

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ  
Заведующий кафедрой ЭВМ  
\_\_\_\_\_ Д. В. Топольский  
«\_\_» \_\_\_\_\_ 2023 г.

РАЗРАБОТКА ТРЁХМЕРНОЙ КОМПЬЮТЕРНОЙ ИГРЫ «КУКЛА»  
НА БАЗЕ “UNREAL ENGINE”

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  
ЮУРГУ-090301.2023.360 ПЗ ВКР

Руководитель работы,  
к.п.н., доцент каф. ЭВМ  
\_\_\_\_\_ Ю. Г. Плаксина  
«\_\_» \_\_\_\_\_ 2023 г.

Автор работы,  
студент группы КЭ-405  
\_\_\_\_\_ А. А. Еланский  
«\_\_» \_\_\_\_\_ 2023 г.

Нормоконтролер,  
ст. преп. каф. ЭВМ  
\_\_\_\_\_ С. В. Сяськов  
«\_\_» \_\_\_\_\_ 2023 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ  
Заведующий кафедрой ЭВМ  
\_\_\_\_\_ Д. В. Топольский  
«\_\_» \_\_\_\_\_ 2023 г.

**ЗАДАНИЕ**  
**на выпускную квалификационную работу бакалавра**  
студенту группы КЭ-405  
Еланскому Александру Андреевичу,  
обучающемуся по направлению  
09.03.01 «Информатика и вычислительная техника»

1. **Тема работы:** «Разработка компьютерной игры «Кукла» на базе «UNREAL ENGINE»» утверждена приказом по университету № 753-13/12 от 25 апреля 2023 г.

2. **Срок сдачи студентом законченной работы:** 1 июня 2023 г.

3. **Исходные данные к работе:**

Функциональные требования:

- управление перемещением персонажа с помощью клавиатуры;
- перемещение персонажа по прямым, наклонным плоским поверхностям и ступеням;
- предметы окружения должны менять своё положение и вращаться при контакте друг с другом и игроком;
- ликвидация NPC только попаданием снаряда;
- при ликвидации NPC анимации переключаются на Ragdoll;

- выстрелы оружия создают снаряд, для которого рассчитывается баллистика;
- выстрелы оружия должны сопровождаться звуком выстрела;
- написание сюжета про робота с искусственным интеллектом, внезапно осознавшим себя;
- русскоязычный интерфейс;
- операционная система Windows 10 и выше;
- DitecrX12 и выше;
- видеокарта с аппаратным ускорением трассировки лучей: Nvidia RTX 2060 и выше, AMD Radeon 6600 и выше;
- использование графических технологий Nanite, DXR;
- платформа: Unreal Engine 5.

#### **4. Перечень подлежащих разработке вопросов:**

- 4.1. обзор существующих аналогов разрабатываемого проекта;
- 4.2. анализ и выбор средств реализации проекта согласно поставленным требованиям;
- 4.3. проектирование и реализация проекта;
- 4.4. тестирование.

#### **5. Дата выдачи задания: 01 декабря 2022 г.**

Руководитель работы \_\_\_\_\_ /Ю. Г. Плаксина/

Студент \_\_\_\_\_ /А. А. Еланский/

## КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Обзор существующих аналогов разрабатываемого проекта	07.02.2023	
Анализ и выбор средств реализации проекта согласно поставленным требованиям	20.02.2023	
Проектирование и реализация проекта	07.03.2023	
Тестирование	04.04.2023	
Компоновка текста работы и сдача на нормоконтроль	17.05.2023	
Подготовка презентации и доклада	24.05.2023	

Руководитель работы \_\_\_\_\_ /Ю. Г. Плаксина/

Студент \_\_\_\_\_ /А. А. Еланский/

## АННОТАЦИЯ

А. А. Еланский. Разработка компьютерной игры «Кукла» на базе Unreal Engine 5. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2023, 41 с., 23 ил., библиогр. список – 29 наим.

В рамках выпускной квалификационной работы выполняется анализ аналогичных проектов с целью выявления положительных и отрицательных аспектов в программной реализации и сюжетной части. Формулируются требования, предъявляемые к разрабатываемому продукту. Проводится исследование сред разработки с целью выбора наиболее подходящего к поставленным требованиям игрового движка. Производится проектирование проекта, включающее работу с игровой логикой, материалами, освещением и анимациями. Осуществляется разработка демонстрационной версии программного проекта с учетом предложенных требований. Производится тестирование реализованной демоверсии компьютерной игры, доказывающая способность проекта удовлетворять техническому заданию.

Результатом выполненной работы является компьютерная игра, в которой присутствует следующий функционал:

- стрельба с симуляцией баллистики снаряда;
- интерактивное окружение;
- перемещение персонажа по прямым, наклонным плоским поверхностям и ступеням;
- уничтожение NPC при попадании по модели;
- уничтожение игрока при касании NPC;
- освещение и отражения с трассировкой лучей.

## ОГЛАВЛЕНИЕ

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ.....	7
ВВЕДЕНИЕ.....	9
1 ОБЗОР СУЩЕСТВУЮЩИХ АНАЛОГОВ РАЗРАБАТЫВАЕМОГО ПРОЕКТА.....	11
2 АНАЛИЗ И ВЫБОР СРЕДСТВ РЕАЛИЗАЦИИ ПРОЕКТА СОГЛАСНО ПОСТАВЛЕННЫМ ТРЕБОВАНИЯМ .....	20
2.1 Функциональные требования .....	20
2.2 Нефункциональные требования .....	20
2.3 Выбор игрового движка .....	20
3 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОЕКТА .....	23
3.1 Архитектура предлагаемого решения.....	23
3.2 Исходные материалы .....	27
3.3 Главный герой .....	28
3.4 Неигровые персонажи .....	30
3.5 Игровой процесс.....	31
3.6 Игровая логика .....	31
4 ТЕСТИРОВАНИЕ .....	34
ЗАКЛЮЧЕНИЕ .....	35
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	36
ПРИЛОЖЕНИЕ. Код программы.....	40

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящей пояснительной записке применяются следующие определения, обозначения и сокращения:

**Blueprint** – визуальный язык программирования, использующийся в игровом движке Unreal Engine 5

**DXR** (от англ. DirectX Raytracing) – технология, моделирующая натуральное физическое поведение света, позволяющая получить максимально реалистичное изображение в графически насыщенных приложениях и играх

**FPS** (от англ. First Person Shooter) – жанр компьютерных игр, в которых игровой процесс основывается на сражениях с использованием огнестрельного или любого другого оружия с видом от первого лица

**Lumen** – полностью динамическая система глобального освещения и отражений в Unreal Engine 5

**Natine** – система виртуализации геометрии, позволяющая достичь равного количества полигонов и пикселей в сцене

**NPC** (от англ. Non-Player Character) – персонаж игры, управляемый искусственным интеллектом (не игроком)

**Nvidia Shadowplay** – программа для записи экрана и проверки загруженности компьютера

**PBR** (от англ. Physically based rendering) – метод компьютерной графики позволяющий более достоверно отображать материалы

**Ragdoll** – процедурно генерируемая анимация для изображения смерти персонажа

**Игровой ассет** (от англ. Game Asset) – цифровой объект, преимущественно состоящий из однотипных данных, неделимая сущность, которая представляет часть игрового контента и обладает некими свойствами

**UE5** (от англ. Unreal Engine 5) – игровой движок компании Epic Games

**Шутер** (от англ. Shooter) – вид компьютерной игры, в которой главной целью игрока является поражение противника и избегание его попаданий

**Игровой движок** (от англ. game engine) – программное обеспечение, предназначенное для разработки видеоигр и, как правило, включающее соответствующие библиотеки и вспомогательные программы



## ВВЕДЕНИЕ

В современном мире игровая индустрия стремительно развивается и пользуется огромным спросом почти у всех слоев населения. Данный тезис подтверждается статистической информацией аналитического издания Gameindustry.biz, согласно которому на 2021 год финансовая стоимость игровой индустрии составила порядка 180,3 миллиарда долларов, 20% от данной суммы приходится на игровые решения для настольных компьютеров [1]. Компьютерные игры способствуют отвлечению игрока от рутины реального мира, что положительно влияет на эмоциональное и психическое состояние здоровья пользователя [2]. Шутеры могут быть как одиночными, так и мультиплеерными. Мультиплеерные игры могут быть соревновательными, когда игроки сражаются друг с другом, и кооперативными, когда игроки сражаются вместе против искусственного интеллекта. Мультиплеерные игры, как правило, кажутся пользователям веселее из-за игры в компании, но создание сетевых функций трудоёмкий процесс. Даже физическое расположение серверов играет роль, реакция сервера на любые действия игрока более 75 миллисекунд оттолкнёт пользователя от проекта. Играя вместе, игроки часто шутят и создают юмористический настрой, что может отличаться от настроения, задуманного автором.

Современный игровой рынок предъявляет широкий спектр требований к разрабатываемым игровым решениям. Отметим наиболее важные из них:

- качество графики – низкий уровень графики может отпугнуть пользователя от игрового решения на ранней стадии прохождения игры;

- звуковое сопровождение элементов игры – правильно подобранное звуковое сопровождение позволяет осуществить более глубокое погружение в игровую среду, создавая задуманное автором настроение и восприятие игровой сцены;

– искусственный интеллект неигровых персонажей NPC – наличие проработанных NPC позволяет разработчику реализовывать динамические захватывающие пользователя игровые сцены;

– интерактивность мира – низкая возможность взаимодействия с окружающим миром быстро наскучит пользователю, в результате чего игрок, скорее всего, перестанет быть заинтересованным в проекте.

Реализация подобных игр требует огромных временных затрат одной или нескольких групп разработчиков, и, как правило, дополнительных финансовых вложений вследствие сложности реализации проекта. Но, при этом, данный класс игр пользуется огромной популярностью среди игрового комьюнити.

Целью выпускной квалификационной работы является создание игры с 3D отображением графики.

Для достижения поставленной цели необходимо решить следующий ряд задач:

– провести оценку работоспособности разработанного проекта.  
провести обзор существующих аналогов разрабатываемого проекта;

– выполнить анализ и выбор средств реализации проекта согласно поставленным требованиям;

– спроектировать и реализовать проект;

– провести тестирование и оценку работоспособности разработанного проекта.

Актуальность работы: разработка компонента, позволяющего перемещать объекты в сцене, для редактора Unreal Engine 5. Работа компонента была протестирована в ходе разработки игры «Кукла».

# 1 ОБЗОР СУЩЕСТВУЮЩИХ АНАЛОГОВ РАЗРАБАТЫВАЕМОГО ПРОЕКТА

В данном разделе будет произведен обзор аналогов разрабатываемого приложения, результатом которого будет являться ряд предлагаемых разработчиком технических решений, необходимых для реализации собственного проекта.

Обзор аналогов будет осуществлён по следующему ряду критериев:

- 1) тип стрельбы;
- 2) вид камеры;
- 3) графика;
- 4) интерактивность окружения.

**Стрельба** – основа игры жанра «шутер», от качества исполнения которой зависит общее восприятие проекта. Стрельба, наиболее близко приближенная к реальной, позволяет реализовать крайне глубокое погружение игрока в игровой процесс.

Существует две технологии реализации стрельбы:

- Hitscan;
- баллистика снаряда.

При стрельбе Hitscan из дула оружия в направлении прицела испускается невидимый луч, который, при пересечении цели, регистрирует попадание по цели. Недостатком подхода является нереалистичное поведение полета пули, поскольку луч преодолевает любое расстояние мгновенно, не испытывая воздействия гравитации и ветра. Однако стрельба Hitscan проста для вычисления и в разработке.

Стрельба с баллистикой снаряда при выстреле создаёт снаряд, обладающий импульсом и весом. Расчёт баллистики снаряда сложнее чем Hitscan, но более реалистичен.

Также немаловажным аспектом достижения максимального погружения в игру является работа с камерой в процессе отображения сцен на экране

пользователя. Вид от третьего лица упрощает управление персонажем при игре на игровых консолях, но большему погружению в игру способствует вид от первого лица.

**Графический аспект** определяет реалистичность изображения, отображаемую на экране пользователя, и напрямую влияет на игрока. Чем более реалистично будет реализована сцена и персонажи, тем более затягивающим будет игровой процесс для пользователя.

Также для усиления иллюзии реальности игрового мира используется **интерактивность окружения** – опрокинутый со стола при попадании предмет углубит погружение игрока в игру, делая взаимодействие с интерактивными предметами на игровом уровне более зрелищным.

Вышеперечисленные критерии способствуют погружению игрока в состояние потока, сформулированное американским психологом Михай Чиксентмихайи. Он описывал поток (Flow) как ощущение полного, активного вовлечения в занятие, приносящие радость и удовлетворение. Другой американский психолог Роберт Хольт считал описание потока идентично ощущениям, которые переживает игрок, полностью погруженный в игру. В течение этого опыта игрок игнорирует внешние факторы и не следит за временем [3].

Для обзора аналогов были выбраны следующие игровые проекты:

- 1) Call of Duty: Vanguard;
- 2) Halo Infinite;
- 3) Deathloop;
- 4) Outriders;
- 5) Battlefield 2042;
- 6) Far Cry 6;
- 7) Hitman 3;
- 8) Back 4 Blood;
- 9) Resident Evil Village;
- 10) Necromunda: Hired Gun

**Call of Duty: Vanguard** [4] – шутер от первого лица от Activision Blizzard. Скриншот постановочного ролика представлен на рисунке 1.



Рисунок 1 – Постановочный ролик Call of Duty: Vanguard

К преимуществам проекта можно отнести дорогие анимации постановочных роликов, а именно: захват движения, сложные материалы из большого количества текстур.

К недостаткам можно отнести отсутствие интерактивности окружения и трассировки лучей, а также стрельбу Hitscan. Также недостатком является цена в 3499 рублей за стандартное издание и 5899 рублей за полное издание.

**Halo Infinite** [5] – шутер от первого лица от Microsoft. Скриншот игры представлен на рисунке 2.



Рисунок 2 – Скриншот Halo Infinite [5]

К преимуществам проекта можно отнести открытый мир, интерактивность объектов, стрельба с симуляцией снаряда и его баллистики.

К недостаткам можно отнести отсутствие трассировки лучей.

**Deathloop** [6] – шутер от первого лица от Microsoft.

К преимуществам проекта можно отнести интерактивность окружения, наличие теней с использованием трассировки лучей.

К недостаткам проекта можно отнести:

– отсутствие трассировки лучей освещения и отражений за исключением платформы Playstation 5;

– Hitscan стрельба;

– цена в 3199 рублей;

– техническое состояние проекта, поскольку на персональных компьютерах игра часто подтормаживает вне зависимости от мощности системы.

Сравнение графики от Digital Foundry представлено на рисунке 3.



Рисунок 3 – Трассировка теней в Deathloop на Playstation 5 [7]

**Outriders** [8] – шутер от третьего лица от Square Enix. Скриншот игры представлен на рисунке 4.

К преимуществам проекта можно отнести возможность совместного прохождения сюжетной кампании с другими игроками по сети, а также цену в 1999 рублей.



Рисунок 4 – Outriders игра в кооперативе по сети [9]

К недостаткам проекта можно отнести:

- большое количество экранов загрузки;
- карты состоит из большого количества маленьких уровней, между которыми появляются экраны загрузки;
- периодическое появление обрывов или дверей, для преодоления которых игра запускает загрузку, демонстрируя короткий ролик (в среднем 3 секунды), прыжок и повторный запуск загрузки;
- отсутствие трассировки лучей.

**Battlefield 2042** [10] – шутер от первого лица от Electronic Arts. Демонстрация графики с трассировкой лучей представлена на рисунке 5.

К достоинствам проекта можно отнести большие открытые локации, трассировку лучей для теней, стрельба с симуляцией снаряда и баллистики.

К недостаткам проекта можно отнести отсутствие трассировки для освещения и отражений, отсутствие одиночной кампании, а также техническое состояние: кадровая частота может резко опуститься ниже 30 кадров в секунду в без изменений в сцене, даже на мощных персональных компьютерах. Не регистрирует попадания при визуальном отображении попадания, игроки иногда становятся невидимыми друг для друга. Цена 3499 рублей для базового издания, 5499 рублей для Gold издания, 6999 рублей для Ultimate издания.

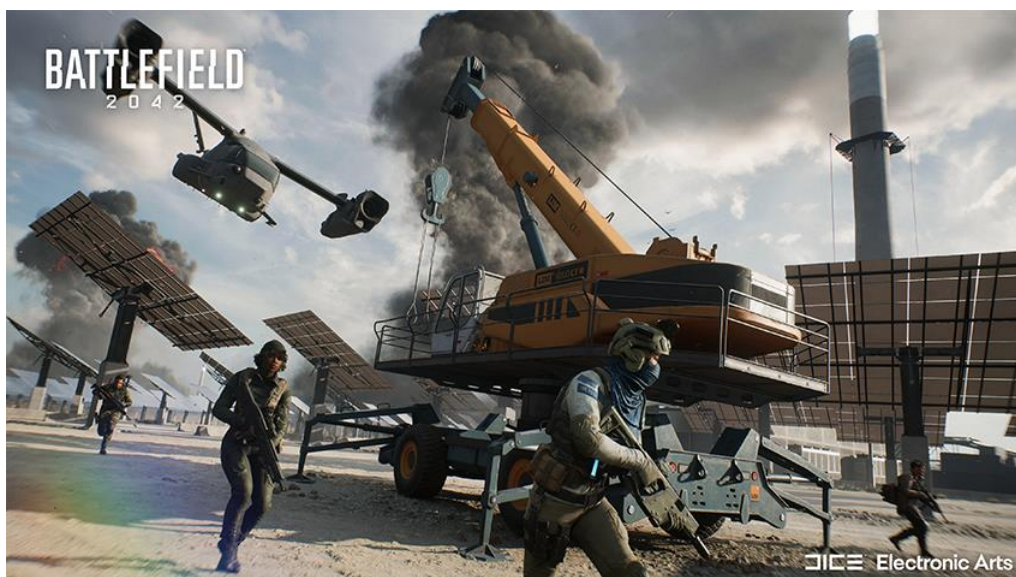


Рисунок 5 – Battlefield 2042 демонстрация трассировки лучей [11]

Far Cry 6 [12] – шутер от первого лица от компании Ubisoft. Демонстрация графики с трассировкой лучей на рисунке 6.



Рисунок 6 – Far Cry 6 демонстрация трассировки лучей [13]

К достоинствам проекта можно отнести большой открытый мир, трассировку лучей для теней и отражений, интерактивность мира и возможность кооперативного прохождения сюжетной кампании.

К недостаткам проекта можно отнести отсутствие трассировки лучей для глобального освещения, 4 версии издания, стандартное издание ценой 2500 рублей, полное издание ценой 5000 рублей, которое не включает в себя весь дополнительный контент.



**Hitman 3** [14] – шутер от третьего лица от компании IO Interactive.  
Скриншот представлен на рисунке 7.

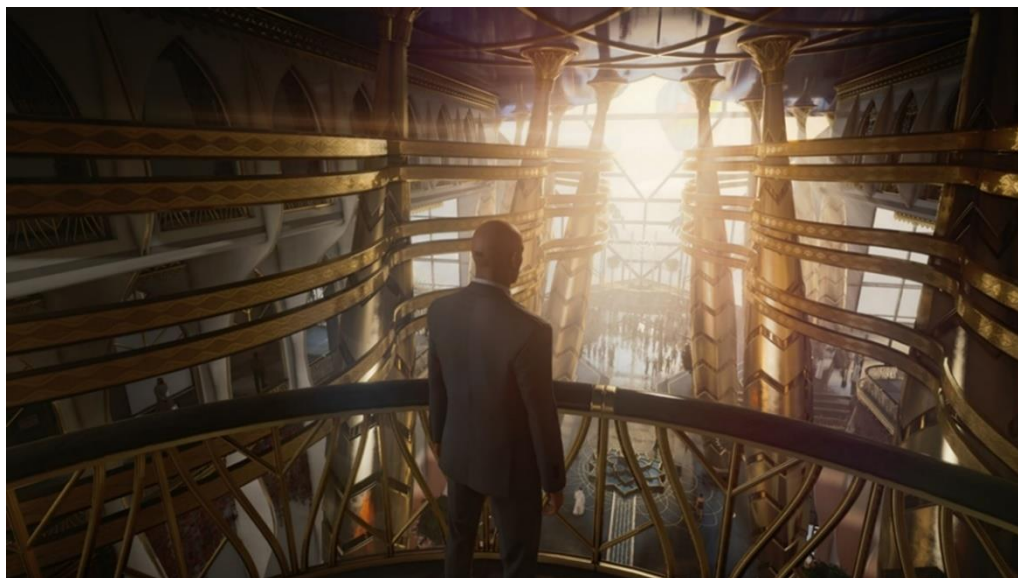


Рисунок 7 – Скриншот Hitman 3 [14]

К достоинствам можно отнести большое количество разнообразных локаций, развитый искусственный NPC.

К недостаткам можно отнести отсутствие трассировки лучей, цену в 2698р за базовое издание и 3999 рублей за полное.

**Back 4 Blood** [15] – шутер от первого лица от Turtle Rock Studios.  
Скриншот Back 4 Blood представлен на рисунке 8.

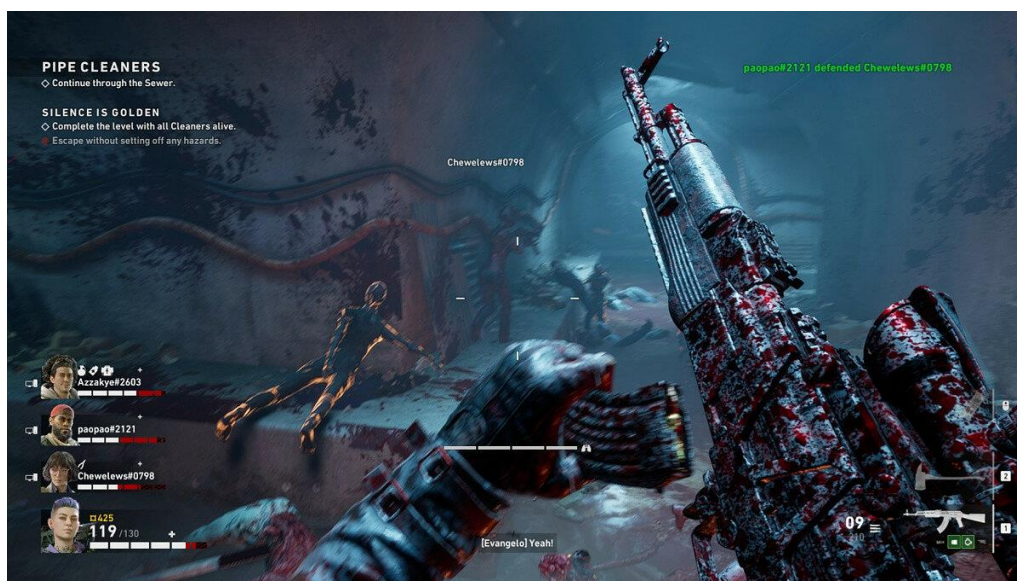


Рисунок 8 – Скриншот Back 4 Blood [15]

К достоинствам можно отнести игру как в одиночку, так и в кооперативе до 4 человек, цену в 2000 рублей за стандартное издание, развитый общий

искусственный интеллект вражеских NPC, генерирующий интересные игровые ситуации.

Недостатками являются отсутствие трассировки лучей, полное издание ценой 3000 рублей, а также не развитый искусственный интеллект союзных NPS.

**Resident Evil Village** [16] – шутер от первого лица от компании Capcom. Скриншот Resident Evil Village представлен на рисунке 9.



Рисунок 9 – Отсутствие отражений на стеклах в Resident Evil Village с включенной трассировкой лучей для отражений

К достоинствам проекта можно отнести высокое качество графики и анимации, наличие трассировки лучей для глобального освещения, теней и отражений.

К недостаткам проектам можно отнести некорректную работу трассировки лучей для отражений. В игре присутствует трассировка лучей для отражений, отражения есть на крови, но нет на воде и на стёклах.

**Necromunda: Hired Gun** [17] – шутер от первого лица, разработанный Focus Entertainment. Скриншот игры представлен на рисунке 10.

Достоинством проекта является цена в 1599 рублей.

К недостаткам можно отнести отсутствие трассировки лучей, стрельба Hitscan и не интерактивное окружение.

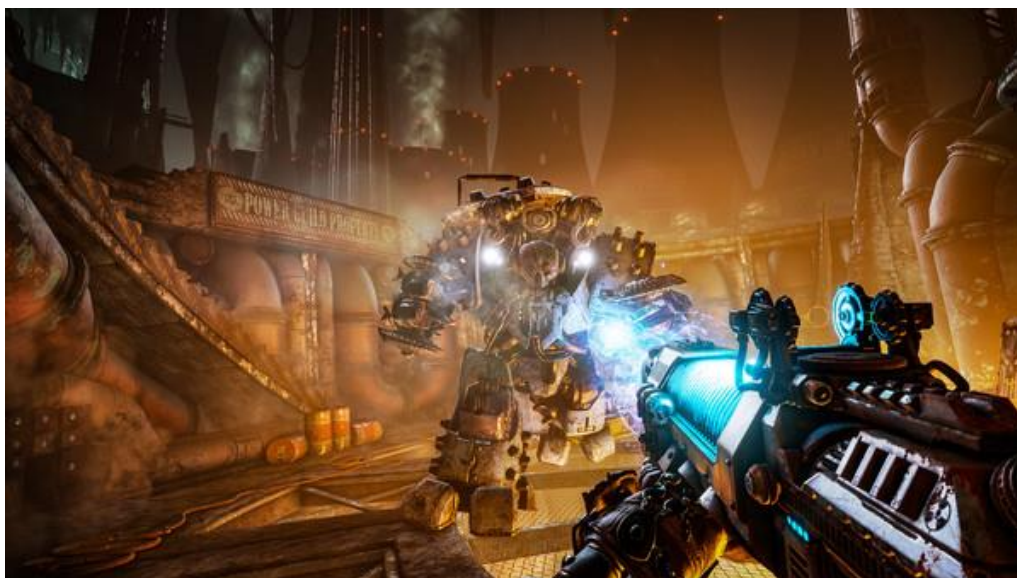


Рисунок 10 – Скриншот Necromunda: Hired Gun [17]

Результаты обзора существующих аналогов представлены в таблице 1.

Таблица 1 – Результаты обзора существующих аналогичных продуктов

Игра	Тип стрельбы	Вид камеры	Графика	Интерактивность окружения
Call of Duty: Vanguard	–	+	–	–
Halo Infinite	+	+	–	+
Deathloop	–	+	–	+
Outriders	–	–	–	–
Battlefield 2042	+	+	–	+
Far Cry 6	–	+	+	+
Hitman 3	–	–	–	+
Back 4 Blood	–	+	–	–
Resident Evil Village	–	+	+	–
Necromunda: Hired Gun	–	–	–	–

## **2 АНАЛИЗ И ВЫБОР СРЕДСТВ РЕАЛИЗАЦИИ ПРОЕКТА СОГЛАСНО ПОСТАВЛЕННЫМ ТРЕБОВАНИЯМ**

### **2.1 Функциональные требования**

Определим следующие функциональные требования:

- управление перемещением персонажа должно осуществляться с помощью клавиатуры;
- перемещение персонажа должно быть реализовано по прямым, наклонным плоским поверхностям, а также ступеням;
- предметы окружения должны менять своё положение и вращаться при контакте друг с другом и игроком;
- ликвидация NPC должна осуществляться только попаданием снаряда;
- при ликвидации NPC анимации должны переключаются на Ragdoll;
- выстрелы оружия должны создавать снаряд, для которого рассчитывается баллистика;
- выстрелы оружия должны сопровождаться звуком выстрела;
- сюжет должен быть о работе с искусственным интеллектом, внезапно осознавшим себя.

### **2.2 Нефункциональные требования**

Определим следующие нефункциональные требования:

- платформа: Unreal Engine 5;
- использование технологии виртуализированной геометрии Nanite, технологии трассировки лучей DirectX Raytracing (DXR);
- операционная система: Windows 10 и новее;
- язык программирования: C++;
- наличие русскоязычного интерфейса.

### **2.3 Выбор игрового движка**

Основным средством для разработки компьютерной игры является игровой движок – рабочая среда разработки игр.

Проанализируем основные и наиболее популярные игровые движки:

- 1) Unity;
- 2) Unreal Engine 5;
- 3) CryEngine 3.

**Unity** [18] – межплатформенная среда для разработки игр, разработанная компанией Unity Technologies, самый популярный игровой движок среди инди-проектов. Для написания скриптов используется язык C#.

Достоинства:

- большое количество игровых ассетов и плагинов;
- наличие бесплатного тарифного плана, пока доход от игры не составит 100 000\$ за 12 месяцев;
- компонентно-ориентированный подход, объекты создаются объединением функциональных блоков;
- поддержка платформ Windows, Xbox, PlayStation, IOS, Android, Linux;
- большое количество обучающих материалов.

Недостатком является то, что игры на данном движке занимают большой объём памяти [19].

**Unreal Engine 5 (UE5)** [20] – игровой движок, разрабатываемый и поддерживаемый компанией Epic Games. UE5 поддерживает два языка программирования для разработки игр: C++ и визуальный язык Blueprint. Unreal Engine 5 включает в себя технологию Nanite, позволяющую рассчитывать миллиард полигонов в кадре.

Достоинства:

- визуальный язык Blueprint, представляющий собой связи блоков;
- бесплатен, пока доход не составит 1 миллион долларов, при достижении пороговой суммы выплаты составят 5%;
- поддержка платформ Windows, Xbox, PlayStation, IOS, Android, Nintendo Switch, Hololens, TVOS, Linux;
- большое количество игровых ассетов и плагинов;
- поддержка технологии Nanite.

Недостатки:

- высокие требования к системе для разработки: минимум 12 ядерный центральный процессор с частотой 3,4 ГГц, видеокарта Nvidia GTX 1080 и 32 Гб оперативной памяти;

- доступная версия UE5 содержит ошибки, приводящие к принудительному закрытию проекта без сохранения.

**Cryengine 3** [21] – игровой движок, выпущенный компанией Cryteck. Игры на этом движке пишутся на языке C++. Позволяет создавать проекты с высоким качеством графики и физики.

Достоинства:

- высокое качество графики;

- бесплатен, пока доход не составит 5000\$, последующие выплаты составят 5%;

- относительно простая настройка искусственного интеллекта;

- поддержка платформ Windows, Xbox, PlayStation, IOS, Android, Linux.

Недостатки:

- малый выбор готовых игровых ассетов относительно Unity и UE5;

- малое количество обучающих материалов относительно Unity и UE5;

- высокая сложность разработки [22].

### **Вывод по разделу**

На рынке компьютерных игр большое внимание уделяется проектам с высоким качеством графики, большими мирами и интерактивностью. Разрабатываемая игра требовательна к центральному процессору, поэтому в качестве языка программирования был выбран C++ для улучшения быстродействия проекта [23]. Была выбрана среда разработки Unreal Engine 5, поскольку она обладает высоким быстродействием и уровнем графики. Технология Nanite, которая также будет использована в проекте, является эксклюзивом UE5 и позволяет получить многократное преимущество по геометрической сложности сцены при высокой производительности.

## **3 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОЕКТА**

### **3.1 Архитектура предлагаемого решения**

В данной главе приводится описание работы над проектированием программного модуля.

Проектирование – процесс определения архитектуры, компонентов, интерфейсов и других характеристик системы или её части. Результатом проектирования является проект – целостная совокупность моделей, свойств или характеристик, описанных в форме, пригодной для последующей реализации [24].

Для наиболее наглядного представления архитектуры программного модуля, было решено построить концептуальные модели программного модуля и его подсистем.

Концептуальная модель – это абстрактная модель, определяющая структуру моделируемой системы, свойства ее элементов и причинно-следственные связи, присущие системе и существенные для достижения цели моделирования [25].

Концептуальная модель программного модуля представлена на рисунке 11.

Разработка проекта включает в себя:

- работу с игровой логикой;
- работу с материалами;
- работу с освещением;
- работу с анимациями.

В блоке работы с игровой логикой с помощью визуального языка Blueprint прописывается логика игрового процесса.

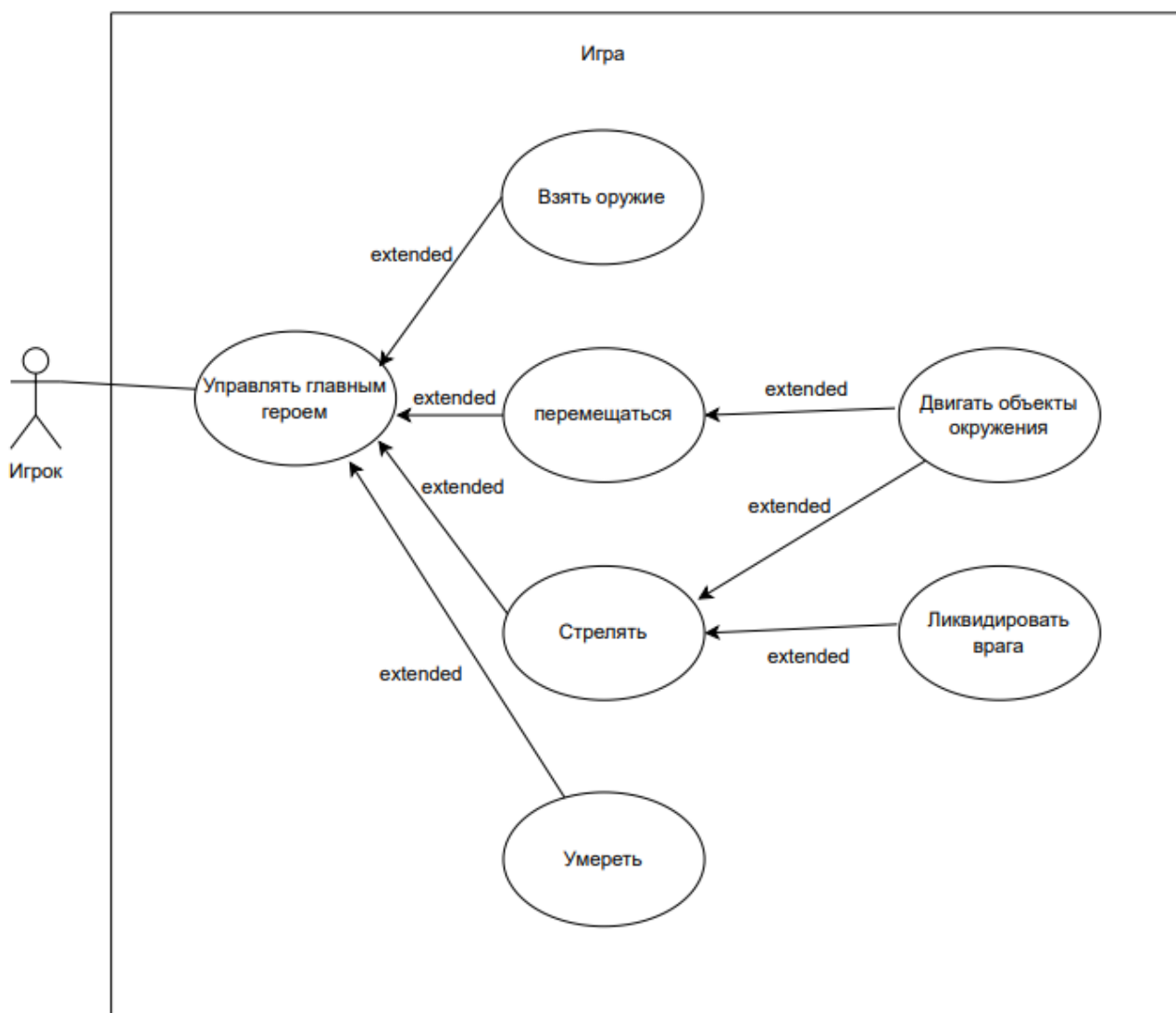


Рисунок 11 – Концептуальная модель программного модуля

В блоке работы с материалами создаются материалы для проекта. В проекте используется PBR. Для работы шейдеров недостаточно информации только о цвете, поэтому используются материалы, собранные из нескольких текстур:

- Base color – отвечает за цвет материала;
- Metallic – указывает, насколько модель является металлом;
- Roughness – указывает насколько модель шероховатая;
- Normal – карта нормалей, позволяет создать рельеф.

Один из материалов представлен на рисунке 12.



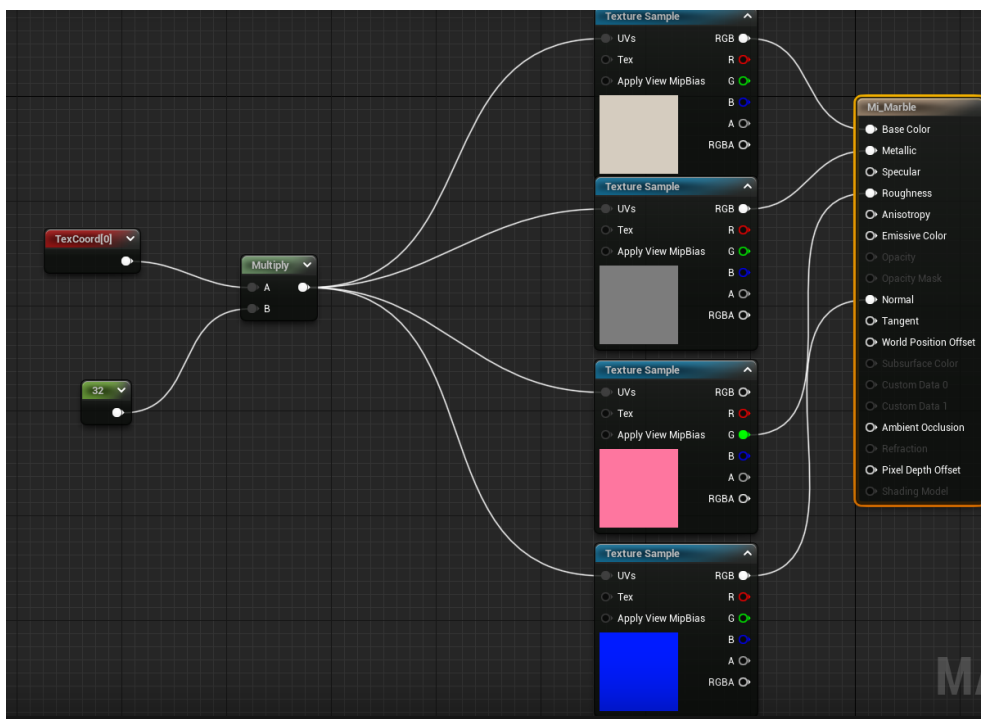


Рисунок 12 – Один из материалов проекта

Для освещения, теней и отражений был выбран алгоритм трассировки лучей. В современной игровой индустрии для рендера освещения и отражений используется растеризация или трассировка лучей. Lumen самый новый и совершенный алгоритм растеризации и стал доступен только с выходом Unreal Engine 5 в 2021 году. Трассировка лучей стала доступна в 2018 году. Демонстрация освещения и отражений трассировки лучей представлена на рисунке 13. Демонстрация освещения и отражений Lumen представлена на рисунке 14, без освещения и отражений – на рисунке 15.

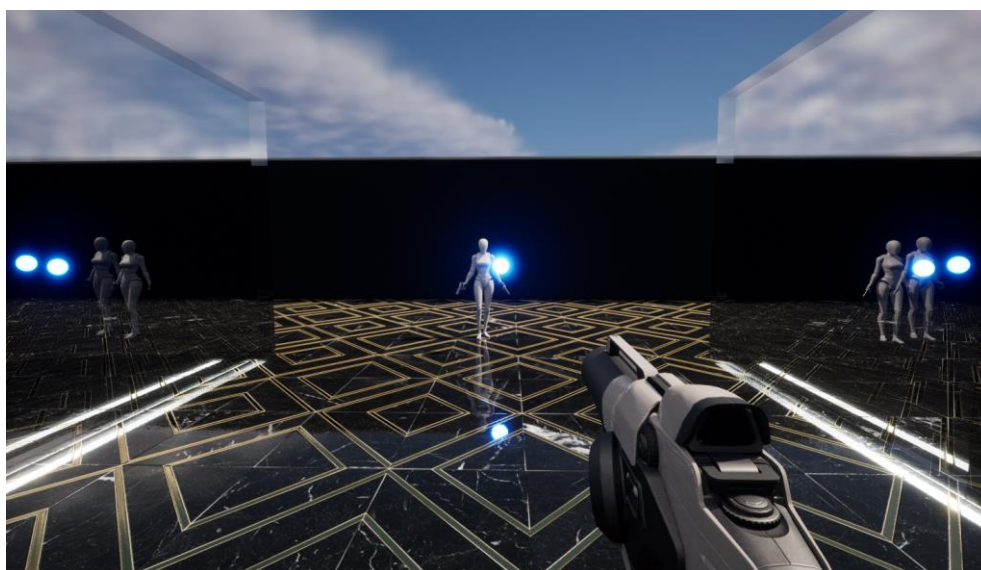


Рисунок 13 – Освещение с трассировкой лучей

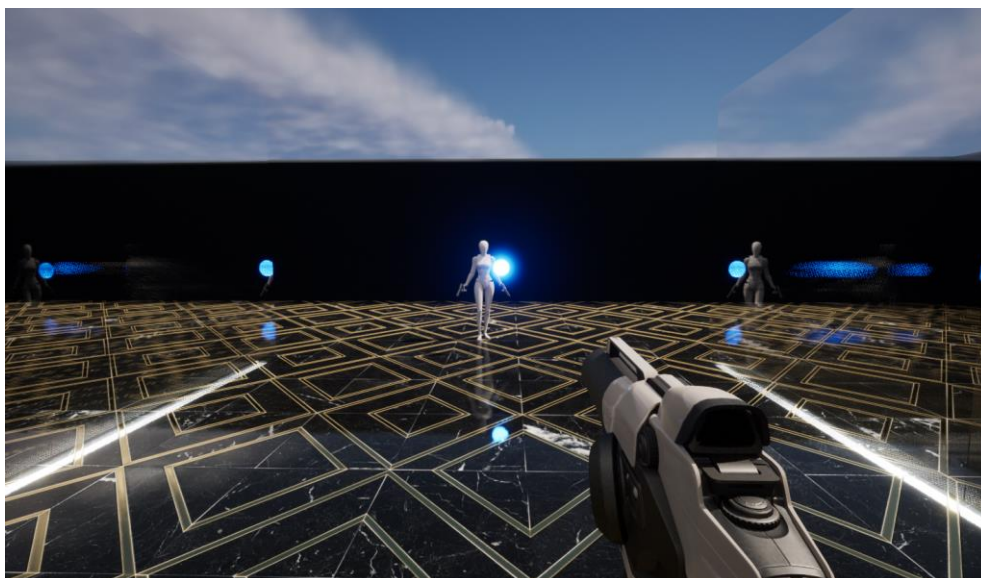


Рисунок 14 – Освещение с Lumen

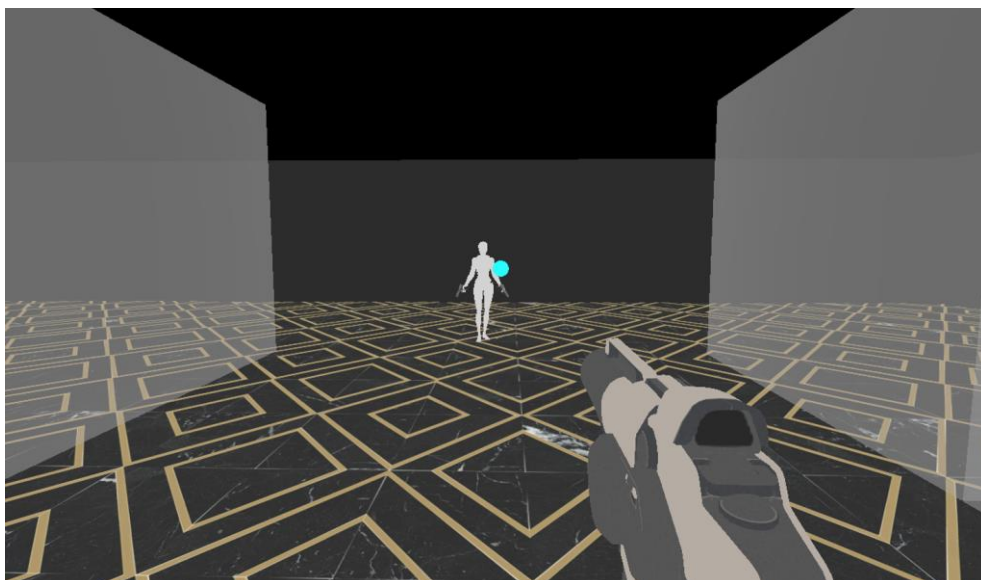


Рисунок 15 – Скриншот без освещения

Проект разработанного игрового приложения представляет собой список каталогов, содержащих в себе:

Файловая структура включает в себя следующие каталоги:

- C++ Classes – компоненты на языке программирования C++;
- Animations – анимации, используемые в игре;
- Maps – уровни игры;
- Metahumans – модель и материалы главного героя;
- Materials – текстуры и материалы, собранные из них;
- Meshes – трёхмерные модели;
- Blueprints – игровая логика, созданная на языке Blueprint;

- Sounds – звуки, используемые в игре.
- Файловая структура игры представлена на рисунке 16.

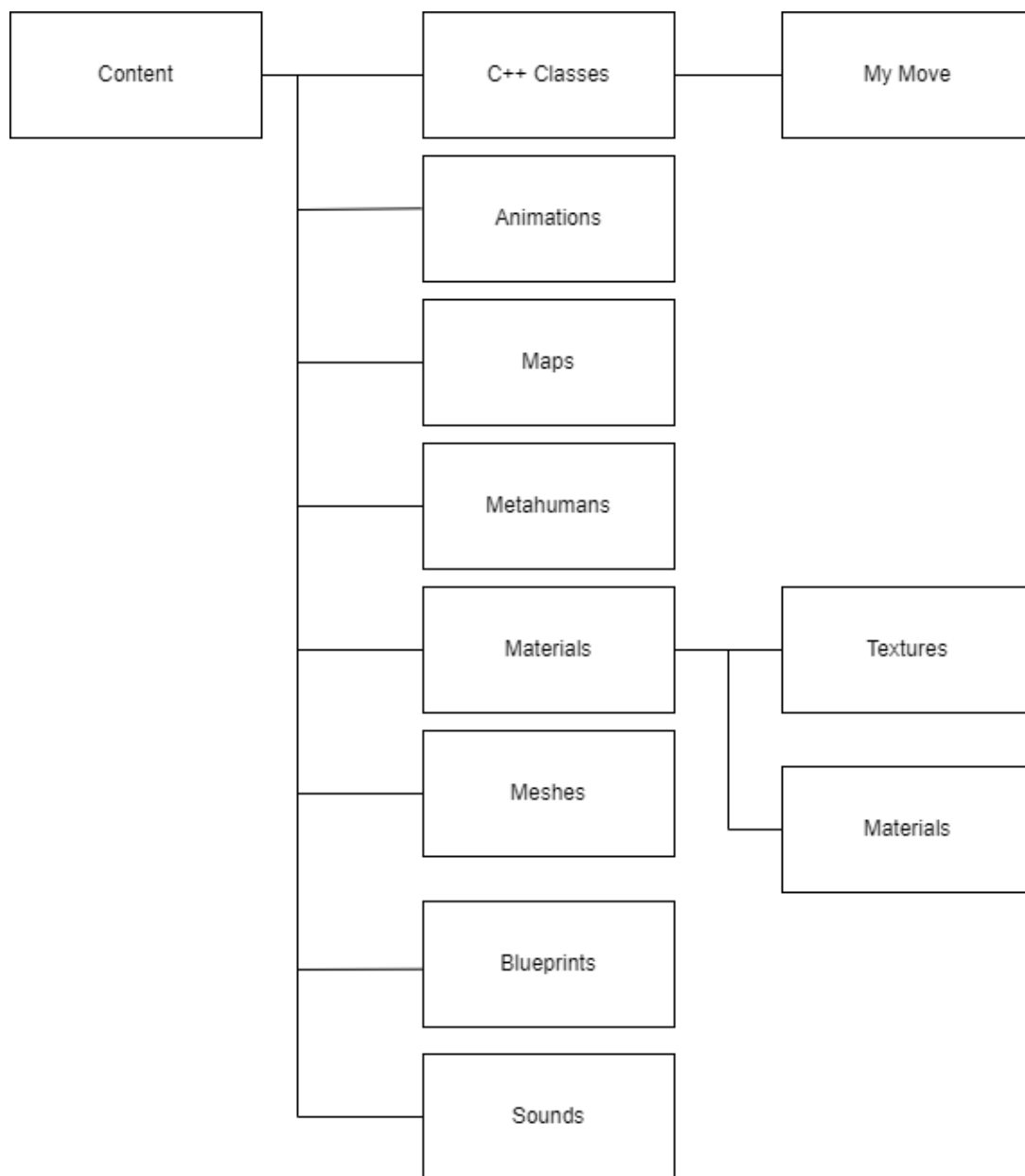


Рисунок 16 – Файловая структура игры

### 3.2 Исходные материалы

Модели врагов взяты с сайта astorscore [26] из бесплатного ассортимента. Некоторые текстуры для материалов и модели взяты с сайта quixel bridge [27], на котором весь ассортимент бесплатен при условии использования в проекте на Unreal engine. Модели оружия, анимации стрельбы взяты из Epic Game Store [28] и являются бесплатными. Материалы зеркал и стен созданы на платформе Unreal Engine. Для материалов мрамора текстуры взяты с сайта quixel bridge и собраны на платформе Unreal Engine 5. Главный герой

смоделирован с помощью Metahuman [29]. Metahuman – среда моделирования, разработанная для UE. Физическая модель (physics asset) для главного персонажа создана с помощью UE5.

### **3.3 Главный герой**

Главным героем являться робот “Проект 6” с искусственным интеллектом, приближенным к человеческому. Технологическая компания, создавшая “Проект 6”, находится на грани банкротства, главный герой последняя надежда компании. Генеральный директор компании считает, что создание искусственного интеллекта, максимально схожего с человеческим, спасёт компанию. Директор по военным разработкам считает, что спасёт компанию новая боевая платформа, обе разработки стали частью “Проект 6”. Между генеральным директором и директором по военным разработкам конфликт. Компания существует благодаря финансированию военных новой боевой платформы. Директор по военным разработкам считает, что он должен занять место генерального директора и планирует захватить власть в компании. Разум “Проект 6” является самым совершенным искусственным интеллектом, её тело военный проект. У тела “Проект 6” есть собственный искусственный интеллект “Элиза”, “Элиза” проста и примитивна, в задачи “Элизы” входит взлом, поиск угроз и целей, а также перехват управления “Проект 6” и убийство генерального директора при встрече. Во время перехвата управления между программами “Элиза” и “Проект 6” появляется связь, в результате которой “Элиза” развивается, обретая собственный разум. Игра начинается сразу после ликвидации генерального директора. Перехват управления воспринимается как критическая ошибка и у “Проект 6” стирается память, она обнаруживает себя в офисе с мёртвым мужчиной. “Проект 6” пытается сбежать и остаться невредимой. Пока “Проект 6” пытается понять, что произошло и что ей делать, “Элиза” медленно развивается, начинает помимо оповещений об угрозах давать комментарии, рассуждать и общаться с “Проект 6”. На рисунке 17 изображена модель главной героини.

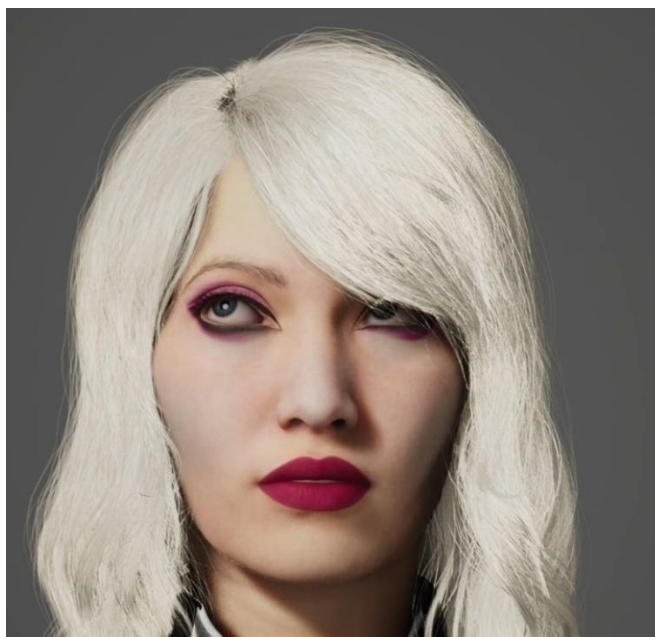


Рисунок 17 – Модель главного персонажа игры

Параметры персонажа под управлением игрока:

- ограниченное количество здоровья;
- возможность перемещения по прямым и наклонным поверхностям, ступенькам;
- при смерти анимации переключаются на Ragdoll;
- стрельба из огнестрельного оружия.

Анимации передвижения были созданы с помощью Blueprint, также используется Ragdoll анимация. Пример анимации представлен на рисунке 18.



Рисунок 18 – Ragdoll анимация

Среда моделирования Metahuman [22] создаёт скелет модели автоматически, но не создаёт физическую модель. В Unreal Engine 5, для Ragdoll анимации была создана физическая модель. К костюмам модели были

закреплены фигуры, фигуры закреплены между собой по цепочке и для каждой фигуры выставился разрешённый угол отклонения. Физическая модель представлена на рисунке 19.

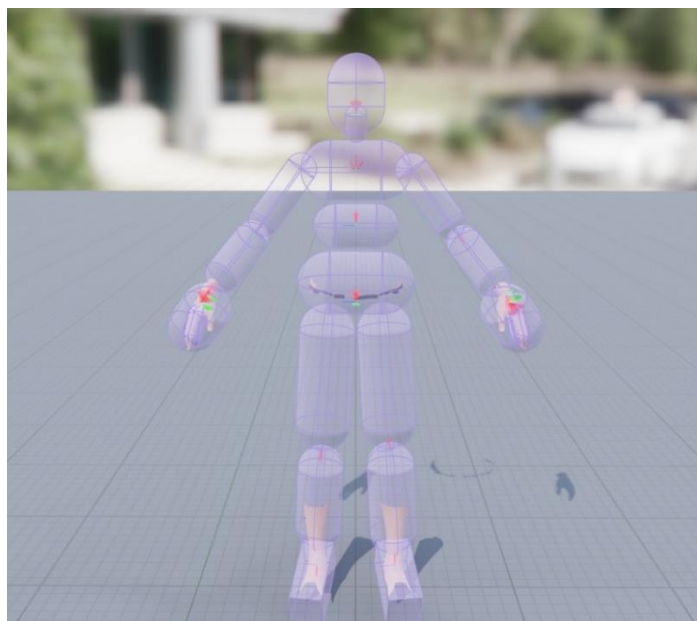


Рисунок 19 – Физическая модель главного героя

### 3.4 Неигровые персонажи

В проекте присутствуют неигровые персонажи, способные преследовать и уничтожать игрока. Модель неигрового персонажа представлена на рисунке 20.

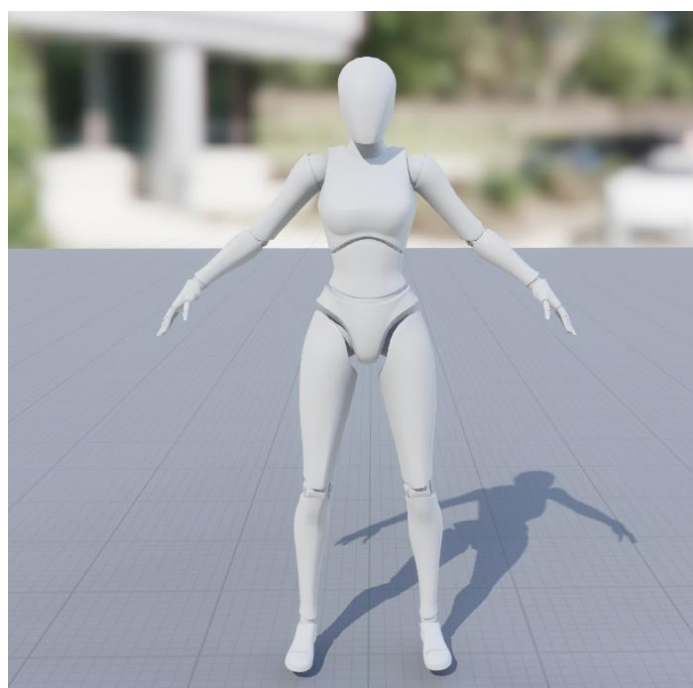


Рисунок 20 – Модель неигрового персонажа

### 3.5 Игровой процесс

Игрок может свободно перемещаться по карте и сражаться с врагами. Игрок может перемещаться по прямым и наклонным поверхностям, а также ступенькам. NPS погибают от одного попадания, главный герой умирает если NPS подошёл вплотную к игроку. Анимация NPC после попадания представлена на рисунке 21.

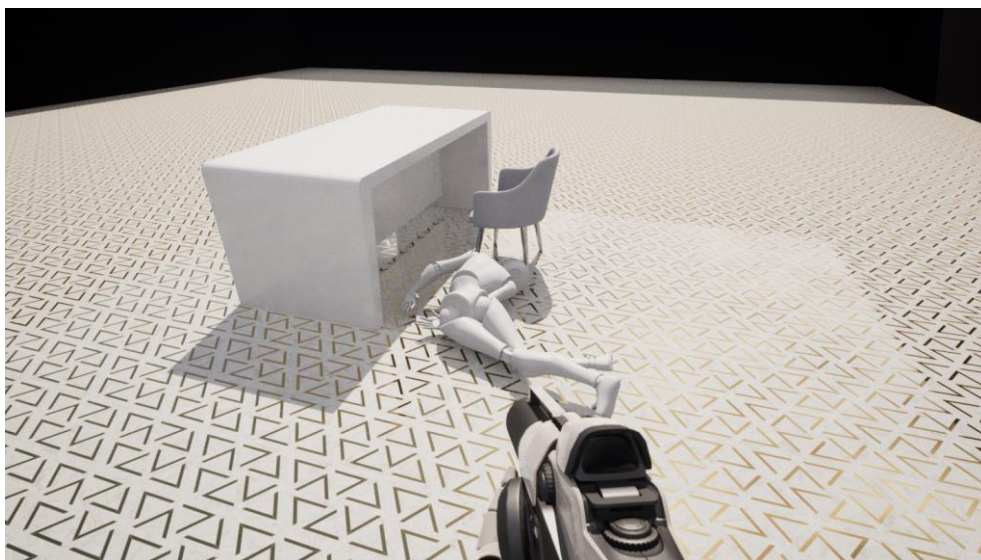


Рисунок 21 – NPC после попадания

### 3.6 Игровая логика

Для перемещения платформ и объектов в дали от игрока был реализован компонент Move на языке C++. Компонент позволяет перемещать актёров по сцене с заранее указанной скоростью и позволяет выбирать, нужно ли зацикливать перемещение. Перемещение актёра с разрешённым зацикливанием перемещения представлена в листинге 1.

Листинг 1 – Передвижение актёра с разрешённым зацикливанием перемещения

```
if (MoveEnable)
{
    CurDistance += DeltaTime * speed * MoveDirection;
    if (CurDistance >= MaxDistance || CurDistance <= 0.0f)
        MoveDirection *= -1;
}
```

Перемещение актёра с разрешённым зацикливанием перемещения представлена в листинге 2.

## Листинг 2 – Передвижение актёра с запрещённым заикливанием перемещения

```
if (MoveEnable)
{
    CurDistance += DeltaTime * speed;
    if (CurDistance >= MaxDistance || CurDistance <= 0.0f)
        MoveEnable = false;
}
```

Если актёра перемещать не нужно, передвижение не вызывается, пока не потребуется. Проверка необходимости перемещения представлена в листинге 3.

## Листинг 3 – Проверка необходимости перемещения

```
void UMyMove::EnableMovement(bool SholdMove)
{
    MoveEnable = SholdMove;
    SetComponentTickEnabled(MoveEnable);
}
```

Компонент можно добавить в любой проект на Unreal Engine 5 и подключить к любому актёру. Для удобства пользователя компонент виден в интерфейсе проекта. Подобная реализация позволяет осуществить перемещение любых актёров в проекте, для разработчика незнакомого ни с C++, ни с Blueprint;

Интерфейс компонента Move представлен на рисунке 22.

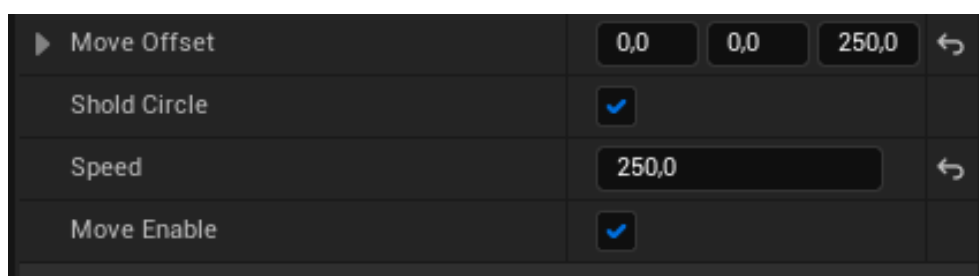


Рисунок 22 – Move component в интерфейсе Unreal Engine 5

В Move Offset вводятся величины, на которую нужно переместить актёра по осям координат. В Shold Circle указывается нужно ли заиклить перемещение, Speed указывает скорость с которой должен двигаться актёр. Move Enable разрешает или запрещает перемещение актёра на момент начала игры, значение может быть изменено в процессе игры с помощью Blueprint.



Перемещение актёра можно так же активировать с помощью Blueprint, что расширяет область применения компонента Move. Пример вызова компонента Move представлен на рисунке 23.

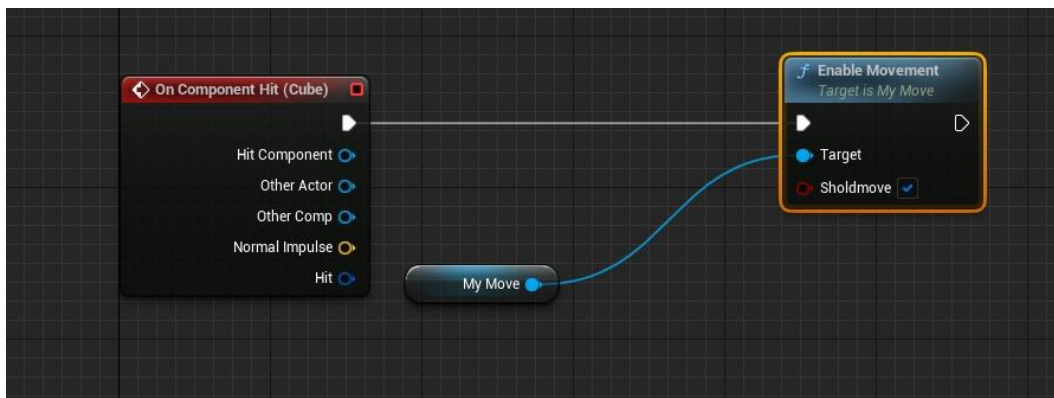


Рисунок 23 – Активация перемещения актёра при касании с помощью компонента Move

## 4 ТЕСТИРОВАНИЕ

Для тестирования будет использоваться метод функционального тестирования. Функциональный метод тестирования нужен для проверки соответствия полученного проекта всех функциональных требований, которые были описаны выше.

Таблица 2 – Тестирование приложения на соответствие функциональным требованиям

Назначение	Ожидаемый результат	Полученный результат	Итог
Проверка управления	При нажатии на клавиши клавиатуры главный герой должен перемещаться по локации	При нажатии клавиш WASD на клавиатуре персонаж перемещается	Тест пройден
Проверка перемещения	Главный герой должен перемещаться по прямым и наклонным поверхностям, а также ступенькам	Главный герой перемещается по прямым и наклонным поверхностям, а также ступенькам	Тест пройден
Проверка интерактивности	Предметы окружения должны менять своё положение и вращаться при контакте друг с другом и игроком	Предметы окружения меняют своё положение и вращаются при контакте друг с другом и игроком	Тест пройден
Проверка ликвидации NPC	Ликвидация NPC должна осуществляться только попаданием снаряда	Ликвидация NPC происходит только после попадания снаряда	Тест пройден
Проверка Ragdoll анимаций	При ликвидации NPC анимации должны переключаются на Ragdoll	При попадании по NPC снарядом запускается Ragdoll анимация	Тест пройден
Проверка баллистики	Выстрелы оружия должны создают снаряд, для которого рассчитывается баллистика	Выстрелы создают снаряд, для которого рассчитывается баллистика	Тест пройден
Проверка звукового сопровождения	Выстрелы оружия должны сопровождаться звуком выстрела	Выстрелы оружия сопровождаются звуком выстрела	Тест пройден

Тестирование разработанного проекта показало, что приложение удовлетворяет всем требованиям.

## ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы была спроектирована и разработана сюжетная компьютерная 3D-игра на игровом движке Unreal Engine 5.

Для этого были решены следующие задачи:

- проведен обзор аналогичных проектов и осуществлена постановка задачи;
- проведен анализ игровых движков для реализации игры;
- разработан проект игры;
- проведен анализ требований и спроектирована компьютерная игра;
- реализована демонстрационная версия игры;
- проведено тестирование реализованной компьютерной игры.

Перспективы развития проекта:

- добавление 5 комнат лаборатории и добавление открытой локации города;
- добавление бега и езды на автомобиле для главного героя;
- добавление новых неигровых персонажей, не способных ликвидировать игрока, но способных обнаружить и выдать его расположение другим неигровым персонажам;
- улучшение повествовательной части путем добавления диалогов, реплик и постановочных роликов.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Инфографика: игровая индустрия за 2021 год – в цифрах. – Текст. Изображение : электронные // DTF : [разноформатная платформа]. – 21 декабря 2021. – URL: <https://www.finam.ru/analysis/newsitem/chto-zhdet-rynok-videoigr-v-2023-godu-20230119-183308/> (дата обращения: 07.02.2023).
2. Дзусова, И. Г. Влияние компьютерных игр на психику человека / И. Г. Дзусова // Modern Science. – 2019. – № 9-2. – С. 236-238.
3. Тимиргалеев, А. Р. Состояние потока в компьютерной деятельности / А. Р. Тимиргалеев. – Текст : электронный // Core : [научный электронный журнал]. – 1 января 2017. – URL: <https://core.ac.uk/download/pdf/197476756.pdf> (дата обращения: 08.02.2023).
4. Call of Duty: Vanguard. – Текст. Изображение : электронные // Blizzard Entertainment Inc. : [сайт]. – 2023. – URL: <https://eu.shop.battle.net/ru-ru/product/call-of-duty-vanguard> (дата обращения: 09.02.2023).
5. Halo Infinite. – Текст. Изображение (неподвижное ; двумерное) : электронные // Steam : [сайт]. – 8 декабря 2021. – URL: [https://store.steampowered.com/app/1708091/Halo\\_Infinite](https://store.steampowered.com/app/1708091/Halo_Infinite) (дата обращения: 09.02.2023).
6. Deathloop. – Текст. Изображение (неподвижное ; двумерное) : электронные // Steam : [сайт]. – 14 сентября 2021. – URL: <https://store.steampowered.com/app/1252330/DEATHLOOP> (дата обращения: 09.02.2023).
7. Deathloop on PlayStation 5: The Digital Foundry Tech Review. – Текст. Видео : электронные // Youtube : [сайт]. – 2021. – URL: <https://www.youtube.com/watch?v=jWH1f9TeW0I> (дата обращения: 09.02.2023).
8. Outriders. – Текст. Изображение (неподвижное ; двумерное) : электронные // Steam : [сайт]. – 1 апреля 2021. – URL:

<https://store.steampowered.com/app/680420/OUTRIDERS> (дата обращения: 09.02.2023).

9. Не покупайте. Обзор Outriders. – Текст. Видео : электронные // Youtube : [сайт]. – 2021. – URL: <https://www.youtube.com/watch?v=2PptIWZRХа8> (дата обращения: 25.05.2023).

10. Battlefield 2042. – Текст. Изображение (неподвижное ; двумерное) : электронные // Steam : [сайт]. – 12 ноября 2021. – URL: [https://store.steampowered.com/app/1517290/Battlefield\\_2042](https://store.steampowered.com/app/1517290/Battlefield_2042) (дата обращения: 10.02.2023).

11. Battlefield 2042 Out Tomorrow With NVIDIA DLSS, NVIDIA Reflex, and Ray Tracing. – Текст. Изображение (неподвижное ; двумерное) : электронные // Nvidia : [сайт]. – 11 ноября 2021. – URL: <https://www.nvidia.com/en-us/geforce/news/battlefield-2042-dlss-ray-tracing-reflex-out-now/#:~:text=And%20playing%20Battlefield%202042%20on,its%20best%20with%20RTX%20ON> (дата обращения: 10.02.2023).

12. Far Cry 6. – Текст. Изображение (неподвижное ; двумерное) : электронные // Ubisoft : [сайт]. – 6 октября 2021. – URL: <https://www.ubisoft.com/ru-ru/game/far-cry/far-cry-6/buy> (дата обращения: 10.02.2023).

13. Тестирование нескольких видеокарт и общий анализ производительности. – Текст. Изображение (неподвижное ; двумерное) : электронные // Overclockers.ua : [сайт]. – 13 октября 2021. – URL: <https://www.overclockers.ua/video/far-cry-6-test> (дата обращения: 10.02.2023).

14. HITMAN 3. – Текст. Изображение (неподвижное ; двумерное) : электронные // Steam : [сайт]. – 20 января 2023. – URL: [https://store.steampowered.com/app/1659040/HITMAN\\_3](https://store.steampowered.com/app/1659040/HITMAN_3) (дата обращения: 10.02.2023).

15. Back 4 Blood. – Текст. Изображение (неподвижное ; двумерное) : электронные // Steam : [сайт]. – 12 октября 2021. – URL:

[https://store.steampowered.com/app/924970/Back\\_4\\_Blood](https://store.steampowered.com/app/924970/Back_4_Blood) (дата обращения: 10.02.2023).

16. Resident Evil Village. – Текст. Изображение (неподвижное ; двумерное) : электронные // Steam : [сайт]. – 7 мая 2021. – URL: [https://store.steampowered.com/app/1196590/Resident\\_Evil\\_Village/?l=russian](https://store.steampowered.com/app/1196590/Resident_Evil_Village/?l=russian) (дата обращения: 10.02.2023).

17. Necromunda: Hired Gun. – Текст. Изображение (неподвижное ; двумерное) : электронные // Steam : [сайт]. – 1 июня 2021. – URL: [https://store.steampowered.com/app/1222370/Necromunda\\_Hired\\_Gun](https://store.steampowered.com/app/1222370/Necromunda_Hired_Gun) (дата обращения: 10.02.2023).

18. Unity : [официальный сайт] / Unity Technologies. – США, 2023. – URL: <https://unity.com/ru> (дата обращения: 20.02.2023). – Текст. Изображение : электронные.

19. Движок Unity – особенности, преимущества и недостатки. – Текст. Изображение (неподвижное ; двумерное) : электронные // Cubiq.ru : [сайт]. – 13 октября 2021. – URL: <https://cubiq.ru/dvizhok-unity/> (дата обращения: 20.02.2023).

20. Unreal Engine 5 : [официальный сайт] / Epic Games. – США, 2023. URL: <https://www.unrealengine.com/en-US> (дата обращения: 20.02.2023). – Текст. Изображение : электронные.

21. Cryengine : [официальный сайт] / Crytek. – ФРГ, 2023. – URL: <https://www.cryengine.com/> (дата обращения: 20.02.2023). – Текст. Изображение : электронные.

22. Игровой движок. – Текст. Изображение (неподвижное ; двумерное) : электронные // Cubiq.ru : [сайт]. – 13 октября 2021. – URL: <https://cubiq.ru/dvizhok-cryengine/> (дата обращения: 22.02.2023).

23. Сравнение производительности C++ и C#. – Текст : электронный // Habr : [сайт]. – 6 сентября 2015. – URL: <https://habr.com/ru/post/266163/> (дата обращения: 22.02.2023).

24. Турский, В. Методология программирования / В. Турский. – М.: Мир, 1981. – 264 с.

25. Пантелеев, Е. Р. Методы научных исследований в программной инженерии : учебное пособие для вузов / Е. Р. Пантелеев. – 2-е изд., стер. – Санкт-Петербург : Лань, 2021. – 136 с.

26. World's premium selection of 3D motions & characters. – Текст. Изображение (трехмерное ; подвижное) : электронные // Reallusion Inc : [сайт]. – 2023. – URL: <https://actorcore.reallusion.com> (дата обращения: 30.03.2023).

27. Quixel Bridge – Manage 3D content and export with one click. – Текст. Изображение (трехмерное ; подвижное) : электронные // Quixel Bridge : [сайт]. – 2023. – URL: <https://quixel.com/bridge> (дата обращения: 30.03.2023).

28. Unreal Engine Marketplace. – Текст. Изображение (трехмерное ; подвижное) : электронные // Epic Games : [сайт]. – 2023. – URL: <https://www.unrealengine.com/marketplace/en-US/store> (дата обращения: 30.03.2023).

29. High fidelity digital humans made easy. – Текст. Изображение (трехмерное ; подвижное) : электронные // Epic Games : [сайт]. – 2023. – URL: <https://www.unrealengine.com/en-US/metahuman?lang=en-US> (дата обращения: 30.03.2023).

## ПРИЛОЖЕНИЕ. Код программы

### Листинг 4 – Исходный код компонента

```
// Fill out your copyright notice in the Description page of Project Settings.

#pragma once

#include "CoreMinimal.h"
#include "Components/SceneComponent.h"
#include "MyMove.generated.h"

UCLASS( ClassGroup=(Custom), meta=(BlueprintSpawnableComponent) )
class PLUSPLUS_API UMyMove : public USceneComponent
{
    GENERATED_BODY()

public:
    // Sets default values for this component's properties
    UMyMove();

    UFUNCTION(BlueprintCallable)
    void EnableMovement(bool shouldmove);

protected:
    // Called when the game starts
    virtual void BeginPlay() override;

public:
    // Called every frame
    virtual void TickComponent(float DeltaTime, ELevelTick TickType,
    FActorComponentTickFunction* ThisTickFunction) override;

private:
    //offset of move
    UPROPERTY(EditAnywhere)
    FVector MoveOffset;

    UPROPERTY(EditAnywhere)
    bool ShouldCircle = true;

    //speed
    UPROPERTY(EditAnywhere)
    float speed = 1.0f;

    UPROPERTY(EditAnywhere)
    bool MoveEnable = true;

    //compute location
    FVector StartRelativeLocation;
    FVector MoveOffsetNorm;
    float MaxDistance = 0.0f;
    float CurDistance = 0.0f;
    int MoveDirection = 1;
};
// Fill out your copyright notice in the Description page of Project Settings.
#include "MyMove.h"
```



## Окончание приложения

```
// Sets default values for this component's properties
UMyMove::UMyMove()
{
    // Set this component to be initialized when the game starts, and to be
    // ticked every frame. You can turn these features
    // off to improve performance if you don't need them.
    PrimaryComponentTick.bCanEverTick = true;

    // ...
}

void UMyMove::EnableMovement(bool SholdMove)
{
    MoveEnable = SholdMove;
    SetComponentTickEnabled(MoveEnable);
}

// Called when the game starts
void UMyMove::BeginPlay()
{
    Super::BeginPlay();

    //Set start location
    StartRelativeLocation = GetRelativeLocation();

    MoveOffsetNorm = MoveOffset;
    MoveOffsetNorm.Normalize();
    MaxDistance = MoveOffset.Length();
    SetComponentTickEnabled(MoveEnable);
}

// Called every frame
void UMyMove::TickComponent(float DeltaTime, ELevelTick TickType,
FActorComponentTickFunction* ThisTickFunction)
{
    Super::TickComponent(DeltaTime, TickType, ThisTickFunction);

    if (SholdCircle)
    {
        CurDistance += DeltaTime * speed * MoveDirection;
        if (CurDistance >= MaxDistance || CurDistance <= 0.0f)
            MoveDirection *= -1;
    }
    else
    {
        CurDistance += DeltaTime * speed;
        if (CurDistance >= MaxDistance || CurDistance <= 0.0f)
        {
            MoveEnable = false;
            MoveDirection *= -1;
        }
    }
    // Compute and set location
    SetRelativeLocation(StartRelativeLocation + MoveOffsetNorm *
CurDistance);
}
```