

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ
Школа бакалавриата

ОТЧЕТ

По проекту
«Разработка цифрового двойника робота для симуляционной работы с при-
менением VR»

по дисциплине «Проектный практикум»

Заказчик: Фадеев Виталий Олегович

Куратор: Шестеров Михаил Андреевич

Студенты команды

Свидер М.Ю.

Рудовский Е.А.

Шульгатый В.С.

Савельев Г.Д.

Екатеринбург, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1. Основная часть	5
1.1 Работа участников команд	5
1.1.1 Свидер Матвей Юрьевич	5
1.1.2 Шульгатый Владислав Станиславович.....	5
1.1.3 Рудовский Егор Александрович.....	6
1.1.4 Савельев Григорий Дмитриевич	6
1.2 Разбор требований заказчика	7
1.3 Анализ конкурентов.....	7
1.4 Обзор архитектуры программного продукта	8
1.4.1 Основные компоненты и их связи.....	8
1.4.2 Обоснование архитектурного решения	9
1.5 Методология разработки	9
1.5.1 Процесс разработки	10
1.5.2 Использованные технологии	12
1.6 Результаты тестирования и выявленные ошибки.....	12
1.7 Календарный план проекта	13
ЗАКЛЮЧЕНИЕ	14
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	16
ПРИЛОЖЕНИЕ А (обязательное) Инструкция по управлению роботом.....	17

ВВЕДЕНИЕ

Целью данного проекта является создание цифрового двойника робота в VR-среде для проведения симуляционных испытаний на основе данных, полученных из проблемного интервью. Для достижения поставленной цели были определены следующие задачи: проведение проблемного интервью с целевой аудиторией для выявления ключевых потребностей и требований; разработка архитектуры цифрового двойника; создание прототипа программного продукта на базе Unity; реализация сценариев взаимодействия пользователя с цифровым двойником; тестирование и валидация решения.

Актуальность проекта обусловлена стремительным развитием технологий цифровых двойников и виртуальной реальности, которые все чаще применяются в инженерии, образовании и научных исследованиях. Использование цифрового двойника позволяет минимизировать затраты на физическое тестирование, ускорить процесс отладки алгоритмов и повысить эффективность обучения работе с роботами. Особенно это важно в условиях ограниченных ресурсов и высоких требований к надежности и безопасности робототехнических систем.

Область применения данного программного продукта охватывает широкий круг пользователей — от инженеров и разработчиков, которым необходимо тестировать алгоритмы и моделировать различные сценарии поведения роботов, до образовательных учреждений, где продукт может использоваться как учебный инструмент для практической подготовки студентов. Также он будет полезен исследователям, стартапам и малому бизнесу, заинтересованным в снижении стоимости и времени разработки робототехники.

Ожидается, что по завершении проекта будет создан функциональный прототип цифрового двойника робота в VR-среде, позволяющий проводить симуляции, управлять моделью в режиме реального времени и реализовывать различные сценарии взаимодействия. Планируется, что продукт обеспечит удобство тестирования новых идей, поддержку автономных функций и возможность расширения с учетом будущих требований. Это сделает его перспективным решением для использования в научных, образовательных и коммерческих целях.

1. Основная часть

1.1 Работа участников команд

1.1.1 Свидер Матвей Юрьевич

Роль: Тимлид

Группа: РИ-320932

Как тимлид выполнял достаточно стандартные функции в команде:

- Распределение задач между сокомандниками
- Определение плана выполнения работ проекта
- Организация встреч с куратором
- Создание и защита отчетов для контрольных точек
- Помощь с вопросами поиска информации

Основной акцент делался на координации и поддержке работы всех участников, а также обеспечении эффективного взаимодействия внутри команды.

1.1.2 Шульгатый Владислав Станиславович

Роль: аналитик

Группа: РИ-320913

- Выявление проблемы проекта
- Определение решения проблемы
- Анализ целевой аудитории
- Анализ аналогов
- Разработка плана действий
- Проведение проблемного интервью
- Определение основного функционала

- Определение грядущих задач
- Описание придуманной новой функциональной возможности (датчик черной линии и движение по заданной траектории)
- Подготовка аналитических данных для написания отчета

1.1.3 Рудовский Егор Александрович

Роль: Unity-разработчик

Группа: РИ-320915

Выполнены следующие шаги:

- Подготовка технической основы для дальнейшей работы
- Установка необходимого ПО
- Выбор технологий для разработки
- Работа с 3D-моделью точной копии робота.
- Создание модели в Blender
- Интеграция модели в проект Unity
- Настройка компонентов для реализации управления
- Помощь в написании скриптов для компонентов

1.1.4 Савельев Григорий Дмитриевич

Роль: Unity-разработчик

Группа: РИ-320945

Выполнены следующие задачи:

- Написание скриптов для робота на C#
- Скрипты для движение робота
- Скрипты для управление манипулятором
- Скрипты для управления камерой
- Реализация анимации движения гусениц.

- Реализация датчика, измеряющего расстояние до объекта в камере.
- Реализация самостоятельного движения робота по заданному маршруту (линии)

1.2 Разбор требований заказчика

В ходе проведения проблемного интервью были выявлены ключевые требования заказчика к функциональности цифровой модели робота. Заказчик выразил следующие ожидания:

- реалистичная имитация работы датчика линии — чтобы цифровой двойник точно реагировал на изменение положения относительно линии, аналогично реальному роботу
- точная манипуляция движением — необходимо обеспечить плавное и достоверное поведение робота при движении, включая повороты, остановки и движение по траектории
- возможность проброса изображения с камеры — для полноценной симуляции восприятия окружающей среды, требуется передача видеопотока с виртуальной камеры на интерфейс пользователя.

Указанные требования стали основой для планирования и описания функциональных возможностей цифрового двойника в рамках проекта.

1.3 Анализ конкурентов

Анализ конкурентов представлен в таблице 1.1

Критерий	Наш проект	RoboDK	Siemens	UnityRobotics	VPL	IsaacSim
VR-поддержка	Да	Нет	Нет	Опционально	Нет	Ограничено
Ориентация на обучение	Да	Частично	Нет	Частично	Да	Нет
Доступность	Высокая	Низкая	Низкая	Да	Да	Низкая
Гибкость и кастомизация	Высокая	Средняя	Низкая	Высокая	Низкая	Высокая
Простота освоения	Да	Нет	Нет	Нет	Да	Нет

Таблица 1.1 – Анализ конкурентов

1.4 Обзор архитектуры программного продукта

Проект представляет собой симуляцию робота на гусеничном ходу с манипулятором (клешней) и камерой, разработанный в Unity. Архитектура построена на компонентно-ориентированном подходе, характерном для Unity, с использованием MVC-подобной структуры для разделения логики, управления и представления.

1.4.1 Основные компоненты и их связи

Модель робота

1) Гусеничный движитель:

- Физика движения реализована через HorizontalMovement.
- AutoPilot (зависит от Horizontal Movement) - система автопилота
- Для коллизий используется комбинация Box Collider (основной корпус) и Mesh Collider (детализированные элементы).
- Управление направлением InputAction (вперед/назад, поворот).

2) Манипулятор (клешня)

– Claw, ClawDetector, ClawHolding, ClawController - система хватания клешней.

- Claw - открытие/закрытие клешни,
- ClawDetector(зависит от Claw) - определяет есть ли предмет в клешне,
- ClawHolding (зависит от ClawDetector) - определяет захвачен ли объект двумя клешнями,
- ClawsController(зависит от Claw) - делает возможным управление через клавиши.

3) Камера

– Виртуальная камера, прикрепленная к роботу с возможностью поворота.

- FPVController - переключение на вид от первого лица
- MeasureDistance(зависит от FPVController) - измерение расстояния до ближайшей поверхности

Система управления

- 1) Ввод пользователя - обработка клавиатуры/геймпада через Unity Input System.
- 2) Логика движения - преобразует ввод в команды для гусениц.
- 3) Логика манипулятора - управление клешней (захват, вращение, перемещение).

Физика и окружение (Physics/Environment)

- 1) Unity Physics (Rigidbody, Colliders)
 - Реализация реалистичного взаимодействия с объектами.
 - Объекты для захвата, тестовая площадка.

1.4.2 Обоснование архитектурного решения

1. Компонентный подход:
 - Unity изначально заточена под GameObject + Component, что удобно для роботосимуляции.
 - Каждая часть робота (гусеницы, клешня, камера) — отдельный компонент, что упрощает переиспользование кода.
2. MVC-подобная структура:
 - Четкое разделение на Model (робот), Controller (управление), View (FPV).
 - Позволяет легко заменять логику (например, перейти с ручного управления на автономное).

1.5 Методология разработки

Разработка цифрового двойника велась с использованием итеративного подхода, сочетающего элементы гибкой методологии Agile и прототипирования. Такой подход позволил на каждом этапе вносить улучшения на основе обратной связи и текущих результатов тестирования. В качестве основной среды разработки использовался Unity, так как она обеспечивает достаточную гибкость для работы с VR-технологиями, позволяет реализовать управление, визуализацию и физику модели.

На первом этапе были собраны данные из проблемного интервью, проведённого с представителями целевой аудитории. Это позволило определить, какие функции наиболее важны для пользователей, а также — какие ограничения и требования следует учитывать. На основе полученных данных была составлена архитектура цифрового двойника и определён список базовых функций для реализации MVP (минимально жизнеспособного прототипа).

1.5.1 Процесс разработки

- Управление роботом
 - 1) Было изучено, какие типы скриптов в Unity можно использовать для реализации движения объектов. Особое внимание уделялось HorizontalMovement.cs и AxisRotation.cs, которые были адаптированы под движение корпуса, поворот руки и кисти.
 - 2) Скрипты были привязаны к соответствующим частям 3D-модели робота: корпус, рука, кисть.
 - 3) Реализовано пошаговое управление:
 - а) Вперёд/назад: клавиши W/S
 - б) Поворот корпуса: клавиши A/D
 - в) Вращение руки: клавиши Q/E
 - г) Поворот кисти: клавиши Z/X
 - д) Открытие/закрытие кисти: клавиша Space

- Дополнительные модули
 - 1) Анимация гусениц
 - а) Реализована анимация движения гусениц, синхронизированная с перемещением модели.
 - б) Обеспечена согласованность направления движения и визуального эффекта вращения.
 - 2) Физическая модель и взаимодействие с окружением
 - а) Настроены компоненты Rigidbody и Collider для моделирования веса, инерции и столкновений.
 - б) Робот теперь корректно реагирует на столкновения с объектами сцены, может их отталкивать.
 - в) Улучшена устойчивость модели при наклонах и при касании других тел.
 - 3) Захват объектов клешнёй
 - а) Внедрена система захвата и удержания объектов клешнёй робота.
 - б) Учтена различная масса и форма объектов, что делает захват реалистичным и интерактивным.
 - в) Протестирована надёжность фиксации предметов при движении.
 - 4) Камера и ориентация головы
 - а) Установлена камера первого лица (FPV), смонтированная в «голове» робота.
 - б) Камера выполняет измерение расстояний до объектов с помощью Raycasting.
 - в) Реализована синхронизация направления взгляда камеры с поворотом головы — робот «смотрит» в нужном направлении.
 - 5) Режим полуавтопилота (линейное следование)

- а) Добавлен режим автономного следования по линии на поверхности.
- б) Разработан простой алгоритм распознавания чёрной линии и поворотного управления.
- в) Обеспечен переключатель между ручным и автоматическим режимами.

1.5.2 Использованные технологии

Unity — основная среда разработки и визуализации.

C# — написание логики, управление, взаимодействие.

Unity Physics (Rigidbody, Collider, Raycasting) — физика и столкновения.

Animator и Shader Graph — анимация и визуальные эффекты (гусеницы, захват).

Custom AI Logic — трекинг линии.

Camera & Raycasting — датчики дистанции и ориентация в пространстве.

1.6 Результаты тестирования и выявленные ошибки

Тестирование проводилось поэтапно — после каждой новой функциональной реализации. Использовался ручной тестинг в среде Unity с URP (Universal Render Pipeline). Были протестированы следующие сценарии: Результаты тестирования представлены в таблице 1.2.

Этап	Проверка	Результат
Движение вперёд/назад, поворот корпуса	W/S, A/D	Успешно
Вращение руки и клешни	Q/E, Z/X	Успешно
Работа камеры и датчика	Наведение на объект	Успешно
Трекинг по чёрной линии	Навигация по сцене	Успешно (в простых условиях)
Открытие клешни	Space	Успешно

Таблица 1.2 – Тестирование робота

Выявленные ошибки:

- При первоначальной настройке вращение клешни конфликтовало с поворотом всей руки — решено пересмотром родительской иерархии объектов.
- На раннем этапе камера "проваливалась" внутрь модели — устранено путём корректной привязки к крепёжной точке головы.
- Слишком резкое движение корпуса при повороте — устранено добавлением ограничения скорости поворота и сглаживания через Lerp-функции.
- Сбои в анимации гусениц — решено пересмотром логики запуска анимации в зависимости от скорости трансформа.

1.7 Календарный план проекта

На рисунке 1.1 представлен календарный план проекта.

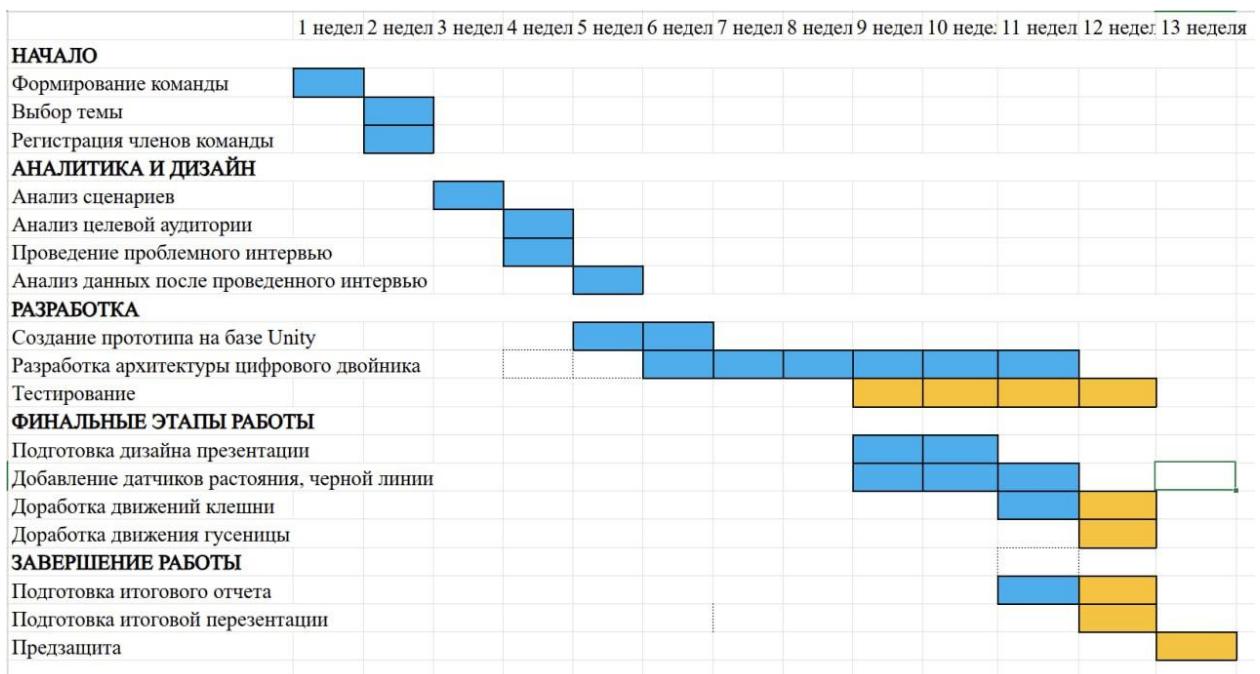


Рисунок 1.1 – Календарный план

ЗАКЛЮЧЕНИЕ

В результате реализации проекта по созданию цифрового двойника робота в VR-среде для проведения симуляционных испытаний, была достигнута основная цель — разработка функционального прототипа, который отвечает ключевым потребностям и требованиям, выявленным в ходе проблемного интервью. Проект успешно выполнил поставленные задачи, включая разработку архитектуры цифрового двойника, создание прототипа на базе Unity, реализацию сценариев взаимодействия пользователя с моделью и тестирование решения.

Актуальность проекта подтверждается растущим интересом к технологиям цифровых двойников и виртуальной реальности, которые находят применение в различных областях, таких как инженерия, образование и научные исследования. Создание цифрового двойника позволяет значительно сократить затраты на физическое тестирование, ускорить процесс отладки алгоритмов и повысить эффективность обучения, что особенно важно в условиях ограниченных ресурсов и высоких требований к надежности робототехнических систем.

Оценка соответствия программного продукта поставленным требованиям показала, что разработанный цифровой двойник успешно выполняет функции, необходимые для тестирования алгоритмов и моделирования различных сценариев поведения роботов. Продукт охватывает широкий круг пользователей, включая инженеров, разработчиков, образовательные учреждения и исследователей, что подтверждает его универсальность и практическую ценность.

Качество программного продукта было оценено на основе результатов тестирования, в ходе которого были выявлены некоторые незначительные дефекты, касающиеся производительности в условиях высокой нагрузки и логики взаимодействия компонентов. Эти недостатки, хотя и не критичные, могут оказывать влияние на общую работоспособность системы, особенно в

сценариях, требующих высокой точности. Рекомендуется провести дополнительную оптимизацию кода и улучшение алгоритмов взаимодействия для повышения стабильности и производительности.

В качестве предложений по улучшению продукта и его дальнейшему развитию можно выделить следующие направления:

6) Расширение функционала: Внедрение дополнительных сценариев и возможностей для более сложных симуляций, что позволит пользователям глубже исследовать поведение робота в различных условиях.

7) Интеграция с внешними системами: Разработка API для взаимодействия с другими программными продуктами и системами, что обеспечит более широкие возможности для анализа данных и интеграции в существующие производственные процессы.

8) Обучение и поддержка пользователей: Было создан обучающий материал для поддержки пользователей (Приложение 1), что поможет им максимально эффективно использовать возможности цифрового двойника.

Проект по созданию цифрового двойника робота в VR-среде не только достиг поставленных целей, но и открыл новые горизонты для дальнейшего развития и улучшения, что позволит значительно повысить его ценность для пользователей и заказчиков.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Баранов, А. В. Виртуальная реальность: технологии и приложения / А. В. Баранов. — М.: Наука, 2020. — 256 с.
2. Кузнецов, И. А. Основы разработки игр на Unity / И. А. Кузнецов. — СПб.: Питер, 2019. — 320 с.
3. Лебедев, С. В. Моделирование и симуляция в виртуальной реальности / С. В. Лебедев. — Екатеринбург: УрФУ, 2021. — 180 с.
4. Михайлов, А. Н. Виртуальная реальность в образовании: возможности и перспективы / А. Н. Михайлов. — М.: Издательство МГУ, 2022. — 210 с.
5. Никифоров, Д. В. Unity 3D: создание игр и приложений / Д. В. Никифоров. — Казань: Казанский университет, 2020. — 300 с.
6. Петров, В. И. Технологии виртуальной реальности: от теории к практике / В. И. Петров. — Новосибирск: Сибирское университетское издательство, 2021. — 250 с.
7. Сидоров, А. А. Моделирование робототехнических систем в среде Unity / А. А. Сидоров. — Ростов-на-Дону: ЮФУ, 2020. — 220 с.
8. Тихомиров, А. С. Виртуальная реальность: принципы и технологии / А. С. Тихомиров. — Москва: Бином. Лаборатория знаний, 2019. — 300 с.
9. Федоров, И. П. Основы проектирования виртуальных симуляторов / И. П. Федоров. — Санкт-Петербург: Лань, 2021. — 240 с.
10. Шевченко, Е. В. Виртуальная реальность и ее применение в инженерии / Е. В. Шевченко. — Екатеринбург: УрГЭУ, 2022. — 190 с.

ПРИЛОЖЕНИЕ А

(обязательное)

Инструкция по управлению роботом

Основные элементы управления

Передвижение робота

- W - движение вперёд
- S - движение назад
- A - поворот корпуса влево
- D - поворот корпуса вправо

Управление манипулятором

- Q - поворот руки влево
- E - поворот руки вправо
- Z - поворот кисти влево
- X - поворот кисти вправо
- Пробел (Space) - открыть/закрыть кисть

Специальные режимы

- Tab - включение/выключение автопилота (робот будет двигаться по заданной программе)
- C - переключение в режим FPV (First Person View - вид от первого лица)