

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

\_\_\_\_\_ 2021 г.  
«\_\_» \_\_\_\_\_

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой ЭВМ

\_\_\_\_\_ Г.И. Радченко  
«\_\_» \_\_\_\_\_ 2021 г.

Разработка компьютерной игры для мобильных платформ «Goway»  
на платформе Unity

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Руководитель работы,

к.п.н., доцент каф. ЭВМ

\_\_\_\_\_ Ю.Г. Плаксина  
«\_\_» \_\_\_\_\_ 2021 г.

Автор работы,

студент группы КЭ-405

\_\_\_\_\_ М.Л. Мохамад  
«\_\_» \_\_\_\_\_ 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 бакалавра**

студенту группы КЭ-405

Мохамад Мустафа Лаитх Мохамад

обучающемся по направлению

09.03.01 «Информатика и вычислительная техника»

- 1. Тема работы:** «Разработка компьютерной игры для мобильных платформ «Goway» на платформе Unity» утверждена приказом по университету от 26 апреля 2021 г. № 714-13/12
- 2. Срок сдачи студентом законченной работы:** 1 июня 2021 г.
- 3. Исходные данные к работе:**
  - Вагнер Б.М. С# Эффективное программирование – М.: Издательство ЛОРИ, 2017. - 320 с.;
  - Мартин Р.С Чистый код: создание, анализ и рефакторинг. – М.: Издательство Питер, 2018. – 464 с.;
  - Хокинг Д.Р. Unity в действии. – Спб: Издательство Питер, 2016. – 704 с.

**4. Перечень подлежащих разработке вопросов:**

- изучить историю возникновения компьютерных игр;
- провести анализ предметной области;
- спроектировать и разработать компьютерную игру;
- протестировать работоспособность разработанной компьютерной игры в нагрузочном тестировании.

**5. Дата выдачи задания:** 1 декабря 2020 г.

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

Студент \_\_\_\_\_/М.Л. Мохамад/

## КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	01.03.2021	
Разработка модели, проектирование	01.04.2021	
Реализация системы	01.05.2021	
Тестирование, отладка, эксперименты	15.05.2021	
Компоновка текста работы и сдача на нормоконтроль	24.05.2019	
Подготовка презентации и доклада	30.05.2021	

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

Студент \_\_\_\_\_ /М.Л. Мохамад/

## Аннотация

М. Л. Мохамад. Разработка компьютерной игры для мобильных платформ «Goway» на платформе Unity. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭКН; 2021, 34 с., 6 ил., библиографический список – 26 наим.

Выпускная квалификационная работа была выполнена с целью разработки компьютерной игры.

В данной работе были изучены существующие методы разработки компьютерных игр, выполнен анализ требований, проектирование и тестирование разработанного приложения. Для разработки использовался движок Unity. Целью работы является создание однопользовательского платформера для персональных компьютеров, содержащий в себе 1 уровень с реализованной основной игровой механикой.

Результатом выполненной работы является правильно функционирующая компьютерная игра.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	7
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	9
1.1 ОБЗОР АНАЛОГОВ .....	9
1.2 АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ .....	12
1.3 ВЫВОД.....	14
2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ .....	16
2.1 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ.....	16
2.2 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ .....	16
3. ПРОЕКТИРОВАНИЕ .....	18
3.1 АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ .....	18
3.2 АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ .....	19
4. РЕАЛИЗАЦИЯ ИГРЫ .....	22
5. ТЕСТИРОВАНИЕ .....	27
6. ЗАКЛЮЧЕНИЕ .....	28
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	29
ПИЛОЖЕНИЕ .....	32

## ВВЕДЕНИЕ

Сейчас стремительно развиваются информационные технологии. Они позволяют быстро решать поставленные задачи, экономить время. Вместе с увеличением рынка персональной электроники так же стремительно быстро растет рынок развлечений – поэтому осуществление проектов по созданию игр в настоящее время очень актуально и можно смело утверждать, что востребованность таковых не уменьшится.

В данной работе рассматривается создание компьютерной игры жанра «платформер» на движке Unity. Игра написана с использованием языка программирования C#.

Аркада (англ. arcade game, arcade genre) — жанр компьютерных игр, характеризующийся коротким по времени, но интенсивным игровым процессом. Платформеры являются поджанром аркады [9].

Платформеры (или «аркады», как их часто называют в исконно-компьютерных журналах) – один из старейших и наиболее востребованных жанров в игровой индустрии. «Марио», «Соник», «Донки Конг», «Кирби», «Крэш» и «Спайро» – все это платформеры, полюбившиеся многим геймерам еще с детства и уже ставшие золотой классикой [18].

Чтобы не тратить время и не разрабатывать приложение с нуля, разработчики пользуются готовыми решениями для игр – игровыми платформами. Одно из популярных решений предоставляет компания Unity Technologies, которая в июне 2005 года выпустила платформу разработки для создания 2D- и 3D игр – Unity. Платформа Unity обладает широким спектром возможностей, удобным и интуитивно понятным интерфейсом. Большим преимуществом Unity является кроссплатформенная разработка, которая делает легким и быстрым портирование игр под такие платформы как Windows, iOS, Android, Windows Phone 8, а также разрабатывать игры для Xbox, PS, Nintendo, веб-браузеров и даже VR [26].

C# – простой, современный объектно-ориентированный язык программирования, но поддерживает также и компонентно-ориентированное программирование. Разработка современных приложений все больше тяготеет к созданию программных компонентов в форме автономных и самоописательных пакетов, реализующих отдельные функциональные возможности. Важная особенность таких компонентов – это модель программирования на основе свойств, методов и событий. Каждый компонент имеет атрибуты, предоставляющие сведения о компоненте, а также встроенные элементы документации. C# предоставляет языковые конструкции, непосредственно поддерживающие такую концепцию работы. Благодаря этому C# отлично подходит для создания и применения программных компонентов [23].

Цель работы: создание компьютерной игры.

Задачи работы:

- провести обзор аналогичных решений и осуществить постановку задачи;
- провести обзор современных средств реализации;
- провести анализ требований и спроектировать компьютерную игру;
- реализовать компьютерную игру;
- провести тестирование реализованной компьютерной игры.



# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 ОБЗОР АНАЛОГОВ

Runner – один из популярных игровых жанров, в центре внимания которых герой-бегун. Практически всегда персонаж бежит автоматически и от игрока требуется лишь корректировать его траекторию движения, вовремя совершать прыжки и уклоняться от препятствий. Это требует в особенности концентрации внимания и хорошей реакции [19].

Рассмотрим самые популярные компьютерные игры этого жанра:

Rayman Origins – игра-платформер из игровой серии Rayman, рассказывающая о приключениях Рэймана и Глобокса. Разработка игры велась на игровом движке UbiArt Framework. Игра была выпущена на Xbox 360, PlayStation 3 и Wii 15 ноября 2011 года. Позже она была выпущена также на Microsoft Windows, PlayStation Vita и Nintendo 3DS.

Главной задачей героя в первой половине является нахождение и освобождение фей, а во второй – королей этих земель. На уровне герой может подобрать множество бонусов: электуны и люмы [4].

Плюсы: динамичный и затягивающий геймплей; дизайн уровней; в меру высокая сложность. Минусы: система сохранений; отсутствие нормального кооператива.

Despicable Me: Minion Rush (с английского языка – Мчащийся Миньон) – аркадный платформер с видом от третьего лица. Это кроссплатформенная игра, разработанная GameLoft. GameLoft использует игровой движок Unreal Engine. Доступна на трех основных консольных системах мобильных игр Windows Phone, Android и iOS. Игровая реальность входит во многие топы от Google Play и App Store. Вам предстоит играть за миньонов Грю в бесконечных забегах. В процессе этих забегов собирайте монетки и разные бонусы. Чем больше монеток соберёте, тем больше сможете улучшить своего персонажа [1].

Плюсы: полностью трехмерный раннер с анимацией персонажей из мультфильма, озвучкой и красочной графикой; простой в освоении игровой процесс; Минусы: freemium-модель навязывает встроенные покупки; игра тормозит на бюджетных планшетах и телефонах, но иногда глючит и на современных устройствах.

Ori and the Blind Forest (с англ. – «Ори и непроглядный лес») – компьютерная игра в жанрах платформер и метроидвания, разработанная студией Moon Studios на базе движка Unity и изданная Microsoft Studios. Выпуск игры для платформ Windows и Xbox One состоялся 11 марта 2015 года, а 27 сентября 2019 года игра была выпущена на платформе Nintendo Switch.

Игрок управляет небольшим белым существом Ори и небольшим лесным духом Сейном, который его защищает. Игровая механика представляет собой типичный платформер, игрок перемещается по большой карте преодолевая различные препятствия и сражается с различными врагами.

Игра отличается красивым оформлением, что, несомненно, выделяет её среди прочих игр данного жанра. За отличным сюжетом, рассказываемом в игре, интересно наблюдать, а звуковое сопровождение отлично задает нужную атмосферу. Минусом можно считать лишь несколько усложненный геймплей из-за неинтуитивного управления [5].

Результаты обзора существующих аналогов представлены в таблице 1.

Таблица 1 - Результаты обзора существующих аналогичных продуктов

Название игры	Кроссплатформенность	Игровой движок
Rayman Origins	+	UbiArt Framework
Despicable Me: Minion Rush	+	Unreal Engine
Ori and the Blind Forest	+	Unity

Проведя анализ существующих разработок, стало понятно, что чистых платформеров в их первоначальном виде стало очень мало, так как они перестали отвечать возросшим требованиям игроков. Чистыми платформерами на данный момент являются в основном переиздания старых популярных игр, которые играют больше на ностальгии игроков.

Однако нельзя сказать, что популярность платформеров как жанра среди игроков сильно упала. Действительно, игр, соответствующих именно этой жанровой категории сейчас не так много, но в тоже время стало выходить больше игр, которые можно причислить не только к платформерам, но и к другим жанрам. Это говорит о том, что жанр «Платформер» стал более нишевым продуктом, чем массовым, а значит для получения больших прибылей, разработчикам приходится добавлять элементы сторонних жанровых категорий видеоигр.

Делая заключительный вывод, можно понять, что для того что бы игра была успешной, нужно разнообразить продукт как геймплейно, так и по возможности добавить интересный и захватывающий сюжет.

Для разработки компьютерной игры основным средством является игровой движок – программное средство, для разработки компьютерных игр.

Игровые движки сразу включают в себя все необходимые алгоритмы для правильного функционирования игры и её разработки.

## 1.2 АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ

В настоящее время существует много платформ для разработки игр. Они значительно облегчают процесс разработки приложения за счёт экономии времени и трудовых ресурсов.

1. Unity – инструмент для создания 2D- и 3D-игр, а также интерактивного контента. Платформа поддерживает два сценарных языка: C# и JavaScript (Модифицирован для Unity).

Плюсы:

- выгодная лицензионная политика;
- легкость использования;
- совместимость с любой платформой;
- популярен среди разработчиков (это означает, что ошибки быстро находят и исправляют).

Минусы:

- ограниченный набор инструментов;
- процесс разработки игры отнимает много времени [14].

2. Unreal Engine позволяет создавать игры для большинства операционных систем и платформ, а также для различных портативных устройств, например, устройствах Apple (iPad, iPhone), управляемых системой iOS и прочих.

Платформа поддерживает различные системы процесса получения изображений (Direct3D, OpenGL, Pixomatic), воспроизведения звука (EAX, OpenAL, DirectSound3D), средств голосового воспроизведения текста, распознавания речи, и имеет различные модули для работы с сетью и поддержки различных устройств ввода.

Плюсы:

- одно из лучших сообществ разработчиков;
- отличная техподдержка;

- совместимость с любой платформой;
- новые инструменты выходят с каждым обновлением;
- широкий ассортимент инструментов для различных целей.

Минусы:

- некоторые разработчики жалуются, что к определенным инструментам сложно привыкнуть.
- подписка стоимостью 19 \$ в месяц и 5 %, если игра заработает более 5,000\$ [14].

3. CryENGINE – невероятно мощная платформа для разработки игр. С помощью данной платформы можно создавать игры под ПК и консоли. Графические особенности CryENGINE значительно превосходят возможности первых двух платформ, поскольку включают архаусный свет, реалистичную физику, продвинутую систему анимации и прочее.

Плюсы:

- функция Flowgraph поможет украсить игру отличной графикой;
- набор функций Fmod для создания мощного звукового сопровождения;
- самый простой процесс создания AI в сегменте;
- начинающему разработчику легко сделать UI.

Минусы:

- относительно небрежная техподдержка бесплатной версии;
- поскольку движок в индустрии сравнительно недавно, ему еще
- только предстоит создать крепкое сообщество;
- относительно высокий порог вхождения [14].

### 1.3 ВЫВОД

В данной работе рассматривается создание компьютерной игры на движке Unity.

Unity – это среда для разработки компьютерных игр, в которой объединены различные программные средства, используемые при создании программного обеспечения – текстовый редактор, компилятор, отладчик и так далее [12].

Движок Unity доступен для Windows и Mac и имеет удобный графический интерфейс, плавный и изменяемый в формате, который подходит разработчику, движок поддерживает разработку 2D и 3D игр и предоставляет инструменты, необходимые для каждого типа отдельно, и содержит инструменты для искусственного интеллекта, такие как выбор лучшего и близкого маршрута с препятствиями, он предоставляет пользовательские инструменты, которые можно вставить в сцену, чтобы начать с простого изменения их настроек, а движок также предоставляет специальные инструменты для создания ярлыков пользовательского интерфейса из пользовательского интерфейса, которые остаются на экране игрока независимо от движения внутренних объектов и ваших потребностей [17].

Первое преимущество – движок Unity3D дает возможность разрабатывать игры, не требуя для этого каких-то особых знаний. Здесь используется компонентно-ориентированный подход, в рамках которого разработчик создает объекты (например, главного героя) и к ним добавляет различные компоненты (например, визуальное отображение персонажа и способы управления им). Благодаря удобному Drag & Drop интерфейсу и функциональному графическому редактору движок позволяет рисовать карты и расставлять объекты в реальном времени и сразу же тестировать получившийся результат.

Второе преимущество – наличие огромной библиотеки ассетов и плагинов, с помощью которых можно значительно ускорить процесс разработки игры. Их можно импортировать и экспортировать, добавлять в игру целые заготовки – уровни, врагов, паттерны поведения ИИ и так далее. Никакой возни с программированием. Многие ассеты доступны бесплатно, другие предлагаются за небольшую сумму, и при желании можно создавать собственный контент, публиковать его в Unity Asset Store и получать от этого прибыль.

Третье преимущество – поддержка огромного количества платформ, технологий, API. Созданные на движке игры можно портировать между ОС Windows, Linux, OS X, Android, iOS, на консоли семейств PlayStation, Xbox, Nintendo, на VR- и AR-устройства. Unity поддерживает DirectX и OpenGL, работает со всеми современными эффектами рендеринга, включая новейшую технологию трассировки лучей в реальном времени [17].

## **2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ**

К общим требованиям для персонального компьютера можно отнести минимальные системные требования:

- операционная система (ОС): Windows 7 или выше/Linux;
- процессор: Intel Core i3 540/AMD A6-3620 или лучше;
- оперативная память: 4 GB;
- видеокарта: видеокарта с поддержкой DirectX 11.0 с 1 GB RAM (NVidia GeForce 460/ AMD Radeon 6850) или лучше;
- DirectX: версии 11;
- место на диске: 2GB;
- звуковая карта: звуковая карта совместимая с DirectX.

Данные системные требования являются минимальными для продукции, которая создается на движке Unity.

### **2.1 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ**

В рамках реализуемого приложения предусмотрен один пользователь, который взаимодействует с проигрываемым сюжетом игры и компонентами меню.

Пользователь может:

- начать уровень 1;
- начать уровень 2;
- настроить игру;
- выйти из игры.

### **2.2 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ**

К нефункциональным требованиям относятся:

- разработанное приложение должно соответствовать определенным минимальным системным требованиям;



- разработанное приложение должно быть написано на языке C# на платформе Unity;
- разработанное приложение должно иметь процедурную генерацию уровней.

### 3. ПРОЕКТИРОВАНИЕ

#### 3.1 АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ

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

- шаблоны объектов;
- игровые сцены;
- скрипты;
- изображения персонажей и фонов;
- анимации.

На рисунке 1 представлена файловая структура приложения.

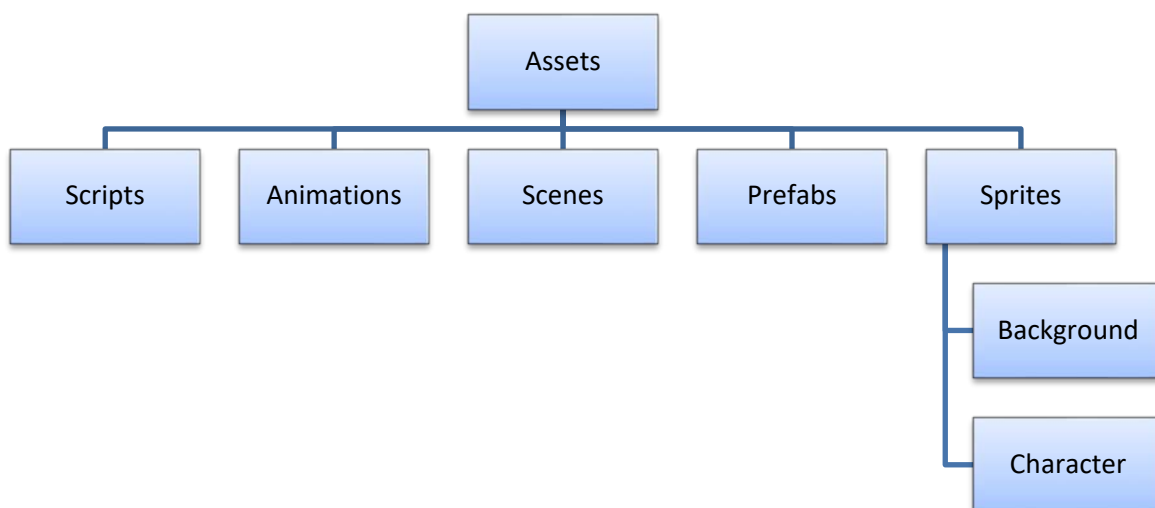


Рисунок 1 – Файловая структура приложения

В директории *Scripts* содержатся скрипты, которые используются в игровом приложении.

В директории *Animations* содержатся анимации персонажа.

В директории *Scenes* содержатся игровые сцены, в которые пользователь может перейти.

В директории Prefabs содержатся готовые шаблоны игровых объектов, при помощи которых можно составить игровой уровень после его проектирования.

В директории Sprites содержатся подкаталоги:

- Background содержит в себе базовые блоки платформ;
- Character содержит в себе спрайт главного персонажа.

## **3.2 АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ**

Компьютерная игра – это компьютерная программа, которая служит для организации игрового процесса (геймплея), связи с партнерами по игре, или сама выступает в качестве партнера.

Игростроение – трудоемкий процесс, требующий от разработчиков полной отдачи и внимания, даже к самым незначительным мелочам. Неважно будь то мелкие детали персонажей или крупные элементы окружающей обстановки. Звук занимает одну из первых строчек в разработке игр.

Хорошо подобранный звук меняет восприятие происходящего, будь то мелкий шорох или какой-то грохот. Игрок может моментально отреагировать на тот или иной звук определенными действиями. Опытный игрок не обращает внимания на определенные действия, сопровождаемые звуком. Ему достаточно лишь услышать, и он сразу определит, какое игровое событие произошло.

Следующая задача – это графика, визуальная составляющая всего проекта, то, что игрок непосредственно видит перед собой. И первое что он видит перед собой в игре это главное меню. Меню в игре должно быть понятным, а его стиль обычно соответствует самой игре. В порядке вещей, если игра, затрагивающая средневековую тематику, будет использовать на заднем плане изображения замков или рыцарей, а сами кнопки и курсор могут представлять собой щиты и меч, а если она затрагивает современные военные действия, меню можно представить в виде тактической карты. Со

временем как-то само собой утвердились и основные кнопки меню, практически во всех играх их число составляет от пяти до семи. Меньше, много подменю, больше, слишком много информации, да и пользователи привыкли, что игровые настройки видео и аудио находятся в подменю кнопки под названием «параметры» или «настройки». Персонажи тоже должны быть сделаны под стать игре, если игра в двух мерном пространстве будут трех мерные модели, это может оказаться не совсем к месту.

Следующая часть – это внутренняя начинка, а именно программный код, насколько он громоздкий, хорошо ли оптимизирован. Скорость работы, если загрузка каждого уровня будет происходить в течение пары десятков минут, у игроков не хватит терпения. Оптимизация, любой программист старается сделать свой код понятным хотя бы для себя, убрать пару лишних переменных, усовершенствовать некоторые алгоритмы под поставленную задачу, потому что любой выпуск программного продукта, в любой сфере, подразумевает сопровождение. Будут выпускаться обновления и дополнения, которые затрагивают оригинальный код.

Тестирование – это еще один вид, как разработки, так и сопровождения. В объектно-ориентированном программировании тестируются классы и методы, это на много упрощает процесс. На некоторых движках разработки классы называются скриптами. Скрипты прикрепляются к объектам. К объекту может быть прикреплено много скриптов и на оборот, несколько объектов могут использовать один скрипт. Тестированию отведен целый большой пункт в разработке любого ПО, разработаны, международные стандарты, ГОСТы, методы и алгоритмы тестирования. Стандарты определяют две связанные модели качества программного обеспечения (ISO 9126–01) – внутреннее качество, внешнее качество, а также набор соответствующих работ по оценке качества программного обеспечения (ISO14598–98) [11]. Качество программного обеспечения, управление

качеством и качество программной инженерии имеют непосредственное отношение к качеству создаваемого программного продукта.

Игровое приложение – это сложный программный продукт, местами даже сложнее некоторых программ, как в разработке, так и в продаже. Они требуют от разработчиков полной самоотдачи и терпения [21].

#### 4. РЕАЛИЗАЦИЯ ИГРЫ

Для разработки игрового приложения была выбрана платформа Unity. Данная платформа имеет низкий порог вхождения, Personal Edition (бесплатную) версию, большое количество обучающих материалов и сообщество разработчиков, в следствии чего с ней можно быстрее начать работать [25].

Название игры: «Goway».

Главным персонажем является «Кубик», который на своем пути собирает такие же по цвету кубики.

Цель игры – набрать наибольшее количество очков.

Игра состоит из двух уровней.

Смысл данной игры: управлять персонажем «Кубик» и помочь ему наибольшее количество кубиков на своем пути. «Кубик» может двигаться вправо и влево. Если «Кубик» столкнется с преградой или с кубиками другого цвета, то игра заканчивается.

В ходе создания приложения для взаимодействия программы с игроком были созданы следующие интерфейсы:

- главное меню;
- интерфейс персонажа.

Для большего удобства обращения игрока с приложением было разработано главное меню, включающее в себя следующие пункты:

- начать игру (“Level 1”, Level 2”);
- настройки (“Options”);
- выход из игры (“Quit”).

На рисунке 2 представлено главное меню.

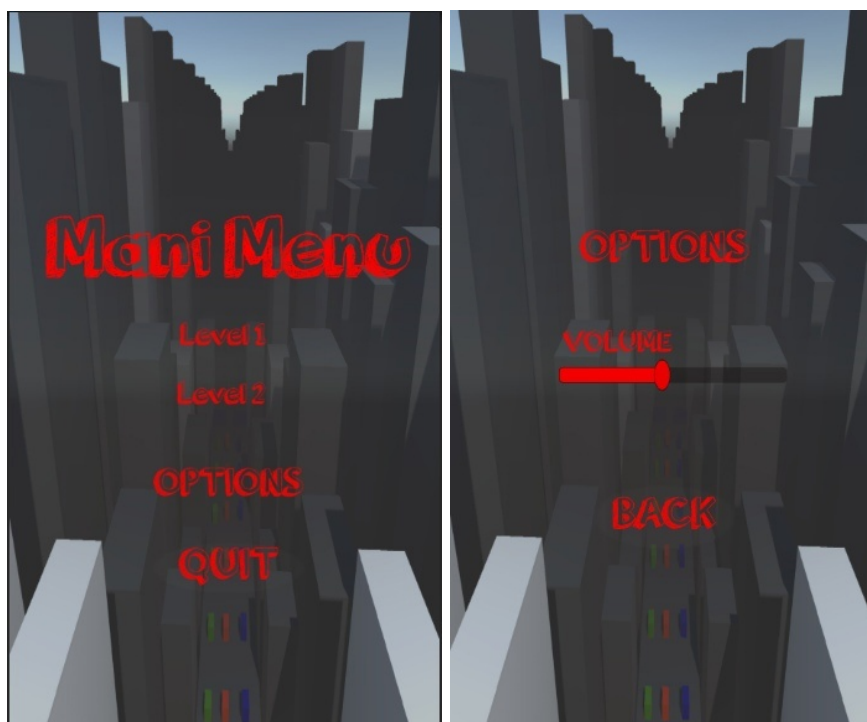


Рисунок 2 – Главное меню

На рисунке 3 представлен интерфейс персонажа.

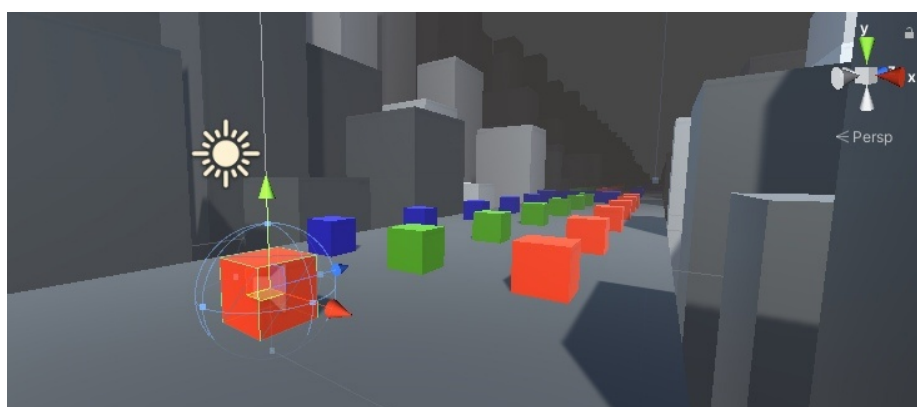


Рисунок 3 – Интерфейс персонажа

На рисунке 4 представлено начало игры.

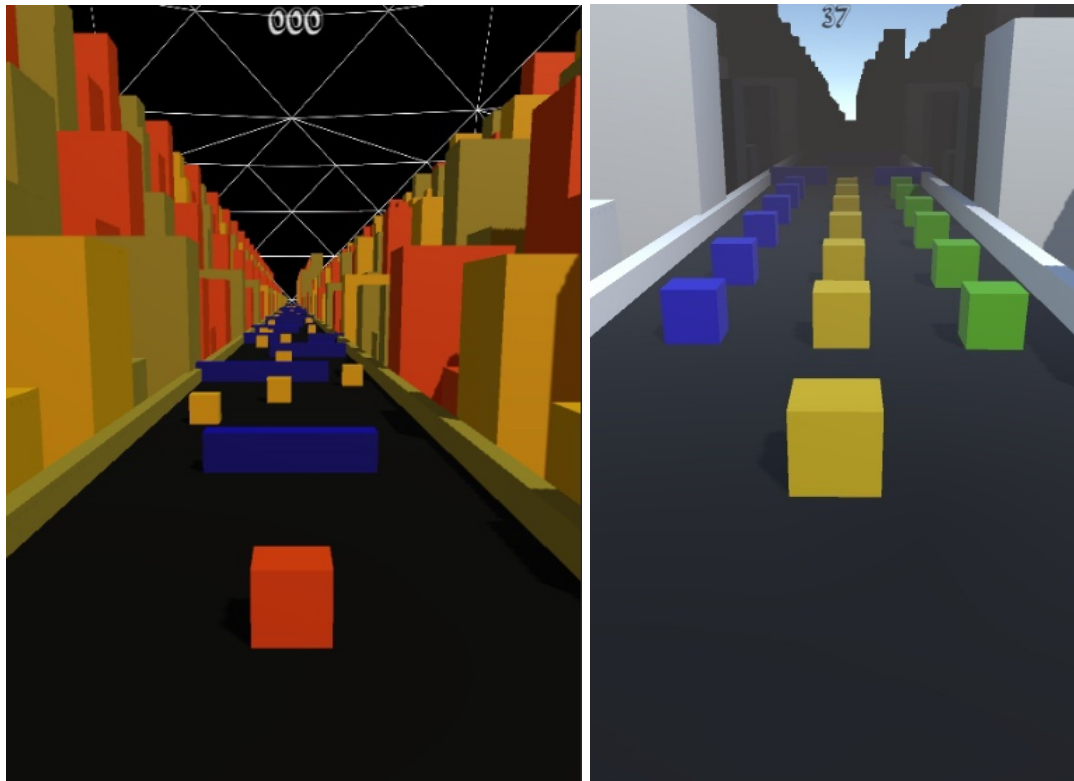


Рисунок 4 – Начало игры

На рисунке 5 представлено завершение игры в случае проигрыша.



Рисунок 5 – Конец игры



На рисунке 6 представлено завершение игры в случае выигрыша.



Рисунок 6 – Конец игры

Некоторые листинги к игре отражены в приложении.

Листинг 1 отвечает за выигрыш в игре. Когда «Кубик» дойдет до определенной точки (конец игры), то появится экран. На этом экране будет «you win thx for playing» и выбор действия начать заново или выйти из игры.

Листинг 2 отвечает за движение игрока «Кубик». «Кубик» автоматически двигается прямо. Управление осуществляется движением пальца по экрану направо и налево.

Листинг 3 отвечает за счетчик очков в игре. Когда «Кубик» собирает себе подобные кубы, то на верхней части экрана увеличиваются очки. 1 куб = 1 очко.

Листинг 4 отвечает за кнопки в меню «level 1», «level 2». При нажатии «level 1» активируется 1 уровень. При нажатии «level 2» активируется 2 уровень.

Листинг 5 отвечает за проигрыш в игре. Когда «Кубик» столкнется с непохожими на него цветами кубами, то появится окно. На нем будет «game over» и выбор действия начать заново или выйти из игры.

## 5. ТЕСТИРОВАНИЕ

Для тестирования приложения использовался метод функционального тестирования.

Функциональное тестирование – это тестирование программного обеспечения в целях проверки реализуемости функциональных требований, то есть способности программного обеспечения в определенных условиях решать задачи, нужные пользователям [16]. В таблице 2 отражено тестирование компьютерной игры «Goway».

Таблица 2 – Тест-план

№	Назначение	Действия	Ожидаемый результат	Полученный результат	Итог
1	Проверка на корректное реагирование персонажа на поступающие команды о передвижении	Используя клавиши движения, пробежать часть уровня	Персонаж будет двигаться в положенном направлении, корректное проигрывание анимации	Персонаж передвигался в нужном направлении, корректное отображение анимации	Пройден
2	Проверка на корректную реализацию события проигрыша игрока	Позволить персонажу столкнуться с преградой	Появляется меню выбора действия	Появилось меню выбора действия	Пройден
3	Проверка на корректную работу главного меню	Запустить уровень, выйти из игры	Есть возможность запустить уровень, либо выйти из игры	В зависимости от выбора запускается уровень, либо закрывается игра	Пройден
4	Проверка на корректный сбор собираемых объектов	Собрать объекты	При сборе объектов, собранный объект уничтожается	При сборе объекта, он исчезает	Пройден

Данное тестирование показывает, что на данном этапе реализации проекта, игра соответствует функциональных требований, выделенных на этапе постановки задачи.

## **6. ЗАКЛЮЧЕНИЕ**

В рамках выполнения выпускной квалификационной работы была разработана компьютерная игра «Goway» на платформе Unity.

В ходе выполнения данной работы была сделана большая работа, начиная с изучения литературы по программированию и заканчивая тестированием и отладкой готового программного продукта.

Были решены следующие задачи:

- проведен обзор аналогичных решений и осуществлена постановка задачи;
- проведен обзор средств реализации;
- проведен анализ требований и спроектирована компьютерная игра;
- реализована демонстрационная версия игры;
- проведено тестирование реализованной версии игры.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Despicable me: minion rush – [https://stopgame.ru/show/50426/despicable\\_me\\_minion\\_rush\\_review](https://stopgame.ru/show/50426/despicable_me_minion_rush_review). Дата обращения 17.04.2021.
2. Jere Mile. Unity 3D and PlayMaker essentials: game development from concept to publishing. Taylor & Francis, CRC Press, 2016. – 507 с.
3. Patrick Felicia. Getting Started with 3D Animation in Unity. 2018. – 159 с. – <https://yadi.sk/d/GGKCL4mAn7I-QQ>. Дата обращения 20.01.2021.
4. Rayman Origins – [https://ru.wikipedia.org/wiki/Rayman\\_Origins](https://ru.wikipedia.org/wiki/Rayman_Origins). Дата обращения 17.04.2021.
5. Ori and the Blind Forest – [https://ru.wikipedia.org/wiki/Ori\\_and\\_the\\_Blind\\_Forest](https://ru.wikipedia.org/wiki/Ori_and_the_Blind_Forest). Дата обращения 17.04.2021.
6. Robert Nystrom. Game Programming Patterns – <http://gameprogrammingpatterns.com/contents.html>. Дата обращения 17.04.2021.
7. Ryan Henson Creighton. Unity 3D Game Development by Example Beginner's Guide. – Packt Publishing, 2010. – 384 с.
8. Албахарь, Д. С# 6.0. Справочник. Полное описание языка / Д. Албахари; 6-е изд.: Пер. с англ. – М. : ООО "И.Д. Вильямс", 2016. – 1040 с.
9. Аркада (жанр) – [https://ru.wikipedia.org/wiki/%D0%90%D1%80%D0%BA%D0%B0%D0%B4%D0%B0\\_\(%D0%B6%D0%B0%D0%BD%D1%80\)](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D0%BA%D0%B0%D0%B4%D0%B0_(%D0%B6%D0%B0%D0%BD%D1%80)). Дата обращения 17.04.2021.
10. Гибсон, Б. Д. Unity и С#. Геймдев от идеи до реализации / Б. Д. Гибсон; 2-е изд. – СПб.: Питер, 2019. – 928 с.
11. ГОСТ Р ИСО/МЭК 9126-93. Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению. – <https://docs.cntd.ru/document/1200009076>. Дата обращения 17.04.2021.

12. Дашко, Ю. В. Основы разработки компьютерных игр / Ю. В. Дашко, А. А. Заика. – М.: Форум, 2009. – 350 с.
13. Дикинсон, К. Оптимизация игр в Unity 5 / К. Дикинсон; пер. с англ. Рагимова Р. Н. – М.: ДМК Пресс, 2017. – 306 с.
14. Какой игровой движок выбрать: Unity, Udk или Cryengine? – <https://3dpara.ru/what-game-engine-to-choose/>. Дата обращения 10.03.2021.
15. Линовес, Д. Виртуальная реальность в Unity / Д. Линовес. – М.: ДМК Пресс, 2016. – 316 с.
16. Майерс, Г. Искусство тестирования программ / Г. Майерс, Т. Баджет. – М: Вильямс, 2012. – 272 с.
17. Мэннинг, Дж. Unity для разработчика. Мобильные мультиплатформенные игры / Дж. Мэннинг, П. Батфилд-Эддисон. – СПб.: Питер, 2018. – 304 с.
18. Нативная или кроссплатформенная разработка – что лучше? – <http://wnfx.ru/nativnaya-ilikrossplatformennaya-razrabotka-chto-luchshe/>. Дата обращения 10.03.2021.
19. О мобильных раннерах и их классификации – <https://stopgame.ru/blogs/topic/95546>. Дата обращения 10.03.2021.
20. Паласиос, Х. Unity 5.x. Программирование искусственного интеллекта в играх / Х. Паласиос; пер. с англ. Р. Н. Рагимова. – М.: ДМК Пресс, 2017. – 272 с.
21. Роллингз, Э. Проектирование и архитектура игр / Э. Роллингз, Д. Моррис; пер. с англ. – М.: Издательский дом «Вильямс», 2006. – С. 39-43.
22. Скиена, С. Алгоритмы. Руководство по разработке / С. Скиена; пер. с англ. – СПб.: БХВ-Петербург, 2011. – 720 с.
23. Стилмен, Э. Изучаем C# / Э. Стилмен, Дж. Грин. – СПб.: Питер, 2014. – 816 с.

24. Торн, А. Основы анимации в Unity / А. Торн; пер. с англ. Р. Рагимова. – М.: ДМК Пресс, 2016. – 176 с.
25. Хокинг, Д. Unity в действии. Мультиплатформенная разработка на C# / Д. Хокинг. – М.: Питер, 2018. – 608 с.
26. Язев, Ю. Обзор самых популярных движков для разработки игр / Ю. Язев. – <https://haker.ru/2014/09/05/game-developmentengines-review/>.  
Дата обращения 10.03.2021.

## ПРИЛОЖЕНИЕ

### Листинг 1 – Выигрыш.

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

public class GameController3 : MonoBehaviour
{
    public GameObject gameWinPanel;
    private void Start()
    {
        gameWinPanel.SetActive(false);
    }
    public void GameWin()
    {
        gameWinPanel.SetActive(true);
    }
    public void Restart()
    {
        SceneManager.LoadScene("Game");
    }
    public void QuitGame()
    {
        Application.Quit();
    }
}
```

### Листинг 2 – Движение игрока.

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

public class PlayerScript : MonoBehaviour
{
    public Rigidbody rigidbody;
    public float force = 100f;
    public float speed = 10f;
    public float maxX = 50f;
    public float minX = -50f;
    // Start is called before the first frame update
    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
        Vector3 playerPos = transform.position;
        playerPos.x = Mathf.Clamp(playerPos.x, minX, maxX);
        transform.position = playerPos;

        TouchMovement();
    }
}
```



```

    void TouchMovement()
    {
        Vector2 touch = Camera.main.ScreenToWorldPoint(Input.mousePosition + new
Vector3(0, 0, 12f));
        if (Input.touchCount > 0)
        {
            transform.position = new Vector3(touch.x,
transform.position.y,transform.position.z);
        }
    }
    private void FixedUpdate()
    {
        rigidbody.AddForce(0, 0, force* Time.deltaTime);
    }
}

```

### Листинг 3 – Счетчик.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine.UI;
using UnityEngine;

public class Score : MonoBehaviour
{
    public Text ScoreText;

    int myScore = 0;
    void Update()
    {
        ScoreText.text = myScore.ToString();
    }
    public void AddScore(int score)
    {
        myScore = myScore + score;
    }
}

```

### Листинг 4 – Кнопки меню.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class MainMenuControlrScript : MonoBehaviour
{
    public Button level01Button;
    public Button level02Button;
    int levelpassed;

    // Start is called before the first frame update
    void Start()
    {
        /*levelpassed = PlayerPrefs.GetInt("LevelPassed");*/
        /*level02Button.interactable = false;*/

        switch (levelpassed)
        {
            case 1:
                level02Button.interactable = true;
                break;
        }
    }
}

```

```

        /* case2:
           level02Button.interactable = true;
           break;*/
    }

}

public void levelToLoad(int level)
{
    SceneManager.LoadScene(level);
}

public void resetPlayerPrefs()
{
    level02Button.interactable = false;
    PlayerPrefs.DeleteAll();
}

}

```

### Листинг 5 – Проигрыш.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine.SceneManagement;
using UnityEngine;

public class GameController : MonoBehaviour
{
    public GameObject gameOverPanel;
    private void Start()
    {
        gameOverPanel.SetActive(false);
    }
    public void GameOver()
    {
        gameOverPanel.SetActive(true);
    }
    public void Restart()
    {
        SceneManager.LoadScene("Game");
    }
    public void QuitGame()
    {
        Application.Quit();
    }
}

```