

РЕШЕНИЕ ПРОБЛЕМЫ НАЛИЧИЯ РУТИННОЙ ЗАДАЧИ — СОЗДАНИЯ СЦЕНЫ ДЛЯ ВИЗУАЛИЗАЦИИ В AUTODESK 3 DS MAX

А.В. Шкляр, А.В. Гуменникова

А.В. Шкляр

Томский политехнический университет

gumennikova_aleksandra@mail.ru

Введение

В условиях глобальной компьютеризации, повышения наукоёмкости деятельности расширяются возможности использования компьютерных технологий в области дизайна. В свою очередь, процесс моделирования и визуализации включает в себя разные этапы, которые можно разделить на творческую деятельность и рутинную работу [1]. Рутинная работа заключается в выполнении одинаковой работы в нескольких проектах. При наличии соответствующего инструмента эту работу можно сделать единой, а потом при необходимости использовать ее результат без временных затрат. При моделировании рутинной работой является создание и настройка сцены для визуализации. Визуализация — термин в компьютерной графике, обозначающий процесс получения изображения модели с помощью компьютерной программы.

Целью работы будет являться написание скрипта для создания сцены [2], состоящей из фона, камеры и двух источников освещения и проведение эксперимента, подтверждающего оптимизацию рутинного процесса настройки сцены для визуализации.

Для достижения цели следует выполнить следующие задачи:

1) написать программу на встроенном скриптовом языке в 3Ds Max, включающую в себя фон, камеру и два источника освещения;

2) создать сцену вручную и сравнить скорость данного процесса и процесса использования созданного скрипта.

Написание программы в MAX Script

Первым этапом написания программы будет создание фона, на которой будет находиться смоделированный объект. Для удобства написания программы были обозначены скриптовые блоки, и заданы переменные. Чтобы не создавать лишних полигонов, количество сегментов в длину «bg_ls» будет равняться 1. Для последующего сгибания плоскости количество сегментов по ширине должно иметь достаточно полигонов. В случае, когда фон находится в вертикальной и горизонтальной плоскостях дает пользователю больше удачных ракурсов для установки камеры. применяем модификатор сгибания Bend. Это достигается путем вызова команды «addmodifier» и указания модифицируемого объекта. В скобках указывается угол сгибания, координата, относительно которой будет происходить преобразование (0- X, 1- Y, 2- Z), создание лимита и указание его значения [3].

```
--background
bg_l=140.0
bg_w=140.0
bg_ls=1 as integer
bg_ws=bg_w as integer
BG=plane length: bg_l width: bg_w
lengthsegs: bg_ls widthsegs: bg_ws wirecolor:
white
addmodifier BG (Bend Angle: -90.0 Axis: 0
limit: true UpperLimit: 19.0)
```

В профессиональной системе визуализации Mental Ray существует стандартная система дневного освещения, предназначенная для освещения открытых пространств, имеет небольшое количество настраиваемых параметров и несколько алгоритмов небесного освещения. Для ее создания была введена переменная «SL» и задан параметр позиции. Для создания направленного источника освещения была создана переменная «MAL», которой присвоена операция miAreaLight, задана позиция и положение target. Была выявлена необходимость уменьшения интенсивности освещения в связи с наличием skylight [5].

```
--light
SL=skylight pos: [-70.0,-70.0,0]
MAL= miAreaLight pos: [-140.0,-30.0,150.0]
AreaEnable: true target: (Targetobject transform:(matrix3 [1,0,0] [0,1,0] [0,0,1] [-2.0,0.0,12.0]))
MAL.multiplier = 0.50
MAL.Area_Type =1
MAL.Rectangle_Width =50.0
MAL.Rectangle_Height =50.0
MAL.falloff =150.0
MAL.hotspot =20.0
```

Для последующей комфортной работы со скриптом была создана камера. Пользователь сможет подобрать нужный ракурс объекта сразу после запуска программы или приступить к масштабированию объектов без потери времени, если во вьюпорте будет отображаться автоматически заданный взгляд камеры на цель.

```
--camera
Targetcamera fov: 45 pos: [-126.0,58.0,70.0]
isSelected:on target:(Targetobject transform: (matrix3 [1,0,0] [0,1,0] [0,0,1] [-2.0,0.0,12.0]))
actionMan.executeAction 0 "40068" --
Camera View
```

Дальнейшее работа с элементами сцены будет удобнее, если у всех объектов будет общий пивот (англ. Pivot). Пивот — это опорная точка 3d объекта, которая является центром масштабирования и поворота модели. Для удобства преобразования следует сгруппировать созданные элементы.

```
select objects
actionMan.executeAction 0 "40140" -- Grouping
```

Быстрый доступ к программе позволит значительно сократить время пользования скриптом.

Возможно произвести два инструмента комфортной оперативной работы с данным скриптом: установка горячих клавиш и создание иконки на панели быстрого доступа.

Для установки горячей клавиши телу скрипта назначается команда «macroScript». Для добавления горячей клавиши в Customize User Interface следует выбрать категорию MyScripts и в поле Hotkey написать желаемое сочетание клавиш. Например, Shift+K.

```
macroScript Standart_Render category:"MyScripts" tooltip:"MyScript"
(
  <Текст скрипта>
)
```

Для добавления кнопки на панель быстрого запуска потребуется вызвать Listner, нажав клавишу F11. В очищенное поле вводится «Filein<название скрипта с расширением>». Выделенный текст следует перетащить на панель быстрого доступа. Далее переименовать кнопку или добавить иконку.

Написание скрипта можно считать окончательным, в результате работы был создан скрипт «Standart_Render» и созданы два инструмента, позволяющие быстро его запустить. Результат использования скрипта в сцене со стандартным примитивом представлен на рисунке 1.

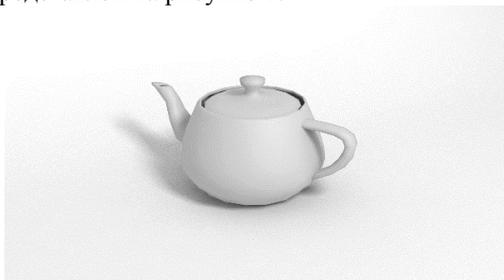


Рис. 1. Визуализация примитива с использованием скрипта

Говоря об особенностях программы следует отметить, что в некоторых случаях возможна необходимость изменения интенсивности освещения. Например, при использовании материалов в цвет фона, а также глянцевых или металлических текстур.

Эксперимент

Для доказательства рациональности использования скрипта «Standart_Render» вместо ручной настройки сцены, был произведен эксперимент. Он заключается в сравнении времени, потраченного на настройку сцены путем использования написанной программы и ручной настройкой сцены.

Предполагается, что программу написали заранее, а в данный момент стоит только выбор: использовать его или настраивать сцену вручную. Поэтому в экспериментальное время «скрипта» не будет включен процесс его написания.

Первым этапом было засечено время, которое пользователь потратит на построение сцены с помощью скрипта. Фактором, задерживающим про-

цесс, будет являться необходимость масштабирования сцены, автоматически созданного скриптом. Также в случае использования материалов, активно отражающих поверхности и свет, потребуется настройка интенсивности освещения. На процесс создания сцены, импортирования объекта и создания первичной визуализации потребовалось 35 секунд. За это короткое время в сцене появились нужные объекты с базовыми настройками.

Второй этап подразумевает создание объектов в сцене вручную, а также настройку их параметров. На процесс настройки камеры, выставления ее цели и настройки параметров протратилось некоторое количество времени. Эксперимент показал, что с первого раза вероятность минимальна создать источники освещения с удачным выбором яркости и расположением направленного источника освещения. Последствием этого выбора является большое количество промежуточных визуализаций, значительно задерживающим процесс создания стандартной сцены. В результате на создание сцены вручную потребовалось 3 минуты 10 секунд. Неоправданный расход времени ушел на процесс создания элементов сцены, выбор удачного положения и корректировку их настроек.

Результаты эксперимента показали, что при использовании скрипта время на создание сцены затрачивается меньше, чем при создании и настройке аналогичных объектов вручную.

Заключение

Таким образом, в процессе работы была написана программа на языке MAX Script, и проведен эксперимент, в результате которого было доказано преимущество использования созданной программы над процессом создания сцены вручную за счет экономии времени на рутинную работу.

Список использованных источников

1. Простое полигональное моделирование // 3D MASTER URL: https://3dmaster.ru/lessons/3dsmax/lesson_tv.html (дата обращения: 23.08.2018).
2. Learning MAXScript // Autodesk 3Ds Max URL: https://help.autodesk.com/view/3DSMAX/2018/ENU/?guid=__files_GUID_4C14F474_CD23_4001_93DF_0F0F9A6025_C7_htm (дата обращения: 23.08.2018).
3. MAXScript // TPU URL: <http://portal.tpu.ru:7777/SHARED/s/SHKLYARAV/disciplines/Subjects/Modeling/info.pdf> (дата обращения: 23.08.2018).
4. Визуализатор Mental Ray // 3d.demiart URL: https://3d.demiart.ru/book/3D-Max-7/Glava_07/Index03.htm (дата обращения: 23.08.2018).
5. Освещение с Mental Ray // 3d.d3S Mir URL: http://www.3dmir.ru/s_tutor/tutor/484.html (дата обращения: 23.08.2018).