

ТАЛЛИННСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет инфотехнологий

Институт автоматике

ISS40LT

Михаил Кононов 134706

**МОДЕЛИРОВАНИЕ ЧАСТИ ГОРОДА В
ПРИЛОЖЕНИЯХ ВИРТУАЛЬНОЙ
РЕАЛЬНОСТИ**

Бакалаврская работа

Руководители: Кристина Васильева, PhD

Доцент

Алексей Тепляков, PhD

Научный сотрудник

Tallinn 2016

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Automaatikainstituut

ISS40LT

Mihhail Kononov 134706

Linnaosa modelleerimine virtuaalreaalsuse rakenduste jaoks

Bakalaureusetöö

Juhendajad: Kristina Vassiljeva, PhD

Dotsent

Aleksei Tepljakov, PhD

Teadur

Tallinn 2016

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Mihhail Kononov

29.08.2016

Аннотация

Данная бакалаврская работа посвящена созданию модели части города, а также ее реализации в Blender и Unreal Engine 4 при помощи очков Oculus Rift, благодаря которым воспроизводится виртуальная реальность. В частности была создана модель небольшой части Ласнамяэ около ледового холла Tondiraba Jäähall, которая включает в себя сам ледовый холл, несколько домов: пятиэтажные и девятиэтажные панельные дома, дороги, детскую площадку, различные фонари, знаки, остановки, заборы, ограждения и прочие разные небольшие объекты.

В программе трехмерного моделирования Blender моделировались все реальные объекты, перечисленные выше. Использовались бесплатно доступные в Интернете модели деревьев и травяного покрытия, а кусты изначально присутствуют в стартовом наборе объектов Unreal Engine 4, разрабатываемого и поддерживаемого компанией Epic Games. В Unreal Engine 4 были импортированы смоделированные объекты и расставлены по карте в соответствие с их реальным местоположением. Также, были настроены параметры материалов, освещения и ландшафта.

Текстуры создавались в растровом графическом редакторе GNU Image Manipulation Program 2 или GIMP 2. Текстуры сделаны из фотографий покрытий разных объектов.

Проект довольно объемный для реализации одним человеком. Поэтому на основе него было написано две дипломные работы, которые освещают разные области проекта. В данной работе пойдет речь об игровом движке Unreal Engine 4, а именно об импортировании моделей, добавлении растительности и проводов, настройке коллизии, материалов моделей, освещении, ландшафте.

Дипломная работа написана на русском языке и содержит текст на 29 страниц, 7 глав и 42 рисунков.

Annotatsioon

Käesolev bakalaureusetöö teema on linnaosa mudeli loomine ja selle realiseerimine Blenderis ja Unreal Engine 4 prillide Oculus Rifti abil, mis kuvavad virtuaalreaalsust. Mudel on loodud väikese osa kohta Lasnamäelt Tondiraba Jäähallini. See koosneb jäähallist, mitmest majast: viie- ja üheksakordsetest paneelmajadest, teedest, lastemänguväljakust, erinevatest tänavavalgustuspostidest, liiklusmärkidest, bussipeatusest, piirdeaedadest ja muudest erinevatest väikesetest objektidest.

Kolmemõõtmelise modelleerimise programmis Blender on modeleeritud kõik ülalpool loetletud reaalsed objektid. Puud ja muru on allalaaditud internetist, aga põõsad olid olemas stardikomplektis Unreal Engine 4, mida arendati ja toetati Epic Games-i abil. Unreal Engine 4 abil imporditi modelleeritud objektid ja paigutati kaardil vastavalt nende reaalsele asukohale. Samuti seadistati materjalide, valgustuse ja maastiku parameetrid.

Tekstuurid loodi rastergraafika redaktoris GNU Image Manipulation Program 2 või GIMP 2. Tekstuurid on tehtud objektide erinevate katete piltidest.

Projekti realiseerimine on üsna kogukas ühele inimesele. Seetõttu oli kirjutatud kahte diplomitööd, missugused kirjeldavad erinevaid projekti osasid. Käesolev diplomitöö koosneb mängumootorist Unreal Engine 4, mudelite importeerimisest, taimestiku ja juhtmete lisamisest, mudelite materjalide seadistamisest, valgustusest ja maastikest.

Lõputöö on kirjutatud vene keeles ning sisaldab teksti 29 leheküljel, 7 peatükki, 42 joonist.

Abstract

The purpose of this bachelor work was to create a small part of the city using Blender and Unreal Engine 4 with a help of Virtual Reality headset Oculus Rift. To be more specific, there was created a model of a small part of Lasnamae district, located near the Tondiraba Ice Hall which includes ice hall itself, some houses: 5 and 9 stories high apartment blocks, roads, playground, various road lights, traffic signs, bus stops, fences, barriers and other small objects.

All the objects listed above were modeled using 3d modeling software called Blender. There were free trees and grass models used for the project and bushes were taken from the originally presented Starter Content of the Unreal Engine 4, developed and supported by Epic Games. All the modeled objects were imported to the Unreal Engine 4 and placed across the map according to their real location. Also, the parameters of materials, lighting and landscape has been configured.

Textures were created in the raster graphics editor GNU Image Manipulator Program 2 or simply GIMP 2. Textures were made using photos of various objects surfaces.

The project is too complex to be done alone, so there were two bachelor works made, describing various parts of the project. This part of thesis is about the Unreal Engine 4 game engine, specifically about models importing, adding foliage, cables, collision, about model's materials, lighting and landscape configuring.

The thesis is in Russian and contains 29 pages of text, 7 chapters, 42 figures.

Sisukord

1. Введение.....	10
1.1 Цели.....	11
1.2 Проблемы.....	11
2. Импорт моделей из Blender в Unreal Engine 4.....	12
3. Настройка свойств материалов.....	14
3.1 Основные настройки материалов.....	14
3.1.1 Базовый цвет.....	16
3.1.2 Металлические свойства.....	17
3.1.3 Зеркальные свойства.....	18
3.1.4 Шероховатость.....	18
3.1.5 Непрозрачность.....	19
3.1.6 Нормаль.....	19
3.1.7 World Position Offset.....	20
3.1.8 Преломление.....	21
3.2 Пример настройки свойств используемого материала.....	22
4. Ландшафт и растительность.....	23
4.1 Ландшафт.....	23
4.2 Растительность.....	28
5. Коллизии.....	31
6. Настройка освещения.....	33
6.1 Постобработка.....	35
7. Взаимодействие проекта с Oculus Rift.....	38
Заключение.....	39
Kasutatud kirjandus.....	42
Lisad.....	43

Jooniste loetelu

Рисунок 1. Пример "Блупринта" [4]	12
Рисунок 2. Импорт моделей	13
Рисунок 3. Материал	14
Рисунок 4. Свойства материала.....	15
Рисунок 5. Окно деталей.....	16
Рисунок 6. Базовый цвет [5]	17
Рисунок 7. Металлические свойства [5].....	17
Рисунок 8. Зеркальные свойства [5]	18
Рисунок 9. Шероховатость [5].....	19
Рисунок 10. Непрозрачность [5].....	19
Рисунок 11. Нормаль [5]	20
Рисунок 12. Движение объекта по спирали и волнами [5].....	21
Рисунок 13. Преломление [5]	21
Рисунок 14. Эффект ветра	22
Рисунок 15. Меню Modes.....	23
Рисунок 16. Блок LayerBlend.....	24
Рисунок 17. Смешивание слоев.....	25
Рисунок 18. Смешивание по весу	25
Рисунок 19. Смешивание по высоте	26
Рисунок 20. Блок SCurve.....	26
Рисунок 21. Блок Landscape Visibility Mask.....	27
Рисунок 22. Landscape Hole Material.....	27
Рисунок 23. Создание ассетов	28
Рисунок 24. Окно Foliage	29
Рисунок 25. Перенос объектов	29
Рисунок 26. Использование конкретного объекта	30
Рисунок 27. Автоматическое создание коллизии.....	31
Рисунок 28. Параметры коллизии.....	31
Рисунок 29. Sky_Sphere, SkyLight и Directional Light.....	33
Рисунок 30. Выбран параметр Movable.....	33
Рисунок 31. Увеличение значения разрешения карты освещения	34
Рисунок 32. Создание Lightmap	34
Рисунок 33. Постобработка	35
Рисунок 34. Постобработка	35
Рисунок 35. Настройки постобработки	36
Рисунок 36. Настройки постобработки	37
Рисунок 37. Использование виртуальных очков в Unreal Engine 4.....	38
Рисунок 38. Скриншот 1	39
Рисунок 39. Скриншот 2	39
Рисунок 40. Скриншот 3	40
Рисунок 41. Скриншот 4	40
Рисунок 42. Скриншот 5	41

1. Введение

Виртуальная реальность – искусственная реальность, созданная техническими средствами, в которой необходимо создать условия того, чтобы человек ощущал себя как можно ближе к реальному миру. Идея виртуальной реальности на сегодняшний день очень популярна, и её технологии стремительно развиваются. Происходит это в том числе за счет того, что производительность домашних компьютеров быстро увеличивается, позволяя визуализировать изображение для обоих глаз с нужной частотой кадров для комфортной работы с приложениями виртуальной реальности. В настоящее время виртуальная реальность используется для игр, медицины, обучения солдат, летчиков и астронавтов, создания автомобилей и т.п. Существует несколько основных видов систем, которые формируют изображение виртуального мира в системах виртуальной реальности, первые коммерческие модели, такие как, например, HTC Vive выходят на рынок и уже доступны для покупки каждому. В этой работе будут использоваться очки виртуальной реальности Oculus Rift Development Kit 2.

В данной дипломной работе, используя игровой движок Unreal Engine 4, представлена модель небольшой части района Ласнамяэ в виртуальной реальности. Главная цель, которая была поставлена – создать как можно более реалистичный и детализированный участок района, чтобы человек, который в нем живет или просто когда-либо находился там в реальном мире, мог сравнить, насколько точно можно воссоздать реальный мир в виртуальности. Иными словами, должен достигаться эффект присутствия. Движок Unreal Engine 4 также позволяет в дальнейшем в случае необходимости продолжить данную работу и дополнить.

Для того чтобы воссоздать небольшой участок мира в Unreal Engine 4, нужно проделать довольно большую работу. Для создания реалистичного виртуального мира помимо моделирования и текстурирования объектов необходимо как можно точнее настроить материал к каждому объекту, настроить освещение, добавить

погодные эффекты, такие как ветер, учесть ландшафт местности, настроить коллизию объектов.

1.1 Цели

В данной работе поставлены следующие цели:

- Импорт моделей из Blender в Unreal Engine 4;
- Расстановка объектов по карте;
- Настройка свойств материалов;
- Настройка освещения;
- Настройка коллизии;
- Добавление ландшафта и растительности;
- Постобработка;
- Работа с очками виртуальной реальности Oculus Rift.

1.2 Проблемы

Проблемы, возникнувшие во время работы в Unreal Engine 4:

- Настройка материала, чтобы сделать его приближенным к настоящему;
- Настройка освещения;
- После просчета шейдеров, ландшафт остается без текстуры.

2. Импорт моделей из Blender в Unreal Engine 4

Unreal Engine 4 – игровой движок, разрабатываемый и поддерживаемый компанией Epic Games. Unreal Engine 4, в основном, является движком для создания компьютерных игр, в других случаях - для работы с графикой в кинематографе, а также в образовательных целях. В игровом движке можно писать игровую логику на языке программирования C++, кроме этого можно использовать нодовую систему "Блупринт", как изображено на Рис. 1.

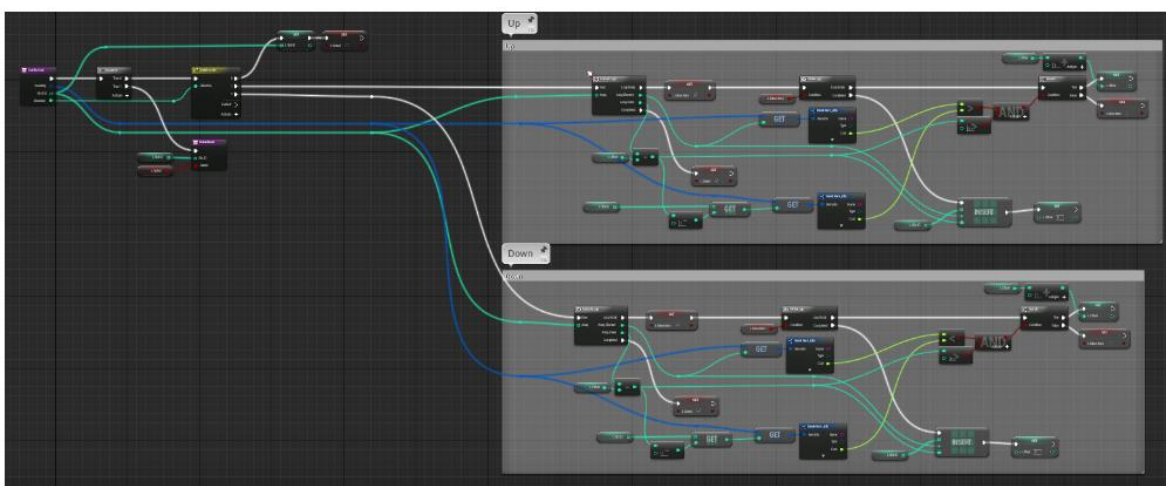


Рисунок 1. Пример "Блупринта" [4]

"Блупринт" – визуальный язык программирования, с помощью которого возможно написать логику игры без применения языка программирования C++. В данной работе не будет применяться "Блупринт", а лишь настройка свойств материалов.

Первым делом необходимо импортировать созданные в Blender модели различных объектов в Unreal Engine 4. Для удобства желательно использовать отдельные папки для разных объектов, а также для удобства расстановки моделей по игровой карте можно материалы и текстуры также распределить по папкам. Импорт моделей выполняется следующим образом: открыть Content Browser, где находятся все модели, текстуры, материалы, загружаемые пользователем или уже загруженный

стартовый набор объектов в игровом движке; выбрать Import и импортировать нужные объекты в формате FBX (Рис. 2).

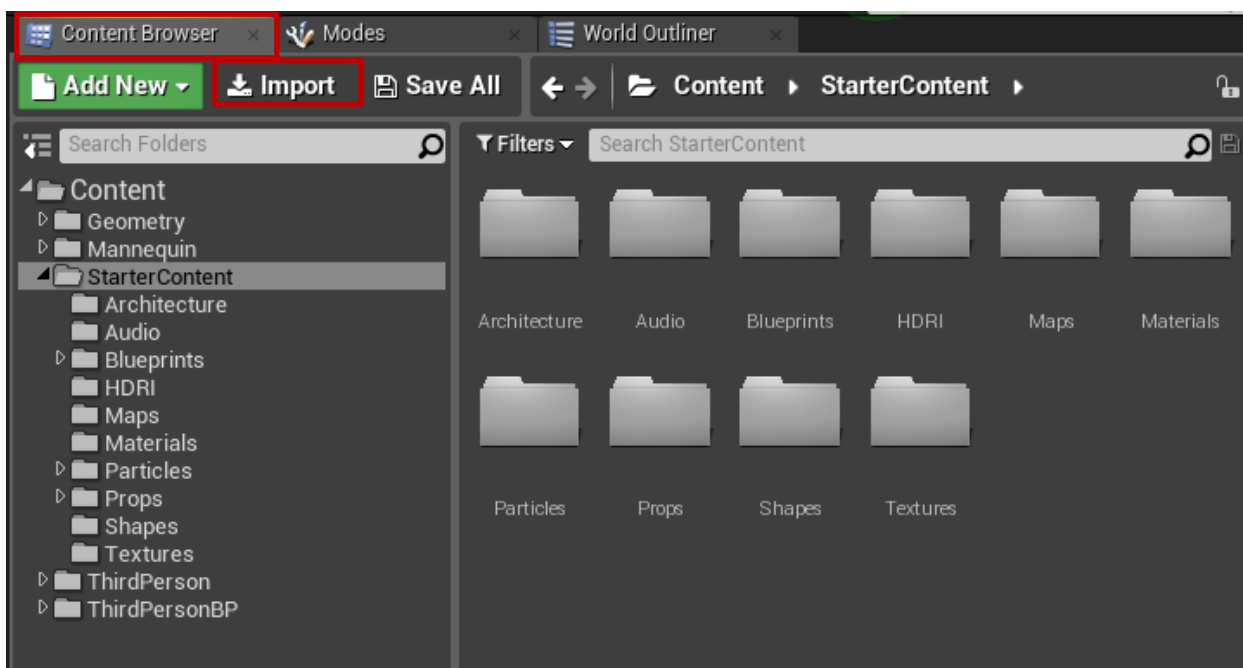


Рисунок 2. Импорт моделей

Далее необходимо расставить все объекты по карте в соответствии с их реальным расположением. Карту можно импортировать вместе с какой-то моделью, либо создать ландшафты и на них сверху наложить спутниковые снимки. Ближе к концу работы, когда необходимо будет построить освещение всей части города, карту, созданную на ландшафтах, придется удалить, в противном случае освещение построится не менее за 2 часа из-за восьми ландшафтов.

3. Настройка свойств материалов

Ознакомление с материалом в игровом движке является одной из главных задач, ведь материал определяет тип и свойства поверхности объекта. Материал контролирует цвет, прозрачность, металлические свойства, шероховатость, зеркальность, сияние поверхности и многое другое.

Когда свет попадает на поверхность объекта, материал определяет то, как поверхность объекта будет взаимодействовать со светом. Эти расчеты производятся с использованием поступающих данных, которые описывают материал: текстуры или цвета, различные параметры настройки свойств.

3.1 Основные настройки материалов

Выбирая какой-либо материал, появляется окно графика, который отображает сеть блоков данного материала. На нем и происходит основная работа над материалом, как показано на Рис. 3.

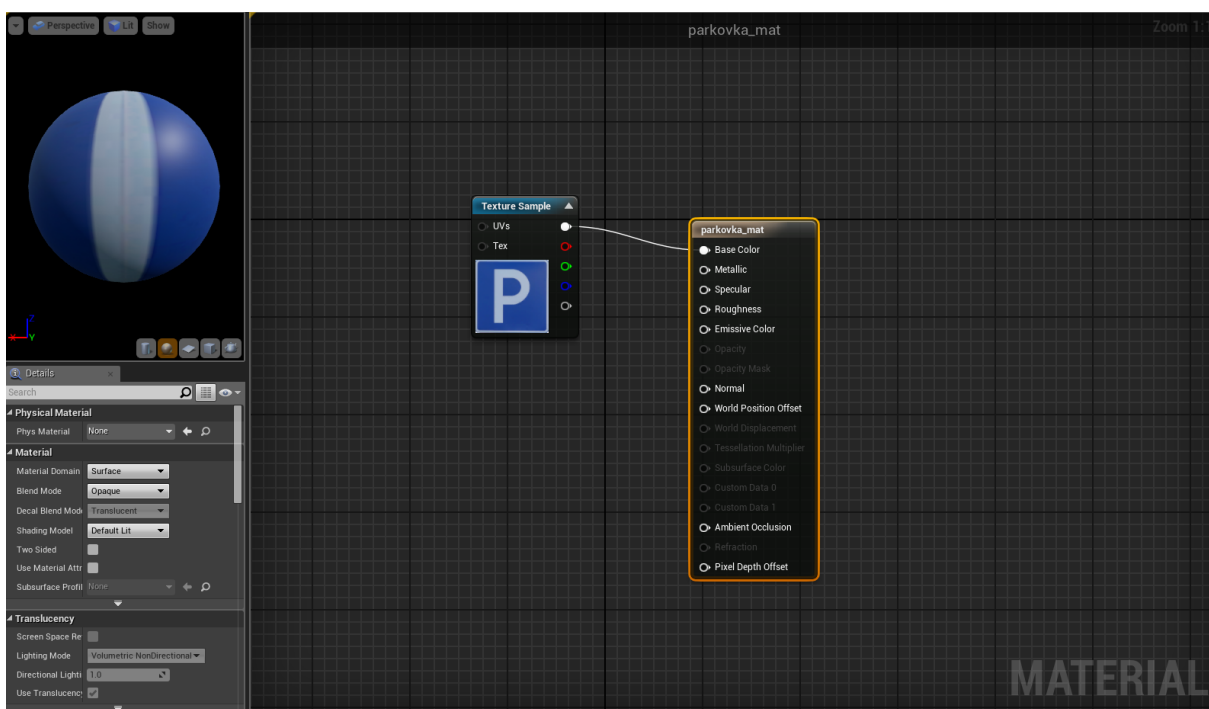


Рисунок 3. Материал

В окне графика имеется описание конечного материала, проиллюстрированного на Рис. 4.

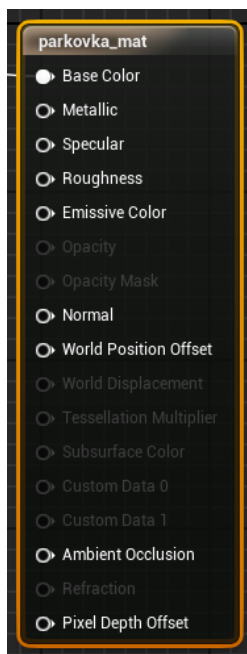


Рисунок 4. Свойства материала

Подавая значения путем функций, констант или текстур на входы, изображенные на Рис. 4, можно из объекта создать почти любую физическую поверхность. Не обязательно использовать все свойства, все неиспользованные в таком случае примут значения по умолчанию. Например, когда создается зеркальный материал, может понадобиться только значение Specular и Roughness, потому что другие свойства не будут иметь значения. Roughness понадобится в том случае, если зеркальная поверхность гладкая, тогда это значение нужно сделать нулевым, в то время как по умолчанию оно равно 0.5. Описание свойств материалов, примененных в данной работе, представлено ниже.

В окне деталей (Рис. 5) в данной работе использовалось два параметра:

- Blend Mode — Определяет, как материал смешивается с пикселями позади. Параметр Opaque значит, что объект непрозрачен и параметры Opacity и Opacity Mask в описании конечного материала как на Рис. 4 неактивны.

Параметр Masked активирует параметр Opacity Mask и позволяет, например, подключить к нему альфа канал текстуры, тем самым сделав его прозрачным. Это можно использовать, к примеру, для текстур листьев деревьев.

Параметр Translucent активирует параметр Opacity и позволяет управлять прозрачностью материала. В данной работе этот параметр используется для материала стекла.

- Two Sided — позволяет сделать материал двухсторонним. Это также можно использовать для материала листьев деревьев.

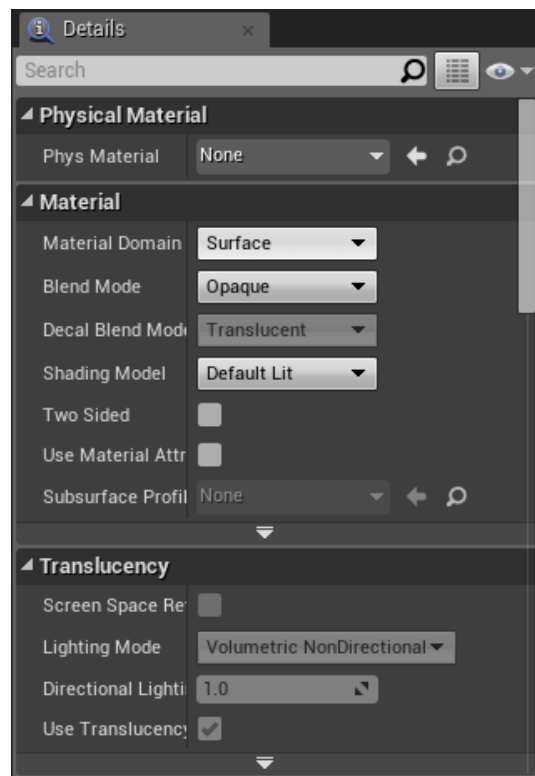


Рисунок 5. Окно деталей

3.1.1 Базовый цвет

Базовый цвет просто определяет цвет материала. Он применяется в значении RGB и каждый цвет находится в интервале от 0 до 1. Вместо цвета можно использовать фотографии, или текстуры, или комбинировать вышеперечисленное. Для большинства материалов в данной работе применялись текстуры как базовый цвет [5].

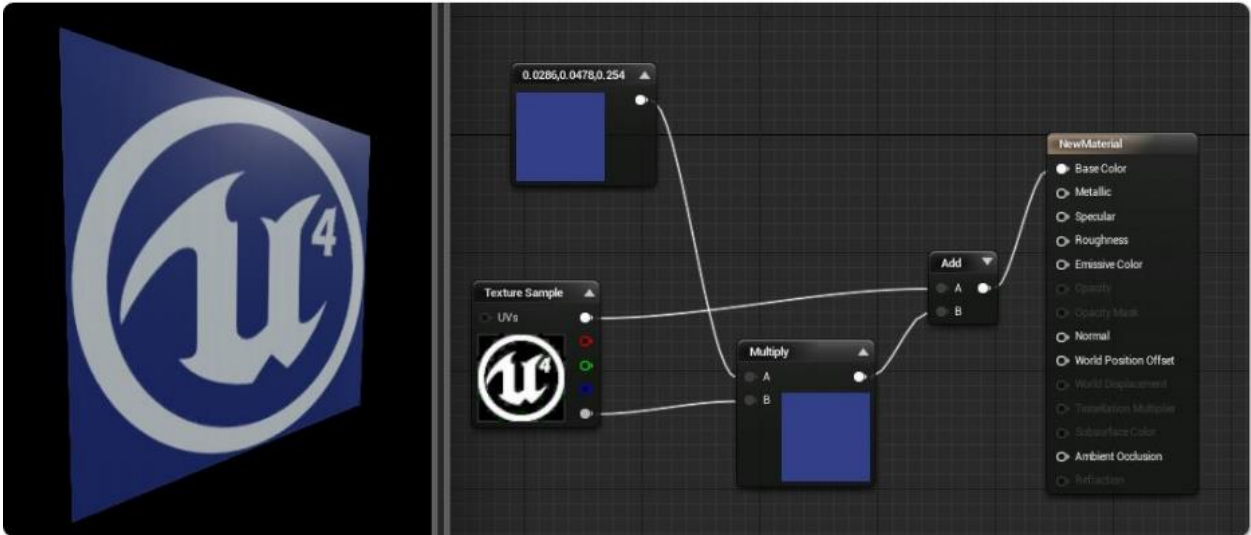


Рисунок 6. Базовый цвет [5]

3.1.2 Металлические свойства

Металлический вход определяет, обладает ли материал металлическими свойствами. Неметаллы имеют значение нулевого значения *Metallic*, чистые металлы имеют соответственно значение 1.0. Если же поверхность, например, ржавая или пыльная потребуется воспользоваться значением в интервале между 0 и 1 [5].

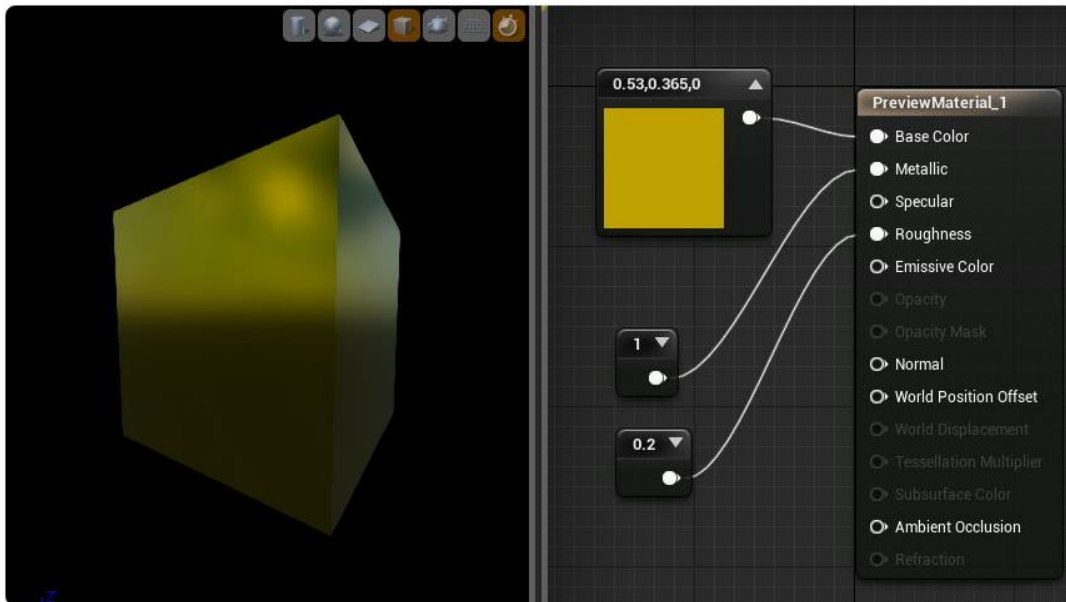


Рисунок 7. Металлические свойства [5]

3.1.3 Зеркальные свойства

В большинстве случаев данное свойство должно оставаться без изменений в качестве значения по умолчанию 0.5. Значение Specular от 0 до 1 определяет отражаемость поверхность для неметаллического материала. На металл данный вход никак не будет влиять [5].

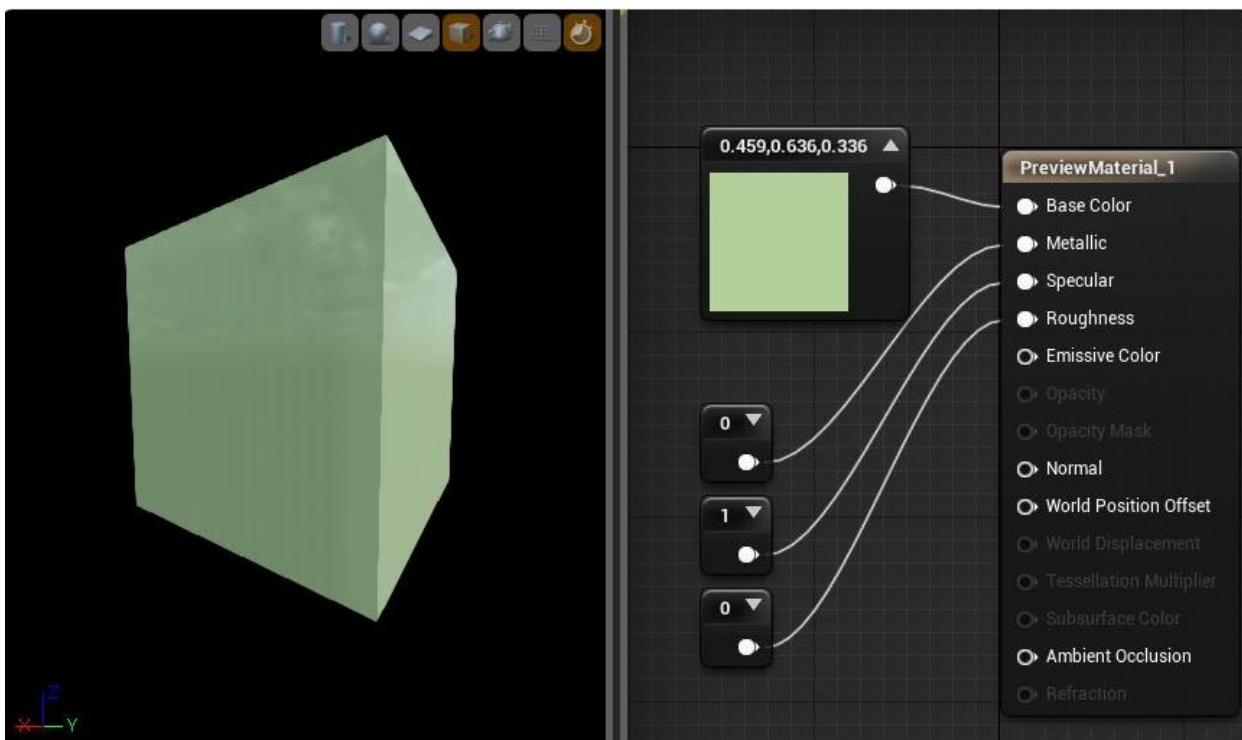


Рисунок 8. Зеркальные свойства [5]

3.1.4 Шероховатость

Вход Roughness определяет, насколько сильно материал будет неровным. Чем больше значение шероховатости имеет материал, тем сильнее от него будет рассеиваться свет в нескольких направлениях. В случае нулевого значения шероховатости будет означать, что материал полностью гладкий. Значение Roughness на 1.0 будет давать полностью матовую или рассеянную поверхность [5].

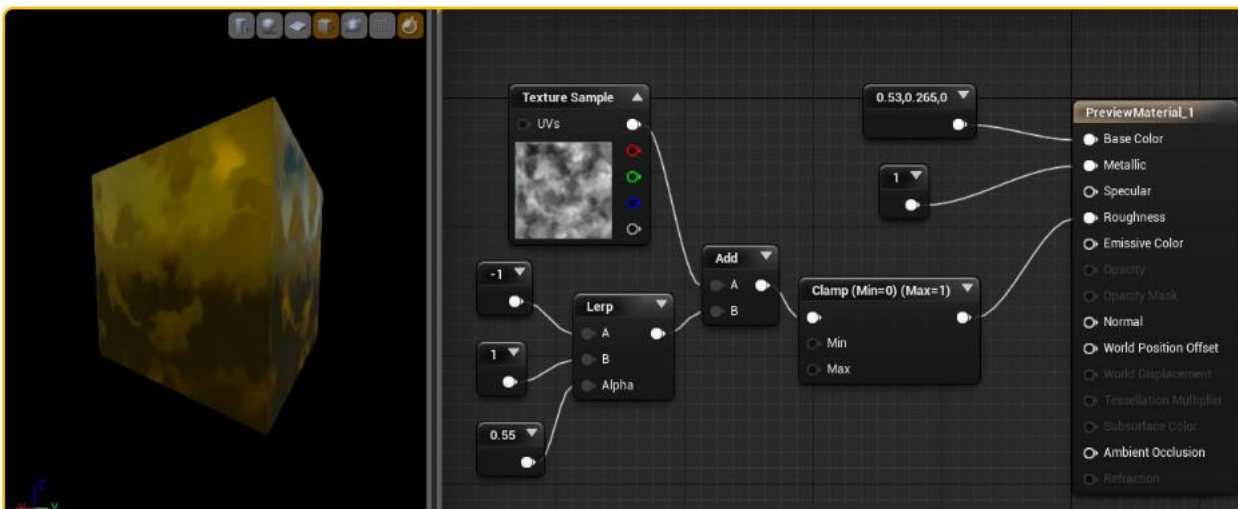


Рисунок 9. Шероховатость [5]

3.1.5 Непрозрачность

Данное свойство определяет прозрачность материала при использовании Translucent Blend Mode. Значение 0.0 дает полностью прозрачный материал, в то время как 1.0 - полностью непрозрачный [5].

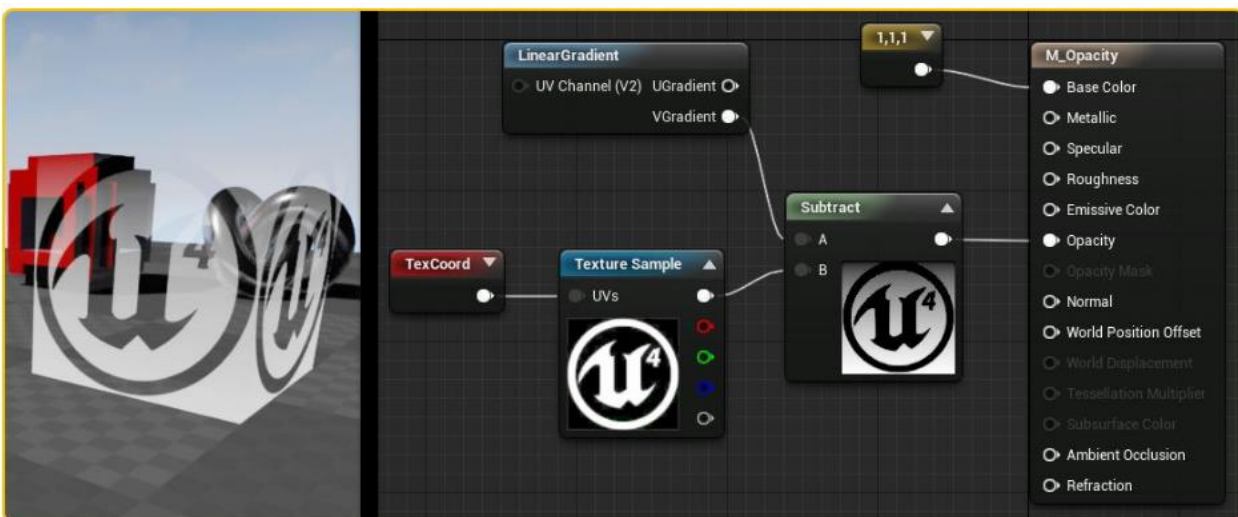


Рисунок 10. Непрозрачность [5]

3.1.6 Нормаль

Вход Normal использует карту нормалей, которая имитирует рельефную поверхность материала. Например, для текстуры кирпичной плитки можно создать карту

нормалей, которая визуально увеличит количество полигонов на объекте, но не повлияет на геометрию объекта, соответственно под углом будет видно, что поверхность ровная [5].

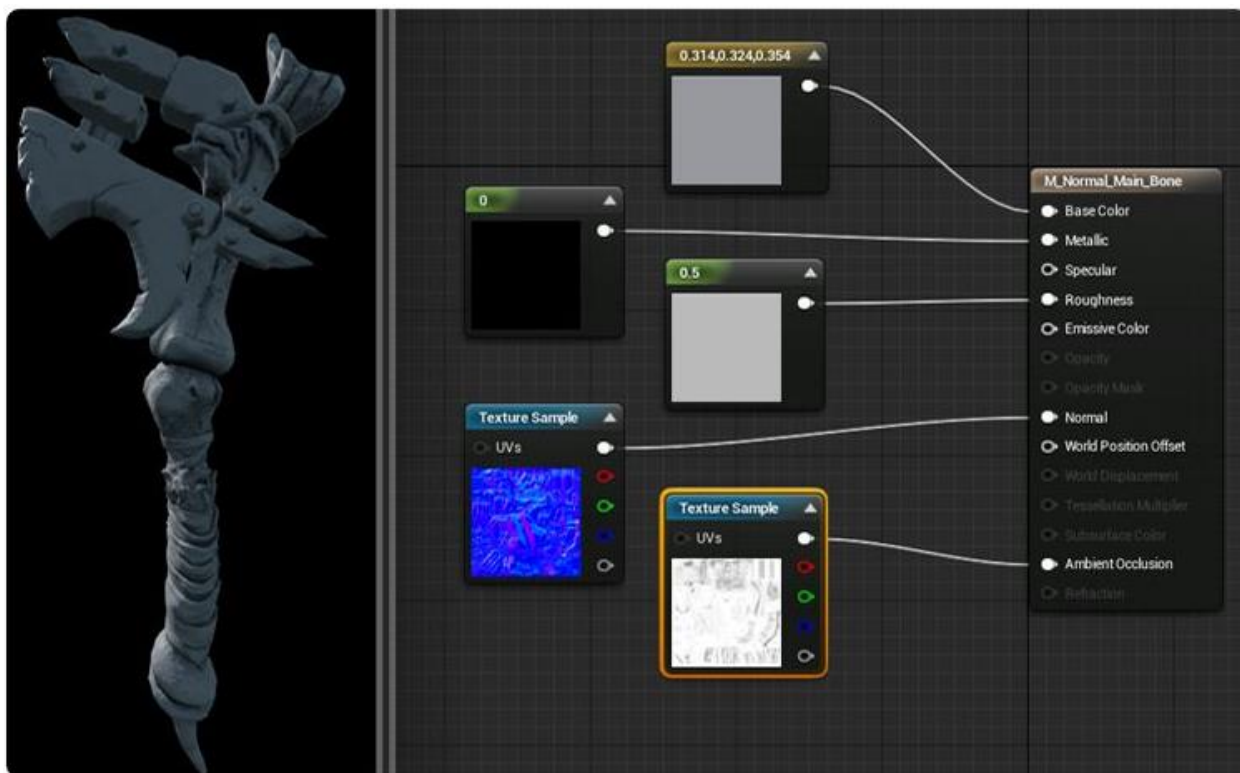


Рисунок 11. Нормаль [5]

3.1.7 World Position Offset

Данное свойство дает возможность управлять вертексами объекта с помощью материала. Это позволяет менять форму объекта, перемещать, менять его и создавать многие другие эффекты. Используется это для того, чтобы какие-то части объекта, например, листья на деревьях могли шевелиться, создавая эффект ветра. Изображение ниже, Рис. 12, показывает, что объект будет двигаться одновременно по спирали и волнами вверх и вниз, как будто танцует [5].

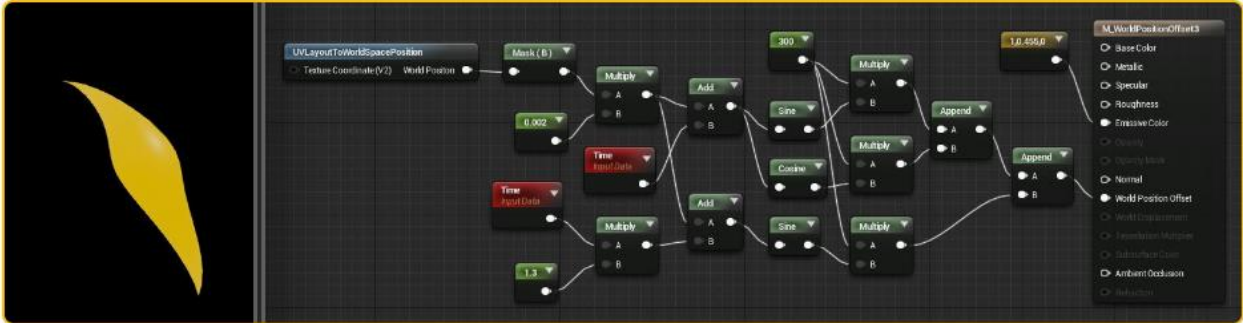


Рисунок 12. Движение объекта по спирали и волнами [5]

3.1.8 Преломление

Данный параметр отвечает за показатель преломления поверхности какого-либо объекта. Это свойство следует применять при создании таких материалов, как стекло, вода, которые преломляют проходящий через них свет. Для воды стоит применять значение 1.00, а стекло требует значение 1.52 [5].

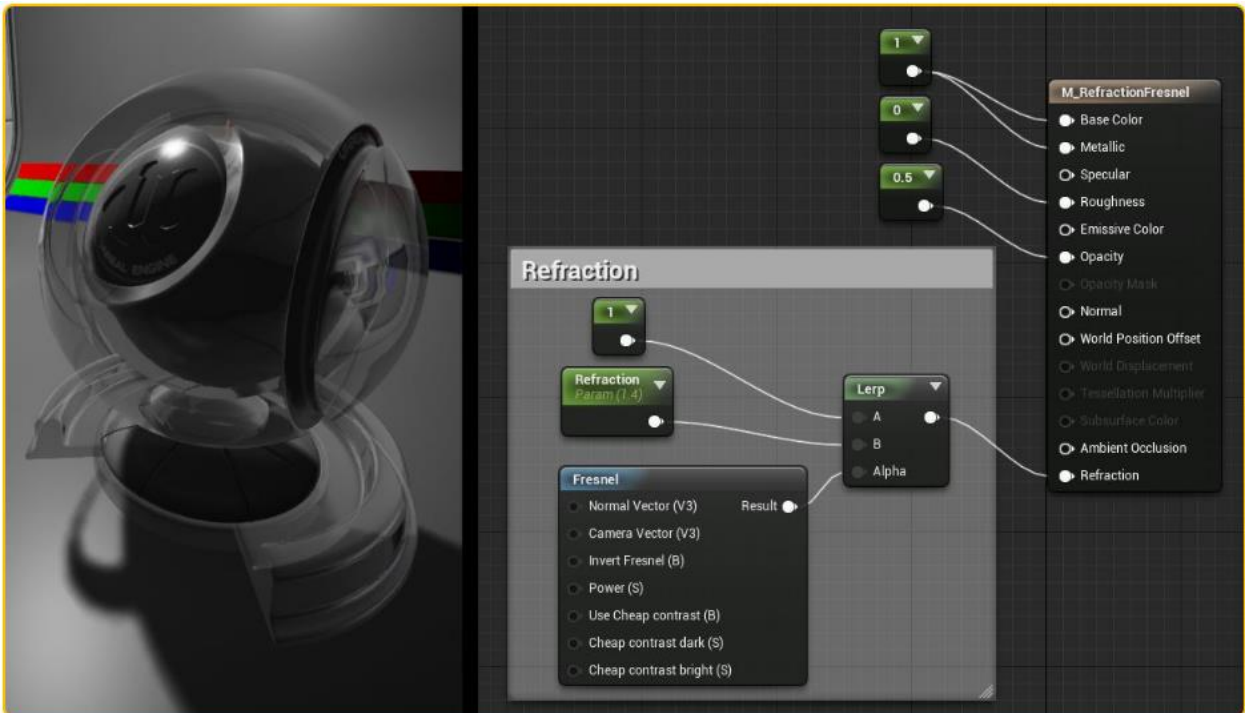


Рисунок 13. Преломление [5]

3.2 Пример настройки свойств используемого материала

Для создания простейшего эффекта ветра можно к выходу World Position Offset подключить блоки Panner и Sine. При этом "скорость" ветра настраивается путем изменения X и Y функции синуса в параметрах блока Panner (Рис. 14).

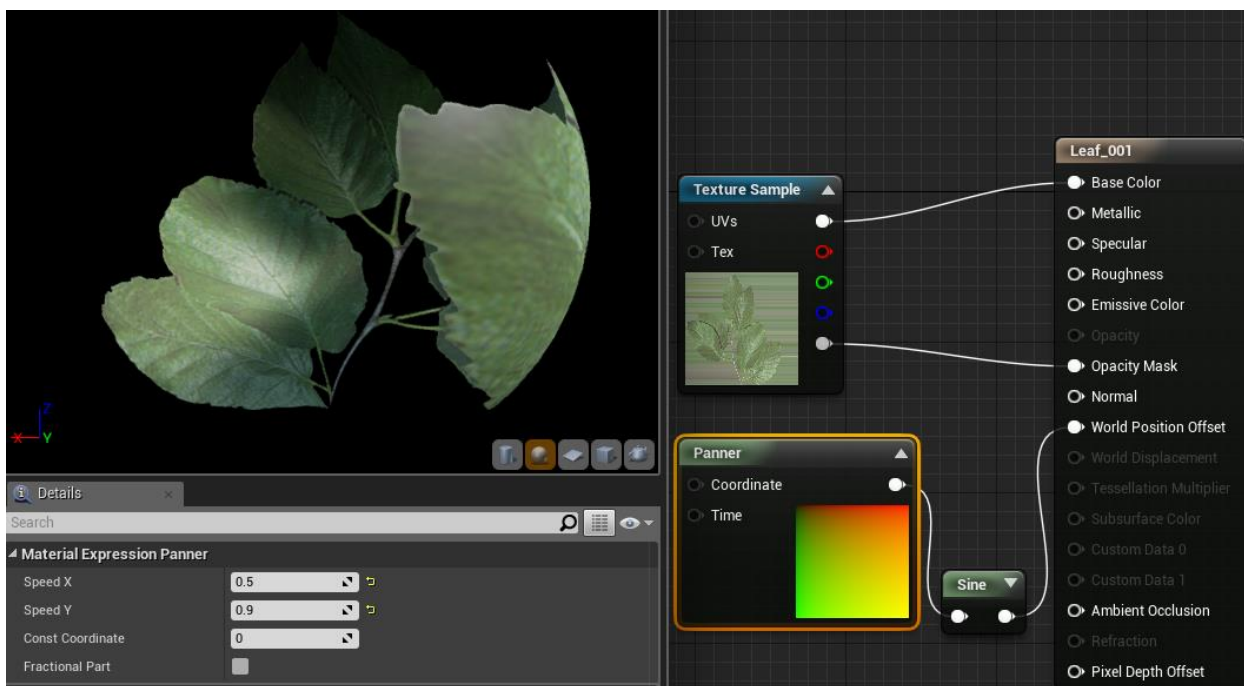


Рисунок 14. Эффект ветра

4. Ландшафт и растительность

4.1 Ландшафт

Система ландшафта позволяет создавать рельеф местности – горы, отверстия для пещер, долины, холмы, возвышения, ямы и многое другое. Также она дает возможность изменять его форму и внешний вид, используя ряд инструментов. В меню Modes в инструменте ландшафта есть три режима, как показано на Рис. 15. Manage – режим, позволяющий создавать и управлять ландшафтом. Sculpt – режим, позволяющий изменять форму ландшафта с помощью набора кистей. Благодаря данному режиму можно создать, например, горы, ямы, равнины и т.п. Paint – режим, необходимый для изменения внешнего вида ландшафта, рисуя на нем текстурами.

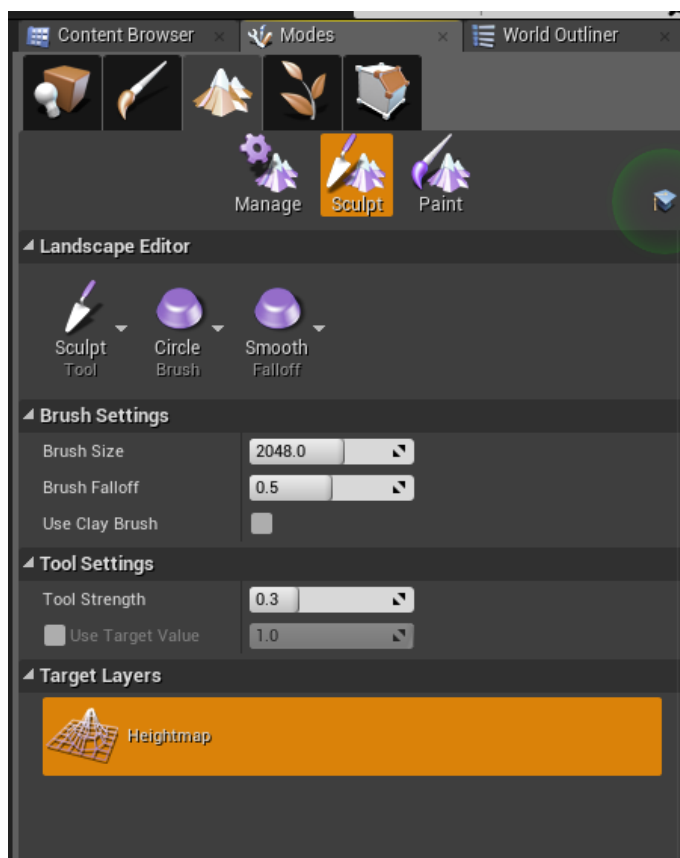


Рисунок 15. Меню Modes

В первую очередь работа в Unreal Engine 4 началась с создания ландшафта со спутниковыми снимками местности. Было создано 8 ландшафтов, на которые сверху накладывались изображения со спутниковых снимков, сделанные с помощью сайта <http://geoportaal.maaamet.ee/>. Один ландшафт имеет размер 150x150 метров, так как это один из рекомендуемых размеров ландшафта. Далее, на поверхности ландшафта расставлялись объекты. После этого создавался последний ландшафт, который служит поверхностью в конечном проекте. После его создания можно было приступить к процессу покраски ландшафта травой, щебнем, песком, гравием.

Для того, чтобы раскрасить ландшафт в разные текстуры, перечисленные выше, нужно создать специальный материал со слоями. Для этого необходимо добавить специальный блок LayerBlend, как представлено на Рис. 16, к которому можно подключить блоки с текстурами.

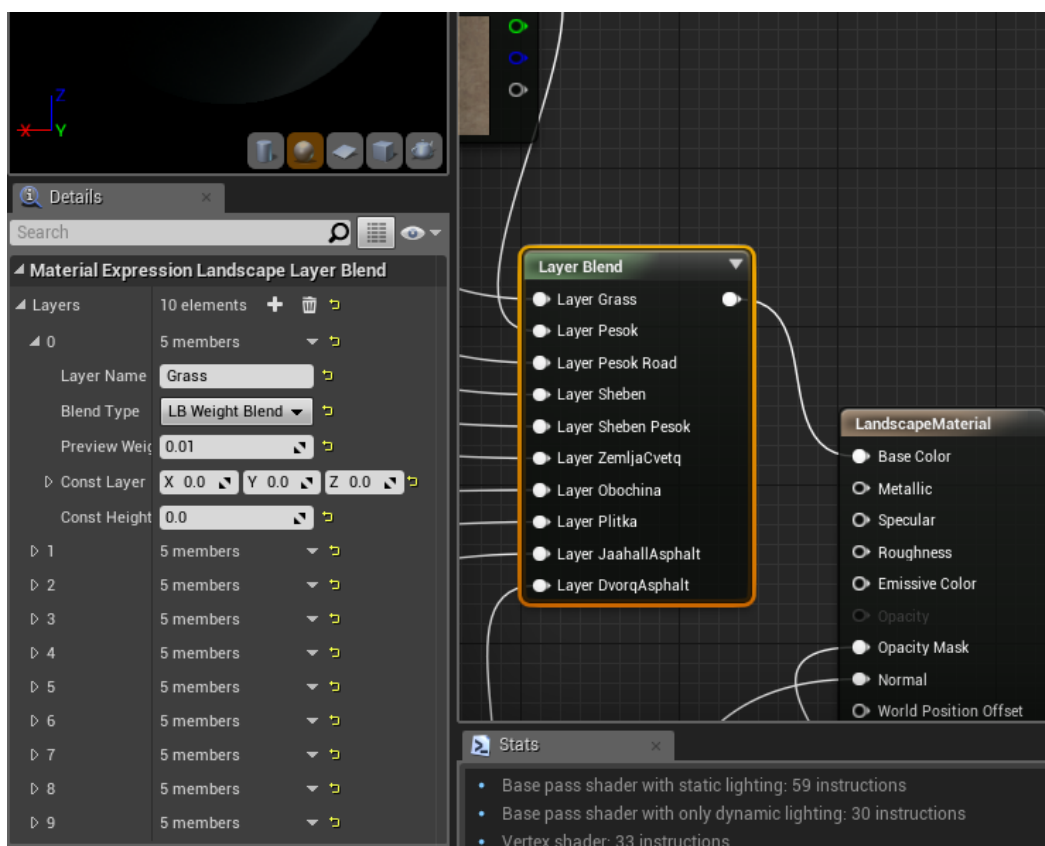


Рисунок 16. Блок LayerBlend

Слои также могут смешиваться между собой разными способами: по весу, по высоте и по альфе, смотри Рис. 17. При смешивании по альфа учитывается порядок слоёв. Например, если слой снега наносится поверх травы и камня, то при его удалении, сначала появится камень, а под ним трава.

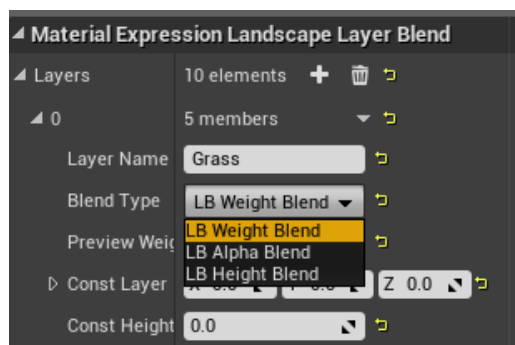


Рисунок 17. Смешивание слоев

В данной работе используются первые два способа смешивания слоев. Если смешивание по весу, как изображено на Рис. 18, – это обычное накладывание слоев друг на друга, то при смешивании по высоте нижний слой заполняет в первую очередь нижнюю часть верхнего слоя, что и показано на Рис. 19.

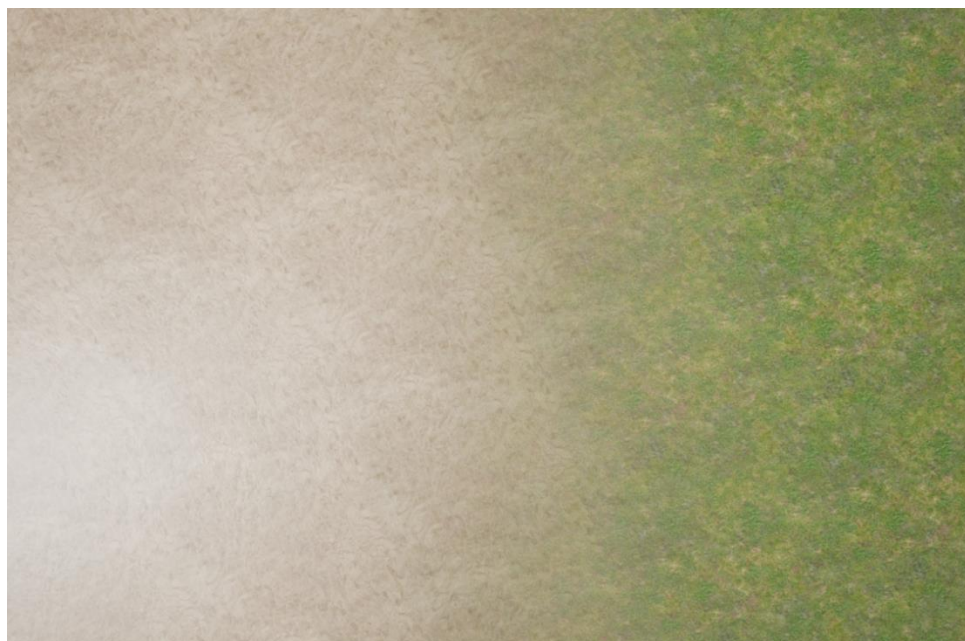


Рисунок 18. Смешивание по весу

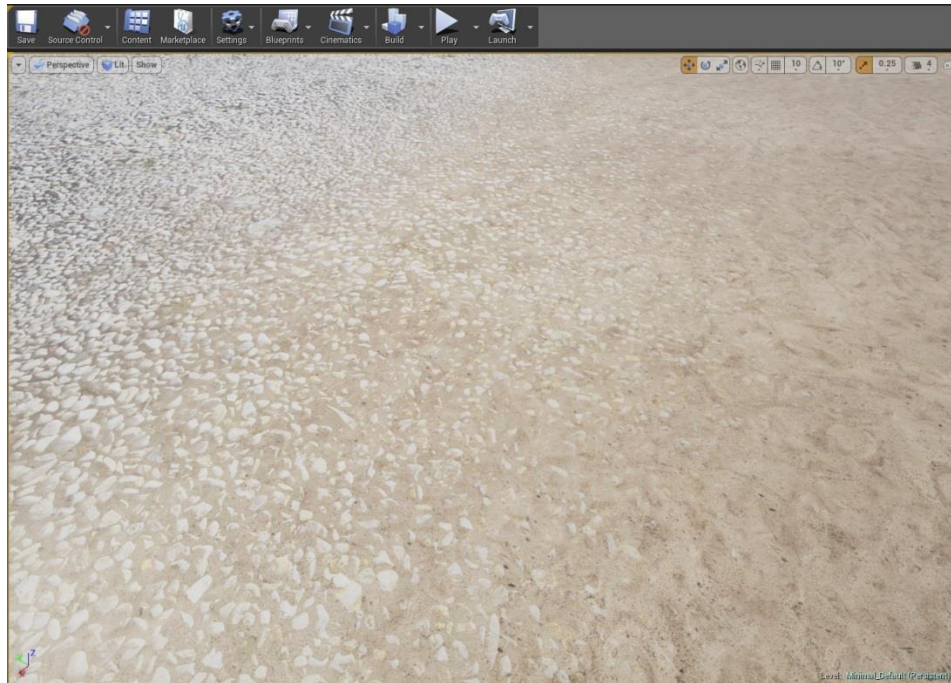


Рисунок 19. Смешивание по высоте

Какая часть текстуры ниже или выше определяется по карте высоты. В этой работе для этого используется та же текстура, но с блоком SCurve (Рис. 20), который добавляет контрастности текстуре. В итоге более темные участки находятся ниже, а более светлые, соответственно, выше.

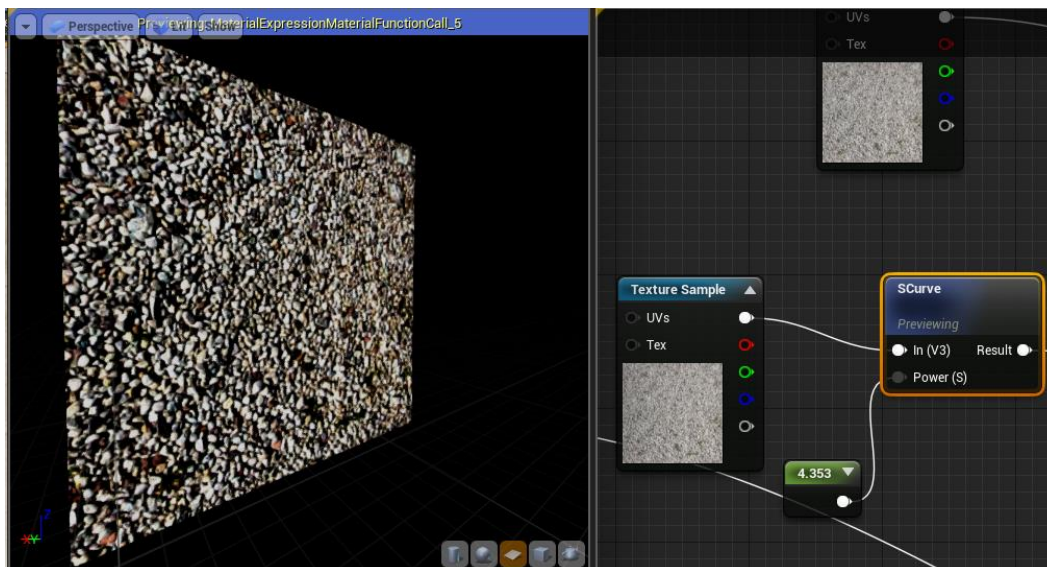


Рисунок 20. Блок SCurve

Также, для того, чтобы делать дыры в ландшафте, необходимо добавить блок Landscape Visibility Mask и подключить его к выходу Opacity Mask, что можно увидеть на Рис. 21.

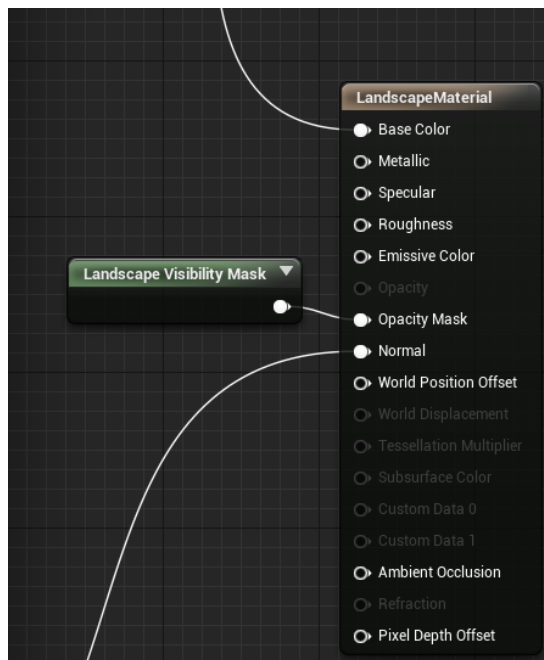


Рисунок 21. Блок Landscape Visibility Mask

Если планируется делать дыры в ландшафте, то необходимо применить также в Landscape Hole Material, смотри Рис. 22.

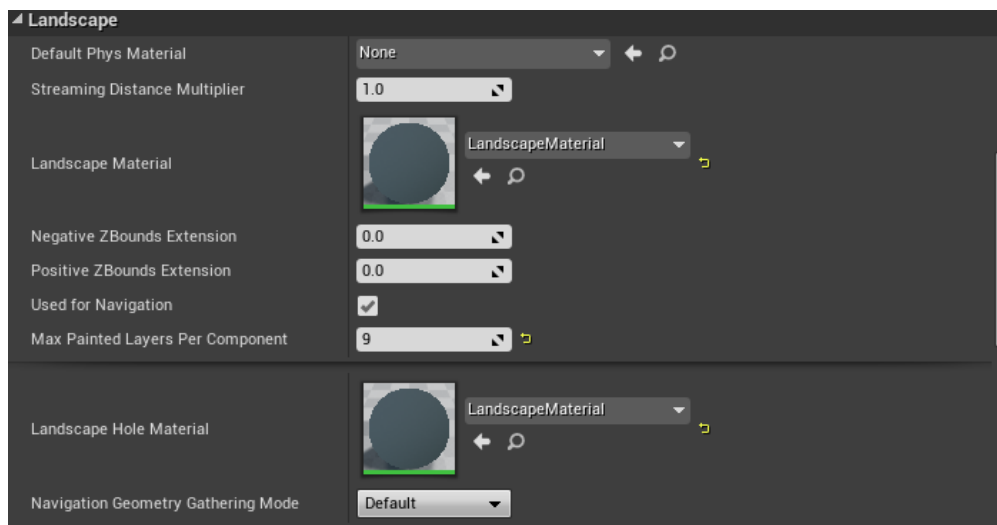


Рисунок 22. Landscape Hole Material

Последним этапом настройки слоев материала ландшафта будет создание ассетов для каждого слоя, как представлено на Рис. 23.

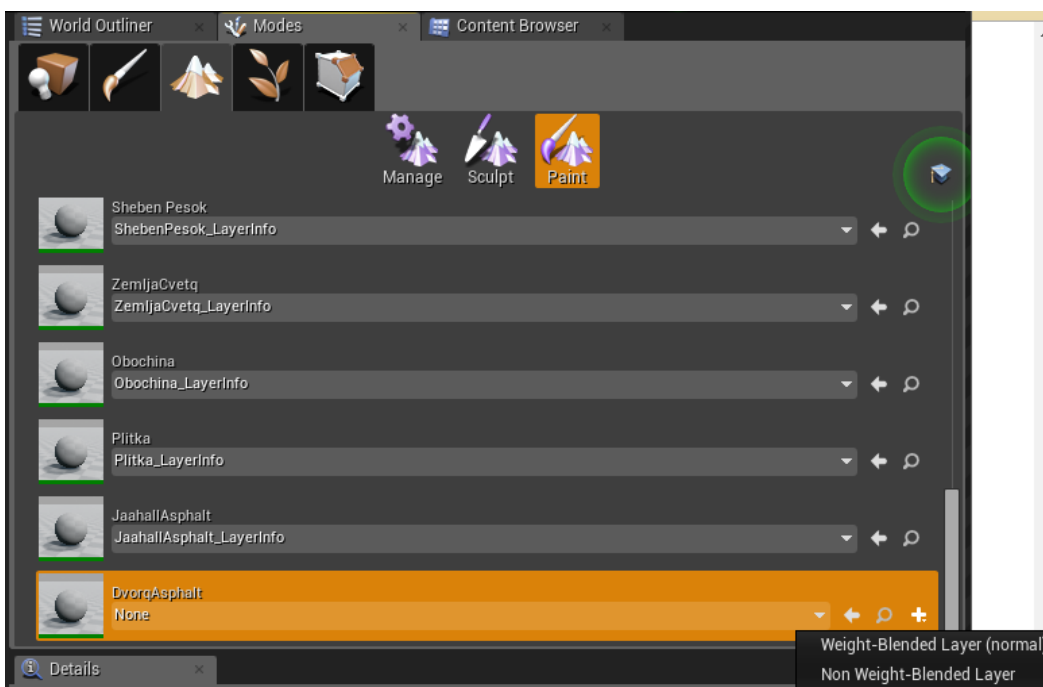


Рисунок 23. Создание ассетов

4.2 Растительность

После того, как на ландшафт раскрашен разными текстурами и на нем расставлены все объекты, работа с ландшафтом подходит к концу. Заключительным этапом становится добавление растительности на всю часть города. Растительность была взята из стартового контента Unreal Engine 4, а также скачана бесплатно из Интернета. Ссылки всех моделей деревьев и травы находятся в списке используемого материала.

Чтобы дополнить часть района деревьями, травой и кустами, сначала следует добавить их в Unreal Engine 4 и после этого открыть окно Foliage, расположенное на панели Modes (Рис. 24). Система Foliage позволяет быстро рисовать или стирать наборы объектов на ландшафте.

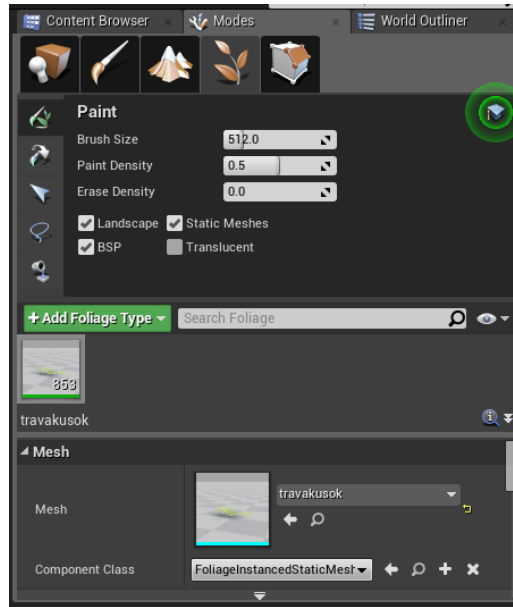


Рисунок 24. Окно Foliage

Необходимо перетащить нужные объекты из Content Browser в Foliage Type, как изображено на Рис. 25. Если используется несколько разных объектов, то наведя курсор на объекты, можно в левом верхнем углу поставить галочку, чтобы использовать определенный объект, как продемонстрировано на Рис. 26. Например, можно загрузить два вида травы в Foliage Type и одновременно рисовать ими на поверхности ландшафта. В окне Foliage можно выбирать размер кисти в Brush Size.

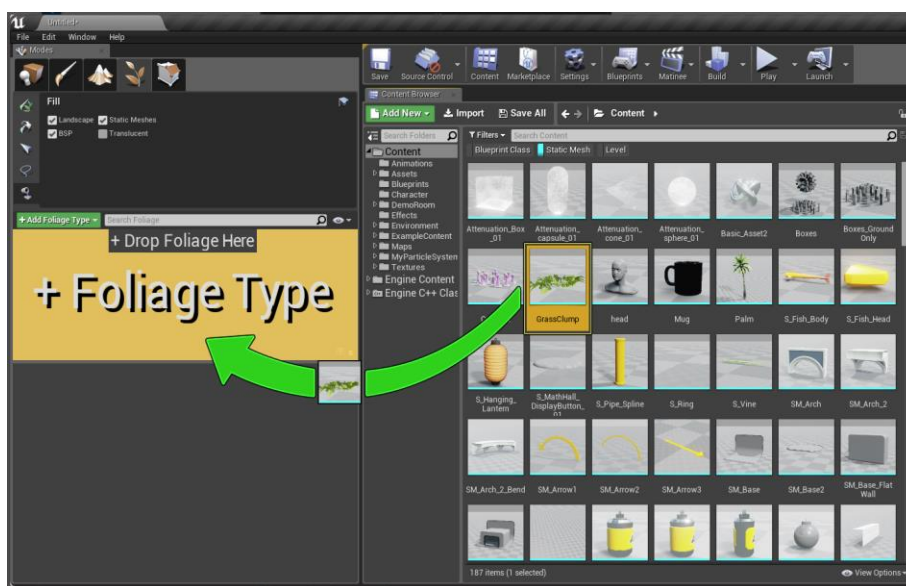


Рисунок 25. Перенос объектов

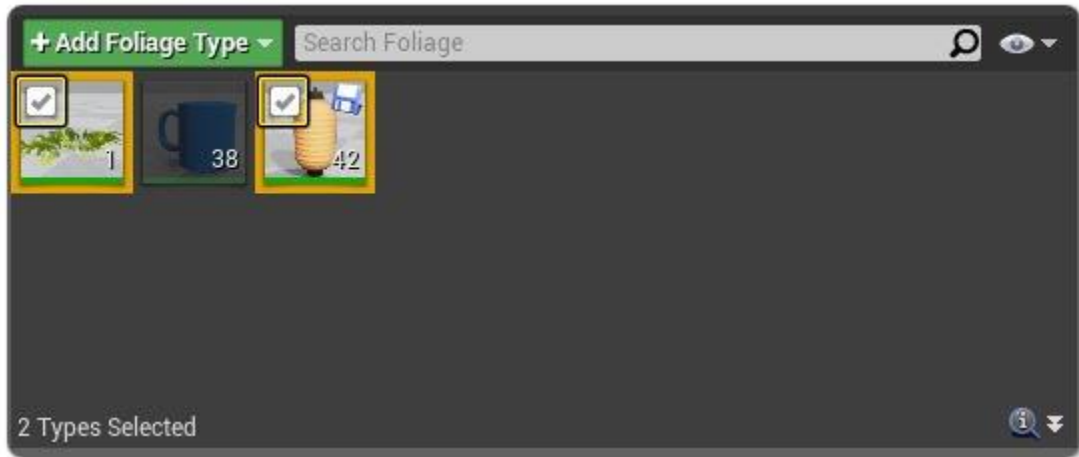


Рисунок 26. Использование конкретного объекта

5. Коллизии

Перед просчетом освещения необходимо проверить коллизии у всех объектов. Настройка коллизий определяет границы объектов для игрока. Чтобы приступить к настройке коллизии, сначала необходимо перейти режим редактирования Static Mesh объекта и удалить изначально созданную коллизию на объекте, если она ему не подходит. Для формирования новой автоматически созданной коллизии, надо выбрать параметр Auto Convex Collision в меню Collision, определить параметры в зависимости от сложности объекта и нажать Apply, как показано на Рис. 27 и Рис. 28.

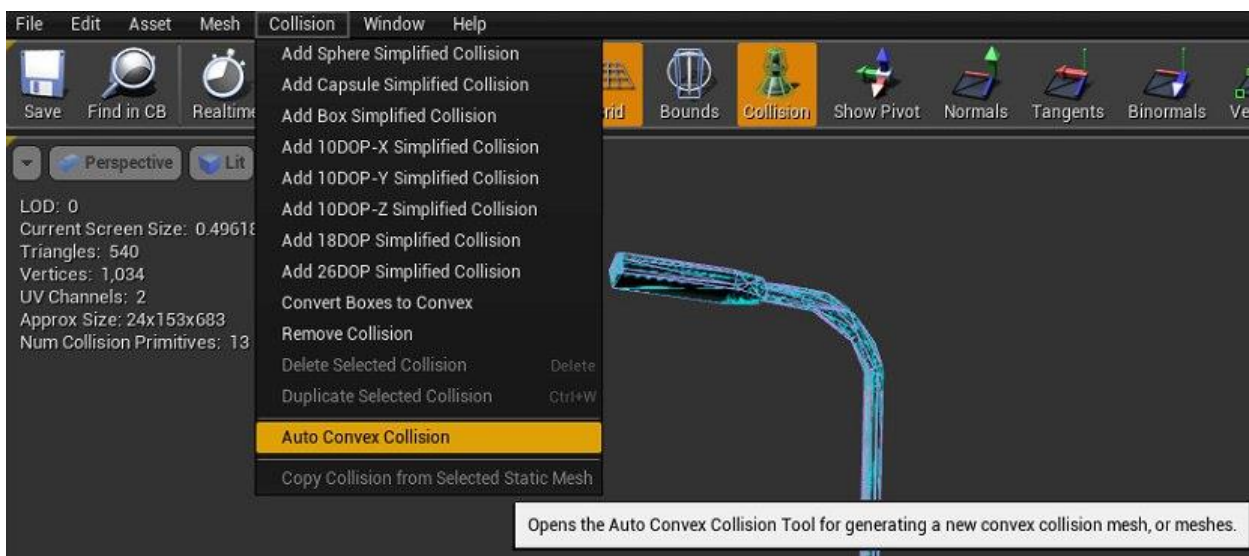


Рисунок 27. Автоматическое создание коллизии



Рисунок 28. Параметры коллизии

После применения данного параметра, Unreal Engine 4 начинает процесс создания целого ряда сложных фигур согласно геометрии объекта и соединяет их, создав меш-коллизии.

6. Настройка освещения

Для полноценного освещения сцены, необходимо добавить в нее три компонента: Sky_Sphere, SkyLight и Directional Light (Рис. 29).



Рисунок 29. Sky_Sphere, SkyLight и Directional Light

- Sky_Sphere – это модель неба в форме сферы, доступная в Unreal Engine 4 в виде блупринта неба с облаками и солнцем из стартового контента. В параметрах была настроена высота солнца, степень прозрачности облаков и их скорость.
- Directional Light имитирует свет от источника, находящегося бесконечно далеко. Этот источник освещения идеально подходит для симуляции солнечного света, который и является основным источником освещения в данной работе.
- SkyLight – захватывает свет неба и освещает текстурой неба на всю сцену. Также, он позволяет реализовать отражение текстуры неба в окнах в любом месте сцены.

SkyLight и Directional Light имеют три главных параметра: Static, Stationary и Movable. В обоих случаях будет использоваться параметр Movable, который представлен на Рис. 30, так как он позволяет получить динамические тени от движущихся на ветру деревьев и травы, а также статические тени от объектов, информация об освещении которых заложена изначально при помощи карт освещенности.



Рисунок 30. Выбран параметр Movable

Для большинства объектов в данном проекте создавалась карта освещения в Blender, которая содержит информацию об освещенности моделей. Благодаря карте освещения, свет просчитывается заранее и движку не приходится визуализировать освещение и тени в реальном времени объектам, для которых это не обязательно. Тем самым увеличивается производительность. Значение разрешения карты освещения (Lightmap Resolution) по умолчанию в Unreal Engine 4 равно 64. Этого разрешения часто не хватает для сложных объектов и его необходимо увеличивать (Рис. 31).

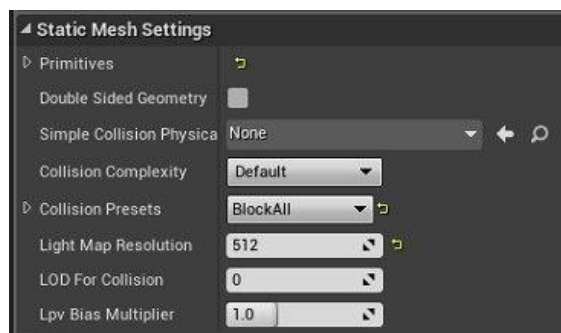


Рисунок 31. Увеличение значения разрешения карты освещения

В случае необходимости, Lightmap можно создать также с помощью Unreal Engine 4, как показано на Рис. 32, однако результаты, как правило, хуже по сравнению с созданием карты освещения в Blender.

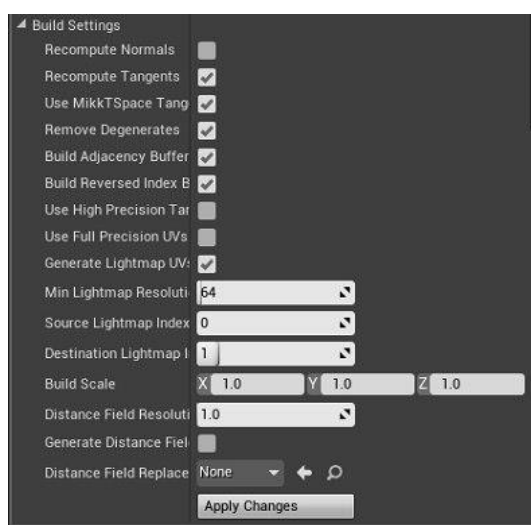


Рисунок 32. Создание Lightmap

6.1 Постобработка

Постобработка – обработка, которая накладывает эффекты на готовую картинку. В данной работе постобработка применяется на все, что попадает в поле зрения камеры персонажа, как проиллюстрировано на Рис. 33 и Рис. 34.

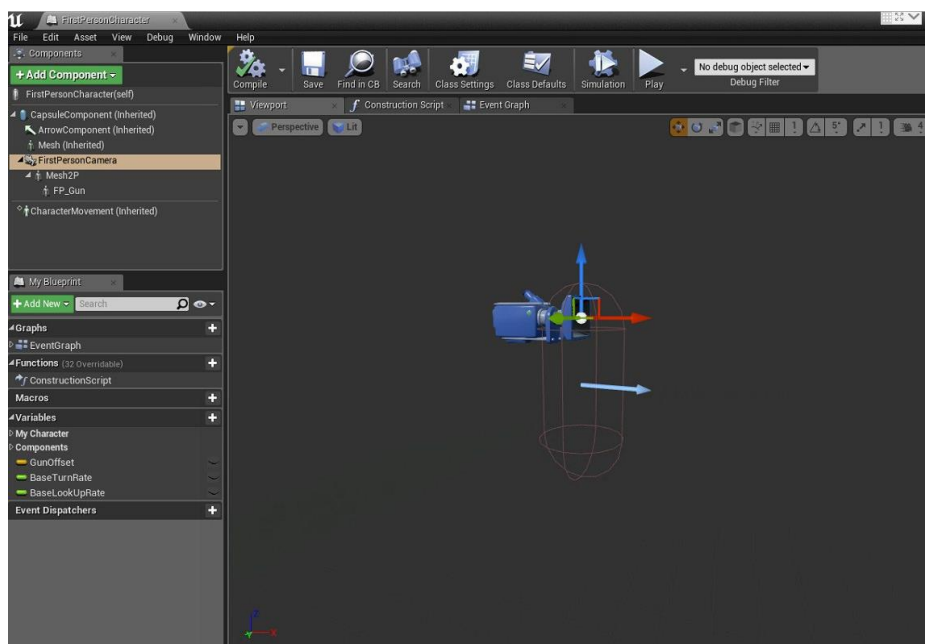


Рисунок 33. Постобработка

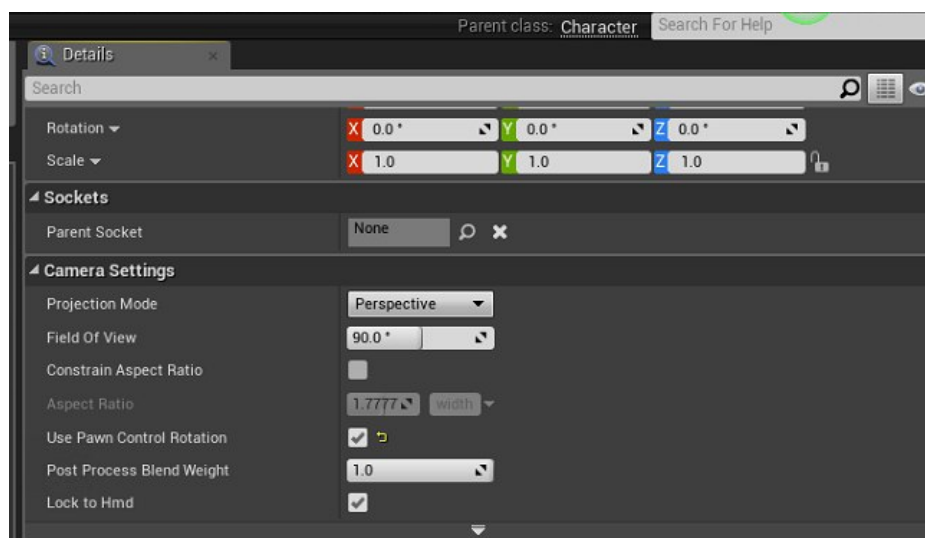


Рисунок 34. Постобработка

Из эффектов постобработки были использованы следующие: увеличение контрастности и насыщенности, добавление эффекта свечения (Bloom), добавление эффекта автовывержки, другими словами эффекта "привыкания глаз", добавление бликов света от солнца, добавление размытия при движении камеры, а также Ambient Occlusion, которые дополнительно затемняет темные участки сцены, что показано на Рис. 35 и Рис. 36.



Рисунок 35. Настройки постобработки

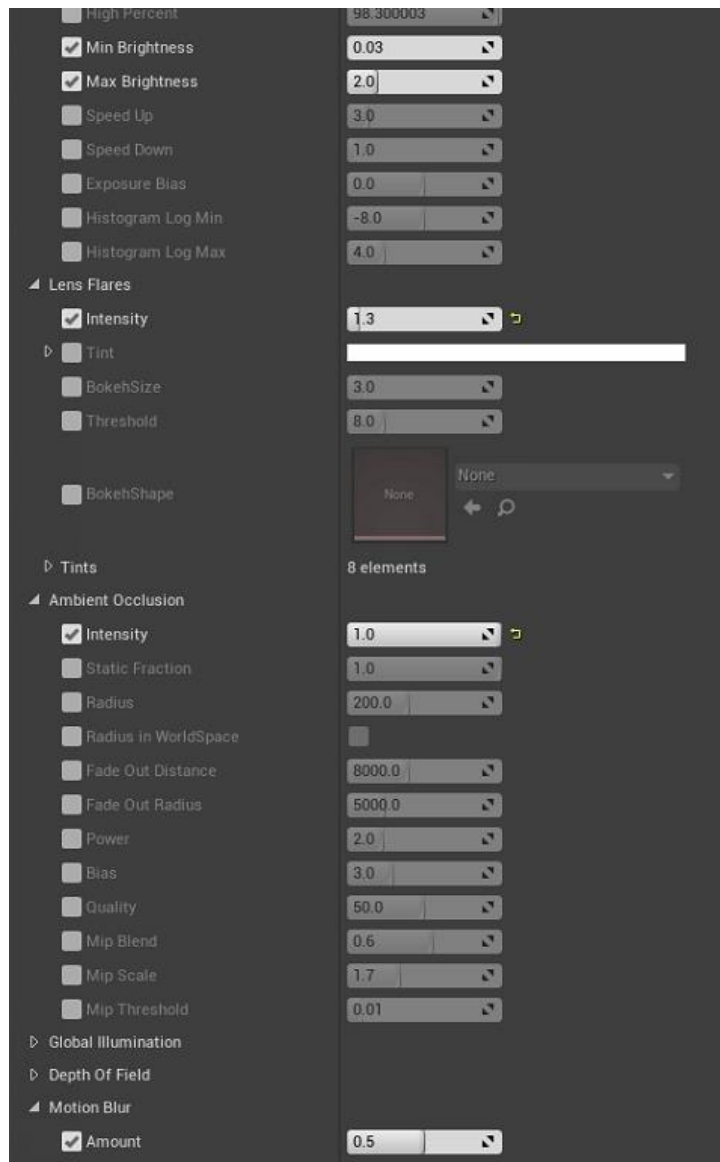


Рисунок 36. Настройки постобработки

7. Взаимодействие проекта с Oculus Rift

Unreal Engine 4 поддерживает Oculus Rift за счет использования плагина, который уже включен в игровой движок. Перед использованием необходимо скачать, установить и запустить программное обеспечение от Oculus и проверить, чтобы вышеназванный плагин был включен. Реализовать проект с помощью очков виртуальной реальности не составляет проблем. Чтобы активировать очки, необходимо в Unreal Engine 4 выбрать VR Preview в меню Play.

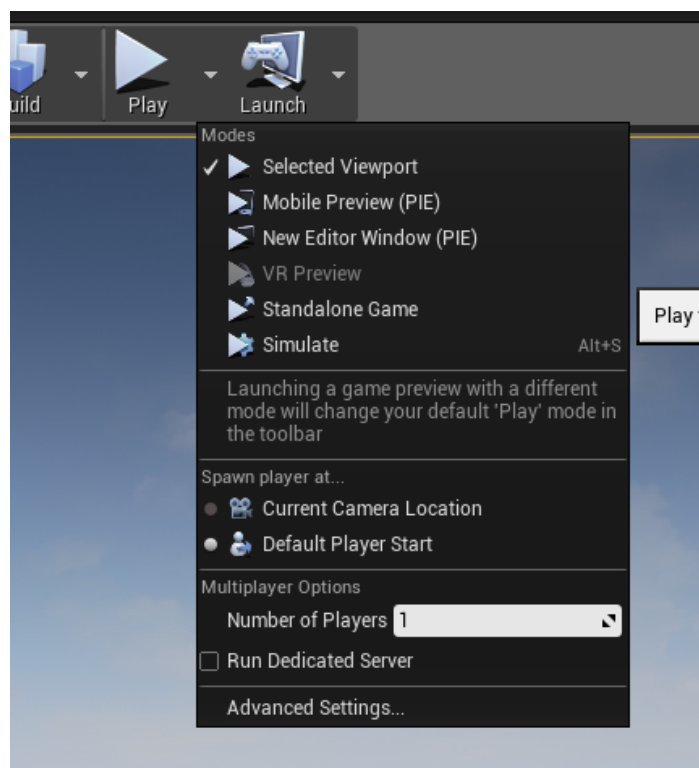


Рисунок 37. Использование виртуальных очков в Unreal Engine 4

Заклучение

Непосредственно перед выводом, несколько скриншотов готового проекта:



Рисунок 38. Скриншот 1



Рисунок 39. Скриншот 2



Рисунок 40. Скриншот 3



Рисунок 41. Скриншот 4



Рисунок 42. Скриншот 5

В заключение могу сказать, что работа была выполнена успешно и все цели достигнуты. При использовании очков виртуальной реальности Oculus Rift ощущается эффект присутствия. Соответственно, если поставлена такая цель, смоделировать часть города, то это можно реализовать с помощью бесплатных программ, таких как GIMP 2, Blender и Unreal Engine 4.

Из минусов можно отметить не совсем стабильный показатель количества кадров в секунду (fps), но при желании можно продолжить оптимизацию и добиться идеального, или близкого к идеальному, результата. Тем не менее, надо учитывать, что для комфортной работы в данном приложении необходим производительный компьютер.

Kasutatud kirjandus

- [1] Unreal Engine Documentation [Last access: 18.08.2016]. [Online]. Available: <https://docs.unrealengine.com/latest/INT/>.
- [2] Документация Unreal Engine 4 [Last access: 18.08.2016]. [Online]. Available: <https://uengine.ru/docs>.
- [3] Oculus Rift: Welcome to the future [Last access: 18.08.2016]. [Online]. Available: <http://www.cnet.com/special-reports/oculus-rift-review/>.
- [4] Unreal Engine 4 для развития своих способностей [Last access: 18.08.2016]. [Online]. Available: <https://habrahabr.ru/post/249965/>.
- [5] Material Inputs [Last access: 18.08.2016]. [Online]. Available: <https://docs.unrealengine.com/latest/INT/Engine/Rendering/Materials/MaterialInputs/index.html>.
- [6] Landscape Outdoor Terrain [Last access: 18.08.2016]. [Online]. Available: <https://docs.unrealengine.com/latest/INT/Engine/Landscape/index.html>.
- [7] Unreal Engine 4 Tutorial: Wind Effect [Last access: 18.08.2016]. [Online]. Available: <https://www.youtube.com/watch?v=bAzKVah0IZg>.
- [8] Collision [Last access: 18.08.2016]. [Online]. Available: <https://docs.unrealengine.com/latest/INT/Engine/Physics/Collision/>.
- [9] Unreal Engine 4 [Last access: 18.08.2016]. [Online]. Available: <https://www.unrealengine.com/blog>.
- [10] GIMP [Last access: 18.08.2016]. [Online]. Available: <https://docs.gimp.org/2.4/ru/gimp-first-steps.html#gimp-concepts-basic>.

Lisad

Список скачанных моделей деревьев и травы из Интернета.

1. Ель <http://tf3dm.com/3d-model/realistic-tree-pack-3-trees-95419.html>.
2. Береза <http://tf3dm.com/3d-model/birch-tree-48016.html>.
3. Пакет различных деревьев <http://www.turbosquid.com/3d-models/free-obj-mode-tree-pack/506851>.
4. Трава <https://www.cgtrader.com/free-3d-models/plant-tree/grass/grass--6>.
5. Клен <https://www.cgtrader.com/free-3d-models/plant-tree/leaf-tree/italian-maple>.
6. Дерево для заднего плана <http://www.turbosquid.com/3d-models/tree-obj-free/785923>.
7. Трава <http://tf3dm.com/3d-model/grass-pack-71861.html>.