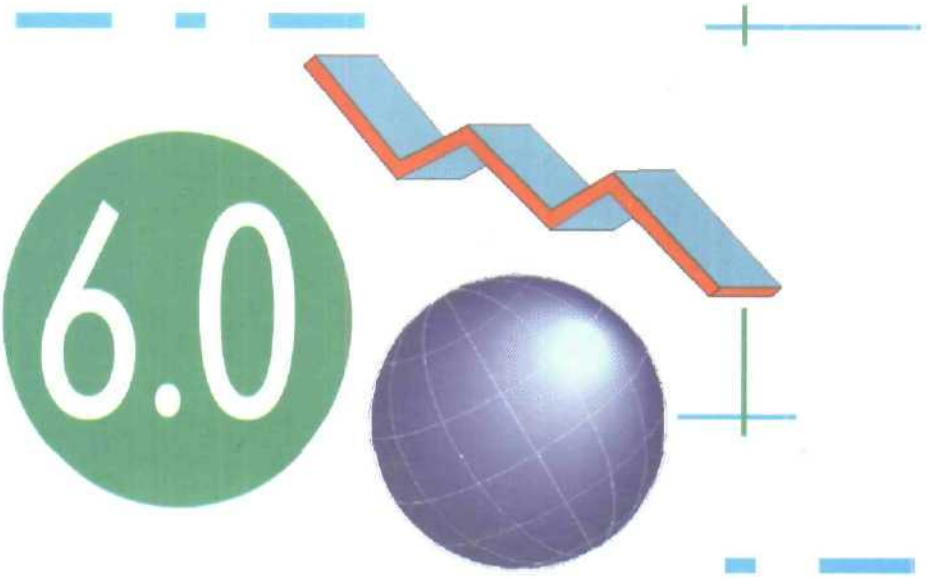
3D Studio MAX

Практический курс



Эта книга посвящена новинкам и усовершенствованным возможностям популярной программы для работы с компьютерной графикой и анимацией. В программе появилось много новинок, о которых могли только мечтать пользователи предыдущих версий 3D Studio Max. Некоторые изменения претерпел интерфейс программы, улучшены средства моделирования геометрии, кардинально обновлены средства работы со светом, пополнено семейство материалов, усовершенствованы методы визуализации и анимации.

Серия книг
«Ваш персональный компьютер»

Иван Слободецкий

3D Studio MAX 6.0
Практический курс

Москва
Редакция «Компьютерная литература»
Издательство «Познавательная книга Пресс»
2004

УДК 004.5
ББК 32.973.26-0 [8.2
С4S

Слободецкий И. М.

С48 3D Studio MAX 6.0: Практический курс, - 2004. - 3S4 с. - Серия книг «Ваш персональный компьютер»

ISBN 5-8321-0602-7

Эта книга посвящена новинкам и усовершенствованным возможностям популярной программы для работы с компьютерной графикой и анимацией — 3D Studio Max. В программе появилось много новинок, о которых могли только мечтать пользователи предыдущих версии 3D Studio Max. Некоторые изменения Претерпел интерфейс программы, улучшены средства моделирования геометрии, кардинально обновлены средства работы со светом, пополнено семейство материалов, усовершенствованы методы визуализации и анимации. В основу книги положено изучение 3D Studio Max на примерах анимационных проектов. Автор подробно раскрыл все этапы процесса производства анимационной продукции с использованием возможностей 3D Studio Max: моделирование, снаряжение, текстурирование персонажей, анимация, освещение, визуализация и компоновка. Каждое новое или усовершенствованное свойство программы рассматривается с практической точки зрения попутно со специальными приемами и методами.

Книга рассчитана в первую очередь на пользователей, уже имеющих опыт работы с предыдущими версиями программы.

УДК 004.5
ББК 32.973.26-018.2

ISDN 5-8321-0602-7

Часть 1. Введение в трехмерный дизайн

Глава 1. Основы гармонии в формах

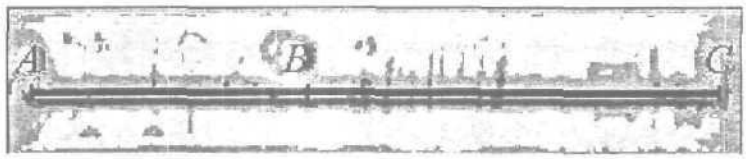
Сегодня 3D графика — такое же полноправное направление в искусстве, как и живопись, фотография. Но несмотря на отсутствие границ в виртуальном мире, где мы при достаточном запасе знаний можем отменить силу тяжести, инерцию, и даже... смерть (любимый код ленивых геймеров), собственное восприятие мы изменить не можем — Бог предусмотрел защиту от наших крэков и патчей, а исходники запрятал далеко на Небесах. Конечно и без багов не обошлось — изображение мы видим перевернутое и уменьшенное, цветовосприятие очень ограниченно из-за малых размеров пятна с колбочками и палочками в глазу, угол зрения у нас в лучшем случае 90 градусов, а самое обидное — это нельзя сохраняться и нажимать «undo». И все же человечество постоянно пыталось что-то изменить, но люди более практичные решили не портить природные дары (все равно получится, как всегда), а изучить их.

Форма и цвет— вот, что всегда интересовало человека в окружающих его объектах. Интерес к какому-либо объекту базируется либо на опыте, либо на интересе — отсутствии опыта (что это за объект? — может ли он мне пригодиться?). Из этого может возникнуть продолжение: «такая форма объекта подходит для того, чтоб...», либо еще более интересное продолжение, которое базируется на воззрениях даосизма: «эта форма бессмысленна, она не может быть полезно использована, но она удивительна, интересна, красива и оригинальна. От нее исходит тепло». В даосизме это явление называется Ву — суть его заключается в том, что что-либо, в нашем случае форма, не несет ни сути, ни содержания, но она близка к совершенству, в ней заключена гармония. Тепло исходит от формы в душу человека. Когда такая форма создана человеком, тепло исходит от души создателя через форму принявшую в себя положительную энергию создателя. Когда человек определяет форму и вкладывает в нее часть своей души, он. как часть природы — гармонического состояния

материи, определяет гармоничную форму. Но оставим подобные рассуждения философам и теологам, а сами рассмотрим научное представление о форме, и сказанное выше просто учтем в наших практических изысканиях. Форма, в сути которой, лежат сочетание золотого сечения и симметрии способствует проявлению ощущения красоты и гармонии. Всякое целое состоит из множества мастей. И все эти части состоят в определенном отношении между собой и целым. Золотое сечение рассматривает определенное гармоническое отношение частей и целого, как проявление совершенства в природе.

Золотое Сечение — гармоническая пропорция

В геометрии пропорцией называют равенство двух или более отношений: a : b = c : d. На примере отрезка, рассмотрим возможные отношения частей отрезка и целого:



- Можно разделить отрезок AC на две равный части AC:AB=AC:BC;
- На две неравный части, которые не образуют пропорции;
- Так, что целое относится к большему, как большее к меньшему.

Последнее отношение и есть золотая пропорция — золотое деление — деление отрезка *в* крайнем и среднем отношении.

Золотое сечение — пропорциональное деление отрезка на неравные части, при котором целый отрезок так относится к большей части, как большая часть относится к меньшей:

AC : BC = BC : AB

или наоборот

АБ : BC = BC : AC

До этого момента, мы использовали только переменные, а сейчас мь: выразим полученные отрезки. Представим целый отрезок равным единице, тогда части будут выражены бесконечной иррациональной дробью:

BC = 0.618 AB = 0,382

Свойства золотого сечения описываются квадратным уравнением:

$x^2 - x - 1 = 0$

Золотое сечение имеет множество интересных свойств, в результате чего, золотою пропорцию наделяли даже магическими свойствами.

Для нахождения отрезков золотой пропорции, можно пентаграмму — геометрическую фигуру, ставшую магическим знаком. Так в перевернутую пентаграмму сатанисты вписывают голову Сатаны, маги и колдуны использовали пентаграмму для своих магических экспериментов, для вызывания духов:

Пентаграмма строится путем вписывания в окружность правильного пятиугольника. Чем же нас так заинтересовала пентаграмма? Тем, что каждый конец пентаграммы представляет собой золотой треугольник. При вершине образуется угол, равный 36°, а основание отложенное на боковую сторону делит ее на отрезки в золотой пропорции.

Вероятней всего знание золотого сечения возникло в Египте. Доказательством этому является то, что пропорции пирамид, храмов соответствовали пропорциям золотого деления. И в самом деле: наклон ребра пирамиды Сахуре = 40°36', Микерина = 41°38', Хеопса = 41°59'. Но самое интересное — изображение зодчего Хесира, держащего в руках из-

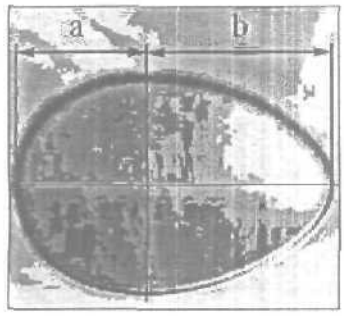
мерительные инструменты, на которых зафиксировано золотое деление. Действительно, при раскопках в Египте, Греции были найдены так называемые «пропорциональные циркули».

С золотым сечением связано имя итальянского математика Фибоначчи. Ряд чисел О, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 и т.д. известен как ряд Фибоначчи. Особенность этой последовательности чисел состоит в том, что каждый ее член, начиная с третьего, равен сумме двух предыдущих 2 + 3 = 5; 3 + 5 = 8;5 + 8 = 13; 8 + 13 = 21; 13 + 21 = 34 и т.д., а отношение смежных чисел ряда приближается к отношению золотого деления. Так, 21 : 34 = 0,617; 34 : 55 = 0,618; 55 : 89 = 0,617.

Это отношение обозначается символом Ф. Только это отношение — 0,618 : 0,382 — дает непрерывное деление отрезка прямой в золотой пропорции, его увеличение или уменьшение до бесконечности, когда меньший отрезок так относится *к* большему, как больший ко всему.

На первый взгляд это просто обычная прогрессия, но на самом деле, это один из законов природы. Рассмотрим развитие какого-либо растения, для наглядности вьющегося. Отросток делает сильный выброс в пространство, останавливается, выпускает листок, делает еще один выброс, но уже короче первого, выпускает листок еще меньшего размера и снова выброс. Если первый выброс принять за 100 единиц, то второй равен 62 единицам, третий — 38, четвертый — 24 и т.д.

И в растительном, и в животном мире определяется формообразующая тенденция природы — симметрия относительно направления роста и движения. В частях проявляется повторение строения целого.



Все, что растет — стремится занять наибольший объем. Для этого оно растет вверх и расстилается по поверхности. Комбинация этих двух процессов заключается в развитии формы в виде спирали.

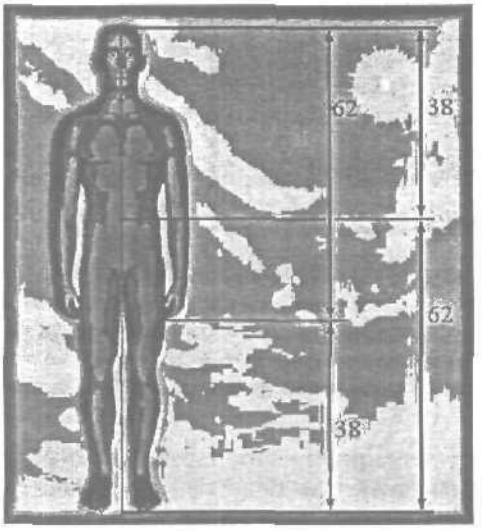
Спиралевидное развитие большинства природных форм отметил еще Гете, он называл спираль «кривой жизни». Молекула ДНК закручена двойной спиралью, паук плетет паутину спиралеобразно, спираль определяется в строении шишек сосны, расположении семян подсолнечника, кактусах, ананасах, расположение листьев на ветке (филотаксис), спиралью закручивается ураган, испуганное стадо северных оленей разбегается по спирали.

Формула спирали была выведена Архимедом, и носит его имя.

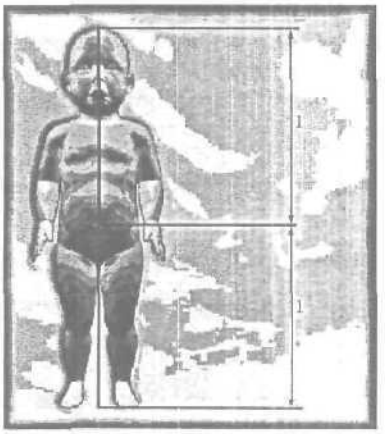
Все эти явления проявляют закономерность ряда Фибоначчи.

Золотое сечение и человек

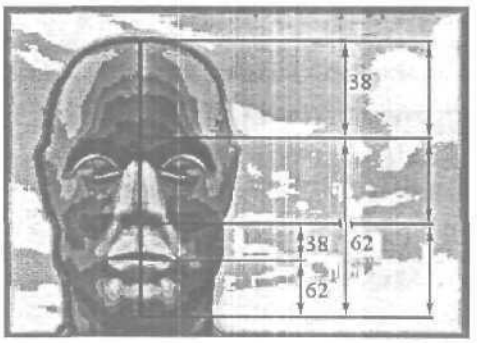
Немецкий профессор Цейзинг в середине 18 столетия проделал огромную работу: он измерил более 2000 тел и высказал предположение, что золотое сечение выражает среднестатистический закон: деление тела точкой пупа — один из основных показателей золотого сечения. Пропорции мужского тела колеблются в пределах среднего отношения 13 : 8 = 1,625. Пропорции золотого сечения проявляются и в отношении других частей тела — длина плеча, предплечья и кисти, кисти и пальцев и т.д.



У маленьких детей (около года) пропорция составляет отношение 1 : 1.

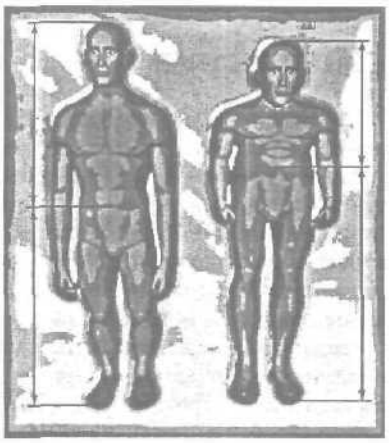


Голова человека тоже проявляет пропорции золотого сечения:



Золотое сечение является основой построения гармоничных форм, так как является абсолютным законом формообразования в природе, частью которой мы являемся. Законы гармонии — есть числовые законы. Поэтому необходимо использовать знания и опыт человечества для развития подходов при определении форм моделей в ваших виртуальных мирах.

Мы рассмотрели среднестатистичного человека, который гармоничен, так как в его формах заключены пропорции золотого сечения. а теперь, мы используем те же модели, но нарушим закон Цейзинга о делении тела точкой пупа, мы будем даже утрировать эти нарушения.



На этой иллюстрации мы уже не видим пропорций золотого деления, а наблюдаем дисгармоничные тела существ. Уродство этих существ связано с несоблюдением гармонических законов природы при их создании.

На данных изображениях существ мы видим, что точка пупа не делит тела в соответствии с золотым (или среднестатистическим?) взаимоотношением частей и целого. Слева это отношение 1 : 1, справа 3 : 4 (или 0,75 : 0,25). На этом изображении ТОЛЬКО ОДНО отношение не соответствует золотой пропорции, так как трудно представить человекоподобное существо, у которого все части тела, органов относятся друг к другу, как 1 : 1. По всей видимости, это было бы существо кубической или сфероидной формы... что никак не похоже на человека.

Моделируя обычного человека, мы, скорее всего, не берем линейку и калькулятор, высчитывать золотые пропорции.

Мы просто интуитивно ощущаем эти формы, ибо формы человеческого существа попадаются нам на глаза чаще, чем что-либо другое, но создавая модель необычного существа, растения, сооружения, нам стоит использовать знания геометрии и золотого сечения, чтобы на результат работы можно было смотреть без отвращения, хотя... если вы добиваетесь как раз чувства отвращения, то вы знаете, что вы должны делать.

В любом случае, знание законов природы (числовых законов), помогает нам как можно быстрее достичь желаемого результата.

Глава 2. Основы реализма в 3D графике

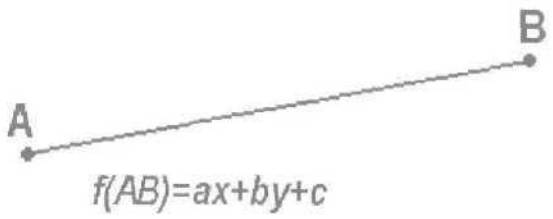
Независимо от того, какой редактор вы используете, вы не раз сталкивались с тем, что просчитанная сиена выглядит плоской, нереальной, даже походит на продукт деятельности чьих-то неумелых рук. Задумываясь над причинами проблем такого рода, вы, наверняка, приходили к выводу, что многие характеристики деталей не соответствуют своим реальным значениям. Текстура камня похожа на бумажную обертку с изображением камня, песок похож на мятую фотографию, за объектом источник света, а тень проецируется не туда, куда нужно... Список можно продолжать еще очень долго. Из всего этого можно сделать вывод, что сцепа состоит из множества мелочей, на первый взгляд неприметных, на второй — слегка портящих общую картину, на третий — раздражающих. Даже не зная причины, интуитивно чувствуются все несоответствия.

Реализм в 3D графике — максимальное приближение к действительности. Создавая сцену, мы создаем модель действительности. Реализм в графике достигается благодаря соблюдению законов природы. Эти законы обуславливаются абсолютными параметрами конкретной вселенной, как среды возникновения, существования и трансформации материи.

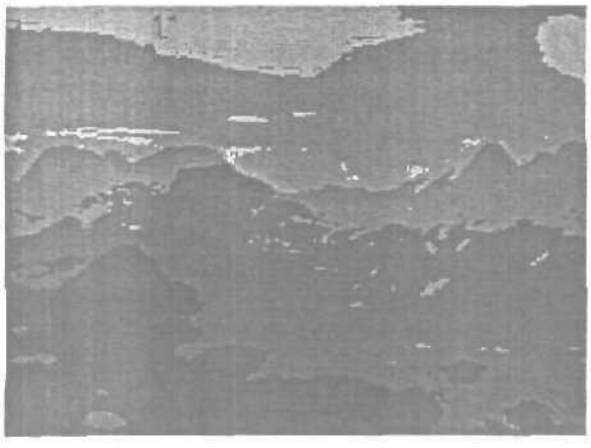
Основные составляющие реализма

Считается, что Вселенная сформировалась из первоначального Хаоса. Но и существующая Вселенная представляет собой все тот же Хаос, то есть, в обывательском представлении — беспорядок. Рассмотрим понятие «порядка», «беспорядка». Порядок — структура, в которой регулярно повторяется один и тот же элемент (одна та же закономерность), который мы называем элементом трансляции. Теперь представим себе, что этот элемент трансляции начинает увеличиваться и стремиться к бесконечности. Таким образом, мы пришли к Хаосу. Итак, Хаос — это порядок с бесконечным элементом трансляции. Мы к этому привыкли и интуитивно ощущаем, на генетическом уровне, но не обращаем на это внимания в силу ограниченности наших воззрений и привычных стереотипов.

В природе очень редко встречаются прямые линии, углы, и, как правило, ассоциируются с плодами рук человеческих. Поэтому прямая линия никак не ассоциируется с природными метаморфозами, то есть Хаосом; координаты множества точек строго подчиняются определенной функции.

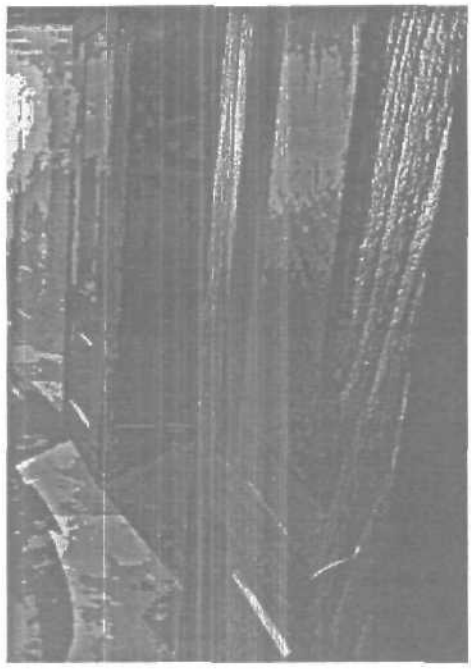


В природе же, не может существовать идентичность, так как такая возможность обратнопропорциональна квадрату количества иных возможностей. Количество этих возможностей стремится к бесконечности, так как каждый параметр можно рассматривать все глубже и глубже в зависимости от спонтанно выбираемой степени аппроксимации. Например, наблюдая парад планет с Земли кажется, что планеты действительно выстроились в одну линию, но при этом видимые размеры планет в миллионы раз меньше их реальных размеров, и если рассмотреть этот парад в определенный момент времени (а это очень важно, потому что Вселенная не статична) и рассмотреть так, чтобы оперировать координатами и размерами планет с относительной погрешностью, хотя бы, в один километр, мы соединили бы центры планет отрезками и никогда бы не получили прямую. Другой пример: горизонт с поверхности Земли кажется прямой линией, а на самом деле... Это и есть принцип относительности. Из сказанного выше следует, что всякое представление материи уникально. Любое повторение портит даже самую милую картинку: обратите внимание, как все портят две одинаковые горы в центральной части картинки.



Первый принцип реализма в графике — хаотичность и спонтанность

Каждый объект можно представить с различной степенью детализации, но число деталей можно уменьшать лишь до того момента, когда объект перестает быть узнаваемым. Каждый объект имеет свои параметры, которые в свою очередь подразделяются на параметры второго уровня, они же в свою очередь — на параметры третьего и так далее, в зависимости от того сколько мы хотим узнать об объекте. Теоретически мы можем узнать столько, сколько захотим, но нам редко нужно знать об объекте хотя бы столько сколько мы можем узнать. Этот принцип возник не просто так, он заимствован у природы. Важнейшие моменты фиксируются, запоминаются, ненужные — забываются и выкидываются. Детализация хоть и не является необходимым фактором, но является тем, что добавляет материальности объекту. Больше деталей — больше ассоциаций с реальностью, а как известно, человек мыслит ассоциативно. Плоские, нетекстурированные объекты выглядят нереально. Текстуры, растянутые в два и более раза искажают представления о самых пропорциональных объектах.

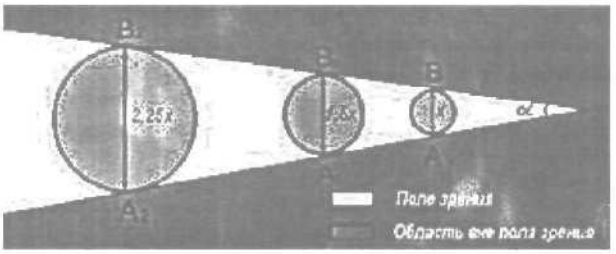


На этой картинке хорошо показан принцип аппроксимации и детализации. Если при меньшей аппроксимации, обусловленной большим расстоянием, одна и та же текстура для дальних дверей вполне подходит, то ближняя дверь выглядит просто ужасно из-за возникшей пикселизации. Закройте ближнюю дверь рукой и вы увидите, что эта картинка могла бы смотреться лучше.

Второй принцип реализма в графике — определенная детализация при соответствующей аппроксимации

Что есть изображение? Почему объекты различных реальных размеров могут иметь одинаковые видимые размеры? Как изменяется цвет и свет?

Ответ на все эти вопросы дает представление понятия зрения. Зрение не измеряет, а соизмеряет. Поэтому, если Солнце, Луна, футбольный мяч и горошина предъявлены зрению как одинаковые угловые величины, они вызовут активность одной и той же группы рецепторов, то и видимые размеры этих объектов будут одинаковы.



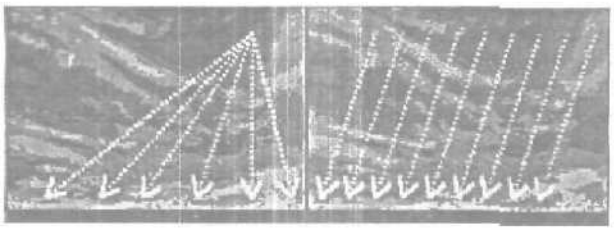
Именно это и показано на рисунке. Угол обзора, то что любители Quak'a привыкли называть FOV, в данном случае такой же, как и угловые величины трех шаров разного диаметра. С геометрической точки зрения эти шары подобны, у них коэффициент подобия 1,5. Градусные меры дуг подобных шаров равны, а длины дуг подобны. Из этого следует, что абсолютный размер не имеет для глаза и для мозга семантического значения. Отсюда понятно, что образ, как сумма размерно-пространственных характеристик выделяется по относительным размерным характеристикам — по соразмерности и пропорции. Характеристика объекта есть определенная соразмерность его частей и целого — пропорция, а неотъемлемой составляющей восприятия образа объекта в пространстве является его угловой размер. Таким образом, реальность и искусство соединены через соразмерность и пропорцию.

Из вышесказанного можно заключить, что факторы, обусловленные физиологией человека, такие, как перспектива, пропорция, являются важнейшими в построении 3D сцены, но поскольку во всех 3D пакетах эти факторы реализуются автоматически, мы рассмотрим то, как незнание этих факторов и, соответственно, неверный выбор методов реализации приводит к некачественному результату. Причиной, по которой мы можем видеть наше окружение, является свет. Именно потому, что все тела отражают свет, мы их можем видеть, исключая такие физические объекты, как черные дыры.

Подробней рассмотрим этот **процесс.**

Свет — электромагнитные волны, длины которых находятся в диапазоне от нескольких нанометров и до десятых долей миллиметра. Природа света дуальна. Именно в оптическом излучении отчетливо проявляются и волновые, и корпускулярные свойства электромагнитного излучения. Волновые свойства обусловливают дифракцию света — грубо говоря, явление огибания лучами света контура непрозрачных тел, интерференцию света — пространственное перераспределение энергии светового излучения при наложении нескольких световых волн и многие другие явления. Одновременно нельзя понять природу света не привлекая представление о свете, как о потоке быстрых частиц — фотонов. Это представление было сформулировано еще Ньютоном. Падающий на поверхность тела поток света частично отражается (reflection), частично проходит сквозь тело — преломляется (refraction), частично поглощается (энергия светового потока сообщается телу — absorption), частично отклоняется (diffraction).

По оптическим законам — яркость света (а следовательно, интенсивность цвета) обратно пропорциональна квадрату расстояния. Наиболее распространенной ошибкой пользователей является применение локальных источников освещения в глобальных масштабах, конкретно реализуемого, виртуального макромира. Например, многие пытаются освещать свои миры используя omni. К чему это приводит, вы можете понять, рассмотрев схемы распространения световых лучей:



Слева вы видите распространение лучей при использовании omni, то есть локального источника для глобального освещения. Справа — реальное освещение. Справа лучи идут не параллельно, между ними расхождение в тысячные доли секунды. Подобное распространение лучей объясняется тем, что в первом случае источник света во столько же раз меньше Земли, во сколько раз, во втором случае, Земля меньше Солнца. А теперь, представьте, как будут проецироваться тени в обоих случаях... Единственный случай, где допустимо применение локальных источников света, таких как omni, для освещения глобальных сцен — космос. Не всегда стоит эмулировать Солнце шаром, долго возиться с текстурой, ведь солнечная поверхность статически гранулирована. Но если начали так детально изображать этот объект, то надо тогда и протуберанцы эмулировать. Это дело не из простых. Рациональнее эмулировать Солнце в космосе, используя omni в качестве источника света и в качестве источника для мощного Lens Flare, а в качестве шара — gizmo.

Третий принцип реализма в графике — реальное освещение глобальным источником света

Кроме света у нас есть еще цвет и текстура. Нужно всегда учитывать, что кроме собственной окраски объекта есть еще множество факторов (например, блики, рефлексы), определяющих конечный воспринимаемый результат. Наиболее существенными факторами являются: свойства поверхности объекта, свойства среды, свойства источника света. В данном случае следует подробней изучить свойства поверхности. Мы видим объекты благодаря свету. Различные свойства поверхности позволяют световым лучам в различной степени отражаться, преломляться, поглощаться. Это обусловливает такие оптические свойства материала как матовость, отражение, блеск, прозрачность. Матовость или блеск определяется шероховатостью или гладкостью поверхности в микромасштабах. Если поверхность достаточно гладкая, то лучи отражаются приблизительно под таким же углом, под каким и падают. Поскольку шероховатая поверхность характеризуется множеством неровностей, то множество лучей, составляющих квант света, отражается не одинаковым образом, а под различными углами.

Каждый материал должен иметь определенную карту выгиба (bump), например, материал камня должен иметь карту выгиба, соответствующую текстуре, со всеми неровностями. В противном случае, различные трещины и неровности камня будут выглядеть неумелой раскраской. При создании анимации, в которой присутствует вода, не обязательно анимировать волны воды, достаточно анимировать карту выгиба волн.

Четвертый принцип реализма в графике — качественные материалы

Кроме линейной перспективы существует также цветовая и контрастная. Закон перспективы гласит, что интенсивность цвета обратно-пропорциональна квадрату расстояния. Объясняется это тем, что световые лучи преломляются в водяных парах, содержащихся в воздухе, а при прохождении границы раздела двух прозрачных сред различной плотности, световой луч меняет свое направление. Этим и объясняется явление преломления. Совокупность цветовой, контрастной и линейной перспективы дает правдоподобную картину рендеринга.

Явление перспективы ощущается вследствие анатомических особенностей глаз.

Пятый принцип реализма в графике — учитывание физиологии человека.

3D графика — достаточно новое направление. Лет десять назад мы и не знали такого понятия виртуальный мир. Слово «виртуальный» бытовало среди физиков-теоретиков в виде сочетания «виртуальные орбитали», где речь шла об электронах...

Фильм «Газонокосильщик» ввел нас в этот мир, ставший сегодня привычным. Теперь мы сами создаем свои миры — наши субъективные ощущения, переживания...

Тем не менее, человек с тех времен, как стал ощущать себя «гомо сапиенс», пытался выразить графически свои переживания, эмоции, наблюдения природы, характеризуемые одним понятием — гармония.

Гармония — общая закономерность в смысле качественного обобщения. Законы гармонии, законы природы — есть числовые законы. Поэтому необходимо использовать знания и опыт человечества для развития подходов при создании виртуальных миров.

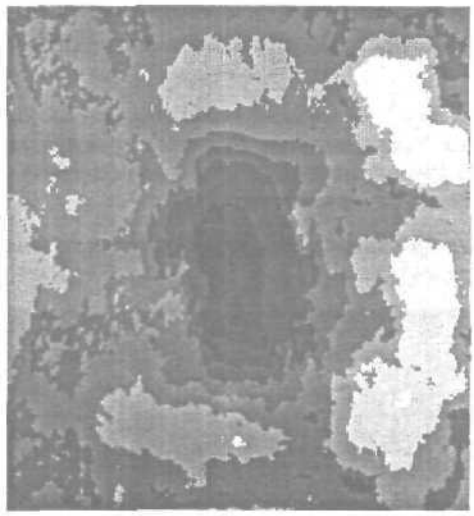
Практическая сторона

Учитывая все вышеописанные аспекты, у вас могут возникнуть только практические трудности. Все приведенные ниже советы применимы к 3D Studio MAX.

Для создания реалистичного ландшафта, привожу небольшую инструкцию:

1. Поскольку оптимальный способ получить реалистичный ландшафт — использование смещения (displace), вам стоит найти Z-карты нескольких гор. Z-карты — черно-белые вертикальные проекции объектов,

где высота объекта определяется интенсивностью. Конечно, их можно создать самому, но это нерационально. Карты можно получить из Bryce или из геологических баз данных. В 2D редакторе, позволяющем работать с объектами, например, Corel Photo-Paint или Adobe PhotoShop, сколлажируйте несколько карт, поменяйте пропорции карт и получите, что-то подобное рисунку.



2. Затем, если вы работаете в 3DS MAX R3 или выше, создайте плоскость (Plane), установите количество сегментов 30x30, а плотность — 6 или 7, в зависимости от размера карты и плоскости, если у вас мощный компьютер, то можно все 10 поставить.

3. Примените Displace. Высота холмов или гор будет зависеть от установленного значения. Включите опцию «центр совместимости»

4. Создайте материал: вместо цвета рассеивания возьмите текстуру камня, травы или скомбинируйте в одном большом материале и траву, и камень. Вы можете, также сделать подошву горы травяной, а вершины каменными. Для этого используйте тип материала «верх/низ» (top/bottom). Значение блеска поставьте очень малым (0-10). Примените карту смещения в качестве карты выпуклости (bump). Примените полученным материал к объекту Plane01.

5. Примените к объекту скаляризацию карты (mapscale).

6. В качестве неба используйте полусферу с диаметром, равным длине меньшей стороны вашей плоскости. Примените к ней материал

неба. Самоосвещение материала неба поставьте 100%. Текстуру неба создайте в зависимости от времени суток, которое вы хотите изобразить.

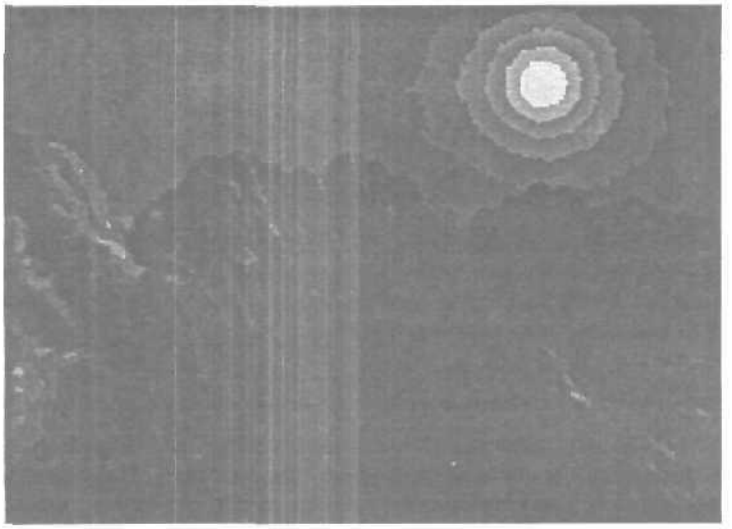
7. В качестве источника освещения используйте «солнце» (sunlight). Определите параметры «месторасположение», «год», «время года», «время суток».

8. Примените Lens Flare. В качестве источника возьмите «солнце».

9. Для большей глубины используйте Motion **Blur. В** качестве центра используйте «солнце».

10. Темное пятно на карте — впадина, ее можно чем-то заполнить: туманом, жидкостью или и тем, и другим. Создайте туман с линейной профессией плотности. Выберите слоеный тип тумана **(layered).** Остальные параметры выберите по вашему усмотрению.

В итоге, если вы все правильно сделали, вы можете получить картинку наподобие этой:



Все вышеописанное применимо и к 3D Studio MAX R2.5 и ниже, но вместо **Plane,** вам придется использовать NURBSurface, но в этом случае вы не сможете получить острых граней, или Box с большим количеством граней, но просчет будет производиться крайне долго по сравнению с Plane.

Глава 3. Основы рендеринга

Для многих людей слово «рендеринг» прочно ассоциируется с другим: «ожидание». Для небольшого же количества людей слово «рендеринг» означает комплекс математических операций с широким диапазоном возможных результатов.

Большинство 3D моделлеров, художников и аниматоров плохо представляют как проходит рендеринг, и чем занимается программа рендерер в процессе работы. Программа рендерер рассматривает 3D сцену с математической точки зрения, вычисляет, как должна выглядеть сцена и затем создает итоговое изображение. Мы все знаем, что разные программы используют разные технологии рендеринга — трассировка лучей (raytrasing), ретуширование (shading) и сглаживание контурных неровностей (antialiasing), для достижения максимальной похожести искусственного изображения на реальную жизнь. Мы также прекрасно знаем, что чем сложнее сцена и выше качество выходного изображения, тем дольше длится процесс рендеринга. Возможно, вы также слышали такой термин, как «потоковый рендеринг» (rendering pipeline). Но в сущности, мало кто задумывался, каким образом происходит процесс рендеринга и от чего зависит конечный результат.

Начнем с цифр — вы наверное не догадывались, но существует более 500 (пятьсот) различных программ для рендеринга. Одни из них встроены в пакеты 3D моделирования, другие поставляются как отдельные продукты, одни могут выполнять огромное число задач, другие созданы для специфических целей. Существуют рендереры, созданный специально для определенных индустрии (например, для военных целей), для определенного оборудования (Амига), для определенных операционных систем (Линукс), и других, иногда самых неожиданных целей (3D аудио). Кроме того, существует некоторое количество бесплатных рендереров, созданных энтузиастами.

Вы спросите зачем столько различных программ одинаковой направленности? Одна из причин — это то, что люди и компании создают подобные программы уже на протяжении 15 лет, кроме того, существует более дюжины платформ, способных поддерживать процесс рендеринга.

Посчитаем: если каждая платформа имеет 2-3 real-lime рендерера (в основном для игр), 2-3 non-real-time рендерера (коммерческие 3D пакеты), 2-3 специализированных рендерера (созданных для военной, либо другой индустрии), добавим сюда неопределенное количество свободно распространяемых рендереров (чаще всего написанных студентами,

хакерами и начинающими программистами) — нот мы и приблизились к реальному числу.

Совершенно необязательно знать про все эти системы рендеринга. Но в зависимости от работы, которую вам необходимо выполнить, вам возможно понадобятся некоторые знания, касающиеся различных программ рендеринга. Например, если вы разрабатываете сцены для игровой индустрии, вам необходимо знать возможности real-time рендерера, на основе которого будет работать игра.

Большинство рендереров делятся на две категории: real-time и non-real-time. Кроме того, все рендереры можно разделить по принципу используемых технологий — scan-line и raytrace. Мы рассмотрим основных представителей этих категории и их различия, чтобы вы смогли самостоятельно выбрать наиболее для вас подходящий.

Потоковый рендеринг (Rendering Pipeline)

Потоковый рендеринг основывается на четырех уровнях.

Все объекты 3D сцены четко определены в пространстве, до того, как вы помещаете объект в сцену, у него уже есть собственное геометрическое описание: длина, высота, ширина и глубина, но он пока никак не связан с другими объектами. В этот этап также может входить и процесс tessellation (когда обсчитываемые поверхности трансформируются в полигоны). Тут нужно быть осторожным, т.к. если вы используете NURBS и поверхности Безье — точность выполнения tessellation виляет на окончательные результаты.

Когда вы помещаете объект в сцену, он приобретает дополнительные геометрические описания, теперь у него есть точная позиция и ориентация по отношению к другим объектам сцены. На этом этапе назначаются параметры поверхностей объектов и устанавливается освещение. Также объектам могут быть присвоены анимационные ключи.

Когда в сцене появляется камера, все объекты приобретают новый параметр — ориентация и местоположение по отношению к камере. На этой стадии все полигоны проверяются на предмет видимости, полигоны, не видимые камерой игнорируются. Этот процесс называется отсеиванием невидимых поверхностей (backface cutting). Камере также могут быть присвоены анимационные атрибуты.

Далее, все объекты трансформируются с точки зрения конечного изображения. Сначала, в зависимости от настроек камеры (фокусное расстояния, поле зрения и т.п.) формируется визуальный коридор. Полигоны полностью за пределами визуального коридора игнорируются, лежащие на границе — усекаются. На этом этапе выполняется множест-

во визуальных трюков, чтобы компенсировать искажение перспективы (более удаленные веши кажутся меньше и т.д.).

Теперь мы входим в ту стадию, которую большинство людей считает собственно процессом рендеринга, хотя технически все вышеперечисленные этапы — это часть потокового рендеринга. Итак, обрезанные и корректно искаженные полигоны проецируются на плоскость (как будто на экран монитора). Каждый полигон преобразовывается в растровый формат, ретушируется, вычисляется глубина цвета. На этом этапе могут применены различные эффекты типа Antialiasing и **Motion Blur.**

Все эти этапы — это части «технического» рендеринга и большинство пакетов 3D моделирования позволяют выполнять их в реальном времени и приступать к финальной стадии только после нажатия кнопки **«RENDER».**

В то время, как начальные этапы потока довольно незамысловаты (чистая математика), последняя стадия может быть выполнена огромным количеством способов, приемов и трюков, призванных ускорить процесс, но в тоже время сделать сцену максимально реальной. К сожалению, многие приемы, используемые для ускорения процесса или достижения определенного визуального эффекта ведут к появлению проблем, которые решаются с помощью новых приемов и трюков (со своими проблемами) — и так далее.

Например, рендереры, основанные на принципе **Radiosity**, берут почти готовую сцену и анализируют количество света, отраженного от других поверхностей, с учетом изменения его спектра, этот свет в свою очередь отражается от поверхности и падает на другую... Эти вычисления повторяются снова и снова. Когда вычисления закончены, каждая поверхность окрашивается и ретушируется в соответствии с результатами вычислений. Главной проблемой такого способа является то, что для ограничения времени выполнения (иначе вычисления длились бы бесконечно), обсчет ведется в соответствии с определенным алгоритмом, но именно в результате этого некоторые поверхности могут быть выключены **из** общего процесса и окрашены некорректно. Тем не менее, такой способ рендеринга дает прекрасные результаты. Но **Radiosity,** который прекрасно подходит для неподвижных сцен, мало пригоден для рендеринга анимаций. В силу особенностей данного принципа, при рендеринге движущихся объектов возможно появление мерцания.

Scanline **рендеринг**

Существует два основных подхода к процессу рендеринга: **scanline и raytracing**. Что же представляет из себя **Scanline** рендеринг? Предположим, что нам необходимо получить изображение разрешением 300x200.

Программа проводит воображаемую линию через каждый пиксель, вычисляя, какие полигоны лежат на пути этой линии и вычисляет необходимый цвет пикселя, в зависимости от того, какие текстуры и цвета были назначены полигонам, встретившимся на пути этой воображаемой линии. Затем берется следующий пиксель, затем следующий — и так до конца.

Основной проблемой этого способа является отделение видимых полигонов от невидимых. В первых поколениях такого рода рендереров процесс вычисления полигонов начинался с самой дальней точки от зрителя и каждый новый полигон закрашивал предыдущий. Такой подход далек от идеала из-за множества ненужных операций. Чтобы решить эту проблему используется Z-буфер. Программа вычисляет все полигоны, лежащие на пути воображаемой линии и назначает каждому полигону Z-значение в зависимости от его удаленности от экрана. Когда настает время рендеринга, обсчитываются только полигоны с наименьшим Z-значением — остальные просто отбрасываются. Такой способ позволяет существенно ускорить процесс рендеринга, но имеет и существенные недостатки, которые будут рассмотрены ниже.

Raytrace рендеринг

Подобно Scanline рендереру, Raytrace рендерер начинает с вычисления воображаемой линии от одного пикселя (кроме того, существует инверсный рендеринг, когда вычисления начинаются от источника света и рендеринг использованием обоих принципов). Когда воображаемая линия встречает на своем пути полигон, случаются три вещи. Сначала вычисляется цвет и яркость на основе прямого освещения источниками света, затем вычисляются углы отражения и преломления (reflection/refraction). На основе параметров отражения/преломления данного полигона луч раздваивается и двигается по двум новым направлениям. Если любой из этих лучей встретит полигон, он раздваивается снова — и так до определенного момента, когда луч либо уйдет за пределы сцены, либо встретит источник света, либо количество отражений/преломлений достигнет определенного установленного числа (recursion level). Когда все лучи закончат свое движение, вычисляется окончательное значение пикселя.

Инверсные трассировщики лучей начинают вычисления от источника света, с огромного количества лучей света, отражающихся от объектов сцены, которые возможно пройдут через нужный пиксель и достигнут зрителя.

Наверняка вы представляете себе, с какой легкостью такие способы вычислений могут поглотить миллиарды вычислений и занять огром-

ное количество времени. Создатели таких рендереров обычно дают пользователю возможность точно указать, сколько отражений луча обсчитывать. Иногда пользователь может пользоваться только уже заранее установленными настройками типа good; better; best.

Выгодой такого способа является точность обсчета отражений и преломлений — свет отражается от зеркала либо проходит через воду именно так, как это происходит в реальной жизни. Такой способ рендеринга позволяет точно вычислять тени. Когда луч встречает полигон, программа пытается провести от этого полигона прямой луч к источнику света, если это невозможно — полигон затеняется.

Основные недостатки — огромное число вычислений и исключение из обсчета определенных свойств света (отраженное освещение). В природе нет абсолютно черных теней, т.к. весь мир наполнен отраженным освещением. Многие рендереры пытаются избавиться от этих недостатков двумя путями: они либо добавляют общее освещение (global illumination), которое освещает всю сцену, либо представляют все объекты сцены как отдельные источники света, способные излучать слабый свет.

Ретуширование

Один из наиболее сложных аспектов рендеринга — это алгоритм ретуширования. Разные методы ретуши могут применяться, чтобы увеличить либо качество, либо скорость рендеринга. Мы все знакомы с такими методами как Flat, Phong, Gouraud, но это далеко не все способы имитации вида поверхности. Каждая техника использует разные принципы и выдает разные результаты, каждая техника использует алгоритмы различной сложности.

Все алгоритмы ретуширования пытаются эмулировать то бесконечное число способов, какими луч света реагирует на поверхность в реальном мире. Отражения (reflections), отражение/поглощение света (transmission/absorbtion), рассеивание света (diffraction), преломление света (refraction), смешивание света различных источников (interference). Проще говоря, алгоритм ретуширования используется для вычисления вида поверхности при разном освещении. Самым простым алгоритмом является Flat shading — одноцветный полигон. Небольшим шагом в сторону улучшения качества работы этого алгоритма является его способность изменять оттенок цвета в зависимости от угла, под которым свет падает на поверхность. Таким образом мы можем произвести простейшую трассировку лучей — проводится воображаемая линия сквозь пиксель экрана пока она не натолкнется на полигон. Затем проводим следующую линию от этого полигона к источнику света и вычисляем получившийся угол. Когда угол вычислен, программа рендерер просто

применяет нужную формулу и вычисляет цвет пикселя. Но подобная техника не рассчитана на вычисление теней, т.к. она работает только с одним источником света и не может учитывать освещение данного полигона другими источниками света. К счастью, большинство рендереров решают эту проблему тем или иным способом и нам не придется напрямую сталкиваться с этой проблемой.

Real-Time против Non-Real-Time

Реал-тайм рендереры (основанные чаще всего на принципе scanline) созданы в первую очередь для скорости, а не для качества. Чаще всего их можно встретить в играх, различных симуляторах реального времени или в пакетах 3D моделирования. Основной выгодой использования принципа Scanline — это скорость, но для того, чтобы эмулировать различные визуальные эффекты она мало подходит. Для их реализации используются различные трюки. Чтобы эмулировать поведение света, сцена просчитывается в несколько проходов.

Например, если необходимо создать эффект объемного дыма, создается текстура дыма и накладывается на анимированную плоскую поверхность, но если нужно посмотреть на этот дым, скажем, сквозь оконное стекло, сцену необходимо просчитать дважды — первый раз для создания дыма, и второй — чтобы наложить на нее текстуру окна. Разумеется, если слишком увлечься созданием эффектов, то это приведет к существенной потере скорости.

Реал-тайм рендереры используют множество трюков для достижения высокого качества изображения при сохранении высокой скорости. Z-буфер уже не используется. Для определения невидимых объектов используется другая техника, основанная на иерархическом положении объектов в сцене. Эти принципы применяются в таких играх, как Doom, Quake и др. Текстуры, в основном те, которые используются для стен, обсчитываются заранее с разных точек зрения и сохраняются в отдельных файлах. Затем в реальном времени просчитывается позиция игрока и накладываются наиболее подходящие текстуры. Другие рендереры (как в Quake 3 Arena) используют другую технику Color by Vertex. Все эти техники призваны ускорить процесс рендеринга, но в тоже время существенно ограничивают художников и моделлеров в осуществлении их фантазии.

Если вы создаете 3D сцену для игры, либо для другой программы где будет использован реал-тайм рендерер, вы должны принимать во внимание все ограничения такого типа рендерера и знать способы обхода этих ограничений.

Не все Scanline рендереры работают в реальном времени. Рендереры основанные на этом принципе также доминируют в индустрии кино и телевидения. Всем известные Pixar's RenderMan и Electric Image's Camera основаны на принципе Scanline. Есть две причины, по которым эта система используется в этих индустриях, первая — высокий уровень фотореализма не так важен, т.к. кадр появляется на экране на доли секунды, и вторая это скорость. Когда мы имеем дело с теми разрешениями, которые используются в кино-теле индустрии, даже оснащенные супермошной техникой студии рендеринга работают на пределе, и использование технологии трассировки лучей может добавить многие месяцы работы.

Нон-реал-тайм рендереры, такие, какие используются в Softimage, NewTek LightWave или Discreet 3D Studio MAX, направлены на создания высоко фотореалистичных изображений (естественно теряя при этом в скорости). Все они в основном базируются на технологии трассировки лучей. Такие рендереры могут также выпускаться отдельно от пакетов моделирования. Рендереры, основанные на технологии трассировки лучей, это идеальный инструмент для достижения высококачественных изображений, особенно для неподвижных изображений, когда зрители могут тщательно рассмотреть работу и выявить возможные недостатки. Такие рендереры могут с легкостью обрабатывать сложные сцены с большим количеством источников света и использованием отражающих/преломляющих поверхностей.

Следует отметить, что большинство систем рендеринга на сегодняшний день используют обе технологии параллельно и занимаются трассировкой лучей только в том случае, если это необходимо (обусловлено присутствующими в сцене материалами).

Отличие программ рендеринга друг от друга

Мы уже выяснили, что работа всех программ рендеринга состоит из стандартных этапов. Чем же отличаются друг от друга разные программы? Было проведено исследование, чтобы выяснить, чем же отличаются программы рендеринга разных компаний. Большинство компаний, разумеется, не желает делиться секретами своих технологий, но кое-что все же удалось узнать.

Разумеется, каждая компания утверждает, что ее продукт уникален, и что недостатки качества изображения зависят от пользователя, не удосужившегося настроить программу и воспользовавшегося предустановленными настройками **(default).**

По заявлению одного из специалистов компании NewTek, люди просто привыкли, что разным продуктам присуще разное качество ко-

нечных изображений. Например продукты Alias и LightWave выдают более органичные изображения, тогда как Max лучше работает с такими материалами как пластик и металл. Все это результат лени, и нежелания тратить время на получение нужного результата. При определенных усилиях можно получить совершенно одинаковые результаты в разных программах.

Кроме того, каждая компания заявляет, что именно ее рендерер является самым быстрым. Но такие заявления очень трудно доказать (впрочем как и опровергнуть), так как для этого необходимо создать равные условия работы — а это довольно сложно, учитывая, что программы работают с разными пакетами 3D моделирования. Что можно сказать наверняка, так это то, что Pixar RenderMan и Electric Image Camera работают более быстро, чем их аналоги, основанные на трассировке лучей.

В методе трассировки лучей довольно мало способов ускорить процесс. Первый и главный способ — мультипроцессорная обработка сцены, такая функция встроена практически во все популярные пакеты 3D моделирования и рендереры. Большинство из них также поддерживают сетевой рендеринг. Все зависит от того, сколько вы готовы платить — LightWave распространяет эти функции бесплатно, в других пакетах необходимо покупать отдельную лицензию на каждую машину, участвующую в рендеринге.

По мнению многих профессионалов Maya обладает самыми лучшими инструментами для анимации мягких тел. Эти инструменты позволяют данной программе создавать очень реалистичную анимацию ткани и воды. Еще одна уникальная черта данной программы — интерактивный рендерер, позволяющий практически мгновенно видеть результаты работы, включая освещение, текстуры, линзы и другие эффекты.

Еще одни трюк этой программы, это то, что перед финальным рендерингом она разбивает сцену на блоки. Для каждого блока оценивается время рендеринга и если оно совпадает с желанием пользователя — блок отправляется на финальный рендеринг, если ориентировочное время рендеринга выходит за установленные рамки, блок разбивается на новые блоки — и так далее. Также Maya проводит оценку видимости объектов перед рендерингом и в процессе рендеринга эти объекты просто пропускаются.

Считается, что LightWave обладает одним из лучших рендереров. По заявлениям NewTek, их продукт является самым быстрым и точным (впрочем, так говорят все компании о своих продуктах). LightWave использует 96-битную глубину цвета. Новая версия пакета будет обладать способностью обсчитывать анимацию методом Radiosity (впервые на

рынке подобных продуктов). Эта программа также поддерживает мультипроцессорный и сетевой рендеринг и выборочную трассировку лучей.

RenderMan является основным рендерером кино-теле индустрии. Его основным преимуществом является скорость и способность обсчитывать сцены высокой сложности (с многими тысячами полигонов). Разработчики данной программы также гордятся качеством выполнения antialiasing и motion blur, точностью настроек камеры.

RenderMan — это scanline рендерер, поэтому для работы ему требуется набор шеидеров (shader) для создания реалистичных световых эффектов. Разработчики программы создали целый язык, на котором можно программировать шейдеры, но это довольно сложно и многие компании нанимают специалистов для программирования шейдеров. Разумеется Pixar обладает стандартной библиотекой шейдеров которую вы можете использовать, кроме того, существует много отдельных студий, занимающихся их созданием.

RenderMan — это только программа рендеринга без возможности моделирования. Вам придется создавать и анимировать сцены Б других программах, а затем передавать их в RenderMan для финального рендеринга. Для этой цели существуют дополнительные программы.

Electric Image Camera является аналогом RenderMan и также ориентирована на кино-теле индустрию. Ее показатели и принцип работы практически совпадает с RenderMan. Единственное различие — программирование шейдеров не является таким сложным процессом и требует лишь знаний C и C++.

3D Studio MAX — это рабочая лошадка 3D индустрии во всех областях: рендеринг, моделирование, анимация. Это, возможно, самая гибкая программа т.к. вся ее работа основана на плагинах. Таким образом вы можете добавить в нее те функции, которые вам необходимы и настроить программу под ваши нужды. В этом Max далеко обогнал всех своих конкурентов. Таким образом, если вам не нравится работа встроенного рендерера, вы можете с легкостью поменять его на дюжину других, которых он поддерживает.

Softimage Mental Ray использует комбинацию scanline и raytracing. Также есть возможность программировать свои собственные шейдеры (если вы знаете C или C++). Если такая задача вас пугает — существует множество уже готовых.

Что нас ждет в будущем

Какие же улучшения в области рендеринга нас ждут в будущем? Phil Miller, исполнительный менеджер Max говорит: «Скорость — это не-

обязательно следующий шаг. Мы уже знаем как повторить большинство тех образов, которые мы видим в реальной жизни. Мы можем повторить 95% эффектов реальной жизни, но каждый новый уровень реализма требует удвоения количества расчетов и при этом только слегка приближает нас к реальности. Возможно теперь, когда производительность процессоров так выросла, мы сможем осуществить это».

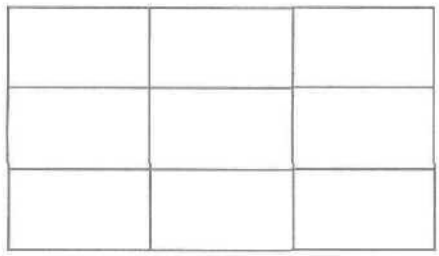
Brad Peeble, сотрудник NewTek соглашается: «В настоящее время процесс развития рендеринга идет в двух направлениях. Первое — реалтайм рендеринг, но с большим уровнем физического реализма и световых эффектов — это нас ждет в скором будущем, с ростом производительности процессоров нам уже не нужно будет имитировать поведение света — мы сможем точно рассчитать его. Второе — обогнать фотореализм. Мы хотим иметь возможность создавать сцены, которые будут выглядеть лучше реальности».

Мы рассмотрели все основные системы рендеринга без особого углубления в высшую математику (ни слова о трансформациях Фурье). К чему же мы пришли? Не нужно ждать слишком многого от рендерера и всегда нужно помнить, что лучший результат требует большего времени. Так что обратитесь к справочникам и руководствам по настройке, если вы хотите добиться нужного вам результата. Если вы работаете для ре;штайм рендерера — вам необходимо будет узнать огромное число специфических черт этой области рендеринга. Если ваша работа связана с кино или телевидением — вам придется разобраться в таких понятиях как Motion Blur, фокусное расстояние и, скорее всего научиться работать с шейдерами. Если вы создаете объекты или сцены, которые вы хотели бы сделать уникальными, то вам придется сделать больше, чем просто пользоваться стандартными установками рендерера. Если же вы просто делаете картинки для собственного удовольствия — возможно вам и не нужно ни в чем разбираться.

Глава 4.
Общие принципы построения композиции

1. В любой 3D сцене должна быть основа — смысловой центр композиции. Разумеется, в любой сцене также должны присутствовать второстепенные объекты и детали, но они не должны отвлекать внимание — смысловой центр должен быть выделен очень четко. Понятие «смысловой центр» совсем не означает центральное (территориально) положение в сиене, как раз наоборот — нельзя помещать основной объект в центре картины, в этом случае она будет выглядеть статичной и малоинтересной. Наиболее выгодным методом расположения смыслового центра яв-

ляется правило третьих частей — смысловой центр должен размещаться в местах пересечения линий. При этом действие предмета (движение, взгляд) должно быть направлено в центр картины.



2. Тщательно подбирайте угол обзора — выберите такое расположение камеры, которое позволяет видеть все аспекты сцены. При моделировании открытых сцен необходимо правильно подбирать расположение линии горизонта. Линия горизонта никогда не должна делить вашу сцену пополам, располагайте ее ближе к низу картины, чтобы подчеркнуть объем и сделать композицию визуально более вместительной; высокая линия горизонта делает композицию более конкретной, сосредоточенной на определенном предмете.

3. Максимально приблизьте камеру к смысловому центру композиции (это может быть либо объект, либо группа объектов) чтобы удалить все ненужные и малозначимые элементы. Крупные планы создают ощущение близости, в то время как вид с большого расстояния позволяет почувствовать глубину композиции.

4. Используйте линии движения внимания. Основные линии движения внимания (в роли их может выступать все: дорога, ограда, поток воды и т.п.) должны вести внутрь композиции. Такие линии можно назвать ведущими, т.к. они удерживают внимание зрителя внутри композиции.

5. Следите за фоном — фон может как улучшить посредственную композицию, так и разрушить прекрасно выполненную сцену. Правильно выбранный фон придет картине нужное настроение — неправильно выбранный — отвлекает внимание. Полезным приемом является расфокусировка фона.

6. Разнообразьте вашу сцену дополнительными деталями (ветка или дерево на переднем плане и т.п.). Подумайте сами, как можно оживить вашу сцену.

Помните, «композиция» — это всего лишь подбор и расположение предметов, входящих в изображение, но в тоже время является одним из важнейших моментов в создании изображений. Перед рендерингом мысленно оглядите всю композицию, проверьте, все ли совпадает с описанными выше правилами — это позволит вам добиться наилучших результатов.

Глава 5. Создание естественного природного окружения

Создание естественного фона, является пожалуй одной из наиболее сложных и одновременно важных задач при построении 3D сцены, и не только из-за сложности технического исполнения. В природе практически нет прямых линий, острых краев и тому подобных элементов, которые так легко создаются в 3D пакетах. Тот факт, что необходимо избавиться от всей этой линейности, делает моделирование природной среды существенно сложнее, чем моделирование индустриальных объектов и ландшафтов.

Конечно, моделирование индустриальных сцен тоже задача не из легких, их сложность состоит в построении огромного количества хаотично расположенных объектов и мелких деталей, таких как урны, металлические сетки, мусорные баки, крышки от бутылок на улицах, трещины асфальта и так далее. При моделировании природы нам этот мусор не нужен, но в этом случае мы сталкиваемся с необходимостью моделировать деревья, мелкие растения, камни. Правда нигде нет следов коррозии и разрушения, которые в изобилии встречаются в любой индустриальной сцене.

Вместо этого мы сталкиваемся с «природным хаосом», наполненным жизнью. Природный хаос, в отличии от индустриального, создан не человеком... из этого следует, что нам необходимо спланировать сцену так, чтобы она не выглядела спланированной (хорошенькая задачка!). Сцена не должна выглядеть систематизированной, иначе нам никогда не добиться фотореализма, при этом, как ни парадоксально, все же необходимо придерживаться определенного порядка, как раз для того, чтобы сцена выглядела натуральной- Довольно скользкий путь, но что делать...

Естественный мир очень детализирован, но при этом все его детали состоят из камней, деревьев и мелких растений. Разумеется есть еще и такие элементы как вода и грязь, но они встречаются только при моделировании определенных типов местности. Если подумать, то природа есть

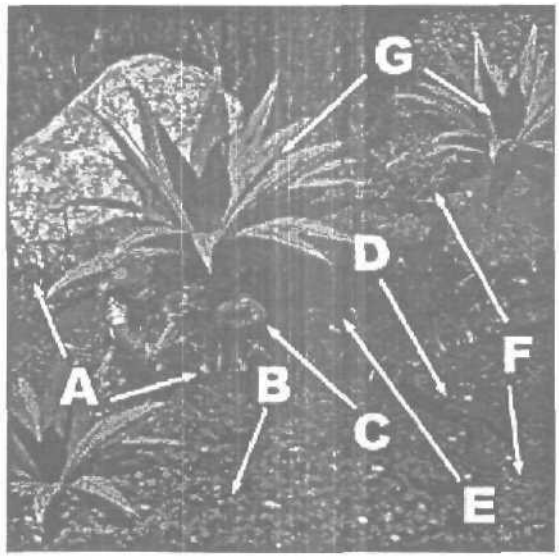
не что иное как несколько типов объектов, составляющих единое целое. Разумеется, деревья, камни и растения могут быть разных видов и размеров. Деревья в природе, как и здания в индустриальных сценах, составляют основу сцены, но не они, а более мелкие детали и предметы отвечают за ее реализм.

Сцены дикой природы (как и индустриальные) состоят из большого объема деталей, но в них мы не встретим грязных, потертых, испачканных маслом и подверженных коррозии поверхностей (я надеюсь, что не встретим). Кроме того, природа еще не так замусорена, как мир созданный человеком. Например, мы никогда не встретим листьев, забрызганных грязью, они могут быть пыльными, но уж никак не испачканными. Конечно в отдельных случаях бывает необходимо добавить некоторые атрибуты индустриального мира — но это отдельные случаи.

В первую очередь перед созданием сцены, мы должны оглядеться вокруг, понять, какие детали придают окружающему миру реальность. Я советую вам перед моделированием сцены изучить различные образцы реальной природы: фотографии, видео, телевидение либо самому выехать за город (или хоть в соседний парк сходить). Самое главное — определить те мелочи, которые позволять сделать вашу сцену более реальной — камешки, мелкие веточки и т.п.

Необходимо полностью сосредоточиться, если мы хотим добиться блестящего результата, давайте рассмотрим пример художественного построения дикой природы на рисунке.

Это кадр из мультфильма о гоблинах (Unfathomable Crag). Как вы можете убедиться, природа здесь показана очень реалистично, сцена наполнена мелкими деталями. Тут есть и случайно расположенные деревья и россыпь различного вида кустиков и еще множество деталей. Это по-настоящему детализированная сцена, хотя количество типов объектов в ней ограничено: дерево, камень, клевер на земле, кактусообразные растения и грязь. Если бы мы планировали индустриальную сцену, нам бы не понадобилось такое количество объектов, но ключом к реализму природных сцен является как раз наличие нескольких разных объектов одного типа, Давайте рассмотрим детали сцены подробнее, чтобы получить более полное представление об их роли в общей сцене. Разобьем сцену на части.



A. Кустики травы

Одним из наиболее часто встречающимся элементом живой природы являются кустики травы, разбросанные там и тут и растущие по краям любого статичного объекта, будь то камень или дерево. Это очень простой элемент для моделирования, но при этом его роль очень важна. Кустики травы добавлены повсюду и это делает сцену намного более реалистичной. Трава вообще является универсальным элементом для создания реалистичных картинок природы, при этом ее формы и размер могут варьироваться по желанию автора.

B. Травяной покров **земли**

Покров земли — это основа любой сиены природы. Он может быть различным, в зависимости оттого, какую природу мы хотим смоделировать. Травяной покров может быть густым или редким, может состоять из *обломков* веток и старых листьев, т.п. Единственное место, наверное, где мы его не увидим, это пустыня. Покров земли на рисунке, это прекрасный пример, как можно смоделировать работу самой Природы. Обратите внимание — покров не везде однородный, в одном месте он густой, в другом проглядывает земля. Различаются оттенки цветов и виды растительности. Такие приемы нельзя переоценить при моделирования природы. В природе трава никогда не растет равномерно, как в парках и на лужайках, созданных человеком, все это необходимо учитывать при проектировании сцен природы.

К различию в плотности, необходимо добавлять разнообразие размеров и ориентации деталей. На изображении это было достигнуто применением Фрактальной Перестановки **(Fractal Noise Displacement).** Вы должны всегда добавлять элемент хаоса, чтобы ваши сцены не выглядели слишком безупречными и гладкими.

C. Случайно расположенные камни

В дикой природе всегда полно различного размера камней, поэтому они являются прекрасным элементом, который поможет вам приблизить вашу сцену к реальности. Тем более, что они едва ли не самый простые и легкие объекты в моделировании. Текстурировать их не менее просто.

D. Голые **участки** земли

Земля закрыта травой не полностью. Открытые участки почвы также являются важной деталью в достижении фотореализма. В природе ничего не растет равномерно, т.к. природная растительность зависит от количества света, и плохо освещенные участки земли чаще всего бывают свободными от растений. На изображении мы видим, что мох покрывает большую часть земли, но в тоже время оставляя свободными некоторые участки. Эта деталь не бросается в глаза сразу, но служит для создания ощущения реализма всего изображения. Неплохо разбросать по этим участкам мелкую гальку, чтобы сделать текстуру земли менее однородной.

E. Корни растений

Корни растении не являются ключевым моментом сцен природы, но в некоторых случаях могут стать неплохой деталью ваших сцен и добавить некоторый беспорядок в расположение предметов. Не нужно злоупотреблять этой деталью, но поэкспериментировать стоит.

F. Мелкие растения

Дикая природа наполнена различного рода растениями, маленькими, большими — разных форм и оттенков, их можно встретить в любом месте. Поэтому ваши пейзажи всегда должны содержать несколько таких растений и обязательно разных видов и размеров. На нашем рисунке не так много места — большую часть занимает отверстие в земле, но несмотря на это на рисунке присутствуют несколько видов растений.

G. Расположение растений

Правильное расположение растений в сцене, является крайне важным шагом в достижении реализма. В природе растения растут там, где есть постоянный источник света и воды. Это совсем не значит, что где-то поблизости обязательно должен быть водоем. В природе, растения чаще всего растут рядом с большими объектами — деревьями, камнями, и в земляных углублениях, где почва остается влажной постоянно.

Как вы видите, для достижения реализма нужно не так уж много — основная задача состоит в расположении объектов и правильном освещении. Если суммировать все вышесказанное — вам нужно: несколько простых элементов, покрытие земли (трава или мох) различной плотности, несколько видов растений, камни и галька и ... все. Все, оказывается, достаточно просто. Главное сделать сцену разнообразной — растения разных видов и высоты, несколько типов травяного покрова и так далее.

На рисунке ниже показан еще один пример успешного построения сцены дикой природы. Данная сцена была выполнена с учетом всех принципов, описанных выше. Попробуйте сами проанализировать ее.

Ключ к успешному созданию органических мирон, это сосредоточение на мелких деталях, в изучении реального мира и в точном повторении в своих работах всех его деталей. Кроме того, прогулки по природе это не только хороший способ дать своим глазам отдохнуть от монитора, но и шанс понаблюдать за живой природой, понять принципы ее композиции и уловить ключевые моменты в расположении объектов. Прогулка с фотоаппаратом по лесу, будет прекрасным началом работы над созданием природы трехмерного мира. Только не надо пропускать мелких деталей, помните — чем больше деталей — тем больше реализма в ваших сиенах.

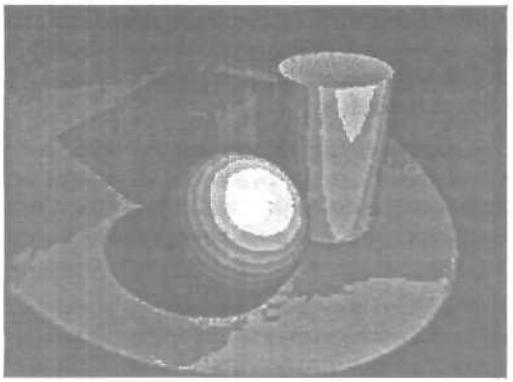
Глава 6. Ставим свет

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 о основных типах света, применяемого для имитации студийного освещения.

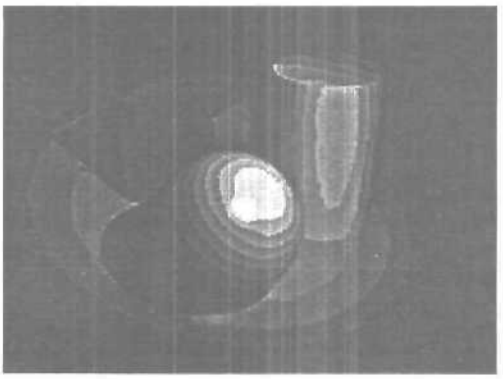
«Мягче, мягче...»

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 «fallow of».



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». Удивительно, почему до этого не додумаются сами программисты.

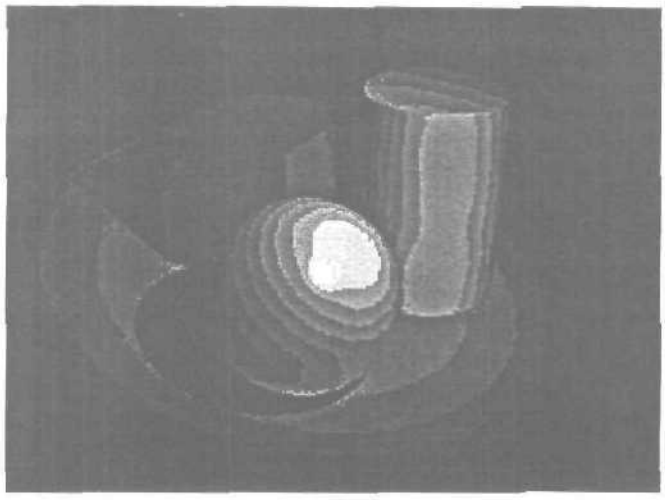


Рефлексируйте!

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 рефлексирующей поверхности (стола).

На рисунке ниже обратите внимание на цветной рефлекс, отбрасываемый кубом на боковую затененную сторону цилиндра.



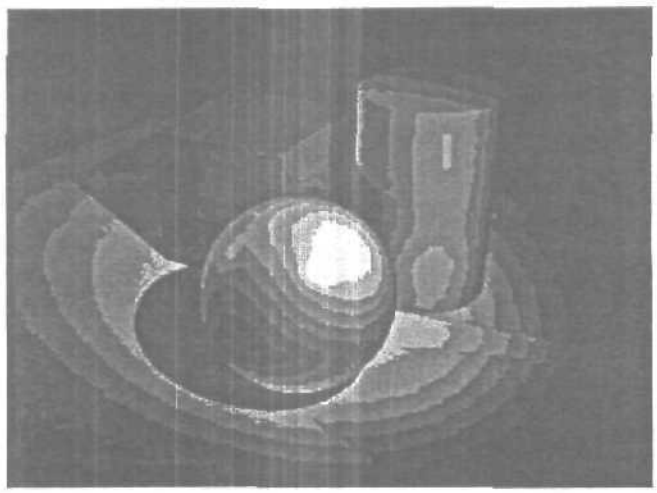
Долой серость!

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 (например, в 3D MAX) есть возможность задать цвет тени. В крайнем случае нужно просто подсветить теневые 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 4-х 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 с ужасом обнаружите, что к десятому кадру ваш центральный персонаж переместился так, что его освещение безнадежно испорчено. Что делать? анимировать еще десять ИС?

Избежать этого поможет простое правило. Максимально используйте возможности уже существующих в сцене ИС, прежде чем решитесь на создание нового. Для избегания «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 7. Принципы классической мультипликации

Видя многие трехмерные персонажи, ловишь себя на мысли, что это бездушные механические марионетки, настолько они отличаются от живых героев мультипликационных фильмов. Тем не менее превратить трехмерную модель в одушевленное существо совсем несложно. Достаточно познакомиться с 12 принципами Диснея и применять их в своих

работах. Хотя эти принципы написаны мультипликаторами и для мультипликаторов, их достаточно легко адаптировать к трехмерной графике.

Принципы Диснея были получены практическим путем, в их основу лег ежедневный опыт мастеров. Они настолько удачно были написаны и результат их применения был на столько эффективным, что принципы стали обязательным предметом изучения сначала для аниматоров студии Диснея, а затем и для мультипликаторов всего мирз.

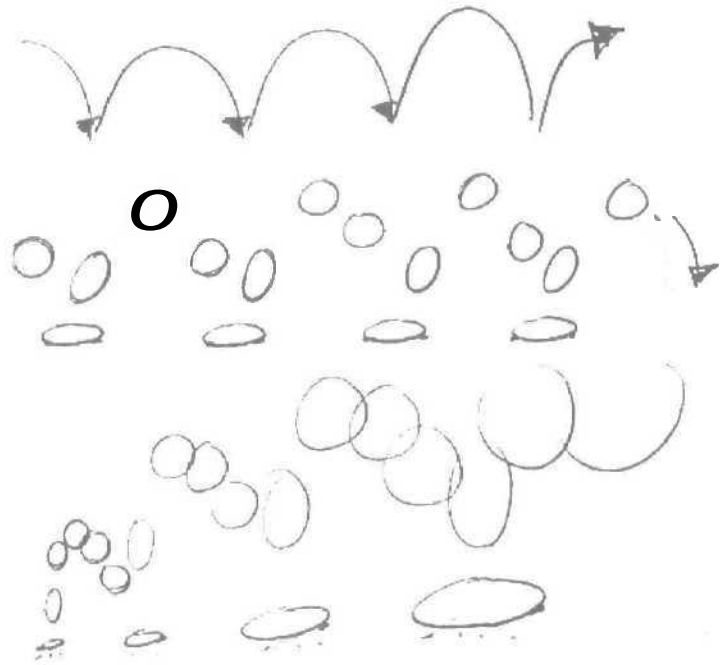
Вот они:

- сжатие и растяжение;
- подготовка, или **упреждение;**
- сценичность;
- рисование «прямовперед» и рисование «от позы к позе»;
- сквозное движение и захлест;
- медленный вход и выход;
- дуги;
- вторичные действия;
- timing, или расчет времени;
- **преувеличение;**
- «крепкий» (профессиональный) рисунок;
- привлекательность.

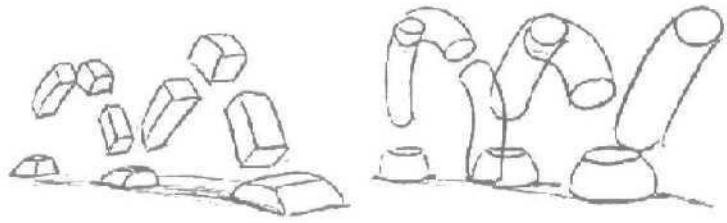
Сжатие и растяжение

Первый принцип мультипликации одновременно и очень прост, и очень важен. Он является основополагающим. С его помощью можно легко «оживить* персонаж, создать иллюзию естественности движения на экране. Основывается он на том, что любое живое тело (да и многие неживые предметы) при движение постоянно то сжимаются, то растягиваются.

Этот принцип легко объяснить на примере прыгающего мяча. Посмотрите на рисунок ниже. Обратите внимание, как шар сжимается в нижней точке каждой дуги в момент столкновения с поверхностью. В следующий момент (в момент отскока) шар растягивается. Эти сжатия и растяжения делают движение естественным, правдоподобным. Они объясняют опытному человеческому глазу причину, по которой мяч прыгает.



Те же манипуляции можно проделывать с другими простыми фигурами.



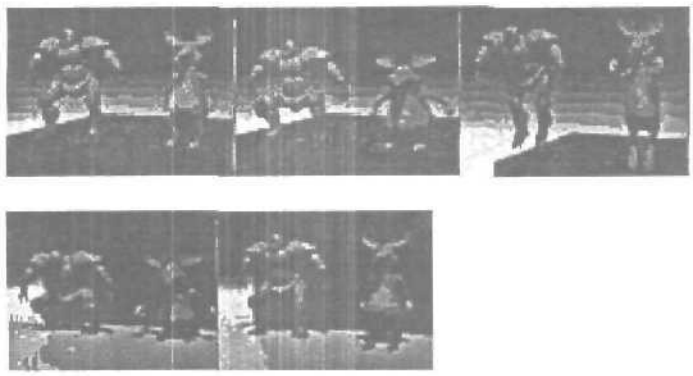
Точно так же нужно анимировать персонажей. Все отличие в более сложных формах.

При использовании этого метода необходимо придерживаться одного правила — объем объекта должен оставаться постоянным. Вертикальное растяжение должно компенсироваться горизонтальным сплющиванием, и наоборот.

Мультипликаторы приводят удачный пример с мешком муки, — при любом броске он меняет форму, но количество муки в нем остается одно и то же. Наверное, поэтому мультипликаторы любят рисовать и практиковаться на классическом «мучном мешке».

Этот метод можно и нужно использовать в 3D сценах. Для того, чтобы ваши персонажи не казались механическими марионетками, окаменелыми изваяниями; чтобы зритель поверил в происходящее на экране, стал сопереживать вашим героям.

Вот пример, использующий этот принцип. Два персонажа прыгают с подставки на землю. Один персонаж — робот, который представляет собой твердое тело, а второй персонаж эластичный и живой — мышка. Принцип сжатие-растяжение работает для обоих персонажей. Но для робота сжатие-растяжение практически незаметно, т.к. металл упруг и плохо поддается сжатию. А прыжок мышки — классический диснеевский принцип сжатия-растяжения. Мускулы и жир, из которых состоит этот персонаж, очень эластичны и легко меняют форму.



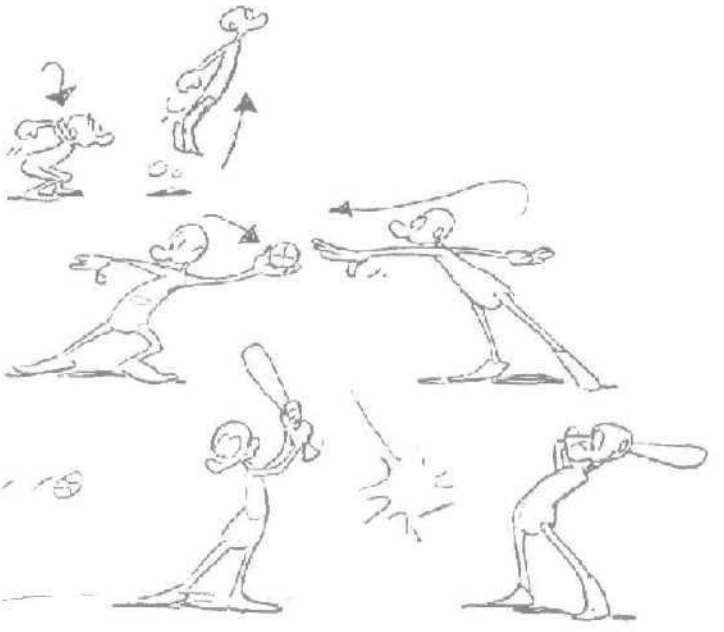
Смело используйте этот метод в своих сценах, небольшие сжатия-растяжения придадут естественности вашим героям. Если же усилить сжатия-растяжения, то можно придать персонажам более мультяшный вид. Изменяя степень сжатия-растяжения можно показать эластичность объекта, изменить настроение сцены и многое другое.

Пробуйте и экспериментируйте!

Подготовка, или упреждение

Вы конечно знаете, что перед тем, как персонаж подпрыгнет, он должен присесть. Наверное, на Луне это лишнее, но на Земле приходится бороться с гравитацией. Поэтому, вначале персонаж должен присесть, и только потом подпрыгнуть. Как правило, каждому действию предшествует Подготовка. Именно Упреждение информирует зрителя о том, что сейчас должно произойти. Зритель смотрит на замахивающегося персонажа и понимает, что через мгновение будет бросок мяча. Достаточно

показать предвкушение удара битой по мячу, и зрителю уже можно не показывать сам момент удара.

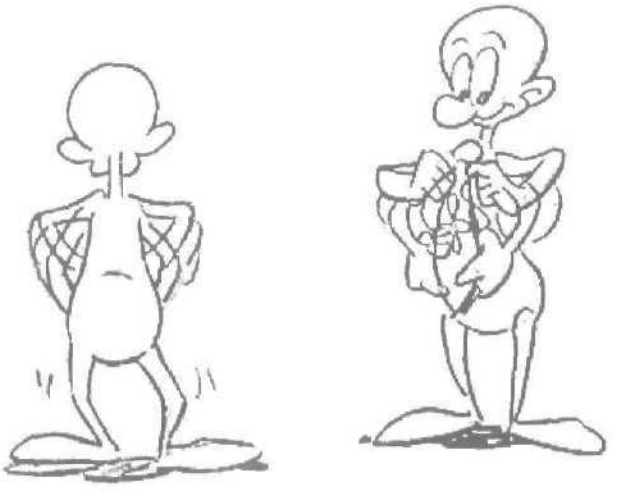


Посмотрите на пример использования этого принципа в 3D. Чтобы забросить тяжелый мешок в машину, человек вначале отклоняется в противоположную сторону, и только затем бросает.

Этот же принцип нужно использовать при анимации рук, ног, головы персонажей. Например, чтобы повернуть голову в сторону, вначале необходимо немного отвести ее в противоположную сторону.

Сценичность

В любой истории важен сюжет и то, как он показан. Действие на экране рассчитано на зрителя, и все происходящее должно быть предельно ясным, понятным и узнаваемым. Например, что делает парень, нарисованный слева? Не совсем понятно. Но стоит его развернуть, как ответ становится очевидным. Он завязывает галстук.



Выражение лица сценично, если оно хорошо читаемо, настроение персонажа сценично, если оно воздействует на зрителя. Характер персонажа должен быть узнаваемым, детали — хорошо заметными, реплики — разборчивыми, текст — доходчивым и т.д.

Движение персонажа не должно скрадываться одеждами, или смазываться неверным выбором угла зрения, или оттесняться на второй план чем-то другим.

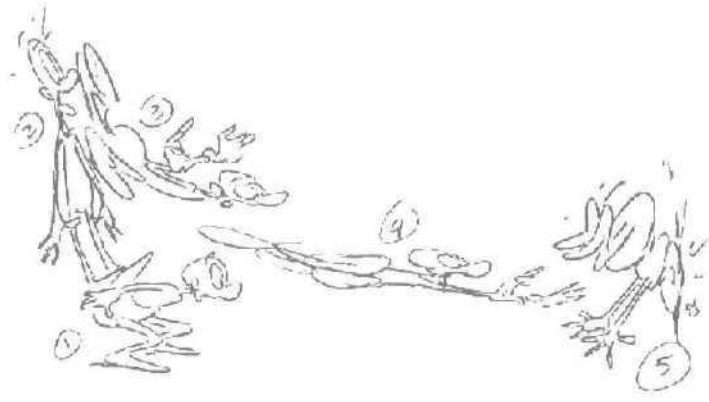
Это в 2D... Л что же в 3D? Все то же самое. Это означает, что каждый кадр должен ясно описывать происходящее на нем. Сделайте пробный рендер и убедитесь, что зритель получит всю нужную для него информацию.

Удостоверьтесь, что вид с камеры подобран правильно, что фон не отвлекает внимания от главного действия, герои хорошо читаемы, а их действия понятны. И только после этого продолжайте.

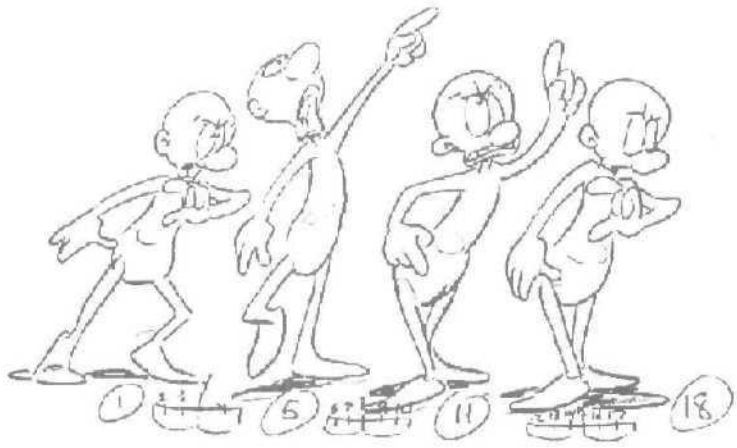
Рисование «прямо вперед» и рисование «от позы к позе»

Существует два подхода при создание мультипликации — прямо вперед и от позы к позе.

Рисование «прямо вперед» означает, что вы рисуете кадр за кадром. Этот метод обычно применяется при создании очень активных (быстрых) сцен.



Рисование «от позы к позе» предполагает, что вы рисуете ключевые позы (фазы) для всей сцены, а потом возвращаетесь и дорисовываете кадры между этими позами. Этот метод самый распространенный в мультипликации.



Этот метод 100%-но знаком всем, работающим в 3D. Рисование «от позы к позе» — это классическая работа с ключевыми кадрами. Но в очень быстрых сиенах рекомендовано использовать метод «прямо вперед», чтобы получить импровизационную анимацию с элементом неожиданности и новизны.

Часть 2. Знакомство с новой версией

Глава 1. Итак, 3ds max 6

В середине октября компания Discreet объявила о начале продаж популярного редактора для работы с трехмерной графикой и анимацией 3ds max 6.

CAD-овский приемник практически полностью завоевал рынок так называемой архитектурной визуализации после того, как был полностью переписан в середине девяностых. Совершенствование пакета продолжаюсь постоянно, по мере того, как росли потребности 3D рынка. Новая версия 3ds max содержит богатый арсенал средств, позволяющий создавать и визуализировать трехмерные сцены. Она включает в себя множество новых долгожданных инструментов, многие из которых были знакомы пользователю как подключаемые модули (плагины).

Глава 2. Поставка

Полная версия 3ds max 6 поставляется на трех дисках. Первый содержит инсталляцию программы, полную документацию (Reference Manual), описание нововведений (New Features Guide), а также программы, необходимые для работы с 3ds max — Direct X 9, Internet Explorer 6, Quick Time 6.1, Autodesk Licence Manager.

На втором диске находятся демо-версии сертифицированных плагинов — Cebas finalRender Stage-1, Cebas finalToon, Sitni Sati AfterBurn3, Sitni Sati DreamScape2, cgCharacter Absolute Character Tools V1.6, Mankua Kaldera и Kaydara HumanIK. Напомним, что программа сертификации плагинов состоит в том, что Discreet заключает соглашение с разработчиками дополнительных модулей о том, что новые версии дополнительных модулей выйдут сразу же после начала продаж нового релиза 3ds max.

Также второй диск содержит полнофункциональные версии плагинов компаний-партнеров Discreet. Среди них — Polytrans, OpenFlight Import/Export, Bionatics EASYNat и RPC.

На третьем диске находятся файлы примеров, бесплатные скрипты и анимация.

Глава 3. Интерфейс

Изменения в интерфейсе можно заметить сразу после загрузки программы. В левой части окна 3ds max добавлена вертикальная панель инструментов модуля для просчета динамики **Reactor** 2. Панель инструментов **Tab Panel** с многочисленными закладками теперь отсутствует. Таким образом разработчики постарались сэкономить для пользователя рабочее пространство. Все настройки этой панели доступны через Главное меню программы и командную панель (Command **Panel).** Однако, в случае необходимости **Tab Panel** можно вернуть на прежнее место. Для этого нужно подгрузить файлы интерфейса (*.CUI) предыдущей версии 3ds max.

Новая панель Extras содержит ниспадающее меню **Render Presets.** С его помощью можно быстро изменять настройки рендеринга, не вызывая диалоговое окно **RenderSccne.**

Была упрошена панель инструментов **Layers.** Это связано с появлением диалогового окна The Layer **Manager,** которое заменило окно настроек **Layer Properties.**

Диалоговые окна **Environment и Rendering Effects** объединены в одно окно **Environment And Effects,** которое содержит две закладки.

Глава 4. Объекты

В связи с тем, что компания Autodesk (подразделением которой является Discreet) объявила о прекращении дальнейшей разработки программы 3D Viz, некоторые функциональные возможности этого пакета были перенесены в 3ds max 6. Одним из таких нововведений стали группы объектов AEC **Extended, Doors** (двери), **Windows** (окна) и **Stairs** (лестницы).

В четвертом и пятом релизах 3ds max, в отличие от более ранних версий, отсутствовали такие необходимые для архитектурной визуализации объекты, как окна и двери. Этот недостаток можно было устранить подключением бесплатных плагинов Doors и Windows, которые разрабатывал сам Discreet. В шестой версии окна и двери снова были добавлены в список объектов 3ds max. Настройки этих объектов совпадают с настройками вышеупомянутых плагинов для 3ds max 5.

Группа объектов Doors позволяет создать три типа дверей — Pivot, Sliding и BiFold. Первые напоминают обычные входные двери, вторые — двери купе, а третьи — автобуса. Пользователь имеет возможность создавать парные и одинарные двери, регулировать размер дверной рамы, самих объектов и даже толщину стекол, если таковые имеются. Параметр Open позволяет указать, насколько двери открыты или закрыты.

Группа объектов Windows позволяет добавлять в сиену шесть типов окон — Awning, Fixed, Projected, Casement, Pivoted, Sliding. Их основное отличие — в способе открытия.

- Sliding Window — «отъезжает в сторону», подобно раздвижным стеклам на книжной полке;
- Pivoted Window — открывается таким образом, что оконная рама вращается вокруг своей горизонтальной оси;
- **Awning Window — поднимается вверх;**
- Casement — самый распространенный тип окна. Открывается подобно двери;
- Projected — состоит из нескольких частей, открывающихся в разные стороны;
- Fixed — вообще не открывается.

Следующая группа объектов Stairs также является необходимым инструментом для проектирования архитектурных сооружений. В 3ds max 6 можно создавать четыре типа лестниц: прямую (Straight Stair), винтовую (Spiral Stair), лестницу L-типа и U-типа.

Объекты Stairs могут быть открытыми, закрытыми и с основой. Отдельно регулируется наличие перил с правой и левой сторон, их высота и расположение относительно ступенек, глубина и ширина ступенек. Для спиральной лестницы дополнительно указывается радиус и направление (по часовой стрелке и против часовой), наличие или отсутствие опоры.

В группу AEC Extended входят объекты Foliage (растительность), Railing (ограда, перила) и Wall (стена). Railing и Wall, как и описанные выше объекты, применяются в архитектурном моделировании. Если настройки объектов Railing и Wall достаточно просты, параметры объекта Foliage требуют более детального рассмотрения.

Моделирование растительности в 3D сопряжено с большими трудностями. Для того чтобы созданное дерево выглядело реалистично, необходимо не только подобрать качественную текстуру, но и смоделировать сложную геометрическую модель. Таких моделей до сих пор в стандартном инструментарии 3ds max не было. Для создания растительности использовались разнообразные плагины — **TreeStorm,** TreeShop, Druid и пр. В 3ds max 6 появился свой инструмент для моделирования флоры.

При помощи Foliage можно создавать растительные объекты, которые загружаются из библиотеки **Plant Library.** Создаваемому объекту автоматически назначается свой материал. Для того чтобы деревья и кусты не были похожи один на другой, используется параметр **Seed,** который определяет случайное расположение веток и листьев объекта.

Проблема сложной геометрии листьев решена таким образом, что все листья и цветы представлены в виде плоскостей, к которым применяется текстура с использованием карты прозрачности. Таким образом, в процессе визуализации изображения ненужные участки плоскости становятся невидимыми, и листья приобретают нужную форму.

Раз уж речь зашла о растительности, то нельзя не упомянуть плагин **Bionatics** EASYNat, полнофункциональная версия которого входит в поставку полной версии 3ds max 6. Этот плагин добавляет в группу объектов Bionatics новый объект **EASYNat,** а также встраивается в главное меню 3ds max. Работа этого дополнительного модуля напоминает объект **Foliage,** однако, функциональных возможностей у плагина значительно больше.

При помощи **Bionatics EASYNat** можно не только моделировать всевозможную растительность, но и имитировать рост растений. Кроме этого, растения могут иметь разный вид в зависимости от времени года. Плагин содержит в прилагаемой библиотеке четыре растения. Остальные библиотеки можно докупить отдельно на сайте Bionatics.

Для создания трехмерной модели существует много приемов - NURBS-моделирование, полигональное, при помощи булеановских операций и т.д. В 3ds max 6 разработчики добавили новый тип объекта Blob Mesh, который открывает перед пользователем возможность создания трехмерных тел при помощи метаболов. Этот объект расположен на

командной панели в группе **Compound** Objects. Работать с метаболами можно двумя способами.

Первый способ привычен для тех, кто уже работал с метаболами — поверхность составляется из отдельных объектов. Второй же способ состоит в том, что любой объект можно преобразовать в метаболический. При этом каждая вершина преобразованного объекта будет обладать свойствами метабола. **Blob Mesh** удобно использовать вместе с модулем для работы с частицами **Particle Flow.**

Глава 5.
Модификаторы

Новый модификатор **Shell,** который можно найти в списке **Editable** Mesh на закладке Modify командной панели, воздействует на оболочку **Editable Mesh,** придавая ей толщину. Этот модификатор можно применять к полигональным, **patch** и NURBS-поверхностям.

Подвергся изменениям модификатор **Vertex Paint.** Теперь он позволяет рисовать на отдельных слоях и смешивать их, составляя новую цветовую гамму (до 99 каналов (map channels».

Кисти (**Brush**), при помощи которых пользователь может, например, раскрасить губы трехмерного персонажа, реагируют на чувствительность виртуального надавливания. Рисование кистью происходит с использованием той же технологии, которая задействована в работе модификатора Skin. То есть, пользователь полностью управляет размером и силой виртуальной кисти, а также может настроить ее профиль. В окне настроек Painter **Options** с помощью кривой можно создать профиль кисти, который ему необходим. В случае надобности, например, при раскрашивании глаз трехмерного существа, можно использовать режим симметричной модели.

Глава 6.
Particle Flow

Одним из главных нововведений 3ds max 6 можно считать модуль для работы с частицами Particle Flow. Этот модуль помогает создать практически любой эффект, связанный с частицами — брызги воды, разбивание объекта на фрагменты, сноп искр и пр.

Particle Flow — это совсем новая разработка Discreet. Он был опробован как плагин к 3ds max 5.1 и завоевал немалую популярность у пользователей.

Работа Particle Flow базируется на событийно-управляемой модели. До его появления такой принцип использовался только дополнительными модулями 3ds max — Cebas Thinking Particles и Digimation Particle Studio. Теперь же пользователь может проделывать эффекты с частицами при помощи стандартного инструментария 3ds max.

Настройки Particle Flow на первый взгляд могут показаться сложными, однако, принцип работы настолько продуман, что разобраться с ним можно довольно быстро. Дли того чтобы понять, как работает Particle Flow, необходимо иметь базовые знания о структуре модуля.

Интерфейс Particle Flow представлен окном Particle View. Его можно вызвать при помощи одноименной кнопки свитка Setup настроек объекта PFSource. В окне Particle Flow представлена диаграмма событий, описывающая системы частиц. Каждое событие (event) состоит из группы операторов (operator) и критериев (test). Стрелки на диаграмме показывают направление протекания события (например, от события 2 к событию 3).

Каждый оператор, который может находиться в отдельно взятом событии, определяет поведение частиц. Он может повлиять на форму частицы, скорость, размер, материал и на другие характеристики. Критерий служит связующим звеном между двумя событиями. Он определяет условие, при котором будет возможен переход от одного события к другому.

Например, критерий Speed Test работает таким образом, что переход к следующему событию возможен только для тех частиц, которые достигли определенной скорости. Таким образом, если следующее событие содержит оператор, определяющий другую форму частиц, в процессе анимации частицы, удовлетворяющие данному критерию, будут изменять форму. Выделив оператор или критерий в списке события, можно управлять его параметрами, которые отображаются в правой части окна Particle View.

Весь перечень доступных операторов и критериев располагается в нижней части окна Particle View. Для того чтобы добавить какое-нибудь действие в событие, необходимо просто перетащить иконку оператора или критерия на рабочую область диаграммы. Если перетащить действие на пустую область, будет создано новое событие.

Для того чтобы указать направление протекания событий, необходимо захватить курсором выступ на диаграмме события (он находится напротив критерия) и перетянуть на условную мишень в виде кружочка в верхней части второго события.

Когда курсор изменит форму, кнопку мыши можно отпустить. События будут связаны, на что будет указывать соединяющая стрелка междуними.

Каждое событие можно сделать неактивным. Для этого нужно нажать на лампочку в правом верхнем углу события. Ни один из операторов отключенного события не влияет на поток частиц.

В трехмерной сцене, *к* которой используется сложный эффект с частицами, диаграмма Particle Flow может выглядеть громоздкой. Для того чтобы работать с такой диаграммой событий было удобнее, можно использовать инструменты для управления видом в окне событий, расположенные в правом нижнем углу окна Particle View.

Глава 7. Материалы

В редакторе материалов 3ds max 6 появился новый тип материала Architectural. Этот материал, как и некоторые другие новшества, изначально разрабатывался для системы визуализации VIZ-рендер.

С помощью Architectural вы можете быстро создавать реалистичные материалы высокого качества. Трехмерные модели, к которым применен данный тип материала, обладают физически правильными свойствами. Они позволяют добиться реалистичного изображения при условии использования источников света Photometric Lights и системы просчетаосвещенияGlobalIllumination.

Еще одно нововведение в 3ds max 6 связано с появлением фотореалистичного рендерера Mental Ray. На его настройках мы остановимся ниже, а пока рассмотрим изменения, которые он вносит в редактор материалов.

Если в настройках 3ds max в качестве системы визуализации выбран Mental Ray, у пользователя появляется возможность работать с тремя дополнительными типами материала — Mental Ray, DGS и Glass.

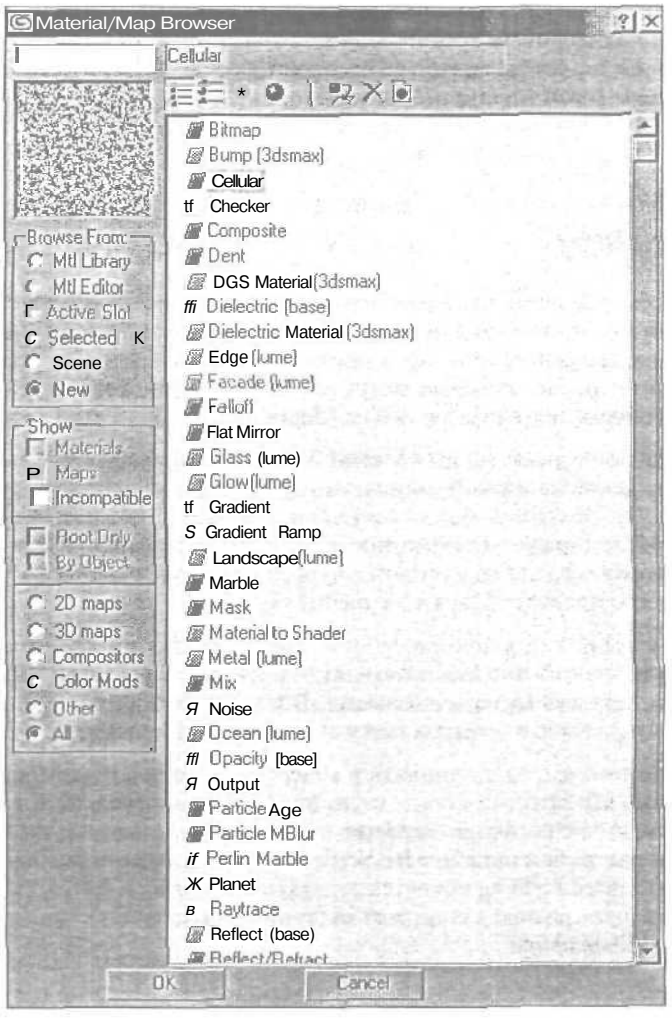
Первый тип материала — Mental Ray — состоит из шейдера поверхности (surface) и девяти дополнительных шейдеров, определяющих характеристики материала.

Материал DGS управляет цветом лучей, рассеиваемых материалом (diffuse), формой блика (glossy) и силой отблеска (specular).

Тип Glass позволяет управлять основными настройками материала «стекло».

Нужно отметить, что понятие шейдера для материалов Mental Ray значительно шире, чем для Scanline Renderer. Шейдером в 3D-моделировании принято называть алгоритм, определяющий поведение отраженных лучей света от поверхности.

Для Mental Ray же понятие «шейдер» означает определенный алгоритм визуализации изображения. Поэтому если вы используете в качестве визуализатора **Mental Ray.** вы можете работать с дополнительными шеидерами, аналогично тому, **как вы** работаете **с текстурными** картами 3ds max. В диалоговом **окне Material Map Browser** шейдеры **Mental Ray** будут представлены желтыми **иконками** (напомним, **что иконки** стандартных текстурных карт зеленого цвета).



Список шейдеров в окне Material Map Browser может изменяться в зависимости от того, какому параметру назначается шейдер. Например, если попытаться назначить шейдер в качестве параметра Contour материала **Mental Ray,** будет доступно девять шейдеров. Если же назначать шейдер в качестве параметра **Bump,** можно увидеть только три доступных шейдера.

Когда вы используете стандартный метод визуализации или любой другой рендерер, кроме Mental Ray, в редакторе материалов шейдеры визуализатора обычно отображаются в виде темных и светлых пятен или вообще не отображаются. И напротив: если используется Mental Ray, в сцене будет корректно визуализировано большинство стандартных материалов и текстурных карт 3ds max.

Глава 8. Mental Ray

Несмотря на то, что аппарат рендеринга 3ds max постоянно совершенствуется, он давно уже перестал отвечать запросам современной 3D-индустрии. На рынке 3D-софта появилось несколько конкурентоспособных рендереров, которые могут заменить **Default** Scanline **Renderer.** Одним из таких визуализаторов был **Mental Ray.**

В новой версии 3ds max Mental Ray интегрирован в программу как рендерер, альтернативный стандартному алгоритму визуализации. Напомним, что Discreet — не единственная компания, которая заключила сделку с Mental Images. Еше в конце прошлого года компания Alias Wavefront (которая сегодня известна уже просто как Alias) объявила об интеграции своего продукта Maya 4.5 с mental ray 1.5.

Внешний вид диалогового окна **Render** Scene претерпел изменения. По умолчанию оно имеет пять закладок — **Common, Render Elements, Raytracer, Advanced Lighting и Renderer.** В зависимости от того, какой тип визуализатора выбран, эти закладки могут изменять название.

Для того чтобы установить в качестве визуализатора Mental Ray, необходимо вызвать диалоговое окно **Render** Scene, перейти на закладку Common и в свитке Assign **Renderer** выбрать соответствующую строку. При этом настройки окна **Render** Scene изменятся. Вместо закладок **Raytracer и Advanced Lighting** появятся закладки **Processing и Indirect Illumination.** Последняя закладка содержит настройки каустики, а также параметры Global **Illumination.**

Чтобы Mental Ray просчитывал сцену с максимальной точностью и скоростью, нужно использовать источники света MRAreaOmni и MRAreaSpot.

Однако, он корректно просчитывает освещенность сцены, даже если в ней используются стандартные источники света. В качестве карты теней лучше всего использовать Ray Traced Shadows (в этом случае просчет идет трассировщиком лучей Mental Ray) и Mental Ray Shadow Map. Кроме этого, можно использовать стандартную карту теней Shadow Map, однако, собственная карта рендерера показывает лучшие результаты.

Среди дополнительных возможностей Mental Ray можно отметить:

- Эффект «смазанного движения» (Motion Blur)
- Эффект «глубина резкости» (Depth Of Field)
- Детальную прорисовку текстур (Displacement)
- Распределенный рендеринг (Distributed Rendering)
- Использование шейдеров камеры (Camera Shaders) для получения эффектов линз (Lens Effect) и прочих.

Глава 9. Reactor 2

Reactor 2 представляет собой одну из самых мощных утилит для моделирования физических задач и незаменимый инструмент при просчете динамики в сценах 3ds max. При помощи Reactor'а вы сможете просчитать поведение тел при соударении, имитацию водной поверхности, материи и многое другое. Reactor, как и некоторые описанные выше модули 3ds max, ранее был отдельным плагином, однако, начиная с 3ds max 5 входит в стандартную поставку программы.

Реактор 2 полностью интегрирован в 3ds max, о чем свидетельствует вертикальная панель с настройками модуля в левой части экрана. Изменения, которые коснулись Reactor 2, направлены, в первую очередь, на увеличение стабильности работы и скорости просчета сцен.

Reactor умеет работать со следующими группами объектов:

- Rigid Bodies (твердые тела)
- Soft Bodies (гибкие тела)
- Deforming Meshes (деформируемые поверхности)

- **Rope** (веревка)
- **Constraints** (конструкции)
- **Actions** (воздействия)
- **Water** (вода).

Все эти группы находятся в категории **Helpers** и Space **Warps** командной панели **Create** в группе объектов Reactor.

Обновленная версия Реактора работает с новыми типами конструкций — **Cooperative Constraints.** В их числе:

- **Rag Doll Constraints** — данный тип конструкций позволяет поворачивать и вращать тела на угол, не превышающий заданное значение. Примером данной конструкции может *служить* плечевой сустав руки.
- **Hinge Constraints** — данный тип конструкций позволяет осуществлять движения одного тела относительно другого вокруг заданной оси. Примером данной конструкции может служить локтевой сустав руки и колено.
- Prismatic **Constraints** — данный тип конструкций позволяет производить поступательные движения, подобные тем, которые осуществляют роботы и другие механизмы
- *CarWheel Constraints* — данный тип конструкций был разработан специально для симуляции поведения колес, прикрепленных к ходовой части, шасси.

В процессе работы над сценой удобно использовать опцию **Preview in Window** (просмотр в оконном режиме), появляется окно с логотипом Havok, внутри которого будет аппаратно отрендерен первый кадр, по виду напоминающий компьютерную игру.

Окно можно заставить проиграть анимацию, если вверху в ниспадающем окошке Simulation выбрать строчку **Play/Pause.** Здесь же можно указать прорисовку сетчатой оболочки для каждого объекта, по которой плагин просчитывает соударения.

Новая версия Реактора может хранить все данные относительно соударений всех твердых тел (rigid bodies) в процессе просчета. Эта информация запоминается и может быть вызвана при помощи MAXScript или сохранена в текстовом файле. Это файл содержит данные о скорости движения тел, координатах точек соударений и пр.

Глава 10. Другие нововведения

Интересным новшеством 3ds max 6 стала возможность рендеринга из командной строки (Command **Line** Rendering). Эта функция позволяет производить пакетный рендеринг, не устанавливая вручную параметры файла *.max. Рендеринг из командной строки производится благодаря утилите 3dsmaxcmd.exe, которая находится в папке 3dsmax6. Функция **Command** Line **Rendering** предназначена для профессионалов. Она может пытаться выполнить заведомо некорректные настройки сцены. При этом вы не увидите никаких сообщений, предупреждающих о возможной ошибке, например, о перезаписи существующего файла, некорректном отображении текстуры и пр. Рендеринг из командной строки также удобно использовать в процессе сетевого рендеринга и на мультипроцессорных системах.

Создавая трехмерную сцену, 3D-аниматор может выбрать удачный ракурс окна проекции, который желательно было бы получить в объекта ве виртуальной камеры. Ранее для этого требовалось выполнить две операции: сначала создать в сцене камеру, а затем выполнить команду **Match Camera to** View. В 3ds max 6 эти два действия заменены одним — **Create Camera From View** (сочетание клавиш CTRL+C). Камера создается автоматически, и пользователю не приходится выполнять лишнее действие.

В 3ds max 6 была встроена поддержка Direct X 9. Это позволяет в режиме real-time просматривать в окне проекции шейдеры Direct X 9 (файлы *.fx).

Еще одна интересная функция новой версии 3ds max — **Grab** Viewport. Она позволяет делать скриншот активного окна проекции, который затем можно сохранить в растровый файл.

Новая версия 3ds max поддерживает экспорт файлов Shockwave 3D (*.w3d) с учетом текстур, освещения и анимации. После установки параметров экспорта будет доступен предварительный просмотр файлов.

Еще одним приятным нововведением 3ds max 6 стала поддержка формата *.HDRI (High Dynamic Range Image). Эти файлы могут быть использованы в качестве текстурной растровой карты, в качестве карт отражения (Reflection) и **Radiance,** для имитации окружающей среды.

Как вы могли убедиться, 3ds max 6 содержит большое количество нововведений и усовершенствований. В целом они способствуют увеличению эффективности работы в программе. Можно даже говорить о том,

что 3ds max 6 отличается от пятой версии так же, как 3ds max 5 отличался от третьего релиза. Между четвертой и пятой версией такого количества различий не было.

Однако, несмотря на положительные изменения, в *программе* имеются существенные недостатки. Так, например, не был исправлен алгоритм создания булеановских объектов. Что же касается фотореалистичного рендера **Mental** Ray, то скорость просчета картинки с его помощью далеко не самая лучшая, поэтому большинство пользователей наверняка и дальше будут использовать подключаемые рендереры Vray, Final **Render** Stage 1 и пр.

Кроме того, новая версия 3ds max была перекомпилирована и поэтому не поддерживает плагины от предыдущей версии. Правда, большинство компаний, разрабатывающих коммерческие модули, уже успели выпустить переделанные под новую версию плагины. К сожалению, производители бесплатных дополнений пока не выкладывают перекомпилированные версии плагинов, поэтому многие пользователи не спешат переходить на новую версию, в которой не хватает привычных инструментов.

И, наконец, самым главным недостатком 3ds max 6 можно считать его нестабильную работу. В процессе работы то и дело на экране возникает окно с предложением отправить разработчикам отчет о произошедшем сбое, и программа закрывается. Остается надеяться, что в скором времени разработчики выпустят патч к программе, исправляющий если не все, то хотя бы большую часть ошибок.

Часть 3. Учимся мастерству

Глава 1. Создание развевающегося на ветру флага

Прежде всего, нам необходимо построить объект **HEXAMESH,** который собственно и будет выполнять роль полотнища флага. **CREATE ➪ REM PRIMITIVES O HEXAMESH -** в окне **FRONT** рисуем прямоугольную сетку. В поле **HEXAGONS** ставим значение 10. (Этот параметр определяет степень эластичности объекта и выбран опытным путем. При значении 4, флаг получается как резиновый).

Мы построили **MESH — но** это пока просто статичная заготовка. После наложения модификатора **CLOTHREYS,** она начнет подчиняться законам тяготения, т.е. просто будет падать вниз.

На заготовку можно воздействовать ветром. Но только ветром из **CLOTHREYS!** Тогда полотно будет не просто падать, а развеваться как настоящий флаг. Ну....точно так, как на НТВ в заставках.

Нам надо, чтоб левый край флага был жестко прикреплен к флагштоку, поэтому его надо будет закрепить, прикрепив к какому-либо статичному объекту, который при рендере будет скрыт.

Для этого создаем рядом (ни и коем случае объекты не должны пересекаться) любой **PRIMITIVE** объект, скажем BOX01.

Почему не **DUMMY?** Модификатор **CLOTHREYS** не понимает его. Во всяком случае у меня так и не получилось использовать в качестве объекта привязки **DUMMY.**

При рендере мы **BOX01** конечно скроем за ненадобностью, а пока он нам нужен как «точка опоры» в пространстве.

К HEXA01 применяем модификатор **CLOTHREYS (MODIFY ➪ MORE ➪ CLOTHREYS).**

Далее надо создать сцену. **MAKE SCENE** — введем любое название для сцены.

Таким образом у нас на рабочем столе создана сцена, в которой участвует объект HEXA01. Теперь нужно указать модификатору, какие еще объекты будут участвовать в нашеи сцене. **ADD OBJECT TO SCENE** — добавляем **BOXOI.** Обратите внимание, что в данный моменту нас получились выделенными оба объекта (HEXA01 **и** BOX01). Это то, что нам *именно* и надо!

Теперь поработаем над полотнищем. Нам нужно указать модификатору, какие участки флага должны активно двигаться, а какие — быть жестко закреплены. Например, левый край флага должен быть жестким, ведь он у нас прикреплен к флагштоку. Для этого, надо воздействовать на **HEXA01** на уровне подобъектов **(SUB-OBJECT).** Но, обратите внимание, кнопка **SUB-OBJECT** пока не активна. Укажем модификатору, на какие из объектов он должен воздействовать. **Кнопка MAKE FABRIC.** На вопрос о введении HEXA01 **в FABRIC** ответить утвердительно. Перед нами панель **FABRIC PARAMETERS** — оставим пока все как есть,

Жмем OK. Появится окно с запросом на объект BOX01.

Внимание! На вопрос о введении **BOXOI** в **FABRIC** ответить **ОТРИЦАТЕЛЬНО.** Нам не надо, чтобы наша «точка опоры» тоже стала падать вниз *или колыхаться от ветра.*

Выделяем объект HEXA01. Теперь у нас стала активной кнопка **SUB-OBJECT** в стеке модификаторов. Жмем ее. В окне **SELECTION LEVEL** выбран уровень **VERTEX.** Это то, что нам надо.

Выделяем группу вершин, которая должна быть фиксированной. Достаточно 1-го вертикального ряда с левого края. (Потом мы на него наложим флагшток). Делаем MAKE **GROUP** и называем ее **FIXED.** Внизу в окне она у нас появляется с пометкой **«NOT** FIXED». Это означает, что группа создана, но ни к какому объекту не прикреплена.

Выбираем в нижнем окошке созданную группу FIXED **и** жмем **FIX GROUP TO OBJECT.** Появляется список **MODEL NODES,** в **нем** выбираем **BOX01.**

Все. Теперь левая грань флага у нас привязана к объекту **BOXOI.** а так как он статичен, группа **FIXED** будет оставаться на месте, в то время как все полотнище будет двигаться. Если сейчас нажать кнопку **PLAY** в 3DS MAX, то **мы** никаких движений в сиене не увидим. Надо ее просчитать.

Отжимаем SUB-OBJECT, давим **START** CALCULATION. На запрос о количестве кадров ставим от **0 до 100 и OK. SPEED** — оставим пока 30. На все вопросы жмем **CONTINUE.** На запрос **SAVE MAX FILE -** как вам угодно.

Начался покадровый просчет. Мы видим, как правая часть флага под собственным весом опускается вниз и слегка колыхаясь, повисает под собственным весом.

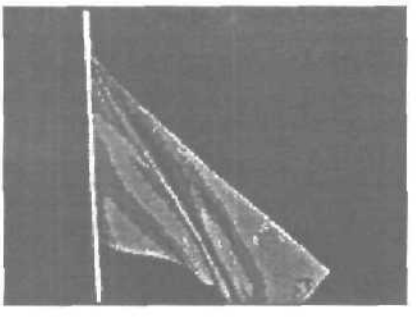
Этого нам недостаточно, нужно заставить ее двигаться. Попробуем добавить ветра. В окне **WIND X** вводим 500. Далее **MAKE FABRIC,** На запрос об изменении параметров ответить **YES.** Перед вами окно **FABRIC PARAMETERS.** В поле **AIR FRICTION** ввести **40.** Мы увеличили скорость ветра и силу трения флага о ветер.

Делаем снова **START CALCULATION и** пожалуйста — настоящий развевающийся флаг.

Поэкспериментируйте со скоростью ветра, с его направлением, с силой трения полотнища о ветер.

Когда все настройки с полотнищем закончены и внешний вид развевающегося флага нас вполне устроил, остается наложить на него флагшток и скрыть **BOX01.**

Если вам нужно натянуть текстуру на флаг, то делать это надо до того, как объекту **HEXA01** назначен модификатор CLOTHREY.



Глава 2. Ползающая многоножка

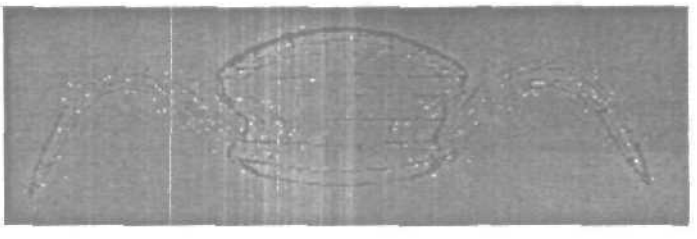
Создание и анимация многоножки является тем первичным опытом, который будет полезен для 3D аниматоров. MAX имеет некоторые полезные инструментальные средства, которые можно использовать для создания и анимации такой сложной, сегментированной конструкции.

Тело многоножки состоит из 19 полноразмерных сегментов, каждый с его собственной парой ножек, плюс три несколько уменьшенных

— шея и сегменты хвоста. Общее количество ножек равно 42. Головка с антеннами соединяется с укороченным сегментом шеи (имеющим две ножки, которые внешне выглядят как клыки), который в свою очередь опирается на первый сегмент тела. Задняя часть тела насекомого содержит уменьшенный 21-й сегмент с двумя удлиненными ножками, которые используются для захвата иди сжатия.

Начните строить секцию тела из примитивов: box, который имеет три сегмента на каждой стороне. Сузьте box, используя модификатор **Editable Mesh** в режиме SubObject **C Vertex,** чтобы создать на одном конце выступ подобный шейке и на другом выступ похожий па нависающую губу. Потом сожмите box на треть и примените ко всем граням модификатор MeshSmooth.

Сформируйте ножки: выберите по одной грани с каждой стороны сегмента тела, которые расположены в центре (ниже середины) и примените к ним модификатор **Extrude,** направленный наружу. Формируя последние участки ножек, вытяните (Extrude) выбранную грань несколько (около 5) раз, масштабируя ее и изменяя углы направления вытягивания. Затем, отделите ножки от сегмента тела, чтобы анимировать их. (Ножки преднамеренно сделаны простыми, без многократных сочленений, которые усложнили бы процесс анимации.) Чтобы отделить их от сегмента тела, выберите все грани одной ножки, находясь в режиме **SubObject, а** затем отделите (Detach) их. Эта команда отсоединяет все грани, формируя объект с новым именем. Повторите те же действия для противоположной ножки и затем выйдите из модификатора.



Простые изогнутые ножки были сделаны вытягиванием и масштабированием граней. Устранение многократных сочленений упростило процесс анимации цикла шагов.

Чтобы установить оси вращения для ножек, откройте **Hierarchy Panel и** используйте команду Adjust Pivot ➪ **Affect Pivot Only,** чтобы переместить центр вращения в основание каждой ножки. Затем, свяжите каждую ножку с сегментом тела. Сейчас, первый сегмент многоножки готов к анимации одного цикла движения. Лучше анимировать ножки

прежде, чем формировать остальную часть модели, так как все остальные ножки будут Instances первых двух, и ключи анимации должны быть применены к ним прежде, чем они будут клонированы. Иначе, остальные 19 Instances сегментов тела и пар ног не будут правильно наследовать движение. Нельзя просто делать **Instances** из первой пары ножек и затем перемещать их в нужное место, чтобы формировать многоножку, потому что тогда все ножки будут дочерними только по отношению к первоначальному сегменту и будут повторять только его движения. Необходимо создавать только один Instance сегмента тела и его пары ножек от каждого предыдущего сегмента, для того чтобы анимация заработала правильно.

Создание Первого Шага

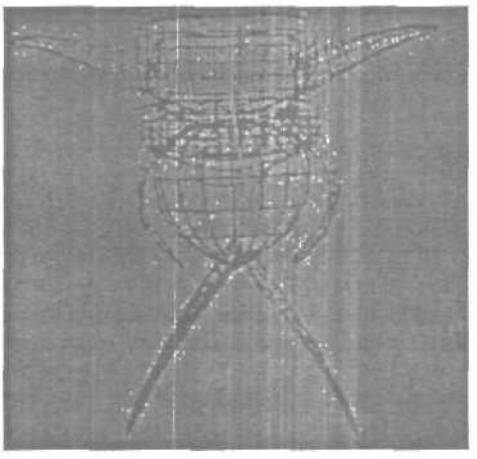
Предлагается создать короткий цикл шагов, который будет повторяться несколько раз. Установите длину анимации равную **500** кадрам, переместите **Time Slider** на 10 кадр, включите режим анимации и на виде сверху поверните обе ножки вперед по направлению к заостренному концу сегмента тела. В 20 кадре, поверните обе ножки в обратную сторону, к задней части сегмента тела. Сейчас закончена первая половина цикла шагов, в которой ножки двигаются вне контакта с землей. В 25 кадре поверните, их наполовину вперед и приподнимите так, чтобы они оказались над уровнем земли. Наконец, в 30 кадре поверните их полностью вперед и вниз, чтобы они вошли в контакт с землей и были точно в том месте, в котором они находились в 10 кадре. Теперь вы завершили один цикла шагов.

Следующие действия сделают шаги непрерывными для всей последовательности анимации, Окно **Track View** имеет мощные возможности для редактирования анимации, включая способность применять различные контроллеры и промежуточные ключи к любому треку анимации. Хитрость в создании циклической анимации состоит в том, чтобы сообщить системе о повторении анимации, используя диалоговое окно Out-of-Range Keys. Эти ключи распространяют анимацию за пределы диапазона уже установленных ключей; поэтому и называются **Out-of-Range.**

Нажмите кнопку **Curves Out-of-Range Types,** она откроет диалоговое окно, которое предлагает шесть способов повторения анимации. Это Constant (постоянная), **Ping-Pong** (пинг-понг), **Loop** (петля), **Linear** (линейная), **Relative Repeat** (относительное повторение) и Cycle (цикл) анимации. Выберите **Cycle, и** затем щелкните на кнопку Play. Ноги будут плавно и непрерывно двигаться на протяжении всех 500 кадров.

Создание Instances Сегментов Тела

Создайте сегмент шеи с ножками-клыками, клонируя первый сегмент и масштабируя его. Затем сформируйте головку из сферы и добавьте к ней антенны, после чего свяжите все части головы вместе. Не прикрепляйте эти части к первому сегменту тела до тех пор, пока не закончите создание оставшихся 19 сегментов тела.



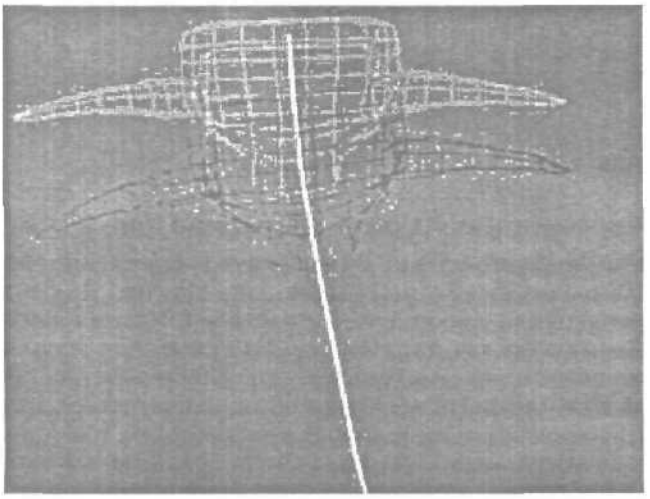
Теперь можно заняться созданием **19 Instances** первого сегмента тела. Выберите первый сегмент и его ноги на виде слева, и нажмите кнопку Array. В открывшемся диалоговом окне, установите преобразование вдоль оси X с шагом -38.0 единиц, а число копий равное 19. Затем нажмите OK и выберите **Instance** как тип преобразования. Получилось 20 идентичных сегментов тела (с 40 ногами), которые перекрываются друг с другом примерно на 20 процентов. Нажмите **Play,** теперь все сегменты дружно будут перебирать ножками.

Создание Пути Анимации

Чтобы собрать насекомое, используйте способность MAX присваивать одну и ту же анимацию любому количеству объектов, а затем корректировать их положение для выравнивания друг относительно друга. Чтобы создать путь движения сороконожки, на виде сверху нарисуйте изогнутый сплайн, начинающийся от передней части первого сегмента тела. Затем выделите первый сегмент и откройте Motion **Panel** и **в** меню Assign Controller выделите контроллер Position, использующий Besier Position **Controller.** После чего, назначьте контроллер **Path,** который заставит сегмент следовать по выбранному сплайновому пути.

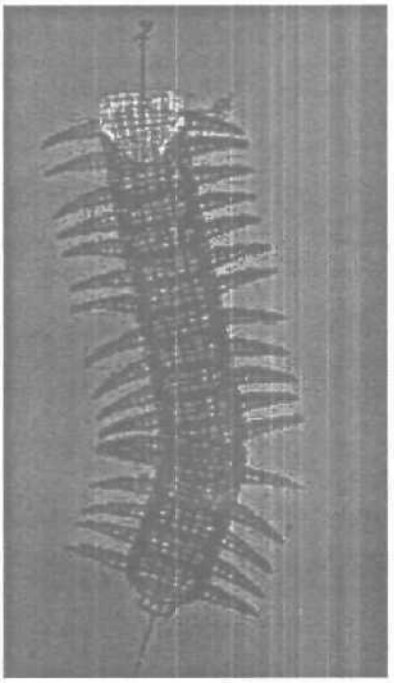
Как только контроллер **Path** будет назначен, в диалоге параметров пути нажмите кнопку Pick Path, а затем на виде сверху кликните на сплайновый путь. Сегмент займет начальную позицию на своем пути. В меню параметров пути следует установить Constant Velocity по оси Y и нажать Flip. Эти действия выровняют сегмент, так чтобы он правильно следовал по пути не ускоряясь и не замедляясь в течение анимации. Последний шаг очень важен, поскольку он предохранит другие сегменты от сталкивания и наложения во время движения.

Затем, установите **Time Slider** в значение 10 и назначьте новый контроллер **Path** для первого **Instance** сегмента тела. Нажмите кнопку **Pick** Path, чтобы новый сегмент расположился следом за первым сегментом. Теперь необходимо откорректировать его относительную позицию так, чтобы новый сегмент следовал за первым. Вернитесь в меню параметров контроллера пути и уменьшите Percentage spinner примерно на -1.8. Это заставит новый сегмент тела переместиться на небольшое расстояние назад от начала пути, пока пересечение с первым сегментом не составит примерно 20 процентов. Повторите эти действия для всех 19 Instances сегмента тела, продвигая **Time Slider,** пока не получите все тело многоножки, собранное на линии пути.



Контроллер **Path** применялся к каждому **Instance** сегменту тела, чтобы задать движение по сплайновому пути. Несколько объектов могут двигаться по одному пути.

Все **19** Instances сегмента тела двигаются по заданному пути. Затем необходимо откорректировать параметры относительного отклонения.



СинхронизацияКлючейДвиженияНог

Сейчас, когда все сегменты равномерно установлены и движутся по пути, необходимо изменить анимацию движения ног. На данный момент они двигаются все вместе, подобно весельной лодке с командой в 40 человек. Многоножка так не двигается. На самом деле движение ног многоножки носит волнообразный характер. Чтобы этого достичь, зайдите в **Track View и** откройте все 40 треков анимации, отображая ключевые кадры для всех ног. Ключи изображены как точки, а весь диапазон анимации с 20 кадровым циклом представлен черной линией с квадратиками на каждом конце.

Вторая пара ног прикоснется к земле только после того, как первая пара выполнит весь цикл движения, то есть необходимо переместить вперед во времени диапазон анимации второй пары ног. Аналогично анимируется третья пара ног, она касается земли только после того, как вторая пара пройдет свой последний ключ движения. Таким образом,

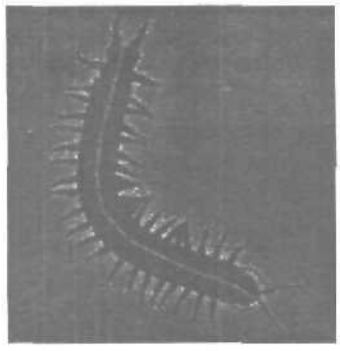
необходимо переместить диапазон анимации слегка дальше диапазона второй пары. Следите за движением каждой пары ног, перемещая **Time Slider** вперед и назад, пока не получите правильное движение, потом переходите к следующей паре ног и повторяйте весь процесс заново. По окончании процесса получается волнообразное движения ног от передней части тела к задней.

Чтобы корректировать время, когда каждая пара ножек должна соприкасаться с землей, нужно было слегка смещать полосу **Key Range в Track** View, а затем применил **Out of Range Key,** чтобы сделать анимацию ножек непрерывной.

В природе существует много насекомых с волнообразным движением ног — от задней части к передней. Получить это движение просто, надо полностью повторить вышеупомянутый процесс, начиная с последнего сегмента. Когда многоножка бежит, ее тело изгибается, принимая S-образную форму. При этом меньшее число ног касается земли, а их движение чередуется на противоположных сторонах тела. Не стесняйтесь экспериментировать с моделью.

Присоединение Головы и Хвоста

Когда анимация закончена, можно присоединить голову и хвост. Расположите голову перед первым сегментом тела, а затем свяжите ее с ним. Войдите в **Track View и** удалите ключи анимации из сегмента шеи так, чтобы ноги-когти не двигались. Затем, укоротите последние сегменты ножек, после чего поверните их так, чтобы они указывали вперед. И, наконец, скопируйте последний сегмент (на сей раз **Copy, а не Instance)** и уменьшите его, чтобы создать сегмент хвоста в виде клещей. Разместите ноги позади сегмента, так чтобы они тащились за насекомым. Потом удлините их и поверните навстречу друг другу, чтобы ноги-клещи были как у настоящей многоножки.

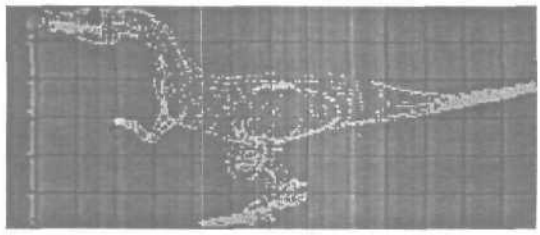


Последний шаг — примените материалы к сегментам тела и ногам. Используйте стандартный материал с диффузной картой MIX, состоящей на 60 процентов из Bitmap **Asphalt** и на 40 процентов из карты **Cellular**, чтобы сделать оболочку многоножки более натуральной. Диффузная карта MIX относится к тем типам карт, которые при создании позволяют смешивать две отдельные карты вместе в изменяемом процентном отношении. Добавьте также карту **BUMP** к **Bitmap** Asphalt со значением 115 процентов. Установите Opacity равную 92 процентам, чтобы придать насекомому легкую прозрачность. Затем, добавьте **Target Camera, и** осветите насекомое с помощью **Omni** с отбрасыванием теней. Постройте большой куб и примените материал поверхности земли, чтобы многоножке было по чем ходить.

На сегодняшний день такой метод анимации достаточно прост. Необходимо научиться работать с многочисленными частями тела и большим количеством ног. Потом эти знания можно будет применить к различным проектам: от каботажного судна с пассажирами до идущей колонны людей или омаров и даже к полету стаи уток. Вывод: чтобы упростить создание анимации большого числа одинаковых объектов, необходимо использовать **Instances** при копировании как объектов, так и их ключей анимации. Затем необходима синхронизация каждого момента движения, чтобы движение внешне выглядело естественным.

Глава 3. Основы инверсной кинематики

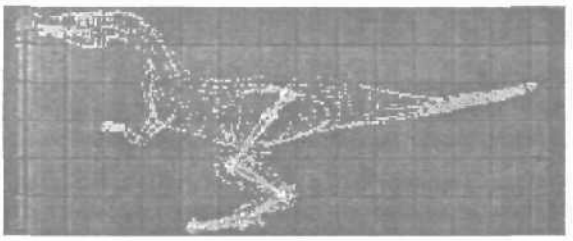
Подготовка персонажа к анимации не такой сложный процесс как многим кажется. Многие пользуются для этого популярным пакетом Character Studio, но на самом деле, Max обладает практически всеми инструментами для решения этой проблемы. К примеру, для создания динозавра с гладкими и реалистичными движениями необходимо правильно подготовить модель к анимации. На рисунке одна из моделей динозавров — наша задача создать для нее скелет, который позволил бы в последствии правильно ее анимировать.



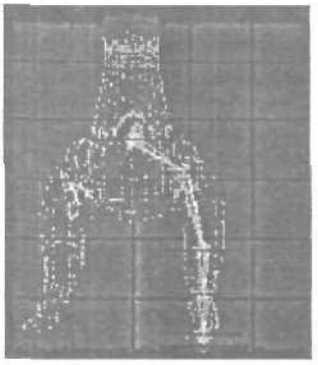
Для создания скелета мы воспользуемся стандартным инструментом Max'a — **Bones.** Этот инструмент находится в **Create Panel** в подменю **Systems.**

При создании первой кости убедитесь, что установлен параметр **Assign to** Root, а параметр Create End Effects отменен — мы создадим **End Effects** позднее.

Самая важная вещью при создании при создании скелета это правильно начать. Начинать нужно со спины, устанавливая кости по цепочкам — например, конечным звеном в скелете ноги будет палец.

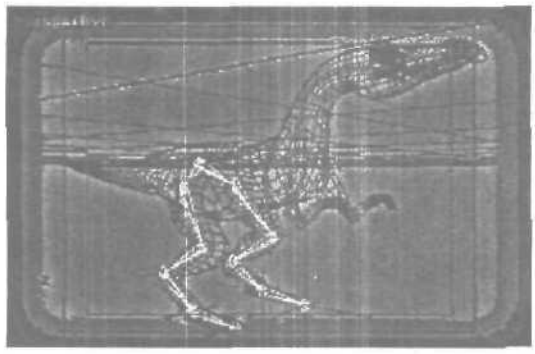


Далее мы подгоним получившуюся цепочку костей под модель.



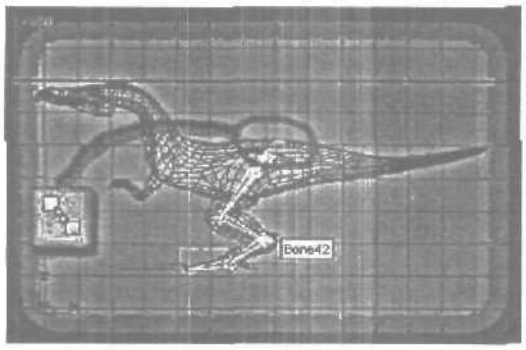
После того, как вас устроит получившаяся модель ноги, можно приступать к созданию скелета для другой ноги. Сначала кликните на спинную кость — таким образом обе ноги будут объединены в один скелет.

Теперь создадим **End Effectors. В** панели **Create** в подменю **Helpers** выберите **Dummy.** Создайте по одному Dummy для каждой ноги. Используя Non-uniformal **Scale** подгоните их под размер ноги. Эти **Dummies** будут служить контроллерами при анимации скелета.



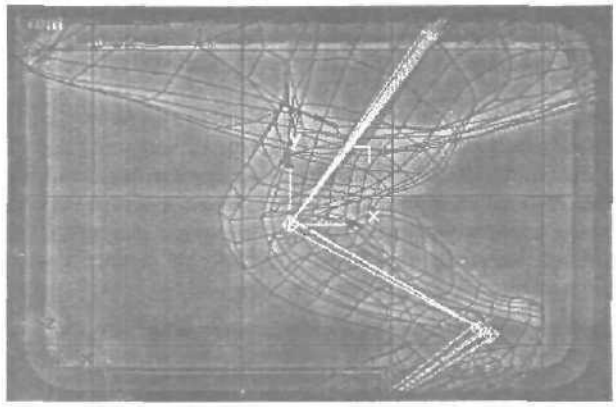
Теперь выберите палеи левой ноги и откройте панель Motion. Здесь мы создадим End Effectors. Нажмите кнопку Create и кликните на соответствующие Dummy и кость. То же самое и для второй ноги

Теперь создадим Dummy, который будет использоваться как основной контроллер. Расположим его слегка над костями ног. Если бы мы делали скелет человека, основноиконтроллер располагался бы в районе живота. Теперь свяжем (Link) кости ног с контроллером (обязательно в этой последовательности!) — и все готово.

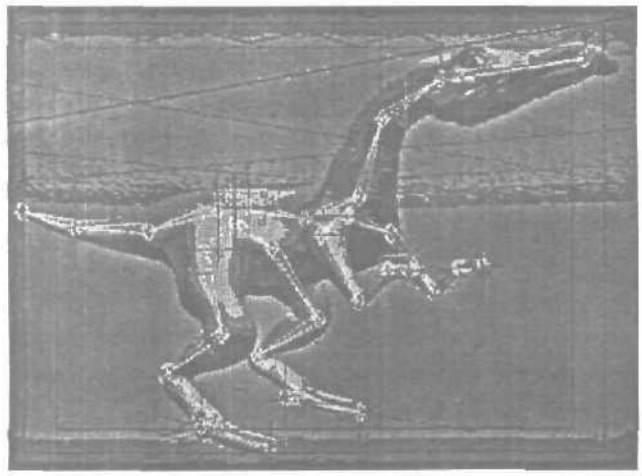


Теперь у нас есть правильно расположенные скелеты ног, но если мы слегка переместим контроллер ноги, выяснится, что движения скелета далеки от реальности. Для исправления этого откроем панель Hierarchy подменю IK — здесь содержится вся информация о всех костях скелета. Нам необходимо, чтобы колено гнулось только в нужном направлении. Для этого в панели можно задать активные углы изгиба, ограничения изгиба и многое другое. Теперь при изменении положения контроллера ноги мы получим более реалистичные движения нашего

персонажа. Если вас не устраивает полученный результат вы можете сколько угодно экспериментировать с настройками для достижения идеала.



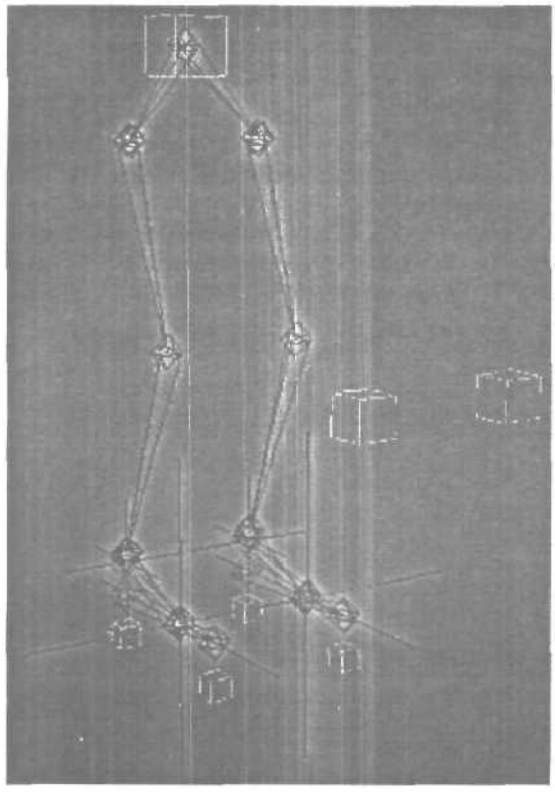
Вот собственно и все - практически все - остались только передние конечности, голова, спина и хвост. Все делается по той же технологии, что и ноги с одним исключением - настройки панели IK – для каждого соединения необходимо установить нужные параметры, чтобы движение персонажа было как можно более реально. Кроме того необходимо помнить, что каждую новую цепочку необходимо начинать от главного контроллера, иначе ничего работать не будет.



На рисунке хорошо видно расположение **контролеров,** движения **конечностей и** строение костей **скелета.**

Глава 4.
Инверсная кинематика — урок второй

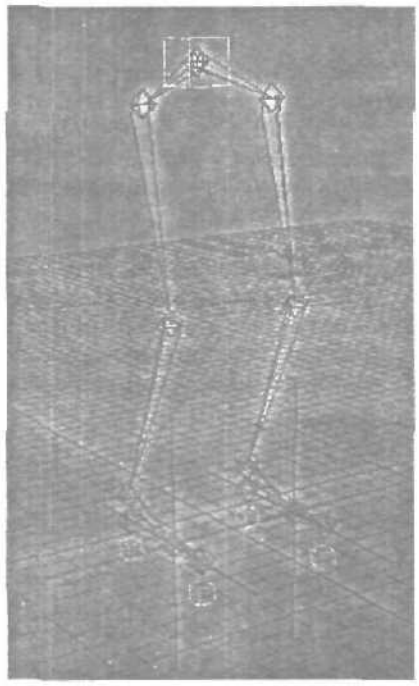
Инверсная кинематика является незаслуженно забытым инструментом для анимации **персонажей. Данный урок рассчитан на** опытных пользователей **3D Studio** Max, знакомых с **общими принципами IK (Inverse Kinematics). Мы рассмотрим два основных способа построения** цепочек IK. **Данные методы могут** быть **использованы и в других** пакетах **3D графики — Alias** Wavefront's Maya **и Softimage 3D —** инструменты **могут различаться, но основная** идея **одна и та же.**



Создание скелетов

Первый способ

Итак, посмотрим на рисунок ниже и займемся его изучением. Как вы видите, скелет представляет собой обычную двуногую модель гуманоида. Кости ноги организованы в единую цепь.



Обратите внимание на серые цепочки костей, образующие стопы. Эти цепочки не связаны с основным скелетом ноги, чтобы обеспечить прямую кинематику движения пальцев ног и пяточной области. Данные цепочки будут использоваться также в качестве контроллера анимации ноги. Такой вид контроллера позволяет использовать одновременно и инверсную кинематику, и прямую кинематику. Эти цепочки мы будем называть «Стопой». В дальнейшем, принцип построения стопы будет рассмотрен подробнее, не расстраивайтесь, если вы не до конца разобрались в этом. Для начала усвойте то, что Стопа не является частью цепочки костей ноги. Такая организация скелета одинакова для обоих принципов IK.

Объекты **Dummy,** показанные на рисунке, служат для осуществления всех видов анимации скелета.

«ЗА» и «ПРОТИВ»

ЗА:

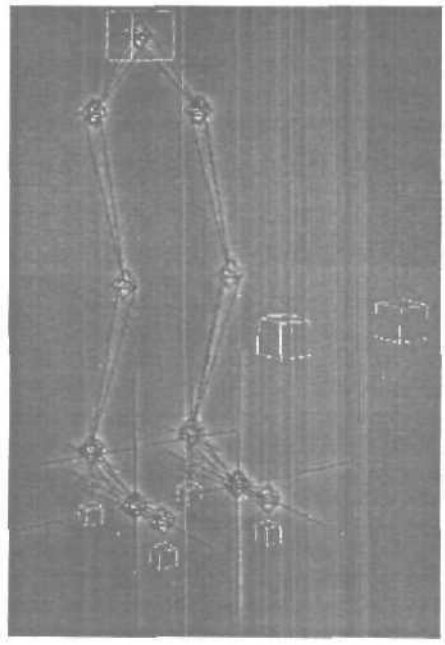
- Такой скелет дает возможность осуществлять все виды анимации
- Есть возможность выполнять как Инверсную, так и Прямую Кинематику

ПРОТИВ:

- Некорректная анимация колена при вращении кости бедра

Второй способ

Данное построение скелета основано на тех же приемах что и в Первом Способе, но в данном случае решается проблема некорректной анимации колена. В обоих принципах используются одни и те же контроллеры, различие состоит в структуре скелета — добавлен дополнительный сустав с контроллером для управления движением колена.



«ЗА» и «ПРОТИВ»

ЗА:

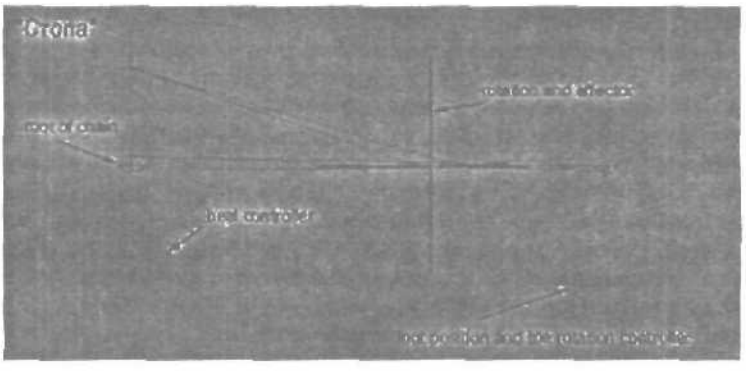
- Такой скелет дает возможность осуществлять все виды анимации
- Есть возможность выполнять как Инверсную, так и Прямую Кинематику
- Позволяет контролировать верхний вектор и управлять движением колена, что решает проблему, возникают™ в Первом Принципе

ПРОТИВ:

- Теперь скелет состоит из двух разных цепочек, что может вызвать трудности при анимации
- Наличие нового коленного сустава может слегка влиять на взаимодействие движения костей скелета

Строение «Стопы»

Стопа выполняет две функции — контролирует движение и помогает визуализировать процесс анимации. Она состоит из одной цепочки костей (bones). Основание Стопы, в отличие от всех остальных костей в цепочке, не имеет контроллера «New IK».



Итак, Стопа состоит из трех костей, как вы видите эта цепочка полностью повторяет расположение костей в реальном скелете за исключением пяточной кости.

В опорной части стопы находится rotation end effector, который отвечает за вращение пяточной области. Этот effector связан **(link) с Dummy**

объектом — «Heal Controller». Вращая **Heal Controller** в локальных координатах, мы можем управлять вращением пяточной области ноги.

Теперь нам нужен контроллер, управляющий позицией и вращением самой стопы. Для этого необходимо просто связать корневую кость Стопы **и** Heal Controller **Dummy** в качестве **parent** объекта с **Dummy** контроллером позиции стопы.

Теперь мы имеем полностью рабочую Стопу. Пользуясь двумя **Dummy** контроллерами мы можем управлять позицией, вращением и подъемом стопы.

Такое построение стопы является ключевым моментом Первого и Второго способа построения скелета. Теперь, когда мы разобрались со строением Стопы, вернемся к нашим Двум Способам.

Тип первый

Первый Способ — самый подходящий для знакомства с Инверсной Кинематикой, т.к. чем больше вы используете ее для анимации персонажей, тем более ясным становится для вас, что иногда вы просто не можете добиться желательного результата, что побуждает к более детальному изучению этого инструмента.

Сначала разберемся, что такое «ограничитель». Под ограничителем мы понимаем любой объект, ограничивающий передвижения другого объекта. В роли такого объекта может выступать любой предмет — dummy, **end effector,** даже источник света. Обычная связка объектов (link) - это простейшая форма ограничения, хотя на самом деле при связывании объектов друг с другом полного ограничения движений не происходит. Например, если мы свяжем сферу и куб, мы сможем анимировать эту сферу отдельно от куба, хотя перемещения куба и будут влиять на сферу. Настоящая ограничительная связь не позволяет анимировать ограничиваемый объект, это можно сделать только перемещением контроллера и анимация полностью зависит от типа и свойств ограничительной связи. Существует несколько видов ограничителей, но для полного понимания этой статьи необходимо понять только один — так называемый «Направляющий вектор» — в Маya он называется Pole **Constraint,** а в Softimage — Up-Vector **Constraint.**

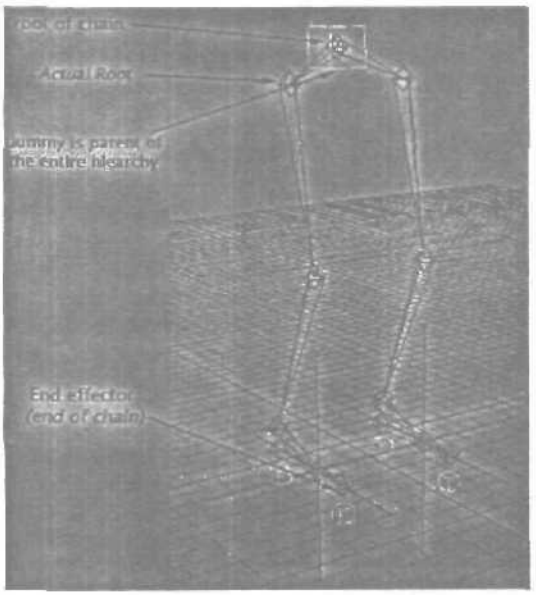
Зачем говорить о Первом Способе IK, если он имеет недостатки?

Выше было упомянуто, что Первый Способ далеко неидеален, зачем же тогда вообще о нем упоминать, можете спросить меня вы. Ответ прост — в Маya и Softimage вы можете пользоваться Первым Способом спокойно, т.к. в этих пакетах присутствует вышеупомянутый ограничитель. Таким образом, каждая создаваемая вами цепочка костей автомати-

чески снабжается этим ограничителем, который определяет ориентацию корневой кости. Этот контроллер дает возможность ограничивать вращение цепочки одной плоскостью. В Max'е такого ограничителя нет и нам придется пользоваться только тем, что предлагает Max. Мы можем обойти это неудобство с помощью небольшого трюка, который конечно не так хорош как встроенные ограничители Maya и Softimage, но тем не менее позволяет контролировать плоскость перемещения цепочки костей.

Детальное изучение Первого Способа построения IK

Ниже показана полная иерархия скелета ноги. Мы видим, что объект Dummy занимает в ней первое место (parent). Но на самом деле корневая кость не совсем отвечает всем признакам корневой кости — ей не назначен IK-контроллер. Таким образом основной костью можно считать бедро — назовем эту точку «действительная корневая кость» (Actual Root). В этом месте начинается влияние IK. В Maya такую цепочку создать легче, т.к. там есть специальный инструмент Handle, с помощью которого можно легко вручную выстроить всю иерархию скелета. Max не обладает подобным инсфументом и нам нужно быть осторожными при построении скелета.

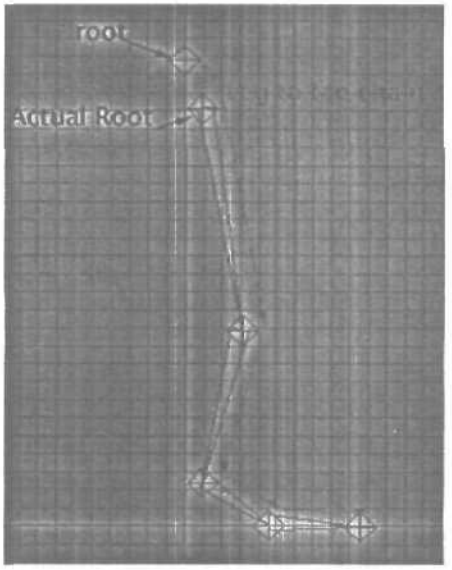


Чтобы правильно выстроить иерархию костей скелета в первую очередь необходимо создать «Корень» (Root).

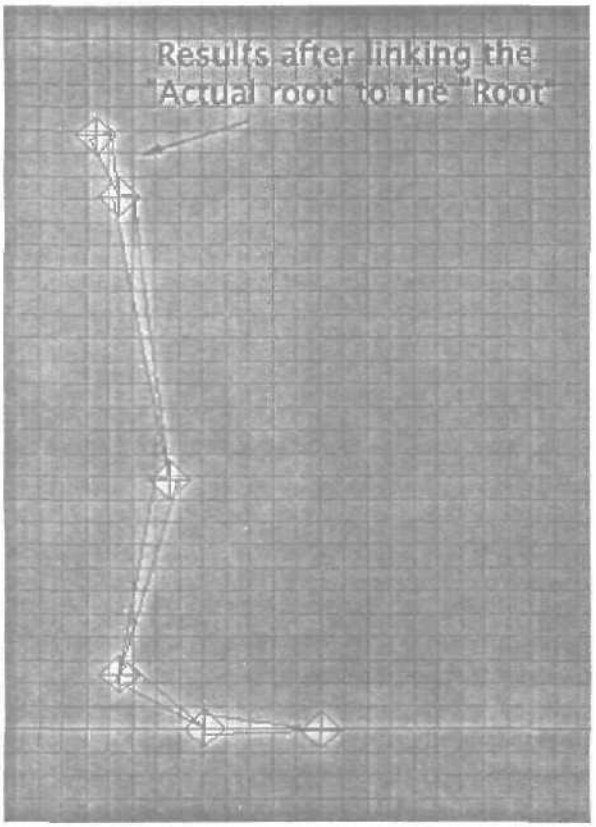
- откройте панель **Create/Systems.**
- кликните **Bones.**
- отключите параметр Assign to **Root.**
- кликните в окне вида чтобы создать **Root, а** затем еще раз, чтобы создать кость.
- теперь выберите кость и удалите ее. Таким образом останется только **Root.**

Этот объект не обладает контроллером IK, и значит, когда мы создадим скелет ноги и свяжем его с ним, цепочка IK будет начинаться не с него, а с «Действительного корня» **(Actual Root)** — что нам и нужно.

Теперь создайте скелет ноги начиная с **Actual Root.** Помните, нам нужна новая цепочка, поэтому не начинайте с объекта, который мы только что создали, а создайте отдельную цепочку. Для этого включите параметр **Assign to Root** и отключите **Create End Effector.** Когда весь скелет будет готов — свяжите **Root и Actual Root** — после этого будет автоматически создана кость, объединяющая два объекта в одну цепочку.

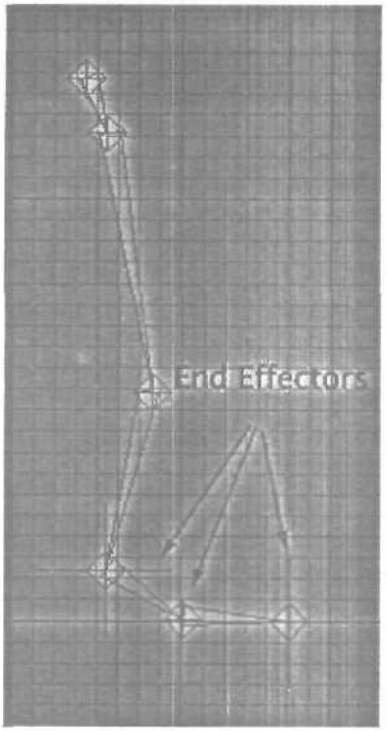


У вас должно получиться нечто подобное:



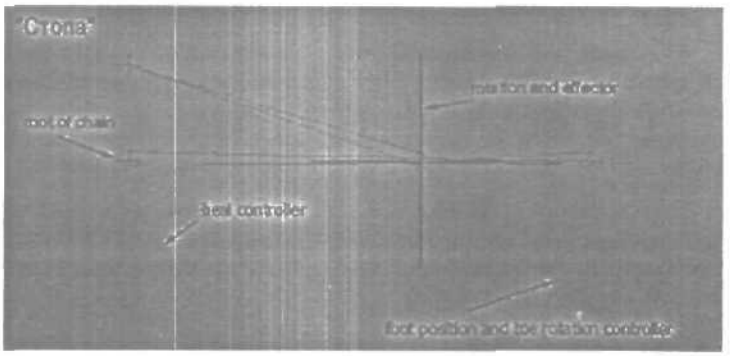
Помните, что материал данного урока подразумевает знание основ Инверсной Кинематики в Max. Для экономии времени мы не будем рассматривать создание полного скелета. Если вам что-либо не понятно, обратитесь к справочным материалам по 3D Studio Max и по инструменту **Bones**.

Дальше мы создадим **End Effector** на пальце ноги (т.е. на последней кости в цепочке). Кроме того нам понадобятся еще два End Effector'а для обеспечения реалистичной анимации. Эти эффекторы будут управляться Стопой.



Создание «Стопы»

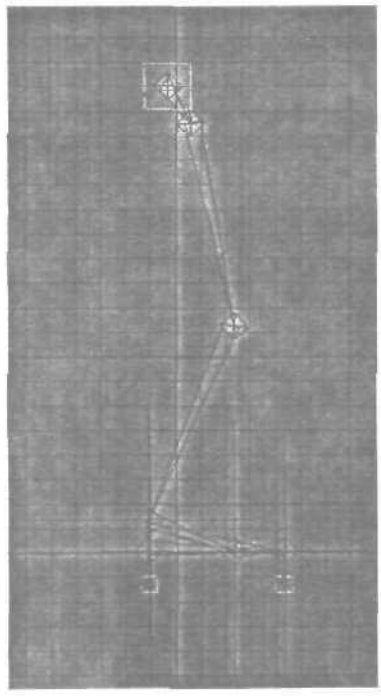
Перед **тем как добавлять Стопу в наш скелет вы** должны создать ее.



Посмотрите на рисунок выше. В Стопе Корневая кость в цепочке (Root) также не имеет IK контроллера, таким образом при создании первой кости необходимо отключить параметр Assign to Root. Следуя рисунку создать Стопу не сложно — первая кость идет от корня до самого конца пальца, затем возвращается до середины и затем вверх к пятке. Стопа должна в точности повторять основной скелет, для этого вы можете сначала создать макет стопы, а затем переместить суставы в нужное положение. Когда модель Стопы готова, вы можете создать **End Effector,** как показано на рисунке выше.

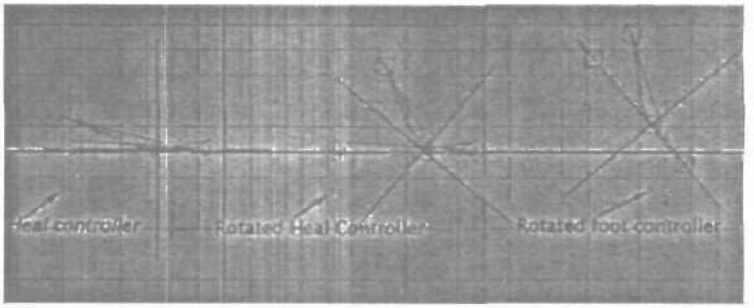
Далее нам понадобятся два объекта **Dummy** — один под пяткой, другой под конечным суставом пальца. Теперь свяжите **(link)** Корень **(Root)** стопы и **Dummy** под конечным суставом — в результате получим Foot **Position Controller.**

Теперь свяжем **(link) End Effector и Heal Controller.** Для этого откройте панель **Motion** и выберите наш **End Effector.** Внизу панели вы увидите функцию «**End Effector** Parent», кликните **Link** и выберите «**Heal Controller** Dummy».



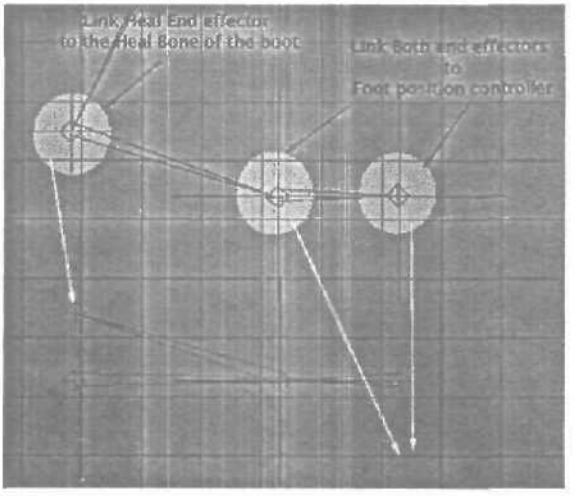
Теперь у вас должно получиться что-то похожее на рисунок выше. Обратите внимание, что создан еще один **Dummy** в основе всего скелета. Вы должны сделать то же самое, а так же связать **Dummy с Root** всей системы, сделав его первым в иерархии (Parent).

Проверим что у нас получилось. Попробуйте вращать контроллеры, как показано на рисунке ниже. У вас должно получиться тоже самое.



Теперь у нас есть возможность выполнить полную анимацию стопы при ходьбе — осталось присоединить нашу «Стопу» к основному скелету.

Это довольно просто, необходимо лишь связать **End Effector**'ы реального скелета с нужными элементами стопы, сделав их основными **(parent).**



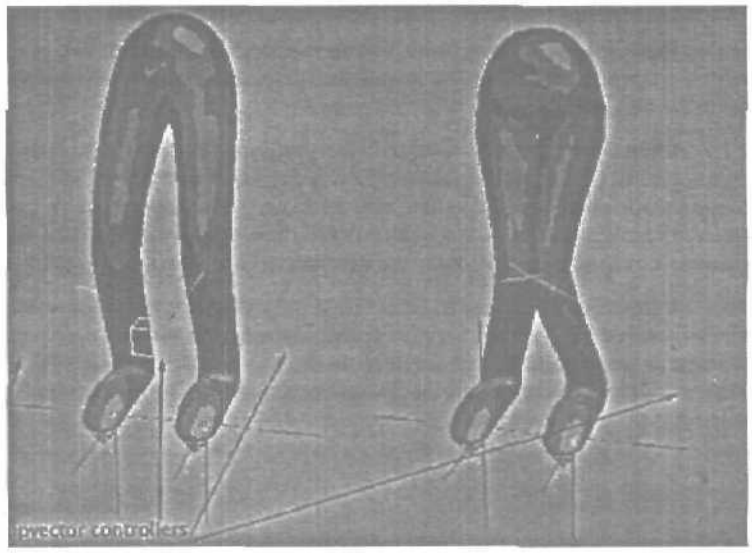
На рисунке выше точно показано, каким образом соединять (link) объекты. Обратите внимание, что в вашем случае основной скелет и Стопа должны совпадать — на рисунке я сдвинул стопу немного вниз, чтобы яснее показать порядок связи — вам этого делать не нужно.

Мы создали полностью рабочий скелет ноги. Вы можете протестировать его работу.

Почему Первый Способ IK работает не во всех ситуациях?

Первый Способ построения цепочек IK — это хороший инструмент для начинающего аниматора, но если вы занимаетесь анимацией всерьез, то скоро наткнетесь на его слабые стороны. Уже упоминалось о таком типе ограничителя как «Направляющий вектор», большинство пользователей Softimage и Maya знакомы с этим термином, но Maxтаким инструментом не обладает. Таким образом пользователи Max вынуждены так или иначе обходить этот недостаток, что несколько усложняет работу. Что ж, нам придется найти другой способ контролировать плоскость вращения нашей цепочки IK.

А зачем он нужен, этот «Ограничитель по Направляющему Вектору»? Наиболее часто этот инструмент используется для контроля над движением коленных суставов.



Вы можете видеть, как работает этот ограничитель. Скелет на рисунке построен по Второму Способу IK. Так как Max не имеет в своем распоряжении вышеозначенного ограничителя, нам придется обмануть программу, для этого служат специальные контроллеры направления коленей. Такой способ построения скелета будет более подробно рассмотрен далее.

Итак, на рисунке мы используем Dummy объекты в качестве контроллеров направления коленных суставов. С помощью этих контроллеров мы можем заставить колени смотреть либо вперед, либо друг на друга, либо как вам угодно. Такой контроль над коленными суставами очень важен. Без такого контроля в вашем скелете невозможно будет отрегулировать степень свободы вращения коленного сустава (как в реальном человеческом скелете).

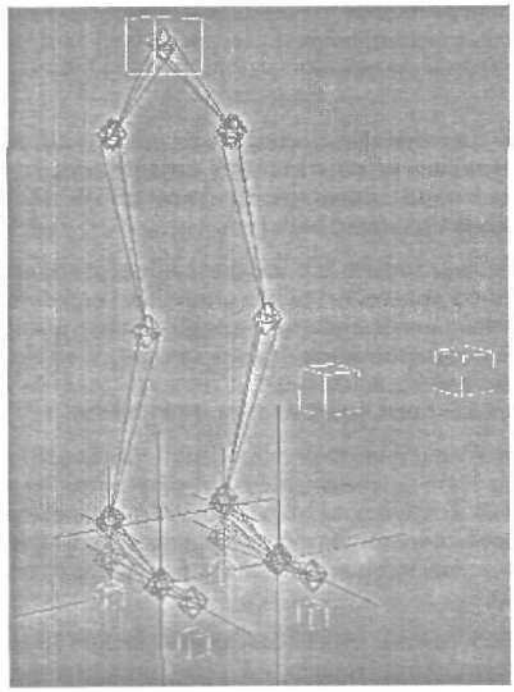
Необходимо заметить, что всем известный пакет Character Studio с его инструментом **Biped** обладает всеми необходимыми средствами для анимации двуногих персонажей. Эта программа является наилучшим выбором при анимации таких персонажей т.к. она намного превосходит возможности, которыми обладают Softimage и Maya. Но если нам нужно анимировать модель, не подходящую под определение «двуногий» — здесь Character Studio бессильна и нам придется искать другой способ. И IK — прекрасный инструмент для выполнения этой задачи.

Второй Способ построения цепочки IK

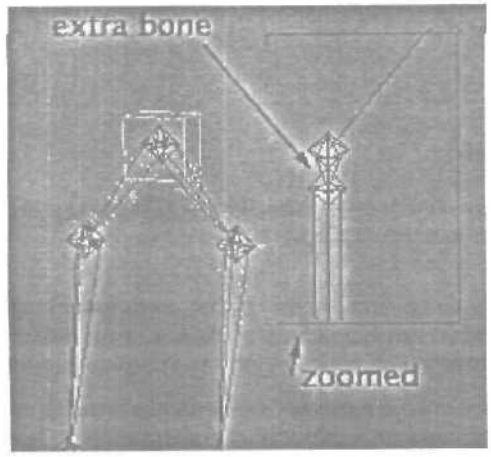
Приступим к изучению Второго Способа. Очень похоже на первый, не правда ли? Но если посмотреть внимательнее, обнаружатся некоторые различия.

Основных различий два. В отличие от Первого Способа, вся основная цепь поделена на две части — верхнюю и нижнюю. Корень новой IK цепочки обладает собственным IK контроллером и имеет только две степени свободы (вы можете обратиться к готовому файлу сцены, чтобы понять, какими плоскостями ограничено движения).

Второе отличие — это наличие дополнительной маленькой кости в основании бедра. Эта кость дает возможность контролировать плоскость вращения всей цепочки. Эта кость не обладает собственным IK контроллером, вместо этого у нее есть «направляющий контроллер» который контролирует ее ориентацию.



Ниже на рисунке дополнительная кость показана более подробно.



Скелет состоит теперь из двух раздельных цепочек, такая организация позволяет правильно анимировать скелет, используя контроллеры направления. Такая система прекрасно работает, необходимо только быть осторожнее.

Итак, во Втором Способе присутствуют два новых контроллера, которые выполняют функции «ограничителя по направляющему вектору», они, совместно с маленькой дополнительной костью в бедре ограничивают плоскость вращения всего скелета ноги.

Наилучшим способом понять все вышеизложенное является самостоятельное сравнение Первого и Второго способа с помощью сцен-примеров. Приводится два варианта сцены: первый — без использования Expressions, второй — с использованием. Нужно сказать, что Expressions дают более полный контроль над анимацией, но при это существенно замедляют процесс — решайте сами, пользоваться ими или нет.

Внимание! Данный прием не является полной имитацией работы настоящего «ограничителя по направляющему вектору». Реальный ограничитель работает только и одной плоскости вращения, в нашем же случае, контроллеры направления работают в двух плоскостях. Таким образом при анимировании очень важно передвигать их не только вправо и влево, но и вверх и вниз!

Не что ж, мы разобрались в двух способах построения IK цепочек (надеюсь, что разобрались). Хотя IK является очень мощным средством в анимации, все же советуем вам пользоваться Biped если это возможно, т.к. этот инструмент практически не имеет недостатков. Работать с IK в Max необходимо с осторожностью, в связи с некоторой нестабильностью этого инструмента.

Глава 5. Сигаретный дым

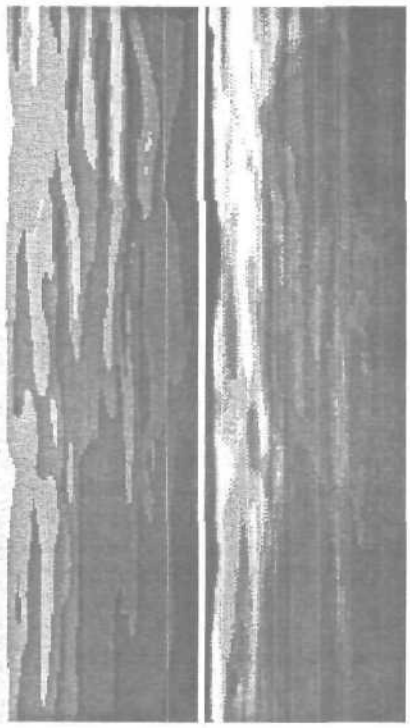
Довольно часто при создании реалистичной 3D сцены приходится сталкиваться с проблемой создания реалистичного дыма — например сигаретного.

Сразу хочу предупредить, что такой способ создания сигаретного дыма рассчитан прежде всего па статичные сцены, анимировать его довольно сложно, но при этом такой способ дает прекрасные результаты.

Смею предположить, что сигаретный дым на картинке практически не отличим от настоящего. Вся сцена была выполнена с помощью RhinoCeros и 3D Max без применения плагинов.

Итак, давайте разберемся, как же выполнен дым на этой картинке. Весь дым, это не что иное, как геометрическая фигура, выполненная из двух извивающихся кривых в Rhino. Достаточно легко сделать подобную геометрическую фигуру, необходимо только помнить, что при ее моделировании надо следить, чтобы плоскость не пересекала саму себя, т.к. это приведет к появлению резких границ в дыме при финальном рендеринге. Rhino и Max при создании **NURBS** генерируют собственные UV координаты, что существенно облегчает текстурирование.

Для создания реалистичного сигаретного дыма нам потребуется правильно текстурировать полученную фигуру. Для этого сделаем две текстурные карты.



Карту **diffuse** можно получить распылив белый, голубой и темно-голубой цвета в одном направлении, затем применить **Motion Blur** — вы можете добиться таких же результатов и другими, более удобными для вас способами. Вторая карта — opacity, это черно-белый вариант первой карты, слегка более контрастированная и с затемненными краями, для обеспечения мягких переходов. Осталось применить эти текстуры к нашему дыму. Для этого создадим материал со следующими параметрами:

blue ambient color (RGB: 13 / 130 / 255)
filter = 130
shinyness = 0
sh.strength = 0
self illumination = 70 (значение этого параметра зависит от общий освещенности сцены)

Затем добавим diffuse map. Дальше поработаем над прозрачностью с использовал mix material с черным цветом в одном слоте и нашей картой прозрачности в другом, в качестве filter использован falloff material, пользуясь настройками этого материала можно добиться максимального сходства с настоящим дымом.

Как вы видите, нет ничего сложного в создании сигаретного дыма — ни Particles, ни Volume Apparatus, ни эффекты Video Post не были использованы — весь дым, это *чистая* геометрия. Разумеется, этого существенно осложняет его анимацию, но при желании вы можете поэкспериментировать с ripple space warp — может это решит проблему.

Глава 6. Создание VRML

Продукт компании Discreet — 3D Studio Max обладает встроенным экспортером в VRML формат, который способен переводить MAX файлы в VRML.

Для работы нам потребуется не только 3DS Max, но и VRML вьювер, а также текстовый редактор. Кроме того, если вы собираетесь работать с текстурами, необходимо иметь программу обработки 2D графики (Photoshop).

VRML вьювер

Самый простой способ просмотреть VRML файл — это воспользоваться стандартным браузером (Netscape Navigator, Communicator или Microsoft Internet Explorer), снабженным VRML плагином. Помните, что различные плагины и вьюверы рендерят файлы по-разному, и если вы создаете VRML для широкой публикации — будет лучше если вы протестируете его в разных программах.

Текстовый редактор

Подобно HTML VRML — это формат, основанный на ASCII и вам понадобится текстовый редактор для проверки и модификации кода, к примеру MS Word.

Графический редактор

Всем понятно, что при создании текстур программу, подобную Photoshop трудно заменить.

Введение в VRML

Давайте разберемся — что же такое VRML? Данный формат напрямую связан с понятием Virtual Reality (VR) — интерактивный трехмерный мир, по которому пользователь может перемещаться в зависимости от своего желания. В этом мире пользователь называется «avatar». Virtual Reality Modeling Language (VRML) — это ASCII язык, используемый для описания подобных миров. Подобно HTML, это язык, не являющийся чьей-либо собственностью и его может использовать каждый совершенно свободно, он поддерживает использование текста, графики, анимации и звука и был принят за общий стандарт. VRML браузеры осуществляют рендеринг таких миров в реальном времени.

При создании 3D сцен для последующего экспорта в VRML вы всегда должны пользоваться следующими правилами:

Минимизация количества полигонов

VRML сцены не должны содержать больше 3000 полигонов, в противном случае могут возникнуть проблемы, и не только со скоростью загрузки.

Анимация с использованием простейших трансформаций

Анимация простейших трансформаций (масштабирование, перемещение, вращение) занимает гораздо меньше места в файлах VRML, чем анимации, использующая координатную интерполяцию. При координатной интерполяции VRML экспортер вынужден просчитывать каждую вершину отдельно, что отнимает время и существенно увеличивает размер конечного файла. Координатная интерполяция используется при анимации с использованием spacewarps, **taper,** модификаторов **bend и twist,** а также при анимации с использованием стека модификаторов и параметров объекта.

Материалы

Пользуйтесь исключительно материалами типов **standard** и **multi/sub-object.** Другие типы материалов (composite, **morpher, raytrace**) экспортироваться не будут.

Текстуры

Использование текстур увеличивает размер конечного файла, поэтому пользуйтесь текстурами небольшого размера и не злоупотребляйте ими.

Освещение

Если вы используете только стандартное освещение Max, скорее всего объекты вашей сцены не будут достаточно освещены. Обычно бы-

вает необходимо добавить дополнительные источники освещения и при этом учитывать возможные манипуляции и передвижения пользователя. Если вы используете несколько источников освещения — уменьшите у них параметр multiplier, иначе сцена будет слишком яркой.

Камеры

Обязательно используйте хотя бы одну камеру — она будет стартовой точкой вашего мира. При использовании нескольких камер вам будет предложен выбор — какую из них сделать стартовой. Все камеры сиены экспортируются в VRML файл и пользователь сможет легко переключаться между ними. Таким образом, использование нескольких камер в разных точках сцены сделает ее более удобной для изучения. Имя камеры становится именем вьюпорта, поэтому давайте камерам разумные имена, чтобы пользователь знал, куда отправит его та или иная камера.

Полигоны

Перед экспортом в VRML необходимо прятать те полигоны, которые невозможно увидеть. Спрятанные полигоны **(hidden** faces) не будут экспортироваться, что позволит существенно сократить размер файла.

Объекты

Используйте примитивы **(sphere,** cylinder, **cone, box** и т.д.) где только возможно. Конвертирование объекта в editable **mesh** увеличит размер файла даже при минимальных изменениях геометрии.

Основные шаги при создании VRML сцен

Ниже приведен наиболее выгодный порядок действий по созданию VRML сцен.

- создание геометрии;
- создание источников освещения и камер;
- присваивание объектам материалов и текстур;
- анимирование;
- вставка вспомогательных VRML-объектов;
- экспорт в VRML;
- использование барузера для тестирования файла;
- редактирование файла в текстовом редакторе (при необходимости).

Вставка вспомогательных VRML-объектов

Без вспомогательных VRML-объектов (можно назвать их частью VRML-интерфейса) пользователь не сможет передвигаться по виртуальному миру и правильная их настойка может сделать созданный вами мир по-настоящему интерактивным.

Первый шаг является общим по созданию любого вспомогательного объекта (helper object):

- Откройте панель Creation и выберите вкладку Helpers.
- В выпадающем меню выберите VRML.
- Выберите тип объекта (Anchor, **AudioClip,** Background и т.д.), нажав на соответствующую кнопку.
- Во вьюпорте Max кликните мышкой, чтобы создать объект. Если вы не уверенны в своих действиях — создавайте объект в **Top viewport.**

Создание двух типов вспомогательных объектов — NavInfo и TouchSensor

NavInfo

Этот объект сообщает браузеру следующую информацию:

- какой тип навигации разрешить;
- возможно ли пользователю включить **«headlight»;**
- как далеко пользователь может видеть;
- как быстро может передвигаться пользователь;
- размер пользователя в виртуальном мире (например — используется для расчета столкновений);
- как близко может подойти пользователь к предмету не столкнувшись с ним;
- высота пользователя над уровнем ландшафта;
- как высоко может подняться пользователь;
- размер вспомогательного объекта **NavInfo** во вьюпорте Max (не влияет на окончательный размер файла).

В NavInfo справа находятся необходимые параметры. Приведем их подробную расшифровку:

The navigation type - «FLY».

```
The «headlight» - но отмечен.
The Visibility Limit - 1000.
Speed - 3.0.
Avatar Size:
Collision: 0.25
Terrain: 1.6
Step Height: 0.75
Icon Size is 144.605
```

Значения параметров

-- TYPE --

Этот параметр назначает способ, которым пользователь (avatar) перемещается по миру — **WALK, EXAMINE, FLY, NONE.** NavInfo не позволяет присваивать несколько значений этому параметру, но существует возможность отредактировать полученный файл вручную и добавить дополнительный способы передвижения — пример приводится ниже:

```
DEF NavInfo01 NavigationInfo {
avatarSize [0.25, 1.6, 0.75]
headlight FALSE
speed 3
type «EXAMINE, FLY, WALK»
}
```

-- HEADLIGHT --

Включение этой опции добавляет дополнительный направленный источник освещения, всегда направленный в сторону взгляда пользователя. Всегда лучше отключить этот параметр и добавить свои дополнительные источники освещения. С другой стороны, если вы поленились как следует осветить ваш мир — включите эту функцию, иначе ваш мир будет выглядеть черным.

Согласно спецификации VRML, параметр Headlight должен определять — разрешено или нет пользователю включать этот источник освещения, в реальности же, в Cosmo Player он просто указывает, включен ли этот источник сразу, и даже при отключенном параметре пользователь может при желании включить **HEADLIGHT.**

-- VISIBILITY LIMIT -

Этот параметр назначает расстояние, при котором объекты становятся видимыми пользователю. Если он выставлен на 0, пользователь сможет видеть всю сцену сразу.

-- SPEED --

Стандартная скорость перемещения равна I, но удобнее пользоваться большими значениями — поэкспериментировать со значениями от 3 до 5.

-- AVATAR SIZE --

Этот параметр используется в основном для расчета столкновений (collision).

-- COLLISION --

Минимальное расстояние, на которое пользователь может приблизиться к объекту, не столкнувшись с ним.

-- TERRAIN --

Высота пользователя над поверхностью.

-- STEP HEIGHT -

Максимальная высота объекта, через который пользователь может переступить не столкнувшись с ним.

TouchSensor

Пользоваться этим объектом достаточно легко — после создания объекта, вы просто выбираете объект, который будет включать действие а затем объект, который будет выполнять действие. Пользователь, кликнув на объект-переключатель, заставляет объект-исполнитель выполнить назначенную ему анимацию.

Экспорт в VRML

Вовремя экспорта в VRML у вас появляется возможность сконфигурировать некоторые важные параметры:

- **Normals** — генерирует нормали, увеличивая размер файла Выбирайте этот параметр если в вашей сцене используются **smoothing groups;**
- **Indentation** — делает код файла более читабельным;
- **Primitives** — экспортирует собственные VRML-примитивы вместо примитивов Max — уменьшает размер файла;
- **Color Per Vertex** — экспортирует цвет вершин для геометрии. Если вы выбираете этот параметр, вам необходимо будет выбрать источник — Vertex **Color Source**

- использовать ли настройки Max, основанные на освещении и материалах;

- **Coordinate** Interpolators — включает возможность экспорта анимации, основанной на координатной интерполяции. Если вы помечаете этот параметр, экспортер предупредит вас о возможной несовместимости типов анимации;
- Export Hidden Objects - позволяет экспортировать спрятанные объекты, делая их видимыми в VRML;
- **Flip-Book** — экспортирует сцену в несколько файлов с различным **sample** rate;
- Polygons **Type** — определяет, каким образом геометрические грани будут описываться в VRML файле. По умолчанию установлен «triangles» — создает треугольные грани, **«quads»** — создает четырехугольные грани где возможно, если такой возможности нет — пользуется треугольными. «Ngons» — создает максимально возможное число граней. «Visible edges» — разрывает грани на видимых углах;
- Initial **View** — выдает список всех камер и вьюпортов сцены и позволяет выбрать начальный вид;
- **Initial Navigation Info** — выбирает объект NavInfo, который будет загружаться со сценой;
- **Initial Background** — выбор одноименного *вспомогательного* объекта для загрузки со сиеной;
- **Initial Fog** — выбор одноименного вспомогательного объекта для загрузки со сценой;
- **Digits of Precision** — генерация глубины изображения — по умолчанию 4, уменьшение числа уменьшает размер файла, но более высокое значение может потребоваться если какие-либо предметы находятся более чем в 100.000 юнитов от центра;
- Show **Progress Bar** — визуализирует процесс создания VRML файла;
- **Bitmap URL prefix** — указывает на адрес расположения текстурных карт использованных в сцене;
- **Samples Rate** — влияет на количество ключей анимации - понижение значения уменьшает размер файла;

- World Info — некоторые браузеры позволяют показывать информацию о сиене.

Вот собственно и все. Думаю с помощью этого урока создание миров VRML станет для вас простым делом.

Глава 7. Морской пейзаж

Эта глава о том, как получить реалистичный материал воды для поверхности моря.

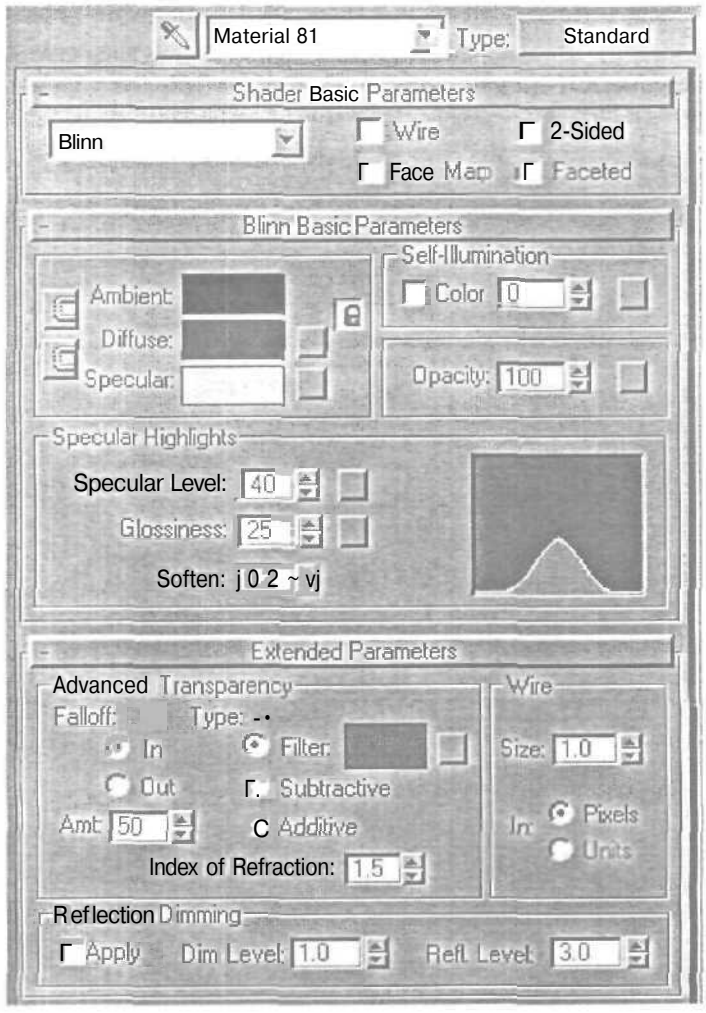
Вначале давайте установим фон нашей сцены. Для этого выберите в меню **Rendering** пункт Environment. В появившемся окне кликните по кнопке **Environment Map и** подберите для фона любую подходящую текстуру неба.

Создайте объект **Plane** длинной примерно 200 (параметр Length) и шириной около 100 (параметр Width). Количество сегментов по длине и ширине (параметры **Length** Segs и **Width** Segs) должно быть достаточно большим, — от 50 до 100. Это пригодится нам в дальнейшем.

Перейдем в раздел Modify и добавим нашему объекту модификатор Noise. Включите параметр Fractal, Установите **Roughness** равным 0.1, а **Iterations** — 8.0. Немного поэкспериментируйте с этими параметрами чтобы получить подходящий результат. Параметр **Strength** по оси **Z** уста-

новите равным 20 или около того. Это значение определяет насколько большими будут волны. У нас получилась волнистая поверхность, нужно добавить материал.

Откройте редактор материалов и выберите любой слот. Убедитесь, что тип заливки — Blinn. Установите те же параметры, что и на рисунке:

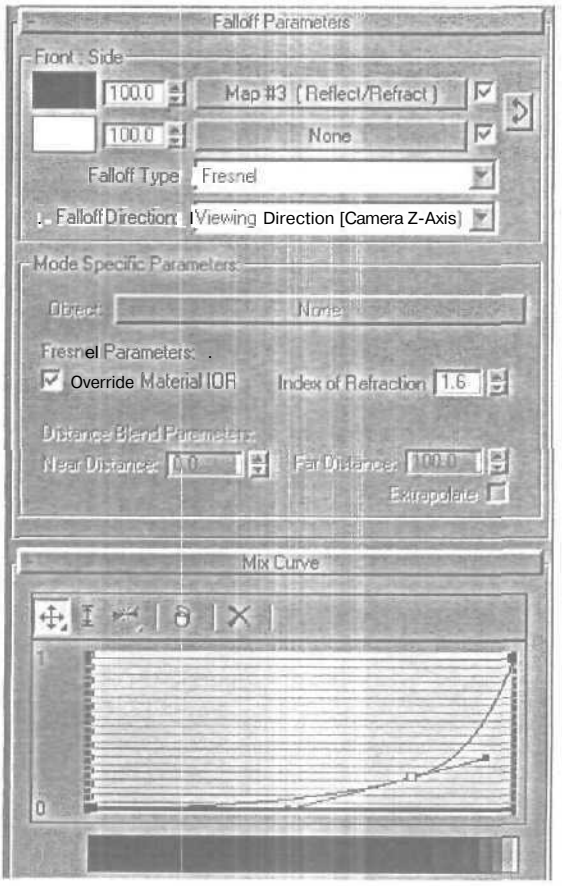


В текстурах включите карту шероховатости **(Bump) и** установите ее размер 20.

Кликните по кнопке Bump Map и выберите Noise. Установите тип шума (Noise type) — Fractal, а размер шума (Size) — 0.5. Вернитесь обратно, используя кнопку **Go To** Parent.

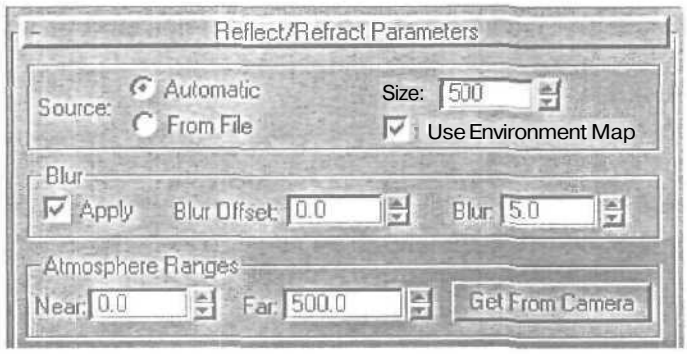
Включите карту отражении **(Reflection) и** установите ее размер 40.

Кликните по кнопке **Reflection Map** и выберите Falloff. Установите те же параметры, что и на рисунке. На графике **Mix Curve** добавьте третью точку и постройте график, подобный изображенному на рисунке.



Кликните вверху по кнопке напротив окошка с черным цветом и установите Reflect/Refract.

Установите параметры так же, как на рисунке. Это заставит материал отражать небо, заданное в качестве фона нашей сиены.



Вот и все. Присвойте полученный материал объекту **Plane.** Осталось установить нужный вид в окне **Perspective и** отрендерить. Если вы все сделали правильно, то у вас должен получиться небольшой морской пейзаж.

Глава 8.
Пузырь, еще пузырь...

В данной главе будет рассказано о том, как можно создавать мыльные пузыри, не затрачивая при этом особых усилий. Не пугайтесь! Вам не придется создавать и анимировать сотни сфер! Более того, вам даже не понадобится пользоваться никакими дополнительными модулями! 3D Studio Max и любой графический редактор (подойдет даже Windows'овский Paint) — все, что нам сегодня потребуется.

На первый взгляд, создание пузырей — дело весьма трудоемкое и может отнять кучу времени. Однако, если вспомнить про системы частиц и попробовать наложить на них текстурную карту, то результаты получатся поистине впечатляющие. Давайте с вами и попробуем сейчас это сделать.

Создание текстурной карты

Прежде всего, создадим текстурную карту с изображением одного пузыря. Для этого в центре нашей сцены поместим сферу.

Можно использовать следующие параметры:

X, Y, Z: 0, 0. 0
Radius: 100
Segments 64

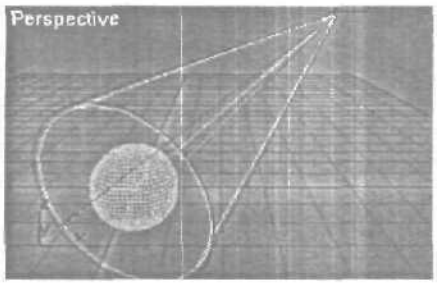
Теперь займемся освещением. Добавьте в сцену источник света типа Target Spot с примерно следующими мировыми координатами:

Spot - X, Y. Z: 500, -200, 500
Spot Target - X Y, Z: 0, 0, 0

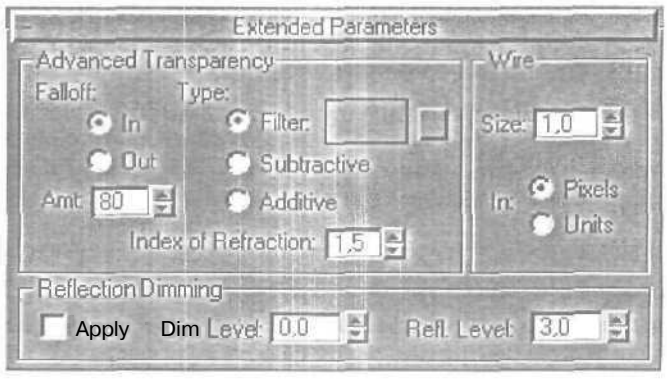
А цвет этого источника света сделайте белым:

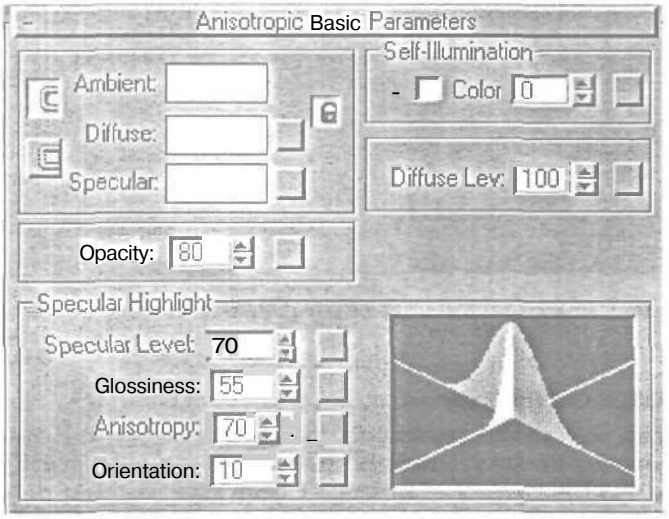
R, G. B: 255, 255, 255

Примерно так должна выглядеть ваша сцена на данном этапе:



Теперь настала пора открыть Редактор Материалов (Material Editor). Активизируйте свободный слот (Slot) для создания нового материала. В разворачивающемся списке установите тип Shading в Anisotropic. Заприте цвет Ambient вместе с цветом Diffuse, придав обоим значения R, G, B: 255, 255, 255 (чисто белый цвет). Остальные параметры установиге такими, как на следующих рисунках:





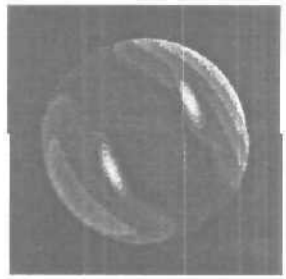
Заметьте, что цвет фильтра **(Filter)** в **расширенных параметрах** (Extended **Parameters)** имеет значение:

R, G, Б: 200, 200. 200.

Теперь примените получившийся материал к сфере. Отрендерьте (визуализируйте) сцену в окне вида Перспектива (Perspective), предварительно установив опцию **Force 2-Sided** (Меню **Rendering ⇨ Render,** в группе Options). Результат рендеринга (визуализации) сохраните.

Должен заметить, что наши настройки материала для пузыря кому-то могут показаться не идеальными. Каждый из вас не только может. но и должен попробовать создать свой пузырь! Экспериментируйте с настройками. Критичным в этой сцене является лишь цвет фона. Поскольку мы будем использовать полученную карту не только как диффузную, но и как карту непрозрачности, то цвет фона должен оставаться черным. По умолчанию так оно и есть. но если у вас цвет фона другой, вы можете его заново установить — меню **Rendering C Environment,** и затем щелкните па образце цвета **Background color.**

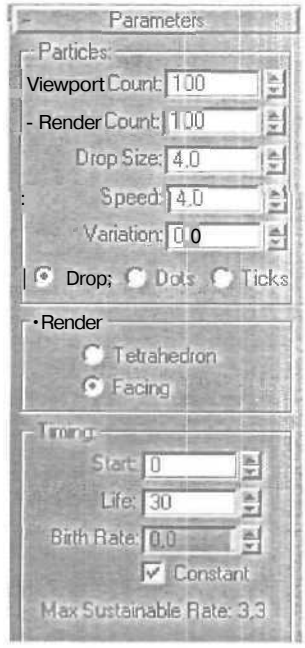
В PhotoShop или любом другом графическом редакторе обрежьте полученную текстурную карту так, чтобы она стала квадратной. Иначе, все наши пузыри будут вытянуты (обратите внимание, экспериментируя с искажением текстурной карты, вы можете получить довольно забавные результаты). Сохраните полученное изображение. Наконец, текстурная карта для наших пузырей готова.



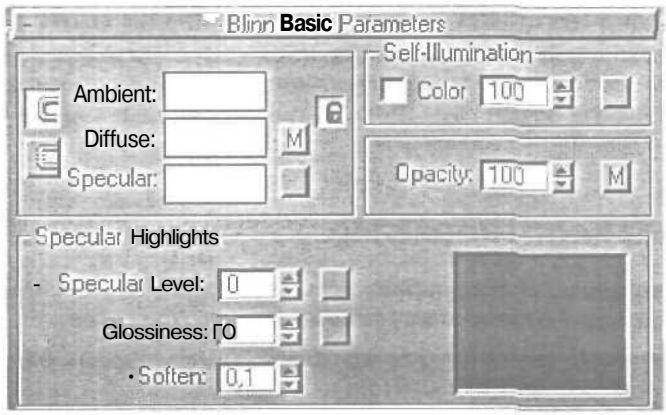
Теперь можно закрыть эту сцену. Она нам больше не понадобится.

Создаем пузыри

Итак, пора творить чудеса! Открываем новую сцену. Идем на командную панель Create ➪ Geometry, далее в разворачивающемся списке выбираем Particle Systems и к свитке Object Type щелкаем по кнопке Spray. Создайте данный объект в своей сиене, просто шелкая и перетаскивая курсор. Установите параметры, показанные на рисунке:



Теперь нужно создать материал для нашей системы частиц. Открываем Редактор Материалов (Material Editor) и, активизировав свободный слот, устанавливаем значения в соответствии с рисунком:



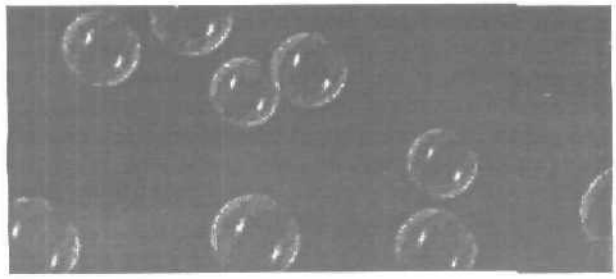
В свитке расширенных параметров (Extended Parameters) на этот раз изменяем только цвет фильтра:

Filter - R, G, B: 150, 150, 150

Далее, разворачиваем свиток **Maps и** загружаем в параметр Diffuse карту, которую мы ранее приготовили. Затем, грузим эту же самую карту в параметр **Opacity** (можно просто перетащить карту с кнопки для **Diffuse** на кнопку для Opacity).

Теперь осталось лишь назначить материал нашему объекту Spray. Делаем это. Перетащим ползунок анимации **(Time** Slider) из нулевого кадра в, скажем, тридцатый кадр. И перед рендерингом (т.е. визуализацией) не забудем включить опцию **Force 2-Sided.**

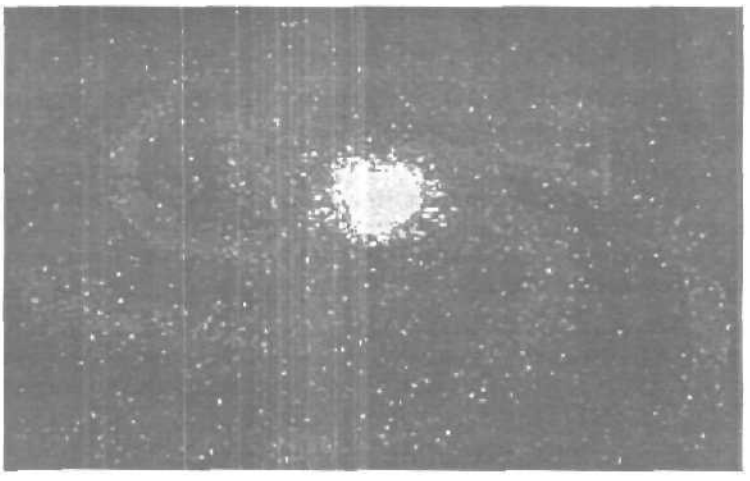
Вот и все! Каковы затраты, но каков результат!



Сделайте рендеринг активного временного сегмента в файл, получите эффектный AVI'шник. Если всерьез браться за анимацию нашего Spray'я, то подумайте о том, чтобы связать его с каким-нибудь исказителем пространства, например ветром **(Wind).** Кстати, ПУЗЫРИ это не единственное, для чего можно использовать данную технику. Какна счет того, чтобы организовать листопад в своей сцене? Не бойтесь экспериментировать с параметрами нашего объекта Spray. Настраивайте их под нужды своих сцен.

Глава 9. Спиральная галактика

Ниже будет продемонстрировано, как создать спиральную галактику, используя системы частиц типа **Spray и** эффекты **Glow.**



Итак, начните новую сиену. В окне проекции **Front** создайте объект типа **Spray.** На панели **Create** щелкните на кнопке **Geometry** и выберите из разворачивающегося списка **Particle Systems** вместо Standard **Primitives.** Убедитесь, что ось объекта находится в самом центре. Для этого щелкните на кнопке **Move,** выделите объект **Spray** и вновь щелкните на кнопке **Move,** но теперь уже правой кнопкой мыши. В появившемся окне **Move Transform Type-In** установите все значения в 0 (просто щелкая правой кнопкой на соответствующих спиннерах).

На панели **Modify** установите параметры для нашего объекта Spray следующим образом:

Viewpoint Count: 100
Render Count: 1000
Drop Size: 15
Speed: 2
Variation: 0
Render: Tetrahedron
Start: -200
Life: 200
Constant: On
Emitter Width: 30
Emitter Length: 10

Если частицы слишком велики или слишком малы, параметр Drop Size можно изменить. Мы устанавливаем параметр **Start** в **-200,** потому что мы собираемся «закрутить» нашу галактику, чтобы получить эффект спирали.

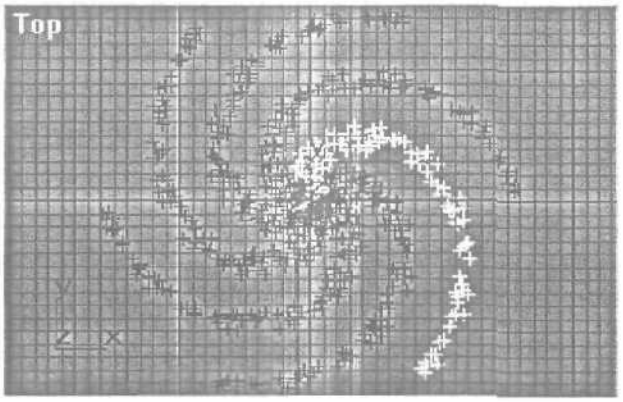
Щелкните на кнопке Animate, установите ползунок анимации в кадр 100. Принимал во внимание, что вы начали новую сцену с установленным по умолчанию активным сегментом в 100 кадров, это конечный кадр анимации. Теперь в окне проекции **Top** вращайте наш **Spray** на 90 градусов (ось Z). Проще всего воспользоваться угловой привязкой: щелкните на переключателе **Angle** Snap **Toggle.** Выключите кнопку Animate. Итак, вы создали первый объект Spray.

Щелкните на нем правой кнопкой мыши, выберите из контекстного меню **Properties.** Установите **Object Channel** в **1.** Это идентифицирует нашу систему частиц, когда мы будем применять эффект Glow.

Теперь предстоит часть посложнее. Откройте **Track View и** разверните так, чтобы все треки объекта Spray были хорошо видны. Мы собираемся переместить трек Transform влево. Это вынудит начать анимацию еще до первого кадра, поэтому, когда вы отрендерите (визуализируете) первый кадр, результат более будет походить на спираль, чем на частицы, выстроенные в линию. Переместите черную линию влево так, чтобы ее крайняя точка находилась в нулевом кадре, и чтобы вся эта черная линия очутилась в темно серой зоне.

Выберите трек **Rotation.** Щелкните на кнопке вызова диалога Parameter **Curve** Out-of-Range **Types** и выберите из предлагаемых вариантов Relative **Repeat.** Это заставит наш **Spray** вечно продолжать скручивание. Другими словами, вы сможете изменять число кадров анимации, но процессу скручивания это не повредит.

Теперь давайте создадим массив из нашего объекта. Выделите Spray, и поверните его в окне проекции Top на 60 градусов при нажатой клавише **Shift.** Установите число копий в 5. После этого сцена должна выглядеть примерно так:



Клонируйте один из полученных объектов Spray, но не перемещая его. Просто выберите **Clone** из меню **Edit.** На панели Modify установите параметры **Emitter Width в 100 и Render** Count в 500. Это придаст более естественный вид.

Создайте еще один массив объектов **Spray,** добавив 5 копий через каждые 60 градусов, точно так же, как на шаге S.

Большинство галактик имеет плотный центр, поэтому нашим намерением на данном этапе является создать желтый центр. В окне проекции **Top** создайте объект **PCIoud (Create ➪ Geometry ➪ Particle Systems ➪ Object Type)** и поместите его в центре сцены. Установите параметры следующим образом:

Свиток Basic Parameters:

Particle Formation: Sphere Emitter
Rad/Len: 55

Свиток **Particle Generation:**

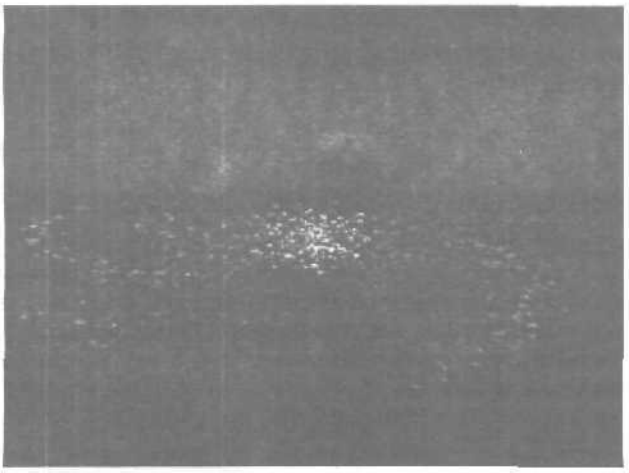
Use Rate: 600
Display Until: «длина вашей анимация»
Life: «длина вашей анимации»
Size: 2

Оставьте **все остальное, как** есть.

В окне проекции Front уменьшите масштаб эмиттера объекта PCloud вдоль оси Y, используя неоднородное масштабирование (Non-Uniform Scale). Щелкните правой кнопкой мыши на эмиттере объекта PCloud и выберите Properties. Установите Object Channel в 2.

Откройте Редактор Материалов и установите цвет Diffuse в светло синий, например: R — О, G — 126, В = 255. Установите Specular Level и Glossiness в 0, а Self-Illumination в 100. Создайте еще один точно такой же материал, но установите цвет Diffuse в светло оранжево-желтый, например: R = 255, С = 207, В = 0. Назначьте синий материал всем объектам типа Spray, а второй материал — объекту типа PCIoud.

Создайте камеру. Поместите ее так, чтобы она находилась перед галактикой. Когда вы отрендерите (визуализируете) сцену, должно получиться примерно следующее изображение.



В меню Rendering откройте Video Post. Щелкните на кнопке Add Scene Event. Из разворачивающегося списка выберите камеру, которую вы только что создали. Затем щелкните на кнопке Add Image Filter Event и выберите Lens Effects Glow. Щелкните на кнопке Setup, и установите следующие параметры;

Вкладка **Properties:**
Source: Object ID: 1
Filter: Perimeter Alpha
Вкладка Preferences:
Size: 3
Intensity: 50

Щелкните на кнопке OK. Создайте другой фильтр **Lens Effects Glow со** следующими назначениями:

Вкладка **Properties:**

Source: Object ID: 2
Filler All

Вкладка **Preferences:**

Size: 4
User: R: 255, G: 199, B. C
Intensity: 45

Создайте эффект Starfield со следующими параметрами:

Dimmest Star: 140
StarSize: от 1 до 1.5
Motion Blur: Use: Off
Count: 60000

Отрендерьте изображение при разрешении 400x300, и ваша спиральная галактика готова!

Глава 10. Создание модели скрипки методом NURBS моделирования

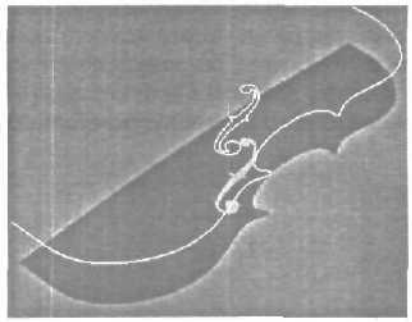
Сушествует бесконечное количество методов и техник, которые нужно освоить, необходимо знать возможности программы, изучить интерфейс и так далее. Но с точки зрения художника, вам необходимо развивать инстинкты и нарабатывать методы, чтобы облегчить творческую работу.

Этим инстинктам и методам можно научиться, либо выучить путем наблюдения за работой профессионалов. Основным условием творческой работы является способность представить себе проект в целом, обобщить задачу. В большинстве случаев работа с трехмерной графикой состоит в решении стратегических задач. Для достижения профессионального уровня работы, необходимо в самом начале изучить все поставленные задачи и рассмотреть все возможные стратегические альтернативы. Желательно проделать эту процедуру несколько раз, т.к. и большинстве случаев решения, принятые в самом начале работы, вдальнейшем ограничивают творческую свободу. Необходимо точно продумать, каким образом двигаться от общего к частностям, и сделать это таким образом, чтобы существовала свобода добавлять новые детали в дальнейшем, не загоняя себя в угол.

Модель скрипки была сделана в 3DS Max, но в ней нет ничего уникального и эта модель может быть создана в любой программе, имеющей инструменты **NURBS.**

На рисунке вы видите законченную модель скрипки. Конечно эта модель не является совершенством и те, кто хорошо знаком с музыкальными инструментами конечно найдет некоторые изъяны.

Для начала нужно выделить основные объекты, которые можно смоделировать по отдельности: корпус, гриф, подставку, струнодержатель и проч. Корпус скрипки был самым большим объектом и поэтому решено было начать с него. Наилучшим способом было вырезать из плоского объекта нужную форму (trimming), а затем, с помощью другой формы вырезать f-прорези. Обе эти задачи были достаточно сложны. Пришлось сначала создать половину поверхности, таким образом было необходимо вырезать лишь одну f-прорезь в одной поверхности.



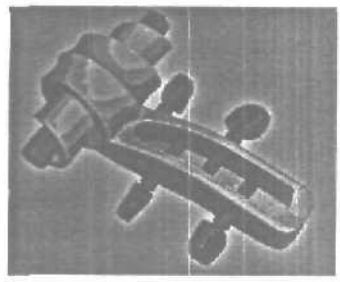
Но у этого метода есть свои недостатки, т.к. деки у скрипки слегка выпуклы, и чтобы сымитировать эту выпуклость, и одновременно предотвратить появление видимых швов на месте стыка двух половинок было очень сложно.

Затем была создана нижняя дека с помощью зеркального копирования верхней, кривая, образующая обрез деки была экструдирована для создания боковины. Отдельных усилий потребовало создание небольших выпуклостей на местах стыка боковины и обоих дек. Эти элементы были созданы из смешанных поверхностей (blend surfaces) для обеспечения целостности и затем присоединены к модели. Чтобы выполнить эту задачу потребовалось много экспериментировать.

Следующим этапом было создание шейки. Для создания этого элемента был выбран инструмент **Path Deformation.** Был нарисовал сплайн **NURBS** в форме нужной нам спирали, затем создана модель шейки с характерными изгибами углов и деформацией поверхности. Осталось только применить к полученной модели нужный модификатор. После этого редактируя сплайн спирали можно контролировать изгиб шейки.

Недостатком этого метода является то, что с его помощью невозможно создать хорошего завершения для завитка. Пришлось идти на компромисс и создавать новый объект. Этот объект конечно не смог абсолютно точно вписаться в шейку, но недостатки настолько малы, что их можно не принимать во внимание.

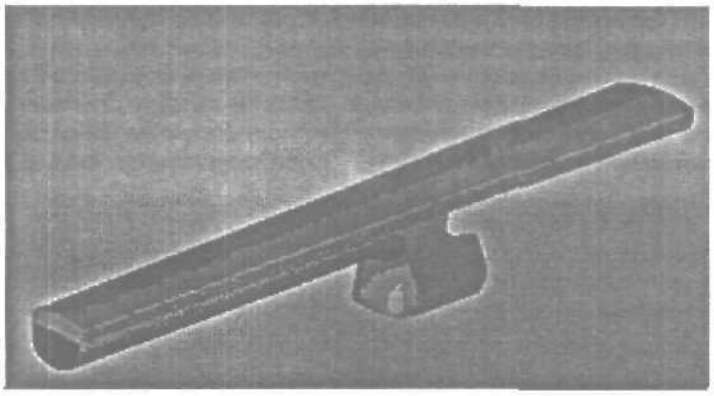
Чтобы создать пространство для размещения колков, понадобилось вырезать кусок поверхности шейки, экструдировать полученное отверстие вовнутрь и создать поверхность для дна. Сами колки это объекты, созданные с помощью модификатора **Lathe** и затем обрезанные по краям.



Изгиб грифа также потребовал серьезных размышлений — был нужен плавный изгиб, но при этом необходимо было сохранить острые грани.

Для решения этой задачи существует множество путей, но решено было создать объект с помощью модификатора **Loft,** а затем применить Bend для получения изгиба. Затем обрезаем лишнее для получения острых граней. Понадобилось «всего» четыре среза, т.к. часть грифа срыта внутри скрипки и не требует детализации.

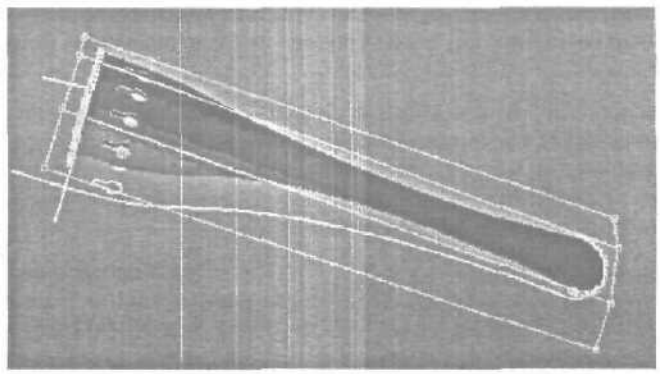
Для моделирования панели на грифе была использована та же кривая, что и для обреза грифа. Конечно пришлось удлинить ее и слегка модифицировать. После этого было достаточно легко соединить оба объекта в одну бесшовную структуру.



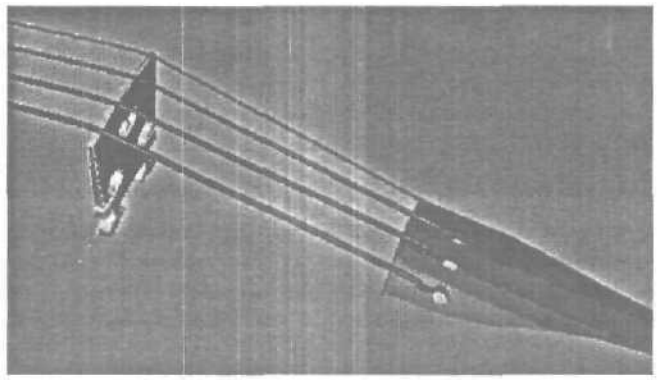
Подставка — это было еще одно серьезное упражнение в «выпиливании по объекту». Фигурные отверстия были сделаны с помощью проецирования шаблона на поверхность для обрезки. Эта задача была довольно сложна, т.к. решено было в этот раз избежать моделирования по частям.

Струнодержатель тоже создавался путем многочисленных вырезов. Как и в случае с корпусом скрипки пришлось Применять внешние и внутренние отсечения, но теперь отсекающие кривые не были такой сложной формы.

Были использованы две кривые для формирования объекта, затем вырезано по очереди четыре одинаковых прорези — в итоге получилось шесть, но это сработало, поскольку ни один вырез не был сложным.



Струны тоже были утомительной работой, но принцип был прост. Были экструдированы маленькие округлые кривые вдоль пути, обозначенного еще одной кривой — в максе этот процесс называется **Rail Sweep.** Этот метод предоставляет довольно большую свободу действий, можно изменять толщину струн, а также редактировать кривую пути для точного совпадения с прорезями в струнодержателе и наматывания на колки.



В процессе работы регулярно собирались все объекты вместе, чтобы посмотреть что получается. В конце все равно оказалось, что части не подходят друг к другу и потребовалось некоторое время, чтобы настроить все объекты и собрать вместе. Когда работа была закона, скрипка выглядела как показано на рисунке, готовая к экспериментам с материалами.

Метод моделирования, описанный выше, достаточно сложный для восприятия, и если вы только начинаете работать с поверхностями и кривыми **NURBS,** то у вас обязательно возникнет немало трудностей, когда вы попытаетесь реализовать материал главы на практике.

Некоторые советы по моделированию NURBS

- Для вас гораздо легче будет нарисовать любой профиль для обрезки простым инструментом «линия» (Line) и затем присоединить **(Attach)** полученную форму к вашей **NURBS** поверхности. При этом кривая, для сохранения точной формы будет разбита на отрезки. Эти отрезки могут быть по очереди спроецированы на плоскость для получения формы выреза. Перед проекцией можно объединить отрезки методом **Join,** но при этом необходимо быть готовым к тому, что кривая потеряет первоначальную форму.
- Вырезы выполняются инструментом **Multicurve Trimmed Surface.** Для этого кривая-шаблон, по которой производится обрез должна быть приаттачена к **NURBS** структуре и находиться на поверхности объекта. Для этого необходимо либо создать кривую прямо на плоскости **(Point Curve on Surface),** либо спроецировать на плоскость приаттаченую кривую (Normal **Projected Curve** или **Vector Projected Curve).**
- При создании любых поверхностей **NURBS** необходимо следить за направлением нормалей **(Normal),** при необходимости выполняя операцию **Flip** Normals. Либо,

если вы не хотите об этом задумываться — примените к объекту материал с пометкой 2 Sided.

- **Самым** сложным объектом действительно **оказалась** шейка. В результате, после многих экспериментов, шейка была создана инструментом **1 Sweep Rail.** Для этого необходимо нарисовать сплайн спирали, и затем разместить на ней профили шейки, которые тоже представляют собой замкнутые сплайны. При использовании этого метода необходимо следить, чтобы профили обязательно пересекались с кривой пути, иначе результаты будут непредсказуемые. Также надо будет отрегулировать настройки по умолчанию полученной поверхности.

Напоследок надо отметить, что вся модель скрипки состоит только из поверхностей и сплайнов **NURBS,** без применения других объектов.

Глава 11. Футбольный мяч

Эта глава поможет вам создать «настоящий» футбольный мяч в 3D Studio MAX. Метод, описанный в нем, более продвинутый, нежели создание обычной сферы с наложенной на нее текстурой. Дело в том, что мяч такого типа может разглядываться с близкого расстояния и еще у него есть небольшие трещины между кожаными частями, которые формируют мяч.

1. Во первых, запустите 3D Studio MAX, и разверните окно с видом «Top» используя кнопку «Min/Max **Toggle*.**

2. Теперь создайте n-угольник (NGon) выбрав **«Create/Shapes ➪** NGon», установите параметр «Circumscribed» и количество сторон равным 5. Зажмите левую клавишу мыши в окне проекции и тащите пока, радиус n-угольника не будет равен примерно 100.

3. Щелкните по кнопке «Select and move», а затем еще раз, но уже правой кнопкой мыши.

4. В появившемся окне «Move **Transform Type-In»** в форме «Absolute: World» введите 0 во всех трех полях. Теперь центр вращения n-угольника будет расположен в точке (0,0,0). Закройте окно.

5. То же, что в п.3, но с кнопкой «Select **and Rotate».**

6. В появившемся окне «Rotate **Transform Type-In»** в форме «Offset: **Screen»** наберите -90 в поле Z. Закройте окно.

7. Теперь в закладке **«Modify»** установите радиус n-угольника равным 68,819.

Как известно, мяч состоит не только из пятиугольников, поэтому нужно сделать еще один n-угольник с шестью сторонами.

8. Выберите **«Create/Shapes ➪** NGon», установите параметр **«Inscribed»,** а количество сторон сделайте равным 6. Зажмите левую клавишу мыши в окне проекции и тащите, пока радиус n-угольника не будет равен примерно **100.**

9. Сразу же после создания n-угольника щелкните на закладке «Modify» и в поле «radius» введите 100,

10. Теперь нам нужно выровнять оба n-угольника. Щелкните по кнопке «3D Snap **Toggle»** правой кнопкой мыши; в появившемся окне «Grid **and** Snap **Settings»** поставьте галочку напротив «Vertex». Закройте окно.

11. Нажмите «s», если кнопка **«3D Snap** Toggle» еще не активизирована.

12. Щелкните по кнопке «Select **and move»,** затем наведите курсор на одну из вершин шестиугольника, должен появиться маленький крестик, говорящий об активизации режима «Snap». Зажмите левую кнопку мыши и тяните все это дело к соответствующей вершине пятиугольника, после отпустите кнопку мыши.

У вас должны получиться 2 n-угольника, выровненные друг относительно друга, потому что их ребра равны.

Пока n-угольники не повернуты, неплохо будет задать им UVW координаты.

13. Выделите пятиугольник. Примените к нему модификатор «UVW Map». Теперь сделаем скриншот экрана, нажав **«Print Screen* на** клавиатуре. Загружайте ваш любимый графический редактор, вставляйте в него полученный образ и сохраняйте фрагмент в пределах желтых линий UVW карты. Данный фрагмент поможет вам создать текстуру, когда вы закончите моделирование.

14. То же самое нужно применить к шестиугольнику.

Теперь нужно сделать повернутые копии этих 2-х частей для образования мяча. Для этого мы должны изменить центр вращения шестиугольника.

15. Выделите шестиугольник, выберете **«Hierarchy ➪ Affect Pivot Only»** и щелкните на кнопке **«3D Snap Toggle»** правой кнопкой мыши, в окне **«Grid and Snap Settings»** поставьте галочку напротив **«Pivot».** Закройте окно.

16. Щелкните по кнопке «Select **and move» и** тяните центр вращения шестиугольника к левой верхней вершине пятиугольника. После этого центр вращения должен оказаться точно в этой вершине.

17. Нажмите **«Affect Pivot Only»** еще раз, что бы выключить этот режим.

18. Теперь, вместе с выделенным шестиугольником, щелкните по кнопке **«Select and Rotate»** сначала правой, затем левой кнопками мыши.

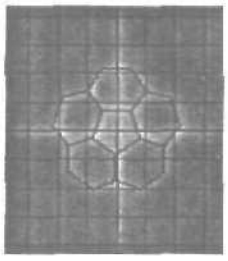
19. В появившемся окне **«Rotate Transform Type-In»** в форме **«Offset: Screen»** введите 37,377 в поле X, Закройте окно.

20. Щелкните по кнопке **«3D Snap Toggle»** правой кнопкой мыши и уберите галочку напротив **«Pivot».**

21. Выделите шестиугольник и наведите курсор на его левую нижнюю вершину, зажмите левую кнопку мыши и, удерживая **Shift,** потяните объект к другой вершине. Затем отпускаете все кнопки.

22. Теперь вращайте n-угольник на 72 градуса вокруг оси Z, выбрав, как и раньше «Select **and Rotate»;** вводить угол нужно в форме **«Offset: Screen»,** в поле Z.

23. Повторите это еще 3 раза (имеется в виду п.21 и п.22., а манипуляции, конечно же, надо производить с последним скопированным шестиугольником) и вы получите, то, что изображено на картинке.

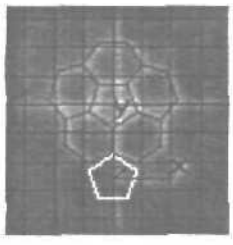


На этом этапе у нас уже почти готова нижняя часть мяча, но до конца еще далеко.

24. Выделите пятиугольник. Нажмите **«Select and Move»**, зажмите **Shift** и левой кнопкой мыши потяните объект вниз, так, что бы он оказался под мячом.

25. Оставив новый n-угольник выделенным, выберем «Select **and Rotate**» и повернем объект на 180 градусов вокруг оси Z.

26. Выберите «Select and Move» и наведите курсор на верхнюю вершину n-угольника. Затем, зажав левую кнопку мыши, потяните объект вверх и установите, как показано на картинке.



27. Оставляя объект выделенным, выберите **«Hierarchy ➪ Affect Pivot Only»**, а затем нажмите кнопку «Align» и щелкните на пятиугольнике.

28. После этого в форме **Align Selection** (Screen) поставьте галочку напротив «У **position»,** в форме **«Current Object»** выберите **«Pivot Point»,** а в форме «Target **Object»** выберите **«Maximum»** и нажмите **OK.**

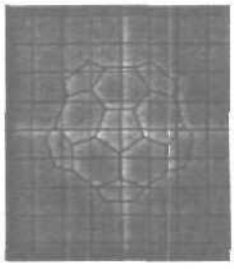
Теперь, надеюсь, центр вращения установлен в верхней вершине.

29. Нажмите «Affect Pivot Only» еще раз, что бы выключить этот режим.

30. Теперь, сначала левой, затем правой кнопками мыши щелкните на «Select **and** Rotate», в окне **«Rotate Transform** Type-In» в форме «Absolute: **World»** в поле У введите 63,442. Закройте окно.

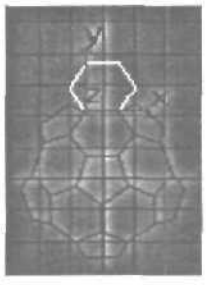
Теперь у нас есть еще один, точно подогнанный, кусочек мяча.

31. Сделайте последовательно 4 копии этого пятиугольника, поворачивая каждую на 72 градуса вокруг оси Z и устанавливая куда надо при помощи **Vertex Snap.** У вас должно получиться что-то вроде этого.



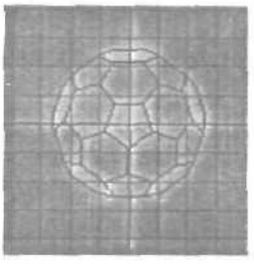
32. Выделите шестиугольник, расположенный вверху сцены,

33. Скопируйте его так, что бы центр вращения копии расположился в левой верхней вершине шестиугольника. Как показано на картинке.



34. Оставляя объект выделенным, в окне **«Rotate Transform Type-In»,** надеюсь, вы уже знаете, как его открыть, в поле X введите 79,178.

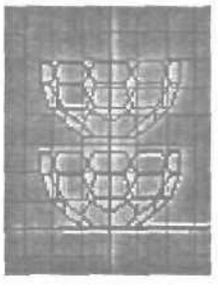
35. Теперь сделайте последовательно 4 копии этого шестиугольника, поворачивая каждую на 72 градуса вокруг оси Z и устанавливая куда надо при помощи **Vertex** Snap. Точно так же как вы сделали это недавно с пятиугольником- Результат будет примерно таким.



Как видите, вы уже создали нечто круглой формы и это мы будем использовать в качестве нижней части мяча.

36. А сейчас нажмите «f» на клавиатуре. Вы должны переключить окно проекции в вид «Front».

37. Выделите все n-угольники, а затем скопируйте их (так, как на картинке) и вы получите 2 половинки мяча.



38. Оставляя новые объекты выделенными, щелкните по кнопке «Select and Rotate» и поверните их на 180 градусов вокруг оси Z.

39. Теперь нажмите «t», что бы переключиться в вид «Top». Теперь, оставляя новые n-угольники выделенными, поверните их на 36 градусов вокруг оси Z.

40. Можете нажать «и», что бы переключиться в вид «User», и покрутиться вокруг мяча, нажав на кнопку «Arc Rotate».

41. Вы увидите, что мяч уже почти готов, есть только небольшое расстояние между половинками, избавьтесь от него, используя кнопку «Select and Move» и режим «Vertex snap».

Вот теперь у нас получился мяч, правда, чуть плоский по краям. Ликвидируем этот недостаток.

42. Нажмите снова «t» для переключения в вид «top».

43. Выделите все n-угольники, предварительно нажав клавишу «h» на клавиатуре.

44. Примените к ним модификатор **«Edit Mesh»**, затем заходите **в** «Sub-Object ⇨ Faces* и выделяйте все грани. Теперь в самом низу находите форму **«Normals» и** жмете на кнопку **«Flip».**

45. Оставляя все грани выделенными, примените к объекту модификатор «Meshsmooth». Поставьте галочку напротив **«Quad Output»,** а параметр «Iterations» задайте равным 3.

46. Теперь примените к объекту модификатор «Spherify». После этого ваш мяч должен стать круглым.

47. Примените к объекту модификатор «Face **Extrude».** Установите параметр **«Amount»** равным 3, а параметру **«Scale»** установите значение 95.

48. Примените модификатор «Meshsmooth» еще раз. Поставьте галочки напротив **«Quad Output», «Apply To whole Mesh» и «Smooth Result»,** а параметр «Iterations» пусть будет равен 2. Если нам не нужна такая высокая детализация, можно выбрать 1.

Вот теперь у вас получилась отличная модель мяча, осталось только назначить ей текстуры для пущего реализма. Для создания текстур можно использовать те скриншоты, которые мы сделали в начале урока.

Урок бесспорно полезный, но следует заметить следующее. Шаги с 1 по 41 этого урока описывают, как создать «мяч, правда чуть плоский по краям».

Этого же можно достичь гораздо быстрее, если создать стандартный объект Hedra с параметрами **Family:** Dodec/Icos и Family **Parameters:** P=0.35, Q=0.0, выполнить **Convert To:** Editable **Mesh и** дальше по тексту урока.

Глава 12.
Объединение вершин (welding)

Полигональное моделирование — это основа всего моделирования и каждый серьезный моделлер должен это понимать и знать наизусть тот набор средств, который предоставляет в этой области та программа, которой он пользуется. Объект 3d max — **Editable Poly** на первый взгляд похож на **Editable Mesh,** но тем не менее сильно отличается от него набо-

ром свойств и поведением.

Editable Poly довольно сильно отличается от Editable Mesh, и гораздо *лучше* разобраться в его особенностях путем проведения простых тестов, чем столкнуться с ними на этапе выполнения важной работы. Editable Poly гораздо более гибок и удобен в использовании, чем Editable Mesh.

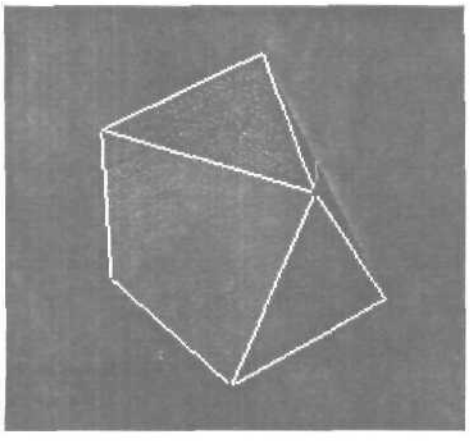
Давайте разберемся, чем же они отличаются.

Объединение вершин (Welding) — это базовая операция при полигональном моделировании. Эта операция стала традиционной и широко используется при работе с Editable Mesh. На уровне операций с вершинами (Vertex sub-object mode) есть две опции объединения вершин — Selected и Target.

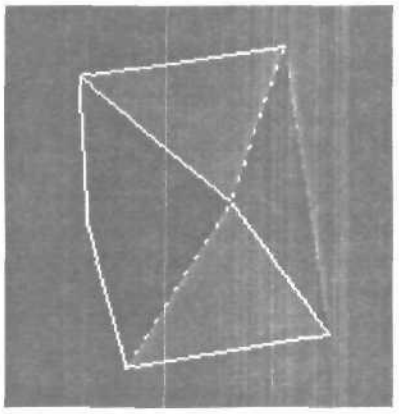
Для использования метода «Selected» вам необходимо сначала выбрать все вершины, которые БЫ хотите объединить, затем вы нажимаете кнопку «Selected» и все вершины, находящиеся на пороговом расстоянии (threshold distance) объединяются в одну.

Часто приходится настраивать пороговое расстояние для гарантии успешного результата.

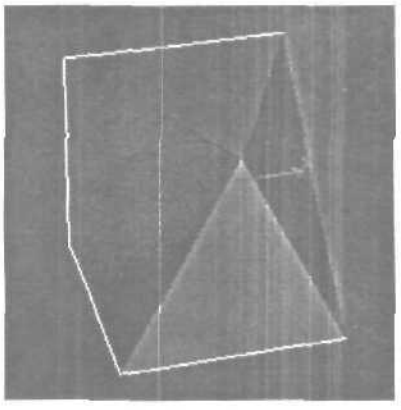
Если после установки необходимого порогового расстояния нажать кнопку «Selected», то грань между двумя вершинами исчезнет и вершины объединятся в одну на середине их прежнего расстояния.



Все выглядит достаточно просто, пока мы не переключимся в режим работы с гранями (Edge sub-object mode) — здесь мы видим появившиеся странные пунктирные линии.

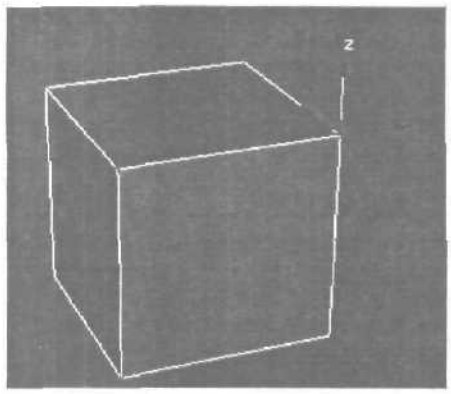


Как выяснилось процесс объединения вершин привел к созданию двойных граней. Вместо положенных четырех граней, встречающихся в вершине — их на самом деле восемь. Каждая появившаяся грань, это на самом деле две грани, находящиеся на минимальном расстоянии друг от друга. В этом вы можете легко убедиться выбрав грани и установив параметр **display** на панели свойств Editable **Mesh.** На самом деле, ситуация еще хуже, т.к. диагонали, связывающие вершины также удвоились.

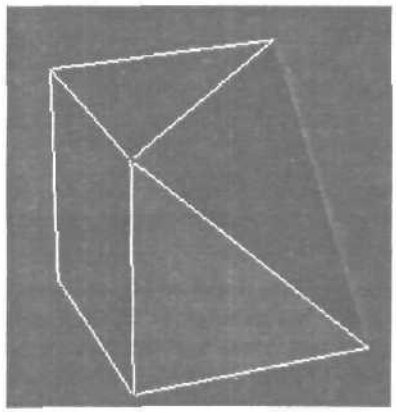


Что же это значит в практическом смысле? Существование таких двойных граней может в последствии проявиться при дальнейшей работе над объектом, когда результаты ваших действий будут не всегда адекватными. Но на это приходится идти, т.к. процедура объединения вершин практически не заменима.

Следующий способ объединения вершин — **Target** — перетаскивание одной вершине к другой. Вершина, которую вы перемещаете будет объединена с вершиной на которую вы ее переместите и займет ее положение. Техника заключается в следующем — вы нажимаете кнопку «Target», затем выбираете нужную вершину.

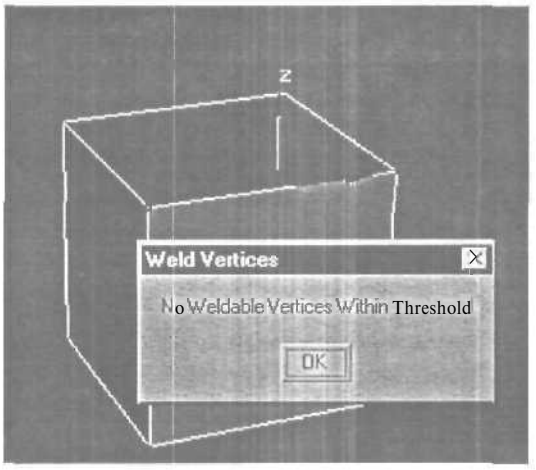


Появятся оси координат. Затем надо перетащить вершину к той, с которой ее необходимо слить, вершина при этом не двигается, только небольшая иконка появится при достижении вершины. Отпустите кнопку мыши и вершины объединятся. Не забудьте отключить режим **Target.**



Если вы перейдете в режим редактирования граней — вы найдете те же случаи дублирования граней. Фактически — если вы свернете (**collapse)** грань между двумя вершинами — вы получите тот же результат. И решения этой проблемы при работе с Editable **Mesh** нет.

Теперь перейдем **к** Editable Poly. В режиме редактирования вершин **(Vertex** sub-object **mode)** мы увидим те же опции Selected и Target, но если мы попробуем их использовать, то получим следующее сообщение:

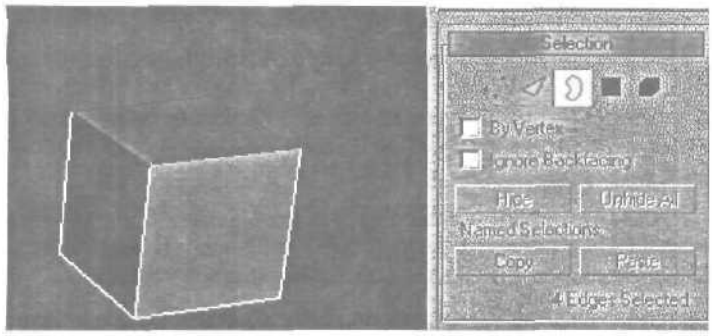


Первой мыслью будет, что пороговое расстояние слишком маленькое, и из-за этого вершины и не объединяются, но вы можете устанавливать любое расстояние и все равно получите то же сообщение. Дело в том, что не выбрано вершин, которые могут быть объединены.

Что же, выходит в Editable **Poly** нельзя объединять вершины? Да. но с одним исключением — чтобы объединить вершины в **Editable Poly** необходимо выбрать связывающие их грани и свернуть их (collapse). Главный плюс этого подхода это то, что при этом не создается дубликатов граней как в **Editable Mesh.**

Попробуем воспользоваться опцией **Target** — тоже без эффекта. Мы не получим даже сообщения об ошибке и не сможем даже выбрать вершины — в этом случае альтернативы нет, увы.

Резонный вопрос — зачем же тогда эти опции вынесены на панель **Editable Poly,** если они не действуют? Они действуют! **Их** функции работают, но в очень ограниченных, но при этом в очень важных случаях. Объект **Editable Poly** предоставляет новый уровень редактирования -- Border. Идея состоит в том, что на этом уровне можно производить операции над всеми гранями, образующими бордюр. Например, если выделить верхний полигон куба и удалить его, то в режиме **Border** можно будет выделить все грани, образующие бордюр.



Проще говоря — объединить вершины можно только в случае, если эти вершины окружают пустое пространство объекта. Основным достоинством **Editable** Mesh является то, что он позволяет объединить два отдельных объекта и получить один. Этот прием используется повсеместно когда необходимо соединитьдва объекта вместе (например зеркальные половинки объекта).

Глава 13. Взрывная волна

Одним из вариантов генерации взрывной волны может быть создание круглой плоскости, и наложение на нее текстур, при этом можно усилить эффект через **Video Post.** Этот метод оправдывает себя, тем не менее, он требует создания новых текстур для каждого нового эффекта.

В этой главе объясняется, как создать заготовку для генерации нескольких различных типов взрывных волн. Для этого нам не понадобится что-либо, поставляемое отдельно от стандартного пакета MAX.

Для успешного выполнения урока, вы должны ориентироваться в интерфейсе MAX, уметь создавать простые примитивы и масштабировать их по одной из осей, уметь создавать простую анимацию, уметь назначать эффекты материалам в редакторе материалов.

Эта глава научит вас работать с такими средствами MAX, как **Rendering** Effects, **Particle Arrays и Video Post** (VP). Если вы уже знакомы с ними, то урок не покажется вам трудным, если нет, не волнуйтесь. Все будет подробно описано.

Обзор Сцены

Наша сцена будет построена в три этапа. Первым является создание массива частиц **(Particle Array)** и так называемого дистрибутивного объекта. Вторым этапом будет настройка **Rendering Effects,** которые мы применим к массиву частиц. Финальным же этапом будет настойка эффектов в **Video Post**.

Дистрибутивный объект будет использоваться для контроля расширения частиц. Также эффекты, назначенные дистрибутивному объекту, мы будем использовать для усиления эффектов частиц.

Использование **VP** и **Rendering Effects** раздельно позволяет расширить набор эффектов, назначенных дистрибутивному объекту, Это также упрощает создание сцены в таком виде, в котором мы ее задумали. В нашем случае вы будете иметь дело только с VP, когда захотите повлиять на дистрибутивный объект. Для частиц же мы будем использовать **Rendering Effects.**

Введение в массивы частиц (Particle Arrays)

Система **Particle Array** (PArray) может использоваться для заполнения частицами некоторого объекта. Из сказанного выше ясно, что им будет дистрибутивный объект. С помощью системы **PArray** можно создавать очень крутые взрывные эффекты, особенно, когда задействованы деформаторы пространства **(Space Warps).**

Перед настройкой системы PArray мы сначала создадим дистрибутивный объект. Затем мы установим **PArray** эмиттер. Довольно просто, не так ли? Затем мы можем настроить миллион параметров и, сперва, это выглядит жутковато. Хотя, конечно, не миллион, но все равно довольно много. К счастью, для достижения желаемого результата нам придется заострить внимание всего на нескольких из них.

Создание дистрибутивного объекта

Моделирование дистрибутивного объекта для нашей сцены не потребует никаких специальных усилий. Это будет обычный плоский тор. Запустите MAX или выполните команду **Reset** для создания новой сцены. В окне **Perspective** создайте тор. **Не имеет** значения, как он будет выглядеть. После этого зайдите в панель **Modify.** Можно, к примеру, использовать такие параметры:

```
Radius 1 = 95
Radius 2 = 9
Rotation = O
Twist = O
```

```
Segments = 48
Sides = 20
```

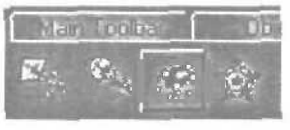
Вы можете изменить их в соответствии со своими вкусами, зависящими от того какую форму вы желаете получить. Установите флажок напротив параметра Generate **Mapping Coords.** Щелкните правой кнопкой по объекту и выберите Properties. Задайте параметр G-Buffer Object Channel равным 1.

Следующим шагом будет сплющивание тора. Выберите окно с видом Left. Теперь надо сплющить тор по оси Y при помощи кнопки Non **Uniform** Scale. Уменьшите его толщину примерно раза в 2. Наша сиена будет содержать всего два объекта, поэтому необязательно изменить имя тора. Задайте ему светло-коричневый цвет.

Когда закончите, сохраните ваш результат под именем shock1.max.

Создание PArray и настройка анимации

На главной панели выберите **Particles.** Это третья по счету кнопка слева, как показано на рисунке. Щелкните по кнопке и в окне перспективы создайте **PArray** эмиттер.



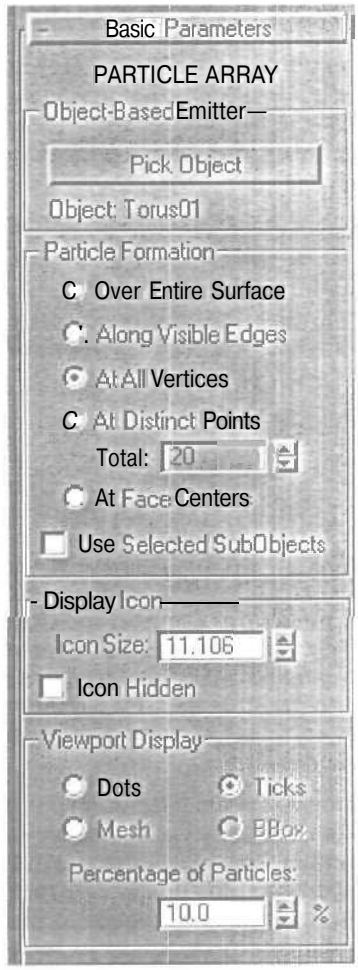
Пока что сцена довольно проста, поэтому давайте настроим анимацию и сконцентрируемся на эффектах, дабы разнообразить это руководство. Установите бегунок в нулевой кадр и включите режим анимации. Теперь выберите режим **Uniform Scale и** уменьшите тор до относительно малых размеров. Это будет начальное состояние нашей взрывной волны. Установите **PArray** приблизительно в центре тора, но при этом не обязательно быть предельно точным, короче, на глазок. Когда результат вас удовлетворит, выключите режим анимации.

Установив бегунок в кадр под номером 100, включите режим анимации. Выберите режим **Uniform Scale.** А теперь увеличивайте масштаб тора, пока его размеры не превысят пределы окна перспективы. Возможно, для достижения правильного эффекта, вам придется отрендерить пробную анимацию и, опираясь на полученный результат, отрегулировать финальный размер тора. В противном случае может оказаться, что взрывная волна будет распространяться недостаточно быстро.

Выключите режим анимации и подвигайте бегунок туда-сюда. Вы увидите, как тор быстро меняется в размерах. Сохраните вашу работу.

Настройка дистрибутивного объекта

Установив бегунок в нулевой кадр и выбрав **PArray** в окне **Perspective,** щелкните по закладке Modify. Потяните панель параметров, так, чтобы свиток **Basic Parameters** был полностью виден. Соответствующая часть этой панели изображена на рисунке.



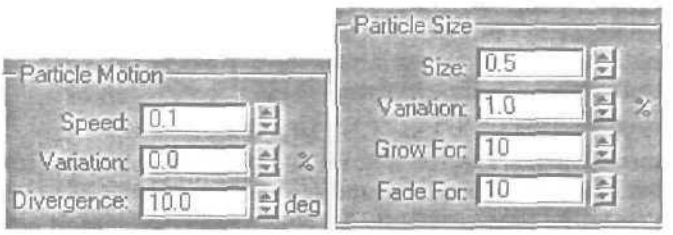
Теперь щелкните по кнопке **Pick Object** и выберите тор. Это сделает его дистрибутивным объектом для массива **PArray.** Раздел Particle Formation очень крут тем, что каждый его параметр обеспечивает нас но-

вым представлением эффекта. В этом уроке использовался параметр **At All Vertices.** В результате получится увеличивающееся кольцо, состоящее из маленьких подколец.

В разделе **Viewport Display** установите параметр **Percentage of Particles** равным 10.0, а для отображения частиц выберите **Ticks.** Это означает, что количество частиц, отображаемых в вашем вьюпорте, будет равно 10% от общего количества и выглядеть они будут как метки,

Генерация частиц

Сверните свиток **Basic Parameters** и разверните свиток **Particle** Generation. Обратите внимание на два важных параметра: Speed в разделе **Particle Motion** установите равным 0.1, а Size в разделе **Particle** Size установите равным 0.5.



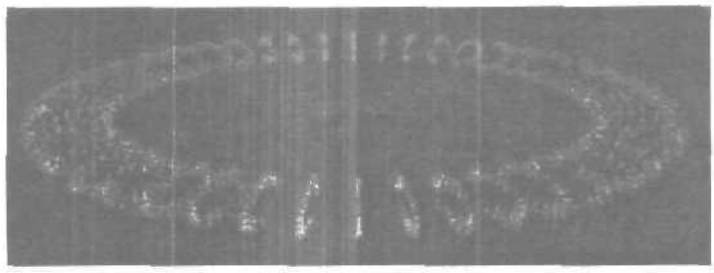
Параметр **Use Rate** контролирует количество генерируемых частиц на кадр. Его изменение влияет на плотность взрывной волны.

Значения параметров в разделе **Particle liming** вы должны установить по своему вкусу. **Emit Start и Emit** Stop указывают на то, в каких кадрах появляются и исчезают частицы. От этого зависит, насколько быстро будет затухать взрывная волна. Если вы считаете, что затухание идет очень быстро, попытайтесь в качестве конечного кадра задать 50 или 60. Кроме того, можно увеличить параметр **Life.**

Наконец, разверните свиток **Particle Type и** выберите **Standard Particles** в разделе **Particle Types,** а **в** разделе Standard **Particles** установите **Triangle.**

Материалы частиц и Rendering Effects

Установите бегунок в 45-й кадр. Произведите тестовый рендеринг окна перспективы. Должно выглядеть неплохо. Обратите внимание, как частицы расширяются вместе с тором. Они размещены в кольцах, огибающих тор. Мы можем изменить отображение эффекта, просто поменяв параметры, которые указывают на то, как частицы будут размещаться относительно дистрибутивного объекта.



Откройте **Material Editor** и выберите пустой слот, Для достижения эффекта взрывной волны нам нужно сделать тор прозрачным. Задайте материалу нулевую прозрачность **(Opacity) и** примените материал к тору. Выберите другой пустой слот. Назовите его ParticleMaterial. Для первого раза мы сделаем бледно-синий эффект, поэтому установите для цвета Diffuse значения **130:130:255, а** для **Ambient** — 16:200:255. Вы можете изменить их позже для создания других эффектов. Установите параметр Self-Illumination равным 25, а **Material** Effects **Channel** равным 1 (7-я слева кнопка под материалами).

Свиток **Particle Type** должен быть развернут. Внизу должна быть кнопка **Get Material From.** Под этой кнопкой, выберите **Picked Emitter.** Это означает, что при назначении материала эмиттеру, он будет так же назначаться частицам. Выберите PArray эмиттер. Перейдите в редактор материалов и назначьте наш ParticleMaterial эмиттеру.

Произведите другой тестовый рендеринг. Уже ближе, но не совсем. В главном меню зайдите в **Rendering ➪ Effects.** Щелкните по кнопке **Add.** Откроется окно **Add Effects.** Выберите **Lens Effects,** а затем нажмите OK. В окне **Rendering Effects** разверните свиток **Lens** Effects **Parameters.** Выберите Glow, а затем щелкните по кнопке со значком >.

Перейдите к свитку **Glow Element.** Выберите панель **Options.** В разделе Image **Sources** включите параметр Effects ID и задайте ему значение 1. Это наложит свечение на любой материал, ID которого равен 1, как, например у **ParticleMaterial.** Перейдите в панель **Parameters.** Задайте для параметра Size значение 0.1. Мы хотим, чтобы свечение вокруг частиц было очень маленьким. Параметр **Intensity** может быть равен **40** или 50. Для **Use Source Color** оставьте 0.0. Применим эффект под названием **Radial Color.** Поставьте флажки напротив его двух составных цветов. Затем для цвета, который слева, установите значения **170:170:255,** а того, что справа — 255:255:255. Это создаст светло-голубое свечение вокруг частиц. Если вам нужен другой оттенок, то измените значения левого цветового образца.

Теперь отрендерите и сохраните ваш результат.

Настройка параметров Video Post

Мы использовали Rendering Effects для контроля за свечением частиц. Так же можно дополнить или усилить контраст картинки, применив эффекты к тору. Для этого нужно будет настроить Video Post (VP). Video Post несет в себе некоторые преимущества. Так мы без особых усилий можем отдельно контролировать эффекты частиц и тора. Эффект Glow в VP имеет опцию Electric, задействовав которую можно создать очень реальное свечение, имея нескольких материалов. Если вы хотите добавить в сцену эффект вспышки, то это легко сделать в окне Video Post.

В главном меню зайдите в Rendering ➪ Video Post. В появившемся окне Video Post наведите курсор на каждую из активных кнопок вверху и прочтите всплывающий текст, для получения представления об их функциях. На данный момент наш набор эффектов в VP равен нулю. Поэтому нажмите на кнопку Add Scene Event. В появившемся окне Add Scene Event выберите Perspective и нажмите OK.

В окне VP нажмите кнопку Add Image Filter Event. В появившемся окне Add Image Filter Event выберите Lens Effect Glow и нажмите OK. После этих манипуляций сцена будет рендериться из окна перспективы. Эффект свечения будет применяться уже после рендеринга, а информацию для него будет предоставлять рендер. Отсюда вывод — VP эффекты являются двумерными.

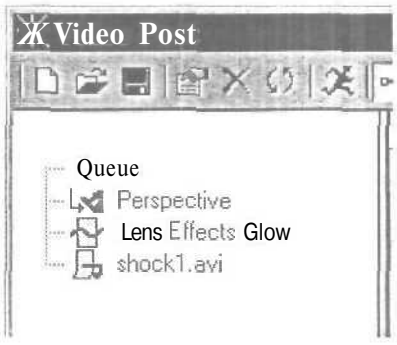
После применения эффекта свечения мы должны отрендерить анимацию в видео-файл. В окне VP нажмите кнопку Add Image Output Event. В появившемся окне Add Image Output Event щелкните по кнопке Files. Выберите нужный каталог, в качестве имени задайте shock1.avi и нажмите OK. Появится окно Video Compression. Выберите Full Frames (Uncompressed) — Полные кадры (без сжатия). Нажмите OK.

Обшей ошибкой при настройке VP является добавление сразу нескольких событий. Результат будет не таким, какой вам нужен. Поэтому старайтесь настраивать каждое событие, а затем добавлять следующее. Если что-то будет выглядеть не так, не волнуйтесь. Выберите каждое событие по отдельности и удалите. Удалив все, попытайтесь еще раз.

После того, как вы все выполнили, сохраните ваш результат под новым именем, например shock2.max. Иногда полезно сохранять крупные изменения в новый файл.

Настройка эффекта Glow

В окне VP ваш набор событий должен выглядеть аналогично приведенному рисунку. Щелкните дважды на Lens Effect Glow. В появившемся окне нажмите кнопку Setup. Затем включите кнопки Preview и VP Queue. Сцена будет отрендерена с сохранением информации обо всех объектах.

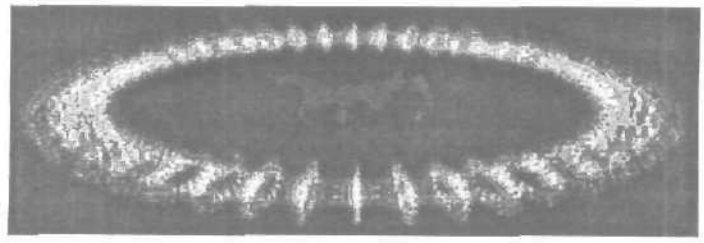


Теперь попробуем назначить небольшое свечение тору. Панель Properties должна быть активной. Убедитесь, что параметр Object ID в разделе Source включен и равен 1. В разделе Filter включите AH.

Окно предварительного просмотра будет обновляться каждый раз после изменения каких-либо параметров. Свечение получилось довольно большим. Перейдите в Preferences. В разделе Effect задайте значение для Size 0.5. В разделе Color укажите Gradient. Мы будем использовать стандартные градиентные установки. Теперь вместо пустого пространства между кольцами мы неплохое свечение. Изменяя VP настройки, вы можете сделать свечение более сильным или прозрачным.

Если вы отрендерите изображение в разрешении 640x480, то VP эффекты могут показаться более качественными, нежели в окне предварительного просмотра. Вообще полезно рендерить несколько тестовых кадров после того, как вы все настроили.

И раз уж речь пошла о рендеринге, то в окне VP щелкните по кнопке, на которой изображен бегущий человек. Она называется Execute Sequence. В появившемся окне в разделе Time Output выберите Single и укажите кадр, например 48. Задайте размер кадра и нажмите кнопку Render.



Сделайте тестовый рендеринг 0, 20, 60, 80 и 100 кадров, чтобы убедиться, что VP настройки вас устраивают, а кольцо тает в нужном кадре. Когда вы будете удовлетворены, сохраните вашу работу под именем shock3.max. Теперь можно рендерить сцену в .AVI файл. Для этого надо выбрать Range в разделе Time Output. Из всех кадров будет построена анимация. Вы можете указать начальный и конечный кадры для тестирования отдельного участка анимации. Это полезно для сложных сцен с долгим рендерингом.

Проиграйте видео и настройте движение кольца и VP параметры в соответствии со своими вкусами.

Вариации

Настоящим бонусом в этой сцене является то, что вы можете создавать многочисленные другие эффекты взрывов простой сменой нескольких параметров. Выделите эмиттер и зайдите в закладку Modify. В свитке Basic Parameters в разделе Particle Formation установите:

1. Over entire surface
2. Along visible edges
3. At face centers

Каждый параметр образует новый эффект. Попробуйте изменить количество частиц на кадр. Так же можно поменять цвета материала, эффектов и градиента в VP.

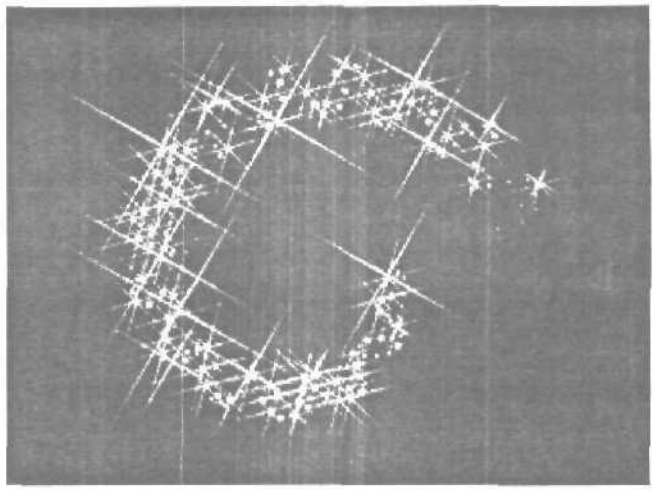
Другой возможностью является наложение текстуры на тор. Для этого его нужно сделать почти прозрачным. Зайдите в настройки Lens Effect Glow. В панели Inferno выберите Electric. Также можно задать эффект свечения лишь краям тора, а не всему объекту. Для этого в панели Properties в разделе Filter установите Edge и выключите All.

Варьируя все эти параметры, вы получите новые эффекты. И нет нужды создавать текстурные карты. Просто меняйте некоторые параметры и выбирайте, то, что вам нужно. Еще можно добавить Lens Effects Flare для усиления эффекта взрыва.

Глава 14.
Использование фильтра Highlight

Фильтр Highlight является одним из модулей набора плагинов Lens Effects для Video Post и входит в стандартный пакет 3D Studio Max. Эффект Highlight заключается в том, что яркие участки визуализированной геометрии после применения фильтра отображаются в форме звезд.

Данный фильтр полезно применять к объектам, которые обладают сверкающими материалами. Однако в этой главе речь пойдет о создании так называемой «волшебной пыли» и се анимации.



Осваивая материал данной главы, вы познакомитесь:

- **С эффектом Highlight.**
- С системой частиц SuperSpray и ее параметрами.
- С объемной деформацией Path Follow и ее параметрами.
- С основами модуля Video Post.

Для того чтобы наша будущая анимация длилась хотя бы с десяток секунд, давайте увеличим активный временной сегмент до 300 кадров.

Щелкните на кнопке Time Configuration. В появившемся диалоге Time Configuration в группе Animation увеличьте значение спиннера Length до 300.

Щелкните на кнопке «OK».

Для того чтобы в дальнейшем нам было удобнее следить за сценой, перейдем в 150-й кадр. Просто перетащите «тайм-слайдер» (Time Slider) в кадр 150.

На командной панели Create, в категории Geometry, из разворачивающегося списка выберите разновидность Particle Systems (Системы частиц). В свитке Object Type щелкните на кнопке Super Spray и создайте данную систему частиц в окне Top.

Вообще-то, для иллюстрации эффекта Highlight можно было бы использовать простой Spray. Здесь выбран Super Spray единственно потому, что данный тип предоставляет больше возможностей для контроля над частицами.

Для настройки параметров только что созданной системы частиц перейдем на панель Modify. Если вы нечаянно сняли выделение с Super Spray, то вновь выделите его.

В свитке Basic Parameters в группе Viewport Display установите значение параметра Percentage of Particles в 100%. Этот параметр отвечает за то, какой процент реально созданных частиц будет отображаться в окнах проекций. Так как много частиц мы создавать не планируем, поэтому для наглядности пусть все они будут нам видны.

Разверните свиток Particle Generation.

В группе Particle Quantity установите опцию Use Rate, а соответствующий спиннер в 2. Тем самым мы установили, что в каждом кадре будет испускаться по две частицы.

В группе Particle Motion установите значение параметра Speed в 1. Данный параметр отвечает за скорость движения частиц — количество текущих единиц измерения за кадр.

В группе **Particle Timing:**

- -Значение параметра Emit Start оставьте таким, какое оно есть — 0. Данный параметр отвечает за то. в каком кадре частицы появятся в сцене.
- Значение параметра Emit Stop установите в 150. Данный параметр отвечает за то, в каком кадре перестанут появляться новые частицы.
- Значение параметра Display Until установите в 300. Данный параметр определяет в каком кадре все частицы перестанут отображаться, в независимости от прочих параметров.

- Значение параметра Life установите в 150. Данный параметр определяет количество кадров в течение которых каждая отдельная частица будет отображаться на сиене.

В группе **Particle Size:**

- Значение параметра Size оставьте таким, какое оно есть — 1. Данный параметр определяет размер частиц.
- Значение параметра Grow For установите в 1. Данный параметр определяет число кадров, в течение которых частица с момента своего появления в сцене вырастает до размера, определенного в спиннере Size.
- Значение параметра Fade For установите в 15. Данный параметр определяет число кадров, в течение которых частица уменьшится до размера 1/10 значения, установленного в параметре Size прежде чем она прекратит свое существование на сцене.

Теперь разверните свиток Particle Type:

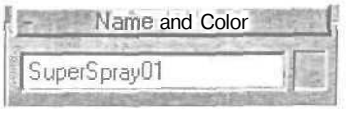
- В группе Particle Types должна быть установлена опция Standard Particles. Опции данной группы определяют три типа частиц. В зависимости от того, какой тип частиц вы выбираете, соответствующие группы параметров становятся доступными внизу данного свитка.
- Итак, в настоящий момент активна группа параметров Standard Particles. Здесь вам предлагается выбрать один из нескольких типов стандартных частиц. Выберите Facing. Это визуализирует каждую частицу нашей системы как квадрат, причем который в любом окне проекций всегда направлен лицевой стороной к зрителю.

Для того чтобы в пост-обработке мы смогли применить фильтр именно к нашему Super Spray, необходимо назначить ему идентификатор.

- Наведя курсор на значок системы частиц SuperSpray01, щелкните правой кнопкой мыши. Значок системы частиц при этом должен быть выделен.
- Из контекстного меню выберите Properties.
- В появившемся диалоге в группе G-Buffer установите в спиннере Object Channel значение 1.
- Щелкните на кнопке «OK».

Конечно, правильнее было бы назначить нашим частицам материал, но для целей данной статьи вполне достаточно просто изменить цвет объекта **Super SprayOI.**

- Щелкните на цветовом квадратике, что справа от имени объекта. Это можно сделать либо на панели Create, либо на панели Modify.



- В появившемся диалоге выберите светло голубой цвет: R: 175, G: 215, B: 255.
- Щелкните на кнопке «OK», чтобы закрыть диалог **«Object Color».**

Чтобы придать направление потоку наших частиц, необходимо создать путь.

- Вернитесь на командную панель **Create.**
- Выберите категорию **Shapes. В** разворачивающемся списке должна быть выбрана разновидность **Splines.**
- В свитке **Object Type** щелкните на кнопке **Helix (Спираль).**
- В окне Top создайте сплайн данного типа со следующими параметрами:
  Radius 1: 55
  Radius 2: 155
  Height: -150
  Turns: 1
  Bias: 0
  CW: on
- Поместите созданную спираль так, чтобы ее начало примерно (но не абсолютно!) совпадало со значком системы частиц.

Для того чтобы заставить частицы Super Spray двигаться вдоль только что созданного пути, необходимо использовать объемную деформацию **Path Follow.**

- На командной панели **Create** выберите категорию Space **Warps.**

- Из разворачивающегося списка выберите разновидность Particles Only.
- В свитке Object Type щелкните на кнопке Path Follow.
- В окне Top создайте объемную деформацию выбранного типа.

Выберите на Главной панели инструментов инструмент Bind to Space Warp, и свяжите объемную деформацию PathFollowObject01 с системой частиц SuperSpray01.

Перейдите на панель Modify и установите следующие параметры свитка Basic Parameters:

- В группе Current Path щелкните сначала на кнопке с надписью «Pick Shape *Object*», а *затем* в окне Top на сплайне HelixOl. Тем самым вы указываете путь для объемной деформации.

В группе **Motion** Timing:

- Значение параметра Start Frame оставьте таким, какое оно есть — 0. Данный параметр устанавливает, в каком кадре объемная деформация начинает оказывать влияние на нашу систему частиц.
- Значение параметра Travel Time установите в 150. Данный параметр определяет, сколько времени (т.е. количество кадров) уйдет у отдельной частицы на то, чтобы пройти весь указанный путь.
- Значение параметра Last Frame установите в 300. Данный параметр устанавливает, в каком кадре объемная деформация Path Follow завершает свое влияние на систему частиц.

В группе Particle Motion:

- Установите опцию Along Offset Splines. При данной опции влияние объемной деформации Path Follow на систему частиц зависит от расстояния между первой вершиной пути — в нашем случае HelixOl, и источником частиц — SuperSprayOl. Помните, как на седьмом шаге последним пунктом создания спирали мы совмещали ее начало со значком Super Spray?
- Установите флажок Constant Speed, чтобы все частицы вдоль пути двигались с одинаковой скоростью.

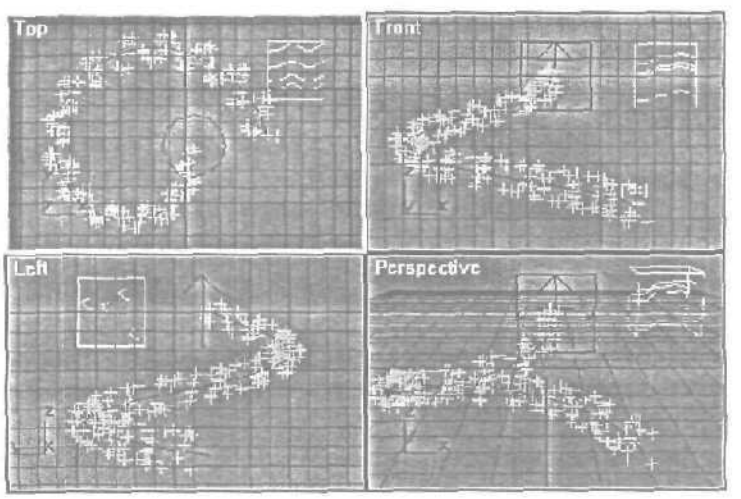
В спиннере Stream **Taper** установите максимальное значение - 99%. Данный параметр определяет степень влияния, которое будет оказывать на частицы одна из следующих опций:

- Converge: При движении вдоль пути частицы сходятся по направлению к пути.
- Diverge: При движении вдоль пути частицы расходятся. Следует установить именно эту опцию, чтобы создать эффект будто бы частицы испускаются из одной точки и при движении постепенно расходятся в стороны.
- Both: При движении вдоль пути одна часть частиц сходится, другая — расходится.

В спиннере **Stream Swirl** установите значение 15. Данная опция устанавливает количество оборотов, которые совершат частицы при своем движении вдоль пути. Следующие опции устанавливают направление для совершаемых оборотов:

- **Clockwise:** По часовой стрелке.
- **Counterclockwise:** Против часовой стрелки.
- **Bidirectional:** Одна часть частиц движется по часовой стрелке, другая — против часовой. Следует установить именно эту опцию.

Сцена готова!



Вся черная работа выполнена. Осталось самое интересное — непосредственно познакомится с фильтром Lens Effects **Highlight.**

Откройте окно Video Post: Меню **О Rendering ➪ Video Post.**

Щелкните на кнопке Add Scene Event.

- В появившемся окне выберите из разворачивающегося списка *строку* Top, вил из окна проекции которой мы хотим визуализировать.
- Щелкните на кнопке «ОК».

Щелкните на кнопке Add Image Filter Event.

- В появившемся окне выберите из разворачивающегося списка фильтр Lens **Effects** Highlight.
- Щелкните на кнопке Setup.

В появившемся окне на вкладке **Properties** проверьте, чтобы установлены были лишь следующие параметры:

- В группе Source только опция **Object ID: I**
- В группе **Filter** только опция **Bright,** а значение в соседнем спиннере 160.

На вкладке **Geometry** измените только параметр Clamp — 7.

На вкладке **Preferences** измените следующие параметры:

- В группе **Effect** установите параметр **Size** в **15,** а параметр **Points в 6.**
- В группе **Color** установите опцию **Gradient.**

На вкладке **Gradients:**

- Щелкните на первом градиенте, который называется **Radial Color,** неподалеку от его начала.

- Дважды щелкните на появившемся флажке. В появившемся диалоге выберите цвет для флажка, например, R: 250, G: 60, B: 120.
- Щелкните на кнопке **Close** в диалоге «Color Selector: Color».

Щелкните на кнопке «OK» в диалоге **Lens** EffectsHighlight. Она может быть не видна из-за панели задач Windows. Перетащите окно диалога чуть выше.

Для большего эффекта создайте второй фильтр Highlight. Дли этого:

- Щелкните на кнопке **Add Image**Filter **Event.**
- В появившемся окне выберите из разворачивающегося списка фильтр Lens Effects Highlight.
- Щелкните на кнопке Setup.

В появившемся окне на вкладке **Properties** проверьте, чтобы установлены были лишь следующие параметры:

- В группе Source только опция **Object ID: 1**
- В группе **Filter** только опция Bright, а значение в соседнем спиннере 255.

На вкладке **Geometry** измените параметр Clamp — 50.

На вкладке **Preferences** измените следующие параметры:

- В группе Effect установите параметр **Size** в 40, а параметр **Points в 4.**
- В группе Color установите опцию **Gradient.**
- На вкладке **Gradients** оставьте все как есть.

Щелкните на кнопке «OK» в диалоге Lens **Effects** Highlight.

Щелкните на кнопке **Add Image** Output **Event.**

В появившемся окне щелкните на кнопке с надписью «Files».

В новом окне выберите тип файла итоговой визуализации, например, AVI-файл.

Дайте новому файлу имя и выберите папку, где желаете его сохранить.

Щелкните на кнопке «Сохранить».

В окне **«Сжатие видеозаписей»** щелкните на кнопке «OK», принимая предложенный кодек.

В окне **«Add** Image **Output** Event» также щелкните на кнопке «OK».

В окне «Video **Post»** щелкните на кнопке **Execute Sequence.**

Проверьте, чтобы в диалоге **"Execute Video** Post» в группе **Time Output** была выбрана опция Range (Диапазон) и именно с нулевого по трехсотый кадр.

В группе *Output Size* размер 320x240.

И щелкните на кнопке **«Render».**

Теперь осталось дождаться окончания визуализации и с помощью Меню ➪ File **O View File** оценить конечный результат. Процесс визуализации всех 300 кадров даже на медленных машинах займет не более 10 минут.

Дополнение

Те, кто знаком с окном Track View, могут внести дополнительное оживление в наши «блестки» перед тем, как визуализировать сцену.

Откройте окно **Track View.**

Доберитесь до трека *Video* **Post** ➪ Второй *фильтр Lens* **Effects** Highlight ➪ **Angle и** выделите его.

Добавьте по одному ключу в кадре 0 и в кадре 300 данного трека,

В ключе из 300-го кадра установите параметр Value в **-1050** (обратите внимание на знак «-»).

Закройте окно **Track** View.

Данные действия приведут к тому, что «бОльшие звездочки» нашей волшебной пыли во время движения дополнительно совершат три оборота против часовой стрелки. Вы можете точно также придать вращение и «меньшим звездочкам», внеся изменения в соответствующий трек первого фильтра Lens Effect Highlight.

В случае проблем

Если вы визуализируете сцену с разрешением отличным от 320x240, возможно, вам потребуется соответственно настроить параметр, отвечающий за размер частиц — параметр **Size** из свитка **Particle Generation** объекта **SuperSprayOl.**

Если у вас частицы при движении вдоль пути выстраиваются «в нитку», скорее всего вы слишком точно совместили первую вершину сплайна-пути со значком системы частиц. Помните, для PathFollowObjectOl мы установили опцию Along **Offset** Splines. Теперь вы можете, перемещая значок SuperSprayOl настраивать движение частиц так, как вам будет угодно.

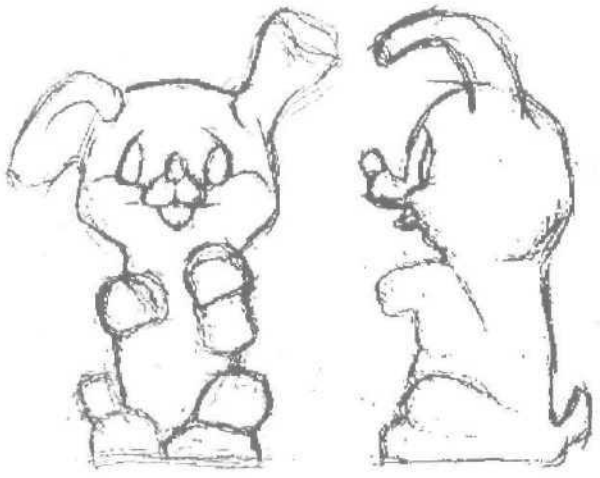
Глава 15.
Создание модели игрушечной собачонки

Все начинается с выбора персонажа для будущей модели, а так как далее речь пойдет о патчевом моделировании, то и персонажем может быть все, что угодно — от простых геометрических фигур до изображения людей, животных и птиц.

Если вы не можете найти материал для будущей работы дома, то можно обратиться в библиотеку, но в любом случае вы должны четко представлять что должно получиться в конце работы. Если ваше воображение не может нарисовать полную картину до начала работы, то это тоже не беда — можно все корректировать во время работы.

Ну что ж, для того, что бы облегчить себе жизнь решено было взять реально существующую собачонку и решил сделать модель. Для этого было сделано два наброска в фас и профиль, на основе которых и будут вестись все дальнейшие построения.

Первым делом надо было нарисовать фронтальный и боковой вид.



Рисунки должны иметь одинаковые пропорции, иначе возникнут проблемы с подгонкой, что создает путаницу во время работы.

Далее есть два пути.

Первый — это поставить рисунки в качестве background во **viewport** (Views ➪ Viewport **Bakground**) — фронтальный вид во фронтальный вьюпорт и боковой в **Right** соответственно. Кроме того, не забудьте включить «галочку» напротив **Match** Bitmap и **Lock Zoom/Pan, ну** и конечно **Display Background.** Это не позволит искажать изображение и привяжет ваш рисунок к создаваемому объекту. Удобно пользоваться сочетанием клавиш **Ctrl+Alt+B** для корректировки объекта по отношению к фону.

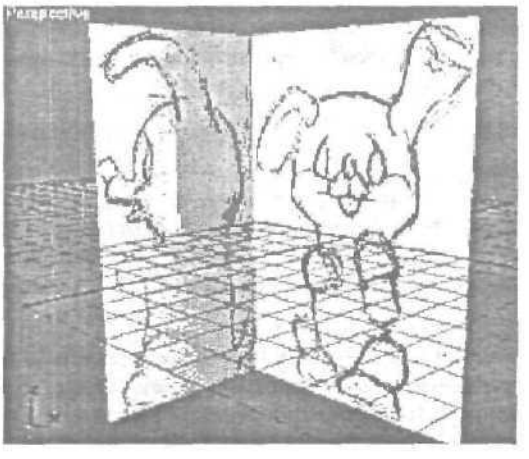
И **второй**, которым мы в дальнейшем и воспользуемся, это создание т.н. виртуальной студии. Для этого надо на фронтальном виде создать объект **Plane**, с пропорциями, соответствующим пропорциям фронтального рисунка. К примеру, 115.3 x 83.4.

Далее нужно сделать копию этой плоскости, перетащив в сторону плоскость с нажатой клавишей Shift и указав в качестве метода копирования Copy. Во второй плоскости, которая будет являться подложкой для бокового вида, надо изменить ширину с 83.4 на 58.2 (для соответствия рисунку бокового вида). Далее эту плоскость на виде сверху поворачиваем на 90 градусов и с помощью кнопки **Align,** которая находится на панели инструментов, выравниваем две плоскости так, что бы они вместе создавали букву *«Г»*.

Теперь следующим шагом будет наложение на плоскости рисунков. Для этого в редакторе материалов создадим два материала на один из которых поместим в качестве Diffuse Color рисунок фронтального вида (для этого щелкаем мышкой на кнопке рядом с цветом Diffuse и выбираем карту Bitmap, щелкнув на которой, в свою очередь, выбираем наш отсканированный рисунок фронтального вида).

Далее присваиваем этот материал объекту Plane01, что соответствует фронтальному виду. Следует заметить, что в редакторе материалов надо активизировать кнопку Show map in viewport, а в настройках вида (Front и Right) выставить Smooth+Highlights, иначе вы не увидите свои рисунки во вьюпортах. Те же операции надо произвести и для второго материала, накладываемого на Plane02.

В итоге должно получиться нечто похожее на изображение внизу.



Справедливости ради надо отметить, что мы не воспользовались видом сверху, не потому, что он не нужен вообще, а потому, что в нашем случае вид сверху не столь уж. критичен для построения модели, к тому же сам оригинал у нас перед глазами и мы имеем возможность рассмотреть собачонку во всех подробностях.

Вот мы и подошли к непосредственному созданию самой модели. Начинать будем с построения сплайна по характерным точкам на рисунке. Для этого сделайте активным вид сбоку и выбрав во вкладке Create ➪ Shapes ➪ Line начните построение сплайна.

Сразу же хотелось бы дать несколько советов.

Начнем с того, что строить модель можно по-разному, но если модель имеет симметричные детали (или вся модель симметрична), то логичнее всего делать одну половинку, а затем ее скопировать и присоединить к существующей. Далее, если вы не используете для построения модели предварительные рисунки или фотографии, то построение надо вести от общего к частному, т.е. вначале создаете сплайнами контуры всей модели, *а* затем приступаете к ее детализации.

Сплайн удобнее строить, используя Bezier Corner, причем длина ручек Безье должна соответствовать приблизительно 1/3 расстояния между вершинами, тогда модель получается более сглаженной. Хотя иногда полезно пренебрегать этим правилом для создания криволинейной поверхности, используя минимальное количество точек.

Кстати о точках. Чем меньше точек будет задействовано для создания модели — тем лучше. Хотя и здесь все зависит от конечной цели. Если вы, скажем, собираетесь анимировать свою модель, то она должна на выходе иметь достаточное количество полигонов что бы не казаться неестественной при анимации.

Итак, продолжим.

Голова нашей собачонки симметрична — с нее и начнем. Ниже показана последовательность построения точек. Старайтесь точки ставить только в тех местах, где сплайн изменяет направление движения. Опять же хочется заметить, что количество точек в процессе работы может изменяться. Их можно добавлять по мере необходимости или наоборот, удалять лишние.

Построенный сплайн будет являться серединной линией, используя которую, будем строить остальные сплайны. На кончике носа построим два других сплайна, *принадлежащих* объекту **Line01** (*контур* головы) и соединяющих противоположные точки. Каждая линия имеет дополнительную точку посередине. Эти точки понадобятся для проведения через них линии от кончика носа. Полезно при проведении таких операций использовать 3D Snap Toggle — это избавит вас от необходимости совмещать две точки вручную.

Очень важно контролировать весь процесс создания новых точек в *других* окнах проекций. После *построения* новых *сплайнов* их вершины нужно поставить на место. Крайние точки уже стоят там, где им и положено быть, а вот средние надо на фронтальном виде сдвинуть вправо так, что бы они соответствовали контурам рисунка. Вид сверху еще раз позволяет проконтролировать их местоположение. Таким образом добавляя к основному сплайну дополнительные мы будем опоясывать сплайнами всю модель.

На начальном этапе обучения довольно сложно определить, где должны проходить сплайны и, соответственно, в каких местах располагать контрольные точки, поэтому как один из вариантов можно предложить способ предварительного нанесения карандашом линий на рисунок модели, а еще лучше если есть возможность (как в нашем случае с игрушкой) нанести эти линии на сам объект, с которого создаем модель. Хочется обратить ваше внимание на то, что при построении сетки из сплайнов, как можно меньше должно быть ячеек с тремя вершинами т.к. они создают проблемы с наложением текстур и анимацией.

Сетка из сплайнов является каркасом для создания поверхности. Раньше на ее основе мы бы создавали модель из патчей, теряя кучу времени на выравнивание патча по сплайнам, но спасибо Питеру Ватье, который создал модуль Surface. Этот модуль поможет вам без лишних затрат времени превратить сплайны в готовую патч-модель. Им и воспользуемся, тем более, что он присутствует в стандартном наборе модификаторов.

Хорошей практикой является тот способ построения модели, при котором во время построения каркаса из сплайнов вы можете видеть модель покрытую патчами. Для этого надо воспользоваться одним «хитрым» способом: вам надо создать копию модели и применить к ней модификатор **Surface.**

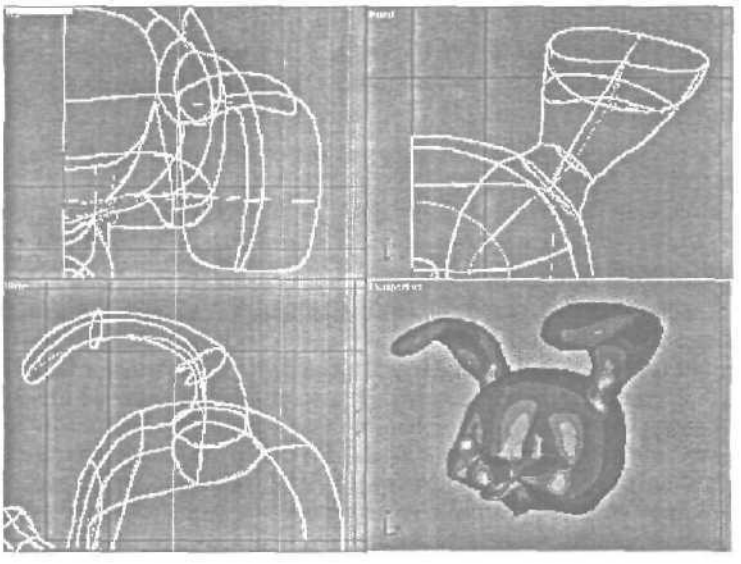
Делается это так. Удерживая **Shift** перетаскиваем нашу модель (пока еще только сплайны) на свободное место позади нашей виртуальной студии и затем в качестве метода копирования указываем **Reference.** Далее, навешиваем на копию модификатор **Surface** и делаем **Mirror,** выбрав **Instance.** После этого две половинки модели надо совместить (но не «сваривать»), чтобы получился цельный объект. Этот клонированный объект и поместим в окно перспективы. Иногда можно для этого объекта поставить в сиене несколько камер, для того, чтобы можно было контролировать модель из нескольких наиболее характерных точек и быстро между ними переключаться.

Продолжаем добавлять сплайны и постепенно создавать каркас, контролируя создаваемые сплайны во всех окнах проекций.

Пришло время немного отвлечься от непосредственного построения модели и выслушать несколько советов. Чем больше мы создаем дополнительных сплайнов, тем сложнее затем разобраться в их взаимном расположении, но тем не менее есть способы разобраться в этой паутине сплайнов. Во-первых, не забывайте о том, что работая со сплайнами и вершинами, вы всегда можете с помощью команды **Hide в** свитке Geometry спрятать мешающие вам сплайны. Во-вторых, очень удобно использовать **Viewport Clipping** в окнах проекций для отсечения мешающих

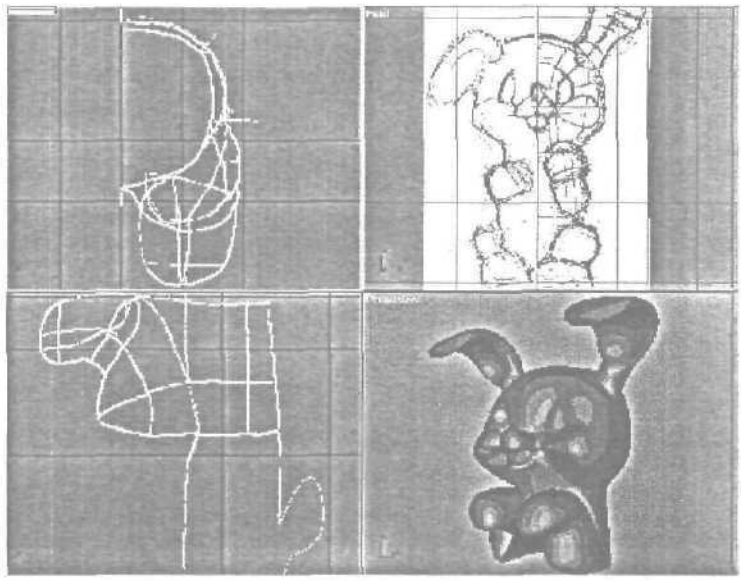
плоскостей пространства, с которыми вы в данный момент не работаете. Особенно это полезно при работе с видом сверху, когда из-за большого скопления линий и узлов трудно разыскать нужную точку внизу модели. Таким образом, вы оставляете в поле зрения только те сплайны, с которыми непосредственно в данный момент работаете. Кроме всего прочего хочется сказать пару слов о том, что аккуратность в моделировании — это не последнее дело. Обращайте внимание на то, чтобы сплайны не только повторяли контуры рисунка, по и имели плавные линии, повторяющие геометрию модели.

Итак, продолжим нашу работу. Построим некоторое количество сплайнов формирующих пасть и сделаем правое ухо.



Как вы уже, наверное, успели заметить, мы не стремились к детализации за счет увеличения количества сплайнов. Как раз наоборот, как говорилось об этом выше, чем меньше их будет, тем лучше (при условии, что это не повлияет в худшую сторону на внешнем виде модели).

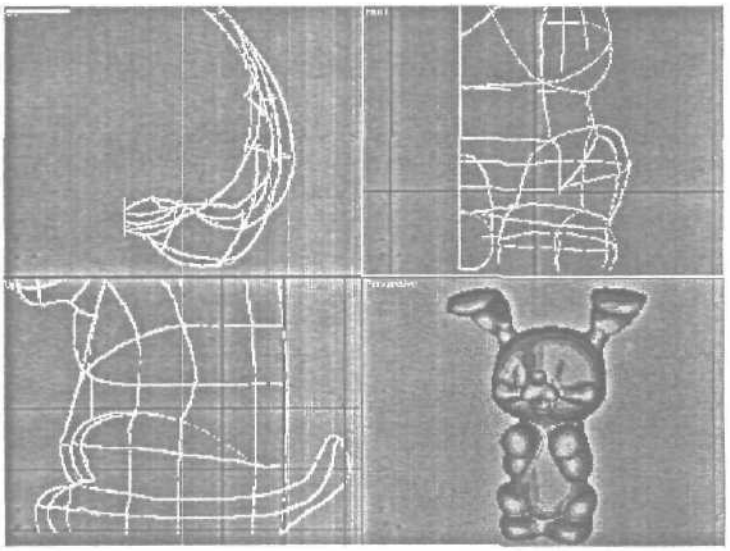
Исходя из этого, ухо построено всего из пяти сплайнов, три из которых формируют поперечное сечение. Напоминаем, что мы строим только правую половинку модели, поэтому пока не будем обращать внимание на то, что левое *ухо* не соответствует рисунку. Несколько позже мы вернемся к этому вопросу, а пока будем наращивать сплайны, продвигаясь вниз.



В процессе работы с конечными точками сплайнов, MAX предлагает их склеить, если они расположены достаточно близко друг к другу. Обычно делать этого не стоит по той простой причине, что в дальнейшем будет затруднена корректировка сплайнов. Кроме того в параметрах модификатора Surface надо активизировать Remove interior patches. Это позволит убрать внутренние патчи (грубо говоря перегородки) с таких мест как уши и лапы.

В некоторых случаях, когда вы применили модификатор Surface к построенным сплайнам и не получили желаемый результат (появились дыры, патчи перевернуты или скручены), причиной может быть не только присутствие пятой точки, но и неправильная ориентация какого-то сплайна по отношению к остальным. Для этого в модификаторе Edit Spline, при выделенном сплайне, надо попробовать применить Revers или Mirror.

На рисунке внизу вы видите практически построенную половинку собаки, которая благодаря примененным модификаторам и маленькой уловке, о которой писалось выше, смотрится как цельная форма.



Сейчас, когда основная работа сделана, пришло время придать непринужденность позе и отойти от осевой симметрии. Для этого удалим нашу копию модели с модификатором **Surface** — в таком виде как сейчас она нам больше не понадобится, а для рабочих сплайнов (половинки, над которой мы работали) сделаем копию (mirror) и объединим их.

Делается это так: вначале делаем зеркальную копию, затем вновь образованные сплайны приаттачиваем к уже существующим и, наконец, объединяем внутренние вершины двух половинок. В результате всех этих операций вы получите полный каркас из сплайнов и вот сейчас можно приступать к непосредственному редактированию сплайнов для придания модели индивидуальности. Достигается это путем простого перетаскивания вершин левой половины модели на новые места, согласно эскиза виртуальной студии.

Закончив с расстановкой вершин на свои места можно сплайны снова покрыть поверхностью с помощью модификатора Surface и в таком виде считать работу с геометрией законченной. А сейчас, после того, как закончили с геометрией можно приступать к отекстуриванию модели. Процесс этот достаточно сложный и трудоемкий, требующий отдельной статьи, поэтому здесь будут описаны только основные шаги создания текстуры.

Т.к. у нас сложная геометрия и ее невозможно корректно покрыть одной текстурой, то придется создавать **Multi/Sub-Object** материал, а сле-

довательно и объект нужно предварительно перевести в **mesh,** для этого «прибиваем» стек модификаторов и в качестве результирующей выбираем **Editable Mesh** — это нам позволит назначить ID материалов.

Конечно, если у вас в избытке памяти, то вы можете просто добавить в стек модификатор **Edit Mesh**. Далее надо проанализировать нашу модель на предмет выявления отдельных деталей, на которые будут наноситься материалы.

Их получилось 10: тело, правое ухо, левое ухо — перед, левое ухо — зад, кончик носа, верхняя челюсть, нижняя челюсть, правая лапа, левая лапа, хвост. У вас может получиться другое количество материалов, все зависит от ваших предпочтений, но главное — это выбрать места и способ наложения материалов так, чтобы они как можно меньше деформировались на геометрии.

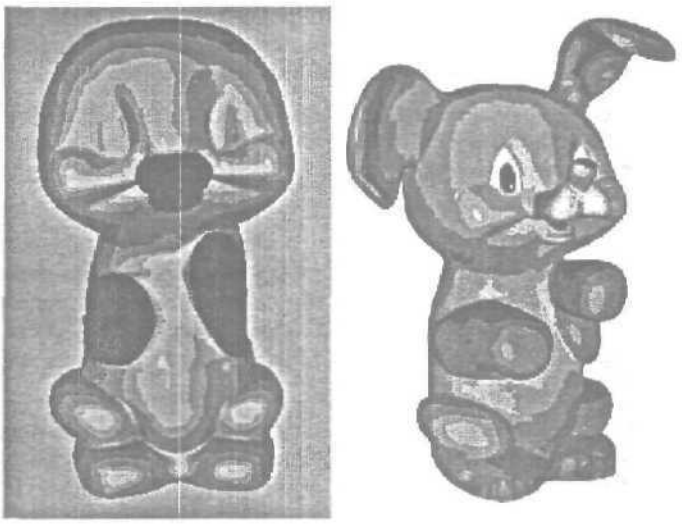
Итак, приступим к назначению ID для тела, для этого в качестве Selection **Level** выбираем **Poligon** и выделяем всю модель, а затем нажав и удерживая Alt удаляем все ненужные нам полигоны. Не забывайте контролировать процесс в других окнах проекций, а так же поворачивая модель.

Но лучшим способом контроля выделенных полигонов является скрытие ненужных полигонов командой **Hide,** тогда сразу становится видно, если где-то упущен полигон, либо наоборот— выделены лишние.

Сейчас нужно в **Material ID** поставить номер материала (в нашем случае это I) и запомнить этот номер для **Multi/Sub-Object** материала (номер ID и номер материала должны соответствовать). После этого в стек добавляется **UVW Map и в** качестве параметра наложения карты выбирается **Cylindrical,** как наиболее подходящую в данном случае.

На уровне Sub **Object** — **Gizmo** нужно выделить линию стыка карты, и желательно, чтобы она оказалась в наименее приметном месте (например на спине или сбоку).

После этого есть два *пути:* первый — это воспользоваться плагином все того же Питера Ватье, который называется Unwrap **Objeck Texture** и находится во вкладке утилиты, либо второй — программой **Deep Paint** и ее подключаемым к MAX-у плагином. И в том и в другом случае результат будет одинаков, за исключением того, что **Deep Paint** позволяет рисовать прямо на трехмерном объекте и выполнять доводку в программе растровой графики (например Photoshop), а после **Unwrap Objeck Texture** нужно сразу все делать в такой программе.



Мы не стали «усложнять себе жизнь» и перегнали объект в **Deep Paint,** там наметив основные цветовые пятна для Diffuse карты и места выдавливания для карты **Bump,** а затем все доработали *к* Фотошопе. Карту выдавливания и карту цвета для тела

Все дальнейшие действия производятся по аналогии с описанными выше. Единственное, на что нужно обратить внимание — это на пропорции карт. Т.к. в нашем случае карта для тела является самой большой, то все остальные карты должны быть пропорционально меньше, иначе вы не сможете добиться корректной стыковки двух соседних карт. Так же полезно пользоваться инструментом «штамп» в Фотошопе для более точной подгонки соседних карт в местах стыковки.

Ну вот, собственно и все, что можно рассказать о создании моделисобачонки.

Глава 16. Интеграция 3D в фотографию

Многим по роду своей деятельности часто приходится иметь дело с проектами, которые требуют размещения рекламных конструкций прямо на фотографин. Область применения такого подхода очень широко: от архитектуры, коллажей до создания видео.

Предположим, что у нас есть идея для проекта (создание стелы для банка), ее конструктивное решение. Но заказчику мало просто описать проект, он хочет видеть каким он будет в жизни. Ну что ж, попробуем выполнить пожелания заказчика.

Итак, приступим.

Все начинается с фотографии. Первым делом нужно определиться с каким качеством мы должны будем распечатать проект для заказчика, отсюда будет зависеть разрешение при сканировании. Если фотография 9x12, а печать будет на листе формата А4, то сканировать надо с разрешением 300 dpi и предпочтительнее работать с форматом TIFF.

Подготовительная часть закончена, запускаем MAX. Первым делом надо разместить фотографию в качестве **background** в окне перспективы. Для этого выбираем в меню Rendering ➪ **Environment** ➪ Environment Map, и в качестве карты Bitmap выбираем нашу фотографию.

Обратите внимание на то, что когда откроется диалоговое окно «Select bitmat image file» и вы активизируете изображение, внизу, в графе Statistics будет дано разрешение в пикселах, надо запомнить или записать это значение для параметров рендера.

Жмем OK и на кнопке появляется название нашего файла. Иногда требуется небольшая коррекция для фотографии в процессе работы и для того, чтобы это было возможным в MAX, достаточно скопировать ее в редактор материалов. Для этого открываем редактор **Tools** ➪ **Material** Editor и кликнув по кнопке в окне **Environment** (если вы его еще не закрыли) перетаскиваем ее в любой свободный слот в редакторе материалов, а в качестве метода копирования выбираем instance. Там, во вкладке Out-

put, достаточно настроек, чтобы подкорректировать при необходимости изображение,

Полдела сделано. Далее надо прописать это изображение в окно перспективы. Для этого откроем Views ➪ Viewport Background и там активизируем Match Bitmap, Display Background, Lock Zoom/Pan и в последнюю очередь Use Environment Background. В качестве Viewport выбираем перспективу и после нажатия на OK получаем наше изображение в окне перспективы. Ну и чтобы совсем закончить с настройками, связанными с фотографией, перейдем к настройке рендера Rendering ➪ Render. Вот тут нам и пригодится размер фотографии в пикселах — их нужно прописать в Output Size в качестве ширины и высоты — это сохранит пропорции фотографии и отрендерит изображение с максимальным качеством. Кроме того для тестовых рендеров часто используются настройки Draft Render, где сохраняя параметр Image Aspect (т.е. пропорции) можно уменьшить в несколько раз размеры изображения для скорейшего рендера сцены.

Далее приступаем к анализу нашей фотографии. Надо определить точку, с которой производилась съемка (что необходимо для правильной постановки камеры в сцене), а так же проанализировать свет и тени (это пригодится для выставления светильников).

Начнем с камеры. Логично предположить, что фотоаппарат находился на уровне глаз фотографа, значит и камеру в сцене надо выставить на высоте 1600-1700 мм (за отметку земли возьмем начало координат по оси Z). Target камеры будет находиться несколько выше, т.к. фотография сделана под небольшим углом. Конечно, идеальным вариантом было бы знать реальные размеры объектов на фотографии (например столбов) и расстояния до точки съемки, но попробуем разобраться без них.

Что касается света, то судя по теням от машины и столба можно предположить, что солнце находилось слева и немного впереди.

А сейчас нужно сделать небольшое отступление и сказать несколько слов по поводу последовательности в работе такого характера. Что касается данного примера, то можно было вначале построить всю геометрию (плоскость земли и саму стелу), согласно реальных размеров, а лишь только потом начинать работу с фотографией. Так вы не будете загружать понапрасну свой компьютер. Но гораздо чаще приходится строить объекты используя фотографию, т.е. заниматься подгонкой прямо по месту (например, добавить несколько мелких деталей в интерьер комнаты), потому и была описана именно такая последовательность.

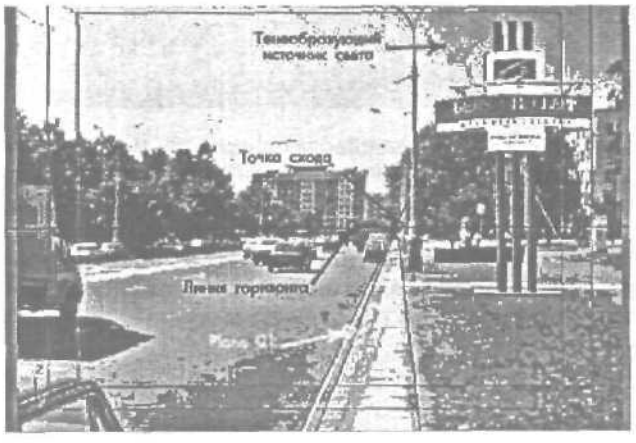
Ну, что ж, вернемся к работе.

Предположим, что мы уже раньше сделали модель стелы и сейчас только экспортируем ее в сцену. Далее «подведем» под ее плоскость (Create ➪ Geometry ➪ Plane), которая будет лежать на уровне земли и в дальнейшем будет использована для «принятия» тени от стелы. Исходя из этих соображений размер плоскости может быть любой, но не менее того, на который уместится тень (ведь это ее основное назначение). Кроме того, как вы увидите ниже, используются грани плоскости для выравнивания относительно точек схода. На этом с геометрией закончили — теперь вернемся к Камере.

Построим на виде сверху камеру **(Create ➪ Cameras O** Target) так, чтобы она была направлена на «лицо» стелы, затем, не снимая выделения активизируем в главной панели значок трансформации **(Select and** Move) **и** кликаем по нему правой кнопкой мыши, после чего появится окно для ввода числовых значений трансформации, где в поле оси Z введем 1700 (расстояние от земли до камеры).

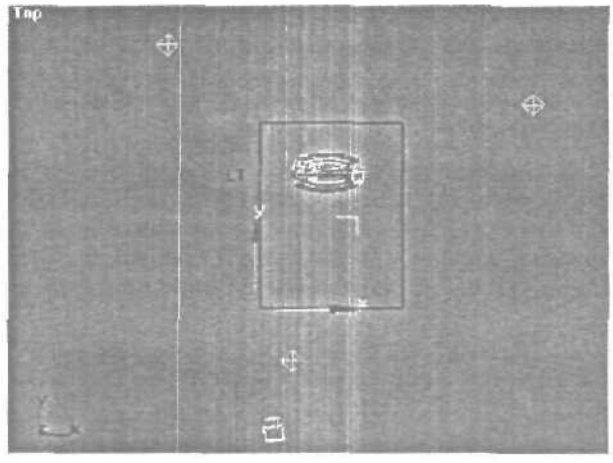
Сейчас можно окно перспективы заменить на вид из камеры и продолжить настройки. Для согласования линии горизонта камеры с горизонтом на фотографии надо включить показ горизонта камеры во вьюпорте. Для этого, при выделенной камере, зайдите в панель **Modify** и активизируйте Show **Horizont** во вкладке **Parameters.**

На рисунке внизу представлены составляющие правильной настройки камеры и освещения. Для наглядности проведены (можно это сделать, подключив воображение) две линии параллельно бордюрному камню и по краю газона (в жизни они обычно параллельны). На пересечении мы получили точку схода, а следовательно и линию горизонта фотографии.

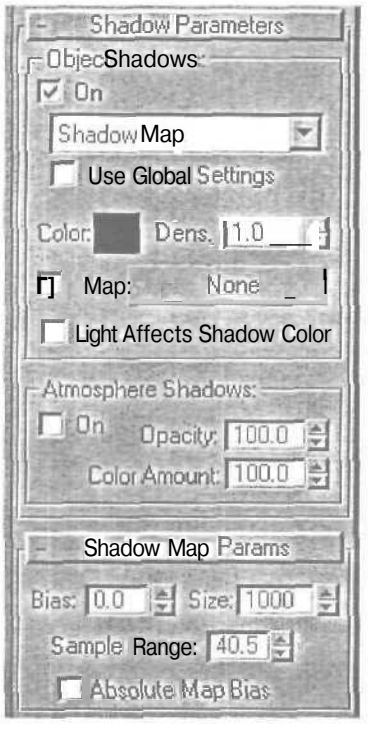


Сейчас используя инструмент Select **and** Move, передвиньте Target камеры по оси Z гак, чтобы черная горизонтальная линия (горизонт камеры) совпал с точкой схода. Все, по оси Z камеру двигать больше не нужно — она заняла свое место. Осталось только перемещать ее по осям X и Y для того, чтобы объект встал на то место, которое для него предназначено. Стеллу предполагается расположить перпендикулярно дороге, следовательно по левой грани плоскости **Plane 01** (а стела согласована с этой плоскостью) можно выставить всю геометрию. Для того, чтобы проще было настраивать камеру, в Object Propertis для **Plane 01** (окно появляется после нажатия правой кнопкой мыши на **Plane** 01) установлен **Display** As **Box.**

На рисунке вверху плоскость обозначена белой линией, а крайняя левая линия указывает на то, что OH;I выставлена параллельно дороге. А на рисунке внизу показан вид сверху, где видно окончательное расположение камеры и светильников.

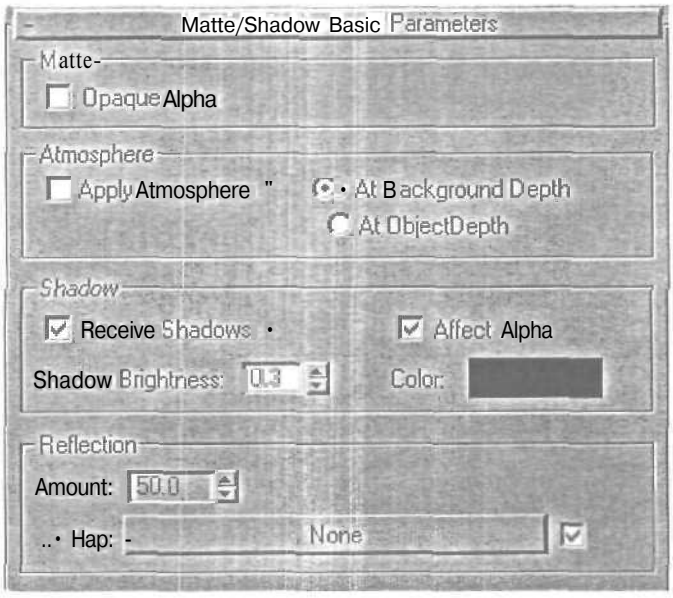


Сейчас несколько слов по поводу освещения. Всего в сцене 4 светильника: три **Omni** (заполняющие) и **Target Direct. Для** источника света, который будет генерить тени, выбран Target **Direct,** как наиболее близкий по отбрасыванию теней к солнцу (у солнца лучи почти параллельны). Источник выставлен согласно описанному выше анализу — слева и немного впереди. Для более реалистичной тени применен **Shadow Map** с параметрами, которые вы можете видеть ниже.

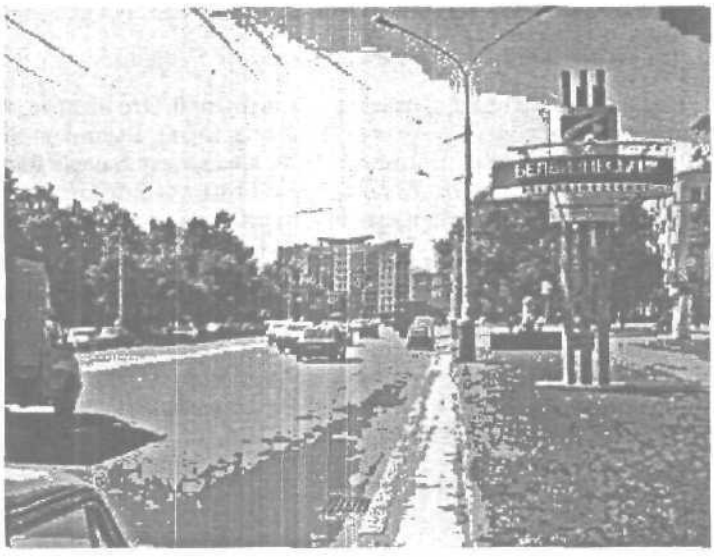


Как вы можете видеть, параметр Bias равен 0. Это не позволяет тени «отрываться» от объекта, который ее отбрасывает. Выставив Size равным 1000 тень была сделана более ровной, а параметр **Sample Range** «генерит» размытые края тени. Все эти параметры подбираются опытным путем в зависимости от задач и выходного размера картинки.

Далее поговорим о материалах, точнее об одном материале, который назначается **Plane 01. Это Matte/Shadow** материал — именно он позволяет принимать тени, сам при этом оставаясь невидимым. Ниже вы можете увидеть настройки этого материала — они достаточно просты. Просто установите флажок **Receive Shadow и Affect Alpha.** Кроме того, с помощью параметра **Shadow** Brightness можно изменить яркость тени, а параметр **Color** меняет цвет тени.



Остальные материалы настраиваются как обычно.



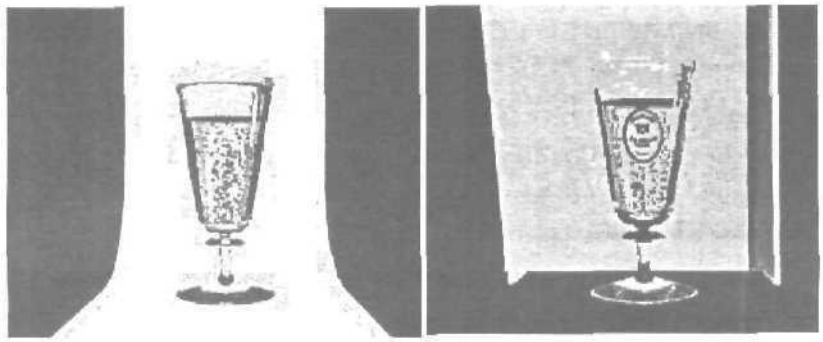
И напоследок несколько советов. В нашем случае тень от стелы падала на ровную землю, а что если тень падает на стену здания с колоннами, окнами и пилястрами? Все не так страшно, просто надо сымитировать выступы и впадины (достаточно, если это будет весьма приблизительно) и с расстояния вы ничего не заметите. Другое дело, если объект показан крупным планом — тут придется потрудиться. Кроме того фотографии чаще всего имеют глубину резкости и надо подбирать параметр сглаживания при рендере, чтобы объект не казался инородным телом. Это же относится и к подбору цвета для светильников (надо попасть в цветовую гамму фотографии).

Глава 17. Бокал пива

Во многих студиях чаще всего цифровую камеру приходится замещать 3dMAX. Но всегда хочется сделать рекламу как на западных каналах.

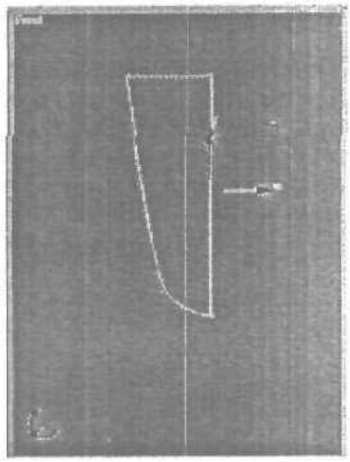
Для качественного результата требуется опыт создания реалистичной воды и стекла.

Вот что получилось при изготовлении рекламы пива на желтом и белом фонах. Как видите есть разница.

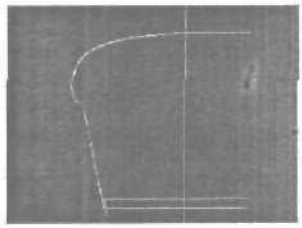


Для начала сделаем бокал (каждый из вас может сделать свою форму бокала). Из панели Create выбираем сплайн и рисуем контур бокала.

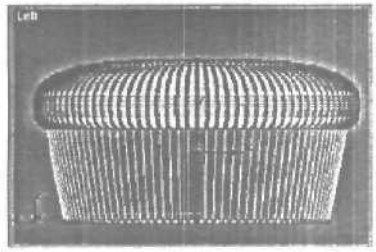
Рисуем само пиво, т.е. то, что налито в бокал. Тоже при помощи сплайна.



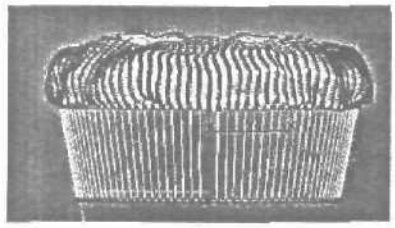
К обоим сплайнам применяем модификатор **Lathe** с достаточным количеством сегментов, в нашем случае их 50. И поскольку ось модификатора Lathe совпадает с Пивот Пойнтом сплайна, нужно потом скорректировать положение оси вращения. Координаты расположения бокала в пространстве: 21.127; **-168.176;** -77.3. Это важно, т.к. присутствие других предметов в сцене имеет свое значение. Теперь сделаем пену. Пиво же — пенное! Тоже при помощи комбинации **Spline-Lathe:**



Теперь к объекту-пене применяем модификатор **Volume Select.**



Теперь очередь Noise, Scale=20, StrenthX=Y=Z=10.

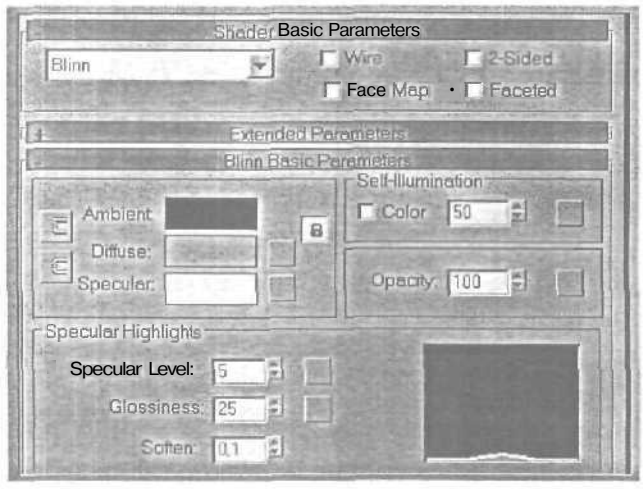


Все три объекта при помощи команды Align выравниваются по осям Y и X. Теперь нам понадобится среда или в 3D-шном простонародье — environment. Создаем 3 бокса, с размерами 1000, 600, 10. Координаты в пространстве:

box center: 22;163;222
box right: 207;138;222
box left: -242;163;222

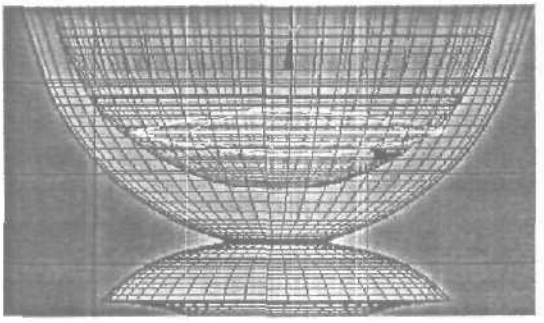
Этот бокс повернут по оси Y на -95.5 град.

Всем трем присвоен стандартный материал, с shader — Blinn, Цвет: Ambient 0,0,0; Difuse 255,198,85; Specular 255,255,255. Self-illumination 50. Все остальное как на рисунке ниже:

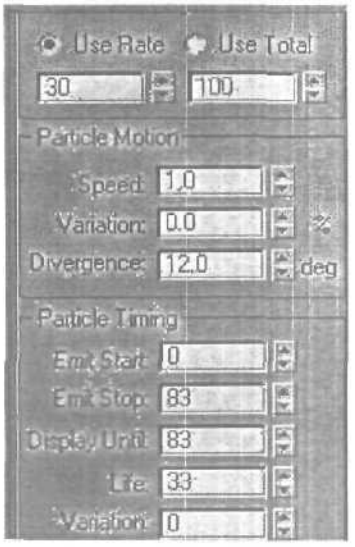


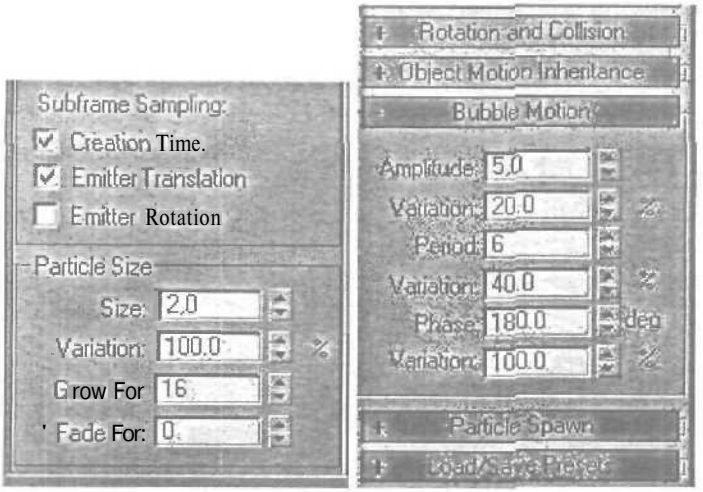
Пол бокал ставим Quad Patch, также стандартный материал с полностью черным цветом. Ему при желании можно применить карту Flat Mirror на позицию Reflection.

Теперь нужно сделать «булъки», т.е. пузырьки. Для этого на дне чаши располагаем **OilTank:**



Создаем PArray, который располагается, на том же месте, что и **OilTank. В** свитке **Basic** Parameters нажимаем кнопку **Pick Object** и выбираем **OilTank, он и** будет теперь нашим эмиттером. В **Particle Formation** выбираем At **Distinct Points** и задаем ему значение 20. Это позволит нашим пузырькам появляться сразу с 20 различных точек, что придаст природную хаотичность. Не забудьте в свойствах объекта **OilTank** выключить флажок **Renderable.** В свитке Load/Save Presets загружаем стандартную настройку **Bubbles** и добавляем кое-какие детали.



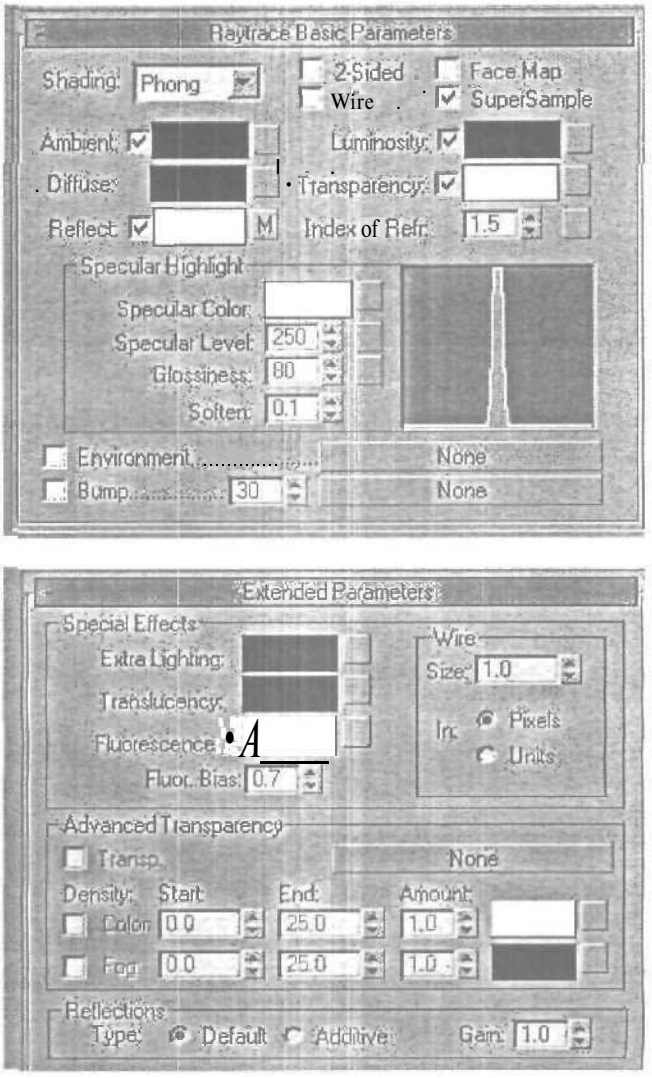


Теперь Свет. Его много:

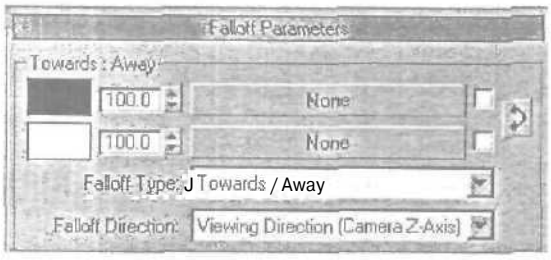
Light	Position (X,Y,Z)	Hotspot/Falloff	Color	Multiplier	Cast Shadows	Exclude
Omni 1	56,28,-37	---	255,255,255	06	no	box right, PENA
Omni 2	139,-390,163	---	255,255,255	1 C	no	PENA
Spot 1	-492,-678,-187	38/55	255,255,255	1 0	yes	boxright,PENA
Spot 1 Target	-18, 222,-19					
Spot 2	-337,-902,789	100/102	255,255,255	1.0	no	PArray
Spot2 Target	-12,-141,28					
Spot 3	793 ,-287,50	70/90	255,255,255	1.0	yes	box right, PENA
Spot3 Target	98,-234,19					
Spot 4	58,-823,364	48/50	240,224,206	0.6	yes	Beer cap, box right
Spot 4 Target	58,-137,179					
Spot 5	583,-902,789	100/102	255,255,255	1.0	no	3eer Cap, Beer liquid, Lox right ,0il Tank. PEMA, Quard Patch
Spot 5 Target	-12, -141,28					

Ну-с. теперь приступим к более важной части; материал пива и стекла. Начнем со стекла.

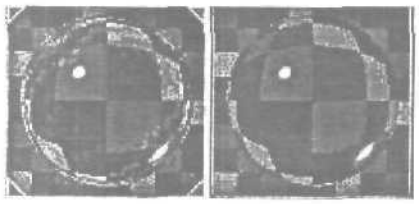
В Материал-редакторе открываем библиотеку RayTraced_01.mat, из нее выбираем материал Glass Clear. И делаем следующие изменения:



Вот так выглядят настройки для карты Reflect:

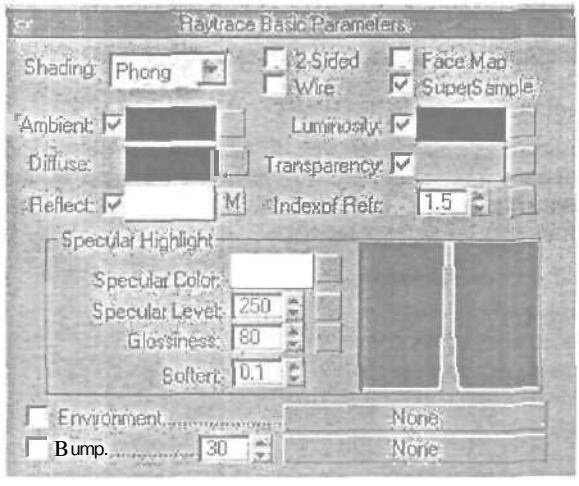


А вот так выглядят измененный (слева) и стандартный материалы для стекла:

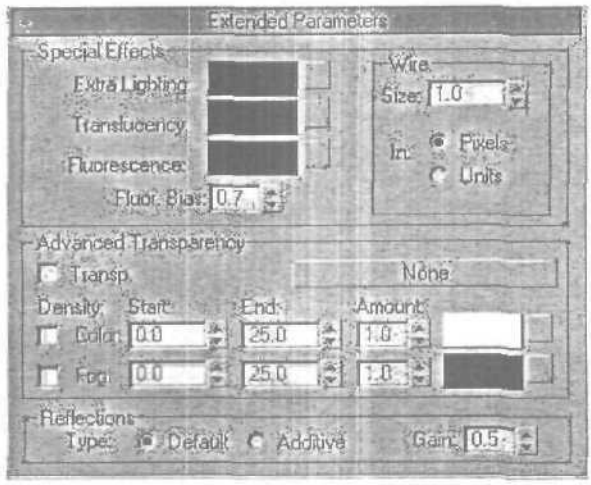


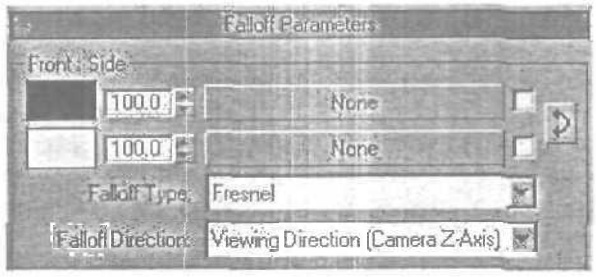
Цвет, который вы видите белым и есть белый на 100%, то же касается черного.

Теперь сотворим материал для жидкости пива. Из той же библиотеки RayTraced_01.mat выбираем **GlassYellow**. А потом:



Желтый цвет: 255, **214,** 62.





Этот материал мы присвоим к Пиву.

А теперь материал для пузырьков. Скопируйте этот материал для пива в свободный слот и назовите его **Beer Bubbles**. Меняем только настройку прозрачности **Transparency** **Yellow: 248,235,144.** Цвет среды «environment» для всей сцены — черный на 100%.

Материал для пены: **Standart,** Phong.

Ambient: 0.0.0
Diffuse=Specular=255.255.255
Self-Illumination: 195.195.195
Specular Level =100
Glossiness=5

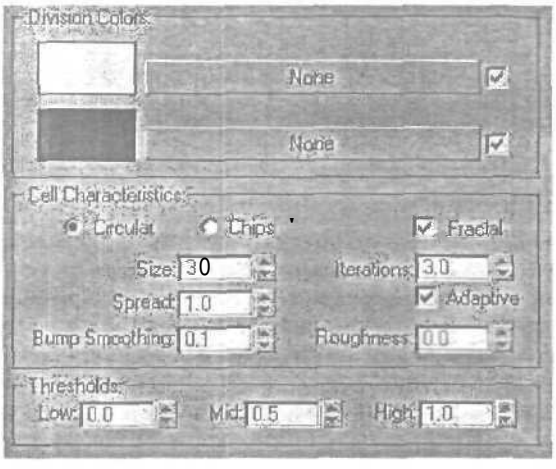
Теперь карты:

Diffuse Color (100)= Opacity(100) = Bump(200) = Cellular

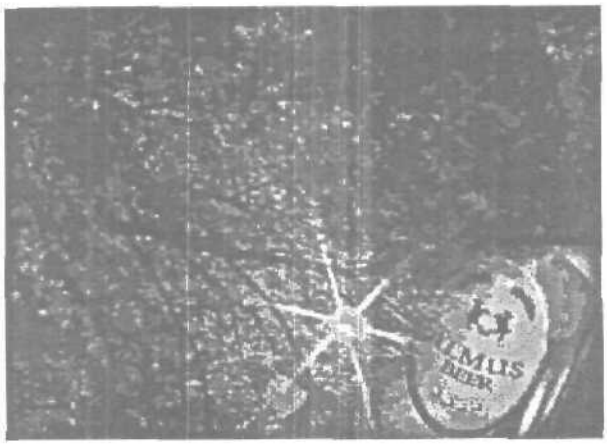
Применив ее **(Cellular)** на одну карту, на другие копируем ее как образец. Настройки следующие:

Cell Color=255.255.255
Division #1=243.243.243
Division #2=119.119.119
Ceil Characteristics - Circular

Дальше, **как на рисунке:**



Ну, вроде все, ставьте на рендер. Вот кадры **из** рекламы:



На рисунках вы можете видеть и сравнить заснятую на S-VHS бутылку и сделанную анимацию.

Т.к. материал **Raytrace** не только пропускает лучи, но и отражает и преломляет их, значит нужно не только правильно установить настройки выше упомянутых параметров, но и создать для этого материала подходящую среду. От этого свойства этого материала будут более реалистичными. Художники-фотографы так и поступают. И еще, т.к. мы работаем под Windows, то и в MAXe одного и того же результата можно добиться как минимум тремя различными способами. И поэтому, с помощью карт Thin **Wall Refraction и Reflect/Refract**, с применением шести-файловой среды даст почти такие же результаты. Смысл в том, что считать будет легче.

Глава 18.
Объекты в фокусе камеры

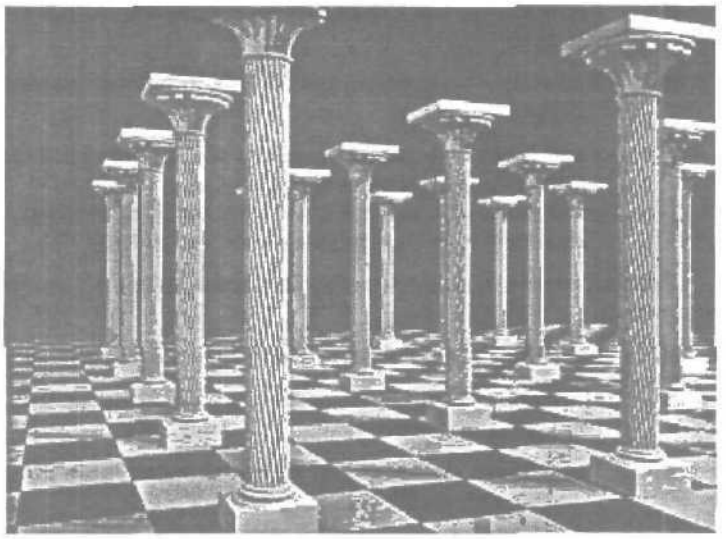
Вы когда-нибудь задумывались над тем, почему порой сцены, созданные в программах 3D моделирования, выглядят неестественными. Тому может быть несколько причин: начиная от неверно выставленного освещения и заканчивая неправильным наложением текстур. Но в этой главе хотелось бы затронуть такую тему, как глубина резкости и фокусное расстояние. Ни для кого не секрет, что глядя на фотографию, экран телевизора или просто на какие-либо предметы на улице или дома, мы не видим все одинаково четким. Это связано с особенностью строения глаза, устройством фотоаппарата и камеры. И невзирая на такой очевидный

факт многие пренебрегают глубиной резкости в своих работах. Ну, что ж, попробуем разобраться, что же такое глубина резкости и фокусное расстояние, применительно к сценам в 3D MAX.

Мы рассмотрим четыре подхода к решению этой задачи: первый — это использование встроенного **blur из Video** Post, фильтра **Depth** of Field (глубина резкости) из **Rendering Effects,** альтернативного метода, использованного для выполнения одного из рекламных проектов, связанных с оптикой, а так же возможностей настройки камеры.

Что ж, начнем.

Для начала надо создать какую-нибудь сцену, которая будет служить нам в качестве полигона для рассмотрения наших задач. Наверное уже становится традицией для такого рода примеров использовать колонны — не будем ее нарушать.



Далее нужно установить в сцене камеру и хочется по этому поводу сделать небольшое отступление. Многие дизайнеры, на 80% работающие с перспективой, отодвигают роль камеры на задний план. Так вот, такое решение — ошибочно. Дело в том, что кроме того, что камера позволяет более гибко настраивать сцену, вид из камеры получается более естественным, не говоря уже о многом другом.

Итак, у нас есть сцена и есть камера. Стоит обратить ваше внимание на то, что **target** камеры установлены на колонну, находящуюся в

центре (выделенную цветом) для *того, чтобы* при настройках глубины резкости, где используются параметры камеры, эта колонна всегда оставалась в фокусе. Опять же глубина резкости будет настраиваться именно по этой колонне — так будет проще сравнить результаты всех методов.

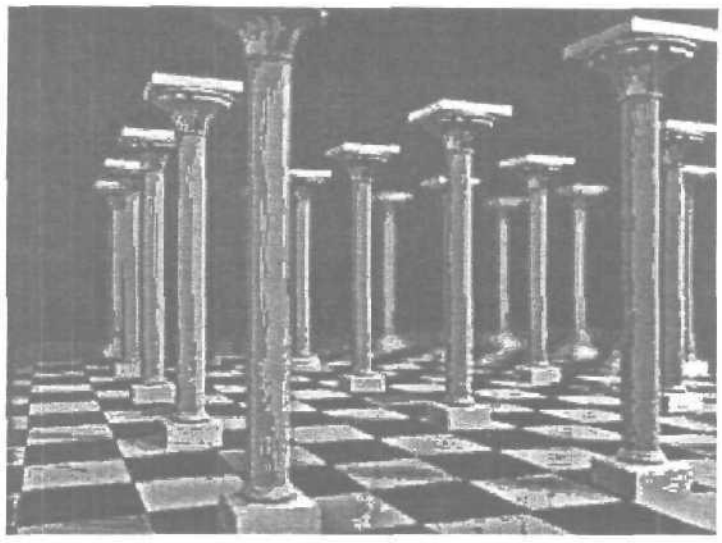
После того, как мы выставили камеру, идем в меню **Rendering ➪ Video Post,** жмем на кнопку Add Scene **Event** и выбираем вид из камеры. После этого нужно добавить сам фильтр. Жмем на кнопку **Add Image Filter Event-** Из списка выбираем: **Lens Effects Focus и** жмем «Ok». Потом в **Video Post** там, где слева показана структура очередности задач, выделяем **Lens Effects Focus и** щелкаем по нему два раза, затем в появившемся окне по кнопке **Setup и** после этого попадаем в окно настройки параметров **Lens Effects Focus.** Для того, чтобы увидеть в окне настройки вашу сцену, надо активизировать кнопку **VP Queue.** Далее обратимся к настройкам этого окна. В левой части находятся три переключателя:

- **Scene Blur** — эффект размытия применяется ко всей сцене в целом;
- **Radial Blur** — размытие сцены происходит радиально, от центра к ее краям
- **Focal Node** — позволяет выбрать объект, который будет являться центром фокусировки. Собственно им и воспользуемся, выбрав в качестве объекта колонну, находящуюся в центре (выделенную цветом).

Справа, в окне, есть еще несколько параметров, на которых стоит остановиться.

- **Affect Alfa** — воздействие размытия на альфа канал;
- **Horiz/Vert Focal** Loss — величина размытия сцены по горизонтали и вертикали;
- **Focal Range** — определяет на каком расстоянии от центра (объекта) начинается размытие;
- **Focal Limit** — расстояние, на котором будет достигнут максимум размытия сцены.

«Поиграв» немного с настройками рендерим сцену, нажав **Execute Sequence** (иконка с бегущим человечком) в **Video Post** и получаем вот такое изображение.



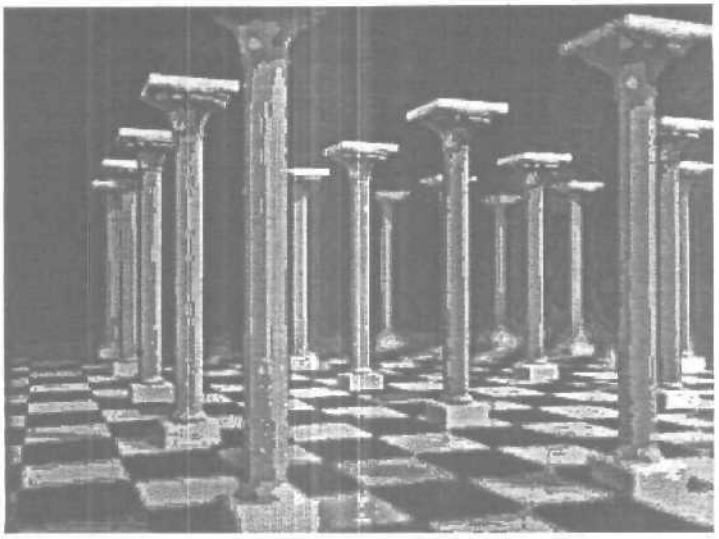
Далее приступаем к описанию второго способа установки глубины резкости в сцене. Воспользовавшись все той же сценой зайдем в меню **Rendering ➪ Effects** и для загрузки фильтра нажимаем на кнопку **Add,** после чего выбираем эффект **Depth of Field** (глубина резкости).

Как только вы активизируете этот фильтр в нижней части окна появится свиток **Depth of Field Parameters,** который позволяет настраивать глубину резкости в сцене. Итак, обратимся к параметрам этого фильтра. Они во многом повторяют настройки, рассмотренные выше для **Lens Effects** Focus, поэтому остановимся только на некоторых из них.

Pick Cam. поля Cameras позволяет выбрать одну или несколько камер прямо из окна проекций. В поле **Focal Point** можно выбрать для центра фокусировки объект **(Pick Node),** указав его в любом из окон проекций, либо использовать камеру (Use **Camera),** воспользовавшись фокусным расстоянием камеры.

Ну и в поле Focal **Parameters** можно установить Use **Camera,** тогда характеристики глубины резкости (такие как диапазон и лимит) будут определяться параметрами выбранной камеры. Пожалуй, все не так сложно...

Попробуйте изменить настройки этого окна в ту или иную сторону и посмотрите на результат. Плод таких манипуляций с настройками вы можете посмотреть ниже.



Ну а сейчас перейдем к описанию более сложного метода, но более правильного с физической точки зрения. Как вы уже, наверное, успели заметить, два предыдущих метода хороши своей простотой, но они не совсем верно передают эффект глубины резкости и размытие появляется там где его как казалось бы не должно быть и наоборот. Вот потому-то, когда понадобилось делать рекламный плакат, необходимо было искать альтернативу этим двум методам.

Заключается этот метод в том, чтобы отрендерить последовательность кадров, изменяя положение камеры в пространстве (target камеры при этом остается на месте), а затем собрать все кадры вместе.

Взяв все ту же сцену с камерой построим дополнительный объект — это будет круг **(Create** ⇨ Splines ⇨ Circle). Радиус круга должен быть небольшим. Затем расположите этот сплайн так, чтобы камера оказалась внутри круга, а сам сплайн стал перпендикулярно **target** камеры. Не нужно все это выстраивать с абсолютной точностью, применяя дополнительные средства выравнивания, достаточно воспользоваться инструментами трансформации **Move и** Rotate.

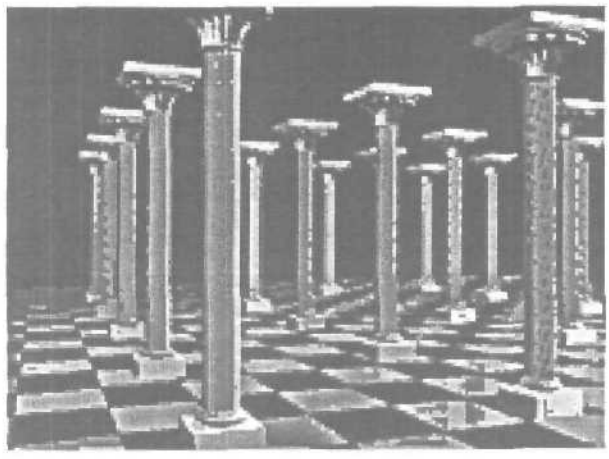
Следующим шагом будет привязка камеры к сплайну. Для этого выделяем камеру и идем на командную панель Motion, кликаем по кнопке **Parameters** и в свитке Assign **Controller** выделяем строку **Position,** затем щелкните по кнопке со значком контроллера (в верхнем левом углу свитка), после чего выбираем из списка контроллер **Path Constraint,** после

этого появится вкладка **Path Parameters. В** этом свитке кликните по кнопке **Add Path** и выберите во вьюпорте круг, После всех этих манипуляций камера займет свое место в начале сплайна.

Несколько позже нам понадобится отрендерить последовательность кадров, поэтому следующим шагом будет определение количества кадров для рендера. В этом вопросе надо проявить определенную гибкость: чем больше кадров — тем качественнее будет конечный результат, но тем дольше все будет рендериться. Остановимся на 10 кадрах. И делается это так; кликните правой кнопкой мыши на изображении ключа, рядом с большой кнопкой **Animate** (справа внизу) и в появившемся окне в поле Animation, напротив **End Time** введите число кадров и нажмите на кнопку **Re-scale Time,** появится еще одно окно, где просто кликните на «OK» и еще раз на «OK» для закрытия окна **Time** Configuration.

После этого передвигаем ползунок таймера на последний кадр и нажимаем на кнопку Animate для того, чтобы начать запись и возвращаемся во вкладку **Motion ➪ Path Parameters,** где в поле **Path Options** напротив % **Along Path** выставляем цифру **100** (это значит, что камера пройдет по сплайну полный путь). Выполнив все эти действия, еще раз нажимаем на кнопку **Animate** для того, чтобы остановить запись, в итоге у вас появится два ключа анимации на первом и на последнем кадре.

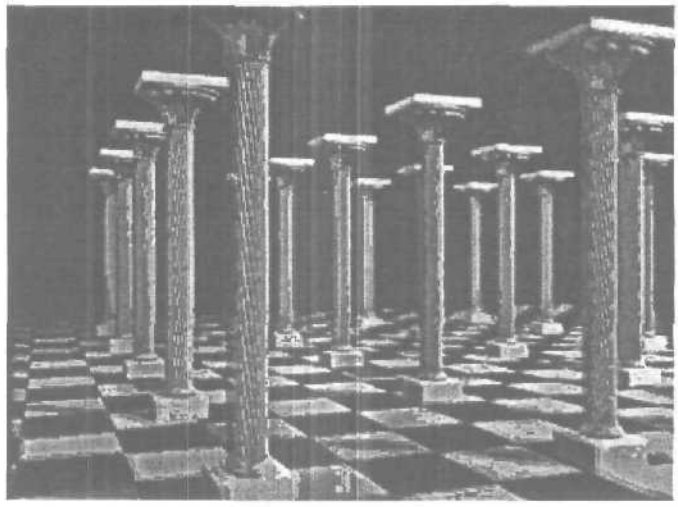
Далее приступим к непосредственной настройке размытия кадра. Для этого надо выделить круг, активизировать вкладку **Modify** на командной панели, после чего получим доступ к настройке радиуса круга. После этого запускаем анимацию, активизировав окно камеры и нажав кнопку **Play Animation.** Внизу вы видите, что получилось.



Нужно заметить, что если у вас в сцене много объектов или слабенький компьютер, то вы можете во вкладке Propertis установить для объектов свойство **Display As Box,** либо в окне камеры в выпадающем меню выбрать Other ⇨ **Bounding Box.** Так или иначе, но здесь вы должны посмотреть на смешение объектов в процессе выполнения анимации. Если оно слишком велико (следовательно и размытие будет большим), значит нужно уменьшить радиус круга. После этих настроек можно приступить к рендеру. Заходим в меню **Rendering** ⇨ **Render,** активизируем **Active Time Segment,** для того чтобы рендерились все 10 кадров. Ниже, в поле **Render** Output ставим флажок рядом с **Save** Fale и указываем директорию на диске, куда сохранятся наши кадры, а так же тип файла.

После всего этого жмем на кнопку **Render** и, в MAX вся работа закончена. Осталось только собрать все отрендеренные кадры воедино. Для этого можно воспользоваться программой Photoshop (или использовать любую другую).

После того, как вы загрузите все 10 кадров в Photoshop надо выбрать один любой кадр (файл) и на него перенести изображения с оставшихся 9. Для этого активизируйте инструмент **Move** на панели инструментов и удерживая нажатой клавишу **Shift** (это обеспечит полное совпадение слоев), перетаскивайте изображения на выбранный файл. В результате у вас получится изображение состоящее из 10 слоев. После этого каждому слою назначьте **Opacity** (прозрачность) равную 35% (подбирается опытным путем). Вот и все, остается только все это сохранить как один файл. Результат, как говорится, на лицо.



И напоследок рассмотрим новую возможность настройки глубины резкости. Все то, что с таким трудом мы проделывали в предыдущем методе, нашло свое отражение во вкладке **Depth of Field** настроек камеры target (Create ➪ **Cameras** ➪ **Target** или **Free** ➪ **Parameters ri**> Multi-Pass Effect ➪ **Depth Of Field).**

Итак, рассмотрим подробнее эти возможности. Обратимся к полю **Multi-Pass** Effect, именно здесь надо активизировать **(Enable)** данный процесс, выбрав в качестве метода **Depth of Field** или **Motion** Blur, а кнопка **Preview** позволяет просмотреть результат прямо во вьюпорте.

Далее перейдем к свитку **Depth of Field Pframeters,** там в группе **Focal Depth** есть флажок Use Target **Distance** и если он включен. то **target** камеры используется как точка, вокруг которой смещается камера, Если флажок снят, то для установки глубины, с которой начнется смешаться камера служит поле **Focal Depth.** Далее следует группа **Sampling.** В ней вы можете установить показ проходов во вьюпорте **Display Passes,** далее **Use** Original **Location** определяет будут ли начинаться проходы с первоначального положения камеры, или камера будет смешена.

Total Passes — число проходов для генерации эффекта.

Sample Radius — значение, на которое будет сдвинута сцена, чтобы гснерить расплывчатость. Увеличение этого значения ведет к усилению эффекта.

Sample Bias — смещает действие эффекта (размытия) к началу или к концу проходов (по умолчанию находится посередине).

Следом идет группа **Pass Blending.** В этой группе вы можете управлять сглаживанием мультипроходного рендера (во вьюпорте не виден). Активизированный флажок Normalize Weights помогает избавится от артефактов при рендере, увеличивая степень размытия.

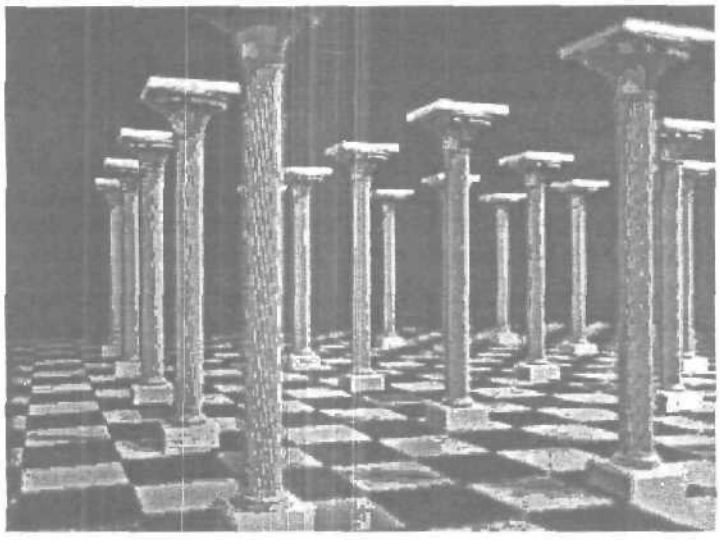
Dither Strength — управляет количеством полутонов в мультипроходах.

Tile Size — устанавливает размер перекрытия слоев, используемого в передаче полутонов.

Вот и все основные параметры для **Depth of Field.**

Motion Blur Pframeters почти полностью повторяют вышеперечисленные настройки, потому останавливаться на них не будем.

Ну. а то, что можно сотворить с нашей сценой, используя **Multipass Depth of Field** параметры для камеры вы можете видеть на рисунке внизу.



Итак, подведем итог. Улучшить вашу сцену можно многими способами, но не забывайте и про глубину резкости, особенно если вы моделируете большие пространства. Не следует забывать об этом и тогда, когда надо сконцентрировать внимание зрителя на одном предмете в сцепе. Рассмотренные выше примеры помогут вам в этом — останется только выбрать наиболее подходящий для вас метод.

Глава 19. Эффект Flow-Mo в 3ds max

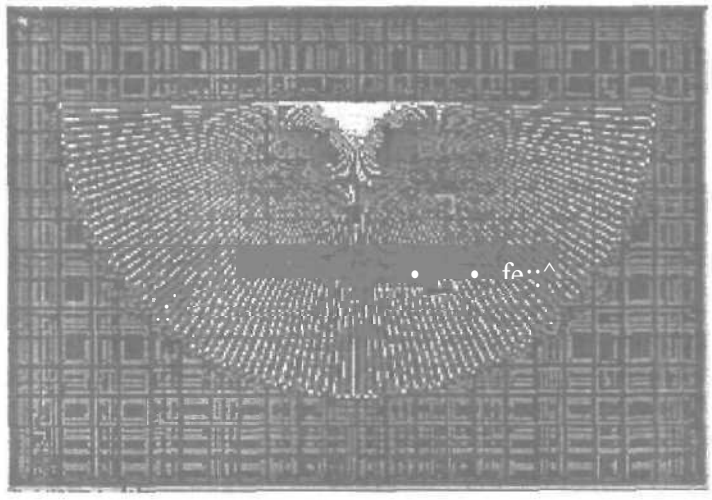
Помните в фильме «Матрица», был такой эпизод, в самом начале фильма полиция врывается в комнату, где сидит Тринити, и она от них, извините, «отмахивается». Помните кадр «разборки» с первым полицейским? Она подпрыгивает, камера поворачивается вокруг них более чем на 180 градусов, «немая сцена», как у Н.В. Гоголя. Красота.

Изобразить такое в «Максе», а затем собрать в «After **Effects**» или **«Premiere»** можно. Тяжело, но можно. Принцип довольно прост. Нужно чтобы в определенном кадре «сработало» сразу несколько камер. Им нужно отснять по одному кадру, и они должны быть расположены по круговой (вернее «секторной») траектории. Как это сделать, попытаемся вам объяснить. В качестве примера в сцене использованы **«Super Spray» и** «PArray» (несколько «дымков» и взрыв чайника (а кто не без греха?)).

Берем стандартные «Максовские» сто кадров (именно 100, а не 101, как по умолчанию), это четыре секунды, если по 25 кадров в секунду (стандартный PAL). Первая камера снимает с 1-го по 50-й, вторая с 50-го по 100-й (в итоге правда получилось 8 секунд и один кадр, но это лучше чем 7 сек. и 24 кадра). Далее, решаем на сколько градусов будем поворачиваться вокруг объекта съемки, и за сколько секунд. У меня поворот на 180 градусов за 4 секунды (100 кадров).

Для того чтобы равномерно установить «камеры», был сделан шаблон, по которому они были выставлены.

В роли шаблона — стандартный конус. Мне надо за 100 кадров повернуться на 180 градусов, исходя из этого выставляем количество сегментов конуса — 101. Почему 101? Потому, что вершин на одну меньше — если сегментов 101, то вершин 100). Ставим галочку в «Slice *On»,* и в «Slice **To»** ставим 180. Конвертируем конус в «**Editable Mesh».** Заготовка готова.



Начинаем устанавливать камеры. «**Camera01»** и «**Camera02»** необходимо расположить и «зашевелить» так, чтобы в 50-м кадре они находились на месте крайних вершин конуса, каждая у своей вершины, разумеется.

Теперь о «однокадровых» камерах. Поскольку их много, сто штук все таки, то их нужно «устанавливать» группами по 10 штук. И в монитор влезают, и не запутаешься. Итак «ставим» десять камер примерно у вершин конуса. «Первую» и «последнюю», естественно «не предлагать». Хо-

тя на самом деле, первая и последняя вершины конуса находятся в центре оного объекта, и они нам даром не нужны. Нам нужны вершины с 3-й по 102-ю.

Во 2-й вершине в 50-м кадре заканчивает движение «Camera01», а из 103-й, в том же кадре, начинает движение «Camera02». Итак выбираем первую из десяти созданных «однокадровых» камер, и перемещаем ее в вершину конуса, которая у нас под номером 3 числится.

Маленькое разъяснение для тех, кто пока не знает. Если включить режим редактирования вершин у нашего конуса, нажать кнопочку «Move» и выделить любую вершину, то внизу в трех окошечках появятся координаты выбранной вершины. Со следующими так же. Все «target» у всех камер естественно находятся в одной точке.

Настоятельно не рекомендуется повторять вышесказанное в темное время суток, именуемое «ночью». Можно очень сильно запутаться. Будет очень обидно.

Ну вот, камеры поставили, в сцену дымок запихнули, чаек и фитилек. Теперь полезли в «Video Post». А без него ни куда. С «Camera01» и «Camera02» надеемся все ясно.

С «однокадровыми» то же, понятно. «Camera01» и «Camera02» рендерят «tga», каждая в свой каталог, «однокадровые» в свой. В нашем случае это: «Cam 01», «Cam 02» и «Cam Move», находятся они на диске «С», в «корне».

В «After Effects» дважды кликаем в окне «Project», выбираем из каталога «Cam 01» первую таргу, ставим галочку напротив «Targa Sequence», потом открываем и на всякий случай игнорируем альфа-канал. Повторяем для следующих каталогов. Программа ругнется правда, «мол потеряла я, каюсь, кое какие «тарги». Но вы ей не верьте, ничего не потеряно. Все на месте.

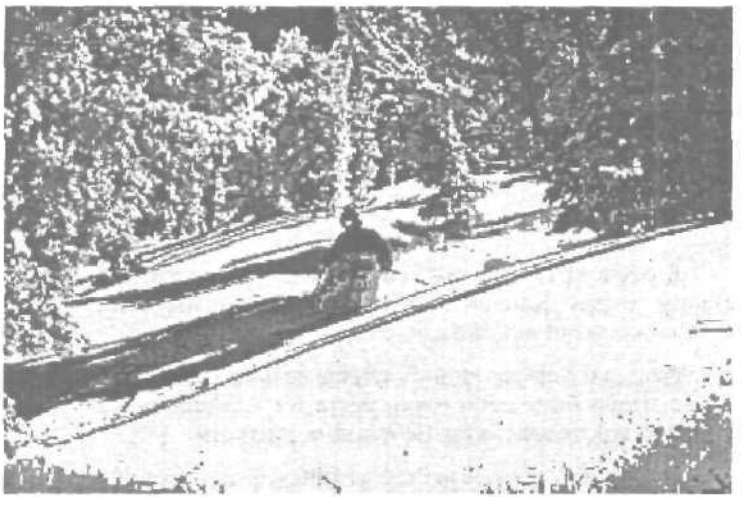
В «Adobe Premiere» собираем немножко по другому. Для начала лезем в «File» ➪ «Preferences» ➪ «General/Still Image» и ставим «Default Duration» — «1 frames». Затем в окошке «Project» кликаем правой кнопкой мыщи и выбираем «Import» ➪ «Folder» и таки да выбираем каталоги с таргами.

Глава 20.
Еще раз о 3D и фотографии

Итак...

Урок первый

Предположим, что у нас есть фотография зимнего пейзажа, такая как эта;



И почему бы нам не подшутить над знакомыми и не показать им на этой фотографин Снежного Человека.

В данной главе постараемся описать несложный способ создания следов на снегу. Сразу же сделаем небольшое отступление и скажем, что таким способом можно выполнить и другие задачи, например в интерьер комнаты (фото) повесить бра и высветить участок стены или вывести некоторые детали фотографии в объем.

Ну, что ж начнем. Для начала нам понадобится растровое изображение следов Снежного Человека. Можно поступить очень просто - воспользоваться стандартными инструментом Adobe Photoshop, который называется **Custom** Shape tool, где присутствуют эти самые следы и вот, что получилось:

Далее в MAX-е на виде сверху создаем плоскость **(Create ➪ Plane)** и ставим камеру. Камера должна максимально соответствовать участку земли, на который накладывается (ракурс, уклон и т.д).

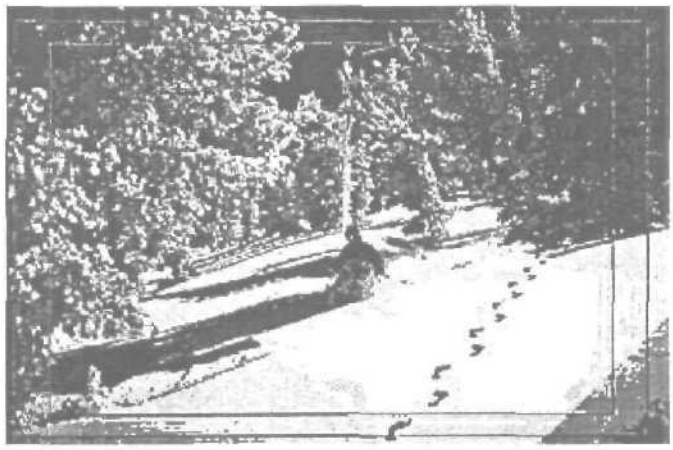
Размер и форма в данном случае не имеют решающего значения — главное, чтобы было достаточно места для наложения текстуры. Что касается текстур, то здесь есть небольшие тонкости.

Во-первых, в качестве карты Diffuse надо загрузить то же изображение, что и для фона, но только во вкладке **Coordinates** установить Environ **и Screen.**

Во-вторых, **Specular Level и Glossiness** в поле **Specular Highlights** настраиваемого материала должны быть установлены в ноль, иначе появятся засвеченные места на плоскости.

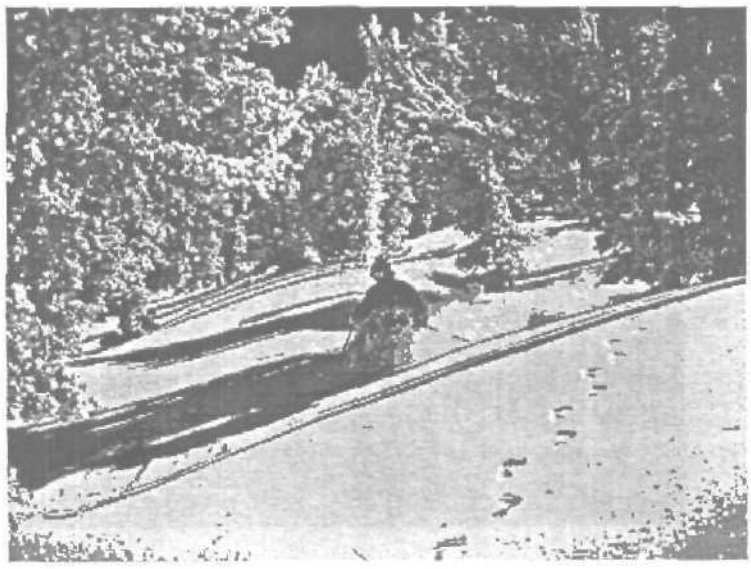
В качестве карты выдавливания *мы поместили следы и* в настройках немного увеличили размытие. Значение **Bump** составило 200, но в зависимости от фотографии, освещения и размера карты может сильно изменяться в ту или иную сторону.

И вот что у вас должно получиться:

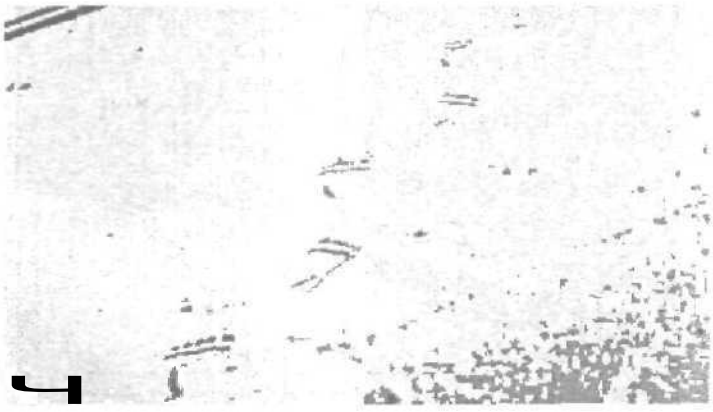


Далее надо установить в сцену светильники.

В данном случае использовано два **Omni:** один основной, второй заполняющий. Здесь надо учитывать не только освещение фотографии, но и настроить Multipler осветителей так, чтобы при рендере плоскость полностью «сливалась» с фоном. К сожалению в этой части все придется настроить методом подбора. И вот что в итоге должно получиться:



Посмотрим немного крупнее...

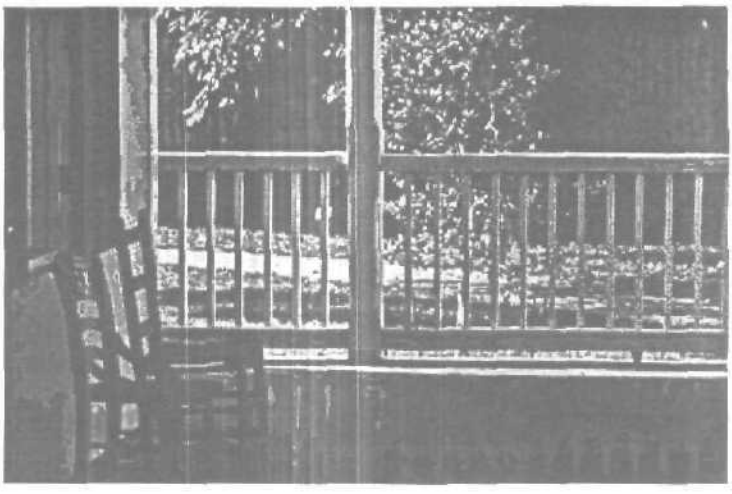


Пожалуй неплохо.

Урок второй

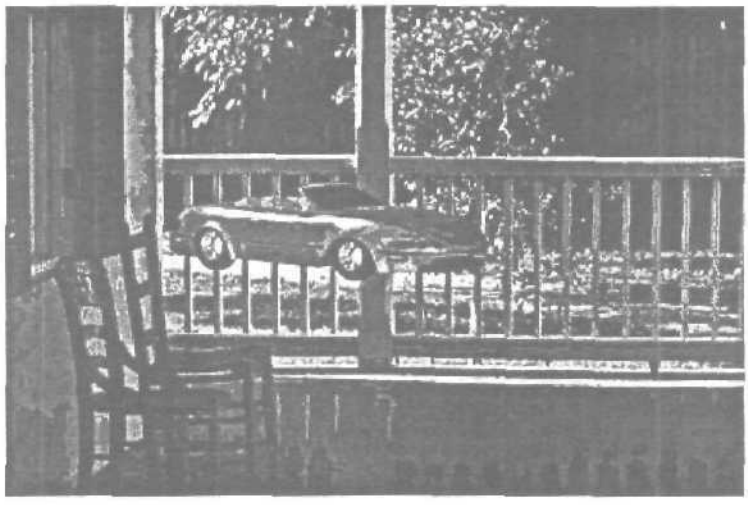
В этом уроке мы разберем способ маскирования объектов для того, чтобы они могли быть частично или полностью (если того требует создание анимации) перекрыты фотографией.

Возьмем для примера вот такую фотографию

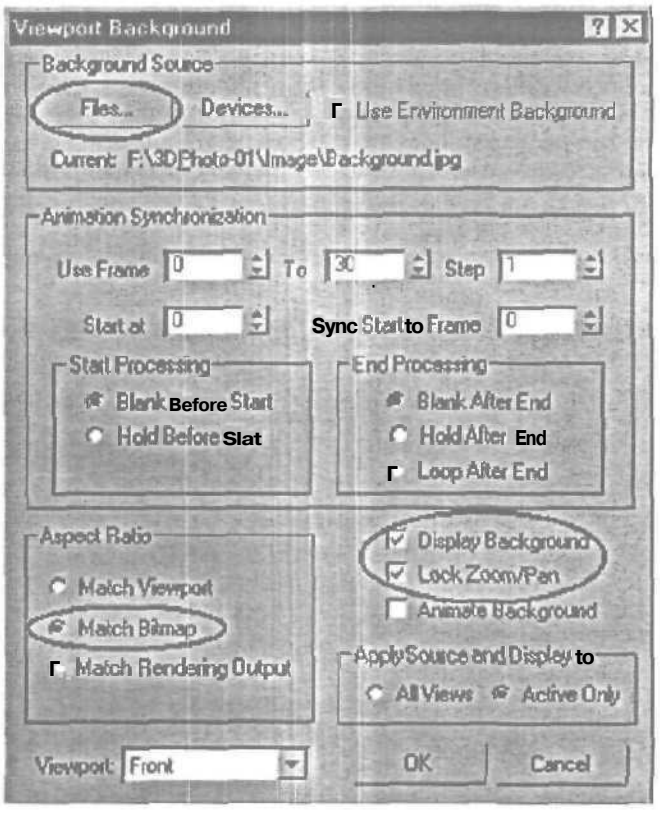


И поставим на задний план модель автомобиля.

Как установить фотографию в качестве фона, согласовать вид из камеры и фотографию, установить свет и т.д. описано в уроках упоминавшихся выше, поэтому приступим непосредственно к описанию маскирования. На изображении снизу вы видите, что получается, если просто отрендерить сцену с автомобилем — уменьшенная копил авто как бы зависла где-то над крыльцом, хотя все сделано правильно. Такое впечатление создается из-за того, что ограждения крыльца оказались за автомобилем и нет тени, которая так же помогает ориентироваться в пространстве.



Ну, что ж попробуем исправить положение. Для начала нам понадобиться построить плоскость **(Create** ➪ Plane), подвести ее под автомобиль и назначить материал Matter/Shadow с параметрами для принятия тени от автомобиля. Затем надо построить объект, который по форме будет соответствовать тем деталям, которые надо открыть на фото (пока они скрыты автомобилем). Сразу же отметим, что если это все делается для анимации, то объекты должны быть объемными копиями тех, которые на фотографии, иначе это не сработает при повороте или движении камеры. Все, о чем будет говориться ниже справедливо только для статических (т.е. неподвижных) сцен. Дальше есть два пути: если форма, которую надо построить не сложная, то ее можно строить прямо на виде из камеры, если же сложная, то надо фотографию в качестве фона вставлять так, как указано ниже.



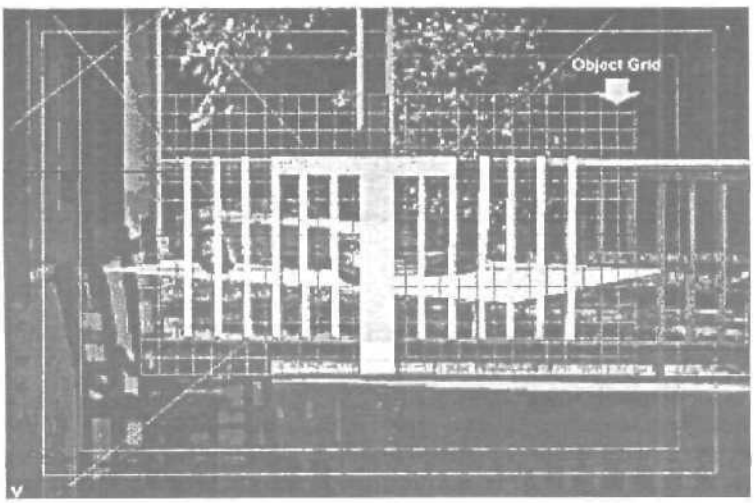
Этот способ постановки фотографии в качестве фона позволяет масштабировать объект вместе с фотографией и следовательно приблизить фотографию для проработки сложных участков. В качестве объекта маскирующего автомобиль были использованы сплайны (Create ➪ Shapes O Rectangle), после чего наложен Edit Mesh. Для того, чтобы ничего не мешало и не тормозило работу, можно скрыть все ненужные в данное время объекты (в нашем случае автомобиль, плоскость, камера и светильники). Для этого в командной панели Display при выделенном сплайне нажимаем кнопку Hide Unselected, а по окончании всех построений и выравниваний — Unhide All.

После создания геометрии ее надо выровнять относительно вида из камеры (до сих пор построение происходило во фронтальном виде). Для этого активизируйте вид из камеры и при выделенном сплайне выравниваем его относительно оси Z (т.е. перпендикулярно взгляду) (Tools

➪Aling to **View** ➪ **Aling** Z). Сейчас надо на виде сверху переместить сплайн так. чтобы он в окне перспективы занял свое место относительно фотографии, после чего с помощью инструмента **Select And Uniform** Scale масштабируем до нужных нам размеров.

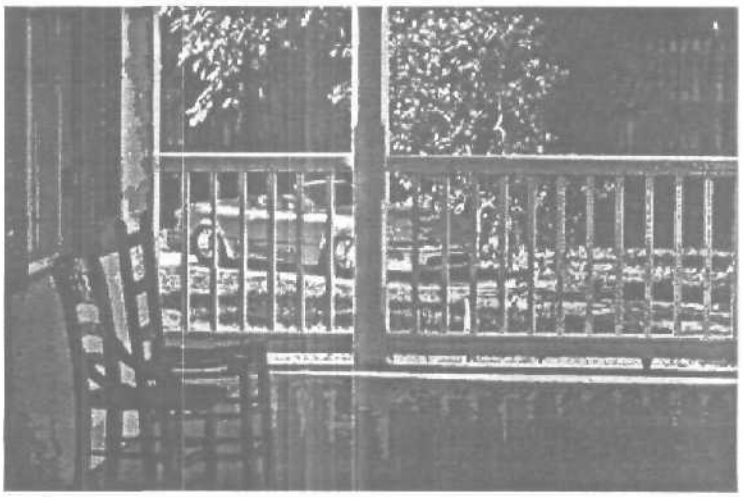
Второй способ несколько проще. Вначале строим объект сетки **(Create** ➪ **Helpers** ➪ Grid), затем при активном виде из камеры выравниваем ее перпендикулярно виду **(Views** ➪ **Grids** C Aling **Grid to** View), после чего надо поменять систему координат на **Screen.**

На виде сверху проверьте, не оказался ли у вас объект сетки позади камеры и если так, то передвиньте его на передний план. После этого, прямо здесь, в окне камеры, можно строить геометрию. Вот, что получилось:



После того, как все поставлено на свои места, можно приступить к наложению материала на созданный сплайн. Для этого нам понадобится Matte/Shadow материал вот с такими параметрами;

И вот окончательный рендеринг:



Глава 21.
Создание текстур в DEEP PAINT 3D

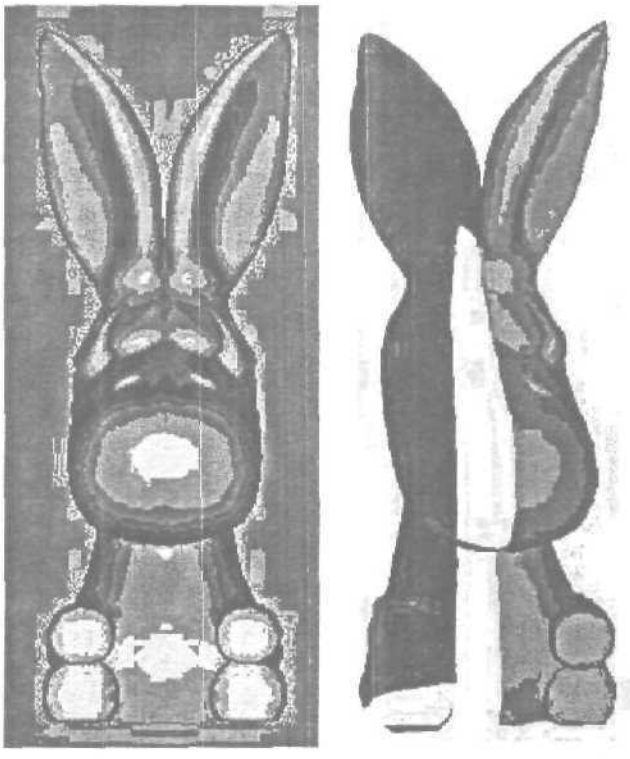
Перед тем, как начать описание технической стороны урока, хотелось бы сказать пару слов о самой программе DEEP PAINT 3D. Она позволяет значительно упростить процесс создания текстур, превращая его из достаточно утомительного в увлекательное занятие. Можно отметить три важные особенности: первое — это возможность интерактивно на-

кладывать UV Мар, т.е. изменяя положение осей вы в соседнем окне имеете возможность видеть как изменяется развернутый материал. Второе — это возможность рисовать прямо на модели и работать одновременно в MAX — Deep Paint — Photoshop. Ну а третье — это компактность текстурных карт, когда на одной графической карте можно развернуть сложную модель.

Итак приступим. Вот он, герой нашего сегодняшнего урока – прошу любить и жаловать.

Планируя накладывать текстуры в Deep Paint вы должны заранее, еще на стадии моделирования, позаботится о том, чтобы правильно «собрать» модель и подготовить ее для экспорта. Дело в том, что большинство моделей создается методом клонирования половинки модели относительно оси симметрии. В этой, как казалось бы простой операции, кроется один важный для Deep Paint нюанс. Дело в том, что 3D MAX воспринимает объект созданный методом Mirror так, как мы привыкли это видеть, т.е. с нормалями, повернутыми наружу, а Deep Paint «понимает» этот процесс по другому — с нормалями повернутыми внутрь, т.е. клонированный объект оказывается вывернутый наизнанку.

Посмотрите на изображения, расположенные внизу. На левом вы видите две половинки объекта в MAX-е, а на правой то же, но в Deep Paint.



В связи с этим в силу вступает первое правило: никогда не приатачивайте оригинальный объект к объекту клонированному методом Mirror. Делать нужно наоборот, выбрав оригинал, с помощью команды **Attach** присоединить вновь созданный объект. Таким образом все нормали примут положение как в оригинале, в противном случае у вас все нормали окажутся повернутыми внутрь. И даже если вы затем с помощью модификатора Normals **в** MAX-е или в Deep Paint поменяете положение нормалей на противоположное, все равно для этих двух программ останутся разночтения в положении нормалей, что в свою очередь не позволит наложить правильно текстуры. Второе, что нужно сделать перед экспортом, это добавить в стек модификаторов **Edit Mesh,** если ваш объект уже не является редактируемой сеткой. К сожалению MercatorUV, который предоставляет нам поразительную гибкость в наложении коорди-

натных карт, корректно работает только с такими объектами. Конечно, можно обойтись и без него, воспользовавшись максовским UVW Map, но при этом мы потеряем часть возможностей Deep Paint, кроме того, что стек модификаторов вырастет пропорционально количеству координатных карт и материал, который понадобится для объекта будет не стандартный, a Multi/Sub-Object.

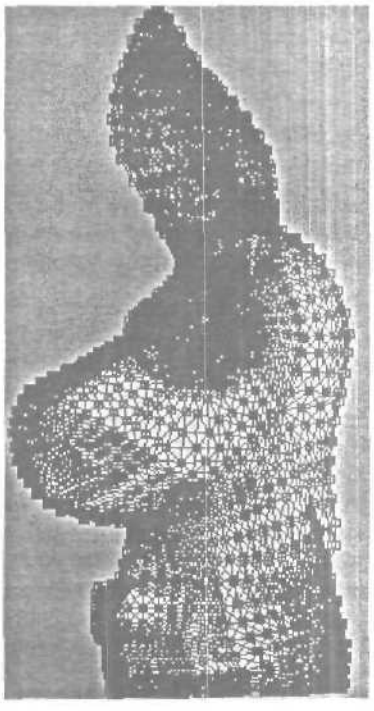
После того, как вы все это сделаете, нужно будет присвоить объекту материал (любой). На данном этапе это не важно, просто Deep Paint-у нужно знать, куда скидывать создаваемые карты. И только сейчас можно экспортировать объект в Deep Paint 3D. Для этого выделяем объект, идем во вкладку Utilities на командной панели и находим кнопку **Deep Paint 3D,** хотя, скорее всего, он будет скрываться за надписью **More.** Когда развернется панель нажимаем на кнопку **Paint и** попадаем в программу Deep Paint 3D.

Программа Deep Paint 3D довольно проста и интуитивно понятна. Так что для человека, знакомого с программами растровой графики не составит труда за вечер разобраться с ее работой. Л за сим продолжим.

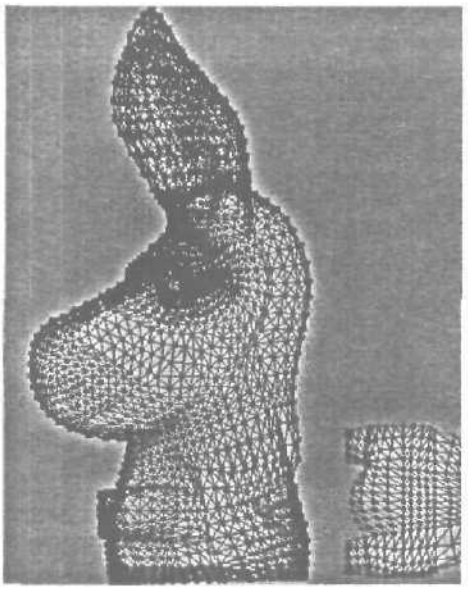
После экспорта объекта в Deep Paint 3D появится окно «**Material** Import» — здесь все просто, жмем OK и получаем свой объект в окне проекта. Далее идем в командную панель, выбираем вкладку **Элементы** и активизируем **Слои (Command Panel ➪ Elements (F7) ➪ Layers).** Здесь, если у вас не были заданы карты в MAX-е, будут пустые ячейки под буквами **C** (Color Channel), **B (Bump Channel), G (Glow** Channel), **S (Shininess Channel) и O (Opacity** Channel). Щелкаем дважды по пустой ячейке под буквой С, после этого появляется окно Add **New Channel,** выбираем там **Nothing и** попадаем в новое окно, в котором нам предстоит выбрать размер материала в пикселах. Здесь все просто: если модель простая, то можете отставить значения, предлагаемые по умолчанию или ниже, если же сложная, то не менее 1000x1000 (конечно если у вас не совсем старый компьютер). Так же можно задать каналы для выдавливания и если нужно то и для свечения, сияния и прозрачности.

Следующим шагом будет назначение объекту плоских проекционных координат. Для этого нажмите на кнопку Map в правой части **Главнон панели** (вверху), это вызовет окно MercatorUV. Именно здесь нам предстоит разместить на текстурной карте расправленную проекцию сетчатой оболочки. В зависимости от объекта вам предстоит выбрать на какие части и какие проекционные координаты накладывать. В данном случае предположим, что нам понадобится планарная карта для низа модели, цилиндрическая для тела, сферическая для головы и гривы и планарная для ушей. Конечно, в зависимости от задачи их может быть больше либо меньше. Первым делом жмем на кнопку Select All в правой части

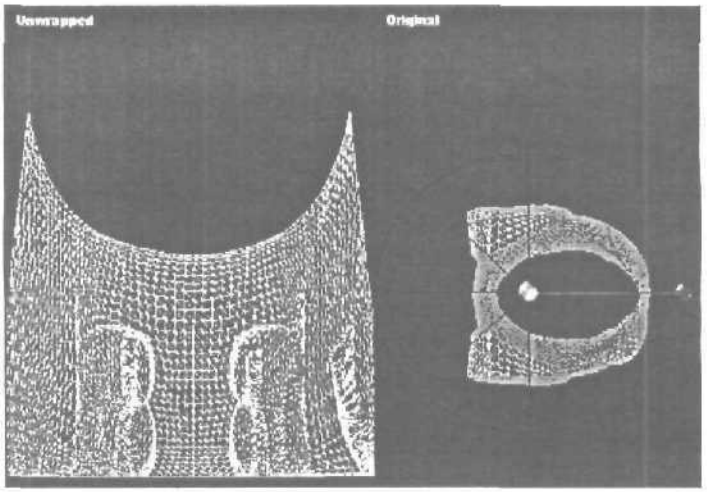
окна MercatorUV, затем на кнопку Map, а в появившемся окне на **Interactive Planar,** после чего появится еще одно окно — **Interactive Mapper.** Там **в** выпадающем меню Aling **View** выбираем **Front** и жмем OK. В итоге **в MercatorUV** мы получим плоскую проекцию сетчатой оболочки, а для того, чтобы она «вписалась» в размеры нашей текстурной карты нажимаем кнопку Pack и в появившемся окне оставляем активной лишь надпись **Equal Areas.** Результат всех этих манипуляций с кнопками и окнами вы можете видеть на изображении внизу.



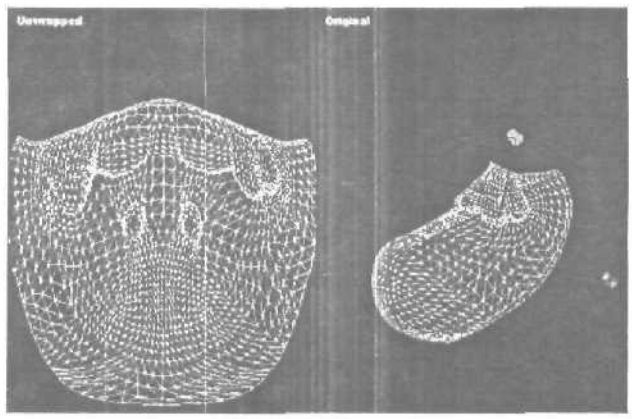
Сейчас, выбрав инструмент Лассо, выделяем самую нижнюю плоскость, активизировав точки (Points) в качестве подобъекта для выделения, после чего жмем на кнопку Map. выбираем планарную систему координат и в окне **Interactive Mapper** вид сверху **(Top).** После нажатия на OK выделенная плоскость примет вид как на рисунке снизу. Если у вас по какой-либо причине развертка нижней части ослика попадет на основное изображение, не беда, просто не снимая выделения с этой группы передвиньте ее с помощью инструмента **Move** на любое свободное место (можно даже за пределы карты).



Дальше повторяется та же последовательность действий, но только для тела. Единственное отличие состоит в том, что здесь применяется цилиндрическая система координат. Как разворачивается проекция для тела видно на изображении внизу.

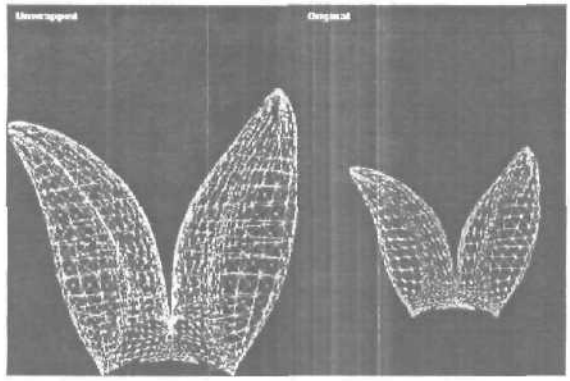


А вот так будет выглядеть голова ослика после применения сферической карты координат:



Обратите внимание на то, что поворачивая оси координат в правой части окна, одновременно меняется развертка, представленная в левой. Таким образом вы можете выбрать оптимальное решение для данной карты.

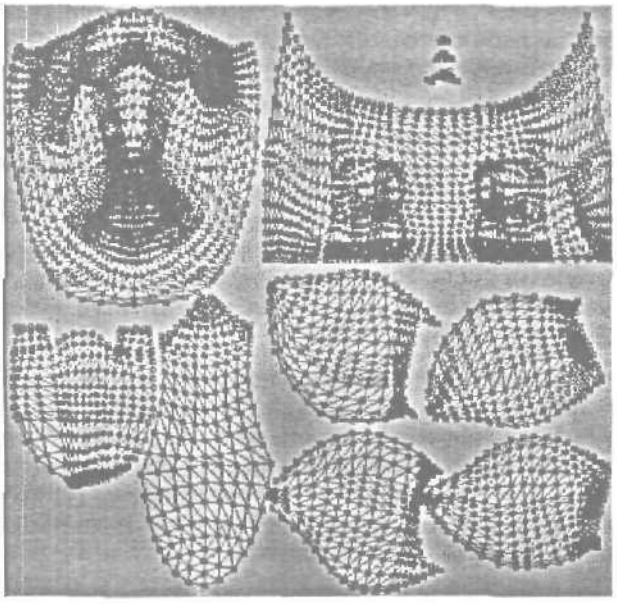
Такой же подход, как описан выше, будет применяться и для гривы ослика. А вот уши требуют несколько иного решения. Так как одно ухо скрыто другим, то мы не можем сразу выделить их по отдельности, но мы можем это сделать в два этапа. В первый раз, выбрав планарную систему координат поворачиваем их так, чтобы можно было выделять каждое ухо по отдельности.



Во второй, выделяем одно из ушей ослика и снова повторяем все операции с планарной системой координат, причем поворачиваем ухо так, чтобы плоская сторона уха «смотрела» на нас. После всего, что мы проделали у нас окажется проекция сетчатой оболочки уха. но только двухсторонняя (передняя и задняя часть уха в одной плоскости). Для того, что бы их разделить, надо нажать на кнопку **Lift** (излишне говорить, что ухо должно быть выделено), в результате чего передняя и задняя часть уха будут представлены по отдельности. Те же операции нужно повторить и для второго уха.

После того, как все части оказались развернутыми в плоские проекции можно заняться их упорядочением в рамках текстурной карты. Для этого выбираем кнопку **Pack,** оставляем значения по умолчанию и жмем **OK.** При желании можно вручную перераспределит все объекты на плоскости, для этого в правой части Mercator-а есть все необходимые инструменты,

Итак, вот что получилось:



Только после того, как плоские проекционные координаты назначены и развернуты в пределах текстурной карты, можно приступать непосредственно к самому текстурированию. Здесь все относительно просто: либо рисуем прямо по трехмерному объекту, либо в режиме про-

екции (т.е. когда объект представлен в виде плоскости). На данном этапе может быть несколько вариантов работы, предпочтительней делать наброски «цветовых пятен» (или мест, где будет применятся карта выдавливания) в Deep Paint 3D, а детальную проработку производить в Photoshop. Сразу же обратим ваше внимание на то, что проблемные места, такие как стыковка двух и более подобъектов сетчатой оболочки (к примеру головы и ушей) лучше текстурировать, используя режим проекции. Причем рисовать можно как на самом объекте, так и на карте цвета (либо любой другой), при этом изменения будут вноситься одновременно во все открытые окна. При текстурировании на картах для удобства можно включить отображение сетки, кликнув правой кнопкой мыши на изображении и выбрав **Display** Wireframe. Если же вы предпочитаете работать в Photoshop, то для экспорта нужно нажать кнопку **Export Material To Photoshop,** а для того что бы забрать карты из Photoshop на кнопку **Fetch the Material from Photoshop.** Для этой же операции можно воспользоваться и плагином, который при инсталляции ставится в Photoshop. Если вы экспортируете материалы в Photoshop в режиме проекции, то для рисования нужно использовать слои **Paint Layer.**

По поводу рисования на объекте нет никаких особых хитростей, но все-таки хотелось дать несколько небольших советов.

Начнем с того, что в Deep Paint достаточно много предустановленных материалов, и это может значительно облегчить вашу работу, причем без особого труда вы можете создавать свои собственные наборы и модифицировать уже существующие (найти их можно в **Командной панели** во вкладке **Presets).** Достаточно удобно при нажатой клавише **Ctrl,** интерактивно, при помощи мыши (или дигитайзера), изменять размер и прозрачность рабочей кисти. Ну, а раз зашла речь о дигитайзере, то надо сказать, что авторы программы постарались и сделали все, чтобы работать с дигитайзером было очень удобно.

Также хочется обратить ваше внимание на то, что на панели инструментов есть переключатели **(Color toggle, Bump toggle и** т.д.), которые позволяют включать (либо выключать) каналы, на которые будет наноситься изображение. Таким образом можно рисовать сразу на нескольких картах. Кроме всего прочего Deep Paint поддерживает работу с большим количеством слоев и позволяет импортировать и накладывать текстурные изображения (к примеру, фотографии).

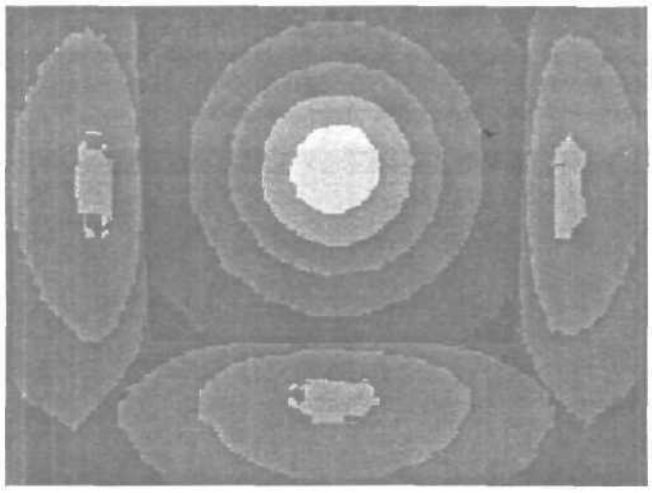
Ну вот мы и завершили процесс текстурирования, осталось только отправить материал обратно в MAX. Для этого существует кнопка **Send Material to 3D Application.** Кстати, если вы не все закончили, то можно сохраниться во внутреннем формате Deep Paint и затем продолжить. После экспорта в MAX в стеке модификаторов появится в верхней строке но-

вый модификатор Mercator UV, а в редакторе материалов, в присвоенном объекту материале появятся текстурные карты.

Глава 22.
Пара шагов к финальному рендеру

Сначала хотелось бы обратить внимание читателя на некоторые моменты, которые почему-то игнорируются многими (начинающими) пользователями Макса, желающими получить хорошую картинку.

Первое, это бессмысленное пихание **Omni** в сцены.

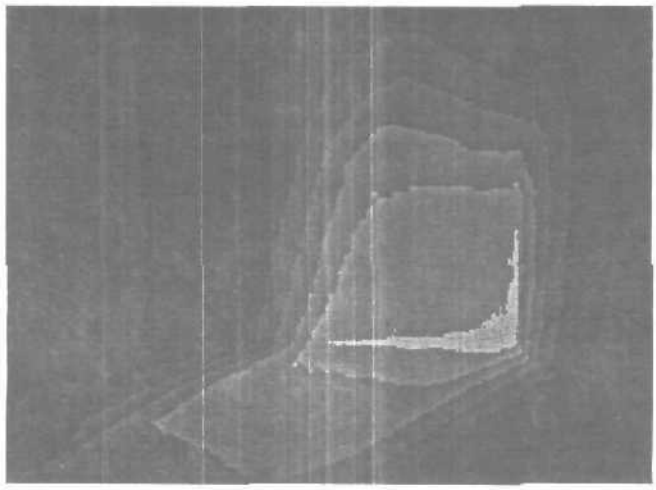


Знакомая картина? Висит Омни и делает черт знает что... На него смотрит юзер, и тоже не знает что делать... Юзер! Ты должен знать о законе Ламберта. Освещенность поверхности прямо пропорциональна синусу утла между поверхностью и лучом света. То есть, если луч светит в упор на поверхность, то получаемое освещение максимально, если же угол = 0 или отрицателен, поверхность не освещена вовсе.

Закон этот верен, но на практике мы сталкиваемся с двумя проблемами. Первая проблема состоит в синусоидальном законе. Поскольку он никак не контролируется, то при близких и «средних» расстояниях область поверхности, получающая максимальную освещенность мала и кругла, а большая часть поверхности остается освещена плохо. Вторая состоит в том, что сам Омни не имеет физических габаритов, это просто

математическая точка. Вместе в первой причиной это обстоятельство делает Omni не самым лучшим светильником. Первая проблема решается введением GI, поскольку в реальной жизни поверхности очень часто освещены в бОльшей мере отраженным светом, а не прямым, и как следствие, освещение того или иного объема пространства довольно равномерно.

А вторая проблема... вторая проблема не решается и не будет решаться никогда, потому что закон Ламберта неизменен. Отсюда правило — юзайте светильники адекватной формы и размера. Обратите внимание на Direct светильники — они ведь могут быть прямоугольниками любого размера, и не только квадратными... плюс еще можно карту положить, можно одним светильником заделать скрытый свет на потолке. Вообще, учитесь ставить свет по следующей схеме: сначала наиболее крупные и значимые светильники, потом добавление различных локальных подсветок/затемнений. Важно представлять и понимать себе размер-интенсивность-значимость каждого светильника в сцене.

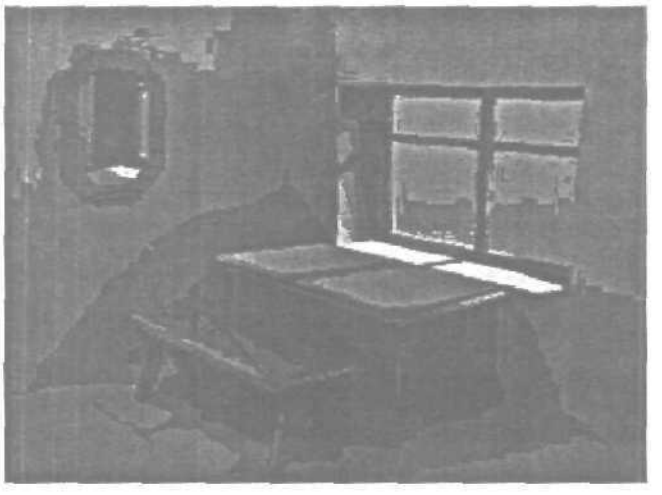


На рисунке сверху та же коробка, что и ранее, но с ONB шеидером.

Второй момент, на который хотелось бы обратить внимание, касается Final Render'a. Некоторые погоняют его с дефолтными настройками недельку и идут искать счастья дальше. Это не есть хорошо. На сегодня не так уж много быстрых рендереров, которые можно использовать в производстве, а не в целях самообразования или «поиграться». С Final Render можно работать, и именно та сотня параметров, которая в нем есть, и позволяет контролировать каждый аспект материала или света.

Сложно? Но можно! Возможно. VRay или Brazil обгонят Final Render. Вполне возможно. Но такой свободы и реализма, как в Final Render пока нигде нет. Изучение тонкостей этой системы даст вам полное понимание процесса, одинакового для *всех* современных систем рендеринга. Поэтому, к тем кто только приступает к изучению Final Render просьба: не думайте, что дефолтные установки есть самые быстрые или самые качественные. Иногда они просто нечто среднее между скоростью и качеством, иногда они просто выключены. Дефолтные настройки могут легко быть в два раза тормознее оптимальных.

Ну да хватит, начнем урок. Откроем сцену, к примеру, вот такую:



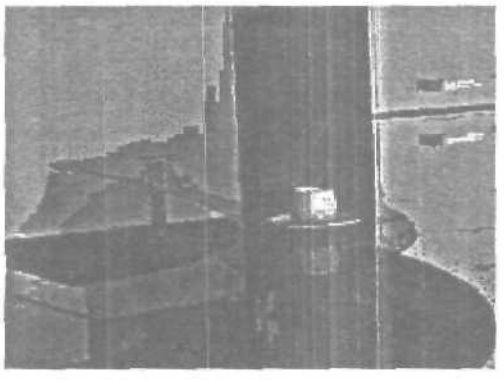
«Обычная сцена», — скажете вы. Ну да, так и есть. Фокус лишь в освещении. На самом деле одной стены в этой комнате нет, а в свойствах GI стоит галочка Consider **background**. А на задник повешен самый обыкновенный градиент (Gradient **map**). За стенами есть еще direct, который имитирует солнце. И вуаля, всего с одним светильником мы имеем более-менее приемлемое освещение. Довольно быстро и без затей.

Теперь обратите внимание на зеркало. Некоторые спрашивают, «как мне сделать нормальное зеркало». Прежде всего, вы должны знать, какого цвета зеркало. У зеркала нет своего цвета, оно лишь отражает. «Нет цвета», то есть **diffuse** = 0,0,0 RGB. Это единственное условие, далее начинаются тонкости. Первая тонкость состоит в том, что в Final Render поверхность начинает идеально отражать при **refl.Level** = 50% и цвете фильтра = 255, 255, 255 RGB. Поскольку же зеркала отражают свет не идеально, то обычно зеркалу нужно ставить **refl.level** =35..45%. И в зави-

симости от того, бронзовое зеркало или серебряное, надо немножко изменить цвет фильтра (Filter). Вуаля, зеркальце готово.

Практически то же самое и со стеклом. У стекла конечно свой цвет более заметен (скажем, зеленый), но его проще ввести не через **diffuse**, а пугем подкрашивания фильтра. Если же стекло не сильно прозрачно, то тут уже нужно пользовать **diffuse** color. Для всяких там стеклянных столиков и полочек очень круто смотрится стекло с **diffuse** = 0-0-0, **refl** = ~17%.

Когда стеклянная поверхность довольно большая, то конечно надо вспоминать правило Френеля, и либо класть карту **Falloff**, либо использовать кнопку Advanced **Fresnel**. Последний способ, правда, нельзя контролировать при помощи карты, а жаль. Ах ну да, и не забывайте, что у стекла IOR = 1,5...1,7.



Насчет бликов. Блик — это отраженный свет. Поскольку раньше у нас не было **area-light**'ов, то вместо них мы зачастую использовали **point-light**'ы (точечные источники). Это привадило к созданию ненатуральных бликов (для краткости, псевдо-бликов). Чтобы хоть как-то отобразить больший динамический диапазон сцены, для этих бликов приходилось выкручивать параметры за 100%. Более-менее приличный блик для стекла был **160/70 (Spec.level = 160, Glossiness = 65,** Blinn).

Более продвинутые пользователи имитировали реальные блики следующим образом: они создавали реальную геометрию в виде источника света, и на нее натягивали белый материал, у которого на **diffuse** наложена карта **Output.** В этой карте можно значительно увеличить **Output и RGB level** материала. Если, скажем, поставить этот параметр равным 5, то получившийся источник света будет в пять раз белее (ярче) обычного белого (255, 255, 255 **RGB).** Выставляя для бликующих поверхностей малые значения отражательной способности, типа 1...10%, на выходе полу-

чаем физически корректные блики. Что, надо заметить, смотрится довольно эффектно после всех этих бесконечных псевдо-бликов. Как вы вероятно догадались, данная техника является уже приближением к технике HDRI, поскольку несколько искусственно растягивает динамический диапазон сцены.

Заканчивая это несколько странное поучение, обратимся к читателю с просьбой не забывать о цвете света. Серый свет в природе редок, пожалуй это только когда в ясную погоду небо полностью закрыто облаками, именно это время ценят художники, потому что в эту пору можно рисовать пейзажи или этюды в их естественных цветах, без влияния прямого света или рефлексов. Эта область наименее освещена в литературе, и довольно сложна в освоении — но именно она вносит финальный штрих в реализм, или особый шарм, или особую изюминку. Именно потому что это сложная сфера, хотелось чтобы больше людей пытались освоить правильное применение цвета при расстановке света.

Глава 23. Как нарисовать книгу

Книжка... вроде бы саму книжку как таковую нарисовать не сложно. Главная сложность в отрисовке страничек (или создания их иллюзии). Сначала надо будет создать box с такими параметрами:

length=100
width=100
hight=10
length segs=1
width&hight segs=10

Теперь применим к нашему **box** Free Form — деформацию. Для этого переключаемся в режим **modify** и выбираем модификатор **FFD2x2x2.** 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 перемесите их вверх. Теперь выделите поочередно **control points** слева и справа и слегка поверните их.

Ну вот один лист книги у нас готов. Скопируем его и отобразим симметрично оси Y.

Теперь займемся обложкой. Нарисуйте сплайн.

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

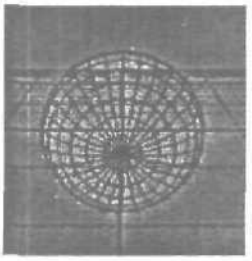
Ну вот книга и готова, по желанию к ней можно добавить новые элементы, уголки, подставку или еще что-либо.

Глава 24. Как сделать реалистичный взрыв

Первое что приходит на ум — использовать простейший **bomb,** но при использовании этого модификатора ничего хорошего и реалистичного обычно не получается. По этому 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 вот эту сферу.

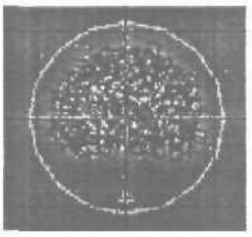


Расположим модификатор **bomb** ниже нашей сферы, зададим ему следующие параметры:

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

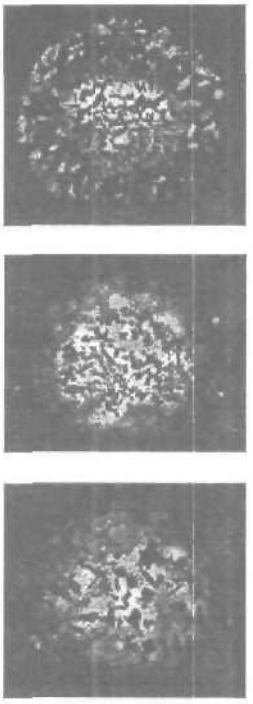
И свяжем его с объектом который хотим взорвать.

Можете проиграть полученную анимацию и заметить 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:



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 самом **combustion** меняем параметр **stretch на 1.0.**

Теперь можно попробовать отрендерить несколько ключевых кадров.



В случае если вам не понравился результат — вы всегда можете поменять параметры **combustion** и bomb.

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

Воспользуемся предыдущим примером, только **combustion** уберем. Теперь создадим **path gird,** которая будет служит объектом, на который будет накладываться секвенция. Создадим материал, где в качестве карты **diffuse и opacity** будут использоваться секвенции. Наложим полученный материал на наш **path gird, 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

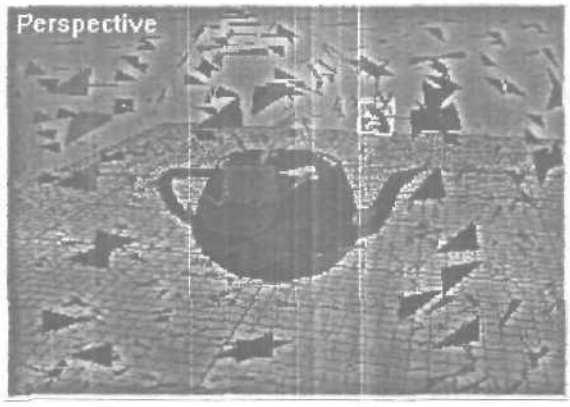
Часто используемым эффектом (особенно в космических сценах, анимациях) является взрыв. Для взрыва в max'е предусмотрен стандартный способ — Combustion и модификатор Bomb. Но это не самый хороший способ, и часто результат не устраивает — дело в том, что Bomb разделяет объект, в лучшем случае, на правильные формы групп face'ов (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 для всяких издевательств — стандартном max'овском чайнике.

Создаем Create ➪ Particle Systems O Particle Array. Располагаем этот Particle Array где-нибудь в сцене (не важно где). Выбираем объект, который будем взрывать (Pick Object). Ставим следующие установки:

- Viewport Display — Mesh
- Particle Type — Object Fragments

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 следующимиустановками:

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

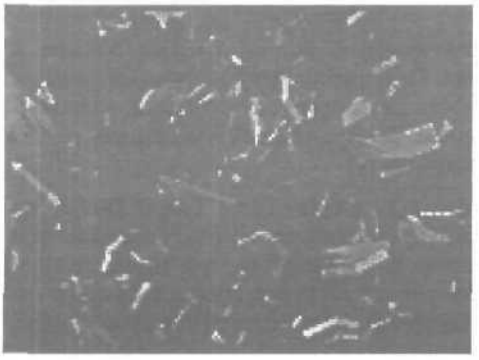
Particle Type: Object Fragment Controls: Number of Chunks — очень важный параметр, регулирует количество кусков-осколков, на которые разлетится чайник. Если надо, чтобы объект не разлетался на face'ы, то поставьте нужный параметр.

Particle Rotation: Spin Speed Controls: **Spin Time — еще** один важный параметр для увеличения реалистичности — контролирует вращение осколков. Если этот параметр не 0, то осколок будетделать 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.**



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 имеют толщину, то есть, считается, что чайник был полым и толстостенным.
- Осколки вращаются в полете.
- Осколки представляют собой стандартные particle'ы, и к ним можно применять любые space-warp'ы (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'a для обработки ситуаций ударения осколков о предметы — подумайте, как здорово будет выглядеть, 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.

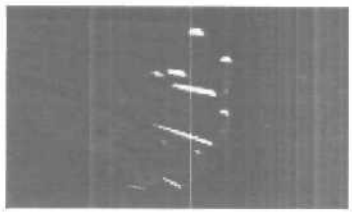
Глава 25. Как сделать параллельные лучи света

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

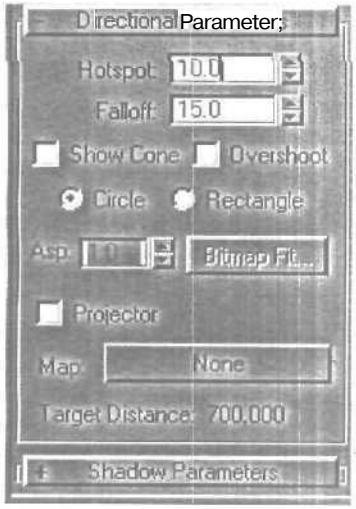
Способ №1

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).**



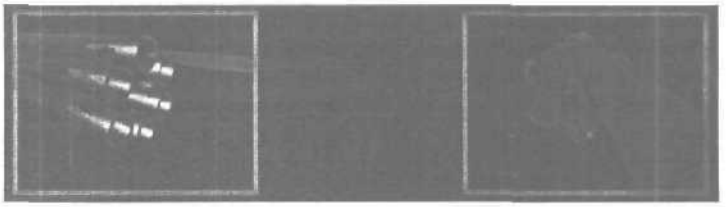
Если у вас изначально не было поставлено источников света, которые бы освещали сами стволы, поставьте их сейчас. Они больше не будут освещаться автоматически, так как вы добавили источник света.



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 26. Создание фонтана

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

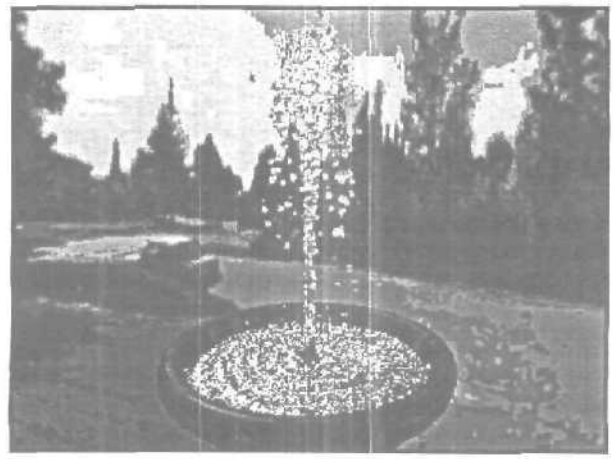
На вырывающийся из сопла поток воды действуют две основные силы — земное притяжение с одной стороны и центробежная сила с другой (последняя возникает при выходе воды под напором из круглой трубки). Форма является результатом взаимодействия капель воды — разлетаясь из сплошной струи, они рассыпаются на множество отдельных

брызг. В 3ds max есть все необходимые инструменты для имитации вышеперечисленных эффектов — SpaceWarps: **Gravity, Motor, PS: Blizzard** — система частиц с поддержкой клэй-технологии, построенной на «слипании» частей, подобно каплям жидкости в невесомости.

Итак, приступим! Создав в проекции **TOP Particle Blizzard,** перейдите во **FRONT** и поверните частицы на 180 градусов. Опять же 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. Далее для отображения в окнах можно выбрать **MESH (default ticks).** Но наберитесь терпения или, что еще лучше, измените **Emission c** 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 27.
Создание зеркальных поверхностей

Зеркальные поверхности — один из особых предметов гордости Autodesk и Kinetix — ведь именно их 3DS'ка смогла быть не только трехмерным 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, то есть, моделировать в них особенно нельзя.

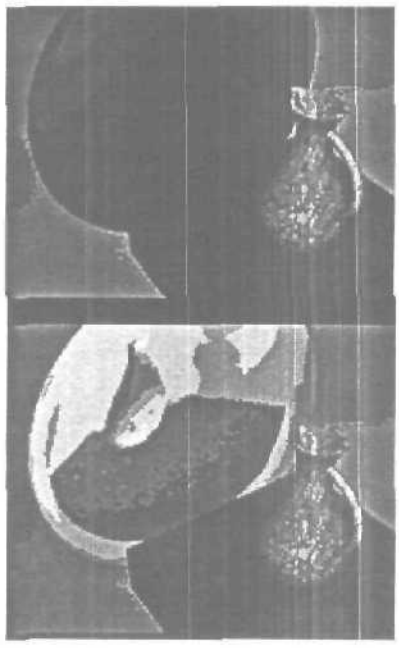
Мы рассмотрим четыре основных способа создания зеркал, по возрастающей сложности.

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.

Все — рендерите.



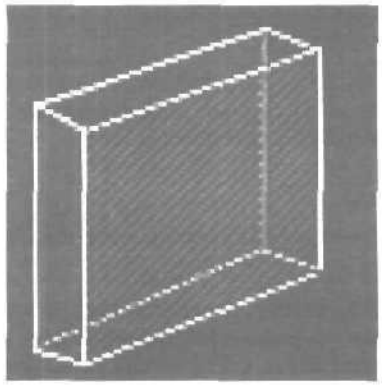
Быстрый и простой метод. Но это единственное его достоинство. Дальше идут одни недостатки. Во-первых — это очень плохое качество изображения. Даже на той сжатой из 640x480 до 200x150 картинке вы можете заметить ребристость границы — поверьте, она выглядит действительно отвратительно. Во-вторых, это качество 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.

Способ №2 — Flat Mirror

Этот способ применим только для идеально ровных поверхностей. Если у вас хотя бы на одну миллионную координата сдвинута — уже не сработает.

К тому же, способ не так прост, как №1.

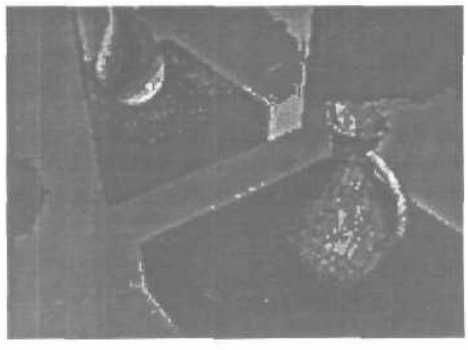
Итак, приступим. Создайте **Box** так, как показано на картинке.



Материал нам понадобится не простой, а многокомпонентный, а точнее двухкомпонентный. 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. 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). Теперь самое интересное. Берете свой Box, и добавляете модификатор **Edit Mesh** (можно также сконвертить 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. Все — мы готовы к рендерингу.



Хорошее качество изображения — пожалуй, почти идеальное. 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.

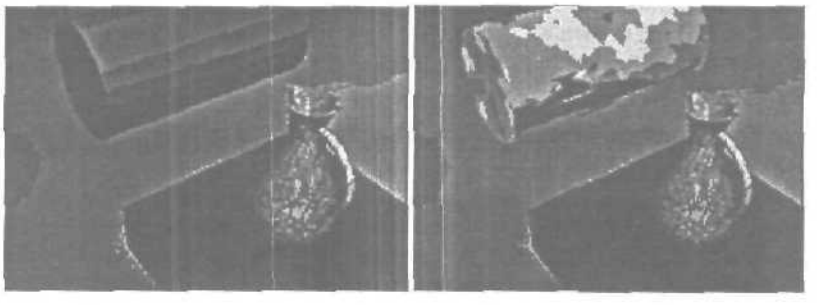
Способ №3 — Raytrace

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) 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 **тар** — **Raytrace** (процесс аналогичен заданию Reflect/Refract из №1). Присвойте материал объекту и ренедерите.

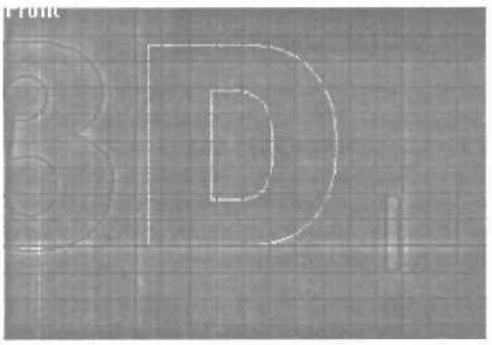


Идеальное качество изображений. Очень низкая скорость просчета — самая низкая из всех методов. Проблема в том, что встроенный рейтресер max'a очень и очень медленный по сравнению со своими аналогами.

Глава 28. Эффект растущего текста

Это очень простой эффект, его знают практически все бывалые max'еры. Построен он на том, что в 3ds max'е можно анимировать все — даже настройки деформаций лофта. Попробуем сейчас анимацию такого растущего из ничего лофта простого текста.

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

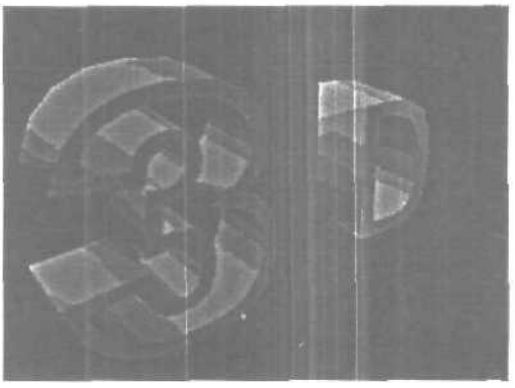


Но и тут уже кроется опасность! Буква *«D» на* самом деле состоит из двух сплайнов — внутреннего и внешнего! Для устранения этого недостатка буквы D, нам придется разделить ее на два разных сплайна. Выберите ее и во вкладке **Modify** примените модификатор **Edit** Spline **(Редактирование сплайна).** 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, которое мы будет двигать по пути-тексту. Пойдя по пути наименьшего сопротивления, создадим простой прямоугольник **(Create ➪ Spline ➪ Rectangle).**

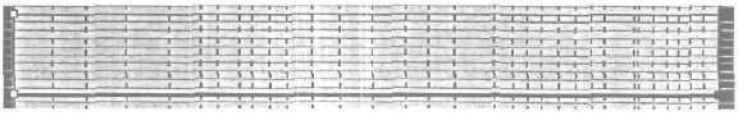
Все готово для лофтинга текста. Выберите один из сплайнов текста и сделайте **Create ➪ Lofting ➪ Loft.** Теперь нажмите **Get Shape** и выберите свое сечение — прямоугольник (если Get Shape не нажимается, то это означает, что вы где-то создали не одинарный сплайн, а двойной —

такую букву, как например «D» или «O»). Одна буква есть! Проделайте ту же операцию для каждого сплайна-пути и получите полный текст.

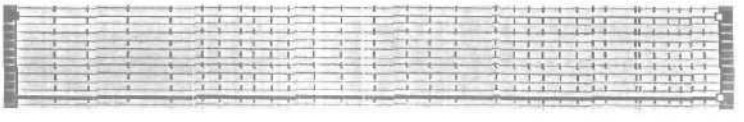


Осталось его проанимировать — это, в обшем, не так сложно. Мы будем анимировать деформацию масштабирования. Изначально (на первом кадре анимации) у нас не должно быть видно ничего — то есть, по всей длине пути лофт должен иметь масштабирование до 0%. В конечном кадре должно быть все на 100%. А так как нам нужно, чтобы буквы как бы 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 нажмите его. Теперь нажмите большую красную кнопку **Animate** на главном экране 3ds max 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, как показано на этом рисунке:



Все! Анимация готова! Можете отрендерить и всем показывать!

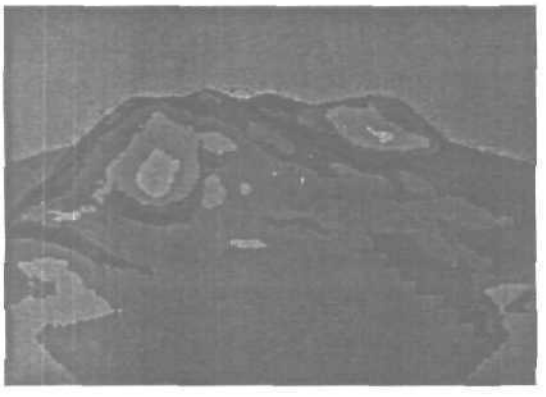
Глава 29.
Создание ландшафтов — Mountain Texturing

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 пикселов 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

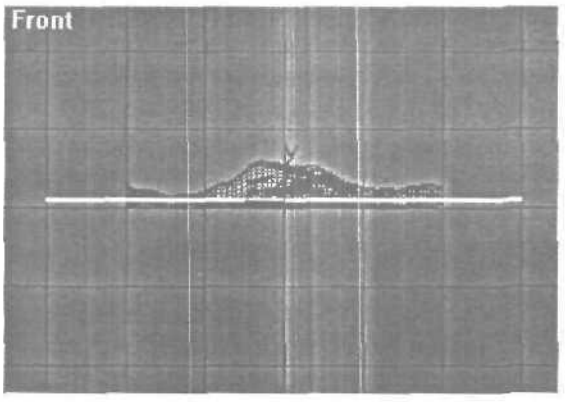
Можно не рисовать ничего, а просто создать этот самый box и хорошенько потаскать его за вершины, не забывая **Affect** Rigion держать включенным или вообще взять **patch** или **NURBS** и разгуляться всерьез.

Способ №3

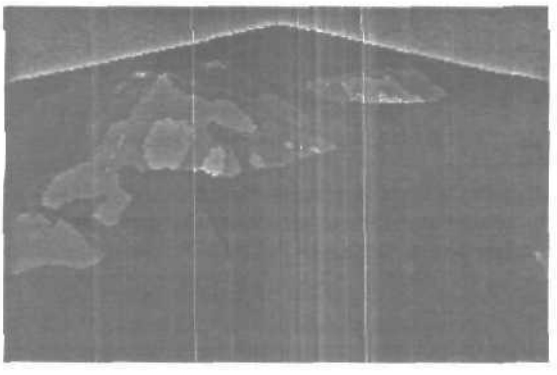
Если рисовать не получается или времени нет, в таком случае надо поискать плагин, который все делает сам, благо для max'а он есть и может быть даже не один. Плагин **Mountain** как раз **и** создает подходящую поверхность. Но вот, горы есть и сразу возникает вторая проблема, а как **их** собственно оттекстурить? Для первого способа всегда остается со-

зданная карта, которую можно ограниченно использовать как маску или blend-карту, чтобы получить более естественный вид гор. А если мы любим баловаться плагинами и таскать box'ы и patch'и за уши, это что же, круглый сирота получается?

Если гора не нравится, то всегда можно поиграться с Seed'ом (генератором случайных чисел в плагине), и заделать себе такую, чтобы по душе была.

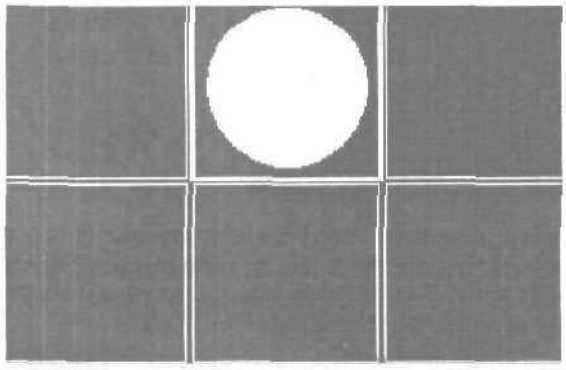


А теперь самое время создать немного дополнительной геометрии... В проекции Top создаем box, покрывающий нашу гору, как бык овцу.

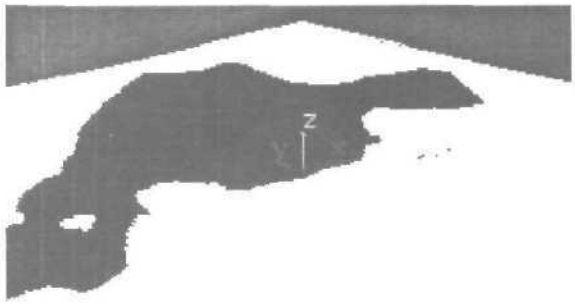


Сразу возникает вопрос, а зачем мы это делаем? А для того чтобы создать эти самые карты для масок, и blend'ов. Но перед основным шоу позвольте сделать еще несколько загадочных пасов руками. Создадим еще пару материалов...

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

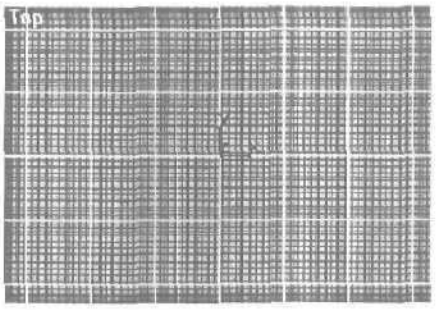


После этого присваиваем белый материал box'y, а черный — нашим скалистым горам.

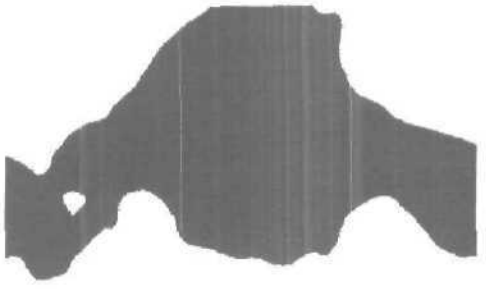


Переходим в вид Top, добиваемся с помощью zoom'a чтобы гора занимала весь вид по ширине. Зачем? Поскольку полученной картинкой в дальнейшем мы собираемся воспользоваться для наложения текстур на горы, то желательно чтобы стороны картинки были пропорциональны сторонам гор и если не О'фегулировать вид, то потом предстоит немного поработать в Photoshop'e, урезая картинку до нужного размера.

Рендерим в разрешении 300 на 300.

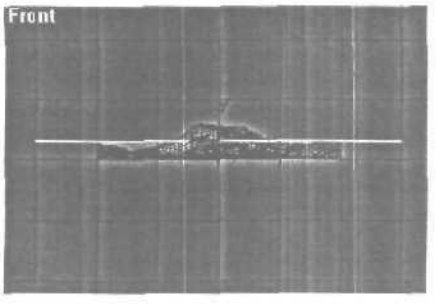


Спокойно! Это не шкура освежеванной коровы, а наша первая карта. Почему первая? Ну сами представьте, горы высокие, внизу травка растет, чуть повыше каменные стены до небес и на самом верху снежка напорошено.



Так вот, с полученной картой мы реализуем только травку и камни, а снежок-то и не выпадет, так что продолжаем...

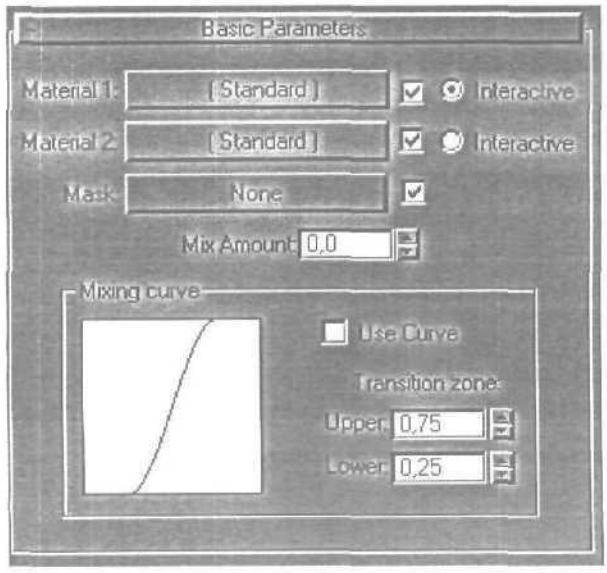
Перемещаем box повыше, не забывая, что часть горы, которая возвышается над box'ом будет в снегу.



Когда подходящий уровень найден снова напрягаем CPU рендером...

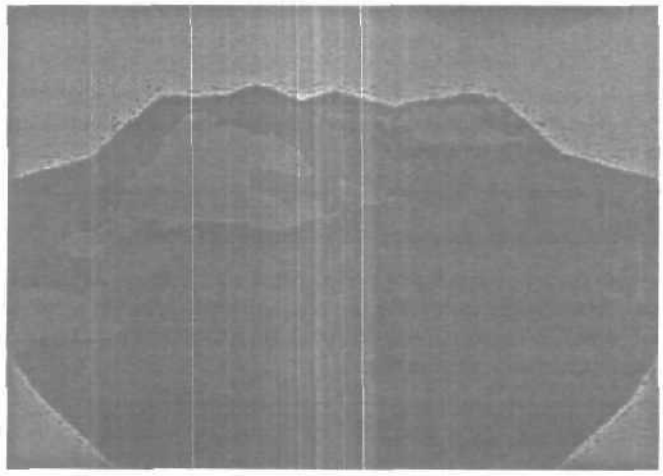
Теперь у нас есть все необходимое, чтобы сделать красивую текстуру. Идем в **Material Editor** и начинаем химичить. Сначала изменим тип материала со **Standart'a** на **Blend**.

Первую маску засовываем в слот **Mask** как **Bitmap**, **Material I** делаем красным, **Material** 2 синим. Позже мы, конечно, заменим это красно-синее безобразие на настоящие текстуры, но для наглядности так лучше. Накладываем материал, рендерим...

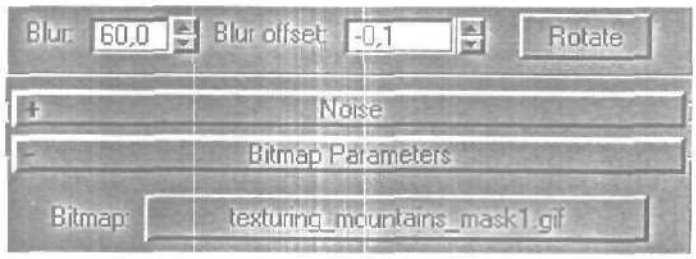


Сложно предположить, что 3ds max скажет, после того как вы нажмете на кнопочку Render. Во-первых, он может хмуро заявить, что на вашем Box'е, то есть — горе, нет mapping'а, тогда бегите скорее в панель Modify и наложите mapping с помощью UVW-Mapping модификатора. Во-вторых, текстура может неправильно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

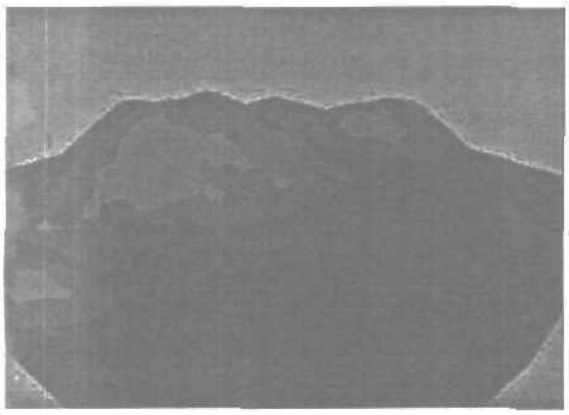
Резкая граница смотрится весьма неестественно, неплохо бы ее разблурить.



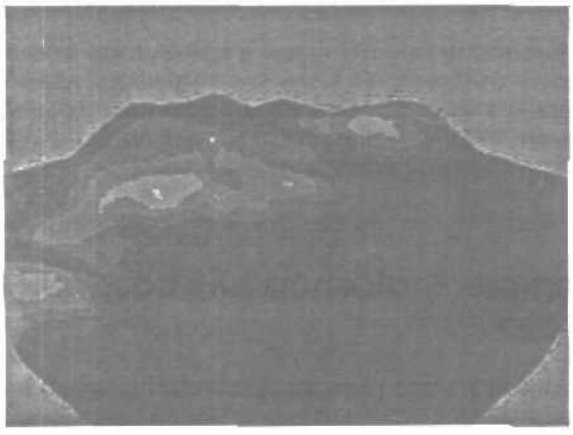
Не стоит делать этого в Photoshop'е. В 3ds max'е получится не хуже. Заходим в слот Mask и в параметрах Bitmap'а изменяем значение Blur.



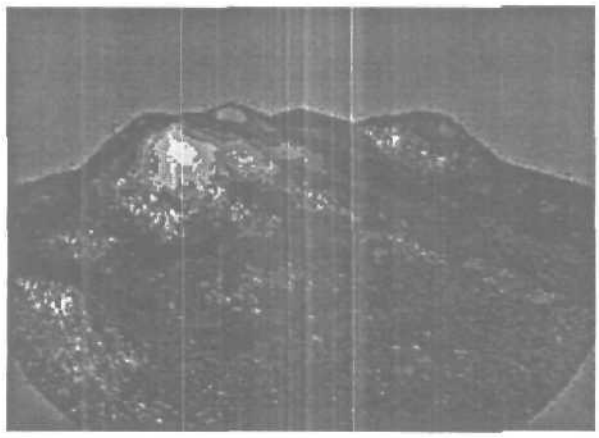
При значении 60 картинка заметно похорошела.



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, разблуриваем ее известным способом, в слоте **Material** 2 создаем материал под цвет крокодила Гены. Рендерим.



А теперь осталось только заменить все эти красно-сине-зеленые несуразности на нормальные материалы. Вместо синенького — травку, вместо зеленого — камень, а красное на снежок заменить.



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

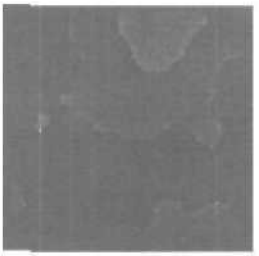
Вспомогательные материалы (у нас в примере это белый и черный) можно сделать светло-серым и темно-серым соответственно, или добавить какую-нибудь текстуру в слот **diffuse** и должным образом уменьшить интенсивность **self-illumination,** тогда материалы будут просвечивать сквозь друг друга в самых неожиданных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

Глава 30. Использование Photoshop для создания текстур

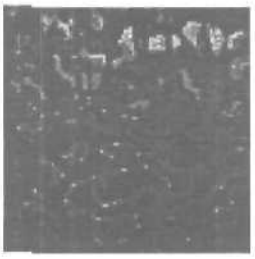
Работая с любой 3D-программой, часто встает вопрос о текстурировании объектов — в простейшем случае создании плоских изображений, которое бы проектировалось на нужный объект. Опишем несколько приемов, используемых для получения разных текстур в Photoshop.

Помятость (bump)

Откроем новый файл и наделаем в нем черно-белых облаков. Для этого reset-ьте цвета (кнопка «D») и сделайте **Filter ➪ Render ➪ Clouds.**

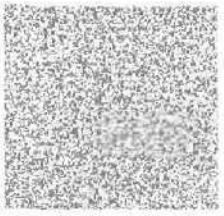


Теперь на полученное применим фильтр **Distort ➪ Ocean Ripple** со средним размером ряби и высокой силой. Увеличьте контраст изображения на 40-50 единиц (**Image ➪ Adjust ➪ Brightness-Contrast**) — и перед вами хорощая **bump-карта** для накладывания, например, для изображений помятости ткани или сморщенности.

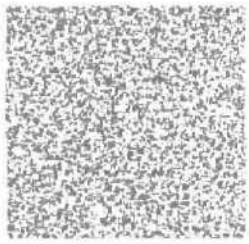


Расщелины (bump, diffuse)

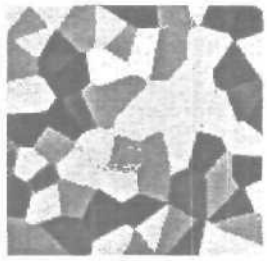
Откройте новый файл, скажем 150 на 150. Возьмите два первоначальных цвета (черный и белый — кнопка «D») и примените **Filter ➪ Noise ➪ Add Noise** с параметром Monochromatic и уровнем около 75.



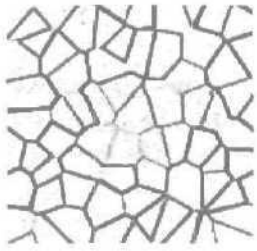
Теперь сделайте **Image ➪ Adjust ➪** Brightness-Contrast и увеличьте контраст где-то на +50 или +55. Этот параметр подбирается на глаз, так, чтобы общий суммарный цвет всех точек примерно был на 50% серым. После этого нужно еще немножко разблурить **(Filter ➪ Blur ➪ Blur)** картинку.



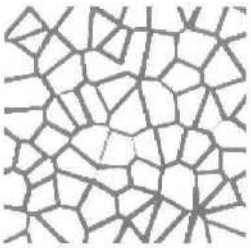
Все это теперь надо превратить в расщелины. Применим фильтр Crystalize со средним уровнем «кристаллизации» (например, 25).



Ну, что, начинают уже проглядываться расщелины? Осталось немного — еще один фильтр. На этот раз — **Filter ➪ Stylize ➪ Find Edges.**

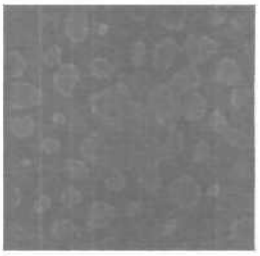


Ну вот практически и все. Осталось лишь сделать все расщелины более-менее равномерными. Сделайте Image ➪ Adjust ➪ **Levels.** Параметры такие: **Input** Levels — 194 1.00 255, а все остальное — оставьте, как есть. Все — можно использовать как bump или раскрасить текстурой, как **diffuse**. Одно «но» — 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).



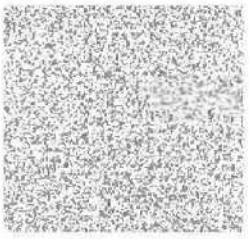
Пузырьки (bump, diffuse)

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

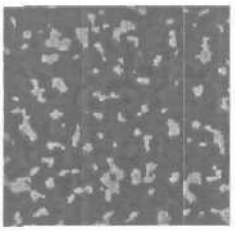


Каменистая поверхность (bump, diffuse)

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



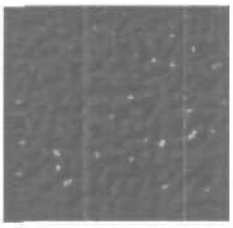
Теперь размытие — **Filter** ➪ **Blur** ➪ **Gaussian** Blur, коэффициент — 1-2.



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

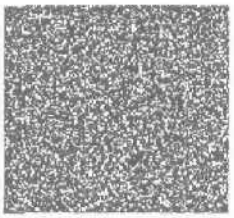
Height=3
Amount=100%

Ну и для диффуза — раскрасить.

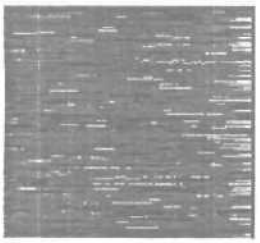


Дерево (diffuse)

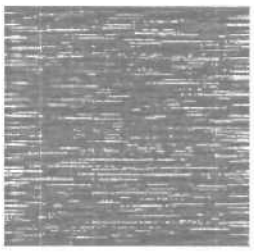
Создаем новую картинку, и, как всегда, добавляем шум — **Filter ➪ Noise ➪ Add Noise.** Лучше не монохроматичный.



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



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



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

Hue - 35
Saturation - 60
Lightness - +20

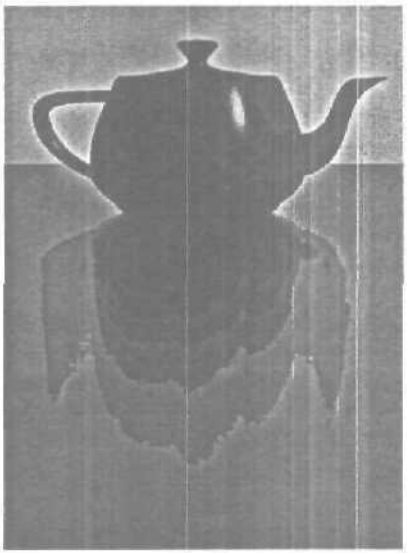
Получается что-то похожее на полированный паркет. Параметры дерева легко меняются, как параметры окраски. Осталось только отрезать края (там тоже получилась интересная текстура — текстура «расщепленного дерева» — если нужна — сохраните ее в отдельный файл), и затайлить все.

Глава 31.
Анизотропные отражения

Многие видели некоторые современные картинки, отредеренные в Brazil-e, многие также заметили тип шейдера в 3ds max называемого анизотропный (anisotropic), ну и 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 то же что и обычные отражения, 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.



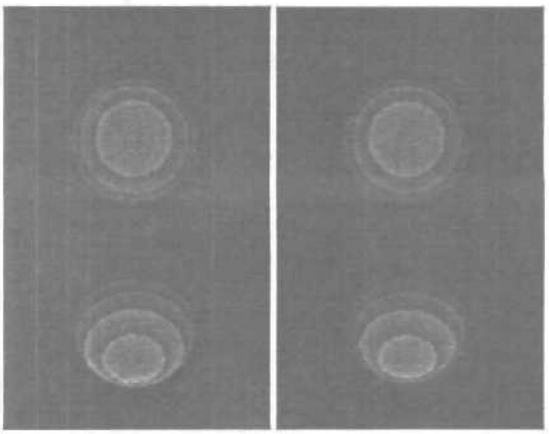
Типы объектов, имеющих анизотропные отражения

Все, что имеет мелкую зернистость, идущую, преимущественно, в одном направлении. Хороший пример — волосы; металлы, обработанные щеткой, так называемые шероховатые металлы (brushed metals), или

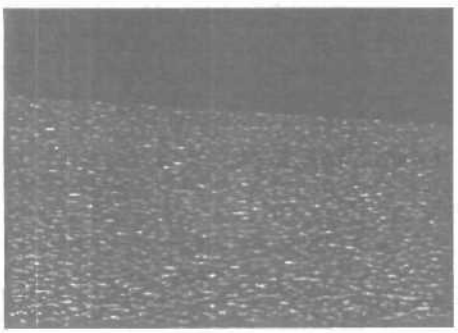
отражения на взволнованной воде (например, капающим дождем). Компакт-диски, тоже анизотропные, отражение искажается мелкими канавками, идущими по кругу вдоль CD.

Причина анизотропного отражения

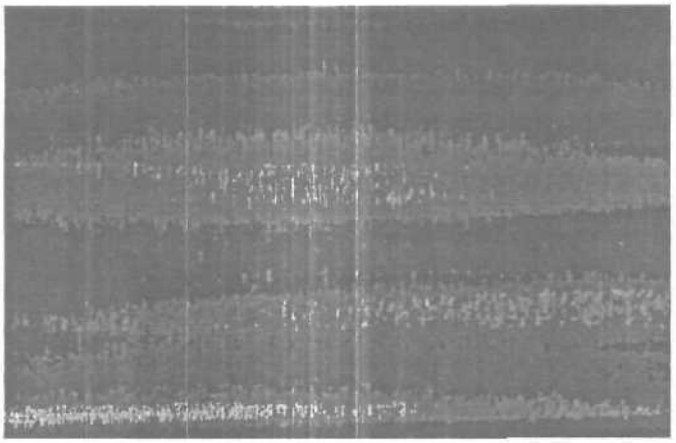
Мелкие неровности на поверхности делают отражения равномерно размытыми (если неровности умеренно неравномерны). Если эти мелкие выпуклости образуют «структурный узор», тогда «узор» может влиять на то, как выглядит отражение. Вот две сферы, первая с четким отражением, вторая с размытым.

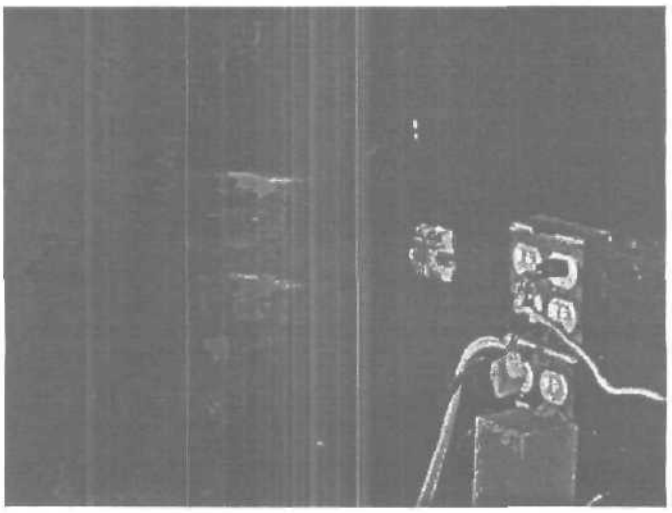


Если бы это было фото «реального мира», размытое отражение было бы вызвано мельчайшими хаотическими неровностями на поверхности пола, неровности, возможно, настолько малы, что вы можете их не заметить, но вы заметите, как они влияют на отражение.



Возьмем эти два примера реальных шероховатых металлов.

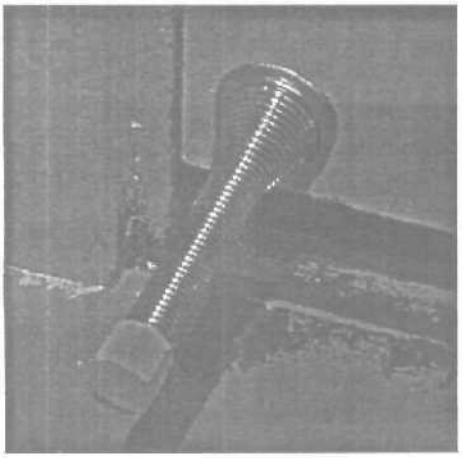




Первый рисунок — это приближенная рельса, пример, где вы можете увидеть некоторую мелкую зернистость, идущую в одном направлении. Второй рисунок — это фото дверцы холодильника, поверхность которой изготовлена из похожего материала, и сделано это фото с большего расстояния. Обратите внимание, как отражения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

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

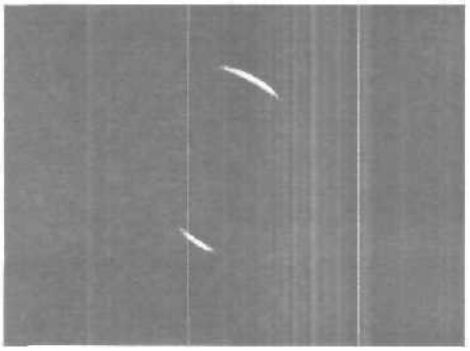
Вот, например, источник света и его отражение на дверной пружине.



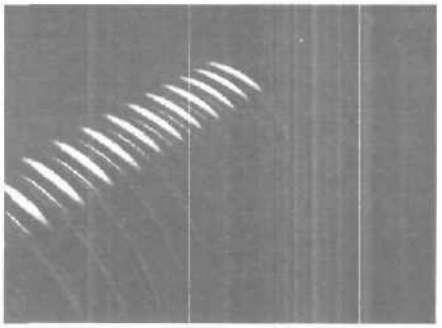
Если мы посмотрим на дверную пружину ближе, мы можем представить ее как набор кругообразных поверхностей, каждая из которых

отражает свет, исходящий сверху. Если бы было только одно кольцо, вы бы видели единственное отражение, по 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.

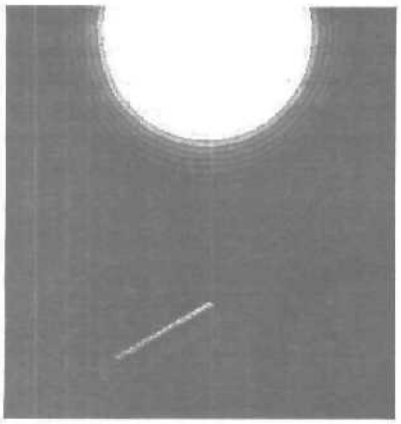
Вот пример этого же феномена смоделированного компьютерным способом, это одно кольцо, отражающее источник света, расположенный сверху.



И несколько колец, выстроенных в ряд.

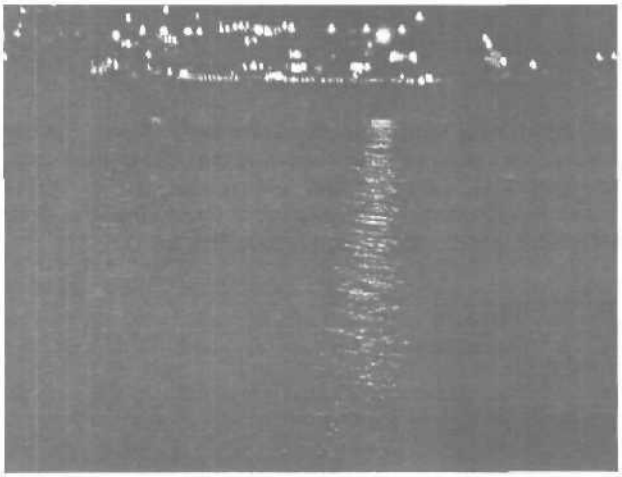


А вот этот же пример, но с расстояния подальше.



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

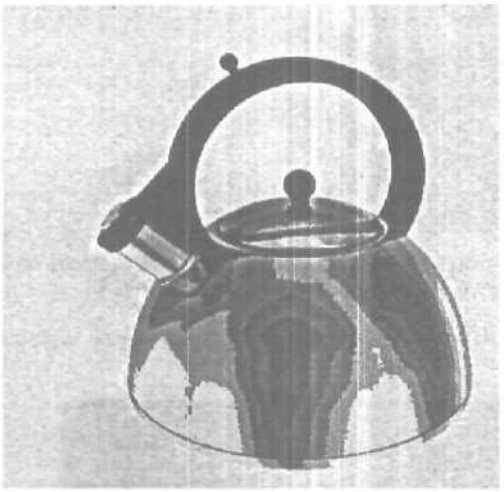
Другой пример для рассмотрения — это отражения на воде в ночное время.



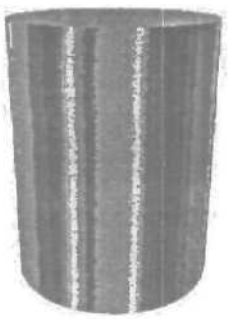
Простой круглый источник света отражается как длинный вытянутый световой блик, из-за ряби на воде, рябь, которая выполняет функцию похожую на функцию спиралей дверной пружины, или канавок металла с шероховатостями, только в увеличенном масштабе.

Анизотропные отражения или мелкие неровности

Поскольку анизотропные отражения представляют собой лишь отражения, искаженные мелкими выпуклостями на поверхности, почему бы просто не сделать эти мелкие выпуклости на смоделированной нами поверхности? Ну, конечно, 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 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

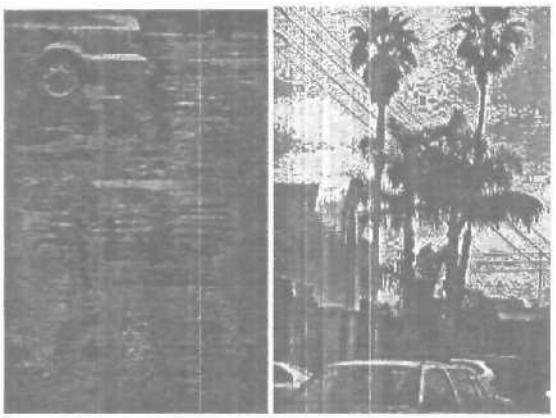
Первый снимок, это несколько машин, остановившихся на светофоре, заметьте, как свет их фар размыт в воде, которая, в свою очередь, искажается капающим дождем, а также неровной поверхностью дороги.

Заметьте, также как в итоге прекращается каждое отражение. Как указывалось ранее, говоря о дверной пружине, 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. Величи-

на «беспорядочности» на поверхности, так же как и угол вашего обзора, и яркость отражаемого объекта, все это влияет на то, как выглядит отражение.

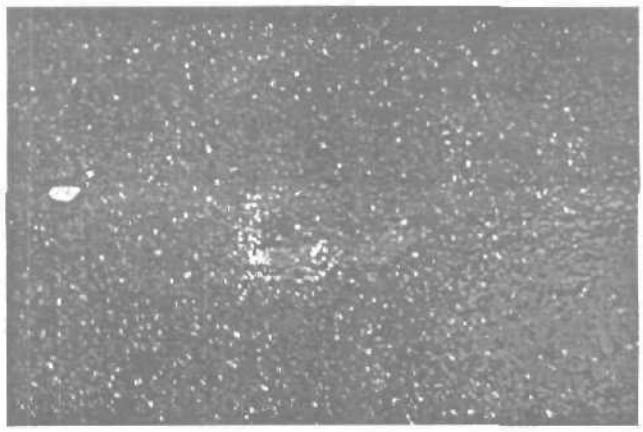
Вот свет стоп-фар.

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

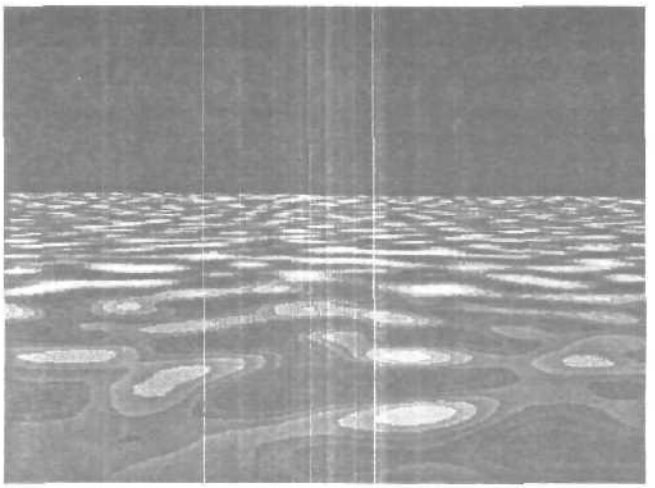


Рассмотрите эти две фотографии. Если вы исследуете их внимательно, заметите, чем дальше вы от отражения, тем больше оно размыто. Вот свет фар машины:

А вот фото, сделанное при приближении к отражению:

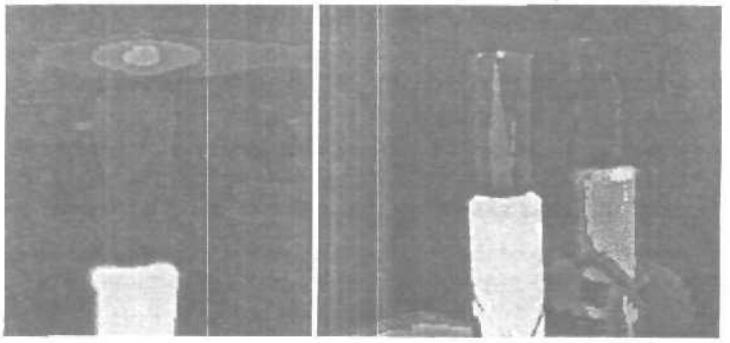


Заметьте, что отражение размыто неровностями поверхности дороги, но эффект анизотропии исчез. Эффект анизотропии изменяется в зависимости от расстояния, а также угла, например, рассмотрите этот шум (noise), назначенный плоскости:



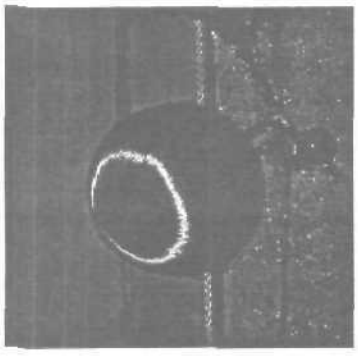
Это тот же однородный нойз на поверхности плоскости. И обратите внимание, ближний нойз выглядит больше, и по мере удаления он становится меньше и узор становится более «плотным». Так и отражения будут искажаться по-разному, отдаляясь, отражения переходят в неровности (bump pattern).

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 же 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 похож на подопечный анизотропный 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.



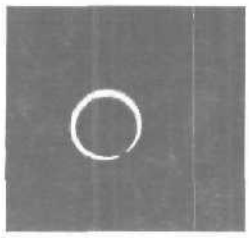
Итак, сфера с анизотропным бликом должна выглядеть примерно также, однако в 3ds max не так.

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.

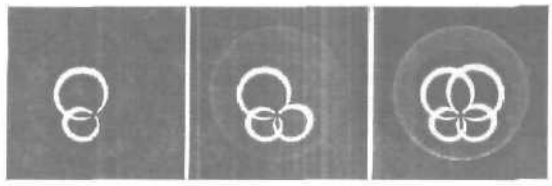
Но это подходит не для всех ситуаций. Например, для верхней части цилиндра, нет способов установить как вам нужно световой блик для выпуклостей, идущих по круговому узору (circular pattern), это означает,

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.

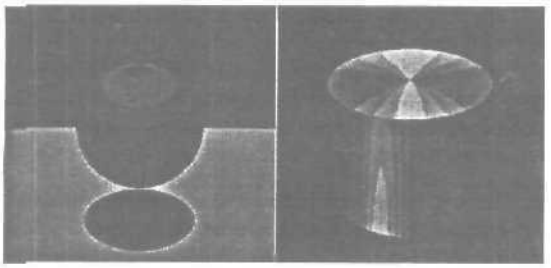


Однако, так как он предназначен для волос, а не для геометрии, у него существует компонента «просвечивания», «опустошающая» свет. Например, добавление большего количества источников света в вашу сцену начнет все ярче высвечивать рассеивающую составляющую (diffuse component)...



Решив проблему с рассеиванием (diffuse), отражающий блик (specular highlight) все же отбрасывает яркий свет. Например, если установить свет совсем за сферой, и назначить hair shader, то все еще получаем отражающий блик на поверхности, даже если свет на нее не направляется. Новая версия hair shader-a 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 с помощью Animal Logic's Maxman translater. Может получиться неплохой блик, даже на вершине (середины) 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.

Глава 32.
Создание 3D-живности

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

Вообще-то 3ds max предоставляет возможность использовать для анимации живых персонажей «bones», но возиться с расстановкой «костей» и IK-solver'ов 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.

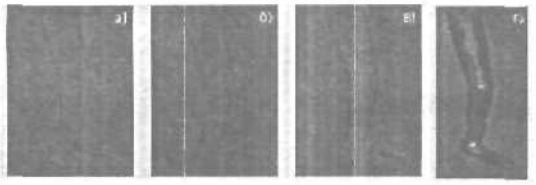
Чтобы проанимировать персонажа, его сначала нужно создать. Этим и займемся. Создадим небольшого человека, а затем — короткую анимацию с ним в главной роли.

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.



Теперь при помощи **Move, Rotate и Uniform scale** размещаем полукруги по всей длине ноги. Если полукругов не хватает, создаем копии.

Последний полукруг, 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 полукруг и на панели **Modify** ищем кнопочку **Attach.** Кликаем сначала на ней, затем последовательно на всех элементах нашего каркаса. Не забудьте и о практически невидимом последнем полукруге!

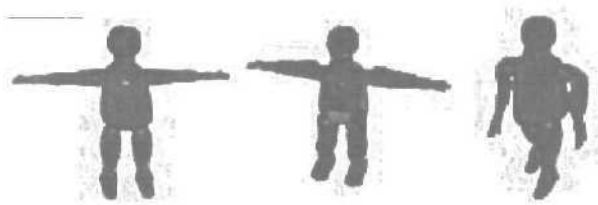
Дальше все просто. К полученному большому сплайну применяем модификатор **CrossSection**, в параметрах которого устанавливаем **Smooth** вместо **Linear.** Применяем модификатор **Surface** и получаем «обтянутый» каркас.

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).

Далее 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 сметчика 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.



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 вы-

шеупомянутый 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 подгоню: 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.**

Подсказка: если персонаж толстоват, а его руки при ходьбе задевают туловище — вернитесь в Figure mode и расширьте плечи персонажа. Если вы разместили «кости» правильно, то всяческих «отстающих» от тела точек быть не должно. Если же таковые появились (особенно часто подобное случается в районе пальцев), существует два пути для исправления этого безобразия. 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** параметрах (во вкладке **Modify)** увеличить параметры Radial scale, **parent и child overlap.** Можно рендерить все в AVI'шку.

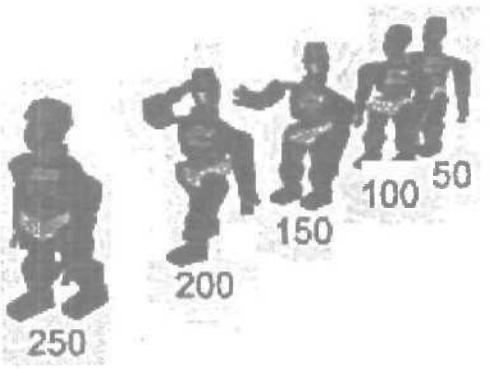
Подведем итоги. Мы создали простую анимацию, базирующуюся на шагах. Теперь давайте ее слегка усложним. Для начала перейдем в **Footstep mode** и, предварительно выделив все шаги, удалим их одним кликом по кнопке **Delete footsteps.**

Кликаем **Create multiple footsteps** и в поле **Number of footsteps** вводим какое-нибудь крупное число, например «15». Дальше, 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, а что дальше? Допустим, пусть на определенном этапе ходьбы он повернет голову налево и отсалютует. Пацифисты могут заставить персонажа просто почесать голову.

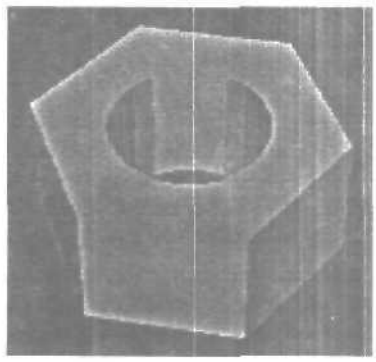
Приступим к анимации. Прокрутите Time slider примерно до того момента, когда скелет (вы же спрятали **Mesh,** не так ли?) будет в районе пятого footstep'a — допустим, кадр 84. Выделите голову скелета (как-то не хочется называть это черепом). По временной шкале видим, что keyframe'ы у головы присутствуют в момент каждого шага. Давайте при помощи того же Time slider'a посмотрим, где бы нам желательно закончить поворот головы, то есть повернуть ее назад. Кадр 174, записали? Выделяем и удаляем все keyframe'bi 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.

Между прочим, к пустой голове (я имею в виду голову, не покрытую головным убором) руку не прикладывают. Чтобы сделать человечку кепку/фуражку/колпак, создайте этот объект и просто «прилинкуйте» (select and link) его к голове (к Mesh'у линковать бесполезно), А дальше 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.



Глава 33. Создание правильной гайки с помощью лофта Rhino

Этот туториал совсем не сложен. Он скорее направлен лаже не на получение результата, а на получение некоего убеждения в мощности инструмента Rhino3D, имя которому Loft. Итак, мы будем рисовать гайку. Все и так умеют, да? Случайно не экструднутый сплайн, так как на рисунке?

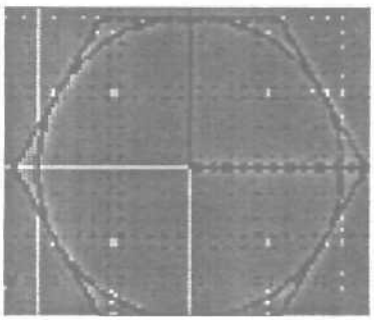


Теперь идите, отыщите где-нибудь гайку и посмотрите на нес. Что, похоже? Не очень, правда? Ладно, приступим к рисованию правильной гайки.

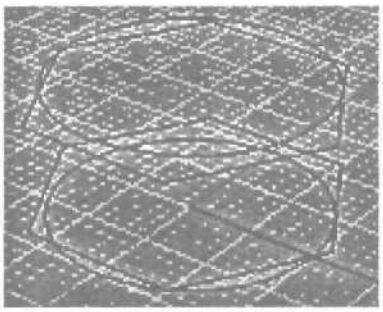
Настоящая гайка — это лофт двух вещей — окружностей и описанных вокруг них шестиугольников. Начнем с окружности. С помощью простейшего инструмента Circle нарисуем окружность. Теперь выберем инструмент Polygon и введем следующие коррекции с клавиатуры:

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

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



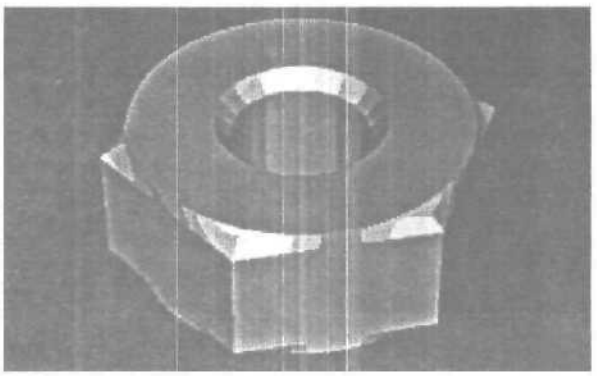
Теперь надо откопировать (Copy) оба объекта и разместить их друг над другом, так, как показано на рисунке.



Теперь все готово к лофту.

Если хочется, то 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. Ну как? Больше похоже?



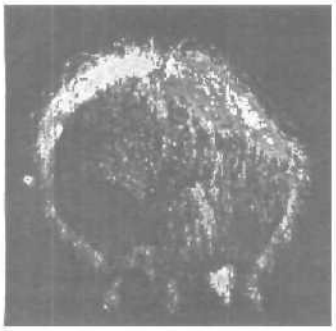
Глава 34.
Создание волос с применением plug-in Shag: Fur

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 полумиллионофейсные сцены, которые, к сожалению, пока не каждый может себе позволить. А хитрая система систем партиклов плоха тем, что практически невозможно выполнить анимацию полученной модели — партиклы-то движутся, и остановить их без дополнительных plug-in'ов не так-то просто. Да и сами партиклы тоже достаточно прожорливы в смысле железа.

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

С помощью использования специальных векторов, возможна настройка таких сложных опций, как подгибание волос — ведь не у всех же 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, естественно, можно анимировать.

Sub-Material'ы 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, листва деревьев, вообще всевозможная растительность, оригинальные эффекты (вместо обычного bump можно сделать что-то действительно объемное), при небольшом желании с его помощью можно даже моделировать какие-то объекты (например, когти у злобного пришельца получатся очень интересными, если их сделать не монолитной геометрией, а из жесткой лоснящейся щетины).

К сожалению, эти эффекты не так легко даются. Основные концепции (особенно векторы) даются относительно сложно для понима-

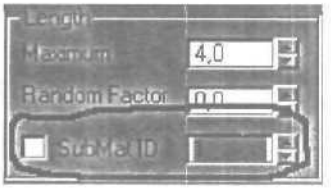
ния, а если учесть, что хотя Shag: Fur и очень быстрый, но простая сцена рендерится все равно долго (где-то минуты 2-3), то на обучение уходит где-то с пол-дня. Туториалы, идущие с plug-in'ом, достаточно примитивны и не показывают самой сути.

Можно сказать, что в целом plug-in получился очень хорошим, и был бы просто отличным, если бы не некоторая сложность в обучении и использовании — ну, а это на самом деле объяснимо тем, что никто до этого ни с чем подобным не сталкивался и, соответственно, общаешься с «новыми технологиями». Plug-in'у можно обещать долгую жизнь, хорошую продаваемость и множество применений.

Волосатая дубинка

Предположим, у нас есть дубинка, хорошая дубинка, но хотелось бы, чтоб она была волосатой! Никак не иначе, а именно чтобы самый «дубинистый» конец был просто лохматым и темным, а «недубинистый» — светлым и почти бритым! Да еще и чтоб на волосы было похоже! Представили? Значит пора браться за дело!

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.



Как вы думаете, что это? Правильно! Именно здесь незадачливый программист заложил возможность присвоения текстуры параметру, просто объекту надо назначить мультиматериал, **diffuse** которых будет использоваться в качестве текстуры. Оттуда же можно вытащить и **цвет** волос. Приступим!

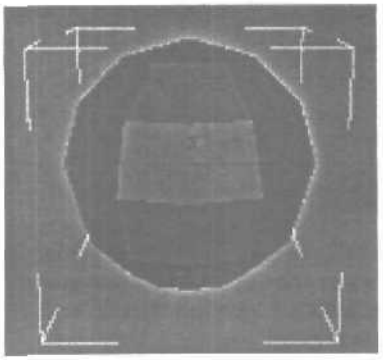
Перейдите в **Material Editor и** измените тип материала дубинки на Multi-SubObject, сохранив сам материал дубинки. Проследите, чтобы объект имел на всей поверхности материал с ID1. Теперь прикинем — на ID2 положим положим длину волос, на ID3 — плотность, на ID4 — материал. Все, что осталось сделать — 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 Betid. Отрегулировав параметры, стоит все же сконвертить свет в Hair Enabled (это делается в модуле Shag: Render/Tools) и просмотреть результат.

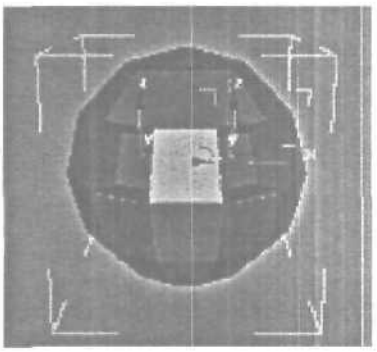
Глава 35. Создание простой головы

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

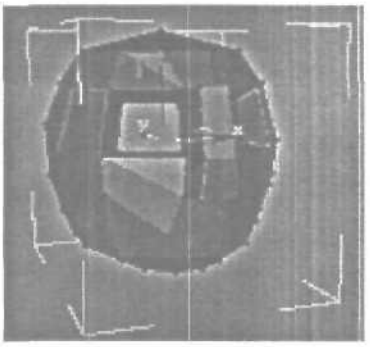
1. Начинается все со сферы: 12 сегментов без сглаживания. Сразу же применяем Edit Mesh и поворачиваем все вершины на 15 градусов, дабы в будущем иметь правильные локальные координаты.



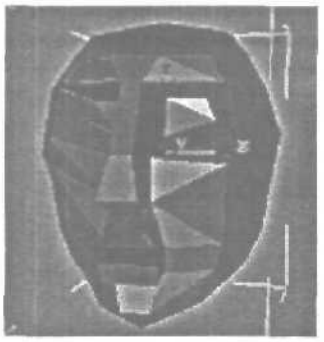
2. Определимся с глазами, для этого выделяем два соответствующих полигона и с помощью Extrude вдавливаем примерно так, как показано на рисунке:



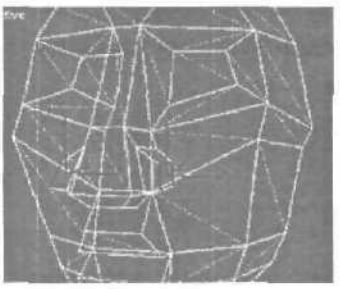
3. Теперь заготовка для носи. Выделяем два полигона между глазами и выдавливаем **Extrude**'ом (далее — просто «выдавливаем»). Две верхние точки возвращаем на место и объединяем с соответствующими точками основания носа (Weld ➪Target). Двигая точки, придаем носу более человеческий вид. Можно также выдавить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



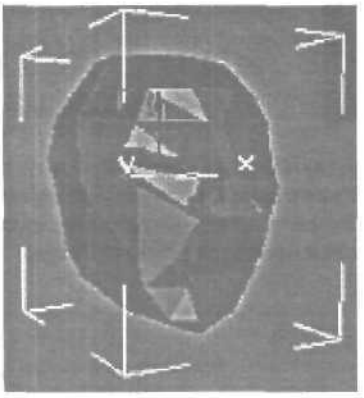
4. Так как в исходной сфере не достаточно вершин, дополнительные (особенно для верхней и нижней частей головы) можно получить, выдавливая «макушку» и «подбородок» и выравнивая полученные точки в соответствие с требуемой формой головы.



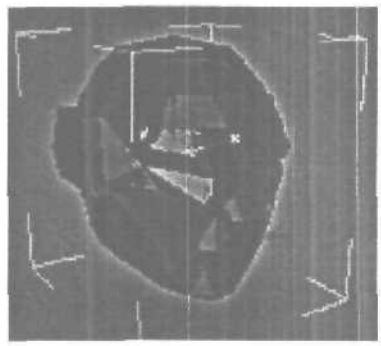
5. Ноздри вьщавливаются в два приема из боковых поверхностей носа.



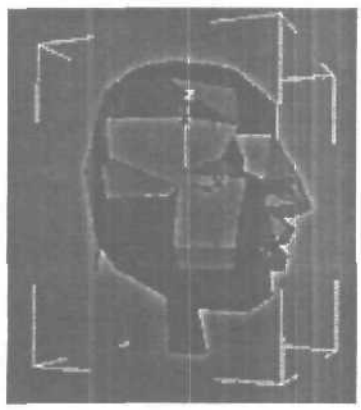
6. После всего проделанного голова имеет примерно такой вид.



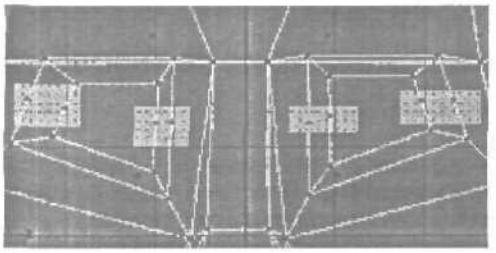
7. Можно сделать и уши. Как и с носом, выдавливанием и возвращением передних точек на место. Оставшиеся боковые поверхности также выдавливаем для пущей лопоухости.



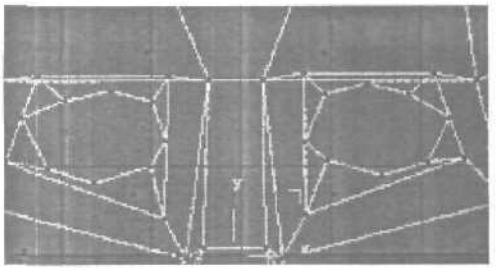
8. На этом же этапе можно сделать шею. Выдавливанием.



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)**



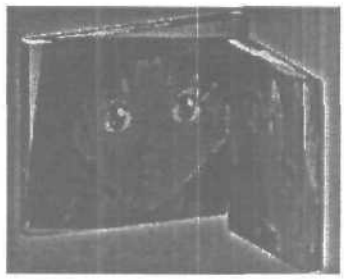
10. Также делим верхние *и* нижние границы глазниц. Двигаем вершины как надо.



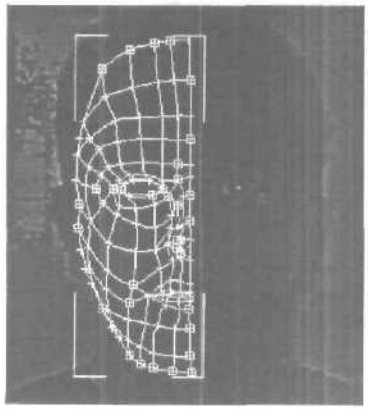
11. Собственно, пора делать **MeshSmooth**. Имеем простую, но уже человекоподобную голову. Дальнейшая обработка по вкусу, рекомендуется текстурирование и выборочное дополнительное сглаживание.

Глава 36. Моделирование головы с использованием Surface Tools

Первым делом, вы должны создать с помощью сплайнов контуры вашего лица в области просмотра Front. Просто обведите контур половины головы. После, 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 вы получили поверхность содержащую три или четыре вершины. Чтобы облегчить работу при создании дополнительных сплайнов, включите кнопку «3D Snap» с параметром **«End Points».**



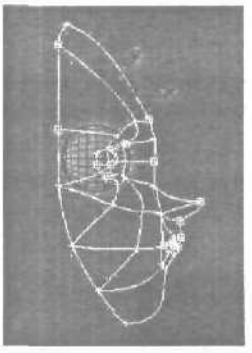
Обратите внимание, голова «аниме», обладает не лучшим контуром, так как я старался сохранить очень низкую детализацию. Далее, в процессе работы, я добавил необходимое количество сплайнов, для получения нужной детализации.



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

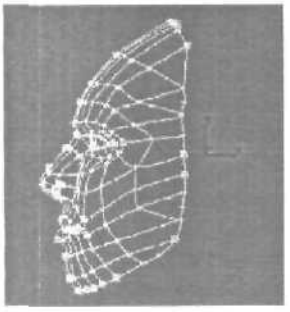
Добавление объема

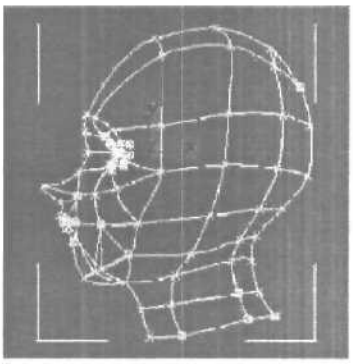
После того, как вы создадите основные сплайны спереди, то начинается самое интересное. Вторым шагом для вас станет добавление объема к созданным сплайнам. Используйте для этого фотографии вида спереди и сбоку, переместите каждую вершину, пока она не станет соответствовать вашим фотографиям.



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

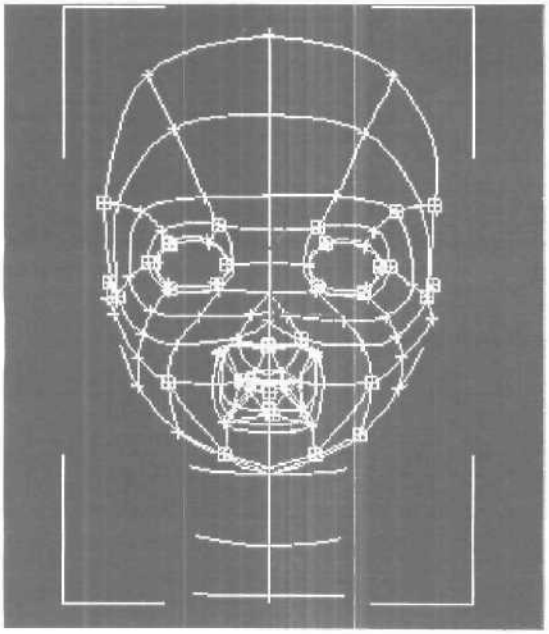
Добавление объема к сплайнам в 3ds max — наиболее длительная и нудная часть моделирования. Можно потратить часы, дни и недели, чтобы получить необходимый каркас с нужной детализацией. Но все затраченное время на это, позволит вам в дальнейшем получить очень качественную модель.





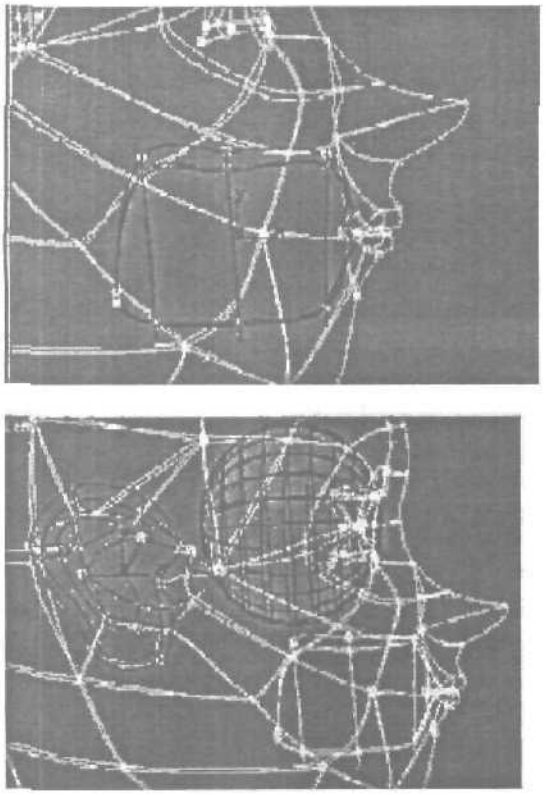
После того, как вы решите, что половина головы больше не нуждается в редактировании, то настало время для получения целой модели головы.

Для этого вы должны зеркально отразить эту половину и полученную копию разместить в нужном месте. Затем совпадающие вершины двух половинок необходимо соединить.

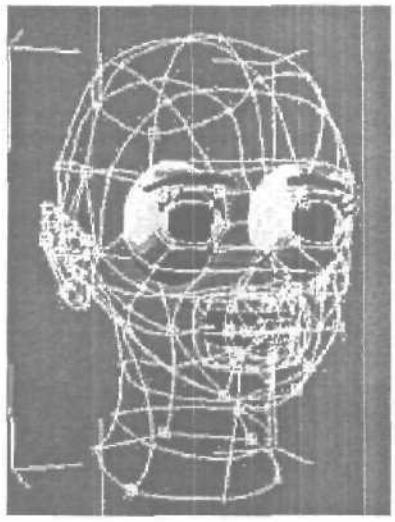


Доводка

Я обычно создаю форму подобно мешку, которая соединена со 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 37. 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 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, У=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 Strendth 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 OK. Измените цвет нового объекта на цвет морской волны.

3. Переместите в видовом окне **Left** объект Camera01 вправо в точку с координатами 60 ДО. 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.

Поверхность моря

1. Выберите объект **Quad** PatchOl 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 PatchOl.**

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).

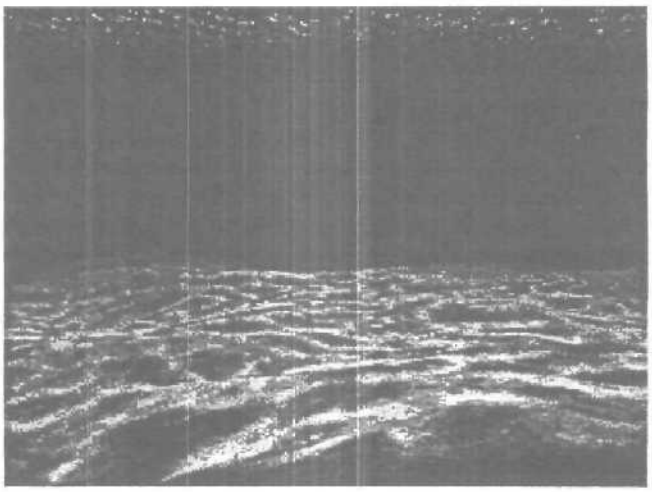
2. Измените во вкладке Modify цвет прожектора на ярко белый. Параметр **Multiplier** установите на 2,0. В группе **Attenuation** включите опции **Use** и **Show,** затем установите значения параметров Start **Range и End** Attenuation равными 95 и 100 соответственно. В разворачивающейся панели **Spotlight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

выберите опцию **Rectangle**. И наконец, в разворачивающейся панели Shadow **Parameters** включите опцию **Cast Shadows.**

3. Включите опцию **Projector,** чтобы включить объект **SpotlightOl** 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

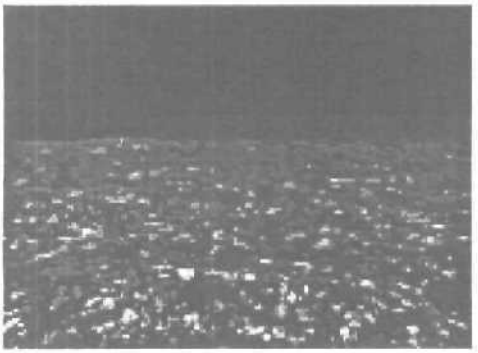
Итак, приступим к созданию подводной сценки на небольшой глубине от поверхности моря или озера.

Предположим, что погода солнечная, на поверхности лето в разгаре, ветра почти нет; так, легкий бриз, ну и волнения тоже почти никакого — только рябь небольшая. В этом уроке не ставится задача заполнить водные толщи тварями плавучими и ползучими, 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:

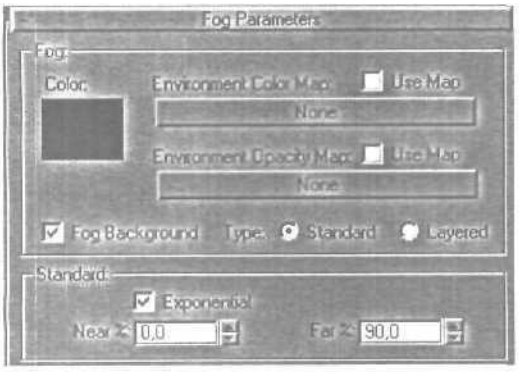
Length: 1000,0
Width: 1000,0
Height: 0,0
Length Segs: 40
Width Segs: 40

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

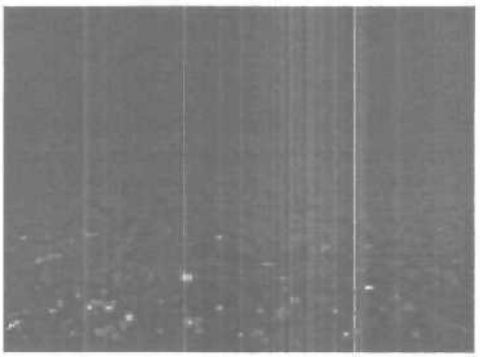


Дальше будем создавать эффект водной толщи.

В главном меню, пункт **Rendering,** войдите в **Environment** и в разделе Atmocphere 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.

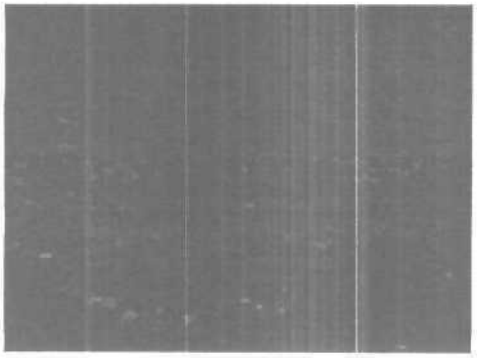


И, наконец, создадим блики на дне и солнечные лучи в толще воды.

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'a** и перетащите кнопку Map из параметров света на свободную ячейку материала. Теперь можно выставить нужные параметры.

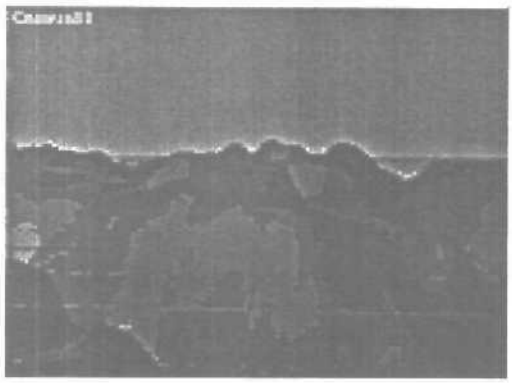
Для вывода на экран лучей света, **в** Environment'e необходимо выставить еще один эффект **Volume** Light.

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

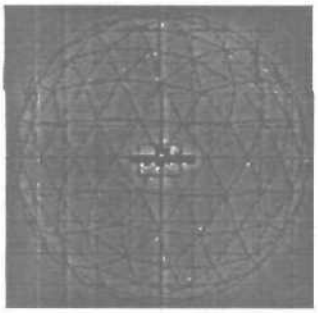


Глава 38.
Как сделать облака, море и волны

Итак у нас есть ландшафт.

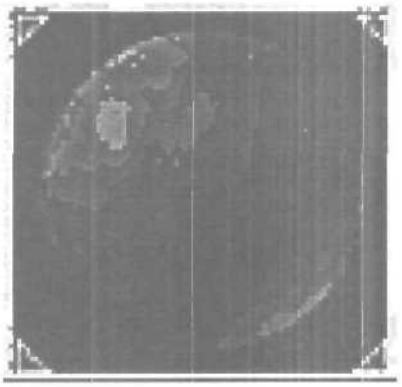


Нужно его оживить. Ну конечно мы наложим на него текстуру. Подберем освещение, но нам нужно добавить небо! Делается это так. Рисуем сферу которая заключала бы в себя всю нашу сцены можно не скромничать и сделать большую сферу.



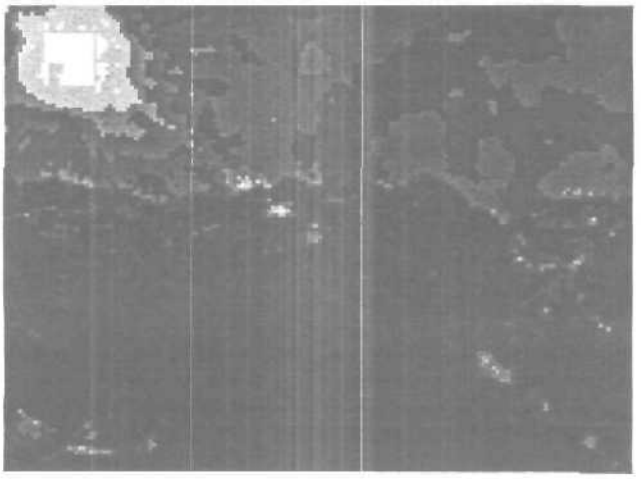
Теперь займемся подготовкой материала для нашего неба. Для того чтобы иметь возможность анимировать его в дальнейшем воспользуемся следующий вариантом: создадим материал диффузной картой которого будет Noise.

Вот как он будет выглядеть:

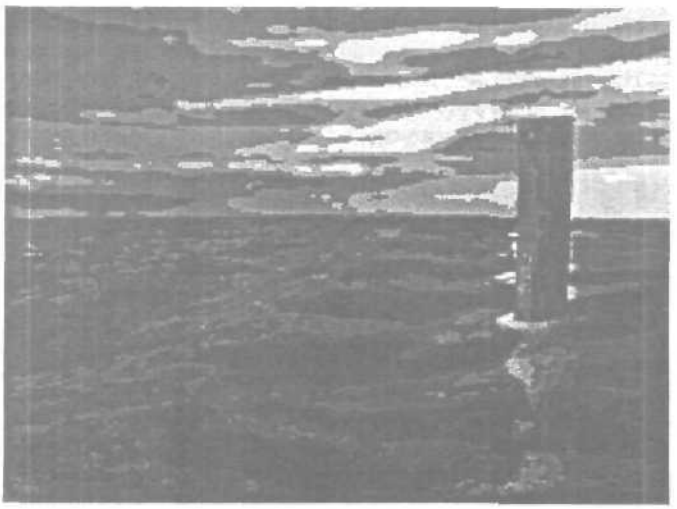


Назначим его на нашу сферу. Попробуем отрендерить... О-па, ничего не видно? Конечно, потому что мы не воспользовались модификатором **normals.** Нам нужно как бы вывернуть наизнанку нашу сферу, то есть, нам нужно повернуть ее нормали внутрь. Делается это путем включения **flip** normals.

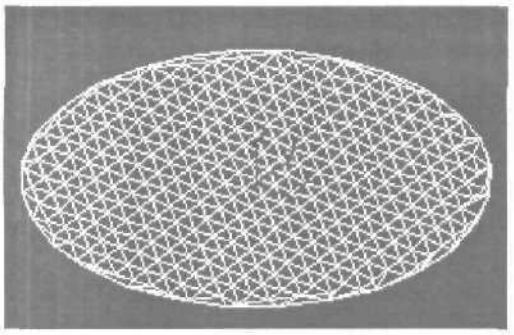
Ну вот и наш ландшафт с облаками. Для пущего реализма я добавил солнце с бликами. Делается это путем махинаций в **video post** (путем добавления стандартного плага под названием **Lens Flare).**



Теперь взгляните на эту картинку:

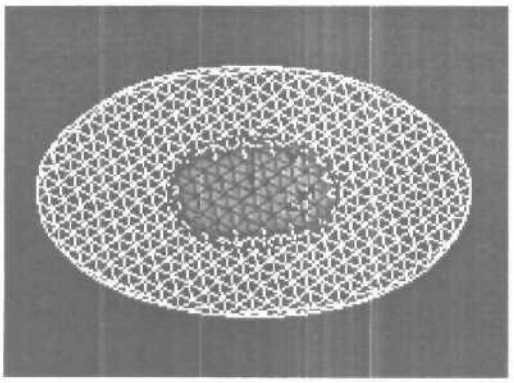


Начнем с простого, — со сплайна, или, точнее, с окружности. Сделайте большую окружность, которая послужит контуром вашего океана. Затем примените к ней модификатор **extrude. He** забудьте установить для **capping** опцию **grid.**

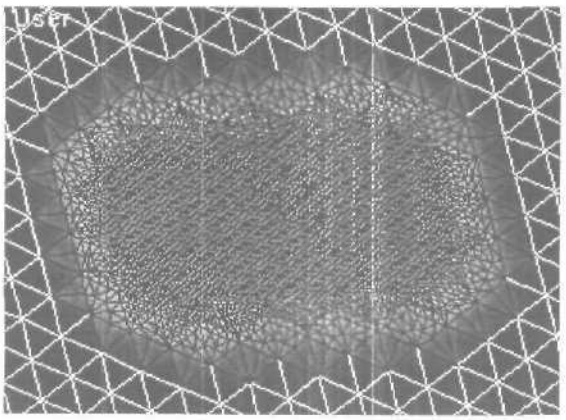


У вас получится нечто в этом роде, фактически, то же самое. Мы еще недостаточно продвинулись, чтобы можно было что-нибудь напутать. Затем следует применить модификатор **mesh edit.** Теперь нужно сделать волны. Прежде всего, нужно выбрать грани, которые будут для этого нужны. Конечно, можно выбрать их и вручную, но это доставит большое неудобство в случае, если захочется поменять положение каме-

ры. 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 tesselate с двумя или тремя итерациями. У вас должно получиться следующее:

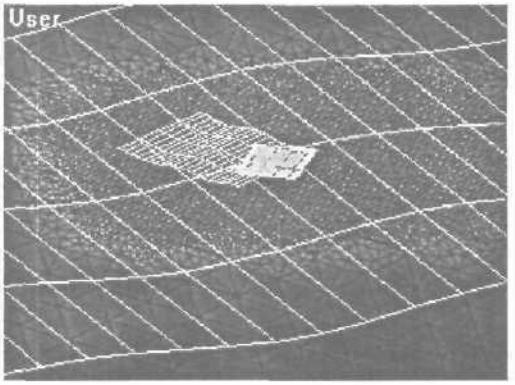


Посередине круга появится густая сетка. Наступила пора сделать волны. Это можно было бы сделать при помощи модификатора wave, но, боюсь, тогда придется слишком много думать. Итак, воспользуемся пространственными волнами.

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



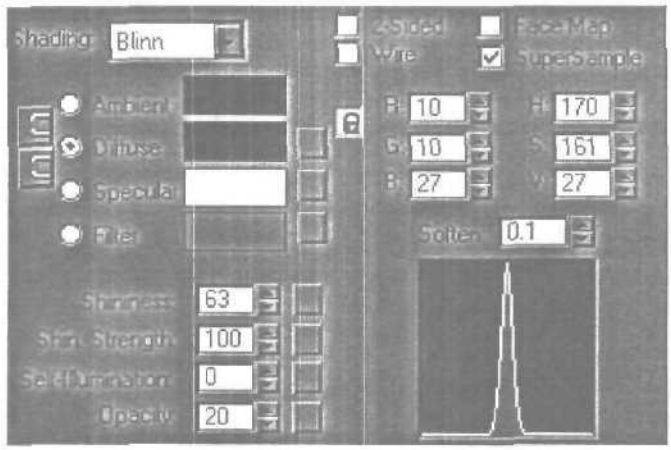
Не забудьте связать волны с поверхностью воды, — воспользуйтесь для этого кнопкой:

Поскольку воздействие будет оказано только на мозаичные участки **(tesselated),** 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 tesselate 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, посколысу 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 shineness. Пусть у графика яркости будет такой же высокий пик, как на рисунке ниже.



Скопируйте все установки с картинки, за исключением supersample.

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.** Для тех, кто не знает, что это такое, — bump mapping — это создание иллюзии текстуры для бедных — то есть, без реального ее моделирования.

Классический метод создания текстуры воды, — использование карты шума (noise). Мы воспользуемся этим приемом в несколько переиначенном виде. Мы 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 I 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. О 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 еше 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 размешена камера, но поленился. Это была бы довольно трудная работа, поэтому я решил выбрать такое значение параметра, которое подошло бы 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.

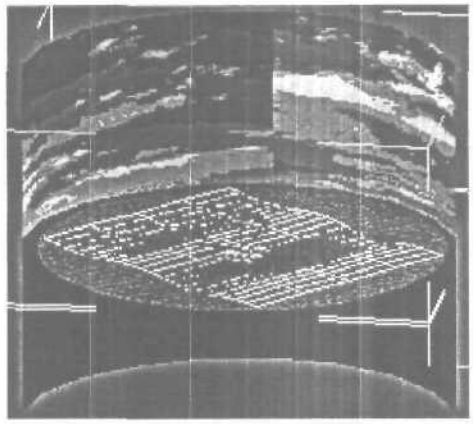
Стоит удостовериться, что **mapping type** имеет параметры **environment и spherical.** Чтобы усложнить картинку, я несколько раз ее размножил. Теперь надо перейти в environment setting в меню **render.** Нажмите на кнопку рядом с надписью **background** и перейдите в редактор материалов. Выберите текстуру 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

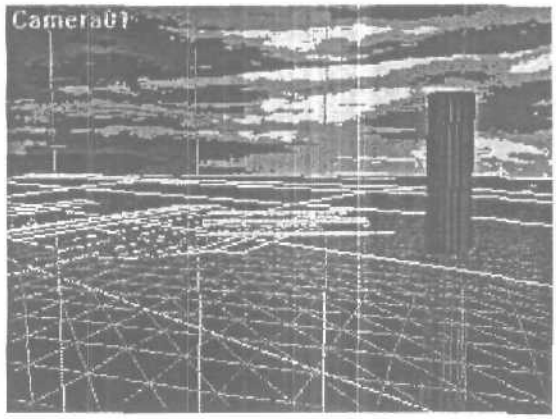
Поверьте, я перепробовал все, причем неоднократно. Но лучшим в этой ситуации оказалась thinwall. Этот вид хорош еще и тем, что у него мало настроек. Я просто установил **bump** effect =3, чтобы больше внима-

ния уделялось карте неровностей. Уровень рефракции я установил равным 50%, а IOR (индекс рефракции) — равным 1.8. Я знаю, что это не IOR воды, но выглядело все прилично.

Еще неплохо было бы, 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 днищем, 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 прекрасно сработало.



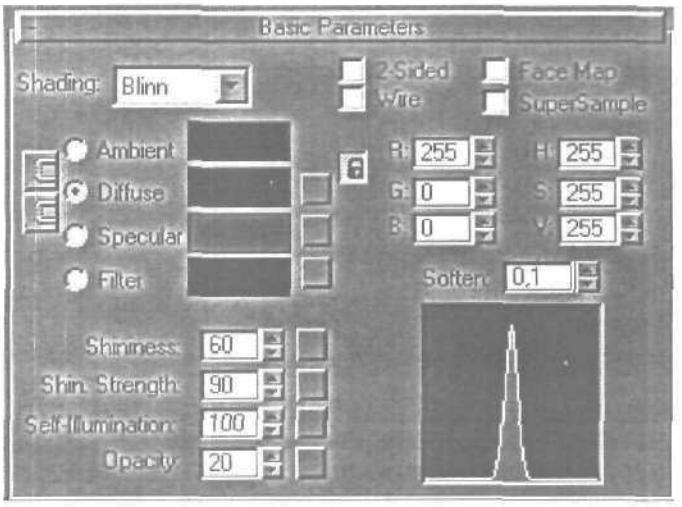
Затем я поместил внутрь камеру. Вод вид из этой камеры:



Вот и все. Еще я добавил немного motion blur для воды, взяв стандартные значения. Затем запустил рендеринг через video post, воспользовавшись фокусом **lensFX** (который обламывается девять раз 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.

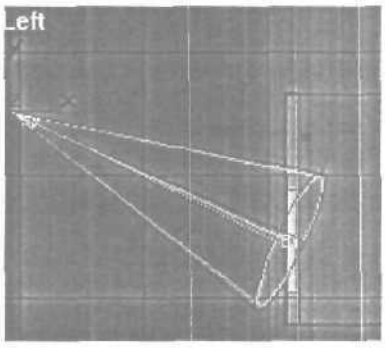
Глава 39. Объемный свет на примере мозаичного окна

Объяснять как нарисовать раму и вставить туда стекла, думаю, надобности нет, поэтому сразу перейдем к созданию материала для цветных стекол. Начнем с красного. Откройте панель **Material** Editor и выставьте следующие параметры:



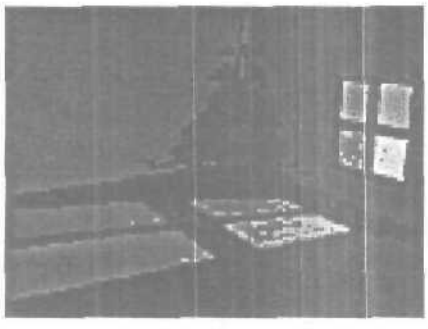
Как вы заметили, цвет выставляется только на Diffuse (цвет стекла) и **Filter** (просвечиваемый цвет). Непрозрачность, самосвечение и блеск подберите экспериментально, так, чтобы ваше стекло выглядело как настоящее. Таким же образом создаются и все остальные цвета. Теперь соберите все в кучу: стекла, 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.



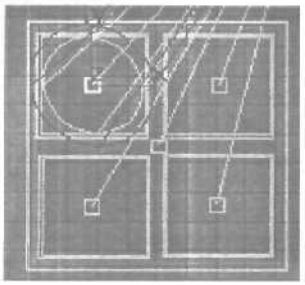
В разделе Shadow **Parameter** установите метку на **Cast Shadows и Use Ray-Traced** Shadows.

После рендеринга у вас должна получиться вот такая сцена.

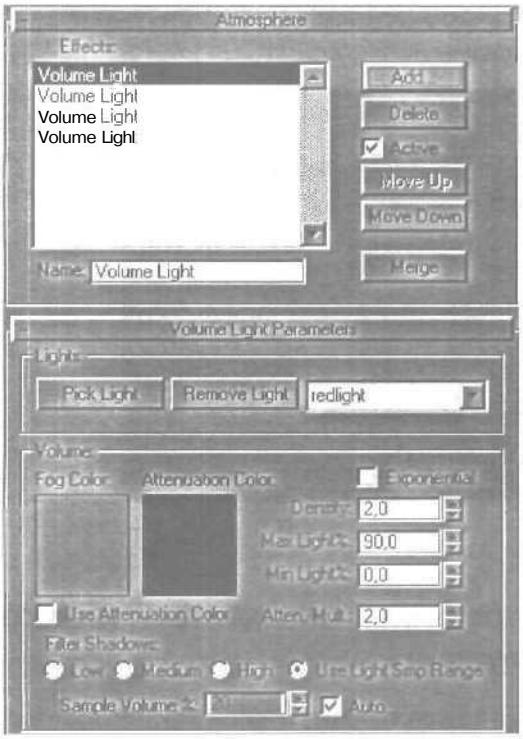


В большинстве случаев этого бывает вполне достаточно, но можно еще добавить реалистичности — придать видимость лучам света. Достигается это созданием отдельных источников **Target Spot** с одним центром, но с различным направлением луча — для каждого стекла свой источник света.

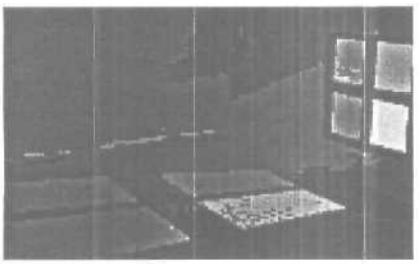
В General Parameters для каждого источника, нажав кнопку **Exclude,** отключите освещение объектов, на которые падает луч, чтобы не было лишних световых пятен, а в Spotlight **Parameters** установите размер светового пятна не больше цветного стеклышка, на которое падает этот луч. Для удобства определения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

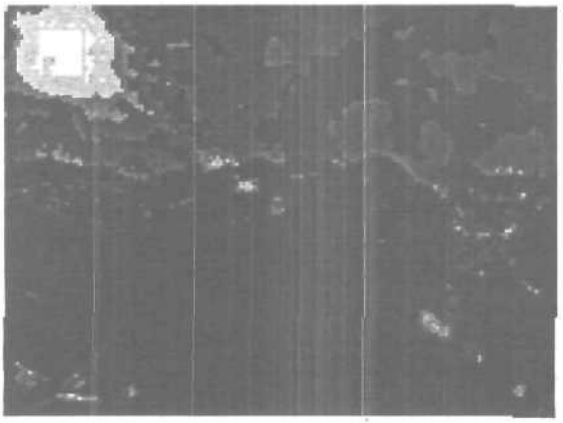


Естественно, для каждого луча Fog Color должен быть своего цвета.



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.

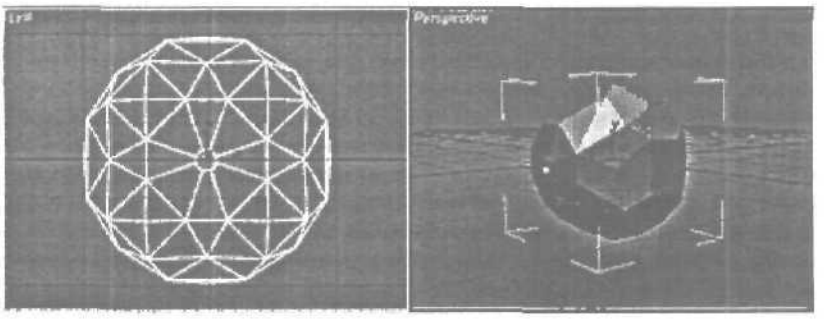


Глава 40. Бриллианты, рубины, аметисты... или как делать бижутерию

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

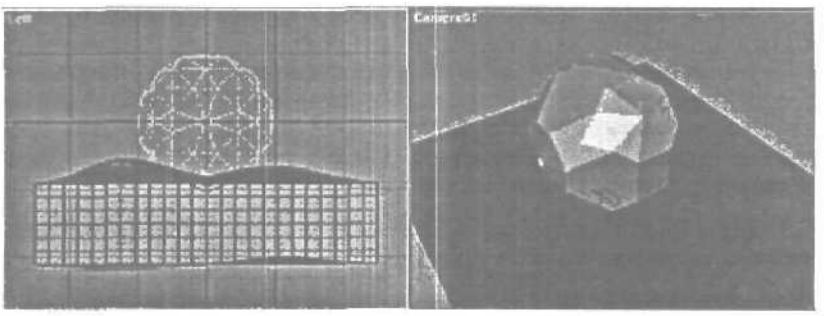
Перво-наперво нужно создать форму будущего граненого брилльянтика — в данном конкретном случае я пошел по наипростейшему пути — примером нам будет служить обычная бусинка. Почему не камешек огранки под колечко? Да потому, что это не просто Sphere без сглаживания — натуральная огранка гораздо сложнее.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

А 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 (enviropment) 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.

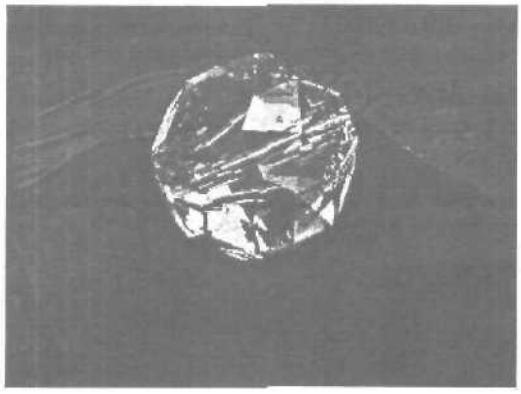


Именно материал определит, насколько хорошо мы справились с задачей. Не буду тратить время на окрашивание стола с пуфиком — посмотрите прилагаемую сиену, да и для бриллиантика всего расписывать не буду, лишь обратим внимание на ключевые моменты. Как я уже упоминал, «игра» кристалла складывается из отражения и преломления окружающей среды и своих собственных бликов. Это — ключевое слово, достаточно понять, что это за блики и сразу станет ясно, «как его делать». Все мы в школе изучали основы оптики и знаем, что такое разложение света через призму. Однако в 3D подавляющая часть народа забывает об этом эффекте, и даже те, кто помнит, начинают искать софт, считающий дефракцию. 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 **enviropment** для отражений и преломлений!

К счастью рэйтрэйсовый материал 3ds max позволит нам сделать это, и не только это. Требования, предъявляемые к карте отражения — это 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 «1 **sided**». сила отражения должна лишь немногим превышать половину. И, 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 41. Создание молнии

Рисуем каркас будущей молнии.

Для этого просто расположите во фронтальном окне проекции линии так, как вам кажется выглядит молния.

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.

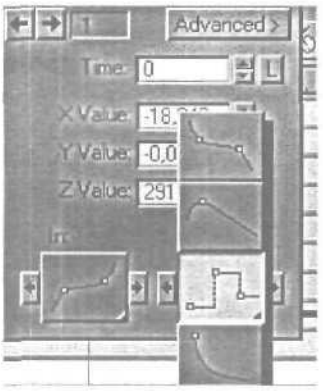
В командной панели во вкладке **Modify** выберите модификатор Noise и установите следующие параметры:

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 куда-нибудь за пределы выводимой части проекционного окна и нажмите большую кнопку **Animate.** Установите в окошке текущего кадра 20 кадр и переместите молнию в ви-

димую часть экрана. Затем установите 25 кадр и снова сместите объект с глаз долой. Отключите анимацию.

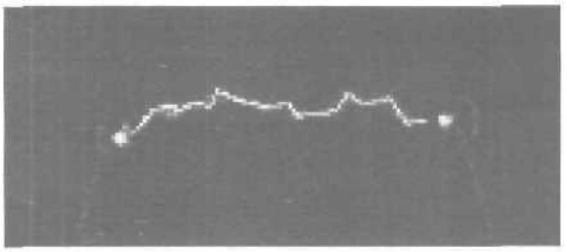
Откройте окно Track View и кликнув правой кнопкой по каждой ключевой точке измените параметры перемещения объекта во входной и выходной точке между ключевыми кадрами так, чтобы движение было скачкообразным.



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

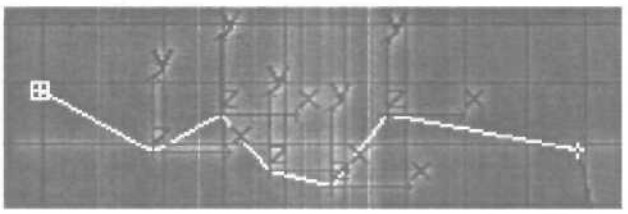
Откройте панель Video **Post** и установите все необходимое в окошке -Queue. Измените установки параметров **Lens Effect Glow;** Свиток **Preferences Effect — Size: 2,0 и Color —** User (выставьте такой же цвет **как и** в **Material** Editor'e). Все, молния готова. Для вывода конечной анимации естественно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

А можно нарисовать и такую молнию, здесь можно даже без лофтинга обойтись.



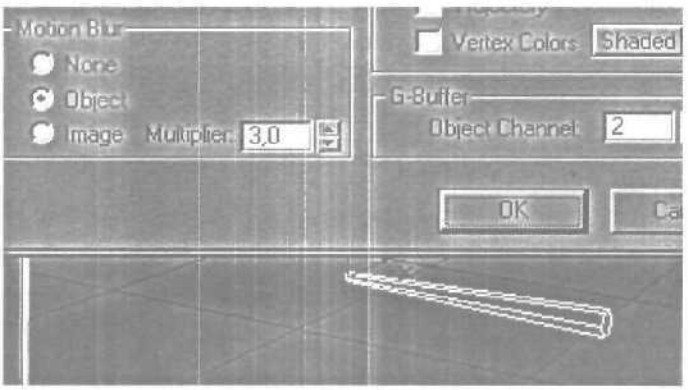
Глава 42.
Создание фотонной пушки

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,**

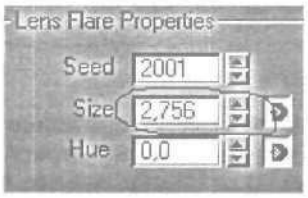


И еще тонкость: чтобы при вращении всей установки разряды не разбегались во все стороны, надо в **Hierarchy** сдвинуть все **Affect Pivot Only** к центру вращения.

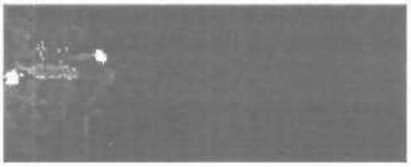
Для центрального лазера использован конус радиусы которого по началу одинаковые, а затем дальний от установки немного расширяться и под коней весь конус сдвигается в направлении выстрела, естественно все операции проделываются со включенной анимацией. Для придания реалистичности в **Video** Post'e **к** цилиндру применен **Lens Effect Glow и и Object Properties** установлен Motion Blur.



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

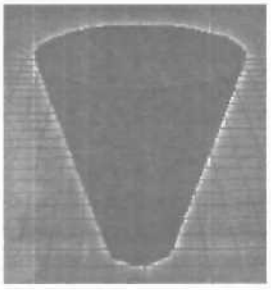


Картинка, конечно, не может передать всех тонкостей, на самом деле сцена намного красивей.



Глава 43.
Простой способ создания пара

Для создания пара понадобиться изготовить простой конус.



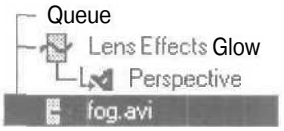
Кликните по конусу правой кнопкой мыши, выберете Properties... и в появившемся меню, около надписи G-Buffer, поставьте **Object Channel:** значение 1.

Назначьте полученному объекту материал с такими параметрами.

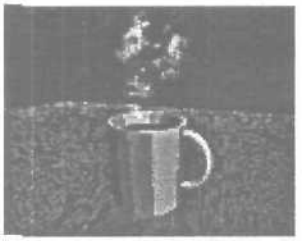
В главном меню, команда **Rendering,** выберете функцию **Video** Poct, нажмите кнопку Add Scene Event и установите, если еще не установлен, вид на сцену (Camera, **Perspective).** В результате в **Video Post'e** в окошке -Queue появится выбранный вид. Кнопкой **Add Image Filter Event** присоедините **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).

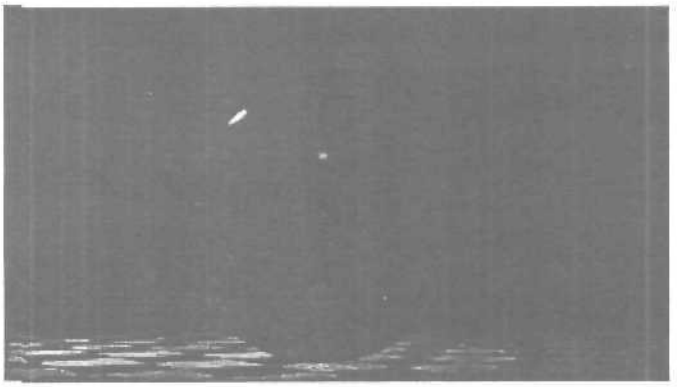
Подтвердите установки кнопкой **OK.** Добавьте кнопкой **Add Image Output Event** (проследите, что бы на панели **-Queue** не была выбрана ни одна установка) на панели **Video Post'a** формат и параметры выводимого файла.



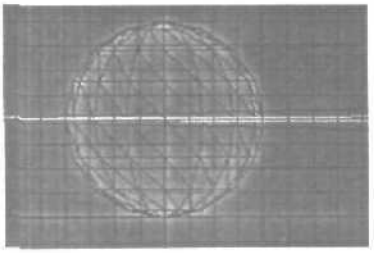
Теперь, если выделить на панели **-Queue** установку вывода файла и нажать кнопку **Execute** Sequence, готовый файл будет записываться уже с присоединенным эффектом. Остаюсь дорисовать на сцене сопутствующие пару объекты и у вас получится что-то в это роде.



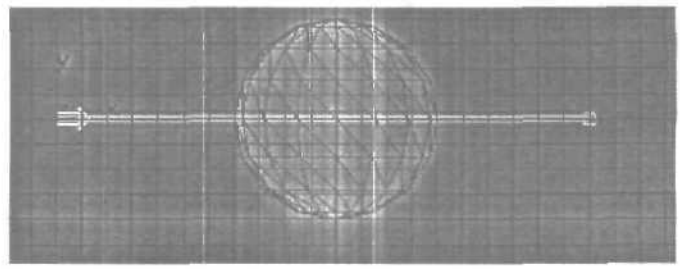
Глава 44.
Создание реалистичного лазера



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 — **Spotlight,** но при этом, даже, если мы сжимаем конус этого источника света до очень малого утла, то все равно мы никогда не получим лазерный луч.

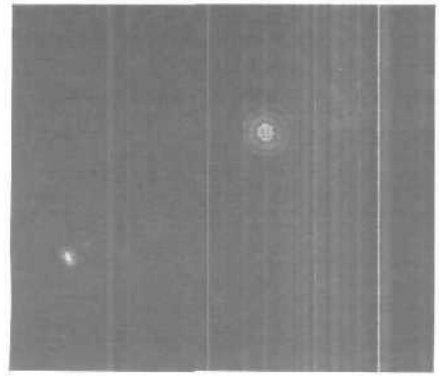


К счастью, в 3ds max, есть источник света **Direct,** который имеет параллельные лучи света, в отличие от **Spotlight.** Поэтому, будем использовать его для нашего лазера.

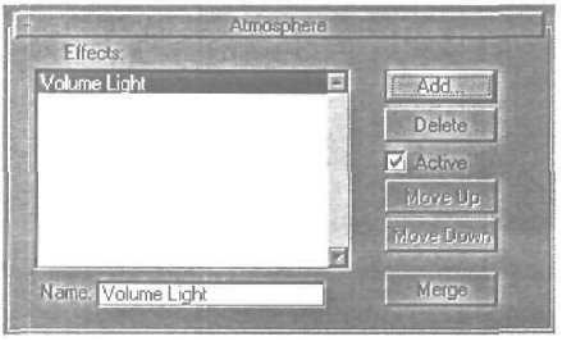


Источник света должен иметь очень узкий пучок света. Я использовал следующие параметры настройки:

Также я использовал красный цвет для источника света и увеличил **значение multiplier:** После рендера, эффект лазера должен быть похожим на кое-что вроде этого:



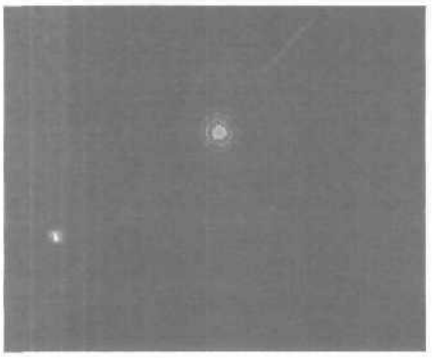
Это хорошо походит для типичной «съемки в лоб», в которой кто-то нацелится на объект и на объекте лазер имеет красную точку. Но мы хотим большего. Для этого мы используем параметры настройки Environment. Добавьте эффект Volume **Light** Environment:



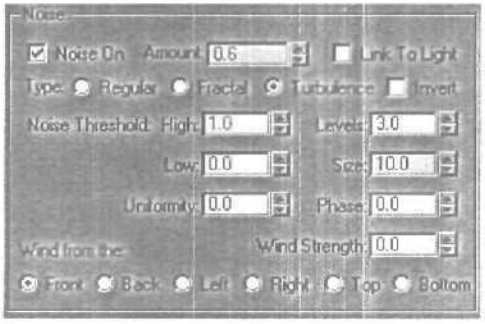
Выберите источник света **Direct для** этого эффекта:

Установите цвет тумана на красный, как у источника света и установите плотность **(Density)** равное **13.**

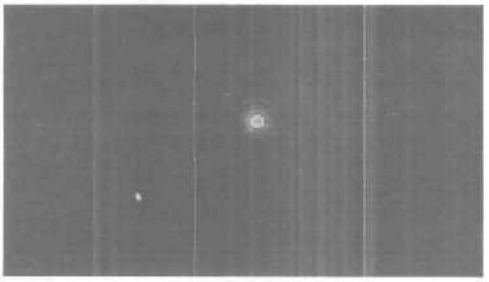
Теперь луч станет видимым. Но если вы стремитесь к реализму, то вы будете должны признать, что это действительно дрянно-выглядящее изображение. Лазерные лучи никогда не видимы в чистом виде и имеют некоторое количество пыли или дыма в окружающей среде.



Включите **Noise.** Я использовал следующие параметры настройки:

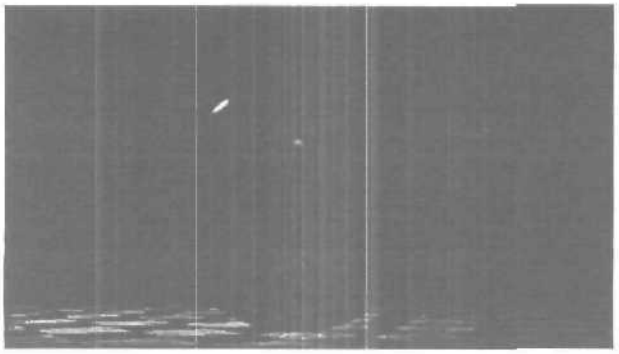


Теперь намного лучше. Разве не так?



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.

Лазер, проходящий через стеклянную сферу:



Глава 45.
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 **OmniOl** значения RGB — 255,204,0. В группе **Attenuation** значение Stert Range установите равным 50, а **End Range** равным 60. Включите опции Use и **Show.** Для Omni02 значения **RGB — 255,104,0.** В группе **Attenuation** значение Stert **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 **OmniOl.** В группе **Volume** измените значение **Density** на 30. В группе Noise включите опцию Noise **On,** измените значение **Amount на 0,5.**

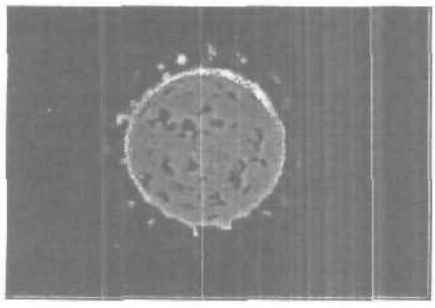
4. Щелкните на кнопке **Add...,** из появившегося списка выберите **Volume Light.** Щелкните на кнопке **Pick Light** и в любом видовом окне выберите источник света Omni02. В группе **Volume** измените значение **Density** на **30.** В группе **Noise** включите опцию **Noise On,** измените значения Amount 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. Щелкните на кнопке с надписью **Map** #1 **(Noise).** В окне **Put to Material Editor** выберите **Slot 6.**

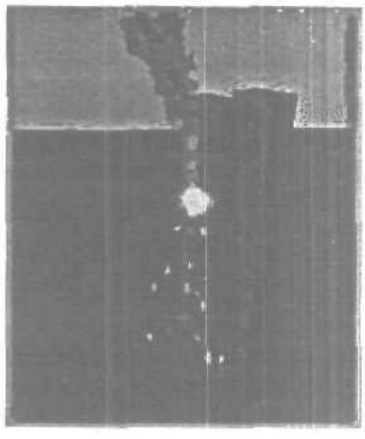
6. Войдите в **Material Editor.** Убедитесь, что активен слот 6, В разворачивающейся панели **Maps** щелкните на кнопке справа от опции **Diffuse.** В появившемся окне **Material/Map Browser дважды** щелкните на элементе Noise. В разворачивающейся панели **Noise Parameters** установите **Size** равным **0,1,** Low = 0,65. Щелкните на кнопке с надписью **None** справа от **Color** #1. **В** появившемся окне Material/Map **Browser** дважды щелкните на элементе 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**

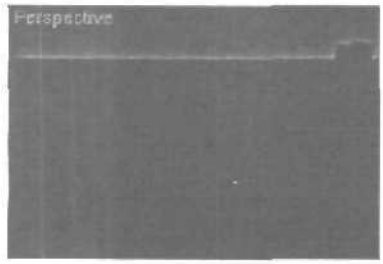
дважды щелкните на элементе Noise. В разворачивающейся панели Noise **Parameters** установите Size равным 10, **High = 0,71.** Щелкните на черный прямоугольник Color **#1** и установите значения **RGB = 246,255,0.** Щелкните на белый прямоугольник **Color** #2 и установите значения RGB = 237,255,0. Щелкните 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 **SphercOl,** чтобы сделать его активным. Щелкните на кнопке **Assign** Material to Selection, чтобы назначить материал сфере. Попробуйте отрендерить сцену.



Глава 46. Имитация работы сварочного аппарата



1. Создаем геометрию забора.



2. Объект для искр.

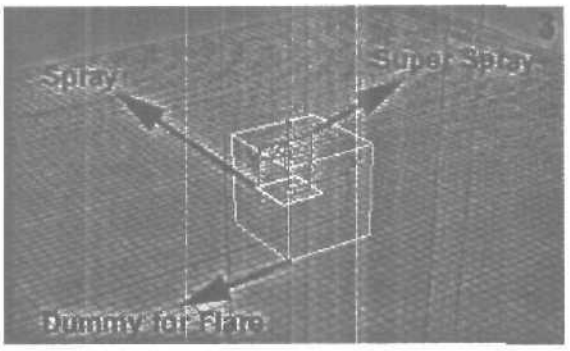
Создайте объект для имитации искр от сварки посредством редактирования коробки **EditMesh,** затем разместите его куда-нибудь, где его не будет видно.

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 на используется для искр) со следующими параметрами: **Off Axis: O, Spread: 81; Off Plane: 0, Spread: 180; V. Display — Mesh; Use Total: 500; E. Start: 1; E. Stop: 60; Display Until: 100; Ufe: 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:



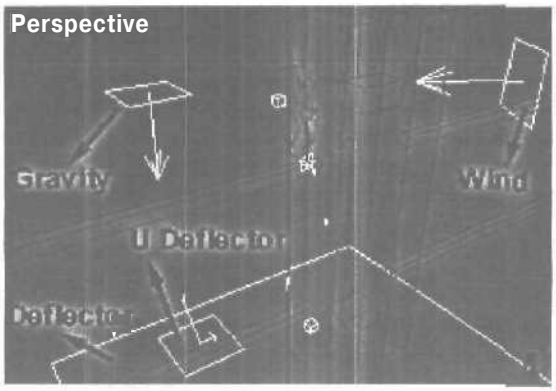
4. SpaceWarp.

4.1. Создайте Deflector, (он будет использоваться для отскакивания частиц от «земли») установите параметр Bounch **в 0,2,** после разместите, как на рисунке ниже и привяжите к искрам.

4.2. Создайте UDeflector, (он используется для отскакивания частиц от забора) установите Bounch равным 0,1, теперь возьмите и сгруппируете 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 на рисунке, и привяжите к дыму.



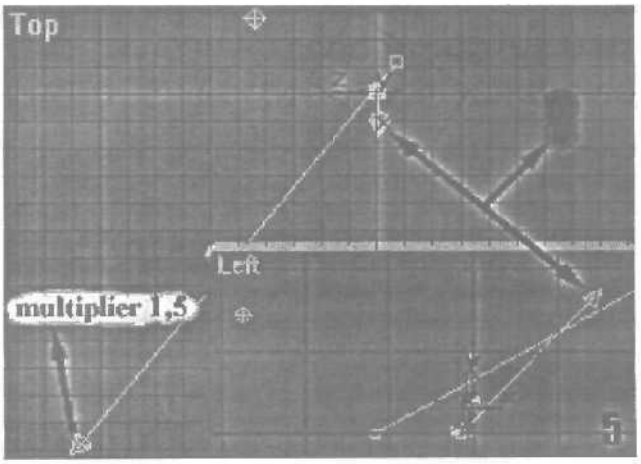
5. Материал **для искр и** дыма.

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 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 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.**

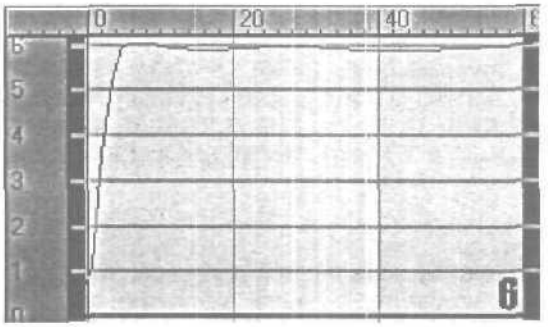


7. Glow **and Flare + Flare Animation.**

7.1. В Video Poste повести камеру (или перспективу); **Lens Effects Glow,** в нем установите Mat ID: 1; Size: **I; 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 О **Video Post** ➪ **Lens Flare** ➪ Size и исправьте линию до такой как на рисунке:



Глава 47. Тонкости и хитрости создания металлических поверхностей

Оглянитесь вокруг, наверняка вы увидите множество предметов, где используется метал. Это и не удивительно, ведь после того, как древний человек научился обрабатывать металлическую руду, начался бурный рост цивилизации, и теперь уже не наши прародители прятались и убегали от диких зверей, а дикие звери стали скрываться от человека вооруженного металлическими ножами или же стрелами с металлическими наконечниками. В прочем очень скоро зверья поубавилось, и охотники переключились на «ноу серов».

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 мною и все они разные, секрет успеш-

ного моделирования и заключен в том, на сколько хорошо вы понимаете различия характеристик металлов.

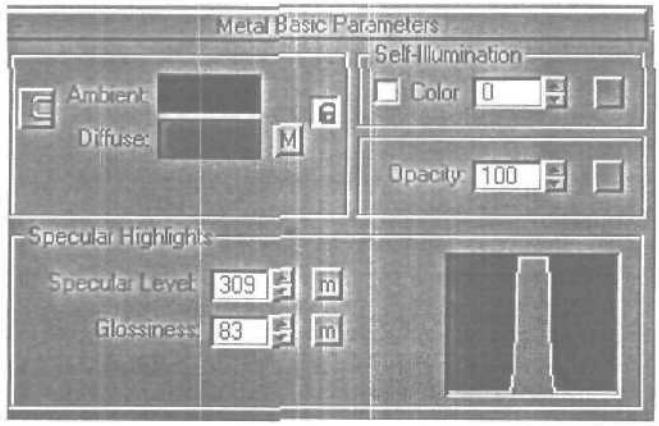
Первое чем отличаются металлы это отражение, а зависят они от твердости самого металла, так, например хром и медь, у меди отражение очень мутное, а в отражении хромированного бампера автомобиля можно заметить все изъяны дорожного полотна.

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 бликов практически не видно, так как они очень тусклые и размытые. Это, 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 чрезвычайно удобно. Недостаток у этого метода только один, цена создания мелких деталей, для их хорошего качества придется добавить пару сотен тысяч фейсов.

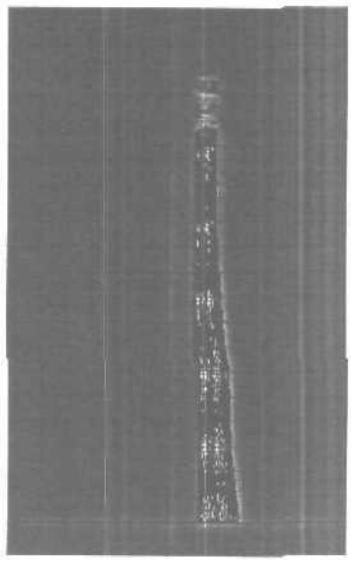
Bump напротив геометрию не изменяет, это плюс, но в результате не возможно создать сильное смещение поверхности. Особенно заметно это при взгляде на края объекта и его тень, так как здесь влияние бампа обнаружить не удается. Следовательно, бамп это для мелких деталей, что в нашем случае и необходимо. Наиболее грандиозных результатов удается добиться, комбинируя эти два способа, классическим примером этого является кирпичная стена когда бамп используется для нанесения мелких дефектов, царапин и трещин, а дисплейс для получения общего рельефа стены.

Для выбора алгоритма тонирования есть несколько вариантов. Прежде всего, это «Метал», доставшийся нам еще с DOS'овских времен, ветераны хорошо помнят это время и возможно когда собираются за рюмкой чая со слезой вспоминают это прекрасное время. Одним из достоинств этого алгоритма является задать такие параметры **Glossiness** и Specular Livel, что яркость блика будет либо постоянна на всем его протяжении, либо даже спадать к центру блика.



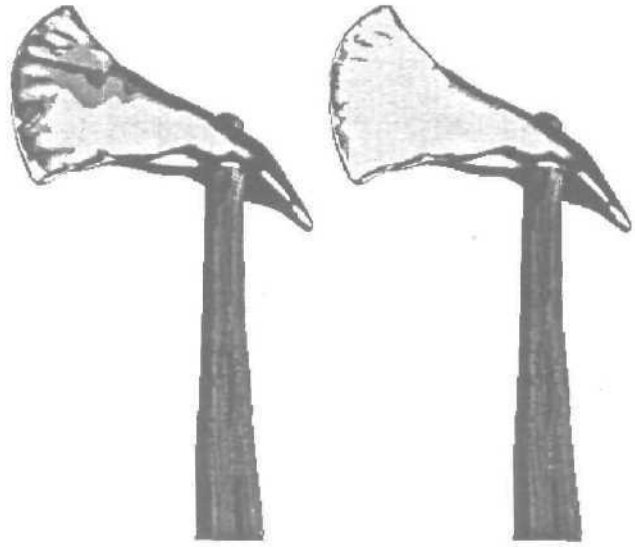
Давайте попробуем изучить этот материал.

Для начала создадим тестовую с цену, я использовал модель топора взятую из коллекции бесплатных моделей, на одном из множества Интернет-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 сейчас повозимся. Установите цвет диффузии в серый, задайте какое-нибудь неприлично большое значение параметру **Specular** Livel, ну к примеру 400, **а Glossiness — 80,** это обеспечивает нам очень яркие и точные блики.

Получилось не плохо, но недостаточно реалистично:

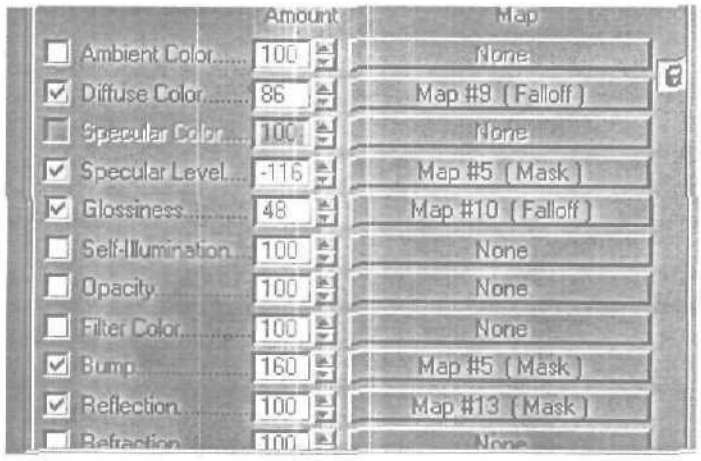


Теперь добавим отражение, раскройте свиток Maps и добавьте карту **RayTracc в** канал Reflection. Двигаясь дальше, добавим в слот Diffuse **color** карту Falloff, это обеспечит более темные края, у нашего боевого топора. Скопируйте ее в канал Glossiness, одновременно уменьшив влияние этих карт до 80-50 процентов.

Настала пора состарить немного наш топор, ведь он у нас не на складе лежал, а участвовал в боевых действиях. Наложим на канал бамп-карту Noise, очень маленького размера, от **0.01** до 2. И затем, чтобы придать большего реализма, скройте часть нойза маской, по-хорошему конечно нужно эту маску нарисовать, учитывая, где в реальности на объекте могут возникнуть вмятины и царапины, но сейчас сойдет и что-то типа карты **Smoke** с большим размером зерна.

Чтобы еще больше улучшить внешний вид металла, скопируйте карту бамп в слот Specular **Livcl** и задайте ей отрицательное значение, те-

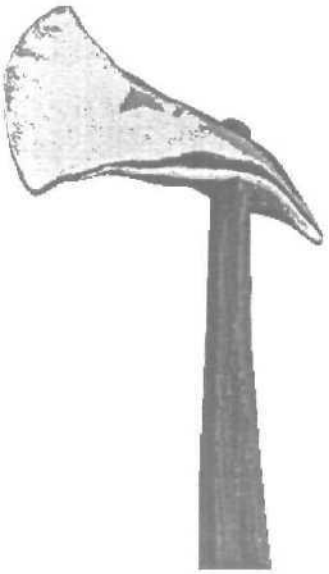
перь неровность поверхности будет оказывать значительное влияние нл силу бликов, и в завершении нужно будет замаскировать канал отражения так же как это было сделана с картой бамп.



Согласитесь, получился, достойный представитель оружия наших предков, пролежавший в земле энное количество лет.

Теперь давайте со следующим алгоритмом тонирования, **STRAUSS,** он не напрягает большим количеством параметров, и подходит в тех случаях, когда нужно быстро получить неплохой результат.

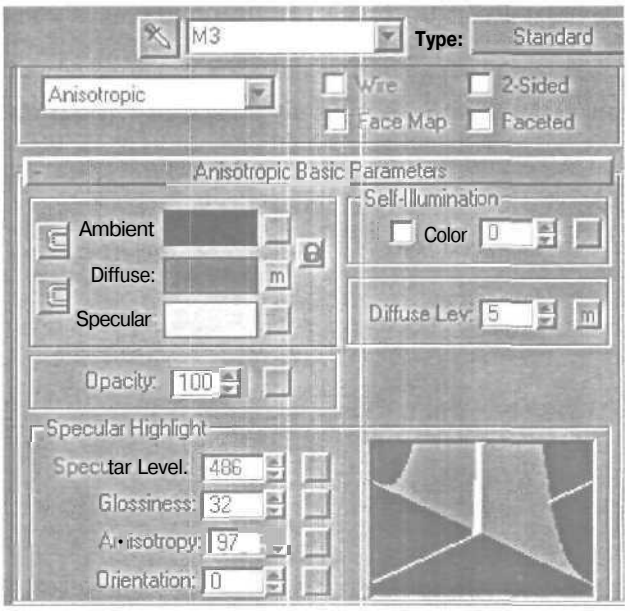
Основным органом управления является параметр Metalness, чем он больше, тем более металистая поверхность.



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** Livel и anisotropy, значение первого очень небольшое, 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 подвергнутьанимации.



Посмотрите, какие шикарные блики дает этот материал, потратьте немного лишнего времени на изучение его параметров и вы будете вознаграждены.

Глава 48.
Создание стеклянных и металлических поверхностей

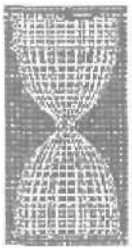
Давайте нарисуем обычные песочные часы.

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 рендереров-рейтрейсеров (**RayMax**, **RayGun**).

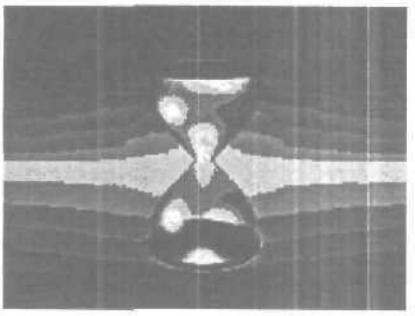
Начнем с примера использования **RayTrace.**

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 рендерера. 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 рендерера вам нужно зайти и раздел **File-Preferences и там в** закладке **Render** найти пункт Assign и выбрать необходимый вам рендерер (при этом он должен быть подключен — находиться в каталоге Plugins).

В панели настройки рендерера 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 рендерера и процедура подключения практически такие же. Единственная особенность рендерера — он значительно быстрее предыдущего, хотя и картинки у него не отличаются высоким качеством.

Имеется возможность указать тип материала (Glass, Mctall), 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).

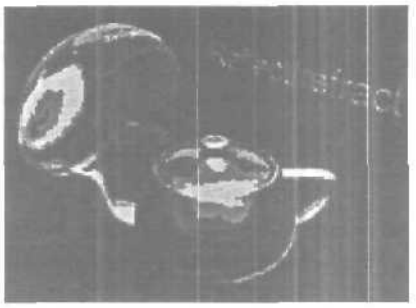
Глава 49.
Три способа создания зеркал

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 веши, как bump, 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, 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.

2 способ — Flat Mirror

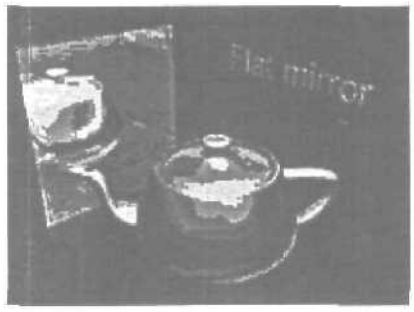
Этот способ применим только для идеально ровных поверхностей. Если у вас хотя бы на одну миллионную координата сдвинута — уже не сработает- К тому же, способ не так прост, как первый.

Итак, приступим. Создайте Box. Материал нам понадобится не простой, а многокомпонентный, а точнее двухкомпонентный. Создайте новый материал Multi **O Sub-Object.** Сделайте Set **Number** и укажите два компонента. Нам надо создать два простых материала: один — собствен-

но 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). Теперь самое интересное. Берете свой Box, и добавляете модификатор Edit Mesh (можно также сконвертить Box в Editable Mesh). Теперь переключаете Sub-Object на Face и выделяете поверхность.

Теперь ищите Material ID и вводите туда 2. Все — мы готовы к рендерингу.



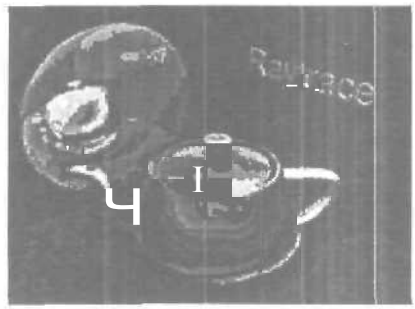
Хорошее качество изображения — пожалуй, почти идеальное. Но только для идеально плоских поверхностей- Метод необходим там, где нужны точные, качественные и самое главное быстро читающиеся зеркала. По сравнению с рейтрейсингом, этот метод дает очень большой выигрыш в быстродействии.

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) 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 сферическом зеркале. Сделайте сферу и расположите его примерно так, как показано на картинке.



Создайте новый материал и в Reflection Map задаем новый тар — Raytrace (процесс аналогичен заданию Reflect/Refract из первого способа). Присвойте материал объекту и ренедерите.

Идеальное качество изображений. Очень низкая скорость просчета — самая низкая из всех методов. Проблема в том, что встроенный рейтресер 3ds max'a очень и очень медленный по сравнению со своими аналогами.

Приложения

Расширения 3ds max

Вся система 3ds max состоит из plug-in'ов. Что такое plug-in? **Plug-in** — это программный модуль 3ds max, который позволяет добавить какую-то новую функцию в основную программу. Основная программа 3ds max (так называемая core) писалась с расчетом на то, что все действия будут делать plug-in'ы.

Взглянем поближе на директорию 3ds **max** (будем считать, что она установлена у вас в \3dsmax). Вы увидите три поддиректории, относящиеся к plug-in'ам:

- **STDPLUGS** — в этой директории хранятся так называемые стандартные plug-in'bi. Они служат, например, для рисования примитивов, импорта файлов для текстур.
- **PLUGINS** — здесь хранятся нестандартные plug-in'bi, например, написанные пользователями. Они позволяют расширить диапазон функций 3ds max.
- PLUGCFG — здесь обычно лежат файлы конфигурации plug-in'ов. В основном туда их кладут стандартные plug-in'bi.

Как выглядят plug-in'bi? Plug-in поставляется в виде одного (реже - нескольких) файлов в расширениями .DL? и некоторыми другими. Это переименованный обычный DLL Windows. По расширению, данному файлу можно определить тип plug-in'a:

- BMF — (BitMap Filter) Фильтры файлов графики.
- BMI — (BitMap Import) Поддержка форматов файлов графики.
- DLC — (Controller) Анимационный контроллер.
- DLE — (Export) Экспорт сцен в другие форматы.
- DLF — (Font) Возможность использовать другие типы шрифтов.
- DLI — (Import) Импорт снен из других форматов.

- DLM — (Modifier) Новый модификатор.
- DLO — (Object) Создает объекты.
- DLR — (Render) Рендеринг и объекты рендеринга.
- DLS — (Shape) Все, связанное со сплайнами.
- DLT — (Texture) Процедурные текстуры и material editor,
- DLU — (Utility) MAX-утилиты.
- FLT — (FiLTer) VideoPost-фильтр.

Ссылки на ресурсы Internet

http:\ \ www.cray.onego.ru/3d

Русский 3D Центр. Все о 3D: вебринг, статьи, советы, галерея, тесты железа.

http:\ \ www.sim.ru/lightwave3d

Русскоязычный сайт по LightWave.

http:\ \ www.newtekniques.com/TekTicker

Ежедневные новости от журнала NewTekniques.

http:\ \ www.animationartist.com

Он-лайновый журнал Animation Artist. Обновляется ежедневно: 2d/3d новости, туторы, интервью, книги...

http:\ \ www.digitalproducer.com

Он-лайновый журнал Digital Producer.

http:\ \ www.3dup.com

Мощный специализированный на 3d поисковик. Сюда попал из-за своего раздела Spotlights, обозревающего последние 3d новости.

http:\ \ www. VFXPfo.com

VFXPro — ежедневные новости в области визуальных эффектов (FX): новый софт, хард, фильмы, новости компаний, предложения работы.

http:\\www.3dartist.com

Он-лайновый вариант журнала «3D Artists». Основное направление — технологии создания трехмерных сцен, обсуждение новых пакетов и инструментов.

http:\\www.cgw.com

Он-лайновый вариант журнала «Computer Graphics World», отдающий предпочтение таким темам, как CAD, анимация, визуализация и мультимедиа.

http:\\www.serious3d.com

Журнал профессиональных туторов «Serious 3D». Все очень интересно, но практически все только за деньги.

http:\\www.visualmagic.awn.com

Журнал визуальных эффектов «Visual Magic». Периодически отсюда можно скачать (в PDF формате) полную версию одноименного бумажного журнала.

http:\\www.625-net.com

Журнал «625». Не совсем о 3d, но зато по-русски.

http:\\www.3dcafe.com

Один из самых популярных 3d сайтов, где можно найти очень много полезных вещей. Все, что можно взять бесплатно (туторы, текстуры, модели, шрифты, звуки...) в разделе FREE STUFF.

http:\\www.3dark.com

Еще один ведущий 3d сайт. Масса разнообразной информации по 3d.

http:\\www.members.xoom.com/surender99

Сайт поддержки турнира SU.RENDER 99 между подписчиками эхоконференции FIDO SU.RENDER.

http:\\www.raph.com/3dartists

Галерея «3d Artists», самая большая 3d галерея.

http:\\www.avalon1.viewpoint.com

Одна из самых больших библиотек моделей, текстур и утилит.

Список использованной литературы

Золотое сечение: Три взгляда на природу гармонии
Шевелев И. Ш., Марутаев М. А , Шмелев И. П.

Золотое сечение в живописи
Ковалев Ф.В.

Коды золотой пропорции
Стахов А.

Advanced 3D Realism
Bill Fleming

Анимация в компьютерных играх
Алексей Орлов

3ds max 6
Сергей Бондаренко, Марина Двораковская

3D Studio MAX
Сергей Гашников

3D Studio MAX
Виталий Башкатов

3D Studio MAX
Олег Татарников

Ландшафт
Михаил Якшин

Mountain Texturing
Юрий Стоцкий

Текстура ландшафта
Михаил Якшин

Море
Виталий Башкатов

Взрыв
Михаил Шатилов

Лучи лазера
МихаилЯкшин

Создание фонтана
Виталий Башкатов

Зеркальные поверхности
Михаил Якшин

Растущий текст
Михаил Якшин

Plug-in'ы 3ds max
Михаил Якшин

Волосатая дубинка
Виталий Башкатов

Shag: Fur
Михаил Якшин

Создание текстур
Михаил Якшин

Анизотропные отражения
Перевод: Александр Ергашов alish@maiUka.ru

3D-живность
Максим Зиннатов

Моделирование головы с использованием сплайнов
Виталий Македонский

Простая голова
Чернышев Петр

Создание поверхности моря
Виталий Башкатов

Море, волны...
Steve Johnson

Подводный пейзаж
Дмитрий Шляхтенко

Ландшафт
Аверин Александр

Сварка
Владимир Шуменко

Брилианты, рубины, аметисты.... или как делать бижутерию
Виталий Башкатов

Настоящая молния
Дмитрий Шляхтенко

Фотонная пушка
Дмитрий Шляхтенко

Создание реалистичного лазера
Виталий Македонский

Стеклянные и металлические поверхности
Аверин Александр

Содержание

Часть 1. Введение в трехмерный дизайн

Глава 1. Основы гармонии в формах . . . 3
Глава 2. Основы реализма в 3D графике . . . 10
Глава 3. Основы рендеринга . . . 19
Глава 4. Общие принципы построения композиции . . . 2S
Глава 5. Создание естественного природного окружения . . . 30
Глава 6. Ставим свет . . . 35
Глава 7. Принципы классической мультипликации . . . 38

Часть 2. Знакомство с новой версией

Глава 1. Итак, 3ds max 6 . . . 47
Глава 2. Поставка . . . 47
Глава 3. Интерфейс . . . 48
Глава 4. Объекты . . . 48
Глава 5. Модификаторы . . . 51
Глава 6. Particle Flow . . . 51
Глава 7. Материалы . . . 53
Глава 8. Mental Ray . . . 56
Глава 9. Reactor 2 . . . 57
Глава 10. Другие нововведения . . . 59

Часть 3. Учимся мастерству

Глава 1. Создание развевающегося на ветру флага . . . 61
Глава 2. Ползающая многоножка . . . 63
Глава 3. Основы инверсной кинематики . . . 70
Глава 4. Инверсная кинематика - урок второй . . . 74
Глава 5. Сигаретный дым . . . 88
Глава 6. Создание VRML . . . 91

Глава 7. Морской пейзаж 98
Глава 8. Пузырь, еще пузырь 101
Глава 9. Спиральная галактика 106
Глава 10. Создание модели скрипки методом NURBS моделирования . . . . . ПО
Глава П. Футбольный мяч 116
Глава 12. Объединение вершин (welding) 122
Глава 13. Взрывная волна 127
Глава 14. Использование фильтра Highlight 136
Глава 15. Создание модели игрушечной собачонки 145
Глава 16. Интеграция 3D в фотографию 154
Глава 17. Бокал пива 161
Глава 18. Объекты в фокусе камеры 170
Глава 19. Эффект Flow-Mo В 3ds max 178
Глава 20. Еще раз о 3D и фотографии 181
Глава 21. Создание текстур в DEEP PAINT 3D 188
Глава 22. Пара шагов к финальному рендеру 197
Глава 23. Как нарисовать книгу 201
Глава 24. Как сделать реалистичный взрыв 202
Глава 25. Как сделать параллельные лучи света 207
Глава 26. Создание фонтана 209
Глава 27. Создание зеркальных поверхностей 211
Глава 28. Эффект растущего текста 215
Глава 29. Создание ландшафтов - Mountain Texturing 217
Глава 30. Использование Photoshop для создания текстур 224
Глава 31. Анизотропные отражения 230
Глава 32. Создание 3D-живности 243
Глава 33. Создание правильной гайки с помощью лофта Rhino 248
Глава 34. Создание волос с применением plug-in Shag: Fur 250
Глава 35. Создание простой головы 253
Глава 36. Моделирование головы с использованием Surface Tools 258
Глава 37. Создание подводных эффектов 263

Глава 38. Как сделать облака, море и волны 269
Глава 39. Объемный свет на примере мозаичного окна 277
Глава 40. Бриллианты, рубины, аметисты... или как делать бижутерию 280
Глава 41. Создание молнии 283
Глава 42. Создание фотонной пушки 286
Глава 43. Простой способ создания пара 287
Глава 44. Создание реалистичного лазера 289
Глава 45. Создание космического эффекта - солнце 293
Глава 46. Имитация работы сварочного аппарата 294
Глава 47. Тонкости и хитрости создания металлических поверхностей 298
Глава 48. Создание **стеклянных** и металлических поверхностей 305
Глава 49. Три способа создания зеркал 307

Приложения

Расширения 3ds max 311
Ссылки на ресурсы Internet 312
Список использованной литературы 314

Научно-популярное издание

Серия книг
«Ваш персональный компьютер»

Якушев Дмитрий Михайлович

3D Studio MAX 6.0
Все, что Вы хотели знать, но боялись спросить

Оригинал-макет ***И. В. Царик***
Художник ***О. К. Алехин***
Художественный редактор *М. Л. Мишин*
Технический редактор *К. В. Шапиро*
Корректоры *Л. С. Зимина, К. В. Толкачева*

Подписано в печать с диапозитивов 10.12.2003. Формат 60×90/16.
Бумага офсетная. Печать офсетная. Усл печ. л. 20. Тираж 1000 экз.


9 785832 106021