Оглавление

Часть II. Низкоуровневые системы движка

Часть III. Графика, движение и звук

Часть IV. Игровой процесс

Часть V. Подведение итогов

12 Системы анимации

В большинстве современных 3D-игр центральную роль играют *персонажи* — часто это люди или человекоподобные существа, иногда животные или монстры. Отличительной чертой персонажей является то, что они должны двигаться плавно и органично. Это порождает множество дополнительных технических проблем, помимо тех, которые требуется решить в случае моделирования и анимации таких твердых объектов, как транспортные средства, снаряды, футбольные мячи и фигуры тетриса. Задачу наделения персонажей естественно выглядящими движениями решает компонент движка, называемый *системой анимации персонажей*.

Как мы увидим далее, система анимации предоставляет разработчикам игр мощный набор инструментов, которые можно применять не только к персонажам, но и к другим объектам. Возможности системы анимации могут с успехом задействоваться в любом мало-мальски подвижном объекте игры. Поэтому, когда вы видите в игре транспортное средство с подвижными частями, шарнирно сочлененный механизм, слегка покачивающиеся на ветру деревья или даже взрывающееся здание, велика вероятность того, что в этом объекте хотя бы отчасти используется анимационная система игрового движка.

12.1. Разновидности анимации персонажей

Технологии анимации персонажей прошли долгий путь развития со времен серии игр *Donkey Kong*. В первых играх использовались простейшие методы имитации движений живых существ. По мере роста возможностей игрового аппаратного обеспечения стало возможным применение более сложных методов в режиме реального времени.

Сегодня в распоряжении разработчиков игр имеется множество мощных методов анимации. В этом разделе мы кратко коснемся эволюции анимации персонажей и в общих чертах рассмотрим три метода, которые получили наибольшее распространение в современных игровых движках.

12.1.1. Келевая анимация

Предшественником всех методов игровой анимации является *традиционная рисованная анимация*. Речь идет о методе, который использовался уже в первых мультипликационных фильмах. Иллюзия движения создается отображением последовательности быстро сменяющихся неподвижных изображений — *кадров*. 3D-рендеринг в реальном времени можно рассматривать как электронную разновидность традиционной анимации, в которой перед зрителем проходит последовательность неподвижных полноэкранных изображений, создавая иллюзию движения.

Особой разновидностью традиционной анимации является *келевая анимация*. *Кель* — это прозрачный лист целлулоида (cel — от слова celluloid), на котором можно рисовать изображения. Размещая анимированную последовательность келей поверх фиксированного фонового рисунка, можно создавать иллюзию движения, не перерисовывая раз за разом статический фон.

Аналогом келевой анимации в компьютерной графике является так называемая *спрайтовая анимация*. Спрайт — это небольшое растровое изображение, которое можно накладывать поверх полноэкранного фонового изображения, не искажая его. Обычно оно отрисовывается с помощью специализированного графического аппаратного обеспечения. То есть в двухмерной игровой анимации спрайт играет ту же роль, которую играл кель в традиционной анимации. Этот метод анимации был основным в эпоху двухмерных игр. На рис. 12.1 показана хорошо известная последовательность спрайтов, применявшаяся для создания иллюзии бегущего гуманоидного персонажа практически во всех играх для приставок Mattel Intellivision. Эта последовательность кадров организована таким образом, чтобы она плавно проигрывалась и при бесконечном повторении, — такой подход называется *циклической анимацией*. Пользуясь современной терминологией, данную анимацию можно назвать *циклом бега*, поскольку она создает иллюзию того, что персонаж бежит. Обычно персонаж имеет несколько циклов анимации: различные циклы бездействия, цикл ходьбы и цикл бега.

Рис. 12.1. Последовательность спрайтов, применявшаяся в большинстве игр для приставок Intellivision

12.1.2. Жесткая иерархическая анимация

Системы анимации первых 3D-игр, таких как *Doom*, незначительно отличались от спрайтовой анимации: монстры в этих играх представляли собой не что иное, как обращенные к камере квадранты, создающие иллюзию движения за счет отображения на них последовательности текстурных растровых изображений, называемых

анимированными текстурами. Эта техника используется и сегодня для объектов с низким разрешением и/или удаленных, таких как толпа на стадионе или множество солдат, сражающихся на заднем плане. Однако развитие трехмерной графики потребовало применения для высококачественных персонажей переднего плана более совершенных методов анимации персонажей.

Изначально для трехмерной анимации персонажей использовали метод *жесткой иерархической анимации*. При этом персонаж моделируется как набор жестких фигур. Так, гуманоидный персонаж обычно разбивается на следующие составные части: таз, торс, верхние и нижние части рук, верхние и нижние части ног, кисти рук, ступни ног, голова. Эти жесткие части связываются друг с другом в иерархическом порядке, подобно тому как соединены в суставах кости млекопитающих животных. Это заставляет персонаж двигаться естественным для него образом. Так, например, когда начинает двигаться верхняя часть руки, за ней автоматически следуют нижняя часть руки и кисть. Обычно в качестве корневого узла иерархии выступает таз, его непосредственными дочерними узлами являются торс и верхние части ног и т. д.:

```
Pelvis
    Torso
        UpperRightArm
            LowerRightArm
                RightHand
        UpperLeftArm
            UpperLeftArm
                LeftHand
        Head
    UpperRightLeg
        LowerRightLeg
            RightFoot
    UpperLeftLeg
        UpperLeftLeg
            LeftFoot
```

Большой проблемой жесткой иерархической анимации является то, что тело персонажа часто начинает выглядеть не очень привлекательно из-за «растрескивания» в суставах (рис. 12.2). Этот метод хорошо подходит для роботов и машин, которые действительно состоят из жестких компонентов, но не выдерживает никакой критики в случае его применения к персонажам, которые хоть в какой-то мере обладают плотью.

12.1.3. Повершинная анимация и морфинг-мишени

Жесткая иерархическая анимация, часто выглядит неестественно именно в силу своей жесткости. Поэтому нужно перемещать отдельные вершины таким образом, чтобы треугольники могли растягиваться, обеспечивая более естественный вид движений.

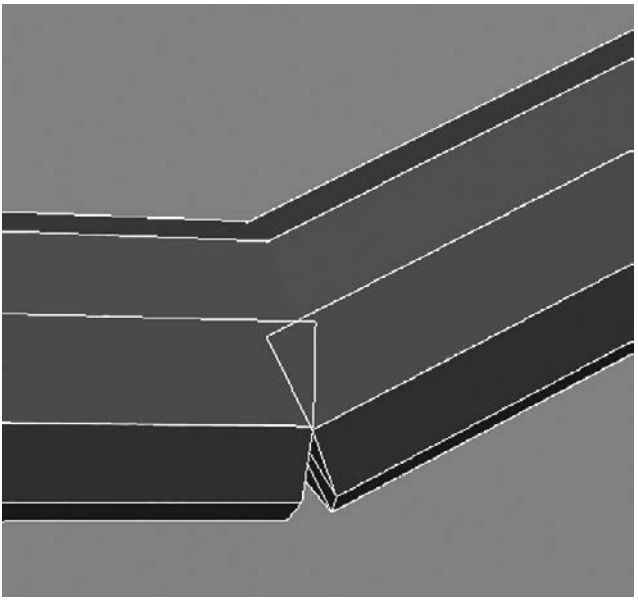

Рис. 12.2. «Растрескивание» в суставах — большая проблема жесткой иерархической анимации

Помимо других способов, для этого можно использовать такой прямолинейный метод, как *повершинная анимация*. При этом вершины сетки анимируются художником, после чего данные о движении экспортируются и используются в качестве указания игровому движку о том, как следует перемещать каждую из них в динамическом режиме. Данный метод позволяет деформировать сетку практически любым вообразимым способом (ограничением является лишь степень тесселяции поверхности). Однако это информационно емкий метод, поскольку для каждой вершины сетки требуется сохранять изменяющиеся во времени данные о движении. В силу этого он практически не применяется в играх в режиме реального времени.

В некоторых играх в режиме реального времени используется разновидность этой техники, известная как *анимация по морфинг-мишеням*. При этом аниматор перемещает вершины сетки, задействуя сравнительно небольшой набор фиксированных крайних поз. Анимация выполняется *смешиванием* двух или большего количества фиксированных поз в динамическом режиме. Положение каждой вершины рассчитывается посредством простой линейной интерполяции (linear interpolation, LERP) между положениями вершины в каждой из крайних поз.

Метод морфинг-мишеней часто используется для лицевой анимации, потому что у человеческого лица чрезвычайно сложная анатомия и оно приводится в движение примерно 50 мышцами. Это дает аниматору полный контроль над каждой

Рис. 12.3. Набор лицевых морфинг-мишеней для персонажа Элли из игры The Last of Us: Remastered (снимок экрана с сайта https://www.gameenginebook.com)

вершиной лицевой сетки, позволяя реализовывать как едва уловимые, так и очень большие движения, хорошо имитирующие мускулатуру лица. На рис. 12.3 показано, как может выглядеть набор лицевых морфинг-мишеней.

По мере роста вычислительных мощностей некоторые студии вместо морфинг-мишеней начали использовать суставные лицевые риги, включающие в себя сотни суставов. Другие студии сочетают оба метода, создавая базовое выражение лица с помощью суставных ригов и внося небольшие изменения с помощью морфинг-мишеней.

12.1.4. Скиновая анимация

По мере дальнейшего роста возможностей игрового аппаратного обеспечения был разработан метод *скиновой анимации*. Он обладает почти всеми плюсами поверхностной анимации и анимации по морфинг-мишеням, позволяя деформировать треугольники анимируемой сетки. Все это сочетается с намного более высокой эффективностью в плане производительности и потребления памяти, характерной для жесткой иерархической анимации. С помощью этого метода можно довольно реалистично имитировать движение кожи и одежды.

Скиновая анимация была впервые применена в таких играх, как *Super Mario 64*, и по сей день остается наиболее распространенным методом анимации как в индустрии компьютерных игр, так и в кино. Персонажи многих известных современных игр и кинофильмов как минимум частично, а то и полностью анимировались с помощью этого метода. Это, в частности, динозавры из фильма «Парк юрского периода», Солид Снейк из серии игр *Metal Gear*, Голлум из фильма «Властелин колец», Натан Дрейк из игры *Uncharted*, Базз Лайтер из мультфильма «История игрушек», Маркус Феникс из серии игр *Gears of War* и Джоэл из серии игр *The Last of Us*. Далее в этой главе мы сосредоточимся главным образом на изучении скиновой/скелетной анимации.

Как и в жесткой иерархической анимации, в скиновой анимации составляется *скелет* из жестких костей. Но вместо того, чтобы отрисовывать эти жесткие части на экране, их оставляют скрытыми. К суставам скелета привязывается так называемый скин — плавная непрерывная триангулярная сетка, вершины которой отслеживают движения суставов. Для каждой вершины можно определить весовые коэффициенты влияния нескольких суставов, что обеспечивает реалистичное растяжение скина при их движении.

На рис. 12.4 представлен игровой персонаж Горностай Крэнк *(Crank the Weasel)*, созданный Эриком Браунингом в 2001 году для компании Midway Home Entertainment. Внешняя оболочка, или скин, Крэнка состоит из сетки треуголь-

ников, как и любая другая 3D-модель. А внутри него мы видим жесткие кости и суставы, которые приводят скин в движение.

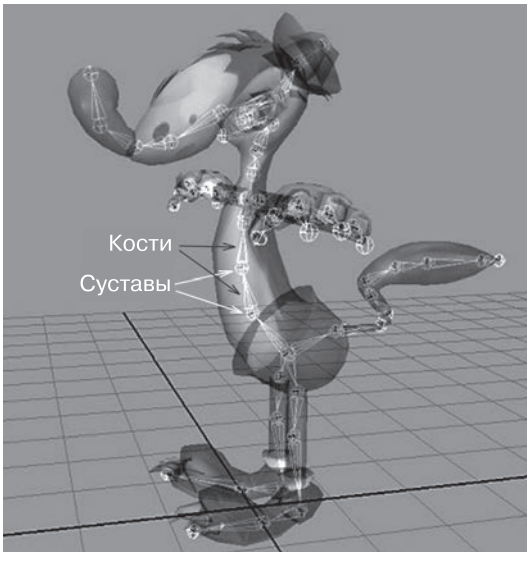

Рис. 12.4. Горностай Крэнк с внутренней скелетной структурой

12.1.5. Методы анимации как методы сжатия данных

В идеализированно гибкой системе анимации у аниматора будет контроль буквально над каждым бесконечно малым участком поверхности объекта. Конечно, такой подход приведет к созданию анимации, потенциально содержащей бесконечно большой объем данных. Анимация вершин триангулярной сетки является, по сути, упрощением этого идеального случая — мы *сжимаем* информацию, необходимую для описания анимации, ограничиваясь только возможностью перемещать вершины. (Анимация набора контрольных точек представляет собой аналог вершинной анимации для моделей, состоящих из патчей более высокого порядка.) Морфинг-мишени можно рассматривать как дополнительный уровень сжатия, обеспечиваемый за счет наложения на систему дополнительных ограничений, допускающих перемещение вершин только по линейным траекториям между фиксированным количеством заранее заданных положений. Скелетная анимация, по сути, представляет собой еще один способ сжатия данных вершинной анимации путем наложения ограничений. В этом случае ограничение состоит в том, что сравнительно большое количество вершин должно отслеживать движения сравнительно небольшого количества суставов скелета.

Решая, какой из методов анимации лучше выбрать в том или ином случае, часто полезно думать о них как о методах сжатия, во многом аналогичных методам

сжатия видео. В общем случае нужно выбирать метод анимации, который обеспечивает наилучшее сжатие, не создавая неприемлемых визуальных артефактов. Скелетная анимация дает наилучшее сжатие, когда движение одного сустава транслируется в движение множества вершин. Так, скелет позволяет очень эффективно перемещать конечности персонажа, поскольку они ведут себя почти как твердые тела.

Однако движения лица обычно намного сложнее и требуют более независимого перемещения отдельных вершин. Для того чтобы скелетный метод обеспечивал достаточно реалистичную анимацию лица, количество используемых суставов должно быть ненамного меньше количества вершин в сетке, что снижает его эффективность в качестве метода сжатия. Это один из главных доводов в пользу того, чтобы использовать для анимации лица метод морфинг-мишеней, а не скелетный подход. (Другой веский довод таков: применение морфинг-мишеней обычно более привычный для аниматоров способ работы.)

12.2. Скелеты

Скелет состоит из *иерархии* жестких элементов, называемых *суставами*. Хотя в игровой индустрии под суставами и костями часто понимают одно и то же, такое употребление слова «кости» не совсем корректно. Если судить с технической точки зрения, то суставы — это объекты, которыми напрямую манипулирует аниматор, а кости — пустое пространство между ними. В качестве примера рассмотрим тазобедренный сустав модели Горностая Крэнка. Он соединен с четырьмя другими суставами (в хвосте, позвоночнике и двух ногах) и в силу этого образует четыре выходящие из него кости (рис. 12.5). Игровые движки не обращают никакого

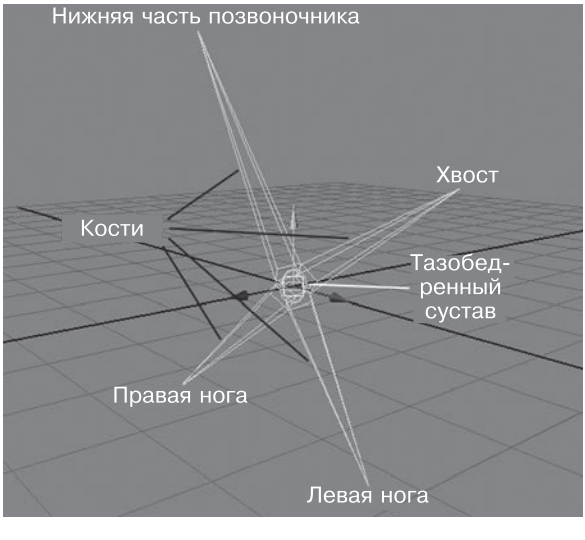

Рис. 12.5. Тазобедренный сустав персонажа соединен с четырьмя другими суставами (в хвосте, позвоночнике и двух ногах) и, соответственно, образует четыре кости

внимания на кости — для них важны только суставы. Поэтому, когда вы слышите слово «кость» в контексте игровой индустрии, знайте, что при этом почти всегда имеются в виду суставы.

12.2.1. Скелетная иерархия

Как мы уже упоминали, суставы в скелете образуют иерархию или древовидную структуру. Один сустав выбирается корневым, а все остальные становятся по отношению к нему дочерними, внучатыми и т. д. Типичная иерархия суставов для скиновой анимации выглядит практически так же, как типичная иерархия для жесткой иерархической анимации. Так, иерархия суставов гуманоидного персонажа может выглядеть примерно так, как показано на рис. 12.6.

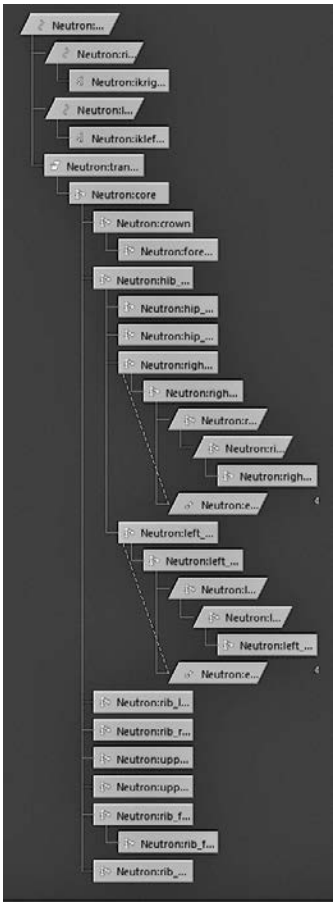

Рис. 12.6. Пример скелетной иерархии, представленной в виде гиперграфа в приложении Maya

Любому суставу присваивается индекс в диапазоне от 0 до $N - 1$. Поскольку каждый сустав имеет один и только один родительский сустав, иерархическая структура скелета может быть полностью определена путем сохранения в каждом суставе индекса его родительского сустава. Так как у корневого сустава нет родительского сустава, его индексу родительского сустава обычно присваивается недопустимое значение, например –1.

12.2.2. Представление скелета в памяти

Для представления скелета обычно используется небольшая структура данных верхнего уровня, содержащая массив структур данных отдельных суставов. Чаще всего выбирается такой порядок следования суставов в массиве, при котором дочерний сустав располагается после родительского. Это означает, что корнем скелета всегда является нулевой сустав.

Для обращения к отдельным суставам в пределах структуры данных обычно используются *индексы суставов*. Дочерний сустав ссылается на родительский, указывая его индекс. Аналогичным образом в заскиненной триангулярной сетке вершина ссылается по индексу на суставы, к которым привязана. Это гораздо эффективнее, чем ссылаться на суставы по имени, с точки зрения как потребления памяти (если принять максимальное число суставов в скелете равным 256, индекс сустава может занимать всего 8 бит), так и затрат времени на поиск нужного сустава (мы можем мгновенно перейти к нужному суставу в массиве, используя его индекс).

Структуры данных отдельных суставов обычно содержат следующую информацию:

- *имя* сустава в виде строки либо ее хэшированного 32-битного идентификатора;
- индекс родительского сустава в скелете;
- *обратную матрицу преобразования позы привязки* для данного сустава. Поза привязки сустава представляет собой данные о том, какими были положение, ориентация и масштаб этого сустава в момент его связывания с вершинами сетки скина. По причинам, которые будут подробно рассмотрены в последующих разделах, обычно сохраняется *обратная матрица* этого преобразования.

Типичная структура данных скелета может выглядеть примерно так:

```
struct Joint
{
    Matrix4x3   m_invBindPose; // обратная матрица преобразования позы привязки
    const char* m_name;        // удобное для чтения человеком имя сустава
    U8          m_iParent;     // индекс родительского сустава или 0xFF
                               // в случае корневого сустава
};

struct Skeleton
{
    U32     m_jointCount;  // количество суставов
    Joint*  m_aJoint;      // массив суставов
};
```

12.3. Позы и положения

Какой бы подход ни использовался для создания анимации (жесткий иерархический или скиновый/скелетный), она всегда воспроизводится постепенно. Чтобы создать иллюзию движения, тело персонажа последовательно принимает статические *позы*, которые быстро чередуются на экране обычно с частотой 30 или 60 *поз в секунду* (на самом деле, как вы увидите в подразделе 12.4.1, вместо строгого отображения каждой позы часто используют *интерполяцию* между соседними позами). В скелетной анимации поза скелета напрямую управляет вершинами меша, а позиционирование является основным инструментом аниматора, с помощью которого тот вдыхает жизнь в своих персонажей. Поэтому очевидно, что прежде, чем мы сможем анимировать скелет, нужно понять, как придать ему *позу*.

Скелет позиционируется за счет произвольного вращения, перемещения и иногда масштабирования его суставов. *Положение* сустава определяется его положением, направлением и масштабом относительно какой-то системы отсчета. Положение сустава обычно представлено в виде матрицы размерностью 4×4 или 4×3 или же структуры данных SRT (scale, quaternion rotation, vector translation — масштаб, кватернионное вращение, векторное перемещение). Поза скелета — это просто набор положений всех его суставов, который обычно имеет вид массива матриц или структур SRT.

12.3.1. Поза привязки

На рис. 12.7 показаны две позы одного и того же скелета. Слева мы видим специальную *позу привязки*, которую иногда называют *позой отсчета* или *позой покоя*. Это состояние трехмерного меша, в котором он находится перед привязкой к скелету (отсюда и название). Иными словами, это поза, которую принял бы обычный, нескиновый треугольный меш без какого-либо скелета. Эту позу также называют *Т-образной*, поскольку персонаж напоминает по форме букву «Т» — стоит со слегка расставленными ногами и вытянутыми руками. Эта стойка была выбрана потому, что в ней конечности отдалены от туловища и друг друга, что облегчает процесс привязки вершин к суставам.

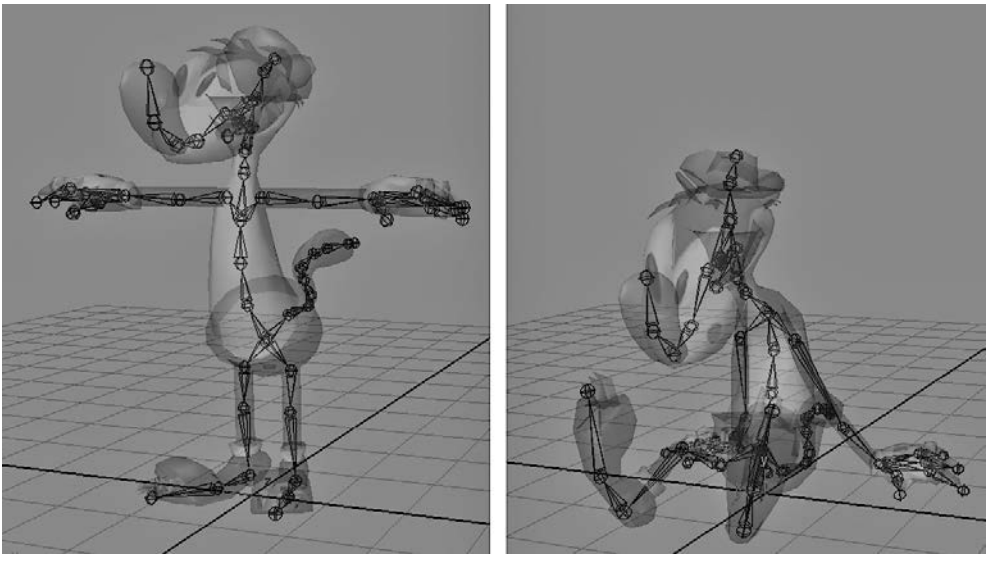

Рис. 12.7. Две позы одного и того же скелета. Слева показана поза привязки

12.3.2. Локальные позы

Положения дочерних суставов чаще всего указываются относительно *родительских*. Это обеспечивает естественное движение суставов. Например, если мы повернем плечевой сустав, а положения локтя, запястья и пальцев оставим без изменений, вокруг плеча жестко повернется вся рука, как и ожидалось. Позу или положение, заданные относительно родительского сустава, иногда называют *локальными*. Локальные позы почти всегда хранятся в формате SRT. Причины этого мы исследуем при обсуждении слияния анимации.

Многие пакеты 3D-моделирования, такие как Maya, отображают суставы в виде небольших сфер. Однако, помимо положения, у сустава также есть поворот и масштаб, поэтому его визуализация может быть немного обманчивой. На самом деле

сустав определяет координатное пространство, которое принципиально ничем не отличается от других пространств, рассмотренных ранее, таких как пространство модели, пространство игрового мира или пространство просмотра. Поэтому его лучше представить себе в виде набора осей в прямоугольной системе координат. Maya дает пользователю возможность отображать локальные оси координат сустава (рис. 12.8).

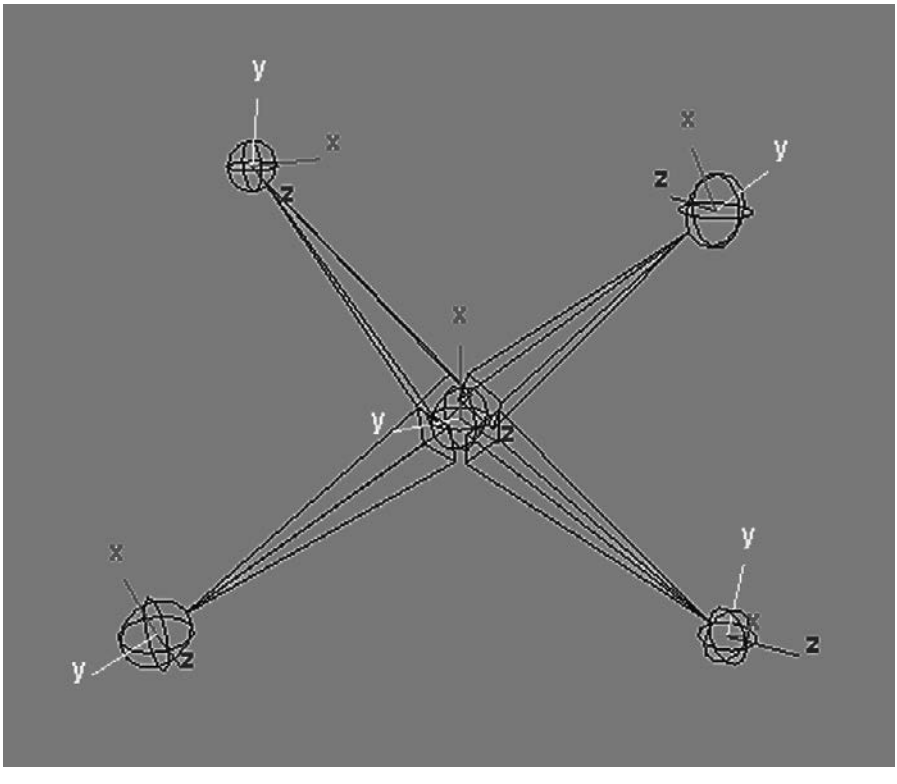

Рис. 12.8. Каждый сустав в скелетной иерархии определяет оси координат в локальном пространстве, известном как пространство сустава

В математическом смысле положение сустава — не что иное, как *аффинное преобразование*. Положение сустава j можно записать в виде матрицы преобразования $\mathbf{P}_j$ размером 4×4, которая состоит из вектора преобразования $\mathbf{T}_j$, диагональной матрицы масштабирования $\mathbf{S}_j$ размером 3×3 и матрицы вращения $\mathbf{R}_j$ того же размера. Позу всего скелета $\mathbf{P}_{\text{skel}}$ можно записать в виде набора всех поз $\mathbf{P}_j$, где j находится в диапазоне от 0 до $N - 1$:

$$\mathbf{P}_j = \begin{bmatrix} \mathbf{S}_j\mathbf{R}_j & 0 \\ \mathbf{T}_j & 0 \end{bmatrix}; \; \mathbf{P}_{\text{skel}} = \left\{\mathbf{P}_j\right\}\Big|_{j=0}^{N-1}.$$