

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»

институт информационных технологий и технологического образования
кафедра информационных технологий и электронного обучения

Основная профессиональная образовательная программа
Направление подготовки 09.03.01 Информатика и вычислительная техника
Направленность (профиль) «Технологии разработки программного обеспечения»
форма обучения – очная

Курсовая работа

по дисциплине «Технологии компьютерного моделирования»

Технологии создания и анимации 3D моделей.

Обучающегося 2 курса
Козырькова Игоря Викторовича

Руководитель:
д.п.н, профессор
Власова Е. З.

«_____» _____ 2021 г.

Санкт-Петербург
2021

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
ПОСТАНОВКА ЗАДАЧИ	4
ОСНОВНАЯ ЧАСТЬ.....	4
1.Теоретическая часть.....	4
Принципы 3D – моделирования:	4
Принципы работы UV:	5
Принцип работы текстур:	7
Принципы работы анимации:	8
Работа с Unreal Engine:	9
2.Практическая часть:	11
1.Blocking:	13
2.Middle poly:	14
3.High poly:	15
4.Low poly:	16
5.UV – развертка:	17
6.«Запечка» карт:.....	18
7.Текстурирование:	19
8.Анимирование модели:	20
9.Работа с Unreal Engine.	21
ЗАКЛЮЧЕНИЕ	26
ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ	27
ПРИЛОЖЕНИЕ А	30

ВВЕДЕНИЕ

Трехмерная графика – это раздел компьютерной графики, который посвящен созданию 3D (трехмерных) объектов.

Главная задача 3D-моделирования – это разработать зрительный объемный образ желаемого объекта. Трехмерная графика работает по принципу построения геометрической проекции объекта в трех измерениях на плоскость, такой как, например, экран компьютера с помощью специализированных программ. Однако трехмерная графика не обязательно включает в себя проецирование на плоскость.

В сегодняшнем мире 3D графика окружает нас каждый день, будь то кинематограф игры и огромное количество видеоконтента, и за всей этой красотой стоит обширный процесс создания 3D – моделей.

Цель данной курсовой работы заключается в исследовании технологии создания и анимации 3D моделей на примере трехмерной модели автомобиля.

Для решения поставленных задач необходимы следующие действия:

- Создать непосредственно 3d-модель автомобиля.
- Создать и применить текстуры к модели.
- Создать анимацию с помощью игрового движка Unreal Engine 4

ПОСТАНОВКА ЗАДАЧИ

Исследовать и описать процесс создания, текстурирования и анимации 3d – моделей на примере автомобиля

ОСНОВНАЯ ЧАСТЬ

1. Теоретическая часть

Чтобы лучше понять термины, которые используются в данной работе, обратимся к теории.

Принципы 3D – моделирования:

Построение трехмерных геометрических объектов, базируется на основе прямоугольной системы координат, которая называется «Декартова система координат» в честь французского ученого Рене Декарта.

Аббревиатура 3D – Это условное обозначение графики в трехмерном исполнении, что означает «three-dimensional» - имеющей три измерения.

Трехмерные модели подразделяют на три типа по функциональному назначению:

Объектно-ориентированное конструирование или же каркасное моделирование низкого уровня. Объекты получаемые в результате данного типа моделирования, называются каркасными или проволочными. Модели такого типа не содержат информации и поверхности. Преимущество такого типа состоит в минимальных затратах оперативной памяти.

Поверхностное моделирование: в отличие от каркасного моделирования, данный тип включает в свой состав поверхности, которые образуют визуальный контур изображаемой поверхности.

Твердотельное моделирование – это самое полное и самое достоверное построение реального объекта. Результатом построения геометрического тела таким методом является монолитный образец нового изделия.

Любая, даже самая сложная фигура состоит из множества простых фигур: такая фигура как шар или точнее сфера в компьютерной графике состоит из множества треугольников и четырехугольников. Чем количество простых составляющих больше, тем более гладкой, детализированной и сферической является поверхность.

Итак, разберемся из каких элементов состоит 3D – модель:

- 1.Вершина (Vertex) – это абстрактная геометрическая точка имеющая координаты в системе координат X,Y и Z. Вершиной она называется по той причине, что является крайней точкой либо замкнутого полигона, либо объемной фигуры.
- 2.Грань (edge) – отрезок прямой, соединяющий две вершины. Также, как и точка, грань – это не самостоятельный объект, а лишь элемент, связывающий полигоны.
- 3.Полигоны (poly, polygon) – основная функциональная составляющая: Плоская многоугольная фигура. Обычно 3d редакторы работают с треугольниками и четырехугольниками.
- 4.Нормаль – нормалью называется вектор, перпендикулярный поверхности в каждой данной её точке. С помощью нормалей работает технология сглаживания поверхностей, а также карта нормалей (Normal Map).

Непосредственно из данных элементов строиться сама объемная фигура. Но сама форма объекта — это еще не все. На 3D модель накладываются текстуры, благодаря ним, модели начинают выглядеть реалистично.

Принципы работы UV:

Начнем с того, что имея готовую 3d модель мы не имеем возможности накладывать на нее текстуры. Предварительно необходимо сделать ее развертку.

UV-РАЗВЕРТКА:

Для понимания этого этапа вспомним, например, как выглядит наша планета, она представляет собой сферу, но карты поверхности земли, которые мы привыкли видеть двухмерные, соответственно, для того чтобы отобразить 3х мерный объект на плоскости нужно его как бы разрезать.

UV-преобразование или развёртка в трёхмерной графике — это соответствие между координатами на поверхности трехмерного объекта (X, Y, Z) и координатами на текстуре (U, V). Значения U и V обычно изменяются от 0 до 1. Зачастую развертка модели производится вручную, но на сегодняшний день есть инструменты, помогающие ускорить этот процесс.

UV – развертка необходима для наклеивания на нее текстур.

Сам процесс их создания начинается с так называемой «запечки» текстурных карт.

Запечка текстур:

Суть процесса запечки заключается в следующем: мы, имея модель с небольшим количеством полигонов (low poly) делаем так, чтобы она выглядела как модель с миллионами полигонов (high poly). Это обусловлено тем, что ресурсы компьютеров ограничены и мы не можем позволить себе использовать большое количество полигонов. В результате данного процесса мы получим следующие текстурные карты:

- Карта нормали - Normal map
- Карта затенения - Ambient occlusion
- Карта кривизны - Curvature map

NORMAL MAP

Для понимания возьмем простой куб.

Карта нормалей у такого объекта будет выглядеть как сплошная синяя картинка.

По сути, это означает отсутствие Normal map. Простой синий цвет означает, что vertex нормали перпендикулярны поверхности.

. У карты нормалей очень простая технология: каждый пиксель создает на поверхности модели виртуальную вертекс нормаль.

Нейтральный голубой цвет (128,128,255 RGB) означает, что оригинальная вертекс нормаль не будет изменяться. Красный канал отвечает за отклонение нормали влево или вправо, зеленый — вверх или вниз. Синий имитирует глубину

С помощью карт нормалей мы можем создать геометрию, которой на самом деле нет, но на которой будут работать блики и затенения, как на реальной геометрии, при этом сэкономяв огромное количество полигонов.

AMBIENT OCLUSSION MAP (карта затенения)

Карта затенения показывает самые глубокие тени в объекте — в основном это тени в углублениях и на пересечениях. Эту карту используют в 3D движках для создания корректного освещения, а также она очень полезна в текстурировании.

Главная цель этой карты состоит в том чтобы создать затенение на этапе создания модели, рассчитав ее всего один раз, на одном компьютере, а не непосредственно в 3d движке на каждом так как происходит расчет освещения, а этот процесс требует большое количество мощностей компьютера.

CURVATURE MAP

Данная карта на сам внешний вид модели не влияет. Она предназначена для создания затенения в мелких углублениях, она схожа с картой ambient occlusion, но она в свою очередь создает крупные затенения между отдельными объектами, как карта кривизны в свою очередь делает карту мелких затемнений(трещин, сколов, стыков)

Принцип работы текстур:

Теперь же разберемся по какому принципу работают сами текстуры, на 3D-моделях:

В трёхмерной графике с самого начала её жизни были реализованы самые разные подходы к отображению на экране, пока не стало понятно, что

воспроизведение механизмов реальности даёт результаты более реальные, так сказать.

Как мы все знаем, мы не могли бы видеть ни один объект, если бы объекты не отражали свет, который на них падает. Все это работает очень просто — луч света падает на стол, отражается и попадает вам на сетчатку глаза.

В зависимости от поверхности, от ее состояния (отражает ли оно, как зеркало, или имеет шероховатость, есть ли капли грязи / воды / кофе на поверхности) вам в глаз приходит уже видоизмененный луч (скажем так).

То есть то, как мы видим объекты, выстраивается из того, как свет отражается от поверхности.

Текстуры подчиняются принципу Physically Based Rendering, что означает “Основанный на физике рендер” и описывает, какие параметры должны учитываться, чтобы объект выглядел максимально реалистично.

Основные PBR параметры:

- Base Color (цвет)
- Roughness. (шероховатость)
- Metallic. (металлик)
- Height. (высота)

Эти функции описывают поведение луча при взаимодействии с поверхностью: как он рассеивается (то есть проникает в верхний слой и возвращается), зеркально отражается или проходит через прозрачный материал.

Принципы работы анимации:

Начнем с того, что любая 3D-модель, которая имеет способность двигаться в 100% случаев имеет в своей структуре так называемые «кости» (или от англ. bones). Эти «кости» выполняют схожие функции с костями в теле человека. Главная их задача, это дать понять самой модели относительно каких точек ей двигаться.

Принцип заключается в том, что мультипликатор или моделер создаёт скелет, представляющий собой как правило древообразную структуру костей, в которой каждая последующая кость «привязана» к предыдущей, то есть повторяет за ней движения и повороты с учётом иерархии в скелете. Далее каждая вершина модели «привязывается» к какой-либо кости скелета. Таким образом, при движении отдельной кости двигаются и все вершины, привязанные к ней. Благодаря этому задача аниматора сильно упрощается, потому что отпадает необходимость анимировать отдельно каждую вершину модели, а достаточно лишь задавать положение и поворот костей скелета.

Например, имея, допустим, модель человека, нам необходимо анимировать его руку. Чтобы это сделать нам необходимо распределить кости. Главная кость будет находиться в центре самой модели человека, именно от ее положения будет зависеть положение всей модели. Далее необходимо поставить кость в саму руку, и ее в свою очередь необходимо привязать к главной кости. Дело в том, что рука не может двигаться относительно мировых осей координат, ей необходимо дать систему координат, в центре которой будет главная кость, относительно которой будет и двигаться рука. Ведь не может двигаться рука независимо от самого тела.

Работа с Unreal Engine:

Практически вся работа с движком и объектами в сцене происходит в системе под названием Blueprint.

Blueprint (англ. blueprint - план, чертеж) — система визуального скриптинга, представляющая собой визуальный интерфейс для создания элементов, используемая в игровом движке Unreal Engine. Данная система позволяет использовать почти полный потенциал программирования. Вся работа в этой системе осуществляется при помощи нодов.

Ноды(узлы) – это функциональные блоки, благодаря которым мы можем создавать всевозможные алгоритмы, фактически с их помощью мы делаем блок

схему, по которой уже и работает алгоритм на автоматически построенном коде на C++.

Конечно, используя UnrealEngine, мы не ограничены только использованием Blueprint, у нас есть возможность писать наш алгоритм сразу на C++.

Говоря об анимации автомобиля в движении, мы подразумеваем использование встроенных инструментов. Unreal Engine предоставляет уже готовую физико-математическую модель движения автомобиля, благодаря этому, пользователю остаётся лишь, применить эти возможности к своей модели: распределить точки движения и оси вращения на костях, задать силу деформации подвески, мощность и крутящий момент двигателя, передаточные числа коробки передач, силу трения колес с дорогой, массу автомобиля, центр тяжести, звук автомобиля и т.д.

2. Практическая часть:

В данном разделе будет представлен процесс создания и анимации 3D модели автомобиля.

Для начала разберемся, как происходит сам процесс моделирования.

Весь принцип моделирования построен на том, что мы из простых геометрических фигур или в 3d это называется примитивами, пытаемся сделать более сложную форму.

Изменение же самих фигур основано на принципе полигонального моделирования.

Полигональное моделирование – низкоуровневое моделирование, которое позволяет визуализировать объект с помощью полигональной сетки.

Полигональные сетки состоят из таких подобъектов:

- вершина – точка соединения ребер, их может быть сколько угодно;
- ребра – линии, выступающие границами граней;
- грани или полигоны – ячейки сетки, участки плоскости, которые имеют чаще всего треугольную или четырехугольную форму.

Полигональная сетка может состоять из огромного количества одинаковых ячеек.

Для создания трехмерной модели объекта необходимо работать с подобъектами. Это реализуется с использованием таких режимов моделирования, как:

- Ву vertex (редактирование по вершине). В данном режиме мы можем выделять отдельные точки, а также массив точек. Их мы можем двигать, масштабировать и вращать. Так же в зависимости от используемого ПО варианты инструментов могут отличаться, но вот список основных:

1. Weld (инструмент слияния выбранных точек)

2. Chamfer (создания фаски от выделенной точки)
 3. Extrude (выдавливание формы на месте выделенной точки)
- By edges (редактирование граней). В данном режиме имеется возможность трансформировать грани. Инструменты схожи с инструментами, что используются для точек, но есть и уникальные:
 1. Extrude.
 2. Chamfer.
 3. Weld
 4. Bridge (создание моста из полигонов между выделенными гранями).
 5. Connect (создание новых граней между выделенными)
 - By border (редактирование граней у которых отсутствуют соседние полигоны, о есть фактически это периметр отверстия в модели). В данном режиме используются следующие инструменты:
 1. Extrude
 2. Chamfer
 3. Bridge
 4. Connect
 5. Cap (Закрытия отверстия полигонами)
 - By polygons (Редактирование по полигонам). В данном режиме используются следующий список инструментов:
 1. Extrude
 2. Bridge
 3. Bevel (практически тоже самое, что и Extrude, только имеет возможность редактирования размеров полигона на конце выдавливания)
 4. Inset (создание пропорционального полигона выделенному полигону).

От программы к программе эти инструменты могут немного отличаться или иметь просто другое название, но в основном принцип работы во всем доступном ПО не отличается друг от друга и основан на использовании этих режимов и инструментов.

Теперь перейдем к созданию нашего автомобиля и разберемся в основных этапах работы.

1.Blocking:

Процесс создания 3d модели начинается с этапа под названием – blocking. На данном этапе нам необходимо создать простейшие объекты и формы, соответствующие реальным размерам и пропорциям объекта. Например, мы создаем примитивы по форме и размерам кабины, капота, колес и т.д. В дальнейшем эти объекты помогут при более детальном моделировании не уйти от реальных форм и размеров. Для нас это важно, так как если размер не будет реален, это может отразиться на итоговом разрешении текстур, а если будет нарушена форма объекта, то это просто будет выглядеть нереалистично.

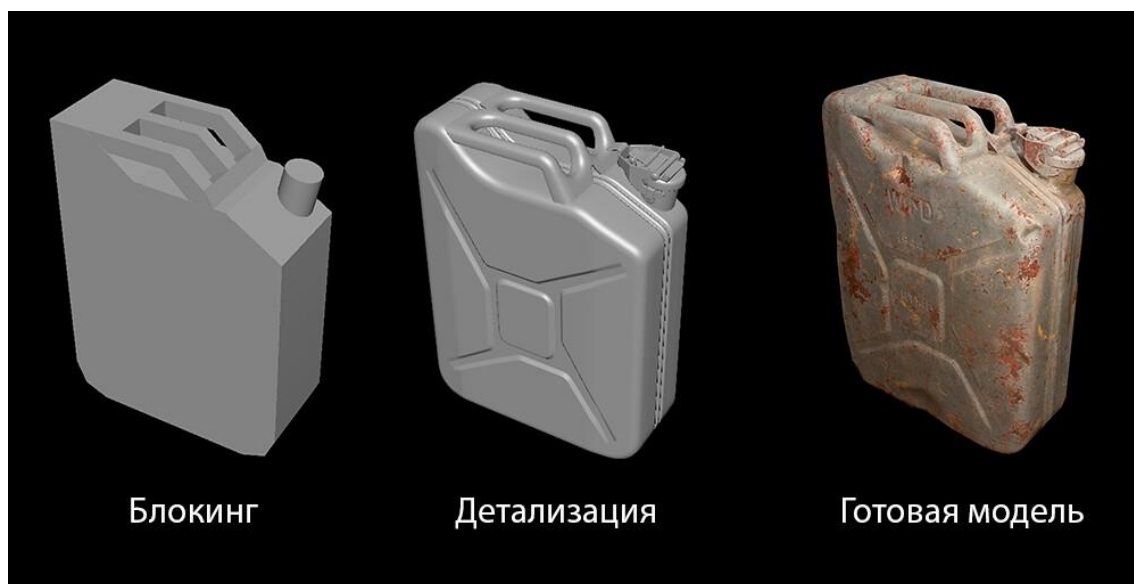


Рис. 1

В результате данного этапа мы должны получить примитивную модель с основными формами и размерами.

2.Middle poly:

На данном этапе наша основная задача создать всю основную форму объекта. Детализация на этом уровне значительно выше и в нашем случае количество полигонов может превышать сотни тысяч. Фактически на этом этапе мы и создаем нашу машину, создавая даже самые мелкие детали. При работе нам стоит обратить внимание на полигональную стеку. В будущем будет необходимо создать высокополигональную модель, а для этого на этапе middle poly необходимо создавать сетку особым образом.

Дело в том, что создание high poly достигается множественным разбиением сетки на более мелкие полигоны, и для корректной работы данной системы, необходимо при моделировании создавать как можно меньше треугольников и фигур с количеством углов больше 4, то есть четырехугольники для нас в приоритете. Это работает следующим образом: один полигон разбивается на 4 равных части, и новые полигоны принимают усредненные координаты между соседними полигонами, так происходит сглаживание формы.

В результате, на данном этапе у нас должна получиться полностью готовая модель машины, но она нам не подходит, так как ее количество полигонов слишком велико, и при этом сама модель выглядит недостаточно хорошо.

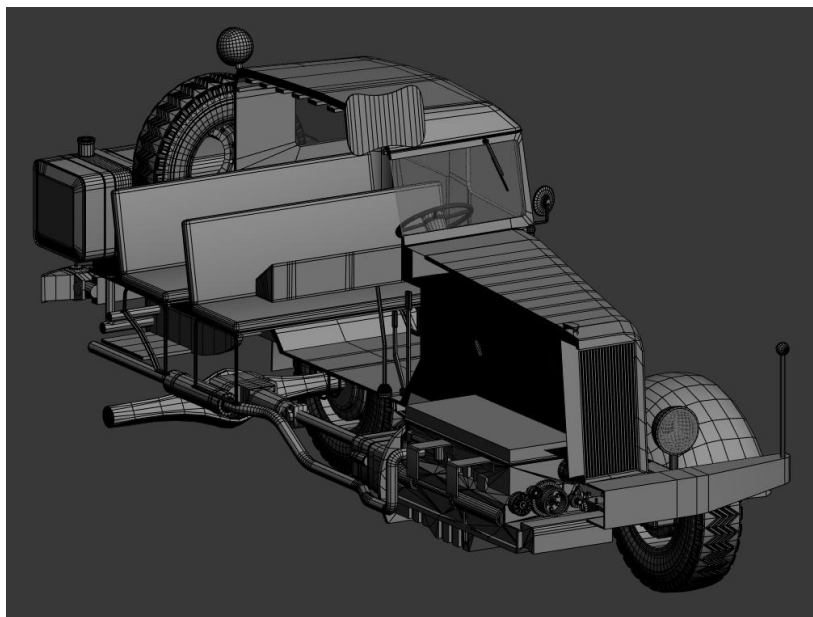


Рис. 2

На данном рисунке мы видим, что нет половины у многих частей модели. Это обусловлено тем. Что так упрощается процесс моделирования. У симметричных частей будет рациональнее сделать только половину, а вторую часть просто скопировать.

3.High poly:

После создания middle poly, мы дважды копируем нашу модель и распределяем на разные слои отображения: одна из копий – это наша high poly, а вторая копия – это будущая low poly.

Разберемся с high poly. Вся работа на этом этапе в основном заключается в том, что мы подготавливаем нашу полигональную сетку с middle poly для Low poly. Нам необходимо привести сетку к такому виду, что у нас останутся в основном четырехугольники, а также наша задача заключается в добавлении поддерживающих петель(loops). С помощью них наша геометрия не будет искривляться в результате разбиения в тех местах где мы добавим эти петли, и в зависимости от их расположения мы будем получать различную степень сглаживания: чем ближе поддерживающие петли к граням модели, тем более острая получится грань после разбиения, соответственно, чем дальше, тем более плавной получится грань.

В результате, на этом этапе мы получим такую же по форму модели, как и на middle poly, только имеющую уже не сотни тысяч, а миллионы полигонов, в нашем случае их количество около 30 млн.

Цель высоко полигональной модели – это выглядеть наиболее приближенно к реальной машине, а большое количество полигонов необходимо для того, чтобы сгладить все грани, ведь в реальном мире нет ничего идеально острого, все объекты имеют даже незаметную фаску.



Рис 3

4.Low poly:

Теперь нам необходимо максимально уменьшить количество полигонов. Мы берем модель с middle poly и начинаем удалять все лишнее: грани, точки и полигоны. При этом нам нужно избегать полигонов с количеством углов более четырех, а самым лучшим вариантом будет использовать треугольники, так как 3D движки работают с треугольниками и их вертекс – нормальями, поэтому при экспорте модели в движок, полигоны с количеством углов от пяти и более разбиваются на треугольники и часто это может происходить некорректно.

В результате упрощения всей геометрии мы получим модель с 43000 полигонами, данное количество вполне приемлемо для модели такого рода.



Рис. 4 Low poly

Чтобы заметить разницу между high poly и low poly можно заострить свое внимание на колесах, на high poly мы видим рельеф протектора, на low poly такого нет, так как это заметно уменьшает количество полигонов. Наличие протектора на шинах мы обеспечим, используя карту normal после «запечки».

5.UV – развертка:

Далее мы разрезаем нашу модель на отдельные кусочки для того, чтобы у нас появилась возможность наносить текстуры на модель.

Обычно, для небольших моделей характерно располагать все куски на одном квадрате UV. Дело в том что размер этого квадрата ограничен, по UV Он размером $[0;1]$ по U и V соответственно, поэтому чем больше по размерам наша модель, тем большее и ее размер поверхности, следовательно нам нужно поместить большую площадь на квадрат UV, и в результате снижается разрешение текстур, что

сказывается на внешнем виде модели. Поэтому для того, чтобы сохранить разрешение текстур, мы будем использовать 6 текстурных сетов или , проще говоря, 6 квадратов UV. На них будут располагаться такие элементы, как:

- Body (кабина и мелкие элементы на ней)
- Bottom (Элементы рамы, подвески)
- Glass (стекла кабины и фар)
- Interior (все элементы интерьера)
- Other (навесное оборудование: ступени, крылья и т.д)
- Wheels (колеса и диски)

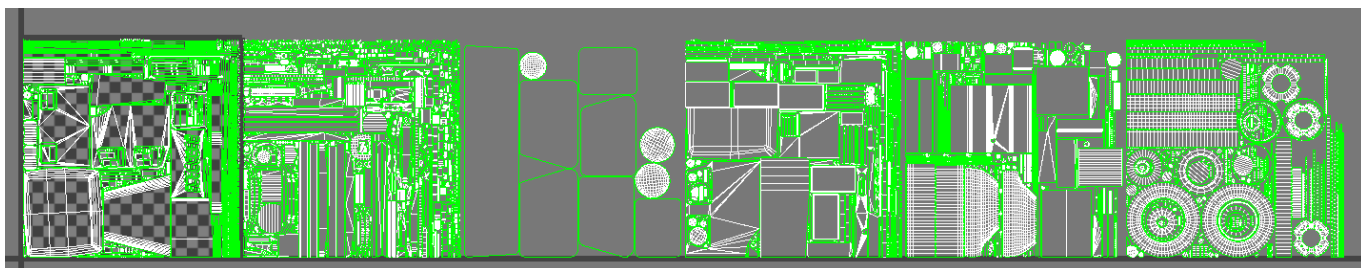


Рис. 5 UV - развертка.

Самое ресурсозатратное для 3D движка – это не количество полигонов, а количество текстур и используя 6 текстурных сетов мы заметно утяжеляем модель, но при этом мы заметно повышаем разрешение текстур, тем самым улучшая внешний вид модели.

6.«Запечка» карт:

На этом этапе нам необходимо выполнить следующее: мы должны создать текстуры normal, curvature и ambient occlusion карт для нашей low poly модели на основе high poly. Этот процесс выполняется с помощью специального ПО. Мы закидываем нашу low poly и high poly модель, далее мы указываем разрешение карт (в нашем случае это 4096*4096 пикселей), и на основе высоко полигональной модели происходит расчет карт. В результате мы получаем модель, которая выглядит как высоко полигональная, при этом имея всего 44359 полигонов.

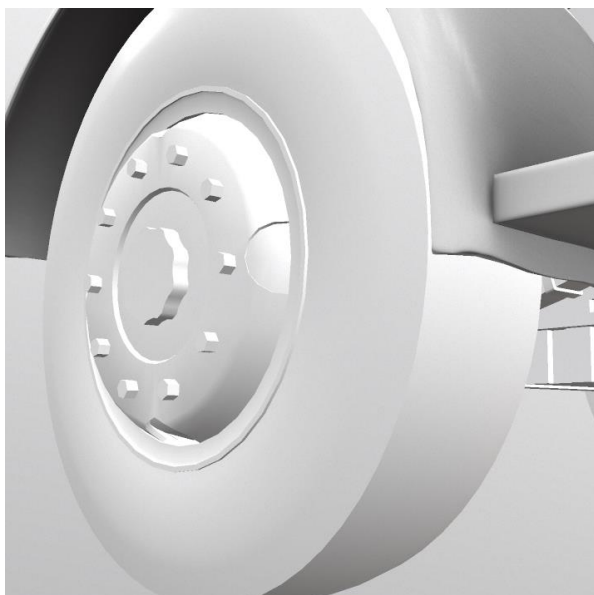


Рис.6 До «запечки»



Рис.7 После «запечки»

На данных рисунках видно как low poly модель выглядела до и после «запечки» текстурных карт.

7.Текстурирование:

Теперь стоит разобраться как происходит сам процесс текстурирования, для этого я использую программу Substance Painter

1 этап: назначение материалов: в библиотеке данного ПО имеется множество уже готовых материалов и мы просто выбираем необходимый и назначаем на нужные объекты или полигоны, а если же нам материал не подходит мы можем настроить все его параметры.

Весь принцип назначения материалов и т.п. заключается в использовании так называемых слоёв-масок. Основные задачи маски - маскировать или скрывать какую-либо область изображения или ограничивать область действия какого-либо инструмента. Маска - это своего рода черно-белое изображение, содержащее черный цвет, белый цвет и все оттенки серого. Белый цвет проявляет (100% непрозрачности), черный цвет - маскирует (100% прозрачности), а серый цвет - предназначен для создания разной степени прозрачности.

2 этап: создания разнообразия по цвету и отражениям. В жизни нет практически ничего идеально однородного, как по цвету так и по отражениям. Это достигается путем создания слоёв с немного отличающимися цветами. Путем использования неоднородных масок мы можем создать разнообразие. Такой же принцип мы будем использовать и для roughness(шероховатость) канала.

3 этап: добавление царапин, и сколов, грязи в трещины ямки и т.д.. Это достигается путем использования генераторов встроенных в Substance Painter. Они могут, используя карту нормали, создавать маски, с помощью которых и можно добавить грязь, пыль, царапин, и т.д

При выполнении всего описанного нами процесса мы получаем следующий результат:

8.Анимирование модели:

Теперь когда мы имеем готовую модель и текстуры, нам необходимо создать анимацию. Для этого вернемся в программу, в которой мы моделировали машину и добавим ей кости. Для начала проверим что центр нашей модели находится ровно в начале координат по осям X и Y, а также чтобы модель смотрела в направлении оси X, это необходимо для корректной работы 3D движка. Теперь нам необходимо разместить главную кость в координаты (0;0;0), относительно нее будет происходить движение всей модели, колес, подвески. Далее разместим кости в центре колес, а также в центре рессор и по краям заднего и переднего мостов, а также в центре кардана, для того чтобы он крутился вместе с движением машины как в реальном мире.

Теперь назначим иерархию костей: кости колес, заднего и переднего мостов, и кардана назначаем к главной кости, кости рессор привязываем к костям колес, так как положение подвески зависит от положения колес.

Теперь привязываем нашу модель костям: Корпус, рама и все что на навешено на них привязываем к главной кости (Эти части будут менять координаты вслед за изменением координат главной кости). Модель колес,

кардана, мостов и рессор привязывается к костям колес, кардана, мостов и рессор соответственно.

Далее для корректной работы подвески необходимо назначить силу воздействия кости на рессору, то есть чем полигоны ближе к кости, которая находится в центре рессоры, тем сильнее координаты этих точек будут соответствовать координатам кости. Простыми словами это значит, что в центре рессоры деформация будет сильнее чем по краям.

9. Работа с Unreal Engine.

Импортируем модель в движок. После добавления нашей модели в движок мы начинаем с того, что создаем Animation blueprint для нашей машины в окне контента нажав правой кнопкой мыши и выбрав нашу модель. Именно этот элемент будет отвечать за ее движение. Нажав правой кнопкой мыши, мы находим в списке Mesh Space Ref Pose и к этому элементу подключаем элемент Wheel Handler for WheeledVehicle. Мы создали необходимые элементы для работы нашей физики автомобиля, но если мы остановимся только на этом у нас будет некорректно работать движение переднего и заднего мостов. Чтобы это исправить нам нужно сделать так чтобы кости мостов брали значение координат у колес, это позволит им двигаться вслед за колесами. Для этого создадим элементы под названием Copy bone и выберем кости на которую нужно скопировать значение и кость, с которой нужно скопировать это значение, так мы делаем для всех костей переднего и заднего моста.

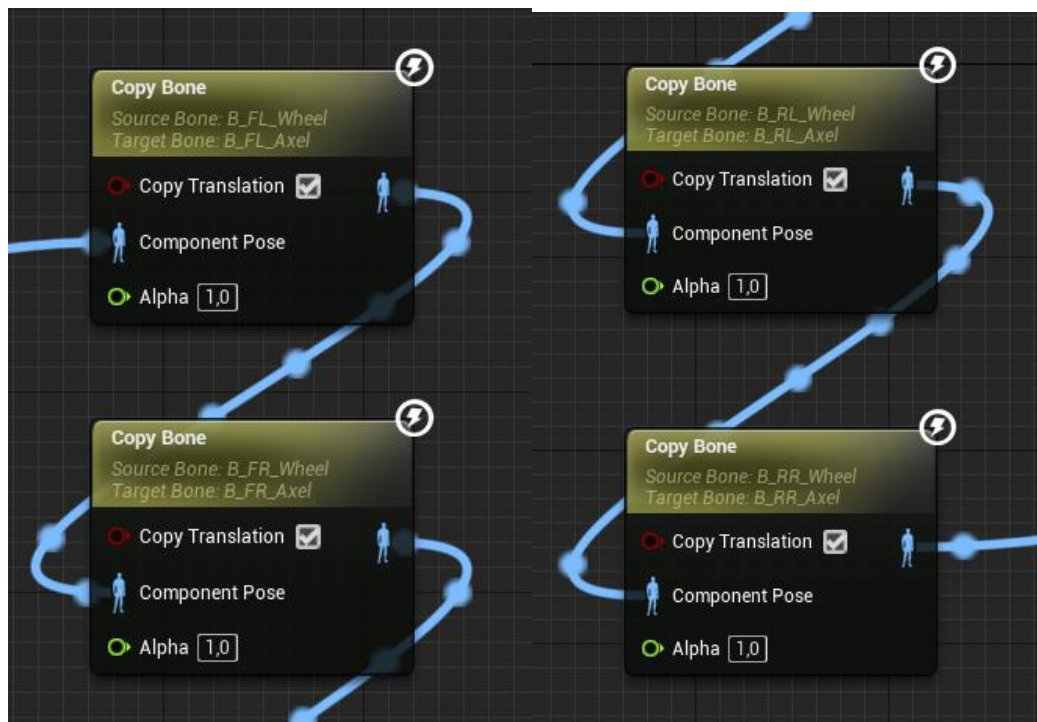


Рис. 8

Рис. 9

Для того чтобы помимо движения самого автомобиля у нас крутился кардан мы добавляем элемент под названием **Rotation Angle**(угол вращения колес), соединяем его с блоком **multiply**(умножение) , так как колеса вращаются значительно медленнее, чем кардан. В моем случае угол вращения колес умножается на 5.

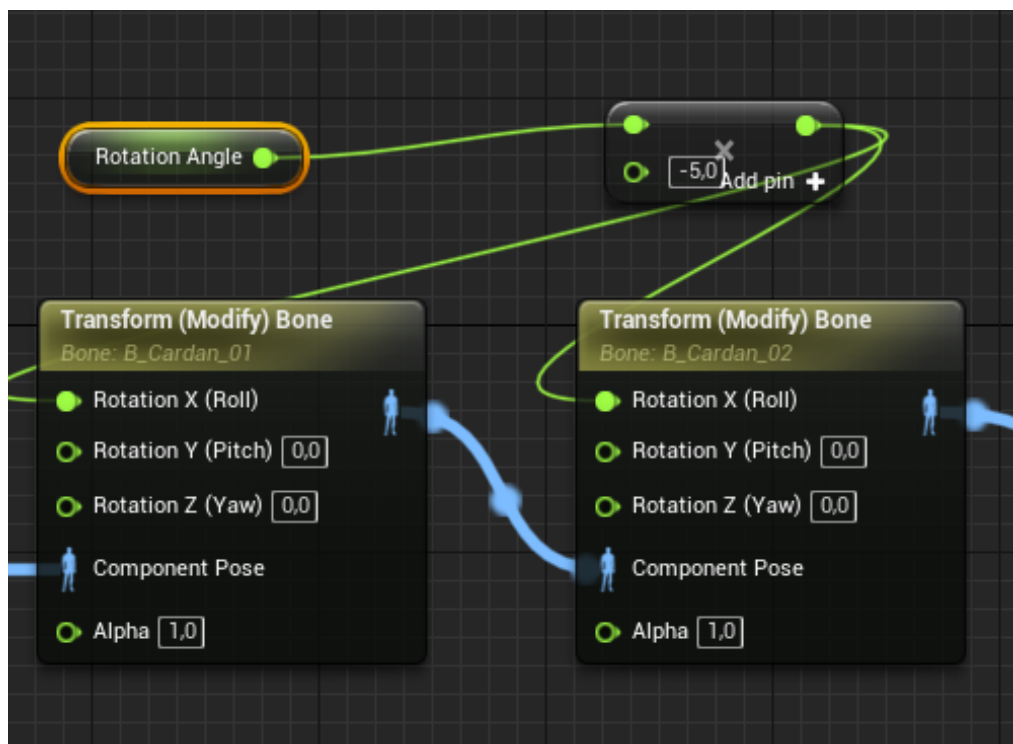


Рис. 10

Далее нам необходимо настроить колеса. В списке контента добавим Blueprint class под названием VehicleWheel для передних и такой же блок для задних колес. Открыв эти меню мы можем настроить массу наших колес, их радиус, а также угол поворота колес. Для задних колес мы поставим значение поворота в 0 градусов, а для передних 35 градусов.

Следующим этапом будет назначить необходимые клавиши для управление нашим автомобилем. Переходим ко вкладке Settings, далее project settings. Затем в разделе Engine находим вкладку input, далее в разделе Bindings нажимаем на плюс и добавляем необходимые нам клавиши, такие как: W(вперед),A(влево),S(назад),D(вправо),Space(ручной тормоз),L(свет),Mouse Y и Mouse X(оси движения мышью).

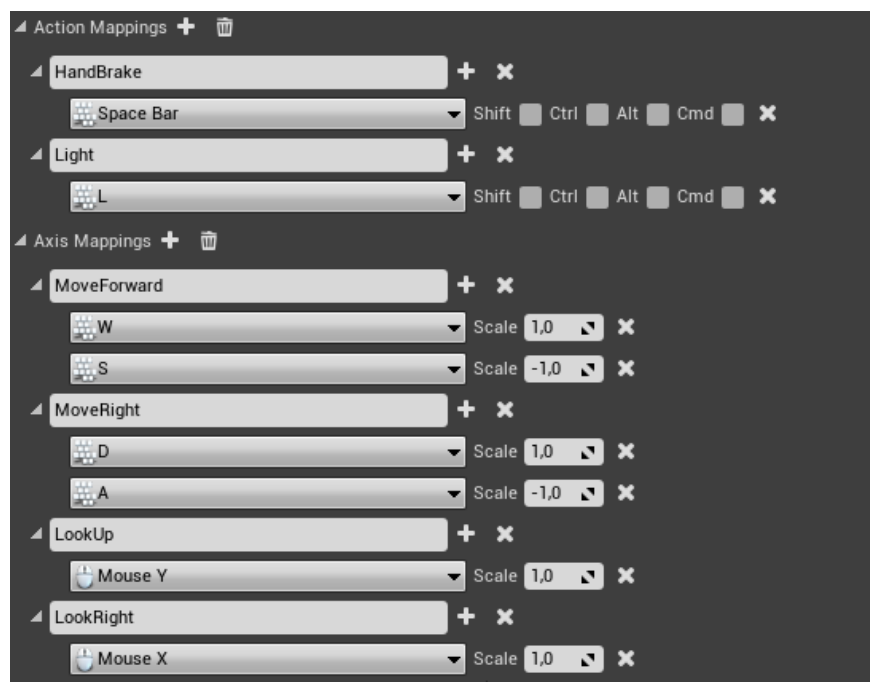


Рис. 11

Добавив кнопки, нам необходимо обеспечить их работоспособность, в разделе контента создадим Blueprint, в котором добавим блоки, соответствующие названиям, кнопок которые мы добавляли ранее. Также создадим блок VehicleMovement – это главный блок в этом blueprint, отвечающий за движение машины. Соединим имеющиеся ноды следующим образом:

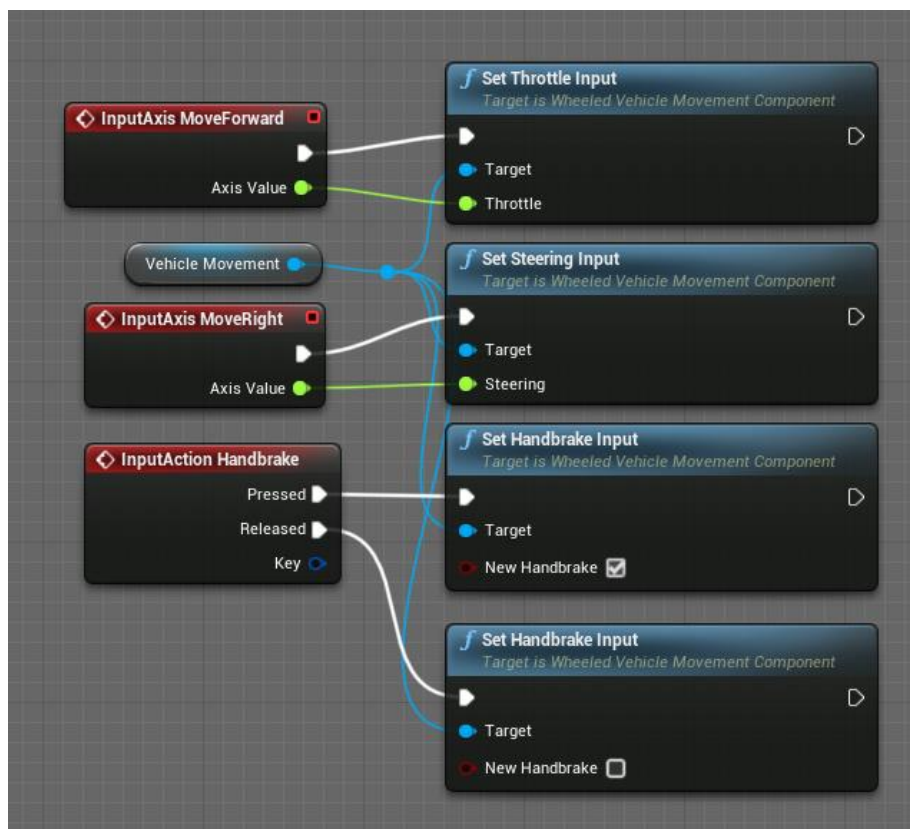


Рис. 12

Теперь нам нужно добавить нашей машине звук. В реальном мире звук автомобиля изменяется в зависимости от оборотов двигателя, то есть чем выше обороты, тем выше частота звука. По такому же принципу будет работать звук в нашем случае. Для этого мы создаем ноды в уже созданном blueprint, которые будут брать значение оборотов двигателя из блока Get Engine Rotation Speed и передавать их в Set Pitch Multiplier, выглядит это следующим образом:

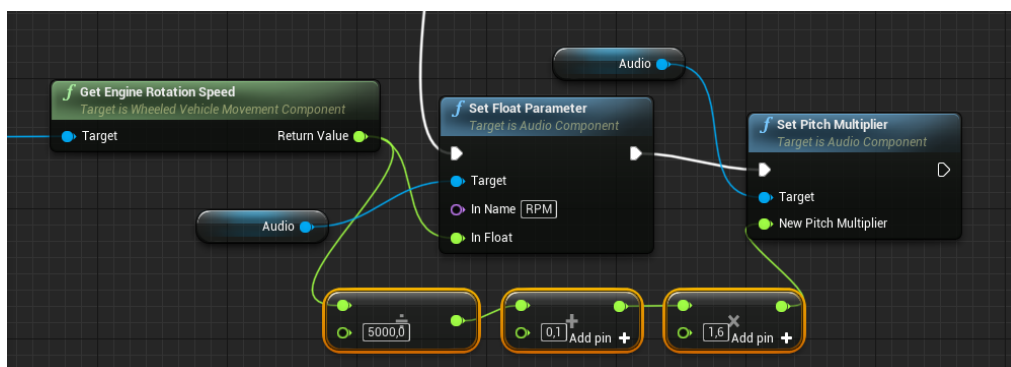


Рис. 13

Теперь нам необходимо реализовать работоспособность фар. Для этого в боковом меню выберем раздел Light и добавим в сцену spot light для каждой фары. Разместим эти источники света в центре фар. Выбрав в списке объектов наши

источники света, можем настроить силу, дальность свечения и цвет светового излучения. Для того чтобы у нас была возможность включать свет по нажатию кнопки «L» перейдем в наш blueprint и создадим ноды. Нам необходимо сделать так, что при нажатии значение интенсивности света становилось максимальных, а при повторном нажатии становилось равным нулю, а затем значение селектора запоминалось. Добавим ноду, отвечающую за нашу кнопку, а также ноды, которые отвечают за интенсивность. Соединим следующим образом:

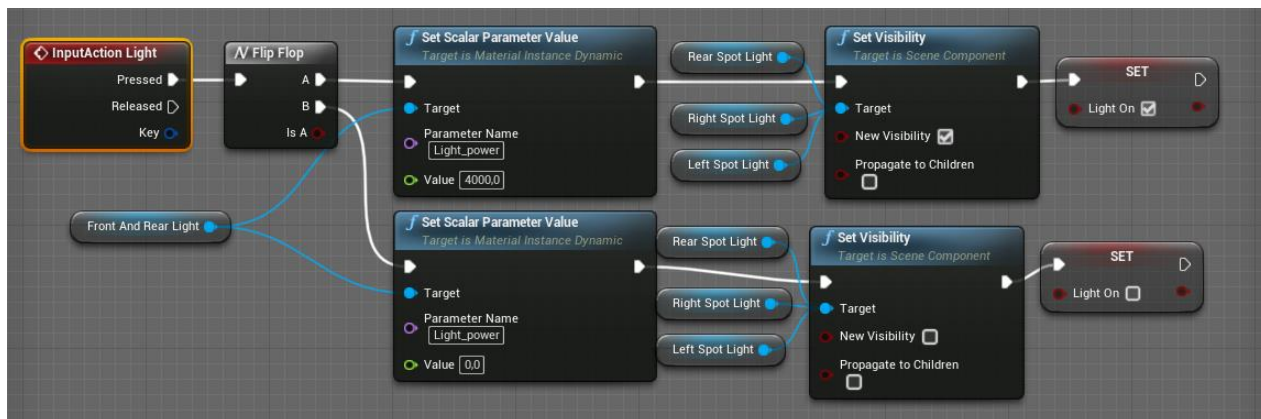


Рис. 14

Для того чтобы у нас была возможность комфортно управлять нашей машиной, нам нужно добавить камеру, с которой мы будем наблюдать за ней. Переходим в наш blueprint, далее add component и добавляем spring arm(рука), затем, выделив spring arm добавляем уже элемент camera. Далее используя правое меню настраиваем длину нашей руки, а также передвигаем ее, занимая положение немного позади и сверху от машины.

Все, наша машина готова. Мы можем нажать кнопку «play» и протестировать ее в движении. Также, зайдя в blueprint и выбрав из списка объектов машину, можно настроить ее характеристики, такие как масса, мощность двигателя, силу трения колес, обороты двигателя, полку крутящего момента и т.д.

Демонстрационный ролик анимированной машины: (см. в приложении А).

ЗАКЛЮЧЕНИЕ

При написании данной курсовой работы были выполнены следующие задачи:

1. Исследован и описан процесс создания 3D моделей, на примере создания автомобиля посредством использования студенческих версий таких программ, как: 3D Studio Max, Marmoset и Substance Painter.
2. Исследован и описан процесс анимации автомобиля посредством использования бесплатного 3D движка Unreal Engine 4.

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Меженин А.В. Технологии разработки 3D – моделей. Учебное пособие . – СПб: Университет ИТМО, 2018. – 100 с. – URL: <https://books.ifmo.ru/file/pdf/2287.pdf> (дата обращения 15.05.2021). Режим доступа: свободный.
2. Демяненко Я.М. Компьютерная графика. Рельефное текстурирование. Учебное пособие. – Южный федеральный университет, Институт математики, механики и компьютерных наук, 2019 –80 с. – URL: http://edu.mmcs.sfedu.ru/pluginfile.php/51313/mod_resource/content/1/OpenGL_2_2.pdf (Дата обращения 14.05.2021) режим доступа: свободный.
3. Меженин А.В. Технологии 3d моделирования для создания образовательных ресурсов. Учебное пособие. – СПб., 2008. - 112 с. – URL: http://window.edu.ru/resource/846/74846/files/3d_modelling.pdf (дата обращения: 17.05.2021). Режим доступа: свободный.
4. Д.А. Булгаков, А.А. Никитина, Н.Н. Решетникова. Основы разработки 3D – сцен в пакете 3Ds Max. Учебное пособие. – СПб: Федеральное государственное автономное учреждение высшего профессионального образования «СПбГУАП», 2012. – 44 с. – URL: https://fs.guap.ru/k44/trud/bulgakov_nikitin_reshetnikova.pdf (дата обращения 20.05.2021). Режим доступа: свободный.
5. Куксон Арам, Райан Даулингсок, Клинтон Крамплер. Разработка игра на Unreal Engine 4 за 24 часа; [перевод с анлийского М.А. Райтмана]. – Москва: Эксмо, 2019. – 528 с. : ил. – (Мировой компьютерный бстселлер. Геймдизайн). – URL: <https://cdn1.ozone.ru/s3/multimedia-6/6007253934.pdf> (дата обращения 19.05.2021) Режим доступа: свободный.
6. Томми Транн. Тьюториал по Unreal Engine. Часть 1: знакомство с движком [Электронный ресурс]. - URL: <https://habr.com/ru/post/344394/> (дата обращения 16.05.2021). Режим доступа: свободный.

7. Текстурирование 3d – моделей: [Электронный ресурс]. – URL:<https://renderart.ru/articles/texturing-materials> (дата обращения 15.05.2021). Режим доступа: свободный.
8. Загарских А.С., Хорошавин А.А., Александров Э.Э., Введение в разработку компьютерных игр– СПб: Университет ИТМО, 2019. – 79 с. – URL: <https://books.ifmo.ru/file/pdf/2646.pdf> (Дата обращения 16.05.2021). Режим доступа: свободный.
9. Из чего состоит 3d модель. Создание 3d моделей. [Электронный ресурс] -URL: <http://cpu3d.com/animation/iz-chego-sostoit-3d-model-sozdanie-3d/> (дата обращения 16.05.2021). Режим доступа: свободный.
- 10.Как работает рендеринг 3D – игр: текстурирование и фильтрация текстур. [Электронный ресурс]. – URL: <https://temofeev.ru/info/articles/kak-rabotaet-rendering-3d-igr-teksturirovanie-i-filtratsiya-tekstur/> (дата обращения 16.05.2021). Режим доступа: свободный.
- 11.Unreal Engine 4. Учебник для начинающих: введение в основы. [Электронный ресурс] – URL: <https://stdpub.com/unrealengine/unreal-engine-4-uchebnik-dlya-nachinayushhih-vvedenie-v-osnovy> (дата обращения 20.05.2021). Режим доступа: свободный.
- 12.<https://klona.ua/blog/3d-modelirovanie/teksturirovanie-modeliruyte-pravilno>
- 13.Все про Bake (Запечку) как этап AAA – пайплайна. [Электронный ресурс] – URL: <https://render.ru/ru/XYZ/post/17128> (дата обращения 20.05.2021). Режим доступа: свободный.
- 14.Текстурирование: моделируйте правильно: [Электронный ресурс]. – URL:https://www.kgasu.ru/upload/iblock/c56/mu_3dstudiomax.pdf (Дата обращения 20.05.2021). Режим доступа: свободный.
- 15.Горелик А.Г. Самоучитель 3ds Max 2014. – СПб.: БВХ. Петербург, 2014. -554 с.: ил. – (Самоучитель). [Учебное пособие]. – URL : <http://bookash.pro/ru/book/188593/samouchitel-3ds-Max-2014-aleksandr-gorelik> (Дата обращения:20.05.2021). Режим доступа: свободный.

- 16.3D – моделирование.Autodesk: [Электронный ресурс]. – URL: <https://www.autodesk.ru/solutions/3d-modeling-software> (Дата обращения: 21.05.2021). Режим доступа: свободный.
- 17.В.С. Хворост. Внедрение пайплайна разработки моделей для игр в обучающий процесс: БГТУ, Минск: [Учебное пособие]. – URL: https://elib.belstu.by/bitstream/123456789/33444/1/Hvorost_Vnedrenie_pajplajna.pdf (Дата обращения: 21.05.2021) Режим доступа: свободный.
- 18.Н.К. Трубочкина. Технология создания полнометражных 2d и 3d фильмов с использованием фрактальных слоев: [Ученое пособие]. – URL: <https://publications.hse.ru/mirror/pubs/share/folder/3o5gcskrng/direct/201784596.pdf> (Дата обращения: 18.05.2021) Режим доступа: свободный.

ПРИЛОЖЕНИЕ А

Демонстрационный ролик движения автомобиле по ссылке:

Animated old truck | Unreal Engine 4. //Видеохостинг YouTube [Электронный ресурс]. – URL: <https://www.youtube.com/watch?v=F2Qak40Eaa4> Режим доступа: свободный.