

СОЗДАНИЕ СОБСТВЕННОГО СКЕЛЕТА В 3DS MAX

В данной статье авторы производят риггинг части тела персонажа на основе собственных скриптов для создания гибкой анимации и корректной работы моделей и текстур.

ВВЕДЕНИЕ

В современном мире компьютерные игры предлагают игроку не только интересную историю и захватывающий геймплей, но и детально проработанных персонажей и моделей окружения. Создание эффекта погружения игрока должно производиться путем создания впечатляющих моделей и соответствующих им анимаций. Чтобы персонажи ожили и рассказали зрителю свою историю, предстоит проделать кропотливую работу по созданию нужного скелета, благодаря которому можно создать не только реалистичную анимацию, но и воплотить самые безумные идеи. Целью данного исследования является изучение существующих скриптов и инструментов в 3Ds Max, на основе которых можно создать собственный скелет.

I. СОЗДАНИЕ РАСТЯГИВАЮЩЕЙСЯ КОСТИ

Данный прием отлично подходит для подготовки к анимации карикатурных персонажей. Для того, чтобы расширить возможности системы ИК (инверсной кинематики) в отношении автоматического растягивания костей при перемещении ИК Goal понадобится написать новый сценарий MAXScript.

1. Построение ноги

Для построения ноги необходим скелет из костей, которые находятся в окне Bone Tools, вызываемый из меню Animation. Во фронтальной проекции они располагаются таким образом, как показано на рисунке 1.

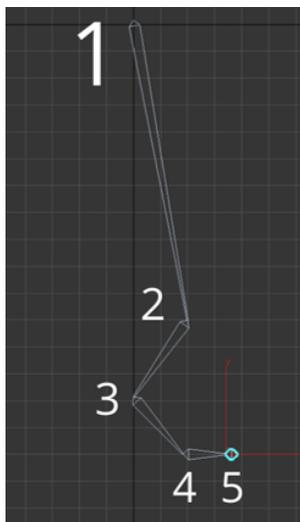


Рис. 1 – Расположение системы костей

2. Назначение инверсной кинематики (ИК)

Для зависимости всех костей между собой создается цепочка инверсной кинематики (IK Chain): Animation->ИК Solvers->HI Solver, затем последовательно выбираются кости Bone005 и Bone001. Таким образом, выстраивается иерархия, при которой изменение положения одной кости ведет к изменению положения всей цепочки.

3. Процесс сетапа.

Для определения положения кости Bone002 следует назначить контроллер XYZ Position на параметр Position для дальнейшего использования в скрипте: Graph Editors->Track View->Position->ПКМ->Assign Controller. Далее из списка выбирается контроллер Position XYZ и нажимается кнопка Make Default. Такая же процедура производится для кости Bone003. Теперь нужно определить максимальное расстояние между ИК Goal и Bone001, после которого кости будут начинать растягиваться. ИК Goal перемещается вниз до тех пор, пока нога не станет совершенно прямой. В окне MAXScript Listener (F11) записывается следующее выражение: `distance IKChain001.posBone001.pos` (вычисляется максимальная длина кости в разогнутом состоянии). Полученное числовое значение будет использовано в третьей строке сценария MAXScript.

II. СОЗДАНИЕ СКРИПТА ДЛЯ РАСТЯЖЕНИЯ КОСТИ

Выделяя кость Bone002 и назначая на позиции X Position контроллер Float Script, появляется окно редактора Script Controller. Значение, которое записано в окне редактора, понадобится для четвертой строки скрипта.

Сам скрипт имеет следующий вид:

```
dependsOn $IKChain001.pos.controller
$Bone001.pos.controller
d = distance $IKChain001.pos
$Bone001.pos
ex = (d - 184.612)
orig = 111.804
if (ex < 0) then ex = 0
p = orig + (ex/2)
```

В первых двух строках сценария MAXScript используется команда `dependsOn`. Она позволяет сообщить 3ds max'у о необходимости выполнения расчета по данному сценарию в зависимости от изменения любого числа параметров объекта. Таким образом, пересчет этого сценария осуществляется при изменении положения цели ИК и кости Bone001.

В третьей и четвертой строках, как и прежде, измеряется расстояние между целью ИК и костью Bone001. Полученное значение присваивается соответствующей переменной.

В пятой строке расстояние, на которое растянута нога, вычитается из текущего расстояния, и полученное значение присваивается соответствующей переменной.

В шестой строке значение исходного положения кости Bone001 по оси X присваивается соответствующей переменной.

В седьмой строке проверяется условие отсутствия растягивания ноги, и, если это условие истинно, значение расстояния, на которое растянута нога, устанавливается равным нулю.

В восьмой, последней строке скрипта, фактически устанавливается значение положения по оси X, которое и определяет растягивание ноги. Для получения равномерного растягивания костей (этот скрипт будет повторно использован для кости Bone003) значение исходного положения по оси X складывается со значением растягивания, деленного на 2.

После того, как сценарий написан, кнопка Evaluate (Вычислить) запускает скрипт.

Далее такая же процедура производится для кости Bone003 (процедура назначения контроллера Float Script), чтобы растягивалась кость Bone002. При этом значение orig должно быть равным значению, которое появляется в окне редактора Script Controller при выделении кости Bone003. Остальной скрипт остается без изменений.

Результат проделанной работы отображен на рисунке 2.



Рис. 2 – Поведение системы костей после применения скрипта

Скоморох Кристина Сергеевна, студент 3 курса факультета информационных технологий и управления БГУИР, skomorokh03@gmail.com.

Киватыцкая Дарья Игоревна, студент 3 курса факультета информационных технологий и управления БГУИР, keksfbrgs@gmail.com.

Научный руководитель: Кукин Дмитрий Петрович, кандидат технических наук, доцент, заведующий кафедрой вычислительных методов и программирования БГУИР, kukin@bsuir.by.

III. ПРИМЕНЕНИЕ ПОЛУЧЕННОГО СКЕЛЕТА К МОДЕЛИ

Для демонстрации результата работы скрипта к модели ноги применяется модификатор skin, который позволяет соединить скелет с моделью. Таким образом, растяжение костей приводит к растяжению ноги (см. рисунок 3).

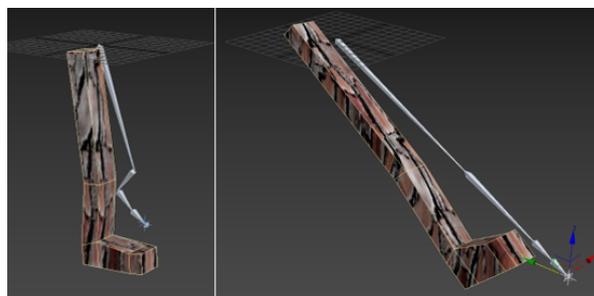


Рис. 3 – Результат проделанной работы

Таким образом, при растяжении ноги можно заметить, что наложенная текстура не искажается, а подстраивается под проекцию Vox, посредством которого наложена текстура (UVW Map).

IV. ВЫВОДЫ

В данной статье был представлен скрипт, который позволяет растягивать кости без вреда для геометрии и наложения текстур, применимый для любой части тела персонажа. Данная функция имеет несколько преимуществ:

1. Создание сюрреалистичных персонажей и анимаций;
2. Наложение неискаженных текстур;
3. Обеспечение плавного растяжения костей.

1. Clinton, Y. Game Character Modeling and Animation with 3ds Max/ Y. Clinton// Autodesk Media and Entertainment. – 2007. – С. 237-301.
2. Изучаем скрипты в Max3 [Electronic resource]/ – Mode of access: <https://render.ru/ru/a.misharin/post/11715?page=4/>. – Date of access: 27.02.2000
3. 3ds Max CAT and MAXScript [Electronic resource]/ – Mode of access: <https://help.autodesk.com/view/MAXDEV/2022/ENU/?guid=GUID-7664D95A-A27D-4356-8442-F53064920BF8/>.