МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

**Лабораторные работы**

**по курсу «Методы, средства и технологии мультимедиа»**

Выполнила: Алексюнина Ю.В.

Группа: М8О-407

Преподаватель: А.В. Крапивенко

Москва, 2020

**Моделирование виртуальных миров в программах**

**«Internet Space Builder» и «Unity 3D»**

1. **Цели**

Создание виртуальной трёхмерной обстановки для работы с ней через сеть Internet. Знакомство и экспорт в формат VRML 97. Знакомство с Unity.

1. **Задание**

Импортировать результаты работы ЛР №1 в среды ISB и Unity. Проверить соответствие цветов и текстур. Добавить к телу вращения два любых вида произвольной интерактивности и проверить результаты работы с помощью plug-in'ов к веб-браузерам.

В выводах описать сравнительные преимущества каждой из исследуемых программных сред.

1. **ПО**

ISB 3.0, Cortona VRML Client, Unity 5.6.4f1

1. **Теория**

Система виртуальной реальности – комплекс программно-аппаратных средств, создающих эффект погружения пользователя в искусственно созданную трехмерную среду. Максимальный эффект достигается при возможности интерактивного взаимодействия пользователя с окружающей виртуальной средой.

Области использования VR-технологий: визуализация, моделирование, навигация, развлечения, игры.

Одной из наиболее популярных областей применения, помимо игровых виртуальных сред, являются интерактивные WEB-приложения:

* Виртуальные руководства пользователя с 3D-разделами
* Конструкторские и инженерные приложения со сложным моделированием
* Курсы дистанционного обучения
* Сценарное 3D-моделирование событий
* Виртуальные экскурсии по городам, музеям, замкам
* Дизайн внутренних интерьеров и тд

Для создания виртуальных средств в среде Internet используется широкий спектр программных средств:

1. Панорамные видеотехнологии – создают виртуальную среду, в которой угол обзора составляет 360 градусов за счет склейки отдельных изображений в круговую панораму. Для получения качественных изображений требуется специальный круговой штатив с автоматизацией шага съемки или вращающаяся панорамная камера.

Чтобы можно было рассматривать конкретные объекты с любой степенью детализации и с любого угла, они фотографируются или рендерятся отдельно и компонуются с общей панорамой.

1. VRML – язык моделирования виртуальной реальности, описания трехмерных сцен, в котором определены иерархические преобразования, источники света, возможность произвольной смены точки наблюдения, геометрические тела, анимация, различные свойства материалов и атмосферных сред.

Он обеспечивает технологию для интеграции двумерных и трехмерных объектов текста и мультимедийных данных в единой согласованной модели виртуального мира.

Среди объектов VRML-графики есть аватар – анимированный объект, часто с человеческими чертами, с которым можно взаимодействовать и который может управлять как пользователем, так и компьютером.

Средства работы с VRML-сценами:

1. Трехмерные среды для создания VRML-сцен (ISB)
2. Средства визуализации VRML-сцен(Cortona, CosmoPlayer): работает в соответствии с установками конкретного web-сервера. Клиентский браузер посылает запросы на web-сервер, который должен возвратить запрошенный документ в формате стандарта MIME. Браузер анализирует эту информацию и с помощью плагинов воспроизводит VRML-сцены.
3. Прямое текстовое кодирование на языке VRML:

Файл VRML состоит из следующей послед-ти функциональных блоков: заголовка, описания прототипов, описания трехмерной сцены, маршрутов событий. Каждая строка, начинающаяся со знака «#», считается комментарием, кроме заголовка файла. После заголовка идут описания трехмерных объектов, их параметров и свойств, а также события и их обработчики для создания динамики в виртуальном мире.

Основные понятия языка:

1. Имена – регистро-зависимые, не кириллица, не управляющие символы от 0 до 32, без синтаксических символов, без цифр в начале.
2. Узлы – все трехмерные тела и любые мультимедийные сообщения(весь трехмерный мир представляет собой дерево объектов)

Типы:

А) группировочные – объединение вложенных узлов в ветви «дерева сцены» - родители

Б) конечные - определяют конкретные геометрические примитивы, звуки, видео, камеры; не могут иметь вложенных узлов-детей, кроме вспомогательных

В) вспомогательные – специфические узлы, располагающиеся только в определенных узлах: сенсоры, узлы-интерполяторы, которые меняют объекты сцены в соответствии с заданным набором опорных точек, и некоторые другие.

3. Поля и события.

Поля – уникальные параметры, отличающие данный объект от других аналогичных, каждое поле имеет свой тип и значение по умолчанию.

События – средства обмена инф-цией между узлами; входящие и исходящие события.

В VRML определены следующие примитивы: куб, сфера, цилиндр, конус.

Цвет фигуры определяется с помощью объекта Material, либо <Texture2>(текстура).

По умолчанию любой описанный объект будет располагаться точно по центру окна браузера, поэтому, если описать два одинаковых цилиндра, то они сольются друг с другом. Поэтому нужно поменять координаты второго цилиндра с помощью узла **Translation**(определяет координаты объекта)

Узлы, модифицирующие св-ва фигур, действуют на все фигуры. Чтобы ограничить область их действия, фигуры необходимо сгруппировать с помощью узла **Separator**

**Rotation** – вращение фигур вокруг осей координат

**Scale** – масштаб по x,y,z

**Anchor** – возможность перемещения между виртуальными мирами или переход по гиперссылке

Сенсоры окружающей среды – контролируют течение процессов в окружающей среде(течение времени и расположение пользователя):

* Сенсор времени – не имеет конкретной позиции в виртуальном мире и ассоциированных с ним геометрических объектов; может использоваться для порождения событий или управления узлами-интерполяторами
* Сенсор видимости – представляет собой некоторый параллелепипед, генерирующий события, когда он попадет в/выходит из поля зрения пользователя, по которым можно точно определить время и характер произошедшего
* Сенсор приближения – похож на сенсор видимости, но также отслеживает перемещения внутри параллелепипеда и генерирует сообщения при изменении позиции/ориентации пользователя
* Сенсор коллизий – включение/выключение распознавания коллизий(столкновений) между потомками этого узла и пользователем

Узлы-манипуляторы:

* Датчик касания – реагирует на взаимодействие курсора мыши с геометрическими объектами – потомками своего родительского узла.

Датчики буксировки:

* Сферический сенсор – вращение объекта вокруг центральной точки самого сенсора
* Цилиндрический сенсор – вращение только вокруг одной оси
* Плоский сенсор – перемещение вдоль осей X и Y в локальной системе координат

Unity3d - это очень гибкий мультиплатформенный движок, предоставляющий большую свободу действий пользователю и работающий под различными операционными системами. Возможность импортирования моделей напрямую из 3d редактора, является одной из самых главных. В Unity3d чаще всего создаются компьютерные игры.

1. **Ход выполнения**

Работа с Unity. Экспортируем из 3Ds max комнату(при экспорте в \*.fbx) указываем AxisConversion: Z-up – нужно, так как Unity использует систему координат с осью Y, направленной вверх.

Затем импортируем ее в Unity. Подгружаем заново все материалы (важно, чтобы они не были с одинаковыми именами). Возникла проблема с материалом зеркала, поскольку его невозможно было экспортировать из 3Ds max корректно, и я попыталась написать скрипт для отражения. К сожалению, он работает не полностью.

Необходимо взаимодействие с предметами сцены. Для этого воспользуемся FPSController(управление персонажем от первого лица) из StandartAssets(они скачиваются и импортируются из AssetStore).

Нужно добавить несколько видов произвольной интерактивности, и для этого были выбраны следующие предметы: дверца шкафа (открывается/закрывается со звуком), миска (падает со звуком), юла (начинает покачиваться со звуком). Все взаимодействия осуществляются по клику.

Для каждого из предметов, а также для FPSController был написан C#-скрипт с последовательностями действий в каждый момент времени.

Основная сложность заключалась в прописывании определенных параметров при написании скриптов (подбор угла/расстояния).

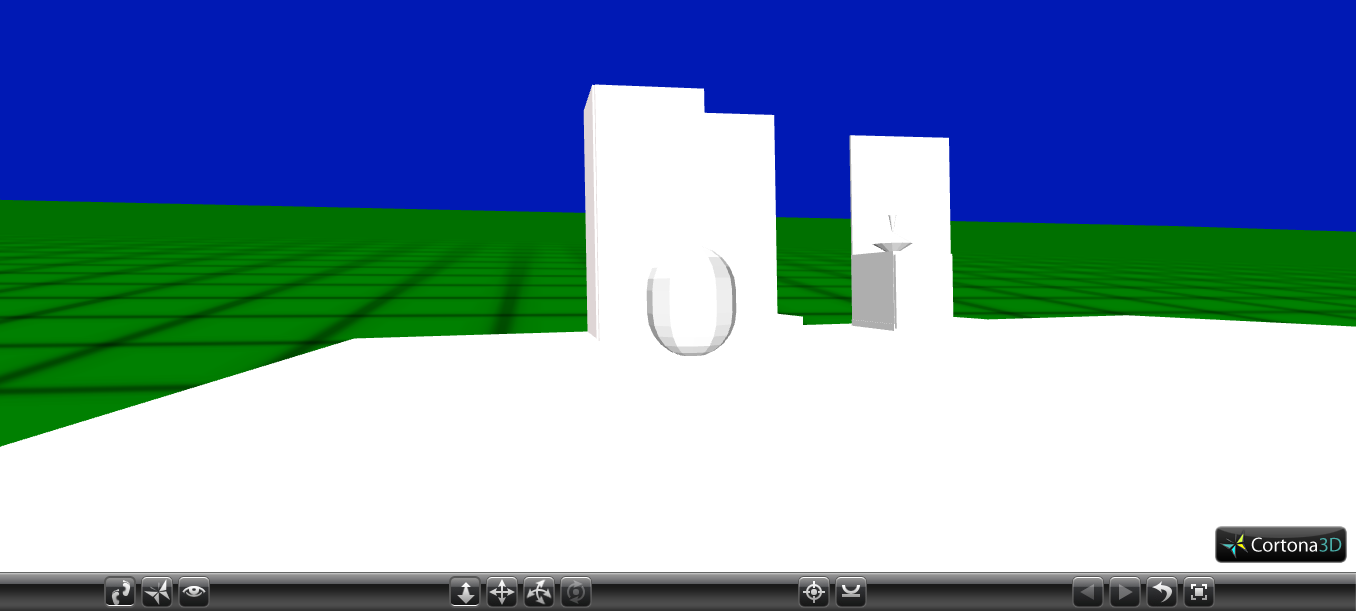
После того, как в режиме Game все взаимодействия начали работать корректно, установила Unity WebGL для работы проекта в браузере. К сожалению, современные версии Google Chrome/Mozilla Firefox не поддерживают данный плагин из соображений безопасности, если его запускать с компьютера, отдельно собрав и сохранив перед этим. Но если нажать кнопку «Build and Run», то запустится небольшой локальный веб-сервер, через который страница загрузится.

Далее, я перешла к работе с программами Internet Space Builder 3.0 и Cortona VRML Client. Но сначала вернулась к проекту комнаты в 3Ds max и ре-экспортировала его в другой формат файла, так как оказалось, что ISB воспринимает на импорт только файлы формата \*.3ds. При импорте сцены я обнаружила, что она слишком большая по некоторым критериям для урезанной версии программы. А поскольку полную версию программы было найти практически невозможно, пришлось перетащить все примитивы отдельно на сцену в ISB(как самостоятельные объекты), а затем удалять по одному объекту до тех пор, пока кнопка сохранения не стала «Enabled».

Также, даже при исходном импорте перенеслись не все объекты, небольшая часть из них пропала. Когда, наконец, у меня появилась возможность сохранения сцены, нужно было сделать 2 вида произвольной интерактивности. В данном проекте – при нажатии на предполагаемое авокадо издается звук, а при клике на юлу – переход по ссылке. После завершения работы с ISB была установлена Cortona VRML Client в качестве плагина для браузера. Примечательно, что данная программа «заставила» мой ноутбук «вспомнить» о существовании Internet Explorer, поскольку только в нем она может работать корректно, не применяя лишних усилий к ее установке.

1. **Результат**

Для ISB 3.0 & Cortona VRML Client:



Для Unity3D & Unity WebGL:



1. **Вывод**

Если сравнивать Unity3D и ISB, то, на самом деле, их довольно сложно сравнивать, поскольку ISB 3.0 – ПО, выпущенное относительно давно и имеющее куда меньший функционал, чем Unity. Конечно же, Unity абсолютно превосходит ISB при выполнении данной работы, кроме, разве что, одного пункта: для добавления интерактивности в ISB потребовалось куда меньше времени и усилий, чем в Unity. В остальном же Unity – явный лидер в этой «гонке».