

серия книг

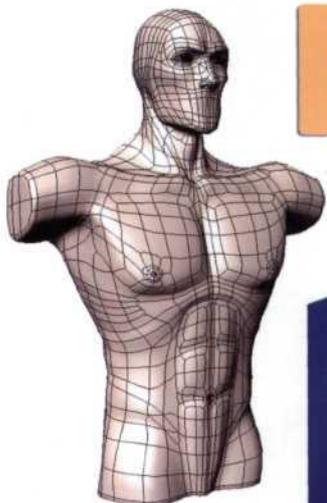
АРТ-ДИЗАЙН НА ПЕРСОНАЛЬНОМ КОМПЬЮТЕРЕ

**Autodesk**  
Media and Entertainment

ВСЕ, ЧТО ВЫ ХОТЕЛИ  
ЗНАТЬ, НО БОЯЛИСЬ  
СПРОСИТЬ ●●●●

Теперь Вам  
не потребуются  
десятки и сотни книг  
по 3D Studio Max!

# 3D STUDIO Max 7.0



Неофициальное  
пособие  
по созданию  
трехмерных  
анимационных  
объектов  
на персональном  
компьютере



[www.book-press.ru](http://www.book-press.ru)

# Неофициальное пособие по созданию трехмерных анимационных объектов на персональном компьютере

---

**Ф. М. Мильчин**

## **3D Studio Max 7.0: Все, что Вы хотели знать, но боялись спросить**

Автор (составитель) этой книги не несет ответственности за использование материалов, которые опубликованы в этом издании. Вся информация дана исключительно в образовательных целях. Ни при каких условиях ответственность за какие-либо последствия от использования этой книги в практических целях не может возлагаться на автора.

Москва



Литературное агентство «Бук-Пресс и К»  
2005

УДК 004.5  
ББК 32.973.26-018.2  
М60

Издание выпущено совместно с  
ООО «Издательским домом «Таврия»

**Мильчин, Ф. М.**

**М60** 3D Studio Max 7.0: Все, что Вы хотели знать, но боялись спросить. Неофициальное пособие по созданию трехмерных анимационных объектов на персональном компьютере / Ф. М. Мильчин. - М.: Бук-Пресс и К., 2005. - 368 с. - (Арт-дизайн на персональном компьютере).

ISBN 5-88548-507-9

Современный мир находится в постоянном движении. Все, что было изобретено в прошедшие десятилетия, объединяется сегодня в новые мощные технологии, а то, что еще вчера было примером блестящих передовых решений, становится обыденным. Глобальные перемены коснулись и 3D-графики. Ее возможности трудно переоценить: она вносит не только больший реализм в компьютерное искусство, кино- и видеоиндустрию, но оказывает влияние и на способы нашего общения с машиной. Несмотря на то, что компьютерное проектирование и работа с трехмерными объектами практикуются уже в течение нескольких десятилетий, только в последние два-три года, после появления мощных домашних компьютеров, стали наблюдаться серьезный прогресс и беспрецедентная динамика развития 3D-графики.

3D Studio Max — наиболее распространенное в мире программное обеспечение для 3D-моделирования, анимации и визуализации, включающее высокопроизводительные инструменты, необходимые для создания зрелищных кинофильмов и телевизионных заставок, современных компьютерных игр и презентационных материалов. 3D Studio Max 7.0 — это реализм, выразительность, производительность и гибкость.

Теперь вам не потребуются десятки и сотни книг по 3D Studio Max. Просто начните самостоятельно работать с программой с помощью этой книги! Познакомьтесь с последней версией пакета трехмерной компьютерной графики и анимации 3D Studio Max 7.0, легко и быстро изучите основные приемы работы с программой, освоите базовый набор инструментов, при помощи которого вы сможете создавать как простые, так и достаточно сложные анимационные объекты. В книге также рассмотрены возможности модификации трехмерных объектов, а также редактор материалов и способы визуализации трехмерных сцен.

Для начинающих пользователей персонального компьютера, профессионалов, а также для тех, кто увлекается графикой и ее оживлением.

УДК 004.5  
ББК 32.973.26-018.2

ISBN 5-88548-507-9

© Ф. М. Мильчин, составление, 2005  
© Литературное агентство «Бук-Прес и К.», 2005

# Часть 1.

## 3ds max — седьмая версия

### Глава 1.

#### Основы 3D технологий

Современный мир находится в постоянном движении. Все, что было изобретено в прошедшие десятилетия, объединяется сегодня в новые мощные технологии, а то, что еще вчера было примером блестящих передовых решений, становится обыденным.

Глобальные перемены коснулись и 3D-графики. Ее возможности трудно переоценить: она вносит не только больший реализм в компьютерное искусство, кино- и видеоиндустрию, но оказывает влияние и на способы нашего общения с машиной. Несмотря на то, что компьютерное проектирование и работа с трехмерными объектами практикуются уже в течение нескольких десятилетий, только в последние два-три года, после появления мощных домашних компьютеров, стали наблюдаться серьезный прогресс и беспрецедентная динамика развития 3D-графики.

#### Представление объектов

3D-объект — это объемное тело, у которого есть длина, ширина и глубина — свойства, которых у компьютерных изображений не было в течение многих лет. Ведь в действительности 3D-объекты существуют только в памяти компьютера и отображаются на плоской поверхности экрана по законам перспективы. Трехмерный объект характеризуется своей формой и текстурой поверхности.

Форма — это геометрия объекта, которая в простейшем случае описывается серией взаимосвязанных в трехмерном пространстве точек (вершин) и многоугольников (граней — замкнутых двумерных фигур с тремя или более сторонами). Например, куб имеет восемь вершин и шесть граней. Раньше для простоты в большинстве случаев в качестве базовых многоугольников выбирались треугольники. Несколько сотен или даже тысяч таких треугольников соединялись друг с другом и образовывали сложные трехмерные сетки. Теперь же в трехмерных пакетах все больше используется моделирование гладких поверхностей в форме сплайнов, или так называемое твердотельное моделирование (solid mod-

eling). Причем вычислительные мощности современных компьютеров возросли настолько, что позволяют отображать такие поверхности даже в текстурированном виде практически в реальном времени.

Текстура (лат. *textura*) — это особенности строения твердого вещества, обусловленные характером расположения его составных частей (кристаллов, зерен и т. п.). В компьютерной графике текстура описывает самые разные свойства поверхности объектов: цвет, прозрачность, отражающую способность, шероховатость и т. п. Для текстур обычно используются изображения, имитирующие материал, из которого сделана поверхность моделируемого объекта или так называемые карты поверхности, отражения, рельефа и т. п.

На моделируемой сцене размещаются источники света, при необходимости вводится осветляющий (обычный) или затемняющий (для эмуляции ночных условий) туман и задается точка обзора (камера).

Далее в дело вступает компьютер, которому для отображения 3D-объекта на экране, приходится выполнять весьма точные математические расчеты:

- ◆ сначала производится разбиение поверхности (Tessellation), то есть объект, образованный плоскими или криволинейными поверхностями, приводится, как правило, к сеточному виду, причем обычно поверхность разбивается на простейшие элементы — треугольники. Для криволинейных поверхностей производится аппроксимация с точностью, достаточной для имитации гладкой поверхности;
- ◆ затем выполняются геометрические преобразования (Geometry Transformation) 3D-объекта, то есть перемещения, вращения и масштабирование. Трансформация в математическом смысле — это перемножение координат вершин на матрицу соответствующего преобразования;
- ◆ рассчитывается освещенность сцены (Lighting) в зависимости от количества источников, их расположения и типа: рассеянный свет (Ambient), точечный источник света (Spot), направленный источник света (Omni) и т. п.;
- ◆ затем выполняется отсечение (Clipping), то есть для снижения непроизводительных затрат выявляются те части объекта, которые будут видны с заданной точки обзора. С помощью отсечения из дальнейшего

рассмотрения удаляются части объекта (если они заданы), выходящие за границы экрана или не попадающие в область просчета, расположенную между ближней и дальней плоскостями отсечения, то есть между Near и Far Clipping planes. Другими словами, обычно задается некоторая область в пространстве, по выходе из которой объекты или их части не рассматриваются;

- ◆ далее, на основании данных об освещенности, начинает работать один из алгоритмов затенения (Shading). Чаще всего используются простые методы: плоскостной (Flat), Гуро (Gouraud) и Фонга (Phong). При плоскостном затенении на основании данных об освещенности вершин треугольника рассчитывается среднее значение освещенности всей его поверхности. Этот метод самый быстрый, но при использовании его практически невозможно получить сколько-нибудь реальное изображение гладкой поверхности. При затенении по методу Гуро на основании данных об освещенности вершин интерполируются значения освещенности по ребрам треугольника, а после этого по градиенту рассчитываются значения освещенности для каждой точки ограниченного ими треугольника. Метод дает существенно лучший результат по качеству, но порождает эффект звездчатых бликов. Затенение по Фонгу работает совершенно иначе: нормали, первоначально заданные в вершинах, интерполируются по всей поверхности объекта (причем для внутренних точек можно задать свой закон изменения нормали), имитируя таким образом произвольную кривизну, а уже затем на основании данных о такой «кривизне», расположении источников света и положении наблюдателя для каждой точки поверхности рассчитывается ее освещенность. Последний метод дает один из лучших результатов и широко используется в программах 3D-моделирования;
- ◆ затем, используя информацию о глубине, компьютер проецирует полученную поверхность на плоскость (Projection) для создания на двумерном экране иллюзии глубины (обозначает перспективу);
- ◆ удаляются невидимые поверхности (Hidden Surface Removal), то есть те части объектов, которые закрыты другими элементами или объектами;

- ◆ и, наконец, при помощи более сложных алгоритмов выполняются окончательный просчет и растеризация (Rasterisation) 3D-объекта. На этом этапе компьютер использует информацию о текстурах (их может быть несколько для одной поверхности) с учетом их взаимодействия, то есть производится учет прозрачности, перспективы, уровня и метода детализации. На этом этапе окончательно выясняется, как объект взаимодействует со светом: какого он должен быть цвета, какое количество света должно отражаться от каждой его грани, должны ли возникать какие-либо тени и отражения. На этом этапе применяются алгоритмы трассировки луча (Ray Tracing) или излучательности (Radiosity) — самые сложные и качественные методы построения реалистичных изображений. Наиболее распространен алгоритм обратной трассировки луча: от наблюдателя через точку строящегося изображения проводят луч и, учитывая все его отражения от объектов, вычисляют цвет результирующего пиксела. Излучательность — это алгоритм просчета, при котором фотореалистичные изображения создаются путем просчета взаимных отражений света между объектами. Без этого часто кажется, что в трехмерной сцене чего-то не хватает. Классический пример: отсветы лавы на стенах кратера вулкана или блики на лицах от костра. Однако при просчете этим методом теряется такой визуальный компонент, как отраженная энергия, порожденная отражениями и световыми бликами. Поэтому иногда применяют метод гибридной излучательности, который выполняет просчет как рассеивающегося света (по алгоритму излучательности), так и отраженного (по алгоритму трассировки луча), и в результате получаются сцены, демонстрирующие оба этих аспекта. Некоторые эффекты добавляются интерактивно уже на этапе обработки изображения и мгновенно появляются при визуализации поверх уже готовой картинкой;
- ◆ окончательная обработка (Post processing) — самая последняя стадия работы с уже готовым изображением. На этом этапе можно сделать, например, сглаживание границ объектов после их растеризации (Antialiasing), в частном случае это удаление так называемого лестничного эффекта, проявляющегося на наклонных линиях. Для чего изображение, например, изначально рассчитывается в большем разрешении, чем выводимое на экран, и

уменьшается с вычислением среднего значения цвета для матрицы из нескольких рядов стоящих пикселей исходного изображения.

## Моделирование

По мере развития 3D-индустрии проблема моделирования встает особенно остро. И хотя многие разработчики достаточно опытные, чтобы моделировать сложные конструкции самим, не у всех есть время для детального исследования, необходимого при такой работе.

Выходят из положения по-разному: обращаются к художнику с просьбой тщательно и аккуратно выстроить модель по фотографиям или в фирму типа Viewpoint DataLabs, которая за \$300-1000 продаст уже готовую полигональную или NURBS-модель нужного объекта или сделает ее на заказ. Такая модель состоит из нескольких десятков тысяч многоугольников и поставляется в нескольких форматах, включая форматы наиболее популярных 3D-пакетов. В придачу к ней за небольшую дополнительную плату дадут и полный набор редактируемых текстур, что еще более ускорит работу.

Масштабные 3D-модели, как, например, эту, можно строить разными способами: по фотографиям, по чертежам или непосредственно отсканировав макет.

Полигональная (то есть каркасная) модель — это ядро любого 3D-проекта, по сути, основная несущая конструкция, на которую накладываются поверхностные текстуры, образующие достоверный, близкий к реальному миру объект. Такие модели начали разрабатывать еще в 1960-х годах и применять в системах автоматизированного проектирования. Традиционно они строятся из полигональных сеток. Количество многоугольников в таких сетках соответствует количеству деталей объекта, и чем более детальными становятся модели, тем большее количество многоугольников они насчитывают. Этот процесс имеет свою обратную сторону: чем больше многоугольников входит в модель, тем большие мощности нужны для работы с нею. Эту особенность необходимо учитывать, планируя изготовление той или иной модели.

Относительно новая область 3D-моделирования — NURBS-модели. NURBS (Non-Uniform Rational B-Splines — неоднородные рациональные B-сплайны) — это математически определяемые кривые, или сплайны. Криволинейные поверхности, построенные из таких сплайнов, получаются плавными и гладкими, в отличие от традиционных полигональных сеток, которым все труднее соперничать с реальностью. В настоящее время такие модели составляют лишь небольшую долю от общего числа используемых в работе, но, вероятно, с ростом мастерства

аниматоров и художников их количество будет расти, особенно после того, как в таких пакетах, как 3D Studio MAX 2, начали предлагать NURBS-инструментарий в дешевом секторе рынка. Особую ценность в данной ситуации приобретают усилия, направленные на ускорение просчета таких моделей, поскольку, как и прежде, важно, чтобы они были сконструированы с минимальным количеством наложений, срезов и т. п.

Методы, используемые при построении моделей, варьируются не только от компании к компании, но и от проекта к проекту. Воссоздать динозавра или построить мотоцикл «Харлей-Дэвидсон» — это очень разные задачи. Причем в самом процессе моделирования большинство компаний используют разные 3D-пакеты. На первом месте стоит задача построить достоверную поверхность. При этом по-прежнему широко используются фотографические материалы, планы, чертежи и рисунки (особенно в небольших компаниях), хотя все большее распространение получают различные методы оцифровки объектов.

Три неотъемлемых составных части трехмерного объекта. На первом этапе объект представляет собой набор вершин (у куба их восемь), соединенных многоугольниками. На втором этапе в изображении появляется перспектива, а на третьем — становится видна текстура поверхности с деталями

Многие специалисты для оцифровки своих объектов используют, например, сканер 8ft FaqArm — более совершенный вариант оригинального 3D-сканера, созданного в начале 1980-х годов. По сути, это контактный шуп, который при помощи нескольких потенциометров, установленных на складной арматуре с шарнирными соединениями, считывает информацию о том, в каком месте находится головка, и преобразует эту информацию в координаты X, Y и Z в трехмерном пространстве. Достаточно сделать необходимое количество замеров, и сетка готова.

Устройство это довольно сложное по сравнению с примитивными первоначальными считывателями. В сканере применена система противовесов; он автоматически учитывает изменения температуры и компенсирует соответствующие расширения и сжатия материалов. Это портативное устройство может работать с объектами вписывающимися в сферу диаметром до 3,65 м и имеет точность до 0,3 мм.

Система MicroScribe 3D компании Immersion Corporation позволяет построить трехмерную модель оцифровав объект вручную по специальной нанесенной на объект сетке

Существует аналогичная, но более дешевая, технология. Компания Immersion Corporation, специализирующаяся на 3D-сканерах и яв-

ляющаяся лидером в области периферийных игровых устройств с обратной связью, представляет 3D-сканеры нижнего уровня. Ее MicroScribe 3D имеет рабочую сферу диаметром всего 1,27 м, зато стоит \$3000.

Одно из основных преимуществ таких систем — высочайшая степень контроля за процессом оцифровки со стороны оператора-модельщика. Как и в случае применения технологии оцифровки движения (motion capture), после оцифровки неподвижной модели неизбежно требуется довольно трудоемкая дополнительная обработка, в ходе которой данные сначала нужно вычистить, а затем привести в пригодную для работы форму. Однако, тщательно спланировав исходную сетку, многие из этих трудностей можно избежать. Сначала на исходный объект карандашом наносятся контрольные линии; затем оператор-модельщик решает, где к сетке нужно добавить деталей, а где следовать естественным изгибам — все это в зависимости от назначения модели. Цель всех этих манипуляций — добиться того, чтобы окончательная сетка была как можно более рациональной.

В идеале нужно оцифровывать непосредственно тот объект, модель которого строится. Когда это невозможно, часто вместо него используется физический макет.

Конечно, самая передовая технология захвата статичных 3D-данных — это технология лазерного сканирования. Побудительным толчком к началу исследований в области оптических технологий послужили низкая скорость и недостаточная точность контактных датчиков. Из трех основных направлений, по которым могла бы развиваться эта технология (сканирование по точкам, зонам и полосам), лучшие результаты за самое короткое время показала технология сканирования по полосам.

Ее суть заключается в том, что на поверхность модели проецируется световая полоса, и ее положение записывается внешними видеокамерами. Таким образом, по мере сканирования модели от одного края до другого, выстраивается точный образ ее поверхности. Проблема в том, что необходимо обрабатывать при этом огромный объем генерируемых данных. Чтобы полностью отснять 3D-объект, необходимо сделать несколько сканов. Сведение этих сканов в единую непрерывную модель, удаление дублирующих данных и отсеивание ошибок (которые неизбежно возникают, когда обрабатывается громадное количество точек, выражающееся семизначными числами) — задача, конечно, непростая.

Лос-анджелесская компания Cyberware стала первой фирмой, выпустившей систему такого рода на рынок в конце 1980-х годов. Вслед за сканером с прагматичным названием Head Scanner (в буквальном переводе «головной сканер» или «сканер головы») в начале 1990-х годов был выпущен Whole Body Scanner — «сканер всего тела». Технология оказа-

лась довольно популярной, и в течение нескольких лет появилось довольно много производителей подобных устройств. В конце концов они эволюционировали настолько, что стало возможным сканирование не только формы поверхности, но и цвета (текстуры) за один проход. Работают эти устройства чрезвычайно быстро и точно (например, сканирование головы занимает не более 15 секунд, и за это время генерируется около 250 тысяч опорных точек), но по-прежнему остаются очень дорогими. Скажем, WB4 Body Scanner компании Cyberware обойдется в 410 тысяч долларов, а расценки на сканирование в специализированных бюро начинаются от двух тысяч долларов за час.

Эта технология имеет и другие недостатки. Возникают проблемы с отражающими поверхностями (что неудивительно), а также с выступами, расщелинами и другими участками, где трудно отслеживать световую полосу. Кроме того, поскольку процесс полностью автоматизирован, оператор-модельщик оказывается выключенным из рабочего цикла, то есть он не может приступить к работе до тех пор, пока не закончится сканирование. В результате получаются слишком громоздкие сетки, для которых требуется большой объем дополнительной обработки.

Однако на рынке представлены и более дешевые сканеры, нижний уровень цен которых составляет примерно 20 тысяч долларов. Специалисты большинства компаний убеждены, что эти устройства достойны того, чтобы работать над снижением их цен. Но даже если лазерные сканеры или контактные датчики действительно когда-нибудь выйдут на массовый потребительский рынок, перед пользователем все равно будет стоять все тот же вопрос: «Строить или покупать?», то есть или тратить несколько недель на то, чтобы найти или вылепить модель, отсканировать ее, а затем вычистить полученные данные и упаковать их в приличную 3D-модель, или подобрать в каталоге готовую? Судя по всему, расширение библиотек каркасных моделей продолжится и в дальнейшем.

## Визуализация

Отображение трехмерных объектов — очень трудоемкая задача, даже если речь идет только о визуализации. Если же объект необходимо еще и анимировать, компьютеру приходится выполнять такие вычисления несколько десятков раз в секунду. А если объект не один, да еще с многочисленными источниками света, дымками, туманностями и похожими эффектами? Ведь подобные сцены сегодня встречаются почти в каждой компьютерной игре.

Основная часть функций отображения трехмерных объектов на плоском экране при моделировании встроена сегодня прямо в операционную систему, чтобы программистам не приходилось писать их само-

стоятельно. Это делается посредством так называемых API (Application Programming Interface) — обширных библиотек функций, упрощающих работу с 3D-объектами и их перерисовку на экране. На PC родным API является Direct3D, который (как часть DirectX) поставляется в комплекте с операционной системой Windows 95/98. На компьютерах Mac эта роль отводится QuickDraw 3D.

Однако, помимо этого, существуют и сотни других 3D API, ориентированных на выполнение различных задач: от сложного 3D-моделирования до простой 3D-игры. Для трехмерных игр важнейшим фактором является скорость: чтобы игра была интересной, необходимо перерисовывать объекты на экране как можно быстрее. В задачах трехмерного моделирования, напротив, важно чтобы объекты перерисовывались с максимальной точностью. Поэтому в таких программах окончательный просчет запускается редко — это очень длительная процедура, а все построения приходится делать под управлением все того же API.

Для ускорения отображения 3D-графики на экране изображения перерисовываются в память на виртуальные экраны (кодовые буферы), аналогично снимкам экранов в памяти компьютера. В одном из буферов может быть записана базовая форма объекта, в другом — наложенная на него текстура, в остальных — освещение, прозрачность, альфа-канал (проще говоря, маска) и другие эффекты. Таким образом, перед выдачей объекта на экран компьютер должен за один проход скомпилировать все необходимые графические эффекты и сгенерировать изображение. Обратная сторона такого ускорения — колоссальные объемы необходимой оперативной памяти, причем, чем выше разрешение экрана, тем больше памяти требуется на такие буферы.

Окончательный просчет и построение общей композиции сцены отнимают много времени, поэтому большую часть элементов этой картинки лучше купить в готовом виде в банке моделей — в результате сроки производства радикально сократятся.

Однако для окончательной визуализации изображения применяются сложные программные решения практически без использования аппаратуры. Наиболее известными пакетами для фотореалистичного, программируемого и интерактивного просчета являются Mental Ray компании Mental Images и RenderMan компании Pixar. Mental Ray имеет продвинутые решения для просчета с использованием алгоритма трассировки луча (как известно, RenderMan трассировку лучей не использует). Компания Mental Images производит, по общему признанию, лучший алгоритм просчета по алгоритму трассировки луча, который остается в то же время и самым быстрым, а также полностью распределяемым по сети. Только RenderMan дает лучшее качество, чем MentalRay, но для этого

RenderMan нужно тщательно программировать. Оба пакета имеют свои собственные библиотеки алгоритмов визуализации и процедурных текстур, а научившись писать свои собственные алгоритмы (шейдеры), можно повысить коэффициент полезного действия этих программ.

RenderMan, в свою очередь, является промышленным стандартом для алгоритмов просчета и используется практически во всех проектах, связанных с киноиндустрией.

### Аппаратура

В последние годы во все графические карты стали встраиваться 3D-акселераторы, которые берут на себя большую часть работы центрального процессора. На таких картах обычно устанавливается очень быстрый специализированный процессор, дополнительный чип — 2D RAMDAC (цифроаналоговый преобразователь) и комплект очень быстрой памяти. Сегодня типичная карта имеет 8 МБ памяти, а на некоторых есть по 32 МБ и даже больше.

Поначалу 3D-акселераторы использовали 3D API, «родные» для тех платформ, для которых они разрабатывались: на PC использовались акселераторы DirectX, на Mac — QuickDraw 3D. Однако быстроразвивающаяся индустрия компьютерных игр породила массу новых API и соответствующих чипов-акселераторов. Некоторые из них со временем исчезли с рынка, другие до сих пор процветают. Производители аппаратных средств (таких, как 3Dfx) разработали набор графических микросхем Voodoo (впоследствии ему на смену пришел Voodoo2). Благодаря своей чудовищной скорости и относительно простому для программирования API под названием Glaid, этот набор графических микросхем завоевал чрезвычайную популярность как среди любителей компьютерных игр, так и среди разработчиков и производителей графических карт.

Однако для 3D-моделирования он не подходит, так как может об- считывать изображение только на полный экран, а не в окно. Кроме того, его возможности ограничены всего лишь 16-разрядным цветом, сравнительно небольшими текстурными картами и несложными моделями (для игр всего этого вполне достаточно, поскольку здесь важнее всего скорость).

К счастью, в 1992 году компания Silicon Graphics разработала потенциальную систему и для профессиональных проблем — это язык описания графических данных OpenGL. Он был разработан в виде 3D API для приложений высокого уровня. Помимо возможности отображать объекты с текстурными картами высокого разрешения, он позволяет точно просчитывать освещение сцен и вводить в них туман, OpenGL обладает еще и кроссплатформенной совместимостью. Кроме того, зало-

женные в нем процедуры обладают достаточной скоростью для того, чтобы их можно было использовать и в играх. Конечно, OpenGL не обеспечивает такой производительности, как 3Dfx Glaid, но небольшие потери в скорости компенсируются той легкостью, с которой этот API можно переносить с одной платформы на другую и даже с одного графического акселератора на другой, так как большинство из них при наличии соответствующих драйверов работают с OpenGL.

Обычно, если 3D-акселератор имеет только аппаратный блок растеризации (как правило, с текстурированием), он выполняет только последние стадии просчета. Если имеется так называемое ядро подготовки треугольников (Triangle Setup Engine), то ускоритель выполняет при обработке изображения еще и триангуляцию. Если у него есть еще и специализированный геометрический процессор (Geometry Engine), то он берет на себя все стадии обработки изображения.

Сейчас принято выделять в 3D-акселераторе несколько основных частей, характеристики которых дают базовое представление о 3D-акселераторе. Это блок геометрии, блок растеризации и память, включающая кадровый буфер (Frame buffer), буфер глубины (Z-буфер), буфер шаблонов и специальную текстурную память (Texture memory).

### Выбор акселератора

Если вы хотите подобрать себе 3D-акселератор, нужно заранее ознакомиться с профессиональной терминологией. Сложный мир трехмерной графики породил массу терминов: Z-буфер, билинейная и трилинейная фильтрация и др. К счастью, и здесь есть общий знаменатель — Frame Rate, то есть скорость смены кадров — количество кадров, которые карта может обрабатывать за одну секунду. Этот параметр зависит от того, насколько разнородные вещи отображаются на экране, и от сложности моделей, однако многие производители называют средние скорости, характерные для игр, просто в качестве отправной точки.

Если у вас новая модель PC с портом AGP (Accelerated Graphics Port), то вам нужна карта именно такого типа, поскольку она сможет взаимодействовать с процессором и оперативной памятью вашего компьютера гораздо быстрее, чем карта с интерфейсом PCI (Peripheral Component Interconnect). Следовательно, повысится скорость отображения графики на вашем дисплее. Помните, что 3D-программы страшно прожорливы: им нужна большая оперативная память, поэтому покупать необходимо карты с максимальным объемом оперативной памяти. Для профессионального 3D-моделирования и просчета можно было бы рекомендовать иметь на карте память 32 МБ или больше. Кроме того, необходимо, чтобы карта обеспечивала работу OpenGL, и чтобы к

ней прилагался набор приличных драйверов низкого уровня, поскольку сегодня в профессиональном трехмерном мире именно этот API де-факто становится стандартом. В большинстве 3D-приложений и даже в некоторых играх в самом ближайшем будущем будет, скорее всего, использоваться именно этот API.

В последние годы 3D-акселератор (карта или процессор) стал такой же важной деталью компьютера, как и центральный процессор. Без него отображение трехмерных объектов и работа с ними проходит очень медленно.

Основными способами, применяемыми для увеличения мощности 3D-ускорителей, являются: повышение скорости работы графического процессора и памяти (повышение тактовой частоты), увеличение пропускной способности интерфейса памяти, добавление дополнительных модулей обработки 3D-графики и, наконец, применение геометрических ускорителей.

Увеличение скорости работы графического процессора за счет обыкновенного повышения тактовой частоты является самым простым и, пожалуй, самым дешевым способом увеличения скорости работы 3D-ускорителя, однако этот путь сопряжен с проблемами технологического плана. Повышение частоты работы неизбежно увеличивает энергопотребление, соответственно и нагрев микросхемы. Принимая во внимание, что сложность микросхем растет очень быстрыми темпами и число транзисторов, соответственно и площадь самих микросхем, неуклонно возрастают, нельзя не увидеть, что эта проблема встает сегодня с особой остротой, и уменьшение ее влияния возможно только с переходом на 0,25 мкм технологию и далее.

Повышение скорости работы памяти имеет значение в том случае, когда скорость обработки графики намного превышает скорость передачи необходимых для обработки данных. Учитывая глубокую конвейеризацию и детерминированность всех процессов в 3D-графике, это достаточно эффективный путь, особенно при наличии нескольких обрабатывающих модулей, имеющих общую шину передачи данных. К тому же снимается проблема, связанная с наличием быстрого RAMDAC, которому нужен постоянный доступ к памяти.

Увеличение числа модулей обработки 3D-графики в настоящее время идет двумя путями. Первый связан с увеличением числа модулей, отвечающих за наложение текстур (обычно их используется два). Это помогает при мультитекстурировании, когда каждый модуль накладывает одну из нескольких текстур на объект. Другой путь — это простое наращивание числа самих 3D-ускорителей и/или геометрических процессоров. Компания Metabyte сообщила в начале года о разработке технологии

PGC, позволяющей ставить практически любые 3D-акселераторы параллельно, чтобы каждый ускоритель обрабатывал только свою часть изображения.

Установка на ускоритель геометрического процессора позволяет снизить нагрузку на центральный процессор в части геометрической обработки сцены и достичь более высоких скоростей обработки графики. В этом случае центральный процессор практически не участвует в графической обработке, и общая производительность определяется только скоростью самого 3D-ускорителя. По заявлениям разработчиков, производительность их геометрических процессоров на порядки превышает производительность центрального процессора на специализированных графических операциях.

Можно высказать предположение, что в недалеком будущем появится самое разнообразное и исключительное по мощности аппаратное и программное обеспечение для 3D-графики, а цены на него упадут. Ожидается массовый выпуск недорогих 3D-акселераторов, оборудованных геометрическим процессором, которые будут поддерживать и большие текстуры, и компрессию, и аппаратный (32-разрядный) просчет. Однако не стоит расслабляться. В таких условиях особое значение приобретают обучение и постоянное совершенствование технологий применения нового оборудования и программ. При этом требования к профессионализму аниматора не только не снижаются, а наоборот, возрастают. Как говорил Джордж Лукас: «Работу над кинофильмом никогда нельзя закончить, ее можно только прекратить».

## Fahrenheit

Компании SGI и Microsoft работают над новым 3D API (Application Programming Interface) под кодовым названием Fahrenheit. Цель данного совместного проекта — соединить переработанные API OpenGL и

Direct 3D в единую кроссплатформенную 3D-архитектуру, которая позволит разработчикам программного обеспечения писать программы для операционных систем Windows и Unix. Идея состоит в том, чтобы использовать развитые 3D-функции OpenGL и доступность Direct 3D.

Создания API Fahrenheit приведет к тому, что сторонним фирмам станет гораздо проще разрабатывать программное обеспечение и аппаратное обеспечение, а у пользователей появится единое решение, как для игр, так и для профессиональных приложений, таких, как CAD/CAM и 3D-моделирование. Будущие версии драйверов для уже существующих наборов микросхем смогут извлекать выгоду из этого проекта, вызывая подпрограммы непосредственно из операционной системы. В результате

карты-акселераторы станут дешевыми, а программное обеспечение более совместимым. Поставки разработчикам различных компонентов программного обеспечения в рамках проекта Fahrenheit начнутся в следующем году, а приложения на его основе появятся к концу 2000 года (хотя никакой официальной даты пока назначено не было). Fahrenheit вряд ли потеснит Voodoo на рынке компьютерных игр, но для профессионалов, то есть для вас, он будет иметь огромное значение.

## Глава 2.

### Новая версия программного обеспечения для профессиональной трехмерной анимации и моделирования

3ds max — лидер рынка программного обеспечения для трехмерного моделирования, анимации и визуализации. Новая версия 3D Studio MAX 7.0 предлагает еще больше усовершенствованных возможностей, которые удовлетворяют возрастающим нуждам создателей трехмерных моделей и аниматоров, а также разработчикам интерактивных игр. Открытая архитектура 3D Studio MAX позволяет аниматорам воспользоваться преимуществами использования более чем ста подключаемых приложений, чтобы быстро и легко добавлять впечатляющие эффекты. Более того, с помощью 3D Studio MAX SDK они могут даже создавать свои собственные приложения-модули, чтобы придавать оригинальный вид своим творениям.

3ds max — наиболее распространенное в мире программное обеспечение для 3D-моделирования, анимации и визуализации, включающее высокопроизводительные инструменты, необходимые для создания зрелищных кинофильмов и телевизионных заставок, современных компьютерных игр и презентационных материалов. Новый 3ds max 7.0 — это реализм, выразительность, производительность и гибкость. При создании 3D Studio Max 7.0 программисты Discreet старались наиболее полно учесть потребности разработчиков компьютерных игр — значительной аудитории пользователей этого продукта. В этом им помогли многие известные разработчики игр, например, Sega. По данным Discreet, пакет 3D Studio Max используют 80% разработчиков, связанных с игровой индустрией. Из новинок, созданных с помощью этого ПО, можно назвать Grand Theft Auto, Splinter Cell, Knight of the Old Republic и Neverwinter Nights.

Новый 3D Studio Max получил функции управления слоями, расширенный набор команд для управления через консоль (в том числе команды для рендринга), полностью переработанный режим упрощенного обзора для работы с большими и сложными сценами, поддержку динамических шейдеров, распределенных вычислений и формата HDR (High Dynamic Range Image) для улучшения работы с освещением и отражениями.

Интересным новшеством является режим Vehicle Dynamics, который позволяет обращаться с ним как с простым объектом после задания основных параметров колесного транспортного средства. В состав пакета также вошла дополнительная программа для рендеринга Mental Ray 3.2, лицензированная у Mental Images.

Как показала практика, 80% пользователей 3D Studio Max — это разработчики видеоигр. Именно поэтому к разработкам 6-й версии продукта была привлечена одна из крупнейших компаний по производству компьютерных игр — Sega.

В новой версии 3D Studio Max 7.0 реализован механизм управления слоями. Также был расширен набор консольных команд. К примеру, теперь можно управлять с консоли рендерингом.

Разработчиков компьютерных игр порадует поддержка динамических шейдеров, а также новый режим, который значительно увеличивает производительность при работе со сложными и большими сценами. Улучшены и алгоритмы обработки освещения и отражения.

Профессиональные дизайнеры теперь также смогут использовать в работе с 3D Studio Max 6 проекты, созданные в продуктах компании Autodesk, Inc., к примеру Architectural Desktop (ADT) 2004. Также в новую версию 3D Studio Max будут добавлены собственные средства для работы с архитектурными элементами сцен из различных материалов.

### **Требования в программному обеспечению**

- ◆ XP Professional (SP1), Windows 2000 (SP4) или XP Home (SP1)
- ◆ Internet Explorer 6
- ◆ DirectX 9 рекомендуется (DirectX 8.1 минимум)
- ◆ OpenGL

### Требования к «железу»

- ◆ Intel PIII или более поздний, или AMD 300Mhz минимум (Dual Intel Xeon или dual AMD Athlon рекомендовано)
- ◆ 512 MB RAM и 500 MB места на жестком диске минимум, (1GB RAM и 2GB дискового пространства рекомендовано)
- ◆ графическая поддержка карты 1024x768 цветов на 16 бит с RAM 64 МБ. (OpenGL и Direct3D поддерживающее ускорение аппаратных средств ПК; 3D графический акселератор 1280x1024 цвета на 24 бита с RAM 256 МБ (предпочтительно))
- ◆ Windows-compliant pointing device (оптимизация для Microsoft Intellimouse ).
- ◆ Wacom Intuos или ему подобное
- ◆ CD-ROM drive
- ◆ Дополнительно: звуковая карта и спикеры; кабель для TCP/IPcompliant network; 3D hardware graphics acceleration;
- ◆ video input and output devices;
- ◆ джойстик;
- ◆ инструменты midi;
- ◆ мышь с 3 кнопками

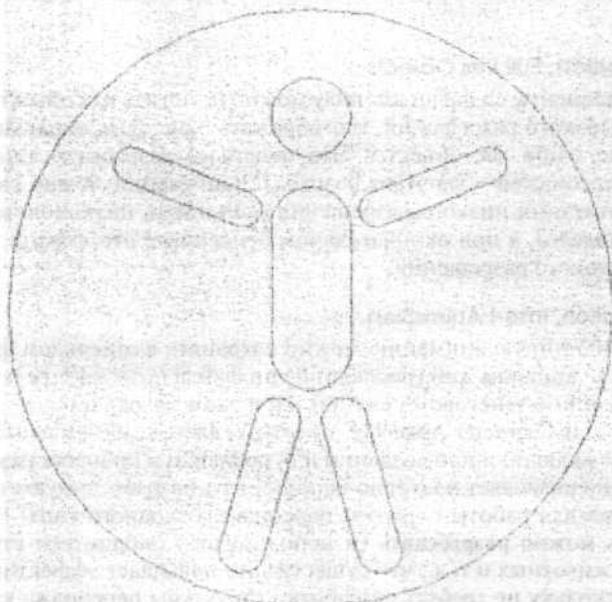
## Часть 2.

# Моделирование

### Глава 1.

## Конструкция персонажа

**Character Assembly** — инструмент управления структурой персонажа как единым целым. Представляет собой специфический тип группы, объединяющей элементы оснастки персонажа — кости (скелетон), оболочку (кожу), цепочки инверсной кинематики и анимацию (контроллеры, хелперы). После создания **Character Assembly** обозначается значком особого вида, расположенным у ног персонажа:



Фактически **Character Assembly** ничего нового не создает, это всего лишь операция группирования сформированных ранее (и другими инструментами) элементов структуры персонажа. Зато после создания **character assembly**, для работы с ним (через панель **modify**) доступны замечательные новые средства.

### Set as Skin Pose

Задание позы скелетона в видовом окне в качестве позы для связывания костей с кожей (полигонным каркасом модели) модификатором **Skin**. Может применяться повторно, при этом все параметры связывания каркаса и костей пересчитываются.

### Assume Skin Pose

Заставляет скелетон принять позу, заданную как **Skin Pose**.

### Skin Pose Mode

Режим редактирования позы скелетона, принятой в качестве **Skin Pose**. Может применяться в любом месте разработки анимации, при этом текущая поза меняется на **Skin Pose**, а после завершения настройки (отжатия кнопки **Skin Pose Mode**), скелетон возвращается в исходное положение.

### Low Res Objects, Full Res Objects

Все элементы **character assembly** можно пометить как объекты низкого или высокого разрешения, и отображать в видовых окнах либо те, либо другие, либо все объекты. Это облегчает разработку анимации сложных персонажей — на этапе разработки анимации можно работать только с объектами низкого разрешения (например, низкополигонным каркасом модели), а при окончательном рендеринге отображать только объекты высокого разрешения.

### Save Animation, Insert Animation

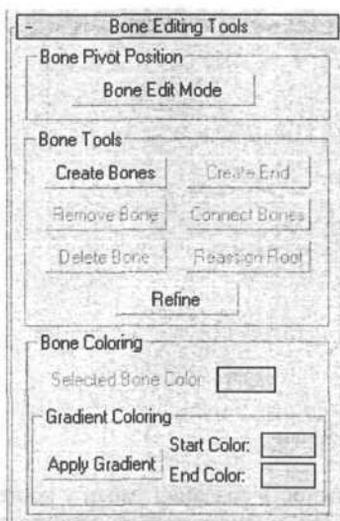
Разработанную анимацию можно сохранять в отдельном файле с разрешением **.anm** или **.xml** (последний тип файла позволяет редактирование анимации в текстовом режиме). При этом сохраняется не только анимация, но и **Character Assembly**, то есть, скелетон, кинематика и все, что мы в нее включили при создании или добавили в процессе разработки. Эта опция позволяет повторно использовать разработанную структуру персонажа для работы с другим персонажем сходного типа. Например, теперь можно разработать (и использовать) библиотеки структур для людей, животных и т. д., что существенно повышает эффективность работы, поскольку не требует разработки структуры персонажа каждый раз с «нуля». Кроме того, повторно можно использовать не только струк-

туру персонажа, но сохраненную анимацию — ее можно вставлять для анимации другого персонажа или сливать с созданной ранее.

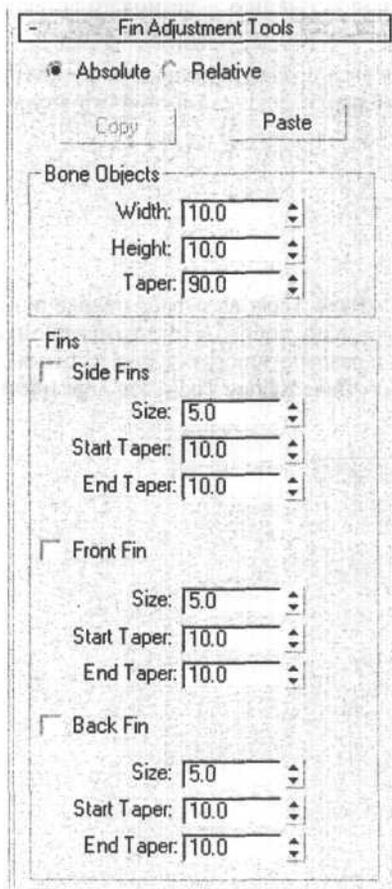
**Character Assembly** может быть создан не только на основе структуры персонажа, но и на любой иерархически связанной структуре других объектов.

## Глава 2. Bone Tools

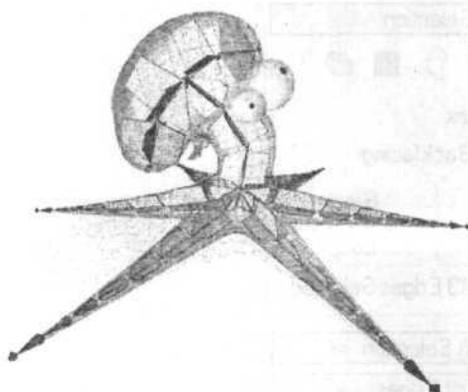
Выбор **Character menu** ⇨ **Bone Tools** вызывает появление нового выпадающего меню, где в одном месте собраны инструменты, наиболее типичные для создания костей и разработки структур на их основе. В меню представлены три закладки — **Bone Editing Tools**, **Fin Adjustment Tools**, **Object Properties**.



**Bone Editing Tools** — позволяет создавать, удалять, вставлять и редактировать кости, при включенной кнопке **Bone Edit Mode** изменение длины кости вызывает соответствующее смещение связанных костей, так что цепочка не рвется. Для костей можно назначать различный цвет, в том числе и градиентный.



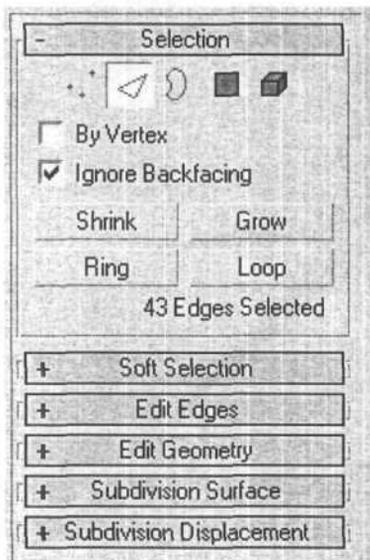
**Fin Adjustment Tools** – возможность создания ребер у костей, управления их размером и формой. Несомненно, сильно облегчает визуальную связь со структурой, в особенности, сложной, и будет полезна при создании связей с кожей. Вкупе с назначением цвета позволяет придавать костным структурам весьма колоритный вид, например, такой:



**Object Properties** – эта группа параметров сменила место жительства. Ранее она находилась на закладке свойств объекта – **Object Properties**.

### Глава 3. Editable Polygon

Для этого инструмента добавлены новые рабочие операции: **Quickslice tool**, **extrusion along a spline** (экструзия вдоль сплайна), **Hinge From Edge** (поворот относительно заданной оси), **constraints** (направляющие), средства увеличения и уменьшения количества выделенных элементов. Значительной переработке подвергся рабочий интерфейс.



Закладок теперь всего шесть. Операции, специфические для каждого подобъекта, вынесены на закладку **Edit Elements** (**Edit Vertex**, **Edit Edges**, **Edit Borders**, **Edit Polygons**, **Edit Elements** – в зависимости от того, с каким подобъектом ведется работа). Операции наиболее универсальные (общие для всех элементов) вынесены на закладку **Edit Geometry**. Кроме того, многие операции обзавелись окном интерактивного предварительного просмотра, удобного для экспериментирования.

Полностью изменилась закладка **Selection**, здесь появились новые средства автоматического расширения/уменьшения количества выделенных элементов – кнопки **Shrink** (уменьшить), **Grow** (расширить), **Loop** (петля), **Ring** (кольцо). Выделяем элемент, например ребро, нажимаем кнопку **Grow**, при этом выделяются все примыкающие к выделенному ребра. **Shrink** работает противоположным образом – снимает выделение. **Loop** и **Ring** также расширяют область выделения, но используют несколько иной механизм, чем **Grow**.

Закладка **Soft Selection** имеет новую кнопку **Shaded Face Toggle**, переключающую режимы выделения с затенением и без затенения. Работает со всеми подобъектами – вершинами, гранями, границами, полигонами и элементами.

Рабочая функциональность **Editable Polygon** значительно усилена за счет введения новых операций. Все множество рабочих операций можно разделить на две группы: универсальные, или общие для всех подобъектов, и специфические.

Универсальные, расположены обычно на закладке **Edit Geometry**:

### QuickSlice

Позволяет быстро разрезать объект без манипуляций с **gizmo**. Выделяем подобъект, нажимаем **QuickSlice**, затем щелкаем в начальной точке резки и в конечной.

### Cut

Резка, позволяет создавать новые грани и полигоны одним щелчком мыши.

### Constraints

Направляющие. Предназначается для использования в работе с подобъектами ограничительных направляющих — ребер или граней.

### Repeat Last

Повторяет последнюю выполненную команду. Работает не со всеми типами трансформаций.

### Hide Unselected

Отключает отображение на экране всех невыделенных подобъектов.

Специфические рабочие операции, находятся как правило на закладке **Edit Element**:

### Для вершин

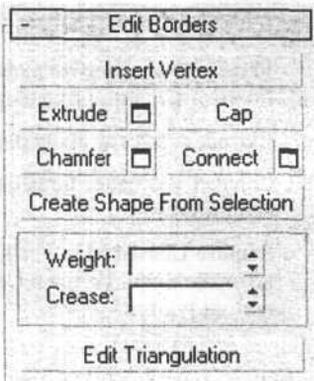
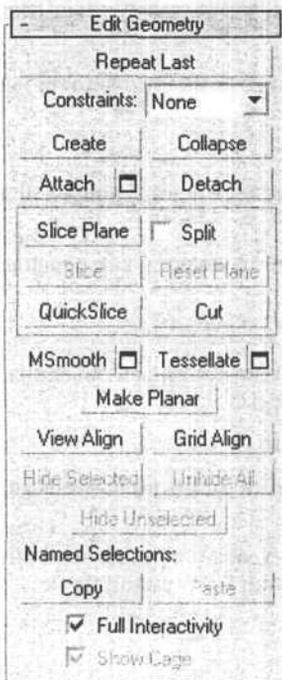
- ◆ **Remove** — удаляет выделенные вершины и объединяет полигоны, использовавшие их;
- ◆ **Extrude** — экструзия вершин;
- ◆ **Connect** — создает новые грани между выделенными вершинами;
- ◆ **Remove Unused Map Verts** — удаляет лишние вершины карт, которые появляются в Unwrap UVW editor, но не используются.

## Для ребер

- ◆ **Remove** – удаляет выделенные грани и объединяет полигоны, использовавшие их;
- ◆ **Extrude** – экструзия вершин с окном предварительного просмотра;
- ◆ **Connect** – создает новые грани между частями выделенных граней;
- ◆ **Detach** – отсоединяет выделенные грани и полигоны им принадлежащие от объекта, создает на их основе совершенно новый объект.

## Для границ

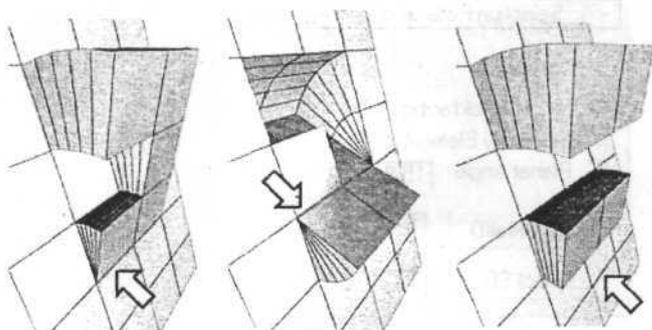
Граница создается вокруг пустот или дыр в поверхности объекта – новый подобъект, поэтому все операции для работы с ним можно считать новыми:



Для полигонов/элементов

- ◆ **Inset** – выполняет операцию **bevel** без смещения полигона по высоте, то есть, новые полигоны лежат в плоскости исходного полигона;
- ◆ **Flip** – изменяет на противоположное направление нормали выделенных полигонов;
- ◆ **Extrude Along Spline** – операция **Extrude** для выделенных полигонов выполняется вдоль выбранного сплайна;
- ◆ **Hinge From Edge** – поворачивает выделенные полигоны относительно выбранной оси – выделенного ребра, имеет окно предварительного просмотра.

Различные результаты работы **Hinge From Edge** в зависимости от выбора оси поворота:



## Глава 4.

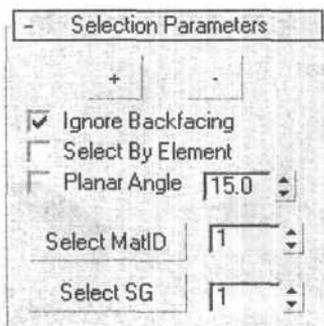
### Модификатор Unwrap UVW

Модификатор радикально переработан. Можно сказать, что фактически мы имеем дело с качественно новым инструментом. Новый модификатор Unwrap UVW теперь предназначен для выделения набора объектов модели (вершин, ребер, граней), создания плоских координатных карт для таких наборов и их редактирования – подстройки координат под накладываемую текстуру – все это не выходя из редактора Unwrap UVW!

Говоря другими словами, у обычной методики создания координатных карт различного типа (прямоугольных, цилиндрических, сфери-

ческих) появилась альтернатива — работать только с плоскими картами (картами, развернутыми на плоскость) на подобъектном уровне. Если пользоваться аналогиями, это напоминает изготовление игрушек обклеивкой кусочками бумаги. С другой стороны, возможность делать выделения на подобъектном уровне и автоматически сохранять их как именованные выделения (**named selection** — новая «фича» пятого макс) не выходя из редактора Unwrap UVW отменяет необходимость делать и хранить такие выделения с помощью стека. Модификатор может работать абсолютно самостоятельно, создавая собственные плоские координаты, или совместно с модификатором UVW Map, обрабатывая созданные им координаты, в том числе не только плоские. Unwrap UVW работает с объектами типа Mesh, Patch, Polygon, HSDS и NURBS.

Соответственно, серьезно переработан рабочий интерфейс модификатора и редактора координат. Прежде всего, появились различные инструменты выделения подобъектов:



Выделения можно расширять/уменьшать кнопками «+», «-»; выделять элементы модели, выделять подобъекты с помощью текстурных каналов или по группам сглаживания.

Редактор координат **Edit UVW** обзавелся строкой меню, новым инструментом **Freeform Mode** и панелью **Options**.

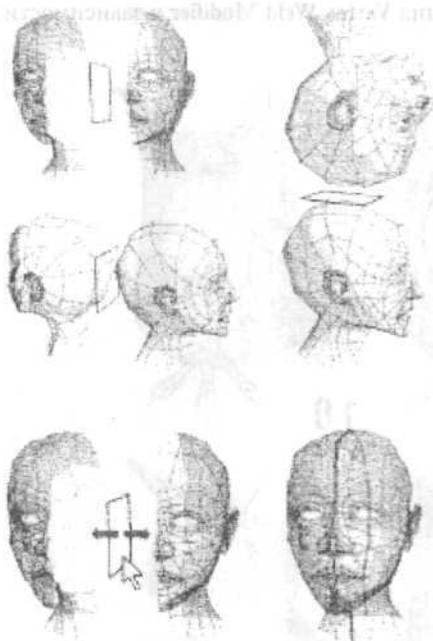
В меню редактора представлены дополнительные инструменты для работы с выделениями; три дополнительных способа построения разворота координатных карт на плоскость и средства для их сшивки; инструмент редактирования координат **Sketch Vertices**.

Создаваемые плоские карты удобно использовать как ссылочные для рисования растровых текстур — для чего подготовленная карта пере-

таскивается через буфер обмена (ALT+PrintScreen), например в Adobe Photoshop.

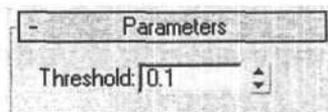
## Глава 5.

### Модификаторы: Symmetry Modifier, Vertex Weld Modifier, Edit Normals Modifier

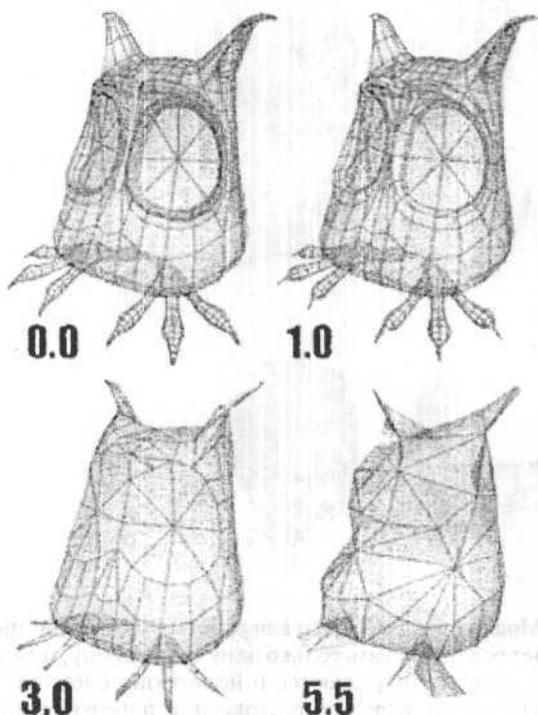


**Symmetry Modifier** предназначен для работы с симметричными моделями и позволяет редактировать только одну ее половину, при этом все изменения интерактивно отображаются и на второй половине. Выполняет три основные задачи: отражение каркасной поверхности относительно плоскостей X, Y и Z; разрезает поверхность и при необходимости удаляет отрезанные части; автоматически соединяет (weld) вершины вдоль общего шва.

**Vertex Weld Modifier** упрощает модель за счет слияния близких или совпадающих вершин. Соединяются все вершины, попадающие в заданный интервал **Threshold**.

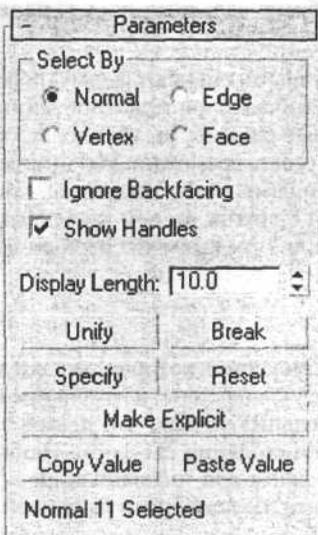


Результат применения **Vertex Weld Modifier** в зависимости от значения параметра **Threshold**:



**Edit Normals Modifier** предназначается для разработчиков игр и позволяет вручную менять направление нормали в каждой вершине каркасной поверхности. Изменение направления нормали воздействует на способ отражения поверхностью света, меняя направление нормали

можно локально управлять отражением. При визуализации стандартным рендером 3ds max этот эффект не визуализируется, а отображается только в видовом окне при использовании видеодрайвера Direct3D.



## Глава 6. Сплайновое моделирование

Кубический сплайн — кусочно-гладкая параметрическая кривая, используется для аппроксимации формы, которая имеет более сложное задание. Существует разнообразные способы представления кубических сплайнов, используемые в прикладных программах графики, каждый из них отличается различными свойствами и характеризуется своей областью использования.

### Bezier-сплайны

Эти сплайны используются и в 2D прикладных программах для векторного рисунка, и в анимации по ключевым кадрам для задания траектории движения. Такие сплайны обычно управляются набором контрольных точек по кривой, с возможностью настройки и управления формой в окрестности этой точки. Управляя наклоном (производной) кривой, проходящей через данную точку, и гладкостью в ее окрестности,

получают необходимое приближение. Обычно изменяется только наклон, и с обеих сторон обеспечивается гладкость. В некоторых случаях, однако, симметрия может быть нарушена, наклон — тот же самый, но вторая производная меняется, или обе производные в точке меняются, и мы получаем излом.

Простота интерфейса, когда сплайны меняют форму, переходя от узла к узлу, обеспечивает большое удобство при моделировании и анимации. Однако существуют некоторые сложности, преодолеть которые при такой форме задания кривой не всегда возможно. Мы иногда можем увидеть, как при движении внезапно изменяется ускорение, поскольку объект переходит на участок кривой с другой второй производной при смене ключевого кадра. Движение при этом выглядит рваным и неестественным.

### **В-сплайны**

Следующий класс кубических сплайнов, которые решают эту проблему. В-сплайны сохраняют тангенс угла наклона, и изменение кривизны является непрерывным в контрольных точках, что делает этот тип кривой оптимальным для анимации и создания 2D-форм. Однако цена этого — отсутствие удобного интерфейса для интерполяции — форма сплайна задана контрольными точками, не лежащими на кривой.

### **NURBS (неоднородный рациональный В-сплайн)**

Термин, про который каждый знает, что это хорошо, но немногие могут объяснить — почему. NURBS — единица «контрольного списка», которую каждая трехмерная программа графики должна иметь по маркетинговым причинам. NURBS — специальный тип В-сплайна, разработанный для использования в трехмерной графике. NURBS — тип В-сплайна и обладает всеми свойствами, которые есть у В-сплайнов, со всеми их преимуществами и недостатками. Он также неоднороден, что полезно для установки ключевых кадров в произвольных точках времени.

Однако трансформации объектов, заданных в такой форме, более сложны и требуют большого количества длительных операций с плавающей точкой, которые не могут быть оптимизированы. Поэтому большинство пакетов компьютерной графики, особенно на PC, избегают NURBS-представления, справедливо считая скорость вычислений важнейшим фактором.

Конечно, NURBS представление более точно отображает многие поверхности, такие, например, как круги и эллипсы. Это делает NURBS более интересным для систем автоматизированного проектирования и

производства прикладных программ механического моделирования. Однако для аниматоров круги, которые могут быть сделаны B-сплайнами, или даже обычными Bezier-сплайнами достаточно близки к реальным. Если вы в большей степени заинтересованы в создании произведения искусства, чем в использовании высокой технологии, можете не вспоминать NURBS.

При применении аппаратного отображения поверхностей графической системой компьютера, может пригодиться другая особенность NURBS: возможность линейных преобразований самих контрольных точек и оптимизация вычислений на уровне их структуры, с последующим аппаратным отображением самой поверхности, и тогда этот способ представления получит большее распространение.

## Глава 7.

### Просчет и фотореализм

Термин «фотореализм» используется, когда нужно описать компьютерное изображение, которое выглядит также реально, как на фотографии. Процесс компьютерной генерации изображения (просчет) или рендеринг, оценивался на заре своего появления исключительно с точки зрения фотореалистичности. Однако с развитием компьютерной графики этот критерий отходит на все более дальний план. Дело даже не только в том, что компьютерная графика все больше переходит в область искусства, а и в том, что основные принципы компьютерных построений, а следовательно, и все приемы и средства, которые позволяют добиваться впечатляющих эффектов, к реализму никакого отношения не имеют, и рождаются подчас исключительно в недрах самой компьютерной графики.

Рассмотрим основные методы визуализации компьютерной модели. Наиболее простым является *метод Гуро*, основанный на определении освещенности грани в ее вершинах с последующей интерполяцией получившихся величин на всю грань. Этот метод не обеспечивает гладкости изменения закраски и имеет много дефектов на изображаемой поверхности, однако является самым оптимальным в вычислительном отношении, поэтому часто используется в предварительных расчетах.

Как и метод Гуро, следующий метод — *Фонга*, при расчетах также опирается на интерполирование закраски, но в отличие от предыдущего, здесь интерполируется значение самого вектора нормали в данной точке, которое затем используется для вычисления цвета пиксела в соответствии с законами физики.

Произвольно меняя вектор «нормали» в точке, мы можем получать из острых ребер — гладкие, а из гладких поверхностей — шершавые. Также здесь можно изменять цвет и свойства поверхностей, используя специальную картинку — «текстуру» или «карту» того или иного свойства материала.

Существуют различные способы наложения текстур или непосредственной генерации в каждой точке по заданной формуле (такие текстуры называются процедурными, в отличие от простых, проективных). Несомненным достоинством моделей Гуро и Фонга является их сравнительная простота. Однако вследствие значительных упрощений получаемый результат не всегда оказывается удовлетворительным, поэтому приходится пользоваться более совершенными методами, например методом трассировки лучей (**ray-tracing**) или методом излучательности (**radiosity**).

При построении изображения методом трассировки лучей посылаются в заданном направлении для оценки приходящей оттуда световой энергии. Эта энергия определяется освещенностью поверхности, встретившейся на пути луча, с учетом преломления, отражения и рассеяния. Поступая таким образом, можно определить освещенность каждой поверхности сцены и построить требуемое изображение.

Однако полный просчет лучей от всех источников, называемый прямой трассировкой, из-за огромного количества вычислений не применяется. Кроме того, большая часть работы на прослеживание лучей, не попавших в глаз наблюдателю и определение освещенности невидимых поверхностей, окажется проведенной впустую. Поэтому на практике реализуется так называемая обратная трассировка, при которой отслеживаются только лучи, попавшие в глаз наблюдателя.

Основными недостатками метода трассировки лучей является неэффективность работы с диффузными поверхностями (которые, в основном, и существуют в природе) и то, что определение освещенности поверхностей проводится параллельно с построением изображения и зависит от положения наблюдателя.

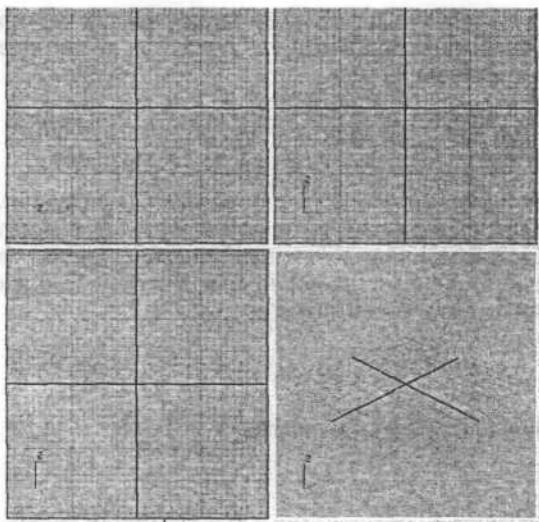
Метод излучательности устраняет эти недостатки, обеспечивая высокую точность при работе с диффузными объектами и отдельное вычисление глобальной освещенности, независимо от положения наблюдателя. В основе метода лежит закон сохранения энергии в замкнутой системе. Все объекты разбиваются на фрагменты, и для этих фрагментов составляются уравнения баланса энергии. Грубо говоря, каждый элемент сцены выступает в роли источника света. Картина в этом случае наиболее достоверная, но вычислительная сложность непомерна даже для специализированных рабочих станций. Поэтому те немногие реализации,

которые существуют сегодня, используются только при просчете одиночных изображений.

И, наконец, делая общие выводы о процессе 3D-визуализации, важно подчеркнуть необходимость качественного алгоритма просчета. И даже не в смысле математики, положенной в его основе, а в смысле его высокопрофессиональной реализации. **Renderman** фирмы Pixar не использует ни алгоритм трассировки лучей, ни, тем более, излучательность, однако считается одним из самых качественных. Именно этот инструмент использовался для расчета почти всех широко известных работ для видео и кино.

## Глава 8. Интерфейс программы

Плавно переходим к работе в 3ds max 7.0. Бегло ознакомимся с оболочкой и попробуем выявить элементы, представляющие для нас наибольший интерес, а заодно и просто попривыкнуть к обстановке, дабы на будущее знать примерно, что где лежит.



Перед нами 4 проекции создаваемого нами мира. Их расположение, тип и количество можно менять. Активируется тот или иной вид кликом на его названии в левом верхнем углу.

Далее видим довольно типичное меню.



То, что нас непосредственно может интересовать, объяснений не требует. Более того, многие его команды продублированы на других панелях.

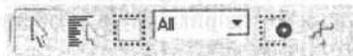
Далее видим набор всевозможных инструментов.



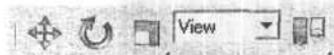
Рассмотрим наиболее важные из них подробнее.



Это классические **Undo** и **Redo**. Распространяются на множество шагов, но существуют и необратимые операции.



Курсор для выбора объектов, меню типов выбора (прямоугольник, эллипс, полигон), фильтр выбираемых объектов, вызов окна списка объектов для выбора по имени.



Эти кнопки представляют собой самые простые операции, которые можно совершить с объектом в пространстве — переместить, покрутить, пропорционально или непропорционально уменьшить или увеличить, опираясь при этом на различные системы координат, с опцией ограничения их одной или двумя осями.



Кнопка вызова редактора материалов.



Основной доступ к рендереру — модулю, занимающемуся построением готовых картинок из сделанной нами геометрии, присвоенных объектам поверхностей и расставленных источников света.



Панель контроля видов/проекций создаваемого мира для перемещения точки зрения наблюдателя, изменения направления и угла обзора, оптимизации вида — не влияют на созданную геометрию (за исключением положения камеры).

Все, что мы до настоящего момента успели рассмотреть, больше напоминает своеобразный рабочий стол, на котором под рукой — в относительном (техническом) беспорядке — всегда самое нужное. Теперь на очереди строгая иерархическая система создания и модификации объектов — «шкаф» с разложенными по полочкам и ящикам заготовками и инструментами. Вот как выглядит верхний уровень этой иерархии:



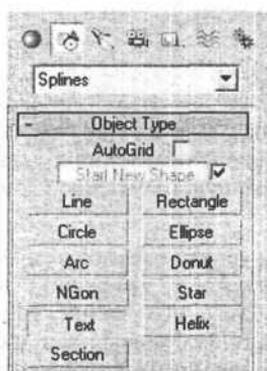
Работа по созданию модели, как правило, начинается с простейших двух- или трехмерных элементов, таких как прямая линия, сплайн, полигон, куб, шар. В результате взаимодействия примитивов по различным схемам и наложения на объекты или их части соответствующих модификаторов возникают более сложные и приближающиеся к конечному идеалу модели.



В меню **Create** мы, в первую очередь, сталкиваемся с возможностью выбора типа элемента, который мы хотим создать: стандартный ли примитив, двухмерную фигуру, а может, источник света, камеру или что-то другое.

## Глава 9. Моделирование текста

Придадим объем некоему тексту. Найти «создатель текста» не составляет большого труда — он именно там, где ему и положено быть, среди двумерных примитивов, в категории **Splines** на панели **Create**. Появившаяся внизу диалоговая панель для ввода текста, особых объяснений не требует, теперь можно выбрать шрифт и содержание.



В первом случае рекомендуется для начала ограничить выбор шрифтами «потолще» и без большого количества острых углов и узких переходов, хотя, в принципе, подойдет что угодно — в нашем случае это

был простой **Arial Bold**. Когда выбор сделан, кликнем левой кнопкой мыши в центр поля **Front**. Вот и наш текст.



Самое время немного поиграть с режимом перемещения, перетаскивая свеже созданный текст в разные стороны, а также с контроллерами видов.

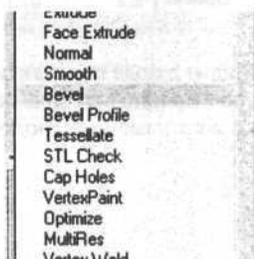


Разобраться, что делает каждый из них, достаточно просто.



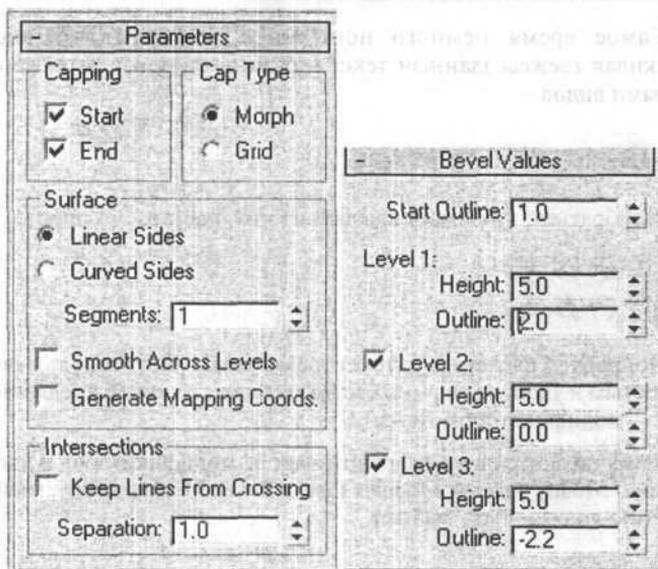
Попробуем сделать наш текст трехмерным. Мы воспользуемся самым простым и удобным из множества методов, который предлагает нам 3ds max — модификатором **Bevel**.

**Bevel** находится в стандартном наборе модификаторов и доступен через меню **Modify panel** ⇨ **Modifier List** ⇨ **Bevel**. Модифицируемый текст естественно должен быть выбран.



Коротко о том, как работает этот модификатор: к контуру, взятому за основу, достраиваются на заданном расстоянии от него дополнительные «этажи» (**levels**) — максимум 3 штуки — из его копий, по необходимости расширенных или суженных (**outline**). Несколько сжимая контур текста при удалении от основы можно получить «мягкую» выпуклую поверхность.

При желании вы можете посвятить некоторое время экспериментам с различными параметрами, однако для ориентации здесь приведена неплохо работающая комбинация (размер текста — 100). Пользоваться параметром **keep lines from crossing** рекомендуется только в крайнем случае, когда иным способом невозможно удержать линии контура от наложения друг на друга и образования артефактов при изменении **outline** — связанный с ним алгоритм требует весьма интенсивных вычислений.



То, что у нас должно получиться, выглядит в поле перспективного обзора (**perspective**), в котором мы, ради исключительно эстетических целей, с помощью контроллеров видов несколько изменили положение точки наблюдения и масштаб, примерно так:



Сейчас вполне можно задаться вопросом, с какой, собственно, стати, объект в перспективном обзоре освещен, и чем? — ведь мы пока не создавали источников света. Дело в том, что для облегчения нам жизни

на первых этапах работы со сценой 3ds max по умолчанию добавляет «параллельную» освещенность без теней. Организацией правильного освещения мы займемся несколько позже, а пока давайте посвятим некоторое время замене глупого шероховатого пластика (примерно так выглядит материал по умолчанию, цвет выбирается случайным образом), из которого сейчас сделан наш текст, на что-нибудь более оригинальное.

## Глава 10. Хромированный металл

По сложившейся традиции обучения первым шагам в 3D используется хромированный металл.

Прежде, чем мы нажмем на кнопку вызова редактора поверхностей, сделаем небольшое теоретическое отступление.

Поверхности объектов в окружающем нас мире характеризуются множеством параметров — они по-разному поглощают, отражают и преломляют свет. Поскольку рендерер стремится добиться максимального реализма при построении изображений, можно быть готовым к тому, что материалы в виртуальной сцене также должны быть снабжены всевозможнейшими характеристиками, отражающими их оптические свойства.

Для начала разберемся с простейшей из них — цветом. В редакторе материалов 3ds max базовых цветов — четыре:

- ◆ **Ambient** — цвет объекта с неосвещенной прямой светом стороны.
- ◆ **Diffuse** — цвет объекта с освещенной белым светом стороны.
- ◆ **Specular** — цвет ореола блика от яркого источника света.
- ◆ **Filter** — «окраска» света, проходящего через отчасти прозрачный материал. Этот пункт, строго говоря, на конечный цвет объекта прямо не влияет.

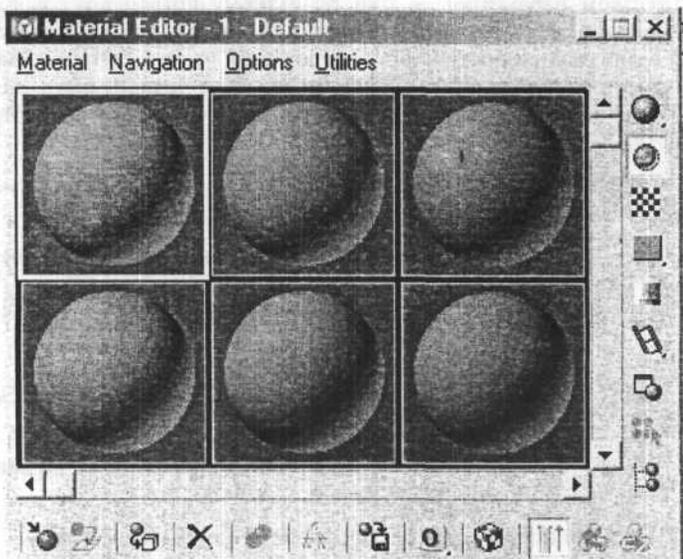
«Блестящность» объекта определяется двумя параметрами:

- ◆ **Shininess** — размер блика.
- ◆ **Shininess Strength** — яркость блика.

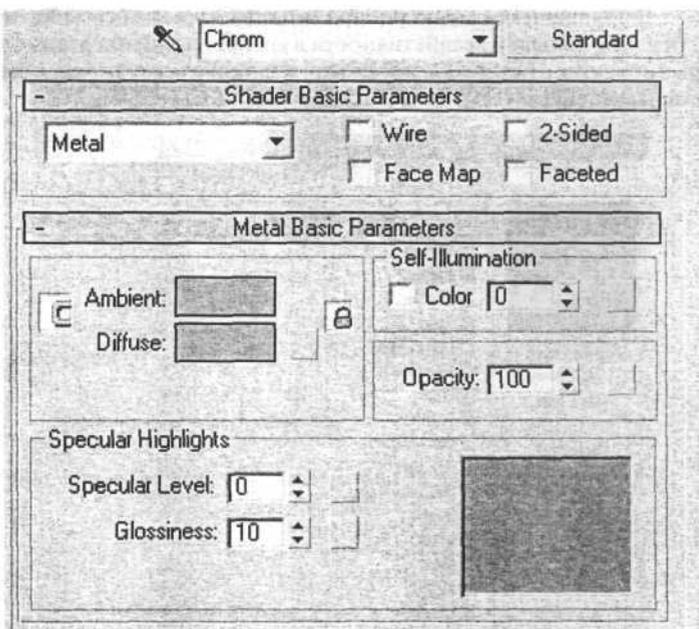
Кроме того, мы можем регулировать прозрачность (точнее, непрозрачность) объекта параметром **Opacity**, и сделать его светящимся, установив **Self-Illumination**.

Изученного нами сейчас достаточно для моделирования поверхности объекта, в реальном мире сделанного из одноцветного материала равномерной матовости/блестящести. Однако поверхности реальных объектов достаточно редко одноцветны и равномерны. Гораздо чаще они текстурированы: на них бывают изображения, неровности, они обладают меняющейся в зависимости от точки объекта блестящестью и прозрачностью. А иногда в них к тому же что-то отражается. Как быть? Редактор материалов предлагает наиболее логичное решение — нанести на объект причитающиеся ему текстуры в виде плоских картинок. Итак, вооруженные приобретенными знаниями, вперед.

В стандартном варианте редактор поверхностей предлагает 6 ячеек (**slot**) для одновременной работы с материалами и текстурами. В каждой из них — образец материала на освещенном шаре (вариант — кубе или цилиндре).

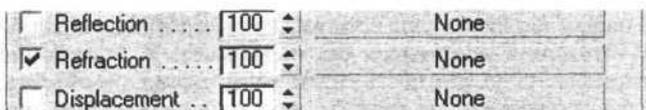


Конечно, можно выбрать готовый материал из стандартной библиотеки, входящей в состав стандартной инсталляции пакета, но в процессе обучения лучше попробовать создать свой собственный материал от начала и до конца.



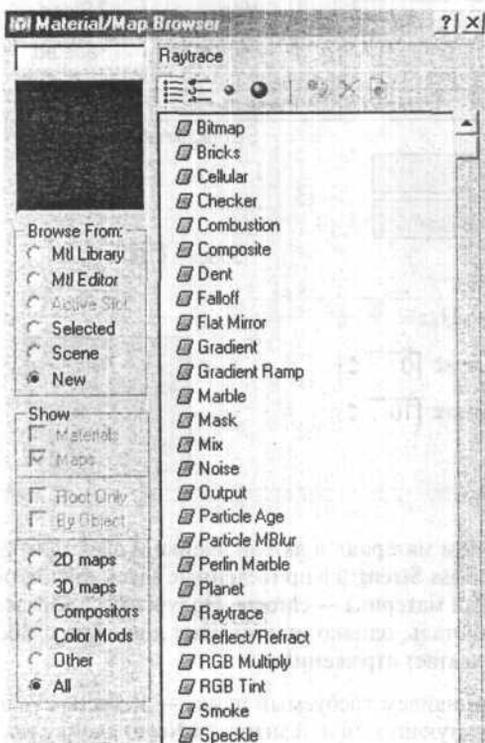
За основу возьмем материал в любой ячейке и выставим **Shading**, цвета, **Shininess** и **Shininess Strength** в приведенные здесь значения. Заодно можно и назвать наш материал — **chrome**. Полученный образец материала хромированную сталь, однако, напоминает довольно слабо. Проблема очевидна — не хватает отражений.

На панели **Maps** найдем требуемый пункт — **Reflection** — и ткнем мышкой в пока еще пустующую (в ней значится **None**) ячейку текстуры.



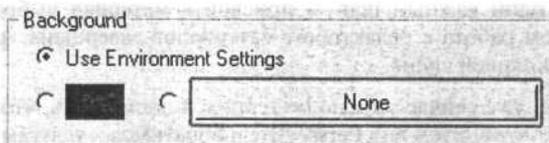
Небольшое замечание касательно терминологии: под текстурами (**map**) мы будем понимать нечто большее, чем проецируемые на поверхность объекта картинки (они являются лишь одним из параметров), в это понятие входят и так называемые процедуральные текстуры (изображения не хранятся в файле, а генерируются программно), а также многие другие данные, такие как искажения, размытость, фильтры и прочее.

В появившемся меню выбора типа текстуры выберем **Raytrace** для достижения большей реалистичности в воспроизведении отражений. Теперь в объектах с поверхностью из этого материала отражались бы другие элементы сцены.



В нашем случае в сцене есть только выстроенные в ряд буквы, поэтому отражений получилось бы очень мало. К счастью, текстура **Raytrace** предлагает возможность эмулировать окружение объекта другой текстурой — в нашем случае это будет просто некая картинка, имитирующая хромированную поверхность.

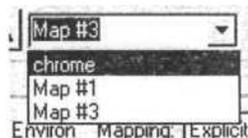
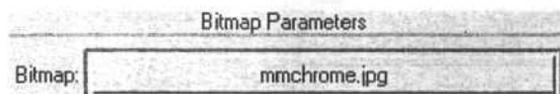
Обнаружив параметр **Background**, кликнем в опять же пока недействующую ячейку текстуры фона.



В появившемся, уже знакомом нам меню параметров типа текстуры на этот раз выберем **Bitmap** — ведь сейчас мы хотим использовать вполне конкретную картинку.



Наконец, установим параметр **Bitmap**, выбрав соответствующий файл.



В процессе создания материала мы незаметно спустились по лестнице иерархии на 2 уровня вниз (от самой поверхности к одной из ее текстур и далее к ее вспомогательному элементу). Вернемся обратно.

Наше творение должно теперь выглядеть примерно так:



Остался один важный шаг — присвоить материал выбранному объекту. На этом работа с редактором материалов завершена, закроем его и вернемся к нашей сцене.

Наверное, уже сейчас ужасно не терпится посмотреть, что же мы там натворили. Активируем вид **Perspective** и обратимся к услугам рендера (**Render Scene**), выставим размер картинке поменьше (320x200 вполне подойдет) и ткнем в кнопку **Render** (в будущем, когда все параметры рендеринга выставлены, можно пользоваться **Quick Render**, расположенной справа от **Render Scene**). Придется подождать, ведь алгоритмы **Raytracing** требуют множества вычислений.

Обнаружить панель создания источников не так сложно:



- ◆ **Spot** — конус света из определенной точки.
- ◆ **Free** — направленный поток параллельных лучей.
- ◆ **Omni** — точечный источник, излучающий равномерно во все стороны.

Для освещения сцены нам потребуются три лампы типа **Omni** потнее и один **Spot** поярче. Начнем с **Omni**.

Выбрав **Omni** и слегка уменьшив яркость примерно до **90**, ткнем левой кнопкой мыши в окне **Top** где-нибудь перед текстом — первый источник света создан. Еще две копии его можно сделать посредством клонирования — **Shift+drag-and-drop** в режиме перемещения.



Теперь очередь **Spot**. Установим его яркость примерно на 130. Контроллером видов **Zoom** изменим масштаб сцены в окне **Top** так, чтобы можно было создать центр новой лампы на достаточном расстоянии от текста, ткнем левой кнопкой мыши и протянем линию — основное направление **Spot**. Теперь осталось разместить созданные лампочки, пользуясь видами **Top**, **Left** и **Front**.

Теперь настал момент для финального рендеринга.



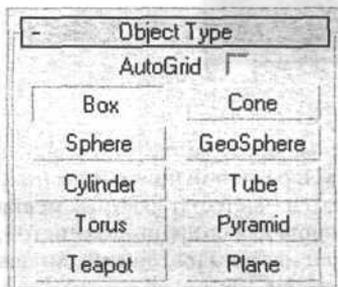
## Глава 11.

### Моделирование ландшафта

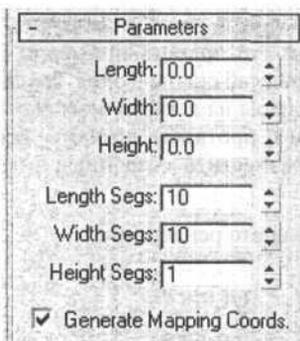
Иногда (и даже очень часто) бывает необходимо создать реалистичный природный ландшафт. Для 3DS была написана такая уйма **plug-in**'ов, позволяющих их моделировать (**3DSurf**, **Miner**), а в 3ds max есть несколько способов, позволяющих получить ландшафт, не прибегая к нестандартным **plug-in**'ам.

#### Способ №1

Создадим простой, но эффектный ландшафт. Для этого воспользуемся примитивом **Box**.



При создании параллелепипеда укажите количество сегментов по двум измерениям побольше, как показано на рисунке.



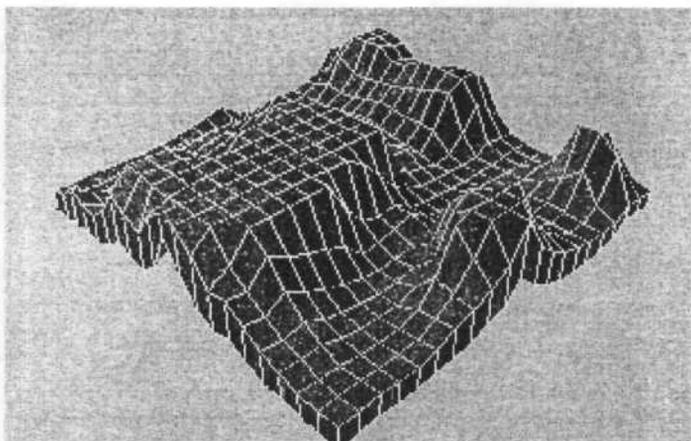
Теперь на получившийся сетчатый объект примените модификатор **Noise**. Проставьте флажок **Fractal**, **Roughness 0.0**, **Iterations 6.0**, и **Strength** по **Z** равную **50** (или другую, в зависимости от ваших масштабов).

Ландшафт готов. Добавьте материал и небо по вкусу.



## Способ №2

Ландшафты, создаваемые предыдущим методом, имеют один недостаток — некоторую монотонность. В реальной природе вы никогда не встретите горного массива, состоящего из часто натянутых мелких холмиков. Так что предыдущий метод хорош для создания холмистой, но не гористой местности. А если нам нужен настоящий горный ландшафт, то можно воспользоваться следующим путем.



Начнем так же, как и в предыдущем способе — создадим **Box** с множеством вершин. Теперь мы воспользуемся модификатором **Displace**, который изменяет геометрию предмета по картинке или по **map**'у.

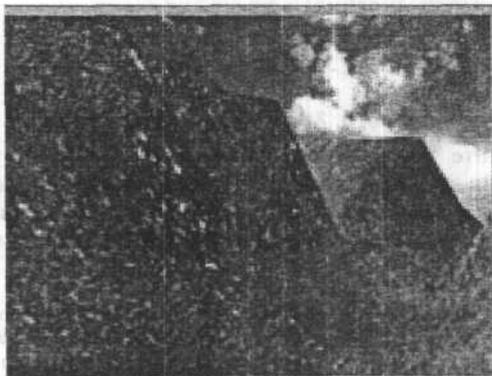
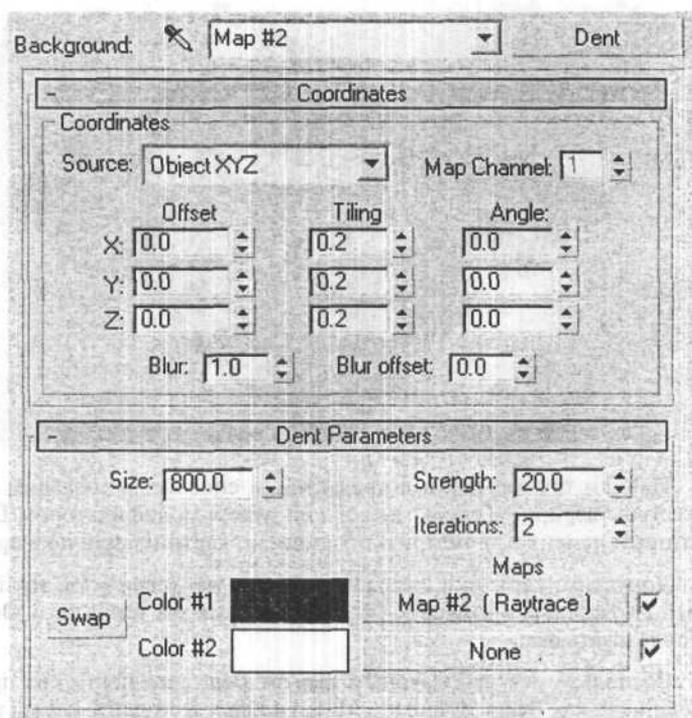
Допустим, картинки возвышений у вас нет (если есть, то вперед, она используется, как в **Bump**'е — чем светлее пиксел, тем выше). Остается использовать **map**.

Воспользуемся процедурным **map**'ом **Dent**. Это почти его прямое назначение — как **Bump**. Войдите в **Material Editor** и создайте новый **map** — **Dent**. Укажите следующие параметры:

```
Size=800
Strength=20
Iterations=2
X, Y, Z Tiling каждый по 0.2
```

Теперь примените модификатор **Displace** на ваш **Box**, указав **Strength**, равную 35 (или другую — зависит от вашего масштаба), и **Dent-map**, который вы только что создали. Если вы хотите использовать не **map**, а просто картинку, то вместо **Map** нажмите **Bitmap**, и укажите на свою картинку.

Первоначальный **Box** должен искривиться. Те его вершины, которые при накладывании на него **map**'а должны быть окрашены светлее, поднимутся, а те, которые темнее — опустятся. Теперь можно наложить на него текстуру, причем заметьте, что можно и не добавлять дополнительно модификатор **UVW Mapping**, так как сам **Displace** работает как **UVW**, если включить опцию **Apply Mapping**.

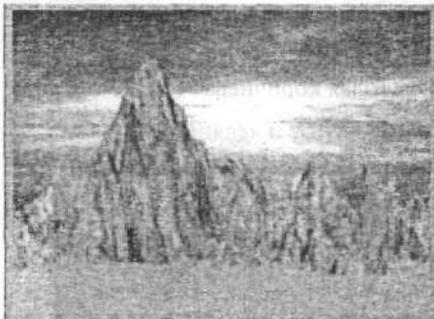


Это вариант готового отрендеренного ландшафта — вид со дна каньона.

### Способ №3

Если все-таки вам не нравится ни один из способов выше, то вам ничего не остается, как только достать один из многочисленных plug-in'ов для 3ds max — генераторов фрактальных ландшафтов.

Пример использования одного из них — **Mountain** вы видите ниже. Без его помощи трудно создать, например, такие реалистичные крутые каменистые берега.



## Глава 12. Моделирование текстуры ландшафта

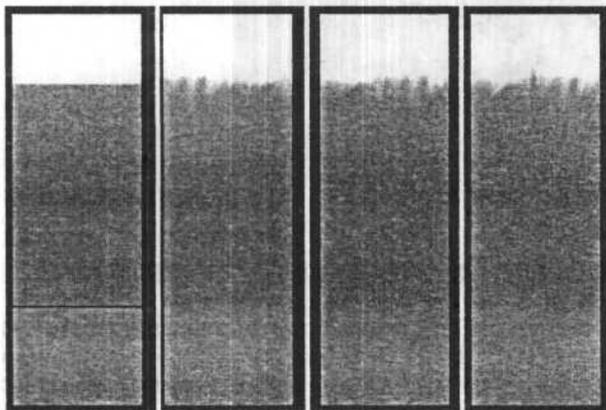
Допустим, вы уже освоили технику создания ландшафтов. Но есть одна маленькая проблема — эти способы отлично отвечают вашим запросам, когда вам нужны марсианские горы или лунные поверхности (то есть, поверхности однородного цвета — например, коричневого), а создать реалистичный ландшафт земного мира сложно. В этой главе рассказано о том, каким образом раскрасить ранее созданный ландшафт, чтобы он был похож на фрагмент реалистичного земного рельефа.



Для начала создадим ландшафт любым из способов, описанных в предыдущей главе. В этой картинке использован **plug-in Mountain**.

Теперь задумаемся о том, что мы хотим увидеть в реальном земном ландшафте. (Кстати, 3D-художникам часто приходится думать о реальной природе явлений — привыкайте к этому). Итак, нам нужно следующее:

- ◆ В нижних точках рельефа — светло-зеленая трава.
- ◆ Выше — темно-зеленые кустарники и деревья.
- ◆ Еще выше становится холоднее и там нет никакой растительности — голая коричневая земля.
- ◆ На самом верху — снежные и ледяные шапки гор.



Теперь подумаем, как можно это сделать в 3ds max 7.0. На различных сайтах несложно найти **plug-in Terrain Material**. Этот инструмент позволяет делать переход от одного материала к другому по высоте. Нам это не совсем подходит, так как у нас должно быть минимум четыре материала. Так что воспользуемся **UVW Mapping**'ом 3ds max'a, нанеся текстуру особым образом.

Закройте или прикройте 3ds max и запустите свой графический редактор (в данном случае предпочтительнее Fractal Design Painter, но подойдет и Adobe Photoshop). Мы не хотим делать сложные текстуры, так что у нас будет все просто. Создайте новый рисунок-полоску, например 100x300. Теперь нанесите на него градиентами отдельные цветовые участки.

Теперь создадим плавные переходы от одного градиента к другому. Для этого в большинстве редакторов есть функция **Smudge** (размазывание) или аналогичная (вот почему предпочтительнее FDP). Процесс размазывания очень творческий, но на самом деле требует минимального опыта и средней мышки.

Все практически готово. Осталось затайлить текстуру. Сделайте **Filter** ⇨ **Misc** ⇨ **Offset** и введите в X половину размера картинке по горизонтали. Картинка сместится и будет виден шов тайлов. Теперь размажьте и его, и получите окончательный вариант текстуры. Сохраните ее в один из каталогов **map'ов** 3ds max и закрывайте редактор.

Снова откройте 3ds max и загрузите свою сцену с ландшафтом. Идите в **Material Editor** и возьмите новый материал. Присвойте ему в **Diffuse Map Bitmap'ом** нашу текстуру рельефа (для контроля лучше установите кубик, а не шарик). Подержайте за остальные настройки материал, чтобы он выглядел получше (уберите блеск). Присвойте материал своему объекту — ландшафту.



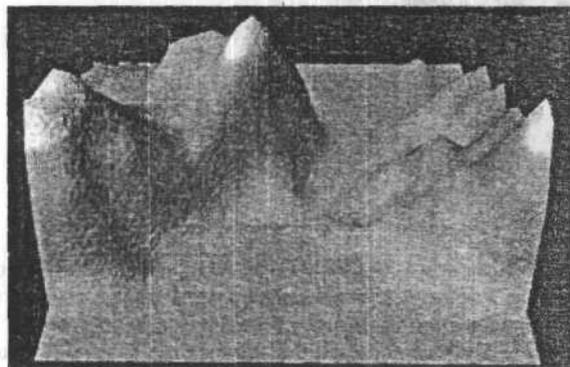
Теперь надо присвоить объекту **UVW Mapping**. Добавьте модификатор **UVW Mapping**. Убедитесь, что он имеет тип **Planar** и нажмите магическую кнопку **Sub-Object**. Разверните прямоугольник так, чтобы он был перпендикулярен плоскости ландшафта, как показано на рисунке. Отожмите **Sub-Object** и кликните на **Fit**. Теперь можно рендерить.

Получилось неплохо. Снизу у нас все зеленое, выше — коричневое, а шапки гор — белые. Но выглядит все это нереалистично из-за того, что у нас плоская и гладкая текстура, а в природе ничего такого нет. Надо бы добавить немного **Noise'a**.



Переключимся в **Material Editor** и изменим тип **Diffuse Map**'а с **Bitmap** на **Mix**. На вопрос о том, хотим ли мы оставить прошлый **Bitmap** как **Sub-Map**, ответьте «да». Теперь во второй ячейке создайте **Noise Map** с черно-белым **Noise**'ом помельче (размер где-нибудь 3-5). Теперь поставьте **Mix Amount 0.15** и посмотрите на результат. Задайте **Bump Map** как черно-белый **Noise** помельче (1-2) и отрегулируйте силу **Bump**'а, чтобы он хорошо смотрелся.

Теперь можно рендерить.



## Глава 13. Моделирование морского ландшафта

Возьмемся за море. Для придания большей реалистичности вспомним, какое море мы видели в обычной жизни. Море плещется и сверкает в лучах солнца. Погода — спокойная, вид с берега, волнение — 0.5-1 балл, тип погоды — наиболее благоприятный для 3D. Будем счи-

тать, что поблизости нет речек с грязными водами, давно не было ни дождя, ни шторма. В общем, круг задач поставлен и описан — можно приступать.

Начнем с понимания моря в такую погоду. Во-первых, динамика — волны небольшие, ветра почти нет, а значит и нет барашков на волнах. Легкая рябь, да и только. Потом цвет — кроме того, что есть понятие «цвета морской волны», надо бы напомнить, что вода в море окрашена по-разному — ближе к берегу она зеленее, дальше от берега стремится к ультрамарину. Да и взвешенные в воде частицы дают о себе знать — солнечные лучи, преломленные поверхностью, придают воде легкий разброс по температуре цвета и светлости.

Во **viewport'e TOP** создайте **CV Surface** размера **1000x1500** (лучше работать именно в таких больших масштабах, чтобы не было мучений с длинами волн вроде **0.00001**). Это поверхность моря. Теперь скопируйте эту поверхность, переместив ее с **Shift** во **viewport'e FRONT** по оси **Y** единиц на **100** — это будущие облака. (Лучше сразу обзывать все соответствующими именами — море пусть будет «море», облака — «небо»).

Теперь нужно создать «окружение» — полусферу с центром координат в точке **0,0,0** радиуса **1000**. Далее эту сферу надо переместить к нижнему краю «моря» во **viewport'e TOP** так, чтобы центр был на нижнем крае, а край сферы касался дальнего конца объекта «море». Теперь смасштабируйте сферу во вьюпорте **Front** через **Non-Uniform scale** по оси **Y** процентов на **50**. После этого выберите вместе «небо» и «окружение», разверните им нормали, присвоив модификатор **Normal**.

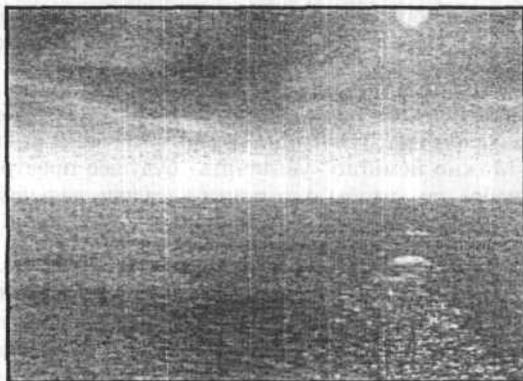
Осталось создать направленную камеру во **viewport'e TOP** (ближе к нижнему краю с прицелом вверх). Лучше принять значение угла камеры, равное **35** мм. Можно немного «увеличить» будущее пространство сцены, повернув «небо» так, чтобы оно почти касалось дальнего края «моря». С геометрией покончено.

Можно приступать к материалам. Приступим к разукрашиванию — здесь, коротко о главном. Начнем с «окружения». Это фактически купол неба, а значит и цвет его — градиентная заливка. Создадим соответствующий материал с текстурой **Gradient** на **Diffuse** — от небесно-синего до голубого (или желтоватого, если рядом город). Для корректности наложения текстуры примените модификатор **UVW Mapping** во **viewport'e FRONT** так, чтобы карта была параллельно **viewport'у** и корректно «сидела» на объекте. Этот материал должен светиться (**Self Illumination = 90-100**). Добавим облака — это белая поверхность с маской прозрачности **Noise (Fractal)** и самосветимостью **50** процентов. Поигравшись параметрами **High** и **Low**, можно регулировать плотность облаков.

Теперь перейдем к главному объекту — к морю. Для начала разберемся с цветом: при удалении от камеры он должен густеть, значит, нужно применить градиент. Внутренние блики воды — это **Noise (Turbulence и Inverted)**. Значит карта цвета будет **Mix Градиента и шума** на 0.05-0.1. Волны на море не бывают одного типа. Есть побольше — основные, вызванные ветром, землетрясением, и мелкая рябь. Для бампа мы тоже используем микс карт — воды и шума. Вода будет создавать крупные волны, а шум — рябь. Кроме того, можно при помощи ряби показать неравномерность ветра и морских течений.

Для этого нужно создать **MASK** течений и использовать его для ряби. В нашем случае подойдет и просто обычный шум большого размера, следовательно, задача упрощается — вместо белого цвета шума-маски будем использовать мелкую карту шума ряби. Можно усложнить задачу, задав обычный шум для белого и **turbulent**'ный для черного — это будет ближе всего к истине. Ну и, конечно же, отражения. В качестве карты отражений в данной сцене можно смело использовать **Raytrace**, так как она не содержит в себе никаких глюков. Но главное — запомните, что коэффициент отражения воды не так велик, как кажется. Не переборщите.

Теперь осталось осветить картину. Положение, когда вы смотрите на море против света, кажется наиболее реальным.



Для света используем 1 основной **Direct Light** с **Raytrace**'овыми тенями, исключив из его освещения «окружение». Кроме того, один точечный источник покажет нам положение солнца.

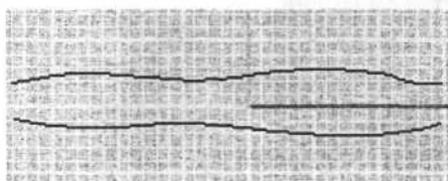
Осталось добавить **Flare**'иков и тумана по вкусу и море готово. Запомним, что в данном случае можно все анимировать.

## Глава 14.

# Моделирование руки по четырем огибающим

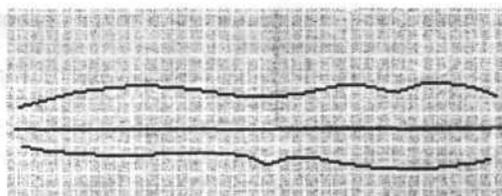
**Rhino3D** — один из самых лучших NURBS-моделлеров. Этот совет посвящен моделированию в нем органической поверхности — руки. Он направлен на то, чтобы показать один из самых базовых методов построения поверхностей — построение по четырем огибающим.

Запустите **Rhino3D** и начните новый проект. Для начала нам нужны собственно 4 огибающие кривые для руки. Для их построения воспользуемся инструментом **Curve**, рисуя интерполированные кривые.



Выберите вид **Top** и постройте две кривые, так, как показано на рисунке — они будут означать граничные значения будущей поверхности нашей руки. Сначала она немного утолщается, потом утоньшается к локтю, и затем идет достаточно большой изгиб (только не переборщите) — это мускулы — бицепсы.

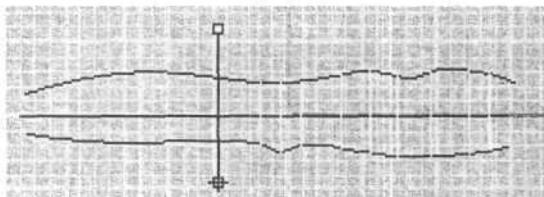
Теперь надо построить еще две кривые — ограничение по виду спереди. Перейдите в вид **Front** и постройте еще две кривые. Можете придать им некоторую бугристость — обычно мускулы в вертикальном положении напрягаются больше, чем в горизонтальном. Не забудьте про локоть.



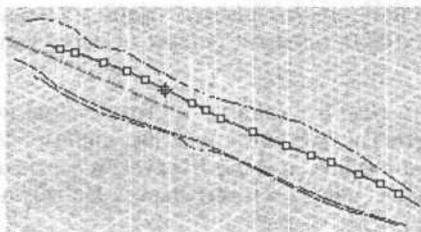
На этом творческая работа почти окончена. Остальное — дело техники. Теперь нам надо построить много сечений нашей будущей поверхности и пролофтить их. **Rhino3D** выполняет построение сечений поверх-

ности, заданной четырьмя огибающими без проблем — используйте команду **CSec Profile** из **Curve Tools**.

Теперь очень аккуратно выберите свои четыре кривые в следующей последовательности: передняя, верхняя, задняя, нижняя (в общем, в любой последовательности по, или против часовой стрелки). Для этого удобно использовать вид **Perspective**. Теперь нам надо указать, где проводить сечения. Так как у нас достаточно сложная форма, лучше сделать это вручную. Сечения проводятся так же, как линии в любом виде, хотя лучше всего использовать **Front**. Проведите их около десяти-пятнадцати, обязательно выделяя основные выпуклости и вогнутости поверхностей. Для проведения прямых линий удобно зажимать **Shift** — тогда линии будут проводиться только параллельно сетке — как в **Photoshop**.

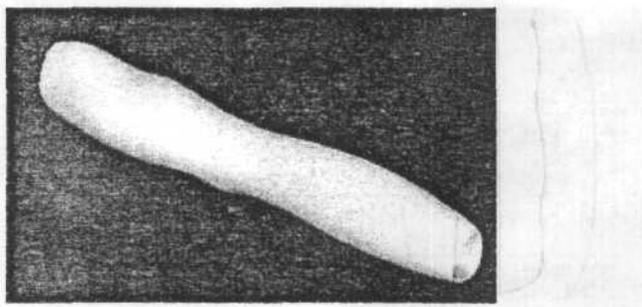


Закончив делать сечения, нажмите **Enter**. Сечения уже выделены, осталось только сделать лофт в меню **Surface** ⇨ **Loft**.



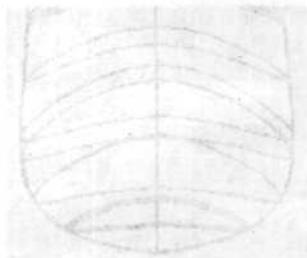
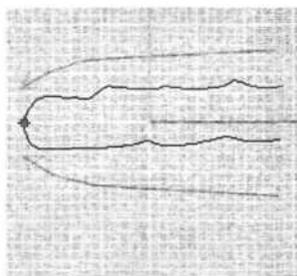
Теперь нужно, чтобы векторы направления и точки приложения векторов были одинаковы для всех сечений. Если что-то не так — поправьте, подвинув вектор или изменив направление командой **Flip**. Есть еще команда **Auto**, которая сделает все автоматом, но не всегда правильно. Нажимайте **Enter** и в диалоге говорите **ОК**. Проверьте, чтобы в диалоге **Closed Loft** был выключен. Чтобы при будущих построениях эта куча сечений вам не мешалась, пока она выбрана, нажмите **Hide**.

Конечный результат — рендеринг.

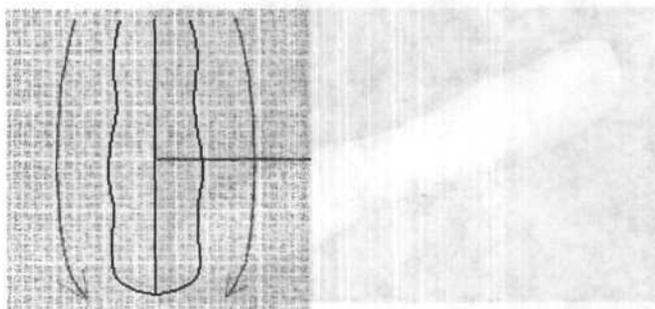


## Глава 15. Моделирование одного пальца руки

Здесь будет описан еще один распространенный метод — лофтинг огибающих. Нарисуйте инструментом **Curve** две кривые в виде **Right**. Главное — это чтобы их концы совпадали. Для этого проще всего воспользоваться инструментом **Snap** — курсор будет ходить по клеточкам сетки. Следите также, чтобы вы вели обе кривые в одном направлении, то есть, если вы заканчиваете первую кривую общей точкой со **Snap**, то вторую ведите в том же направлении (они показаны на рисунке красными стрелками).



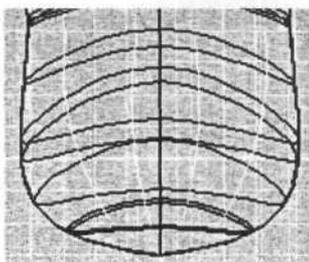
Теперь сделайте еще две кривые в виде **Top**. Они также должны смыкаться в одной и той же точке — снова задействуйте **Snap**. Так как сверху палец симметричен, вы можете воспользоваться инструментом **Mirror** для того, чтобы нарисовав лишь одну кривую, отзеркалить ее и получить вторую. Не забывайте также про направления! Ведите кривую в направлении, показанном стрелкой.



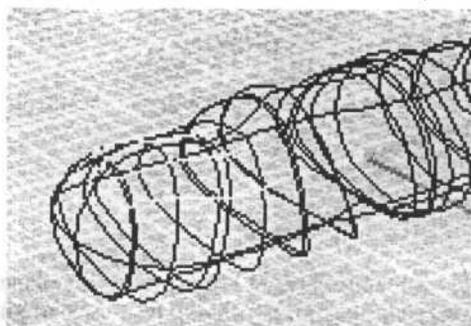
Теперь снимите все выделения и выполните команду **Loft**. Теперь по очереди (по/против часовой стрелки) в виде **Perspective** выберите все кривые. После выбора всех четырех нажмите **Enter**. В появившемся окошке обязательно включите **Closed Loft**.

Палец в принципе готов. А для тех, кто не прочь сделать еще кое-что для реалистичности, предлагаем еще одну идею. Попробуем создать ногти.

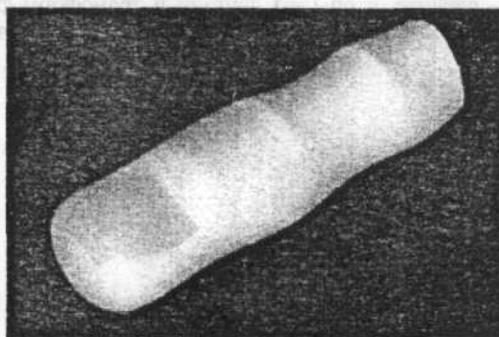
На конце пальца, где заранее предусмотрено место для ногтя, прочертите четыре кривые — ограничители ногтя. Две из них можно даже отзеркалить. Выделите все четыре кривые. Теперь инструментом **Project** спроектируйте эти кривые на поверхность пальца.



Получилось 3 серии кривых: одна — изначальная — посередине пальца, вторая — лежащая сверху пальца и третья — снизу. Выберите и удалите первую и третью. То, что осталось, выберите по очереди и выполните команду **Loft**. В окошке опций уберите **Closed Loft** и, чтобы не перегружать сцену ненужными деталями, установите **Rebuild with 5 points**. Картина готова.



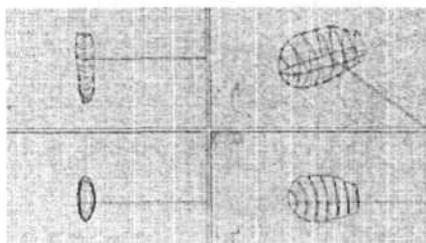
Покрасьте получившуюся пластинку в какой-нибудь другой цвет, можете немножко поднять ее для видимости и отрендерите.



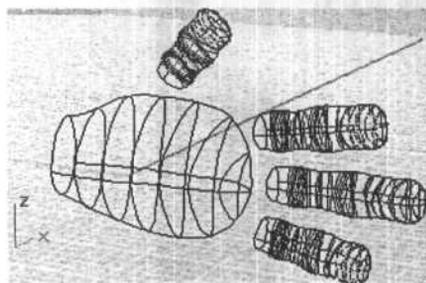
## Глава 16. Моделирование полной руки — ладонь, пальцы, плечо, предплечье

Итак, у нас уже есть палец и рука. Пора сделать полную руку — пальцы, ладонь, предплечье и плечо. Задача поставлена — сделать ладонь, собрать ладонь и пальцы в кисть руки и присоединить кисть к предплечью.

Сначала — ладонь. Она выполняется тем же методом, что и палец (лотфинг огибающих). Единственное уточнение: следите за направлением кривых, не забудьте про **Closed Loft** и для облегчения последующего редактирования сделайте **Rebuild with 8 points**.

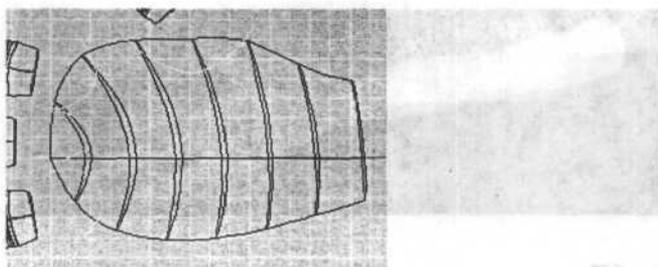


Теперь начинается самое интересное. Подгрузите ваш палец (**File** ⇒ **Merge**). Поверните его, если нужно, передвиньте и смасштабируйте так, чтобы он подходил к ладони. Теперь с помощью команд **Copy** и **Rotate** (с опцией «С» — **Copy**) размножьте свой палец. Каждый палец надо немножко изменить — повернуть, растянуть (**Scale-ID**). Если у вас пальцы с ногтями — следите, чтобы вы двигали и трансформировали ногти вместе с пальцами. У вас должно получиться примерно вот это.



Пусть будет всего 4 пальца. Посмотрим на то, что у нас получилось. У нас есть 4 поверхности с «дырками» — пальцы и одна поверхность с одной дыркой для предплечья — ладонь. Чтобы присоединить пальцы к ладони, можно воспользоваться командой **Blend Surface**, но для нее нам нужны «дырки» с обеих сторон — и в пальцах, и в ладони. Нужно вырезать отверстия в ладони!

Проведите 4 кривые так, как показано на рисунке.

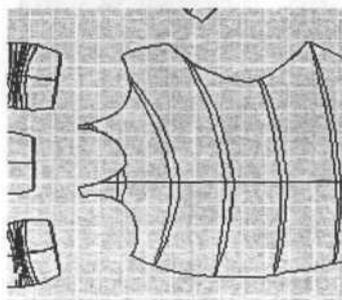


Этими кривыми мы как бы отрубим часть ладони. Они должны быть как можно более аккуратными и ни в коем случае не пересекаться, хотя и располагаться достаточно близко друг к другу.

Просто так отрезать ими мы не можем, так как нужно, чтобы кривые лежали на поверхности объекта ладони. Спроектируем их на нее — команда **Project**. После проектирования ваши первоначально нарисованные кривые уже не нужны, их можно удалить.

Настало время для команды **Split** в ее основном применении. Нажмите эту кнопку, выберите ладонь как объект и все три кривые как **Cutting Objects** (режущие объекты). Жмите **Enter** и ждите. Эта операция — одна из самых долгих по времени, так что для сложных объектов порой приходится ждать несколько минут.

Все готово — нужные нам куски отрезаны. Выделите их и удалите. Режущие кривые нам теперь тоже не нужны — можете их спрятать или стереть. Дырки есть — можно использовать **Blend Surface**. Выполните эту команду (она находится в **Surface Tools**).



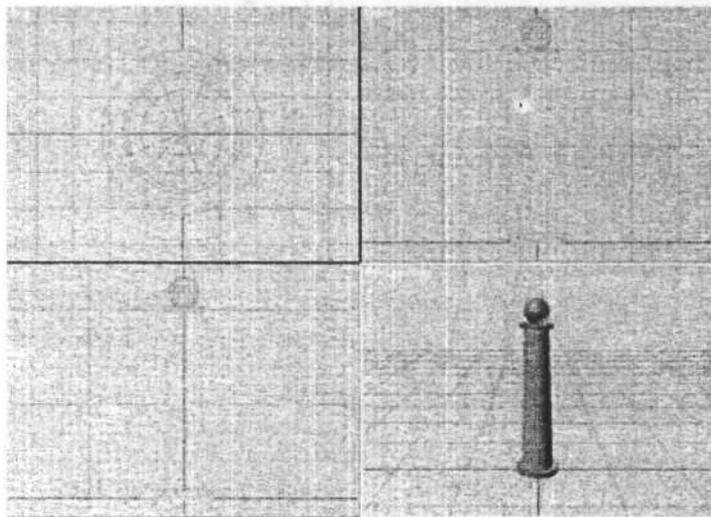
Теперь приставить предплечье к кисти руки не составит никакого труда, тем более, что дырки уже есть — осталось только сделать **Blend**.



## Глава 17. 3D-моделирование

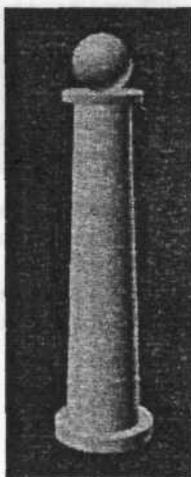
3D-моделирование — одно из самых интересных приложений в области компьютерных технологий. С помощью современных программ человек может фактически создать свой собственный мир трехмерных образов и «дорабатывать» его по своему усмотрению. Ни один современный боевик, комедия или просто интересный кинофильм уже не представляется нам без серии потрясающих спецэффектов, которые заставляют многих не верить, что происходящее на экране нарисовано на компьютере, а не происходило на съемочной площадке «взаправду».

Допустим, мы хотим нарисовать простенькую колонну с шаром на вершине. Колонна будет усеченным конусом с цоколем и опорой.



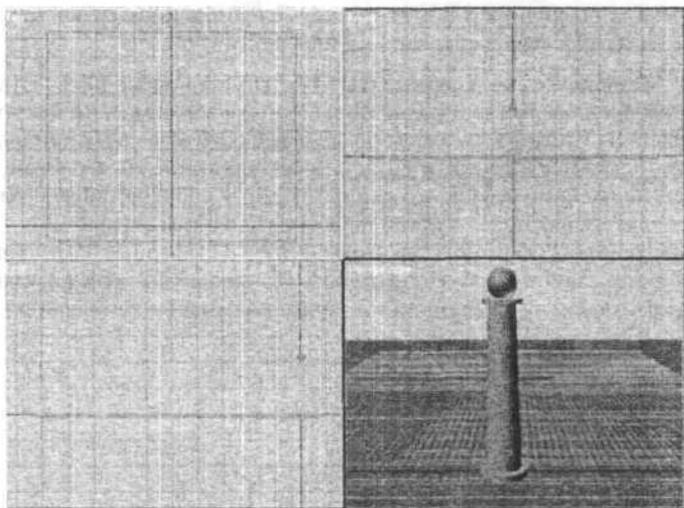
Как вы видите, 3ds max 7.0 по умолчанию показывает три проекции — сверху, слева и спереди — и перспективу.

Однако в окне перспективы показывается очень некачественное, черновое изображение, предназначенное для максимально быстрой перерисовки. Многие возможности 3ds max и спецэффекты становятся видны только после визуализации или рендеринга. Суть его в том, что из простейших примитивов, созданных вами в процессе моделирования, генерируется ряд поверхностей по весьма сложным алгоритмам — с учетом источников света, текстур, отражения, атмосферных эффектов строится фотореалистичное изображение. Попросим 3ds max визуализировать нам эту сцену с параметрами, принимаемыми по умолчанию.

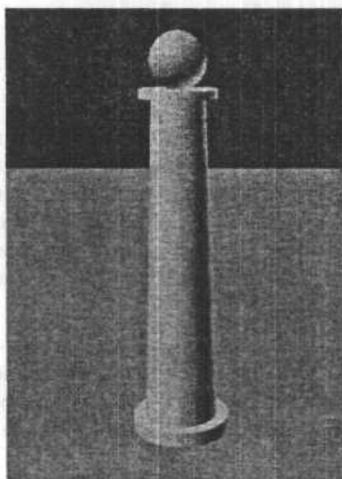


Не правда ли, не слишком «фотореалистичное» изображение? Но это только начало. На представленном изображении цвета объектов оставлены по умолчанию, поэтому на границе колонны и основания видна неприятная граница, несмотря на то, что она слегка смягчена фильтром антиалиасинга (**antialiasing**) 3ds max. Этот фильтр предупреждает резкие, «рубленные» кромки на границе двух объектов путем их взаимного размывтия.

Логично предположить, что цоколь, колонна и основание — единое целое. Для более качественного рендеринга с учетом этого, объединим до разные объекты путем создания **Compound** — составного объекта **Boolean**. Кроме того, добавим большой прямоугольник под колонну, олицетворяющий «почву».



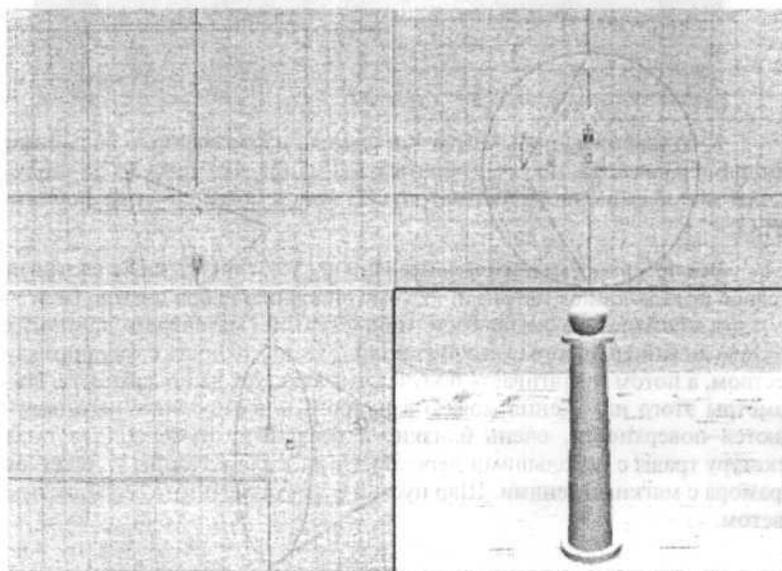
Как мы видим, за счет того, что в сцену был добавлен несоизмерно большой объект — прямоугольник, «основание» — видовые окна, значительно уменьшили свой масштаб из-за применения операции *Fit*, то есть, «подогнать» масштаб под размеры сцены. Перспективу же мы специально увеличили и сдвинули, чтобы последующая визуализация получилась более наглядной.



Теперь картина начинает вырисовываться. Использовался стандартный источник света, так как мы еще не определили своих источников. При визуализации был сделан более качественный шейдинг, то есть, мягкий переход окраски граней объекта с учетом его освещенности. Однако, что и говорит, объект выглядит скучновато и не эффектно. Как можно это побороть?

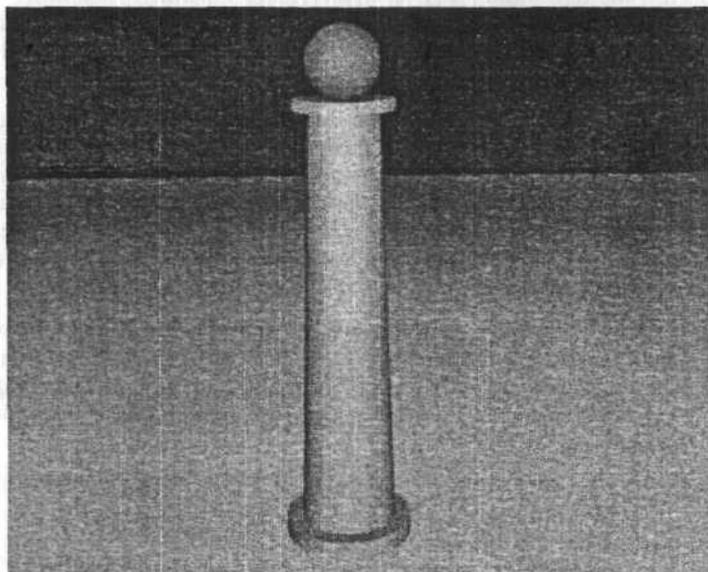
После того, как мы создали простую геометрическую модель, стоит выбрать для нее «сцену». Сначала создадим камеру. Применяемый по умолчанию в нижнем правом окне вид **Перспектива**, в принципе, неплох, но он не позволяет нам легко и быстро менять точку обзора. Если же мы создадим камеру и выберем ее в окне перспективы, то, меняя ее расположение и ориентацию в других видах, сможем в реальном времени отслеживать изменения «пейзажа» в окне камеры.

Кроме того, создадим подсветку для сцены. Пусть колонна будет освещена мягким боковым светом, а ее задний вид теряется в полутени. Это важно, так как мы не успели создать обстановки на заднем фоне и вид уходящего в никуда основания сцены будет выглядеть очень непривлекательно.



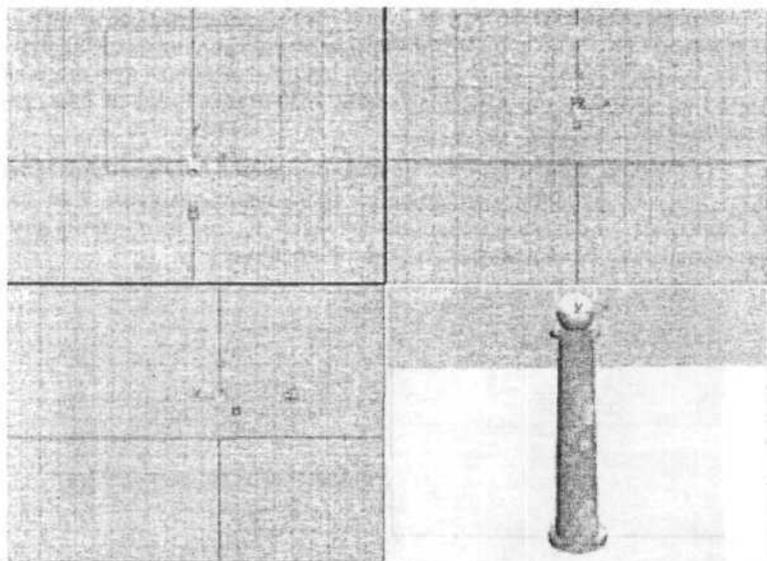
Не обращайте внимания на «засвеченность» правого нижнего вида — дело в том, что 3ds max некорректно отображает освещенность сце-

ны в окне предварительного просмотра при наличии сложных источников света.

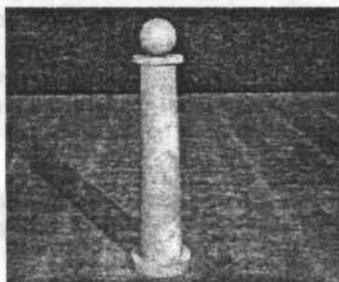


К сожалению, приходится записывать изображение с большими потерями в качестве, поэтому обратите внимание, что полосам на «полу» сцены мы обязаны не визуализатору 3ds max, а погрешностям формата JPEG.

Все поверхности в реальной жизни представляют из себя сложный и слабо поддающийся математическому описанию набор цветов. Поэтому, хотя в библиотеке 3ds max есть «процедурные» материалы, во многих случаях дизайнер просто сканирует реальную поверхность с хорошим качеством, а потом «натягивает» полученную текстуру на поверхность. Параметры этого наложения можно варьировать, в результате чего получаются поверхности, очень близкие к реальным. Выберем для пола текстуру травы с небольшими неровностями, а для колонны — текстуру мрамора с мягкими тенями. Шар пускай будет отсвечивать серебристым цветом.

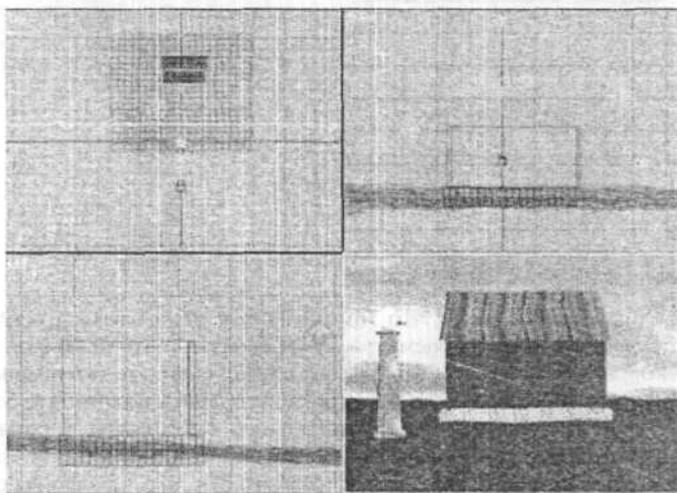


Обратите внимание, что 3ds max может показывать текстуры и в окне предварительного просмотра. Надо заметить, что делает он это не очень корректно, поэтому этот уровень визуализации годится только для приблизительной прикидки. Вы можете исправить положение, установив на свой компьютер профессиональную видеокарту, ускоряющую отображение OpenGL-сцен. К тому же подобная карта будет визуализировать предварительный вид сцены намного быстрее, так что вы сможете даже на полном экране просмотреть будущий видеоклип с неплохим качеством. На скорость окончательной визуализации специальная видеокарта никак не влияет.



Теперь сцена уже годится для того, чтобы распечатать ее на струйном принтере. Но на ней пока что нечего рассмотреть. Любая сцена нуждается в заднем плане. У нас же на заднем плане — зияющая пустота. Как сделать так, чтобы на картинке был какой-то фон и колонна не была такой одинокой?

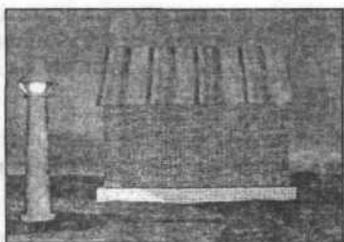
Придумаем фон, немного трансформируем нашу колонну. Она станет уличным фонарем, освещающим траву перед домиком. Сам домик набросаем, не создавая подробных деталей. Кроме того, слегка изомнем «траву», создавая иллюзию неровной полянки.



Как видно, 3ds max умеет создавать достаточно наглядное изображение в окне камеры (да и в любом другом). Оно вполне годится для предварительного просмотра сцены. Правда, из-за того, что в окне просмотра неправильно учитываются источники свет (точнее, они обчисляются слишком примитивно), сцена выглядит излишне светлой. Кроме того, никаких теней и иных световых эффектов вы в окне не увидите. Зато на достаточно мощном процессоре или с помощью специализированной видеоплаты, вы сможете просматривать анимацию в большом окне чуть ли не в режиме Real-time.

Чтобы подобрать лучшее соотношение между качеством предварительного изображения и эффективностью работы, можно выбирать из различных режимов качества отображения.

Естественно, изображение будет значительно приятнее, если мы его визуализируем.

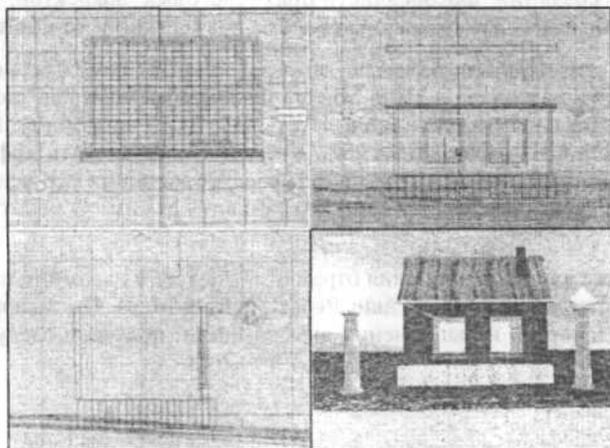


Не правда ли, не слишком похоже на картинку в окне просмотра. Добавились световые эффекты от фонаря, сама картинка стала более темной. Текстуры визуализировались с гораздо лучшим качеством.

В составе следующей сцены применены в основном материалы типа **Bitmap**. То есть, с помощью сканированного растра мы пытаемся изобразить материалы реального мира.

Но вещи обычно еще больше свойств: прозрачность, отражение, преломление, пропускание света и так далее.

Попробуем повысить реалистичность нашей сцены, используя в ней материалы с этими свойствами.



Во первых, более подробно детализируем сцену. Домику стоит прерисовать конек на крыше, окна с подоконниками, трубу, и добавить

вторую колонну, она же — фонарь для симметрии. Кроме того, пусть домик будет стоять на берегу озера.

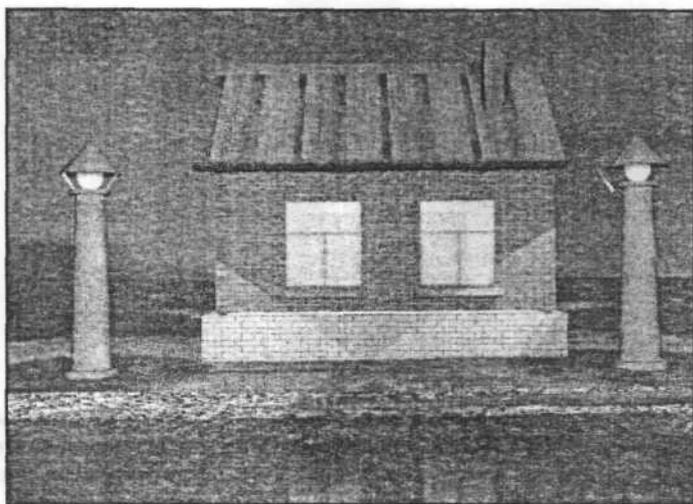
Первый из материалов с расширенными свойствами — это материал занавесок и оконного стекла. Занавески сделаны наполовину прозрачными, а стекло — прозрачным полностью. В результате мы можем видеть свет, льющийся из окна на берег и траву.

Воду сделаем чуть прозрачной, чтобы показать элементы дна. Кроме того, применим очень интересный, но крайне ресурсоемкий прием — создание отражающего свет материала.

Как известно, все материалы, кроме абсолютно черных тел, отражают свет, но такими в реальном мире являются только звезды — «карлики», которые свет также поглощают и пропускают. Все эти параметры различны для света с разной длиной волн. Для того, чтобы синтезировать такие эффекты в пакете трехмерной графики, используется методика трассировки лучей — **ray tracing**. Ее суть состоит в том, что каждый каждый луч света, испускаемый источником, рассматривается отдельно. При этом для каждого встретившегося предмета, исходя из заданных пользователем типов материалов, вычисляем коэффициент пропускания, отражения и поглощения. После этого продолжаем отслеживать путь модифицированного луча до тех пор, пока это необходимо. Количество шагов трассировки лучей может быть задано пользователем как параметр подсистемы визуализации. К сожалению, этот метод, необходимый для создания фотореалистичных световых эффектов, очень ресурсоемок и зачастую увеличивает время визуализации во много раз.

Спускаясь с высот на землю, заметим, что в 3ds max 7.0 есть специальная карта материала — **Flat Mirror**, предназначенная для создания плоских зеркальных поверхностей. Пусть вода будет спокойной. Поставим коэффициент отражения в 25%. Кстати, чтобы ощутить эффект от всех этих прозрачностей и отражений, следует заставить все основные источники света отбрасывать тень и установить режим обсчета теней в **Raytraced**.

Также для использования отражения следует в параметрах визуализации установить флажок **Auto Reflect/Refract Maps**. По завершении всех этих действий и выполнения визуализации, получаем следующий результат:



Не правда ли, отражение домика и фонарей в воде придают этой сцене реалистичный и интересный вид?

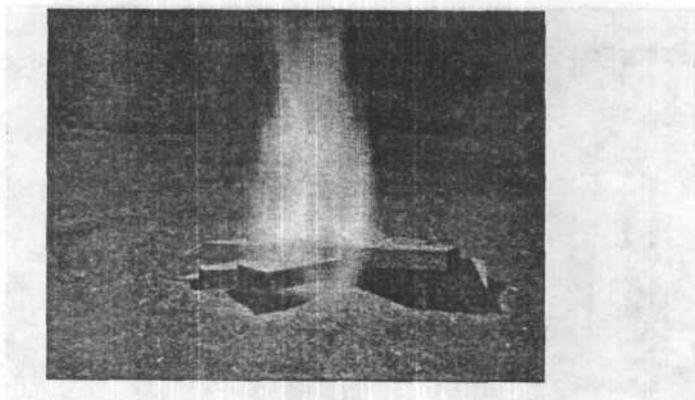
К сожалению, время визуализации увеличилось в 10 раз.

В следующем шаге мы применим и другие «продвинутые» материалы 3ds max для придания нашей сцене еще большей реалистичности и законченности.

В составе 3ds max имеются инструменты, придающие дополнительную реалистичность изображению. Это так называемые атмосферные эффекты. Они, в отличие от геометрических примитивов, представляют из себя скорее растровый эффект, хотя их параметры и задаются аналитически.

Первый эффект — это туман. Подробнее с ним можно ознакомиться здесь. Существуют и его разновидности — трехмерный туман, имеющий больше параметров настройки.

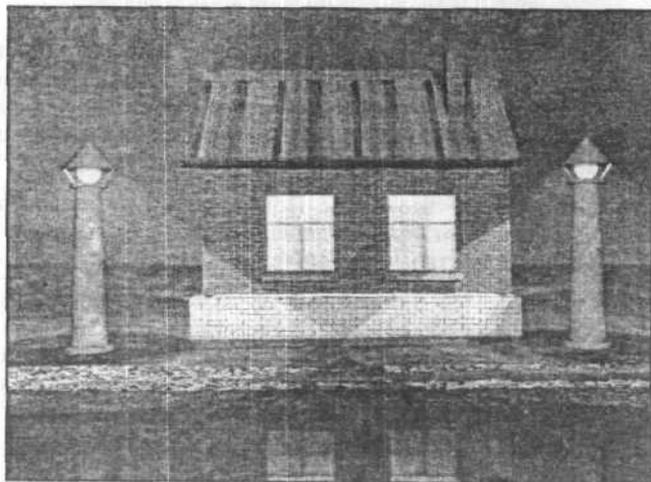
В 3ds max имеется интересный эффект под общим названием **Combustion**. Он не имеет какой-то определенной характеристики, так как при желании, с его помощью можно смоделировать весьма широкий класс объектов — от пламени пожара до взрыва бомбы или дыма от костра.



Это один из примеров применения эффекта. На самом деле, **Combustion** имеет большое количество параметров, действие которых редко можно точно предугадать. Чаще приходится подбирать значения разных коэффициентов, добиваясь желаемого эффекта.

И, наконец, один из самых интересных эффектов — объемный свет — **Volume Light**. Представьте себе, что сквозь узкое отверстие в темной пыльной комнате светит лучик солнца, и вы поймете механизм действия этого эффекта.

Вот пример применения такого света в нашей сцене:

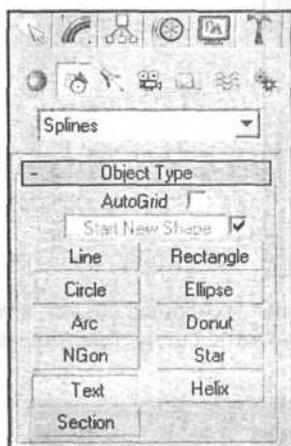


Если вы заметили, фонари теперь не только отбрасывают тени, но и виден их свет, отражаемый пылью в воздухе.

Используя опцию синтеза цветного объемного света, можно создавать интересные динамические ролики, где **Volume Light** будет исходить от вывесок, источников света и прочих светящихся объектов.

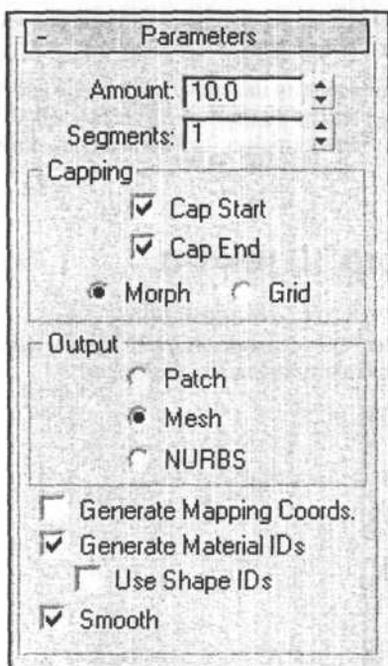
## Глава 18. Создание красивого 3D текста

Сначала нам следует создать плоский, «обычный» текст — таким вы видите его в своем любимом текстовом редакторе. На рисунке показано расположение примитива «текст» на панели 3ds max 7.0.



Нажмите кнопку **Text**. В раскрывшемся свитке выберите параметры создаваемого объекта, в том числе шрифт, размер шрифта, выравнивание, стиль и кернинг.

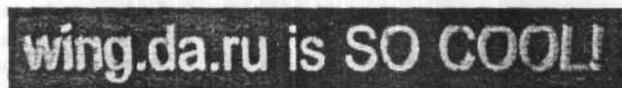
В результате вы получите плоский текст, отображаемый в видовом окне белыми линиями. Для создания настоящего текста следует применить операцию **Extrude**. Эту операцию можно сравнить с выдавливанием на торт крема из специального шприца. Укажите толщину создаваемого объекта. Не стоит делать его слишком «толстым» — это плохо скажется на его виде после визуализации.



Теперь мы получили полноценный трехмерный объект, к тому же достаточно сложный. Не стоит перегружать сцену текстом, так как потребуются больше памяти.



Конечно, такое изображение выглядит скучновато. Добавим источник света и наложим на текст текстуру:



Теперь раскрасьте это изображение, как вам нравится.

## Глава 19.

### Создание изображения лампы

#### Черновое построение

На проекции **Front** сплайнами (**Create ⇨ Shapes ⇨ Splines ⇨ Line**) строим заготовки для объектов: ножка лампы, патрон, лампочка, абажур, часть держателя абажура. Для простоты построения делаем все сегменты прямолинейными.

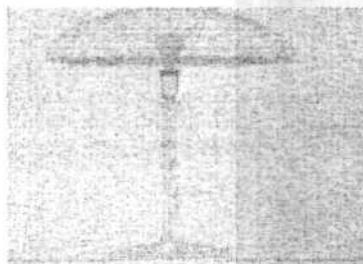
#### Уточнение формы сплайнов

Уточняем форму каждого сплайна. Для этого выбираем сплайн, и на закладке **Modify** выставляем параметры группы **Interpolation** для удаления угловатостей сегментов сплайна. При нажатой кнопке **Sub-Object** в режиме **Vertex** редактируем форму сплайна.

*Совет:* В первую очередь нужно выделить все вертексы, щелкнуть на любом выбранном вертексе правой кнопкой мыши и установить тип вертекса **Smooth**, затем **Bezier** или **Bezier-Corner**. Это обычно облегчает дальнейшее редактирование формы сплайна.

#### Создание из сплайнов объемных объектов

Для каждого объекта, кроме части держателя абажура, применяем модификатор **Lathe**. Сразу после применения модификатора необходимо установить нужное выравнивание оси вращения (и/или направление оси).



В данном случае для всех объектов хватает нажать кнопку **Align:Min**. Необходимо установить чекбокс **Weld Core** для слияния вертексов у полюсов объекта. **Flip Normals** используется при необходимости. **Segments** можно увеличить для уменьшения угловатости объектов (особенно для крупных деталей).

Для создания части держателя абажура будем использовать **Loft** (**Create** ⇨ **Geometry** ⇨ **Loft Object** ⇨ **Loft**). Построенный сплайн будем использовать как путь для лофт-объекта. Необходимо еще построить сплайн-сечение объекта: **Create** ⇨ **Shapes** ⇨ **Splines** ⇨ **Circle** с небольшим радиусом.

Выбираем сплайн, который будет путем, нажимаем кнопку **Loft**, нажимаем кнопку **Get Shape** и указываем курсором на объект-сечение, то есть, окружность. Для более мягкой формы в параметрах лофт-объекта устанавливаем **Skin Parameters** ⇨ **Options** ⇨ **Path Steps** 10.

Для того, чтобы видеть на всех видах лофт-объект полностью отмечаем чекбокс **Display** ⇨ **Skin**.

### Создание держателя абажура

Для клонирования объекта поворотом необходимо правильно установить **pivot-point** этого объекта. Отмечаем построенную часть держателя абажура. На закладке **Hierarchy** (кнопка **Pivot** нажата) нажимаем кнопку **Affect Pivot Only** и совмещаем ставший доступным **pivot** в виде **Top** с осью вращения лампы. Отжимаем кнопку **Affect Pivot Only**.

Клонировуем объект. При активном виде **Top** в **toolbar** нажимаем кнопку **Array** и в появившемся диалоге устанавливаем: вращение по оси **Z** на 120 градусов, **Array Dimensoins 1D Count**=3, **Type of Object** — **Instance**.

Теперь доработаем два куска трубы: крепление к ножке лампы и кольцо, на котором лежит абажур.



## Глава 20.

### Пэчечное моделирование

Перед тем, как приступить непосредственно к моделированию, надо понять, что требуется для построения модели.

Из инструментов нам понадобятся **Line (Spline)**, **Quad Patch** и модификатор **Edit Patch**.

#### Анализ

Для быстрого и правильного построения модели необходимо совершенно четко себе представить, что вы хотите получить в итоге. То есть, закрыть глаза и представить в уме моделируемый объект. Посмотреть на него с разных сторон. Постараться выявить закономерности его формы.

Возможно ли расчленить объект на более простые составные части, которые без ущерба для окончательного результата могут быть выделены в самостоятельные объекты.

Является ли объект симметричным (осе-, зеркально- и прочим), содержит ли подобные части? Если да, то вполне вероятно, что можно вначале построить только часть объекта, а остальные части получить с помощью того или иного вида клонирования, возможно, с последующей доводкой откопированной части. Например, при моделировании кисти руки можно непосредственно моделировать только два пальца: «большой» и любой другой. Дальнейшие рассуждения относятся к объекту целиком (если он не содержит повторяющихся частей) или к части объекта.

Далее необходимо проанализировать форму объекта. Провести по его поверхности характерные линии. Эти линии должны служить как бы каркасом объекта, помещенного в координатное пространство (центр объекта совпадает с началом координат). Какие линии будут характерными?

- ◆ Линии сечения объекта координатными плоскостями.
- ◆ Линии, ограничивающие отверстия.

Часто характерными линиями могут являться линии, проведенные по выпуклостям или вогнутостям объекта. На этапе освоения рекомендовано использовать их. Однако эти элементы поверхности могут быть смоделированы и другими способами.

## Синтез

Линии  $3ds \max'a$  (сплайны) строятся по опорным точкам (вертексам). Вертексы занимают одну из ключевых позиций в данной методике. Поэтому после того, как каркас из характерных линий построен, надо хотя бы примерно представить, в каких местах этих линий будут опорные точки и сколько их должно быть. Полезно также попытаться соединить вертексы соседних характерных линий дополнительными линиями, чтобы получился более детализированный каркас. Каждая ячейка этого каркаса должна представлять собой криволинейный трех- или четырехугольник. Если это требование не выполняется, то необходимо достроить вспомогательные линии. Если получившийся каркас имеет много треугольных ячеек, то он построен неверно. Треугольные ячейки должны использоваться только в случае необходимости, так как осложняют построение и окончательную доводку модели. Попробуйте построить каркас немного по другому и/или добавить вертексы в характерные линии. Желательно, чтобы ячейка не имела резких (больше чем на 90 градусов по одной из осей) перегибов.

Почему ячейки каркаса должны иметь три или четыре угла? Потому что в  $3ds \max'e$  используются два вида patch'ей: треугольный и четырехугольный. Каждая ячейка каркаса должна быть закрыта одним patch'em.

Чем больше линий каркаса, тем точнее будет модель, и тем больше времени потребуется на моделирование (временные затраты возрастают нелинейно).

Теперь можно приступать непосредственно к моделированию.

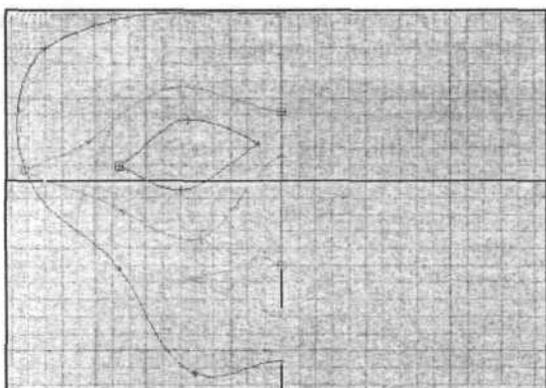
Для примера построим модель некоей маски.

### Шаг 1. Выявление закономерностей

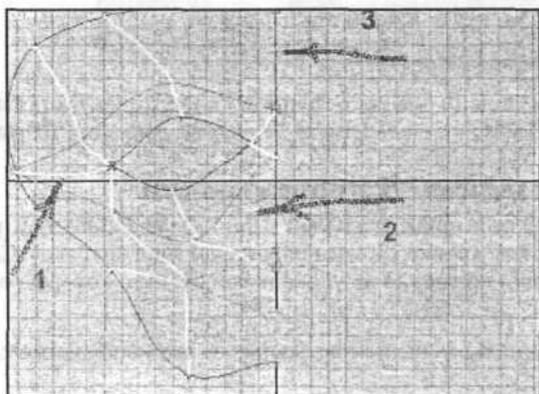
Маска является симметричной относительно плоскости YZ. Следовательно, можно строить только половину (допустим, левую).

### Шаг 2. Построение характерных линий

На виде спереди и слева строим сплайнами характерные линии. При регулировке сплайнов обращаем внимание на то, чтобы «правые» (на виде спереди) вертексы имели координату X, равную 0 (для облегчения в дальнейшем сопряжения половинок).



Мы выбрали в качестве характерных линий линии сечения координатными плоскостями, линии, ограничивающие рот (губы) и глаз, а также линии переносица-бровь-висок и переносица-скула-висок.

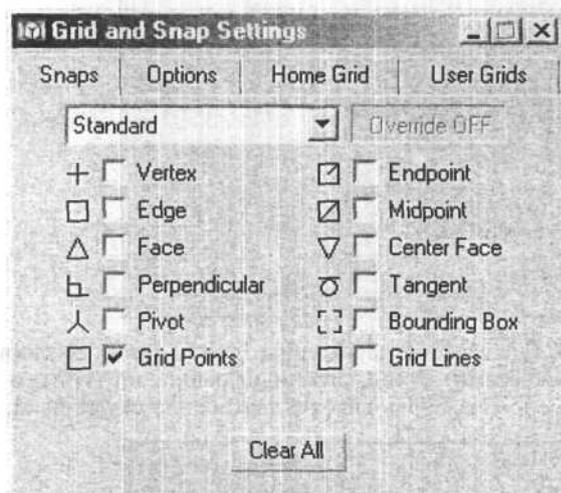


Далее окончательно регулируем сплайны. При построении линий и при дальнейшем построении модели удобно пользоваться «прилипалкой» (Snap).



Режим «прилипания» переключается с клавиатуры клавишей S. Перед использованием необходимо указать, к каким объектам будет прилипать курсор. Для этого на кнопке Snap Toggle необходимо щелкнуть

правой клавишей мыши. Появится диалог настройки. Для наших целей необходимо установить прилипание к вертексам.



Для каждого сплайна (можно групповой операцией) устанавливаем флажок **Vertex Ticks** в разделе **Display Properties**, закладки **Display**.

### Шаг 3. Создание Patch-поверхности

Создаем простой прямоугольный patch (**Quad Patch**) **1x1**, размером примерно равным ячейке каркаса над глазом (вид спереди). Сдвигаем его (на виде слева) примерно на уровень той ячейки, которую он должен закрывать.

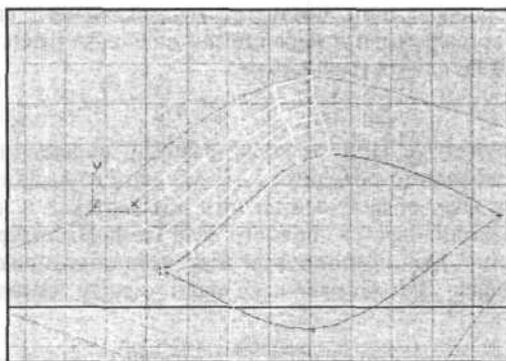
Если «прилипалка» еще не настроена, то это необходимо сделать сейчас.

Работаем в режиме **3D Snap** (прилипание к вертексам).

Накладываем модификатор **Edit Patch**.

**A**

**Sub-Object — Vertex.** Дополнительно запретим показ каркаса patch'a: сбрасываем флаг **Lattice** в группе **Display** раздела **Edit Vertex**. Включаем «прилипалку» (клавиша **S**). Беремся по очереди за каждый вертекс patch'a и «прилипаем» его к соответствующему вертексу характерных линий.

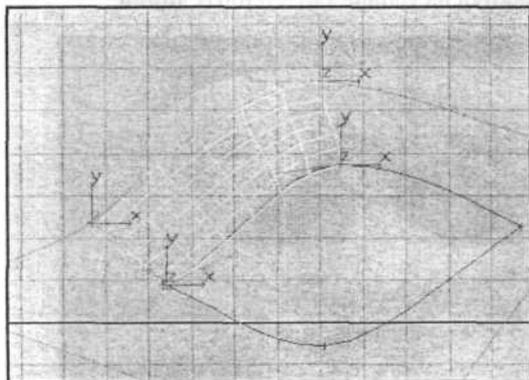


Б

Запрещаем прилипание (S) и «в черновую» регулируем форму patch'a тангентами. Общее правило: тангенты соседних вертексов (по общей стороне) не должны перекрываться в одной плоскости.

Также желательно, чтобы для соседних вертексов тангенты доходили до 1/3 стороны.

Окончательная регулировка формы поверхности тангентами осуществляется после всех «пристыковок» к данному и соседним с ним вертексам.



В

Устанавливаем Sub-Object Edge (ребро, сторона), выбираем одну из сторон (при отключенном Display Lattice просто перемещается трип-

лет координат, а при включенном, соответствующий элемент каркаса (**Lattice**) подсвечивается красным) и наращиваем еще один patch кнопкой **Add Quad** (или **Add Tri** для треугольного).

Г

Переходим к пункту А. При стыковке patch'ей, наращенных на различных ребрах, необходимо объединять (**Weld**) вертексы. Для этого при **Sub-Object Vertex** надо областью выделить оба вертекса и нажать кнопку **Weld**. Обычно увеличивается параметр **Weld Threshold** для гарантированного срабатывания. При объединении вертексов объединяются и соответствующие ребра, что исключает щели между соседними patch'ами.

Если на выделенном вертексе щелкнуть правой кнопкой мыши, появится горячее меню. В нем будут два пункта, отвечающие за тип вертекса: **Coplanar** и **Corner**. Эти типы вертексов patch'a соответствуют типам вертексов сплайна **Bezier** и **Bezier-Corner**.

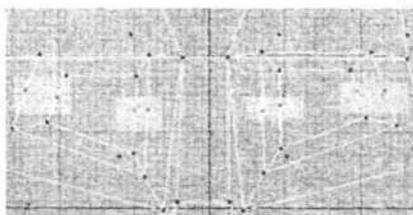
#### Шаг 4. Окончательная модель

Для этого установим **Pivot Point** в начало координат и отзеркалим созданный patch-объект как **Instance**.

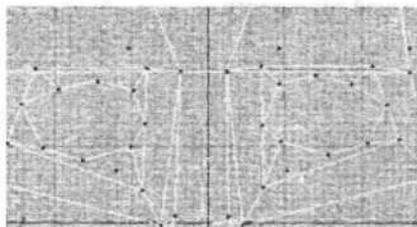
Теперь видны допущенные ошибки. На крае половинки маски (по центру) не точно выставлены тангенты: заметен шов. От внешнего уголка глаза идут «морщины» — надо дополнительно регулировать вертексы. Недостаточна проработка на уровне характерных линий.



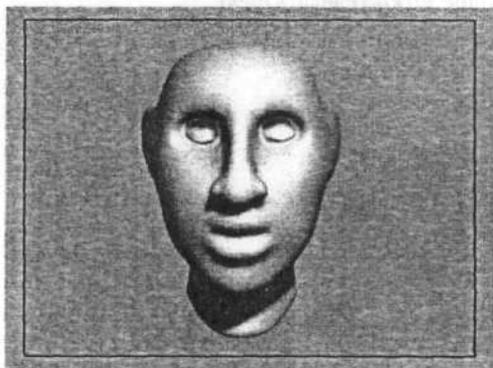
9. Делаем глаза. «Тонкое» место, привожу самый простой, но, возможно, не самый аккуратный путь решения: в режиме **Sub object: Edges** делим указанные ребра пополам. Затем удаляем внутренние полигоны. Подсвеченные вершины попарно объединяются (**Weld** ⇔ **Target**)



10. Также делим верхние и нижние границы глазниц. Двигаем вершины в нужном направлении.



11. Теперь пора делать **MeshSmooth**. Получается простая, но уже человекоподобная голова. Дальнейшая обработка по вкусу, рекомендуется текстурирование и выборочное дополнительное сглаживание.



6. После всего проделанного голова имеет примерно такой вид.

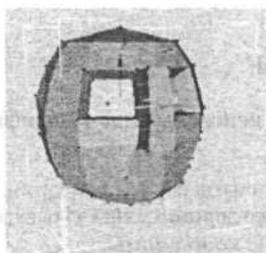


7. Можно сделать и уши. Как и с носом, с помощью выдавливания и возвращения передних точек на место. Оставшиеся боковые поверхности также выдавливаем для пушей лопухости.



8. На этом же этапе можно сделать шею.

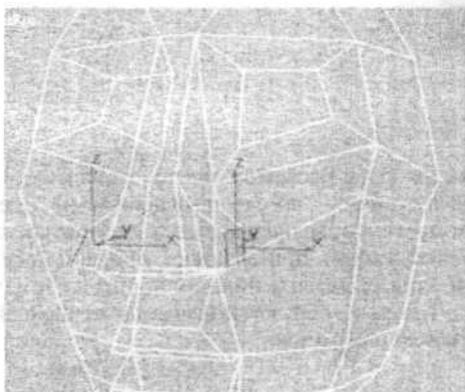




4. Так как в исходной сфере недостаточно вершин, дополнительные (особенно для верхней и нижней частей головы) можно получить, выдавливая «макушку» и «подбородок» и выравнивая полученные точки в соответствие с требуемой формой головы.



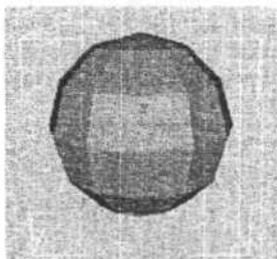
5. Ноздри выдавливаются в два приема из боковых поверхностей носа.



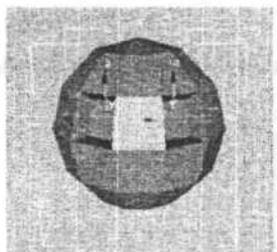
## Глава 15. Создание простой головы

Пример создания модели головы с использованием модификатора **MeshSmooth** (3ds max).

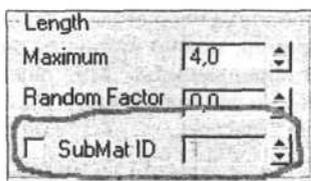
1. Начинается все со сферы: 12 сегментов без сглаживания. Сразу же применяем **Edit Mesh** и поворачиваем все вершины на 15 градусов, дабы в будущем иметь правильные локальные координаты.



2. Определимся с глазами. Для этого выделяем два соответствующих полигона и с помощью **Extrude** вдавливаем примерно так, как показано на рисунке:



3. Теперь заготовка для носа. Выделяем два полигона между глазами и выдавливаем **Extrude** ом (далее — просто «выдавливает»). Две верхние точки возвращаем на место и объединяем с соответствующими точками основания носа (**Weld** ⇌ **Target**). Двигая точки, придаем носу более человеческий вид. Можно также выдавить нижний полигон с тем, чтобы подчеркнуть кончик носа. Таким же образом получаем заготовки для губ: отдельно выдавливая нижнюю и верхнюю.



Именно здесь незадачливый программист заложил возможность присвоения текстуры параметру, объекту надо назначить мультиматериал, **diffuse** которых будет использоваться в качестве текстуры. Оттуда же можно вытащить и цвет волос.

Перейдите в **Material Editor** и измените тип материала дубинки на **Multi-SubObject**, сохранив сам материал дубинки. Проследите, чтобы объект имел на всей поверхности материал с **ID1**. Теперь прикинем — на **ID2** положим положим длину волос, на **ID3** — плотность, на **ID4** — материал. Все, что осталось сделать — это положить **gradient** на **diffuse** в **ID12** и настроить **ID3**. Обратите внимание, что волосам более присущ металлический блеск.

Осталось лишь подогнуть волосы. Для этого создайте **Lean Vectors** (**Create** ⇨ **Helper** ⇨ **ShagFur** ⇨ **Vector**) так, чтобы максимальная зона влияния покрыла треть дубинки, а минимальная с запасом покрывала все целиком. Назначьте этот вектор в качестве **Lean Direction** и добавьте немного **Bend**. Отрегулировав параметры, стоит все же сконвертировать свет в **Hair Enabled** (это делается в модуле **Shag: Render/Tools**) и посмотреть результат.



**Sub-Material** в качестве параметров также существенно облегчают работу и улучшают результаты — например, можно назначить меппинг **Noise** и на объекте частично проступят «проплешины». Материал волос, длину, плотность распыления — все можно регулировать.



У **Shag: Fur** куча применений. Не считая тривиальных — волосы и шерсть — это трава, листья деревьев, всевозможная растительность, оригинальные эффекты (вместо обычного **bump** можно сделать что-то действительно объемное), при желании с его помощью можно даже моделировать какие-то объекты (например, когти у злобного прищельца получатся очень интересными, если их сделать не монолитной геометрией, а из жесткой лоснящейся шетины).

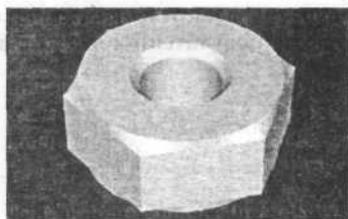
К сожалению, эти эффекты не так легко даются. Основные концепции (особенно векторы) даются относительно сложно для понимания, а если учесть, что хотя **Shag: Fur** и очень быстрый, но простая сцена рендерится все равно долго (где-то минуты 2-3), то на обучение уходит где-то с пол-дня. Тьюториалы, идущие с **plug-in**, достаточно примитивны и не показывают самой сути.

Можно сказать, что в целом **plug-in** получился очень хорошим, и был бы просто отличным, если бы не некоторая сложность в обучении и использовании. **Plug-in** можно обещать долгую жизнь, хорошую продаваемость и множество вариантов применений.

### Волосатая дубинка

Предположим, у нас есть дубинка, которую вы хотите сделать волосатой, чтобы «дубинистый» конец был просто лохматым и темным, а «недубинистый» — светлым и почти бритым.

Загрузим **plug-in Shag: Fur** и назначим в качестве **furry object** нашу неудавшуюся палицу. Естественно — будем «шерстить» целиком. Но сначала остановитесь и внимательно взгляните на изображение. Особенно стоит задуматься над сочетанием, показанным ниже.



## Глава 14. Создание волос с применением plug-in Shag: Fur

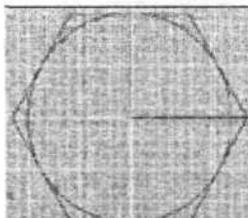
Основной отличительной чертой этого «генератора волос» является его скорость. Дело в том, что он не генерирует вообще никакой геометрии. Без него волосы обычно делаются либо созданием **вручную/полуавтоматически (Compound Scatter)** геометрии волос, либо хитрыми системами партиклов. Первое — достаточно сложный и трудоемкий процесс, создающий полумиллионифейсные сцены. А хитрая система систем партиклов плоха тем, что практически невозможно выполнить анимацию полученной модели — партиклы движутся, и остановить их без дополнительных plug-in'ов непросто. Да и сами партиклы тоже достаточно прожорливы в смысле железа.

**Shag: Fur** делает свои волосы не за счет геометрии, а как «атмосферный эффект», примерно так же, как **combustion**. Используя специальные **Hair-Enabled Lights** (свои собственные источники освещения), этот plug-in достигает даже эффекта тени от волос на объекте.

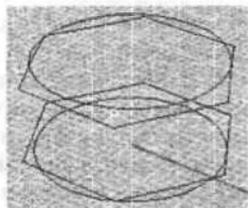


С помощью использования специальных векторов возможна настройка таких сложных опций, как подгибание волос — ведь не у всех волосы стоят торчком. Этими же векторами можно уложить волосы так, чтобы у шерсти было направление — то есть, «по шерсти» и «против шерсти». И все это можно анимировать.

Теперь постройте многоугольник, указав тот же центр, что и у окружности, и тот же радиус. У вас должно получиться то, что показано на рисунке.



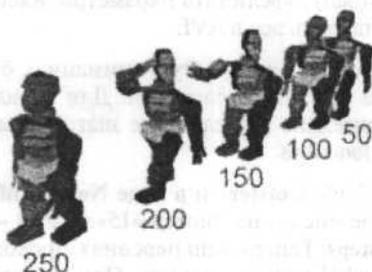
Теперь надо скопировать (**Copy**) оба объекта и разместить их друг над другом, так, как показано на рисунке.



Теперь все готово к лофту.

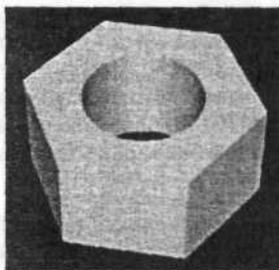
При желании можно вставить еще несколько окружностей для внутренней части гайки и фасок, но сейчас для нас это несущественно, поэтому не будем на этом останавливаться. Выбираем **Loft** и выделяем окружность, шестиугольник, шестиугольник и окружность в порядке снизу вверх или сверху вниз. Для правильной подгонки линий связи проще всего использовать автоматику (**auto**). Теперь нам надо не плавную поверхность, а строгую, четкую и техничную, так что выбираем **Style: Straight Sections**. Все, лофтим.

Теперь, если хочется, то **Cap Planar Holes** или переделайте все с окружностями и **Closed Loft**'ом. Вот и результат.



## Глава 13. Создание правильной гайки с помощью лофта Rhino

Этот tutorial совсем не сложен. Он скорее направлен даже не на получение результата, а на получение некоего убеждения в мощности инструмента Rhino3D, имя которому Loft. Мы будем рисовать гайку. Но это случайно не экструднутый сплайн, как на рисунке?



Приступим к рисованию правильной гайки.

Настоящая гайка — это лофт двух вещей — окружностей и описанных вокруг них шестиугольников. Начнем с окружности. С помощью простейшего инструмента **Circle** нарисуем окружность. Теперь выберем инструмент **Polygon** и введем следующие коррекции с клавиатуры:

- ◆ **Num** — количество сторон;
- ◆ **6** — у нас шестиугольник;
- ◆ **Circ** — наш шестиугольник.

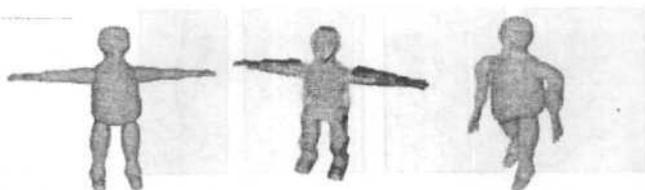
параметрах (во вкладке **Modify**) увеличить параметры **Radial scale**, **parent** и **child overlap**. Можно рендерить все в **AVI**.

Подведем итоги. Мы создали простую анимацию, базирующуюся на шагах. Теперь давайте ее слегка усложним. Для начала перейдем в **Footstep mode** и, предварительно выделив все шаги, удалим их одним кликом по кнопке **Delete footsteps**.

Кликаем **Create multiple footsteps** и в поле **Number of footsteps** вводим какое-нибудь крупное число, например «15». Далее — как обычно: **create keys for inactive footsteps**. Теперь наш персонаж проходит в несколько раз большее расстояние! Но это не главное. При создании чуть более сложных движений, чем ходьба, **Mesh** персонажа будет нам мешать. Чтобы избежать этого, выделите его и во вкладке **Display**, кликните **Hide selected**. Воспроизведите анимацию. Ходьба есть. Допустим, на определенном этапе ходьбы он повернет голову налево и отсалютует. **Пацифисты** могут заставить персонажа просто почесать голову.

Приступим к анимации. Прокрутите **Time slider** примерно до того момента, когда скелет (вы же спрятали **Mesh**, не так ли?) будет в районе пятого **footstep** — допустим, кадр 84. Выделите голову скелета (как-то не хочется называть это черепом). По временной шкале видим, что **keyframe** у головы присутствуют в момент каждого шага. Давайте при помощи того же **Time slider** а посмотрим, где нам желательно закончить поворот головы, то есть повернуть ее назад. Кадр 174. Выделяем и удаляем все **keyframe** головы между этими кадрами. Теперь нажмем кнопку **Animate** и, отсчитав примерно полсекунды от начала поворота (пусть будет кадр 100), поворачиваем голову примерно на 50° влево. При помощи клавиши «**Shift**» копируем этот ключевой кадр в кадр 160. Воспроизводим анимацию. Теперь делаем нечто подобное с правой рукой, но, кроме поворота, придется воспользоваться перемещением.

Чтобы сделать человечку кепку/фуражку/колпак, создайте этот объект и просто «прилинкуйте» (**select and link**) его к голове (к **Mesh** линковать бесполезно). Далее все просто — **Display** ⇌ **Unhide All**; **Modify** ⇌ **Physique** ⇌ **Hide attached nodes**; **Rendering** ⇌ **Render**. Кстати, добавив какой-нибудь фон, поставив в направлении поворота головы персонажа фотографию любимого прапорщика и добавив музыку из «**Army Men**», можно сделать интересный клип.

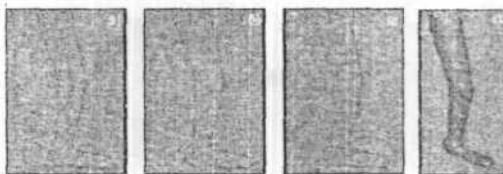


Создадим для него скелет. На панели **Create** нажимаем клавишу **Systems**, далее кликаем на **Biped** и растягиваем прямоугольник так, чтобы его высота соответствовала росту нашего персонажа. Во вкладке **Create** задаем необходимое количество пальцев, хвостов и так далее. Чтобы вышеупомянутый персонаж не мешался, выделяем его и во вкладке **Display**, щелкаем **Freeze Selected**. Выделяем какую-либо часть скелета и переходим во вкладку **Motion**. Кликом по кнопке с человечком переходим в режим **Figure mode**. В этом режиме мы «подгоняем» скелет под размеры персонажа. Выделите центр тяжести (такой ромб внутри таза скелета), переместите его туда, где, собственно, у вашего персонажа таз. Кстати, этот самый центр тяжести удобно будет «забить» в **Named Selections**. Мы будем довольно часто к нему обращаться.

Еще один нюанс при подгонке таких частей тела, как пальцы и хвосты: будет лучше, если их размеры будут превышать размеры пальцев персонажа, то есть слегка «выпирать». Можно выйти из **Figure mode**.

Приступаем к самой ответственной части работы. Выделяем нашего персонажа и применяем к нему модификатор **Physique**. Кликнув по кнопке **Attach to Node**, а затем по нашему центру тяжести (иногда это легче бывает сделать через **Select by Name**), «надеваем» персонажа на скелет. Здесь лучше сохранить свою работу. Делаем небольшую проверку: кликаем по кнопке с изображенными на ней ступнями (**Footstep mode**), а затем — **Create Multiple Footsteps** ⇨ **Ok**. Последний штрих — **Create keys for inactive footsteps**. Кстати, чтобы спрятать скелет от посторонних глаз, выделяем тело персонажа и в параметрах модификатора **Physique** отмечаем галочкой **Hide attached Nodes**.

Если персонаж толстоват, а его руки при ходьбе задевают туловище — вернитесь в **Figure mode** и расширьте плечи персонажа. При правильном размещении «костей», «отстающих» от тела точек быть не должно. Если таковые появились (особенно часто подобное случается в районе пальцев), существует два пути для исправления. Первый — самый простой, но не всегда эффективный — вернуться в **Figure mode** и «вправить» все неправильно размещенные «кости». Второй путь — выделить тело персонажа и перейти к модификатору **Physique** на уровне подобъекта **Envelopes**. Далее необходимо кликнуть на «проблемной» **Envelope** и в



Теперь при помощи **Move**, **Rotate** и **Uniform scale** размещаем полу-  
 круги по всей длине ноги. Если полуциркуль не хватает, создаем копии.

Последний полуциркуль, находящийся на самом конце ступни, уменьшите до каких-нибудь ничтожных размеров — так, чтобы его даже не было видно, иначе нога превратится в трубу. Большая часть работы сделана. Удалим контур (если он сделан из **Line**) — он нам больше не понадобится. Выделяем первый полуциркуль и на панели **Modify** ищем кнопку **Attach**. Кликаем сначала на ней, затем последовательно на всех элементах нашего каркаса. Не забудьте и о практически невидимом последнем полуциркулье!

Дальше все просто. К полученному большому сплайну применяем модификатор **CrossSection**, в параметрах которого устанавливаем **Smooth** вместо **Linear**. Применяем модификатор **Surface** и получаем «обтянутый» каркас.

Если поверхность «вывернута наизнанку» — отмечаем галочкой **flip normals**. Параметр **Patch Steps** настройте по своему усмотрению. Теперь полученную поверхность конвертируем в **Editable Mesh**.

Кликнув на кнопку **Mirror selected objects** на главной панели, выбираем соответствующую ось, в **Clone Selection** — **copy**. Получаем отличную зеркальную копию нашей половинчатой ноги. Прикрепить ее ко второй половине можно с помощью кнопки **Attach** (кстати, таким же способом эту ногу можно прикрепить и к туловищу).

Переходим к подобъектам **vertex** и в проекции **Front**, выделяем все **vertex**ы, расположенные вертикально по середине ноги. Кликом по кнопке **Weld Selected** добиваемся полного слияния половинок ноги (значение счетчика после кнопки **Selected**, скорее всего, придется увеличить). К готовой ноге примените модификатор **Optimize** или **MultiRes** — всегда нужно стараться сократить количество полигонов в сцене.

Таким же образом создаются туловище, руки и голова. Но перейдем к анимации. Допустим, у нас есть вот такой персонаж.

## Глава 12.

### Создание 3D-живности

3ds max имеет огромное количество встроенных возможностей, открывая тем самым огромные просторы для аниматора. Эти просторы расширяются при использовании внешних модулей, один из них — **Character Studio**.

3ds max предоставляет возможность использовать для анимации живых персонажей «bones», но возиться с расстановкой «костей» и IK-solver слишком утомительно. В **Character Studio** готовый «скелет» с заданным числом пальцев на руках и ногах, фаланг этих самых пальцев, хвостов и косичек появится в два щелчка мыши. Остается лишь подогнать его размеры под персонажа.

Чтобы анимировать персонажа, его сначала нужно создать. Сформируем небольшого человека, а затем — короткую анимацию с ним в главной роли.

Приведем лишь вкратце два основных метода создания персонажей.

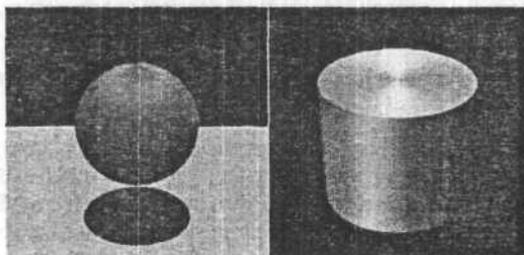
**Первый метод:** создать box с достаточным количеством height, width и length segments, затем преобразовать его в редактируемый mesh (щелчок правой кнопкой по объекту, **Convert to** ⇨ **Convert to Editable Mesh**). Далее параллелепипед редактируется на уровне подобъектов (vertex, edge, face, polygon).

**Второй метод:** создание персонажа при помощи модификатора **Surface**. Опишем этот процесс пошагово, на примере моделирования ноги. Для начала начертите при помощи объекта **Line** контур ноги в боковой проекции или, если у вас есть этот контур в графическом файле, задайте **Viewport background**.

Этот контур будет для нас подсказкой при рисовании ноги. Теперь в проекции **Top** чертим круг (**Create** ⇨ **Shapes** ⇨ **Circle**) и при помощи **Uniform scale** подгоняем его диаметр под диаметр «ноги» в самой верхней точке. Нам нужно отсечь половину круга. Для этого конвертируем его в **Editable Spline**, на уровне подобъекта **Segment** выделяем половину круга и нажимаем магическую клавишу «Del». При нажатой клавише «Shift» сдвигаем полученный полукруг вниз и в поле **Number of copies** вводим «14». Можно больше, можно меньше. Скорее всего, по ходу работы мы сделаем еще несколько копий.

Решив проблему с рассеиванием (diffuse), отражающий блик (specular highlight) все же отбрасывает яркий свет. Например, если установить свет за сферой и назначить hair shader, то все еще остается отражающий блик на поверхности, даже если свет на нее не направляется. Новая версия hair shader-а предоставляет пользователю некоторый контроль над величиной полупрозрачности (translucency), но в общем, этот шейдер рекомендован только для волос (ворса, ниток), а не для геометрии.

Стандартный анизотропный шейдер в Renderman (standard ward anisotropic renderman shader) работает отлично, если вы пользуетесь PRMan или BMRT, переводя результат в 3ds max 7.0 с помощью **Animal Logic's Maxman translator**. Может получиться неплохой блик даже на вершине (середине) цилиндра (в случае использования nurbs цилиндра вместо стандартной фигуры в 3ds max). Вот два изображения, просчитанные в BMRT с применением ward anisotropic shader.



Последнее фото «реального мира», составляющий пару для рождественского орнамента, представленного выше, на этот раз конической формы. Обратите внимание на ровное отражение источника света на шарике внизу конуса, на гладкой поверхности, как шар, вы получаете простое отражение, но на анизотропной поверхности, такой как конус, вы получаете совсем другой тип отражения света.

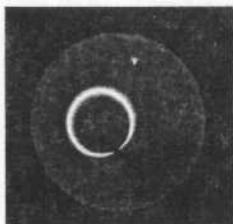


Это из-за того, что расположение светового блика не верно. Изменение ориентации (Orientation) с 0 на 90 сменит направление «несуществующих» неровностей, следовательно, изменит вид блика.

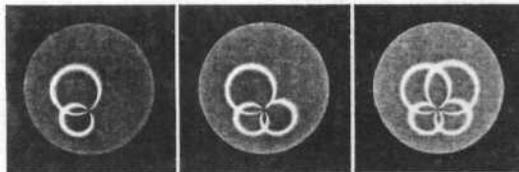


Но это подходит не для всех ситуаций. Например, для верхней части цилиндра, нет способов установить как вам нужно световой блик для выпуклостей, идущих по круговому узору (circular pattern), это означает, что вы не сможете сделать компакт диск. Бликом тяжело управлять, например, очень трудно получить хороший тонкий размытый блик, который полностью сходится в полный круг на вершине объекта.

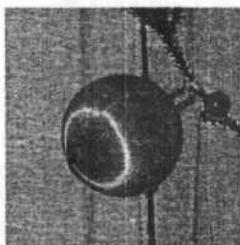
Другой анизотропный шейдер поставляется с Digimation Shag:Hair. Шейдер Hair, имеет неплохой отражающий компонент.



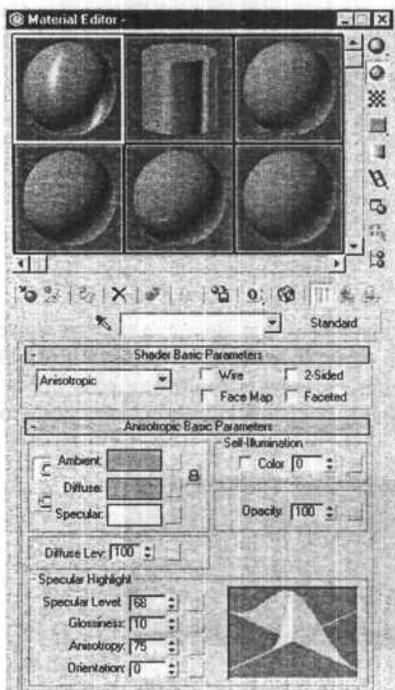
Однако так как он предназначен для волос, а не для геометрии, у него существует компонента «просвечивания», «опустошающая» свет. Например, добавление большего количества источников света в вашу сцену начнет все ярче высвечивать рассеивающую составляющую (diffuse component).

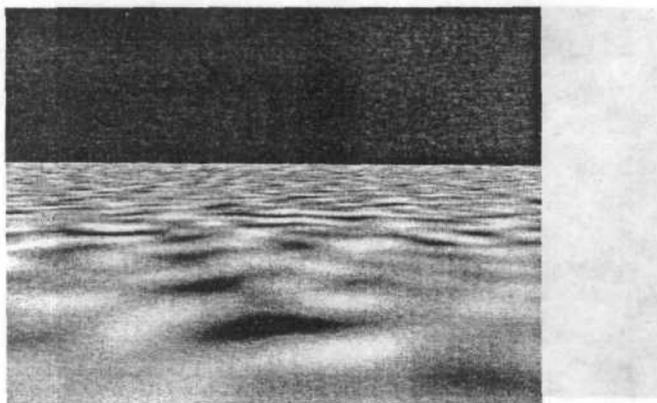


блик-шейдер смешанный дополнительно с рассеивающим Ламберт-шейдером (Lambertian diffuse shader). К сожалению, анизотропный шейдер имеет некоторые проблемы; к примеру, возьмем этот образец из реального мира — рождественское украшение, сделанное из тонких синтетических волокон, натянутых в одном направлении.



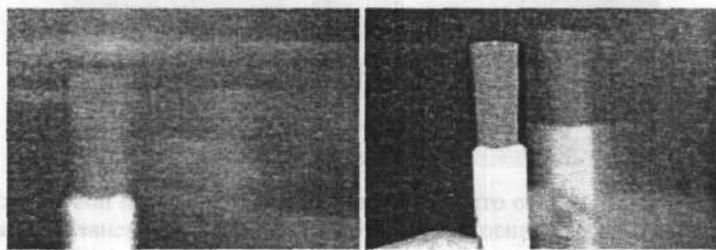
Итак, сфера с анизотропным бликом должна выглядеть примерно также, однако в 3ds max 7.0 не так.





Это тот же однородный нойз на поверхности плоскости. Обратите внимание, ближний нойз выглядит больше, и по мере удаления он становится меньше и узор становится более «плотным». Так и отражения будут искажаться по-разному, отдаляясь, отражения переходят в неровности (bump pattern).

Вот похожий феномен, рассматривающий расстояние вместо угла обзора. Заметьте, ручка и моя рука, отраженные поверхностью (холодильника) очень размыты на расстоянии более 30 см, и не совсем размыты по мере приближения к поверхности.



### Анизотропные шейдеры для точечных источников света

В 3ds max включен анизотропный шейдер. Что это такое? Как обсуждалось ранее, световые блики это лишь отражения источников света. А анизотропный блик это лишь анизотропное отражение, но которое работает для точечных источников света, а не для реальных отражений. Анизотропный шейдер 3ds max 7.0 похож на подопечный анизотропный



А вот фото, сделанное в приближении к отражению:



Заметьте, что отражение размыто неровностями поверхности дороги, но эффект анизотропии исчез. Этот эффект изменяется в зависимости от расстояния, а также угла. Например, рассмотрите этот шум (noise), назначенный плоскости:

Вот свет стоп-фар.



А чтобы вы не думали, что этот эффект присущ лишь источникам света, вот размытое отражение деревьев:



Рассмотрите эти две фотографии. Если вы исследуете их внимательно, заметите, чем дальше вы от отражения, тем больше оно размыто. Вот свет фар машины:

рисованные карты (painted maps), подходящие для этого. Так, например, для верхней части цилиндра вы можете нарисовать карту, с царапинами, исходящими в круговом направлении, а затем наложить ее на верхнюю часть цилиндра. Цилиндр будет выглядеть как CD.

Хотя эта техника работает, она требует много точных настроек для получения необходимого размера и частоты неровностей, чтобы изготовить нужный вам тип искажения. Используя отдельный процесс для деформирования отражений из неровностей, вы получаете больше контроля, так как можете управлять величинами выпуклости и растяжения независимо друг от друга.

### Отражения при дожде

Первый снимок — это несколько машин, остановившихся на светофоре. Обратите внимание, как свет их фар размыт в воде, которая, в свою очередь, искажается капающим дождем, а также неровной поверхностью дороги.



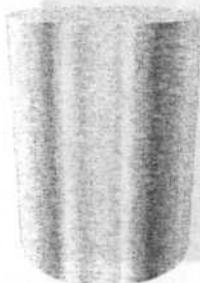
Заметьте также, как в итоге прекращается каждое отражение. Как указывалось ранее, отражение на каждой неровности существует слегка под разным углом, так в результате, угол с которого неровность получает отражение настолько мал, что неровности перестают отражать, что и приводит к прекращению отражения. Величина «беспорядочности» на поверхности, так же как и угол вашего обзора, и яркость отражаемого объекта, все это влияет на то, как выглядит отражение.

### Анизотропные отражения или мелкие неровности

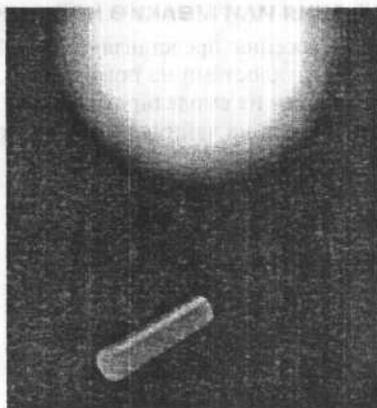
Поскольку анизотропные отражения представляют собой лишь отражения, искаженные мелкими выпуклостями на поверхности, почему бы не сделать эти мелкие выпуклости на смоделированной нами поверхности? Можно, взгляните, например, на чайник, который отрендерен в **Ghost**.



Для создания этой картинки использован «обман с неровностями». Как это делалось: сначала была создана гладкая отражающая хромированная поверхность. Затем была добавлена очень смягченная, маленькая, слегка вытянутая карта неровности (*bump map*), деформирующая отражение, преимущественно вдоль одной оси.

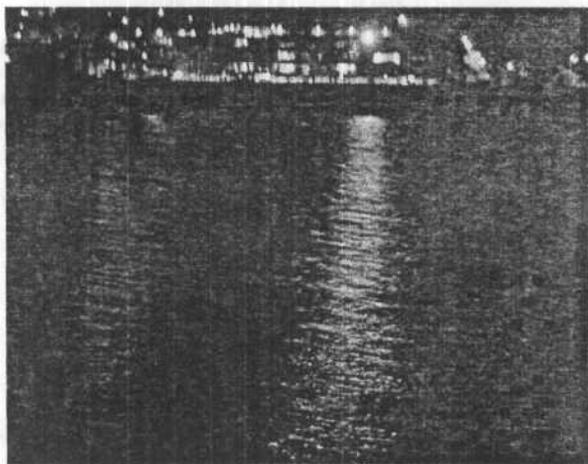


Вот материал, который был использован для чайника на обычном цилиндре. Вы не обязаны использовать процедурную карту шумов (*noise procedural map*) для неровностей (*bump*), вы можете также использовать



Заметьте, как ряд меньших отражений, образует теперь одно большое длинное и растянутое отражение-блик. Цч

Другой пример для рассмотрения — это отражения на воде в ночное время.



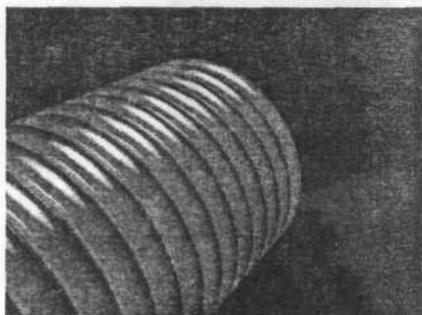
Простой круглый источник света отражается как длинный вытянутый световой блик, из-за ряби на воде, рябь, которая выполняет функцию похожую на функцию спиралей дверной пружины, или канавок металла с шероховатостями, только в увеличенном масштабе.

видели единственное отражение, но поместите рядом второе кольцо, получите два отражения рядом друг с другом. Но это не будет совсем одним и тем же отражением, так как поверхности располагаются слегка под разными углами к предмету, который они отражают. Расположим связку таких колец вместе, и отдалимся, до тех пор, пока совсем не будут видны выпуклости, заметьте, как отражение совершенного кругообразного источника света, теперь выглядит как отражение длинного, продолговатого источника света.

Вот пример этого же феномена, смоделированного компьютерным способом. Это одно кольцо, отражающее источник света, расположенный сверху.



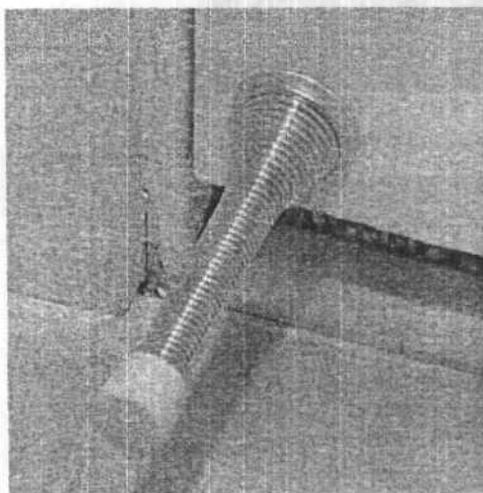
И несколько колец, выстроенных в ряд.



А вот этот же пример, но с расстояния подальше.

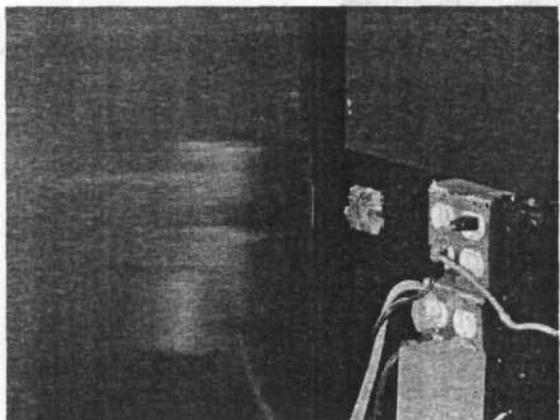
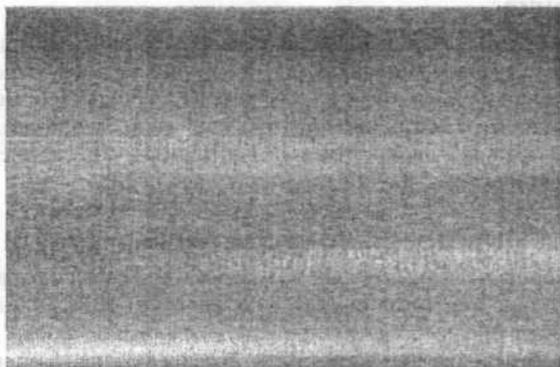


Вот, например, источник света и его отражение на дверной пружине.



Если мы посмотрим на дверную пружину ближе, можно представить ее как набор кругообразных поверхностей, каждая из которых отражает свет, исходящий сверху. Если бы было только одно кольцо, вы бы

Возьмем эти два примера реальных шероховатых металлов.



Первый рисунок — это приближенная рельса, пример, где вы можете увидеть некоторую мелкую зернистость, идущую в одном направлении. Второй рисунок — это фото дверцы холодильника, поверхность которой изготовлена из похожего материала, и сделано это фото с большого расстояния. Обратите внимание, как отражения розеток и проводов размываются в определенном направлении вместо равномерного размывания во всех направлениях.

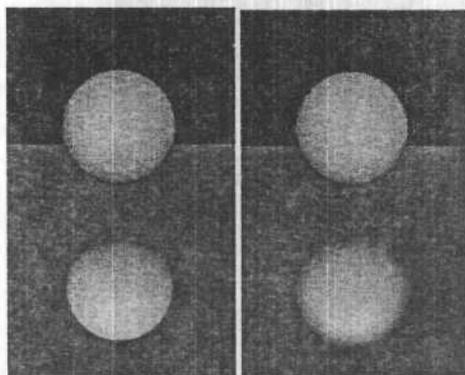
Не вдаваясь в технические подробности, опишем происходящее: вместо того, чтобы видеть одно отражение на поверхности, вы видите отражения, повторяющиеся многократно на неровных участках поверхности, и отражения эти смешиваются, образуя деформированное отражение.

## Типы объектов, имеющих анизотропные отражения

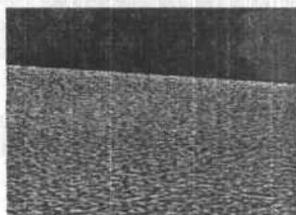
Это все, объекты, которые имеют мелкую зернистость, идущую, преимущественно, в одном направлении. Хороший пример — волосы; металлы, обработанные щеткой, так называемые шероховатые металлы (brushed metals), или отражения на взволнованной воде (например, капающим дождем). Компакт-диски, тоже анизотропные, отражение искажается мелкими канавками, идущими по кругу вдоль CD.

## Причина анизотропного отражения

Мелкие неровности на поверхности делают отражения равномерно размытыми (если неровности умеренно неравномерны). Если эти мелкие выпуклости образуют «структурный узор», тогда «узор» может влиять на то, как выглядит отражение. Вот две сферы, первая с четким отражением, вторая — с размытым.



Если бы это было фото «реального мира», размытое отражение было бы вызвано мельчайшими хаотическими неровностями на поверхности пола, неровности, возможно, настолько малы, что вы можете их не заметить, но вы увидите, как они влияют на отражение.



зать края (там тоже получилась интересная текстура — текстура «расщепленного дерева» — если нужна — сохраните ее в отдельный файл), и затайлить все.

## Глава 11.

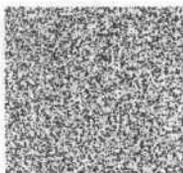
### Анизотропные отражения

Многие видели некоторые современные картинки, отрендеренные в Brazil-e, на них заметен тип шейдера, который в 3ds max 7.0 называется анизотропным (anisotropic), все мы сталкиваемся с эффектом анизотропии ежедневно, даже не подозревая, что мы его видим.

Анизотропные — anisotropic — неравные физические свойства (предмета) вдоль различных осей.

Анизотропные отражения аналогичны обычным за исключением того, что они растянуты или размыты, на основе расположения мелких неровностей (выпуклостей, волокон или царапин), существующих на отражающей поверхности. Например: пол с анизотропным отражением, и как видите, чайник размыт по оси Y, но не совсем размыт по оси X.





Теперь делаем **Filter** ⇨ **Blur** ⇨ **Motion Blur** с расстоянием в 30-40 пикселей. Получились полосы, похожие на текстуру дерева, только очень размытые.



Делаем **Filter** ⇨ **Sharpen** ⇨ **Sharpen** для выделения текстуры дерева. Все — теперь осталось только раскрасить ее.



Раскрашиваем: **Image** ⇨ **Adjust** ⇨ **Hue-Saturation**, со включенной опцией **Colorize**, ориентировочные параметры:

Hue - 35

Saturation - 60

Lightness - +20



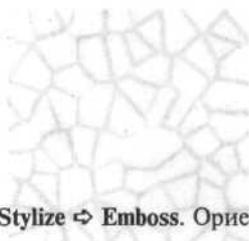
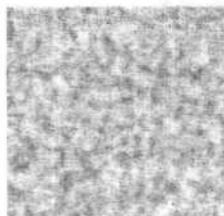
Получается что-то похожее на полированный паркет. Параметры дерева легко меняются, как параметры окраски. Осталось только отре-

### Каменистая поверхность (bump, diffuse)

Создайте новый файл. Начнем с шума — традиционно, **Filter ⇨ Noise ⇨ Add Noise, Monochromatic**.



Теперь размытие — **Filter ⇨ Blur ⇨ Gaussian Blur**, коэффициент — 1-2.

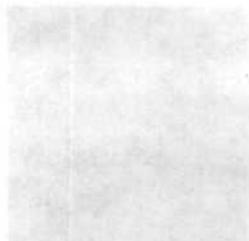
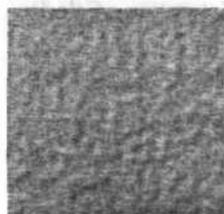


И, наконец, выпуклости — **Filter ⇨ Stylize ⇨ Emboss**. Ориентировочные параметры:

Height=3

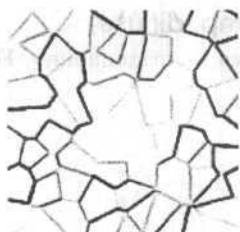
Amount=100%

Ну и для диффуза — раскрасить.

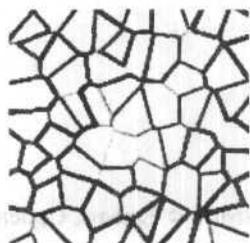


### Дерево (diffuse)

Создаем новую картинку, и, как всегда, добавляем шум, лучше не монохроматичный, — **Filter ⇨ Noise ⇨ Add Noise**.

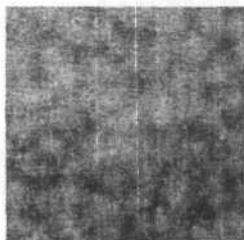


Ну вот практически и все. Осталось лишь сделать все расщелины более-менее равномерными. Сделайте **Image** ⇨ **Adjust** ⇨ **Levels**. Параметры такие: **Input Levels** — 194 1.00 255, а все остальное оставьте, как есть. Можно использовать как **bump** или раскрасить текстурой, как **diffuse**. Одно «но» — эту текстуру очень сложно затайлить стандартными методами, так что советуем делать сразу столько, сколько нужно (пусть одна будет большая, типа 2000 на 2000, но чтобы не тайлилась).



### Пузырьки (bump, diffuse)

Возьмите текстуру из предыдущего способа и сделайте **Filter** ⇨ **Blur** ⇨ **Gaussian Blur** с параметром 4. Для **bump** сойдет и так, для диффуза раскрасьте как надо через **Image** ⇨ **Adjust** ⇨ **Hue-Saturation** с **Colorize**.

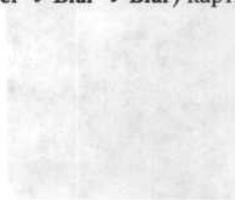


### Расщелины (bump, diffuse)

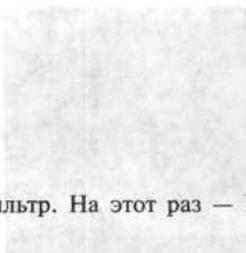
Откройте новый файл, скажем, 150 на 150. Возьмите два первоначальных цвета (черный и белый — кнопка «D») и примените **Filter** ⇨ **Noise** ⇨ **Add Noise** с параметром **Monochromatic** и уровнем около 75.



Теперь сделайте **Image** ⇨ **Adjust** ⇨ **Brightness-Contrast** и увеличьте контраст до +50 или +55. Этот параметр подбирается на глаз, так, чтобы общий суммарный цвет всех точек примерно был на 50% серым. После этого нужно еще немножко разблурить (**Filter** ⇨ **Blur** ⇨ **Blur**) картинку.



Все это теперь надо превратить в расщелины. Применим фильтр **Crystalize** со средним уровнем «кристаллизации» (например, 25).



Осталось немного — еще один фильтр. На этот раз — **Filter** ⇨ **Stylize** ⇨ **Find Edges**.

## Глава 10. Использование Photoshop для создания текстур

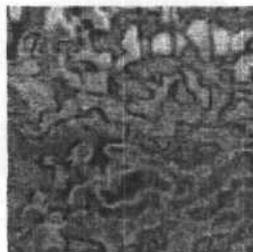
При работе с любой 3D-программой часто встает вопрос о текстуровании объектов — в простейшем случае создании плоских изображений, которое бы проектировалось на нужный объект. Опишем несколько приемов, используемых для получения разных текстур в Photoshop.

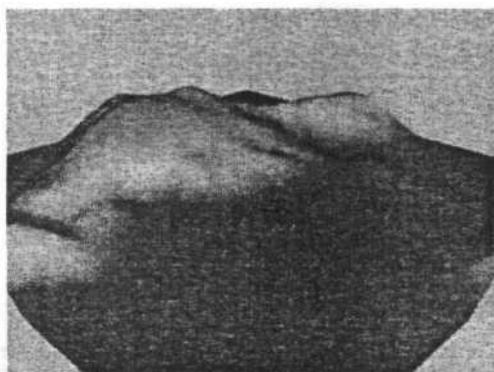
### Помятость (bump)

Откроем новый файл и сделаем в нем черно-белые облака. Для этого **reset**-ьте цвета (кнопка «D») и сделайте **Filter** ⇨ **Render** ⇨ **Clouds**.



Теперь применим фильтр **Distort** ⇨ **Ocean Ripple** со средним размером ряби и высокой силой. Увеличьте контраст изображения на 40-50 единиц (**Image** ⇨ **Adjust** ⇨ **Brightness-Contrast**) — и перед вами хорошая **bump**-карта для накладывания, например, для изображений помятости ткани или сморщенности.





А теперь осталось исправить цвета. Вместо синего — траву, вместо зеленого — камень, а красный заменить на снежок.



Метод можно расширять и усложнять.

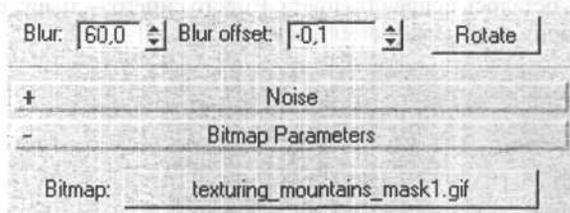
Вместо трех материалов создать четыре, пять или даже больше, если надо отобразить различные скальные породы или просто разнообразить путь к вершине.

Вох можно не только постепенно поднимать, но и наклонять на небольшой градус, чтобы граница не выглядела слишком горизонтальной.

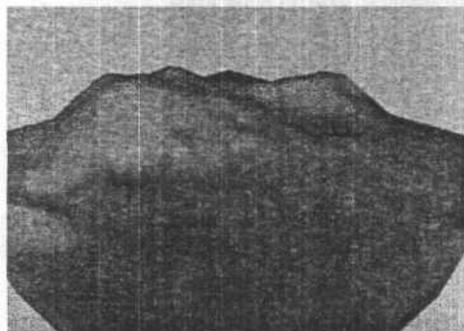
Вспомогательные материалы (у нас в примере это белый и черный) можно сделать светло-серым и темно-серым соответственно, или добавить какую-нибудь текстуру в слот **diffuse** и должным образом уменьшить интенсивность **self-illumination**, тогда материалы будут просвечивать сквозь друг друга в самых неожиданных местах.



Не стоит делать этого в Photoshop. В 3ds max'e получится не хуже. Заходим в слот **Mask** и в параметрах **Bitmap**'а изменяем значение **Blur**.



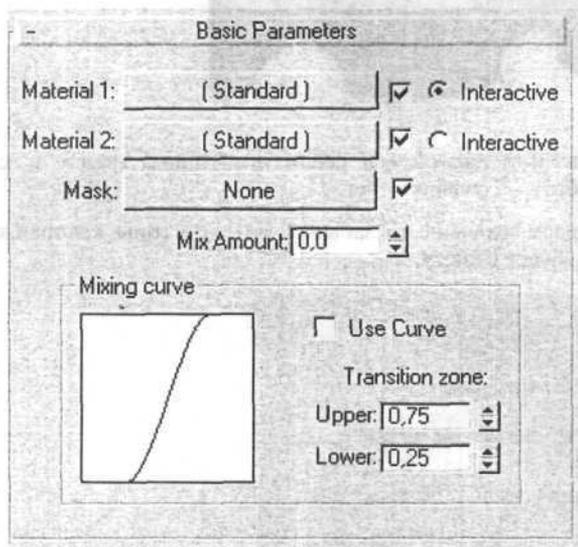
При значении 60 картинка заметно похорошела.



Помнится, еще был обещан снежок. Так вот, чтобы он выпал, надо поработать. Заходим в **Material 1** и меняем его тип со **Standart**'а на **Blend**. 3ds max предлагает сохранить нынешний красный материал как суб-материал — соглашайтесь. А дальше та же история, слоту **Mask** назначаем вторую маску, разблюриваем ее известным способом, в слоте **Material 2** создаем материал под цвет крокодила Гены. Рендерим.

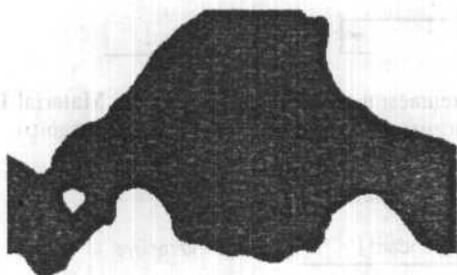


Первую маску помещаем в слот **Mask** как **Bitmap**, **Material 1** делаем красным, **Material 2** синим. Накладываем материал, рендерим.



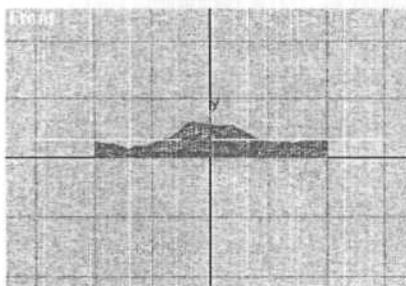
Сложно предположить, что 3ds max 7.0 скажет, после того как вы нажмете на кнопку **Render**. Во-первых, он может хмуро заявить, что на вашем **Box**, то есть — горе, нет **mapping**, тогда через панель **Modify** наложите **mapping** с помощью **UVW-Mapping** модификатора. Во-вторых, текстура может неправильно отображаться, тогда следует пойти в тот же модификатор **UVW-Mapping** и подкрутить **Gizmo**, на **Sub-Level'e**.

Резкая граница смотрится весьма неестественно, неплохо бы ее разблурить.



С полученной картой мы реализуем только травку и камни. Продолжим работу – создадим снег.

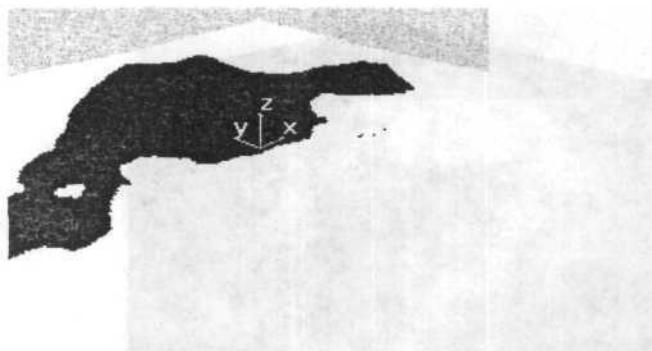
Перемещаем **box** выше, не забывая, что часть горы, которая возвышается над **box**, будет в снегу.



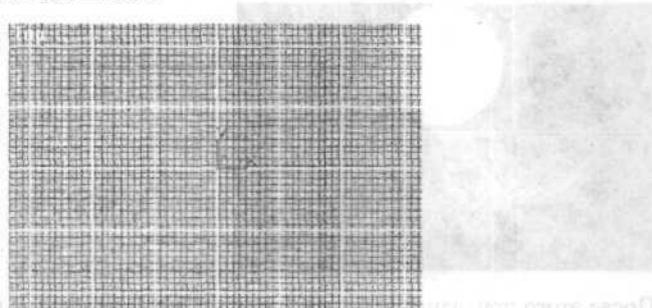
Когда подходящий уровень найден, снова напрягаем CPU рендером.



Теперь у нас есть все необходимое, чтобы сделать красивую текстуру. Идем в **Material Editor**. Сначала изменим тип материала со **Standart'a** на **Blend**.



Переходим в вид **Тор**, добиваемся с помощью **zoom**, чтобы гора занимала весь вид по ширине. Поскольку полученной картинкой в дальнейшем мы собираемся воспользоваться для наложения текстур на горы, то желательно, чтобы стороны картинке были пропорциональны сторонам гор. Если не отрегулировать вид, то потом предстоит немного поработать в Photoshop, урезая картинку до нужного размера. Рендерим в разрешении 300 на 300.

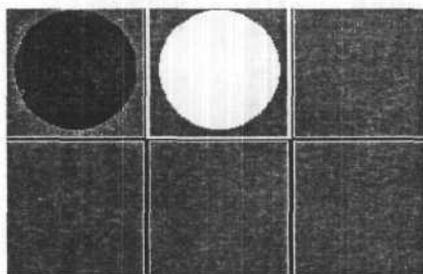


Это наша первая карта.

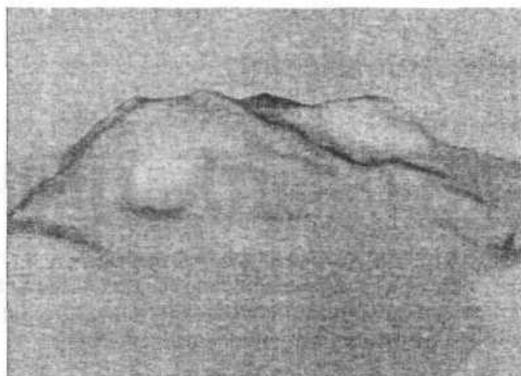


Это делается для того, чтобы создать карты для масок и **blend**.  
Перед этим создадим еще пару материалов.

Один материал совершенно черный и самосветящийся на 100%,  
второй материал настолько же бел, насколько черен первый и светится с  
не меньшей силой.



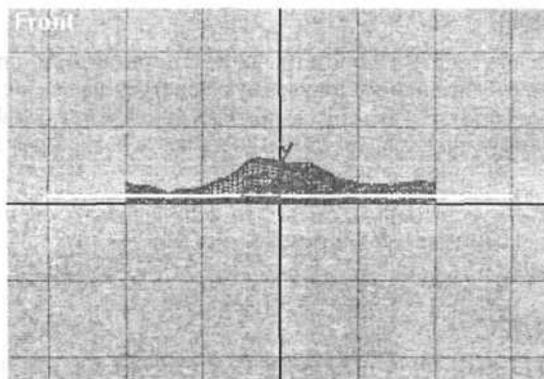
После этого присваиваем белый материал **box**, а черный — нашим  
скалистым горам.



### Способ №3

Если рисовать не получается или времени нет, нужно поискать плагин, который все делает сам. Плагин **Mountain** создает подходящую поверхность. Возникает вопрос — как оттекстурировать горы? Для первого способа всегда остается созданная карта, которую можно ограниченно использовать как маску или blend-карту, чтобы получить более естественный вид гор.

Если гора не нравится, то всегда можно поиграться с **Seed**'ом (генератором случайных чисел в плагине).



Теперь самое время создать немного дополнительной геометрии. В проекции **Top** создаем **box**.

## Глава 8. Использование anti-aliasing

Сравните эти две картинки:



Чувствуете разницу? На картинке слева (без anti-alias'a) изображение выглядит ступенчатым, а на картинке справа (с anti-alias'ом) закругления букв обработаны серыми точками, что создает впечатление округлой формы, хотя на двух картинках одинаковое разрешение.

Используйте **anti-alias** всегда для создания более плавных изображений — в 3D и 2D. Неплохо было бы иметь и anti-alias крупных шрифтов при просмотре web-страничек. Для этого вы можете использовать MS Plus!, в котором есть функция **Smooth edges of screen fonts** или Windows со встроенной функцией.

## Глава 9. Mountain Texturing

Для создания ландшафтов есть три простых способа:

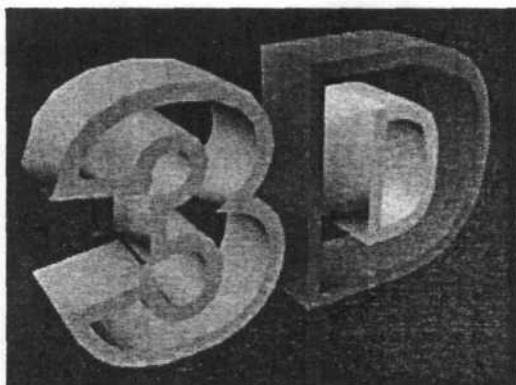
### Способ №1

Нарисовать вид сверху в любом графическом редакторе, а потом с помощью **displacement**'а выдавить полученную текстуру на **box**'е с достаточной плотностью сетки. В этом случае яркость пикселей **grayscale**-карты определяет глубину, на которую та или иная часть **box**'а будет выдавлена.

Если провести достаточно времени, то можно создать очень хороший ландшафт и, самое главное, он будет выглядеть именно так, как вы и хотели.

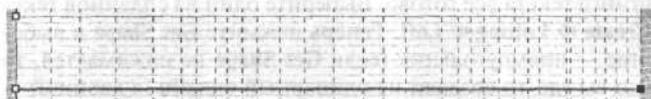
### Способ №2

Можно не рисовать ничего, а просто создать **box** и хорошенько потаскать его за вершины, не забывая держать **Affect Region** включенным или вообще взять **patch** или **NURBS** и углубиться в работу.

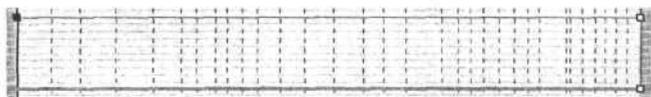


Осталось его проанимировать, что в общем сделать несложно. Мы будем анимировать деформацию масштабирования. Изначально (на первом кадре анимации) у нас не должно быть видно ничего — то есть по всей длине пути лофт должен иметь масштабирование до 0%. В конечном кадре должно быть все на 100%. А так как нам нужно, чтобы буквы «разматывались» из ленты, то будем двигать пределы этого перехода 100% в 0% от начала к концу пути по мере анимации.

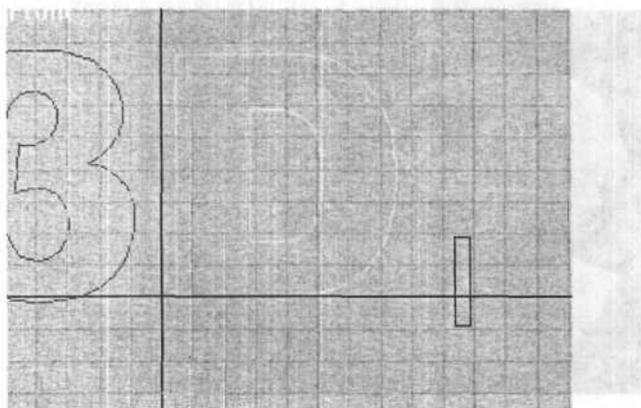
Возьмите один из получившихся лофтов и перейдите к его вкладке **Modify**. В самом низу найдите **Deformations/Scale** и кликните его. Теперь нажмите большую красную кнопку **Animate** на главном экране 3ds max 7.0 и перемотайте движок кадров на начало вашей анимации. Добавьте на кривую деформации две точки с помощью команды **Insert Control Point** и расположите их, как показано на рисунке:



Теперь зафиксируйте второе положение точек в позиции конечного кадра анимации:



Все! Анимация готова! Можете отрендерить и всем показывать!



Но и тут уже есть ловушка. Буква «D» на самом деле состоит из двух сплайнов — внутреннего и внешнего. Для устранения этого недостатка буквы D нам придется разделить ее на два разных сплайна. Выделите ее и во вкладке **Modify** примените модификатор **Edit Spline (Редактирование сплайна)**. Теперь в **Sub-Object** выберите пункт **Spline** и выделите один из сплайнов буквы «D», который бы вы хотели отделить. Прокрутим вниз правую панель с кнопками и внизу увидим кнопку **Detach**. Смелее нажимайте и пишите название для нового объекта — отделенного сплайна!

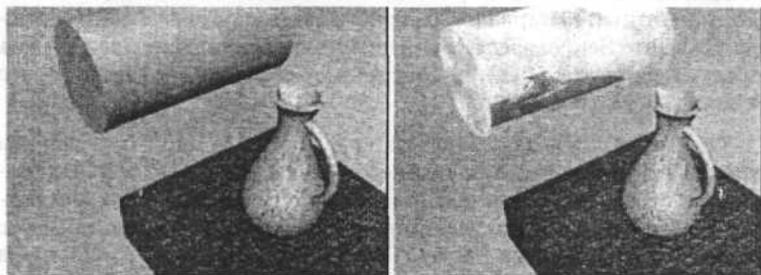
Теперь надо создать сплайн-сечение. Пойдя по пути наименьшего сопротивления, создадим простой прямоугольник (**Create ⇨ Spline ⇨ Rectangle**).

Для лофтинга текста все готово. Выберите один из сплайнов текста и сделайте **Create ⇨ Lofting ⇨ Loft**. Теперь нажмите **Get Shape** и выберите свое сечение — прямоугольник (если **Get Shape** не нажимается, то это означает, что вы где-то создали не одинарный сплайн, а двойной — такую букву, как например «D» или «O»). Одна буква готова. Прodelайте ту же операцию для каждого сплайна-пути и получите полный текст.

### Способ №3 — Raytrace

**Raytrace** — это встроенный рендерер 3ds max 7.0, доступный еще во второй версии. Рендеринг или трассировка лучей — это способ построения изображения, при котором каждая точка поверхности, выбираемая с каким-либо шагом дискретизации испускает «луч», который просчитывается по обычным физическим законам (преломления, отражения) и, таким образом, получается идеальная картинка. Минусы такого подхода тоже очевидны — очень низкая скорость по сравнению со всем остальным. Продемонстрируем его на не самом традиционном цилиндрическом зеркале. Сделайте цилиндр и расположите его примерно так, как показано на картинке.

Создайте новый материал и в **Reflection Map** задаем новый **map** — **Raytrace** (процесс аналогичен заданию **Reflect/Refract** из №1). Присвойте материал объекту и рендерите.



Идеальное качество изображений, но скорость просчета — самая низкая из всех методов. Проблема в том, что встроенный рендерер max'a очень и очень медленный по сравнению со своими аналогами.

## Глава 7.

### Эффект растущего текста

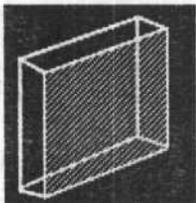
Это очень простой эффект, построенный на том, что в 3ds max'e можно анимировать все, даже настройки деформаций лофта. Попробуем сейчас анимацию такого растущего из ничего лофта простого текста.

Первый шаг — создание сплайна текста. Здесь первая тонкость — сплайн должен быть замкнутым многоугольником (плавным или угловатым), а значит, надо создавать каждую букву текста по отдельности (**Create** ⇨ **Spline** ⇨ **Text**). Для простоты создадим надпись из всего двух букв (вернее цифры и буквы) — «3D».

### Способ №2 — Flat Mirror

Этот способ применим только для идеально ровных поверхностей. Если у вас хотя бы на одну миллионную координата сдвинута — уже не работает. К тому же, способ не так прост, как №1.

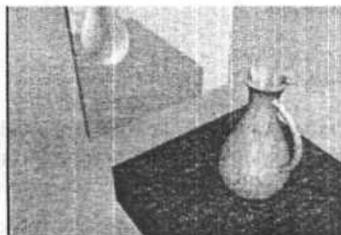
Приступим. Создайте **Box** так, как показано на картинке.



Нам понадобится не простой, а многокомпонентный, а точнее двухкомпонентный материал. Создайте новый материал **Multi ⇔ Sub-Object**. Сделайте **Set Number** и укажите **2** компонента. Нам надо создать два простых материала: один — собственно зеркало, а другой — то, что будет как бы обрамлять его.

Тыкаем мышкой во второй материал, если нужно — регулируем основные параметры, но самое главное — создаем **Flat Mirror Map** в разделе **Reflect Map** (так же, как и **Reflect ⇔ Refract**). Теперь самое интересное. Берете свой **Box**, и добавляете модификатор **Edit Mesh** (можно также сконвертировать **Box** в **Editable Mesh**). Теперь переключаете **Sub-Object** на **Face** и выделяете ту поверхность, что показана штриховкой на вышеприведенном рисунке.

Теперь ищите **Material ID** и вводите туда **2**. Теперь можно провести рендеринг.



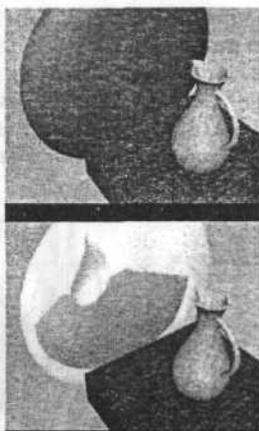
Качество изображения высокое. Но только для идеально плоских поверхностей. Метод необходим там, где нужны точные, качественные и самое главное быстро читающиеся зеркала. По сравнению с рейтрейсингом этот метод дает очень большой выигрыш в быстродействии.

мерным моделлером, но еще и крутой рендеринговой системой и рендерить такие вещи, как **bump**, отражения, прозрачность. До этого, конечно были системы типа POVR, но они, в основном, — рендереры, и моделировать в них нельзя.

Мы рассмотрим четыре основных способа создания зеркал, по возрастающей сложности.

### Способ №1 — Reflect/Refract

Создадим сферу и сплющим ее так, чтобы она была похожа по форме на линзу. Теперь идем в **Material Editor** и создаем новый материал. Можете оставить все основные установки так, как они есть — вполне подойдет почти любой шейдинг, блеск. Наша цель — раздел **Maps** и пункт **Reflect** в нем. Нажмите на него, и из появившегося списка **New** выберите **Reflect** ⇨ **Refract Map**. Можете не менять никаких установок, вернуться в основной материал и присвоить этот материал вашему объекту — сплющенной сфере. Теперь можно рендерить.



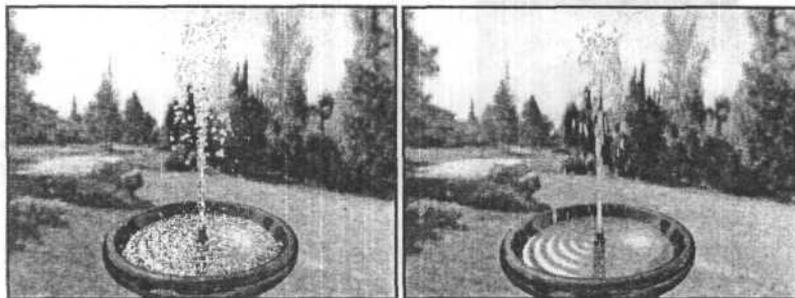
Основное достоинство этого метода — высокая скорость и простота. Но дальше идут одни недостатки. Во-первых, это очень низкое качество изображения. Даже на той сжатой из 640x480 до 200x150 картинке вы можете заметить ребристость границы — поверьте, она выглядит действительно отвратительно. Во-вторых, это качество изображения (количество пикселей) уменьшается с отхождением объекта от сферической формы. Для идеальной сферы этот способ очень привлекателен, но его ни в коем случае нельзя использовать на плоских объектах, и не рекомендуется на несферических объектах.

вышеперечисленных эффектов — **SpaceWarps: Gravity, Motor, PS: Blizzard** — система частиц с поддержкой клэй-технологии, построенной на «слипании» частей, подобно каплям жидкости в невесомости.

Начнем со следующих шагов. Создав в проекции **TOP Particle Blizzard**, перейдите во **FRONT** и поверните частицы на **180** градусов. В проекции **TOP** создайте **gravity** (в любом месте) и **MOTOR** (с центром посередине системы частиц), активируйте их (**Bind to SpaceWarp**). Просмотрите получившуюся анимацию.

Наполнив содержанием, переходим к форме — выбрав в настройках **Blizzard Particle type**, измените его на **MetaParticle**. Далее необходимо настроить параметры сетки, по которой из частиц будет выстраиваться результирующий объект. Наиболее оптимален такой вариант — для **ViewPort** половина размера частицы и четверть для визуализации. Чтобы получить отображение в окнах можно выбрать **MESH (default ticks)**. Однако наберитесь терпения или, что еще лучше, измените **Emission с Rate** на **Total**, далее цифра зависит от мощности вашего компьютера.

Мы создали основную струю воды и крупные капли, таким же образом создаются мелкие брызги, с той лишь разницей, что используются обычные сфероидные частицы вместо **MetaParticles** для экономии ресурсов. Ниже приведены две картинки — одна использует обычные частицы с **Motion Blur** (16 минут на кадр), другая — моментальный снимок «настоящей» струи из метачастиц (6.5 часов на кадр при максимальной детализации).



## Глава 6.

### Создание зеркальных поверхностей

Зеркальные поверхности — один из особых предметов гордости Autodesk и Kinetix — ведь именно их 3DS'ка смогла быть не только трех-

Теперь самое важное: чтобы свет было видно, добавьте **Volume Light**. Войдите в меню **Rendering** ⇨ **Environment** и в **Atmosphere** добавьте **Volume Light**, затем нажмите **Pick Light** и выберите ваш **direction**-источник света.

Самое интересное — это настройки **Volume Light**:

- ◆ **Fog Color** — цвет самого света.
- ◆ **Attenuation Color** — цвет света, в который свет будет переходить при затухании.
- ◆ **Density** — «плотность» света в процентах. 100% — непрозрачный свет.
- ◆ **Max Light** — максимальный уровень света. «Плотность» реально не может быть больше его.
- ◆ **Min Light** — минимальный уровень света. Уровень света вне луча.
- ◆ **Atten.Mult** — коэффициент затухания.



Все, луч готов!

## Глава 5. Создание фонтана

Перед тем, как браться за изготовление самого фонтана (с технической точки зрения это не так сложно), необходимо ответить на несколько базовых вопросов: как оно движется и почему образует именно эту форму.

На вырывающийся из сопла поток воды действуют две основные силы — земное притяжение с одной стороны и центробежная сила с другой (последняя возникает при выходе воды под напором из круглой трубки). Форма является результатом взаимодействия капель воды — разлетаясь из сплошной струи, они рассыпаются на множество отдельных брызг. В 3ds max 7.0 есть все необходимые инструменты для имитации

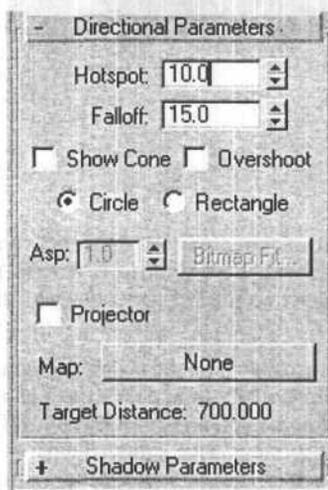
### Способ №1

Этот способ заключается в использовании **Volume Light** на **Directional** источнике света. Посмотрим его действие на примере: создадим простую сцену — что-то типа лучевого пулемета, взяв массив из 6 трубок, скрепив их кольцом. Теперь попробуем сделать так, чтобы из верхнего дула вырывался луч света.

Создадим стандартный источник света **Target Direct**, так чтобы сам источник находится в дуле (неглубоко, но и не совсем на поверхности), а цель источника далеко на предполагаемой линии лазера. Теперь проставьте параметры **Hotspot** и **Falloff** так, чтобы они соответствовали размерам вашего дула (например, дуло имеет радиус 30, **Falloff** — 15 и **Hotspot** — 10).



Если у вас изначально не было поставлено источников света, которые бы освещали сами стволы, поставьте их сейчас. Они больше не будут освещаться автоматически, так как вы добавили источник света.





Заметьте на этой картинке основные отличие **Bomb'a** от использования **PArray**:

- ◆ Осколки имеют произвольную форму — не одинарный **face**, и не простая группа **face'ов**.
- ◆ Осколки обладают таким параметром, как толщина, то есть считается, что чайник был полым и толстостенным.
- ◆ Осколки вращаются в полете.
- ◆ Осколки представляют собой стандартные **particle**, и к ним можно применять любые **space-warp** (такие как **gravity**, **wind**).

Можно разделить объект на куски по принципу сглаживания, то есть объект разделится по линиям около острых углов и прочих хрупких мест.

Можно использовать функции **spawn** для обработки ситуаций удара осколков о предметы — подумайте, как здорово будет выглядеть картина, когда чайник распадается на два осколка, потом каждый из них еще на два, каждый еще и так далее.

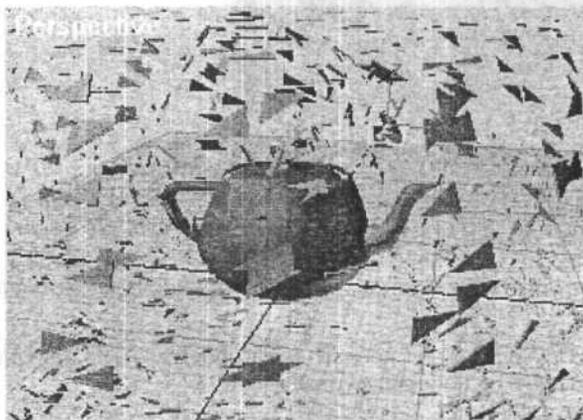
Рендерить сцену надо через **VideoPost**. С ней показывается чайник, падающий со стола на пол и разбивающийся на осколки.

## Глава 4.

### Как сделать параллельные лучи света

Иногда можно создать хороший эффект, показав, например, момент выстрела лазерного орудия, стреляющего пучком параллельных лучей. Этого эффекта можно достигнуть несколькими способами.

Все. Основной взрыв уже готов — запустите анимацию, и увидите, как ваш объект взорвется и разлетится на отдельные **face**'ы (как и **Bomb**).



Если вы хотите что-то более сложное — поэкспериментируйте со следующими установками:

**Particle Type: Object Fragment Controls: Thickness** — толщина осколков. Так как чайник имеет достаточно толстые стенки, надо увеличить этот параметр.

**Particle Type: Object Fragment Controls: Number of Chunks** — очень важный параметр, регулирует количество осколков, на которые разлетится чайник. Если надо, чтобы объект не разлетелся на **face**, поставьте нужный параметр.

**Particle Rotation: Spin Speed Controls: Spin Time** — еще один важный параметр для увеличения реалистичности — контролирует вращение осколков. Если этот параметр не 0, то осколок будет делать полный оборот за это количество кадров.

Для придания окончательного вида сцене, можно отрегулировать скорость и другие общие параметры — ведь **Particle Array** — это стандартный **Particle System**. В большинстве случаев также надо спрятать основной объект в момент взрыва (в отличие от **Bomb**, **PArray** этого не делает) в **Track View** ⇨ **Add Visibility Tracks** ⇨ **Visibility**. Время начала и конца взрыва также можно регулировать в основных настройках **PArray**.

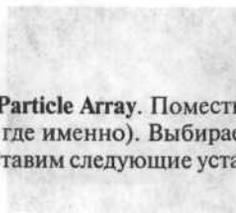
Если вам не понравился результат, всегда можно изменить параметры **combustion** и **bomb**.

**Пример 2.** Создание того же эффекта при помощи **Bomb** и готовых секвенций взрывов (**Pyromania**).

Воспользуемся предыдущим примером, только уберем **combustion**. Теперь создадим **path grid**, которая будет служить объектом для наложения секвенции. Создадим материал, где в качестве карты **diffuse** и **opacity** будут использоваться секвенции. Наложим полученный материал на наш **path grid**, и подберем **mapping** таким образом, чтобы взрыв выглядел реалистично. После этого подбираем момент, когда нужно делать взрыв, изменяем параметры **bomb** для того, чтобы разлетающиеся обломки и сам взрыв хорошо сочетались.

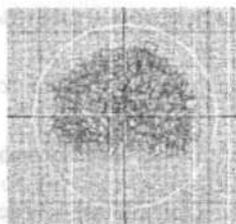
## Как сделать реалистичный взрыв и осколки от объекта

Часто используемым эффектом (особенно в космических сценах, анимациях) является взрыв. Для взрыва в *max*'е предусмотрен стандартный способ — **Combustion** и модификатор **Bomb**. Но это не самый хороший способ, и часто результат не устраивает — дело в том, что **Bomb** разделяет объект, в лучшем случае, на правильные формы групп **face** (в худшем — на отдельные **face**'ы). Если же нужен более сложный эффект — например, разделение объекта на несколько неровных кусков — используйте следующий способ. Демонстрация будет происходить на традиционном объекте — стандартном чайнике.



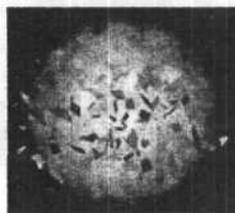
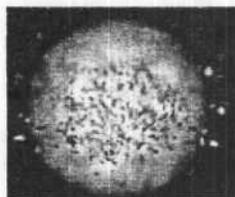
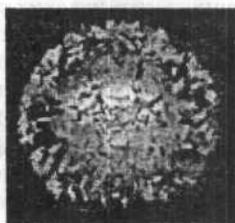
Создаем **Create** ⇨ **Particle Systems** ⇨ **Particle Array**. Поместим этот **Particle Array** где-нибудь в сцене (неважно, где именно). Выбираем объект, который будем взрывать (**Pick Object**). Ставим следующие установки:

- ◆ Viewport Display — Mesh
- ◆ Particle Type — Object Fragments



В меню **Rendering-environment** выберем **combustion**. При помощи **pickgizmo** выберем наш габаритный контейнер — **spheregizmo**. Займемся настройками самого **combustion**. Включаем **explosion**, заходим в **setup explosion** и задаем там **starttime** 10. Кроме того, в **combustion** меняем параметр **stretch** на 1.0.

Теперь можно попробовать отрендерить несколько ключевых кадров.



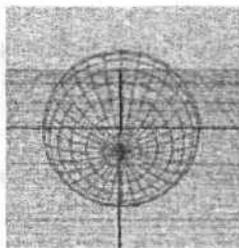
## Глава 3.

### Как сделать реалистичный взрыв

Первое, что приходит на ум — использовать простейший **bomb**, но при использовании этого модификатора ничего хорошего и реалистичного обычно не получается. Поэтому приведем два варианта даю два совета по поводу того как можно улучшить эффект создаваемый **bomb**. Оба примера основаны на совмещении **bomb** с другими эффектами.

**Пример 1.** Создание взрыва при помощи **bomb+combustion**.

Итак для начала создадим то, что будем взрывать — например, сферу.



Расположим модификатор **bomb** ниже нашей сферы, зададим ему следующие параметры:

```
strength=1,5;
spin=1,5;
FragmentSize min=1, max=5;
gravity=1,0;
chaos=0,5;
detonation=10
```

И свяжем его с объектом, который хотим взорвать.

Можете проиграть полученную анимацию. При этом вы заметите, что взрыв происходит на 10 кадре, а затем обломки разлетаются. Теперь займемся **combustion**. Для начала создадим **helper-atmosferic apparatus-spheregizmo**. Расположим его так, как показано на рисунке:

Выделяем два верхних левых **control point** и сдвигаем их вдоль оси X вправо.

Теперь пройдемся по получившейся призме модификатором **noise (fractal)**.

Единственное, что вам нужно запомнить — применять модификатор аккуратно, внимательно наблюдая за изменениями, и использовать его только по оси X.

Теперь приступаем к следующему этапу. Переводим в режим **edit mesh** и выделяем все правые крайние вершины (**vertex**). Выбираем модификатор **uniform scale** и выбираем ось воздействия X.

Продолжайте применять модификатор до тех пор, пока не получите практически ровный край.

Приступим к дальнейшим модификациям. Теперь применяем модификатор **FFD 4x4x4**. Выделите **control points** и переместите их вверх. Теперь выделите поочередно **control points** слева и справа и слегка поверните их.

Один лист книги у нас готов. Скопируем его и отобразим симметрично оси Y.

Займемся обложкой. Нарисуйте сплайн.

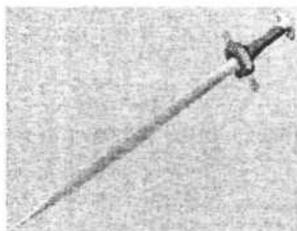


Теперь примените к нему **extrude**, так, чтобы у получившегося объекта края выходили за книгу.

Книга готова, по желанию к ней можно добавить новые элементы, например уголки или подставку.



Теперь перейдем к нашей модели, выберем режим **MeshEdit** в котором мы ее создавали. Отмечая нужные **face**, назначим им определенные номера **material** (если вы все правильно делаете, то на экране будет видно, как отдельные участки меча раскрашиваются материалами). Вот что должно получиться в результате:



Нажмите кнопку **Render**, чтобы получить достаточно красивый меч:



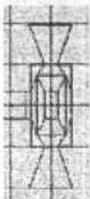
## Глава 2. Как нарисовать книгу

Это задание несложно. Главная задача — отрисовать странички (или создать их иллюзии). Сначала надо будет создать **box** с такими параметрами:

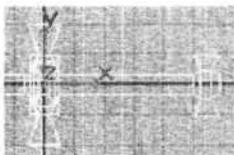
```
length=100  
width=100  
height=10  
length segs=1  
width&height=10
```

Теперь применим к нашему **box Free Form** — деформацию. Для этого переключаемся в режим **modify** и выбираем модификатор **FFD2x2x2**.

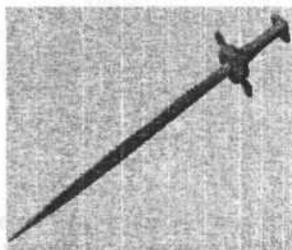
Например, так:



Теперь изменим рукоятку и ее набалдашник. Все по той же технологии.



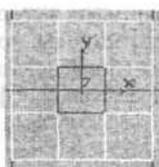
Получилось что-то отдаленно напоминающее меч. Но это еще не все. Применим к нашей модели модификатор **meshsmooth**. Попробовав изменить параметр **strength**, можно получать различные эффекты.



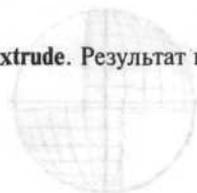
При этом помните, что вы в любой момент можете вернуться к своему изначальному объекту, подправить, изменить, добавить детальности.

Теперь давайте сделаем на наш меч текстуры и наложим их. Для этого воспользуемся материалом **multi/sub-object**. Создадим этот материал из трех.

Из Золота (**Gold Dark**), красного стекла (**Red Glass**) и металла (**Burshed Metal**). Каждому из материалов назначьте свой номер.



Теперь вытянем этот face при помощи **extrude**. Результат показан на рисунке:



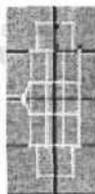
Затем перейдем в режим **vertex** и изменим расстояние между нижними вершинами, как показано на рисунке:



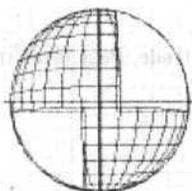
Все это заготовка для будущего лезвия нашего меча. Теперь выберем измененный нами **face** и снова вытянем его при помощи **extrude**. Придайте этому **face** вид как на рисунке:



С лезвием пока все. Теперь займемся рукояткой и гардой. Изменим ее в том стиле в каком пожелаете.



Самое время воспользоваться булевыми операциями, встроенными в 3ds max. Эти операции скрываются там же, где примитивы, но в другой подгруппе — **compound objects**. После применения (последовательно) двух **boolean** на нашу сферу и параллелепипеда, у вас получится сфера со срезанными краями:



Теперь надо создать для нее приличный материал, например **ray-trace**, который замечательно подходит для создания металлических и стеклянных поверхностей. В итоге получилась металлическая сфера со срезанными краями.



Примитивы являются очень мощным инструментом моделинга. Для того чтобы удостовериться в этом, вы можете посмотреть еще одно описание о том, как из одного **Box** при помощи **Edit Mesh** и **MeshSmooth** нарисовать настоящий меч.

Для начала создадим обыкновенный **box** с параметрами:

**length=20**

**width=10**

**hight=20**

**Width, Hight, Length steps=3**

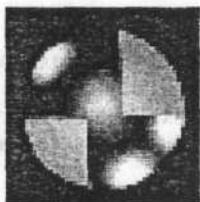
Теперь начнем его редактировать. К примеру, основой для меча будет гарда, поэтому мы будем вытягивать из нее все остальное. Перейдем в режим **edit mech** и выберем режим редактирования **face**.

Теперь выбираем один из центральных **face**.

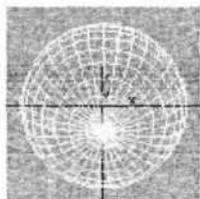
- ◆ Pyramid — пирамиды;
- ◆ Teapot — чайник с изменяемыми размерами;
- ◆ Prism — призмы

С этими базовыми элементами можно нарисовать практически все. Самое главное — научиться разбивать тело, которое вы хотите нарисовать на примитивы.

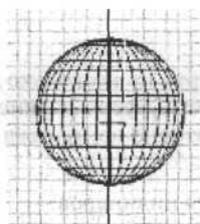
Вот, например, сфера со срезанными (частично) краями (попробуйте нарисовать ее при помощи **Loft**):



Для начала создаем обычную сферу (размеры не имеют значения) для лучшего качества результата задайте побольше **segments**:



Теперь нарисуйте два параллелепипеда:



Расположите их по краям нашей сферы, так чтобы они накладывались на нее.

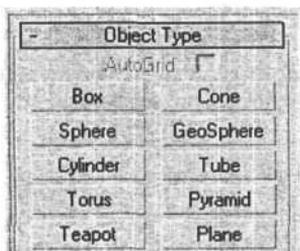
# Часть 9.

## Советы

### Глава 1.

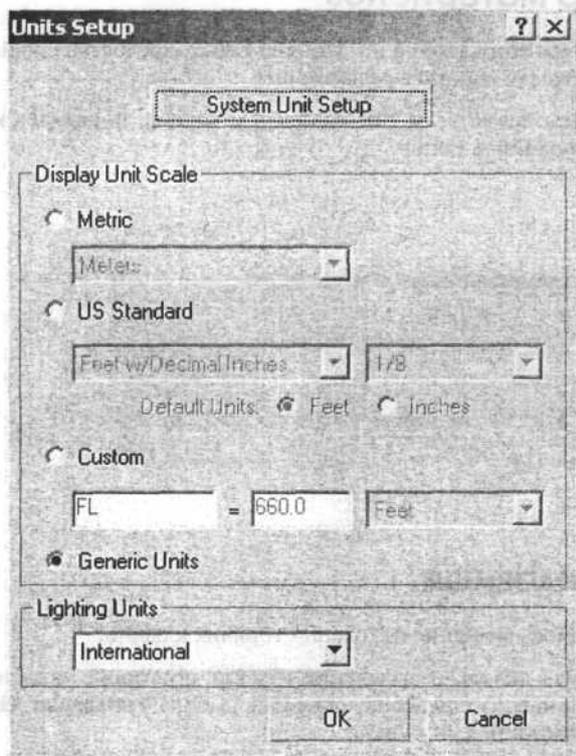
### Примитивы в 3ds max

Уже по названию кажется, что ничего интересного с их помощью не сделать. Это не так. При помощи различных модификаторов (таких как **Edit Mesh**, **Bend**, **Taper**, **Boolean** и многих других можно делать удивительные вещи. Прячутся примитивы на панели **Create** в пункте **Geometry**.



- ◆ **Box** — инструмент для создания различных кубов и параллелепипедов;
- ◆ **Cone** — средство для рисования конусов (как усеченных, так и обычных);
- ◆ **Sphere** — сферы;
- ◆ **GeoSphere** — то же самое что и сферы, но несколько другой конструкции (кстати получить из этой сферы эллипсоид или полусферу очень просто, достаточно поиграться с параметрами);
- ◆ **Cylinder** — инструмент для создания цилиндров;
- ◆ **Tube** — функция для рисования труб;
- ◆ **Torus** — средство для создания тороидов (очень подходит для рисования колец);

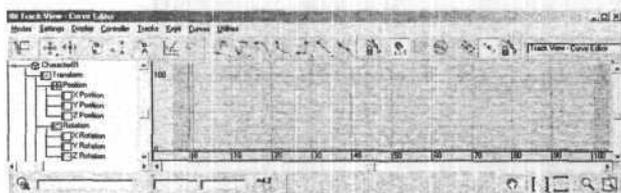
Упрощена и установка единиц измерения, а диалог **Units Setup** предоставляет возможность установки системных единиц.



## Глава 4. Редактор материалов

Появилась строка меню и у **Material Editor**, предоставляющая альтернативный способ работы с редактором.

В нижнем левом углу, под слайдером кадров, появилась кнопка вызова редактора **Curve Editor**.

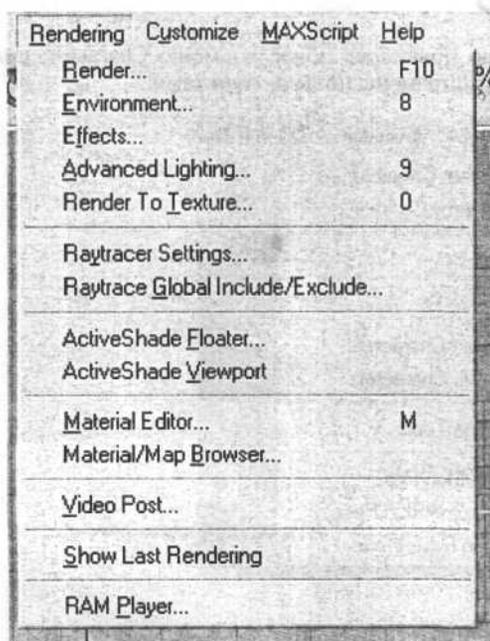
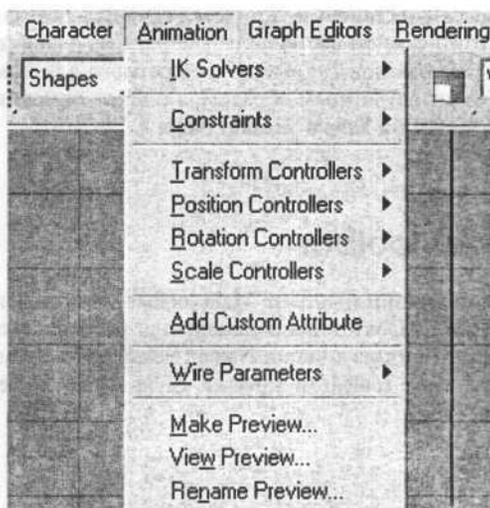


## Глава 5. Горячие клавиши

Изменилось значение некоторых горячих клавиш:

- ◆ **Q** — включает выделение объекта, повторное нажатие приводит к перебору способов выбора: **rectangular**, **circular**, **fence** и новый — **lasso**;
- ◆ **W** — вызывает **Select And Move**; видовое окно теперь максимизируется по **ALT+W**;
- ◆ **E** — включает **Select And Rotate**;
- ◆ **R** — активизирует **Select And Scale**, повторное нажатие **R** циклически перебирает режимы масштабирования: **uniform**, **non-uniform** и **squash**.

Координатная сетка **home grid** видовых окон **Perspective**, **Camera** и **Spotlight** теперь имеют фиксированный размер, другими словами, сетка не масштабируется при **zoom** или **pan**. Это важно при работе с новыми фотометрическими источниками света.



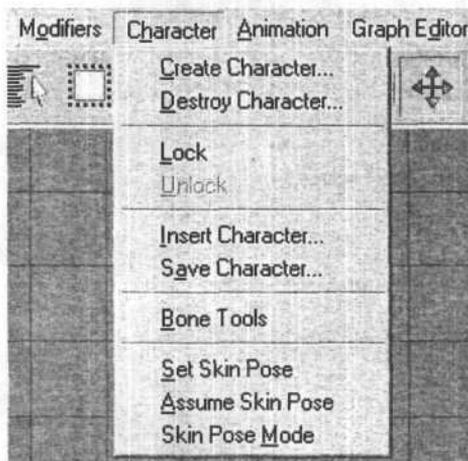
При запуске программы теперь не придется скучать — показывается загрузочный экран с описанием нескольких выбранных клавиатурных сокращений. Всего таких экранов 20, выбор конкретного для показа во время загрузки происходит случайным образом, а полное описание «горячих» клавиш можно найти по **Help** ⇨ **Hotkey Map**.

## Глава 2. Изменения Main Toolbar

По щелчку правой кнопки мыши на **Main toolbar** доступны: панель с закладками **Tab panel**, **Axis Constraints** и панель для работы со слоями **Layers**; здесь же появилась кнопка диалога **Named Selection Sets**; сменили «место жительства» на **Main toolbar** и кнопки **Snap toggle** и **Keyboard Shortcut Override toggle**.

## Глава 3. Главное меню

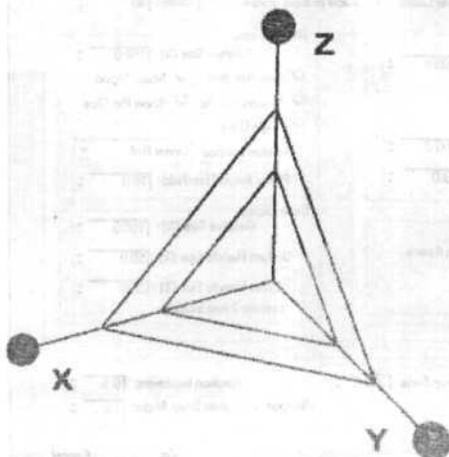
В главном меню появилось новое подменю **Character**; подменю **Animations** и **Rendering** дополнены новыми пунктами.



**Rotate gizmo**, отвечающий за вращение объектов, заново переделан на основе концепции трекбола. Работать с поворотами стало гораздо удобнее и вращать можно по осям X, Y, Z, свободно и в плоскости, перпендикулярной видовому окну. Величина угла поворота отображается интерактивно подсветкой меняющегося с поворотом углового сектора.



**Scale gizmo**, отвечающий за масштабирование, так же стал интерактивным и получил манипуляторы для выполнения масштабирования в соответствующих координатных плоскостях. Позволяет выполнять как **Uniform**, так и **Non-Uniform** масштабирование без обращения к **main toolbar**.

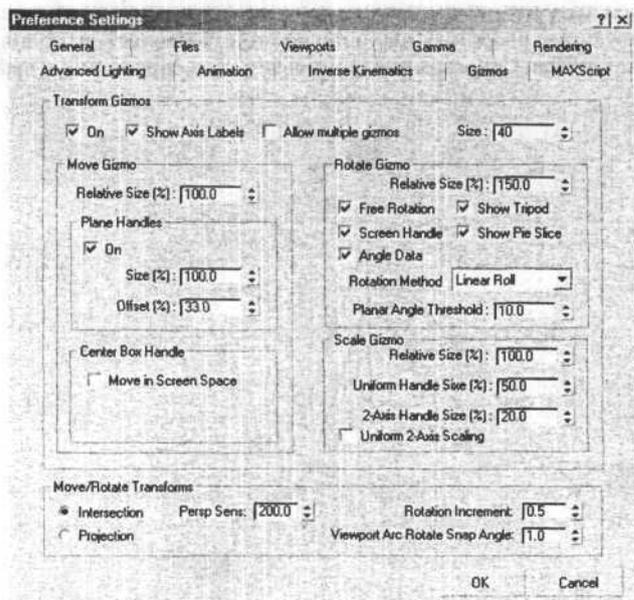


# Часть 8. Интерфейс

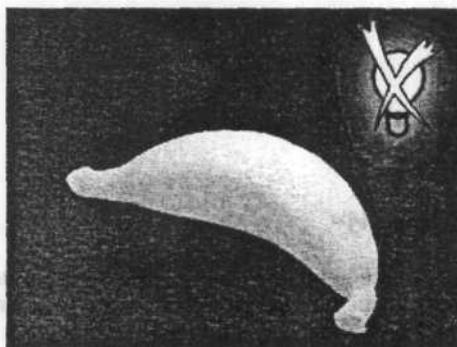
## Глава 1. Манипуляторы трансформаций объекта

Манипуляторы трансформаций объекта в видовых окнах стали более удобными и информативными.

**Move gizmo**, отвечающий за перемещение объекта, обзавелся манипуляторами **plane constrain** (перемещение в плоскости XY, YZ, XZ или параллельно видовому окну). При активизации соответствующий манипулятор подсвечивается прямоугольником желтого цвета. Размер и смещение манипуляторов могут устанавливаться в **Gizmos panel** диалога **Preferences**.

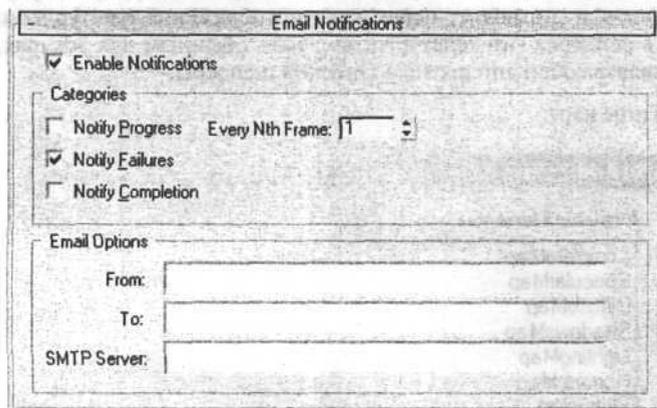


«Выпеченная» карта освещенности позволяет визуализировать банан даже при отключенных источниках света:



## Глава 2. Извещение о статусе рендеринга по почте

Аналогично сетевому, 3ds max 7.0 при обычном рендеринге может отправлять сообщения по электронной почте:



Сетевой рендеринг теперь выполняется при помощи программы **backburner**, функциональность сетевого рендеринга почти не изменилась.

# Часть 7.

## Рендеринг

### Глава 1. «Текстурные выпечки»

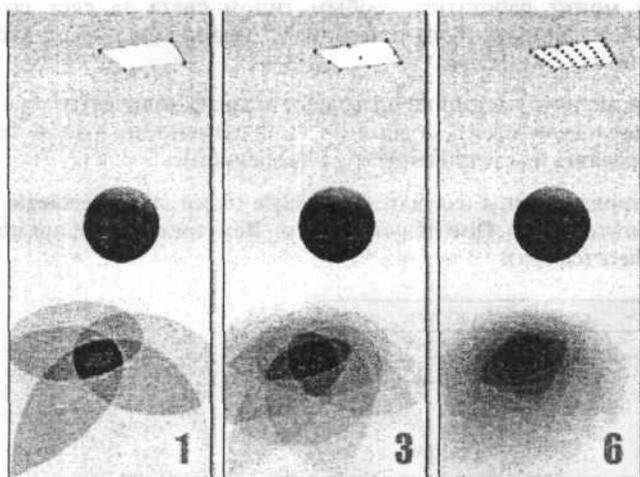
Рендеринг для текстур (**rendering to textures**, также известный как «текстурная выпечка») представляет собой процесс создания карт освещенных объектов в соответствии с тем, как они визуализируются в сцене, для последующего их наложения на те же объекты. Появление возможности делать «текстурные выпечки» обусловлено экономией времени при расчете анимации, а также тем, что такие текстуры могут экспортироваться в игры, поддерживающие Direct3D.

Для создания выпечки используется специальный механизм **Render to texture**, модификатор **Automatic Flatten UV** (автоматическая версия модификатора **Unwrap UVW**) и новый материал **Shell**. **Shell material** является контейнером, содержащим исходную текстурную карту и новую «выпечку», позволяет переключать отображение текстур в видовом окне и в рендере. Тип текстур может быть обычным для 3ds max, либо представлять собой аппаратные Direct3D шейдеры.

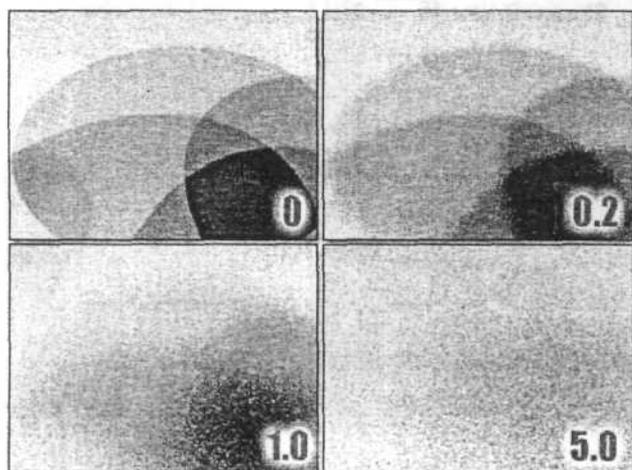
Типы карт:



## Влияние параметра Shadow Integrity...



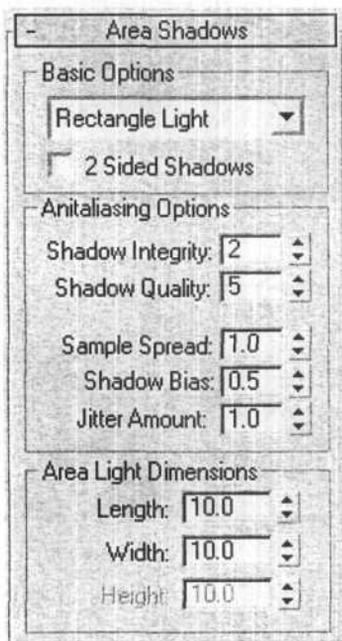
## ...и параметра Jitter Amount



**Area Shadow** – генератор теней от протяженных (area) источников, который может работать с любым типом света за счет создания виртуального протяженного источника с заданными геометрическими размерами.

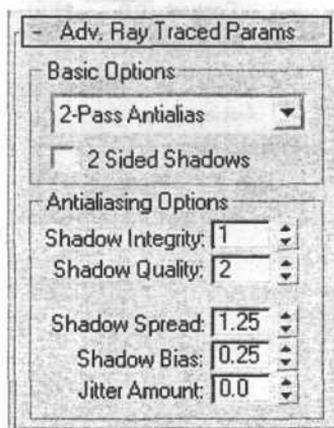
Кроме того, настройки позволяют задавать количество теней (элементарных излучателей), сглаженность и размытость границ теней, а также управлять их смешиванием (растворением).

Основные типы излучателей: **Simple** (один луч), **Rectangle** (плоский прямоугольник), **Disc** (плоский диск), **Box** (трехмерный прямоугольник), **Sphere** (сфера).

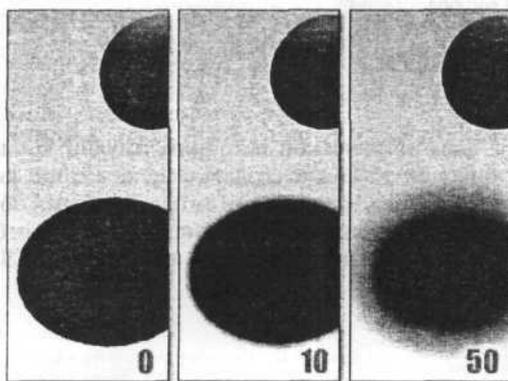


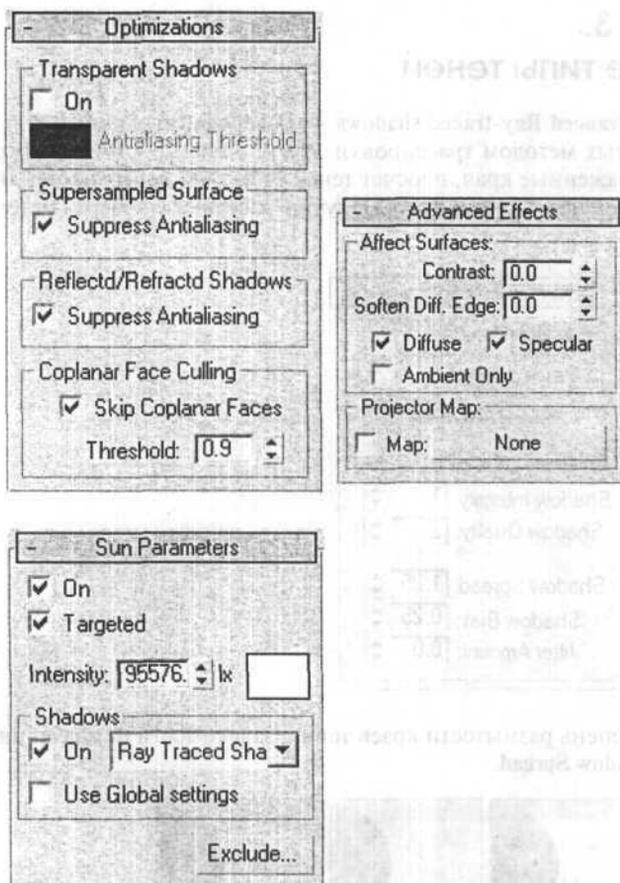
## Глава 3. Новые типы теней

**Advanced Ray-traced shadows** — усовершенствованный тип теней, получаемых методом трассировки лучей. Главные изменения: размытые и сглаженные края, просчет теней от задних поверхностей объекта (относительно источника света), лучше работают с новыми источниками света.



Степень размытости краев тени в зависимости от значения параметра **Shadow Spread**.



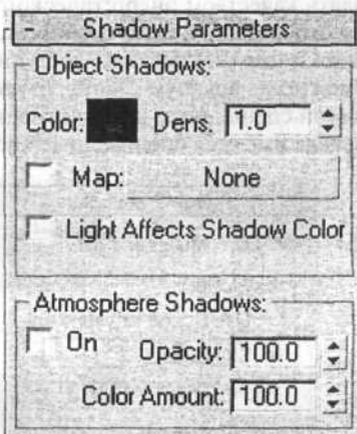
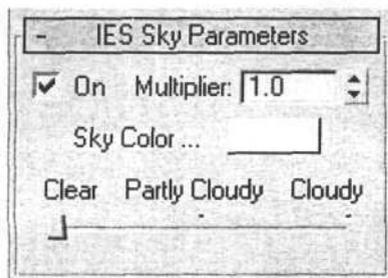


**IES Sun Light** – фотометрический источник, эмулирующий солнечный свет (солнце, наконец-то можно создать в сцене солнце, которое будет рендериться). Может взаимодействовать с атмосферными объектами, в частности, с облаками – менять интенсивность лучей и отбрасывать тень от них. Остальные закладки параметров аналогичны **IES Sky**.

**Skylight** — нефотометрический источник, предназначенный для симуляции дневного освещения, дает наилучшие результаты при использовании с **Light Tracer**.



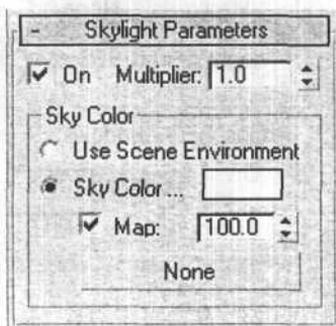
**IES Sky** — фотометрический источник, эмулирующий атмосферные эффекты — рассеянный свет солнца. Может использоваться как самостоятельно, так и в составе системы **daylight**. В последнем случае его параметры увязываются с параметрами источника **IES Sun Light**, эмулирующего солнечный свет. Работает только с **radiosity** или **light trace**. Может взаимодействовать с атмосферными объектами.



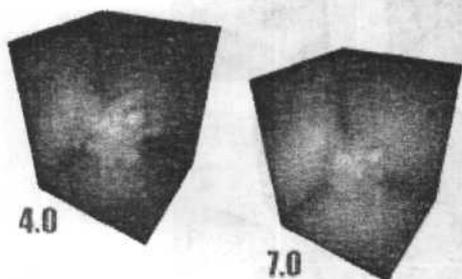
ветствовать реальной, поскольку для помещений высотой в 100 метров и в 100 дюймов будут получаться принципиально разные решения), необходимость работать только с фотометрическими источниками света (другие источники будут заменяться фотометрическими аналогами), а для воспроизведения естественного освещения использовать только **IES sun** и **IES Sky** (новые источники света, введенные специально для метода излучательности) или в крайнем случае, **Direct Light** и **skylight**.

Еще один нюанс — **Exposure control** применяется для установления соответствия просчитанного освещения его отображению на экране из-за того, что компьютерный монитор не в состоянии отобразить весь видимый спектр освещения. Для визуализации сцены методом излучательности сначала необходимо определить способ его взаимодействия с другими рендерами, затем запустить просчет решения. После того, как он будет выполнен, **radiosity**-решение будет автоматически подставляться в процессе рендеринга **scan-line** или **ray trace**, в том числе и при расчете анимации, если она обусловлена перемещением только камеры. Найденное решение остается правильным до тех пор, пока не изменится освещение в сцене как прямыми манипуляциями с источниками света, так и косвенно — за счет изменения положения объектов и элементов сцены. После таких изменений решение необходимо пересчитывать.

Возможности настроек, влияющих на качество **radiosity**-решения, вполне заслуживают эпитета «изошренные». Судите сами: вы можете управлять качеством энергетического распределения; качеством визуального отображения поверхностей за счет сглаживания освещенности соседних поверхностей; вам доступна интерактивная фильтрация шума, возможность вручную задать разбиение сцены на **radiosity**-элементы как для всей сцены, так и пообъектно; вы даже сможете рисовать светом — в арсенале имеются специальные кисточки!



- ◆ **Volumes amount** — меняет максимальное значение количества света, собираемого от **volumetric** эффектов.

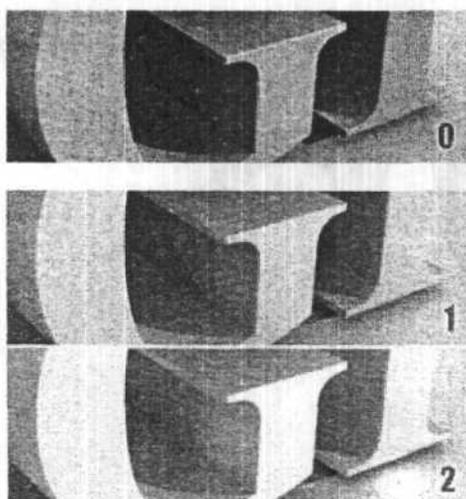


## Глава 2. Radiosity

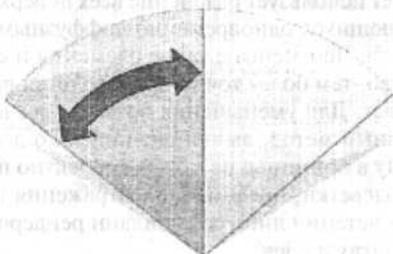
Возможность рендеринга методом **radiosity** (излучательности) стала доступна в 3ds max начиная со второй версии, в виде плагина. Кто с ним работал, почувствует себя с одной стороны «в своей тарелке», с другой — будет приятно удивлен качеством и полнотой возможностей нового radiosity-рендера пятого max'a.

Метод **radiosity** рассчитывает дополнительное освещение, обусловленное диффузным рассеянием света поверхностями всех объектов сцены на основе физически корректной модели поведения фотонов (метод **radiosity**, используемый в 3ds max 7.0 носит название **SRR — stochastic relaxation radiosity**). Расчет использует разбиение всех поверхностей на небольшие элементы, являющихся одновременно диффузными излучателями и приемниками света. Чем меньше такие элементы и соответственно, больше их количество, тем более точно производится расчет, тем больше времени он занимает. Для уменьшения времени расчетов в 3ds max применяется рекурсивный метод, выполняющийся до достижения заданной точности. **Radiosity** в принципе не может корректно просчитывать прямое освещение, подсветки, преломления/отражения и поэтому должен использоваться в сочетании либо со сканлайн рендером, либо с методом трассировки лучей (**ray tracing**).

В целом, для максимально реалистичного расчета освещенности в 3ds max необходимо использовать целый комплекс средств: это и пара **radiosity/ray trace**, и соответствующие материалы, и источники света. Кроме того, придется учитывать размерность сцены (она должна соот-



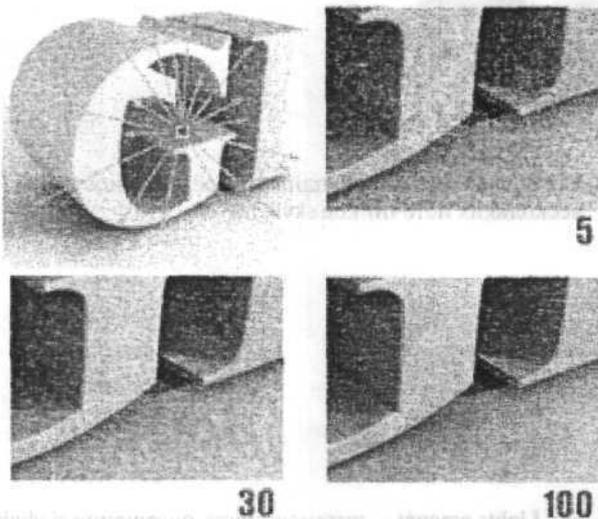
- ◆ **Ray Bias** – как и **Ray-Trace Bias** для теней, позиционирует отраженный свет. Используется для коррекции артефактов.
- ◆ **Bounces** – число трассируемых отраженных лучей света.
- ◆ **Cone Angle** – определяет угол конуса перенацеливания лучей.



- ◆ **Volumes toggle** – расчет освещения для **volumetric** источников, таких как **Volume Light** или **Volume Fog**.



- ◆ **Color Filter** – фильтрует весь свет, падающий на объект, может использоваться для окраски.
- ◆ **Filter Size** – размер фильтра в пикселах, используемый для уменьшения шума.
- ◆ **Rays/Sample** – число лучей на пиксел. Большие значения увеличивают сглаженность.



- ◆ **Extra Ambient** – если цвет отличен от черного, добавляет цвет.

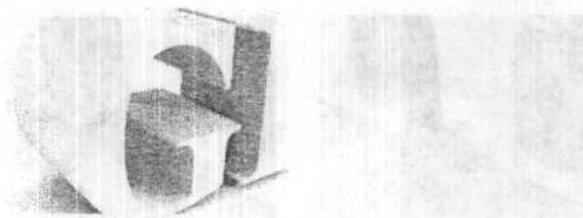
- ◆ **Global Multiplier** — общий уровень освещенности в сцене.



- ◆ **Object Multiplier** — уровень отражаемого объектами света.



- ◆ **Sky Lights toggle** — накопление освещенности от нескольких источников skylight.



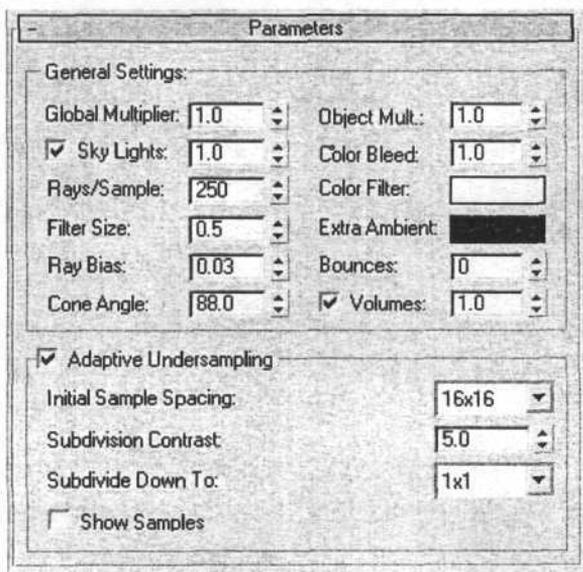
- ◆ **Sky Lights amount** — интенсивность освещения skylight.

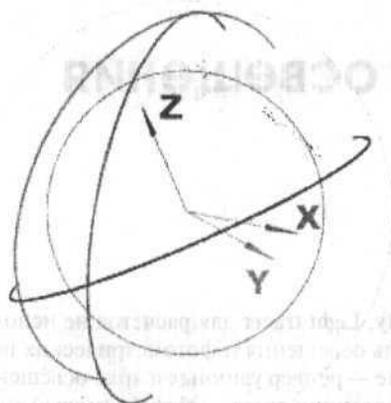
# Часть 6.

## Элементы освещения

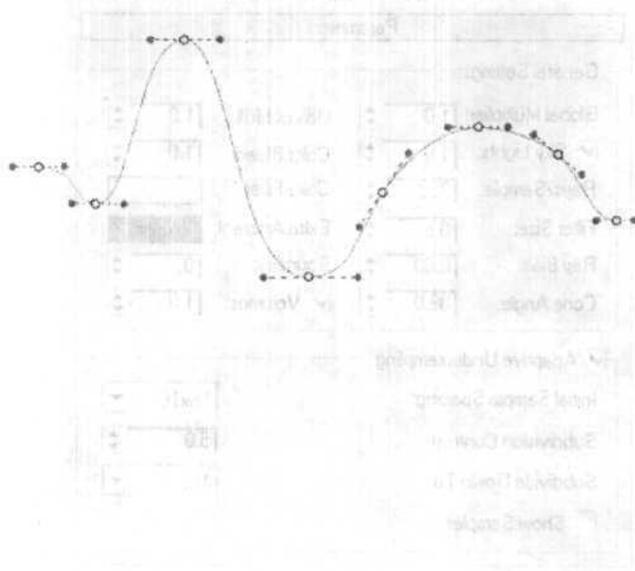
### Глава 1. Light Tracer

В отличие от **radiosity**, **Light tracer** для расчетов не использует физически корректную модель освещения и фотометрических источников света. Основное назначение — рендер уличных и ярко освещенных сцен. Основным используемым источником света — **skylight** (новый тип освещения), дополнительные источники могут использоваться для проработки теней. Для включения расчетов методом **Light Tracer** и доступа к его настройкам выбираем **Rendering** ⇨ **Advanced Lighting** ⇨ **Light Tracer**. Объекты можно избирательно исключать из расчетов, отключая соответствующую опцию в **Object Properties dialog** (свойства объекта).

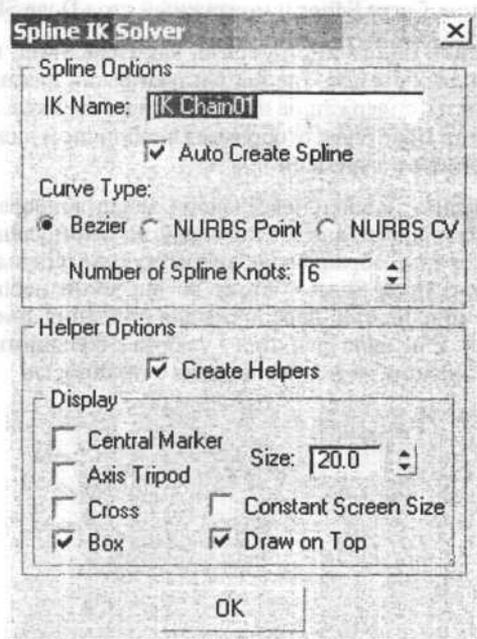




Для обеспечения гладких переходов между анимационными ключами введен новый тип интерполяции **Auto Tangent**, назначаемый по умолчанию. Это позволяет избежать недостатков (отклонений), свойственных использовавшейся ранее интерполяции типа **Smooth**.



Появился новый тип управления для инверсной кинематики — **Spline IK**, использующий сплайны в качестве формы для цепочек ИК. Предназначается для моделирования «веревочных» объектов типа длинных тонких хвостов, змей, щупальцев, веревок. Может использоваться и в качестве направляющего пути (**Path Constraint**), например для анимации ползушей змеи.



Для интерактивного вращения объектов при помощи контроллера **Euler XYZ** введена координатная система поворотов **Gimbal**. Эта система обеспечивает однозначное соответствие между ориентацией координатных осей и трексов Euler XYZ.

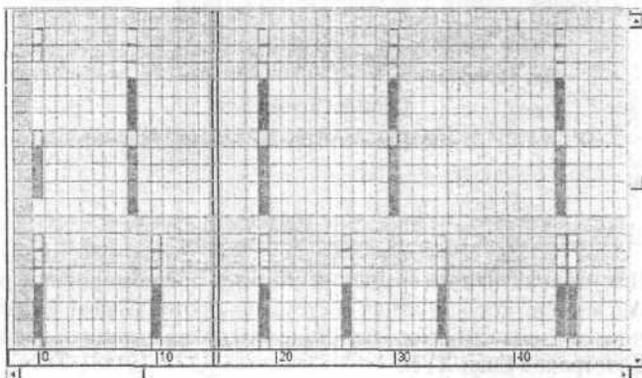
При использовании других координатных систем поворот относительно одной из осей всегда меняет не менее двух трексов. Следовательно, поворот относительно одной оси координатной системы Gimbal меняет только один трек, что облегчает редактирование функциональных кривых движения.



Изменения внесены и в **Track View** — улучшен интерфейс и появились два отдельных специализированных редактора: редактор функциональных кривых движения **Curve Editor** и монтажный стол **Dope Sheet**.

Идея возникновения такого инструмента, как **Dope Sheet**, происходит из практики реальных съемок, где используются так называемые **X-sheet** — монтажные листы, содержащие инструкции по съемке для оператора. В 3ds max редактор **Dope Sheet** отображает в табличном виде ключи анимации, расставленные по времени.

Здесь можно выделить любой или все ключи, масштабировать, перемещать, копировать их или выполнять с ними другие манипуляции непосредственно в редакторе, не обращаясь к видовому окну сцены. Наиболее общее применение **Dope Sheet** состоит во внесении небольших искажений в движения конечностей персонажа для придания анимации большей естественности. Еще одна функция — работа с большими группами персонажей для придания их перемещениям случайности.



**Dope Sheet** работает в двух режимах: **Edit Ranges** и **Edit Keys**. В режиме **Edit Keys** в таблице для опорных кадров отображаются цветные ключи (каждый цвет соответствует определенному виду трансформации) и ключи для промежуточной анимации, легко различающиеся по внешнему виду (здесь не показаны). Для выделения нескольких ключей доступен режим **Soft selection**, что особенно полезно при обработке ключей данных захвата движения из-за их огромного количества.

# Часть 5.

## Анимация в программе

### Глава 1.

### Элементы нелинейной анимации

Начиная с данного релиза, в 3ds max вводятся элементы нелинейной (pose-to-pose) анимации. В идеале это позволяет создавать последовательность опорных сцен, независимо придавать каждой из них желаемый вид, расставлять эти сцены по времени и создавать между опорными кадрами промежуточную анимацию.

Преимущество такого способа перед методом **Auto Key** заключается в получении возможности экспериментировать с различными идеями как для опорных кадров, так и для промежуточных анимаций, без необходимости каждый раз заново переделывать большой кусок работы. Кроме того, именно так работают в традиционной анимации. Теперь в сцену можно вносить изменения, и на их основе избирательно определять анимационные ключи для определенных треков и объектов.

Вместо кнопки «**Animate**» появились три новые:

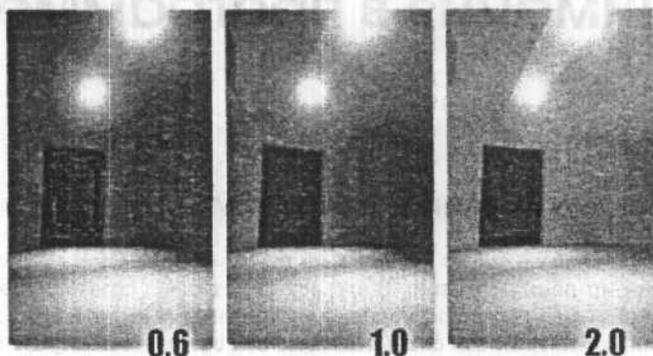
A rectangular button with the text "Auto Key" in a light-colored font on a dark background.

Аналогична по функциям старой «**Animate**» — нажали кнопку, переместили слайдер к нужному кадру, внесли изменения в сцену, снова переместили слайдер. Линейная анимация по-прежнему с нами.

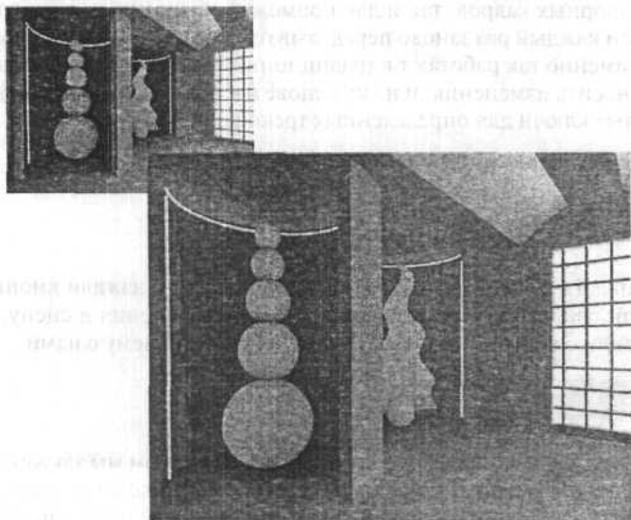
A rectangular button with the text "Set Key" in a light-colored font on a dark background.

Нажав **Set Key** переходим в режим, при котором мы также сначала перемещаемся в нужную временную точку. Но прежде чем изменять или трансформировать параметры объекта, мы определяем треки, на которых будут создаваться анимационные ключи (с помощью иконки **Keyable** в **Track View** и различных трековых фильтров). И только после этого вносим изменения в сцену и когда то, что получается принимает удовлетворительный вид жмем кнопку, которая и создает анимационные ключи.

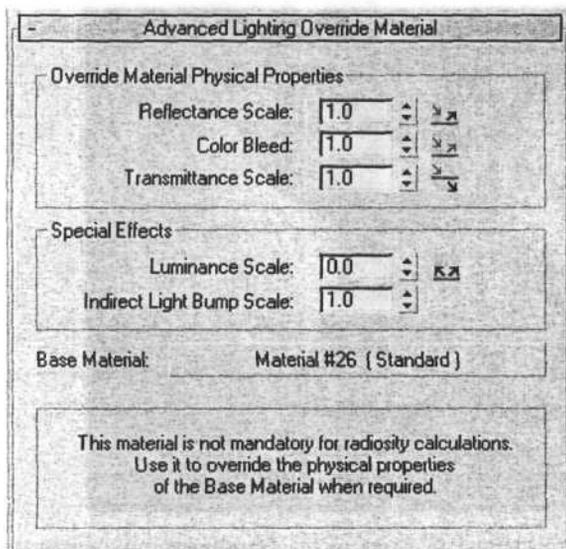
Здесь ситуация обратная — увеличение силы отражения и изменение цвета для материала пола позволило осветить излишне темную комнату. Цифрами указано значение параметра **Reflectance Scale**.



На этом рисунке показан пример учета взноса от самосветящегося материала (неоновые источники света) с помощью AOL.

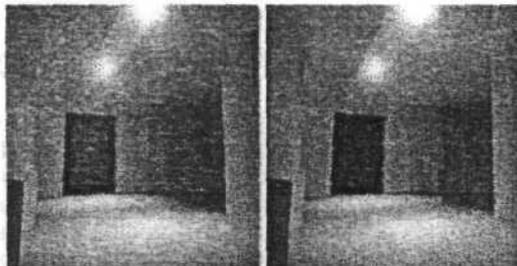


тов, таких, например как взнос в общее освещение от самосветящихся объектов.



Параметры материала **Advanced Lighting Override** позволяют настраивать степень отражения, цвет отраженного и проходящего (для прозрачных объектов) света. Группа параметров **Special Effects** предназначена для учета самосветящихся объектов и масштабирования эффекта **Bump**.

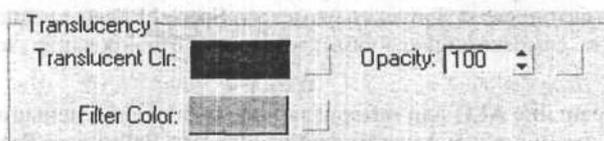
Применение ALO для материала пола позволило уменьшить степень его отражения в сцене (настройкой параметра **Reflectance Scale**), визуализированной методом **radiosity**.



объекта, приобретая немного другой цвет, и выходит через его поверхность. Сам объект при этом оставляет впечатление легкой светимости.



Параметры настройки материала **Translucent** становятся доступны после выбора у стандартного материала на закладке **Translucent shader**. Новые параметры:



### Глава 3. Advanced Lighting Override Material (ALO)

Вспомогательный материал, предназначенный для совместной работы с **radiosity** и **light tracing**. **Advanced Lighting Override Material** всегда является дополнением к базовому материалу, в качестве которого может выступать любой визуализируемый тип материала 3ds max. Используется ALO для достижения двух основных целей: настройка свойств материала для расчетов **radiosity** или **light tracing** и для создания спецэффек-

# Часть 4.

# Материалы и шейдеры

## Глава 1.

### Ink 'n Paint

Теперь появилась возможность создавать изображения, стилизованные под рисованные. Но если вы решите отрендерить сцену с объектами, которым уже назначены обычные материалы, то сделать их «рисованными» сразу не получится. Сначала придется создать новые материалы типа **Ink 'n Paint** и заново назначить их объектам, поскольку рисованный тип визуализации достигается на уровне материала, а не на уровне рендера.

Что же представляет из себя **Ink 'n Paint**? Прежде всего, это цвет, которым собственно и раскрашивается объект (будем дальше именовать его раскраской). Также это линии, которыми прорисовываются контуры объекта и различные его элементы. Раскраска и линии разделены и обладают каждый своим независимым набором настроек. Для цвета — это цвет освещенной области, цвет затененной области и подсветки, для линий — толщина и цвет. Для каждого параметра раскраски и линии можно задавать растровую карту.

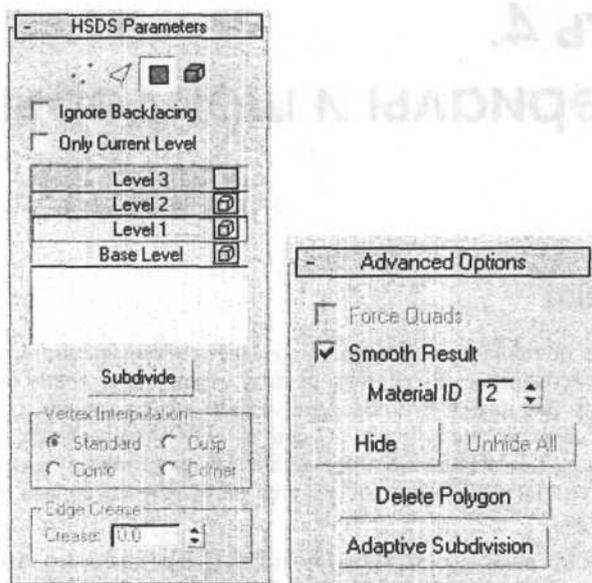
Например, карта шума для контурной линии может придать изображению большее сходство с ручным рисунком. С помощью линий можно обрисовать контур объекта, различные группы сглаживания и части поверхности, имеющие разные текстурные каналы.

Спектр возможностей нового материала довольно широк: ему вполне по силам имитировать ручной рисунок на бумаге, чертеж, картонную аппликацию.

## Глава 2.

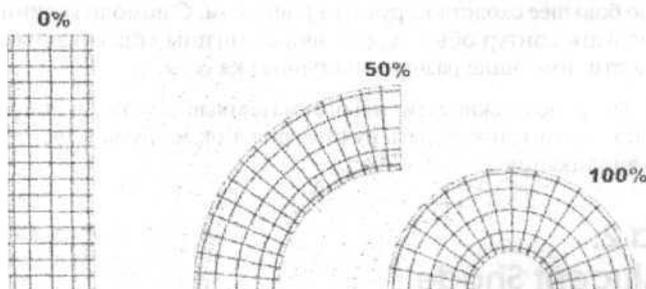
### Translucent Shade

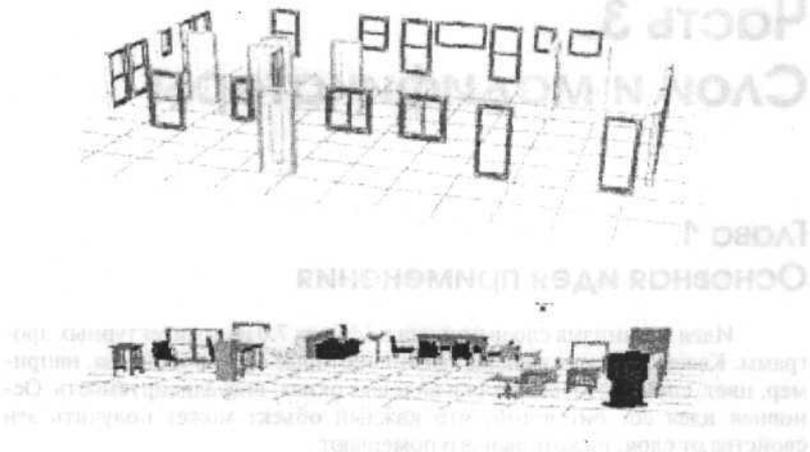
**Translucent** — модный эффект рассеяния света внутри полупрозрачного материала. Свет падает на одну сторону, рассеивается внутри



### Глава 3. Модификатор Morpher

Добавлена поддержка «**progressive morphing**», что позволяет вставлять в морф промежуточные формы, влияя на точность интерполяции.





Диалоговое окно для работы со слоями вызывается по щелчку правой кнопкой мыши на **Main Toolbar**. Способ работы со слоями и объектами несколько запутан и неудобен. Почему бы не предусмотреть возможность перетаскивания объектов между слоями в окне свойств слоев? Вместо этого нужно использовать неудобные манипуляции с выделением объектов в видовом окне и щелкать кнопки на панели **Layers**, да еще и помнить при этом, что и в каком порядке надо нажимать.

Для перетаскивания объекта с одного слоя на другой необходимо, чтобы в этом другом слое уже был хотя бы один объект. Затем вы выделите перетаскиваемый элемент, нажмете иконку **Match Object's Layer** на панели **Layers**, затем щелкните на объекте, который принадлежит слою, куда мы перетаскиваем первый объект. После этого еще наверняка придется проверить свойства слоев, чтобы убедиться, что объект попал туда, куда нужно. В общем идея со слоями неплоха, но рабочий интерфейс неудобный.

## Глава 2. Модификатор HSDS modifier

**HSDS modifier (Hierarchical SubDivision Surfaces)** – оптимизирована скорость работы, так что модификатор стал работать быстрее, и изменен рабочий интерфейс. В частности, у него появился собственный стек для уровней **Subdivide**.

# Часть 3.

## Слои и модификаторы

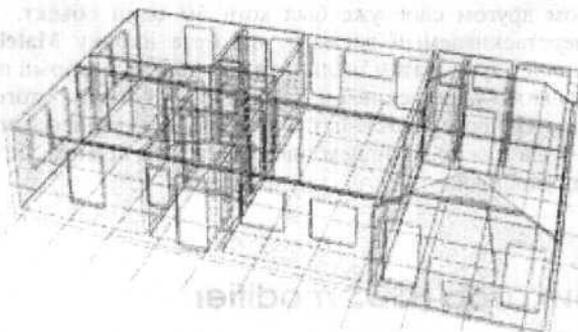
### Глава 1.

#### Основная идея применения

Идея механизма слоев пришла в 3ds max 7.0 из архитектурных программ. Каждый объект в сцене имеет некоторые общие свойства, например, цвет, способ отображения в видовых окнах, визуализируемость. Основная идея состоит в том, что каждый объект может получить эти свойства от слоя, на который его помещают.

Например, можно поместить все высокодетализированные объекты на один слой и задать ему свойства отображения **bounding box**, то есть, вместо объектов будут видны только границы прямоугольных контейнеров, что увеличит скорость работы в видовых окнах.

Более того, каждый новый объект, помещенный в этот слой, может получать от него то же свойство отображения в видовом окне, то есть, **bounding box**.



А теперь — суть метода:

- ◆ Освещаем предмет (сцену) традиционно. (Только у источников света не включаем **Cast Shadows**).
- ◆ Создаем источник света. Например — Spot01.
- ◆ Копируем его так, чтобы копия осталась в том же месте, что и оригинал (!).
- ◆ Далее, в Spot01 включаем **Cast Shadows**.
- ◆ В Spot02 **multiplier** устанавливаем отрицательным, но равным по модулю **multiplier** в Spot01.

Теперь все параметры Spot01 компенсируются Spot02, кроме теней. Мы получили сложный источник света (Spot01+Spot02), который создает тени и не влияет на освещенность сцены! Для удобства можно «подлинковать» оба источника к Dummy-объекту. А используя **Exclude/Include** можно добиться влияния источников только на данный объект(ы).

Тут есть большое поле для экспериментов. Например, можно и в Spot02 включить **Cast Shadows**, но в Spot01 поставить **Sample Range=5** (к примеру), а в Spot02 — равным 0. Тогда мы получим только контур тени объекта. И так далее...

Кто-то может спросить: «К чему все это — ведь в **Exclude/Include** есть опция **Exclude/Include cast shadows**». Действительно, но при этом источник света продолжает освещать предмет (сцену), то есть одним источником все равно не обойтись.

Все написанное выше — не догма, а совет. Не обязательно им всегда пользоваться, к каждой конкретной ситуации нужно подходить по-своему. 3ds max хорош тем, что один и тот же результат можно получить 1000 способами. Надеемся, что это поможет кому-нибудь в работе.

Для окончательного варианта необходимо довести половинку маски до нормального состояния, приаттачить вторую половинку к первой и объединить «пограничные» вертексы. При этой операции, кстати, убеется шов между половинками.

В основном модель готова. Подробности использования инструментов можно найти в меню Help.

### **Дополнительные замечания**

Характерные линии и каркас из характерных линий можно строить на основе различных соображений. Можно брать набор сечений объекта плоскостью или двумя, или тремя ортогональными плоскостями. Существует вариант достраивания вспомогательных линий сплайнами, как и характерные. Иногда это помогает, иногда начинают мешать лишние линии и вертексы при регулировке patch'ей.

Теоретически можно обойтись без каркаса из сплайнов.

## **Глава 21.**

### **Создание раздельного освещения и затенения сцены**

Крупные части изображения, не обязательно 3D, их внешний вид, композиция действуют на сознание зрителя, а мелкие — на подсознание. Человек может даже не понимать, что конкретно его смущает. Картинка «смотрится» тогда, когда мелкие ее детали не вызывают у человека подсознательного конфликта с тем, что он привык видеть. Задача художника — учитывать все мелочи при построении изображения. При моделировании (особенно если нужно добиться «фотореализма») огромную роль для качества имиджа играет правильное освещение сцены. В реальной жизни любой предмет освещается далеко не одним источником света. Первоначально их может быть один, два... (скажем, солнце, лампа), но свет от них, попадая на предметы, окружающие моделируемый, отражается (рассеивается) и тем самым мы получаем еще множество источников. Поэтому добиться требуемого результата, манипулируя источниками света, которые одновременно и освещают, и создают тени — нелегко, а в некоторых случаях — невозможно. Гораздо удобнее сначала осветить предмет так, чтобы он выглядел «как живой», а затем, независимо от освещения, создать нужные тени. 3ds max 7.0 позволяет это сделать достаточно просто.

## Глава 16. Моделирование головы с использованием Surface Tools

Первым делом вы должны создать с помощью сплайнов контуры вашего лица в области просмотра **Front**. Обведите контур половины головы. Затем обведите контуры вокруг глаз, носовой части и рта, также можете обвести дополнительные детали, которые вы хотите подчеркнуть в модели. Далее, добавьте поперечные сечения так, чтобы создать поверхность, содержащую три или четыре вершины. Чтобы облегчить работу при создании дополнительных сплайнов, включите кнопку «3D Snap» с параметром «End Points».



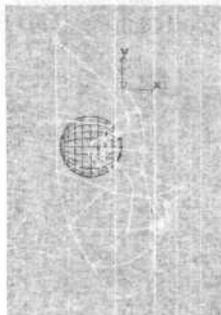
Обратите внимание, голова «аниме», обладает не лучшим контуром, так как я старался сохранить очень низкую детализацию. Далее, в процессе работы, я добавил необходимое количество сплайнов для получения нужной детализации.



Это изображение головы содержит более детальный контур из сплайнов. Фактически, было намного больше сплайнов, но лишние я удалил в процессе работы.

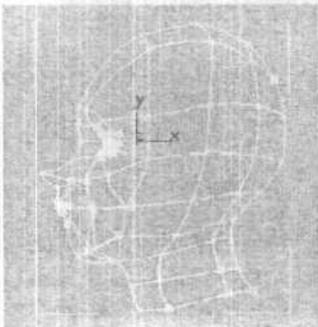
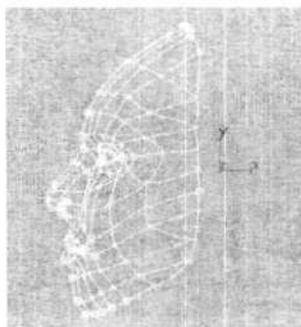
### Добавление объема

После того, как вы создадите основные сплайны спереди, то начинается самое интересное. Вторым шагом для вас станет добавление объема к созданным сплайнам. Используйте для этого фотографии вида спереди и сбоку, переместите каждую вершину, пока она не станет соответствовать вашим фотографиям.

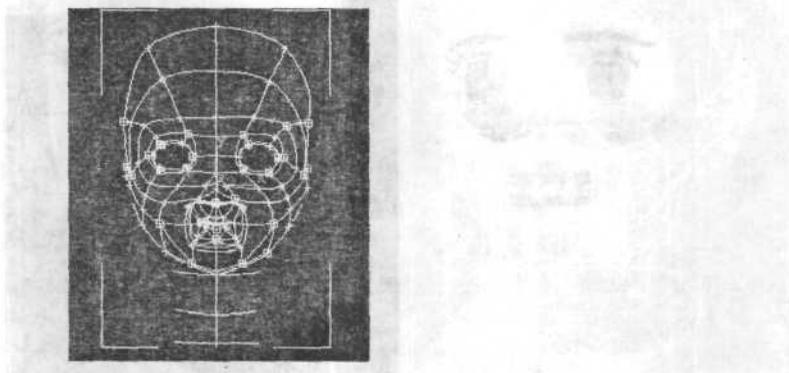


Это изображение содержит голову «аниме» после перемещения вершин в 3ds max.

Добавление объема к сплайнам в 3ds max 7.0 — наиболее длительная и нудная часть моделирования. Можно потратить часы, дни и недели, чтобы получить необходимый каркас с нужной детализацией. Затраченное время на это позволит вам в дальнейшем получить очень качественную модель.

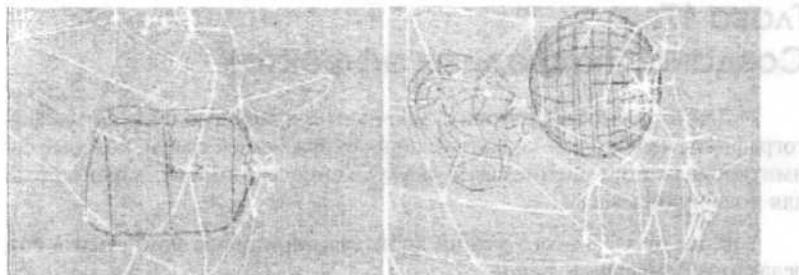


После того, как вы решите, что половина головы больше не нужна в редактировании, настало время для получения целой модели головы. Для этого вы должны зеркально отразить эту половину и полученную копию разместить в нужном месте. Затем совпадающие вершины двух половинок необходимо соединить.

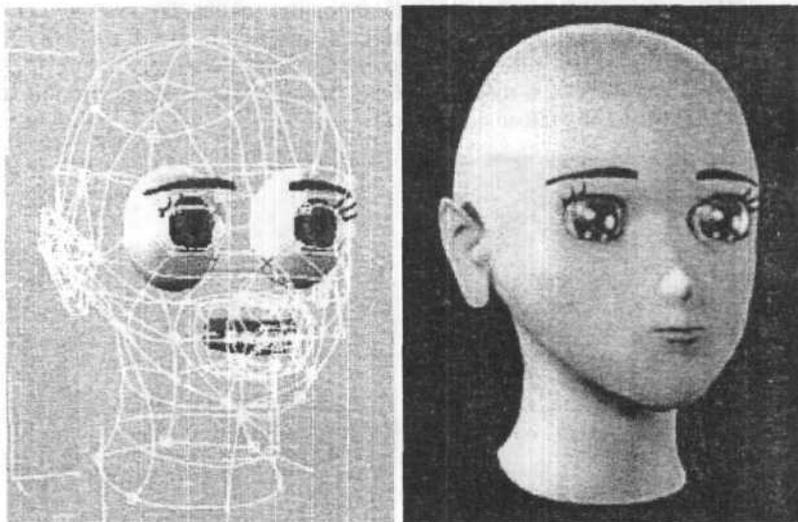


### Доводка

Я обычно создаю форму подобно мешку, которая соединена с внутренней частью губ моей модели лица. Это позволяет мне получить закрытый рот. Язык и зубы я обычно создаю как отдельные полигональные объекты. Нижнее изображение слева показывает размещение сплайнов для внутренней части рта.



Уши могут быть довольно сложными для моделирования, поэтому я предпочитаю создавать их отдельно. После создания уха я объединяю его с главной моделью, сваривая и устанавливая сплайны там, где необходимо.



Здесь показано изображение готовой модели.

Для наложения текстуры я применил цилиндрическую карту вокруг головы, затем сделал развернутую текстуру, чтобы видеть полигоны. В Photoshop нарисовал текстуры. Для фотореалистичных моделей я использую отсканированные фотографии с доработкой по модели.

## Глава 17.

### Создание подводных эффектов

Для того, чтобы повторить мрачность, присущую подводным фотографиям, необходимо создать условия окружающей среды, которые бы имитировали поглощение света и ухудшение видимости, характерные для подводной массы.

В 3ds max 7.0 этого можно добиться с помощью компонента Fog редактора Environment Editor.

#### Подводная мгла

1. Включите режим привязки к координатной сетке 2D Snap.
2. В видовом окне Top создайте объект Quad Patch, расположив его верхний левый угол в точке с координатами  $X=-100$ ,  $Y=100$ , а нижний

правый угол — в точке  $X=100$ ,  $Y=-100$  (Объект **Quad Patch** должен иметь длину и ширину в 200 единиц).

3. В разворачивающейся панели **Parameters** включите опцию **Generate Mapping Coordinates**. Примените к объекту **Quad Patch** текстуру песчаного дна.

4. Разместите в видовом окне **Top** камеру приблизительно в точке с координатами  $0, -40, 0$ . Перетащите ее точку съемки вертикально вверх до края объекта **Quad Patch**. При этом перемещение по оси  $Y$  должно составить приблизительно 102 единицы. В видовом окне **Left** переместите тело **Camera01** по оси  $Y$  вверх на 10 единиц.

5. Во вкладке **Modify** перейдите к параметрам группы **Environment Ranges** разворачивающейся панели **Parameters**. Включите опцию **Show** и замените значение 1000 параметра **Far Range** значением 125. Теперь измените представление в видовом окне **Perspective** видом из камеры **Camera01**.

### Глубинный свет

1. Щелкните на вкладке **Create** на кнопке **Light**, а затем — на кнопке **Omni**. В группе **Color** установите для параметров **RGB** значения 255, 255, 255. В видовом окне **Top** поместите источник света **Omni** приблизительно в точке с координатами  $50, -50, 0$  ( $x, y, z$  — соответственно).

2. В видовом окне **Left** переместите элемент **Omni** вверх по оси  $Y$  на 50 единиц.

3. Щелкните на объекте **Quad Patch**, чтобы выбрать его. Примените к нему модификатор **Edit Mesh** и установите для его опции **Sub-Object Selection Level** значение **Face**.

4. В любом видовом окне выберите весь объект **Quad Patch**. Во вкладке **Modify** в разворачивающейся панели **Edit Face** найдите группу **Tessellation**. Не изменяя текущих установок, щелкните на кнопке **Tessellate** (объект стал более сложным).

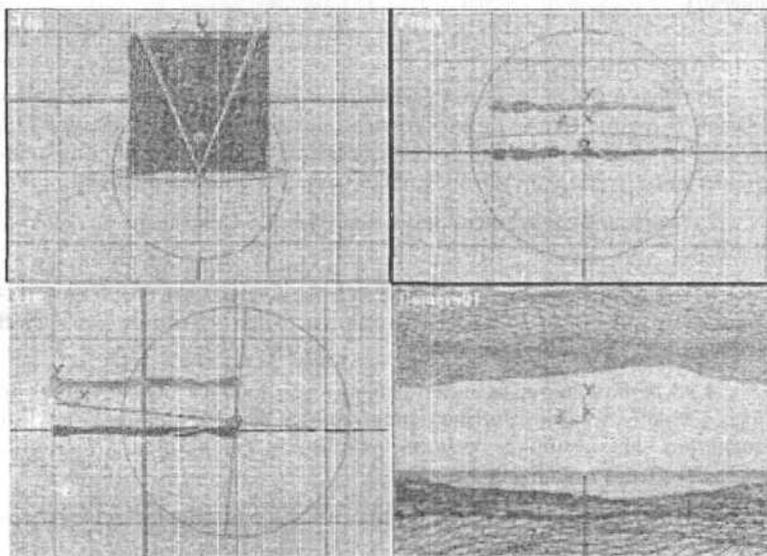
5. Щелкните на кнопке **Tessellate** еще два раза. Вернитесь в разворачивающуюся панель **Modifiers**.

6. Щелкните на кнопке **Noise**. Когда появится разворачивающаяся панель параметров **Noise**, включите опцию **Fractal**, а затем в группе **Strength** установите значения координат  $X=10$ ,  $Y=10$ ,  $Z=15$ . Объект **Quad Patch** станет напоминать поверхность с мягкими перепадами уровня.

### Поверхность океана

1. В видовом окне **Left** выберите объект **Quad Patch**.
2. Щелкните на кнопке **Mirror Selected Objects**, расположенной в верхней части экрана. Когда появится диалоговое окно **Mirror: Screen Coordinates**, в группе **Mirror Axis** выберите **Y** и установите для параметра **Offset** значение 50. В группе **Clone Slection** выберите опцию **Copy**, а затем щелкните на кнопке **ОК**. Измените цвет нового объекта на цвет морской волны.
3. Переместите в видовом окне **Left** объект **Camera01** вправо в точку с координатами **60,0,0**. А объект **Camera01 Target** в точку **0,30,0**.
4. В видовом окне **Front** переместите источник света **Omni** влево к точке **X=-50**, тем самым вы поместите источник света в центр сцены.

После этого экран должен выглядеть так:



### Поверхность моря

1. Выберите объект **Quad Patch01** и щелкните на вкладке **Modify**. Вы увидите, что объект стал ярко-красным. Если заглянуть в стек модификаторов (**Modifier Stack**), то видно, что модификатор **Noise** стоит первым в списке (то есть последним применен к объекту).

2. Щелкните на кнопке **Remove Modifier from the Stack** несколько раз, пока объект не примет первоначальный вид, с которого вы начинали работу. Убедитесь, что в разворачивающейся панели **Parameters** включена опция **Generate Mapping Coordinates**.

3. Откройте **Material Editor**, щелкните на свободный слот — он будет активизирован. Создадим новый материал для поверхности моря. Щелкните на поле **Ambient** и установите для его параметров **RGB** значения **0,8,16** (темно сине-зеленый цвет).

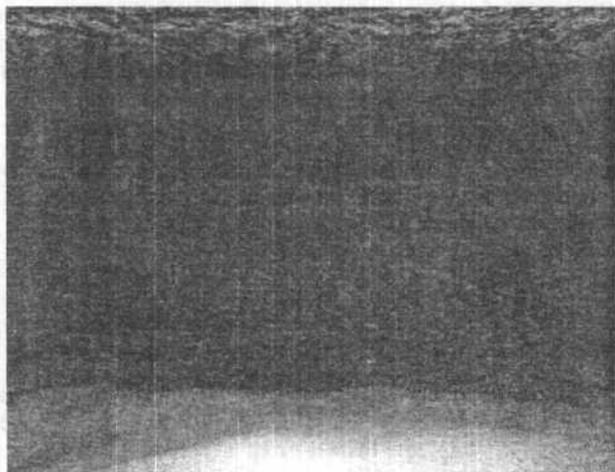
Замените для цвета **Diffuse** значение параметров **RGB** на **64,160,196**. Замените цвет **Specular** чисто белым цветом или RGB-значениями: **255,255,255**. Установите значение параметра **Shininess** равным **50**, а **Shininess Strength** равным **100**. Затем включите опцию **Soften**.

4. Откройте разворачивающуюся панель **Maps** и щелкните на кнопке справа от опции **Shininess**. Когда появится диалоговое окно **Material/Map Browser**, в группе **Browse From** выберите опцию **New**. Затем дважды щелкните на элементе **Noise** в списке справа. Вернувшись в редактор материалов, в разворачивающейся панели **Noise Parameters** для параметра **Noise Type** выберите опцию **Fractal**. В поле **Size** установите значение **4**. Щелкните на поле **Color#1**. В открывшемся окне **Color Selector** установите для параметров **RGB** значения **0,16,32**. Оставьте поле **Color#2** без изменений. Закройте диалоговое окно **Color Selector**.

5. Щелкните на кнопке **Go to Parent**, чтобы вернуться на верхний уровень редактора материалов, затем перетащите текстуру **Tex #1 (Noise)** из слота **Shininess** к слоту **Bump**. Когда появится диалоговое окно **Copy Method**, щелкните на опции **Instance** (но не **Copy!**), а затем — на кнопке **OK**. После этого увеличьте значение параметра **Bump Map** до максимума — **999**. Затем перетащите текстуру **Tex #1 (Noise)** из слота **Shininess** в слот **Reflection** и щелкните на кнопке **OK**, чтобы поместить копию этой текстуры.

6. Щелкните на кнопке **Reflection**, для параметра **Noise Type** выберите значение **Turbulence** и замените значение параметра **Size** с **4,0** на **1**. Щелкните на кнопке **Go to Parent**, чтобы вернуться на верхний уровень редактора и на кнопке **Assign Material to Selection**, чтобы назначить этот материал объекту **Quad Patch01**.

Попробуйте отрендерить сцену.



### Узоры на дне

1. В видовом окне **Left** создайте прожектор (**Target Spot**) приблизительно в позиции с координатами **0,0,48**, а точку направления разместите в центре объекта **Quad Patch** (дна океана).

2. Измените во вкладке **Modify** цвет прожектора на ярко белый. Параметр **Multiplier** установите на **2,0**. В группе **Attenuation** включите опции **Use** и **Show**, затем установите значения параметров **Start Range** и **End Attenuation** равными **95** и **100** соответственно. В разворачивающейся панели **Spotlight Parameters** установите значение параметра **Hot Spot** равным **120**, а значение **Falloff** равным **125**. Включите опцию **Show Cone** и выберите **Rectangle**. И наконец, в разворачивающейся панели **Shadow Parameters** включите **Cast Shadows**.

3. Включите опцию **Projector**, чтобы включить объект **Spotlight01** в свет прожектора, затем щелкните на кнопке **Assign**. Когда появится окно **Material/Map Browser**, дважды щелкните на элементе **Noise**. Нажмите на кнопку **Map**, поместите **Map #1** в слот **4** окна **Material Editor** и щелкните **OK**.

4. Вернитесь в диалоговое окно **Material Editor**. Активизируйте слот **4**. Для параметра **Noise Type** выберите значение **Turbulence**. Измените значение **Size** с **25,0** на **5** и щелкните на кнопке **Swap**.

Сделайте рендеринг.



### Подводный пейзаж

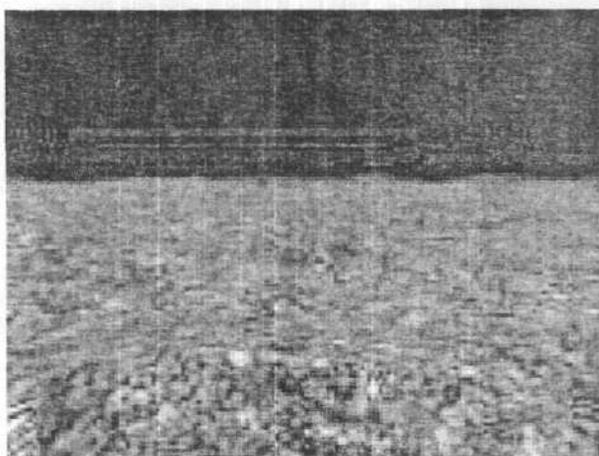
Итак, приступим к созданию подводной сценки на небольшой глубине от поверхности моря или озера.

Предположим, что погода солнечная, на поверхности лето в разгаре, волнения и ветра почти нет. В этом уроке не ставится задача заполнить водные толщи живыми существами, поэтому пейзаж будет довольно пустынным, но, тем не менее, не совсем безжизненным: воду будут пронзать преломляющиеся солнечные лучи, а по дну будут бегать солнечные блики от мелких волн. В общем, вполне реальная картина, все равно, что нырнуть в подводной маске где-нибудь на чистом озере.

При создании поверхности дна используется объект **Box** с параметрами:

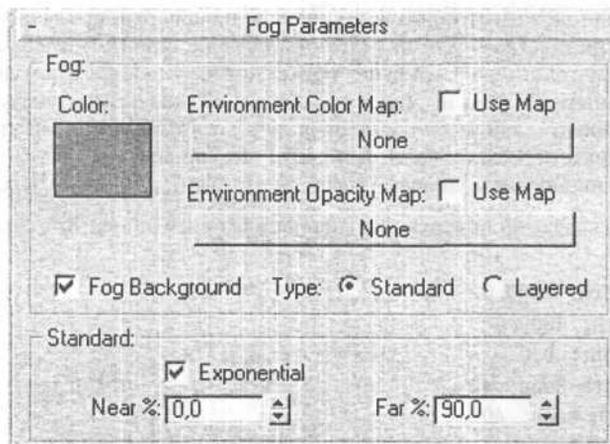
Length: 1000,0  
Width: 1000,0  
Height: 0,0  
Length Segs: 40  
Width Segs: 40

Присвойте объекту модификатор **Noise** с параметрами **Strength: Z: 20,0** и **Fractal**, остальные параметры по умолчанию. Теперь выберите материал с каменной или песчаной структурой, присвойте его **Box**'у и дно готово.

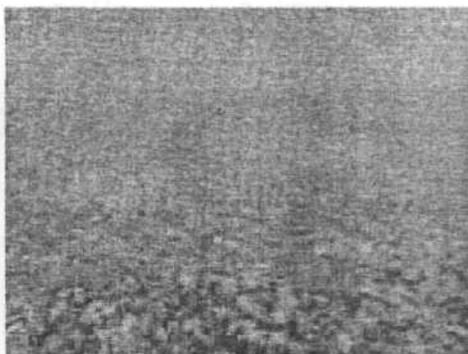


Дальше будем создавать эффект водной толщи.

В главном меню, пункт **Rendering**, войдите в **Environment** и в разделе **Atmosphere** при помощи кнопки **Add...** присоедините эффект **Fog** (туман) с параметрами как на рисунке.

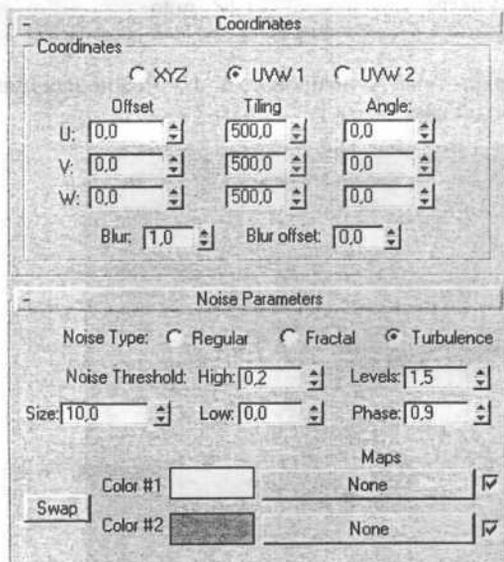


Чтобы туман был на нужном месте, а не где попало, необходимо сцену рассматривать через камеру с параметром **Far Range: 1000,0** — **1500,0**, а **Near Range** оставить равным **0,0**.

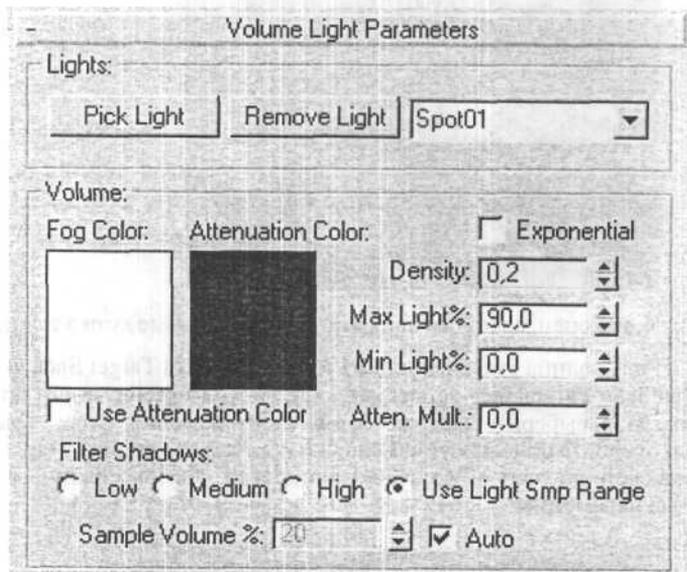


И, наконец, создадим блики на дне и солнечные лучи в толще воды.

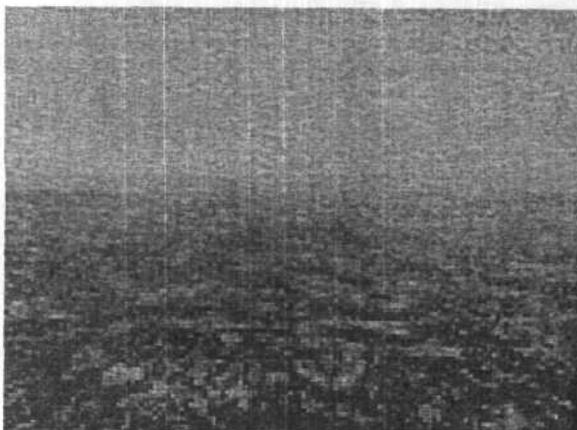
Роль солнца будет исполнять источник света **Target Spot**. В области **Spotlight Parameters** установите параметр **Projector** и, щелкнув по кнопке **Map**, выберите материал **Noise**. Чтобы изменить параметры материала, откройте окно **Material Editor**'а и перетащите кнопку **Map** из параметров света на свободную ячейку материала. Теперь можно выставить нужные параметры.



Для вывода на экран лучей света в **Environment'e** необходимо выставить еще один эффект **Volume Light**.

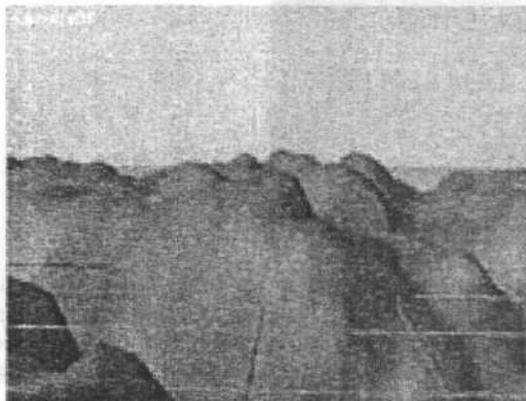


Вот и все, сцена готова. Анимировать блики можно изменяя в материале **Noise** параметр **Phase**.

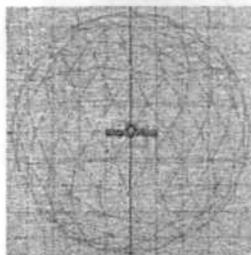


## Глава 18. Как сделать облака

Итак у нас есть ландшафт.

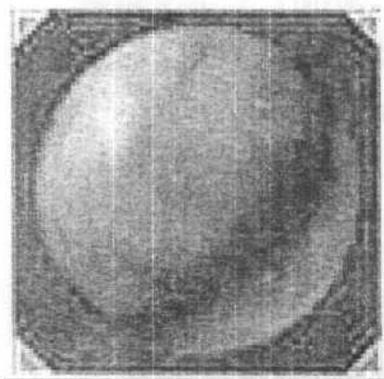


Нужно его оживить. Можно наложить на него текстуру. Подберем освещение, но нам нужно добавить небо. Делается это так. Рисуем сферу, которая заключала бы в себя всю нашу сцены. Можно сделать большую сферу.



Теперь займемся подготовкой материала для нашего неба. Для того, чтобы иметь возможность анимировать его в дальнейшем, воспользуемся следующим вариантом: создадим материал диффузной картой которого будет Noise.

Вот как он будет выглядеть:



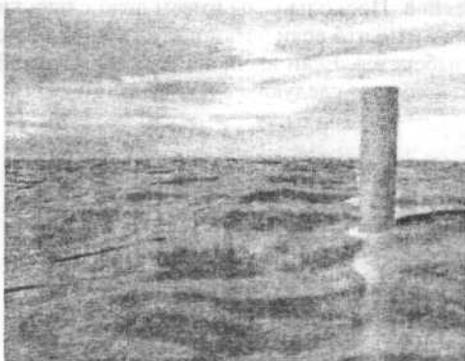
Назначим его на нашу сферу. Попробуем отрендерить... Ничего не видно? Конечно, потому что мы не воспользовались модификатором **normals**. Нам нужно как бы вывернуть наизнанку нашу сферу, то есть повернуть ее нормали внутрь. Делается это путем включения **flip normals**.

Ну вот и наш ландшафт с облаками. Для пушего реализма я добавил солнце с бликами. Делается с помощью махинаций в **video post** (путем добавления стандартного плага под названием **Lens Flare**).

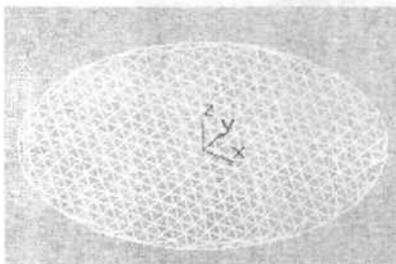
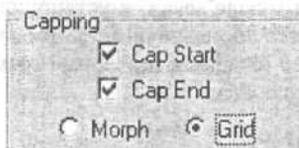


## Глава 19. Море, волны...

Взгляните на эту картинку:

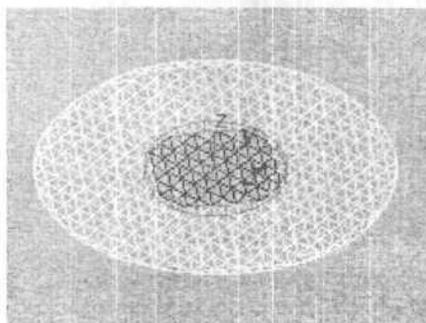


Начнем с простого — со сплайна, или, точнее, с окружности. Сделайте большую окружность, которая послужит контуром вашего океана. Затем примените к ней модификатор **extrude**. Не забудьте установить для **capping** опцию **grid**.

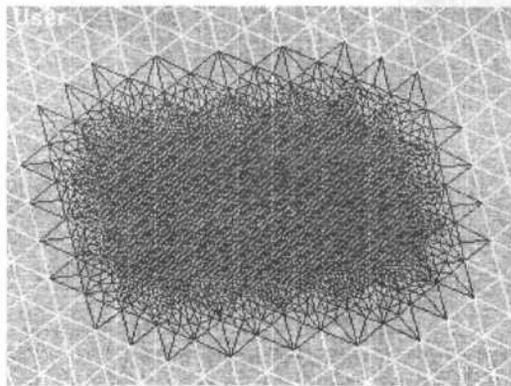


У вас получится нечто в этом роде. Мы еще недостаточно продвинулись, чтобы можно было что-нибудь напугать. Затем следует применить модификатор **mesh edit**. Теперь нужно сделать волны. Прежде всего,

выберите нужные для этого грани. Конечно, можно выбрать их и вручную, но это доставит большое неудобство в случае, если захочется поменять положение камеры. Воспользуемся для этого модификатором **volume select**. Таким образом мы сможем динамически выбирать грани, содержащиеся в контуре (**gizmo**). Установите параметры модификатора **select faces** и **cylinder selection**. Поскольку мы имеем дело с плоским объектом, вы можете обнаружить, что если установите гизмо слишком маленьким, то ничего не выберется. Если вы проведете 2D масштабирование в плоскости X/Y, то получите следующее изображение:



Теперь у нас выбран небольшой участок, состоящий из граней. Перед использованием модификатора **volume select**, я установил видимость всех граней сетки, чтобы видеть все происходящее. Затем я применяю к выбранной области модификатор **tessellate** с двумя или тремя итерациями. У вас должно получиться следующее:

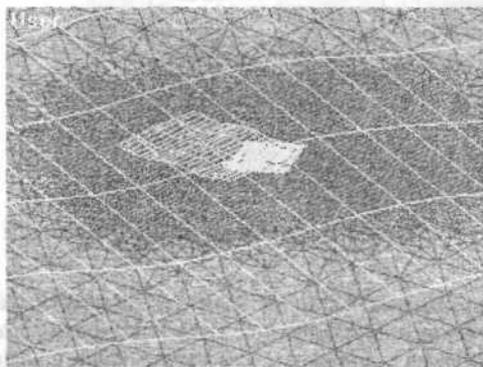


Посередине круга появится густая сетка. Наступила пора сделать волны. Это можно было бы сделать при помощи модификатора *wave*, но, боюсь, тогда придется слишком много думать. Итак, воспользуемся пространственными волнами.

Сейчас я собираюсь пристегнуть три волны — одну очень большую, которая изобразит движение типа прилива, и еще две маленькие почти одинаковые волны, повернутые на разные углы относительно большой волны.

Не забудьте анимировать фазу волн, чтобы было видно, как они движутся. Я использовал очень небольшую фазу — до 3 на 500 кадров. Постфактум, мне кажется, что это слишком долго, но, в общем, получилось ничего. Фаза большой волны должна быть совсем маленькой. Имейте в виду, что объекты, движущиеся медленно, кажутся больше.

Это картинка с волнами.



Не забудьте связать волны с поверхностью воды, — воспользуйтесь для этого кнопкой:

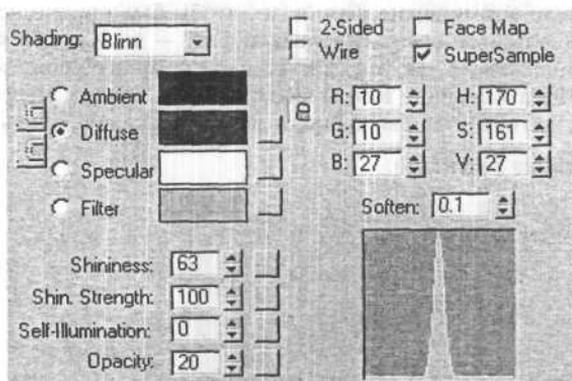


Поскольку воздействие будет оказано только на мозаичные участки (*tessellated*), внешняя область круга останется ровной, то есть горизонт будет спокойным. Если же вы хотите увидеть волнение и по краям, как в пруду, то ничто не мешает вам применить *tessellate* после *mesh select*.

Чтобы добиться натурализма, добавим текстуру. Итак, какие же у воды свойства? Только не говорите, что она мокрая. Она прозрачная, зеркальная, и искажает предметы, когда смотришь сквозь нее.

Включаем редактор материалов. Создаем новый материал **water**. Задаем прозрачность 20% — не стоит задавать 0, поскольку вода, все-таки, не совсем прозрачна. Теперь установим цвета для **diffuse** и **ambient** — темно синий и очень темно синий соответственно. Чаще всего вы не будете видеть эти цвета, но все равно нужно их определить.

Для цвета **specular** назначьте белый. Вода блестит, поэтому определите нужным образом параметр **shininess**. Пусть у графика яркости будет такой же высокий пик, как на рисунке ниже.



Скопируйте все установки с картинки за исключением **super-sample**.

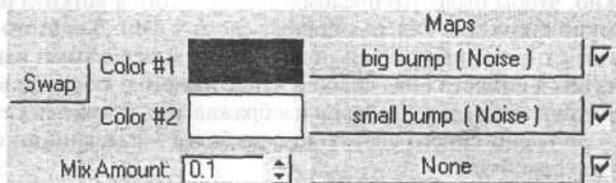
Созданные нами пространственные волны создадут движение воды, но у реальной воды бывает куда более сложная текстура поверхности. Чтобы решить эту проблему, прибегнем к помощи **bump mapping**, это инструмент для создания иллюзии текстуры без реального ее моделирования.

Классический метод создания текстуры воды — использование карты шума (**noise**). Мы воспользуемся этим приемом в несколько переименованном виде. Построим двухуровневую карту. Для этого я взял карту **mix**, нажав на кнопку рядом с надписью **bump** и выбрав ее из списка. Карта типа **mix** предназначена для смешивания между собой двух исходных карт.

Перейдите к первой компоненте и выберите шум. Назовите эту карту «**Big noise**». Установите размер для шума приблизительно 25. Затем перейдите ко второй компоненте, снова выберите шум и назовите карту «**small noise**». Установите размер этого шума в 1 и поднимите нижний порог до 0.4. Это нужно, чтобы изменить разницу между белым и черным.

С такими параметрами мы получаем очень темный фон с ярко-белыми вкраплениями.

Вернитесь к смешанной карте и установите соотношение между двумя типами шума. **0** означает присутствие только шума «**big noise**», а **1** — что у вас используется только «**small noise**». Я воспользовался величиной в районе **0.1** — много «**big noise**» с едва заметным налетом «**small noise**».



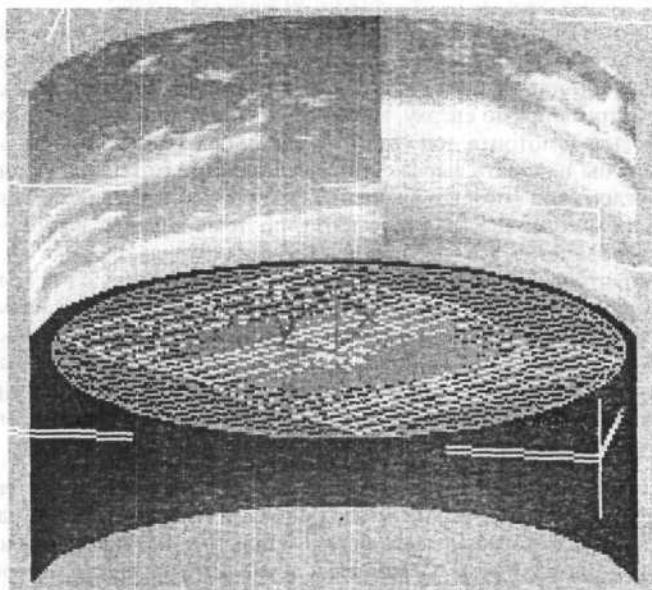
С этим материалом осталось сделать еще пару фокусов — добавить немного отражающей способности и пустить рябь. В реальности отражающая способность зависит от угла, под которым мы смотрим на воду. Если вы смотрите прямо сверху, то она вообще почти ничего не отражает. Но если вы посмотрите под углом, то отражающая способность возрастет. Я мог бы написать выражение, связывающее отражающую способность с углом, под которым размещена камера, но оно слишком сложно. Это была бы довольно трудная работа, поэтому я решил выбрать такое значение параметра, которое подойдет во всех случаях. Прежде всего, необходимо, чтобы синий цвет отчетливо просматривался во всех случаях, поэтому я установил значение **20%**. Для отражения я собирался воспользоваться готовой картинкой, но потом решил, что сгенерированное отражение будет иметь лучшую детализацию. На этом этапе нужно выбрать, что именно будет отражаться, и настроить окружение. Я выбрал в редакторе материалов пустой слот и нажал кнопку «**get material**». Затем я выбрал из списка **bitmap** и взял для отражения картинку с облаками.

Стоит удостовериться, что **mapping type** имеет параметры **environment** и **spherical**. Чтобы усложнить картинку, я несколько раз ее размножил. Теперь надо перейти в **environment setting** в меню **render**. Нажмите на кнопку рядом с надписью **background** и перейдите в редактор материалов. Выберите текстуру облаков. Это означает, что в воде будут отражаться облака. Теперь надо сделать рябь. Это называется рефракцией. Существует три вида рефракции:

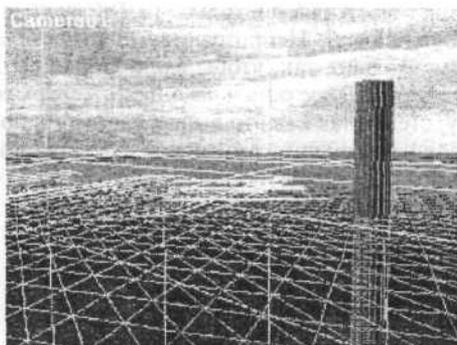
- ◆ **refelect/refract**
- ◆ **thinwall refraction**
- ◆ **raytraced**

Я перепробовал все, причем неоднократно. Но лучшим в этой ситуации оказалась **thinwall**. Этот вид хорош еще и тем, что у него мало настроек. Я установил **bump effect = 3**, чтобы больше внимания уделялось карте неровностей. Уровень рефракции в данном случае равен **50%**, а **IOR** (индекс рефракции) — равным **1.8**. Я знаю, что это не IOR воды, но выглядело все прилично.

Нужно, чтобы было что преломлять. Для этого я воткнул ржавый металлический стержень. Теперь осталось сделать фон. Для этого я взял два цилиндра с отрезанным верхом и дном, и вывернутыми наизнанку. Один из них я поместил над водой и нанес на него изображение неба, а другой поместил под воду, чтобы он изображал дно. Я закрасил его градиентом — от темно-синего сверху до еще более насыщенного синего снизу. Все это сработало.



Затем я поместил внутрь камеру. Вид из этой камеры:

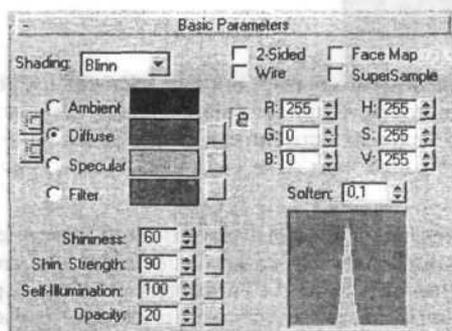


Вот и все. Еще я добавил немного **motion blur** для воды, взяв стандартные значения. Затем запустил рендеринг через **video post**, воспользовавшись фокусом **lensFX** (который работает только в одном случае из десяти, но на этот раз все прошло гладко). Возможно, если бы я переделывал это сейчас, то сделал бы волны немного побыстрее, но в остальном все получилось вполне прилично.

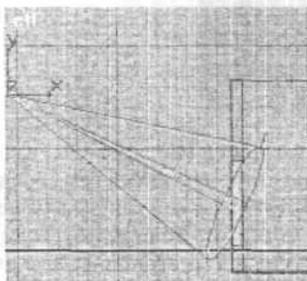
## Глава 20.

### Объемный свет на примере мозаичного окна

На этом уровне вы уже умеете рисовать раму и вставлять туда стекла, поэтому сразу перейдем к созданию материала для цветных стекол. Начнем с красного. Откройте панель **Material Editor** и выставьте следующие параметры:

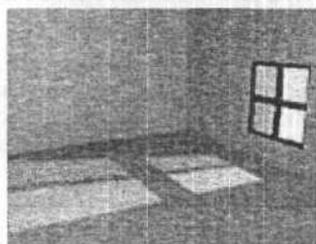


Как вы заметили, цвет выставляется только на **Diffuse** (цвет стекла) и **Filter** (просвечиваемый цвет). Непрозрачность, самосвечение и блеск подберите экспериментально, так, чтобы ваше стекло выглядело как настоящее. Так же создаются и все остальные цвета. Теперь соберите все в кучу: стекла, раму, стены, так как это было у вас задумано (в моем примере это просто маленькое окошко в комнате) и осветите мозаичное окно с внешней стороны так, чтобы луч света проходя через стекло падал на пол. В качестве источника света выберите **Target Spot**.



В разделе **Shadow Parameter** установите метку на **Cast Shadows** и **Use Ray-Traced Shadows**.

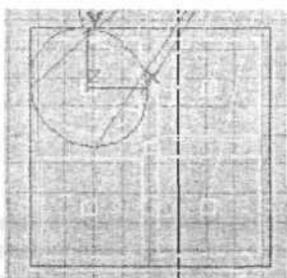
После рендеринга у вас должна получиться вот такая сцена.



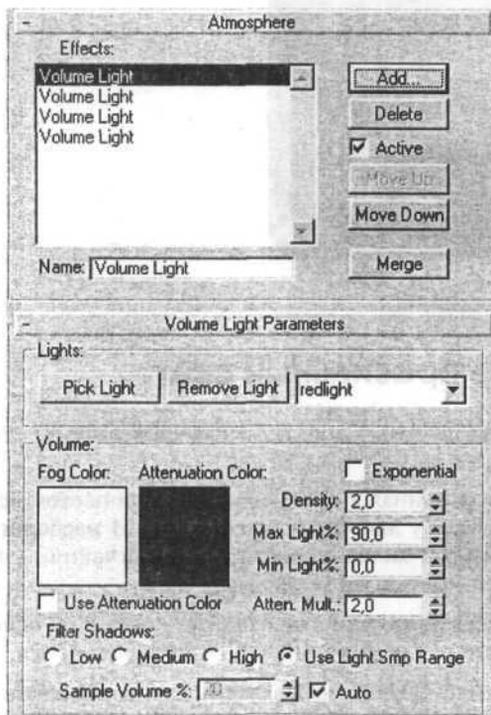
В большинстве случаев этого бывает вполне достаточно, но можно еще добавить реалистичности — придать видимость лучам света. Достигается это созданием отдельных источников **Target Spot** с одним центром, но с различным направлением луча — для каждого стекла свой источник света.

В **General Parameters** для каждого источника, нажав кнопку **Exclude**, отключите освещение объектов, на которые падает луч, чтобы не было лишних световых пятен, а в **Spotlight Parameters** установите размер светового пятна не больше цветного стеклышка, на которое падает этот

луч. Для удобства определения размера светового пятна располагайте концы лучей прямо на уровне поверхности стекла.



Затем в главном меню **Rendering/Environment** добавьте для каждого луча эффект объемного света, например, с такими параметрами:



Естественно, для каждого луча **Fog Color** должен быть своего цвета.



Ну, вот в принципе и все.

Один маленький совет: если в вашей сцене не четыре, а несколько десятков или сотен стеклышек, нет нужды включать сотню источников света, можно раскидать несколько цветных лучей по краям, а по середине пустить один белый луч.



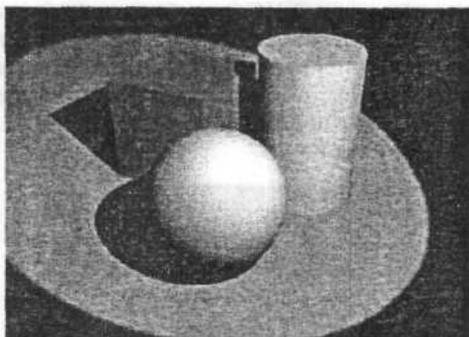
## Глава 21. Как создавать правильный свет

Освещение — фундаментальная составляющая выразительности вашей сцены.

Сколь угодно вылизанная и сложная модель при плохом освещении ничем не выигрывает у любительских поделок. И наоборот, правильный, продуманный свет заставит поверить, что ваш чайник — настоящий (и необязательно делать его зеркальным).

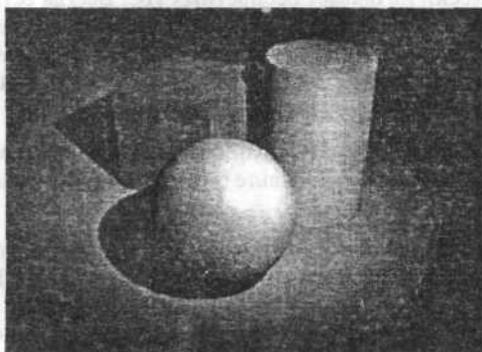
Итак, предположим, что вам уже известно кое-что об основных типах света, применяемого для имитации студийного освещения.

Во многих 3D программах все направленные источники света (далее ИС) по умолчанию имеют довольно большую область максимальной освещенности «hot spot», немногим меньшую чем граница затухания «fall off».



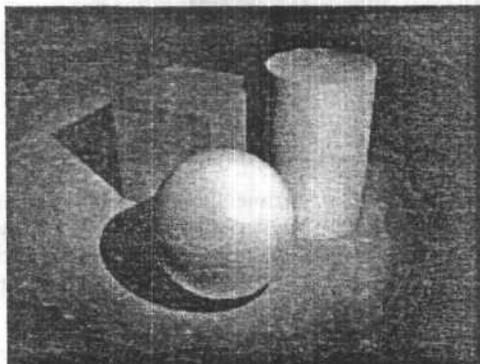
В реальной жизни таких ИС практически не бывает, особенно если речь идет о лампах накаливания и других точечных ИС. В результате мы имеем характерные пересвеченные области на больших плоскостях и мертвые зоны абсолютно однородного освещения, придающие картинке неестественный, жесткий компьютерный вид.

Возьмите себе за правило любому вновь созданному направленному ИС задавать минимальное значение «hot spot».



Полное отсутствие рефлексов (вторичного освещения, возникающего за счет отражения падающего света от светлых поверхностей) гарантировано только в открытом космосе у объекта, рядом с которым нет ничегошеньки вокруг на сотни парсек. Именно такое освещение вы получите осветив вашу сцену единственным ИС. В реальной жизни любой объект освещен, помимо основного ИС, массой рефлексов от своего окружения. Например, ярко освещенный настольной лампой шарик всегда подсвечен снизу светом, отраженным от поверхности стола (не путать рефлекс с зеркальным отражением).

Поставьте под стол маломощный ИС (не забудьте отключить у него функцию отбрасывания теней!), и пусть он освещает ваш персонаж снизу, и немного спереди. Если стол цветной, то не забудьте назначить этому ИС цвет, совпадающий с цветом рефлексирющей поверхности (стола). Обратите внимание на цветной рефлекс, отбрасываемый кубом на боковую затененную сторону цилиндра:

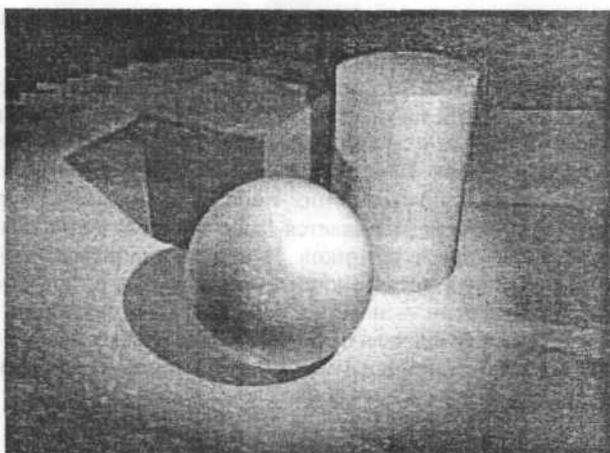


В жизни практически не бывает бесцветных ИС. Всегда есть оттенок, придающий холод свету неоновой лампы, и теплоту солнечному свету и свету ламп накаливания. Только в 3D программах по умолчанию все ИС безжизненно и бесцветно серо-белые. Соответственно, ваша сцена с таким светом будет обладать мертвецкой выразительностью.

Придайте основному вашему ИС оранжевой желтизны, и в сцену проникнет солнечное настроение. Добавьте лунной синевы, и ночная прохлада окутает ее.

Если ваш персонаж — логотип любимого заказчика, или упаковка его продукции, то сам бог велит осветить его цветным светом. В этом случае, особенно если применяется яркое контрастное освещение, следует позаботиться и о цвете дополнительного и контрового освещения и теней.

Наше зрение устроено так, что темные области при контрастном освещении нам кажутся окрашенными в цвет, дополнительный к цвету основного освещения. Например, тени от ярко освещенного теплыми оттенками артиста на сцене кажутся синими. К счастью, во многих современных 3D программах есть возможность задать цвет тени. В крайнем случае нужно просто подсветить теньевые области цветными ИС.



Хотя, как мы убедились, одного ИС почти всегда недостаточно, не стоит увлекаться и перебарщивать с количеством ИС в вашей сцене. Часто уже при четырех и более ИС очень трудно оценить вклад каждого из них в общую картину освещенности. Поставив, как вам кажется, идеальный свет для первого кадра анимации используя всего десяток ИС, вы обнаружите, что к десятому кадру ваш центральный персонаж переместился так, что его освещение безнадежно испорчено. Что делать? Анимировать еще десять ИС?

Избежать этой проблемы поможет простое правило. Максимально используйте возможности уже существующих в сцене ИС, прежде чем решитесь на создание нового. Для избежания «пересвета» поверхностей несколькими ИС используйте включите свойство затухания силы света на расстоянии. Исключайте из освещения дополнительными ИС второстепенные объекты. Кроме вашего времени на настройку освещения этим вы сократите и время на просчет сцены.

## Глава 22. Раздельное освещение и затемнение сцены

Крупные части изображения, не обязательно 3d, их внешний вид, композиция действуют на сознание зрителя, а мелкие — на подсознание. Картинка «смотрится» тогда, когда мелкие ее детали не вызывают у человека подсознательного конфликта с тем, что он привык видеть. То есть

человек может даже не понимать, что конкретно его смущает. Задача художника — учитывать все мелочи при построении изображения.

При моделировании (особенно если нужно добиться «фотореализма») огромное влияние на качество имиджа оказывает правильное освещение сцены. В реальной жизни любой предмет освещается далеко не одним источником света. Первоначально их может быть один, два... (скажем солнце, лампа), но свет от них, попадая на предметы, окружающие моделируемый объект, отражается (рассеивается) и тем самым мы получаем еще множество источников. И поэтому добиться требуемого результата, манипулируя источниками света, которые одновременно и освещают, и создают тени — нелегко, а в некоторых случаях — невозможно. Гораздо удобнее сначала осветить предмет так, чтобы он выглядел «как живой», а затем, независимо от освещения, создать нужные тени. 3ds max позволяет это сделать достаточно просто.

А теперь — суть метода:

1. Освещаем предмет (сцену) традиционно. (Только у источников света не включаем **Cast Shadows**).

2. Создаем источник света. Например — **Spot01**.

3. Копируем его так, чтобы копия осталась в том же месте, что и оригинал.

4. Далее в **Spot01** включаем **Cast Shadows**.

5. В **Spot02 multiplier** устанавливаем отрицательным, но равным по модулю **multiplier** в **Spot01**.

Теперь все параметры, кроме теней, **Spot01** компенсируются **Spot02**. Мы получили сложный источник света (**Spot01+Spot02**), который создает тени и не влияет на освещенность сцены! Для удобства можно «подлинковать» оба источника к **Dummy**-объекту. А используя **Exclude/Include**, можно добиться влияния источников только на данный объект(ы).

Тут есть большое поле для экспериментов. Например, можно и в **Spot02** включить **Cast Shadows**, но в **Spot01** поставить **Sample Range = 5** (к примеру), а в **Spot02** — равным 0. Тогда мы получим только контур тени объекта. И так далее.

Кто-то может спросить: «К чему все это — ведь в **Exclude/Include** есть опция **Exclude/Include cast shadows**». Действительно — есть, но при этом (исключении создания теней) источник света продолжает освещать предмет(сцену), то есть, чтобы не менять освещенности — одним источником все-равно не обойтись.

Кроме того, все написанное выше — не догма, а совет. Не обязательно им всегда пользоваться, к каждой конкретной ситуации нужно подходить по-своему. 3ds max 7.0 хорош тем, что один и тот же результат можно получить 1000 способами.

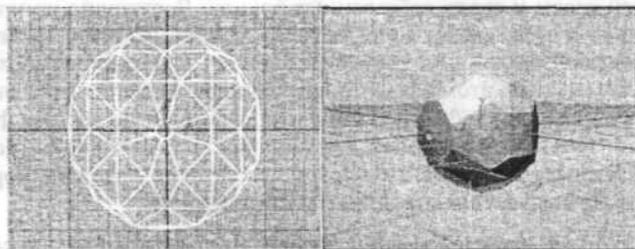
## Глава 23.

### Бриллианты, рубины, аметисты... или как делать бижутерию

Подарите своей девушке кольцо с бриллиантом карат на ...дцать! Ах, студент... Тогда хотя бы красивую бижутерию... Ах, вы учитесь в другом городе... Что сделаешь? А мечтать-то вы хоть не разучились? Тогда — вперед, за МАХ, за работу! Если мечтать умеете, то все будет — и бижутерия, и бриллианты... И не только в МАХ, но тоже не сразу.

Вначале нужно создать форму будущего граненого бриллиантика — в данном конкретном случае я пошел по простейшему пути — примером нам будет служить обычная бусинка. Почему не камешек огранки под колечко? Потому, что это не просто **Sphere** без сглаживания — натуральная огранка гораздо сложнее. А в таких делах, как преломления-отражения, решающим фактором выступает форма — ваш бриллиант не зайграет, если был неправильно огранен.

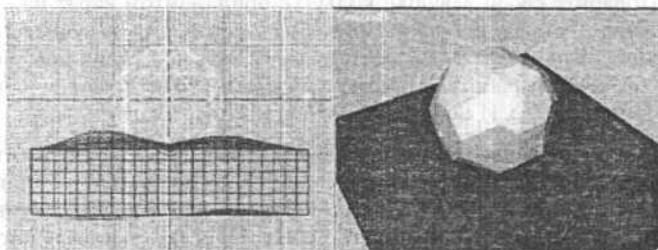
Для бусинки есть отличный примитив — **Hedra**, все уже практически готово, только дырочку проделать в нужном месте.



Здесь кроется еще один подвох, важный при визуализации прозрачных и отражающих предметов, то, что мы часто не замечаем в жизни, потому что оно есть всегда и уже стало привычным — это окружение! Вокруг нас, где бы мы не находились, всегда есть куча предметов, они-то и преломляются, образуя тот самый желанный живой рисунок, полный нюансов. А окружение вокруг отображаемого предмета можно «уплотнить» двумя путями — «навалив» вокруг геометрии и добавив геометрии

вокруг. Естественно, что истина опять «где-то между» — мы добавим немного предметов, подушечку и поверхность стола, а также воспользуемся созданием окружения (environment) для материала.

Итак, у нас есть стол, на нем пуфик, на нем бусинка. Обязательно надо добавить свет, в данном случае подойдет парочка спот-источников — рисующий и «в лоб» (почти), оба с небольшим центральным лучом и широким конусом затухания. Оба источника должны отбрасывать тени, лучше воспользоваться типом «Shadow Maps», иначе прозрачные объекты дадут нереально слабую тень. А теперь подойдем к самому сердцу процесса — созданию материалов.



Именно материал определит, насколько хорошо мы справились с задачей. Не буду тратить время на окрашивание стола с пуфиком — посмотрите прилагаемую сцену, да и для бриллиантика всего расписать не буду, лишь обратим внимание на ключевые моменты. Как я уже упоминал, «игра» кристалла складывается из отражения и преломления окружающей среды и своих собственных бликов. Достаточно понять, что это за блики и сразу станет ясно, «как его делать». Из основ оптики мы знаем, что такое разложение света через призму. Однако при работе в 3D многие забывают об этом эффекте или начинают искать софт, считающий дифракцию. Зачем?! Ведь нам не нужна абсолютно честная физика и оптика — нам нужен результат, а вот он прост для всех линз — появляются «радужные» цвета. Одним словом, достаточно наложить корректную карту **environment** для отражений и преломлений!

К счастью, рэйтрэйсовый материал 3ds max позволит нам сделать это, и не только это. Требования, предъявляемые к карте отражения таковы. Должны быть «чистые» цвета, чередуемые контрастно по яркости и накладываемые способом «additive» на цвет предмета — можно воспользоваться процедурным градиентом, причем необходимо переключить в материале тип наложения отражений — делается это в закладке расширенных параметров. Требования к преломлениям практически идентичны, я бы лишь посоветовал сделать небольшой разброс по цветам в этих двух картах. Кроме того, необходимо настроить сильный блик,

не забыть включить «2 sided», сила отражения должна лишь немногим превышать половину. Во избежание отражения затененных дальних участков сцены рекомендую пользоваться параметрами **Reflection & Refraction Falloff**. Это, пожалуй, и все.

Честно говоря, для окончательного «оживления» этой прелести не помешает добавить «преломленный» бриллиантом свет. Оно, конечно, мелочь, а зато какая...



## Глава 24. Создание молнии

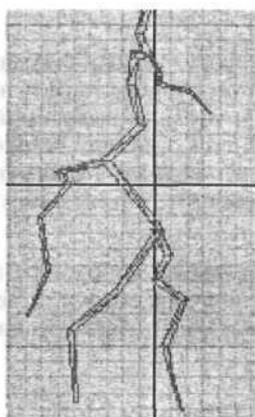
Рисуем каркас будущей молнии.

Для этого расположите во фронтальном окне проекции линии так, как представляете себе молнию.



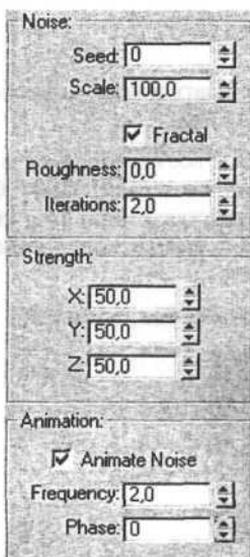
Затем, используя инструмент **Loft**, создайте объемные кривые линии. В качестве **Shape** можно использовать объект **Rectangle**. Во вкладке **Modify** установите для каждого полученного объекта значение **Shape Steps** и **Path Steps**, равное нулю.

В свитке **Deformations** откройте панель деформации **Bevel** и для каждого луча молнии установите свои значения так, чтобы сверху вниз было небольшое уменьшение толщины луча. Например, для самого длинного луча — значения **0,0 — 0,0** и **100,0 — 1,0**, а для самого верхнего короткого — значения **0,0 — 1,0** и **100,0 — 2,0**. Теперь, используя булевскую операцию сложения (**Union**) соедините все лучи в один объект.



Создайте в **Material Editor**'е материал с самосвечением (**Self-Illumination**) равным 100, того цвета, какого по вашему должна быть молния и присвойте его объекту.

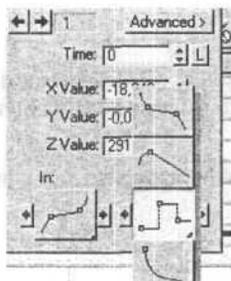
В командной панели во вкладке **Modify** выберите модификатор **Noise** и установите следующие параметры:



Переходим к анимации.

Так как настоящая молния сверкает какие-то доли секунды, то и наша молния из **100** по умолчанию кадров будет присутствовать на экране кадров **5**. Сместите созданный объект за пределы выводимой части проекционного окна и нажмите большую кнопку **Animate**. Установите в окошке текущего кадра **20** кадр и переместите молнию в видимую часть экрана. Затем установите **25** кадр и снова сместите объект с глаз долой. Отключите анимацию.

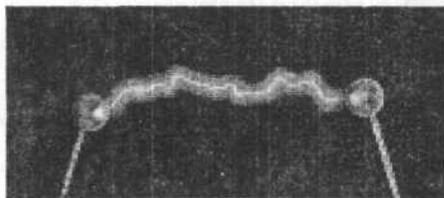
Откройте окно **Track View** и, кликнув правой кнопкой по каждой ключевой точке, измените параметры перемещения объекта во входной и выходной точке между ключевыми кадрами так, чтобы движение было скачкообразным.



С анимацией справились. Осталось придать молнии побольше реалистичности, используя фильтр **Glow**.

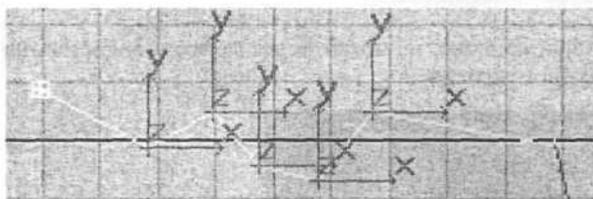
Откройте панель **Video Post** и установите все необходимое в окошке **-Queue**. Измените установки параметров **Lens Effect Glow**: Свиток **Preferences Effect** — **Size: 2,0** и **Color** — **User** (выставьте такой же цвет как и в **Material Editor**'е). Все, молния готова. Для вывода конечной анимации естественно надо использовать **Video Post**.

А можно нарисовать и такую молнию даже без лофтинга.



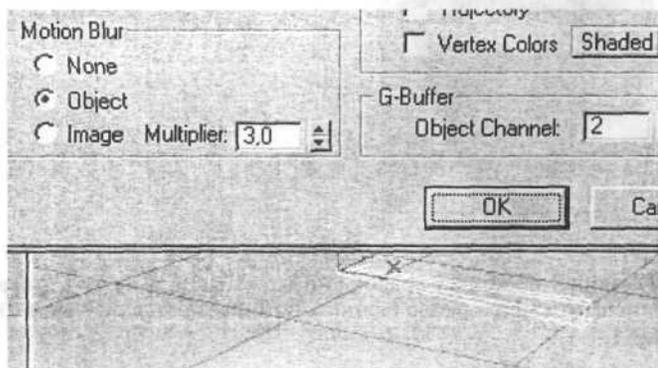
## Глава 25. Создание фотонной пушки

Идея проста, но, на сколько я понимаю, в фильмах и заставках к игрушкам делается все именно таким образом. Для электроразрядов использованы ломанные линии с выставленным параметром **Renderable**. Принципы анимации описаны выше, только перед назначением модификатора **Nois**, чтобы первая и последняя точка были неподвижны, надо, используя **Edit Spline**, выделить все точки на линии кроме первой и последней и только после этого применять **Nois**.

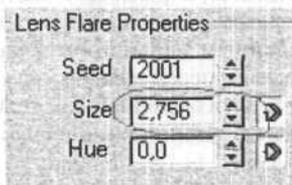


Чтобы при вращении всей установки разряды не разбежались во все стороны, надо в **Hierarchy** сдвинуть все **Affect Pivot Only** к центру вращения.

Для центрального лазера использован конус, радиусы которого поначалу одинаковы, а затем дальний от установки немного расширяется и под конец весь конус сдвигается в направлении выстрела. Все операции прodelываются со включенной анимацией. Для придания реалистичности в **Video Post**'е к цилиндру применен **Lens Effect Glow** и в **Object Properties** установлен **Motion Blur**.



В точке схождения разрядов и лазера установлен источник света **Omni**, к которому применен **Lens Effect Flare**. В начале размер излучения возрастает с **0** до **40**, а затем, после начала движения цилиндра (выстрела), резко гаснет до **0**.

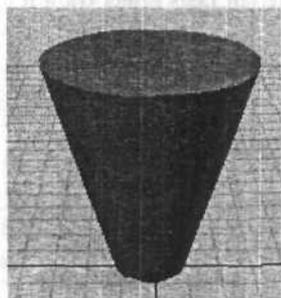


Картинка, конечно, не может передать всех тонкостей, на самом деле сцена намного красивей.



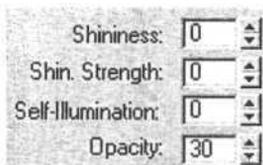
## Глава 26. Простой способ создания пара

Для создания пара понадобится изготовить простой конус.



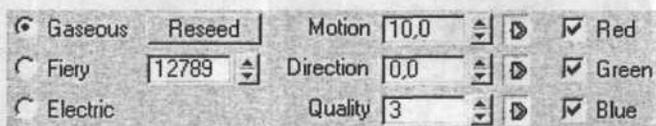
Кликните по конусу правой кнопкой мыши, выберете **Properties...** и в появившемся меню около надписи **G-Buffer** поставьте **Object Channel:** значение **1**.

Назначьте полученному объекту материал с такими параметрами.

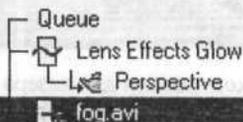


В главном меню, команда **Rendering**, выберите функцию **Video Post**, нажмите кнопку **Add Scene Event** и установите вид на сцену (**Camera**, **Perspective**). В результате в **Video Post**'е в окошке **-Queue** появится выбранный вид. Кнопкой **Add Image Filter Event** присоедините **Filter Plug-In Lens Effect Glow** (сияние).

При помощи кнопки **Setup**, чуть ниже, зайдите в меню установки параметров выбранного фильтра. Кнопкой **Inferno** откройте свиток параметров и установите метки около параметра **Gaseous** и около всех цветовых каналов (**Red**, **Green**, **Blue**).



Подтвердите установки кнопкой **OK**. Добавьте кнопкой **Add Image Output Event** (проследите, что бы на панели **-Queue** не была выбрана ни одна установка) на панели **Video Post**'а формат и параметры выводимого файла.

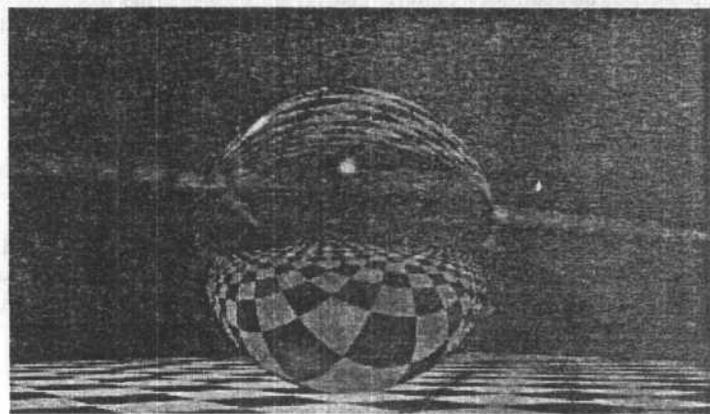


Теперь, если выделить на панели **-Queue** установку вывода файла и нажать кнопку **Execute Sequence**, готовый файл будет записываться уже с присоединенным эффектом. Осталось дорисовать на сцене сопутствующие пару объекты и у вас получится что-то в этом роде.

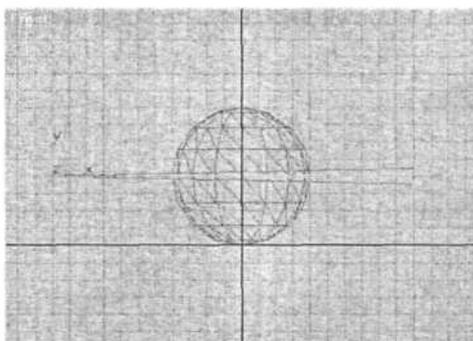


## Глава 27.

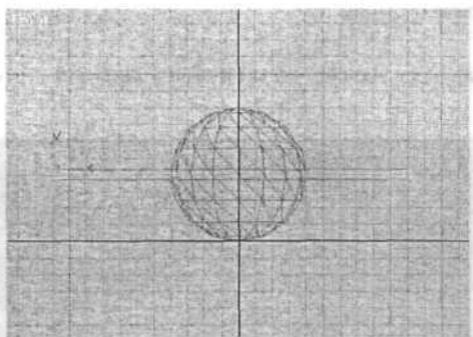
### Создание реалистичного лазера



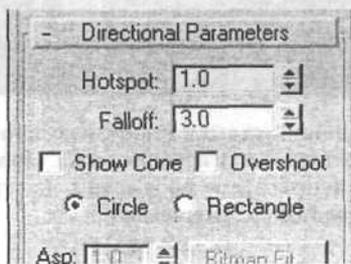
Вместо использования геометрической модели для лазера попробуем создать его с помощью светового луча, так как реальный лазер — это пучок параллельных лучей света большой плотности. С первого взгляда, для этой цели идеально подходит источник света 3ds max 7.0 — **Spotlight**, но при этом даже если мы сжимаем конус этого источника света до очень малого угла, то все равно мы никогда не получим лазерный луч.



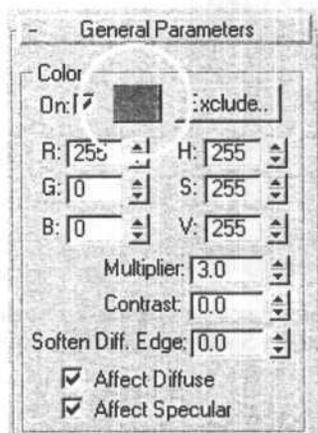
К счастью, в 3ds max 7.0 есть источник света **Direct**, который имеет параллельные лучи света, в отличие от **Spotlight**. Поэтому будем использовать его для нашего лазера.



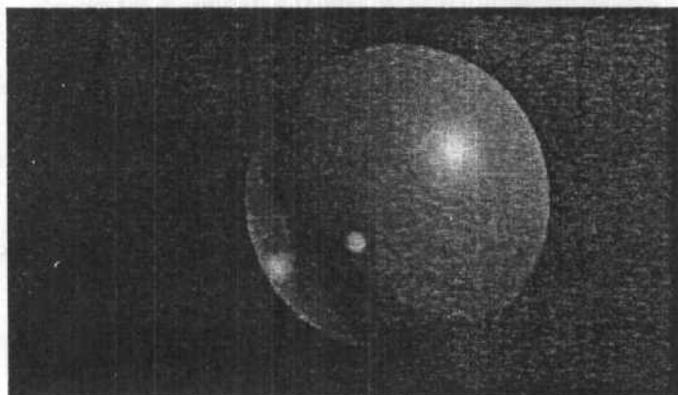
Источник света должен иметь очень узкий пучок света. Я использовал следующие параметры настройки:



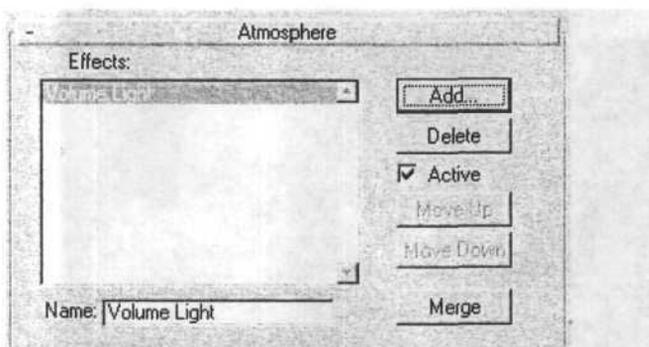
Также я использовал красный цвет для источника света и увеличил значение **multiplier**:



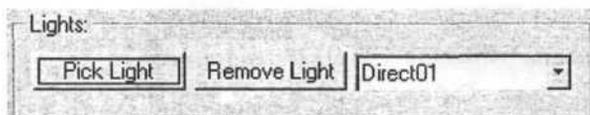
После рендера эффект лазера должен быть похожим это изображение:



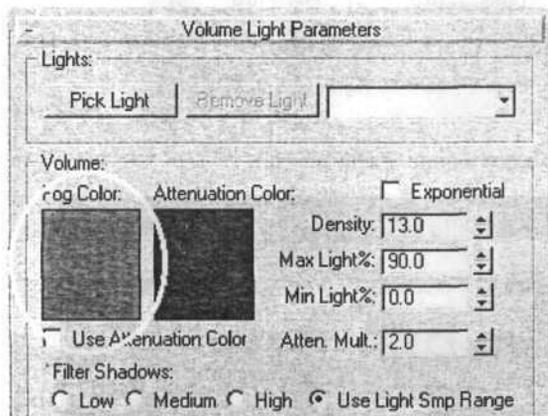
Это хорошо подходит для типичной «съемки в лоб», в которой кто-то нацелится на объект и на объекте появляется красная точка. Но мы хотим большего. Для этого мы используем параметры настройки **Environment**. Добавьте эффект **Volume Light Environment**:



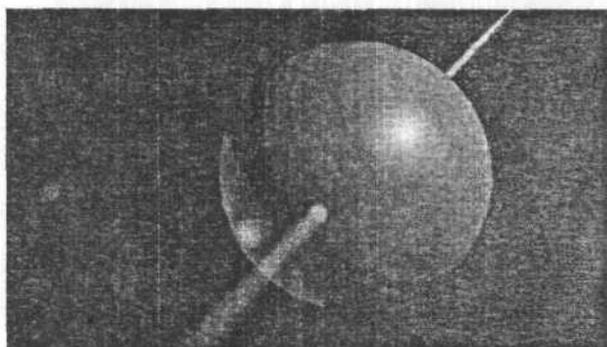
Выберите источник света **Direct** для этого эффекта:



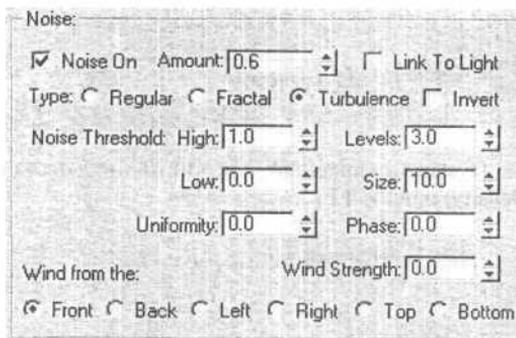
Установите цвет тумана на красный, как у источника света и установите плотность (**Density**) равное 13:



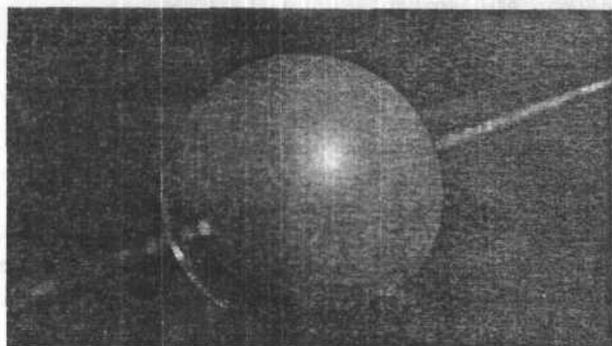
Теперь луч станет видимым. Но качество изображение низкое. Лазерные лучи никогда не видимы в чистом виде и имеют некоторое количество пыли или дыма в окружающей среде.



Включите Noise. Я использовал следующие параметры настройки:



Теперь намного лучше. Разве не так?



Это в основном все, что нужно для получения реалистичного лазера. Теперь вы можете использовать эту модель лазера в любой сцене вместе с объектами обладающими отражением или преломлением.

## Глава 28.

### Создание космического эффекта — солнце

1. В видовом окне **Top** создайте сферу радиусом **50** единиц и с числом сегментов, равным **40**.

2. В середине сферы создайте два источника света **Omni** с настройками: Для **Omni01** значения **RGB** — **255,204,0**. В группе **Attenuation** значения **Start Range** установите равным **50**, а **End Range** равным **60**. Включите опции **Use** и **Show**. Для **Omni02** значения **RGB** — **255,104,0**. В группе **Attenuation** значения **Start Range** установите равным **60**, а **End Range** равным **80**. Включите опции **Use** и **Show**.

3. Войдите в редактор окружения **Environment**. В группе **Atmosphere** щелкните на кнопке **Add...**, из появившегося списка выберите **Volume Light**. Щелкните на кнопке **Pick Light** и в любом видовом окне выберите источник света **Omni01**. В группе **Volume** измените уровень **Density** на **30**. В группе **Noise** включите опцию **Noise On**, измените значение **Amount** на **0,5**.

4. Щелкните на кнопке **Add...**, из появившегося списка выберите **Volume Light**. Щелкните на кнопке **Pick Light** и в любом видовом окне выберите источник света **Omni02**. В группе **Volume** измените значение **Density** на **30**. В группе **Noise** включите опцию **Noise On**, измените значение **Amount** на **0,98**, **Uniformity** — на **-0,1**, **Size** — на **10**.

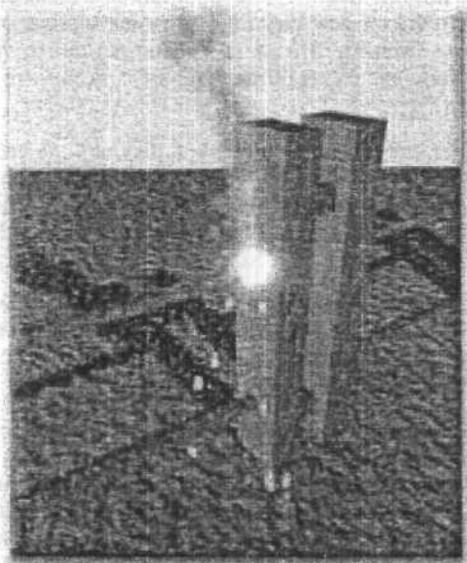
5. В группе **Background** окна **Environment** щелкните на кнопке **Assign...** В появившемся диалоговом окне **Material/Map Browser** дважды щелкните на элементе **Noise**. Выберите кнопку с надписью **Map #1 (Noise)**. В окне **Put to Material Editor** выберите **Slot 6**.

6. Войдите в **Material Editor**. Убедитесь, что активен слот 6. В разворачивающейся панели **Maps** щелкните на кнопке справа от опции **Diffuse**. В появившемся окне **Material/Map Browser** дважды щелкните на элементе **Noise**. В разворачивающейся панели **Noise Parameters** установите **Size** равным **0,1**, **Low** = **0,65**. Щелкните на кнопке с надписью **None** справа от **Color #1**. В появившемся окне **Material/Map Browser** кликните элемент **Noise**. Установите значение **RGB** для **Color #2** равными **22,33,117**.

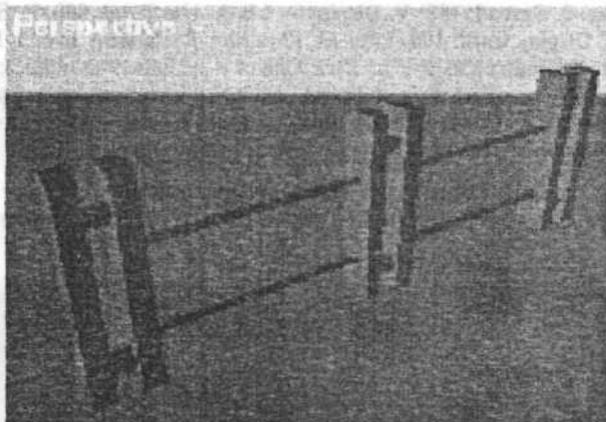
7. Активизируйте слот **1** в **Material Editor**, в разворачивающейся панели **Basic Parameters** установите значение параметра **Self-Illumination** равным **100**. В разворачивающейся панели **Maps** щелкните на кнопке справа от опции **Diffuse**. В появившемся окне **Material/Map Browser** дважды щелкните на элементе **Noise**. В разворачивающейся панели **Noise Parameters** установите **Size** равным **10**, **High** = **0,71**. Щелкните на черный прямоугольник **Color #1** и установите значения **RGB** = **246,255,0**. Щелкните на белый прямоугольник **Color #2** и установите значения **RGB** = **237,255,0**. Щелкните на кнопке с надписью **None** справа от **Color #2**. В появившемся окне **Material/Map Browser** дважды щелкните на элементе **Noise**. Установите значения **High** = **1,0**, **Low** = **0,525**. Установите значения **RGB** для **Color #1** равными **255,174,0**, для **Color #2** — **59,44,0**. В любом видовом окне щелкните на объект **Sphere01**, чтобы сделать его активным. Щелкните на кнопку **Assign Material to Selection**, чтобы назначить материал сфере. Попробуйте отрендерить сцену.

## Глава 29.

### Имитация работы сварочного аппарата



1. Создаем геометрию забора.



## 2. Объект для искр.

Создайте объект для имитации искр от сварки посредством редактирования коробки **EditMesh**, затем разместите его куда-нибудь, где его не будет видно.



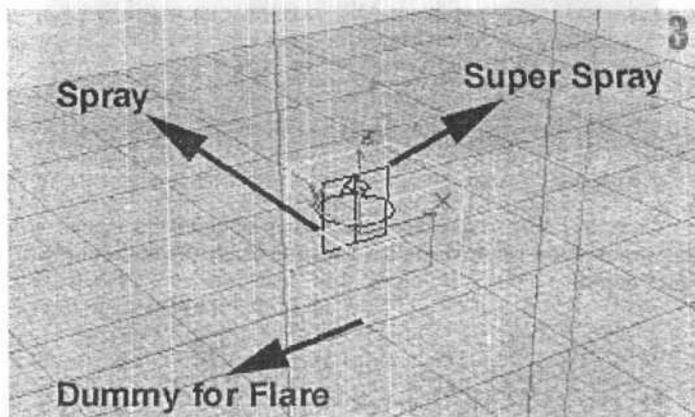
## 3. Создаем Spray и Super Spray.

**3.1.** Создайте систему частиц **Spray**, (у нас она будет использоваться для дыма) со следующими параметрами: **V. Count: 100**; **R. Count: 350**; **Drop Size: 13**; **Speed: 10**; **Variation: 1,145**; **Drops: Render Facing**; **Timing Start 2**, **Life 45**; **Width and Length 12**, затем разместите ее как на рисунке ниже.

**3.2.** Создайте систему **Super Spray** направленной вверх, (она у нас используется для искр) со следующими параметрами: **Off Axis: 0**, **Spread:**

81; Off Plane: 0, Spread: 180; V. Display — Mesh; Use Total: 500; E. Start: 1; E. Stop: 60; Display Until: 100; Life: 13; Variation: 0; Particle Size: 3; Particle type — Inst. Geometry (жмите на Pick Object и назначайте искре); в Part Rotation установите Direction of Travel/ Mblur; в Part. Spawn — Spawn on Collision; Spawns: 4; Affect: 100; Multiplier 1; Speed Chaos Factor: 100 — Fast; Scale Chaos Factor: 100 — Down, разместите как на рисунке ниже.

3.3. Создайте **Dummy**, (он будет использоваться для вспышек) и разместите его как на рисунке, так чтобы его центр не входил в геометрию:



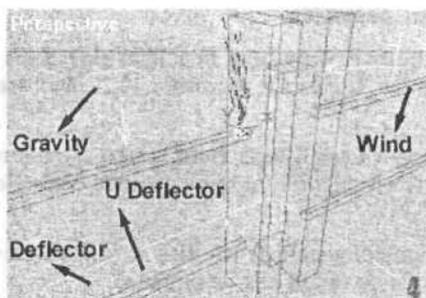
#### 4. SpaceWarp.

4.1. Создайте **Deflector**, (он будет использоваться для отскокивания частиц от «земли») установите параметр **Bounce** в **0,2**, после разместите, как на рисунке ниже и привяжите к искрам.

4.2. Создайте **UDeflector**, (он используется для отскокивания частиц от забора) установите **Bounce** равным **0,1**, теперь возьмите и сгруппируйте самый ближний столб к партиклам и две перекладины в один объект, затем свяжите все это с **UDeflectorom**. После чего привяжите **UDeflector** к дыму и искрам.

4.3. Создайте **Gravity** (гравитация, она и есть гравитация), и привяжите к искрам.

4.4. Создайте **Wind** (ветер будет придавать дыму более реальное перемещение) со следующими параметрами: **Strength: 0,13; Turbulence: 1,0; Scale: 1,0**; разместите его, как показано на рисунке, и привяжите к дыму.



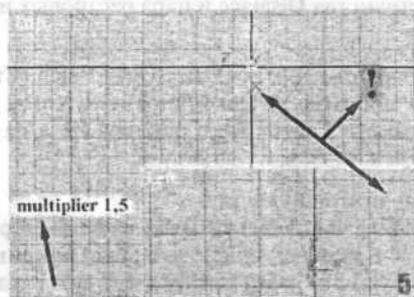
## 5. Материал для искр и дыма.

5.1. Создайте обычный материал со следующими установками: Ambient RGB: 46, 17, 17; Diffuse RGB: 191, 67,0; Specular and Filter: 0; Shininess: 0; Shin. Strength: 0; Self-Illumination: 100; Mat. Effects Channel 1. Назначьте эти параметры искрам.

5.2. Создайте обычный материал со следующими установками: Shininess: 0; Shin. Strength: 0; Self-Illumination: 0; в карте Diffuse установите Smoke: Size: 54,76; Iterations: 5; Exponent: 1,5; Color 1 RGB: 206, 206, 206; Color 2 RGB: 203, 203, 203; в карте Opacity установите Gradient: Color 1 RGB: 0, 0, 0; Color 2 RGB: 32; 32; 32; Color 3 RGB: 67, 67, 67; Gradient Type — Radial; Noise Amount: 0,3; Size: 8; Fractal; Levels: 4,0. Назначьте эти значения дыму.

## 6. Свет.

Обратите внимание на восклицательный знак. Под стрелками указан один Spot Target источник, на самом деле их там два. Установка для первого: Multiplier: 1,0; Cast Shadow; Use Shadow Map; Size 512; для второго: Multiplier: -1,0.

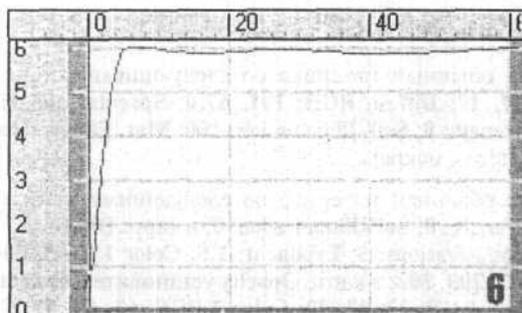


### 7. Glow and Flare + Flare Animation.

7.1. В **Video Poste** повести камеру (или перспективу); **Lens Effects Glow**, в нем установите **Mat ID: 1**; **Size: 1**; **Intensity: 10**; **Color User RGB 255, 121, 0**.

7.2. В **Video Poste** повести **Lens Effects Flare**, в нем загрузите установки файла в архиве вместе со сценой, затем нажмите **Node Source** и выберите **Dummy**.

7.3. Для анимации **Flare** войдите в **Track View** ⇨ **Video Post** ⇨ **Lens Flare** ⇨ **Size** и исправьте линию до такой как на рисунке:



## Глава 30.

### Создание анимированных объектов

#### Следы на снегу

Для создания этого эффекта требуется прямоугольник («земля»), несколько искривлений пространства **Displace** и пара растровых изображений.

Создание «земли»: в проекции **Top**, создайте **Quad Patch (Create** ⇨ **Geometry** ⇨ в меню выбрать **Patch Grids** ⇨ **Quad Patch**) с нужным размером «земли». Установите параметр **Length Segs**: равным **10**, а параметр **Width Segs**: равным **5**.

Создание следов: разверните проекцию «**TOP**» на весь экран, щелкните на **Create** ⇨ **Space Warp** ⇨ **Geometric/Deformable** ⇨ **Displace**. В месте, где должен находиться первый след, создайте искривление **Displace** с размером, соответствующим размерам нужного вам следа. Выделите искривление **Displace** и на панели **Parameters** в блоке **Image** щелк-

ните **Modify** ⇨ **Bitmap None**, выберите нужную вам картинку следа (Она должна быть уже нарисована. Причем, там где след — все белое, а земля вокруг следа — черная). Клонировите **Displace** через главное меню ⇨ **Edit** ⇨ **Clone** и установите переключатель **Copy** в диалоге **Clone Options**. Расположите новые **Displace** в нужные вам места следов.

Для правильного отображения следов на **Quad Patch** выделите **Quad Patch**, примените модификатор **Mesh Select**, выделите все плоскости, находящиеся под **Displace**. При помощи модификатора **Tessellate** при **Iterations = 2** сделайте более мелкую мозаику на выделенных областях.

Привязка **Quad Patch** к **Displace**: если **Quad Patch** по-прежнему выделен, щелкните на **Bind to Space Warp** на панели инструментов (под главным меню). Появится курсор **Bind to Space Warp**. На панели инструментов нажмите **Select by Name**, чтобы появилось окно **Select Space Warp**, выделите объект **Displace01** в списке и щелкните на кнопке **Bind** (на закладке **Modify** на панельке **Modifier Stack** для объекта **Quad Patch** появится **Displace binding**). Повторите такую привязку **Quad Patch** к остальным **Displace**.

Регулировка **Displace** производится таким образом. Переключитесь с **WireFrame** на **Smooth + Highlights** (щелчок правой клавишей на названии видового окна). Выделите **Displace01** и перейдите на закладку **Modify**. На панели **Parameters** в блоке **Displacement**: установите **Strength = 5** и посмотрите результат. Если растровый фон битмапа искажает этот кусок, делая его похожим на приподнятую рамку вокруг следа, то необходимо отрегулировать параметр **Luminance Center** искривления **Displace**: установите флажок **Luminance Center**; затем начинайте медленно изменять значение параметра **Center** в сторону увеличения. Вы увидите, как края пластинки с отпечатком следа опускаются в четырехугольный кусок. Измените **Luminance Center** до **0.5** (для каждого **Displace**). При этом края должны быть полностью плоскими.

Анимация появления следов: на закладке **Modify** сохраните установленное значение **Luminance Center** и установите параметр **Strength = 0**. Включите **Animate**, перейдите к пятому кадру и измените **Strength** на **5** и выключите **Animate**. Откройте **Track View** и разверните треки для **Displace01**. Скопируйте **Strength** из **Displace01** в треки **Strength** остальных **Displace**, сместив ключи по треку для поочередного появления следов.

Когда вам нужно смоделировать постепенно появляющийся след от танка или автомобиля, вместо картинки следа человека подставляйте последовательность кадров (через **AVI**, **FLC**, или другим форматом), на которых белая полоса «следа» на черном фоне проходит от одного края изображения до другого. При этом можно создать только один **Displace** (или два **Displace** — если ваш танк умеет ездить на двух гусеницах).

### Настройка и анимация компьютерной куклы

Одно из базовых правил персонажной анимации состоит в том, что анимация не происходит из точки А в точку В, она происходит в точке А и в точке В (подразумевая под этими точками ключевые позы, замах и перелет, из которых и состоит анимация). Создавая куклу, позирование которой не будет составлять слишком большого труда, надо позаботиться о следующем:

- ◆ Метод управления направлением коленей независимо от стоп и бедер.
- ◆ Стопы должны поворачиваться на 180° в разные стороны.
- ◆ Метод поворота стопы независимо от ноги.
- ◆ Метод поворота пятки независимо от стопы.
- ◆ Метод поворота (подъема) носка независимо от стопы.
- ◆ Метод наклона бедер без необходимости регулирования положения торса.
- ◆ Метод наклона и поворачивания торса без необходимости регулирования положения головы.
- ◆ Правая рука должна легко доставать до левого плеча без сминания плеча (так же и левая рука — до правого плеча)
- ◆ Обе руки должны легко доставать до макушки головы.
- ◆ Метод поворота предплечья.
- ◆ Метод управления направлением локтей независимо от запястий и плеч.
- ◆ Метод, чтобы положить руку на некую поверхность и закрепить ее там.
- ◆ Метод скольжения рукой по поверхности.
- ◆ Метод управления осевой линией движения рук независимо от тела.

Важно помнить, что автоматизирование действий (например, перемещение бедер в зависимости от перемещения ноги) часто приводит к тому, что аниматору приходится бороться с куклой вместо того, чтобы просто придавать ей желаемое положение. Автоматизация полезна только тогда, когда она помогает придавать персонажу определенную позу быстро (например, движок, с помощью которого сгибаются все пальцы и ладонь превращается в кулак).

Автоматизация действий прекрасно выглядит на этапе тестирования, но когда дело доходит до собственно анимации, она начинает мешать. Если я хочу подвинуть одну ногу, это вовсе не значит, что мне нужно, чтобы бедра вдруг сдвинулись, а другая нога стала поворачиваться.

Я встречал много опытных аниматоров, которым приходилось бороться с куклами из-за того, что они были неправильно настроены (и я один из них). Некоторые технические директора почему-то считают, что создать простой и удобный в использовании персонаж означает автоматизировать все, что только можно, чтобы аниматору не нужно было больше ни о чем думать. Это неправильный подход.

Простой и удобный в использовании персонаж легко позируется и не разваливается между позами на части. Тот, кто говорит вам что-либо иное, вероятно, имеет не слишком много опыта в кукольной и 3D-анимации. Компьютер, какими бы замысловатыми ни были программы, не способен создать хорошую анимацию. Анимацию создает аниматор, а компьютер — всего лишь инструмент. Анимация на компьютере ничем не отличается от традиционной рисованной или кукольной. И в том и в другом случае только аниматор решает, как должен двигаться персонаж.

Еще один важный аспект настройки персонажа — это «читаемость» и «понимаемость». Кости и управляющие объекты должны быть хорошо видны и легко выбираемы.

В идеале, аниматор с одного взгляда должен понимать как и для чего построен скелет и как он работает. Я не должен строить догадки или выбирать объект, чтобы видеть его ориентацию. Верите или нет, но этот фактор «читаемости» настолько важен, что может стать определяющим при выборе 3D-программы.

К кукле предъявляется два основных требования. Первое: кукла должна легко позироваться. Второе: структура куклы должна быть проста и понятна.

### Кисти рук

Вместо того, чтобы двигать каждую фалангу, сделайте движок, чтобы управлять сгибанием каждого пальца. Или еще лучше создайте такую систему, которая позволит вам создать «библиотеку» жестов.

### Руки

Самый простой способ анимации рук — это прямая кинематика (FK). Ничего другого для рук и не надо за исключением случаев, когда необходимо «прикрепить» руку к какому-либо объекту. Здесь не обойтись без инверсной кинематики (IK).

### Торс

Для торса используйте FK, работайте вручную, без применения автоматизации.

### Голова

Голова должна поворачиваться независимо от шеи по двум причинам:

- ◆ во-первых, если я хочу, чтоб мой персонаж взглянул на что-нибудь одновременно двигаясь, я не хочу постоянно подправлять положение головы;
- ◆ во-вторых, так гораздо легче выполнять такие принципы традиционной анимации, как перекрывающиеся движения и поочередное сгибание сочленений.

При повороте головы голова либо ведет действие, либо следует за действием. В обоих случаях она движется по дуге отличной от той, по которой движется шея. Движения головы и мигание глаз — это важнейшие элементы выражающие суть характера персонажа и создающие иллюзию живости.

Лучшее решение — привязать ориентацию головы к ориентации основы всего скелета, а не к ориентации шеи.

### Ключицы

Ключицы часто вообще игнорируются, однако попробуйте поднять руки над головой, не используя ключиц и лопаток. Ничего не получится. Поэтому подумайте над тем, как ваш персонаж будет пожимать плечами или хотя бы просто поднимать руки.

### Колени

Обязательно надо предусмотреть возможность поворачивать колени независимо от бедер и стоп. В противном случае с ногами будет очень трудно работать. Если вы не сможете управлять коленями, как вы вообще будете позировать персонажа?

### Стопы

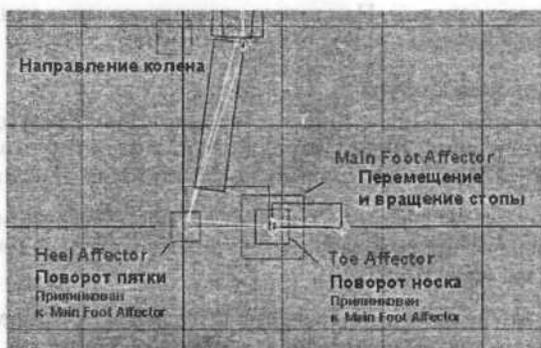
Очень важная часть вашей куклы. Когда вы пользуетесь IK для управления всей нижней половиной персонажа, вам приходится работать главным образом именно со стопами. Существует два подхода в настройке стопы: «от пятки» и «от носка». Я предпочитаю работать от носка по ряду причин:

- ◆ Часто бывает нужно, чтобы персонаж поворачивался на носках. Это трудно сделать работая от пяток.

- ◆ При ходьбе и прыжках персонаж отталкивается носками. Вам может понадобиться, чтобы ваш персонаж прыгал на носках.
- ◆ Попробуйте взобраться по лестнице на пятках. Посмотрим, как высоко вы подниметесь.
- ◆ Вашему персонажу может захотеться встать на цыпочки, чтобы дотянуться до чего-либо
- ◆ Равновесие. Попробуйте сохранить равновесие, стоя на носках, затем на пятках. Что легче?

Независимо от выбранного подхода придумайте метод для поворота персонажа на носках и на пятках. Если вы работаете от пяток, вы все равно должны иметь способ, чтобы ваш персонаж, скажем, мог становиться на цыпочки, а работая от носков, подумайте как сделать, чтобы персонаж мог топнуть пяткой.

Пример настройки ноги:

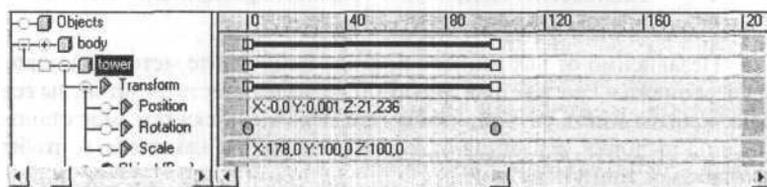


## Стрельба из танка с применением стандартного плагина Combustion

### Шаг 1: Применение Track View

Танк готов, он передвигается. Нам надо, чтобы он стрелял из орудия, и время от времени вертел башню. Определитесь, в каком из кадров он повернет башню и выстрелит. Допустим, у вас 200 кадров, надо чтобы он произвел выстрел в 70-м кадре и вернулся в исходное положение в кадре 130. Поставьте ползунок анимации в кадр 200, в конце. Выделите башню танка и нажмите кнопку **Animate**. Выделив башню, поверните ее в проекции **Top** по оси **X** при помощи кнопки **Select and Rotate**. Если не-

рархическая связь поставлена, то вместе с башней повернется и все остальное, ствол, люки, антенны, пулеметы. Затем отключите кнопку **Animate**. Проиграйте анимацию. При движении танка башня будет поворачиваться, но на протяжении всех 200 кадров, а нам нужно, чтобы она повернулась кадрах как написано выше. Заходим в **Open Track View**. Ищем в иерархическом списке башню танка, к примеру, она у вас названа **tower**. Так как на вершине иерархии танка стоит его корпус (**body**), то башню вы сразу не увидите, щелкните по левому крестику от **body** и раскройте **tower**, это должно выглядеть так:



Черные полосы с квадратиками на концах, это диапазон анимации — туда можно не соваться. Нам нужно отредактировать ключи анимации, которые изображены как жирные, серые точки.

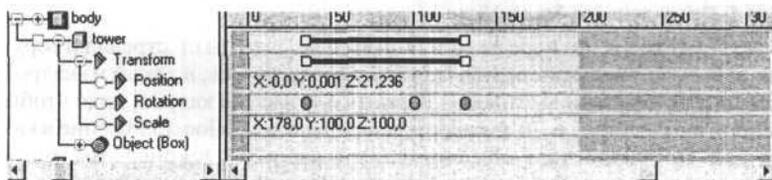
## Шаг 2. Редактирование ключей

Сделаем поворот башни с 70-го по 100 кадр. Для этого выделите первый ключ и внизу меню **Track View**, в окошке, где стоит **0**, измените нулевой кадр на 70-й. Точно так измените последний кадр с 200 на 100.

После того, как вы свернули **Track View** и проиграли анимацию, видно, что башня начинает поворачиваться немного раньше, а точнее — в 40-м кадре, это мы устраним позже.

Нам надо, чтобы она после поворота вернулась в исходное положение. С 70-го кадра она начинает движение, поворачивается в 100-м, заканчивает, а в 130-м делает обратное движение в исходное положение.

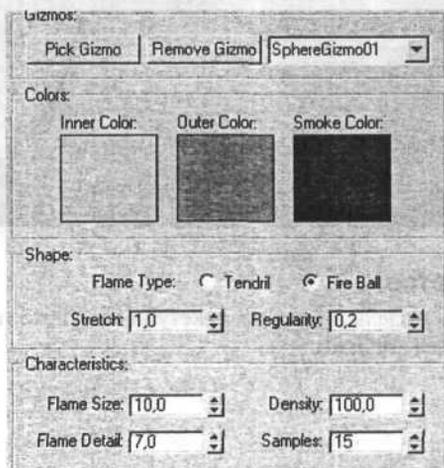
Делаем следующее. Выделяем первый ключ и, удерживая **Shift**, дублируем, передвигая его до 130-го кадра. У вас будет такая картина:



Теперь выровняем по нужным кадрам поворот башни. Выделите все три ключа и нажмите на любом из них правую кнопку мыши. Должно открыться диалоговое окно контроллера управления поворотами. Обнулите значения **Tension/Continuity/Bias** (натяжение, непрерывность, смещение) и проигrajте анимацию. Башня будет двигаться по настроенным выше кадрам. Изменяя положение ключей по кадрам, вы можете поэкспериментировать над поворотами. Главное понять принцип.

### Шаг 3. Выстрел

Для выстрела необходимо создать **Gizmo** выстрела. Заходим в **Create/Helpers/Atmospheric Apparatus** и выбираем **Sphere Gizmo**. Поместите **Gizmo** перед стволом танка и свяжите его с дулом кнопкой **Select and Link**. Размер **Gizmo** можно сделать на ваше усмотрение, смотря какой выхлоп вы хотите получить. Затем заходим **Rendering/Environment**, жмем **Add**, выбираем **Combustion**, **Pick Gizmo** и тыкаем курсором по **Sphere Gizmo**. Далее делаем следующие настройки:



Отсчитаем первый кадр. Он должен быть похож на сферический взрыв. Нам надо, чтобы этот взрыв произошел в момент, когда танк повернул башню примерно в 90-м кадре. Ниже, в настройке **Combustion**, ставим галочки на **Explosion** и **Smoke**, затем в **Setup Explosion** ставим **Start Time 90**, **End Time 100** с интервалом в 10 кадров. Все, теперь выстрел произойдет в соответствии с поворотом башни.

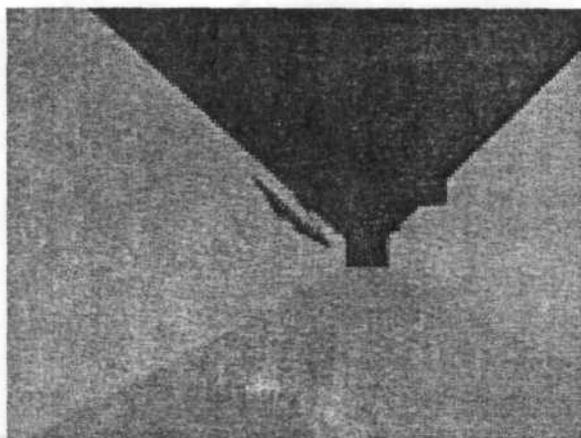
Отсчитайте анимацию и доработайте настройки выстрела по вашему вкусу. Естественно, как написано здесь, выстрел будет слишком

быстрым. Для выстрела из танка подходит интервал протяженностью около 30-ти кадров. Тогда нужно изменять ключи длительности поворота башни. Для придания отдачи башни назад, сдвигайте ее назад в нужном кадре до того, как будете дублировать первый ключ поворота башни. Редактировать ключи отдачи **Track View** можно в параметре **Position**.



### Анимация полетов

Попробуем создать модель полета по макету небольшого городка вслед за неким летательным аппаратом.



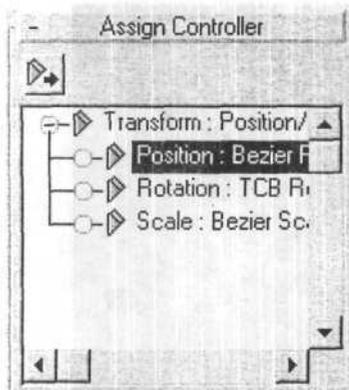
Заставим аппарат следовать по траектории, заданной обыкновенной линией. Камера будет следовать по аналогичной линии.

Итак, в сцене я создал прямоугольник и размножил — это будут домики, нарисовал линией путь для камеры, сделал его копию — путь для аппарата и немного изменил его. Сконструировал из простой сферы некое подобие этого летательного аппарата.

Постройте камеру. Войдите во вкладку **Create** (Создать) в категорию **Cameras** (Камеры) и щелкните кнопку **Free** (Свободная). Щелкните в любом месте окна проекции **Front**. Свободная камера, как и свободный прожектор, строится перпендикулярно окну проекции в направлении вашего взгляда.

Активизируйте окно проекции **Perspective** и щелкните клавишу «С». Вид в окне сменится на вид из камеры. В панели с параметрами камеры в области **Stock Lenses** (Сменные линзы) щелкните кнопку с надписью **15mm** (15-ти миллиметровое фокусное расстояние). Теперь зададим траекторию движения камеры.

При выделенной камере войдите во вкладку **Motion** (Движение). Разверните свиток **Assign Controller** (Назначить контроллер движения), в окне списка назначенных контроллеров выделите строку **Position: Bezier Position** (Позиция: Контроллер управления по Безье) и щелкните кнопку **Assign Controller** (Назначить контроллер).



В появившемся окошке **Assign Position Controller** (Назначить контроллер управления позицией) выделите строку **Path** (Путь) и шелкните **OK**.

Перед вами развернется свиток с параметрами пути.

В области **Curent Path Object** (Текущий объект — путь) шелкните кнопку **Pick Path** (Указать путь) и в окне выбора по имени выберите **Line01**.

В области **Path Options** (Опции пути) установите флажок **Follow** (Следовать) — камера переместится в начало пути.

Началом пути является точка, с которой вы начинали строить линию пути.

Установите флажок **Constant Velocity** (Постоянная скорость) — это заставит камеру сохранять постоянную скорость между двумя точками пути. Иначе на участке пути, находящемся между двумя далеко отстоящими друг от друга точками, камера будет двигаться заметно быстрее, чем на участке пути между близко расположенными точками. Иными словами, камера будет стараться преодолевать участки между двумя точками за один и тот же промежуток времени.

В области **Axis** (Оси) установите переключатель в положение «Y» — по этой оси мы выравниваем камеру вдоль пути.

Воспроизведите анимацию — камера послушно движется по заданному нами пути.

Теперь давайте заставим аппарат двигаться по второму пути.

Выделите аппарат (**Sphere01**) и точно так же, как и для камеры задайте путь следования, только сейчас выберите **Line02**.

Хотя можно было «прицепить» его и к первой линии, я попытался как-то разнообразить анимацию и слегка изменил путь, придав ему некоторые дополнительные искривления.

В области **Path Options** установите флажки **Follow** (Следовать) и **Bank** (Крениться) — это заставит аппарат наклоняться на поворотах пути. Установите флажок **Constant Velocity**. Также выровняйте его по оси **Y**.

Если сейчас посмотреть в окно камеры, то аппарата видно не будет — это потому, что он, так же, как и камера переместился в начальную точку пути (а они у обеих линий совпадают). Нам нужно немного «выдвинуть» его вперед.

В области **Path Parameters** в поле счетчика **%Along Path** (Процентов от начала пути) введите значение **1,5**. Теперь аппарат переместился вперед по отношению к камере и стал виден.

Воспроизведите анимацию. Аппарат не наклоняется во время поворотов из-за того, что задано слишком большое значение **Smoothness** (Сглаженность). Установите в поле счетчика **0,1**. И уменьшите значение **Bank Amount** (Величина крена) до **0,4** — а то иногда на поворотах аппарат переворачивается почти «на спину».

Воспроизведите анимацию.

Теперь заставим камеру не только двигаться, но и постоянно быть обращенной в сторону аппарата.

Для того, чтобы камера могла «следить» за аппаратом, у нее должна быть точка нацеливания. Поэтому удалите свободную камеру и вместо нее постройте Нацеленную камеру (**Target**). Ее лучше создавать в окне **Top**, щелкнув внизу окна и таша мышью до и в сторону аппарата. Установите для нее линзу с фокусным расстоянием **15mm**. Задайте траекторию (**Line01**) аналогично тому, как мы строили траекторию для свободной камеры.

Обратите внимание: теперь появился дополнительный свиток **Look At Parameters** (Параметры управления по линии взгляда). Пока их пропустите и настройте уже знакомые вам параметры со свободной камерой.

Теперь щелкните кнопку **Pick Target** (Указать цель) в свитке **Look At Parameters**. И укажите в качестве цели аппарат (**Sphere01**).

Убедитесь, что выравнивание установлено по оси **Z** и воспроизведите анимацию.

Обратите внимание, что теперь точка нацеливания камеры постоянно обращена в сторону аппарата и он гораздо реже, чем со свободной камерой пропадает из виду, скрываясь за стенами домов.

Но что делать, если нам нужно, чтобы камера двигалась по пути за объектом и при этом могла еще «глазеть» по сторонам? Ответ простой — привязать к аппарату какой-либо объект и точку нацеливания камеры направить на него. Другой вариант — использовать свободную камеру, и в ключевых кадрах вращать ее по нужным осям со включенной кнопкой **Animate**.

Так как второй случай достаточно банален, мы его рассматривать не будем. Изучим первый случай, так как он гораздо более гибок.

Вы можете задать столько вложенных связей объектов, сколько хватит фантазии. При этом можно добиться невероятно замысловатых движений и вращений камеры. Если представить, что к аппарату привязан еще один путь, по которому движется объект, к которому привязан еще один путь, перпендикулярный первому, по которому в свою очередь движется еще один объект, к которому привязана точка нацеливания камеры и так далее...

Увеличьте изображение в окне проекции **Top** в районе летательного аппарата, используя **Ctrl+W**.

Во вкладке **Create** выберите категорию **Helpers** (Вспомогательные объекты) и щелкните кнопку **Dummy** (Пустышка).

Щелкните и тащите мышью около аппарата — построится куб. Расположите его около аппарата, как хотите, перемещая его в окнах проекций.

Теперь привяжем его к аппарату.

На панели инструментов щелкните кнопку **Select and Link** (Выделить и связать), наведите курсор на пустышку и, когда он примет вид связывания, щелкните и тяните мышью к аппарату. Как только курсор опять примет вид связывания, отпустите кнопку мыши. Аппарат мигнет белым, говоря о том, что именно к нему привязана пустышка.

Воспроизведите анимацию. Пустышка послушно следует за леталкой.

Отключите режим связывания, включив какой-либо другой, например **Select and Move**.

Теперь выделите камеру и в панели **Motion** переназначьте линию взгляда камеры с аппарата (**Sphere01**) на пустышку, просто щелкнув

кнопку **Pick Target** в свитке **Look At Parameters** и в окне выбора по имени выбрав **Dummy01**. Теперь, если двигать пустышку, камера будет «следить» за ней.

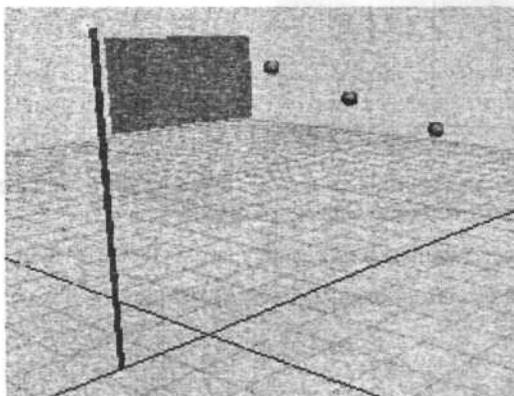
Осталось только в нужных кадрах перемещать пустышку со включенной кнопкой **Animate**. По окончании записи анимации отключите кнопку **Animate**. Воспроизведите полученный результат, и если необходимо — откорректируйте положение пустышки в нужных кадрах при включенной кнопке **Animate**.

### Анимация ткани при помощи Clothreyes

Если вы уже умеете создавать флаг при помощи **Clothreyes**, то можете попробовать при помощи того же самого plugin расстрелять его из какого-нибудь оружия. Для того чтобы это проделать, вам надо будет внести изменения в готовую сцену с флагом, либо создать ее с самого начала.

1. Создаем **hexamesh** (знамя) и цилиндр (древко).
2. Создаем пули, ими могут быть почти любые объекты, я взял три сферы.
3. Создаем два ключа для анимации полета пуль.

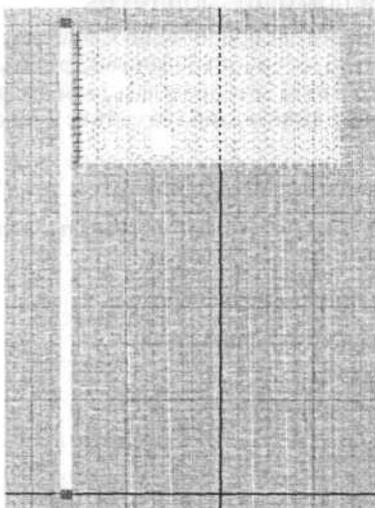
Это примерно то, что должно у вас получиться:



4. Выделяем все объекты в сцене и применяем к ним модификатор **ClothReyes**.
5. Задаем имя для сцены (кнопка **Make Scene**).

6. Назначаем параметры для объектов (**Make Fabric**), на запрос отвечаем положительно только по отношению к **hexamesh**. Основные параметры оставляем по умолчанию, изменяем только взаимодействие с ветром (**AirFriction=30...80**) и придаем свойства для разрушения (флажок **Tearable**). Если хотите, чтобы знамя развивалось на ветру более легко и свободно, измените **mass** и **gravity** в меньшую сторону.

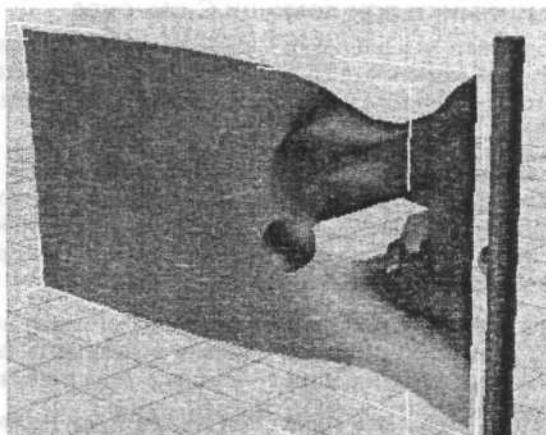
7. Привязываем знамя к древку (иначе оно просто упадет вниз). Переходим в режим редактирования вершин, выделяем вершины на цилиндре и левой стороне **hexamesh**.



8. Задаем имя для группы (кнопка **Make Group**) и прикрепляем знамя к древку (кнопка **Fixed to Object** ⇨ **Cilindr01**).

9. Устанавливаем направление и силу ветра (**Wind x = 500**).

10. Запускаем расчет динамики (**Start Calculation**).

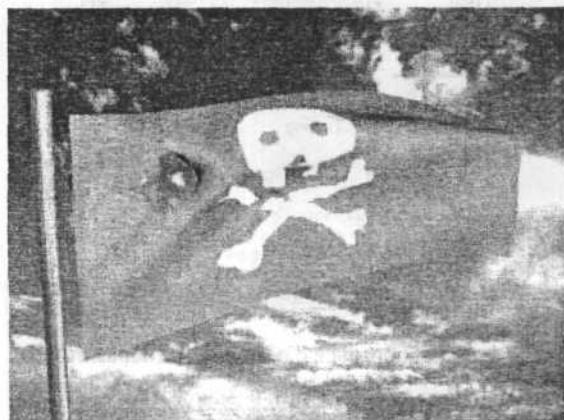


11. Если вы все сделали правильно, то через какое-то время сможете увидеть, как пуля разрывает ткань.

Когда будете работать с текстурой для знамени, обратите внимание на порядок модификаторов в стеке.

Для того, чтобы наложенная вами текстура могла корректно выглядеть, необходимо разместить **UVW mapping** модификатор ниже **Clothreys**.

Примерно так может выглядеть конечный результат:



## Расчет динамики при помощи Clothreys

Многие пользователи используют Clothreys только для анимация ткани, а между тем этот модуль выполняет расчет динамики практически для любых объектов, и работать с ним намного удобнее, чем со стандартным модулем MAX.

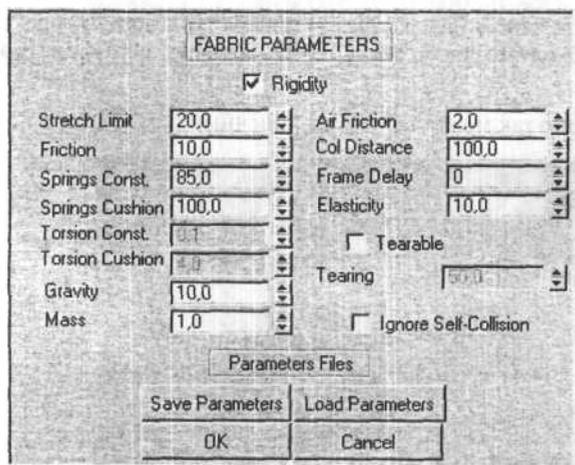
Для того, чтобы вы могли легко управлять динамикой в любой сцене, необходимо усвоить главное правило: все объекты, к которым применен модификатор **Clothreys** взаимодействуют между собой, но ключевые кадры будут просчитаны только для тех объектов, которым вы назначаете свойство **Fabric** после нажатия кнопки **Make Fabric**.

При назначении этого свойства появляется диалоговое окно **FABRIC PARAMETERS**, которое позволяет задавать основные параметры для ткани, а тканью в вашей сцене может быть почти любой объект.

Если вы хотите, чтобы ваш объект был твердым, вам необходимо установить флажок **Rigidity**.

Установка флажка **Tearable** заставит ткань рваться при сильном взаимодействии с другими объектами.

**Ignore Self-Collision** отключает коллизии для объекта.



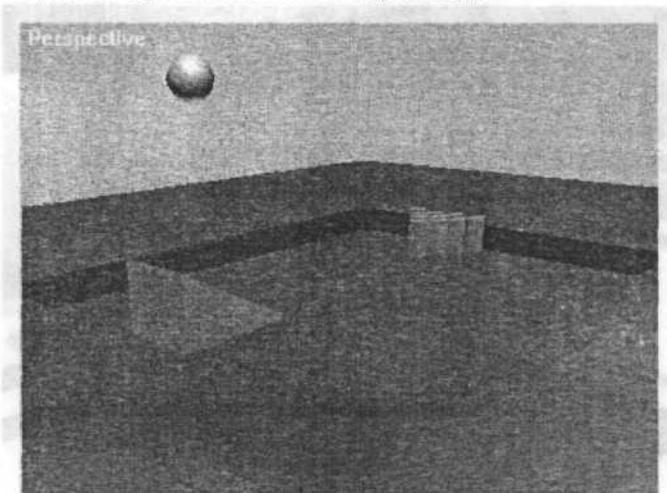
Теперь попробуем разобраться с основными параметрами.

- ◆ **Friction** — сила трения (мин. 0, макс. 100);
- ◆ **Gravity** — гравитация, или ускорение свободного падения;

- ◆ **Mass** — масса;
- ◆ **Air Friction** — чувствительность ткани к взаимодействию (с ветром например);
- ◆ **Elasticity** — упругость для твердых тел (мин. 0, макс. 100).

Ну вот теперь можно смело браться за дело. Прежде вам надо будет решить, что и с чем вы собираетесь сталкивать.

Я для эксперимента взял вот такую сцену:



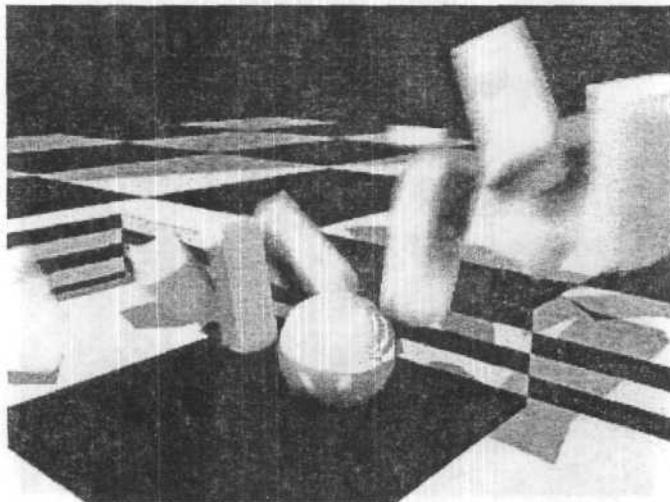
Расставляя объекты, будьте внимательней, нельзя, чтобы поверхности пересекали друг друга! Если ваша сцена уже готова, то можно приступать к главному — назначению модификатора **Clothreyes** и установки свойств для объектов.

1. Выделить все объекты, которые хоть как-то должны взаимодействовать между собой.
2. Выбрать модификатор **Clothreyes (Modifi ⇨ More...)**.
3. Задать имя для сцены (**Make Scene**).
4. Вспоминаем основное правило и назначаем свойства для объектов (**Make Fabric**).

В моей сцене свойство **Fabric** назначено всем подвижным объектам (сфера и кегли, в качестве которых я использую обычный примитив **Box**). Важно также правильно установить параметры для всех подвижных

объектов (для удобства воспользуйтесь кнопками **Load** и **Save Parameters**). Поскольку в моей сцене не используется ткань, я установил для всех объектов флажок **Rigidity**, увеличил силу трения (**Friction**) до 10, параметр **Mass** сделал для сферы побольше (5,0), остальное оставил по умолчанию. 5. Если с параметрами вы закончили, то можно нажимать на кнопку **Start Calculation**, после чего вам останется только ждать когда закончится расчет всех коллизий.

Динамика в действии:

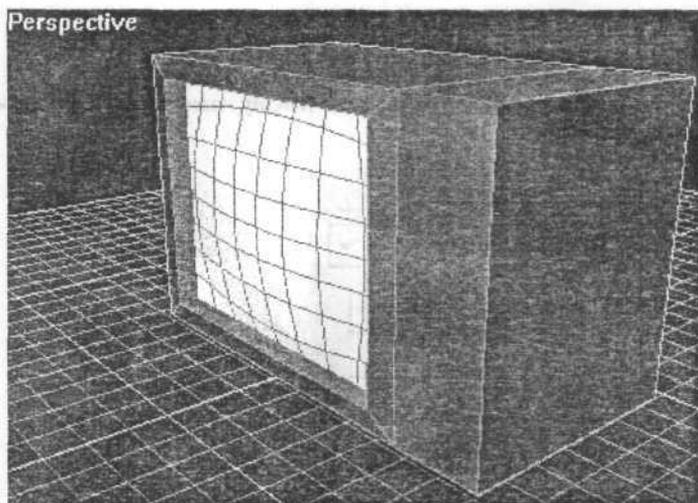


Применений для **Clothreyes** может быть довольно много, часто этот модуль используется для анимации одежды, иногда можно эмулировать анимацию волос (при условии, что анимируется не каждый волос, а целая прядь). Интересные результаты дает анимация цепей.

### Анимированная текстура и ореол кинескопа для телевизора

Здесь мы расскажем, как использовать AVI-файл в качестве текстуры для экрана телевизора, а также как сделать светящийся ореол работающего кинескопа.

Для начала создайте модель вашего телевизора.

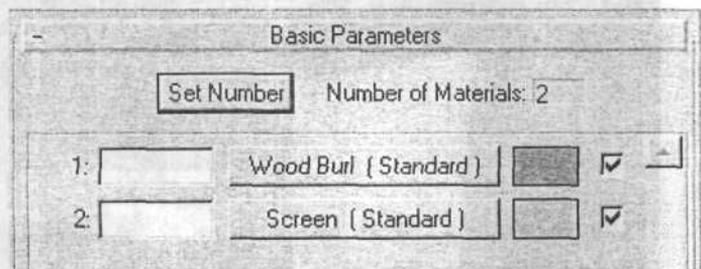


Откройте **Material Editor**.

Выберите первую сферу и нажмите **Type: Standard**.

Выберите **Multi/Sub-Object** из списка.

Установите **Number of Materials = 2**.



Первым материалом у нас будет текстура телевизора, а вторым — экран.

#### Создание текстуры для экрана

Нажмите на материал 2.

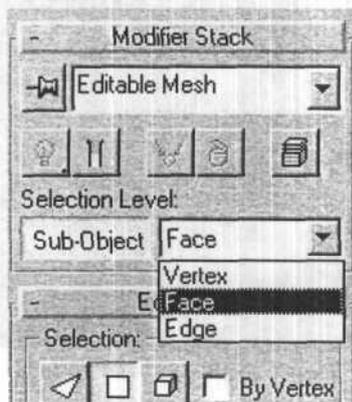
В разделе **Maps** нажмите на кнопку **None** рядом с **Diffuse**.

Выберите **Bitmap** из списка.

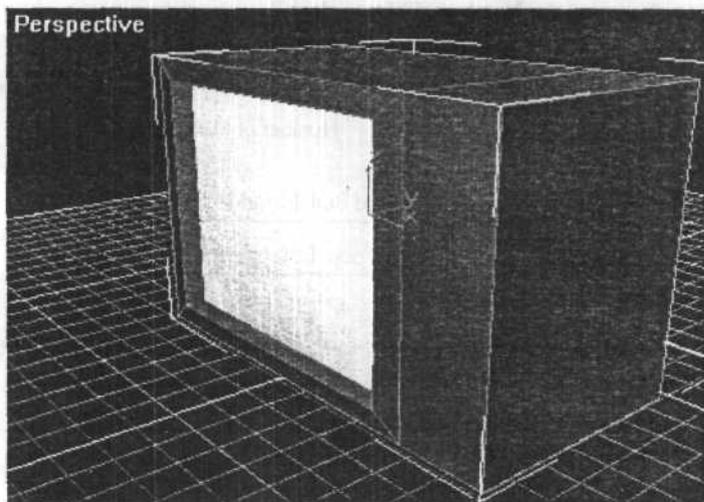
Выберите ваш файл AVI.

Закройте **Material Editor**.

Выберите телевизор. В вкладке **Modify** нажмите кнопку **Sub-Object** и выберите **Face**.

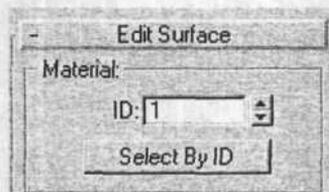


Выберите все грани, которые должны быть корпусом телевизора, а не экраном. Чтобы выбрать несколько граней, нажмите **CTRL**.

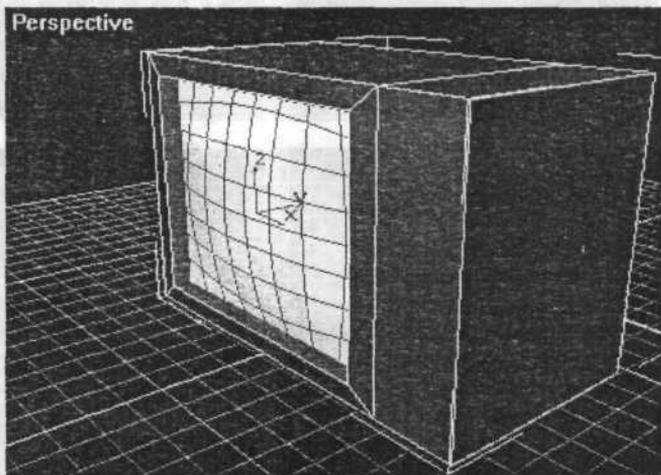


Как только вы выберете все нужные грани, то прокрутите свиток до **Edit Surface**.

Измените **Material: ID** на 1.



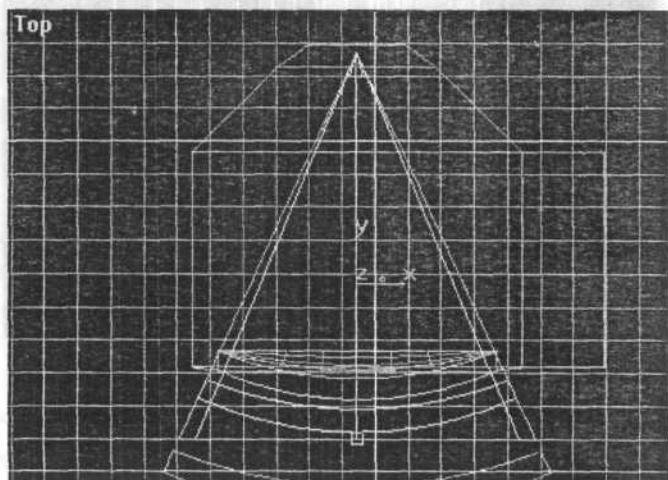
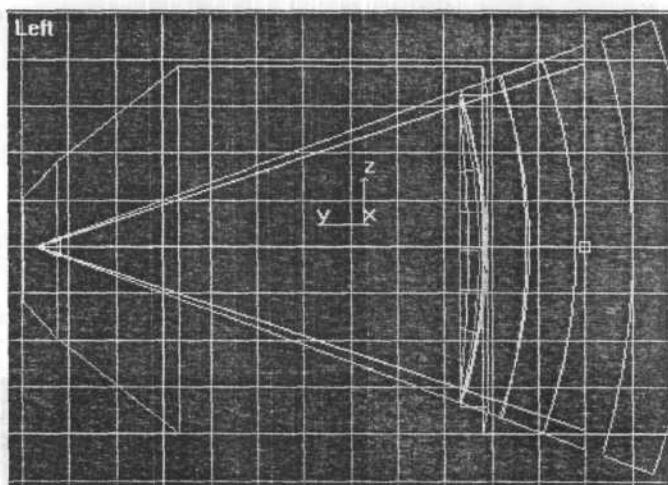
Теперь выделите все грани экрана.



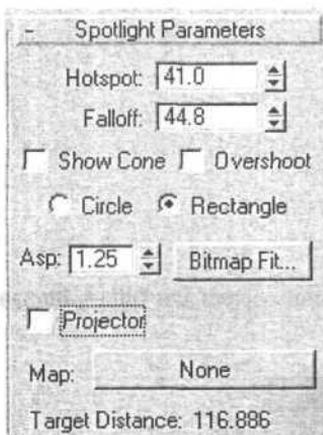
На сей раз измените **Material: ID**: на 2.

Это может быть и все, что нужно для нашего телевизора, но мы добавим некоторые световые эффекты, как у настоящего телевизора — ореол и мерцание.

Создайте источник света **Target Spot** и установите его перпендикулярно экрану телевизора.

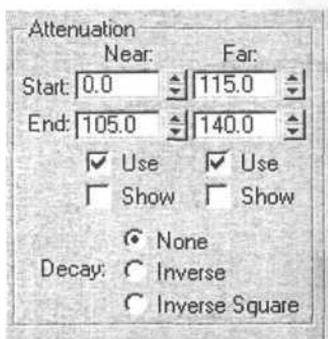


Если ваш экран не имеет формы правильного квадрата, то вы должны будете откорректировать **aspect ratio (ASP)** в параметрах источника света. Для моего экрана я установил значение (приблизительно) **1.25**.



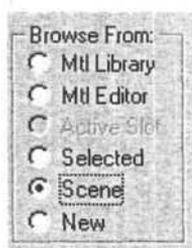
В панели **Modify** (с выбранным источником света) активизируйте переключатели **Use** в **Near** и **Far** в разделе **Attenuation**.

Откорректируйте **End:Near** так, чтобы синий параллелепипед совпадал с размером экрана. **Start:Far** должен быть установлен так, чтобы желтый параллелепипед был расположен перед синим и **End:Far** (коричневый параллелепипед) должен быть немного удален перед желтым.

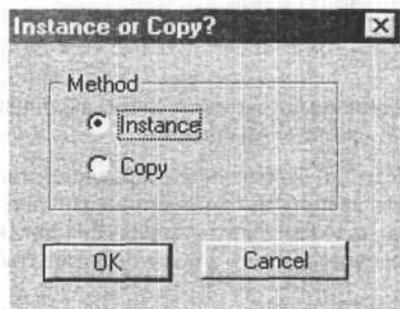


Теперь в **Spotlight Parameters** включите флажок **Projector Map**:

Нажмите «радио» кнопку **Scene**.



Выберите AVI, который вы использовали для **Diffuse Bitmap**.  
Нажмите **Instance** и **OK**.

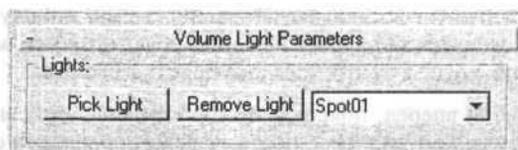


В меню **Rendering** выберите **Environment**.



В разделе **Atmosphere** нажмите **Add** и выберите **Volume Light**.

В **Volume Light Parameters** нажмите кнопку **Pick Light** и выберите ваш источник света.



Теперь откорректируйте **Density** источника света или другие параметры по своему усмотрению.

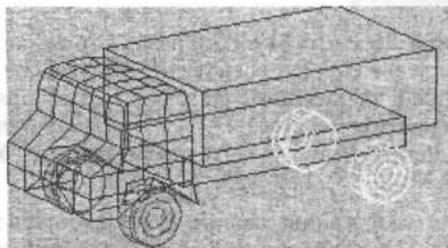
### Движение автомобиля по трассе с использованием Plug-In Roller

Перед тем, как приступить к созданию реалистичной модели движения автомобиля на поворотах, необходимо уточнить, сколько центров вращения задействовано в момент поворота.

В нормальной машине с одним рулем и четырьмя колесами их всего три — один в центре заднего моста (там, где дифференциал) и еще по одному в центре каждого переднего колеса. Впрочем, даже если колес будет шесть, количество центров останется таким же, главная конструктивная особенность — по центру в каждом рулевом колесе и еще центр между всех остальных.

Теперь разберемся, сколько в машине вообще движущихся частей (естественно не имеется в виду работа двигателя, движение руля или ковыряние в носу самого водителя). Два передних колеса — два объекта, они хоть и крутятся в одну сторону, но каждое вращается вокруг своей собственной оси. Два задних — это один объект (хотя конечно в настоящих машинах на повороте одно колесо движется чуть быстрее другого, мы не будем вдаваться в такие тонкости, все равно заметно не будет). И, наконец, сам кузов — это тоже один объект.

Сгруппируйте командой **Group** все части вашего автомобиля так, чтобы у вас получилось четыре объекта: кузов, два передних и оба задних колеса.



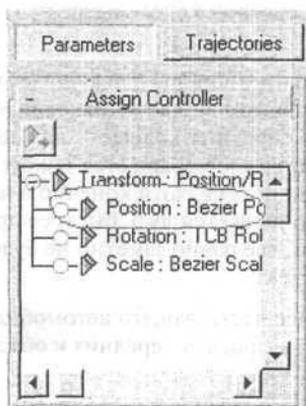
В командной панели во вкладке **Hierarchy**, нажав кнопку **Affect Pivot Only**, выставьте центр для кузова и задних колес в центре заднего моста, а для передних колес в центре каждого переднего колеса.

Немного забегаая вперед, поясню, что движением кузова и колес будут управлять разные контроллеры, поэтому между объектами совсем не обязательно выставлять какие-либо связи, но для прорисовки траектории движения передних колес понадобится временно связать их с кузовом.

При помощи кнопки **Select and Link** присоедините каждое переднее колесо к кузову.

Теперь надо сделать саму дорогу. Поверхностью послужит любой плоский объект, например, **Box**. Траекторию движения можно нарисовать при помощи какого-нибудь инструмента в разделе **Shapes**, например, **Line** или **Point Curve**.

Выделите кузов, в командной панели выберите вкладку **Motion**, нажмите кнопку **Parameters** (если она еще не нажата) и откройте свиток **Assign Controller**.

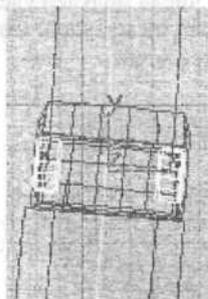


В открывшемся свитке выделите **Position** и назначьте ему контроллер **Path**. В появившемся свитке **Path Parameters** выделите **Follow** (следование) и, нажав кнопку **Pick Path**, щелкните по вашей траектории. В результате вся машина должна переместиться в начало пути. Если машина встала криво или боком, попробуйте поменять параметры **Axis**. У меня все расположилось по оси **Y**.

Теперь, если включить анимацию, машина будет двигаться задней осью прямо по траектории, только передние колеса не поворачиваются.

Дело в том, что у передних колес есть своя собственная траектория, отличная от задней.

Выделите одно из передних колес, в верхней части вкладки **Motion** нажмите кнопку **Trajectories**, и у вас появится траектория движения этого колеса. В свитке **Trajectories** установите параметр **Samples** равным параметру **End Time** и нажмите кнопку **Convert To**. Теперь у вас есть кривая, соответствующая траектории движения одного из передних колес, вторую траекторию сделайте тем же способом, выделив другое колесо.



Снова выделите кузов и в свитке **Path Parameters** выберите **Bank**, измените параметры **Bank Amount** (величина заноса): **-0.1** и **Smoothness** (мягкость рессор): я поставил **0,01**.

Присвоим вращение колесам, используя контроллер **Roll Along Path**, который находится там же где контроллер **Path**. В разделе **Rolling Object Objects** выставьте радиус колес вашей машины и назначьте им траекторию движения кнопкой **Pick Path**.

## Моделирование гусеницы танка с последующей анимацией

Создание гусеницы – непростое дело.

### Шаг 1: Создание модуля гусеницы

Включите кнопки **3d Snap Toggle** и **Use Selection Center**, последняя должна стоять в меню постоянно, иначе модель может не получиться. Зайдите в меню **Create** ⇨ **Standard Primitives**. Нажмите кнопку **Box**. Создайте **box** с параметрами: **length (90,0)**, **width (30,0)**, **height (5)**. **Box** готов. Конечно, можно смоделировать навороченный модуль гусеницы, с шарнирами, шипами, текстурой, чтобы все выглядело как в жизни. Главное – понять принцип.

Не забывайте сохранять ваш файл после каждого шага.

### Шаг 2: Клонирование

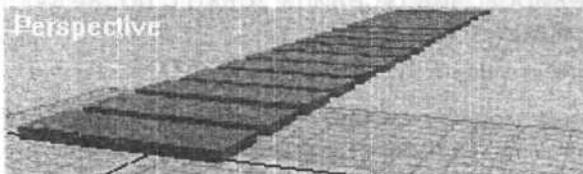
Перейдите в окно проекции **Top** и выделите готовый **box**. Перетащите этот **box** при помощи **Select and Move** вправо или влево примерно на одну клетку в сетке (**grid**), удерживая кнопку **Shift** на клавиатуре. Отпустите кнопку мыши и **Shift** на клавиатуре. У вас должен клонироваться второй **box**. Повторите так пять раз пока у вас не будет пять объектов. Выделите эти пять объектов, и повторите клонирование, соблюдая одинаковую дистанцию между модулями (боксами).

Затем шансы на клонирование большего числа модулей увеличатся. Клонировать дальше, пока не получится около 60 модулей. Создайте группу из модулей, которая и будет будущей гусеницей:



### Шаг 3: Присоединение (Attach)

Выделите первый **box** из всей группы. Зайдите в меню **Modify** и нажмите кнопку **Edit Mesh**. В этом меню отключите активную желтую кнопку **Sub-Object**. Отключив ее, появится кнопка **Attach** (присоединение). Нажмите **Attach** и наведите курсором на соседний от первого модуль, появится крестик. Нажмите кнопку мыши и присоедините второй модуль к первому. Прделайте это с остальными модулями, пока у вас не получится один целый объект. Назовите его «track», гусеницей:



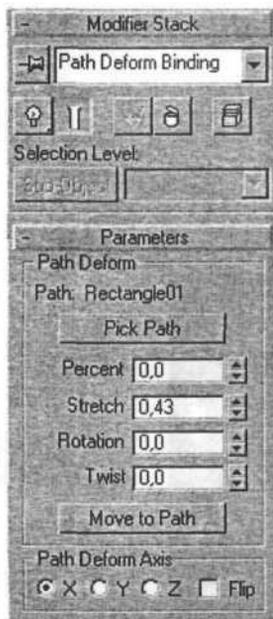
### Шаг 4: Создание профиля гусеницы при помощи сплайна

Снова переходим в меню **Create** ⇒ **Shapes** ⇒ **Splines**. Нажимаем кнопку **Rectangle** (прямоугольник). Создайте **Rectangle** с параметрами: **length (85,0)**, **width (390,0)**, **Corner Radius (0)**. Длина прямоугольника должна получиться примерно в шесть раз меньше, чем длина гусеницы.

### Шаг 5: Закручивание гусеницы по периметру прямоугольника с применением Path Deform Binding

Для начала нужно выровнять гусеницу с прямоугольником. Выделите прямоугольник, нажмите кнопку **Align**, наведите курсор на прямоугольник и нажмите кнопку мыши. Появится окно **Align Selection (Rectangle)**. В самом верху поставьте галочки на всех осях X, Y, Z.

Все, гусеница и прямоугольник выровнены. Выделите гусеницу и зайдите в меню **Modify** ⇨ **More...** (в случае если модификатор не установлен). В списке **Modifiers** установите **Path Deform**, не перепутайте с **Patch Deform**. В меню **Path Deform Binding** нажмите кнопку **Pick Path** и, наведя курсор, щелкните левой кнопкой мыши на прямоугольнике. Гусеница примет причудливую форму, но нам надо, чтобы она располагалась по всему периметру прямоугольника. Сделайте такие настройки:

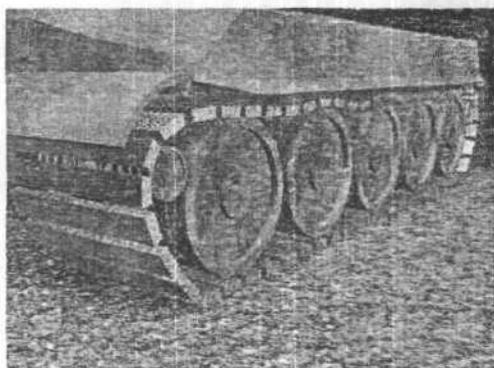
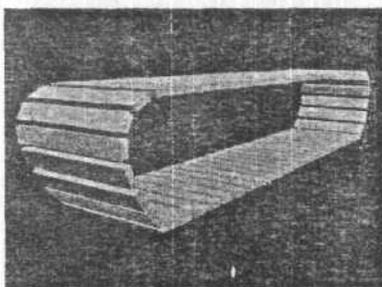


### Шаг 6: Редактируем профиль гусеницы

Получилось подобие конвейера. Выделите прямоугольник, в меню **Modify** нажмите кнопку **Edit Spline**. Отредактируйте профиль гусеницы примерно так, как должно быть у танков, тракторов:



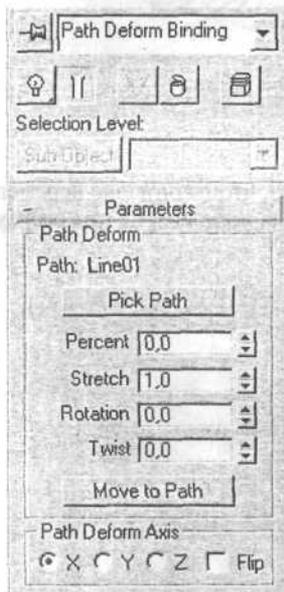
По-моему, результат эффектный, тем более эту гусеницу можно редактировать и естественно анимировать! Назначите текстуру по вкусу. Назначайте текстуру на первый модуль сразу, пока не клонируете остальные. Вставьте колеса, присоедините к соответствующему корпусу танка или трактора и наслаждайтесь. Далее мы сделаем анимацию гусеницы, и движение танка в примере с простым корпусом.



#### Шаг 7: Настройка движения траков

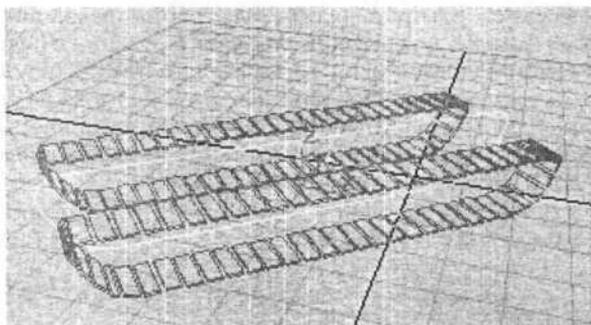
У нас должен сохраниться сплайн, кривая профиля, по которой расположились траки. Не торопитесь с клонированием гусениц. Выделите готовую гусеницу. В свитке **Path Deform Binding** есть параметр **Percent** (процент). При прокрутке ползунком вверх или вниз, гусеницы

должны пошевелиться. В этом и состоит суть работы траков. Включаем большую кнопку **Animate**, и переводим ползунок анимации в положение кадра 100. Параметр **Percent** ставим на **100**. Отключим кнопку **Animate**. Проверим итог, нажав кнопку **Play animation**, гусеница должна работать самостоятельно. Если она крутится не в нужном направлении, измените **Percent** на **-100**. Отключите **Animate**, и переведите ползунок анимации в положение кадра **0**, то есть, в начало.



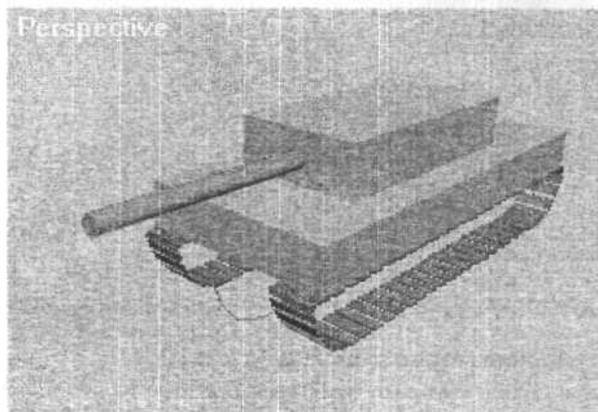
### Шаг 8: Клонирование второй гусеницы

Просто создайте копию второй гусеницы так, как это должно быть у танка, или возьмите расстояние на ваш вкус.

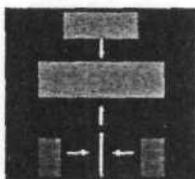


### Шаг 9: Движение в пространстве

Гусеницы движутся, но нам нужно, чтобы они еще перемещались по горизонтальной плоскости. Для начала положим на них сверху нечто похожее на модель корпуса танка.



Начнем связывать все части танка в одно целое. Основной связки будет корпус танка. Нажмите кнопку **Select and Link**, наведите на ствол и щелкните по нему мышкой, не отпуская, подведите к башне. Когда серый кубик станет белым, отпустите кнопку. Теперь ствол привязан к башне. Проверить это можно, потаскав башню в разных направлениях, ствол будет перемещаться вместе с башней. То же самое сделайте с остальными частями танка, учитывая связь с корпусом танка. Придерживайтесь схемы, приведенной на картинке снизу. Башня связывается с корпусом, гусеницы со сплайном, сплайн с корпусом. При перетаскивании одного корпуса все части должны перемещаться вместе с ним.



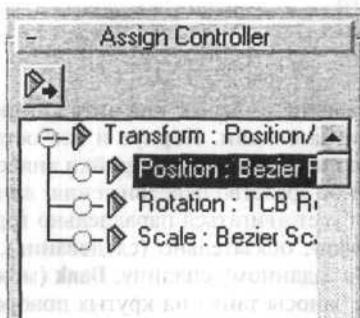
### Шаг 10: Настройка движения гусениц в пространстве

Нарисуйте в проекции **Top** кривую (spline), будущую траекторию движения танка. Постарайтесь нарисовать ее, так что бы вершины находились на равном расстоянии.

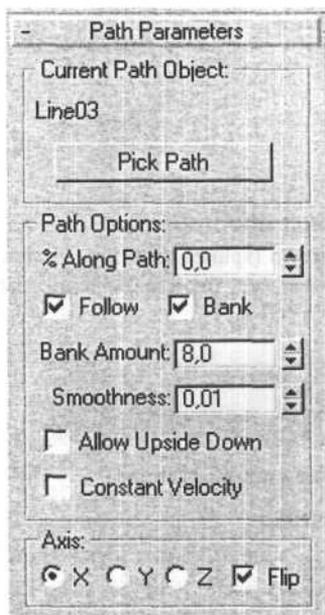


Выделите корпус танка и зайдите в меню **Motion**.

Там будет настройка **Assign Controller**.



Выделяете **Position**: нажимаете зеленую кнопку вверх, в появившемся окне выбираете **Path** и жмите **OK**. Затем, опускаемся ниже и видим настройки **Path Parameters**. Теперь самое важное. Перед тем, как настроить эти параметры нужно нажать кнопку **Animate** и перевести ползунок кадров в конец на 100-й кадр. После этого перейдем к настройкам.



Выделите корпус, если он еще не выделен, нажмите кнопку **Pick Path**, и наведите крестик курсора на сплайн. Корпус и все остальное должно встать на путь вашего сплайна. Сделайте настройки аналогично рисунку. Поскольку танк – наземное средство передвижения, внизу поставьте координату X, тогда танк будет двигаться параллельно горизонтальной плоскости. Поставьте **Follow**, обязательно (следование), тогда ваш танк будет точно двигаться по заданному сплайну. **Bank** (можно не ставить), эта функция отвечает за заносы танка на крутых поворотах, а если поставите, сделайте **Bank Amount** на 8. Если при установке **Follow** танк встал задом, поставьте галочку в окошке **Flip**. Отключите **Animate**. Нажмите **Play animation**, и посмотрите, что получилось. Если все готово, можно отсчитывать анимацию.

Траекторию сплайна можно изменять на ваше усмотрение, самое главное, чтобы расстояние между вершинами сплайна по возможности было одинаковым, иначе танк будет двигаться неравномерно, а рывками. Скорость передвижения танка зависит от количества кадров анимации. Если хотите сделать медленное движение, перед началом всех настроек поставьте, больше кадров от 200 и выше...

## Создание огня с использованием частиц

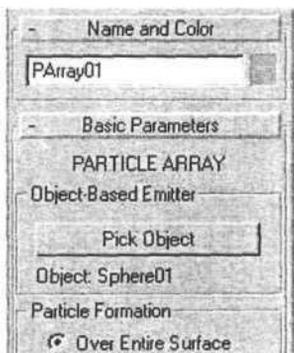
Преимущества: при использовании этой процедуры вы сможете поджигать все что угодно, включая анимированные фигуры людей, деревья. Огонь связывается с движением объекта и его можно видоизменять.



### Часть 1: настройка сцены

Сейчас вы должны решить, что именно собираетесь поджечь. Мы будем рассматривать просто сферу, но то же самое можно сделать с любым объектом, в том числе с телом, модифицированным физикой.

1. Создайте сферу с радиусом **10-20**.
2. Создайте массив частиц и назначьте сферу в качестве источника.



### Часть 2: подбор параметров частиц

1. Для генерации частиц выберите **use rate** и назначьте величину **60-100**.
2. В пункте **motion** установите скорость в **0**.
3. В пункте **timing** величины назначаются в зависимости от ваших потребностей и длины ролика.
4. Установите **Emit stop** равным длине ролика (**100**).
5. Установите **Life** на уровне **20**.
6. Размеры: **size:4, grow for:20, fade for:50**.

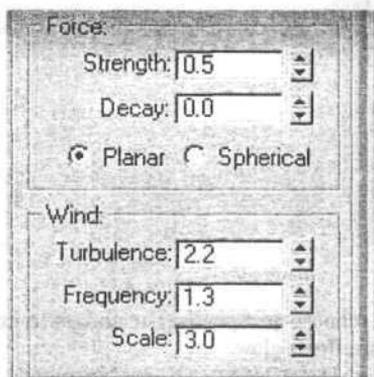
<input checked="" type="radio"/> Use Rate	<input type="radio"/> Use Total
80	100
Particle Motion	
Speed:	0.0
Variation:	0.0 %
Divergence:	10.0 deg
Particle Timing	
Emit Start:	0
Emit Stop:	100
Display Until:	100
Life:	20
Variation:	0
Subframe Sampling:	
<input checked="" type="checkbox"/>	Creation Time
<input checked="" type="checkbox"/>	Emitter Translation
<input type="checkbox"/>	Emitter Rotation
Particle Size	
Size:	4.0
Variation:	0.0 %
Grow For:	20
Fade For:	50

### Часть 3: тип частиц

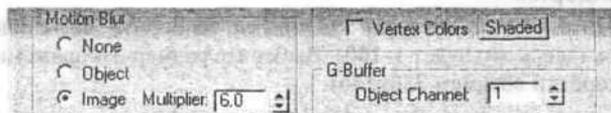
Поскольку это соответствует нашим задачам, для более быстрого рендеринга выберите **constant**.

Теперь добавьте ветер в виде пространственной волны, и посмотрите, как он подействует на частицы. Создайте волну снизу вверх, в направлении сферы и массива частиц.

1. Привяжите массив частиц к ветру и установите для него следующие параметры: **strength:0.5 turbulence:2.2 frequency:1.3 scale:3**.



2. Нажмите правую кнопку мыши на **Parry** ⇨ **parameters** ⇨ **image motion blur** и установите **Image Multiplier** равным **6**, удостоверившись при этом, что он выбран. Установите канал объекта G-буфера равным **1**.



### Часть 4: текстура и материал

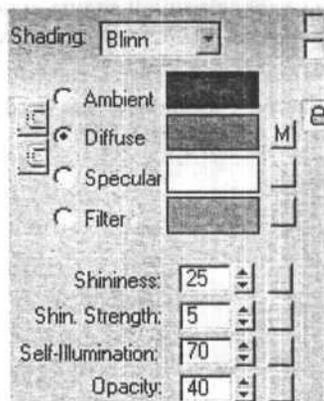
Установим мэппинг по времени и создадим для огня новый материал:

1. Откройте редактор материалов и создайте материал типа **blinn** со следующими установками: **self illumination:55, Opacity:90**.

2. Выберите карту градиента для диффузного типа: **цвет#1**: красный; **цвет #2**: оранжевый; **цвет#3**: светло-голубой. Назначьте позицию второго цвета равной **0.08**.

3. В пункте расширенных параметров материала выберите **falloff** → **out :100**.

Примените созданный материал к массиву частиц.



### Часть 5: Video Post

Выберите **add scene event (choose perspective)** и добавьте событие **image filter**. Назначьте фильтр **lens effects glow**.

1. Выберите следующие установки: во вкладке **preference** назначьте размер **5** и интенсивность **50**. Нажмите **ОК** и вернитесь в **video post**, определите имя выходного файла и сохраните ролик как **avi** или **flc**.

### Часть 6: рендеринг

Нажмите **execute** (бегущий человек). Выберите последовательность кадров (здесь это будет **1-100**). Выберите размер изображения (для большей скорости возьмем **320x240**).

На этом работа завершена. Поскольку массив частиц может иметь в качестве источника любой объект, описанная техника позволяет легко поджигать объекты любой сложности.

## Глава 31.

### Создаем стекло

Многие 3D дизайнеры в своих работах используют предметы из стекла, но добиться по настоящему реалистично выглядящего стекла удается не многим.

Это связано, прежде всего, с тем, что изделия из стекла настолько часто попадают на глаза, что не составляет большого труда определить компьютерную подделку.

Не торопитесь запускать программу, а сходите на кухню и принесите стакан, а еще лучше предварительно вымойте его с мылом, ведь чтобы запачкать виртуальный мир нужно очень много сил и ресурсов, этим мы заниматься не будем.

Когда вспоминают свойства стекла, первое что приходит на ум — это прозрачность и преломление. К сожалению многие этим и ограничиваются. Для того, чтобы реализовать эти простые свойства, в 3ds max предусмотрено несколько способов. Вот они:

- ◆ Карта **Refract/Reflect**;
- ◆ Карта рефракции в тонкой пластине;
- ◆ Карта **Raytrace**;
- ◆ Материал **Raytrace**.

Первые две карты мы отмечаем сразу, так как в этих случаях качество принесено в жертву скорости. Карта **Raytrace** и материал **Raytrace** по многим параметрам схожи, в частности, для расчета отражения и преломления они моделируют реальное поведение световых лучей, и изображения полученное с их помощью (при умелой настройке) выглядит фантастически реально.

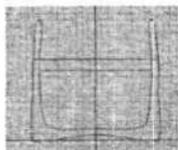
Мы будем использовать материал **Raytrace** так как в отличие от карты у него есть несколько дополнительных параметров без которых нам не удастся добиться реализма. Посмотрите внимательно на характер изменения прозрачности, вы заметили, что к краям она спадает? Это потому, что лучам света приходится проходить сквозь большое количество стекла.

Теперь рассмотрим, как распределяется отражение по поверхности стакана, если вы внимательны, то должны заметить усиление отражения от центра к краям. Если не догадываетесь, почему так происходит, то обратитесь к учебнику физики. Следует также упомянуть о таком чисто компьютерном параметре как количество переотражений/преломлений, в реальной жизни они стремятся к бесконечности, но так как скорость компьютеров не бесконечна, то все программы ограничивают этот параметр очень не большим числом, в 3ds max например по умолчанию стоит 9.

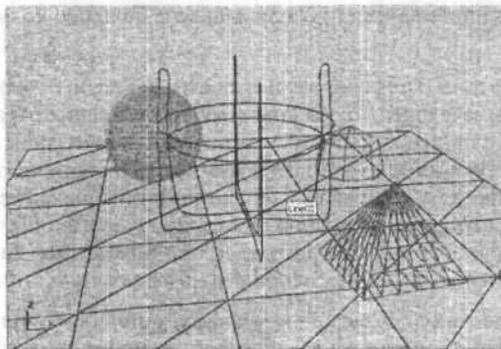
Для моей сцены вполне хватило и 5, а вот установив меньшее значение, вы значительно проиграете в реализме, так как именно пять переотражений позволяют нам увидеть, что отражается на задней стенке ста-

кана. Реально это выражается в виде цветowych пятен на поверхности стекла, которые и придают ей «живой вид». Из основных параметров хотелось бы также упомянуть о бликах, они у стекла жесткие и обусловлены свойствами материала, яркие, что сравните для примера блики на металлической поверхности и резинового мячике.

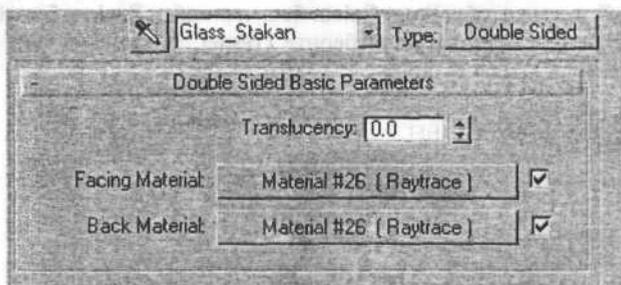
Итак, приступим. Сначала нам нужно создать стакан, проще всего это сделать так, нарисуйте профиль стакана, только учтите у него должно быть две стороны и примените к нему модификатор **Lathe**, только не скупитесь на количество фейсов. Я в своей сцене использовал **NURBS** так как он дает более качественный результат.



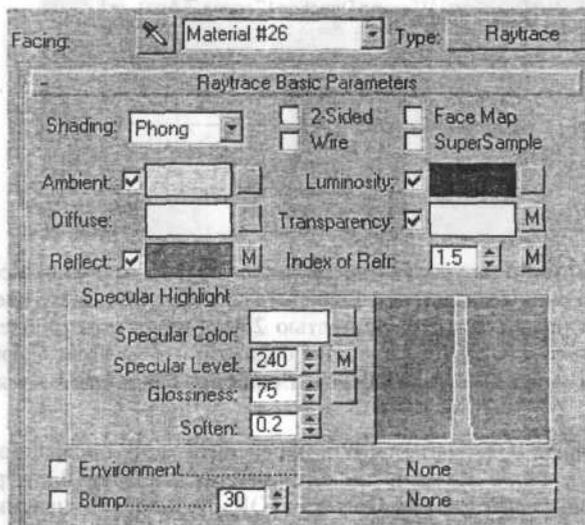
Вот у вас и получился простой стакан, можете поработать над ним еще немного добавляя детали, а затем нужно создать ему окружающую среду. Создайте небольшой **BOX** и поместите его под стакан, пусть это будет стол. Также установите на стол еще несколько простых примитивов, у меня это: сфера, пирамида и непонятная загогулина для общего разнообразия. Разместите эти объекты спереди и сзади стакана, что бы было видно, как он умеет отражать и преломлять. Еще следует создать один большой **BOX** такого размера, чтобы в него поместилась все ваша сцена, это будет комната, примените к нему модификатор **NORMAL** и установите галочку **FLIP**. Осталось настроить камеру и расставить свет, я думаю трех точечной схемы освещения будет достаточно.



Пора приступить к материалам, для начала назначьте всем вспомогательным объектам несложные текстуры. А теперь займемся нашим главным героем. Создадим **Double Sided** материал, на роль **facing** материала назначим **Raytrace** материал, и перенесем его на **Back** выбрав при этом способ копирования как образец.



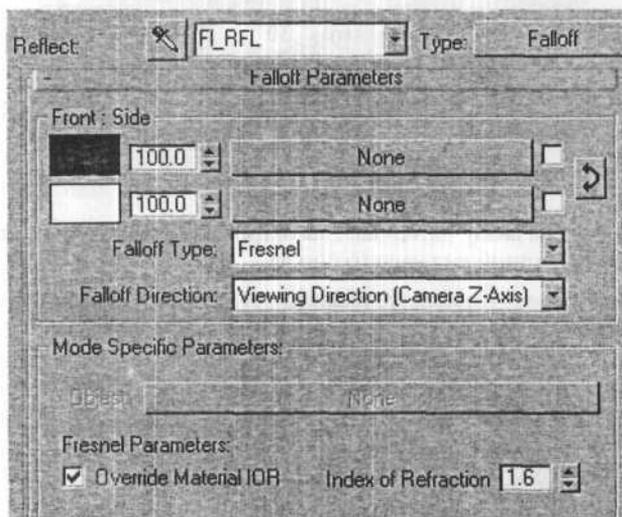
Щелкнув на **facing** материале, вы окажетесь в настройках, которые и определяют внешний вид вашего стекла.



Начнем сверху вниз. В качестве алгоритма тонирования выберите **Fong**, он дает более жесткие блики. Справа поставьте галочку **Super Sampling**, она помогает избежать визуальных артефактов, но, так как эта опция сильно замедляет расчет сцены, то на этапе тестирования ее мож-

но отключить. В качестве цвета диффузии выберите серый с интенсивностью 127.

В строчке **Reflect** выберите небольшое значение интенсивности. 45 вполне хватит, это будет минимальный уровень отражения. Щелкните на пустующей кнопке справа, чтобы назначить на канал отражения карту **Falloff**, важно чтобы **Front Colour** был черный, а **Back** — белый, в качестве алгоритма смешивания выберите **Fresnel**. Это и обеспечит усиление отражения к краям стакана.



Поднимитесь на уровень вверх и скопируйте только что созданную карту на канал **Transparency**. Выберите в качестве базового цвета для прозрачности белый, с интенсивностью 250. Теперь подстройте карту **Falloff** для канала прозрачности таким образом, чтобы **Front Colour** был белым, а **Back** — черным, это обеспечит спад прозрачности к краям стакана.

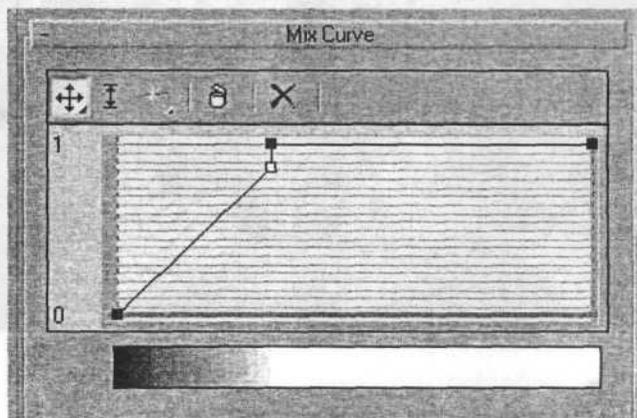
Установите **IOR** равным 1.5, и перенесите карту прозрачности на карту **IOR**, указав при этом способ копирования как **Copy**. В настройках этого канала поставьте галочку **Overdrive IOR** и установите значение равным 1.6 или чуть больше, это обеспечит более сильное преломление света на краях стакана.

Теперь настроим блики. Выставите значения параметров **Shininess** и **Shininess Strength** как 80 и 180 соответственно. Раскройте свиток **Extended Parameters** и найдите параметр **Translucency**, установите

значение цвета, **Value** не более **40**. Этот параметр отвечает за рассеивание света в стекле и позволяет реализовать такие материалы как: замерзшее стекло, рисовая бумага или свечной воск. Далее наш путь лежит в **Raytracer Control**, щелкните на **Global Parameters** и установите количество переотражений равным **5**, впрочем, если у вас очень мощный компьютер то можете оставить **9**.

Поставьте галочку в разделе **Global Ray AntiAliaser** и нажмите **Close**. Позже в этом разделе поиграйте с параметрами размытия. Раскройте свиток **Maps**, и в цифровых полях на против каналов отражения и преломления введите значение **50**, это позволит ослабить влияние карты **Falloff**.

Теперь нужно только поставить сцену на просчет. Для более тонкой настройки вам следует подобрать параметры карты **Falloff**, в частности **Mix Curve**, а также степень ее влияния на отражения и преломления.



Эти параметры зависят от толщины стекла вашего стакана. Нужно добиться такого эффекта, чтобы стенки стакана были слегка темными. Для ускорения процесса настройки я рекомендую назначить карту **Falloff** не на канал прозрачности, а в качестве диффузного цвета стандартного материала — это позволит значительно ускорить процесс настройки **Mix Curve** и затем скопируйте карту в канал отражения или преломления. Впрочем, предварительных расчетов для тонкой настройки не избежать, поэтому на это время отключите **AntiAliaser**, что значительно ускорит процесс. Не переусердствуйте с параметром **Translucency**, так как он слишком размывает изображение и для обычного стекла его значение не должно превышать **95**.

Если в результате рендеринга на изображении видны черные точки размером с один пиксель, то либо уменьшите параметр размытия в настройках **AntiAliaser**, либо измените фильтр с **Fast** на **MultiResolution**, и в настройках аккуратно увеличьте начальное и максимальное количество лучей. Будьте аккуратны с этими настройками, так как они катастрофически влияют на время расчета. Если у вас в запасе нет свободной вечности, то закрасьте в ручную эти пиксели в программе Adobe PhotoShop или аналогичной, впрочем, если вы настолько безумны, что создаете с **Raytrace** эффектами анимацию, то этот путь не для вас.

Итак, создание реалистичного стекла не такая уж трудная задача.



## Глава 32.

### Тонкости и хитрости создания металлических поверхностей

Оглянитесь вокруг. Наверняка вы увидите множество предметов, где используется металл. Вспомните таблицу Менделеева, добрая ее половина посвящена различным металлам, да и «простая» золотая цепь на шее смотрится на много эффектнее, чем разноцветная пластмассовая безделушка. В общем, роль металла в создании реалистичных CGI-образов трудно переоценить. Хотя компьютеры еще не научились выдавать результат в слитках благородного цвета или бумажках с металлизированной полоской, тем не менее, овладев секретами создания «настоящего» металла, вы сможете просить у шефа повышения зарплаты.

Давайте внимательно посмотрим на металлы. Прежде всего, обратим внимание на то, что их очень много и все они разные, секрет успешного моделирования и заключен в том, насколько хорошо вы понимаете различия характеристик металлов.

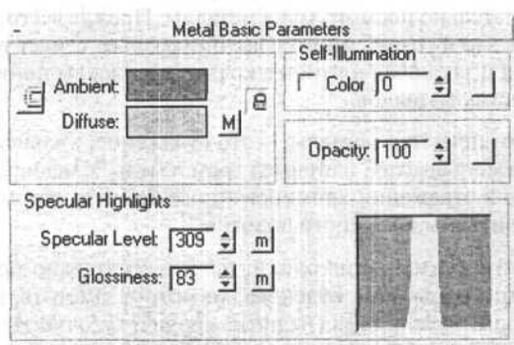
Первое, чем отличаются металлы, — это отражение, а зависит оно от твердости самого металла, так, например хром и медь. У меди отражение очень мутное, а в отражении хромированного бампера автомобиля можно заметить все изъяны дорожного полотна.

Теперь обратим внимание на блики. Если внимательно посмотреть на шашку имени головы тов. Буденова, то можно заметить, постарайтесь не акцентировать свой взгляд на пятна крови и кусочки засохшего мяса, что блики очень яркие и четкие, в отличие от бликов на медных ручках у вашей мебели, где их практически не видно, так как они очень тусклые и размытые. Это явление, как и в случае с отражением, является следствием более «мягкой» поверхности. Вполне логично, что сейчас мы рассмотрим, как реализовано это свойство материала в 3D графике.

Многие уже, наверное, догадались, что я имею ввиду карты **Bump** и **Displeis**, отличаются они влиянием на геометрию объектов, к которым применяются материалы с этими картами. **Displeis** изменяет геометрию и используется в основном для внесения крупных изменений, что очень удобно. Недостаток у этого метода только один — цена создания мелких деталей. Для их хорошего качества придется добавить пару сотен тысяч фэйсов.

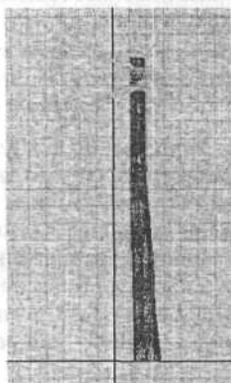
**Bump** напротив геометрию не изменяет, это плюс, но в результате не возможно создать сильное смещение поверхности. Особенно заметно это при взгляде на края объекта и его тень, так как здесь влияние бампа обнаружить не удастся. Следовательно, **Bump** предназначен для мелких деталей, что в нашем случае и необходимо. Наиболее грандиозных результатов удается добиться, комбинируя эти два способа. Классическим примером этого является кирпичная стена, когда **Bump** используется для нанесения мелких дефектов, царапин и трещин, а **Displeis** — для получения общего рельефа стены.

Для выбора алгоритма тонирования есть несколько вариантов. Прежде всего, это «Метал», доставшийся нам еще с DOS'овских времен. Одним из достоинств этого алгоритма является задать такие параметры **Glossiness** и **Specular Level**, что яркость блика будет или постоянна на всем его протяжении, или даже спадать к центру блика.



Давайте попробуем изучить этот материал.

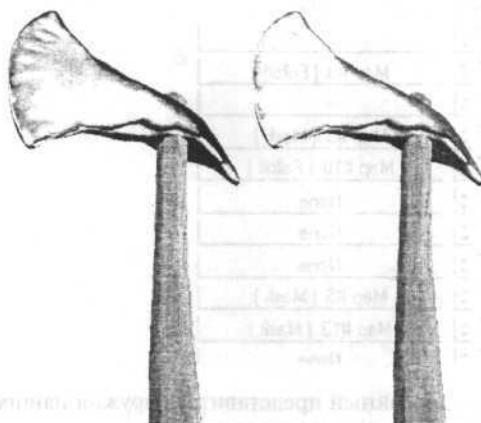
Для начала создадим тестовую сцену. Я использовал модель топора, взятую из коллекции бесплатных моделей на одном из web-сайтов, добавил в модели немного граней, разместил вокруг несколько источников света и наложил на **Background** карту **Meadow1.jpg**, она идет в стандартном наборе текстур поставляемых в комплекте с 3ds max'ом.



Теперь перейдем в редактор материалов. Сделайте «простую деревяшку» и назначьте этот материал на ручку топора, а вот с железякой придется повозиться.

Установите цвет диффузии в серый. Задайте какое-нибудь неприлично большое значение параметру **Specular Level**, к примеру **400**, а **Glossiness** — **80**, это обеспечивает нам очень яркие и точные блики.

Получилось не плохо, но недостаточно реалистично:



Теперь добавим отражение. Раскройте свиток **Maps** и добавьте карту **RayTrace** в канал **Reflection**. Двигаясь дальше, добавим в слот **Diffuse color** карту **Falloff**, это обеспечит более темные края, у нашего боевого топора. Скопируйте ее в канал **Glossiness**, одновременно уменьшив влияние этих карт до 80-50 процентов.

Настала пора состарить немного наш топор. Наложим на канал бамп-карту **Noise**, очень маленького размера, от **0.01** до **2**. И затем, чтобы придать большего реализма, скройте часть нойза маской. Нужно нарисовать эту маску с учетом того, где в реальности на объекте могут возникнуть вмятины и царапины, но сейчас сойдет и что-то типа карты **Smoke** с большим размером зерна.

Чтобы улучшить внешний вид металла, скопируйте карту бамп в слот **Specular Level** и задайте ей отрицательное значение. Теперь неровность поверхности будет оказывать значительное влияние на силу бликов, и в завершении нужно будет замаскировать канал отражения так же, как это было сделано с картой бамп.

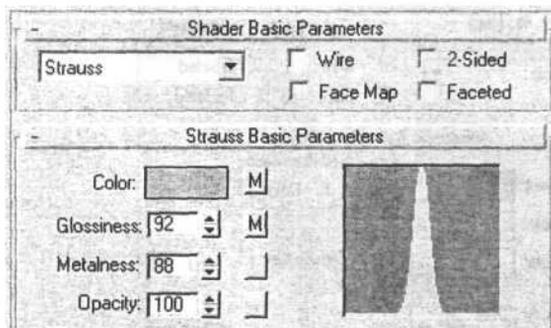
	Amount	Map
<input type="checkbox"/> Ambient Color.....	100	None
<input checked="" type="checkbox"/> Diffuse Color.....	86	Map #9 (Falloff)
<input type="checkbox"/> Specular Color.....	100	None
<input checked="" type="checkbox"/> Specular Level.....	-116	Map #5 (Mask)
<input checked="" type="checkbox"/> Glossiness.....	48	Map #10 (Falloff)
<input type="checkbox"/> Self-Illumination.....	100	None
<input type="checkbox"/> Opacity.....	100	None
<input type="checkbox"/> Filter Color.....	100	None
<input checked="" type="checkbox"/> Bump.....	160	Map #5 (Mask)
<input checked="" type="checkbox"/> Reflection.....	100	Map #13 (Mask)
<input type="checkbox"/> Refraction.....	100	None

Согласитесь, получился достойный представитель оружия наших предков, пролежавший в земле энное количество лет.



Теперь рассмотрим с следующий алгоритм тонирования, **STRAUSS**, он не напрягает большим количеством параметров, и подходит в тех случаях, когда нужно быстро получить неплохой результат.

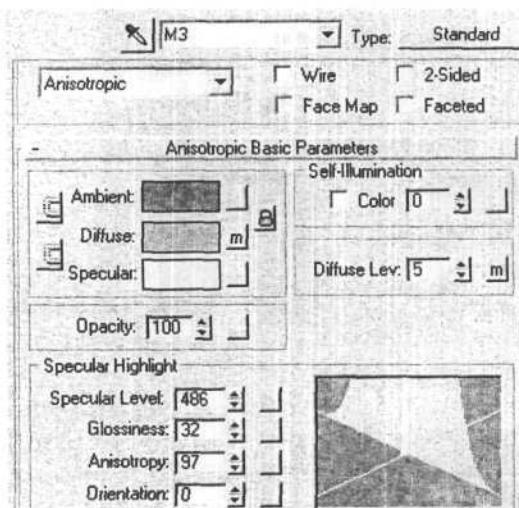
Основным органом управления является параметр **Metalness**: чем он больше, тем более “металлистая” поверхность получается.



Все пакеты 3D графики делятся на HI-END и все остальное. Критерием, по которому происходит отбор, является наличие тех или иных фишек.

Одной из таких фишек является тонирование по алгоритму «Anisotropic».

Два основных параметра, от которых зависит качество металла, это **Dif Level** и **anisotropy**. Значение первого невелико, так как у металлов очень низкий уровень собственного диффузного цвета, а уровень второго чуть больше 50 и зависит от вашего желания. Он определяет вытянутость блика по одной оси, угол можно изменить и даже подвергнуть анимации.



Посмотрите, какие шикарные блики дает этот материал, потратьте немного лишнего времени на изучение его параметров и вы будете вознаграждены.



## Глава 33. Создание стеклянных и металлических поверхностей

Нарисуем обычные песочные часы.

Делается это так: рисуем два цилиндра радиусом примерно 100 единиц и высотой в 10.

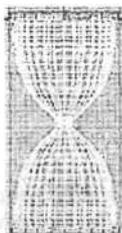


Располагаем их на расстоянии в 300 единиц по высоте, затем рисуем трубу (tube) радиусом в 80 единиц и высотой в 300 единиц, располагаем ее как показано на рисунке:



Теперь займемся деформацией трубы, применяем к ней деформатор **FDD** (box).

Теперь изменим расположение control points деформатора, сожмем их.



После этого переходите в **Material Editor**. Займемся созданием материалов для наших часов. Нам нужно стекло и металл. Для их создания можно воспользоваться:

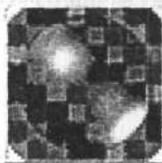
- ◆ входящим в состав 3ds max материалом **RayTrace**;
- ◆ сторонними материалами, входящими в состав рендеров-рейтрейсеров (**RayMax**, **RayGun**).

Начнем с примера использования **RayTrace**.

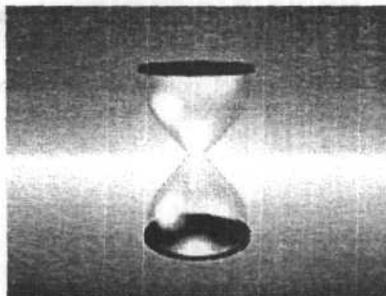
Указываете **Shading — Metall**, немного высветляете раздел **Reflect**, указываете **Shinnes** и **Shinnes Strength 70** и **100** соответственно. Это наш металл.

Теперь займемся стеклом. Оно будет практически таким же как металл за исключением трех пунктов.

Оставьте все как в металле, только добавьте пункт **2-Sided**, измените **Index of Refraction** на **1,3** (коэффициент преломления стекла). Вот ваш материал, наше стекло:



Теперь назначьте их на соответствующие части часов, наложите какой-нибудь рисунок на фон и вперед, рендерим.



Теперь разберемся с оставшимися материалами и подключением внешнего рендера. Для начала рассмотрим **RayMax**. Вы можете либо сконвертировать уже имеющиеся материалы в материалы **RayMax**, либо

создать их заново. Если вы решили идти по второму пути, то вот основные настройки, которые вам нужно будет сделать.

Настройки **RayMax** материала для металла и стекла (разница в том, что в металле мы указываем значение **Reflection=30** и **Refraction=0**, а в стекле наоборот).

Для подключения внешнего рендера вам нужно зайти в раздел **File-Preferences** и там в закладке **Render** найти пункт **Assign** и выбрать необходимый вам рендерер (при этом он должен быть подключен — находится в каталоге **Plugins**).

В панели настройки рендера обратите внимание на пункты **Anti-Aliasing** и **Filter Maps**, они очень помогают в улучшении качества картинки. Кроме того, большие значения в пунктах **Reflection** и **Refraction TL** тоже увеличивают качество и очень сильно замедляют рендеринг.

Перейдем к **RayGun**. Настройки рендера и процедура подключения практически такие же. Единственная особенность рендера — он значительно быстрее предыдущего, хотя и картинки у него не отличаются высоким качеством.

Имеется возможность указать тип материала (**Glass, Metall**), **IOR** (коэффициент преломления), а в остальном настройки похожи на материал **RayMax**.

Результатом рендеринга данной сцены являются стеклянные песочные часы. Хочется заметить, что указаны лишь основные настройки материалов, при использовании различных текстурных карт и дополнительных настроек вы можете получить более сложные эффекты (не забудьте только, что прибавление эффектов сильно сказывается на скорости рендеринга).

## Глава 34.

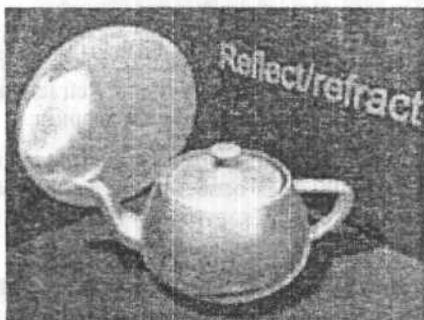
### Три способа создания зеркал

Зеркальные поверхности — один из особых предметов гордости Autodesk и Kinetix — ведь именно их 3ds max смог быть не только трехмерным модельером, но еще и крутой рендеринговой системой и рендерить такие вещи, как **blur**, отражения, прозрачность. До этого, конечно, были системы типа **POVR**, но они, в основном, — рендереры, то есть, моделировать в них особенно нельзя.

Мы рассмотрим три основных способа создания зеркал, по возрастающей сложности.

### 1 способ — Reflect/Refract

Создадим сферу и сплющим ее так, чтобы она была похожа по форме на линзу. Теперь идем в **Material Editor** и создаем новый материал. Можете оставить все основные установки так, как они есть — вполне подойдет почти любой шейдинг, блеск. Наша цель — раздел **Maps** и пункт **Reflect** в нем. Нажмите на него, и из появившегося списка **New** выберите **Reflect ⇔ Refract Map**. Можете не менять никаких установок, вернуться в основной материал и присвоить этот материал вашему объекту — сплющенной сфере. Все — рендерите.



Быстрый и простой метод. Но это единственное его достоинство. Дальше идут одни недостатки. Во-первых — это очень плохое качество изображения. Даже на той сжатой из 640x480 до 200x150 картинке вы можете заметить ребристость границы — поверьте, она выглядит действительно отвратительно. Во-вторых, качество изображения (количество пикселей) уменьшается с отхождением объекта от сферической формы. Для идеальной сферы этот способ очень привлекателен, но его ни в коем случае нельзя использовать на плоских объектах, и не рекомендуется для несферических объектов.

### 2 способ — Flat Mirror

Этот способ применим только для идеально ровных поверхностей. Если у вас хотя бы на одну миллионную координата сдвинута, он уже не годится. К тому же, способ не так прост, как первый.

Итак, приступим. Создайте **Box**. Материал нам понадобится не простой, а многокомпонентный, а точнее двухкомпонентный. Создайте новый материал **Multi ⇔ Sub-Object**. Сделайте **Set Number** и укажите два компонента. Нам надо создать два простых материала: один — собственно зеркало, а другой — то, что будет как бы обрамлять его. Зеркало будет вторым материалом, а обрамление — первым — это важно.

Тыкаем мышкой во второй материал, если нужно — регулируем основные параметры, но самое главное — создаем **Flat Mirror Map** в разделе **Reflect Map** (так же, как и **Reflect/Refract**). Берете свой **Box**, и добавляете модификатор **Edit Mesh** (можно также сконвертировать **Box** в **Editable Mesh**). Теперь переключаете **Sub-Object** на **Face** и выделяете поверхность.

Теперь ищите **Material ID** и вводите туда **2**. Теперь мы готовы к рендерингу.



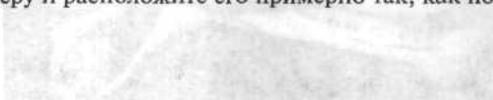
Мы добились хорошего качества изображения, пожалуй, почти идеального. Но только для идеально плоских поверхностей. Метод необходим там, где нужны точные, качественные и самое главное быстро читающиеся зеркала. По сравнению с рейтрейсингом, этот метод дает очень большой выигрыш в быстродействии.

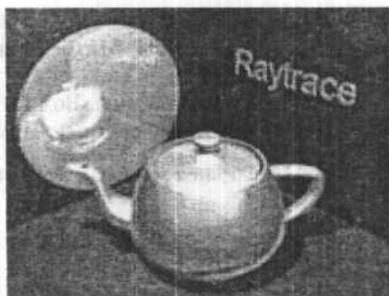
### 3 способ — Raytrace

**Raytrace** — это встроенный рейтрейсер 3ds max, доступный еще во второй версии.

Рейтрейсинг, или трассировка лучей, — это способ построения изображения, при котором каждая точка поверхности, выбираемая с каким-либо шагом дискретизации испускает «луч», который просчитывается по обычным физическим законам (преломления, отражения). Так получается идеальная картинка. Минусы такого подхода тоже очевидны — очень низкая скорость по сравнению со всем остальным.

Продемонстрируем его на не самом традиционном сферическом зеркале. Сделайте сферу и расположите его примерно так, как показано на картинке.





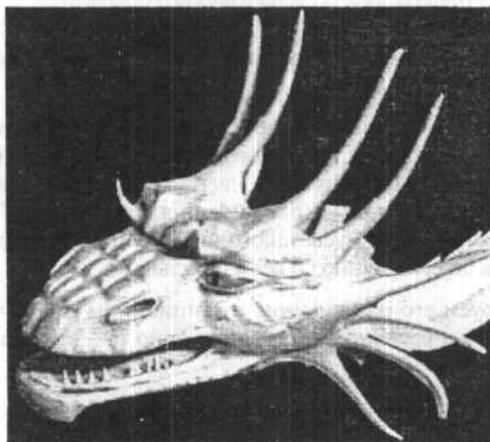
Создайте новый материал и в **Reflection Map** задаем новый **map** — **Raytrace** (процесс аналогичен заданию **Reflect/Refract** из первого способа). Присвойте материал объекту и рендерите.

Идеальное качество изображений. Очень низкая скорость просчета — самая низкая из всех методов. Проблема в том, что встроенный рендерер 3ds max'a очень медленный по сравнению со своими аналогами.

## Глава 35.

### Советы по моделированию зверей

#### Эффективное моделирование персонажей



Рассмотрим два способа: — первый — нечто вроде **BOX MODELING**, но с вариациями. Второй — моделирование с применением **NURBS** поверхностей.

Первый способ неплох но у него есть свои недостатки. Во-первых — трудно бывает добиться соответствия эскизу. Хотя мне удавалось чуть ли не портретное сходство. Во-вторых — получается слишком много полигонов. Хотя и с этим можно бороться.

Попробуем для начала создать дракончика. Начнем с головы. Первое, что надо сделать — это нарисовать его гордый профиль в виде замкнутого сплайна (в сплайне все vertex-сы должны быть **linear**!!).

Затем рисуем несколько таких же «изолиний». Всем этим сплайнам делаем **Extrude** на какую-нибудь заметную высоту. После этого лучше перейти к работе непосредственно с **mesh**-ем. Я обычно просто делаю **collapse all** в стеке модификаторов. Располагаем наши объекты так, чтобы получилось грубое подобие полуголовой дракона, и при этом они перекрывали бы друг друга. Затем двигаем вершины в сетке (**mesh**) так, чтобы подобие стало по возможности менее грубым. Затем для всех объектов по очереди делаем булевскую операцию **union**. Получив **editable mesh**, проверяем ее (советую для удобства отключить все **smoothing groups**). После булевских операций часто обнаруживается множество близкорасположенных паразитных вершин. Их лучше удалить. Для этого можно или, установив **snap vertex**, собрать все соседние вершины к одной и сделать **vertex weld**, или просто выбрать их все и сделать **vertex collapse**.

Надо только следить, чтобы при этом не образовалось перекрывающихся **face**-ов. Попробуем применить к нашему объекту **meshsmooth**. Гладко, но скучно.

Идем дальше. Делаем место под глаз (веки). Техника та же. Рисуем сплайны, редактируем сетку, объединяем булем, редактируем. Так же присоединяем ноздри и рога. Углубляем пасть тоже булем, но **subtraction**. Зубы можно делать таким же способом. А можно немного по-другому, как и чешуйки-пупырышки.

Мелкие чешуйки-пупырышки можно делать при помощи материалов — **bump, specular**. Крупные лучше сделать сеткой. Можно для них использовать **boolean**. А можно применить **compound object shapemerge**. Для этого берем вид нашего зверюги в профиль и рисуем четырехугольники там, где нам хотелось бы видеть чешуйки (конечно, слишком увлекаться не стоит; последствия могут быть ужасными!). Затем все эти квадратики объединяем в один сплайн. После чего делаем нашей полуголове **shapemerge** с этим сплайном. Превращаем полученный объект в сетку. Заходим в подобъект **face**. Те **faces**, которые нам нужны, уже выделены.

Делаем **extrude faces** на минимальную величину (для этого вручную надо ввести значение — например 0.001). Теперь просто двигаем их все вместе так, чтобы наши пупырышки приобрели некую высоту. Затем скалируем на уменьшение, предварительно выбрав в верхней панели скалирование по трем координатам с отдельными центрами. Можно всю эту эпопею с пупырышками повторить сверху, снизу, спереди. Желательно при этом время от времени подчищать лишние **vertexes-faces**, как уже описано.

**Важное замечание:** Конечно, нужно при этом всем использовать и описанные во всех книжках способы — просто **extrude face**, вообще все возможности **edit mesh**. Кроме того, я очень часто пользуюсь всеми разновидностями модификаторов **FFD** в применении к выделенным вершинам.

Борьба с глюками буля. Эти глюки у вас рано или поздно возникнут. Сперва попробуйте, поставив значение **treshold** в **weld vertex** поменьше, сделать **weld** всем **vertex**-ам. Если не помогло, попробуйте сделать **utilites reset X-form** и в стеке — **collapse all**. Если это не помогает — попробуйте сделать **modifiers** — **cap holes**. Тогда остается ловить глюк по частям.

Теперь выберем все **faces** нашей полуголовы, которые образуют осевую плоскость (разделяющую половины головы), и убьем их! Выделим все вершины, ограничивающие образовавшуюся дыру, и отскалируем их в 0 по одной понятно какой, оси. Затем делаем **mirror-copy**, объединяем две сетки, совместив их с применением **snap vertex**, и делаем **weld**.

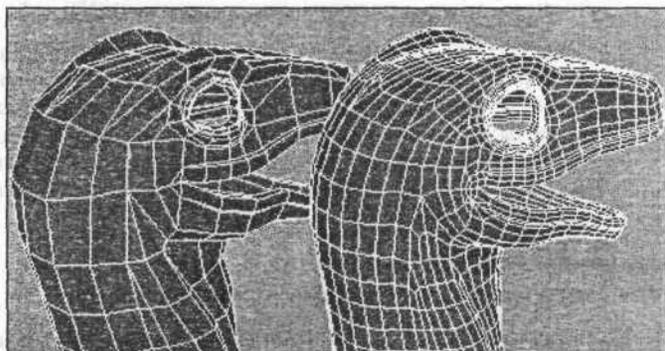


Дальше делаем **Meshsmooth**. Можно и старый — тогда выставим **Quad Output** и **Smooth result**. По-моему, вполне терпимо.

А можно новый. Тогда выбираем NURMS и в подобъекте Vertex выставляем для выбранных вершин Vertex weight. Так можно и зубки поострить, и щечки подгладить.

### Подробное описание MESH и создание головы динозавра

Ниже я объясню технику Mesh моделей в своем понимании. Я начну с самых основ, так что это может быть полезно даже новичку. В то же время, этих сведений вполне достаточно для создания действительно продвинутых моделей. Для успешного Mesh-моделирования требуется глубокое понимание основ, поэтому даже если вы решили, что вполне поняли подход, прочтите этот текст внимательно, — может быть вы сможете найти что-то нужное.



Первая часть служит азбукой и справочником по использованию Mesh-объектов и модификатора Meshsmooth в пакете 3ds max.

#### Mesh моделирование

Mesh-моделирование вместе с Meshsmooth — это самый простой и быстрый метод создания моделей органических тел. Я говорю это, пере-пробовав для создания персонажей технику патчей, surface-инструменты и nurbs. Разумеется, эти методы имеют преимущества в определенных ситуациях, но давайте сосредоточимся на Mesh-моделировании. Этот подход обладает следующими преимуществами:

1. Вы сразу же получаете результат, начиная с грубой модели, и легко подбираете нужные пропорции.

2. По мере продвижения работы, вы можете легко добавлять новые детали — нет необходимости планировать все заранее.

3. У вас есть возможность использовать сеть с разными разрешениями для ускорения анимации во вью-портах.

4. Вы сразу получаете гладкую поверхность — нет нужды в лишних Tri-patches.

5. Ни одна форма не будет слишком сложной для моделирования — это я вам обещаю.

Что такое Mesh (сеть)?

Сеть состоит из следующих основных компонентов:

- ◆ **Vertex** (вершина) — точка в трехмерном пространстве.
- ◆ **Face** (грань) — поверхность, создаваемая между тремя вершинами. Грань видима лишь с одной стороны. С какой именно, зависит от направления нормалей (**Normals**).
- ◆ **Edge** (Ребро) — само по себе оно не является компонентом, а скорее элементом грани. Фактически говорят так: ребра грани. У каждой грани есть три ребра. Ребра могут быть видимыми (**visible**) или невидимыми (**invisible**). Невидимые ребра не показываются в проволочном каркасе, однако, они все-таки показываются пунктирными линиями и их можно выбрать.

### Редактируемая сеть (**Editable mesh**)

Редактируемую сеть можно создать двумя способами:

1. Сожмите стек: в панели **Modify** нажмите **Edit stack**, и, если у вас нет никаких модификаторов, выберите **Convert to: Editable Mesh**.

Если же у вас есть модификаторы, надо их предварительно сжать (**collapse**), а затем, если модель при этом сама по себе не превратилась в редактируемую сеть, нажмите **Convert to: Editable Mesh**.

2. Примените модификатор **Edit Mesh**. В **Editable Mesh** и **Edit Mesh** одинаковые функции (за исключением анимации вершин). Кстати, если в стеке у вас присутствуют модификаторы **Edit Mesh**, то они существенно увеличивают размер файла, поскольку модификатор удваивает геометрию. **Editable Mesh** в 3ds max имеет следующие режимы подобъекта (**sub-object**):

- ◆ **Vertex** — выбор и редактирование вершин;
- ◆ **Edge** — выбор и редактирование ребер;
- ◆ **Face** — выбор и редактирование граней;

- ◆ **Polygon** — выбор граней, лежащих в одной плоскости согласно величине **Planar Thresh**. Представьте себе воду, льющуюся на поверхность: она остановится только на острых углах и, если не выбрана опция **Ignore Visible Edges** (по умолчанию), то на видимых ребрах.

Кстати, при подсчете числа многоугольников в сцене или в игре, имеется в виду число граней.

- ◆ **Element** — выбор всех прилегающих друг к другу граней в сети. Если две грани имеют общее ребро, то обе они будут выбраны.

### Выбор (selecting)

Первое, что может вам потребоваться при редактировании — это выбор того, что вы хотите изменить. В сложных моделях это может оказаться далеко не тривиальным. Здесь я опишу определенные приемы, упрощающие процесс выбора.

Вы можете использовать **Select** или **Region Select** с инструментами **Select**, **Move**, **Rotate** и **Scale**.

1. На основной панели инструментов существует три разновидности выбора области (**Region Select**):

- ◆ Прямоугольная (**Rectangular**);
- ◆ Круговая (**Circular**);
- ◆ Ломаная линия (**Fence**).

Также имеются следующие опции:

- ◆ Выбор внутри и на границе области;
- ◆ Выбор только внутри области.

2. Чтобы добавлять к выборке объекты, держите нажатой клавишу **ctrl**, а чтобы удалять из выборки объекты, нажимайте **alt**.

3. Вы можете сохранить выборки. Это очень полезно в случае сложных выборок. Сделайте выборку и присвойте ей имя в поле поименованной выборки (**Named Selections**) на основной панели инструментов. Когда вам снова потребуется выбрать эти объекты, просто возьмите эту выборку из списка. Эти поименованные выборки являются локальными по отношению к текущему режиму **sub-object**. Они также исчезнут, если вы сожмете стек.

4. **Ignore Backfacing** в свитке **Mesh Selection**. Если эта опция помечена, то вы можете выбирать только те грани, нормали которых в текущем вью-порте смотрят наружу. Испытайте действие этой опции на сферу в окне перспективы.

5. Спрячьте то, что вам сейчас не нужно видеть. Вы не сможете работать эффективно с загроможденным экраном. Исключение элементов, закрывающих вид, также облегчит создание сложных выборок. Просто выберите ребра или грани, которые хотите скрыть и нажмите кнопку **hide** на панели **Edit Mesh Selection**.

6. Мягкая выборка (**soft selection**) на панели **Mesh Soft Selection**. Эта возможность позволяет в определенной мере влиять на подобъекты, не входящие в текущую выборку. Вероятно, наиболее полезен режим **Vertex Sub-Object**. Здесь поле, на которое оказывается влияние, показано цветом, благодаря чему легко понять, что именно будет задействовано. Назначение параметров этой панели понятно из их названий. Для изменения объема, на который будет оказано влияние, используйте параметр **falloff**.

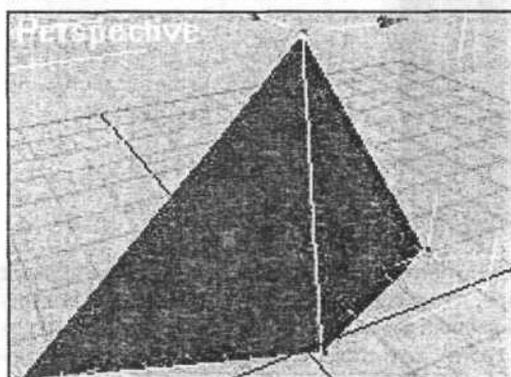
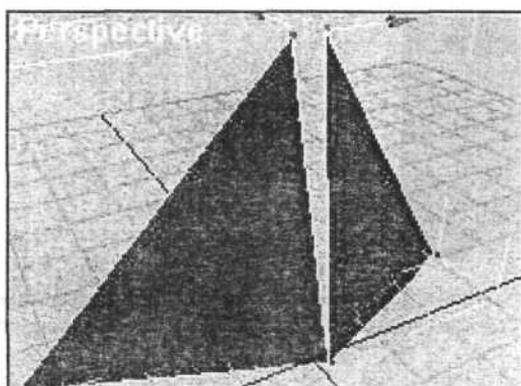
## Моделирование

Как редактировать объект? Прежде всего, можно трансформировать все его части — то есть, их можно перемещать, вращать и масштабировать. При трансформации можно держать нажатой клавишу **shift** и таким образом клонировать выбранный подобъект(ы). Благодаря трансформационному «гизмо» стало возможным масштабирование в окне перспективы. Однако иногда это «гизмо» лишь вводит в заблуждение. Тогда вы можете его отключить/включить нажатием клавиши **x**. Если вы много работаете с вью-портом с тенями (**shaded**), будет разумно включить отображение **edged faces**. Это можно сделать нажатием клавиши **F4**. При этом, если можно так выразиться, проволочный каркас отрисовывается поверх затененного объекта.

Теперь перейдем к подобъектам.

### Вершина (vertex)

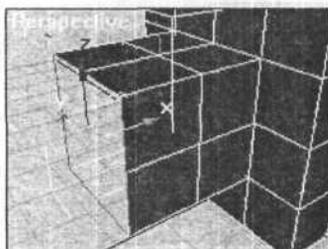
Больше всего корпеть приходится именно над трансформацией вершин. В этом режиме можно также сваривать или сплавлять вершины (**weld**). Это означает объединение двух или более вершин в одну, прикрепляя грани и их ребра к одной общей вершине. Просто выберите вершины, которые хотите сплавить, и нажмите **selected** в группе **weld**, расположенной в панели **edit geometry**. При этом все выбранные вершины, расположенные на расстоянии друг от друга не более порогового (**threshold**), будут сплавлены.



Гораздо реже вам может потребоваться создать (**create**) или разбить (**break**) вершину (также на панели **edit geometry**). Процедура создания вершины вполне очевидна, а действие команды **break** противоположно команде **weld**, то есть, выбранная вершина расщепляется на количество копий, равное количеству прилежащих к ней граней.

### Ребро (edge)

Одним из фундаментальных приемов при **mesh**-моделировании является техника штамповки (**extrusion**). Это можно с равным успехом проделать в режимах **edge** или **face sub-object**. Штамповку ребер можно произвести несколькими способами. Я обычно просто пользуюсь кнопкой **Extrude** или перемещаю ребра с нажатой клавишей **shift**. Последняя возможность обычно используется для клонирования, но в этом режиме дает тот же результат, что и **extrude**.

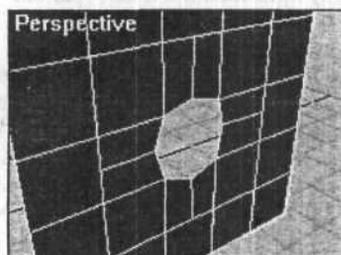


Можно, разумеется, воспользоваться и штамповкой с масштабированием. Это удобно для создания более густой сетки, например, вокруг глаз некоего существа.



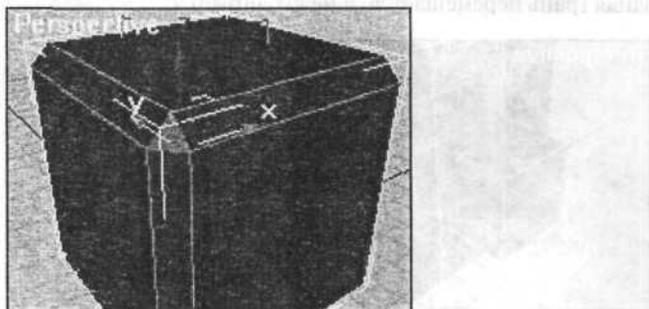
Ведя разговор о создании более частой сетки, стоит обратиться к функции cut. Это то же самое, что и cut в режиме face sub-object. Нажмите cut кнопку на свитке edit geometry и выберите ребро. Теперь вы можете провести линию через те грани, которые хотите разрезать. Вам не придется указывать каждое ребро, которое вы хотите рассечь. Если хотите, можете рассечь линией весь объект. Нажмите правую кнопку мыши, чтобы закончить или прервать операцию.

**Совет:** Проверьте опцию Ignore Backfacing, чтобы не выбирать невидимые ребра.



Попытайтесь выбрать все ребра, чтобы увидеть даже невидимые (точечные линии). Обратите внимание, если вы закончили надрез посередине грани, то дополнительные грани будут созданы автоматически.

Другой полезной функцией является создание фаски (**Chamfer**). Выберите ребра, для которых вы хотите создать фаску, и нажмите кнопку **Chamfer** на панели **Edit Geometry**. Проследите, чтобы случайно не получились пересекающиеся грани.



Внизу панели **Edit Geometry** есть очень полезные кнопки. Первая — это **Select Open Edges**. Нажав ее вы можете проверить, нет ли в вашем mesh'e случайных разрывов. В этом случае, чтобы исправить это положение, вам придется сплавить несколько вершин. Эта возможность также может быть полезна для выбора всех ребер вокруг глаза или в месте крепления ноги.

Вторая кнопка — это **Create Shape from Edges**. Она может вам понадобиться, например, при моделировании окна машины, или при использовании сплайнов для точной подгонки, или для конвертирования в сплайны объекта в целом.

### Mesh ⇄ Spline ⇄ Mesh

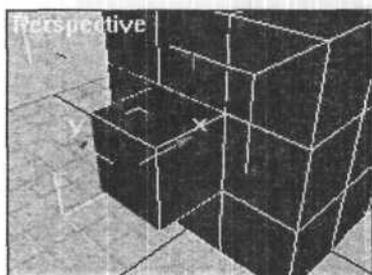
Войдите в режим **edge sub-object**, выберите **Edit ⇄ Select all** и нажмите **Create Shape from Edges**. В появившемся диалоговом окне выберите опции **Linear** и **Ignore Hidden Edges**. Выйдите из режима **edge sub-object** и удалите **mesh**-объект. Выберите сплайновый объект и примените к нему модификатор **Surface**. Выберите опцию **Remove Interior Patches** и сбросьте в 0 параметр **Steps**. Примените модификатор **Edit Mesh** и сожмите стек. Теперь мы в исходной точке!

В панели **Surface Properties** у нас есть две кнопки для определения видимости ребра. Я надеюсь, вы понимаете, как их использовать.

### Грани (Face)

Мы уже говорили о штамповке. Одно из отличий от режима **edge sub-object**, состоит в том, что грань нельзя отштамповать, передвигая с нажатой клавишей **shift**. Из-за этого выбранные грани просто продублируются без привязки к исходному элементу.

Чтобы произвести штамповку вам придется выбрать грани, а затем нажать кнопку **extrude**. Другое отличие состоит в том, что при штамповке исходная грань перемещается, а не дублируется.



**Cut** работает также, как и в режиме **edge sub-object mode**.

В некоторых случаях вам может понадобиться создать новые грани. Это совершается нажатием клавиши **create** и выбором последовательности из трех вершин.

Направление нормали к создаваемой грани зависит от направления обхода точек, — по или против часовой стрелки. Если вы создали грань и не видите изменений на экране, значит, нормаль направлена в противоположную от вас сторону. Перерисуйте грань в другом направлении.

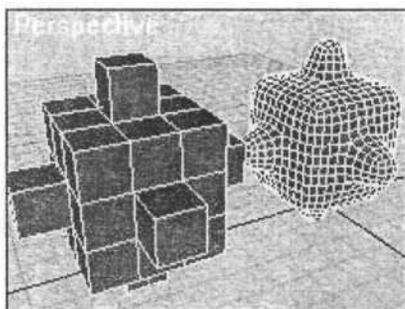
Раз уж мы заговорили о нормалях, остановимся на группе **Normals** в панели **Surface Properties**, где расположена кнопка **flip**. Она позволяет перевернуть нормали выбранных граней.

### Полигоны и элементы

Полигоны и элементы — лишь альтернативные способы выбора граней. Стало быть, редактирование производится так же, как и для граней.

### Сглаживание (Meshsmooth)

Сглаживание — это необходимый элемент **mesh**-моделирования. Без него **mesh**-моделирование не стало бы приемлемым выбором для моделирования объектов с высоким разрешением.



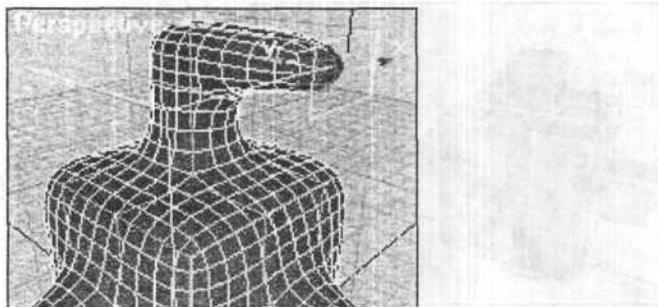
**Meshsmooth** — это модификатор, который обычно применяется последним в стеке. Он не нарушает координат мэппинга или идентификаторов материала соответствующей сетки.

Единственный параметр, который я обычно меняю в этом модификаторе — это **Subdivision Iterations**. Этот параметр определяет, насколько густой и гладкой должна быть сеть. При этом во вью-портах и при рендеринге можно иметь разные величины. Просто выберите опцию **Render Values Iterations** и введите соответствующую величину. Обычно для вью-портов достаточно одной итерации, а при рендеринге эту величину можно увеличить до 2.

Когда **Meshsmooth** применяется к полигонам (эта опция бывает выбрана по умолчанию, и, как правило, это именно то, что нужно), большое значение имеет видимость ребер. Невидимые ребра попросту игнорируются. (Как в режиме **Polygon Sub-Object** модификатора **Edit Mesh** с параметром **Planar Thresh**, равным 180).

Есть еще одна возможность. При сглаживании существует два режима подобъекта: **Vertex** и **Edge**. **Edge** я пока не стану затрагивать, и отмечу лишь, что в режиме **Vertex Sub-Object** можно подбирать вес каждой из вершин, определяющий ее вклад при образовании сглаженной поверхности. Так, геометрия стягивается к вершине, обладающей большим весом.

Может оказаться полезным (хотя я обычно этого не делаю) применить **Meshsmooth**, вернуться в **Editable Mesh** и включить опцию **Show end result**. Это позволит вам увидеть сглаженный объект и проволочный каркас исходного объекта, полностью доступный для редактирования.



### Создание монстров с использованием Plug-in Clay Studio

После установки **Clay Studio** все объекты Plug-in'a можно найти в командной панели во вкладке **Create**.

Начнем с головы или, точнее, с верхней части, причем в этой самой части, как раз, будут сосредоточены все построятельные объекты plug-in'a **Clay Studio Pro**.



Гребешок, голова и нос (хобот) — это **Clay Spline**. К гребешку применен тип **Elipsoid**, к остальным **Round**. Глаза это **Clay Sphere**.



Ямки под зрачки сделаны тем же **Clay Sphere**, только с отрицательным **Influence Field** (область воздействия), в этом случае происходит не слияние объектов, а наоборот, вдавливание. Следующий шаг — создание рук.



Здесь все просто, все те же **Clay Spline**. Если что-то не получилось, всегда можно все поправить при помощи команды **Modify**. Вообще, все объекты **Clay Spline** можно модифицировать также, как и простые равнозначные объекты 3ds max'a.

Осталось приделать, туловище, ноги (ласты), хвост, ну, в общем все что в голову придет.



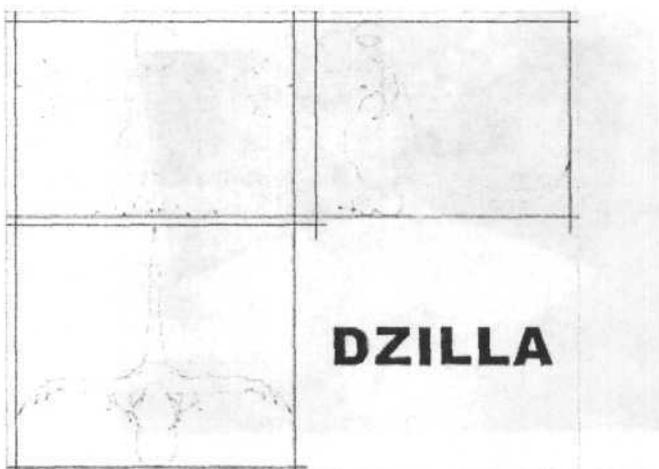
Дело осталось за малым... или большим.

Чтобы все объекты выполнили свои воздействия друг на друга, надо поместить значок **Clay Surface** в каком-либо из окон и нажать кнопку **Absorb Free Primitives** в появившемся меню. В результате объекты **plug-in'a** приобретут форму будущего объекта. Чтобы создать сам объект, в том же меню надо открыть свиток **Utilities** и предварительно выставив в свитке **Parameters** количество деталей будущего объекта, нажать кнопку **Create Snapshot Mesh**. Осталось только удалить все **Clay** объекты и назначить материал оставшемуся **Mesh**-объекту.



### Маленький Годзилла

Меня всегда интересовало создание фантастических существ. Когда я впервые увидел программы трехмерной графики, я сказал себе: «Теперь я могу нарисовать все, о чем я только думал!». Но в то время у меня не было достаточного опыта, чтобы осуществить все, что мне хотелось. Сейчас, когда я у меня такой опыт есть, я хочу поделиться им с вами и вместе создать Дзиллу (*Dzilla*). Дзилла, как вы уже догадались, это маленький Годзилла. На работу также повлияло мое увлечение персонажем японских комиксов Гоном (*Gon*) — маленьким динозавриком с большой головой. Перед созданием Дзиллы я набросал несколько черновых рисунков, чтобы понять, как будет выглядеть мой Дзилла и выбрать наиболее подходящий способ моделирования. Эти рисунки представляют собой виды спереди, сбоку и сверху и будут помогать нам в течение всей работы. Необходимо заметить, что рисунки пропорциональны друг другу, чтобы избежать неточностей в процессе моделирования.



Я использовал 3ds max совместно с **MetaReyes** плагином. Это очень мощный плагин, основанный на принципе метаболов (meatballs) для создания органических форм.

### Создание шаблона

До начала моделирования мы создадим три **Patch Grids** для наложения на них наших рисунков – вид спереди, вид сзади и вид сверху. Откройте **Create Panel**, выберите **Patch Grid** и создайте **Quad Patch**. Размер сеток очень важен, поэтому выбирайте его очень осторожно. Откройте любую программу для обработки графики и посмотрите точные размеры наброска переднего плана в пикселях – в моем случае этот размер составляет 482x673. Размер сетки должен быть следующим:

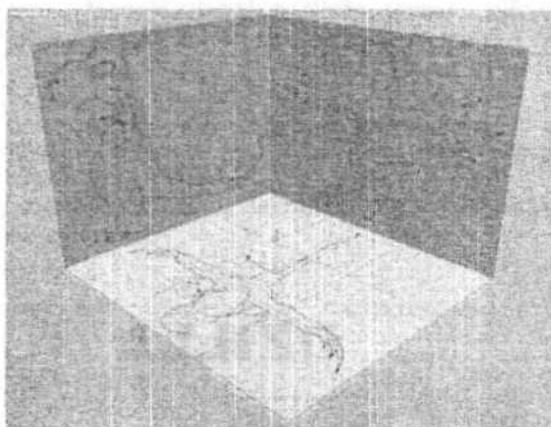
Length 482

Width 673

Убедитесь, что функция **Generate Mapping Coords** включена.

Откройте Редактор Материалов, создайте новый стандартный материал и назовите его «Вид Спереди». В роли диффузии используйте графический файл с видом спереди (например: `dzilla_sketch_front.jpg`). Не забудьте включить функцию **Show Map in Viewport**. Назначьте этот материал **Patch Grid**.

Повторите эту операцию для создания остальных двух видов. В результате должно получиться нечто похожее на то, что изображено на рисунке ниже. Если некоторые изображения наложены неточно, попробуйте откорректировать ее размеры.

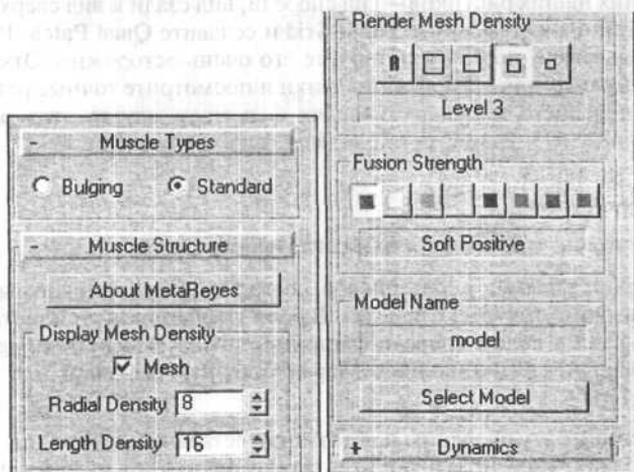


### Моделирование

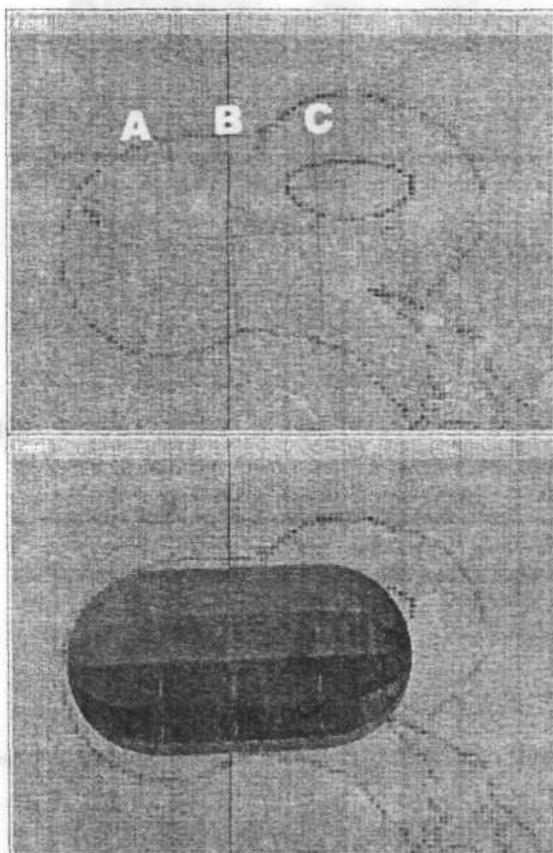
Начнем с головы. Для этого крупный план головы в **Left Viewport**.

Откройте панель **Create** и выберите **MetaReyes object** в меню **REM Primitives**.

Не нужно менять никаких параметров, кроме **render mesh density**, он должен быть отмечен галочкой. Я предпочитаю Level 1 или 2.

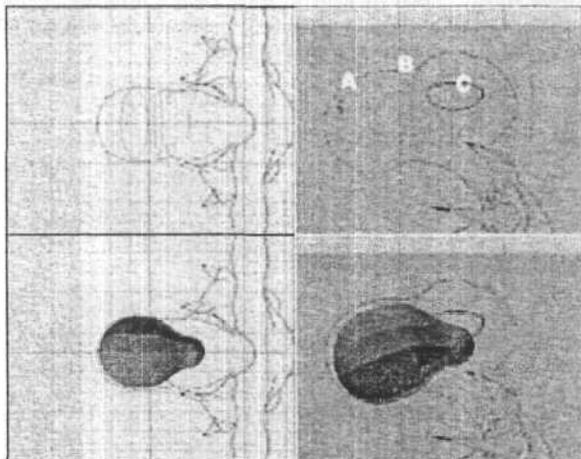


Перейдите в Вид Слева и создайте метамускул (назовем так наш **MetaReyes Object**). Для этого просто создайте первую окружность (A) по размеру кончика носа, затем нажмите **Shift**+левая кнопка мыши – так мы получим второй радиус (B), равный (A); и закончим фигуру созданием третьего радиуса (C) (правая кнопка мыши).

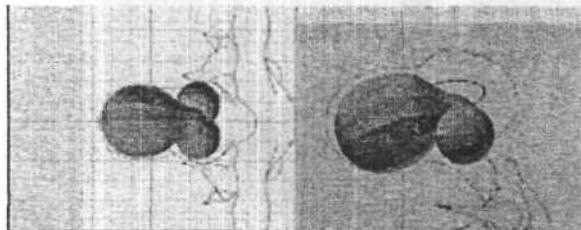


Теперь откройте панель **Modify**, нажмите кнопку **More** и выберите **Edit Muscle**. Займемся корректировкой формы нашего метамускула. Нажмите кнопку **Sub-Object**, переместите вершину (**Vertex**) (A) слегка книзу и выберите функцию **Edit Radii** (выпадающий список следом за **Sub-Object**) и отрегулируйте радиус (A) в соответствии с формой носа на рисунке. Повторите эту операцию с (B) и с (C). Мы могли добиться по-

хожих результатов и без модификатора **Edit Muscle**, но это потребовало бы больше времени.



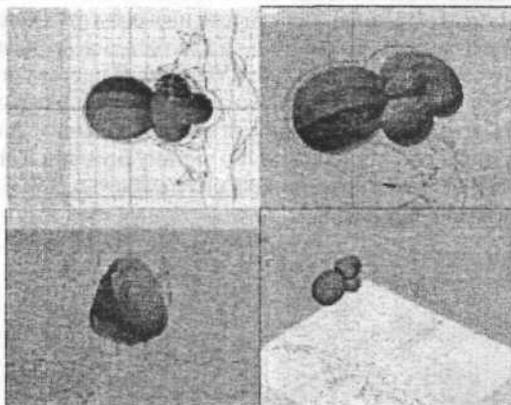
Создайте еще 2 метаболы (**Metaballs**), они послужат для образования челюстей нашего динозаврика.



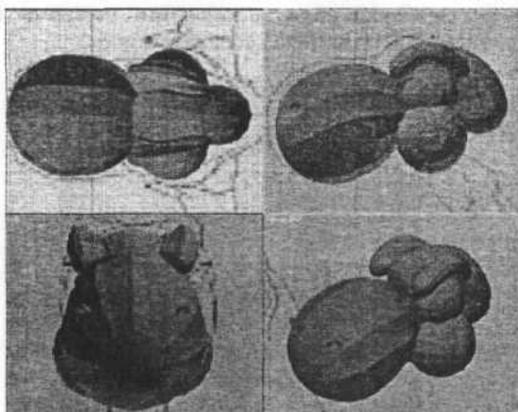
Добавьте еще один метамускул для образования задней части головы.



Затем добавьте два отрицательных метаболо (**Negative Metaballs**) для образования глазных впадин. Для этого нужно выставить параметр **Fusion Strength** в положение **Medium Hard Negative**.

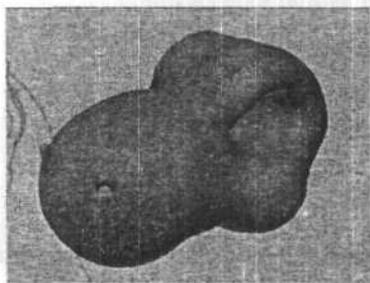


Теперь нам нужно добавить пару метамускулов поверх метаболов зеленого цвета, чтобы наш монстр выглядел более зловеще. Далее нужно добавить еще пару изогнутых метаболов для создания ноздрей. Вы должны получить нечто подобное тому, что изображено ниже.

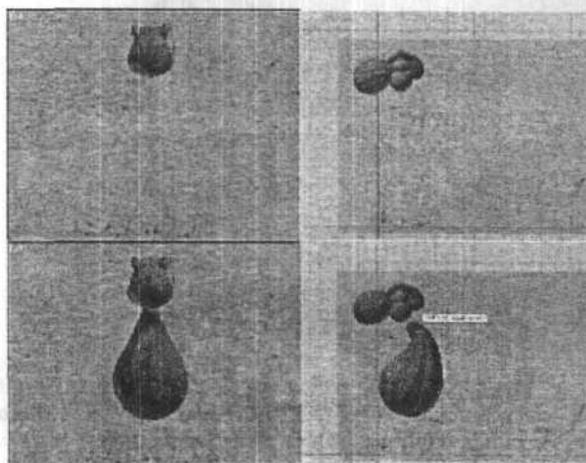


Помните, что каждый раз, когда вы создаете новый метабол вы можете использовать инструмент зеркального отображения (**Mirror Tool**) для создания зеркальной копии (половинки туловища у этой модели симметричны друг другу).

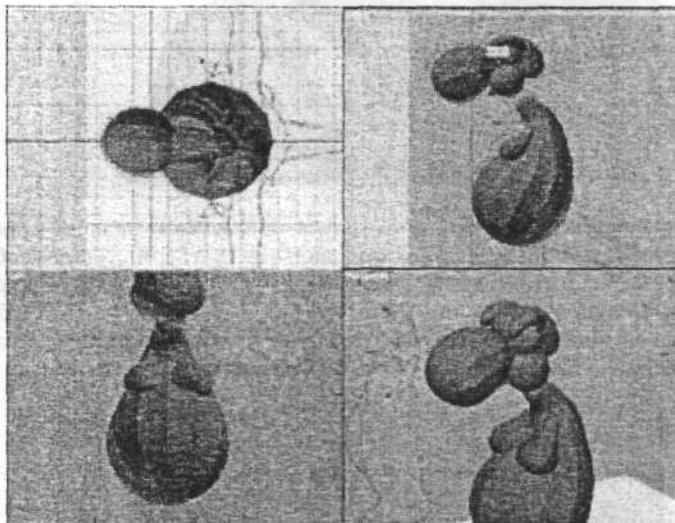
Теперь можно провести тестовый рендеринг получившейся головы. Откройте панель **Utilities** и выберите **MetaReyes Utilities Rollout**. Нажмите на **Select Model** и выберите тот метамускул, который у нас служит для отображения задней части головы. Затем нажмите на **Make Group** — ваша группа будет названа Group00 (нет смысла называть эту группу как-либо еще, так как она будет нам нужна только для тестового рендеринга). Выполните пробный рендеринг в **Perspective Viewport**. Я настоятельно рекомендую вам сделать несколько пробных изображений и разных позиций, чтобы выявить возможные неточности моделирования. В случае, если вас не удовлетворит то, как какой-либо метамускул вписывается в общую форму головы, вы можете подвигать его немного и затем произвести еще один пробный рендеринг.



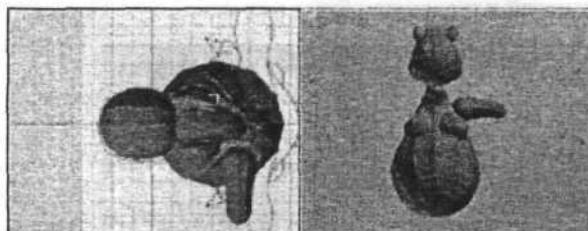
Теперь можно сосредоточиться на создании туловища нашего монстра. Руководствуясь рисунком ниже, создайте живот.



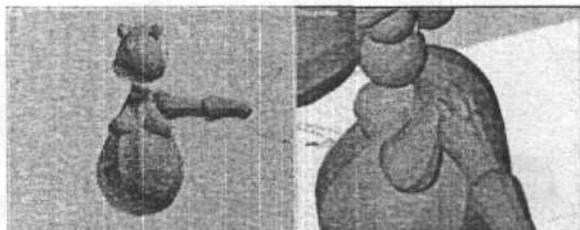
Добавьте два метамускула, чтобы образовать грудь:



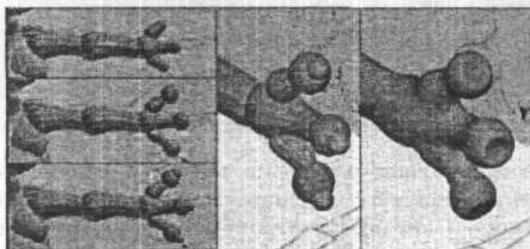
Если у вас все получилось правильно, то можно перейти к созданию рук. Сначала мы полностью смоделируем левую руку, а затем создадим ее зеркальную копию — правую руку. Заметьте, что в **Top Viewport** созданный мной метамускул не соответствует контурам рисунка, но в данном случае такое положение более логично чем на рисунке — иногда в процессе моделирования необходимо полагаться на ваши ощущения больше, чем на базовый рисунок.



Теперь создадим вторую часть руки и добавим несколько мелких деталей к плечу.



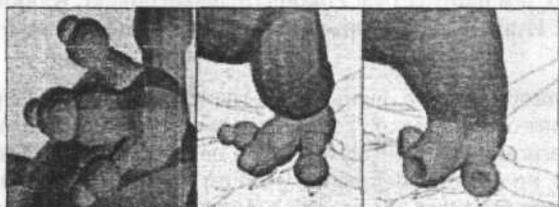
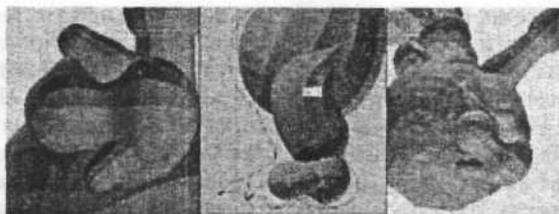
Теперь создайте три метамускула, образующие пальцы, и добавьте три метаболла, чтобы сделать концы пальцев круглыми. Далее необходимо добавить три отрицательных метаболла (**Negative Metaballs**) — параметр **Fusion Strength** в положении **Hard Negative** — чтобы получить углубления для когтей. Выполните тестовый рендеринг, чтобы проверить, достаточно ли хорошо выглядят полученные углубления.



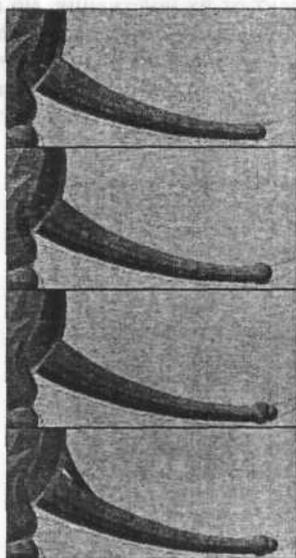
Теперь можно заняться ногами. Эта форма достаточно проста — воспользуйтесь навыками, полученными в процессе создания головы и рук. Следите, чтобы нога и живот перекрывали друг друга. В этом случае переход между ногой и туловищем будет мягче.



На ноге у нашего динозавра также три пальца, причем средний должен быть крупнее остальных. Моделируя пальцы на ноге пользуйтесь теми же принципами, что и для пальцев на руке, не забудьте углубления для когтей.



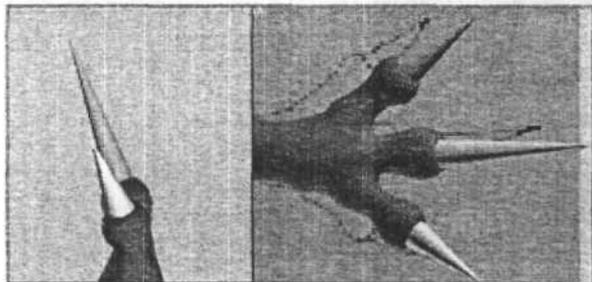
Последний этап моделирования — хвост. Кончик хвоста (также как и кончики пальцев на руках и ногах) должен быть округлым. Выполните пробный рендеринг. Пробные изображения в процессе моделирования обеспечивают больший контроль над моделью и помогают сразу выявить возможные недостатки.



Выберите все части левой руки и левой ноги и создайте их зеркальную копию. Мы почти закончили работу с **MetaReyes** моделированием, единственное, что осталось — создать окончательный объект (**Mesh**). Для этого откройте панель **Utilities** и разверните список **MetaReyes Utilities**, выберите **Select Model** и кликните на одном из метамускулов. Назовите группу **Dzilla** и кликните на **Make Group**. В списке **Mesh Generation** выберите **Select Model** и отметьте группу **Dzilla**, введите имя **Dzilla**, как имя всего вашего объекта. Проверьте параметр **Cube Value**. Чем меньше установленное число, тем плотнее будет окончательный объект. Кликните на **Mesh Generation**. Итак, вы получили окончательный объект — тело нашего динозавра.

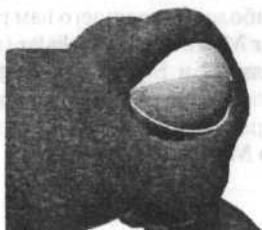
У полученного объекта очень много граней, но зато после рендеринга он выглядит очень гладко. Тем не менее нам нужно оптимизировать его. Оптимизатор, который находится в панели **Modify** работает не очень аккуратно, но можете воспользоваться им. Более выгодная альтернатива ему: **Polygon Cruncher**, он является очень мощным средством оптимизирования моделей. Если у вас его нет — воспользуйтесь стандартным оптимизатором (**Modify Panel** ⇨ **Optimize**).

Теперь перейдем к когтям. Откройте панель **Create** и создайте конус (**Cone**), установите Радиус 1 достаточно большим, чтобы подходить к углублениям, которые мы сделали специально для когтей. Радиус 2 должен быть равным 0. Длина когтей зависит от их положения, но коготь среднего пальца должен быть длиннее остальных. Я пользовался следующими параметрами: **Height Segments=6**; **Cap Segments=1** и **Sides=10**. не забывайте следить, чтобы то, что вы делаете, совпадало с рисунками. Теперь осталось применить модификатор **Bend** к когтям (изгиб когтей на средних пальцах и хвосте должен быть больше).





Осталось сделать глаза. Для этого подойдут вложенные друг в друга сферы (**Spheres**) одна — собственно глаз — несколько меньше радиусом, и вторая — больше радиусом и с параметром **Hemisphere** равным **0.5** — веко. Необходимо, чтобы центры обеих сфер совпадали. Для этого, после создания первой сферы нажмите **Shift** и кликните левой кнопкой на сфере. В результате получится копия первой сферы — отрегулируйте Радиус и параметр **Hemisphere**. Следите, чтобы при создании сфер параметр **Generate Mapping Coords** был включен.



Осталось раскрасить нашего динозаврика.

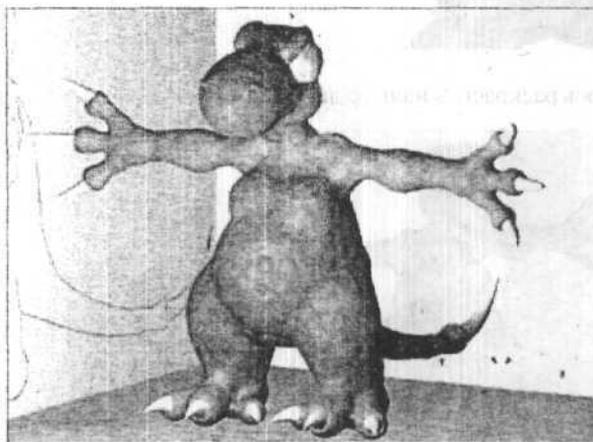


### Текстурирование

Это, пожалуй, самая важная и сложная часть — от нее зависит, насколько реально будет выглядеть окончательное изображение.

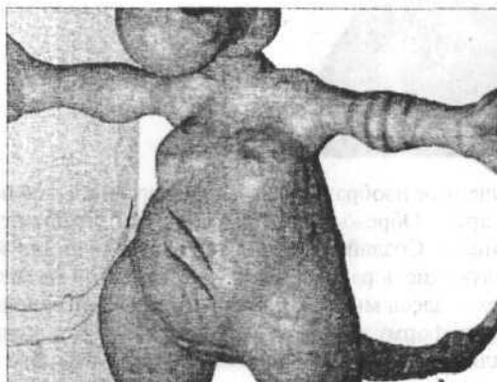
Нам необходимо решить, какую текстуру использовать в роли основной, для того, чтобы швы на модели были менее заметны.

Исходя из того, что наш Дзилла — динозавр, у него должна быть чешуя. Нам нужно создать **Multy/Sub-Object Material**, чтобы корректно работать со всеми текстурами, которые мы собираемся создать. Откройте **Material Editor** и создайте **Multy/Sub-Object Material**. В первой ячейке создайте стандартный материал. Мы назовем этот материал: **SkinGeneral** (Основная кожа). В секции **Bump** мы используем **Cellular map**, величину параметра **Bump** установим равной **70**. Этот тип карты лучше всего подходит для создания чешуи. Нет надобности менять какие-либо параметры, кроме размера чешуй (**size of the cell**) и сглаживания рельефа (**bump smoothing**). Я использовал в первом случае — **80**, а во втором **1.0**. У вас могут получиться другие результаты, так что можете свободно экспериментировать с параметрами для получения наиболее подходящего вам результата. Затем скопируйте полученную **Cellular Map** в ячейку **Cellular** (просто перетащите мышкой). Установите **Shininess=12** и **Shininess Strength=21**. Полученный материал будет служить образцом для создания материалов для остальных частей тела нашего динозавра. Назначьте полученный материал основной модели динозавра (**Assign Material**).



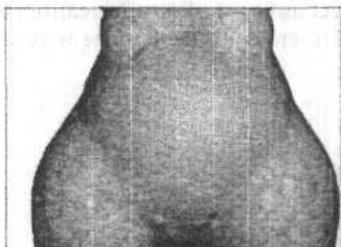
Теперь мы добавим некоторые детали к животу нашего Дзиллы — несколько свежих кровоточащих ран, полученных в схватке. Выделите нужные грани (**Faces**) на животе — для этого откройте панель **Modify**, вы-

берите **Edit Mesh** и в меню **Sub Object** выберите **Face**. Выделите грани на животе нашего динозавра. Откройте список **Edit Surface** и установите **Material ID** равным 2.



Нам необходимо изменить систему проекционных координат (**mapping**) граней живота на **Planar**. Для этого нам понадобится еще один модификатор – **Mesh Select**. Установите **Material ID** равным 2 и нажмите **Select By Material ID**. Эта процедура выделит нужные грани и теперь можно применить **Planar UVW Mapping**, проследите, чтобы Габаритный контейнер (**Gizmo**) находился в нужной позиции.

Второй слот в нашем материале пуст, пока оставим его пустым. Выполните рендеринг фронтального вида и сохраните изображение – так как на грани живота текстура еще не наложена, это позволит вам в последствии более точно определить, где рисовать шрамы.



Откройте полученное изображение в Photoshop, чтобы нарисовать нужные растровые карты. Обрежьте изображение так, чтобы осталась только серая часть живота. Создайте новый слой (**Layer**) и нарисуйте два шрама, используя белую кисть радиусом в один пиксель. Создайте еще один слой поверх всех — здесь мы будем рисовать кровь. Я использовал кисть, чтобы нарисовать форму кровавых подтеков и затем использовал инструмент **Smudge** для окончательной доводки. Затем я немного затемнил кровь в области шрамов, чтобы создать ощущение запекшейся крови с помощью инструмента **Burn**.

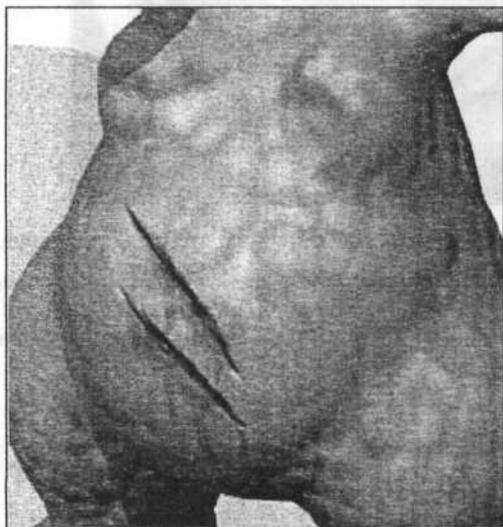
Создайте еще один слой, с удерживая клавиши **Ctrl+Shift** кликните на слоях со шрамами и кровью, полученную выделенную область залейте белым цветом. Теперь создайте черный слой под новыми слоями (но поверх исходного изображения) и сохраните каждый слой в отдельный файл с черным фоном. Назовем их `dzilla_belly_alpha.jpg`, `dzilla_belly_color.jpg` и `dzilla_belly_bump.jpg`.



Вернитесь в 3ds max 7.0 и откройте **Material Editor**. Скопируйте первый внутренний материал (**first sub-map**) во второй слот, назовем но-

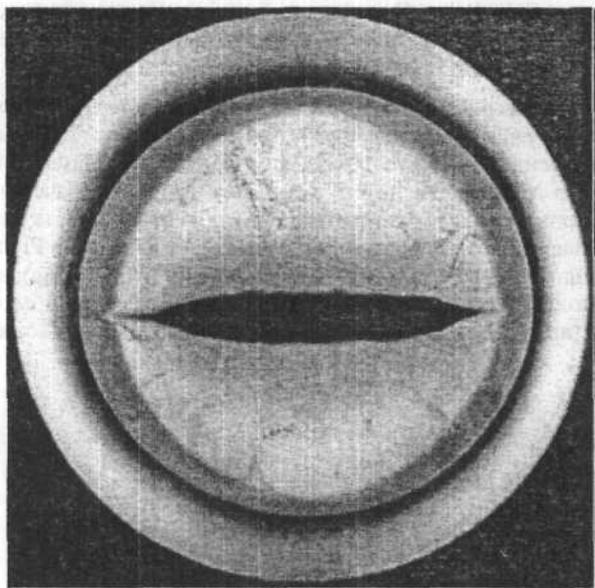
вый материал — **Skin Belly**. Перейдите в секцию **Bump** и кликните на кнопке **Bump** (следующая за **Type**), откроется **Material/Map Browser**, где нам нужен материал типа **Mix**.

На вопрос: **Keep the old map as a sub-map?** ответьте **Да**. В слот 2 мы установим карту **dzilla\_belly\_bump.jpg** в списке **Output** установим **Bump Amount** на 6. Перейдем в родительский материал (**Parent Material**) и установим параметр **Mix Amount** на 0.5. Давайте разберемся, зачем мы это сделали. Я использовал карту типа **Mix** для того, чтобы смешать карту рельефа чешуи (**bump map — cellular**) и новую карту, созданную в Photoshop. Следующим шагом будет создание карты Маски (**Mask Map**) в слоте **Diffuse** — используйте для этого карту **dzilla\_belly\_color.jpg** в слоте для изображения (**map slot**) и **dzilla\_belly\_alpha.jpg** в слоте для Маски (**mask slot**).



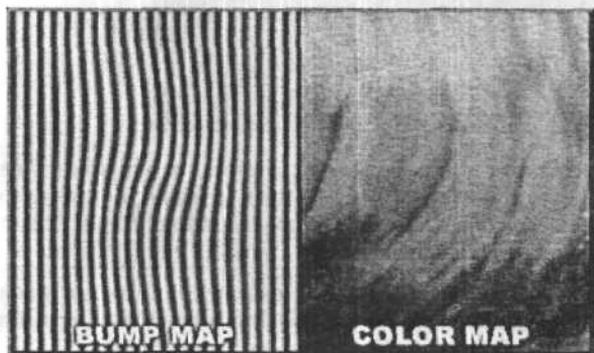
Ту же технику я использовал для затемнения кончиков пальцев и хвоста, я добавил немного грязи на груди и ноздрях, с помощью **Bump map** я сделал нашему динозавру татуировку на спине. Для всех карт я использовал **Planar UVW Mapping**.

Для глаз применена стандартную текстуру. Я подредактировал ее в Photoshop. Применил **Planar UVW Mapping** к сфере-глазу. Параметры материала следующие: **Shininess=37** и **Shininess Strength=100**.

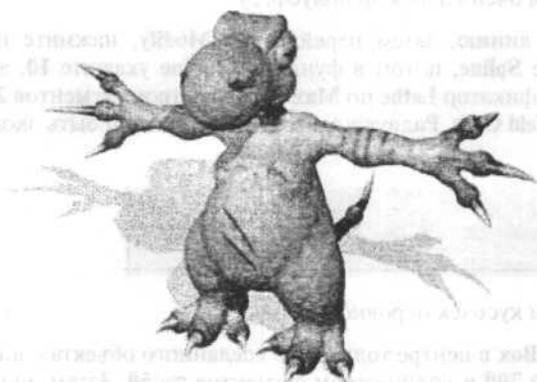


Для век я использовал тот же материал, что и для туловища.

Что касается когтей, я хотел, чтобы основания у них были черные, а кончики желтые. Специальных проекционных координат (UVW Mapping) я не применял. Величину **Bump** я выставил равной 30, **Shininess=29**, **Shininess Strength=52**.



Теперь можно выполнять финальный рендеринг — наш Дзилла готов.



## Глава 36. Моделирование растений

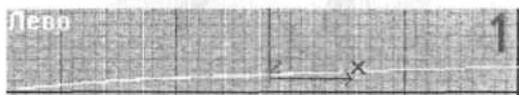
### Облака и трава

Мы научимся создавать небо специальным способом с последующим наложением прозрачного материала **Smoke** для имитации облаков. Траву мы смоделируем при помощи плагина **Shag Fur**, который для это-

го идеально подходит, так как не перегружает сцену вершинами, а делает все за счет объемного света.

#### 1. Создаем очень плоскую полусферу.

Создайте линию, затем перейдите в **Modify**, нажмите на **Sub-Object**, выберите **Spline**, потом в функции **Outline** укажите **10**, а потом примените модификатор **Lathe** по **Max** с количеством сегментов **20** и нажатой опцией **Weld Core**. Радиус у этого объекта должен быть около **4500** единиц.



#### 2. Создаем кусочек неровной земли.

Создайте **Box** в центре только что сделанного объекта с шириной и длиной равной **700** и количеством сегментов по **50**. Затем примените модификатор **Noise: Seed 6, Strength** по **Z = 100**.

#### 3. Делаем небо.

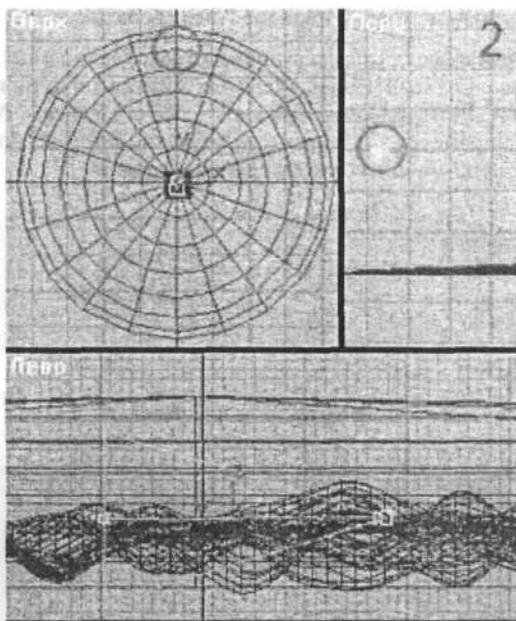
Зайдите в **Environment** кликните на **Color** и выберите **RGB 97, 159, 206**. Теперь щелкните на кнопку **Add** и выберите **Fog**, в нем установите **Standard** и опцию **Exponential**, а также **Far = 25%**. Перейдите в редактор материалов и создайте материал Небо, в **Diffuse** выберите белый цвет, **Self-Illumination** сделайте равным **100**, а **Shin. Strength 0**. В **Opacity** выберите **Smoke** в его параметрах установите **Size 100, Iterations 5, Phase 1, Exponent 2,04, Color 1 — черный, Color 2 — белый**. Назначьте материал «полусфере».

#### 4. Материал для земли.

Создайте материал Земля, выберите картинку и задайте ей **Tiling** равный **15**, в **Filtering** установите **Symmed Area**. Скопируйте полученную карту в **Bump**.

#### 5. Камера и свет.

Создайте **Omni Light** в **Multipier** установите **2**, включите **Cast Shadows, Size 1024**. Разместите. Кликните правой кнопкой на объект «полусфера» и в его параметрах уберите опцию **Cast Shadows**. Камеру создайте над поверхностью земли.



#### 6. Трава.

Зайдите в **Environment**, кликните кнопку **Add**, выберите **Shag Fur**, кликните на **Pick** и выберите землю, на земле появятся палочки — трава, далее **Length Maximum** сделайте равным 9, **Random Factor** 0,75; **Density Per Area** 1,2; **Thickness Maximum** 0,04; **Leaning Amount** 0,85, **Random Factor** 0,35, **Max Angle** 157. В **Bending** включите **Normal** ⇌ **Lean Direction**; **Random Middle Strands** 0,21, **Vertex Strands** 1; **Picture Quality/Memory** 5, **Bizier Steps** 6; **Color Base RGB** 78,100,74, **Tip RGB** 35,122,35, **Bias** 0,03, **Random Factor** 0,12.

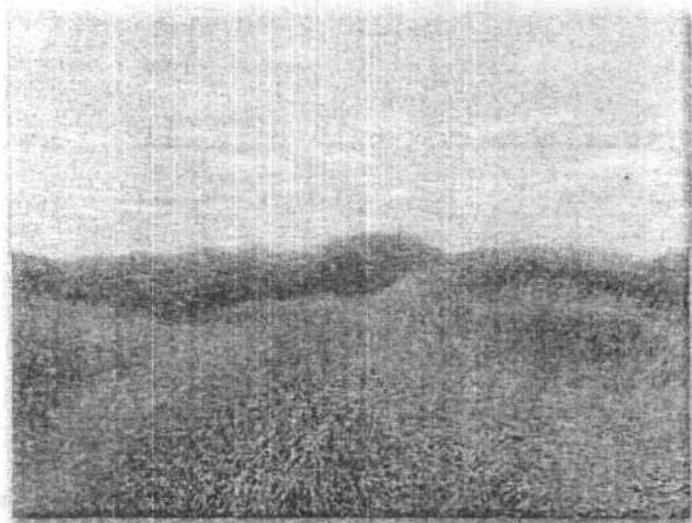
Теперь снова кликните кнопку **Add** и выберите **Shag Render** нажмите кнопку **Convert Light to Hair Enabled**.

#### 7. Дополнение. Анимация и размеры.

Если нужно увеличить размеры облаков, измените параметры **Tiling** с 1 на 0,5-0,3. Анимация производится посредством изменения значения **Phase** с 1 до 1.1 на 3-4 секунды анимации, также можно сделать небольшое перемещение при анимации **Tiling Offset** с 1 до 5 при нормальной погоде, и с 1 до 10 при сильном ветре на каждые 3-4 секунды анимации.

### 8. Как сделать больше травы, и как ее анимировать?

Количество травы увеличивается посредством увеличения значения **Density Per Area** в более высокие значения. Анимация производится в основном за счет анимирования параметра **Vertex Strands**. Можно также изменяемому параметру задать контроллер зашумления, но с ним придется повозиться, чтобы добиться хороших результатов.



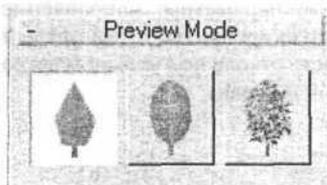
## Глава 37. Plug-in Tree Storm, позволяющий создавать растения

Очень удобный и простой в использовании Plug-in, я бы даже сказал просто объемная библиотека готовых объектов с возможностью незначительной модификации.

Нажав кнопку **Tree...** в свитке **Choose** можно загрузить готовую модель растения из каталога **TREE Library WIN**.

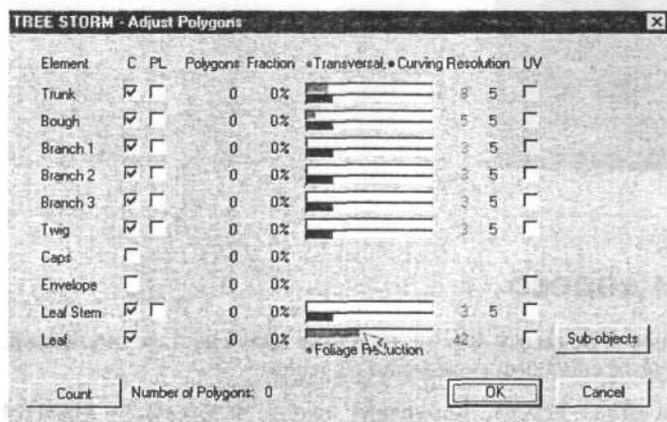
Из относительно большого списка изменяемых параметров хотелось бы выделить всего несколько действительно полезных и реально что-то изменяющих.

## Свиток Preview Mode



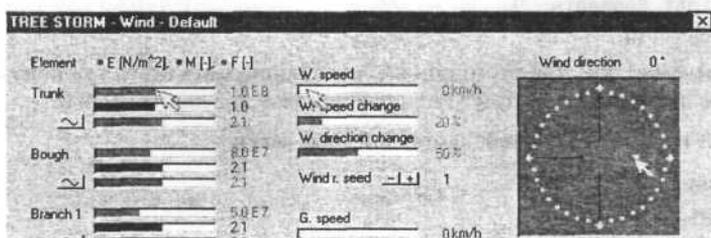
При нажатии определенной иконки в окнах проекции объект будет выводиться с соответствующей детализацией.

## Свиток Polygons, кнопка Adjust

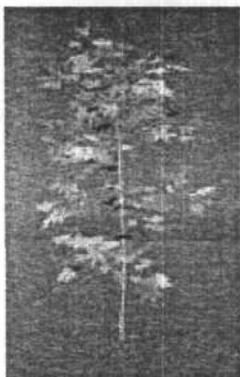


Если изменить положение уровня в самом нижнем индикаторе, напротив надписи **Leaf**, будет изменяться количество листьев на дереве.

## Свиток Wind



Выставьте метку напротив надписи **Active** и кнопкой **Adjust** вызовите меню с параметрами. Основные параметры: направление ветра (красная стрелка на зеленом поле), его скорость (верхний правый индикатор под надписью **W.Speed**) и гибкость ствола под напором ветра (верхний левый индикатор напротив надписи **Trunk**).



## Глава 38. Сажаем дерево

Этот способ не самый легкий, но в тоже время достаточно гибкий. Нарисовать таким способом можно любое дерево.

Рисуем ствол дерева. Командная панель ⇨ **Create** ⇨ **Standard Primitives** ⇨ **Cone**.

Radius 1: 8,0

Radius 2: 0,0

Height: 200,0

Height Segments: 9

Sides: 10

Рисуем ветки. Для этого понадобится нарисовать всего три конуса, радиус в каждой ветке будет уменьшается на 0,5, а длина на 30,0.

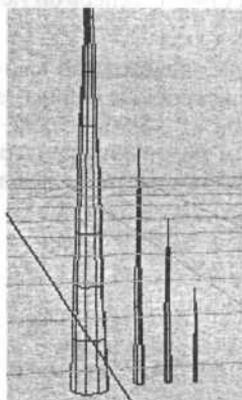
Вот параметры первой, самой длинной ветки.

**Cone:**

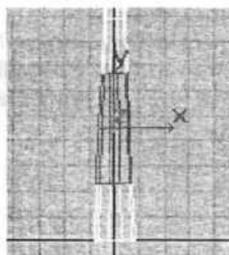
Radius 1: 2,0

Radius 2: 0,0

Height: 100,0  
 Height Segments: 3  
 Sides: 6



Теперь выделите первый конус (будущий ствол), в командной панели во вкладке **Modify** при помощи кнопки **More...** выберите команду **Edit Mesh**. В появившейся области свитков в списке **Sub-Object** установите **Face** и выделите часть поверхности в нижней части конуса.



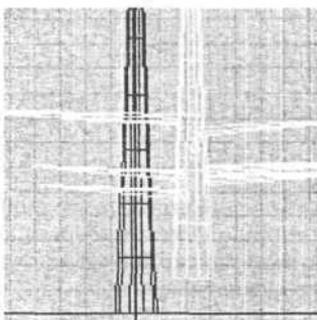
Кликните по надписи **Sub-Object**, чтобы деактивизировать команду и выберите конус самой длинной ветки. Для того, чтобы присоединить ветки к стволу в командной панели выберите вкладку **Create**, нажмите кнопку **Geometry** и в раскрывающемся списке выберите **Compound Object**. Нажмите кнопку **Scatter** и установите следующие значения:

Для **Sours Object Parameters**: **Duplicates**: 7.

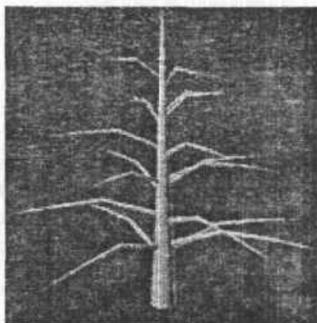
Для **Distribution Object Parameters**: выберите **Use Selected Faces Only** и **Distribute Using - Even**.

Нажмите кнопку **Pick Distribution Object** и кликните по конусу ствола. В результате получится голый ствол с ветками в нижней части. Если теперь немного подвинуть получившийся объект, вы увидите конус, который использовался для построения ствола. Теперь он не нужен и его можно удалить.

Для того, чтобы нарастить ветки выше, необходимо выполнить все предыдущие операции: выбрать **Face** посреди ствола и присоединить конус-ветку со средними параметрами, а затем проделать те же операции для верхушки ствола с последним самым маленьким конусом. После каждой операции не забывайте удалять предыдущие шаблонные объекты.



У вас должно получиться схематичное дерево без листьев. Можно ему немного добавить реалистичности. Используя команду **Modify/Edit Mesh** выделите ствол и часть веток и потяните их немного вниз. Назначьте дереву какой-нибудь коричневый материал.

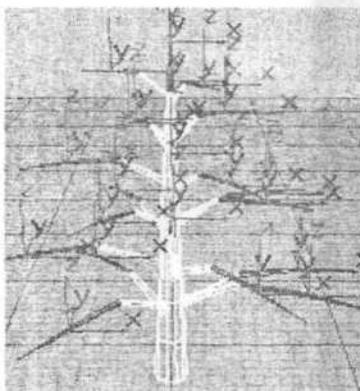


Теперь будем приделывать листья.

В командной панели во вкладке **Create** нажмите кнопку **Shapes** и используя инструмент **Line** нарисуйте небольшой ромбик. Выберите вкладку **Hierarchy** и нажав кнопку **Affect Pivot Only** перетащите центральную ось туда, где у листка должен быть черенок.

Для того, чтобы линии превратились в плоскую поверхность выберите вкладку **Edit Mesh** и нажмите кнопку **Mesh Select**. Затем в редакторе материалов выберите материал с зеленым цветом и присвойте его получившемуся листку.

Теперь, используя уже освоенный инструмент **Scatter**, раскидайте листья по всему дереву.



Выделите на стволе поверхности верхушки и веток, выберите листок и в **Scatter** задайте следующие параметры:

Для **Sours Object Parameters**: **Duplicates**: 1000.

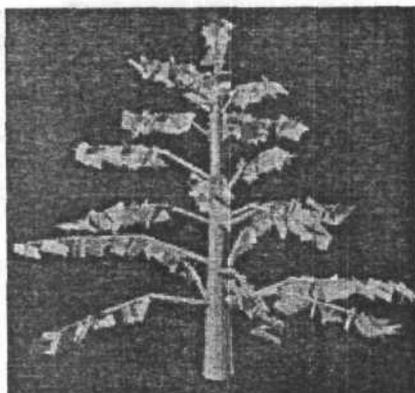
Для **Distribution Object Parameters**: выберите **Perpendicular** и **Use Selected Faces Only**; **Distribute Using** - **Random Faces**.

**Transforms**: X: 360,0 Y: 360,0 Z: 0,0.

**Display** 1,0% и выберите **Hide Distribution Object**.

Теперь нажмите **Pick Distribution Object** и кликните по стволу.

Чтобы получившаяся листва была немного ниже веток, выделите один ствол и немного приподнимите его относительно листьев. Можно еще немного закрутить ствол вокруг своей оси, чтобы ветки не были расположены симметрично друг под другом.



В коллекции растений  
 изобилует разнообразие  
 видов. Среди них можно  
 встретить как редкие  
 виды, так и широко  
 распространенные.  
 Для того чтобы  
 получить более  
 подробные сведения  
 о растениях, посетите  
 музей, где вы  
 сможете познакомиться  
 с ними поближе.



Важнейшим  
 является  
 изучение  
 их в  
 музее  
 и в  
 природе.  
 Для этого  
 необходимо  
 иметь  
 соответствующие  
 инструменты  
 и оборудование.  
 Кроме того,  
 следует  
 соблюдать  
 все правила  
 безопасности  
 при работе  
 с растениями.  
 Только так  
 можно  
 получить  
 достоверные  
 сведения  
 о них.

## Вопросы и ответы

**Как по умолчанию избавиться от сеток в окошках, в каждой новой сцене они восстанавливаются**

Запускаете 3ds max 7.0, делаете все настройки, и записываете файл под именем MAXSTART.MAX.

При каждом старте и при каждом **File ⇨ Reset 3ds max** просто считывает этот файл, и...

Правда, при **File ⇨ New...** так не происходит, но это придется пережить.

**Как при выделении объектов или добавить к уже выделенному, или наоборот снять выделение?**

Клавиша **Ctrl** — добавить к выделенным; клавиша **Alt** — снять выделение; клавиша **Shift** — инвертирование, то есть, уже выделенные станут невыделенными, а невыделенные наоборот...

Как это все работает? Под главным меню нажмите **Select Object** для включения режима выделения объектов, нажмите допустим **Ctrl** и щелкайте по объектам. Они выделяются, при этом старое выделение не снимается. А теперь отпустите **Ctrl**, нажмите **Alt** и опять щелкайте по объектам.

**Как разным частям одного объекта назначить различные материалы?**

Для 3ds max всех версий:

1. В редакторе материалов необходимо сделать составной материал (нажать кнопку справа от **Type**: и выбрать **Multi/Sub-Object**). В настройках материала указать необходимое количество под-материалов (там по умолчанию 6 или 10 в зависимости от версии 3ds max).

После этого надо настроить каждый суб-материал. Порядковым номером суб-материала является **Material ID** (см. п.3).

2. Назначить объекту этот составной материал (кнопочка **Assign Material to Selection**).

3. Применить к объекту модификатор **Modify ⇨ Edit Mesh**. Если не прижата кнопочка **Sub-Object**, прижимаете ее, в меню справа от **Sub-Object** выбираете **Faces**.

Для каждого суб-материала: выбрать все грани, которые должны иметь этот суб-материал, и в свитке **Edit Surface** в секции **Material**: в строке ввода **ID**: указать порядковый номер суб-материала, под которым он виден в **Material Editor**.

**Примечание:** В некоторых объектах, создаваемых плагинами, предусмотрена возможность удобного и быстрого присвоения разных материалов разным участкам объекта. Например, при создании дерева при помощи плагина **Tree Factory** по умолчанию веткам назначается подматериал с **ID=1**, листьям — с **ID=2**, плодам/цветам — с **ID=3**, и остается только создать материал **Multi/SubObject** с тремя подходящими подматериалами, и присвоить его дереву.

### Как сделать зеркальную поверхность в 3ds max 7.0?

В Максе зеркальная поверхность делается тремя способами: один для плоской поверхности, другой — для криволинейной, третий — общий (но медленный).

Для плоской поверхности: в общем случае делается составной материал с двумя под-материалами (в редакторе материалов нажать кнопку справа от слова **Type:**, выбрать **Multi/Sub Object**, назначить **Number of Materials=2**). Один суб-материал — любой стандартный (но не зеркальный), другой — нажимаете на кнопку с названием материала, в свитке **Maps** нажимаете кнопку **None** справа от галочки **Diffuse**, выбираете **Flat Mirror**, в верхнем ряду нажимаете кнопку **Go to Parent**, значение **Amount** для **Diffuse** оставить 100%. Далее работаете как обычно с составным материалом. Для примера, при выделенном отражающем объекте щелкаете кнопку **Assign Material to Selection**, закрываете **Material Editor**, присваиваете объекту модификатор **Edit Mesh**, прижимаете (если не прижата) кнопку **Sub-Object**, справа от нее выбираете **Face**.

Для «отражающих плоскостей» в свитке **Edit Surface** значение **ID**: ставите 2, для всех остальных плоскостей **ID=1**. Если этого не сделать, ничего не будет. Только не стоит на смежные некомпланарные плоскости (то есть, не находящиеся в одной плоскости) ставить один и тот же зеркальный суб-материал — могут вылезти глюки на картинке.

Чтобы кубик был целиком зеркальным, нужно каждой грани назначить отдельный зеркальный суб-материал. Поставьте значение **Number of Materials=6**, и сделайте 6 зеркальных материалов, для каждой грани отдельный.

Для криволинейной поверхности: у нужного материала в свитке **Maps** нажимаете кнопку **None** справа от галочки **Reflection**, выбираете **Reflect/Refract**, один щелчок по кнопке **Go to Parent**, попадем в настройки материала, и назначаем этот материал объекту. Далее — рендерить.

Третий способ: Raytrace-материал — самый медленный, но и дающий самые реалистичные результаты. В редакторе материалов выбирается тип материала **Raytrace** (кнопочка справа от **Type**). В простейшем случае требуют настройки два параметра: **Diffuse** и **Reflect** (степень отражения). Например, в списке **Basic Parameters** можно сделать образец цвета **Reflect** белым вместо черного.

Еще один вариант — у готового материала в свитке **Maps** карте **Diffuse** назначить **RayTrace** (нажать клавишу **None** справа от галочки **Diffuse** и выбрать **Raytrace**, нажать **Go to Parent**) стандартный (но не зеркальный), другой — нажимаете на кнопочку с названием материала, в свитке **Maps** нажимаете кнопочку **None** справа от галочки **Diffuse**, выбираете **Flat Mirror**. На **Diffuse** тоже потянет, но более правильно ставить **Flat Mirror** все-таки на **Reflection**. Тогда для «чистого» зеркала надо выставить следующие параметры материала:

"Ambient" = "Diffuse" = RGB(0, 0, 0);

"Specular" = RGB(255, 255, 255);

"Reflection map" = "100% Flat Mirror"

При этом яркость цвета **Specular** (значение **Value**) будет давать силу отражения (также как и процент на **Reflection map!**), а его цветовой тон (**Hue**) будет тонировать отражение. При просчете **Flat Mirror** отражение складывается с цветом и/или текстурой **Diffuse**. Причем складывается аддитивно. Поэтому то для «чистого» зеркала цвет **Diffuse** — черный.

Всякие частично отражающие материалы будут иметь некий цвет и/или текстуру **Diffuse** и процент **Reflection map** — 5-30%.

Иногда бывает очень нежелательно использовать модификатор **Edit Mesh** или преобразовывать объект к **Editable Mesh**. В этом случае к объекту применяются следующие модификаторы (в стеке они будут в обратном порядке): **Material** (с установкой параметра **ID**), **Mesh Select** (в режиме **Sub-Object** = **Polygon** отмечаются нужные фейсы, **Sub-Object** остается *нажатым*), **Material** (с установкой параметра **ID** для выделенных фейсов) **Mesh Select** (**Sub-Object** отжат).

Для «отражающих плоскостей» в свитке **Edit Surface** значение **ID**: ставите 2, для всех остальных плоскостей **ID**=1.

Я бы посоветовал первым движением отселектить все фейсы и присвоить им **ID**=1, а затем выделить только отражающие фейсы и присвоить им **ID**=2. Это будет гарантировать, что у объекта не осталось фейсов с **ID**=2, который был присвоен Максом при создании объекта.

Чтобы что-то в зеркале отразилось, нужно:

- а. чтобы это что-то существовало...
- б. чтобы камера могла видеть отражение (угол падения = углу отражения)...
- в. чтобы сторона объекта, обращенная к зеркалу была освещена.

**Настройка Glow (свечения) в 3ds max 7.0. Канал объекта и канал материала. Отражение Glow-эффекта на зеркальных объектах. Объект (с Glow-эффектом), закрытый полупрозрачным объектом**

**Glow** — это эффект для VideoPost. Он создает свечение вокруг объекта (как светится неоновая надпись поздним вечером). Работает он через каналы **Channel** объекта или материала. Для работы эффекта соответственно необходимо установить канал материала **Material Effect Channel** (в редакторе **Material Editor** кнопка с цифрой, седьмая слева в горизонтальном ряду) и/или канала объекта (правая кнопка мыши на объекте ⇨ **Properties** ⇨ **G-buffer Object Channel**). Поскольку разные материалы/объекты могут иметь разное значение **номер канала**, то и настройки свечения у них могут быть разные. Очень удобно.

1. Установить канал объекта (или материала) отличным от 0 (значение 0 стоит по умолчанию):

- ◆ для объекта: щелчок на объекте правой клавишей, **Properties**, и меняете **Object ID**.
- ◆ для материала: в **Material Editor** под слотами материалов кнопочка **Material Effects Channel**, жмете и держите, потом назначаете...

2. Из главного меню вызвать **Rendering** ⇨ **Video Post**; создать событие **Просчет сцены** (кнопочка **Add Scene Event**); не выделяя события **Просчет сцены**, создать событие **Обработка изображения** (кнопочка **Add Image Filter Event**), в выпадающем меню выбрать **Lens Effects Glow** (или тот **Glow**, который у вас имеется), кнопка **Setup** — настройка эффекта; и под конец добавить событие **Запись в файл** (кнопочка **Add Image Output Event**).

3. Настройка эффекта. Там все достаточно прозрачно. Под окном **Preview** кнопка **Preview** разрешает показ «preview».

Прижатая кнопка **VP Queue** заставляет показывать в «preview» реальную сцену из очереди **VideoPost** (в противном случае используется тестовая картинка 3ds max'a). Не забудьте поставить в **Object ID** или **Effects ID** нужный номер канала (то число, которое вы назначали в первом пункте).

4. Для получения картинки (то есть, для обсчета сцены) нужно зайти в **Video Post**, и там нажать кнопку **Execute Sequence**.

Через обычный **Render** ничего не получится.

Все настройки анимируются. Ниже — некоторые основные установки **Glow**.

Закладка **Properties**:

- ◆ **Object ID** — тот, который был установлен в п. 1
- ◆ **Material ID** — канал эффектов материала (устанавливается в редакторе материалов **Material Editor** кнопкой **Material Effects Channel**).

Закладка **Preferences**:

- ◆ **Size** — размер эффекта свечения.
- ◆ **Color: Gradient** — свечение будет генерироваться в соответствии с установками, сделанными на странице **Gradients**;
- ◆ **Color: Pixel** — свечение будет рассчитываться, исходя из цвета пикселя объекта;
- ◆ **Color: User** — свечение будет выводиться на основе указанного цвета.
- ◆ **Intensity** — плотность эффекта.

В 3ds max 7.0 в стандартной поставке есть plug-in **Lens Effects** (файл **APOLLO.FLT**). Именно он делает **Glow**. Есть еще и другие plug-in'ы, реализующие **Glow**-эффект: **Glow+**, **RealLensFlare**.

В современных RayTrace'рах для 3ds max есть возможность включать присвоение отражениям такого же **Material ID**, какой используется в материале отражаемого объекта. Другими словами, отражение неоновой надписи в зеркале тоже будет светиться!

Можно сделать нестандартно, и при некоторых условиях это будет работать, то есть, объект, закрытый полупрозрачным объектом, будет обчитываться в **VideoPost**.

В параметрах **Glow** поставьте галочку против **Alpha** и задайте числовой параметр таким, чтобы он был больше **Alpha** полупрозрачного объекта, но меньше **Alpha** внутреннего объекта. Например для непрозрачного (**Alpha=255**) шарика в полупрозрачном (**Opacity=50, Alpha=192**) кубике параметр **Alpha** в **Glow** будет 200.

Альфу можно померить так: отрендерить сцену, нажать в *Toolbag*'е отрендеренной картинке кнопку **Display Alpha Channel** и поводить по картинке указателем мыши с нажатой правой клавишей.

### Как отрендерить объект с тенью, отбрасываемой на невидимый объект?

Нужно получить картинку с Alpha-каналом, с объектом и тенью от него (по Alpha-каналу) на якобы имеющуюся стенку за объектом, при этом на картинке этой стенки не должно быть.

В качестве «стенки» используйте **Box** или **Patch** с материалом **Matte/Shadow**. В редакторе материалов параметры:

- ◆ **Matte: Opaque Alpha** (непрозрачность по альфе) отключить.
- ◆ **Shadow: Receive Shadows** (принимать тени) включить, **Affect Alpha** (тени по альфе) включить.

Остальные параметры — по необходимости.

Результат просчета лучше записывать в формате, поддерживающем Alpha-канал. Например: TIFF, TARGA, PSD.

### Почему некоторые плоскости прорисовываются, а некоторые — нет?

Одна сторона каждой плоскости объекта имеет так называемую «нормаль» (**Normal**). Нормаль — это перпендикулярный вектор, расположенный в центре плоскости. Он используется для определения того, какая сторона плоскости отражает свет. Противоположная сторона плоскости свет не отражает, и не видна при просчете изображения.

Чтобы увидеть нормаль, можно присвоить объекту модификатор **Edit Mesh**, в качестве редактируемых подобъектов (**Sub-Object**) выбрать «плоскости» (**Face**), выделить любую плоскость объекта и в секции **Normals**: отметить **Show Normals**. Нормали будут видны на экране. Другой способ сделать плоскость прорисовываемой всегда — присвоить ей двухсторонний материал (галочка **2-Sided** на свитке **Basic Parameters**).

### А как в *3ds max* сделать нормальные размытые тени?

Использовать плаг **SoftSpot**, а если его нет — смоделировать самому. Принцип **SoftSpot** простой — это 10 ламп по окружности.

Окружность ненулевого диаметра. Лампы идентичны. Тени — то, что надо.

Увеличивайте количество сэмплов для маповых теней и отключите параметр **Absolute Map Bias**.

Для **Shadow Map** есть параметры:

- ◆ **Size** — отвечает за размер просчитываемой карты теней в пикселях;
- ◆ **Smp Range** — количество смешиваемых пикселей, влияет на степень «размазанности» края тени.

Поставьте первый параметр равный 1024, и второй 10-20 — понравится.

### **Почему объекты, которые видно сквозь Raytrace-материал с преломлением, не обрабатываются Anti-Aliasing'ом?**

Anti-Aliasing в Raytrace-материале по умолчанию выключен. Ищите настройки в **Material Editor**.

### **Как сделать так, чтобы стекло было не совсем прозрачным и в то же время отражало объекты?**

В **Material Editor** используйте материал **Mix** — смешанная карта. На один слот выставляется материал стекла, на другой **Flat Mirror**. Далее настройка параметров отражения карты **Flat Mirror**.

Можно обойтись без материала **Mix**, для этого достаточно ослабить тот же **Flat Mirror** путем уменьшения его **Amount**.

### **Как отрендерить сцену так, чтобы задний план в картинках стал прозрачным?**

Самый простой способ — результат просчета записывать в формате, поддерживающем Alpha-канал, например, TIFF, TARGA, PSD.

При этом задний план будет прозрачен там, где картинка заполнена цветом, указанным в главном меню ⇨ **Rendering** ⇨ **Environment** ⇨ **Background** ⇨ **Color**.

Кстати, основано это на эффекте «3ds max считает задний фон абсолютно прозрачным». На картинке в том месте, где видно задний фон (а цвет фона задается в главном меню **Rendering** ⇨ **Environment** ⇨ **Common Parameters** ⇨ **Background**: ⇨ **Color**:) — там прозрачно. А где не видно — там не прозрачно, с градациями.

### **Как рассчитать размеры текстурной карты?**

Далее подразумевается работа с модификатором **UVW Mapping**.

Для точного (без искажений) наложения текстурной карты необходимо соблюсти, как минимум, условие сохранения соотношения сторон самой текстуры (битмапа) и текстурной карты (**Gizmo**).

Для планарной карты это означает, что соотношение ширины и высоты **Gizmo** должны быть равно соотношению ширины и высоты накладываемого битмапа. Кнопка **Bitmap Fit** поможет подогнать размер **Gizmo** по существующему битмэпу.

Обратный ход — создание битмапа по **Gizmo**. Накладываем модификатор **UVW Mapping** и подгоняем **Gizmo** под объект без применения неуниформного масштабирования **Gizmo**. Из полей **Length** и **Width** получаем размеры для битмэпа, точнее, соотношение сторон. Следует заметить, что 3ds max 7.0, по каким-то своим соображениям, делает размер **Gizmo** на 0.1% больше габаритного размера объекта.

Для сферического мэппинга следует творчески применять формулу длины окружности. Соотношение сторон для битмэпа 2:1. Также необходимо учитывать искажения текстурной карты у полюсов сферы.

Выбор соотношения сторон для остальных способов мэппинга вытекает из вышеизложенного.

### Как создать точную планарную текстуру для объекта сложной формы?

Пусть у нас имеется почти плоский объект сложной формы, на который необходимо наложить планарную карту. Допустим, объект — самолет, и нужно очень точно разрисовать его крыло.

Ориентация модели — естественная: вид сверху — плоскость крыла.

Скрываем (**Hide**) все остальные объекты сцены, чтобы ничего не было в кадре. Выбираем вид **Top** и устанавливаем такой масштаб изображения, чтобы объект занимал как можно большую часть окна (без выхода за его границы). Рендерим вид **Top** (полезно на время отключить **Anti-Aliasing** в диалоге рендера, будет лучше).

Сохраняем **Targa** с установленным флажком **Alpha Split** (в диалоге сохранения — кнопка **Setup** после указания имени и формата записываемого файла).

В итоге получаем две картинки: нормальную и силуэтную (то есть, Alpha-канал). В любом растровом редакторе обрезаем лишние черные поля вокруг белого силуэта крыла (не фигурно, а просто прямоугольником, функция **Crop** обычно.). Далее — рисуем на белой части картинку все, что должно быть видно на крыле. Сохраняем полученную карту.

В Максе создаем материал с нарисованной текстурной картой, присваиваем его объекту. Накладываем модификатор **UVW Mapping**.

Обычно дополнительной регулировки не требуется.

Для повышения четкости рисунка можно уменьшить параметр карты **Blur** (в редакторе материалов) до 0.01.

### Текстура Земли. Вид из космоса. Как?

Текстура Земли, как таковая, делается элементарно просто: файл **EARTH.TGA** из стандартной поставки Макса 1.x специально изготовлена для сферического мэппинга. **EARTH.TGA** на цвет **Diffuse**, плюс небольшой **Bump** той же текстурой. Другой вопрос, как сделать реалистичную картину вида из космоса планеты Земля.

Очевидно, что помимо сферы с вышеозначенным материалом необходимо смоделировать атмосферу и поставить правильное освещение и (может быть) дополнительные эффекты.

Атмосферу можно смоделировать несколькими сферами с различными материалами. Я исходил из физического строения атмосферы Земли: несколько слоев облаков (тропосфера) и стратосфера.

Самый нижний слой: редкая, но довольно плотная облачность с хорошо заметными спиральными структурами (подойдет **CLOUDS.TGA** с прозрачностью).

Следующий — плотная атмосфера: полупрозрачный материал с плавным шумом большого размера на **Opacity**, включенным в свитке **Extended Parameters Falloff: = Out**, отмеченным там же **Type: = Filter**.

Следующий — стратосфера: аналогичный полупрозрачный материал с **Opacity** и **Falloff; Refract Map/RayTrace IOR: = 1**; самосветимость около 20.

Радиусы сфер: 6500 — поверхность земли, 6505 — облака, 6520 — плотная атмосфера, 6700 — стратосфера. Получается довольно реалистично. Можно еще добавить карту блеска от воды.

### Как на затекстурированный объект положить картинку, чтобы она была отдельно от всей текстуры объекта. И можно ли так положить несколько картинок?

Присваиваем объекту составной материал **Multi/Sub-Object**. Первый материал настраиваем стандартно (то есть, как вам надо), а у второго в **Diffuse** ставим **Bitmap** (нажать кнопку **Type:** под слотами материалов и в появившемся окошке пункт **Bitmap**). Далее для гарантии применяем к подопытному объекту модификатор **Material ID** и настраиваем на первый материал (например цифру 1). Теперь накладываем **UVW Mapping** для главной текстуры.

Далее применяем модификатор **Mesh Select** и выделяем те плоскости, на которые нужно наложить другую картинку. Не отжимая кнопку

**Sub-Object**, накладываем на выделенные плоскости модификатор **Material ID**, выставляем им соответствующий номер материала (например 2), и делаем еще один, отдельный **UVW Mapping**.

И так по всем под-материалам (если их несколько). При этом каждый последующий **Mesh Select** накладывается просто поверх предыдущего (то есть, у предыдущего **Mesh Select** кнопку **Sub-Object** отжимать не надо).

#### Как сделать так, чтобы у материала в анимации менялся **Bitmap**?

1. Использовать **IFL**.
2. Использовать в качестве картинки **AVI**, **MOV**.
3. Использовать тип материала **Blend**.
4. Использовать тип материала **Mix**.

Хочу присвоить объекту свою текстуру, в **Material Editor** выбираю **Bitmap**, прописываю имя файла, все в списокке появляется, однако с объектом связать не могу: кнопки нужные не активны. Почему?

Сам **Bitmap** не может быть присвоен к объекту. **Bitmap** должен использоваться в материале, а уже сам материал будет присвоен объекту. В самом простейшем случае у материала **Standard**: нажми кнопку **None** справа от галочки **Diffuse** в свитке **Maps**, и вот там выбирай **Bitmap**. Чтобы выйти из режима настроек **Bitmap**, просто нажми кнопку **Go to Parent**. Что бы присвоить материал объекту, при выделенном на сцене объекте нажми кнопку **Assign Material to Selection**. И еще, рекомендую присвоить объекту и настроить модификатор **UVW Map**. Это чтобы **Bitmap** располагался на объекте правильно.

#### Как поставить в картинку в **Background** обычный **BMP**-файл, чтобы он располагался на заднем фоне?

Возьмите любой ненужный материал и задайте в нем для **Diffuse** — **Bitmap**. В параметрах **Bitmap** указываете имя вашего файла (нажмите и пролистайте в **FileDialog**). По умолчанию **Bitmap** в фоне растянется на сферическое окружение. Вращая камеру, вы будете видеть часть виртуальной сферы, на которую будет натянута **Bitmap**. Чтобы этого избежать, в параметрах материала в свитке **Coordinates** отмечаете **Environ**, а правее в меню **Mapping**: выставляете **Screen**.

Нажмите **Go to Parent**, и обзовите свой материал как-нибудь поинтереснее, например: **SceneBGMap** (это в меню под слотами материалов).

Теперь заходите в **Rendering ⇨ Environment**, и нажимаете **Environment Map**. В появившемся **Map Browser** выбираете в секции **Browser From**: пункт **Material Editor**, находите свой материал **SceneBGMap**, выбираете его, и в появившемся окне выбираете **Instance**.

При просчете сцены картинка будет на заднем плане!

**Есть текстура, наложенная на объект (сфера например). Как сделать так, чтобы некоторые части этой текстуры были зеркальными? То есть Reflect**

Рисуете еще одну грейскейловую текстуру с характеристикой отражения. То есть, те места, которые должны отражать — белые, не должны отражать — черные, слегка отражающие — серые. (Удобно делать в Фотошопе, положив на нижний слой твою диффузную текстуру, а потом сохранять отдельно, убрав видимость нижнего слоя). Потом на **Reflection** ставите текстуру **Mask**, для которой устанавливаете в качестве маски вышеописанную картинку, а в качестве **Map** — отражающую текстуру (**Flat Mirror** или **Reflect/Refract**, или **Raytrace**).

**Как сделать так, чтобы при моделировании текстуры на объектах в окнах проекций рисовались?**

Для этого в **Material Editor** надо нажать кнопку **Show Map in Viewport**. Только учтите, она разблокирована только тогда, когда под слотами с материалами на кнопочке **Type**: написано что-нибудь отличное от **Standart**. Другими словами, если в стандартном материале на панели **Maps** нажать кнопку **Diffuse**, выбрать **Bitmap** и назначить картинку, то кнопка разблокируется. И еще, эта кнопка дает возможность показывать только одну текстуру для одного материала. Одновременно включить ее например для **Ambient** и для **Diffuse** не получится.

**Что такое pivot? Где он настраивается?**

**Pivot** — это воображаемая точка объекта, являющаяся центром геометрической трансформации объекта (вращения и масштабирования).

Перемещение **Pivot**'а соответственно изменяет координаты центра вращения/масштабирования объекта.

Закладка **Hierarchy** командной панели (справа), кнопка режима — **Pivot**. Кнопка **Affect Pivot Only** позволяет перемещать и вращать **Pivot**, не трогая самого объекта. Кнопка **Affect Object Only** позволяет перемещать/вращать/масштабировать объект, не трогая **Pivot**. Кнопка **Reset Pivot** возвращает **Pivot** к первоначальному виду, обнуляя все перемещения/вращения/масштабирования. При работе с **Pivot** работает функция **Align** (кнопка в **ToolBar**'е, под главным меню).

### Почему Boolean портит объекты? Что делать?

Есть мнение, что многие ошибки в 3ds max 7.0 происходят от не совсем корректного взлома программы (убрана проверка HardLock-ключа на LPT-порту). Но это не так.

Перед **Boolean** нужно запомнить текущее состояние сцены — из главного меню **Edit ⇔ Hold** (запомнить). Если **Boolean** прошел хорошо — никаких вопросов. Если **Boolean** прошел плохо (объекты исчезли, некоторые точки сместились, или вообще никакого результата), то делаем в главном меню **Edit ⇔ Fetch** (откат), либо **Undo** нужное количество раз.

1. Посмотрите на закладке **Tools ⇔ Measure** параметр **Volume**:. Если он без звездочки в конце, значит можно смело переходить к следующему пункту. Если там стоит звездочка, объем внутри объекта через какую-нибудь дырочку соединяется с объемом сцены, у объекта незамкнутая оболочка.

В качестве решения: для автоматического закрытия всех отверстий в объекте (если нет желания делать это вручную) применяйте модификатор **Modify ⇔ Cap Holes**. Вручную можно создавать новые плоскости в модификаторе **Edit Mesh ⇔ Sub-Object = Face ⇔ Build Face**.

Еще раз проверяем объемы. Если ошибка не исправилась, придется все дырки закрывать вручную, автоматика не справляется.

2. Применяем к объекту модификатор **Modify ⇔ STL-Check**. Если модификатор сообщил, что есть ошибки, исправляем ручками.

3. Желательно, чтобы все плоскости объекта были сориентированы правильно. Речь идет о нормалях, они должны смотреть наружу объекта. Очень часто 3ds max 7.0 переворачивает часть плоскостей нормалью внутрь, что приводит к плохому результату.

В качестве решения: или позволить программе самой все откорректировать (модификатор **Normal**, но он часто ошибается), или сделать все самостоятельно (модификатор **Edit Mesh**, включить режим **Sub-Object = Face**, и, выделяя разные плоскости, правильно их ориентировать кнопочкой **Flip** в свитке **Edit Surface**).

4. В системе **Boolean** есть еще одно хорошее требование: обрабатываемые объекты не должны сами с собой пересекаться. Примером «самопересечения» может послужить резиновое кольцо (бублик, или надутое колесо автомобиля), скрученное так, чтобы получилась цифра «8». В центре такой фигуры и будет «самопересечение». Для 3ds max 7.0 такой случай — губительный.

Переделайте объект заново, избегая такой ошибки.

5. Порой 3ds max спотыкается на том, что два объекта подогнаны друг к другу очень точно, в результате чего один объект «вклинивается» в объем другого объекта на очень маленькое расстояние. Из-за такого маленького «вклинивания» у 3ds max клинит мозговой отдел, и он выдает неправильный результат.

Слегка сдвиньте объекты относительно друг друга.

6. Не рекомендуется обрабатывать объекты, которые значительно различаются в размерах. Другими словами, если вы попытаетесь вычлесть из большого несложного объекта (например из **Box**) объект поменьше, с очень мелкой образующей структурой (например, **Text**), то результат вас безусловно разочарует.

Если же вы укажете для **Box** значения **Length Segs: = 20, Width Segs: = 20**, то все получится совсем иначе.

В качестве решения: разбейте большие плоскости объектов на более мелкие части. Для этого можно применять модификатор **Tessellate** с настройками **Operate On: = Polygon**, режим **Edge, Tension: = 0**. А для укрупнения плоскостей маленьких объектов можно применять модификатор **Optimize** с параметрами: **Smooth Boundaries = off; Material Boundaries = off; Bias = 0; Face Thresh = 0.1** (или меньше).

7. Так же рекомендуется перевести объекты в вид **Editable Mesh**.

8. Кстати, иногда можно попробовать применить **Boolean** ко второму объекту, а операцию задать как **Substraction B-A** (если вам, конечно, нужен именно **Substraction**).

### Как, вместо таскания объекта по экрану, в 3ds max точно задать его координаты?

В верхнем ряду квадратных кнопок (под главным меню) нажать кнопку, соответствующую нужному воздействию на объект (**Select and Move, Select and Rotate** или **Select and Scale**), а потом щелкнуть по этой же кнопке правой клавишей мыши. И можно указывать точные координаты объекту (точке, ребру, линии, плоскости).

### Как заставить 3ds max создавать примитивы в плоскости, определенной тремя произвольными точками?

Вот отличная методика:

1. Создаем треугольник (**Line ⇔ Keyboard Entry**), каждая вершина которого — нужная нам точка.

2. Применяем к нему модификатор **Modify ⇔ Edit Mesh**.

3. Создаем **Create ⇨ Helpers ⇨ Grid**, щелчок правой клавишей мышки на созданном Grid'e, пункт **Activate Grid** — активизируем ее.

4. Из главного меню **Tools ⇨ Align Normals** щелкнуть в геометрическом центре Grid'a, затем щелчок в геометрическом центре нарисованного треугольника. Теперь ориентация Grid точно копирует ориентацию треугольника.

5. В группе кнопок под главным меню в списке **Reference Coordinate System** выбрать **Grid**.

6. Создаем примитивы.

7. Удаляем все лишнее и возвращаемся к **Home grid** (правая кнопка мыши по созданному Grid'у и пункт **Activate Home Grid**).

#### Как сделать видимыми только ребра/вершины объекта?

**Только ребра:** присвоить объекту материал с отмеченным пунктом **Wire** — это чтобы у одного объекта. А чтобы сразу у всех объектов в сцене — в настройках **Rendering'a** отметить галочку **Force Wireframe**.

**Только вершины:** на закладке **Modify** кнопочка **Lattice** (если нет такой, то кнопочка **More**, а потом пункт **Lattice**), в свитке **Geometry** отметить **Junctions Only**.

#### Как в 3ds max 7.0 задать объекту вращение относительно фиксированной точки?

Есть три способа.

**Первый:** настроить pivot.

**Второй:** работа с Point'ом. Создаете **Create ⇨ Helper ⇨ Point**, олицетворяющий перемещаемый центр объекта. Далее под главным меню давите на кнопку **Use Pivot Point Center** (она первая справа от менюшки **Reference Coordinate System**), и не отпускаете некоторое время, и в появившемся наборе кнопок выбираете **Use Transform Coordinate Center**. Затем в меню **Reference Coordinate System** выбираете **Pick** и щелкаете на созданный Point-объект. Можете вращать и масштабировать. При переключении режима трансформации проверяйте, включено ли **Use Transform Coordinate Center**.

**Третий:** работа со **Snap Toggle**.

В нижней правом углу щелкнуть правой клавишей мыши по кнопке **3D Snap Toggle** (она же **2.5D Snap Toggle**, или **2D Snap Toggle**). Появится небольшое окошко **Grid and Snap Settings** (а еще можно в главном меню **View ⇨ Grid and Snap Settings**), в котором можно указать, какие геометрические элементы могут использоваться в качестве центров враще-

ния и масштабирования объекта. А чтобы включить этот режим, нужно надавить описанную кнопочку. Соответственно, если вы в окошке **Grid and Snap Settings** отметите **Vertex**, то сможете двигать/вращать/масштабировать объект относительно любой вершины любого объекта на сцене.

### Как создать камеру так, чтобы вид из нее точь-в-точь повторял вид **Perspective**?

Делаете камеру типа **Free**. Прямо в окне перспективы.

Затем **Align to View** (кнопка в группе **Align**) по оси **Z**.

Во viewport'e камеры подгоняете **Dolly**.

Использование: создать новую камеру (если ее нет), выделить эту самую камеру, правой клавишей мыши щелкнуть в видовом окне **Perspective**, с которым требуется согласовать камеру (чтобы сделать это текущим), а потом в главном меню **Views** ⇨ **Match Camera to View**.

Можно изменить последовательность действий: сделать вид **Perspective** активным, под главным меню нажать кнопочку **Select by Name** и выбрать свою камеру (она выделится), и уже после этого в главном меню **Views** ⇨ **Match Camera to View**.

### Привязываю разные вершины объекта к другим объектам, а потом... любой модификатор работает неправильно

Другими словами, вы создаете прямоугольник **Box01**, создаете два объекта **Dummy** (**Create** ⇨ **Helpers** ⇨ **Dummy**), так?

Далее вы присваиваете объекту **Box01** модификатор **Mesh Select**, выделяете несколько вершин (**Vertex**), сразу присваиваете им модификатор **Linked XForm**, жмете **Pick Control Object**, щелкаете на **Dummy01**? Потом сразу, ничего не изменяя, опять **Mesh Select**, выделяете другие вершины, опять **Linked XForm**, **Pick Control Object** и щелчок на **Dummy02**? Теперь, если перемещать **Dummy01** и **Dummy02**, то прилинкованные к ним подобъекты тоже будут двигаться.

Но есть одно большое «но»! Любой модификатор, изменяющей геометрию объекта (например, **Taper**), начинает действовать на подобъекты, выделенные в последнем модификаторе **Mesh Select**.

Оно и правильно, ведь режим **Sub-Object** у модификаторов **Mesh Select** никто и не выключал. То есть, если **Sub-Object** выключить, то все наладится.

И сразу разладится в другом месте. Поскольку подобъекты перестанут повторять движение за **Dummy**-объектами. Такова природа некоторых модификаторов в 3ds max 7.0.

Что же делать? А решение очень простое! Надо применить к **Box01** еще один модификатор **Mesh Select**, и сразу отжать кнопку **Sub-Object**. Получится как бы «пустой модификатор». Все. Никаких выделенных вершин/ребер/плоскостей.

А значит и остальные модификаторы (например, **Tapet**) будут воздействовать на весь объект.

### Есть способы сделать волосы без применения plug-in'ов?

Сплайнами: **Create** ⇨ **Shapes** ⇨ **Line**. Сделать их разновидностей пять-шесть, **Renderable** отметить, и распределить по поверхности через **Create** ⇨ **Geometry** ⇨ в менюшке выбрать **Compound Objects** ⇨ **Scatter**. Цвет, толщину, вариации распределения сплайнов подбирать по вкусу.

### Модификатор STL-Check

**Open Edge** — открытые ребра. Другими словами ребра, которые принадлежат только одной плоскости. Например, если у кубика **Box** удалить любое ребро — то получится 4 открытых ребра.

**Double Face** — двойные плоскости. Случай, когда есть две одинаковые плоскости, построенные на одних и тех же точках. Например, если у кубика **Box** создать (**Create**) еще четыре точки, дублирующие уже существующие, построить (**Build Face**) на этих точках две новые плоскости, дублирующие уже существующие, и объединить (**Weld**) новые точки со старыми, то получится 4 двойных плоскости.

**Spike** — плоскость-клин. Плоскость имеет только одно общее ребро с объектом, которому принадлежит. Например если у кубика **Box** построить одну плоскость на уже существующих точках — то получится 1 плоскость-клин.

**Multiple Edge** — множественное ребро. Случай, когда одно ребро принадлежит сразу трем (и более) плоскостям. Например если у кубика **Box** построить одну плоскость на уже существующих точках — то получится 5 множественных ребер.

STL-Check очень, кстати, удобная штука для проверки геометрии объекта.

### Включил Snapping и при включении режима «двигать только по X» («Restrict to X») объект свободно перемещается в любом направлении

В нижней части экрана есть кнопка **3D Snap Toggle**, щелкните по ней правой клавишей, и в появившемся окошке **Grid and Snap Settings** на закладке **Options** в секции **Translation** отметьте галочку **Use Axis Constraints**.

**Можно ли скопировать модификатор от одного объекта к другому объекту?**

Выделяете тот объект, от которого надо скопировать модификатор(ы). На закладке **Modify** в свитке **Modifier Stack** нажимаете кнопку **Edit Stack**, выделяете нужные модификаторы и кнопка **Copy**. Затем выделяете второй объект, **Edit Stack** и, выделив модификатор, после которого должны появиться копируемые модификаторы, и нажимаете **Paste**.

**Надо сконвертировать обычный Spline в NURBS-кривую. Как это можно сделать?**

Есть несколько способов.

**Первый:** создать NURBS-кривую, и присоединить к ней все **Spline**, а затем или удалить саму кривую, или сделать **Detach** для самих **Spline**.

**Второй:** сконвертировать **Spline** в **Editable Spline**, а затем этот **Spline** сконвертировать в NURBS-кривую (все это делается через правый щелчок по кнопке **Edit Stack** на закладке **Modify**).

**Третий:** существует небольшой скрипт-файл, проводящий такое конвертирование в автоматическом режиме.

**Расскажите про Snapping**

Прилипалка включается/выключается клавишкой **S** или кнопкой в панели инструментов, расположенной под движком номера кадра.

Если подольше придержать кнопку включения Snapping'a, то появится менюшка для выбора режима работы Snapping'a:

- ◆ **2D Snap** — реагирует только на те элементы, которые находятся в плоскости видового окна, курсор в глубину не перемещается;
- ◆ **2.5D Snap** — реагирует на все включенные элементы, вне зависимости их удаленности от плоскости видового окна, курсор в глубину не перемещается;
- ◆ **3D Snap** — трехмерное прилипание, реакция полностью аналогична **2.5 Snap**, но курсор в глубину перемещается.

Если по кнопке щелкнуть правой кнопкой мыши, то появится диалог **Grid and Snap Settings** (или можно в главном меню **View ⇄ Grid and Snap Settings**). В нем можно настроить набор элементов, к которым будет осуществляться прилипание (grid, vertex, edge).

**Что можно делать с Patch?**

С патчами можно делать все что угодно.

Есть следующая методика работы (в двух словах). 3D-сплайнами рисуем опорные линии моделируемого объекта. Причем vertex'ы располагаем так, чтобы между ними впоследствии мог лечь Patch (Quad или Tri). После того, как каркас из сплайнов сделан, создаем Create ⇨ Patch Grids ⇨ QuadPatch с размером 1x1, применяем модификатор Edit Patch и в режиме 3D Snap Toggle «приклеиваем» контрольные вершины Patch к вершинам каркаса с регулировкой тангент. Далее выделяем одно из ребер (Edge) патча и достраиваем еще кусок и настраиваем его. И так, пока не будет обтянут весь каркас. В местах прилегания двух или более кусков контрольные вершины необходимо объединять (операция Weld).

Можно с Patch работать и без каркаса. Только это сложнее...

### Как объединить две точки двух объектов?

Для этого сначала нужно объединить сами объекты через Attach (выделить один из объектов, на закладке Modify ⇨ кнопочка More.. ⇨ пункт Edit Mesh, отжать кнопочку Sub-Object и нажать на кнопочку Attach, шелкнуть по второму объекту).

### Можно ли очистить стек модификаторов на закладке Modify от всех этих модификаторов UVW Mapping?

Параметры маппинга являются атрибутами самого объекта/суб-объекта.

В случае NURBS — маппинг является атрибутом surface sub-object.

В случае с Mesh — каналы маппинга хранятся вместе с мешем в виде практически таких же Mesh. В случае с Patch маппинг является атрибутом патчевого прямоугольника/треугольника.

Но в любом случае — модификатор маппинга лишь изменяет и настраивает маппинг, а не хранит его.

Соответственно делаете объекту Collapse Stack и все эти Mapping'и исчезают, а их настройки — остаются.

### Что такое Editable Mesh? Как объект привести к такому виду? И зачем это делается?

Дело в том, что в объекты в файлах в формате .MAX записываются не в виде положения точек объектов, а в виде действия над элементом объекта. При считывании файла эти действия обрабатываются соответствующими plug-in'ами, и в результате получается нужный объект...

Другими словами, в файле хранятся параметры прямоугольника, и параметры модификатора Bend (изогнуть) с указанием, на сколько именно градусов пользователь этот прямоугольник изогнул.

Поэтому, прочитав такой файл, можно указать другой модификатору угол, и получить другой результат.

Когда 3ds max считывает файл со сценой, он проделывает все модификации, которые применялись к объекту перед записью. Допустим, во вторник вы создали объект, применили к нему какой-нибудь сложный модификатор (который работал над объектом 2.5 часа), полюбовались на результат, сохранили сцену и ушли домой.

Когда вы в среду придете и загрузите вчерашнюю запись, то 3ds max еще 2.5 часа будет «вспоминать», как именно должен выглядеть объект после применения того страшного модификатора.

Есть команда **Editable Mesh**. Ее работа объясняется просто: к объекту применяются все модификаторы, после чего в памяти хранится уже сам результат работы модификаторов. То есть, не мегабайты информации «какой кубик на какой угол был изогнут», а 100 килобайт информации «точка А находится там-то, точка В — там-то, точка С там-то...»

Функция **Convert to Editable Mesh** находится на закладке **Modify**. Выделяете нужный объект, в свитке **Modifier Stack** есть квадратная кнопка **Edit Stack**, нажимаете ее правой клавишей мыши, появляется или меню (там еще есть пункт **Editable Mesh**, выбираете его...), или окно **Edit Modifier Stack** (там есть кнопочка **Collapse Stack**, жмете ее).

Эта же функция находится на закладке **Utilities**, называется **Collapse**. Результат ее действия абсолютно аналогичен. Только тут надо нажать кнопочку **Collapse Selected**.

**Имеется объект с острыми гранями (куб или цилиндр), как сгладить определенные грани (закруглить)**

Примените к объекту модификатор **Edit Mesh**, после чего выделяете нужную грань объекта (в режиме **Face**, либо в режиме **Edge**), делаете **Extrude**, после чего скалливированием уменьшаете **extrud** нутое. Ручной способ требует длительных приседаний, но зато работает точнее, чем **MeshSmooth**.

**Не могу разобраться с Connect. Как он работает?**

Создайте два шара, в модификаторе **Edit Mesh** выделите у каждого шара по несколько случайных точек и удалите их. После этого примените **Connect**.

**Как размножить объекты по нужному мне пути? Например, хочу цепь намотать на цилиндр. Что мне, каждое звено вручную ставить теперь?!**

Запустите свой объект. А потом просто сделайте **Snapshot** (в главном меню **Tools** ⇨ **Snapshot**), отметив в его настройках **Range** и указав некоторый настройки. Что там для чего — не сложно разобраться путем эксперимента.

**Есть бокс почти плоский. Надо этот бокс погнуть в форме бублика. Я применяю модификатор Bend, а он не гнется! Как сделать?**

Дело в том, что у вашего объекта очень мало вершин (*vertex*) на объекте. А именно вершины являются точками преломления ребер (*edge*) объекта. Они как шарниры, позволяют двум ребрам, соединенным этой вершиной, крутиться относительно друг друга.

Другими словами, простую палку вы не согнете, а вот две палочки, соединенные шарниром — запросто.

Теперь об изгибании **Box**'а. Присмотритесь к нему внимательно. Для удобства шелкните по нему правой клавишей мыши ⇨ **Properties...** ⇨ **Display Properties** ⇨ поставьте **Vertex Ticks**. Видите, объект образован всего 8-ю вершинами, которые находятся по углам объекта.

Как же он будет изгибаться бубликом при таком строении? Идите на закладку **Modify**, и в свитке **Parameters** измените значение **Width Segs.**. Теперь видите, объект состоит из двух половинок, которые можно «сломасть», и центр «слома» будет на месте новеньких вершин.

Кстати, большинство модификаторов используют имеющуюся структуру объектов, не добавляя новых вершин, поэтому есть общее правило: для успешного применения таких модификаторов надо чтобы объект содержал достаточное количество вершин. Ноги растут из того обстоятельства, что ребро (*edge*) не может быть криволинейным. Излом может быть только на стыке двух ребер, в вершине (*vertex*).

**Как в доске вырезать надпись?**

Тут придумал второй путь, но не знаю, как его реализовать. Взять квадрат, вычесть из него текст, потом проэкструдить. Все здорово, только текст из квадрата не вычитается.

Идея правильная, только к **Boolean** имеет очень отдаленное отношение:

1. Рисуем замкнутый сплайн (например, квадрат).
2. В той же плоскости внутри квадрата создаем текст.

3. Выбираем квадрат, закладка **Modify** ⇨ кнопочка **More** ⇨ пункт **Edit Spline**, отжимаем кнопку **Sub-Object**, ждем **Attach** и щелкаем на тексте. Теперь у нас получился один объект из двух сплайнов.

4. Теперь **Modify** ⇨ **More** ⇨ **Extrude** (либо если на закладке **Modify** в свитке **Modifiers** есть кнопочка **Extrude**, то ее).

Но это получится сквозной вырез в квадрате. А если действительно необходимо проделать Boolean-операцию со сплайнами, то:

1-3. См. выше.

4. Нажимаем **Sub-Object** и выбираем **Spline** в выпадающем списке справа от **Sub-Object**.

5. Выделяем первый сплайн (должен стать красным), нажимаем кнопку соответствующей Boolean-операции, ждем кнопку и щелкаем на второй сплайн.

### **Bomb** и его параметры. Как «взорвать» объект?

Для встраивания **Bomb** в объект необходимо нажать кнопку **Bind to Space Wrap** (в ToolBar), щелкнуть на нужном объекте и, не отпуская левой клавиши мыши, дотянуть пунктирную линию до **Bomb**.

При удачном встраивании в стеке объекта появится строка **MeshBomb Binding**.

**Bomb** разрывает объект по фейсам. Для лучшего эффекта необходимо назначить объекту 2-х сторонний материал и желательно наличие достаточно большого количества фейсов.

Сила **Bomb** действует от иконки бомбы во все стороны, с учетом значения параметра **Gravity** (сила тяжести, направленная вниз по мировой оси Z при положительных значениях).

Параметры:

- ◆ **Strength** — сила «взрыва» — скорость разлета осколков.
- ◆ **Spin** — вращение осколков.
- ◆ **Falloff (Falloff On)** — сферическая зона «захвата» фейсов. Фейсы объекта, не попадающие в эту зону, «взорваться» не будут.
- ◆ **Fragment Size: Min, Max** — минимальное и максимальное количество фейсов в одном «осколке».

- ◆ **Gravity** — сила тяжести. 0 — невесомость, больше 0 — осколки будут падать вниз по мировой оси Z, меньше 0 — вверх.
- ◆ **Chaos** — степень хаотичности разлета осколков.
- ◆ **Detonation** — время (по таймлайну) начала взрыва. До этого кадра бомба не работает.
- ◆ **Seed** — начальное значение для датчика случайных чисел.

Кстати, описанный ниже метод позволяет при взрыве получать не тоненькие плоскости, а куски объекта нужной толщины. В настройках **PArray** это есть (**Thickness**: в свитке **Particle Type**).

Для того, чтобы в момент, когда предмет приближается к плоскости разбивания, на него действовала **Bomb**, без ударной волны (то есть, только разделяла на кусочки), а потом чтобы эти кусочки, по действием **Gravity**, или того же **Bomb** долетали последние миллиметры до поверхности, и отскакивали уже от нее, уже разлетаясь в разные стороны, создайте объект **Create** ⇨ **Geometry** ⇨ **Particle Systems** ⇨ **PArray**, на закладке **Modify** выбираете для **Object-Based Emitter** свой объект, а потом в **Particle Type** устанавливаете тип **Object Fragments**. Привязываете **PArray** к **Deflector**.

Теперь создайте **Create** ⇨ **Space Warps** ⇨ **Particles & Dynamics** ⇨ **PBomb**, и именно им взрываете систему частиц **PArray** (а не свой предмет). Кстати, у **PArray** в свитке **Particle Generation** для **Emit Start**: лучше указать кадр, в котором бомба взрывается. И еще, после взрыва предмет лучше скрыть.

### Как спрятать объект на несколько кадров? А чтобы плавно?

Для этого нужно добавить **Visibility Track** в **Track View**.

Вызовите из главного меню **Track View** ⇨ **Open Track View**, найдите и выделите название объекта, который должен показываться/скрываться, и нажмите кнопку **Add Visibility Track**. У выделенного объекта появится **Track Visibility**.

Создавайте в нем ключи. Можете нажать кнопку **Apple Ease Curve**, открыть **Visibility Track** и редактировать ключи в **Ease Curve**. Если ключ имеет значение меньше 79, то он означает «не виден», если равен 79 или больше — «виден».

Есть еще один способ. Он позволяет плавно менять прозрачность материала объекта. Заходите в **Material Editor** при нажатой кнопке **Animation**, и, переходя в нужные кадры, регулируете значение **Opacity** материала. Потом обсчитываете картинку.

**Как заставить двигаться объект по нарисованному пути (линии)?**

Допустим путь в виде линии (закладка **Create** ⇨ кнопочка **Shapes** ⇨ **Line**) и движущийся объект уже создан... Выделяете объект (назовем его **Kirpich**). Так вот, выделяете **Kirpich**, переходите в закладку **Motion**, в свитке **Assign Controller** в списке выделяете пункт **Position: #####** (вместо ##### что-нибудь написано, но это не важно). Потом в этом же свитке нажимаете кнопочку **Assign Controller** и выбираете пункт **Path**, затем в свитке **Path Parameters** нажимаете кнопочку **Pick Path** и щелкаете на пути (линии), по которому должен двигаться объект **Kirpich**. Чтобы объект «смотрел» в ту сторону, в которую движется, нужно поставить галочку **Follow**, а чтобы двигался без ускорения/замедления, выставить галочку **Constant Velocity**.

Если нужно заставить двигаться объект по кривой линии, и при этом изгибаться по этой же линии (как изгибается червяк), то нужно: создать сам объект и линию, по которой он будет двигаться, затем выделить объект, на закладке **Modify** нажать кнопочку **PathDeform** (если нет на закладке, то на этой же закладке кнопочка **More...**, и выбрать пункт **PathDeform**), затем в свитке **Parameters** кнопочка **Pick Path** и щелкнуть на созданную линию.

Анимлируйте значение **Percent**. Все должно заработать. Только необходимо помнить два условия:

1. Движущийся объект должен быть достаточно сегментирован, то есть, иметь большое количество поперечных сегментов.
2. В качестве **Path** в **PathDeform** принимаются **Editable Spline, Line, Helix, NURBS Curve**.

**А как удалить ключи анимации?**

Есть два метода.

**Первый:** В главном меню выбираете **Track View**. Там открываете закладку **Objects** (крестик в кубике), находите по названию свой(и) объект(ы), открываете их, там открываете **Transform**, и видите все свои ключи. Если нажмете правой кнопкой на ключе (белом кружочке), сможете кое-что подправить.

Удаление выделенных (**Ctrl** + левая кнопка мышки) ключей — клавиша **Del**.

**Второй:** В закладках (с правой стороны экрана) выберите **Motion** (одноколесный велосипед), и можете удалять ключи, которые находятся в текущем кадре, нажимая в секции **Delete Key** клавиши: **Position, Rotation** и **Scale**. Только не ошибитесь, клавиши удаления ключей находятся в

секции **Delete Key**. В секции **Create Key** — клавиши создания ключей. А те клавиши, что вообще без секции, указывают, по каким именно ключам перескакивают две маленькие стрелки в свитке **Key Info (Basic)**.

### Как зациклить анимацию по кругу?

Заходите в **Track View**, добираетесь до трека с нужной модификацией объекта. Допустим, движение по кругу у **Box01: Objects** ⇨ **Box01** ⇨ **Transform** ⇨ **Position**. Выделили **Position**.

Теперь в меню кнопочка **Parameter Curve Out-of-Range Types**, и отмечае стрелки под рисунком **Ping Pong**. Или под **Loop**, если нужен резкий скачек в начальное положение.

Вроде должно работать...

### Как сделать так, чтобы объект начал двигаться с 50-го кадра, и двигался до 100-го кадра?

Самое простое, что можно сделать: анимировать кубик с 0 по 100 кадр, а затем в **Track View** сдвинуть нижнюю границу диапазона анимации данного объекта на нужный 50-й кадр. При этом, если имеется несколько ключей анимации, то они будут сдвигаться пропорционально. Если нужно изменить только начальный момент движения, то надо перейти на уровень ключей и сдвигать только первый ключ. **Track View** — крутейшая штука. Рекомендую!

Можно сделать по-другому. При начальном положении кубика установить 50-й кадр, перейти на вкладку **Motion** и создать ключ **Position** (в свитке **PRS Parameters** ⇨ секция **Create Key** ⇨ кнопочка **Position**). После этого перейти на 100-й кадр, нажать кнопочку **Animation** и перетащить кубик в конечную позицию.

Лично мне больше нравится первый способ, так как сразу видны все ключи.

При втором способе вы как бы блуждаете в потемках. Это особенно чувствуется при более-менее сложной анимации.

### Хочу, чтобы один объект перемещался в точном соответствии с перемещениями какого-нибудь Vertex'а другого объекта. Как это сделать?

Если вы сконвертируете объект в **Editable Mesh**, а потом прижмете **Sub-Object**, и при нажатой квадратной кнопочке **Animate** будете двигать вершины объекта, то они (вершины) появятся в **Track View**! Правда, под именами типа **Vertex 1**, но тут уже ничего не поделае. Так вот, можете приделывать к ним **Expression** и...

**Сгруппировал объекты, начинаю их вращать, а они крутятся вокруг своей оси. Что делать?**

Судя по всему, у вас включен режим **Каждый объект вращается вокруг своей оси**. Переключитесь в режим **Все объекты вокруг одной оси**. Делается это так: в группе кнопочек под главным меню нажмите и держите кнопочку справа от меню **Reference Coordinate System**.

Появится набор кнопочек, выберите **Use Selection Center** (названия смотрите в нижней части экрана, в информационной строке).

**Помогите запустить вершину объекта по некоторому пути!**

Давайте рассмотрим случай проведения одной вершины некоторой линии точно по указанному пути (то есть, по другой линии).

Создали две линии, **Line01** (это будет путь) и **Line02** (это обладатель движущейся вершины)? Идем на закладку **Modify**, и к **Line02** применяем модификатор **Extrude** на нулевую высоту — это для того, чтобы после применения модификатора **Edit Mech** (или какого другого) линия не исчезала.

Далее применяем **Edit Mech**, выделяем «двигающуюся вершину», передвигаем ее в точку с координатами (0,0,0), применяем к ней модификатор **XForm**.

После этого в **Track View** у объекта **Line02** должна появиться дорожка **Objects\Line02\Modified Object\XForm**.

Теперь создадим любой объект (это будет временный объект, поэтому можно сделать, например, **Create ⇄ Helpers ⇄ Dummy**), идете на закладку **Motion** и задаете этому объекту движение по пути **Line01**.

Затем, не откладывая в долгий ящик, в **Track View** копируете дорожку **Objects\Dummy01\Transform\Position** в дорожку **Objects\Line02\Modified Object\XForm\Gizmo\Position**.

Жаль, что сама линия **Line02** при **Rendering'e** не видна. Временный объект **Dummy01** можно удалить!

**Назначаю объекту контроллер движения Noise, и он перемещается к центру координат, и все движения происходят относительно него. Как привязать объект к конкретной точке координат?**

Для этого есть несколько способов. Самый изящный — **ListController**. Назначаете на **Position** этот контроллер. Сам по себе он ничем не управляет, а является стеком контроллеров. Движение же является результатом сложения выдаваемых ими координат. В данном случае верхним (active) контроллером ставите обычный **Besier**, в котором доста-

точно одного ключа с нужными вам базовыми координатами, а ниже — контроллер **Noise**.

### Как два объекта в 3ds max сбулить так, чтоб они двигались относительно друг друга?

Надо заранее анимировать объекты, и потом вычесть один из другого. Другой вариант — сначала вычесть, а потом у полученного объекта на закладке **Modify** нажать кнопку **Sub-Object**, справа от нее выбрать пункт **Operands** и анимировать уже эти составляющие объекта.

### Как определить, загружает ли 3ds max какой-либо plug-in?

Есть простой способ: продублировать plug-in, чтобы он находился и в директории `\3dsmax\stdplugins\`, и в директории `\3dsmax\plugins\`.

Если plug-in действительно написан для подходящей версии 3ds max, и он действительно plug-in, то 3ds max 7.0, найдя вторую копию, обязательно выведет сообщение об ошибке:

```
Class <имя_процедуры> from <имя_файла_с_plugin'ом> has
duplicate class ID: not loading. This may be due to a
duplicate plug-in path
```

По наличию этого сообщения можно судить о том, загружается plug-in или нет.

Если 3ds max ругается:

```
<имя_файла_с_plugin'ом> is an absolute version
```

значит у вас версия plug-in'a несовместима с самим 3ds max'ом.

Второй способ: вызвать в главном меню **File ⇨ Summary Info ⇨ Plugin Info**, это список plug-in'ов, которые 3ds max может использовать.

### А можно в Character Studio сделать модель человека? Или в чем еще это можно сделать?

Почему-то многие новички задают этот вопрос. При этом они даже не пытаются (видимо) сделать это в **Character Studio**. А ведь там все просто, все прозрачно до невозможности, все элементарно до не хочу!

Нельзя в **Character Studio** делать модели! Нету там такого. Ведь он предназначен только для того, чтобы управлять созданными моделями. Он просто заставляет модели двигаться так, как это делают живые люди... И не более того.

И еще, plug-in — **Anthropos** создает модели людей из набора примитивов, поставляющихся в комплекте с plug-in'ом (в виде библиотек).

Существуют так же сторонние программы, специально созданные для моделирования людей. Одна из них: **Poser**. Пробуйте...

### **Нужно чтобы любой объект вспыхивал и сгорал превращаясь в пыль (или горящие искры)**

Используй plug-in **Sand Blaster**. Причем можно сделать две копии — одну на пыль, другую на куски. В качестве эмиттера используй свой объект. А в момент распада прячете его через **Visibility**.

### **Можно в 3ds max сделать объект таким, чтобы от него другие объекты сами отталкивались/отскакивали?**

Можно это сделать двумя способами: или plug-in'ом **HyperMatter**, или есть система **Dynamics**.

### **Посоветуйте, как можно заставить в Character Studio манекен стоять на месте?**

Поясню: он дошел, к примеру, до 150 кадра, потом с 151 по 200 начинает двигать частями тела, стоя на месте, но проблема в том, что после 150-го кадра нет **footstep**'ов, поэтому он пишет, что не может создать ключи.

**Character Studio** не позволяет вообще создавать ключи частям **Biped**'а до тех пор, пока не будут созданы эти самые ступи.

Я долго пытался победить эту ситуацию и пришел к такому способу: кучкуете в нужном месте **Footstep** (накладываете их друг на друга), а потом у ног и рук убиваете ключи (через **Track View**, сгенерированные **CS** — и тогда в этом месте бипед остановится и замрет. Вот, включайте **Animate** и вперед.

### **Plug-in Surface Tools планарный текст не смог в нормальный объект превратить. Почему?**

У **Surface Tools** есть концептуальное требование: сплайны своими пересечениями должны образовывать либо треугольники, либо четырехугольники. Причина проста: Питер Ватью (автор плаги) избрал гениальный путь — он заполняет многоугольники, образуемые пересечениями сплайнов, **patch**'ами. А **Patch** в 3ds max, как известно, бывают только трех- и четырехугольными. Поэтому если у вас, например, есть замкнутый сплайн (как в случае с текстом), который имеет более 4-х вертексов, то **Surface Tools** его проигнорирует. Надо соединить часть вертексов змейкой другими сплайнами, чтобы образовать максимум четырехугольные ячейки.

Кстати, настоятельно рекомендую после завершения работы над SurfaceTools-моделью коллапсировать стек. Причина все в тех же patch'ax, которые в 3ds max жрут непомерно много памяти.

Желательно коллапсировать стек и напускать **Optimize** на полученный объект.

### **Для чего служит модификатор CrossSections, входящий в SurfaceTools?**

CrossSection служит для построения cross-сплайнов.

Допустим, вы строите некий шейп. Скажем, круг. С четырьмя вертексами. Потом еще один над первым, потом еще и еще... Потом **Attach** их один к другому по порядку, применяете **CrossSection**.

Он соединит соответствующие вертексы так, чтобы получилось что-то вроде цилиндра из сплайнов. И вот если после этого применить **Surface**, то и получите нужный patch'евый цилиндр.

Есть два правила построения шейпов для **CrossSection**:

1. Шейпы должны иметь одинаковое количество вертексов.
2. Аттачить шейпы друг к другу надо обязательно по порядку.

Если в примере с кругами вы приаттачите к первому последний, а потом какой-либо из середины, то **CrossSection** так и будет строить объект. В результате получите «сжика».

### **Как смоделировать атомный взрыв?**

Пользоваться plug-in'ом **PyroCluster**, в нем есть объект **A-Bomb** (закладка **Create** ⇨ **Geometry** ⇨ в меню **Pyro-Particles** ⇨ **A-Bomb**). Вот именно им такое и делается.

### **Как заставить 3ds max самостоятельно обчислить несколько проектов без помощи оператора?**

Необходимо настроить сетевой рендер. После этого будет возможно ставить проекты в очередь на обсчет (все как при сетевом рендере), но считать будет только один компьютер. Не забудьте запустить и **MANAGER.EXE**, и **SERVER.EXE** и **QUEUEMAN.EXE**.

Второй способ: использование скриптов.

### **Настройка сетевого рендера**

Сетевой рендер возможен при работе под Windows NT, Windows'9x, а также в смешанной сети NT-9x. Также возможен обсчет очереди проектов на одной станции.

Распараллеливание просчета одного проекта по сети может быть выполнено только при генерации последовательности кадров. Каждый отдельный кадр будет целиком считаться на одной станции (на одном компьютере). AVI, FLI и прочие анимационные файлы будут просчитываться целиком на одной станции (на одном компьютере).

Все примеры настроек в этом ответе даны для случая, когда один компьютер должен последовательно просчитать несколько сцен.

Для настройки сетевого рендера необходимо выполнить следующие шаги:

1. Установить и настроить протокол TCP/IP на всех компьютерах.

Для Windows'9x, если нет сетевой карты/модема: Start ⇨ Settings ⇨ Control Panel ⇨ Network ⇨ закладка Configuration ⇨ Add... ⇨ Adapter ⇨ Manufacturers: Microsoft ⇨ Network Adapters: Dial-Up Adapter ⇨ ОК. Двойной щелчок на TCP/IP ⇨ закладка IP Address ⇨ Specify an IP address ⇨ IP Address: 192.168.100.1, Subnet mask: 255.255.255.0.

Для Windows'9x, если есть сетевая карта: Start ⇨ Settings ⇨ Control Panel ⇨ Network ⇨ закладка Configuration ⇨ Add... ⇨ Protocol ⇨ Manufacturers: Microsoft ⇨ Network Protocols: TCP/IP ⇨ ОК. Двойной щелчок на TCP/IP ⇨ закладка IP Address ⇨ Specify an IP address ⇨ IP Address: 192.168.100.1, Subnet mask: 255.255.255.0.

Для Windows'9x, если есть модем: Start ⇨ Settings ⇨ Control Panel ⇨ Network ⇨ двойной щелчок на TCP/IP ⇨ закладка IP Address ⇨ Specify an IP address ⇨ IP Address: 192.168.100.1, Subnet mask: 255.255.255.0.

Для Windows'NT, если нет сетевой карты/модема: Start ⇨ Settings ⇨ Control Panel ⇨ Network ⇨ закладка Adapters ⇨ клавиша Add... ⇨ MS Loopback Adapter ⇨ клавиша ОК ⇨ клавиша ОК. Далее выполнить все инструкции, описанные для случая «Для Windows'NT, если есть сетевая карта/модем».

Для Windows'NT, если есть сетевая карта/модем: Start ⇨ Settings ⇨ Control Panel ⇨ Network ⇨ закладка Identification ⇨ клавиша Change... ⇨ в строке Computer Name: указать имя компьютера (например: GRAPHICS) ⇨ клавиша ОК. Закладка Protocols ⇨ клавиша Add... ⇨ выделить TCP/IP Protocol ⇨ клавиша ОК ⇨ клавиша No. Закладка Services ⇨ клавиша Add... ⇨ выделить Windows Internet Name Service ⇨ клавиша ОК. Клавиша ОК окна Network. Должно появиться окно Microsoft TCP/IP Properties (если окно не появилось, то после перезапуска компьютера Start ⇨ Settings ⇨ Control Panel ⇨ Network ⇨ закладка Protocols ⇨ выделить TCP/IP Protocol ⇨ клавиша Properties, и продолжайте выполнять инструкции как в случае, если бы окно появилось): закладка IP Address ⇨ включить Specify an

**IP address** ⇨ для поля **IP Address**: указать **192.168.100.1** (без кавычек), для **Subnet Mask**: указать **255.255.255.0** (без кавычек), для **Default Gateway**: указать **192.168.100.1** (без кавычек). Зкладка **WINS Address** ⇨ для поля **Primary WINS Server**: указать **192.168.100.1** (без кавычек). Клавиша **OK**.

После настройки перезагрузите компьютер.

Эту операцию нужно проделать на всех рендер-станциях в локальной сети. Рекомендуются IP-серии 192.168.100.\*, где «\*» число от 1 до 255. IP-адреса станций в сети должны быть уникальными. То есть, если у первого компьютера вы указали **IP Address: 192.168.100.1**, то у второго должно быть **IP Address: 192.168.100.2**, и так далее по порядку.

2. Для Windows'9x. Создаете текстовый файл **WINDOWS\hosts.ini** (можно откопировать **hosts.sam**), в нем прописываете соответствие IP-адресов именам станций (по строке для каждой станции в сети), например:

```
192.168.100.1 StationA
192.168.100.2 StationB
```

Для Windows NT файл **WINDOWS\system32\drivers\etc\hosts.ini** с абсолютно аналогичным содержанием.

Этот файл должен быть на каждом компьютере сети.

3. Удалить все файлы и подкаталоги из каталога **3dsmax\NetWork**, или создать такой каталог в случае его отсутствия.

При сетевом рендере необходимо иметь одну управляющую станцию-менеджер и одну или более станций-серверов (на которых будет производиться обсчет). Станция-менеджер может одновременно выполнять функции станции-сервера (возможность просчета очереди проектов на одной машине).

На станции-менеджере необходимо запустить программу **MANAGER.EXE**. При первом запуске возможна только установка параметров программы (оставляем все по умолчанию и выходим из программы). После этого при запуске будет выдаваться окно с логом и кнопки настройки.

На станции-сервере необходимо запустить программу **SERVER.EXE**. При первом запуске возможна только установка параметров. В поле **Manager name or IP-address** вписываем имя станции-менеджера (IP-адрес компьютера, на котором запускается **MANAGER.EXE**), остальное оставляем по умолчанию и выходим из программы.

Пример: при первом запуске в **Manager name or IP-address** указываем **192.168.100.1**.

При любом повторном запуске будет выдано окно с логом и кнопки настройки.

Следует помнить, что все сетевые установки 3ds max 7.0 на всех станциях в сети должны совпадать.

4. Проект, предназначенный для сетевого рендера, не должен быть привязанным к той станции, на которой он был создан. Каждая станция-сервер получает копию проекта. Поэтому, по возможности, пути должны быть в UNC, то есть, \\Station\Volume\Directory\File.Ext.

Исключения составляют те пути, которые совпадают для всех станций-серверов (например путь к стандартной библиотеке карт, ссылка на которую разрешается через диалог **Preferences** в 3ds max).

Проект обязательно должен быть сохранен перед сетевым рендером.

5. Для сетевого рендера на всех станциях-серверах необходимо запустить SERVER.EXE. Компьютер с запущенным SERVER.EXE может участвовать в просчете сцены.

На станции-менеджере необходимо запустить MANAGER.EXE.

Компьютер с запущенным SERVER.EXE может рассылать сцены для просчета другим компьютерам.

6. На станции-менеджере запускаем 3ds max, подготавливаем проект. В диалоге ⇨ главное меню ⇨ **Rendering** ⇨ **Render** отмечаем галочку **NetRender** и нажимаем кнопку **Render**. В следующем диалоге **Network Job Assignment** в редактируемом выпадающем списке **Network Manager** вписываем имя станции-менеджера (пример: вписываем **192.168.100.1**), и нажимаем кнопку **Connect**. После этого в списке, расположенном ниже, появятся имена станций-серверов. В этом списке необходимо отметить те станции, которые должны считать данный проект.

Галочка **Inactive** отложит выполнение просчета до активизации задачи в **Queue Manager** (QUEUEMAN.EXE).

7. На станции-менеджере запускаем программу QUEUEMAN.EXE. Вписываем имя станции-менеджера (пример: **192.168.100.1**), открываем свитки с названиями сцен, заказанных на просчет, и дважды щелкая по названиям станций, готовых к просчету сцены, указываем время, когда станция может участвовать в просчете.

Пример: в случае один компьютер должен просчитать 20 сцен на этом компьютере должны быть запущены SERVER.EXE, MANAGER.EXE и QUEUEMAN.EXE.

### Какая польза 3ds max от наличия в компьютере 3Dfx-карты?

Польза только одна: ускорение прорисовки экрана. Еще появляется возможность подключить систему RenderGL.

OpenGL «для игрушек» сильно отличается от OpenGL, которые используются в 3D-программах. Чипы фирмы 3Dfx не держат нормальный OpenGL в принципе. Для них нет ICD-драйвера. Только MCD-драйвера. А с системой Direct3D в 3ds max работать практически невозможно.

### Как сделать эффект фокусирования на одну точку или один предмет, чтобы вокруг этой точки все было расплывчатым?

Делаете **Rendering** ⇨ **Video Post** ⇨ **Add Image Filter Event** ⇨ **Lenz Effects Focus**. Выбираете тип эффекта: **Focal Node**. Там, кроме всего прочего, указываете объект, который будет в фокусе. Если в качестве **Focal Node object** выбрать объект **Dummy** и придвигать его к камере, то фокусировка с заднего плана плавно перейдет на передний...

### Как окрасить лучи света, проходящие через цветное стекло? И как сделать так, чтобы свет Volume Light не проходил сквозь объекты?

Это в 3ds max'e не реализовано. Если нужно прохождение света через разноцветный витраж: делаете витраж, делаете источник света **Target Spot**. Щелкаете правой клавишей мышки на названии одного из окошек, меню **Views**, выбираете название своего источника света (допустим **Spot01**). Рендерите этот вид и записываете в файл. Выделяете источник света **Spot01**, на закладке **Modify** в свитке **Spotlight Parameters** ставите галочку для **Projector** и нажимаете клавишу **Map**: (на ней перед нажатием написано **None**). В появившемся окне в секции **Browse From**: отмечаете **New** и двойной щелчок на пункте **Bitmap**.

Закрываете это окно и идете в **Material Editor**. Щелкаете на любом квадратике с материалом, нажимаете кнопку **Get Material**, в секции **Browse From**: отмечаете **Scene** и двойной щелчок на материале: **Map #1 (Bitmap) (Environment, Spot01)** Скорее всего у вас будет какое-то другое число вместо #1.

Закрываете окно **Material/Map Browser**, в свитке **Bitmap Parameters** щелкаете кнопочку **Bitmap**: и выбираете свою картинку, которую вы сохранили в начале. В секции **Coordinates** отмечаете **Environ**, в меню **Mapping**: выбираете **Screen**.

Закрываете **Material Editor**. В главном меню **Rendering** ⇨ **Environment** в свитке **Atmosphere** жмете кнопку **Add** и выбираете **Volume Light**. В свитке

**Volume Light Parameters** жмете **Pick Light** и щелкаете на свой источник **Spot01** (который виден на заднем плане).

Чтобы **Volume Light** не просвечивал сквозь объекты, поставь источнику света тени **Shadow Maps** (закладка **Modify**, в свитке **Shadow Parameters** отметить **Use Shadow Maps**).

Теперь рендерите свою картинку, подправляете настройки, рендерите, подправляете, до бесконечности.

### Как заставить 3ds max использовать не один файл film0000.tga, а последовательность картинок film0000.tga...film0029.tga?

Нужно создать простой текстовый файл с расширением **.IFL** (например **FILM.IFL**), в котором отдельными строчками вписать названия используемых файлов. Вот пример:

```
film0000.tga 120
```

```
film0001.tga 100
```

```
.....
```

```
film0028.tga 120
```

```
film0029.tga 120
```

В качестве картинки везде нужно использовать именно этот текстовый файл **FILM.IFL**. Продолжительность одной картинки (в кадрах) назначается цифрой справа от названия картинки.

В 3ds max есть утилита для создания **IFL** файлов. На закладке **Utilities** жмете кнопку **More...**, и выбираете **IFL Manager** Очень удобно, особенно когда **TGA**-шек много.

### Как можно сделать лого в углу экрана?

В **Rendering** ⇨ **VideoPost** помимо камеры (**Add Scene Event**) добавляете **Input Image Event** и загружаете картинку со своим лого (заранее сделанную хоть в том же 3ds max'e). Картинка должна быть с альфа-каналом, чтобы просвечивало сквозь нее. Потом выделяете эти два события, жмете на **Add Image Layer Event**, и выставляете эффект **Alpha Compositor**.

Еще такую процедуру можно проделать в программах видео монтажа.

### Почему 3ds max фон в картинках делает не равномерный? На глаз это не заметно, но при редактировании картинки мешает

3ds max картинку всегда считает в 48bit'ной палитре, а потом переводит ее в палитру картинки, которую должен записывать.

Решение проблемы — записывать картинки в формате с минимальной потерей цветности. Например, не в 24bit **TARGA**, а в 48bit **PNG**.

**Делаю превью, а там вместо нормального видео с текстурами, тенями, все без текстур, частички — крестики! Как сделать видео, чтоб качество было как в рендере, или приблизительно?**

У превью качество превью — быстро и дешево. Хотите качества как в рендере — рендерите. Поставьте разрешение для ролика 240x160, запретите просчет отражений, теней, текстур, эффектов, атмосферы, motion blur, задайте параметры для рейтрейсных материалов (через кнопку **Options**). Естественно все по желанию и необходимости.

Еще можно делать **Display ⇔ Hide Selected/Hide by Name...** для объектов, не нужных в данный момент.

Кстати, именно для этого в 3ds max 7.0 в **Render** есть два режима настроек — для **Production** и для **Draft**-качества. В **Draft** можно кой-какие галочки убрать...

**Почему в сделанном с помощью 3ds max AVI нет звука? Вроде устанавливаю, в 3ds max нормально все прокручивается, со звуком, а в AVI его нет!**

Его там и не должно быть. Звук проигрывается в 3ds max только для синхронизации анимации со звукорядом.

**При обсчете картинки, на которой есть очень сильно приближенный к камере объект, начинают пропадать некоторые части этого объекта, словно их срезали ножом**

Если при сильном приближении камеры к объекту его части начинают исчезать так, словно их отрезали объективом камеры, то вам, вероятно, нужно в ручную настроить положение границ зоны визуализации (clipping planes). Выделите камеру, перейдите на закладку **Modify**, и в секции **Clipping Planes** включите параметр **Clip Manually**. Затем установите для параметра **Near Clip** значение 0, а для **Far Clip** какое-нибудь очень большое значение, например 100 000, и все будет в порядке.

**Как обойти проблему несовместимости форматов при импорте файлов Adobe Illustrator в 3ds max?**

Загвоздка в формате текста, а не самого AI. Многие программы сбрасывают AI в юниконовом формате — в конце строки стоит только код 0x0A, а 3ds max желает в досовском, то есть, в конце строки должно стоять 0x0D, 0x0A. Проблему можно решать различными способами.

1. Открываем в **MultiEdit for Windows**, загружаем файл AI с указанием формата Unix, сохраняем как (**Save As**) файл в формате DOS.

2. Открываем AI-файл в редакторе ДОС Навигатора. Сохраняем с другим именем.

3. Пишем сами на любом языке программирования или ищем простенькую утилиту, которая производит вышеописанное преобразование.

### Как сделать зеркало?

В 3ds max'e зеркальная поверхность делается 3-мя способами: один для плоской поверхности, другой для криволинейной, третий общий (но тормозной).

1. Для криволинейной поверхности делаем стандартный материал с картой отражения (**Reflect/Refract**) на **Reflect (100%)**.

2. Для плоской поверхности в общем случае делается **Multi/Sub Object** материал с 2-мя подматериалами. Один подматериал — любой стандартный (не зеркальный), другой — стандартный с картой **Flat Mirror** на **Diffuse**.

Далее обычная работа с **Multi/Sub Object** материалом. Только не советуем на смежные некопланарные фэйсы ставить зеркальный подматериал — могут вылезти глюки на картинке.

3. Рейтрейсовый материал (**MAX 2+**) — самый медленный, но и дающий самые реалистичные результаты. В редакторе материалов выбирается тип материала **Raytrace**. В простейшем случае требуются настройки 2 параметра: **Diffuse** и **Reflect** (степень отражения).

### Пакетный рендер. Как поставить несколько проектов в очередь для просчета?

Необходимо настроить сетевой рендер. После этого будет возможно ставить проекты в очередь на обсчет (все как при сетевом рендере), но доступная рабочая станция будет в сети только одна.

### Как настроить Glow в 3ds max 7.0?

Эффект в видеопосте. Для его работы необходимо установить канал материала (в редакторе материалов кнопка с цифрой под сэмплами и/или канала объекта. Плагин (стандартный) **Lens Effects (apollo.ft)**.

1. Установить канал объекта отличным от 0 (0 — по умолчанию).

2. В видеопосте кнопки **Add Scene Event** (добавить просчет сцены); **Add Image Filter Event** (добавить фильтр для изображения) в диалоге, в выпадающем списке выбрать **Lens Effects Glow**, кнопка **Setup** — настройка эффекта; **Add Image Output Event** — добавить запись картинку в файл.

3. Настройка эффекта. Там все достаточно прозрачно. Кнопка **Preview** разрешает показ. Кнопка **VP Queue** заставляет показывать в **preview** реальную сцену из очереди видеопоста (в противном случае тестовая картинка 3ds max).

Все настройки анимируются. Ниже — некоторые основные установки.

#### Страница Properties

- ◆ **Object ID** — тот, который был установлен в п. 1;
- ◆ **Material ID** — канал эффектов материала (устанавливается в редакторе материалов кнопкой **Material Effects Channel**).

#### Страница Preferences

- ◆ **Size** — размер эффекта свечения;
- ◆ **Color:**
  - ◆ **Gradient** — свечение будет генерироваться в соответствии с установками, сделанными на странице **Gradients**;
  - ◆ **Pixel** — свечение будет рассчитываться, исходя из цвета пиксела объекта;
  - ◆ **User** — свечение будет выведено на основе из указанного цвета;
  - ◆ **Intensity** — плотность эффекта.

Есть еще плагины, реализующие эффект **Glow**:

- ◆ **Glow+**
- ◆ **RealLensFlare**

# Содержание

## Часть 1. 3ds max — седьмая версия

Глава 1. Основы 3D технологий .....	3
Глава 2. Новая версия программного обеспечения для профессиональной трехмерной анимации и моделирования .....	16

## Часть 2. Моделирование

Глава 1. Конструкция персонажа .....	19
Глава 2. Bone Tools .....	21
Глава 3. Editable Polygon .....	23
Глава 4. Модификатор Unwrap UVW .....	27
Глава 5. Модификаторы: Symmetry Modifier, Vertex Weld Modifier, Edit Normals Modifier .....	29
Глава 6. Сплайновое моделирование .....	31
Глава 7. Просчет и фотореализм .....	33
Глава 8. Интерфейс программы .....	35
Глава 9. Моделирование текста .....	37
Глава 10. Хромированный металл .....	41
Глава 11. Моделирование ландшафта .....	47
Глава 12. Моделирование текстуры ландшафта .....	51
Глава 13. Моделирование морского ландшафта .....	54
Глава 14. Моделирование руки по четырем огибающим .....	57
Глава 15. Моделирование одного пальца руки .....	59
Глава 16. Моделирование полной руки — ладонь, пальцы, плечо, предплечье .....	61
Глава 17. 3D-моделирование .....	64
Глава 18. Создание красивого 3D текста .....	75
Глава 19. Создание изображения лампы .....	77

Глава 20. Пэтчевое моделирование .....	79
Глава 21. Создание раздельного освещения и затенения сцены .....	85

## Часть 3. Слои и модификаторы

Глава 1. Основная идея применения .....	87
Глава 2. Модификатор HSDS modifier .....	88
Глава 3. Модификатор Morpher .....	89

## Часть 4. Материалы и шейдеры

Глава 1. Ink 'n Paint .....	90
Глава 2. Translucent Shade .....	90
Глава 3. Advanced Lighting Override Material (ALO) .....	91

## Часть 5. Анимация в программе

Глава 1. Элементы нелинейной анимации .....	94
---	----

## Часть 6. Элементы освещения

Глава 1. Light Tracer .....	98
Глава 2. Radiosity .....	102
Глава 3. Новые типы теней .....	106

## Часть 7. Рендеринг

Глава 1. «Текстурные выпечки» .....	109
Глава 2. Извещение о статусе рендеринга по почте .....	110

## Часть 8. Интерфейс

Глава 1. Манипуляторы трансформаций объекта .....	111
Глава 2. Изменения Main Toolbar .....	113
Глава 3. Главное меню .....	113
Глава 4. Редактор материалов .....	115
Глава 5. Горячие клавиши .....	115

**Часть 9. Советы**

Глава 1. Примитивы в 3ds max	117
Глава 2. Как нарисовать книгу	122
Глава 3. Как сделать реалистичный взрыв	124
Глава 4. Как сделать параллельные лучи света	128
Глава 5. Создание фонтана	130
Глава 6. Создание зеркальных поверхностей	131
Глава 7. Эффект растущего текста	134
Глава 8. Использование anti-aliasing	137
Глава 9. Mountain Texturing	137
Глава 10. Использование Photoshop для создания текстур	145
Глава 11. Анизотропные отражения	150
Глава 12. Создание 3D-живности	164
Глава 13. Создание правильной гайки с помощью лофта Rhino	168
Глава 14. Создание волос с применением plug-in Shag: Fur	170
Глава 15. Создание простой головы	173
Глава 16. Моделирование головы с использованием Surface Tools	177
Глава 17. Создание подводных эффектов	180
Глава 18. Как сделать облака	189
Глава 19. Море, волны...	191
Глава 20. Объемный свет на примере мозаичного окна	197
Глава 21. Как создавать правильный свет	200
Глава 22. Раздельное освещение и затемнение сцены	203
Глава 23. Бриллианты, рубины, аметисты... или как делать бижутерию	205
Глава 24. Создание молнии	207
Глава 25. Создание фотонной пушки	211
Глава 26. Простой способ создания пара	212
Глава 27. Создание реалистичного лазера	214
Глава 28. Создание космического эффекта — солнце	219
Глава 29. Имитация работы сварочного аппарата	220

Глава 30. Создание анимированных объектов .....	224
Глава 31. Создаем стекло .....	262
Глава 32. Тонкости и хитрости создания металлических поверхностей .....	268
Глава 33. Создание стеклянных и металлических поверхностей .....	275
Глава 34. Три способа создания зеркал .....	277
Глава 35. Советы по моделированию зверей .....	280
Глава 36. Моделирование растений .....	311
Глава 37. Plug-in Tree Storm, позволяющий создавать растения .....	314
Глава 38. Сажаем дерево .....	316
Вопросы и ответы .....	321
Список использованной литературы .....	357

Петрова, А. М.

GPS: Все, что Вы хотели знать, но боялись спросить. Неофициальное пособие по глобальной системе местоопределения / Петрова А. М. — М.: Бук-Пресс и К, 2005. — 352 с. — (Мобильные коммуникации).

ISBN 5-88548-501-3



Определение своего положения с помощью GPS навигатора, отдельного прибора, или устройства, встроенного в карманный компьютер или сотовый телефон, уже стало совершенно обычной вещью. Постепенно столь же привычным становится определение положения объекта с помощью систем телематики на основе GPS/GSM/GPRS, когда на мониторе компьютера или экранчике сотового телефона можно увидеть участок карты с отметкой, где находится другой человек или его автомобиль.

«GPS» — это первые буквы английских слов «Global Positioning System» — глобальная система местоопределения. GPS состоит из 24 искусственных спутников Земли, сети наземных станций слежения за ними

и неограниченного количества пользовательских приемников-вычислителей. «GPS» предназначена для определения текущих координат пользователя на поверхности Земли или в околоземном пространстве. Как и многие многоцелевые вещи в нашем быту, приемник системы глобального позиционирования (GPS) по мере знакомства с ним обнаруживает массу полезных свойств, даже сверх тех, для которых он был приобретен первоначально. Оказывается существует много любопытных вопросов, на который он с легкостью отвечает, — например, какую скорость вы развиваете при ходьбе, какое расстояние вы преодолеваете при занятии бегом и с какой максимальной и средней скоростью, какую скорость вы развили, спускаясь с горы на лыжах, насколько точен спидометр вашего автомобиля. Однако основное его назначение — определение координат.

В этой книге рассмотрены следующие темы: «Глобальная система местоопределения», «Система позиционирования», «Принцип работы систем спутниковой навигации», «Я и GPS», «Основы GPS», «GPS для пользователя системы слежения», «Основные функции GPS-приёмников», «Точность системы слежения». Программное обеспечение: «MacCentre Pocket», «PocketGPS Pro Moscow», «ПалмГИС», «Векторная карта для GPS-навигаторов с поддержкой картографии», «КПК Psion». Тонкости и хитрости: «Ноутбук и GPS», «Беспроводная связь и телематические системы в автомобиле», «Настройки системы координат в GPS-приемнике», «Краткий обзор GPS-приёмников по ценам и характеристикам», «Подключение GPS-навигатора к персональному компьютеру», «Использование GPS в походе», «Как за неделю научиться грамотно ездить по Москве»; «Особенности национальной GPS-навигации». «Железо»: «Выбираем GPS-навигатор», «Pocket Navigator». Автомобильные GPS-навигаторы и эхолоты: «Устройство и основные функции эхолота», «Как выбрать эхолот», «Основные принципы работы эхолота», «Как установить эхолот на моторном судне».



Одобрено редакцией «Бук-пресс и К»

«GPS» — это не только средство для определения координат, но и инструмент для навигации. В этой книге вы узнаете, как использовать GPS-приемник для определения координат, как настроить систему GPS-навигации, как выбрать эхолот, как установить эхолот на моторном судне.

Слободецкий, А. А.

ArchiCAD 9.0: Все, что Вы хотели знать, но боялись спросить. Неофициальное пособие по созданию виртуальных зданий на персональном компьютере / А. А. Слободецкий. — М.: Бук-Пресс и К, 2005. — 320 с. — (САПР и персональный компьютер).

ISBN 5-88548-502-4



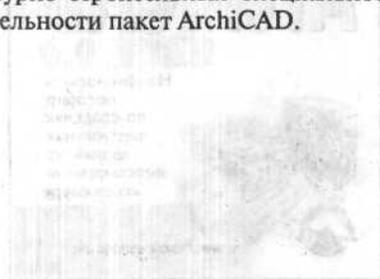
Вне зависимости от того, являетесь ли вы частным архитектором или работаете в проектной отделе крупной строительной фирмы, возводящей небоскребы, управляете крупным офисом или производите строительные конструкции, программа ArchiCAD всегда может предложить решение, которое будет выгодно именно вам. Именно поэтому пакет ArchiCAD изначально разрабатывался как программа, с которой архитектор работает каждый день и через которую общается с другими специалистами.

С помощью этой книги Вы научитесь работать с девятой версией «флагманского» продукта компании Graphisoft (более подробную информацию о компании вы можете найти на сайте [www.graphisoft.com](http://www.graphisoft.com)).

Теперь ArchiCAD основан на уникальной технологии «Виртуального здания», т. е. Вы можете создавать «живые», реальные проекты. Кроме этого, пакет «заточен» под требования современных архитекторов. ArchiCAD 9.0 очень прост и понятен как при двумерном проектировании, так и при работе в трехмерной среде. Ведь каждый архитектор предпочитает инструменты, которые позволяют ему сосредоточиться на том, чему он посвятил жизнь, — на творческой работе. Благодаря ArchiCAD 9.0 архитекторы смогут по-новому взглянуть на процесс архитектурного проектирования и полностью реализовать свои возможности. Интуитивно понятный интерфейс программы позволит работать с трехмерной моделью сооружения («Виртуальным зданием») и опытному, и начинающему пользователю. В результате у архитекторов, использующих ArchiCAD 9.0, останется больше времени для любимого дела — развития архитектурной фантазии! В ArchiCAD 9.0 появилось множество новых функций, которые помогут Вам при создании презентаций проекта и подготовке архитектурной документации.

Войдите в новый ArchiCAD вместе с этой книгой, в которой собраны различные советы, рекомендации и приемы работы.

Для архитекторов, инженеров-строителей, дизайнеров по интерьерам, студентов архитектурно-строительных специальностей и всех, кто использует в своей деятельности пакет ArchiCAD.



Альбов, К. К.

Microsoft Windows XP SP2: Все, что Вы хотели знать, но боялись спросить. Неофициальное пособие по работе с персональным компьютером / К. К. Альбов. — М.: Бук-Пресс и К, 2005. — 336 с. — (Мой персональный компьютер).

ISBN 5-88548-503-5



Нужно признать, что корпорация Microsoft постоянно работает над совершенствованием своего программного обеспечения. Microsoft выпускает обновления и исправления для решения известных проблем. К примеру, пакет Service Pack 2 для Windows XP делает работу в Интернете и использование электронной почты удобнее благодаря новым технологиям безопасности, позволяющим избежать загрузки нежелательного программного обеспечения.

В данной книге содержится «хакерский» (действительно) материал относительно скрытых возможностей, имеющих в операционной системе Microsoft Windows XP. Показывается, каким

