**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**«БАРАНОВИЧСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**Факультет** Инженерный\_

**Кафедра** Информационных технологий и физико-математических дисциплин

**Дата регистрации работы в деканате \_\_\_\_\_\_\_\_\_**

**Дата регистрации работы на кафедре \_\_\_\_\_\_\_\_\_**

**Отметка о допуске к защите \_\_\_\_\_\_\_\_\_**

**Оценка за защиту \_\_\_\_\_\_\_\_\_**

**КУРСОВОЙ ПРОЕКТ**

**по дисциплине:** «Компьютерные системы конечноэлементных расчетов»

**Тема: «**Создание виртуальной лаборатории информационных и робототизированных систем в Unreal Engine**»**

|  |
| --- |
| **Исполнитель:** |
| **Студент 3 курса, группа ИСТ-31** |
| **студент (факультет, курс, группа)** |
| **Веракса Э. А.** |
| **фамилия, имя, отчество** |
|  |
|  |
| **Руководитель:** |
| **ст. преподаватель** |
| **ученое звание, ученая степень, должность** |
| **Шах А. В.** |
| **фамилия, имя, отчество** |

**Барановичи 2024МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**«БАРАНОВИЧСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Факультет\_\_\_\_ инженерный \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра\_информационных технологий и физико-математических дисциплин\_

УТВЕРЖДАЮ

Руководитель секции ИСТ

\_\_\_\_\_\_\_\_\_\_\_Шах А. В.\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (инициалы и фамилия)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024\_ г.

**ЗАДАНИЕ ПО КУРСОВОМУ ПРОЕКТИРОВАНИЮ**

Студенту

\_\_\_\_\_\_\_\_Вераксе Эдуарду Александровичу\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(инициалы и фамилия)

Тема проекта (работы):

«\_\_\_Создание виртуальной лаборатории информационных и робототизированных систем в Unreal Engine\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

Дата выдачи задания – «\_22\_» \_\_\_\_января\_\_\_\_\_\_\_ 2024 г.

Срок сдачи законченного проекта (работы) – «\_02\_» \_\_мая\_\_\_\_\_\_ 2024 г.

Исходные данные к проекту (работе) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Методические указания и перечень заданий содержатся в электронном виде*

*\_на сервере секции информационных систем и технологий по адресу:\_\_\_\_\_*

*\_M:\ методички \ компьютерные системы конечноэлементных расчетов \ курсовые \*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Содержание пояснительной записка (Перечень подлежащих разработке

вопросов) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_Введение\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_1 Теоретический раздел \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_1.1 Постановка задачи\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_1.2 Описание использованных программных сред\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_2 Практический раздел \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_2.1 Построение и описание диаграмм проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_2.2 Проектирование приложения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_2.3 Тестирование приложения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_2.4 Руководство пользователя\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_Заключение\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перечень графического материала (с указанием обязательных чертежей и др.) \_\_\_\_эпюры анализа \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Консультанты по проекту (при их наличии) с указанием относящихся к ним

разделов проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_ст. преподаватель Шах А. В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Календарный график работы над проектом (работой) на весь период

проектирования (выполнения) с указанием сроков выполнения отдельных

этапов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_22.01.24-15.02.24\_\_\_Изучение теоретического материала\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_16.02.24-01.03.24\_\_\_Описнаие архитектуры проекта\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_02.03.22-09.04.22\_\_\_Создание программного продукта \_\_\_\_\_

\_\_\_\_\_10.04.24-20.04.24\_\_\_Тестирование приложения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_21.04.24-02.05.24\_\_\_Оформление пояснительной записки\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель \_\_\_22.01.24\_\_\_\_\_ \_\_\_\_\_Шах А. В. \_\_\_\_\_\_

(подпись, дата) (инициалы и фамилия, уч. степень, звание)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата) (инициалы и фамилия студента)

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**«БАРАНОВИЧСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

РЕЦЕНЗИЯ

на курсовой проект

(регистрационный №\_\_\_\_\_)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студента | | Вераксы Эдуарда Александровича | | | |
|  | | (фамилия, имя, отчество) | | | |
| Факультет | | Инженерный | | | |
| Курс | | 3 | | | |
| Дисциплина | | Компьютерные системы конечноэлементных расчетов | | | |
| Рецензент | | Шах Александр Васильевич | | | |
|  | | (фамилия, имя, отчество) | | | |
| Дата получения к/п для рецензирования | | | |  | |
| Дата возвращения к/п после рецензирования | | | |  | |
| Оценка |  | | Подпись преподавателя-рецензента | |  |
| Текст рецензии: | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

**РЕФЕРАТ**

Курсовой проект: 20 с., 11 рис., 5 источников, 3 прил.

UNREAL ENGINE 5, BLUEPRINT, BLENDER, TWINMOTION, ПРИЛОЖЕНИЕ, ЛАБОРАТОРИЯ.

Объект исследования – моделирование виртуальной лаборатории средствами игрового движка Unreal Engine 5.

Предмет исследования – использование игрового движка Unreal Engine 5 при моделировании виртуальной лаборатории.

Цель проекта – изучение возможностей Unreal Engine при проектировании и визуализации зданий.

Основой для выполнения проекта стала справочная, научная и учебная литература.

При выполнении проекта использовался метод: анализ научно-методической литературы, проектов.

Областью возможного практического применения данной работы является моделирование объектов реального мира на основе игрового движка Unreal Engine 5 совместно с использованием средств моделирования Blender и визуализации Twinmotion.

Автор подтверждает, что приведенный в работе расчетно-аналитический материал правильно и объективно отражает состояние исследуемого процесса, а все заимствованные из литературных и других источников теоретические, методологические и методические положения и концепции сопровождаются ссылками на их авторов.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc165489025)

[1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5](#_Toc165489026)

[1.1 Постановка задачи 5](#_Toc165489027)

[1.2 Описание использованных программных средств 5](#_Toc165489028)

[2 ПРАКТИЧЕСКАЯ ЧАСТЬ 8](#_Toc165489029)

[2.1 Построение и описание диаграмм проекта 8](#_Toc165489030)

[2.2 Проектирование приложения 11](#_Toc165489031)

[2.3 Тестирование приложения 12](#_Toc165489032)

[2.4 Руководство пользователя 14](#_Toc165489033)

[ЗАКЛЮЧЕНИЕ 18](#_Toc165489034)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 19](#_Toc165489035)

[ПРИЛОЖЕНИЯ 20](#_Toc165489036)

# ВВЕДЕНИЕ

В современном мире, где технологии развиваются с невероятной скоростью, создание интерактивных и визуально впечатляющих проектов становится все более доступным. Одним из инструментов, позволяющих реализовать самые разнообразные идеи, является Unreal Engine. Этот движок, разработанный компанией Epic Games, стал неотъемлемой частью индустрии развлечений, образовательных программ и даже в области архитектурного проектирования. Unreal Engine предлагает разработчикам мощные инструменты для создания 3D-графики, анимации, физики и многое другое, что делает его идеальным выбором для реализации самых сложных проектов.

Одним из ключевых преимуществ Unreal Engine является его способность обеспечивать высокую производительность и качество графики, что позволяет создавать проекты, которые могут поражать своим визуальным восприятием. Этот движок поддерживает разработку как для персональных компьютеров, так и для различных платформ, включая мобильные устройства, что делает его универсальным решением для широкого спектра задач. Благодаря своей гибкости и мощным инструментам, Unreal Engine позволяет разработчикам не только реализовывать свои идеи, но и создавать уникальные виртуальные пространства, которые могут быть использованы для обучения, демонстрации проектов, проведения исследований и многого другого.

Создание архитектуры в Unreal Engine открывает новые горизонты для архитекторов и дизайнеров, предоставляя им возможность визуализировать свои проекты в реальном времени. Это позволяет не только лучше понять и представить конечный результат, но и вносить коррективы на ранних этапах разработки, что значительно сокращает время и затраты на проект. Unreal Engine предлагает широкий спектр инструментов для моделирования, текстурирования и освещения, что делает процесс создания архитектурной модели более интуитивным и менее трудоемким.

Unreal Engine 5 (UE5) предлагает ряд значительных преимуществ для проектирования архитектуры, особенно в контексте разработки видеоигр и виртуальной реальности. Вот некоторые из ключевых преимуществ:

* Новейшие Технологии Рендеринга: UE5 использует технологию Nanite, которая позволяет создавать детализированные и высококачественные текстуры и модели без ограничений по размеру. Это значительно упрощает создание сложных архитектурных элементов и окружений.
* Улучшенная Физика: UE5 включает в себя улучшенную систему физики, которая обеспечивает более реалистичное поведение объектов в игровом мире. Это особенно важно для архитектурных проектов, где детализация и реализм являются ключевыми.
* Современные Инструменты и Технологии: UE5 предлагает обновленные инструменты для моделирования, анимации, редактирования сцен и многое другое, что делает процесс разработки более интуитивным и эффективным.
* Поддержка VR и AR: UE5 предлагает мощные инструменты и возможности для разработки виртуальной и дополненной реальности, что открывает новые возможности для создания интерактивных архитектурных проектов.
* Совместимость с Cloud: UE5 поддерживает работу с облачными сервисами, что позволяет разработчикам легко синхронизировать проекты, использовать облачные вычисления для обработки больших объемов данных и обеспечивает гибкость в работе над проектами.
* Большое Сообщество и Документация: UE5 имеет большое и активное сообщество разработчиков, а также обширную документацию и обучающие материалы, что значительно упрощает процесс обучения и решения возникающих проблем.
* Многоплатформенность: UE5 поддерживает разработку игр для множества платформ, включая PC, консоли, мобильные устройства и виртуальную реальность, что делает его универсальным решением для различных проектов [1].

Однако, несмотря на все преимущества, создание архитектуры в Unreal Engine требует от разработчиков определенных навыков и знаний. Для успешной реализации проекта необходимо обладать пониманием принципов работы движка, знать основы 3D-моделирования и программирования, а также уметь эффективно использовать инструменты и функции Unreal Engine. Это может потребовать времени и усилий, но в результате получается возможность создавать проекты, которые могут поражать своим качеством и реалистичностью

1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1 Постановка задачи

Разработка игрового приложения, предназначенного для ознакомления с лабораторией информационных и робототизированных систем.

Программный продукт должен позволять виртуальному аватару пользователя перемащаться по виртуальной лаборатории. Приложение должно запускаться на различных платформах таких как: Windows или Android и поддерживать использование гарнитуры виртуальной реальности.

## 1.2 Описание использованных программных средств

Для моделирования объектов в приложении был использован Blender — программный продукт, предназначенный для создания и редактирования трехмерной графики.

Сам Blender предоставляет пользователю следующий функционал:

* 3D моделирование. Пользователю доступно огромное количество инструментов для создания и редактирования 3D моделей самых разных уровней сложности. Причем моделировать объекты можно при помощи доступных примитивов, полигонов, NURBS-кривых и кривых Безье. Дополнительно предусмотрен функционал для формирования метасфер и управления формой при помощи булевых операций. Не стоит забывать и о технологиях Subdivision Surface и наиболее понятных инструментов создания скульптур. По аналоги с профессиональными сборками, тут предусмотрены модификаторы для изменения формы моделей.
* Создание анимации. Несмотря на то, что софт сам по себе рассчитан на моделирование, анимация представлена тоже неплохо. Можно использовать традиционную скелетную анимацию или риггинг, инверсную кинематику, различные ограничители и многое другое. Все коэффициенты и параметры настраиваются при помощи встроенных инструментов. Дополнительно представлена динамика тел разной твердости и формирование анимации мелких частиц.
* Текстуры. Можно накладывать сразу несколько текстур на один и тот же объект. Есть масса инструментов для текстурирования, в том числе UV-маппинг и функция частичного настраивания. Значительно облегчает работу настройка шейдеров.
* Рисование. Есть много встроенных средств для создания набросок кистями прямо в окне программы. Сейчас эту функцию используют для более удобного формирования двухмерной анимации.
* Инструменты визуализации. Есть сразу несколько предустановленных средств для показа результата работы, а также предусмотрена совместимость со сторонними рендерами от разных разработчиков.
* Видеоредактор. О наличии этой функции могут не догадываться даже очень опытные пользователи. Однако в программе действительно предусмотрен редактор видеороликов с неплохим инструментарием.
* Игровой движок. Последние версии программы оснащаются собственным игровым движком, с помощью которого можно создавать приложения с интерактивными функциями. При наличии определенных навыков, не составит труда внести изменения в создаваемую игру посредством интерфейса Python API [2].

Для построения сцены была использована программа Twinmotion предназначенная для трехмерной визуализации в реальном времени, разработанная для профессионалов в области архитектуры, строительства, городского планирования и озеленения. Она позволяет создавать фотореалистичные изображения проектов, панорамы и стандартные или 360° VR-видео за секунды. Twinmotion сочетает в себе интуитивно понятный интерфейс и мощь Unreal Engine от Epic Games, что делает его простым в освоении и использовании, независимо от размера и сложности проекта, материалов, ИТ-знаний пользователя или предпочитаемого им BIM-решения.

Основные особенности Twinmotion включают:

* Фотореалистичность: Twinmotion использует технологию фотореалистичного рендеринга, что позволяет создавать визуализации, похожие на реальные фотографии.
* Автоматизация: Программа автоматически генерирует текстуры, освещение и эффекты, основываясь на реальных данных, что значительно упрощает процесс создания визуализаций.
* Интеграция с BIM: Twinmotion может быть интегрирован с программами BIM (Building Information Modeling), что позволяет использовать данные из CAD-моделей для создания визуализаций.
* Поддержка VR/AR: Twinmotion поддерживает виртуальную и дополненную реальность, что позволяет пользователям просматривать визуализации в виртуальной среде [3].

Из-за наличия в проекте высокополигональных моделей, которые различные системы могут обрабатывать с трудом была применена технология Nanite представляющая из себя систему визуализации геометрии, которая использует новый формат внутренней сетки и технологию рендеринга для визуализации деталей в пиксельном масштабе и увеличения количества объектов. Она разумно обрабатывает только те детали, которые могут быть восприняты, и не более того [4]

Система Nanite обладает многими преимуществами для создания реалистичных и захватывающих сцен, такими как:

Значительное увеличение сложности геометрии, увеличение количества треугольников и объектов, чем это было возможно ранее в режиме реального времени.

Nanite System имеет множество преимуществ для создания реалистичных и захватывающих сцен, таких как:

* Значительное увеличение сложности геометрии, большее количество треугольников и объектов, чем это было возможно ранее в режиме реального времени.
* Теперь можно напрямую импортировать исходные иллюстрации кинематографического качества, такие как скульптуры ZBrush и фотограмметрические сканы.
* Уровень детализации (LOD) обрабатывается автоматически и больше не требует ручной настройки для отдельных уровней детализации сетки.

Код для взаимодействия пользователя и приложения был написан с помощью языка визуального программирования Blueprint Visual Scripting из-за низких требований к вычислительным мощностям компьютера при обработке команд пользователя.

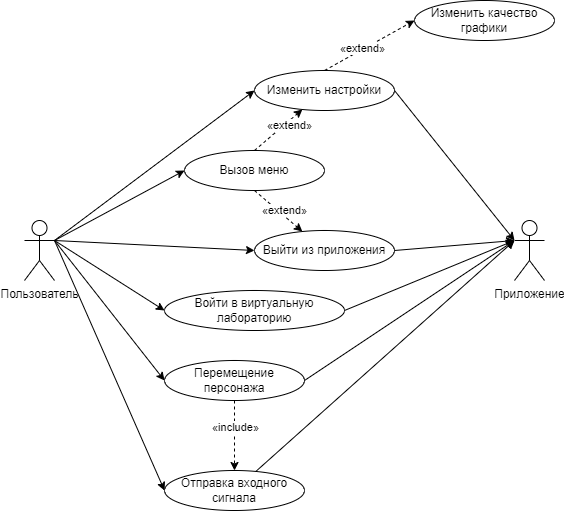
Система Blueprint Visual Scriptingв Unreal Engine — это полноценная система сценариев игрового процесса, основанная на концепции использования интерфейс на основе узлов для создания элементов геймплея из Unreal Editor. Как и многие распространенные скриптовые языки, он используется для определения объектно-ориентированных классов или объектов в движке. При использовании UE5 часто можно обнаружить, что объекты, определенные с помощью Blueprint, в просторечии называются просто «Blueprints».

Эта система является чрезвычайно гибкой и мощной, поскольку она дает возможность дизайнерам использовать практически весь спектр концепций и инструментов, обычно доступных только программистам. В дополнение Специфичная для Blueprint разметка, доступная в реализации C++ в Unreal Engine, позволяет программистам создавать базовые системы, которые могут быть расширены дизайнерами [5].

# 2 ПРАКТИЧЕСКАЯ ЧАСТЬ

## 2.1 Построение и описание диаграмм проекта

Диаграмма Use Case представленная на рисунке 2.1 позволяет описать взаимодействие пользователя и приложения.



**Рисунок 2.1 — Диаграмма вариантов использования**

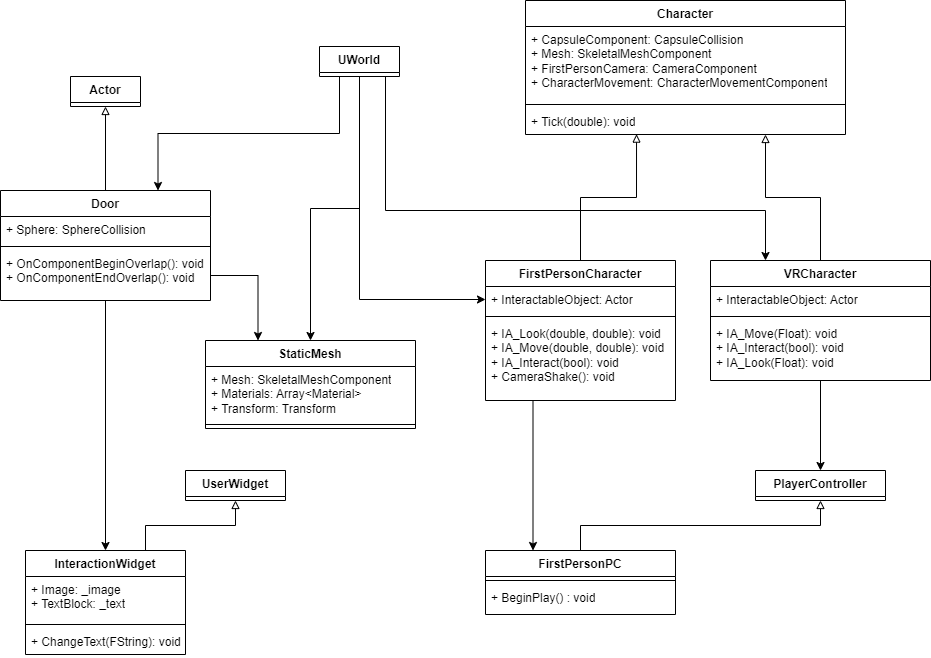
В данной Use Case диаграмме можно наблюдать возможные варианты взаимодействия пользователя и приложения. В данной диаграмме использования представлены 2 актера:

1. Пользователь — представляет человека, использующего программный продукт.
2. Приложение — представляет непосредственно созданный через Unreal Engine программный продукт.

Также используется несколько вариантов использования таких как:

1. Вызов меню — вызывает специальное окно, которое предоставляет пользователю дополнительный функционал в приложении такой как: выход из приложения, вызов меню настроек.
2. Изменить настройки — аналогично предыдущему пункту вызывает специальное окно, позволяющее пользователю изменять настройки приложения. Предусматривает что перед изменением настроек пользователь должен вызвать основное меню.
3. Изменить качество графики — позволяет пользователю изменять графическую часть приложения по заранее стозданным графическим пресетам. Подразумевает вызов меню отвественного за изменение настроек перед непосредственным использованием.
4. Выйти из приложения — предоставляет пользователю возможность прекратить использование программного продукта. В случае если в момент использования запущена виртуальная лаборатория пользователю необходимо перед выходом вызвать основное меню.
5. Войти в виртуальную лабораторию — непосредственный запуск процесса изучения виртуальной лаборатории.
6. Отправка входного сигнала — пользователь отправляет определенный сигнал в приложение путем нажатия клавиш на клавиатуре на который приложение соответствующе реагирует.
7. Перемещение персонажа — подразумевает изменение положения персонажа в пространстве в зависимости от посылаемого пользователем входного сигнала.

Диаграмма классов, представленная на рисунке 2.2 позволяет описать иерархическую структуру классов программного продукта.



**Рисунок 2.2 — Диаграмма классов**

Классы без обозначенных полей и методов такие как: Actor, UWorld, UserWidget и PlayerController являются встроенными в игровой движок и доступ к ним для переопределения у пользователя обычно закрыт.

Класс Character также является базовым, однако в отличии от вышеперечисленных классов его компоненты использовались при проектировании классов наследников.

Перечислим назначение представленных классов:

* UWorld — класс представляющий уровень в приложении на котором расположены все физические объекты.
* Actor — класс представляющий любой объект который можно расположить на уровне.
* Door — наследник класса Actor представляющий собой объект двери на уровне с определенным в нем кодом для взаимодействия с пользователем.
* StaticMesh — класс представляющий часть геометрии состоящей из полигонов, которые могут быть отправлены в видео память и отрендерены видеокартой.
* UserWidget — класс представляющий виджет, который можно расположить на экране пользователя или как объект уровня аналогично обычному GUI приложению.
* InteractionWidget — является наследником класса UserWidget и представляет собой иконку взаимодействия пользователя с объектом. В него входит изображение кнопки, на которую объект реагирует и текст описывающий действие.
* Character — класс представляющий собой репрезентацию пользователя с возможностью перемещения по миру.
* FirstPersonCharacter — является наследником класса Character отвечающий за аватар пользователя с управлением через клавиатуру и мышь, геймпад и смартфон.
* VRCharacter — также наследник класса Character отвечающий за аватар пользователя под управлением шлема виртуальной реальности и соответствующего контроллера.
* PlayerController — представляет функционал считывания ввода пользователя и перевод его в язык действий.
* FirstPersonPC — наследник класса PlayerController выполняющий тот же функционал, но для объекта класса FirstPersonCharacter из-за особенностей управления под VR платформы.

## 2.2 Проектирование приложения

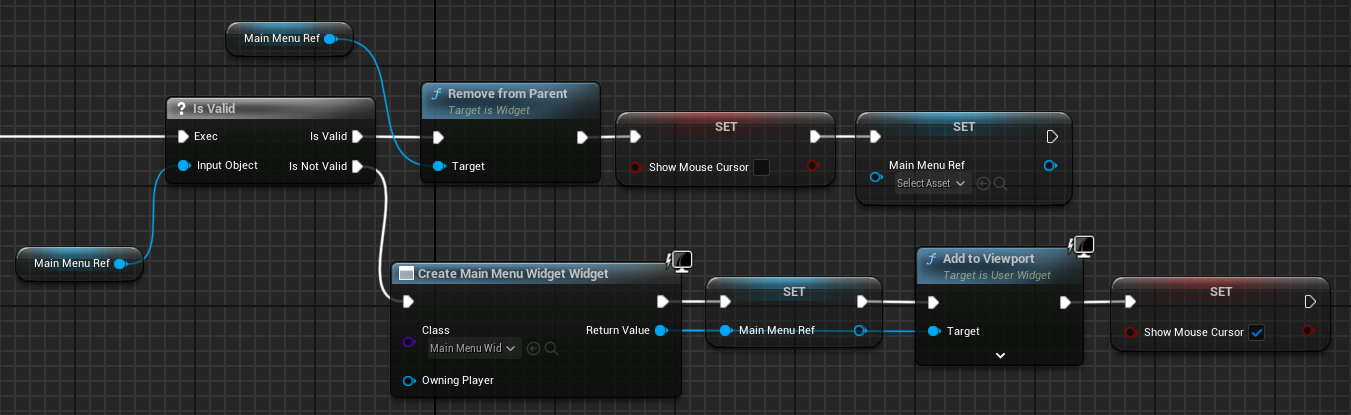
Как уже описывалось в теоретической части проектирование сцены производилось через приложение Twinmotion из-за широкого ассортимента средств для проектирования архитектуры, работы с CAD моделями и удобной интеграции с Unreal Engine.

Перед непосредственным построением виртуальной лаборатории было необходимо создать объекты, которые будут в ней расположены. Для этого было использовано программное обеспечение для создания трехмерной графики Blender при проектировании простых объектов, таких как: вентиляция, дверь, окна и т.д. Однако в приложении также присутствуют достаточно сложные объекты, которые были импортированы через сам Twinmotion из готовых CAD или FBX моделей.

После подготовки всех необходимых моделей и текстур для них происходит их размещение на сцене. Поскольку Twinmotion является программой для проектирования архитектуры в неё можно поместить чертеж здания и расставлять объекты по их размерам и расположению на этом чертеже.

После построения сцены переходим к её переносу в Unreal Engine. Для этого был использован плагин предназначенной для непосредственного переноса сцены в игровой движок. Поскольку сцена импортируется целиком, не учитывая расположенные на ней объекты её редактирование через Unreal Engine становится невозможным. Объекты, с которыми у пользователя предусмотрено взаимодействие добавлялись на сцену после её импорта.

Заключительным этам при проектировании приложения является написание кода для возможности взаимодействия с пользователем. Поскольку программа не подразумевает сложный вычислений, выполняемых кодом, для его написания использовался язык визуального программирования Blueprint Visual Scripting. Пример использования языка представлен на рисунке 2.3.



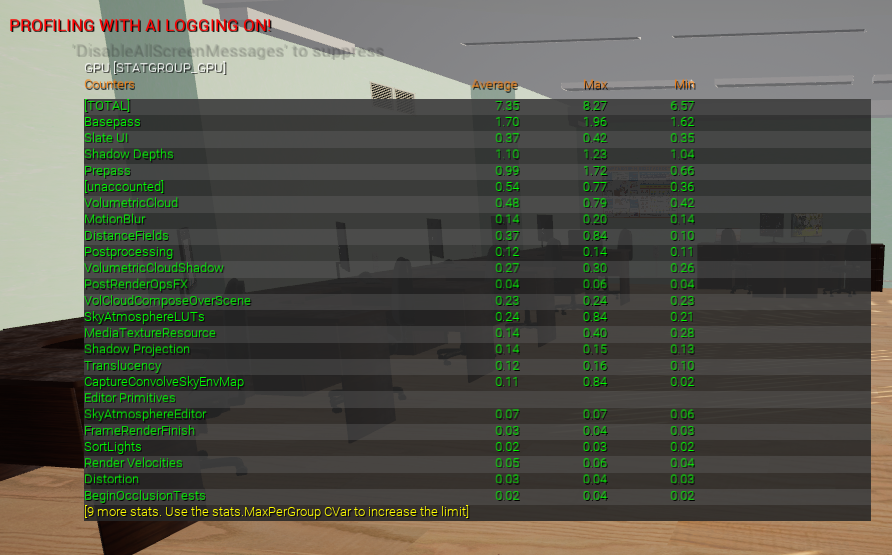
**Рисунок 2.3 — Вызов главного меню**

Данный код используется для вызова виджета главного меню при нажатии на кнопку и его удаления если виджет уже существует на экране.

## 2.3 Тестирование приложения

Тестирование приложения осуществлялось через встроенные в Unreal Engine средства способные анализировать нагрузку на систему, а именно: подсчет использованной памяти, нагруки на видеокарту, нагрузки на процессор и т.д.

Unreal Engine позволяет не только просматривать сколько ресурсов использует приложение, но и какие объекты их потребляют. Пример тестирования нагрузки на видеокарту представлен на рисунке 2.4.



**Рисунок 2.4 — Анализ времени обработки видеокартой**

В данном случае можно наблюдать сколько миллисекунд необходимо для построения кадра и обработки какого-либо параметра, и наибольшее время занимают: Basepass — рендеринг основных объектов сцены, Shadow Depths — глубина теней, Prepass — подготовка данных необходимых для последующего рендеринга сцены. Очевидно, что данные показатели могут изменяться в зависимости от производительности видеокарты.

На рисунке 2.5 представлено использование памяти приложением.



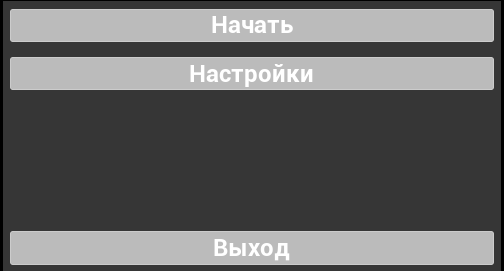
**Рисунок 2.5 — Тестирование использования памяти**

В данном случае можно наблюдать 2 параметра использующих около 3000 мб. памяти каждый, однако это лишь количество памяти, которое движок может выделить для обработки текстур и может быть изменено в настройке проекта. Реально используемая память перечислена ниже где наибольший объем памяти занят текстурами — 304 мб., шейдерами — 118 мб., информацией о StaticMesh объектах — 86 мб. и памятью, выделенной под потоковую передачу — 76 мб. Суммарное потребление памяти составляет около 650 мб.

## 2.4 Руководство пользователя

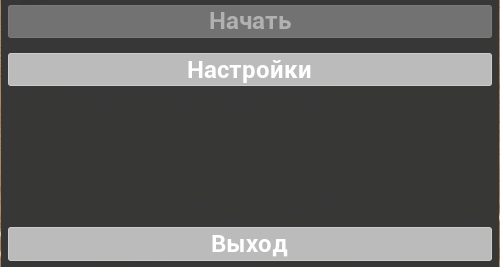
При запуске приложения пользователю отображается основное меню представленное на рисунке 2.6 с возможными вариантами выбора действий, а именно:

1. Начать — создаётся виртуальный аватар пользователя, и подгружается уровень с непосредственно смоделированной лабораторией. Пользователь получает возможность перемещаться в пространстве и рассматривать расположенные в ней объекты.
2. Настройки — вызывает меню настроек позволяя пользователю изменять графическую составляющую приложения по заранее созданным пресетам из 5 пунктов от низкого до кинематографического качества.
3. Выход — непосредственно завершение процесса пользования приложением.



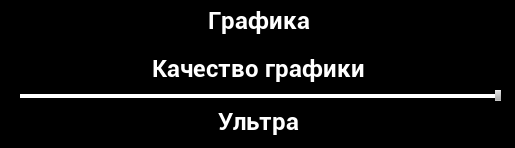
**Рисунок 2.6 — Главное меню**

Аналогичное меню можно вызвать в процессе изучения виртуальной лаборатории при нажатии на клавиатуре клавиши «Esc», однако при этом пункт меню «Начать» будет не доступен для использования. Пример такого меню изображен на рисунке 2.7.



**Рисунок 2.7 — Главное меню вызываемое через «ESC»**

При нажатии на кнопку «Настройки» вызывается меню позволяющее настраивать графику в приложении путем перемещения ползунка под надписью «Качество графики». При настройке графики доступны 5 пресетов: «Низкое», «Среднее», «Высокое», «Очень высокое» и «Ультра» качество графики. Пресеты графики отвечают за следующие графические показатели: масштаб разрешения, дальность прорисовки, сглаживание, тени, глобальное освещение и качество текстур. Пример меню настроек представлен на рисунке 2.8.



**Рисунок 2.8 — Настройка графики**

При нажатии на кнопку «Начать» происходит переход на уровень с расположенной на нем лабораторией. Пример разработанной виртуальной лаборатории можно наблюдать на рисунках 2.9 – 2.10.



**Рисунок 2.9 — Кабинет 5/308 (ракурс 1)**



**Рисунок 2.10 — Кабинет 5/308 (ракурс 2)**

Перемещению между кабинетами 5/308 и 5/301 осуществляется с помощью расположенных на карте дверей, к которым пользователю необходимо подойти и нажать на клавишу «Е». Когда аватар достаточно близко к двери появляется подсказка в виде виджета отображающая на какую клавишу необходимо нажать и в какой кабинет произойдет перемещение. Сам виджет представлен на рисунке 2.11.



**Рисунок 2.11 — Виджет взаимодействия**

Поскольку приложение совместимо с несколькими платформами передвижение и поворот был адаптирован под каждую из них:

1. Клавиатура и мышь — перемещение осуществляется при нажатии на клавиши «W», «A», «S», «D» для продвижения вперед, влево, назад и вправо соответственно. Поворот камеры осуществляется перемещением мыши.
2. Смартфон — на экране устройства появляются 2 джойстика слева и справа по нижней части экрана. Левый джойтик отвечает за перемещение персонажа, а правый за вращение камерой. В добавок изменять направление камеры можно через встроенный в смартфон акселерометр.
3. Геймпад — аналогично управлению со смартфона перемещение персонажа осуществляется через левый стик контроллера, а вращение камерой через правый.

# ЗАКЛЮЧЕНИЕ

В результате выполнения проекта было создано приложение с использованием платформы для разработки игр Unreal Engine 5 демонстрирующее возможности игрового движка для создания проектов архитектуры.

Освоение Unreal Engine требует времени и усилий, но в результате получается возможность создавать проекты, которые могут поражать своим качеством и впечатляющим визуальным восприятием. Этот процесс не только ускоряет разработку, но и позволяет экспериментировать с различными идеями и концепциями в виртуальной среде, что делает его незаменимым инструментом для обучения и проведения исследований.

В эпоху цифровых технологий, когда виртуальная реальность и визуализация становятся все более важными, Unreal Engine занимает центральное место в индустрии архитектуры и дизайна. Его возможности не ограничиваются только созданием архитектурных проектов; с его помощью можно реализовать самые разнообразные идеи, от образовательных программ до виртуальных туров и интерактивных демонстраций.

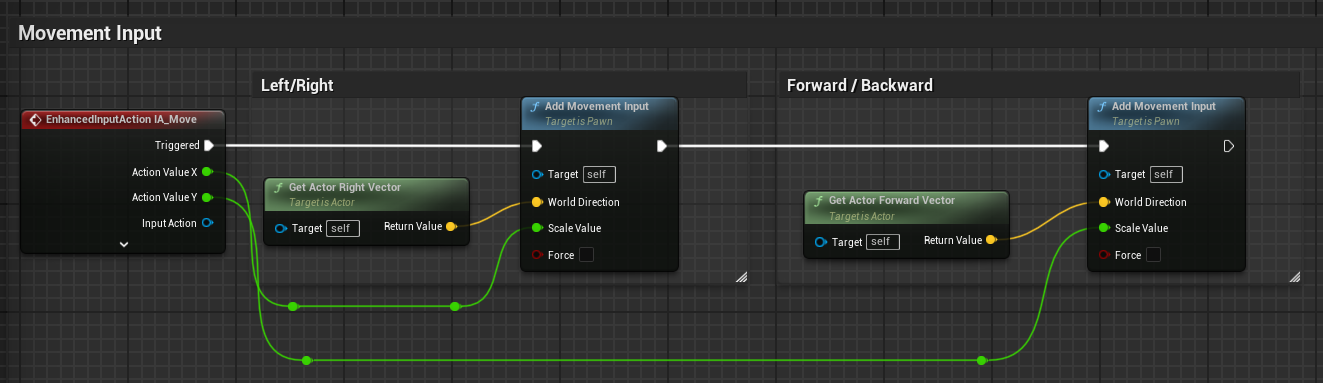
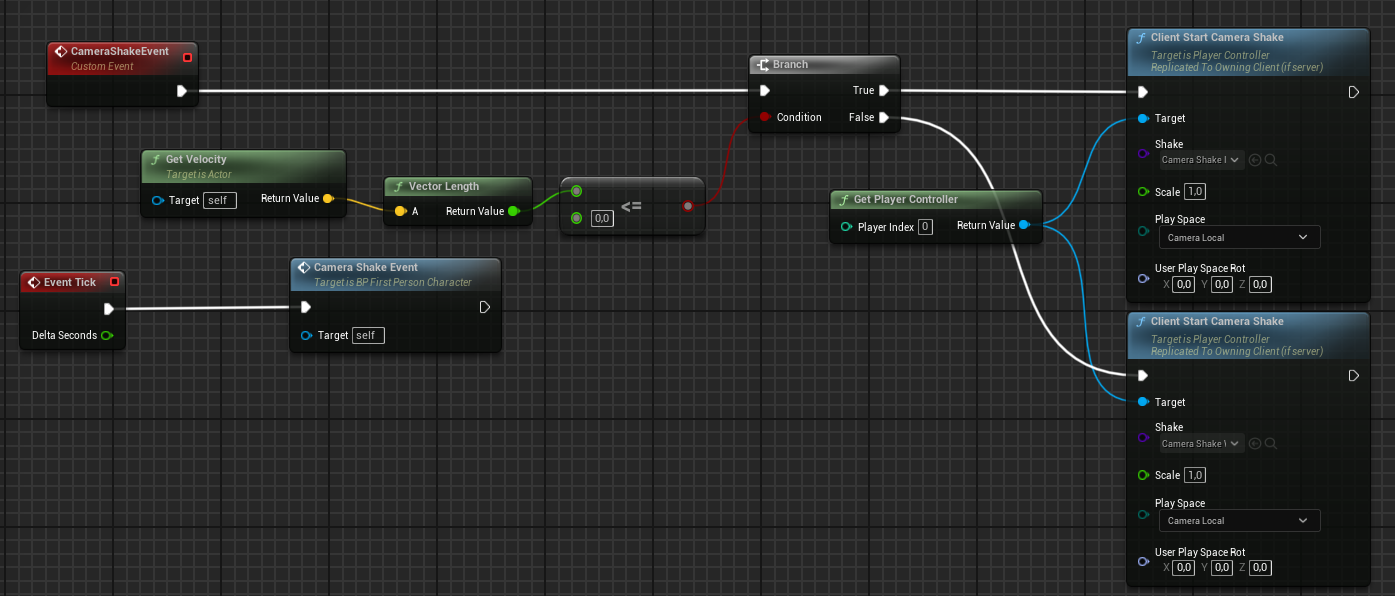
# СПИСОК **ИСПОЛЬЗОВАННЫХ** ИСТОЧНИКОВ

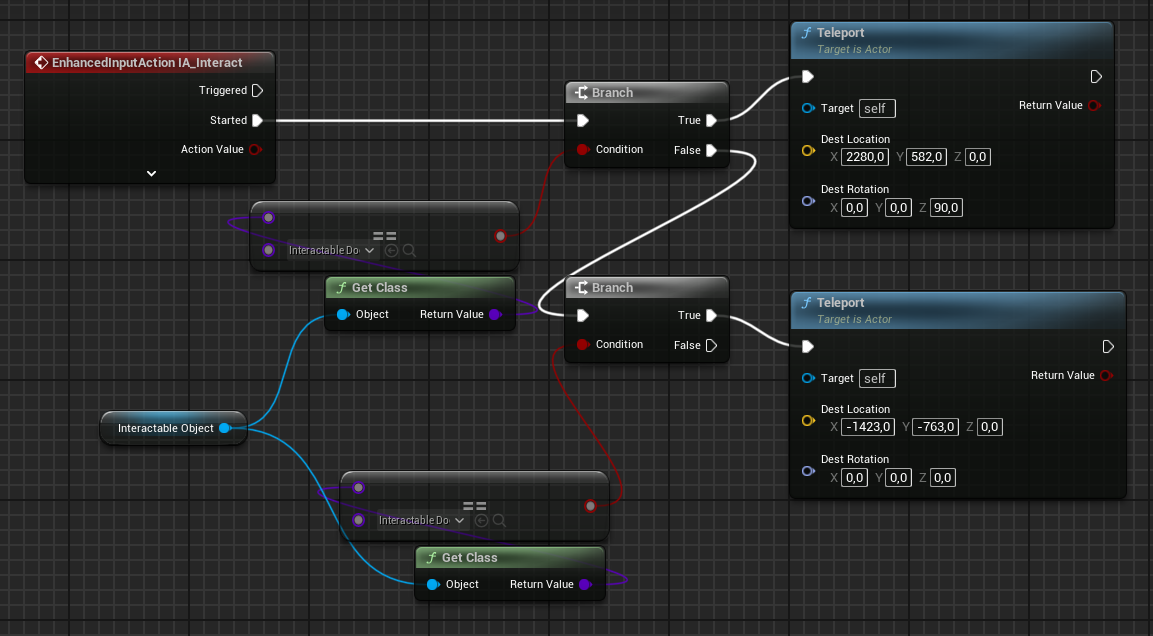
1. Преимущества Unreal Engine [Элек­трон­ный ресурс]. — Режим доступа: https://blog.artcraft.net.ua/page11435532html— Дата доступа: 02.05.2024.
2. Программа Blender 3D [Элек­трон­ный ресурс]. — Режим доступа: https://junior3d.ru/article/blender-3d.html— Дата доступа: 02.05.2024.
3. Twinmotion [Электронный ресурс]. — Режим доступа: https://graphisoft.com/partner-solutions/twinmotion - Дата доступа: 25.04.2024.
4. How to create stunning scenes with Nanite System of Unreal Engine 5 [Элек­трон­ный ресурс]. — Режим доступа: https://www.linkedin.com/pulse/how-create-stunning-scenes-nanite-system-unreal-engine-david-sierra/ — Дата доступа: 02.05.2024.
5. – Blueprint Visual Scription [Элек­трон­ный ресурс]. — Режим доступа: https://docs.unrealengine.com/4.26/en-US/ProgrammingAndScripting/Blueprints — Дата доступа: 02.05.2024.

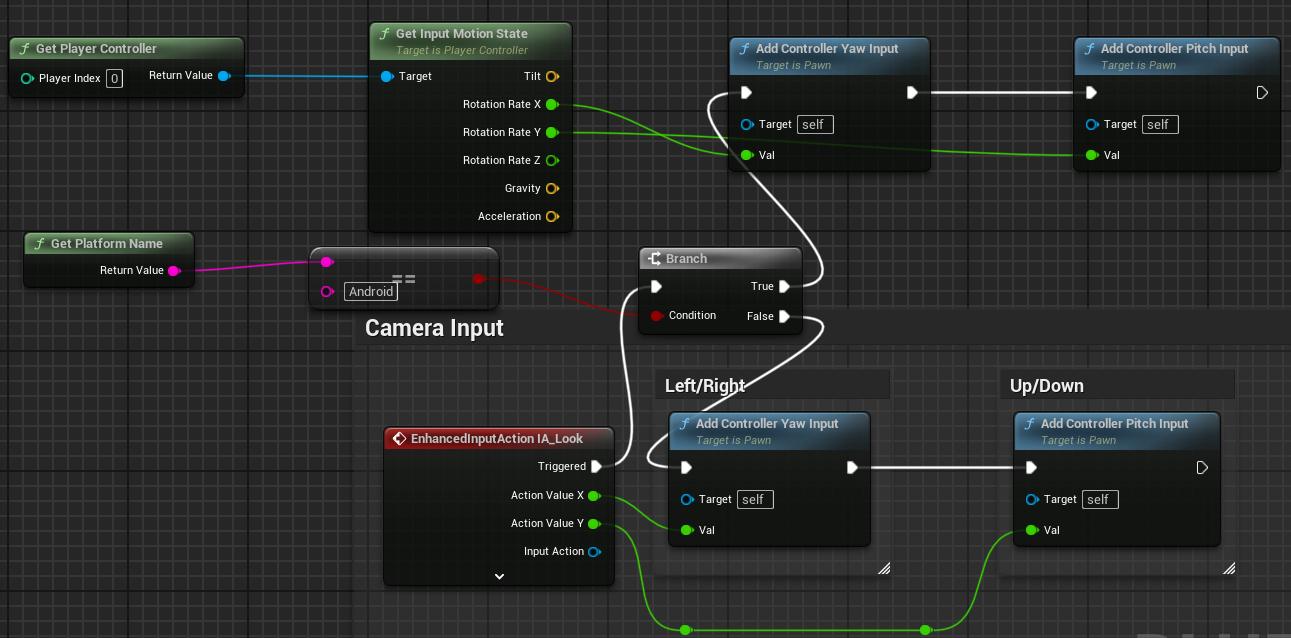
# ПРИЛОЖЕНИЯ

**ПРИЛОЖЕНИЕ А**

**Код FirstPersonCharacter**

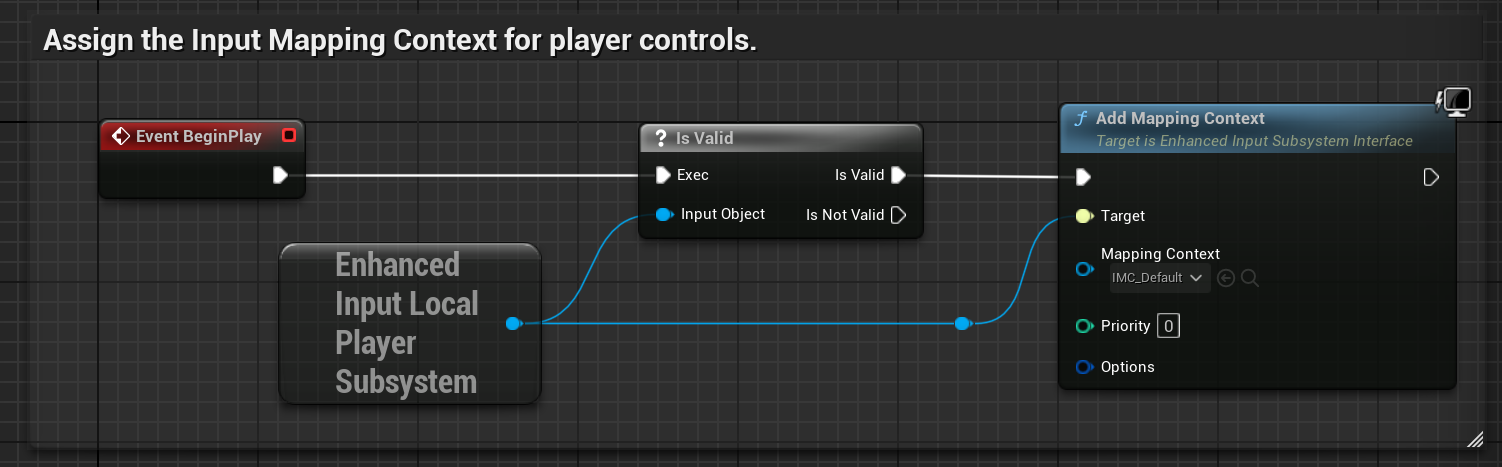
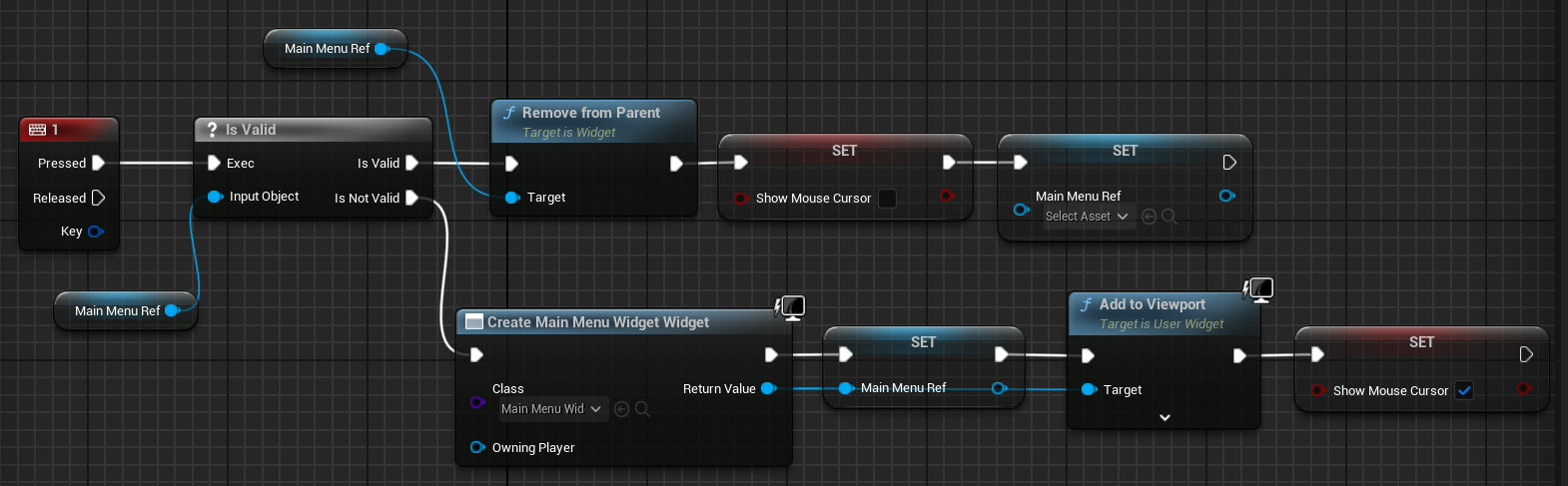
 





**ПРИЛОЖЕНИЕ Б**

**Код FirstPersonPC**



**ПРИЛОЖЕНИЕ В**

**Код VRCharacter**

