

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»

РАБОТА ПРОВЕРЕНА

Рецензент

_____ С.С. Меньшенин
«__» июня 2021 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой ЭВМ

_____ Г.И. Радченко
«__» _____ 2021 г.

Система поддержки indoor соревнований по велосипедному спорту

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Руководитель работы,

к.т.н., доцент каф. ЭВМ

_____ И.Л. Кафтанников
«__» _____ 2021 г.

Автор работы,

студент группы КЭ-222

_____ О.И. Морозов
«__» _____ 2021 г.

Нормоконтролёр,

ст. преп. каф. ЭВМ

_____ С.В. Сяськов
«__» _____ 2021 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

Г.И. Радченко

«___» _____ 2021 г.

ЗАДАНИЕ

на выпускную квалификационную работу магистра
студенту группы КЭ-222
Морозову Олег Ивановичу
обучающемуся по направлению
09.03.01 «Информатика и вычислительная техника»

1. Тема работы: «Система поддержки indoor соревнований по велосипедному спорту» утверждена приказом по университету от 24 апреля 2021 г. № 627

2. Срок сдачи студентом законченной работы: 1 июня 2021 г.

3. Исходные данные к работе:

- языки программирования: C#, Python;
- платформы разработки: Windows, Linux ARM;
- игровой движок – Unity;
- среда разработки – PyCharm Professional;
- микроконтроллер для передачи данных – Raspberry PI Zero W;
- библиотеки ARM: I2C 128x64 OLED Display, asyncio, mpu6050,

VL53L0X.

4. Перечень подлежащих разработке вопросов:

- анализ литературы по теме «умные тренажеры» и «соревнования по велоспорту»;
- рассмотрение существующих аналогов эмуляторов соревнования по велоспорту, оценка их сильных и слабых сторон;
- разработка собственного сервиса соревнования по велоспорту;
- оценка работоспособности сервиса с разными режимами и в разных условиях.

5. Дата выдачи задания: 1 декабря 2020 г.

Руководитель работы _____/И.Л. Кафтанников /

Студент _____/О.И. Морозов /

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	01.03.2021	
Разработка модели, проектирование	01.04.2021	
Реализация аппаратного прототипа велосипедного тренажера	15.05.2021	
Реализация программного прототипа сервиса велосипедных соревнований	01.05.2021	
Тестирование, отладка, эксперименты	15.05.2021	
Компоновка текста работы и сдача на нормоконтроль	24.05.2021	
Подготовка презентации и доклада	30.05.2021	

Руководитель работы _____ /И.Л. Кафтанников /

Студент _____ /О.И. Морозов/

Аннотация

О.И. Морозов. Система поддержки indoor соревнований по велосипедному спорту. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭКН; 2021, 63 с., 29 ил., библиогр. список – 18 наим.

В качестве данной выпускной квалификационной работы производится детальный анализ литературы по темам: домашний симулятор соревнований по велоспорту. Организуется разработка программного обеспечения серверной, интерфейсной, микроконтроллерной частей, а также базы данных. Производится выборка и анализ результатов работы разработанного велотренажера в предложенных режимах. Рассматриваются преимущества и недостатки разработанного программно-аппаратного комплекса. Доказывается способность системы разработанного велосипедного тренажера и сервиса велосипедных соревнований.

Пояснительная записка включает в себя введение, оглавление, основную часть, заключение и библиографический список.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	8
1.АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	10
1.1. ЦЕЛЬ ДИПЛОМНОЙ РАБОТЫ.....	10
1.2. ОБЗОР ЛИТЕРАТУРЫ	10
1.3. ЭТАПЫ РЕШЕНИЯ ЗАДАЧИ.....	12
1.4. ОБЗОР АНАЛОГОВ	12
1.5 АНАЛИЗ И ПОДБОР КОМПОНЕНТОВ АППАРАТНОЙ ЧАСТИ КОМПЛЕКСА	15
1.6. ВЫБОР ВИДЕОМОНИРОРА.....	24
1.7 ВЫБОР ИГРОВОГО ДВИЖКА.....	26
1.8 ВЫВОДЫ ПО ГЛАВЕ 1	29
2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ К КОМПЛЕКСУ	31
2.1 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ.....	31
2.2 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	31
2.2.1 ТРЕБОВАНИЯ АППАРАТНОЙ ЧАСТИ КОМПЛЕКСА	31
2.2.2 ТРЕБОВАНИЯ ПРОГРАММНОЙ ЧАСТИ КОМПЛЕКСА	31
2.2.3 ТРЕБОВАНИЯ К ДОКУМЕНТАЦИИ	32
2.2.4 ТРЕБОВАНИЯ К ДОКУМЕНТАЦИИ	32
2.3 ВЫВОДЫ ПО ГЛАВЕ 2	32
3 ПРОЕКТИРОВАНИЕ.....	33
3.1 ФУНКЦИОНАЛЬНЫЙ СОСТАВ.....	33
3.2 ОБЩАЯ СТРУКТУРА АППАРАТНОЙ ЧАСТИ	34
3.3 СТРУКТУРНЫЙ СОСТАВ ПРОГРАММНОЙ ЧАСТИ	36
3.4 ДИАГРАММА ПРЕЦЕДЕНТОВ ПРИЛОЖЕНИЯ	37
3.4.1 Прецедент «играть»	38
3.4.2 Прецедент «Прогресс».....	38
3.4.3 Прецедент «Настройки»	38
3.4.4 Прецедент «Выход»	39
3.4.5 Прецедент «Тренировка»	39
3.4.6 Прецедент «Многопользовательский режим»	39
3.4.7 Прецедент «Назад»	39
3.4.8 Прецедент «Звук».....	39
3.4.9 Прецедент «Графика».....	39
3.4.10 Прецедент «Управление»	39

3.4.11 Прецедент «Группы»	39
3.4.12 Прецедент «Чемпионаты»	39
3.4.13 Прецедент «VR шлем».....	39
3.4.14 Прецедент «Тренажер»	39
3.5 КАЛИБРОВКА ТРЕНАЖЕРА.....	39
3.6 ВЫВОДЫ ПО ГЛАВЕ 3	40
4 РЕАЛИЗАЦИЯ	41
4.1 РЕАЛИЗАЦИЯ ПРОГРАММНОЙ ЧАСТИ.....	41
4.1.1 ФАЙЛОВАЯ СТРУКТУРА ПРИЛОЖЕНИЯ	41
4.1.2 РЕАЛИЗАЦИЯ ИНТЕРФЕЙСОВ	42
4.2 РЕАЛИЗАЦИЯ АППАРАТНОЙ ЧАСТИ	47
4.3 ВЫВОДЫ ПО ГЛАВЕ 4	48
5 ТЕСТИРОВАНИЕ	49
5.1 ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ.....	49
5.2 ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЕ	49
5.3 ВЫВОДЫ ПО ГЛАВЕ 5	50
ЗАКЛЮЧЕНИЕ	51
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	52
ПРИЛОЖЕНИЕ А СКРИПТ ИНИЦИАЛИЗАЦИИ ШЛЕМА VR.....	54
ПРИЛОЖЕНИЕ Б ПРОГРАММА ТРЕНАЖЕРА НА RASPBERRY PI ZERO W...	59

ВВЕДЕНИЕ

Спорт играет большую роль в жизни людей. Он укрепляет здоровье, воспитывает характер, делает человека сильным и выносливым, закаляет организм. Век назад физические качества – выносливость, сила – ценились людьми. Но роль физической силы падает из-за развития техники и снижения уровня агрессии в обществе. По мере развития технологий работа, требовавшая физической силы, переходит к машинам, а оператору машины особая физическая сила уже не нужна. В настоящее время людей тянет заниматься спортом больше для поддержания здоровья или хобби.

Международный олимпийский комитет (ИОС) провел опрос среди спортсменов и других представителей отрасли, по итогам которого выяснилось, что простой из-за пандемии коронавируса вылился для атлетов в потерю мотивации, а для функционеров и организаторов соревнований — в целый комплекс проблем материального характера [1]. Малоподвижность и отсутствие физической нагрузки пагубно сказывается на здоровье. По результатам исследований, такой образ жизни приводит к смерти 5,3 млн. человек ежегодно.

Описание проблемы

За последние годы отрасль тренировок на велотренажерах активно развивалась, а появление разумных тренажеров и программ стало началом трансформации велоспорта изнутри. Это не только жизнеспособная и довольно реальная альтернатива велосипедной езде, которой пользуются любители и профессионалы. Такие тренировки также доставляют удовольствие. Это идеальный вариант для катания в плохую погоду, при ограниченном количестве времени или при необходимости сфокусироваться на тренировке без мыслей о плохих дорогах. Или же когда «за окном» карантин и приходится находиться в самоизоляции [2].

Появилась цель с подвинуть людей заниматься спортом, превратив спорт в доступную из дома соревновательную платформу со множеством пользователей.

Для решения описанной проблемы следует разработать сервис имитации занятия велоспортом с погружением в виртуальную реальность с физическими нагрузками.

Он поможет маломотивированным людям выбрать любую погоду и ландшафт для занятия спортом, собирать статистику занятия велоспортом и увеличить характеристики выносливости [3].

Решение проводить виртуальные соревнования по велоспорту найдут применение в разных категориях велоспорта:

- начальный уровень – позволит скоротать время с пользой для здоровья в красивых местах реального или вымышленного мира.
- средний уровень – позволит организовать любительское соревнование в кругу своих знакомых или небольших районов с соблюдением ряда нормативов, если некие штрафные санкции ограничивают живую встречу, а система принятия решений поможет явно определить победителей.
- высокий уровень – соревновательный. Организация более строгих соревновательных и регулярных соревнований с составлением рейтинга лучших спортсменов. Данное соревнование может увидеть каждый, если кто из тренирующего или поддерживающего состава не сможет поехать с на реальную встречу, а так он может следить за соревнованием.

Вопросы, которые нужно решить с точки зрения спортивной оценки:

- воссоздать реальные условия соревнования в разных категориях;
- составить Нормы, требования и условия их выполнения по виду спорта.

Вопросы, которые нужно решить с точки зрения IT-специалиста:

- выбрать мобильную платформу для приложения;
- выбрать язык программирования;
- выбрать среду разработки;
- исследовать протоколы связи тренажеров с сервером.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. ЦЕЛЬ ДИПЛОМНОЙ РАБОТЫ

Целью дипломной работы является разработка программно-аппаратного комплекса для интерактивного велотренажера. Он позволит подключить большинство старых моделей велотренажеров к ПК, для эмуляции тренировки в открытом мире или трассе.

Для достижения данной цели необходимо сделать устройство, аппаратная часть которого будет на основе микроконтроллера, устанавливающейся на велотренажере и собирающее данные о количестве оборотов и поворотах, сделанных пользователем и передающее их на ПК через интерфейс USB.

С помощью программной части, данные будут преобразовываться и воспроизводиться в игре, разработанной на Unity.

1.2. ОБЗОР ЛИТЕРАТУРЫ

Был проведен анализ наиболее популярных бесплатных или условно бесплатных средств разработки компьютерных игр. Для сравнения движков был выбран электронный ресурс [4], в котором подробно описаны самые популярные средства разработки игр, предоставляющих свою бесплатную версию пользователю. Эти платформы доступны для свободного использования.

В качестве основного языка программирования используется C# [5]. Данные книги [6,7,8] являются хорошим введением в разработку приложений.

Из рассматриваемых платформ особенно выделились Unity и Unreal Engine 4 [9], так как они понятны для использования, в них схожие возможности и они бесплатны, что очень важно для начинающих программистов. Остальные платформы не подошли, так как они уступали Unity и Unreal Engine 4 по сравниваемым параметрам.

Для разработки игрового приложения была выбрана платформа Unity [10]. Данная платформа имеет низкий порог вхождения, большое количество обучающих материалов [11,12,13,] и сообщество разработчиков, вследствие чего, с ней можно быстрее начать работать.

Raspberry Pi часто используется как мозг робота, домашний сервер или просто компьютер. Во книге Саймона Монк «Raspberry Pi. Сборник рецептов. Решение программных и аппаратных задач» [14] содержится свыше 240 полезных рекомендаций и советов по практическому применению Raspberry Pi. Рассматриваются такие вопросы, как настройка компьютера с Linux, написание программ на Python, управление двигателями и датчиками, а также взаимодействие Raspberry Pi с другими электронными устройствами, включая Arduino и проекты IoT (интернет вещей). Опытный разработчик и автор популярных учебных пособий Саймон Монк знакомит читателей с базовыми принципами построения любительского электронного оборудования, которое основано на популярной микроконтроллерной платформе Raspberry Pi, обладающей невероятно большим потенциалом для применения в серьезных коммерческих проектах.

В книге Адитьи Бхаргавой «Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих» [15] приводится множество примеров. Цель книги — не вывалить на читателя кучу невразумительных формул, а упростить наглядное представление этих концепций. Я также считаю, что мы лучше всего учимся тогда, когда можем вспомнить что-то уже известное, а примеры помогают освежить память.

Для создания и редактирования графической составляющей игры будет использоваться графический редактор Adobe Photoshop. Adobe Photoshop [16] — графический редактор от фирмы Adobe Systems. Он работает как с растровыми изображениями, так и имеет некоторые векторные инструменты. Он является лидером рынка в области коммерческих средств редактирования растровых изображений и наиболее известной программой фирмы Adobe.

Одной из самых популярных программ для 3D моделирования считается Blender. Blender 3D – бесплатный программный продукт, предназначенный для создания и редактирования трехмерной графики. Программа распространена на всех популярных платформах, имеет открытый исходный код и доступна совершенно бесплатно всем желающим, а также есть версия на русском языке [17].

1.3. ЭТАПЫ РЕШЕНИЯ ЗАДАЧИ

Для достижения цели выпускной работы необходимо выполнить

следующие этапы:

1. Анализ существующих решений.
2. Выявление достоинств и недостатков существующих устройств.
3. Разработка программно-аппаратной части комплекса:
 - разработка структуры устройства;
 - выбор компонентов схемы;
 - интеграция с компьютерным приложением;
 - тестирование работы комплекса.

1.4. ОБЗОР АНАЛОГОВ

Учитывая нынешний глобальный кризис, все больше и больше велосипедистов обращают свое внимание на дома, поскольку они надеются сохранить форму в условиях самоизоляции и социального дистанцирования. Один из лучших способов сохранить мотивацию в эти нестабильные времена - добавить в свою жизнь виртуальную программу тренировок. Рассмотрим наиболее популярные.

В настоящее время, на рынке не существует таких решений, которые могли бы модернизировать старые модели велотренажеров в устройства управления движениями в виртуальной симуляции без использования дорогостоящего дополнительного оборудования. Рассмотрим похожие решения.

Zwift

Zwift — это игра с турбо-тренажером, которая позволяет вам подключить турбо-тренажер к компьютеру, iPad, iPhone или Apple TV, позволяя вам кататься с другими велосипедистами в виртуальной среде, тем самым помогая облегчить скуку, связанную с катанием в помещении [18].

Помимо соревнований с другими гонщиками в гонщиках Zwift, те, кто ищет конкретные тренировки, могут получить доступ к тренировкам, разработанным профессиональными тренерами, и они могут быть выполнены в группах с гонщиками, выполняющими усилия с одинаковой интенсивностью на основе

процента от их FTP [19]. Из недостатков, данная платформа не бесплатная. Zwift стоит около 15 долларов в месяц. Однако большим недостатком такой системы является цена устройства, а также интерфейс на иностранном языке.

Onelap

Onelap – Китайский аналог Zwift. В Onelap есть возможность заниматься и развлекаться [20]. Onelap создает реалистичную среду с удивительно четкой графикой деталей, великолепными цветами и физической моделью, которая имитирует градиент, ветер и сопротивление качению, вы можете адаптировать каждый аспект своего аватара, велосипеда и маршрутов - независимо от холмистости маршрут, ровный маршрут или горный маршрут, вы не будете ездить на велосипеде в повторяющейся сцене в пределах 100 км пути. Большинство игроков из Азии, но увеличивается и число европейцев. На данный момент игра бесплатна. Недостатком этой системы также является цена устройства и иностранный интерфейс.

RGT Cycling

RGT Cycling — это приложение для проведения тренировок в помещении, которые симулируют реальные велотрассы по всему миру с помощью умного велотренажера. Вместо того чтобы крутить педали в виртуальном мире, вы будете ездить по известным маршрутам, испытывая реальные ощущения [21].

Также существуют различные структурированные тренировки, разработанные тренерами, и вы можете создать свою собственную гонку, загрузив файл маршрута.

Особенность RGT Cycling заключается в том, что во время гонки вы лучше ощущаете реальность по сравнению с некоторыми конкурентами, если вы движетесь впритык за другим гонщиком, а также создается эффект торможения на поворотах. Если вы хотите обогнать своего соперника, вам придется крутить педали быстрее или прикладывать больше усилий.

RGT Cycling утверждает, что данные о стандартной силе в их приложении являются более реалистичными, чем в других симуляторах велогонок, а скорость

исчисляется исходя из реальных данных, и ваше изображение будет автоматически замедляться, чтобы вы не врезались в гонщика, который едет перед вами.

Недостатком такой системы является цена устройства, иностранный интерфейс и обязательная платная подписка.

VZFit

Главным достоинством данного проекта является то, что в нем используется очки виртуальной реальности, которые позволяют в полной мере насладиться процессом игры. Также, хорошим бонусом является разнообразие режимов игры для такого устройства [22].

Минусы же такого проекта, это то, что для его полного использования необходимо:

- платная подписки;
- наличие виртуального шлема “oculus rift” или “oculus go”;
- интерфейс на иностранном языке.

Таблица 1.4.1 – Сравнение приложений велосоревнования

Название	Zwift	Onelap	RGT Cycling	VZFit
Язык	Английский	Китайский, Английский	Английский	Английский
Интерфейсы связи	BLE, ANT+	ANT+	ANT+, BLE	ANT+, BLE
Режим тренировки	+	+	+	+
Режим профессиональный	+	–	+	–
Наличие режима VR	–	–	–	+
Стоимость оборудования, \$	400	400	400	400 + 200
Стоимость услуги, \$	15	0	15	10

Вывод: Сервисы долгое время пользуются популярностью и надёжно закрепились как основные сервисы организации крупных соревнований. Недостатками можно назвать специальные тренажеры, стоимость которых высока и отсутствие многих моделей на территории РФ. Отсутствие русского языка затрудняет понимание интерфейса российскому пользователю.

1.5 АНАЛИЗ И ПОДБОР КОМПОНЕНТОВ АППАРАТНОЙ ЧАСТИ КОМПЛЕКСА

В данном разделе сравним готовое аппаратное решение для велосипедного соревнования в помещении с компонентами для самостоятельной сборки аппаратного комплекса.

Умный велосипедный тренажер Tacx FLUX S

Tacx FLUX S - мощный и очень тихий тренажер с возможностью автономной работы, а также способные подключаться к различным устройствам при помощи технологии Bluetooth или ANT+. Совместимость станка с популярными приложениями значительно расширяет возможности тренировок, делая их интересными и увлекательными: соревнуйтесь с другими спортсменами в Zwift, загружайте GPS треки велотрасс и опробуйте реальные трассы со всего мира не выходя из дома, дополнив их специально записанным видеорядом. Изображение их онлайн магазина показано на рисунке 1.5.1.



Велостанок Tacx Flux S
Smart Trainer

от 68 100 руб.

Рисунок 1.5.1- Велосипедный тренажер Tacx FLUX S Smart Trainer

Особенности Tacx FLUX S Smart Trainer:

- Мощность тренажера до 1500 Вт;

- Имитация подъемов до 10%;
- Погрешность в измерении мощности до 3%;
- Вес маховика 7кг.

Как множество умных тренажеров, стоимость Tacx FLUX S Smart Trainer сопоставима среднему спортивному велосипеду. Такая стоимость тренажеров отпугивает начинающих пользователей и заставляет отказаться от занятий спортом в помещении.

Велосипедный тренажер Deuter MT-04

Тренажер с блоком переднего колеса и быстросъемным шпагатом. Готовое решение для тренировки на собственном велосипеде в комнате. Подходит для велосипедов с диаметром колеса 26 " ~ 28". Имеется проводной контроллер на 6 скоростей магнитного сопротивления. Внешний вид умного тренажёра показан на рисунке 1.5.2.



Рисунок 1.5.2- Велосипедный тренажер Deuter MT-04

Преимущества:

- максимальное снижающий шум магнитный ролик колеса, разработан с теплоотводящим вентилятором;
- легко собрать без дополнительных инструментов, все включено в посылка;
- 6 уровней скорости для переключения передач, регулировка не нужно снимать с велосипеда;
- дешевле всех тренажёров с регулировкой нагрузки;
- прочный материал, максимальная грузоподъемность более 135 кг.

Недостатки – Минимальная стоимость тренажера — 23 000 рублей. За свои возможности производитель требует высокую стоимость.

Составим список компонентов для сборки собственного умного тренажера. В ходе выполнения выпускной квалификационной работы магистра был проведен анализ наиболее подходящих к данной работе микроконтроллеров, главными требованиями к выбору стали небольшая стоимость устройства, небольшой размер, достаточная мощность для считывания всех сигналов без задержки и возможность подключения по Bluetooth Low Energy.

Raspberry Pi Zero W

Одноплатный компьютер с возможностью подключения к беспроводной локальной сети и Bluetooth. Внешний вид показан на рисунке 1.5.3.



Рисунок 1.5.3 – Raspberry Pi Zero W

Raspberry Pi Zero W обладает всеми необходимыми функциональными возможностями:

- 802.11 b/g/n wireless LAN;
- Bluetooth 4.1+LE;
- одноядерный процессор с тактовой частотой 1 ГГц;
- 512 МБ ОЗУ;
- 28 портов GPIO;
- питание от 5 В, 2А.

ESP32-WROVER-E Espressif

ESP32-WROVER-E и ESP32-WROVER-IE — это два мощных универсальных модуля MCU WiFi-BT-BLE, предназначенных для широкого спектра приложений, от сетей датчиков с низким энергопотреблением до самых сложных задач, таких как кодирование голоса, музыка, потоковая передача и декодирование MP3. Внешний вид показан на рисунке 1.5.4.

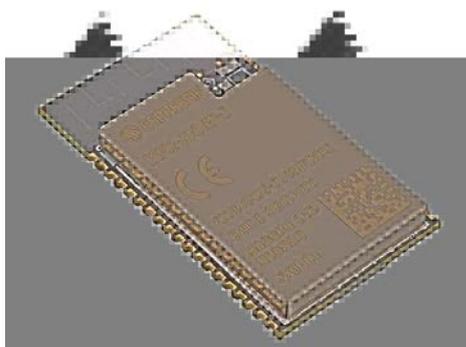


Рисунок 1.5.4 – ESP32-WROVER-E Espressif

ESP32-WROVER-E поставляется с антенной на печатной плате, а ESP32-WROVER-IE - с антенной IPEX. Оба они оснащены внешней флэш-памятью SPI объемом 4 МБ и дополнительной псевдостатической оперативной памятью SPI объемом 8 МБ (PSRAM). Информация в этом техническом описании применима к обоим модулям.

Сравнение двух контроллеров указаны в таблице 1.5.1.

Таблица 1.5.1 – Сравнение микроконтроллеров для комплекса

Характеристики	Raspberry Pi Zero W	ESP32-WROVER-E Espressif
Макс. частота, МГц	1000	240
RAM, Мб	1024	8
ROM, Мб	*	4
I/O выводов	40	38
Цена, руб.	1300	400
Размер, см	3x4	2x3

Вывод: в таблице представлено сравнение микроконтроллеров по основным характеристикам. Из представленных, дорогим, но наиболее подходящим является Raspberry Pi Zero W, так как в соответствии с требованиями размеров и стоимости подходит больше всего.

В умных тренажёрах присутствует динамическое сопротивление, регулируемое специальными программами для тренировок разной интенсивности.

В современных тренажерах нагрузка создается за счет магнитного, воздушного либо гидравлического тормоза. Преимущество в таком тормозе что величина сопротивления прямо пропорциональна поданному напряжению. Из недостатков – высокая цена относительно простых систем торможения.

Интегрированная тормозная система в велосипеде (резиновые колодки, дисковые тормоза) может использоваться как имитатор сопротивления. Преимущество в том, что не нужно его покупать, но подключение к умному тренажеру требует электропривод для регулировки натяжения тормоза.

Универсальным решением послужит двигатель постоянного тока с тахогенератором. Мотор-редуктор можно использовать в качестве тахометра, снимающего через приложенный к колесу вал скорость вращения. А при условии подачи напряжения в обратном направлении движения колеса можно имитировать нагрузку в тренировке.

Выбор двигателя и редуктора

Максимальная сила, с которой велосипедист может давить на педаль, равна его весу. Минимальная сила, понятно, равна нулю. При массе велосипедиста, например, 70 кг, максимальная сила с которой велосипедист может давить на педали, равна $70 \cdot 9,81 = 686,7 \text{ Н}$ [23, 24].

При средней длине педалей 170 мм получим вращающий момент равный:

$$M_{\text{п}} = F \cdot l = 686,7 \cdot 0,17 = 116,73 \text{ Нм}$$

При передаточном соотношении 40 зубьев спереди и 42 сзади получим момент на колесе равный:

$$M = M_{\text{п}} \cdot \frac{42}{40} = 116,73 \cdot \frac{42}{40} = 122,57 \text{ Нм}$$

Вычислим максимальную скорость колеса, исходя из максимального каденса (оборотов в минуту) для среднетренированного человека в 200, и передаточном соотношении 40 зубьев спереди и 11 сзади получим скорость колеса равную:

$$V=K \cdot \frac{42}{11}=200 \cdot \frac{42}{11}=727,27 \text{ об/мин}$$

Теперь выберем редуктор, исходя из полученных данных. Наиболее подходящим является 1ЦУ, с максимальной частотой вращения вала 1000 об/мин, максимальным моментом 850 Нм, передаточным числом равным 4. Тогда двигатель должен иметь частоту вращения не менее 2900 об/мин и момент равный 30 Нм, наиболее подходящим является МПР 102, мощностью 200 Вт, частотой вращения 3000 об/мин, с интегрированным тахогенератором.

Протоколы обмена данными

Готовые решения умных тренажеров имеют беспроводной протокол сопряжения и передачи данных.

ANT+ — технология беспроводной связи, использует нелицензируемый диапазон 2,4 ГГц. Дальность связи приблизительно сопоставима с Bluetooth: спецификация протокола ограничивает ее 30 метрами.

Еще одной полезной особенностью является существенно более низкое энергопотребление. Для смартфона это не столь критично, а вот то, что другие устройства, использующие интерфейс ANT+, могут питаться от плоской батарейки – это куда более важно для разработчиков.

Но ключевым свойством стандарта является его многоканальность. Связь организована по принципу master-slave, и ведущий аппарат способен получать информацию сразу от нескольких ведомых, при этом не мешающих друг другу.

Именно это сделало данный протокол столь привлекательным для создания всевозможного спортивного снаряжения.

В моде есть библиотека, которая прослушивает сигнал беспроводного протокола ANT+ от тренера, совместимого с ANT+ FE-C (Tacx, Wahoo, Elite, Vכוool, Kinetic, Saris и т. Д.), Или измерителя мощности, или беговой дорожки Smart, или стопы. pod, считывает скорость и применяет ее к транспортному средству,

которым управляет ваш персонаж во время игры. Он считывает наклон местности, неровность (материал) и ветер в игре и отправляет всю эту информацию в умный тренажер, чтобы он мог воспроизвести твердость местности, по которой вы на самом деле катаетесь. В таблице 1.5.2 приведено сравнение протокола ANT+ с Bluetooth Low Energy.

Таблица 1.5.2 – Сравнение протоколов беспроводной связи

Технология	ANT/ANT+	Bluetooth Low Energy
Частотный диапазон, ГГц	2.4 – 2.483	2.4 – 2.483
Поддерживаемые сетевые топологии	точка-точка, звезда, кластерное дерево, mesh	точка-точка, звезда
Модуляция	GFSK	GFSK
Ширина канала, МГц	1	2
Протокол	простой	более сложный
Скорость передачи данных, Мбит/с	1	1
Радиус действия, м	50	50
Безопасность	64-битный ключ	128-битный алгоритм шифрования AE
Стоимость, руб.	1000	0 (встроен в контроллер)

Вывод: ANT+ популярный протокол в области умных тренажеров, но довольно дорогой относительно BLE. Но не все сервисы поддерживают умные тренажеры с протоколом BLE.

В условиях хорошо проветриваемых помещений, комфортный теплоотвод за счет конвекции и испарения составляет порядка 585 Вт (при температуре воздуха должна быть 12.8 С). Эта цифра используется при дизайне промышленных цехов.

Учитывая историю о метаболизме, велосипедист выдавал бы в таком цехе максимум 150 Вт на станке. Пределы разумного - 375 Вт [8]. В этом случае ожидается, что весь избыток тепловой энергии, а такой будет порядка 600 Вт поверх охлаждаемых 585 Вт, должен пойти на нагревание. Для велосипедиста массой 70 кг это означает повышение температура тела на 2 градуса в течении 10 минут.

Ни у кого нет желания потренироваться при температуре тела 39 С. Даже при низких скоростях, поток создаваемого воздуха гарантирует охлаждение, которое превышает тепловое рассеивание велосипедиста. Поэтому, рекомендуется использовать вместе с тренажером вентилятор.

Невзирая на неадекватный, с точки зрения физики, воздушный поток, 90 Вт вентилятор создает впечатляющий эффект обдувания. Это вентилятор, если и не остужает, то хотя бы добавляет ощущение комфорта на велотренажере - лужа пота под станком никуда не девается.

Таблица 1.5.3 – Сравнение электровентиляторов разных производителей

Наименование	Уровень шума, дБ	Обдуваемая мощность, Вт	Диаметр, см	Стоимость, руб	Производительность, м3/ч
Electrolux EFF-1004i	48	55	40	3 990	3000
AEG VL 5606 WM	25	100	40	3 290	2500
Rix NPSF-8000		90	45	3 190	3000
Midea FS4543		100	45	2 990	2400

Для вывода краткой информации состояния тренажера установим дисплей SSD1315, изображенный на рисунке 1.5.5.



Рисунок 1.5.5 – Дисплей SSD1315

Для понимания контроллеру аналогового сигнала применим аналогово-цифровой преобразователь ADS1115 изображенный на рисунке 1.5.6.

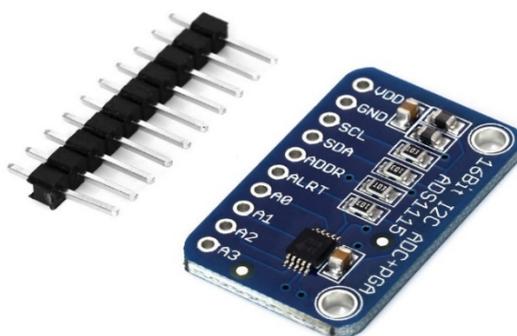


Рисунок 1.5.6 – Аналогово-цифровой преобразователь ADS1115

Для распознавания диаметра колеса велосипеда на тренажер установим датчик расстояния VL53L0X, изображенный на рисунке 1.5.7.

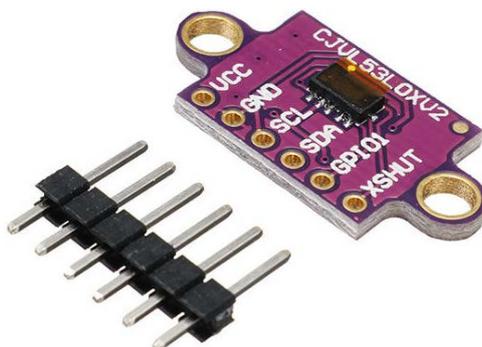


Рисунок 1.5.7 – Модуль расстояний VL53L0X

Для управления торможения мотором-редуктором, были выбраны большие кнопки (12x12 мм), изображённые на рисунке 1.5.8. Кнопка будет установлена на руде и подавать контроллеру сигнал торможения колеса велосипеда.



Рисунок 1.5.8 – Кнопка для тормоза

1.6. ВЫБОР ВИДЕОМОНИРОРА

Виртуальная реальность обещает самые разные вещи, но одно она, безусловно, может сделать эти скучные тренировки в помещении на велосипеде немного более увлекательными.

Погружение в виртуальную реальность (VR) — это ощущение физического присутствия в нефизическом мире. Восприятие создается путем окружения пользователя системы VR изображениями, звуками или другими стимулами, которые создают захватывающую общую среду.

Oculus rift

Основное отличие — есть провод. За счёт подключения к ПК можно использовать шлем в любых VR-играх с отличным качеством картинки. Контроллеры в комплекте те же, что и у Quest — маленькие и удобные, могут долго работать без замены батареек.

Этот шлем — один из наиболее лёгких и удобных в эксплуатации, однако у него физически нельзя отрегулировать расстояние между глазами — только через программу.

Плюсы: приемлемая цена, удобство и простота использования, хороший внутренний inside-out-трекинг, для которого не нужны базовые станции.

Минусы: нельзя отрегулировать расстояние между глазами. Нет официального российского представительства и сервиса в РФ.

Sony HMZ-T1

Это не шлем виртуальной реальности, как может показаться с первого взгляда, но и не простые 3D-очки – возможно правильнее было бы назвать его «стерео-шлем». Стереозэффект достигается за счет использования двух OLED-дисплеев для каждого глаза в отдельности. При такой схеме нет никаких

перекрестных наложений, всякого рода помех и искажений картинка. Картинка должна быть идеально четкой. Идея такой компоновки далеко не нова. Ей десяток-другой лет. Но вот в таком качестве, серийном производстве и началом мировых продаж всех опередила Sony.

Silico MicroDisplay ST1080

Компания Silicon Micro Display разработала портативный дисплей в виде очков. Модель ST1080 поддерживает контент в 2D- и 3D-формате с разрешением Full HD 1080p. Пользователь видит изображение эквивалентное экрану с диагональю 100 дюймов, удаленном на расстояние 3 м. Кроме того, очки обладают прозрачностью 10%, что позволяет использовать их в системах дополненной реальности. Девайс получает видеосигнал через HDMI-кабель, а питание – через USB. Вес модели – 180 г. Цена Silicon Micro Display ST1080 составляет \$800.

Таблица 1.6.1 – Сравнение шлемов виртуальной реальности

Характеристики	Oculus Rift	Sony HMZ-T1	Silico MicroDisplay ST1080
Разрешение	1920×1080	1280×720	1920×1080
Тип экрана	LCD	OLED	LCoS
Разъем для подключения	DVI/HDMI	HDMI 1.4	HDMI 1.4
Способ подачи 3D сигнала	Отдельные экраны для каждого глаза	Frame Packaging	Frame Packaging
Уровень обзора, Градусов по диагонали	110	45	45
Отслеживание движения головы	Да	Нет	Нет

Xiaomi Mi VR и Trinus VR

Многие люди после использования очков виртуальной реальности могут испытывать головокружение. Чтобы избежать этого, Xiaomi использует сам телефон в качестве аппаратного ускорителя, что позволяет повысить чувствительность в 17 раз. Из-за того, что даже малейшее движение головы

улавливается датчиками, достигается полная синхронизация движений головы с видимой виртуальной реальностью. Таким образом можно забыть о головной боли или о каких-либо других неудобствах при использовании очков в течении долгого времени.

TrinusVR - программа для стриминга. Тринус позволяет запускать большинство 3D приложений и игр. На рисунке 1.6.1 показаны способы подключения смартфона в качестве шлема виртуальной реальности.

Для использования стриминга вам необходимы следующие условия:

- VR гарнитура Xiaomi Mi VR или аналог;
- персональный компьютер;
- смартфон с наличием гироскопа, акселерометра и магнитометра;
- быстрое и стабильное Wi-Fi соединение;
- Microsoft .NET;
- сервер, установленный на ПК;
- клиент, установленный на смартфон.

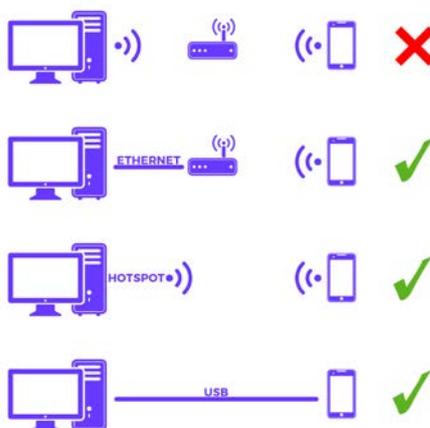


Рисунок 1.6.1- Способы сопряжения смартфона с ПК через TrinusVR

1.7 ВЫБОР ИГРОВОГО ДВИЖКА

Для реализации программной части комплекса, необходимо выбрать игровой движок, на котором будет сделана игра. Игровой движок – это модуль игры, который включает в себя игровую логику. Процесс разработки приложения сильно облегчается за счёт экономии времени и сил, посредством встроенных инструментов. В настоящее время существует огромное количество таких средств.

Для сравнения, были выбраны наиболее популярные среды разработки 3D приложений.

Unity3D

Unity 3D — это мощная среда для разработки 3D игр и приложений. Данная платформа создана в 2005 году. Главный плюс Unity 3D это простота разработки приложений. В данной среде разрабатывается огромное количество игр под различные платформы.

Одним из главных преимуществ использования платформы Unity является ее подробная документация, с описанием всех функциональных возможностей, а также как их правильно применить.

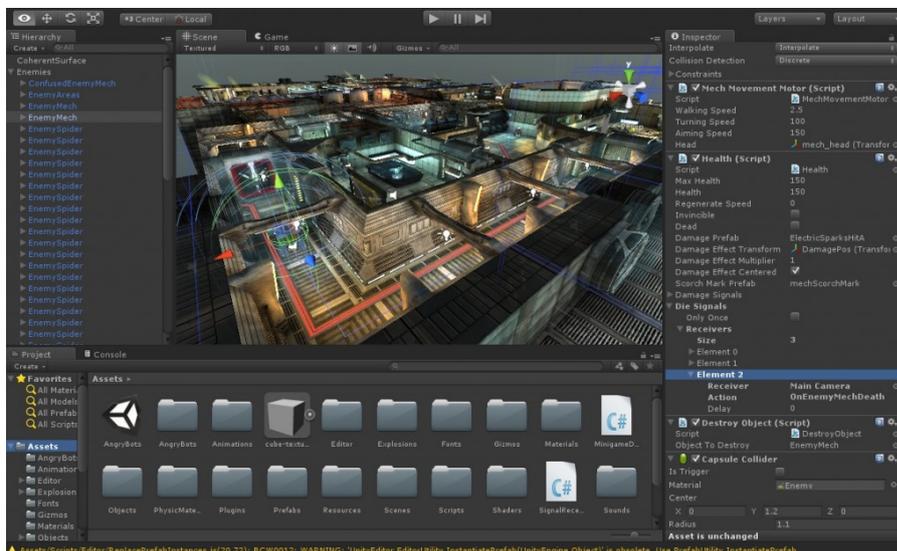


Рисунок 1.6.2 - Интерфейс платформы Unity

Основные возможности и преимущества Unity 3D:

- доступный и понятный интерфейс;
- поддержка двух языков программирования: C# и JavaScript, на которых пишутся скрипты;
- большое сообщество;
- поддержка перетягивания объектов в редакторе;
- возможность дополнения функционала;
- возможность использования систем контроля версий.

Недостатки:

- ограниченный набор инструментов;

– не самая лучшая графика по сравнению с аналогами.

Unreal Engine

Unreal Engine 4 – среда разработки, созданная Epic Games. Unreal Engine 4 – самая популярная среда разработки для создания фильмов и AAA-проектов. Данная платформа обладает высокими графическими возможностями. С Unreal Engine 4 есть возможность разрабатывать игры под PC, Mac, консоли, IOS, Android. В отличие от Unity, UE4 имеет мощный инструмент для дизайна игровых уровней прямо в сцене, достаточно удобную систему Blueprint, не имеющую аналогов, красивый дизайн самой платформы и интуитивность в использовании. Из всех сред разработки, Unreal Engine 4 является самым инновационным. Он сочетает в себе высокую производительность, лучшую графику, простой язык программирования и удобность в использовании.

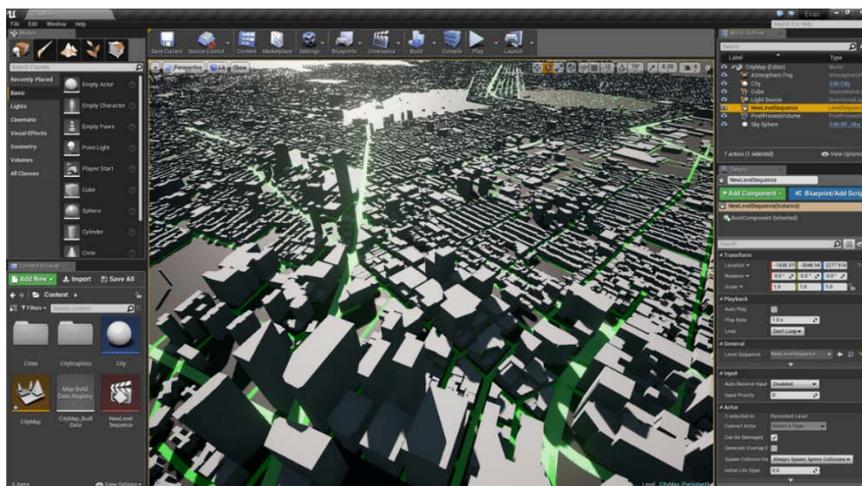


Рисунок 1.6.3 - Интерфейс платформы Unreal Engine

Преимущества:

- большое сообщество;
- возможно, напрямую использовать в проекте файлы с исходным кодом на C++;
- широкий ассортимент инструментов для различных целей;
- совместим с различными платформами.

Недостатки:

- сложно привыкнуть к определенным инструментам;
- небольшой выбор готовых инструментов в официальном магазине;

При выборе средств разработки наиболее важными критериями были:

- порог вхождения;
- поддерживаемые платформы и используемые языки программирования;
- стоимость использования;
- исходный код.

Данные игровые движки схожи по функционалу, они бесплатны, имеют хорошую документацию и поддержку, но среда Unity имеет менее сложный язык проектирования. Таким образом, был выбран игровой движок Unity3D.

1.8 ВЫВОДЫ ПО ГЛАВЕ 1

В ходе анализа предметной области был проведен обзор литературы, которая поможет в разработке работы.

Были рассмотрены проекты, схожие по назначению к разрабатываемую комплексу. Выявлены их достоинства и недостатки.

Были рассмотрены готовые аппаратные решения, схожие по назначению к разрабатываемую комплексу. Выявлены их достоинства и недостатки.

Были рассмотрены устройства сбора и передачи данных. Сравнивая Raspberry Pi Zero, ESP32-WROVER-E Espressif, выбор остановился на Raspberry Pi Zero, так как по необходимым критериям, а это, небольшая стоимость устройства, небольшой размер, достаточная мощность для считывания всех сигналов без задержки и возможность подключения по Bluetooth BLE. Поэтому он подходит больше.

Для считывания данных о движениях колеса, были рассчитаны параметры редуктора и тахогенератора для тренажера.

Для теплового рассеивания велосипедиста в аппаратный комплекс рекомендуется использовать вместе с тренажером вентилятор. Из предложенных вариантов выбран Vionaire ВАС 14 из-за оптимальных характеристик мощности и шума за свою стоимость.

Для подробной индикации состояния был задействован дисплей SSD1315. Для корректного расчёта скорости для собственного велосипеда, был подобран лазерный дальномер VL53L0X.

В качестве средства вывода изображения было выбрано Xiaomi Mi VR в комбинации с приложением Trinus VR. Данное программное обеспечение позволяет видеть смартфон в локальной сети как реальный VR шлем.

Были рассмотрены средства разработки игры, а именно Unity и Unreal Engine 4, они понятны для использования, в них схожие возможности и они бесплатны, что очень важно для программистов. В качестве платформы для разработки игры будет использоваться Unity, так как был небольшой опыт использования данного игрового движка, и он подходит для достижения поставленной цели.

2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ К КОМПЛЕКСУ

Для реализации данной системы необходим следующий набор подсистем:

- приложение на Windows OS. Приложение обеспечивает пользователю симуляцию передвижения на велосипеде;
- графический интерфейс приложения;
- аппаратная реализация системы. Устройство, позволяющее передавать данные о движениях пользователя непосредственно в приложение;
- велотренажер, крепления которого, позволят установить устройство.

2.1 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

- считывание устройством данных о движениях пользователя;
- передача данных устройством о количестве оборотов, сделанных пользователем;
- передача данных устройством о нажатых кнопках направления движения;
- обработка данных, переданных устройством на ПК;
- воспроизведение действий пользователя на виртуальной модели в приложении.

2.2 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

2.2.1 ТРЕБОВАНИЯ АППАРАТНОЙ ЧАСТИ КОМПЛЕКСА

- задержка передачи данных от устройства должна быть минимальной (не более 200мс);
- размеры устройства должны превышать следующих параметров: 20*35*35мм;
- вес устройства не должен превышать 500 грамм.

2.2.2 ТРЕБОВАНИЯ ПРОГРАММНОЙ ЧАСТИ КОМПЛЕКСА

- обработка данных для воспроизведения в приложении не должна быть заметна пользователю;
- изменение пользователем настроек разрешения экрана в приложении;
- возможность начать игру;
- изменение уровня громкости в настройках игры;
- возможность зайти в настройки из начатой игры.

2.2.3 ТРЕБОВАНИЯ К ДОКУМЕНТАЦИИ

Пользовательский интерфейс должен быть на русском языке.

2.2.4 ТРЕБОВАНИЯ К ДОКУМЕНТАЦИИ

В документации на устройство должны содержаться технические характеристики устройства, которые включают следующие требования: вес, размеры, а также инструкцию для пользователя, как правильно установить систему.

2.3 ВЫВОДЫ ПО ГЛАВЕ 2

В ходе определения требований к программно-аппаратному комплексу, были выделены общие, функциональные, нефункциональные, лингвистические, а также требования к документации. Именно по ним и будет выстраиваться дальнейшая работа.

3 ПРОЕКТИРОВАНИЕ

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.

3.1 ФУНКЦИОНАЛЬНЫЙ СОСТАВ

Для описания функционального состава системы можно представить функциональную схему. Данная схема поясняет отдельные виды процессов, протекающих в целостных функциональных блоках. На рисунке 3.1.1 представлена функциональная схема системы.

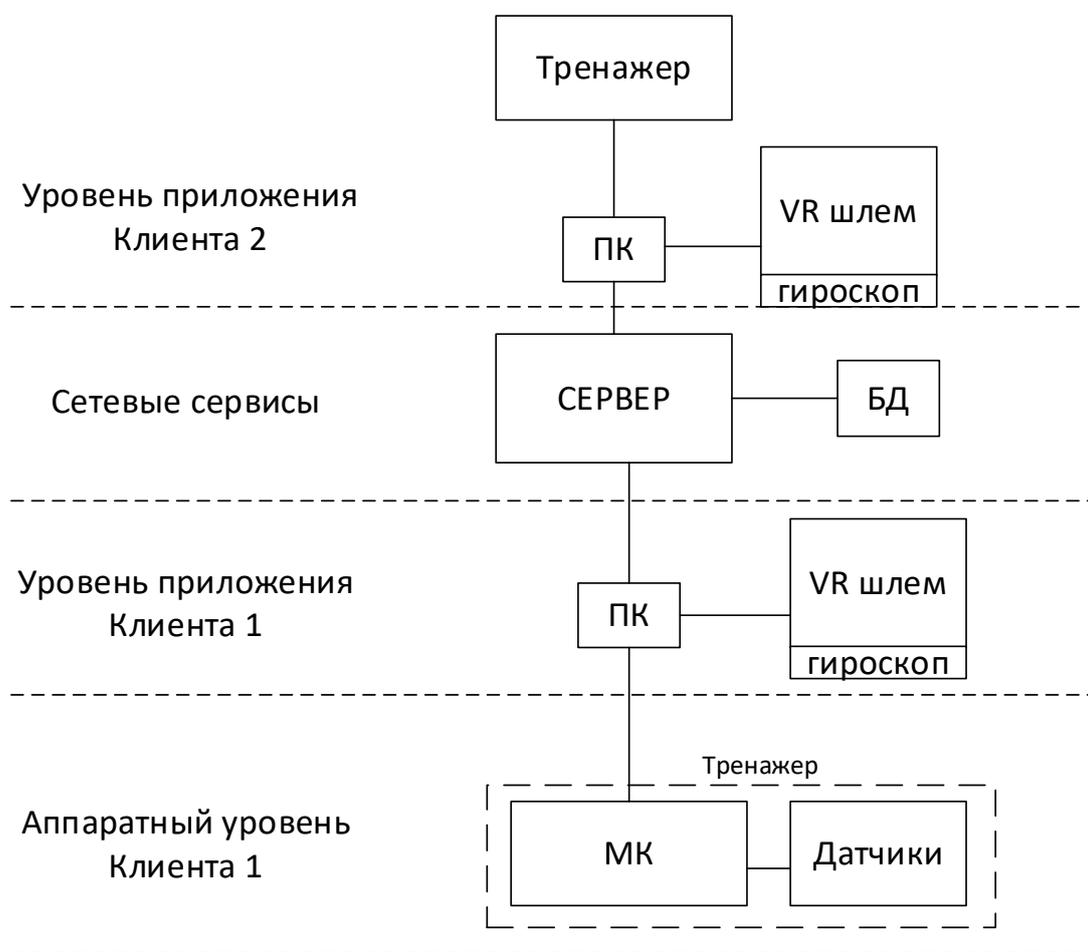


Рисунок 3.1.1 — Функциональная схема системы

Из функциональной схемы видно, что установленные на велотренажер датчики, взаимодействуют с микроконтроллером, который будет передавать

сигналы от них, непосредственно на ПК. На рисунке 3.1.2 отображена схема взаимодействия объектов системы.

Приложение получает эти данные от персонального компьютера и симулирует на виртуальной модели сигналы, полученные от пользователя.

Приложение взаимодействует со шлемом виртуальной реальности, получая от неё значения гироскопа и передавая VR шлему видеоизображение приложения.

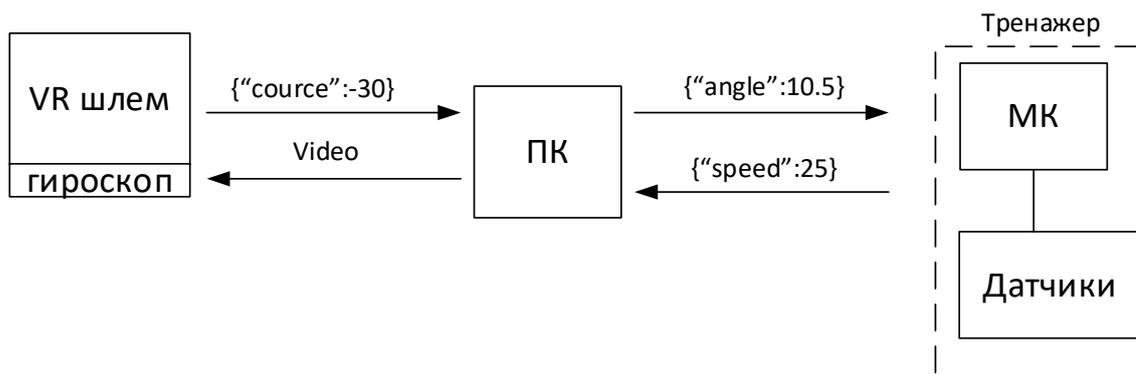


Рисунок 3.1.2 – Схема взаимодействия объектов системы

Клиентское приложение обменивается с сервером показателями пользователя для игровой сессии. Так же сервер хранит статистику о всех пользователях и предоставляет доступ к этой информации по запросу прогресса пользователем.

3.2 ОБЩАЯ СТРУКТУРА АППАРАТНОЙ ЧАСТИ

Аппаратное обеспечение состоит из различных электронных элементов. Далее на рисунке 3.2.1 представлена структурная схема велотренажёра. Управлением всей системы занимается персональный компьютер (ПК), управляющей сигнал в виде задания скорости, момента и направления движения передаётся микроконтроллер (МК), который осуществляет управление блоком преобразования (БП), БП в свою очередь осуществляет управляет исполнительным органом (ИО), представленным двигателем и редуктором. Датчик тока (ДТ) и датчик скорости (ДС) используются как обратная связь для образования замкнутого контура регулирования. Сигналы с сенсоров (С) через МК передаются в ПК для обработки текущей скорости. Аккумуляторная батарея (АКБ), инвертор

(И) используются чтобы накопить и преобразовать энергию педалирования в полезную энергию на нагрузке (Н).

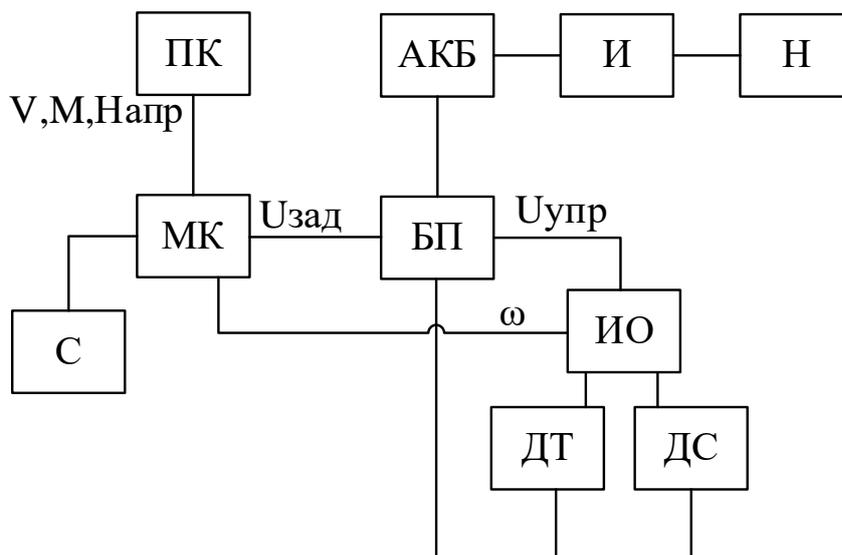


Рисунок 3.2.1 – Структурная схема велотренажера

На рисунке 3.2.2 показана функциональная схема велотренажера. Задающее напряжение $U_{зад}$, поступающее с МК передается в задатчик интенсивности (ЗИ), формирующий темп изменения скорости. Сигналы обратной связи по току ($K_{ост}$) с датчика тока (ДТ) и скорости ($K_{осc}$) с тахогенератора передаются на пропорциональные регуляторы скорости (РС) и тока (РТ), образующих двухконтурную систему подчинённого регулирования с внешним контуром скорости. Особенность данного регулирования заключается в том, что поддержание скорости двигателя дополнительно регулируется и током это значит, что токи не будут выходить за заданные пределы и всегда находиться около заданных значений, избегая перегрева двигателя. Сигналы с регуляторов передаются на блок ШИМ, который осуществляет коммутацию IGBT-транзисторов на работу от сети в режиме торможения или имитации нагрузки, а также зарядку АКБ в режиме педалирования.

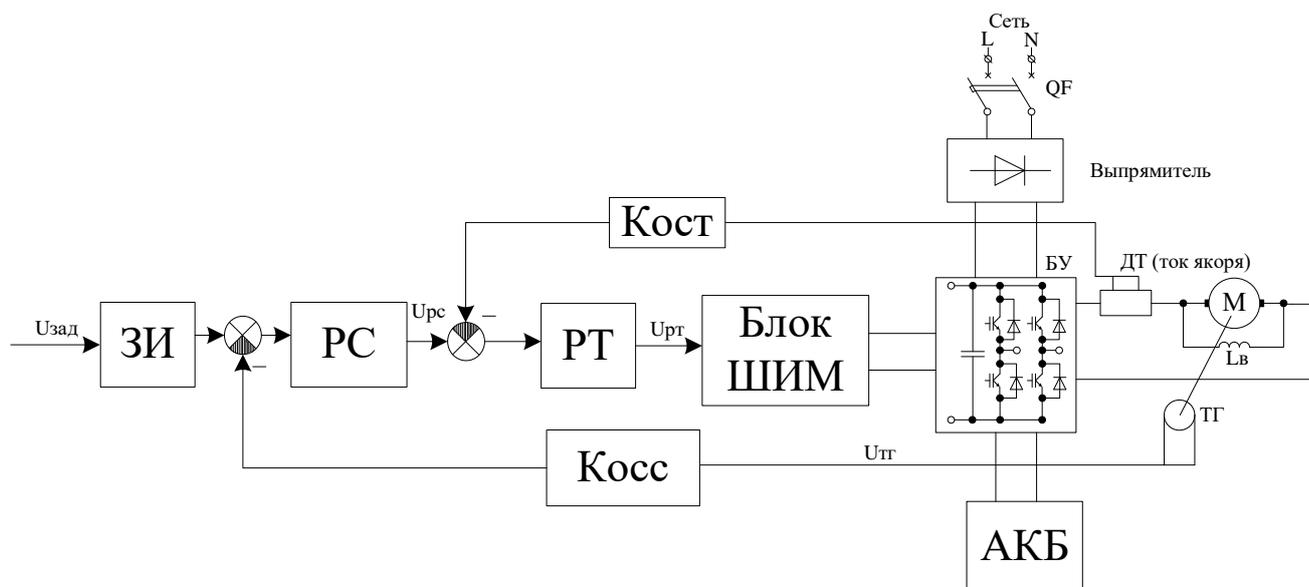


Рисунок 3.2.2 – Функциональная схема велотренажера

3.3 СТРУКТУРНЫЙ СОСТАВ ПРОГРАММНОЙ ЧАСТИ

Для описания структурного состава программной части системы, необходимо построить структурную схему.

Структурная схема – это совокупность элементарных звеньев объекта и связей между ними. Под элементарным звеном подразумевается часть объекта, которая реализует элементарную функцию.

Внутренние компоненты отвечают за логику функционирования и хранение данных, а внешние за визуальное взаимодействие с системой.

Структурная схема системы представлена на рисунке 3.3.3.

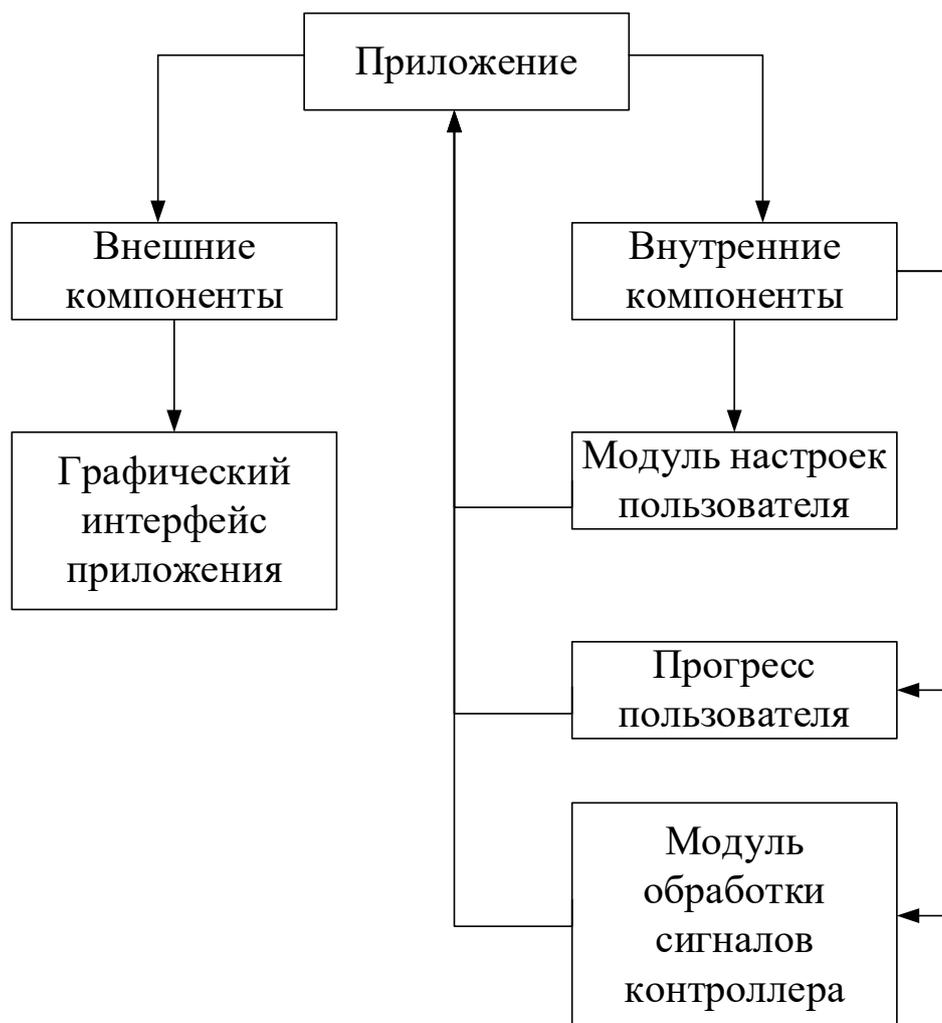


Рисунок 3.3.3 – Структурная схема приложения

Структурная схема приложения включает в себя внутренние и внешние компоненты. Состав внешних компонентов: графический интерфейс приложения. Во внутренние компоненты входят: модуль настроек пользователя, прогресс пользователя и модуль обработки сигналов от контроллера.

3.4 ДИАГРАММА ПРЕЦЕДЕНТОВ ПРИЛОЖЕНИЯ

Диаграмма прецедентов UML, отражающая отношения между актёрами и прецедентами, и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Прецедент — возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный

способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе.

Сформируем модель вариантов использования разрабатываемой системы.

Пользователь взаимодействует с данной системой. Он, находясь в главном меню, может начать игру, просмотреть статистику о себе, заходить в меню настроек и выйти из игры.

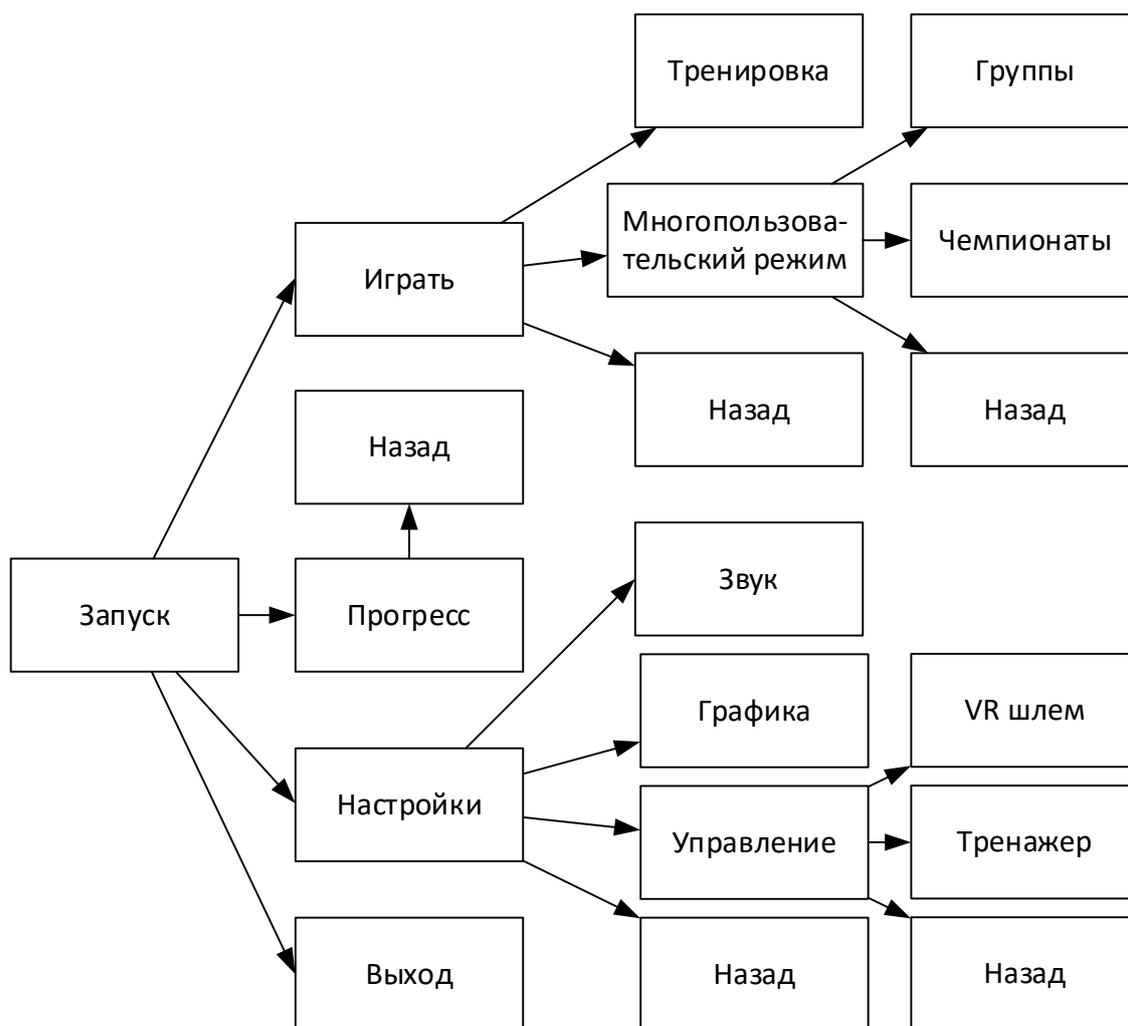


Рисунок 3.4.1 – Диаграмма прецедентов приложения

3.4.1 Прецедент «играть»

Позволяет пользователю перейти в подменю выбора режима игры.

3.4.2 Прецедент «Прогресс»

Позволяет пользователю перейти в подменю прогресса игры.

3.4.3 Прецедент «Настройки»

Позволяет пользователю перейти в меню настроек.

3.4.4 Прецедент «Выход»

Позволяет пользователю выйти из приложения.

3.4.5 Прецедент «Тренировка»

Позволяет пользователю начать управлять велосипедом посредством

3.4.6 Прецедент «Многопользовательский режим»

Позволяет пользователю перейти в подменю выбора многопользовательского режима игры.

3.4.7 Прецедент «Назад»

Позволяет пользователю вернуться в главное меню. Этот вариант использования начинается, когда игрок нажимает кнопку «Назад».

3.4.8 Прецедент «Звук»

Позволяет пользователю перейти в подраздел настройки «Звук».

3.4.9 Прецедент «Графика»

Позволяет пользователю перейти в подраздел настройки «Графика».

3.4.10 Прецедент «Управление»

Позволяет пользователю перейти в подраздел настройки «управление».

3.4.11 Прецедент «Группы»

Позволяет пользователю подключиться к созданным или самостоятельно организованным соревнованиям.

3.4.12 Прецедент «Чемпионаты»

Позволяет пользователю присоединиться глобальным соревнованиям.

3.4.13 Прецедент «VR шлем»

Позволяет пользователю подключить VR шлем.

3.4.14 Прецедент «Тренажер»

Позволяет пользователю найти и подключить тренажер.

3.5 КАЛИБРОВКА ТРЕНАЖЕРА

Для передачи корректных показаний тренажера с установленным велосипедом необходимо производить калибровку аппаратного комплекса под действующий велосипед.

В зависимости от диаметра колеса человек может иметь разные скоростные характеристики. Перед каждым запуском тренажера датчик расстояния будет измерять расстояние от себя до камеры колеса.

Корпус тренажера и положение датчика расстояния неизменяемое и расстояние от центра установки колеса до датчика равно 381 мм (15”).

При условии расстояния от датчика до покрышки колеса, равным 50,8 мм (2”) диаметр колеса вычисляется таким образом:

$$\text{Диаметр колеса} = (381 \text{ мм} - 50,8) * 2 = 660,4 \text{ (26”)}$$

В последствии из тахогенератора и диаметра колеса тренажер передаёт достоверную скорость велосипеда с используемыми размерами колёс. Вычисление скорости велосипеда вычисляется следующим образом:

$$\text{Скорость [км/ч]} = ((\text{скорость вращения колеса [об.мин]} / 60) * \pi * (\text{диаметр колеса [мм]} / 1000000))$$

3.6 ВЫВОДЫ ПО ГЛАВЕ 3

В ходе проектирования комплекса был описан его функциональный состав, показаны структуры аппаратной и программных частей, а также разработана UML-диаграмма вариантов использования из главного меню.

4 РЕАЛИЗАЦИЯ

Для разработки игрового приложения была выбрана платформа Unity. Данная платформа имеет низкий порог вхождения, большое количество обучающих материалов и сообщество разработчиков, в следствии чего с ней можно быстрее начать работать. Для создания и редактирования графической составляющей игры использовался графический редактор Adobe Photoshop.

Для разработки 3D-моделей и анимации использовался Blender 3D. Более сложные 3D модели можно найти в открытом доступе в меню “Assets Store”.

Для программирования микроконтроллера использовался язык Python со средой разработки PyCharm Professional с функцией “SSH Interpretator”.

4.1 РЕАЛИЗАЦИЯ ПРОГРАММНОЙ ЧАСТИ

4.1.1 ФАЙЛОВАЯ СТРУКТУРА ПРИЛОЖЕНИЯ

Разработанный проект содержит каталоги, в которых хранятся:

- 3D-модели;
- игровые сцены;
- скрипты;
- музыка;
- шрифты;
- шаблоны объектов;
- анимации объектов.

Файловая структура представлена на рисунке 4.1.1.1.

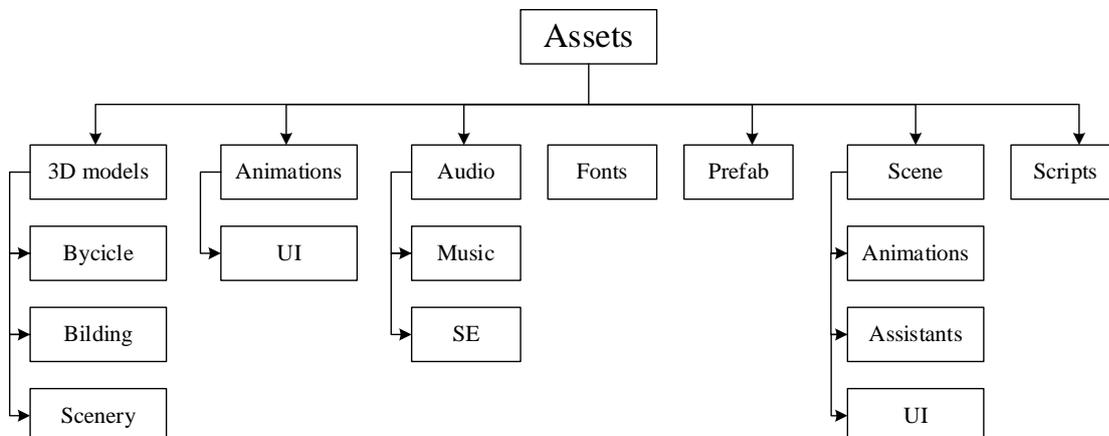


Рисунок 4.1.1.1 – Файловая структура проекта

В папке “Animations” находятся анимации всех объектов игры. В папке “Audio” находятся вся музыка и звуки для игры. В директории “Prefab”» находятся готовые шаблоны игровых объектов. Директория “Scene” содержит сцену игры, в которой происходят все действия. В папке “Scripts” находятся скрипты с описанием всех классов и взаимодействий.

В каталоге “Scripts” содержатся три подкаталога:

- “Animations” содержит в себе скрипты, управляющие анимацией в игре;
- “Assistants” содержит в себе скрипты-помощники, которые управляют логикой всей игры;
- “UI” Содержит в себе скрипты, управляющие графическим интерфейсом.

В каталоге “3D models” содержатся три подкаталога:

- “Bicycle” содержит в себе 3D-модели различных велосипедов;
- “Buildings” содержит в себе 3D-модели всех зданий присутствующих в игре;
- “Scenery” содержит в себе 3D-модели всех второстепенных объектов, таких как столбы, скамейки, деревья и т.д.

4.1.2 РЕАЛИЗАЦИЯ ИНТЕРФЕЙСОВ

В ходе создания игры для взаимодействия пользователя с программой было создано главное меню, показанное на рисунке 4.1.2.1, которое включает в себя следующие пункты:

- играть;
- прогресс;
- настройки;
- выход.

В приложении А показаны фрагменты исходного кода. Полный проект выложен на сервис GitHub [25].

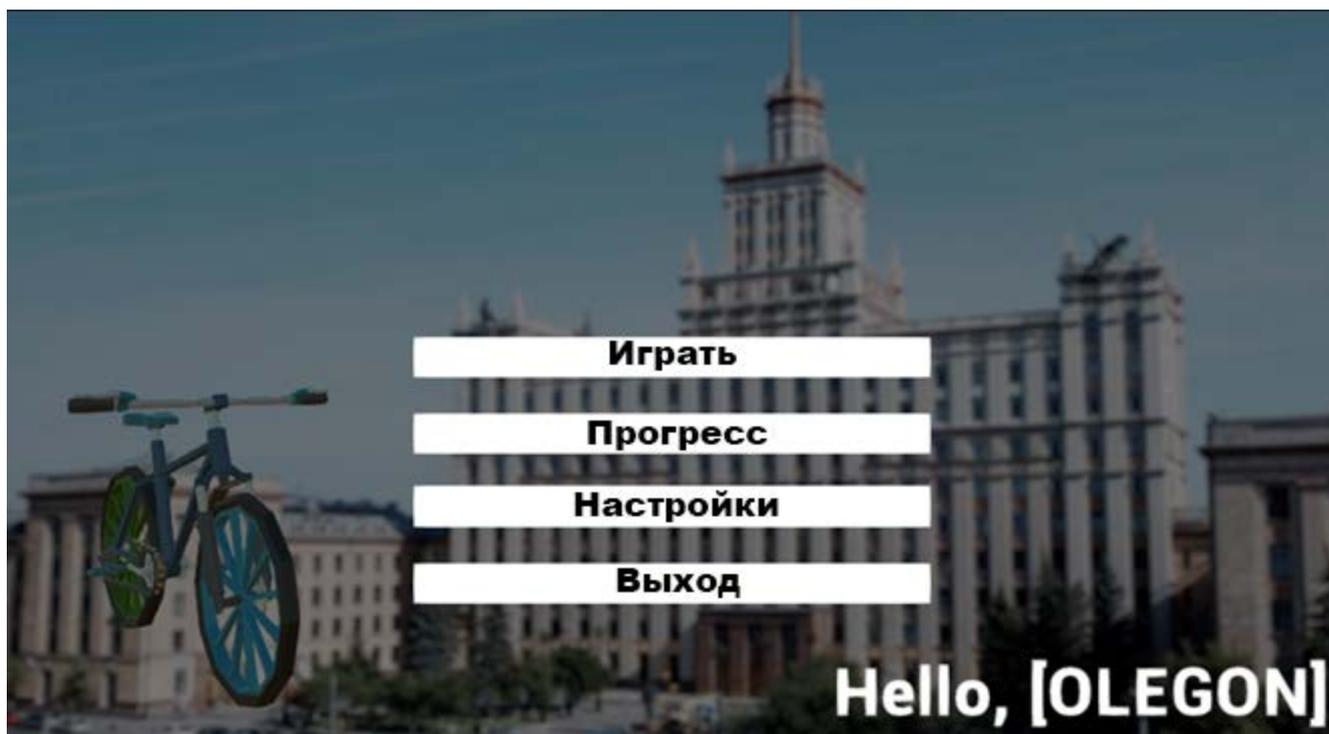


Рисунок 4.1.2.1 - Главное меню

После нажатия на пункт настройки, пользователь попадает в подменю, в котором можно выбрать настройки игры, управления и графики. Они представлены на рисунках 4.1.2.2-4.1.2.4.

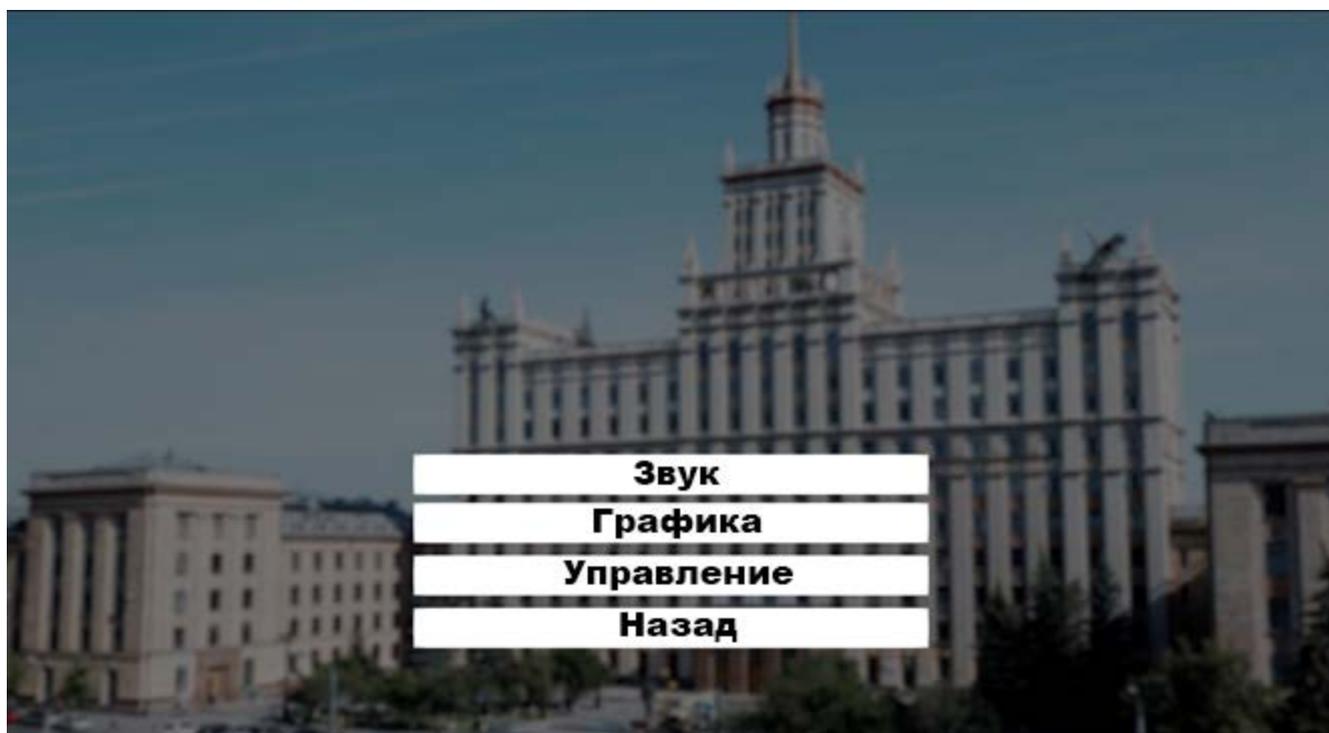


Рисунок 4.1.2.2 – Настройки игры

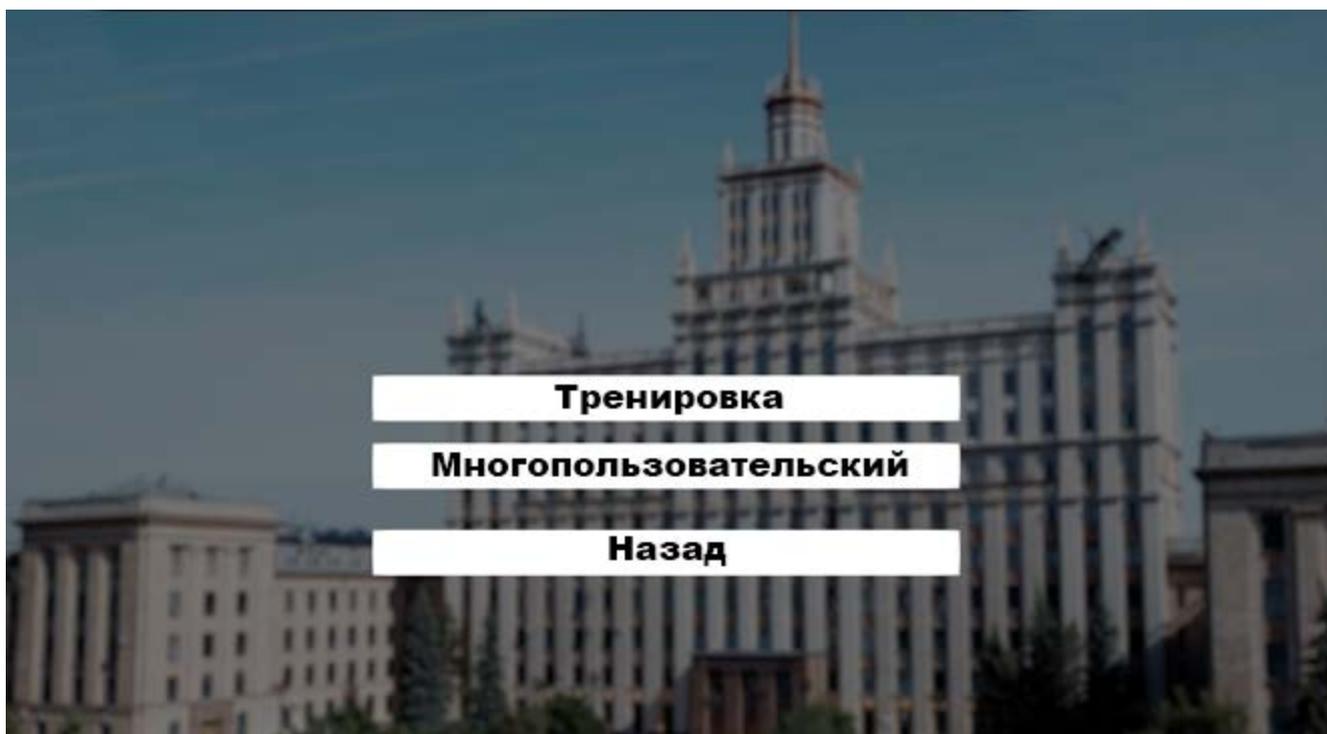


Рисунок 4.1.2.3 – Режимы игры



Рисунок 4.1.2.4 – Настройки управления

При выборе пункта прогресса игры, пользователь может увидеть свои общие результаты, а также результаты за последнюю игровую сессию. Они представлены на рисунке 4.1.2.5.



Рисунок 4.1.2.5 – Общий прогресс игры

При выборе пункта настроек VR шлема, пользователь может добавить свой смартфон с приложением Trinus VR в качестве шлема виртуальной реальности. Скриншоты настроек представлены на рисунках 4.1.2.6 и 4.1.2.7.

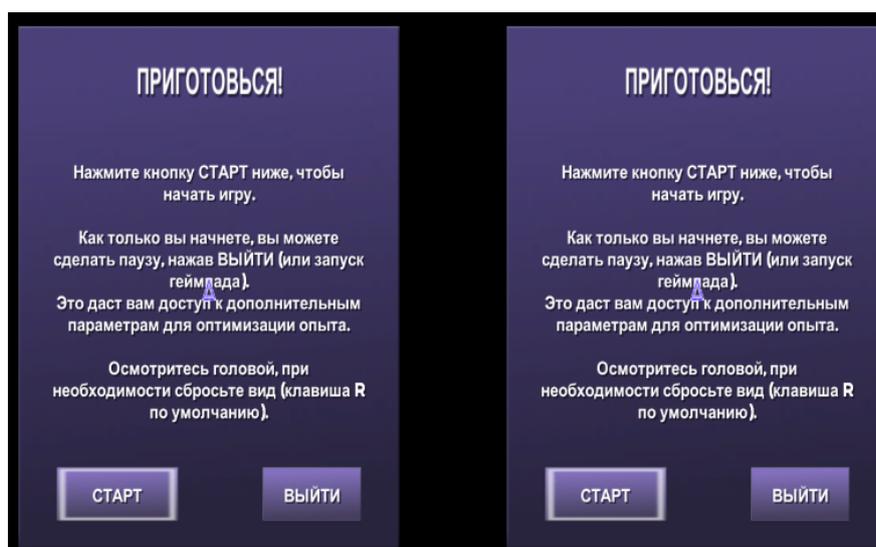


Рисунок 4.12.6 – Включение VR

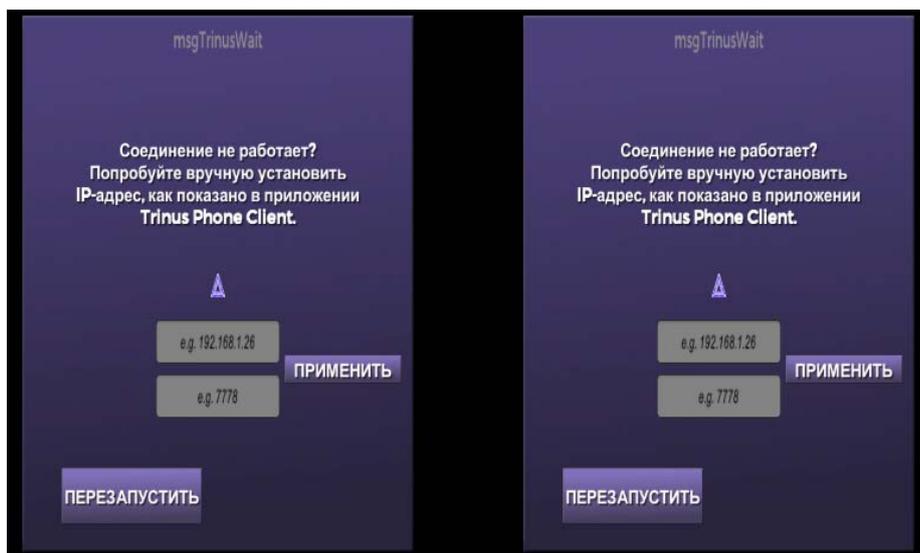


Рисунок 4.1.2.7 – Подключение смартфона

На рисунке 4.1.2.8, показано, как после нажатия на пункт «Тренировка», перед пользователем появляется экран загрузки, после которого появился отсчёт до старта, он перейдет непосредственно к игре.

При запуске игры, пользователь видит перед собой трассу, модель велосипеда, которым может управлять через подключенный велотренажер посредством BLE интерфейса. Также, в левом верхнем углу изображена панель скорости, которая показывает системное время, скорость в км/ч и расстояние в километрах. Процессы изменения этих показателей отображены на рисунке 4.1.2.8.



Рисунок 4.1.2.8 – Тренировка в VR режиме

4.2 РЕАЛИЗАЦИЯ АППАРАТНОЙ ЧАСТИ

Для проверки работоспособности устройства, был собран опытный образец. Его реализация демонстрируется на рисунках 4.2.1. По окончании разработки аппаратной части в контроллер Raspberry PI Zero W записана программа и установлена на автозапуск при включении. Исходный код указан в приложении Б.

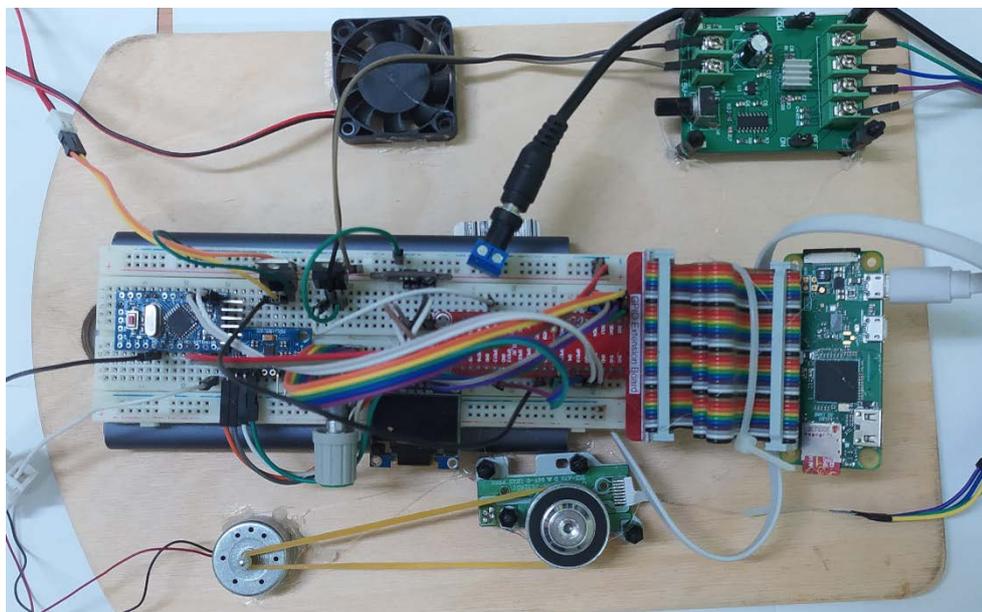


Рисунок 4.2.1 – Стенд имитации тренажера

После прокрутки педали вар передаёт вращательный момент тахогенератору. Тахогенератор преобразует механическое вращение в электрический сигнал. Чем больше скорость прокрутки педалей, тем выше напряжение на выходе тахогенератора. Аналоговый сигнал поступает в АЦП и контроллер получает цифровой сигнал с тахогенератора.

Для управления поворотами, были подключен гироскоп три6050, при наклоне по оси Y значения меняются от +10 до -10. Уровень наклона передаётся приложению и меняет угол поворота велосипеда. При нажатии кнопки тормоза контроллер переключает транзисторные ключи мотора-редуктора в положение короткого замыкания, в следствие чего мотор резко останавливает вращение колеса.

4.3 ВЫВОДЫ ПО ГЛАВЕ 4

Спроектированные ранее устройство и игра, были успешно реализованы. Выполнена возможность управления велотренажером в разработанной компьютерной игре. Смоделированы модели города, внутриигровых объектов, а также велосипеда. Для проверки корректной работы игры, необходимо тестирование.

5 ТЕСТИРОВАНИЕ

5.1 ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ

При тестировании приложения использовался метод функционального тестирования.

Функциональное тестирование – это один из видов тестирования, направленного на проверку соответствий функциональных требований ПО к его реальным характеристикам.

Основной его задачей является подтверждение того, что разрабатываемый программный продукт обладает всем функционалом, требуемым заказчиком.

5.2 ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЕ

Результаты тестирования:

Тест №1. Передвижение пользователя в игре.

Входные данные: пользователь крутит педали на велотренажере.

Ожидаемый результат: пользователь крутит педали на велотренажере.

Полученный результат: совпадает с ожидаемым.

Тест пройден успешно.

Тест №2. Пользователь наклоном головы меняет направление движения на велотренажере.

Входные данные: Пользователь наклоном головы меняет направление движения на велотренажере.

Ожидаемый результат: модель велосипеда в игре поворачивает в нужное направление.

Полученный результат: совпадает с ожидаемым.

Тест пройден успешно.

Тест №3. Выход из игры.

Входные данные: пользователь находится в главном меню.

Ожидаемый результат: пользователь нажимает кнопку выхода из игры, и она закрывается.

Полученный результат: совпадает с ожидаемым.

Тест пройден успешно.

Тест №4. Обновление данных о прогрессе игры.

Входные данные: пользователь находится в игре.

Ожидаемый результат: после преодоления некоторого расстояния, а затем последующего перехода в статистику «прогресса игры», данные, отображенные там, станут другими.

Полученный результат: совпадает с ожидаемым.

Тест пройден успешно.

5.3 ВЫВОДЫ ПО ГЛАВЕ 5

Было проведено функциональное тестирование, чтобы убедиться в правильной работе компонентов приложения, а также правильной работы аппаратной части. Для каждого теста был описан ожидаемый результат и шаги тестирования. Все проводимые тесты были успешно пройдены.

ЗАКЛЮЧЕНИЕ

В рамках дипломной работы был спроектирован и разработан программно-аппаратный комплекс для интерактивного велотренажера.

Программная часть была реализована на платформе Unity, аппаратная часть была основана на одноплатном компьютере Raspberry PI Zero W.

Для достижения этих целей, нужно было выполнить следующие задачи:

1. Проанализировать существующие решения.
2. Выявить достоинства и недостатки существующих устройств.
3. Разработать программно-аппаратную часть комплекса:
 - разработать структуру устройства;
 - выбрать компоненты схемы;
 - интегрировать с компьютерным приложением;
 - протестировать работу комплекса.

В результате выпускной квалификационной работы был разработан комплекс, с помощью которого, пользователь может модернизировать свой велотренажер и заниматься на нем играя в приложение.

В дальнейшем, приложение может быть расширено, расширив типы карт и режимов соревнования, а также добавить многопользовательский режим. Для удалённого доступа к статистике пользователя и новостям соревнований можно создать сайт.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Малоподвижность смертельно опасна. – https://www.bbc.com/russian/science/2012/07/120718_inactivity_research_lancet (дата обращения: 3.03.2021).
2. Спорт и политика в современном мире. – <https://cyberleninka.ru/article/n/sport-i-politika-v-sovremennom-mire/viewer> (дата обращения: 17.03.2021).
3. Всемирная организация здравоохранения. – <https://www.who.int/ru/newsroom/detail/22-11-2019-new-who-led-study-says-majority-of-adolescentsworldwide-are-not-sufficiently-physically-active-putting-their-current-andfuture-health-at-risk> (дата обращения: 20.04.2021).
4. 11 популярных игровых движков в 2021 году. – https://blackcaviar.games/obzor_igrovyyh_dvizhkov (дата обращения: 28.03.2021).
5. C#. – <https://docs.microsoft.com/ru-ru/dotnet/csharp> (дата обращения: 10.03.2021).
6. Агуров, П. С#. Сборник рецептов / П. Агуров. - М.: "БХВ-Петербург", 2012. - 432 с.
7. Вагнер, Б. С# Эффективное программирование / Б. Вагнер. - М.: ЛОРИ, 2013. - 320 с.
8. Рендольф, Н. Visual Studio 2010 для профессионалов / Н. Рендольф. - М.: Диалектика / Вильямс, 2016. - 516 с.
9. Лучший игровой движок по версии пользователей хабра. – <https://habr.com/ru/post/307952> (дата обращения: 28.03.2021).
10. Unity Manual. – <https://docs.unity3d.com> (дата обращения: 08.04.2021).
11. Unreal Engine. – <https://www.unrealengine.com/en-US> (дата обращения: 04.03.2021).
12. Хокинг, Д. Unity в действии. Мультиплатформенная разработка на C#/ Д. Хокинг. - Питер, 2016 – 336 с.
13. Алан, Т. Искусство создания сценариев в Unity / Т. Алан. - Питер, 2016 – 360 с.
14. Монк, С. Raspberry Pi. Сборник рецептов. Решение программных и аппаратных задач / С. Монк. – СПб.: Диалектика, 2017. – 522 с.
15. Бхаргава, А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих / А. Бхаргава - Питер, 2019 – 290 с.
16. Adobe Photoshop. – <https://works.doklad.ru/view/VYUtluuzaZQ0.html> (дата обращения: 17.03.2021).
17. Прахов, А. Blender. 3D-моделирование и анимация. Руководство для начинающих / А. Прахов. - М.: "БХВ-Петербург", 2009. - 516 с.

18. Zwift. – <https://www.zwift.com/get-zwifiting> (дата обращения: 15.02.2021).
19. FTP. – <https://clck.ru/VYMeP> (дата обращения: 15.02.2021)
20. OneLap. – <https://support.onelap.com/hc/en-us> (дата обращения: 15.02.2021).
21. RGT Cycling. – <https://www.rgtcycling.com/about/> (дата обращения: 15.02.2021).
22. VZFit. – https://www.oculus.com/experiences/quest/2088366894520136/?locale=ru_RU (дата обращения: 15.02.2021).
23. Анурьев, В.И. Справочник конструктора – машиностроителя в 3-х томах / В.И Анурьев. – М.: Изд-во «Машиностроение», 1999. – 585 с.
24. Чиликин, М. Г. Общий курс электропривода: учебник для вузов / М. Г. Чиликин. – М.: Энергоиздат, 1981. – 576 с.
25. GitHub проект ПК приложения. - <https://github.com/olegon17/BicycleTrainer2> (дата обращения: 1.06.2021).

ПРИЛОЖЕНИЕ А

СКРИПТ ИНИЦИАЛИЗАЦИИ ШЛЕМА VR

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

namespace trinus{
    public class TrinusCamera : MonoBehaviour
    {
        public enum CAMERA_MODE{
            DISABLED,
            SINGLE,
            DUAL,
            UNITY_VR
        }
        Camera leftCamera;
        Camera rightCamera;
        Camera splitCamera;

        Camera uiCameraMain;
        Camera uiCameraSub;

        Vector3 offset;

        Vector3 defaultRotation;

        public delegate void cameraEnabled(TrinusCamera camera, bool
toggle);
        cameraEnabled cameraEnabledDelegate;

        CAMERA_MODE activeMode = CAMERA_MODE.DISABLED;

        [Tooltip("The Camera mode determines if the view should be
monoscopic or stereoscopic (using Unity VR mode or a two camera rig).
Refer to the manual for details")]
        public CAMERA_MODE defaultMode = CAMERA_MODE.UNITY_VR;

        void Awake(){
            defaultRotation = new
Vector3(transform.localEulerAngles.x , transform.localEulerAngles.y,
transform.localEulerAngles.z);
            Transform t = transform.Find ("LeftCamera");
            if (t != null)
                leftCamera = (Camera)t.GetComponent<Camera> ();
            t = transform.Find ("RightCamera");
            if (t != null)
                rightCamera = (Camera)t.GetComponent<Camera> ();
```

```

t = transform.Find ("SplitCamera");
if (t != null)
    splitCamera = (Camera)t.GetComponent<Camera> ();

    if      (splitCamera      ==      null      &&
transform.GetComponent<Camera>() != null) {
    //TrinusCamera attached to a camera, ignore children
and use that one as main
    splitCamera = transform.GetComponent<Camera>();
    //TODO spawn new left/right cams
    if (leftCamera == null)
        leftCamera = splitCamera;
    if (rightCamera == null)
        rightCamera = splitCamera;
    defaultMode = CAMERA_MODE.SINGLE;
    Debug.Log("TrinusCamera using external camera ("
+ splitCamera + "). Switching to single camera mode");
    }

    uiCameraMain      =      GameObject.Find
("TrinusUICamera").GetComponent<Camera>();
    uiCameraSub      =      uiCameraMain.transform.Find
("Camera").GetComponent<Camera> ();

    setMode(defaultMode);
}

public void setFov(int f){
    if (activeMode == CAMERA_MODE.DUAL) {
        leftCamera.fieldOfView = f;
        rightCamera.fieldOfView = f;
    } else
        splitCamera.fieldOfView = f;

    PlayerPrefs.SetInt ("fov", f);
}
public Vector3 getDefaultRotation(){
    return defaultRotation;
}
public Vector3 getOffset(){
    return offset;
}
public void setOffset(Vector3 offset){
    this.offset = offset;
}

public int getFov(){
    return PlayerPrefs.GetInt ("fov", 100);
}
public void setConvergence(float c){
}
public float getConvergence(){

```

```

        return 0;
    }
    public void setSeparation(float s){
    }
    public float getSeparation(){
        return 0;
    }
    public Camera getMainCamera(){
        if (activeMode == CAMERA_MODE.DUAL)
            return leftCamera;
        return splitCamera;
    }
    public Camera[] getMainDualCamera(){
        return new Camera[]{ leftCamera, rightCamera };
    }
    public Camera getUICamera(){
        return uiCameraMain;
    }

    public CAMERA_MODE getMode(){
        return activeMode;
    }
    public void setDefaultMode(){
        setMode (defaultMode);
    }
    public void setMode(CAMERA_MODE mode){
        if (mode == activeMode)
            return;
        activeMode = mode;

        switch (mode) {
        case CAMERA_MODE.DISABLED:
            leftCamera.gameObject.SetActive(false);
            rightCamera.gameObject.SetActive(false);
            splitCamera.gameObject.SetActive(false);
            break;
        case CAMERA_MODE.SINGLE:
            leftCamera.gameObject.SetActive(false);
            rightCamera.gameObject.SetActive(false);
            splitCamera.gameObject.SetActive(true);
            UnityEngine.XR.XRSettings.enabled = false;

            uiCameraSub.gameObject.SetActive (false);
            uiCameraMain.rect = new Rect (0, 0, 1, 1);
            break;
        case CAMERA_MODE.UNITY_VR:
            leftCamera.gameObject.SetActive (false);
            rightCamera.gameObject.SetActive (false);
            splitCamera.gameObject.SetActive (true);
        }
    }
    UnityEngine.XR.XRSettings.LoadDeviceByName("GoogleVR");
    UnityEngine.XR.XRSettings.enabled = true;

```

```

        uiCameraSub.gameObject.SetActive (false);
        uiCameraMain.rect = new Rect (0, 0, 1, 1);
        break;
    case CAMERA_MODE.DUAL:
        leftCamera.gameObject.SetActive (true);
        rightCamera.gameObject.SetActive (true);
        if (splitCamera != leftCamera)//TrinusCamera
attached to external camera case
            splitCamera.gameObject.SetActive (false);

        uiCameraSub.gameObject.SetActive (true);
        uiCameraSub.gameObject.SetActive (true);
        uiCameraSub.rect = new Rect (0.5f, 0, 0.5f, 1);
        uiCameraMain.rect = new Rect (0, 0, 0.5f, 1);
        break;
    }
    setFov (getFov ());
}
public void setCameraEnabledDelegate(cameraEnabled ce){
    cameraEnabledDelegate = ce;
}
public void OnEnable(){
    enabledCamera ();
}
public void enabledCamera(){
    if (cameraEnabledDelegate != null)
        cameraEnabledDelegate (this, false);
}
public void syncCameras(){
    if (leftCamera != null & rightCamera != null) {
        syncComponents (leftCamera.gameObject,
rightCamera.gameObject);
        syncComponents (rightCamera.gameObject,
leftCamera.gameObject);
    }
}
private void syncComponents(GameObject source, GameObject
dest){
    List<Component> list = new List<Component>();
    source.GetComponents (list);
    foreach(Component component in list){
        copyComponent(dest, component);
    }
}
private static void copyComponent<T>(GameObject gameObject,
T other) where T : Component
{
    Type type = other.GetType();
    if (gameObject.GetComponent(type) != null)
        return;//already exists
}

```

```

Component  newComponent  =  gameObject.AddComponent
(type);

    if (other is Behaviour)
        ((Behaviour)newComponent).enabled  =  (other  as
Behaviour).enabled;

        BindingFlags  flags  =  BindingFlags.Public  |
BindingFlags.NonPublic  |  BindingFlags.Instance  |  BindingFlags.Default
|  BindingFlags.DeclaredOnly;
        PropertyInfo[]  pinfos  =  type.GetProperties(flags);
        foreach (PropertyInfo  pinfo  in  pinfos) {
            if (pinfo.CanWrite) {
                try {
                    pinfo.SetValue(newComponent,
pinfo.GetValue(other, null), null);
                }
                catch { }
            }
        }
        FieldInfo[]  finfos  =  type.GetFields(flags);
        foreach (FieldInfo  finfo  in  finfos) {
            finfo.SetValue(newComponent,
finfo.GetValue(other));
        }
    }
}

```

ПРИЛОЖЕНИЕ Б

ПРОГРАММА ТРЕНАЖЕРА НА RASPBERRY PI ZERO W

```
from mpu6050 import mpu6050
import time
import pickle
import VL53L0X
import Adafruit_SSD1306
import RPi.GPIO as GPIO
import socket
import sys, subprocess, struct
import argparse
import asyncio
import sys
import os
import subprocess
from PIL import Image, ImageFont, ImageDraw
import serial
import bluetooth
print("libs imported")

ROOT_LOG = '/var/log'

class app():
    def __init__(self):
        self.UNIT = 0x1
        self.bufferSize = 1024
        self.serverAddressPort = ('192.168.0.175', 20001)
        self.disp = display()
        print("display inited")
        self.disp.chek_IP()
        self.sock = socket.socket()
        self.sock.connect(self.serverAddressPort)
        self.restartMotionDetector = True

        self.loop = asyncio.get_event_loop()

        self.dic_tasks = {}
        self.tasks = ()
        self.serv_tasks = ()
        self.init_service_task()
        self.tasks = self.serv_tasks
        print(str("CHislo          zadach:          ").encode('utf-8'),
len(self.tasks))
        print("-----zapuskaem planirovshik-----")
        self.loop.run_until_complete(self.task_manager())
        asyncio.run(self.task_manager())

    def init_service_task(self):
```

```

"""
funkcia inicializacii zadach
"""
tasks = []
print("dobavlyaem svoyu korutinu")
my_task2 = self.loop.create_task(self.task_send_udp())
tasks.append(my_task2)

tmp = (t for t in tasks)
self.serv_tasks = tuple(tmp)
print(self.serv_tasks)

async def task_manager(self):
    try:
        self.gather = asyncio.gather(*self.tasks)
        await self.gather
        print("-----zapuskayem vse zadachi-----")
    except asyncio.CancelledError:
        print('Osnovnaya zadacha otmenena!')
    async def task_recive_udp(self):
        while True:
            print ("waiting...")
            await asyncio.sleep(0.5)
            msgFromServer =
self.UDPClientSocket.recvfrom(self.bufferSize)
            array = pickle.loads(msgFromServer[0])
            # address = msgFromServer[1]
            msg = "Message from Server:{}".format(array)
            print (msg)
            if array[0] == "angle":
                print(array[1])
            await asyncio.sleep(1)

    async def task_send_udp(self):

        print("task_send_udp started")

        self.laser=_laser()
        cycle_mm=self.laser.get_range_cycles()
        print (cycle_mm/2.54/10,' inch')
        self.cycle_diam=cycle_mm

        self.encoder = _encoder(self.sock, self.cycle_diam,self.disp)
        self.gyro = _gyroscope()
        while True:
            await asyncio.sleep(1)

```