

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

_____ А.А.Игуменьцев

«__»_____ 2020 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой ЭВМ

_____ Г.И. Радченко

«__»_____ 2020 г.

Разработка программно-аппаратного комплекса для интерактивного
велотренажера

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Руководитель работы,
к.т.н., доцент каф. ЭВМ

_____ Д.В.Топольский

«__»_____ 2020 г.

Автор работы,
студент группы КЭ-222

_____ Е.В.Семиуглов

«__»_____ 2020 г.

Нормоконтролёр,
ст. преп. каф. ЭВМ

_____ Сяськов С.В.

«__»_____ 2020 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
« ___ » _____ 2020 г.

ЗАДАНИЕ

на выпускную квалификационную работу бакалавра
студенту группы КЭ-222
Семиуглову Евгению Вячеславовичу
обучающемуся по направлению
09.04.01 «Информатика и вычислительная техника»

- 1. Тема работы:** «Разработка программно-аппаратного комплекса для интерактивного велотренажера» утверждена приказом по университету от 24.04.2020. № 627
- 2. Срок сдачи студентом законченной работы:** 1 июня 2020 г.
- 3. Исходные данные к работе:** техническое задание.
 - языки программирования – С++ и С#;
 - игровой движок – Unity;
 - платформа разработки – Microsoft Windows;
 - микроконтроллер для передачи данных – Digispark ATtiny85;
 - велотренажер – механический, 1988г.

4. Перечень подлежащих разработке вопросов:

- анализ предметной области;
- определение требований к комплексу;
- проектирование комплекса;
- реализация комплекса;
- тестирование.

5. Дата выдачи задания: 1 декабря 2019 г.

Руководитель работы _____ /*Д.В.Топольский*/

Студент _____ /*Е.В.Семиуглов* /

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и анализ предметной области	01.03.2020	
Определение требований к комплексу	16.03.2020	
Проектирование комплекса	20.04.2020	
Реализация комплекса	01.05.2020	
Тестирование	15.05.2020	
Компоновка текста работы и сдача на нормоконтроль	24.05.2020	
Подготовка презентации и доклада	30.05.2020	

Руководитель работы _____ /Д.В.Топольский/

Студент _____ /Е.В.Семиуглов/

Аннотация

Семиуглов Е.В. Разработка программно-аппаратного комплекса для интерактивного велотренажера. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭЖН; 2020, 76с., 27 ил., библиогр. список – 27 наим.

В рамках выпускной квалификационной работы был разработан программно-аппаратный комплекс для интерактивного велотренажера, состоящий из физического модуля и программной реализации для управления велосипедом в компьютерной игре. Комплекс основан на микроконтроллере ATtiny85 и приложении, разработанном на платформе Unity.

В ходе реализации были выполнены задачи, связанные с созданием управления велотренажером посредством микроконтроллера. Выполнена работа по созданию моделей и анимации. Производится анализ работы приложения.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	8
1.АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	10
1.1. ЦЕЛЬ ДИПЛОМНОЙ РАБОТЫ.....	10
1.2. ОБЗОР ЛИТЕРАТУРЫ.....	10
1.3. ЭТАПЫ РЕШЕНИЯ ЗАДАЧИ.....	13
1.4. ОБЗОР АНАЛОГОВ	13
1.5. АНАЛИЗ И ПОДБОР КОМПОНЕНТОВ АППАРАТНОЙ ЧАСТИ КОМПЛЕКСА	16
1.6. ВЫБОР ИГРОВОГО ДВИЖКА	21
1.7. ВЫВОДЫ ПО ГЛАВЕ 1	24
2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ К КОМПЛЕКСУ	26
2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	26
2.2. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ.....	26
2.2.1 ТРЕБОВАНИЯ АППАРАТНОЙ ЧАСТИ КОМПЛЕКСА	26
2.2.2 ТРЕБОВАНИЯ ПРОГРАММНОЙ ЧАСТИ КОМПЛЕКСА.....	27
2.2.3 ТРЕБОВАНИЯ К ЛИНГВИСТИЧЕСКОМУ ОБЕСПЕЧЕНИЮ..	27
2.2.4 ТРЕБОВАНИЯ К ДОКУМЕНТАЦИИ	27
2.4. ВЫВОДЫ ПО ГЛАВЕ 2.....	27
3. ПРОЕКТИРОВАНИЕ	28
3.1. ФУНКЦИОНАЛЬНЫЙ СОСТАВ.....	28
3.2. ОБЩАЯ СТРУКТУРА АППАРАТНОЙ ЧАСТИ.....	29
3.3. СТРУКТУРНЫЙ СОСТАВ ПРОГРАММНОЙ ЧАСТИ.....	30
3.4. ДИАГРАММА ПРЕЦЕДЕНТОВ ПРИЛОЖЕНИЯ	31
3.5. ВЫВОДЫ ПО ГЛАВЕ 3.....	36

4. РЕАЛИЗАЦИЯ	37
4.1. РЕАЛИЗАЦИЯ ПРОГРАММНОЙ ЧАСТИ	37
4.1.1. ФАЙЛОВАЯ СТРУКТУРА ИГРЫ	37
4.1.2. РЕАЛИЗАЦИЯ ИНТЕРФЕЙСОВ	39
4.2. РЕАЛИЗАЦИЯ АППАРАТНОЙ ЧАСТИ.....	46
4.3. ВЫВОДЫ ПО ГЛАВЕ 4.....	48
5. ТЕСТИРОВАНИЕ	49
5.1. ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ	49
5.2. ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЯ.....	49
5.3. ВЫВОДЫ ПО ГЛАВЕ 5	51
ЗАКЛЮЧЕНИЕ	52
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	53
ПРИЛОЖЕНИЕ А	56
ПРИЛОЖЕНИЕ Б.....	58
ПРИЛОЖЕНИЕ В	59
ПРИЛОЖЕНИЕ Г.....	71

ВВЕДЕНИЕ

Игровая индустрия имеет большое влияние в современном мире. Существует множество игр различных жанров для всех видов электронных устройств: персональных компьютеров, ноутбуков, планшетов, смартфонов, приставок и т.д.

Частично из-за этого, люди стали вести сидячий образ жизни. Ведь ни для кого не секрет, что малоподвижность и отсутствие физической нагрузки пагубно сказывается на здоровье. По результатам исследований [1], такой образ жизни приводит к смерти 5,3 млн. человек ежегодно.

Именно поэтому так важно начать заниматься спортом, будь то бег, плавание или же поездка на велосипеде. Ведение физически активного образа жизни сильно влияет на состояние опорно-двигательной системы, сердечно-сосудистой системы, массы тела и др. [2].

Для привлечения людей к здоровому образу жизни, к нему необходимо повысить интерес. Этого можно достичь, например, совмещением игры с любым физическим тренажером, как делают это производители современных беговых дорожек.

У многих людей дома стоят велотренажеры, но обычно они используются не по прямому назначению, это происходит из-за утраты интереса к процессу тренировки.

Цели и задачи.

Целью работы является разработка программно-аппаратный комплекс, который позволит модернизировать домашний велотренажер в средство управления в новой игре.

Для достижения данной цели должны быть решены следующие задачи:

- провести обзор литературы необходимой для решаемых задач;
- провести анализ и выбор средств реализации устройства управления;
- провести анализ и выбор средств реализации игры;
- спроектировать аппаратную систему;
- спроектировать программную систему;
- реализовать программно-аппаратный комплекс;
- провести тестирование программно-аппаратного комплекса.

Содержание работы.

В первой главе проведен обзор литературы, аналогичных проектов и инструментов, необходимых для реализации комплекса.

Во второй главе описываются требования к комплексу.

В третьей главе для формализации функциональных требований были применены UML-диаграмма вариантов использования.

В четвертой главе описывается файловая структура, представлен игровой процесс.

В пятой главе приведены результаты тестирования игры.

В заключении описаны результаты, полученные в ходе выполнения работы, а также возможные пути развития.

Актуальность.

Данный комплекс является актуальным по нескольким причинам. Во-первых, посредством игровой формы интерес людей к процессу тренировок значительно вырастет, что положительно скажется на здоровье. Во-вторых, когда нет возможности заниматься тренировкой на улице, например, при карантине или плохих погодных условиях, это можно делать дома, не прерывая процесс упражнений.

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

Игровой движок – это тот модуль игры, который включает в себя игровую логику. Он значительно облегчает процесс разработки приложения за счёт экономии времени и трудовых ресурсов и даёт игре возможность запускаться на нескольких платформах. В настоящее время существует огромное количество средств разработки игр.

Был проведен анализ наиболее популярных бесплатных или условно бесплатных средств разработки компьютерных игр. Для сравнения движков был выбран электронный ресурс [2], в котором подробно описаны самые популярные средства разработки игр, предоставляющих свою бесплатную версию пользователю. Эти платформы доступны для свободного использования.

Из рассматриваемых платформ особенно выделились Unity и Unreal Engine 4

[3,4], 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 сравнимым параметрам.

Для разработки игрового приложения была выбрана платформа Unity[5]. Данная платформа имеет низкий порог вхождения, большое количество обучающих материалов [6,7] и сообщество разработчиков, вследствие чего, с ней можно быстрее начать работать.

В качестве основного языка программирования используется C# [8]. Данные книги [9,10,11] являются хорошим введением в разработку приложений.

Кодовую базу было решено хранить в GitHub [12] – крупнейшем веб-сервисе от Microsoft, основанном на системе контроля версий Git. Git позволяет создавать несколько различных, параллельных версий проекта – веток, каждая из которых предназначена для разных целей. Допустим есть работающий проект. Основная ветка проекта – master. В ней хранится последняя стабильная версия. Для добавления в проект новой функциональности необходимо создать отдельную ветку, при этом данные изменения никак не отразятся на основной версии. Можно переключаться между ветками, что приведёт к изменению файлов локальной версии репозитория.

Для создания и редактирования графической составляющей игры будет использоваться графический редактор Adobe Photoshop.

Adobe Photoshop [13] — 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.

Эта программа будет использоваться для создания графической составляющей игры, по ней написано огромное количество статей [14,15], книг [16, 17] и обучающих материалов.

Книга [18] нужна как новичкам, которые только начинают знакомиться с этой программой, так и более опытным пользователям, желающим получить от Adobe Photoshop СС максимум. С помощью простых, но очень хорошо подобранных инструкций, они позволят на практике освоить все инструменты Adobe Photoshop СС.

Для создания моделей в разрабатываемой игре, будет использоваться 3ds Max – Autodesk 2018 [19]. Это профессиональное программное обеспечение для 3D-моделирования, анимации и визуализации при создании игр и проектировании. По этой программе в сети расположено большое количество видео уроков, а также статей и книг.

В данных книгах [20,21,22] подробно описаны этапы создания анимации и проектирования.

Был произведен обзор модулей и платформы быстрой разработки электронных устройств Arduino. Платформа пользуется популярностью во всем мире из-за удобства и простоты языка программирования. Еще одним из преимуществ является открытая архитектура и программный код. Программирование устройства производится через USB без использования программаторов.

С помощью языка С++ [23], можно запрограммировать микроконтроллер, расположенный на плате. Программирование происходит в среде разработки Arduino [24]. Отличительной чертой является автономность работы проектов, основанных на Arduino, но данный факт не исключает взаимодействие с ПО на компьютере. Все ПО можно скачать бесплатно. Также в открытом доступе имеются чертежи схем [25].

1.3. ЭТАПЫ РЕШЕНИЯ ЗАДАЧИ

Для достижения цели выпускной работы необходимо выполнить несколько этапов:

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 компьютерной игрой;
 - тестирование работы комплекса.

1.4. ОБЗОР АНАЛОГОВ

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.

Одним из таких решений, является проект «VZFit» от разработчика «VirZOOM» [26].



Рисунок 1 – Проект «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 игр для такого устройства.

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

- покупка подписки;
- наличие виртуального шлема «oculus rift» или «oculus go»;
- интерфейс на английском языке.

Ещё одним похожим решением является проект «ZWIFT» [27]. Это многопользовательский онлайн проект, который предлагает модернизировать настоящий велосипед, посредством специального устройства, которое будет заменять заднее колесо, пример такого решения представлен на рисунке 2.



Рисунок 2 – Проект «ZWIFT»

Однако большим недостатком такой системы является цена устройства, а также интерфейс на иностранном языке. На рисунке 3 представлены цены на различные типы устройств.



Wahoo Kickr
€1,199.99
Out of stock



Tacx Neo 2T
€1,299.99
Out of stock



Wahoo Kickr Core
€799.99
Out of stock

Рисунок 3 – Стоимость устройств «ZWIFT»

1.5. АНАЛИЗ И ПОДБОР КОМПОНЕНТОВ АППАРАТНОЙ ЧАСТИ КОМПЛЕКСА

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 USB.

Digispark ATtiny85

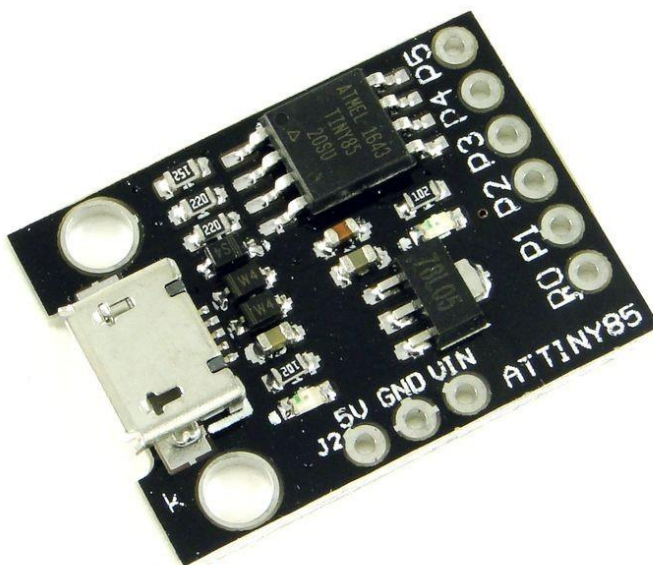


Рисунок 4 – Digispark ATtiny85

Микроконтроллер ATtiny85 вполне функционален, в чем можно убедиться в ходе изучения следующих его характеристик и возможностей:

- для программного кода предусмотрено 8 КБ памяти;
- для исполняемого кода зарезервировано 512Б;

- наличие 6 цифровых пинов;
- два выхода PWM и 4 ADC;
- частота от 1 до 20 МГц.

ATtiny предлагается в двух корпусах, в зависимости от модификации источник питания может быть от 1,8 до 5,5В. В экономичном режиме прибор потребляет от 0,1 мкА, в микроконтроллере реализовано аппаратное прерывание. ATtiny85 по сравнению с другими микроконтроллерами этого семейства имеет максимальную память, что делает его более востребованным на рынке. В нашем случае, была взята модификация, от компании Digispark, которая дает возможность подключения контроллера через USB.

Arduino Micro



Рисунок 5 – Arduino Micro

Микроконтроллер Arduino Micro – плата микроконтроллера на базе ATmega32u4 разработанный совместно с Adafruit. Плата имеет 20 цифровых вход/выходов (из них 7 могут использоваться в качестве выходов ШИМ и 12 – как аналоговые входы), кварцевый генератор частотой 16 МГц, гнездо микро-USB, разъем ICSP и кнопку reset. На ней есть все, что необходимо для работы с микроконтроллером. Форм-фактор контроллера позволяет легко разместить его на макетной плате. Имеет следующие технические характеристики:

- рабочее напряжение 5 В;
- входы/выходы 20;
- каналы ШИМ 7;
- аналоговые входные каналы 12;
- флеш-память 32 Кб из которых 4 Кб используются для загрузчика;
- ОЗУ 2,5 Кб;
- тактовая частота 16 МГц.

Arduino Leonardo

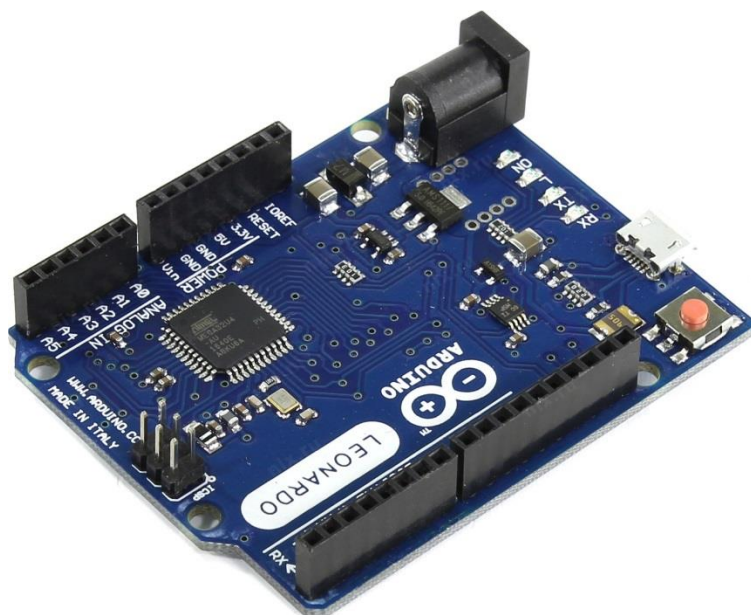


Рисунок 6 – Arduino Leonardo

Плата Arduino Leonardo существенно расширяет возможности для подключения Arduino к компьютеру. Она рассматривается компьютером как стандартное HID-устройство и может напрямую слать нужные управляющие сигналы.

В основе Leonardo лежит достаточно продвинутый микроконтроллер ATmega32u4, на ней можно найти два десятка входов и выходов, разъем micro-USB, кнопка для перезагрузки, а также разъемы ICSP и силовой вход.

Предусмотрена автоматическая перезагрузка (нет необходимости нажимать дополнительную кнопку). Процесс запускается при открытии виртуального CDC COM-порта со скоростью в 1200 бод. После этого система перезагружается. При желании можно активировать процесс вручную с помощью специальной кнопки Reset. При подаче питания контроллер сразу работает с учетом установленного ПО.

- рабочее напряжение 5 В;
- входы/выходы 20;
- каналы ШИМ 7;
- аналоговые входные каналы 12;
- флеш-память 32 Кб из которых 4 Кб используются для загрузчика;
- ОЗУ 2 Кб;
- тактовая частота 16 МГц.

Сравним в таблице 1 подходящие микроконтроллеры по нашим требованиям.

Таблица 1 – Сравнение микроконтроллеров для комплекса

	Digispark ATtiny85	Arduino Micro	Arduino Leonardo
Макс. частота, МГц	20	16	16
RAM, байт	512	2000	2000
Flash, Кб	8	32	32
I/O выводов	6	12	20
Цена, руб.	250	400	450
Размер, см	1,8 × 2,2	4,8 × 1,77	6,9 × 5,3

Вывод: в таблице представлено сравнение микроконтроллеров по основным характеристикам. Из представленных, наиболее подходящим является Digispark ATtiny85, так как в соответствии с требованиями размеров и стоимости подходит больше всего.

Для считывания количества оборотов педалей велотренажера, был выбран датчик Холла TLE4905L представленный на рисунке 7.

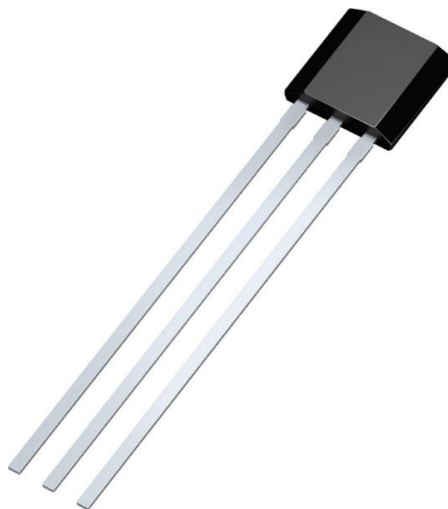


Рисунок 7 – Датчик Холла TLE4905L

Для управления движениями в стороны, были выбраны большие кнопки (12x12 мм), изображённые на рисунке 8.



Рисунок 8 – Кнопка для поворотов

1.6. 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.

Unity3D



Рисунок 9 - Интерфейс платформы Unity

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.

Основные возможности и плюсы 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.

Минусы:

- ограниченный набор инструментов;
- не самая лучшая графика.

Unreal Engine

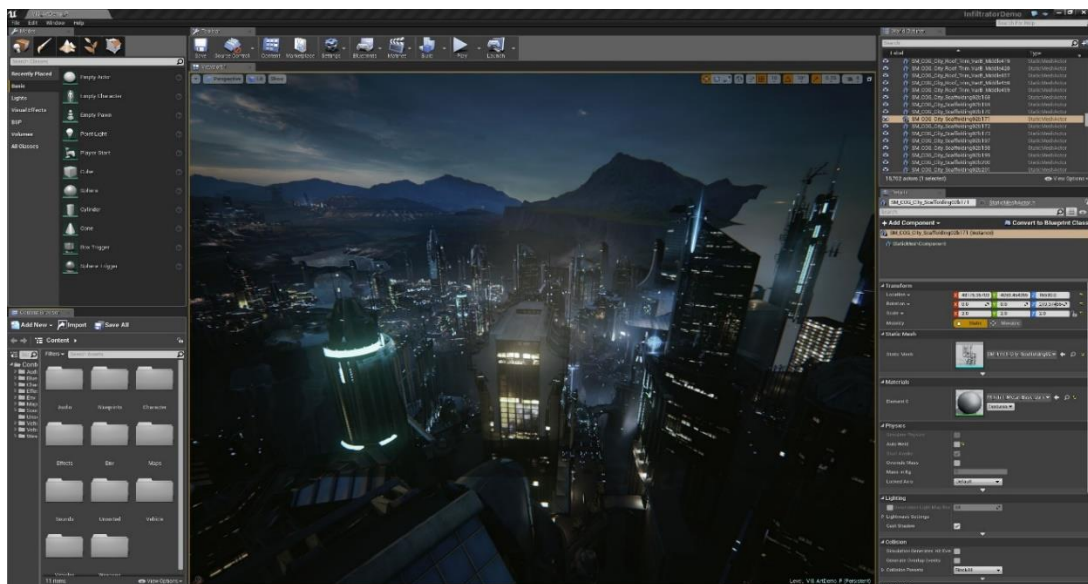


Рисунок 10 - Интерфейс платформы Unreal Engine

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.

Плюсы:

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

Минусы:

- 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;
- цена;
- 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.

1.7. 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 устройства сбора и передачи данных. Сравнивая Digispark ATtiny85, Arduino Micro и Arduino Leonardo, выбор остановился на Digispark ATtiny85, так как по необходимым критериям, а это, небольшая стоимость устройства, небольшой размер, достаточная мощность для считывания всех сигналов без задержки и возможность подключения по USB. Поэтому он подходит больше.

Для считывания данных о движениях пользователя, был выбран датчик Холла KY-003, чтобы определять количество оборотов, сделанных пользователем, а также кнопки, чтобы передавать данные о сделанном повороте вправо или влево.

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 Microsoft Windows. 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 15 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 ЛИНГВИСТИЧЕСКОМУ ОБЕСПЕЧЕНИЮ

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.

2.4. 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 11 представлена функциональная схема системы.

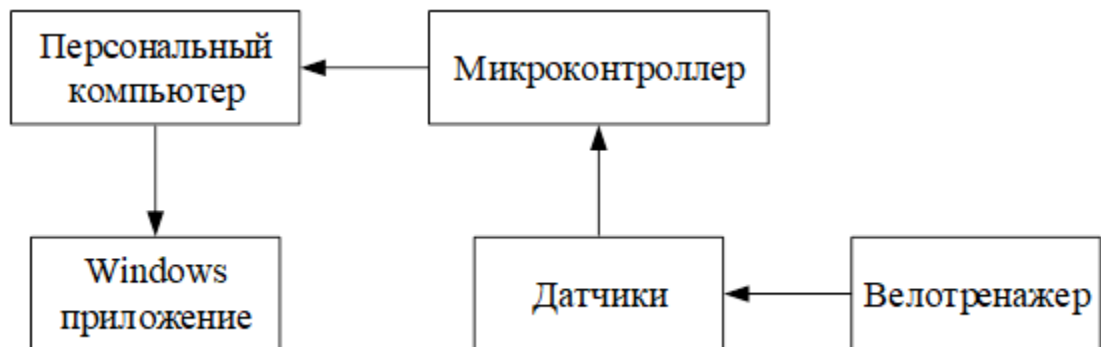


Рисунок 11 — Функциональная схема системы

Из функциональной схемы видно, что установленные на велотренажер датчики, взаимодействуют с микроконтроллером, который будет передавать сигналы от них, непосредственно на ПК.

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

3.2. ОБЩАЯ СТРУКТУРА АППАРАТНОЙ ЧАСТИ

Аппаратное обеспечение состоит из различных электронных элементов. В его состав входят следующие модули:

- питание информационной части от ПК;
- датчики Холла, поворотов направо и налево;
- обмен информацией с системой.

Каждый из которых обладает определенным функционалом, благодаря которому программно-аппаратный комплекс сможет действовать как одно целое. На рисунке 12 представлена структурная схема.

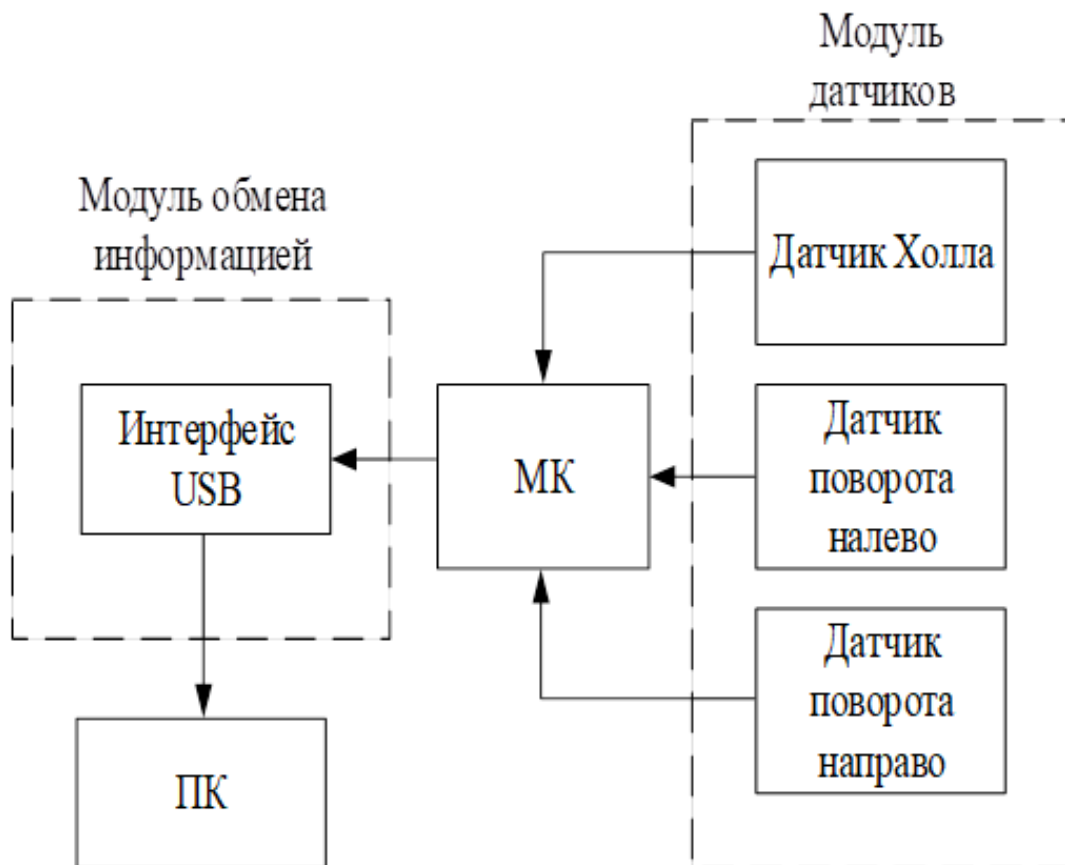


Рисунок 12 — Структурная схема аппаратной части

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 13.

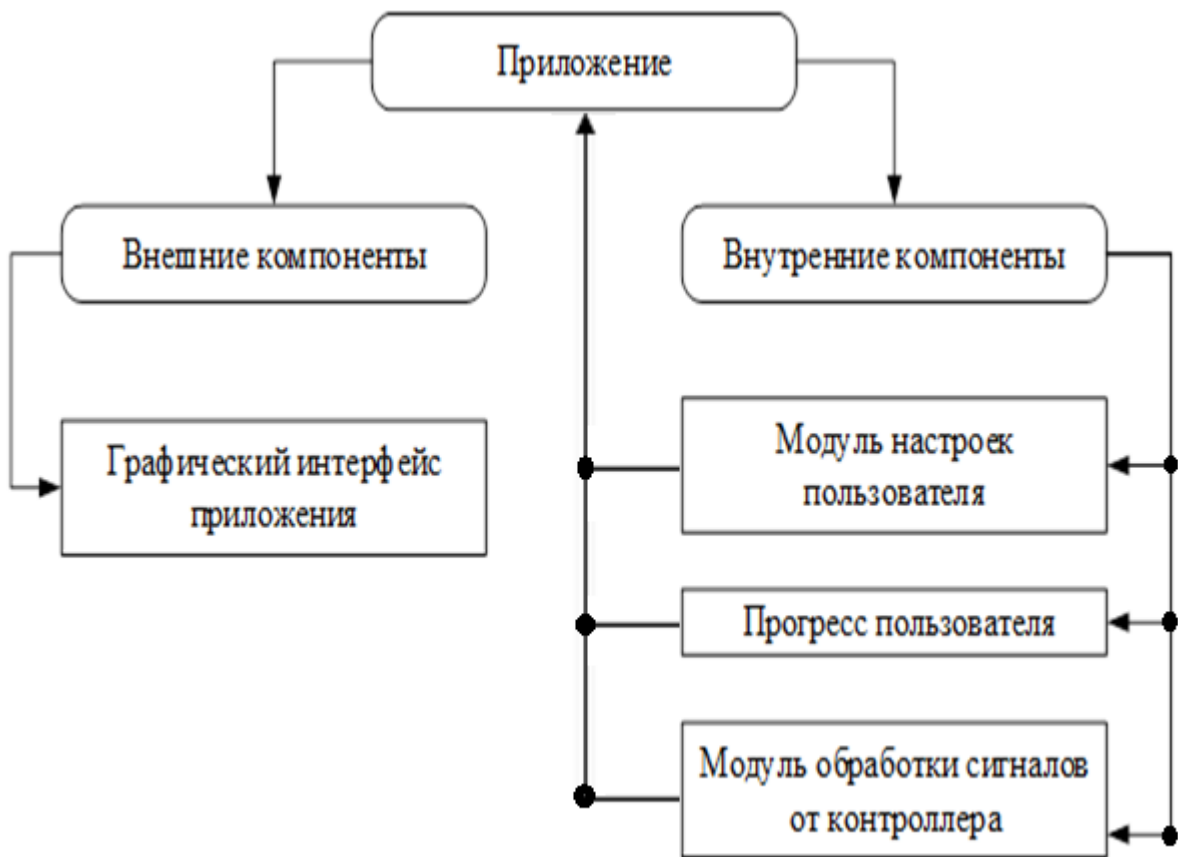


Рисунок 13 — Структурная схема приложения

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, заходить в меню настроек и выйти из игры.



Рисунок 14 – Диаграмма прецедентов главное меню

На основе данных потребностей выделены следующие варианты использования:

- играть;
- прогресс игры;
- настройки;

- общий прогресс;
- прогресс последней игровой сессии;
- выход;
- назад;
- игра;
- управление;
- графика;
- изменение общей громкости;
- изменение громкости музыки;
- изменение громкости эффектов;
- изменение чувствительности контроллера;
- выбор полного экрана;
- изменение разрешения;
- выбор качества текстур.

Рассмотрим более подробно каждый из этих прецедентов.

Прецедент «играть»

Краткое описание

Позволяет пользователю начать управлять велосипедом посредством подключенного к нему велотренажера.

Прецедент «процесс игры»

Краткое описание

Позволяет пользователю перейти в подменю прогресса игры.

Прецедент «настройки»

Краткое описание

Позволяет пользователю перейти в меню настроек. Этот вариант

использования начинается, когда игрок нажимает на кнопку паузы.

Прецедент «общий прогресс»

Краткое описание

Позволяет пользователю просмотреть данные о своем общем прогрессе.

Прецедент «прогресс последней сессии»

Краткое описание

Позволяет пользователю просмотреть данные о своей последней завершенной сессии.

Прецедент «выход»

Краткое описание

Позволяет пользователю выйти из приложения.

Прецедент «назад»

Краткое описание

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

Прецедент «игра»

Краткое описание

Позволяет пользователю перейти в подраздел настройки «игра».

Прецедент «управление»

Краткое описание

Позволяет пользователю перейти в подраздел настройки «управление».

Прецедент «графика»

Краткое описание

Позволяет пользователю перейти в подраздел настройки «графика».

Прецедент «изменение общей громкости»

Краткое описание

Позволяет пользователю изменять общую громкость в игре. Этот вариант использования начинается, когда пользователь меняет общую громкость.

Прецедент «изменение громкости музыки»

Краткое описание

Позволяет пользователю изменять громкость музыки в игре. Этот вариант использования начинается, когда пользователь меняет громкость музыки.

Прецедент «изменение громкости эффектов»

Краткое описание

Позволяет пользователю изменять громкость эффектов в игре. Этот вариант использования начинается, когда пользователь меняет громкость эффектов.

Прецедент «изменение чувствительности контроллера»

Краткое описание

Позволяет пользователю изменять чувствительность контроллера в игре. Этот вариант использования начинается, когда пользователь меняет чувствительность контроллера.

Прецедент «выбор полного экрана»

Краткое описание

Позволяет пользователю выбрать полный экран в игре. Этот вариант использования начинается, когда пользователь меняет выбор полного экрана.

Прецедент «изменение разрешения»

Краткое описание

Позволяет пользователю изменять разрешение экрана. Этот вариант использования начинается, когда пользователь меняет разрешение экрана.

Прецедент «выбор качества текстур»

Краткое описание

Позволяет пользователю выбирать качество текстур. Этот вариант использования начинается, когда пользователь меняет качество текстур.

3.5. 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 Autodesk 3ds Max.

Для программирования микроконтроллера использовался язык C в Arduino IDE.

4.1. РЕАЛИЗАЦИЯ ПРОГРАММНОЙ ЧАСТИ

4.1.1. ФАЙЛОВАЯ СТРУКТУРА ИГРЫ

Разработанный проект содержит каталоги, в которых хранятся:

- анимации объектов;
- музыка;
- шрифты;
- шаблоны объектов;
- игровую сцену;
- скрипты;
- 3D-модели.

Файловая структура представлена на рисунке 15.

В папке Animations находятся анимации всех объектов игры. В папке Audio находятся вся музыка и звуки для игры. В директории Prefabs 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;
- Sceney содержит в себе 3D-модели всех второстепенных объектов, таких как столбы, скамейки, деревья и т.д.

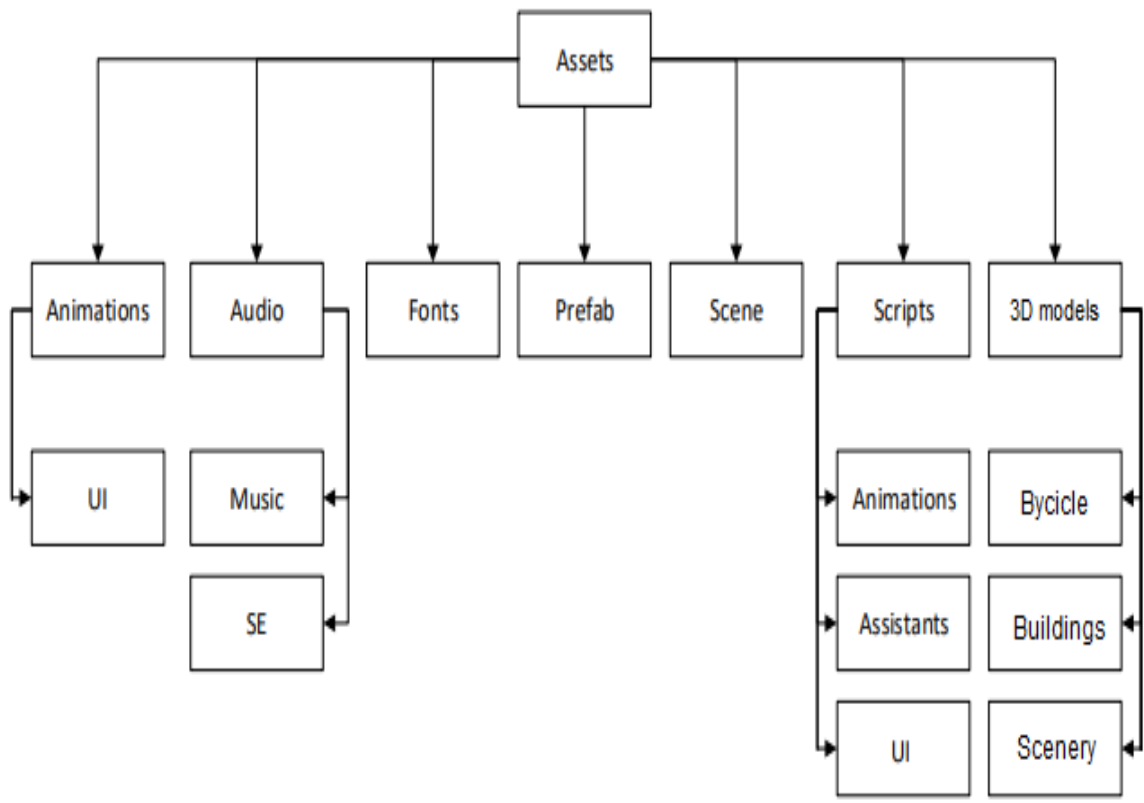


Рисунок 15 – Файловая структура

4.1.2. РЕАЛИЗАЦИЯ ИНТЕРФЕЙСОВ

В ходе создания игры для взаимодействия пользователя с программой было создано главное меню, показанное на рисунке 16, которое включает в себя следующие пункты:

- игра;
- прогресс игры;
- настройки;
- ВЫХОД.

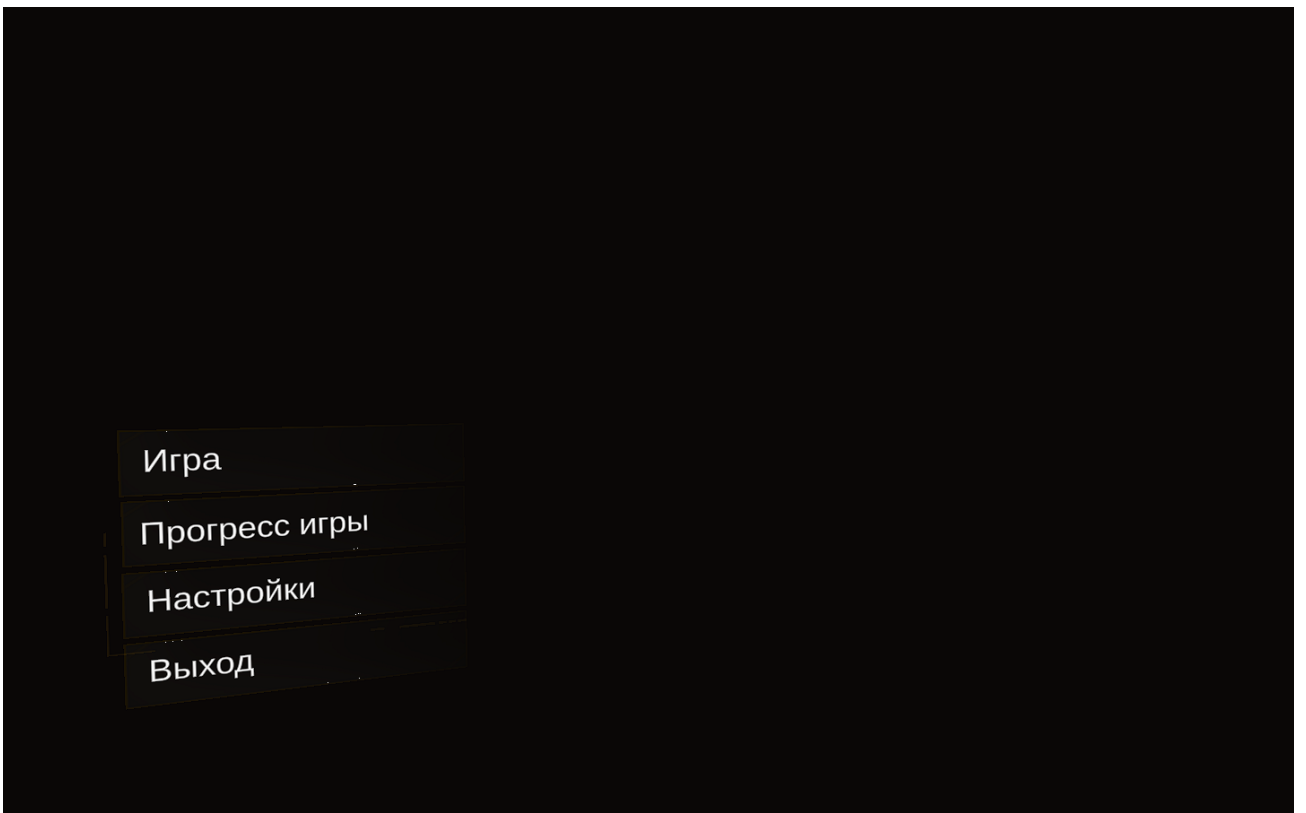


Рисунок 16 - Главное меню

После нажатия на пункт настройки, пользователь попадает в подменю, в котором можно выбрать настройки игры, управления и графики. Они представлены на рисунках 17-19.

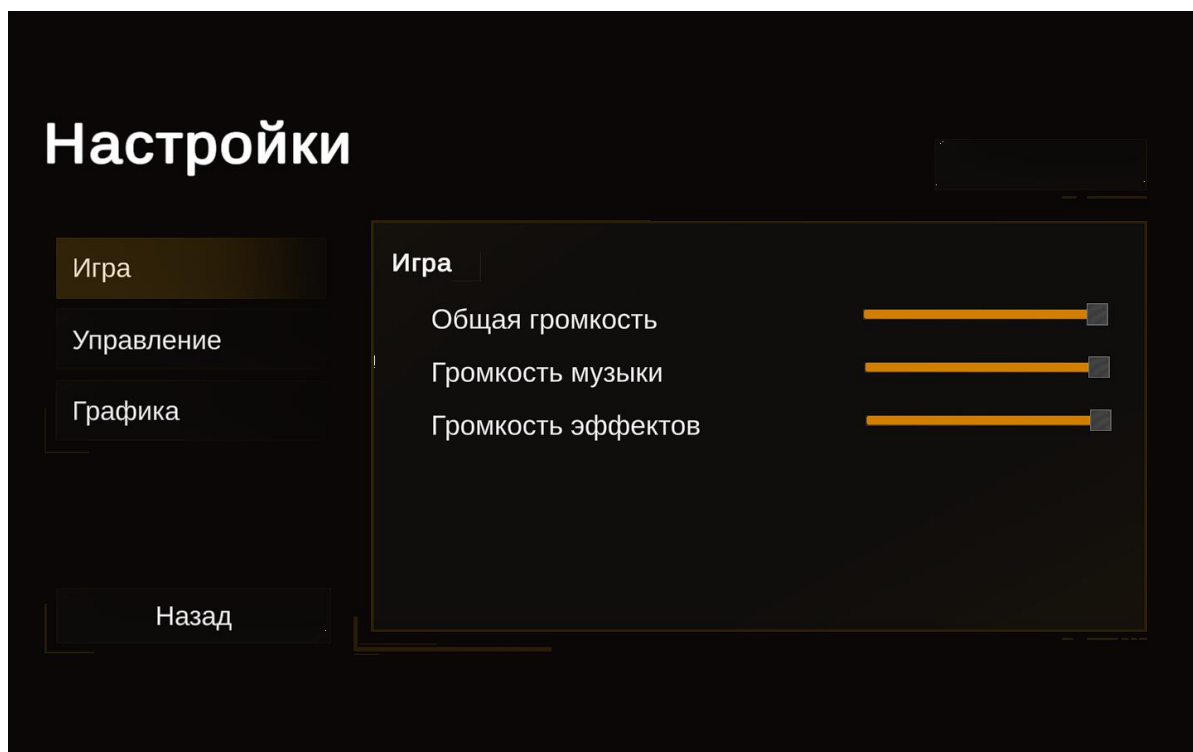


Рисунок 17 – Настройки игры

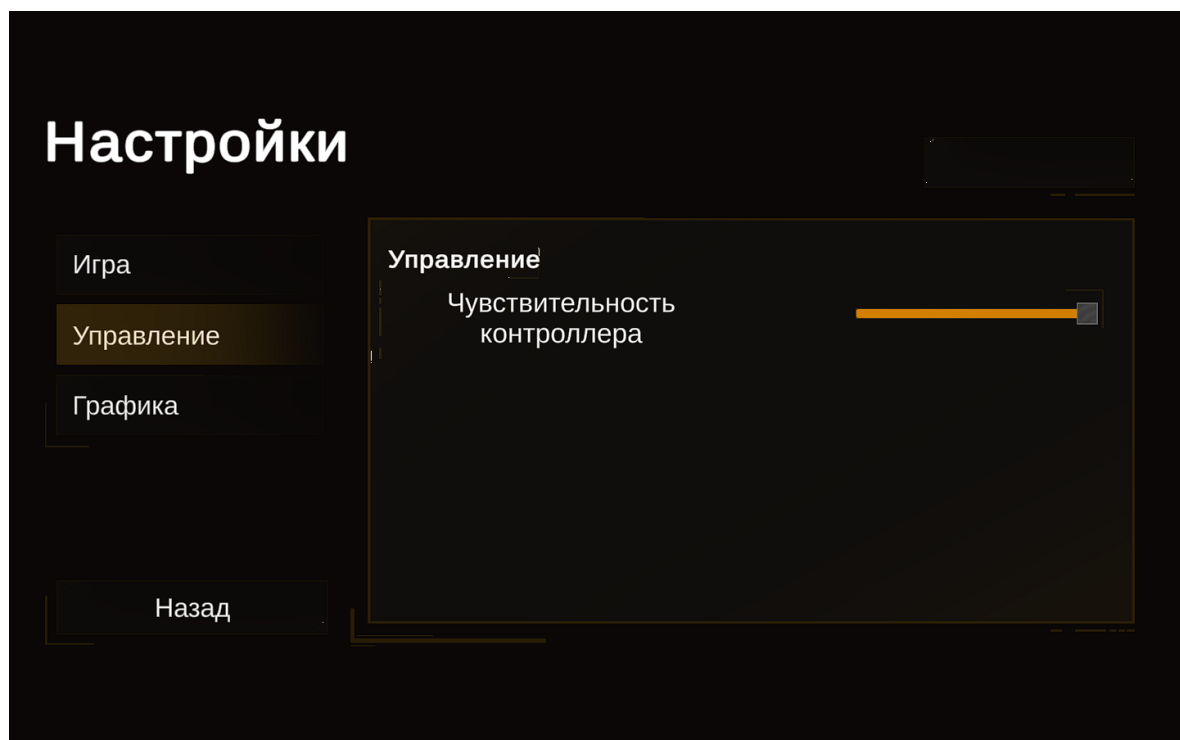


Рисунок 18 – Настройки управления

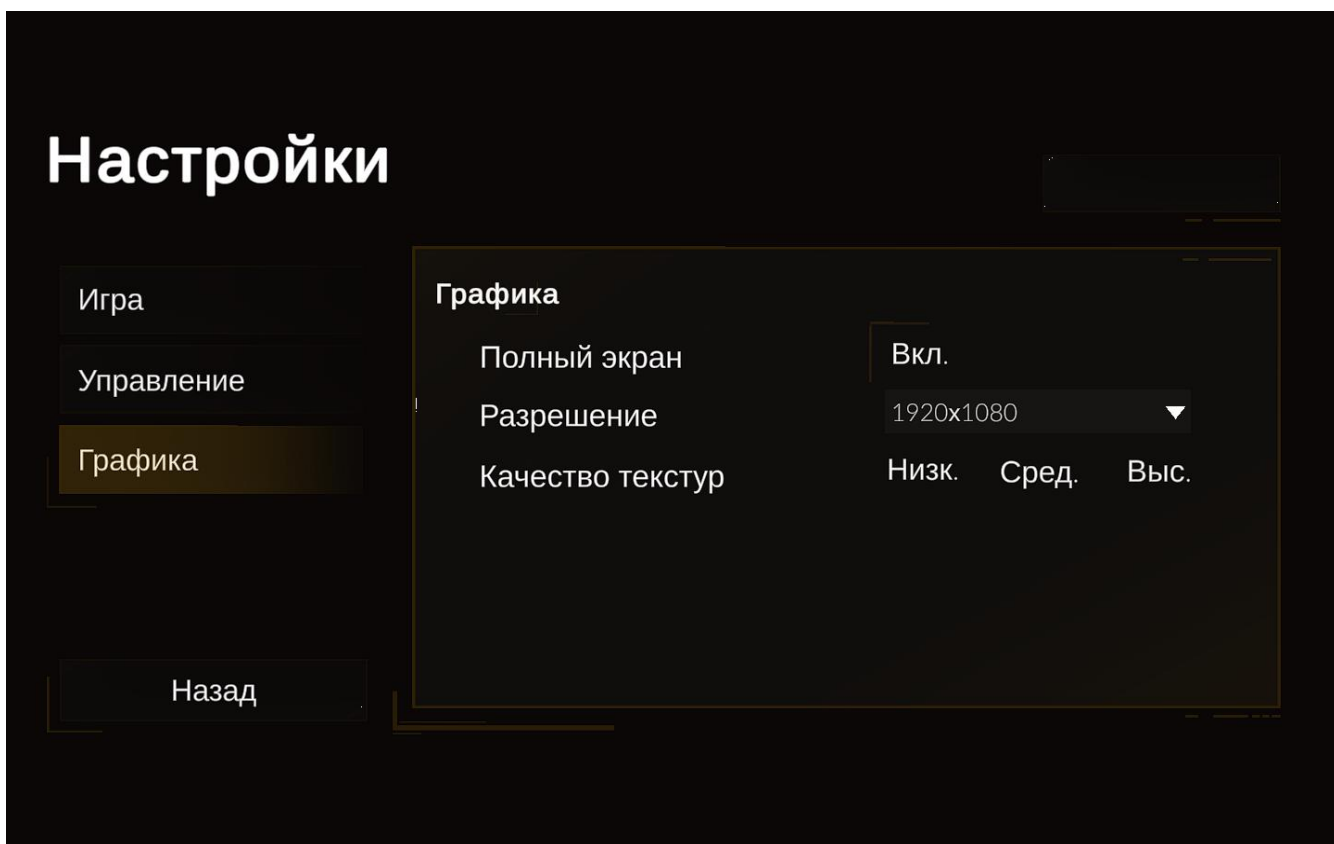


Рисунок 19 – Настройки графики

При выборе пункта прогресса игры, пользователь может увидеть свои общие результаты, а также результаты за последнюю игровую сессию. Они представлены на рисунках 20 и 21.

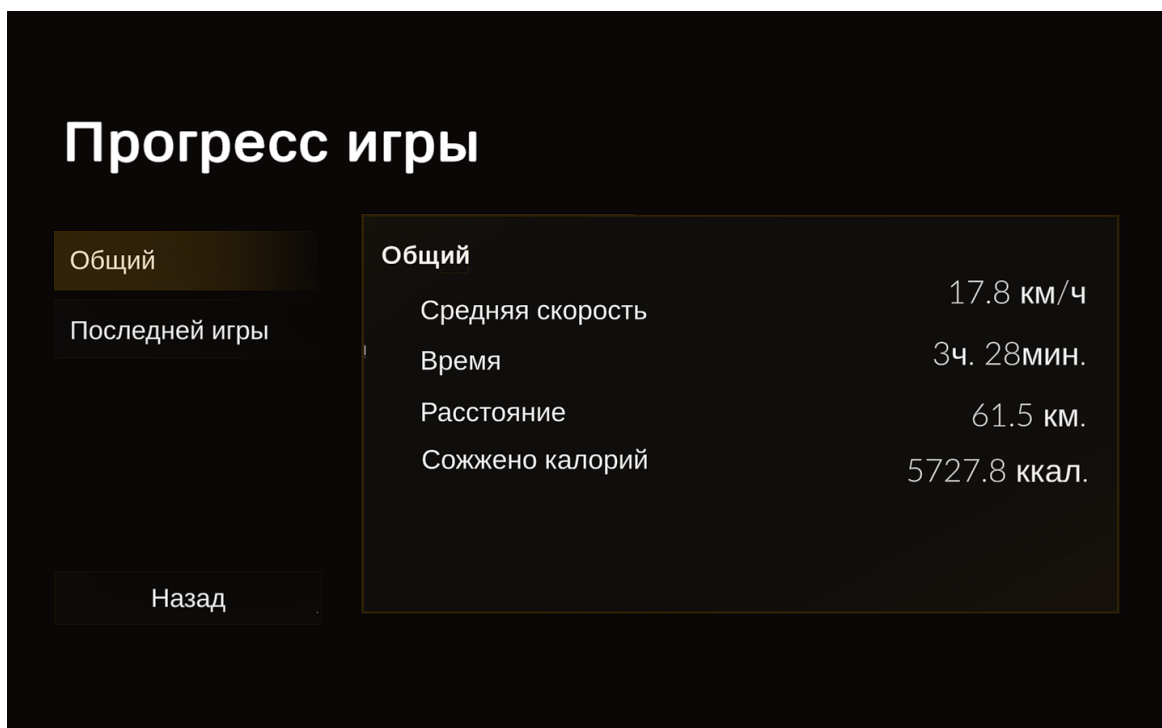


Рисунок 20 – Общий прогресс игры

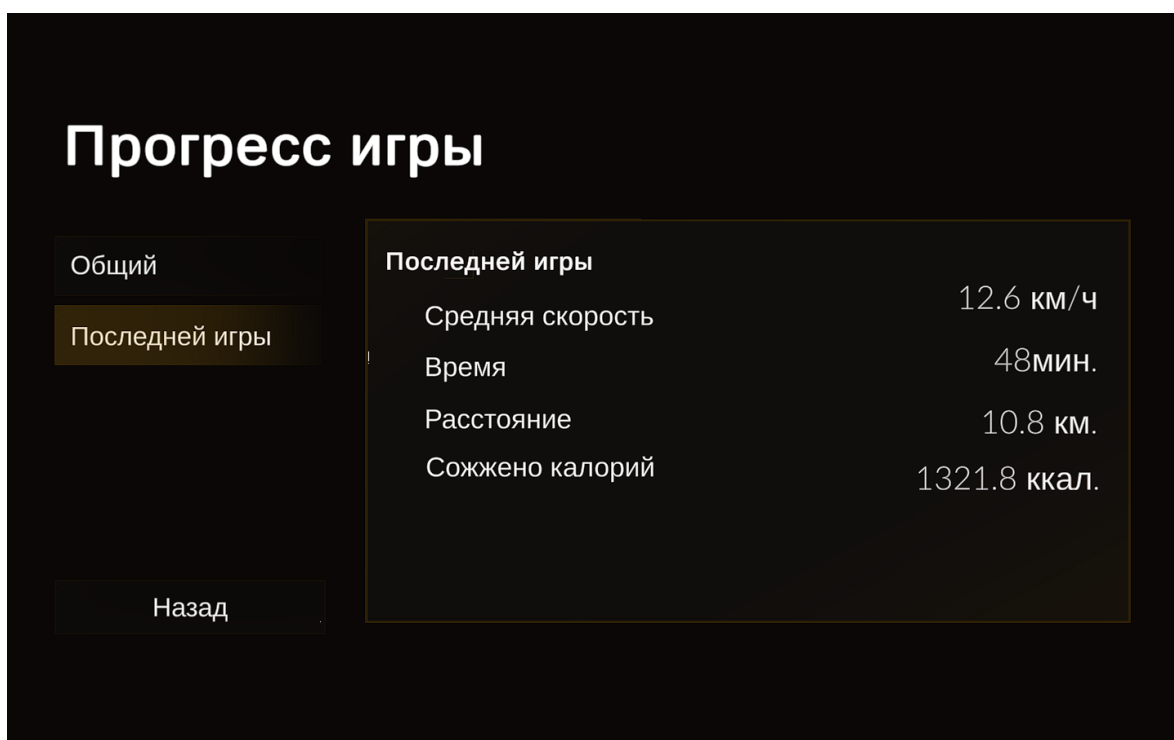


Рисунок 21 – Прогресс за последнюю игровую сессию

На рисунке 22, показано, как после нажатия на пункт игра, перед пользователем появляется экран загрузки, после которого, при нажатии любой из кнопок, он перейдет непосредственно к игре.



Рисунок 22 – Экран загрузки игры

При запуске игры, пользователь видит перед собой город, модель велосипеда, которым может управлять через подключенный велотренажер посредством USB интерфейса. Также, в левом нижнем углу изображен велокомпьютер, который показывает системное время, скорость в км/ч и расстояние в километрах. Процессы изменения этих показателей отображены на рисунках 23 и 24.



Рисунок 23 - Пункт игра



Рисунок 24 – Изменение показателей велокомпьютера

При нажатии паузы, игра переходит в меню, из которого можно так же попасть в настройки или выйти из игры, примером настроек из меню паузы служит рисунок 25.

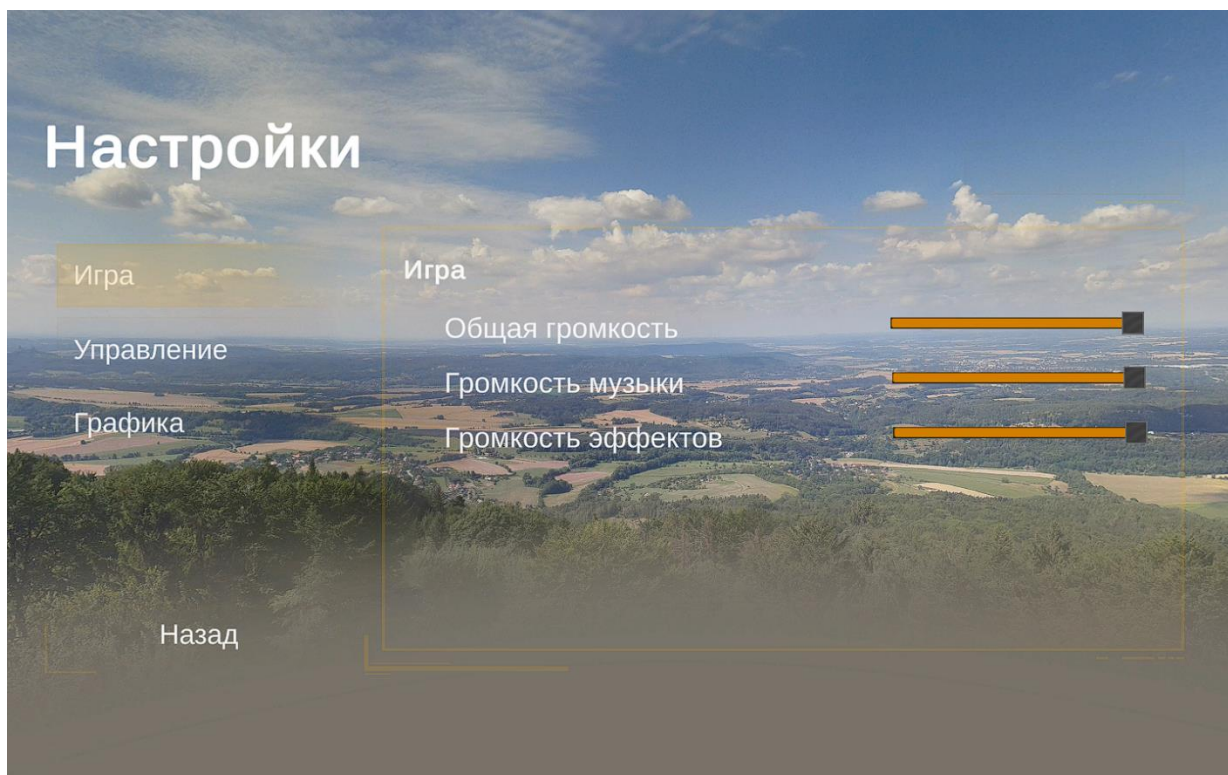


Рисунок 25 – Настройки в меню паузы

4.2. РЕАЛИЗАЦИЯ АППАРАТНОЙ ЧАСТИ

Для проверки работоспособности устройства, был собран опытный образец. Его реализация демонстрируется на рисунках 26 и 27.



Рисунок 26 – Датчик Холла

После прокрутки педали, на котором закреплен магнит, датчик Холла передает сигнал в игру для управления виртуальной моделью велосипеда. Чем больше скорость прокрутки педалей, тем быстрее начинает двигаться модель.

Для управления поворотами, были подключены кнопки, они прикреплены к рулю велотренажера, это показано на рисунке 27.



Рисунок 27 – Кнопки управления велосипедом в игре

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 ТЕСТИРОВАНИЯ

Результаты тестирования:

Тест №1. Передвижение пользователя в игре.

Входные данные: пользователь крутит педали на велотренажере.

Ожидаемый результат: модель велосипеда в игре начнет движение вперед.

Полученный результат: совпадает с ожидаемым.

Тест пройден успешно.

Тест №2. Пользователь нажимает кнопки поворота на велотренажере.

Входные данные: пользователь нажимает кнопки поворота.

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. Изменение громкости.

Входные данные: пользователь находится в настройках игры.

Ожидаемый результат: пользователь перемещает ползунок громкости и громкость изменяется.

Полученный результат: совпадает с ожидаемым.

Тест пройден успешно.

Тест №5. Переход игры в оконный режим.

Входные данные: пользователь находится в настройках графики.

Ожидаемый результат: пользователь нажимает кнопку «в окне» и игра переходит в такой режим.

Полученный результат: совпадает с ожидаемым.

Тест пройден успешно.

Тест №6. Нажатие одновременно двух кнопок поворота.

Входные данные: пользователь нажимает две кнопки управления одновременно.

Ожидаемый результат: модель велосипеда не поворачивает.

Полученный результат: совпадает с ожидаемым.

Тест пройден успешно.

Тест №7. Навигация в игре.

Входные данные: пользователь находится в режиме игры.

Ожидаемый результат: пользователь нажимает кнопку паузы и появляется окно меню.

Полученный результат: совпадает с ожидаемым.

Тест пройден успешно.

Тест №8. 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 микроконтроллере digispark ATtiny85.

Для достижения этих целей, нужно было выполнить следующие задачи:

- провести обзор литературы необходимой для решаемых задач;
- провести анализ и выбор средств реализации устройства управления;
- провести анализ и выбор средств реализации игры;
- спроектировать аппаратную систему;
- спроектировать программную систему;
- реализовать программно-аппаратный комплекс;
- провести тестирование программно-аппаратного 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, добавив туда онлайн составляющую, а также искусственный интеллект. Для распространения игры и продажи готового устройства будет создан сайт.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Малоподвижность смертельно опасна. – https://www.bbc.com/russian/science/2012/07/120718_inactivity_research_lancet. – Дата обращения: 02.03.2020.
2. Всемирная организация здравоохранения. – <https://www.who.int/ru/news-room/detail/22-11-2019-new-who-led-study-says-majority-of-adolescents-worldwide-are-not-sufficiently-physically-active-putting-their-current-and-future-health-at-risk>. – Дата обращения: 20.04.2020.
3. Лучший игровой движок по версии пользователей хабра. – <https://habr.com/ru/post/307952>. – Дата обращения: 02.03.2020.
4. Unreal Engine. – <https://www.unrealengine.com/en-US>. – Дата обращения: 04.03.2020.
5. Unity Manual. – <https://docs.unity3d.com>. – Дата обращения: 04.03.2020.
6. Хокинг, Д. Unity в действии. Мультиплатформенная разработка на С#/ Д. Хокинг. - Питер, 2016 – 336 с.
7. Алан, Т. Искусство создания сценариев в Unity / Т. Алан. - Питер, 2016 – 360 с.
8. С#. – <https://docs.microsoft.com/ru-ru/dotnet/csharp>. – Дата обращения: 10.03.2020.
9. Агуров, П. С#. Сборник рецептов / П. Агуров. - М.: "БХВ-Петербург", 2012. - 432 с.
10. Вагнер, Б. С# Эффективное программирование / Б. Вагнер. - М.: ЛОРИ, 2013. - 320 с.

11. Рендольф, Н. Visual Studio 2010 для профессионалов / Н. Рендольф. - М.: Диалектика / Вильямс, 2016. - 516 с.
12. GitHub. – <https://github.com/pricing>. Дата обращения: 14.03.2020.
13. Photoshop. – <https://www.adobe.com/ru/products/photoshopfamily.html>. – Дата обращения: 16.03.2020.
14. 55 уроков по Adobe Photoshop для новичков – <https://infogra.ru/lessons/55-urokov-po-adobe-photoshop-dlya-novichkov>. – Дата обращения: 17.03.2020.
15. Adobe Photoshop. – <https://works.doklad.ru/view/VYTLuuzaZQ0.html>. – Дата обращения: 17.03.2020.
16. Adobe Photoshop CC. Официальный учебный курс // Эксмо, 2013 - 456 с
Bandler, W. Fuzzy power sets and fuzzy implication operators / W. Bandler, L. Kohout // Fuzzy Sets and Systems. – 1980. – V. 4. – P. 13 – 30.
17. Кошелев, В. Е. Самоучитель Photoshop CS2 / В.Е. Кошелев. - М.: Бином-Пресс, 2012. - 576 с.
18. Adobe Photoshop Adobe Photoshop CC. Официальный учебный курс // Эксмо, 2013 - 456 с.
19. 3ds-Max. – <https://www.autodesk.ru/products/3ds-max/overview>. – Дата обращения 22.03.2020.
20. Верстак, В. 3ds Max 8 на 100% (+ CD-ROM) / В. Верстак, М. Бондаренко, С. Бондаренко. - СПб: Питер - Москва, 2006. - 416 с.
21. Шнейдеров, В. Иллюстрированный самоучитель 3ds max / В. Шнейдеров. - СПб: Питер - Москва, 2006. - 480 с.
22. Маров, М. Энциклопедия 3ds max 6 / М. Маров. - СПб: Питер - Москва, 2006. - 525 с.

23. Campbell Parallel Programming with Microsoft® Visual C++® / Campbell. - Москва: Гостехиздат, 2011. - 784 с.
24. Петин, В.А. Проекты с использованием контроллера Arduino /В.А. Петин. — 2-е изд. — СПб: БХВ-Петербург, 2015. — 400 с.
25. Arduino. – <http://robotosha.ru/arduino/schematic-into-fritzing.html>. – Дата обращения 22.03.2020.
26. VirZOOM. – <https://www.virzoom.com>. – Дата обращения: 25.03.2020.
27. ZWIFT. – <https://www.zwift.com/eu>. – Дата обращения: 25.03.2020.