

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

Вычислительной техники

Кафедра

Вычислительная техника

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

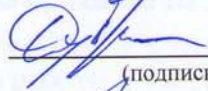
Профиль Вычислительные машины, комплексы, системы и сети

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ
РАБОТА БАКАЛАВРА**

на тему

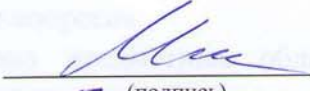
**Приложение на основе устройства виртуальной реальности
Oculus Rift для обучения основам выживания на дикой местности**

Студент  Балаев Кирилл Артемович
(подпись, дата)

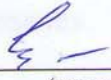
Руководитель  Дубравин А.В.
(подпись, дата)

Нормоконтролёр  18.08.19 Кучин А.В.
(подпись, дата)

Работа допущена к защите (протокол зас. кафедры от 31.05.2019 г. № 38-09-19)

Заведующий кафедрой  Митрохин М.А.
(подпись)

Работа защищена с отметкой отметка (протокол заседания ГЭК от 19.06.19 № 3)

Секретарь ГЭК  Гурин Е.И.
(подпись)

Пенза, 2019

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

Вычислительной техники

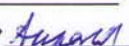
Кафедра

Вычислительная техника

«Утверждаю»

Заведующий кафедрой

 М.А. Митрохин

«21»  2019 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
БАКАЛАВРА**

1. Студент Балаев Кирилл Артемович гр. 15BV1 факультета вычислительной техники.

2. Тема работы: «Приложение на основе устройства виртуальной реальности Oculus Rift для обучения основам выживания на дикой местности».

Тема утверждена приказом ПГУ № 554/о от "29" апреля 2019 г.

3. Руководитель работы к.т.н., доцент каф. ВТ ПГУ Дубравин А.В..

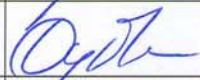
4. Задание на работу (назначение разработки, исходные данные и т.п.).

Разработать приложение на основе устройства виртуальной реальности Oculus Rift. Разработать и отладить взаимодействие модулей приложения. Система должна быть представлена в виде единого приложения с целью обеспечения его применения в качестве обучающего материала.

5. Перечень подлежащих разработке вопросов.

Требуется произвести анализ предметных областей: изучить типы устройств виртуальной реальности, виды и различия программных пакетов для работы с устройствами виртуальной реальности, рассмотреть программное обеспечение для разработки игр. Необходимо изучить методы 3D моделирования.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Изучение предметной области	7	29.04.2019 - 06.05.2019	
2.	Анализ требований	3	07.05.2019 - 10.05.2019	
3.	Выбор технологий, применяемых при проектировании и разработке	3	11.05.2019 - 14.05.2019	
4.	Разработка приложения на основе устройства виртуальной реальности Oculus Rift	10	15.05.2019 - 25.05.2019	
5.	Оформление пояснительной записки	5	26.05.2019 - 31.05.2019	

Дата выдачи задания "29" апреля 2019 г.

Руководитель бакалаврской работы  Дубравин А.В.
(подпись, дата)

Задание к исполнению принял студент  29.04.2019 Балаев К.А.
(подпись, дата)

Работу к защите допустить
Декан факультета  18.6.19 Фионова Л.Р.
(подпись, дата)

РЕФЕРАТ

Пояснительная записка содержит 17 рисунков, 2 таблицы, 30 источников, 2 приложения.

ВИРТУАЛЬНАЯ РЕАЛЬНОСТЬ, OCULUS RIFT, ВЫЖИВАНИЕ, VIRTUAL REALITY, SURVIVAL.

В рамках ВКР было разработано приложение на основе устройства виртуальной реальности Oculus Rift. Разработаны модели дикой местности для обучения. Описаны возможные действия для обучения людей выживанию на дикой местности.

Целью работы являлась разработка приложения на основе устройства виртуальной реальности Oculus Rift для обучения основам выживания на дикой местности.

В процессе работы проводился анализ, сравнение и выбор устройства виртуальной реальности, наиболее подходящей для данного типа задачи, а также выбор программного обеспечения наиболее подходящего для решения данной задачи.

В результате было создано приложение для обучения основам выживания на дикой местности. В дальнейшем планируется расширить спектр знаний предоставляемый приложением.

					ПГУ 2.090301.06.001 ПЗ					
Изм.	Лист	№ докум.	Подпись	Дата	Приложение на основе устройства виртуальной реальности Oculus Rift для обучения основам выживания на дикой местности Пояснительная записка			Лит.	Лист	Листов
Разраб.		Балаев К.А.								
Провер.		Дубравин А.В.							4	
Н. Контр.		Кучин А.В.						ФВТ, каф. ВТ гр. 15ВВ1		
Утверд.		Митрохин М.А.								

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

FPS - Кадровая частота (Frames per Second) — количество сменяемых кадров за единицу времени в компьютерных играх, телевидении и кинематографе.

Моделирование – процесс создания 3D модели.

Модель – 3D объект, который может состоять как из одного объекта, так и из множества, связанных между собой.

ОЗУ – оперативно запоминающее устройство.

ПО - программное обеспечение.

VR – виртуальная реальность

					ПГУ 2.090301.06.001 ПЗ	Лист
						2
Изм.	Лист	№ докум.	Подпись	Дата		

СОДЕРЖАНИЕ

РЕФЕРАТ	1
ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ	2
СОДЕРЖАНИЕ	3
ВВЕДЕНИЕ	4
1. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ	5
1.1. Обзор VR-технологий.....	6
1.1.1. Аппаратное обеспечение (датчики и устройства, шлемы)	9
1.1.2. Программное обеспечение	11
1.2. Программные продукты для разработки игр.....	13
1.3. Программные продукты для 3D моделирования	17
1.4. Методы 3D моделирования	17
2. ПОСТАНОВКА ЗАДАЧИ	20
3. ПРОЕКТИРОВАНИЕ СИСТЕМЫ	21
3.1. Выбор языка программирования и среды разработки.	21
3.2. Выбор языка программирования.....	22
3.3. Выбор платформы разработки	23
4. РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ШЛЕМА ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ	24
4.1. Структура файла скрипта	25
4.2. Описание ключевых классов.....	27
5. ТЕСТИРОВАНИЕ И ОТЛАДКА	29
5.1. Характеристики.....	29
5.2. Тестирование	30
5.3. Выводы	37
ЗАКЛЮЧЕНИЕ	38
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	39
ПРИЛОЖЕНИЕ А. Таблица сравнения средств 3D моделирования	42
ПРИЛОЖЕНИЕ Б. Листинги программы	53
ПРИЛОЖЕНИЕ В Презентация	61

ВВЕДЕНИЕ

В данный момент проблема выживания или обеспечения безопасности, в условиях пребывания человека в экстремальных условиях даже в кратковременном периоде, является проблемой, требующей решения. Обоснованием для необходимости решения задачи я считаю явно просматривающуюся закономерность: чем выше техническое развитие человека, тем он менее приспособлен к борьбе за выживание в дикой местности. При этом не имеется различия в каких условиях это происходит так как человек одинаково плохо справляется с задачей выживания как в условиях крайнего севера, так и в палящих песках пустынь.

Но кроме традиционных методик обучения в недавнем времени появился новый способ. Этот способ представляет собой обучение с использованием VR технологий. В этой работе я предлагаю использование этих технологий для качественного повышения уровня обучения людей выживанию в чрезвычайной ситуации.

Разработка данного программного продукта в первую очередь представляет собой разработку модели реального мира, с которым можно взаимодействовать для более наглядного обучения людей. Поэтому необходимо также рассмотреть, как ПО для разработки данного приложения, а также средства 3д моделирования для создания моделей предметов реального мира.

1. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

Возможность применения шлема виртуальной реальности в обучении были исследованы ранее. В работе[1] Селиванова В.В. были сделаны следующие выводы.

1. Обучающие программы, созданные для Виртуальной реальности, среди прочих положительных эффектов прежде всего стимулируют мышление человека. Эти программы способствует прогрессивному формированию как процессуальных, так и операциональных характеристик мышления, а также развивает формы мыслительной активности. Данное влияние, в конечном итоге, сказывается на более успешном решении обучаемым задач.

2. Работа в программах для обучения в Виртуальной реальности оказывается более продуктивной с точки зрения кратковременной памяти, наблюдательности, устойчивости, концентрации внимания, способности к обобщению и классификации.

3. Работа в программах для Виртуальной реальности формирует специфически познавательную мотивацию, интерес к обучению и создание позитивных, гармоничных психических состояний.

4. Развивающий эффект дидактических программ в Виртуальной реальности определяется объемным изображением познаваемых объектов, широкой возможностью осуществления действий с предметами (анимацией), эффектом присутствия, интерактивностью ситуации, осуществлением визуализации абстрактных моделей и др.

5. Обучающие виртуальные программы хорошо подходят для тренинга профессиональных навыков в различных видах деятельности.

На основе этих выводов я посчитал возможным применения шлема виртуальной реальности в обучении выживанию в дикой местности. Так как те положительные изменения, которые оказывает применение виртуальной

реальности в обучении очень полезны в случае обучения людей выживанию. А наиболее удобным для разработки такого рода приложения оказалось ПО для разработки игр в силу схожести целей для разработки этого приложения и игр жанра Выживание.

1.1. Обзор VR-технологий.

Виртуальная реальность

Виртуальная реальность – это новая концепция использования компьютеров и человеко-машинного интерфейса для создания эффекта трехмерного окружения, в котором пользователь в интерактивном режиме взаимодействует с виртуальными объектами, и при этом создается сильное ощущение трехмерного присутствия[2].

Понятие «виртуальная реальность» приобрело популярность сравнительно недавно. «Виртуальная реальность — это компьютерная система, применяемая для создания искусственного мира, пользователь которой ощущает себя в этом мире, может быть управляем в нем и манипулировать его объектами»[3].

Виртуальная реальность представляет собой некое подобие окружающего нас мира, искусственно созданного с помощью технических средств и представленного в цифровой форме. Создаваемые эффекты проецируются на сознание человека и позволяют испытывать ощущения, максимально приближенные к реальным.

Использование виртуальной реальности в коммуникационной сфере имеет следующий ряд особенностей[2].

1. Наглядная передача информации, сложной для восприятия при использовании традиционных способов обучения (например, моделирование сложных для демонстрации физических опытов, демонстрация стереометрических объектов и т.п.);

2. Способ хранения и демонстрация информации об объемных объектах, имеющих сложную структуру (макеты технических устройств,

моделирующих структуру объекта, которые позволяют ориентироваться нагляднее и проще);

3. Создание виртуальных тренажеров (для операторов техники, врачей и т.п.);

4. Конструирование пространственных объектов, операции с которыми сложны и неудобны с применением традиционного «плоского» интерфейса (компьютерная мышь, клавиатура, планшет и т.п.). Решение данных задач с использованием средств создания виртуальной реальности предполагает различную степень их инженерного совершенства: от стационарных установок, до компактных приставок к персональному компьютеру.

Вышеизложенное относится к виртуальной реальности в ее «классическом» понимании. На сегодняшний день имеются различные интерпретации понятия виртуальной реальности, и соответственно всевозможные ее разновидности.

Примеры виртуальной реальности

Выделяют несколько типов виртуальной реальности, в зависимости от степени погруженности пользователя в создаваемую виртуальную картину. Наиболее простым в исполнении является применение программных и технических средств, создающих 3D-изображение на экране монитора. Пользователь при этом является внешним наблюдателем по отношению к смоделированному трехмерному миру, а также может извне менять заранее predetermined параметры, передвигать объекты на экране, произвольно выбирать ход развития событий и т.д. Существуют средства создания виртуального окружения с эффектом частичной погруженности: специальные шлемы, в которых создается изображение, передаваемое непосредственно на сетчатку пользователя, с использованием дополнительных средств, например, кресел, штурвалов, джойстиков, позволяющих управлять событиями в виртуальном окружении. Комплексы иммерсивной виртуальной реальности, типа CAVE 3D, создают эффект полного присутствия. Такая система состоит из

					ПГУ 2.090301.06.001 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

нескольких экранов, расположенных в форме куба, размером с небольшую комнату, на которые проецируются изображения. Зайдя в такую комнату и надев специальные очки, пользователь не видит ничего кроме окружающей его виртуальной среды, внутри которой он может передвигаться, рассматривать с разных позиций виртуальные объекты и взаимодействовать с ними с помощью управляющих приспособлений.

Данная технология предполагает, что с ее помощью произойдет революция в повседневной жизни человека. В связи с применением технологий виртуальной реальности претерпят изменения практически все сферы бизнеса. Например, туризм (виртуальные путешествия), образование (участие в виртуально воссозданных исторических событиях, наглядные примеры химических и физических процессов и так далее), индустрия развлечений (кинофильмы, где потребитель самостоятельно создает свой уникальный сюжет и является главным героем), коммуникационная сфера (маркетинг, реклама, PR-коммуникации социальных объектов будут осуществляться посредством создания виртуальных пространств) и многие другие.

Для максимально реалистичного погружения в мир виртуальной реальности используются датчики, отслеживающих положение тела, а также различные трекинг-системы, такие как:

1. Системы айтрекинга. Данные системы предназначены для отслеживания движения зрачков глаз и позволяют определить, куда человек смотрит в каждый момент времени.

2. Моушн трекинг. При помощи этих систем отслеживаются любые телодвижения человека и одновременно повторяются в виртуальном мире. Отслеживающую функцию может выполнять как камера, так и специальные устройства, надетые на пользователя.

3. 3D-контроллеры. Чтобы пользователь чувствовал себя комфортно в виртуальной реальности традиционные 2D-контроллеры (мышки, джойстики и др.) заменены на объемные манипуляторы, которые позволяют работать в трехмерном пространстве.

4. Устройства с обратной связью. Данные устройства были разработаны в 90-х годах и предназначались для того, чтобы пользователь мог в буквальном смысле ощутить на себе все происходящее в виртуальном мире. В качестве таких устройств могут использоваться вибрирующие джойстики, установки, вращающиеся кресла и т.д.

На протяжении долгого времени источником 3D-картинки для устройства виртуальной реальности выступал компьютер или пользовательская консоль (например, «PlayStation VR»). В настоящее время в качестве источника 3D-картинки используется смартфон, что позволило в разы снизить стоимость устройства. Это обусловлено следующими факторами:

- современные смартфоны являются высокопроизводительными и способны самостоятельно обрабатывать даже самый «тяжелый» 3D-контент;
- дисплеи смартфонов обладают достаточно высоким разрешением;
- на каждом современном смартфоне уже встроены датчики определения положения в пространстве;[2]

1.1.1. Аппаратное обеспечение (датчики и устройства, шлемы)

В настоящее время среди гаджетов виртуальной реальности самым популярным является шлем или очки виртуальной реальности. Шлемы или очки виртуальной реальности подразделяются два вида. Стационарные, которые необходимо подключать к компьютеру. И мобильные, которые используют смартфоны. Рассмотрим некоторые из устройств подробнее.

Oculus Rift – очки виртуальной реальности, содержащие встроенные датчики и сенсоры, которые обеспечивают реагирование на поведение пользователя и транслируют их поведение в виртуальную реальность. При первичной настройке очков под пользователя обеспечивается максимальная индивидуализация гаджета для его исправной работы: учитывается рост пользователя и его межлинзовое расстояние. Данные очки оснащены механизмами, обеспечивающими минимальное время отклика на действия

пользователя, то есть синхронизация действий пользователя в реальности и виртуальной реальности происходит практически онлайн.[5]

Шлем Sony PlayStation VR – игровой шлем виртуальной реальности, Это шлем виртуальной реальности с OLED-экраном, разрешением видео 1 080 точек и передачей 120 кадров в секунду, углом обзора 360 градусов и 3D-звуком. Шлем работает с приставкой PlayStation 4. В отличие от конкурентов, Sony PlayStation не требует размещения ИК-камер по углам комнаты. Вместо этого встроенная PlayStation Camera отслеживает положение девяти светодиодов в пространстве и корректирует картинку. Отдельно можно приобрести контроллеры Dualshock 4 с PlayStation Camera для интуитивной игры и PS Move для ручного управления. Возможны групповые игры, а также нахождение и коммуникация в виртуальном пространстве нескольких человек.[6]

Samsung Gear VR - одни из лучших очков виртуальной реальности среди мобильных устройств. Девайс оснащен системой трекинга, акселерометром, гироскопом и системой настройки зрения для тех, кто плохо видит. Управляется аппаратной кнопкой и джойстиком. Контент достаточно широкий: игры, видео, короткометражные фильмы, виртуальные музеи и концерты.[7]

HTC Vive VR – данный гаджет представляет собой шлем виртуальной реальности и различные дополняющие и расширяющие его возможности устройства. В комплекте со шлемом идут несколько датчиков, которые необходимо установить в помещении. Данные датчики будут отвечать за контроль движений пользователя. Благодаря этим датчикам, происходит передача реальных движений тела пользователя в виртуальное пространство, что создает впечатление полного присутствия в виртуальном пространстве. Также в систему входят эргономичные контроллеры, считывающие движения тела. Сам шлем помимо проекции картинки имеет множество датчиков и сенсоров, которые реагируют на поведение.[8]

FIBRUM PRO – шлем виртуальной реальности, являющийся разработкой российской компании ООО «Фибрум». Данный шлем работает при помощи смартфона на всех существующих операционных системах. Особенностью

данного шлема является уникальная разработка – универсальные линзы, не требующие дополнительных настроек. При достаточно низкой цене обзор шлема составляет 110 градусов, шлем эргономичен и весьма легок.[9]

Все вышеперечисленные гаджеты в своем функционале имеют возможность интерактивного общения и взаимодействия в дополненном или виртуальном пространстве. Пользователи могут осуществлять непосредственное общение друг с другом с ощущением полного присутствия, могут общаться в чатах, а также поддерживать аудио связь, осуществлять покупки в виртуальных магазинах, посещать виртуальные выставки, концерты, посещать матчи, а также участвовать в бизнес-встречах и переговорах.

При помощи очков виртуальной реальности и других дополнительных гаджетов – специальных датчиков и индикаторов - неограниченное число людей может совместно проводить досуг: совместно просматривать видео, играть как в виртуальные игры, так и в любые настольные игры, участвовать в мероприятиях, не покидая дома, то есть социализироваться, развиваться, получать опыт, коммуницировать, строить отношения и так далее.

1.1.2. Программное обеспечение

Для разработки под шлем виртуальной реальности, помимо самого шлема необходимо использования специализированное ПО. Это ПО ответственно за передачу данных от кода в приложении к аппаратной части, в частно за отслеживание положения шлема и контроллеров в пространстве. Однако стоит сразу отметить что не каждый из представленных ниже программных пакетов подойдет под все шлемы, большинство из них заточены под 1 или несколько конкретных моделей.

1. OpenVR SDK

OpenVR — это программный интерфейс (API) и среда выполнения, позволяющие получить доступ к оборудованию виртуальной реальности от разных поставщиков. При этом OpenVR не требует, чтобы приложения обладали информацией о том, какие устройства они используют [12].

					ПГУ 2.090301.06.001 ПЗ	Лист 11
Изм.	Лист	№ докум.	Подпись	Дата		

Программный пакет с открытым исходным кодом разрабатывается компанией Valve. Поддерживает все шлемы для персональных компьютеров(Oculus Rift, HTC Vive,Windows Mixed Reality). Используется для возможности запуска любых устройств в Steam VR.

Основным плюсом является возможность создать разработчикам единый интерфейс для всех шлемов виртуальной реальности. Также среди плюсов особая система ввода Steam Action, которая позволяет игрока менять раскладку своих контроллеров прямо в игре, а разработчикам для того чтобы это было возможно достаточно объявить действия который могут выполнять при нажатии на кнопку контроллера. И при этом разработчикам нет необходимости настраивать игру под каждый шлем и контроллер игроки могут настроить управление сами, а все аппаратные детали берет на себя SDK. Поддерживает все шлемы для персональных компьютеров(Oculus Rift, HTC Vive,Windows Mixed Reality).

Распространяется свободно.

2. VRTK

VRTK – это созданный любителями программный пакет призванный решать наиболее часто встречающиеся проблемы с разработкой игр в виртуальной реальности также подходит для всех шлемов для персональных компьютеров[13].

Распространяется свободно.

3. PSVR devkit

Программный пакет для разработки под шлем виртуальной реальности PSVR. Доступ к нему можно получить только зарегистрировавшись как разработчик PlayStation . Что требует вноса в размере 2500 долларов. Подходит только для шлема PSVR , который ограничен использование только на платформе PlayStation[6].

4. Oculus SDK

					ПГУ 2.090301.06.001 ПЗ	Лист 12
Изм.	Лист	№ докум.	Подпись	Дата		

Программный пакет от создателей шлема Oculus Rift, Oculus Quest ,Oculus Go. Из преимуществ стоит отметить лучшую аппаратную поддержку вышеуказанных шлемов и поддержку их особых функций. Но к сожалению программный пакет подходит только для шлемов фирмы Oculus[14].

Распространяется свободно.

5. Google VR SDK

Программный пакет от фирмы Google направленный в первую очередь на их продукты Google Cardboard и Google Daydream View. В силу этого у них нет поддержки контроллеров от шлемов HTC Vive и Oculus Rift и больше они подходят для разработки под виртуальную реальность в смартфоне[15].

Распространяется свободно.

1.2. Программные продукты для разработки игр

Программные продукты для разработки игр [16]:

1. Unity ;
2. Torque 2D/3D;
3. CryEngine 3;
4. UDK (Unreal Development Kit) ;

Unity

Unity — это инструмент для разработки двух- и трёхмерных приложений и игр, работающий под операционными системами Windows, Linux и OS X [17].

Unity [18] поддерживает два языка: C# (наиболее используемый) и Javascript. Разработчику необходимо знать один из языков в совершенстве, а другой на среднем уровне, так как некоторые моменты Unity делает только на одном из двух языков.

Объекты в Unity могут быть пустыми, (для объединения несколько объектов в единую группу, т.е. сделать их дочерними (GameObject), содержать

компоненты, с которыми взаимодействуют скрипты, им могут быть названы одним и тем же именем, им могут быть присвоены теги, служащие для того, чтобы скрипт нашел нужный объект).

К объектам в Unity можно присвоить коллайдеры [19]:

1. Box Collider – куб;
2. Sphere Collider – сфера;
3. Character Collider - коллайдер, специально введенный в Unity 5 для персонажей;
4. Mesh Collider - коллайдер, созданный по мешу, то есть повторяющий геометрию объекта;
5. Wheel Collider - коллайдер для колес;
6. Terrain Collider - коллайдер для объекта «terrain»;

Анимировать модели в Unity можно несколькими способами:

1. Создание анимаций в специальных программах, для 3D моделирования;
2. В среде Unity, так как здесь есть компонент для их создания;

Достоинства Unity [17]:

1. Поддержка множества передовых форматов для трехмерных объектов (.FBX, .OBJ);
2. IDE: сочетание редактора сцен (в комплексе общего редактора) с редактором игровых объектов и редакторов скриптов. Дополнительно прилагаются генераторы деревьев и «terrain»;
3. Улучшенные возможности скриптинга (в Unity доступны три языка: JavaScript, C# и разновидность Python's Boo);
4. Кроссплатформенность – как уже упоминалось выше, поддерживаются Windows, MacOS, Wii, iPhone, iPod, iPad, Android, PS3 и XBox 360, не все из которых, доступны в бесплатной версии;

5. Современный уровень графики, способный конкурировать с иными движками. Unity обладает такими возможностями, как deferred освещение, стандартный набор постпроцессинговых эффектов, SSAO, ускоренная проработка лайтмапов (специальные текстуры);
6. Проработанный физический движок;
7. Масштабируемость и производительность. Большую часть простых процессов движок обрабатывает на превосходном уровне;
8. Запуск любого приложения на Unity в веб-плагине;
9. Невысокая цена за полную лицензионную версию для крупного веб-разработчика;

Недостатки Unity [20]:

1. Закрытость кода. Невозможность получения исходных кодов движка даже у платной версии;
2. Невозможность дополнения физики движка сторонними возможностями. Нельзя добавить в движок стороннюю физику;

Torque 2D/3D

Достоинства Torque 2D/3D:

1. Бесплатность;
2. Открытый исходный код;
3. Для воспроизведения звуков T2D использует библиотеку –penAL;
4. Симуляция физики осуществляется посредством движка B-x2D ставшего стандартом в двумерных физических исчислениях;

Недостатки Torque 2D/3D:

1. Доступны не все платформы;

CryEngine 3

Достоинства CryEngine 3:

1. Кроссплатформенность;
2. Поддержка ММО игр;
3. Множество технологий визуализации;
4. Встроенная система анимации;
 - 4.1. индивидуализация персонажей;
 - 4.2. параметрическая скелетная анимация;
 - 4.3. процедурное деформирование движения;
5. Встроенная система ИИ, которая позволяет обрабатывать поведение не только персонажей, но и транспортных средств:
 - 5.1. умные объекты;
 - 5.2. алгоритмы динамического обнаружения пути;
 - 5.3. система, управляемая сценариями;
6. Присутствует поддержка глобальных мультиплеерных игр.

Недостатки CryEngine 3:

1. Бесплатная версия предназначена только для некоммерческого использования;
2. Лицензировать движок может только коммерческое лицо;
3. Закрытый исходный код;

UDK

Достоинства UDK:

1. Бесплатность;
2. Гибкая система анимации позволяет контролировать каждую деталь анимируемого объекта;
3. Анимационная модель контролируется системой AnimTree, которая включает следующие механизмы:
 - 3.1. контроллер смешения (Blend);

- 3.2. контроллер, управляемый данными;
- 3.3. физические, процедурно-скелетные контроллеры;
4. Высоко оптимизированная сетевая архитектура;

Недостатки UDK:

1. Закрытый исходный код;
2. Не кроссплатформенный;
3. Система древовидных классов - дочерние и родительские;

1.3. Программные продукты для 3D моделирования

В приложении А1 представлены критерии основных продуктов для 3D моделирования:

1. 3D Studio Max;
2. Maya;
3. Blender;

Все преимущества и недостатки указанные в приложении А1, имеют одинаковый вес и для каждого пользователя имеют разную ценность. Для одного пользователя недостаток продукта является преимуществом в другом продукте и наоборот.

Были использованы материалы лекций [21] на конференции «CG EVENT» от студии 3DMASTER, статьи пользователей и личный опыт работы.

В качестве среды для разработки 3D моделей был выбран программный продукт 3D Studio Max. Выбор основан на личном опыте работы с данным программным продуктом и перевесе его преимуществ перед другими вариантами. Цена не являлась существенным недостатком, так как для студентов предоставляется бесплатная образовательная версия программы.

1.4. Методы 3D моделирования

1. Сплайновое [22] моделирование (моделирование на основе кривых Безье);

2. NURBS моделирование (поверхностное моделирование на основе кривых NURBS);

3. Полигональное [23] моделирование;

Полигональное моделирование

Практически любой объект в игровом пространстве это набор точек (минимум трех), не лежащих на одной прямой и их образы однозначно задают аффинное преобразование плоскости.

Матричное представление используется, в частности, для записи аффинных преобразований в компьютерной графике. Матричное представление используется в OpenGL и DirectX (где координаты представляются в виде матриц 1×4), где она транспонирована.

Другими словами, каждый полигон состоит из набора точек и приходится хранить информацию о координатах каждой точки. Пример 3D модели и ее представление в виде точек показано на рисунке 1.6.1 И чем больше будет точек (следовательно, и полигонов), тем ниже будет производительность.

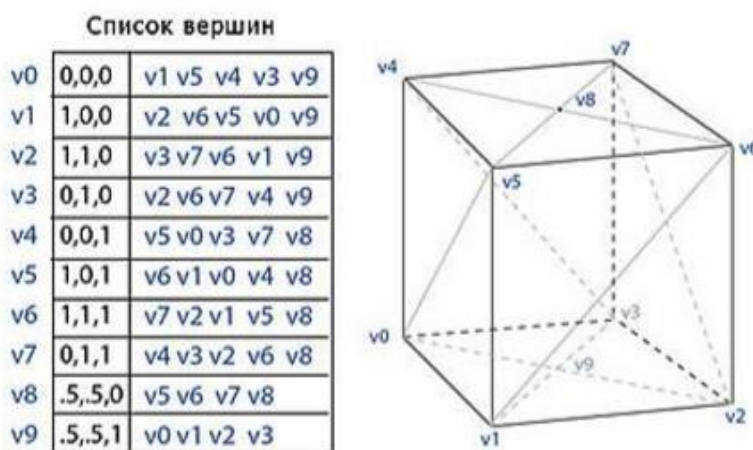


Рисунок 1.1 Полигональная модель

Преимущества полигонального моделирования:

1. Является универсальным;
2. Полнота метода и инструментария;
3. Интеграция с другими пакетами и технологиями (OBJ, FBX);

4. Обеспечивает любую требуемую степень точности;
5. Наиболее полно реализовано во всех современных пакетах;
6. Занимает меньше времени на освоение и получение качественного результата;
7. Простота работы, что снижает время на построение сложного объекта;
8. В отличие от других методов позволяет вносить более детальные изменения в объекте;
9. Естественный переход от низкополигональной к высокополигональной модели [24];
10. Переносимость техник моделирования между пакетами;
11. Перспективы развития;
12. Наличие инструментов скульптинга;
13. Наличие инструментов ретопологии;

Таблица 1 - Сравнение высокополигонального и низкополигонального моделирования

Низкополигональное моделирование	Высокополигональное моделирование
<ol style="list-style-type: none"> 1. Полигоны плоские. 2. Есть сгиб — нужно ребро. 3. 3 или 4 угла в полигоне. 4. Общий объем не нужен. 5. Упрощение объекта. 	<ol style="list-style-type: none"> 1. Полигоны выпуклые. 2. 4-х угольники всегда. 3. 3-угольники на крайний случай. 4. 5-угольники иногда. 5. Есть особые полосы полигонов. 6. Избегание «звездочек» в полигонах. 7. Предъявляет более высокие требования к модели.

2. ПОСТАНОВКА ЗАДАЧИ

Необходимо разработать программу "Приложение для обучения людей основам выживания в дикой местности".

Возможности программы должны предоставлять взаимодействие с объектами в VR, разведение костра, кипячение воды, передвижение по пространству. Для решения этих задач необходимо разработать и реализовать простой, удобный и понятный пользовательский интерфейс.

Программа обязательно должна быть многомодульной: необходимо поделить ее на логические модули. Это во многом упростит добавление в программу новых функций при дальнейшей разработке, а также упростит обнаружение и исправление ошибок при ее отладке и тестировании.

Программа должна быть разработана для работы в операционной системе Microsoft Windows.

					ПГУ 2.090301.06.001 ПЗ	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

3. ПРОЕКТИРОВАНИЕ СИСТЕМЫ

3.1. Выбор языка программирования и среды разработки.

Для разработки приложения был выбран игровой движок Unity по превосходству в количестве преимуществ и менее значимых недостатках, и в силу моего обширного опыта работы с этим движком.

Для Unity существует две наиболее популярных IDE.

MonoDevelop

MonoDevelop - это интегрированная среда разработки (IDE), поставляемая вместе с Unity. IDE сочетает в себе функции текстового редактора с дополнительными возможностями для отладки и выполнения других задач по управлению проектами. Данная среда разработки, предназначенная для создания приложений C#, Java, Boo, Nemerle, Visual Basic .NET, Vala, CIL, C и C++. MonoDevelop является частью проекта Mono. Встроен в дистрибутив Unity3D как средство написания скриптов[25] .

Возможности:

1. Подсветка синтаксиса;
2. Сворачивание кода;
3. Автодополнение кода;
4. Браузер классов;
5. Поддержка плагинов;
6. Встроенный отладчик;
7. Визуальный конструктор форм (GTK#);
8. Модульное тестирование;

Поддерживаемые языки программирования:

1. C#;
2. Visual Basic.Net;
3. C;

4. C++ (до версии 4.0 в Windows);
5. Vala;

Microsoft Visual Studio

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данный продукт позволяет разрабатывать как консольные приложения, так и приложения с графическим интерфейсом [26].

Microsoft Visual Studio позволяет разрабатывать программы с использованием следующих языков программирования: C++, C#, Visual Basic, F#, JavaScript, TypeScript, Python.

Продукты Microsoft для разработчиков давно входят в список наиболее востребованного программного обеспечения для программистов разного уровня. За восемь лет существования на рынке среда разработки Microsoft Visual Studio стала де-факто стандартом создания .NET приложений.

3.2. Выбор языка программирования

Unity изначально поддерживает два языка программирования [27]:

1. C# стандартный в отрасли язык подобный Java или C++;
2. UnityScript, язык, разработанный специально для использования в Unity по образцу JavaScript;

Был выбран язык C#, поскольку он обладает некоторыми преимуществ перед языком JavaScript:

1. В Unity не чистый JavaScript;
2. Обучающей литературы по C# больше;
3. Поддержка и сопровождение C# будет в первую очередь, как заявляют создатели движка Unity;

4. С# находится в пятерке самых востребованных языков программирования по версии GitHub [28] (на 2016 год), что облегчает программирование в команде;

5. Работа программы более стабильная (при смене версий движка Unity);

3.3. Выбор платформы разработки

Исходя из опыта разработки приложений для использования в виртуальной реальности, а также в соотношении цена/качество оптимальным вариантом для разработки был выбран шлем Oculus rift.

Так как шлем является стационарным разрабатываемое приложение возможно сделать более реалистичным как для зрительного обзора, так и в контексте моделирования физики, так как стационарные компьютеры обладают большей мощностью чем смартфоны.

Выбранный шлем выделяется среди конкурентов еще тем что имеет контроллеры движений и систему отслеживания положения в пространстве что в совокупности позволит пользователю в процессе использования приложения передвигаться в приложении естественным образом и естественным же образом использовать свои руки в процессе обучения.

4. РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ШЛЕМА ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

Приложение для обучения людей выживанию состоит из следующих компонентов:

1. 3D модели (импортируется в среду Unity)

1.1. объекты

1.2. управляющие элементы

1.3. текстуры

1.4. UV-карты

2. Составляющие системы

2.1. скрипты

2.1.1. контроллеры

2.1.2. коллайдеры

На рисунке 4.1 показана диаграмма классов. Диаграмма классов — диаграмма, демонстрирующая классы системы, их атрибуты, методы и взаимосвязи между ними. Прежде чем управлять моделью, необходимо импортировать ее в среду Unity. Сама модель была сохранена в одном из самых популярных форматов - .OBJ. Импортировать модель можно двумя способами:

1. Разместить файлы в папке проекта (например, «Assets»).

2. В среде разработки Unity выбрать меню «Assets > Import New Asset».

4.1. Структура файла скрипта

После двойного щелчка на скрипте в Unity, он будет открыт в текстовом редакторе [30]. По умолчанию Unity будет использовать MonoDevelop, но можно выбрать любой редактор из панели External Tools в настройках Unity. Простейшая структура файла скрипта:

Скрипт взаимодействует с внутренними механизмами Unity за счет создания класса, наследованного от встроенного класса, называемого MonoBehaviour. Можно сказать, что это класс для создания нового типа компонента, который может быть прикреплен к игровому объекту. Каждый раз, когда скриптовый компонент присоединяется к игровому объекту, создается новый экземпляр объекта, определенный планом. Имя класса и имя файла должны быть одинаковыми, для того, чтобы скриптовый компонент мог быть присоединен к игровому объекту.

Функция Update - это место для размещения кода, который будет обрабатывать обновление кадра для игрового объекта. Это может быть движение, срабатывание действий и ответная реакция на ввод пользователя, в основном всё, что должно быть обработано с течением времени во игровом процессе. Чтобы позволить функции Update выполнять свою работу, часто бывает полезно инициализировать переменные, считать свойства и осуществить связь с другими игровыми объектами до того, как будут совершены какие-либо действия.

Функция FixedUpdate- исполняет ту же роль что и функция Update, с одним отличием. Функция Update вызывается за 1 секунду столько раз сколько проходит FPS, но в результате при размещении в ней кода отвечающего за взаимодействие с миром пользователя с разными персональными компьютерами могут получить разные результаты в силу того что у одного из них действия будут происходить быстрее чем у другого. А функция FixedUpdate вызовется за 1 секунду четко определенно количество раз, сколько раз выбирается в настройках проекта, по умолчанию 15 раз в секунду. Следовательно, при

обработке действий пользователя в этой функции все пользователи получают одинаковый опыт независимый от технических мощностей их машин.

Функция Start будет вызвана Unity до начала игрового процесса (т.е. до первого вызова функции Update), и это идеальное место для выполнения инициализации. Управление игровым объектом Скрипт определяет только план компонента и, таким образом, никакой его код не будет активирован до тех пор, пока экземпляр скрипта не будет присоединен к игровому объекту. Скрипт можно прикрепить путем перетаскивания ассета скрипта, на игровой объект в панели «Hierarchy» или через окно «Inspector» выбранного игрового объекта. Обращение к компонентам В редакторе Unity свойства компонента можно изменить, используя окно Inspector. Так, например, изменения позиции компонента Transform приведет к изменению позиции игрового объекта. По большей части скрипты также изменяют свойства компонентов для управления игровыми объектами. Разница, в том, что скрипт может изменять значение свойства постепенно со временем или по получению ввода от пользователя. За счет изменения, создания и уничтожения объектов в заданное время может быть реализован любой игровой процесс. Наиболее простым и распространенным является случай, когда скрипту необходимо обратиться к другим компонентам, присоединенных к тому же игровому объекту (GameObject) [29]. Это делается с помощью функции «GetComponent». Объект компонента сохраняют в переменную, это делается в C# посредством следующего синтаксиса:

Если есть ссылка на экземпляр компонента, то можно устанавливать значения его свойств, тех же, которые можно изменить в окне Inspector.

4.2. Описание ключевых классов

Класс Campfire.cs отвечает за создание визуального эффекта огня при поджигании костра. Это происходит только при вызове функции Ignite из вне этого класса объектом с скриптом Lighter.cs.

PlayerMovement.cs отвечает за передвижение персонажа, и за возможность его взаимодействия с миром. Его метод FixedUpdate вызывается каждый FPS, для проверки необходимости передвижения персонажа. А также проверяет активна ли кнопка а на левом контроллере ответственная за взаимодействие с объектами мира.

Lighter.cs – отвечает за создание визуального эффекта пламени на зажигалке и поджигание сгораемых объектов. При включении клавишей а на левом контроллере ,если объект со скриптом Lighter был в руке персонажа то будет показана пламя и этим пламенеем возможно поджигание объектов с тегом “Ignitable”. Проверяется каждый FPS.

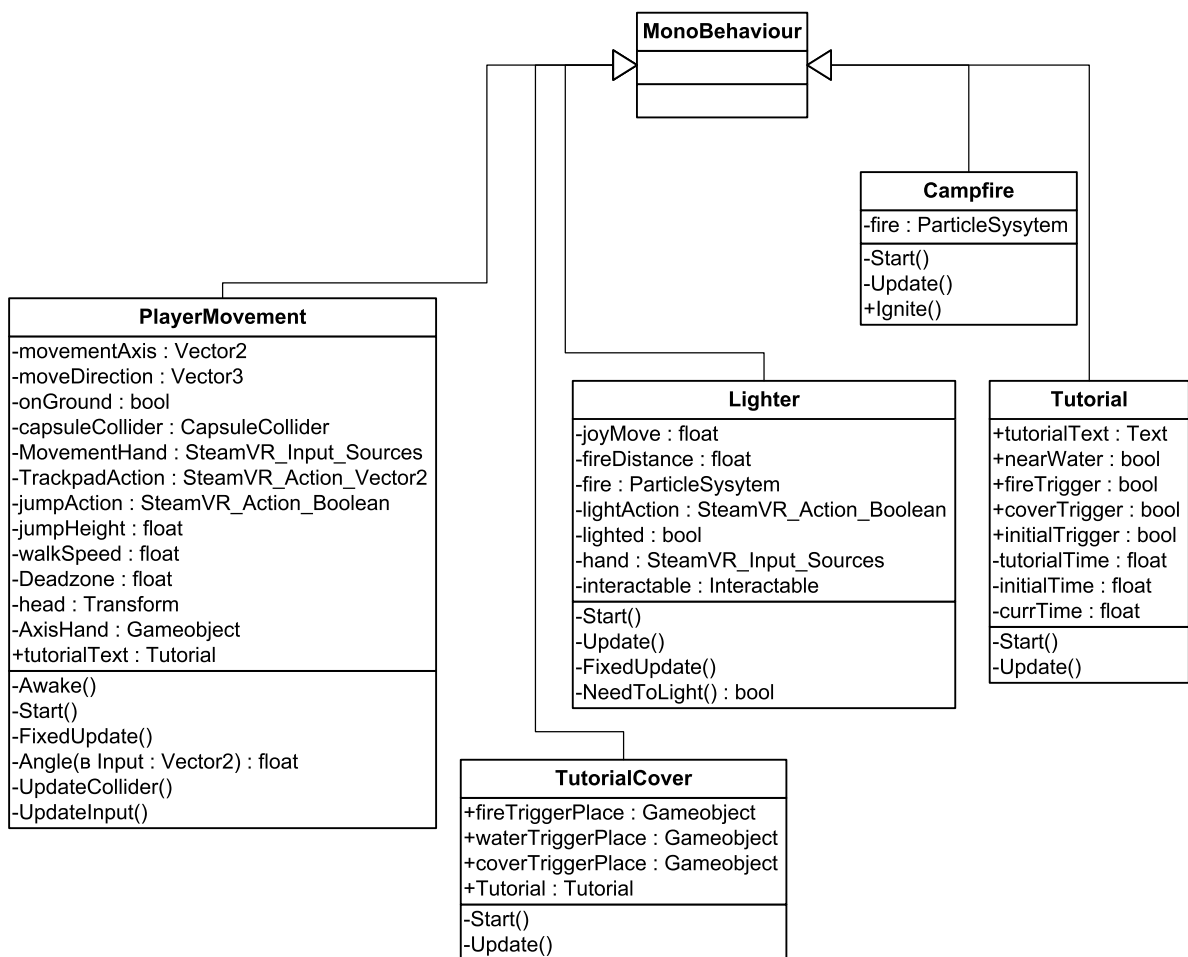


Рисунок 4.1 Диаграмма классов

5. ТЕСТИРОВАНИЕ И ОТЛАДКА

5.1. Характеристики

Тестирование проводилось с использованием шлема виртуальной реальности Oculus Rift CV1. Это шлем с двумя встроенными жидкокристаллическими дисплеями, изображение с которых через линзы направляется на глаза.

Характеристики шлема версии CV1

- Разрешение: 1080×1200 пикселей на каждый глаз
- Частота обновления: 90 Гц, 75 Гц, 72 Гц, 60 Гц
- Угол обзора номинальный: 110°
- Время отклика: 3 мс
- Частота обновления встроенного трекера: 1000 Гц (1000 раз в секунду)
- Датчики: гироскоп, акселерометр, магнитометр, инфракрасные датчики
- Частота обновления трекера, встроенного в камеру: 60 Гц
- Вес: 450 г без кабеля

Характеристики контроллера Oculus Touch

- Соединение по радиоканалу 2,404-2,478 GHz, 2,89 mW одностороннее
- Вибромотор - есть
- Управление - кнопки, джойстик, курок

Шлем представляет собой модель бюджетного уровня. Из недостатков стоит отнести что более современные модели имеют дисплеи с большим

разрешением, более частоту обновления и больший угол обзора, в совокупности это позволяет глубже погрузиться в процесс обучения.

Для тестирования был использован персональный компьютер со следующими характеристиками.

- Процессор Intel Core i5-3570 3.4 GHz
- ОЗУ 8 Гб
- Видеоадаптер Nvidia GeForce GTX 1060 с 6Гб видеопамяти.
- ОС – Windows 10, 64 разрядная.

5.2. Тестирование

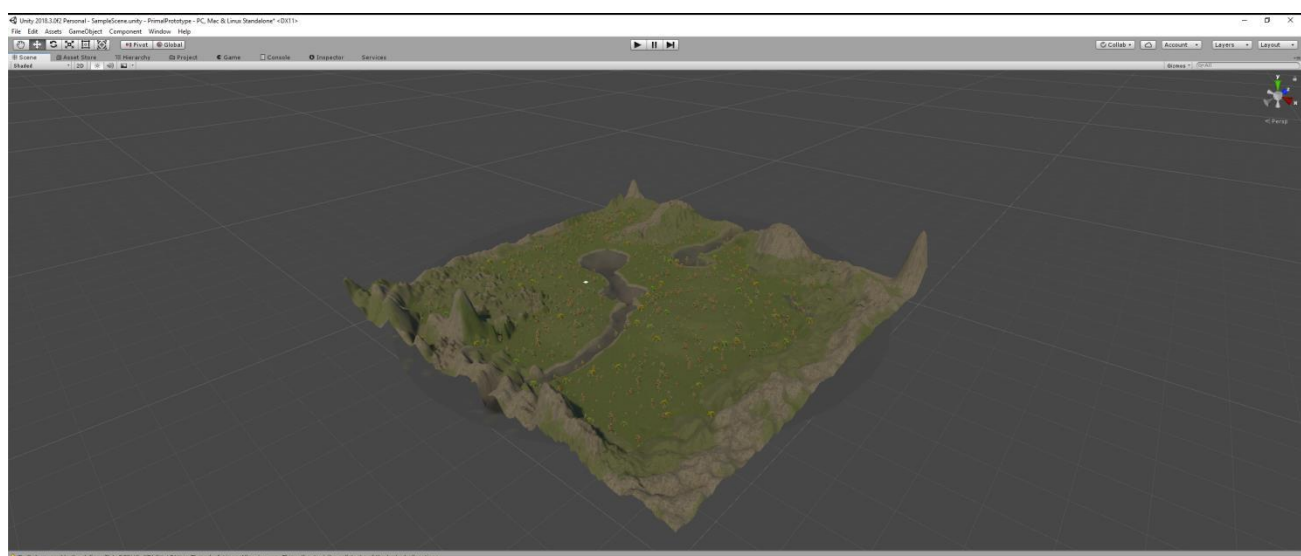


Рисунок 5.1 Общий вид местности

На данном участке есть источник воды растительность и пересеченная местность.

Данный ландшафт позволяет обучить людей основам поиска воды, важности и способов её очистки. Поиску пищи в виде фруктов, ягод, грибов. Способам передвижения по пересеченной местности.

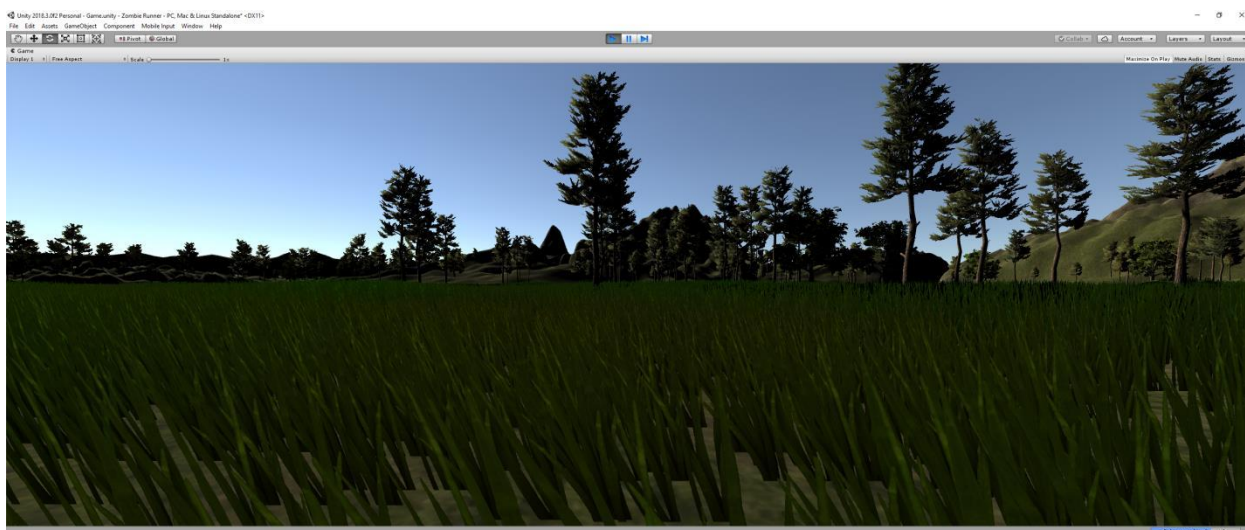


Рисунок 5.2 Общий вид местности

Взаимодействие с объектом производится посредством работы с виртуальными руками пользователя. Помимо самих рук пользователь также видит используемые им контроллеры.

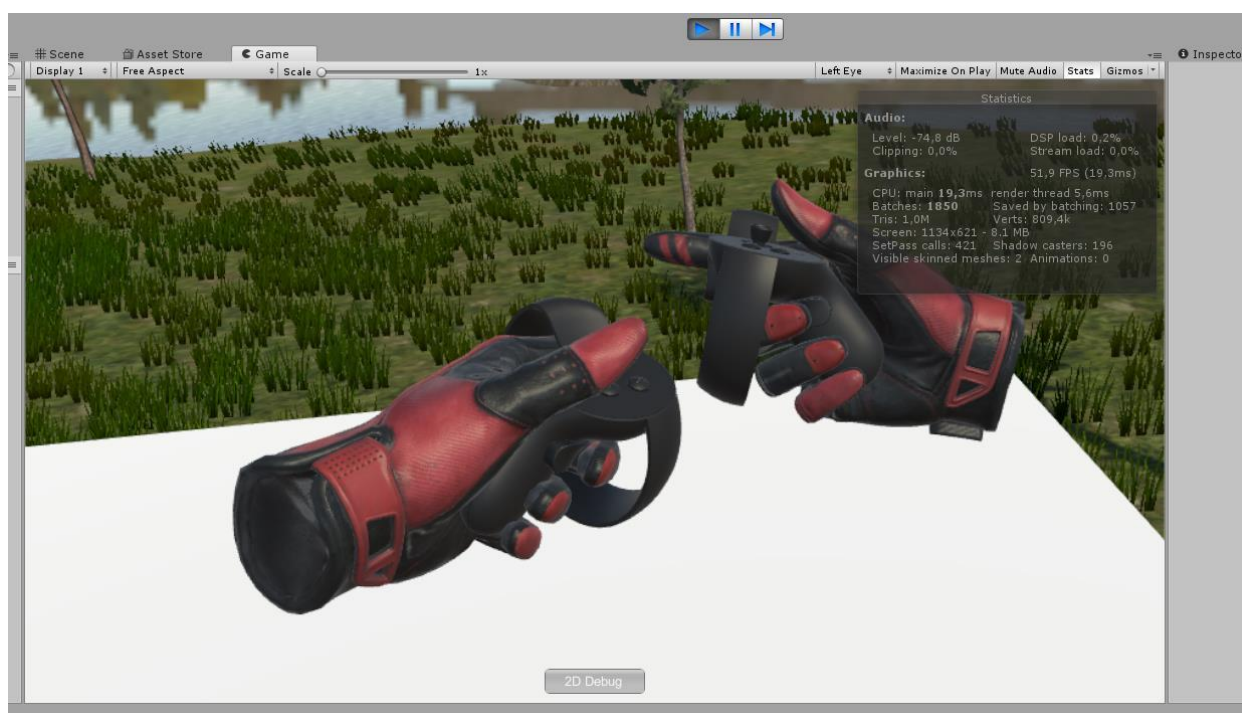


Рисунок 5.3 Виртуальные руки пользователя

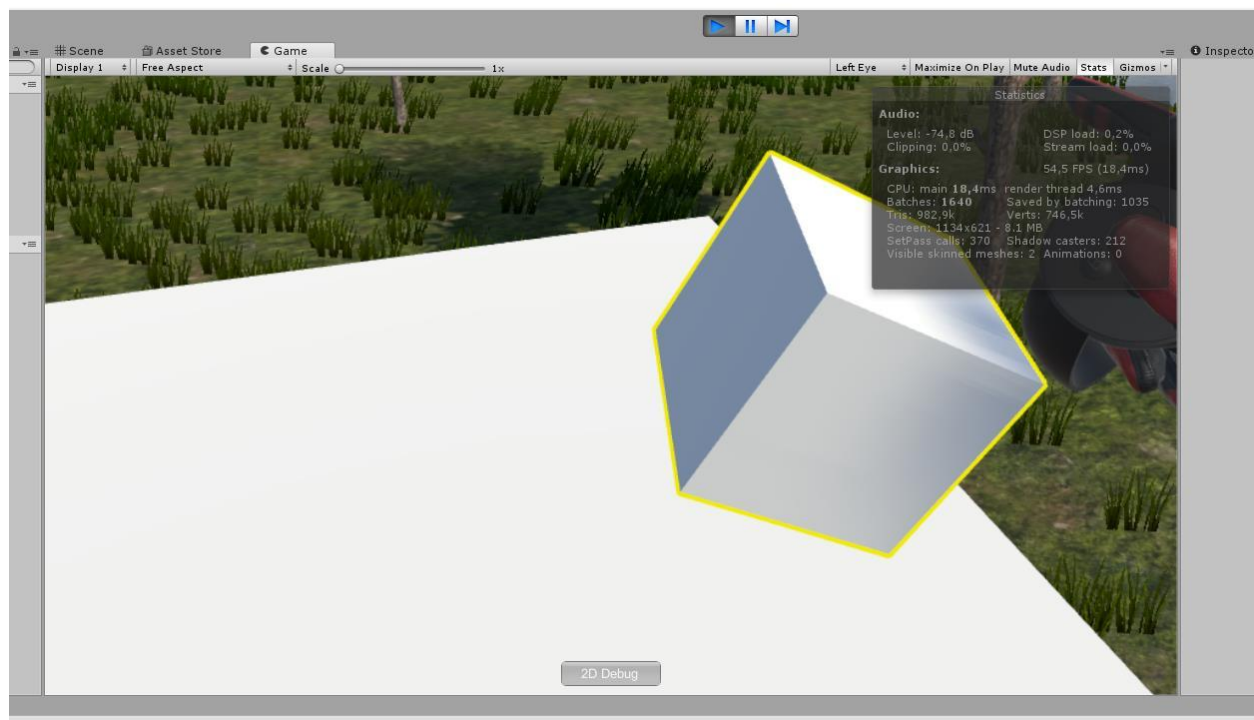


Рисунок 5.4 Рука возле объекта

Если рука пользователя находится достаточно близко к объекту. Для удобства обучения объект подсвечивается по контуру.

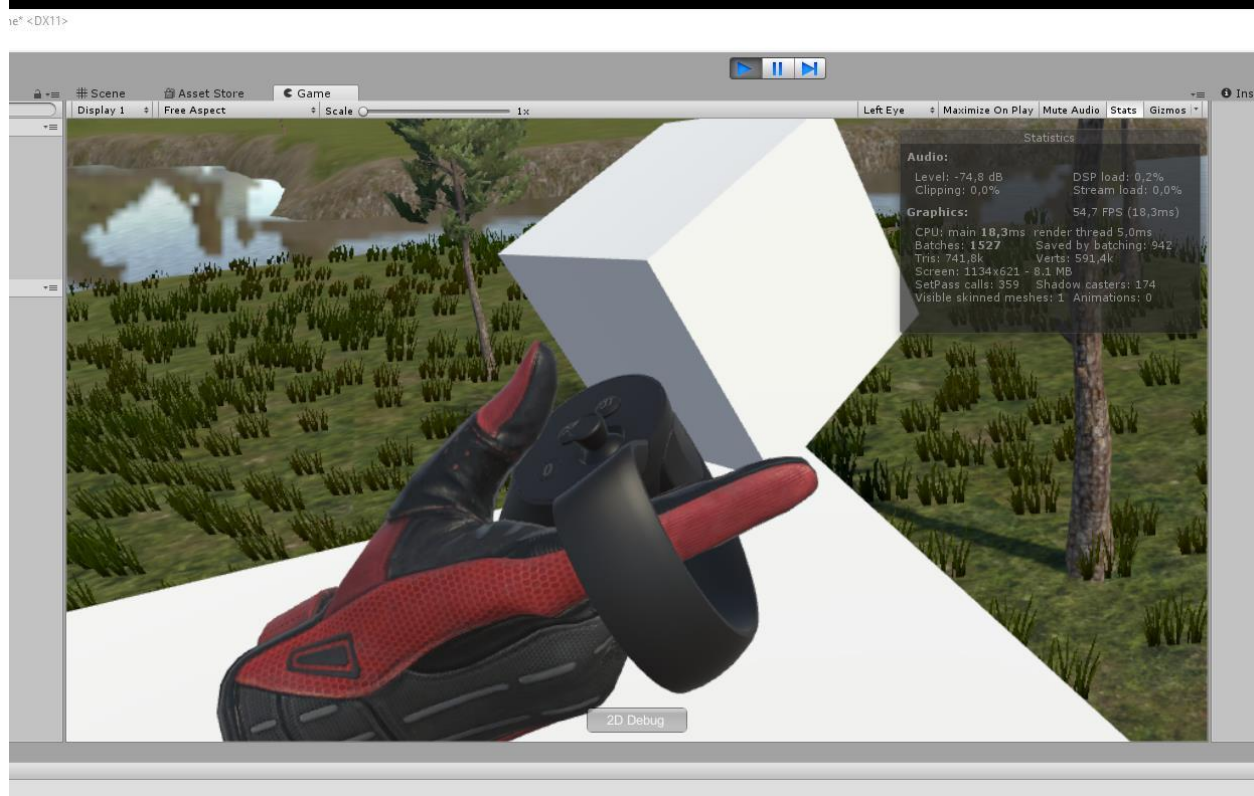


Рисунок 5.5 Объект в руке

Объект схвачен рукой пользователя.

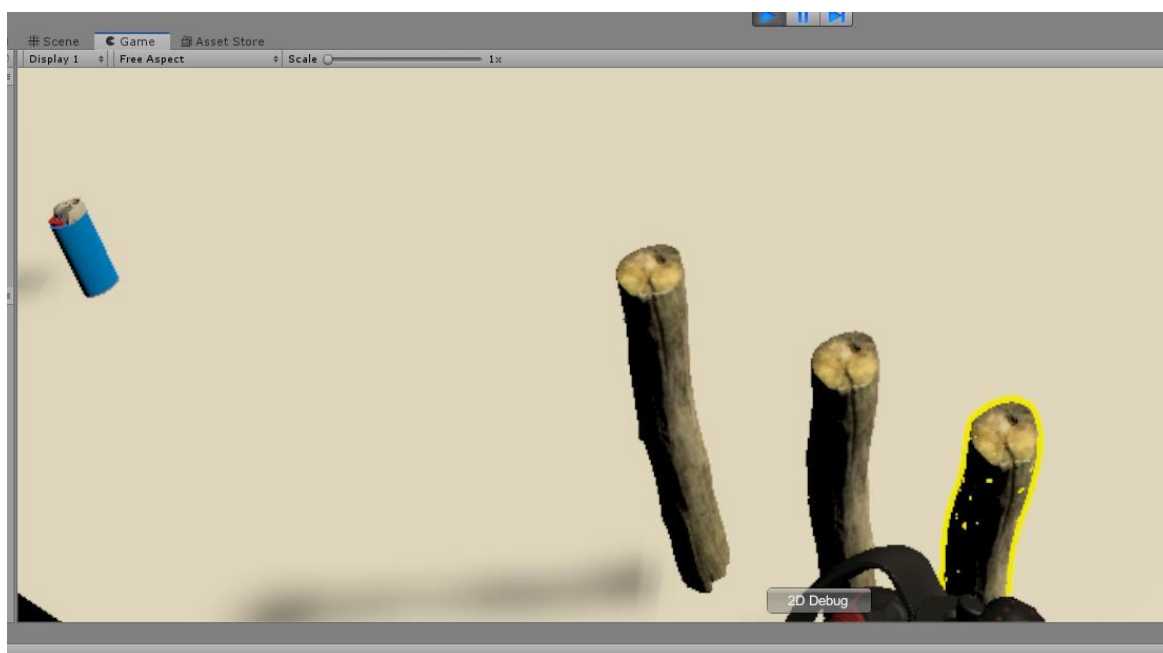


Рисунок 5.6 Объект возле руки

Пример подсветки объекта ветка при поднесении контроллера к объекту.

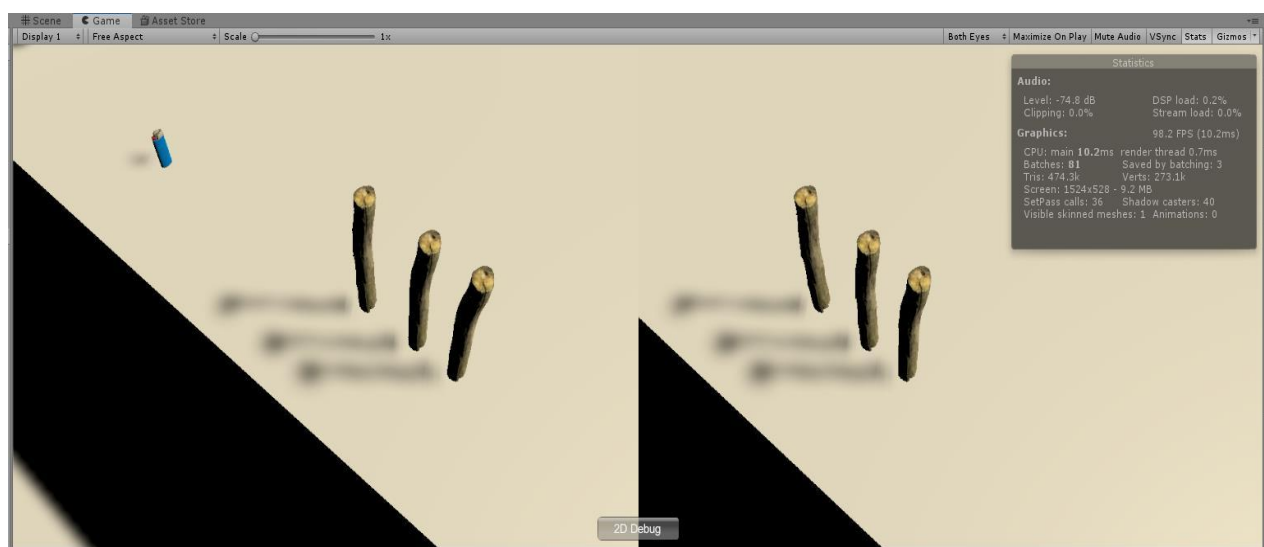


Рисунок 5.7 Объекты

Вид на объекты ветки из глаз пользователя каждый из его глаз имеет свой экран на который выводится изображение.

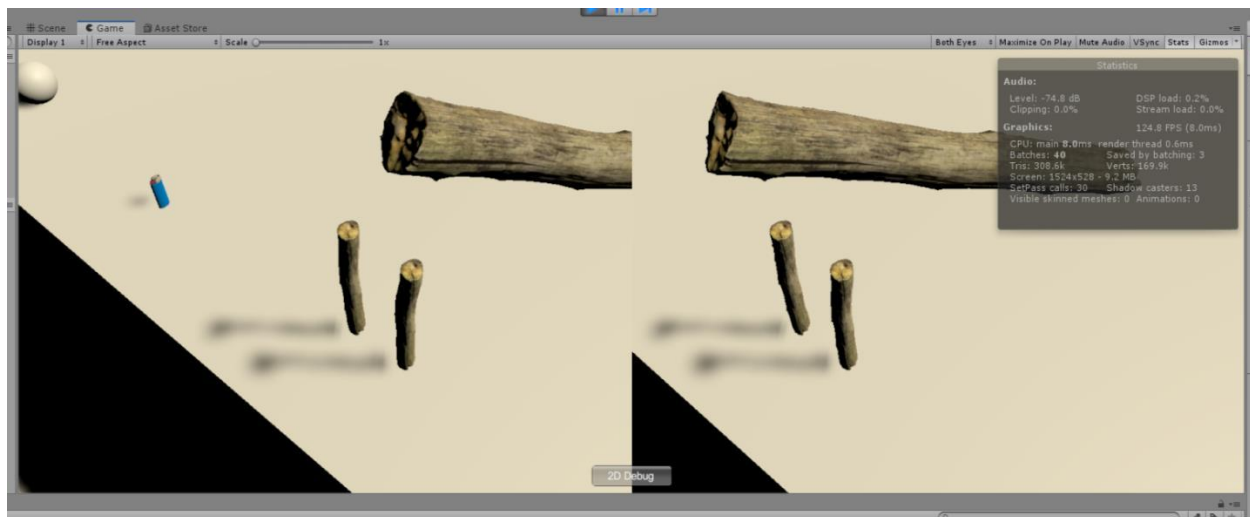


Рисунок 5.8 Объект ветка в руке пользователя

Объект ветка в руке пользователя, поддерживается физика объектов поэтому возможно взаимодействие объектом в руке с остальным миром.

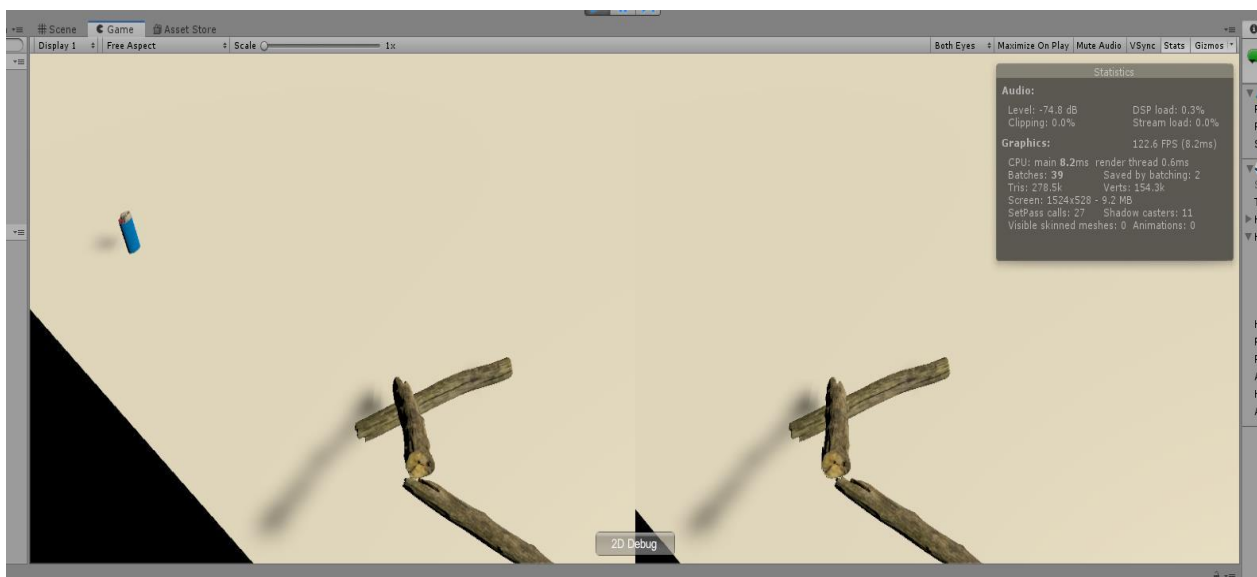


Рисунок 5.9 Взаимодействие объектов

Пример взаимодействия объектов веткой в руке пользователь сбил и подвигал другие ветки.

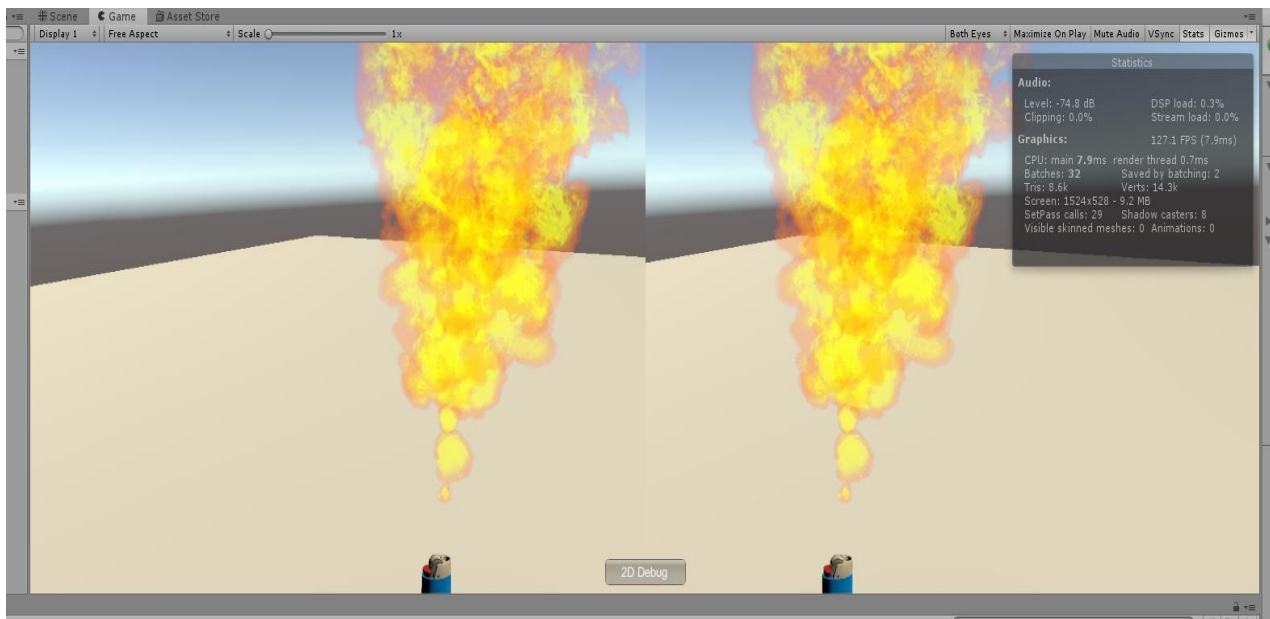


Рисунок 5.10 Активированная зажигалка

Результат применения кнопки а на левом контроллере когда в руке находилась зажигалка.

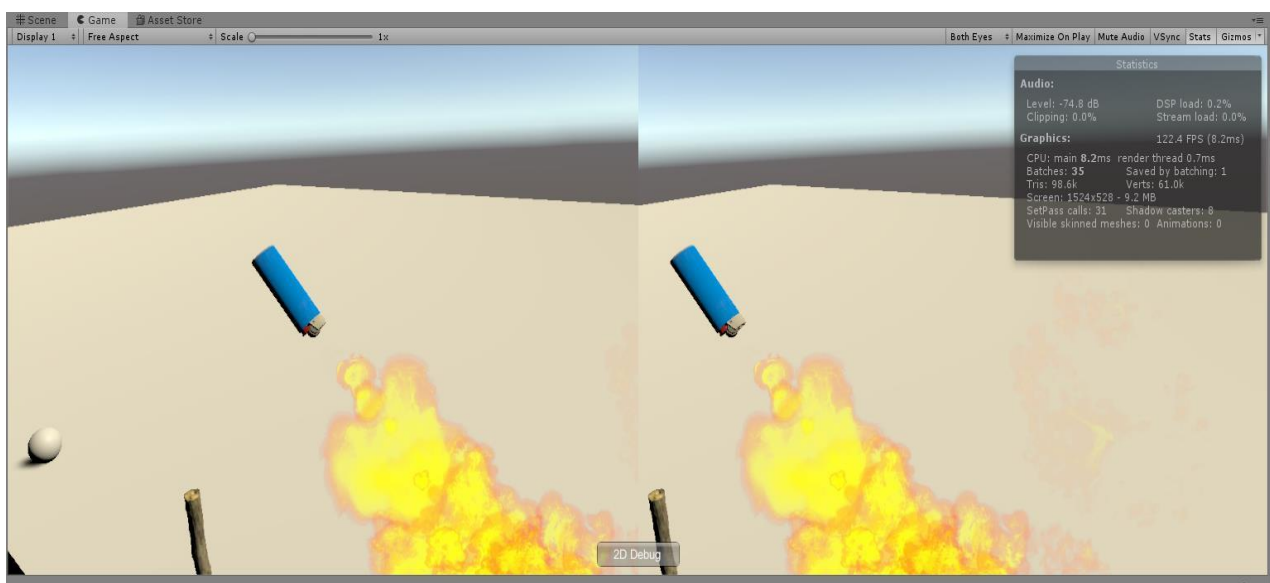


Рисунок 5.11 Поворот пламени

В скрипте Lighter каждый FPS обсчитывается угол наклона объекта зажигалка и соответственно её верху поворачивается генерация огня.

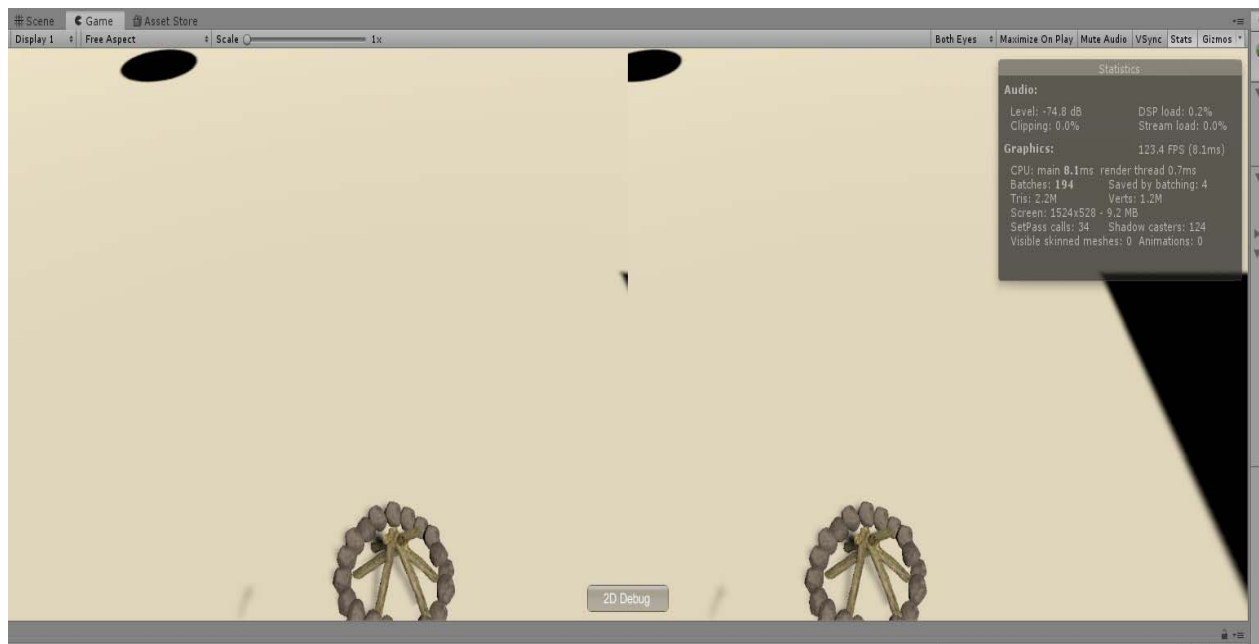


Рисунок 5.12 Объект костер

Объект костер снизу.

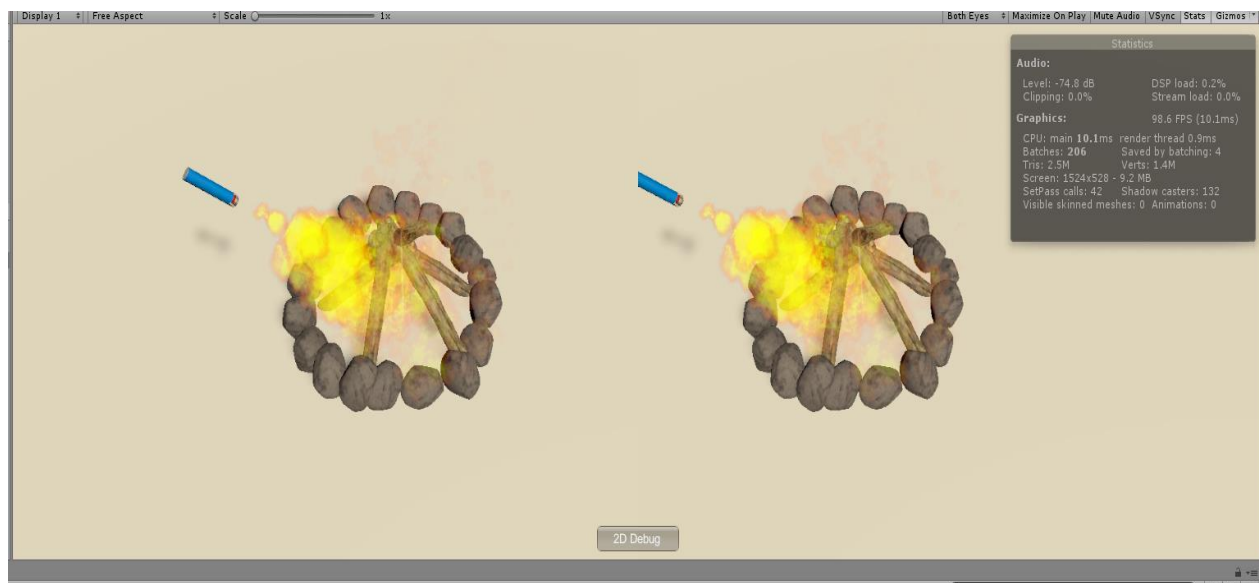


Рисунок 5.13 Зажигание костра

После активации зажигалка каждый FPS испускает луч на длину своего пламени, если на пути луча окажется объекта проверяются его свойства, если в его тегах стоит Ignitable, то вызывается принадлежащий объекту скрипт Campfire.cs и в нем вызывается метод Ignite , который поджигает костер и горит уже без использования зажигалки.

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 2.090301.06.001 ПЗ

Лист

36



Рисунок 5.14 Горящий костер

Костер горит без зажигалки.

5.3. Выводы

Система отработала штатно, количество FPS составляло не менее 90 что удовлетворяет требованиям по быстродействию.

ЗАКЛЮЧЕНИЕ

В ходе выполнения ВКР был проведен анализ предметной области, ПО для создания игр и ПО для 3D моделирования, аппаратные и программные аспекты VR. Изучена среда для разработки игр Unity, включая среду Microsoft Visual Studio.

При помощи детального анализа программный средств для реализации проекта было выбрано наиболее понятное и простое ПО, что позволило избежать трудностей при создании приложения и разработать его в ограниченный срок.

Создано приложение по обучению людей выживанию в дикой местности с возможностью наглядного обучения людей разведению огня.

Как указывалось раньше в моей работе с развитием технического прогресса, человек становится менее приспособлен к жизни и выживанию в диких условиях, следовательно с течением времени данная работа станет еще более актуальной. А с развитием средств VR, создаваемая модель реального мира будет все достовернее.

Дальнейшим логичным развитием этого приложения будет увеличение аспектов обучения как пример: добыча воды и еды. Предоставление пользователю простого и понятного интерфейса, который предоставлял бы ему информацию о полезности тех или иных действий.

Требования технического задания были выполнены в полном объеме и в установленные сроки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Селиванов В.В., Селиванова Л.Н. Виртуальная реальность как метод и средство обучения // Образовательные технологии и общество(№17), 2014, с. 388-389.
2. В. С. Бабенко Виртуальная реальность: Толковый словарь терминов. // ГУАП. - СПб., 2006, с. 87.
3. Хэммит Ф. Виртуальная реальность. // Вашингтон., 1993. С. 104. (спец. Перевод 2008).
4. Технология виртуальной реальности. Виртуальная реальность Virtual Reality (VR) , [Электронный ресурс] URL: http://www.tadviser.ru/index.php/Статья:Виртуальная_реальность
5. Очки виртуальной реальности Oculus rift, [Электронный ресурс] URL: <http://vr-j.ru/oculus-rift-ochki-virtualnoy-realnosti/>
6. PlayStation VR, [Электронный ресурс]: URL: https://ru.wikipedia.org/wiki/PlayStation_VR
7. Обзор Samsung Gear VR (SMR322): универсальные очки виртуальности, [Электронный ресурс]: URL: <http://4pda.ru/2016/03/14/283653>
8. HTC vive шлем виртуальной реальности, [Электронный ресурс]: URL: <https://vr-j.ru/htc-vive-shlem-virtualnoy-realnosti/>
9. Шлем виртуальной реальности, [Электронный ресурс]: URL: <http://fibrum.com/>
10. Селиванов В.В., Селиванова Л.Н. Эффективность использования виртуальной реальности при обучении в юношеском и взрослом возрасте // Непрерывное образование: XXI век.(№ 9), 2015, С. 16-18.
11. Авербух В.Л. К теории компьютерной визуализации // Вычислительные технологии (№ 10), 2005 , С. 21-51.
12. OpenVR [Электронный ресурс] URL: <https://partner.steamgames.com/doc/features/steamvr/openvr?l=russian>
13. VRTK[Электронный ресурс] URL: <https://www.vrtk.io/>
14. OVR SDK[Электронный ресурс] URL: <https://developer.oculus.com/>

15. GoogleVR SDK [Электронный ресурс] URL:
<https://developers.google.com/vr/>
16. Обзор самых популярных движков для разработки игр [Электронный ресурс] URL: // <https://xakep.ru/2014/09/05/game-development-engines-review/>
17. Unity (игровой движок) - Википедия [Электронный ресурс] URL:
<https://ru.wikipedia.org/wiki/Unity>
18. Полный обзор Unity 5 [Электронный ресурс] URL:
<http://devgam.com/polnyj-obzor-unity-5>
19. Unity – Руководство: Коллайдеры (Colliders) [Электронный ресурс] URL:
<https://docs.unity3d.com/2017.3/Documentation/Manual/CollidersOverview.html>
20. Особенности бесплатного кроссплатформенного движка Unity3d [Электронный ресурс] URL: <http://gamesmaker.ru/3d-game-engines/unity3d/osobennosti-besplatnogo-krossplatformennogo-dvizhka-unity3d/>
21. Противостояние 3D-титанов: 3ds Max, Maya, Blender — кто лучше? (часть 1) – YouTube [Электронный ресурс] URL:
<https://www.youtube.com/watch?v=ruIcEgPvl8Y>
22. Харьковский А.В. Лучший самоучитель 3ds Max 2013: Учебный курс. Изд. 4-е. М.: Компьютер-House iBoox.Ru, 2012 С. 477.
23. Полигональная сетка - Википедия [Электронный ресурс] URL:
https://ru.wikipedia.org/wiki/Полигональная_сетка
24. Вард Э. Разработка игрового персонажа: Kila. М.: Game Character Creation Series, 2017, С. 319
25. MonoDevelop - Википедия [Электронный ресурс] URL:
<https://ru.wikipedia.org/wiki/MonoDevelop>
26. Microsoft Visual Studio - Википедия [Электронный ресурс] URL:
https://ru.wikipedia.org/Microsoft_Visual_Studio
27. Unity – Руководство: Создание и Использование Скриптов [Электронный ресурс] URL:

<https://docs.unity3d.com/2017.3/Documentation/Manual/CreatingAndUsingScripts.html>

28. 15 самых популярных языков программирования по версии GitHub
[Электронный ресурс] URL: <https://habr.com/ru/post/310262/>

29. Управление игровыми объектами (GameObjects) с помощью
компонентов [Электронный ресурс] URL:
<https://docs.unity3d.com/2017.3/Documentation/Manual/ControllingGameObjectsComponents.html>

30. Unity – Руководство: Создание и Использование Скриптов
[Электронный ресурс] URL:
<https://docs.unity3d.com/2017.3/Documentation/Manual/CreatingAndUsingScripts.html>

ПРИЛОЖЕНИЕ А. Таблица сравнения средств 3D моделирования

Таблица 2 Сравнение средств 3D моделирования

Инструменты	3D Studio Max	Maya	Blender
Интерфейс	Плюсы:	Плюсы:	Плюсы:
Настройки	Минимализм	Функциональны	Простой
	Дружелюбность	й интерфейс	интерфейс
	Визуальная	Наличие кнопок	Удобные горячие
	понятность	с инструментами	клавиши
	Числовые	Возможности	Разные
	координаты	создания	раскладки
	всегда видны и	собственных	интерфейса
	доступны даже	полок	Фиксация вида
	для подобъектов	Отстегивающеес	при поворотах
	Квад-меню	я меню	Удобная работа
	Разнообразие	Наличие	со студией
	горячих клавиш	практически	Быстрый
	Гибкий стек	всех команд в	поиск команды
	модификаторов	меню, разные	Точный
	Пиктограммы	раскладки меню	клавиатурный
	инструментов	Настройк	ввод
		а абсолютно	большинства
	Минусы:	всего под себя	команд
	Неудобные	Минусы:	Минусы:
	текстовые поля	Много	Выделение
		настроек	правой кнопкой
		Неудобный	мышы
		большой	Несоответстви е
		интерфейс	стандартам
			управления

			видами Нестандартны й интерфейс
--	--	--	---------------------------------------

Продолжение Таблицы 2.

Удобство видовых окон	Плюсы: Есть видовой куб Есть Инструменты для работы с видами	Плюсы: Есть видовой куб Минусы: При выделении объекта появляется сетка	Минусы: Отсутствие поворота окна
Простота создания объектов	Плюсы: Легко найти Параметры создания видны в момент создания	Плюсы: Полнота инструментов	Плюсы: Есть меню, легко доступно
Разнообрази е объектов	Плюсы: Огромное Есть множество готовых решения для архитектурных проектов Есть отличная система создания волос с большим множеством инструментов	Достаточное Плюсы: Модуль PaintEffects, который дает возможность рисовать виртуальной кистью такие трехмерные объекты, как цветы, трава, объемные узоры и прочее	Минусы: Минимально достаточное

Свойства объекта	Плюсы: Наглядная параметризация	Минусы: Нет свойств объекта	Плюсы: Геометрически е объекты не параметризованы
---------------------	---------------------------------------	-----------------------------------	--

Продолжение Таблицы 2.

Моделирова ние	Плюсы: Изобилие параметрических примитивов Полноценный стек модификаторов Изобилие параметризованн ых модификаторов Числовые координаты всегда видны и доступны даже на уровне подобъектов Работа с кривыми безье Удобные сплайновые методы моделирования Мощное полигональное моделирование	Плюсы: Возможность настроить любые параметры через инструмент Channel Box или Attribute Editor Продвинутое полигональное моделирование Работа с кривыми безье Хорошая реализация NURBS Удобное управление гладкими и жесткими ребрами Любую задачу можно решить несколькими способами Более просто решаются	Плюсы: Числовые координаты всегда видны и доступны даже на уровне подобъектов Простые и необходимые модификаторы Удобное полигональное моделирование Работа с кривыми безье Сплайновое моделирование (специфичное) Простая и ясная идеология объекта Скульптинг Самое быстрое моделирование простых
-------------------	--	---	---

	Полное отсутствие не логичных инструментов для полигонального моделирование (функционал)	сложные задачи Существует продукт MayaLT , который сосредоточен	объектов (горячие клавиши) Минусы: Отсутствие параметрических примитивов
--	--	---	--

Продолжение Таблицы 2.

	Наличие уникальных инструментов (мелкая детализация и удобство работы) Превосходный инструмент для ретопологии Есть пред просмотр будущего hipoly объекта во время моделирования low-poly модели Возможность создавать скрипты Есть среда для архитекторов	только на играх Минусы: Нелинейное представление объекта (Отсутствие модификаторов и стека) Необходимост ь обращения с историей объекта Неудобный доступ к координатам подобъектов Есть не логичные инструменты для полигонального	Проблема гладких и жестких ребер Практически отсутствует NURBS моделирование Есть не логичные инструменты для полигонального моделирования (функционал) Позволяет моделировать в 3-5 раз меньше полигонов, чем в аналогичных продуктах
--	--	---	---

	Минусы: Сложная реализация NURBS	моделирования (функционал) Не удобный инструмент ретопологии, многие используют другие программы и скрипты	
--	---	---	--

Продолжение Таблицы 2.

Полнота главного меню	Плюсы: Логично Много общих команд Легко найти Параметры создания видны в момент создания	Плюсы: Меню полное Переключение интерфейса	Минусы: Нестандартное меню Непривычно для начинающего
Контекстное меню	Присутствует, но малоинформатив но	Плюсы: Marking menu	Минусы: Не работает контекстное меню на объекте
Удобство преобразований	Плюсы: Отличное Всегда видны текстовые поля координат	Хорошее	Хорошее, но инструменты маловаты и координаты не совсем очевидны
Стек модификаторов	Плюсы:	Плюсы:	Плюсы:

	Все реализовано через стек Огромное количество модификаторов Удобен как для новичков, так и для профессионалов	Есть история создания объекта Минусы: Нелинейное представление объекта (Отсутствие модификаторов и стека)	Стек есть, но меньше модификаторов
Булевские операции	Специфическая реализация	Минусы: трудная реализация	Плюсы: просто реализованы

Продолжение Таблицы 2.

Переход к подобъектам	Плюсы: По горячим клавишам и модификаторам	Контекстное меню	Через список
Настройки материалов	Минусы: Неудобная реализация	Минусы: Трудная настройка	Плюсы: Легко настраиваются и редактируются
Работа с развертками	Плюсы: На простых объектах удобнее и быстрее выполняет задачу	Плюсы: Наличие системы рисования и ретуширования текстур прямо на самом объекте Удобно	Плюсы: Хорошо справляется с разверткой простых объектов Минусы:

	<p>Минусы:</p> <p>Производительность</p>	<p>заканчивать процесс сложной развертки путем сглаживания кластеров</p> <p>Минусы:</p> <p>Неудобство выполнения</p>	<p>Мало инструментов</p>
Текстурирование и запекание	<p>Плюсы:</p> <p>Легко сделать</p>	<p>Плюсы:</p> <p>Наличие системы</p>	<p>Плюсы:</p> <p>Удобная и</p>

Продолжение Таблицы 2.

(Baking)	<p>быструю развертку или накинуть текстуру</p> <p>Минусы:</p> <p>Запекание</p>	<p>рисования и ретуширования текстур прямо на самом объекте</p> <p>Удобно</p> <p>заканчивать процесс сложной развертки путем сглаживания кластеров</p>	<p>логичная система рисования и ретуширования текстур прямо на самом объекте</p> <p>Удобно делать сложную развертку швами (для персонажей)</p>
Источники света	<p>Плюсы:</p> <p>Есть нацеленные источники света</p>	<p>Минусы:</p> <p>Источники нужно</p>	<p>Наглядно настраиваются и создаются</p>

	Фотореалистичны	настраивать (по теням)	
Визуализация	Плюсы: Вызывается по одной кнопке Множество различных альтернативных визуализаторов Есть визуализатор Vray (преимущество над Maya)	Плюсы: Вызывается по одной кнопке Минусы: Нет бесплатных лицензий для визуализатора Mental Ray	Минусы: Рендер только из камеры Малое количество визуализаторов
Количество обучаемого материала	Плюсы: Огромное множество	Достаточно	Достаточно

Продолжение Таблицы 2.

Востребованность на данный момент	Плюсы: Популярно как и среди новичков, так и профессионалов	Плюсы: Очень востребовано	Минусы: Низкая, большие студии не практически не используют Большая трата на изучение интерфейса сужает круг потенциальных пользователей
-----------------------------------	--	------------------------------	---

Работа с анимацией	Плюсы:	Плюсы:	Плюсы:
	Имеется готовая иерархическая система костей Biped и CAT. Все кости (данных систем) легко настраиваются и интегрируются в другие сцены (возможность работы с файлами)	Самая реалистичная анимация Идеально подходит для профессиональных аниматоров (Используют крупнейшие анимационные студии (Disney, Pixar)) Создаются в большей степени уникальные анимации	Инструменты для удобного скиннинга — встроенный скрипт. Минусы: Не идеальная система инверсной кинематики Приходится много исправлять последствия не точного скиннинга

Продолжение Таблицы 2.

	Существует огромное множество готовых наборов качественных анимаций, созданных с помощью Motion Capture Для иерархии костей не нужно	Превосходная система Motion Capture (Human IK) Многопроходный рендеринг RenderMan Минусы: Приходится анимировать с	
--	--	--	--

	настраивать инверсную кинематику Идеально подходит для простых анимаций для игр Возможность обмениваться файлами анимации Минусы: Анимация не реалистична Без настраиваемой системы костей сложно создавать и управлять обычными костями Плюсы	нуля Нужны вспомогательны е объекты Специфическая работа с костями, долгая настройка Очень сложно вносить из изменения для готовой анимации, вплоть до переделывания проекта	
--	--	--	--

Продолжение Таблицы 2.

Цена	Минусы: Не многие студии могут позволить (Около 200 000 рублей за трехлетнюю лицензию)	Минусы: Не многие студии могут позволить (Около 200 000 рублей за	Плюсы: Бесплатность
------	--	--	------------------------

		трехлетнюю лицензию)	
Технические моменты	<p>Плюсы:</p> <p>Закрытость архитектуры (программное и пользовательское представление почти не связаны)</p> <p>Существует огромное множество готовых скриптов или плагинов MAX Script</p> <p>Минусы:</p> <p>Не очень хорошая стабильность</p> <p>Поддерживает только ОС Windows</p>	<p>Плюсы:</p> <p>язык программирования mel</p> <p>Программистам удобно работать с объектам</p>	<p>Плюсы:</p> <p>Очень высокая производительность для малых проектов</p> <p>Высокая стабильность</p> <p>Низкие требования к ПК</p> <p>Быстрая установка (Установочный файл занимает мало места) на диске</p> <p>Минусы:</p> <p>Недоработка драйверов видеоадаптера</p> <p>Нет поддержки OpenGL</p>

ПРИЛОЖЕНИЕ Б. Листинги программы

Файл «Campfire.cs»

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Campfire : MonoBehaviour
{
    public ParticleSystem fire;
    // Start is called before the first frame update
    void Start() { }
    void Update() { }
    public void Ignite()
    {
        if (!fire.gameObject.activeSelf)
        {
            fire.gameObject.SetActive(true);
            if (!fire.isPlaying)
            {
                fire.Play();
            }
        }
    }
}
```

Файл «PlayerMovement.cs»

```
public class PlayerMovement : MonoBehaviour
{
    private Vector2 movementAxis;
    private Vector3 moveDirection;
    private CapsuleCollider capsuleCollider;
    private bool onGround=true;
    private bool firstTime=true;
    public SteamVR_Input_Sources MovementHand;//Set Hand To Get
Input From
    public SteamVR_Action_Vector2 TrackpadAction;
    public SteamVR_Action_Boolean JumpAction;
    public float jumpHeight;
    public float walkSpeed;
    public float Deadzone;//the Deadzone of the trackpad. used
to prevent unwanted walking.
    public Transform head;
    public GameObject AxisHand;//Hand Controller GameObject
    public PhysicMaterial NoFrictionMaterial;
    public PhysicMaterial FrictionMaterial;
    public Tutorial tutorialText;
    void Awake()
    {
        capsuleCollider = GetComponent<CapsuleCollider>();
    }
}
```

```

void Start()    {
tutorialText.initialTrigger=true;
}
void FixedUpdate()
{
    UpdateInput();
    UpdateCollider();
    if (onGround)
    {
        if (movementAxis.magnitude > Deadzone)
        {
            moveDirection
Quaternion.AngleAxis(Angle(movementAxis)
AxisHand.transform.localRotation.eulerAngles.y,    Vector3.up)
Vector3.forward;//get the angle of the touch and correct it for the
rotation of the controller

        if (firstTime)
        {
            tutorial.initialTrigger = true;
            firstTime = false;
        }
        this.transform.Translate(new
Vector3(moveDirection.x    *    walkSpeed,    0,    moveDirection.z    *
walkSpeed));
    }
}

public static float Angle(Vector2 p_vector2)
{
    if (p_vector2.x < 0)    {    return 360
- (Mathf.Atan2(p_vector2.x, p_vector2.y) * Mathf.Rad2Deg * -1);
}    else    {    return Mathf.Atan2(p_vector2.x,
p_vector2.y) * Mathf.Rad2Deg;    }    }

void UpdateCollider()
{
    float    distanceFromFloor    =
Vector3.Dot(head.localPosition, Vector3.up);
    capsuleCollider.height    =
Mathf.Max(capsuleCollider.radius, distanceFromFloor);
    capsuleCollider.center = head.localPosition - 0.5f *
distanceFromFloor * Vector3.up;
    // transform.localPosition = head.localPosition - 0.5f
* distanceFromFloor * Vector3.up;
}

void UpdateInput()
{
    movementAxis = TrackpadAction.GetAxis(MovementHand);
}
}

```

Файл «Lighter.cs»

namespace Valve.VR.InteractionSystem

					ПГУ 2.090301.06.001 ПЗ	Лист
						54
Изм.	Лист	№ докум.	Подпись	Дата		

```

{    public class Lighter : MonoBehaviour
    {    public float joyMove = 0.1f;
        public float fireDistance;
        public ParticleSystem fire;
        public GameObject test;
        //      public  SteamVR_Action_Vector2  moveAction  =
SteamVR_Input.GetAction<SteamVR_Action_Vector2>("platformer",
"Move");
        //      public  SteamVR_Action_Boolean  jumpAction  =
SteamVR_Input.GetAction<SteamVR_Action_Boolean>("platformer",
"Jump");
        public      SteamVR_Action_Boolean      lightAction      =
SteamVR_Input.GetAction<SteamVR_Action_Boolean>("default",
"ButtonAction1");
        //  private Vector3 movement;
        private bool lighted=false;
        // private float glow;
        private SteamVR_Input_Sources hand;
        private Interactable interactable;

        private void Start()
        {
            interactable = GetComponent<Interactable>();
        }

        // Update is called once per frame
        void FixedUpdate()
        {
            if (NeedToLight())
            {
                Debug.Log("Check true 1");
                if (!lighted)
                {
                    Debug.Log("Lighting");

                    fire.gameObject.SetActive(true);
                    fire.transform.position
this.transform.position;
                    fire.transform.Translate( new Vector3(0, 0,
0.1f));
                    fire.transform.rotation
this.transform.rotation;
                    fire.transform.Rotate(new Vector3(-90, 0,
0));
                    fire.Play();
                    lighted = true;
                }
            }
            else
            {
                fire.transform.position
this.transform.position;
                fire.transform.rotation
this.transform.rotation;
            }
        }
    }
}

```

```

        fire.transform.Rotate(new Vector3(-90, 0,
0));
        fire.transform.Translate(new Vector3(0, 0,
0.1f));
        //Кастуем вверх так как нас интересует верх
зажигалки ведь оттуда исходит пламя
        Vector3 forward =
transform.TransformDirection(Vector3.up) * fireDistance;
        // Ray ray = new
Ray(this.transform.position, forward);
        RaycastHit hit;

if(Physics.Raycast(transform.position, forward, out
hit, fireDistance))
    {
        if(hit.collider.gameObject.tag ==
"Ignitable"){
hit.collider.gameObject.GetComponent<Campfire>().Ignite();
        }
        Debug.DrawRay(transform.position, forward,
Color.red);
    }
else
{
    Debug.Log("Putting down");

    fire.gameObject.SetActive(false);
    lighted = false;
}
private bool NeedToLight()
{
    Debug.Log("Checking");
    if (interactable.attachedToHand)
    {
        hand = interactable.attachedToHand.handType;
        Debug.Log("Check True");
        return lightAction[hand].state;
    }
    else
    {
        return false;
    }
}
}
}

```

Файл «Tutorial.cs»

					ПГУ 2.090301.06.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		56

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Tutorial : MonoBehaviour
{
    public Text tutorialText;
    public bool nearWater;
    public bool fireTrigger;
    public bool coverTrigger;
    public bool initialTrigger;

    private float tutorialTime = 10.0f;
    private float initialTime;
    private float currTime;
    // Start is called before the first frame update
    void Start()
    {
        initialTime = Time.time;
        currTime = Time.time;
    }
    void Update()
    {
        currTime = Time.time;
        if (currTime - initialTime > tutorialTime)
        {
            if (initialTrigger)
            {
                initialTrigger = false;
                initialTime = Time.time;
                tutorialText.text = "Приветствую вас в обучающей
программе по выживанию в дикой местности." +
                " В первую очередь выживание строится на 4
пунктах" +
                "
                1.Еда
                2.Вода
                3.Укрытие
                4.Спасение
                " +
                "Дальнейшие этапы обучения кратко расскажут
вам о них.";
            }
            if (fireTrigger)
            {
                initialTime = Time.time;
                tutorialText.text = "Приветствую тут вы
научитесь использовать огонь. Огонь не только будет согревать вас,
он отгоняет диких зверей, поможет спасателям найти вас, а также
поможет вам вскипятить воду и приготовить еду, чтобы проверить работу
костра возьмите зажигалку левым контроллером и нажмите кнопку а,
после чего поднесите ее к костру.";";
                fireTrigger = false;
            }
            if (nearWater)

```

```

        {
            initialTime = Time.time;
            tutorialText.text = "Вода один из важнейших
факторов вашего выживания в зависимости от физической нагрузки
температуры и влажности без воды вы можете обходиться от 12 до 72
часов. Чем выше температура и ваша физическая нагрузка тем больше и
быстрее вам потребуется вода так как большая её часть будет
выводиться из вашего организма в виде пота для его охлаждения. Вы
можете добыть воду из рек и озер на дикой местности, но не забывайте
её вскипятить чтобы избавиться от бактерий, вода должна кипеть не
менее 5 минут чтобы очиститься от болезнетворных бактерий и
микроорганизмов, также если в воде присутствует осадок не мешает
ее процедить. Помните вода один из важнейших приоритетов так как время
которое вы проживете без еды измеряется днями и даже в редких случаях
неделями, а воды часами при выживании поиск воды ваш первый
приоритет. Помимо разумеется безопасности, так как возле водоемов в
дикой местности могут находиться дикие животные. ";
            nearWater = false;
        }
        if (coverTrigger)
        {
            initialTime = Time.time;
            tutorialText.text = "Если у вас есть вода и еда
, последний этап это укрытие которое не позволит вам промокнуть или
замерзнуть, вы можете соорудить укрытие из палок и веток. Без укрытия
первый дождь или сильный ветер сильно уменьшат ваши шансы на
выживание, так как вы легко можете заболеть и потерять остатки тепла.
После сооружения укрытия старайтесь любыми способами подать сигнал
спасателям где вы находитесь, это могут быть дымовые сигналы от
костра, выложенное камнями или палками слова SOS на открытой
местности или блеск отраженный от зеркала или металла. Не
отчаивайтесь и вы всегда выживете.";
            coverTrigger = false;
        }
    }
}

```

Файл «FireCover.cs»

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class FireCover : MonoBehaviour
{
    public Tutorial tutorial;
    void Start()
    {

    }

    void Update()

```

```

        {
        }
        private void OnTriggerEnter(Collider other)
        {
            if (other.gameObject.GetComponent<PlayerMovement>() !=
null)
            {
                tutorial.fireTrigger = true;
            }
        }
    }
}

```

Файл «WaterCover.cs»

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class FireCover : MonoBehaviour
{
    public Tutorial tutorial;
    void Start()
    {

    }
    void Update()
    {

    }
    private void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.GetComponent<PlayerMovement>() !=
null)
        {
            tutorial.nearWater = true;
        }
    }
}

```

Файл «CoverCover.cs»

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class FireCover : MonoBehaviour
{
    public Tutorial tutorial;
    void Start()
    {

```

```

    }
    void Update()
    {

    }
    private void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.GetComponent<PlayerMovement>() !=
null)
        {
            tutorial.coverTrigger = true;
        }
    }
}

```

					ПГУ 2.090301.06.001 ПЗ	Лист
						60
Изм.	Лист	№ докум.	Подпись	Дата		