

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Российский государственный профессионально-педагогический университет»

## **КОМПЬЮТЕРНАЯ ИГРА ПЛАТФОРМЕР**

Выпускная квалификационная работа  
по направлению подготовки 09.03.02 Информационные системы  
и технологии  
профилю подготовки «Информационные технологии в медиаиндустрии»

Идентификационный номер ВКР: 155

Екатеринбург 2017

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Российский государственный профессионально-педагогический университет»  
Институт инженерно-педагогического образования  
Кафедра информационных систем и технологий

К ЗАЩИТЕ ДОПУСКАЮ

Заведующая кафедрой ИС

\_\_\_\_\_ Н. С. Толстова

« \_\_\_\_ » \_\_\_\_\_ 2017 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
КОМПЬЮТЕРНАЯ ИГРА ПЛАТФОРМЕР

Исполнитель:	И. Л. Байбатыров
обучающийся группы № ИТм-401	
Руководитель:	Н. С. Толстова
к.п.н., зав. каф. ИС	
Нормоконтролер:	Б. А. Редькина

Екатеринбург 2017

## АННОТАЦИЯ

Выпускная квалификационная работа состоит прототипа компьютерной игры и пояснительной записки на 56 страницах, содержит 14 рисунков, 8 таблиц, 25 источников литературы.

Ключевые слова: КОМПЬЮТЕРНАЯ ИГРА, ВИДЕОИГРА, ПЛАТФОРМЕР

**Байбатыров И. Л., Компьютерная игра платформер:** выпускная квалификационная работа / И. Л. Байбатыров; Рос. гос. проф.-пед. ун-т, Ин-т инж.-пед. образования, Каф. информ. систем и технологий. — Екатеринбург, 2017. — 56 с.

В работе рассмотрены проблемы разработки компьютерных игр в России.

Целью работы является создание прототипа однопользовательского двумерного платформера для персональных компьютеров. Для достижения цели были проанализированы современные тенденции в разработке компьютерных игр, исследованы средства разработки компьютерных игр и проекты конкурентов. На основе этих элементов создан прототип однопользовательского двумерного платформера для персональных компьютеров, содержащий в себе 1 уровень с реализованной основной игровой механикой.

Благодаря быстрому и динамичному развитию игровой индустрии в России и в мире, компьютерные игры в скором времени начнут занимать одно из основных направлений в развитии современного общества и массовой культуры.

# СОДЕРЖАНИЕ

Введение.....	4
1 Исследование понятия «Компьютерная игра» и обзор технологий разработки компьютерных игр .....	6
1.1 Исследование понятия и классификация компьютерных игр .....	6
1.2 Общий алгоритм реализации проекта.....	10
1.3 Средства создания компьютерных игр .....	13
1.3.1 Анализ игровых движков .....	13
1.3.2 Анализ графических редакторов .....	16
1.4 Анализ существующих разработок .....	18
2 Разработка проекта.....	24
2.1 Описание целевой аудитории .....	24
2.2 Постановка задачи проекта .....	25
2.2.1 Актуальность, цель и назначение .....	25
2.2.2 Функционал проекта .....	26
2.2.3 Сценарий .....	26
2.2.4 Характеристика оборудования разработки .....	33
2.3 Реализация проекта .....	34
2.3.1 Этап разработки графического оформления .....	34
2.3.2 Этап разработки игры .....	36
2.3.3 Тестирование .....	45
2.3.4 Технические характеристики .....	47
2.3.5 Калькуляция проекта .....	48
Заключение .....	50
Список использованных источников .....	52
Приложение .....	<b>Ошибка! Закладка не определена.</b>

## ВВЕДЕНИЕ

Индустрия компьютерных игр появилась относительно недавно, около 30 лет назад, но уже смогла развиться в огромную отрасль с колоссальными доходами в несколько миллиардов долларов год. Понять подобный внезапный рост популярности виртуальных развлечений очень просто: все это благодаря широкому распространению компьютерных технологий, в том числе появлению сети Интернет. Благодаря этому, в отличие от других видов развлечений, компьютерные игры более доступны для конечного пользователя. Для того что бы просто поиграть игроку нужно иметь только компьютер или игровую приставку и копию самой игры, а с широким распространением сети Интернет ради получения копии игры не нужно выходить из дома. Более того, потребителю не требуется иметь особых знаний для того что бы выбрать подходящую для него игру, в то время как для большинства других видов развлечений необходимо разбираться как минимум в необходимой экипировке.

Так же стоит принять во внимание, что компьютерные игры в последнее время перестали позиционироваться как программы только для отдыха и развлечений. К примеру, сегодня, благодаря использованию игровых технологий, создаются специальные комплексы по симуляции, которые служат для обучения специалистов в различных областях: от лесорубов до пилотов реактивных самолетов.

В России же, к сожалению, все несколько иначе. Игровая индустрия у нас, так же как и в большинстве других стран постсоветского пространства, развита крайне слабо. Связано это с тем, что культура компьютерных развлечений пришла к нам слишком поздно и практически не развивалась. Из-за этого, даже при достаточно высоком спросе, мы имеем слишком малое количество компаний-разработчиков, которые могут составить конкуренцию зарубежным компаниям.

Поэтому развитие технологий в данном направлении можно считать одним из наиболее перспективных, особенно в нашей стране

Объектом исследования является разработка компьютерных игр.

Предмет исследования: технологии разработки компьютерной игры жанра 2D-платформер.

Цель выпускной квалификационной работы - разработать прототип компьютерной игры жанра 2D-платформер средствами среды Unity.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- 1) изучить особенности и состояние компьютерной индустрии России;
- 2) выбрать жанр, вид и платформу для компьютерной игры;
- 3) разработать сценарий, концепцию основных элементов;
- 4) выбрать и изучить средство реализации;
- 5) подготовить необходимые для игры анимации;
- 6) реализация прототипа игры.

# **1 ИССЛЕДОВАНИЕ ПОНЯТИЯ «КОМПЬЮТЕРНАЯ ИГРА» И ОБЗОР ТЕХНОЛОГИЙ РАЗРАБОТКИ КОМПЬЮТЕРНЫХ ИГР**

## **1.1 Исследование понятия и классификация компьютерных игр**

Термином компьютерная игра (или иногда используют неоднозначный термин видеоигра) обозначается компьютерная программа, которая служит для организации игрового процесса (геймплея), связи с партнёром по игре, или сама выступающая в качестве партнёра. Компьютерные игры часто создаются на основе сторонних источников, таких как фильмы или книги, но в последнее время стали появляться обратные случаи, когда по известной игровой серии начинают выпускать дополнительные материалы, расширяющие вселенную игры.

Более того специально разработанные игры могут выступать в качестве обучающего материала или позволяют использовать игроков в научно-исследовательских целях. Такие игры редко выпускаются в широкие массы. По некоторым играм проводятся любительские и профессиональные соревнования, которые называются киберспортом.

Компьютерные игры оказали столь существенное влияние на общество, что в последнее время в информационных технологиях появилась устойчивая тенденция к геймификации для неигрового прикладного программного обеспечения. Таким образом, в некоторых европейских школах начали использовать известную игру Minecraft для обучения, а для нужд армий стали создавать специальные симуляторы для тренировок солдат. Компьютерные игры так же с 2011 года официально признаны правительством США и американским Национальным фондом отдельным видом искусства, наряду с театром и кино, а в России киберспорт официально был причислен к видам спорта [3].

Из всего этого следует, что компьютерные игры плотно влились в нашу нынешнюю жизнь. Более того, сфера их использования за последние 10 лет сильно выросла, теперь игры используют не только для развлечения и отдыха, но и для обучения людей и проведения научных исследований.

Компьютерные игры классифицируют по нескольким основным признакам:

- жанр;
- количество игроков;
- визуальное представление;
- платформа.

Жанр игры определяется целью и основной механикой игры. Основные жанры представлены в таблице 1.

Таблица 1 — Жанровая классификация [2]

Жанр	Особенности	Основные поджанры
Аркада (Arcade)	игры с нарочно примитивным игровым процессом	<ul style="list-style-type: none"> <li>• Платформер</li> <li>• Файтинг</li> <li>• Скролеры</li> </ul>
Приключенческая игра (Adventure)	главной частью игры является история.	<ul style="list-style-type: none"> <li>• Квест</li> <li>• Визуальная новелла</li> </ul>
Ролевая игра (RPG)	главная особенность игры, игрок практически не ограничен в выборе игровых предметов, напарников и диалогов.	<ul style="list-style-type: none"> <li>• Тактическая</li> <li>• Экшен RPG</li> <li>• Японская (Западная)</li> <li>• ZPG</li> </ul>
Экшен (Action)	игра, характеризующая частым и активным нажатием кнопок управления.	<ul style="list-style-type: none"> <li>• Beat'em Up</li> <li>• Шутер</li> <li>• Слешер</li> </ul>



Окончание таблицы 1

Стратегическая игра (Strategy)	Основа игры – необходимость игроку делать нетривиальный выбор.	<ul style="list-style-type: none"> <li>• Пошаговая/Реального времени</li> <li>• Глобальная</li> <li>• Варгейм</li> <li>• Экономические</li> </ul>
Компьютерный симулятор (Simulator)	игрок делает множество упражнений и оттачивает свою технику.	<ul style="list-style-type: none"> <li>• Технические</li> <li>• Симуляторы жизни</li> </ul>
Головоломка (Puzzle)	требует аналитического мышления.	
Обучающая игра (Educational)	игрок обучается во время выполнения каких-либо действий в игре.	
Игрушки (Toys)	программы, взаимодействуя с которыми, игрок получает удовольствие.	

Представленные здесь жанры нельзя назвать полными, так как в последнее время стали появляться игры собственных жанров, которые можно отнести как к одному из представленных здесь жанров, так и к самостоятельному отдельно от них.

По количеству игроков игры разделяются на два вида:

- однопользовательские;
- многопользовательские;

По визуальному представлению компьютерные игры можно разделить на следующие виды:

- текстовые – минимальное графическое представление, общение с игроком проходит с помощью текста;
- 2D – все элементы отрисованы в виде двумерной графики (спрайтов);

- 3D – все элементы отрисованы в виде трехмерной графики (3D – модели).

В данный список не включены такие категории как 2,5D и псевдо-3D, так как они являются лишь разновидностями представленных категорий и технологии их создания практически не отличаются от технологий создания основных представленных критериев.

По типу платформы:

- персональные компьютеры;
- игровые приставки/консоли;
- мобильные телефоны;

К сожалению, представленная здесь классификация не является полной и может быть дополнена. Так, к примеру, к жанрам можно добавить такой специфичный вид игр как музыкальные игры, где основное внимание уделяется музыке и/или звукам, а многопользовательские игры в свою очередь делятся на несколько подкатегорий. Но стоит понимать, что данной классификации достаточно для определения большинства из существующих игр, так как на сегодняшний день не существует однозначной и полной классификации для компьютерных игр. Это происходит из-за того, что многие игры нельзя отнести к каким-либо критериям. Например, игра может соответствовать сразу нескольким жанрам, выйти сразу на нескольких платформах или иметь как однопользовательский режим игры, так и многопользовательский.

Все это обусловлено тем, что игровая индустрия, в отличие от других сфер развлечений, появилась сравнительно недавно. Но как уже было сказано, данной классификации достаточно, для того чтобы определить большинство компьютерных игр существующих на данный момент.

Основываясь на данной классификации компьютерных игр было принято решение разработать **двумерный однопользовательский платформер для персональных компьютеров**. Данное решение было принято потому, что платформер обладает простой и понятной механикой, что означает лег-

кость в разработке для начинающих игровых разработчиков. Так же двумерные игры проще в производстве, в отличие от трехмерных.

## **1.2 Общий алгоритм реализации проекта**

Общий алгоритм разработки компьютерной игры [2] мало чем отличается от алгоритма разработки любого другого программного продукта и включает в себя 3 больших этапа:

- 1) проектирование;
- 2) разработка;
- 3) издание и поддержка.

На этапе проектирования определяются цель игры и средства её разработки.

При определении цели выделяются идея, жанр и сеттинг игры. Идея – это то, что будет побуждать игрока играть в создаваемую игру, и она очень тесно связана с жанром. Так, к примеру, основная идея RPG – позволить игроку прожить свою роль так, как ему захочется, а основная идея шутера – позволить игроку поучаствовать в реальных или вымышленных боевых действиях. Таким образом, определив основные идеи игры, жанр будет подобран практически сразу.

Определившись с жанром и идеей игры, следующим шагом будет выбор сеттинга. Сеттинг – это среда, в которой будет происходить основное действие игры. Он определяет место, время и условия действия. Выбор сеттинга может сильно облегчить разработку сценария для игры, поэтому его лучше выбирать заранее и основываясь на вкусах целевой аудитории.

К средствам разработки в первую очередь относят программный код и игровой движок. От их грамотного выбора зависит как скорость самой разработки, так и работоспособность самого продукта в дальнейшем. Программный код в первую очередь зависит от платформы, для которой будет создаваться компьютерная игра. Например, если игра создаётся для браузеров, то

логично будет использование языка Java или Flash, но, если игра создается для персонального компьютера, оптимальным выбором будет, к примеру, язык программирования C#.

Игровой движок отвечает за низкоуровневое описание физики объектов, правил рендеринга графики и др. При выборе игрового движка первым делом смотрят на его доступность и уже сделанный выбор языка программирования. К примеру, игровой движок Unity позволяет разрабатывать игры на языке C# и он является бесплатным, если средний доход компании не превышает \$100000 в год.

После выбора цели игры и средств разработки, начинается второй этап реализации проекта – разработка.

Разработка самый крупный и долгий этап реализации проекта, он включает в себя большое количество шагов, без которых невозможно создать работоспособный продукт.

Первым делом нужно определиться с сюжетом и игровой механикой. Игровая механика основывается на цели игры, она определяет все объекты и правила, по которым игрок будет взаимодействовать с ними.

Обычно параллельно с разработкой игровой механики идет написание сюжета игры. Сюжет играет не маловажную роль, он определяет то, насколько будет игроку интересно играть в вашу игру. Сюжет представляют в двух вариантах: литературный и режиссёрский сценарии. Литературный сценарий описывает основные события и персонажей игры, которые участвуют в игре. Режиссерский же представляет собой подробное описание уровней игры, событий, которые на этих уровнях происходят

Так же на данном этапе начинается ранняя проработка графической составляющей и дизайна игры. На основе сюжета и заранее обговоренного дизайна, создаются ранние концепт-арты, на основе которых впоследствии будет проработан основной вид игры и персонажей.

После разработки сюжета и игровой механики начинается самая важная часть – разработка самой игры.

На основе сюжета и концепт-артов начинается создание персонажей и объектов игры, параллельно с этим идет разработка игровых уровней. При разработке игровых уровней изначально создается его упрощённый план, на котором схематично изображен сам уровень, а так же изображены предметы, с которыми будет впоследствии взаимодействовать игрок.

Следом после этого создается первая версия уровня. Обычно, она представляет собой просто голую локацию, с минимумом необходимых для прохождения предметов. Эта версия уровня служит для того, что бы протестировать уровень на проходимость. После теста, уровень начинают постепенно заполнять остальными объектами.

Вскоре после создания первых уровней идет сборка первого прототипа игры, который называют альфа-версией игры. Она необходима для того, что бы разработчик мог провести тестирование (альфа-тестирование) основной механики игры, и проверить насколько она соответствует заявленным требованиям. Часто в альфа-версии игры, у объектов даже нет текстур или они вообще представлены в виде абстрактных объектов.

Если альфа-версия игры успешно проходит тестирование, наступает следующий этап разработки – проработка механики и объектов игры.

На данном этапе идет доработка уровней и механики игры, и начинают добавлять первые сюжетные события в игру, такие как видеоролики, сюжетные диалоги и кат-сцены. Так же исправляются первые ошибки и неисправности в коде игры, которые были обнаружены при тестировании альфа-версии игры.

После этого наступает этап создания второго прототипа игры, или, как принято говорить, бета-версии. Бета-версия служит для того, что протестировать игру на неисправности, фактически бета-версия представляет собой практически готовую игру. В ней могут отсутствовать какие-нибудь незначительные элементы, которые не влияют на геймплей игры. При тестировании бета-версии игры проверяется абсолютно все. Часто, особенно, если игра является мультиплеерной, на тестирование приглашаются обычные игроки –

это позволяет сильно ускорить время проведения тестирования, а так же снять часть нагрузки с команды разработки.

Если игра проходит бета-тестирование, она отправляется на окончательную доработку и исправление критических ошибок, после чего идет сборка финальной версии игры и следом наступает релиз игры.

После релиза игры последующая её поддержка. Поддержка заключается в выпуске патчей (файлов исправлений ошибок в готовом продукте). Так же для того, что бы продлить жизненный цикл игры, для неё выпускают дополнительный контент в виде DLC (Downloadable content), которые добавляют различные предметы и/или возможности для игрока.

Таким образом, стало понятно, что этапы разработки компьютерных игр мало отличаются, от этапов разработки любого другого программного продукта.

### **1.3 Средства создания компьютерных игр**

#### **1.3.1 Анализ игровых движков**

Основными средствами разработки игр являются игровые движки, которые отвечают за низкоуровневое описание физики игры, правил рендеринга графики и пр. и графические редакторы для отрисовки графики.

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.

В настоящее время существует огромное количество различных игровых движков. Основные их различия заключаются в поддерживаемых языках программирования, функциональности и, что не менее важно, в стоимости лицензии. При выборе среды разработки основное внимание было уделено

как раз на эти параметры. Для анализа выбрали следующие игровые движки Unreal Engine, Unity, и Game Maker: Studio.

Game Maker: Studio [12] – игровой движок разработанный и поддерживаемый компанией YoYo Games, написан на Delphi (следующая версия на C#). На данный момент является одним из наиболее популярных игровых движков для разработки двумерных игр для всех современных популярных платформ. Для разработки использует собственный упрощенный язык программирования GML (Game Maker Language).

Данный игровой движок имеет несколько версий распространения:

- Trial;
- Desktop;
- Web;
- UWP;
- Mobile;

Каждая из этих версий (кроме Trial) отличается платформой, для которой будет создана игра. Trial – версия является бесплатной и предоставляется для тех, кто хочет сначала опробовать игровой движок, имеет ряд ограничений на количество объектов используемых в игре, а также позволяют компилировать проект только для теста на операционной системе Windows.

Unreal Engine [24] – игровой движок разрабатываемый и поддерживаемый компанией Epic Games.

Первая игра, созданная на этом движке — Unreal — появилась в 1998 году. С тех пор различные версии движка были использованы в более чем сотне игр и других проектов.

Написанный на языке C++, движок позволяет создавать игры для большинства операционных систем и платформ: Microsoft Windows, Linux, Mac OS и Mac OS X; консолей Xbox, Xbox 360, Xbox One, PlayStation 2, PlayStation 3, PlayStation 4, PSP, PS Vita, Wii, Dreamcast, GameCube и др., а также на устройствах управляемых системой iOS и Android.

Для упрощения портирования движков использует модульную систему зависимых компонентов; поддерживает различные системы рендеринга (Direct3D, OpenGL, Pixomatic), воспроизведения звука (EAX, OpenAL, DirectSound3D), средства голосового воспроизведения текста, распознавание речи, модули для работы с сетью и поддержки различных устройств ввода.

2 марта 2015 года Unreal Engine 4 стал бесплатным. Однако, разработчики игр, как и прежде, должны передавать 5 % от прибыли игры компании Epic Games, но при условии, что доходы игры составляют более \$3000 за квартал.

Unity [21] – средство для разработки двух- и трехмерных игр, являющееся одним из наиболее популярных на сегодняшний день систем разработки. Позволяет создавать приложения, работающие под большинством современных операционными системами (Windows, OS X, Windows Phone, Android, Apple iOS, Linux), а также на игровых приставках Wii, PlayStation 3, PlayStation 4, Xbox 360, Xbox One и MotionParallax3D дисплеях (устройства для воспроизведения виртуальных голограмм), например, Nettlebox. Есть возможность создавать приложения для запуска в браузерах с помощью специального подключаемого модуля Unity (Unity Web Player), а также с помощью реализации технологии WebGL. Последние версии Unity позволяют создавать приложения для очков виртуальной реальности.

Редактор Unity имеет простой Drag&Drop интерфейс, который легко настраивать, состоящий из различных окон, благодаря чему можно производить отладку игры прямо в редакторе. Движок поддерживает два сценарных языка: C#, JavaScript (модификация). Расчёты физики производит физический движок PhysX от NVIDIA.

Проект в Unity делится на сцены (уровни) — отдельные файлы, содержащие свои игровые миры со своим набором объектов, сценариев, и настроек. Сцены могут содержать в себе как, собственно, объекты (модели), так и пустые игровые объекты — объекты, которые не имеют модели («пустыш-



ки»). Объекты, в свою очередь содержат наборы компонентов, с которыми и взаимодействуют скрипты.

Unity распространяется бесплатно, но помимо бесплатной, существуют четыре сборки - стандартная Unity, Unity iOS Pro, Android Pro и командная лицензия. Они отличаются стоимостью и функциональностью.

Бесплатная версия имеет некоторые ограничения, однако возможность распространять игры имеется, при условии, что ежегодный доход с игры не превышает 100 000\$. С выходом Unity 5 многие ограничения Free версии были убраны, но ежегодный доход с игры всё также не должен превышать 100 000\$.

Все это делает Unity одним из наиболее приоритетных движков для начинающих игровых разработчиков.

### **1.3.2 Анализ графических редакторов**

Для обработки графической составляющей игры были отобраны три графических редактора: Adobe Photoshop, CorelDRAW Graphics Suite и PaintTool SAI.

Adobe Photoshop [7] – мощный графический редактор для фотографий и рисований, разработанный и поддерживаемый Adobe Systems. Работает с растровыми изображениями, однако имеет несколько векторных инструментов. Лидер рынка в области коммерческих средств редактирования растровых изображений и наиболее известный продукт фирмы Adobe.

Изначально задумывался как мощный графический редактор для растровых изображений в полиграфии, но быстро стал популярен в области Web-дизайна. Позволяет работать в большинстве популярных цветовых моделях.

Все это делает Adobe Photoshop одним из лучших графических редакторов.

CorelDRAW Graphics Suite [8] – векторный графический редактор, разработанный и распространяемый компанией Corel. Является лучшим вектор-

ным редактором. В пакет CorelDRAW Graphics Suite также входит редактор растровой графики Corel PHOTO-PAINT и другие программы — например, для захвата изображений с экрана — Corel CAPTURE. Программа векторизации растровой графики Corel TRACE до 12 версии входила в пакет как самостоятельная программа.

CorelDRAW позволяет импортировать проекты в большинство современных графических форматах, а рисование в векторах позволяет безболезненно изменять размеры изображения, что несомненно является полезным свойством при разработке графического оформления для двумерных игр.

PaintTool SAI [17] - программа, предназначенная для цифрового рисования в среде Microsoft Windows, разработанная японской компанией SYSTEMAX.

Данная программа в первую очередь направлена на тех, кто для рисования использует графические планшеты. Обладает простым и понятным интерфейсом, значительно «легче» относительно Adobe Photoshop, благодаря чему обладает высокой скоростью работы и запуска даже на относительно слабых компьютерах. Поддерживает все популярные графические форматы.

Для разработки проекта был выбран игровой движок Unity, так как он проще в освоении и при этом предоставляет огромный функционал в создании различного рода игр, а бесплатная версия не обладает какими-либо серьёзными ограничениями. Так же он широко распространён среди разработчиков игр в России, что ещё сильнее упрощает его освоение.

Для создания графической составляющей игры было решено использовать Paint Tool SAI, так как к разработке графики был привлечён сторонний человек, который был лучше знаком с данным графическим редактором. Так же данный редактор обладает лучшей совместимостью с графическими планшетами, что упрощает процесс создания графики для игры.

## 1.4 Анализ существующих разработок

В качестве проекта был выбран однопользовательский 2D – платформер, поэтому основных конкурентов стоит искать в той же жанровой категории. Но стоит помнить, что данный жанр потерял свою популярность, из-за чего современные игры в этого жанра часто не являются чистыми платформерами. Самыми яркими представителями в последнее время были:

- Ori and the Blind Forest;
- Mark of the Ninja;
- Dust: An Elysian Tail;
- Limbo;
- DuckTales: Remastered;
- Wings of Vi.

**Ori and the Blind Forest** [16] – игра в жанре платформер разработанная студией Moon Studios и изданная Microsoft Studios для платформ Windows и Xbox One. Игра полностью выполнена в двумерном стиле и написана на игровом движке Unity.

Сюжет рассказывает о лесном духе Ори, который является небольшим листиком огромного духа-дерева, защищающее волшебный лес. Однажды штормовой ветер отрывает Ори и уносит его далеко в лес. Упав на землю он превращается в небольшое белое существо и находит Нару – духа, живущего в лесу и похожего на большого черного медведя, - которое становится для Ори матерью.

Спустя время, темный дух Куро, крадет сердце Древа, и лес начинает стремительно увядать. Вскоре после этого Нару так же погибает и Ори остается один. Он пускается в странствие по погибающему лесу, но вскоре тоже умирает. Но Древо, рядом с которым Ори заканчивает свою жизнь, использует оставшиеся силы, воскрешает его и просит очистить волшебный лес от ТЬМЫ.

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. В игре так же присутствует простая система прокачки в виде небольшого дерева умений. Для покупки новых умений нужно собирать небольшие сгустки энергии, разбросанные по миру игры. Карта мира загружается сразу и полностью, и игрок сам выбирает куда пойти. Единственным ограничением в данном случае является необходимость определенных способностей для прохождения определенных участков.

Игра отличается невероятно 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, но во всем остальном, данную игру можно считать эталоном жанра.

**Mark of the Ninja** [15] – двумерный стелс-экшен, разработанный канадской студией Klei Entertainment для Xbox 360 и ПК, выпущенная в сервисах Xbox Live Arcade и Steam. Данная игра не является чистым платформером, но имеет большое количество атрибутов данного жанра.

Сюжет повествует о безымянном ниндзя, который носит звание Чемпиона клана Хисоми, так как получил особую татуировку, которая усиливает его способности, но взамен постепенно сводит его с ума. Он пытается освободить выживших членов его клана, которые попали в плен после атаки отряда Гессен.

Геймплей игры основан на скрытном прохождении уровня, избегании контактов с противниками и обезвреживании ловушек. Главный герой умеет прятаться в вентиляции, канализационных люках и за предметами интерьера.

Основное оружие — меч-ниндзято, которым можно выполнить бесшумное убийство ничего не подозревающей жертвы и бамбуковые дротики,

предназначенные для отвлечения и уничтожения источников света. Несмотря на наличие смертельного оружия, игра в равной степени ориентирована как на устранение всей охраны уровня, так и на бескровное прохождение. В процессе прохождения покупается и открывается дополнительное снаряжение атакующего и отвлекающего действия.

Практически за каждое действие начисляются очки, на которые можно открыть новые приёмы, приобрести новое или улучшить старое снаряжение. Дополнительно на каждом уровне выдается 3 разных бонусных задания. После выполнения заданий одного типа становится доступным костюм, специализированный для определённого стиля прохождения.

Также на каждом уровне, в скрытых местах можно найти артефакты, которые дадут вам несколько сотен очков. Ещё на уровне присутствуют различные головоломки, которые называются Комнатами испытаний. По прохождении головоломки вы получаете 1 из 3 скрытых на уровне свитков Хи-соми, который также дает дополнительные очки. В основном головоломка заключается в том, чтобы пройти комнату, избежав ловушек.

Данная игра является отличным стелс-экшеном среди двумерных игр подобного жанра, которых не так много производится в последнее время. Обладает стильным графическим оформлением, которое задает атмосферу всей истории. Минималистичный и информативный интерфейс, но несколько перегруженное управление, которое может отбить желание играть дальше.

**Dust: An Elysian Tail** [11] – экшен-РПГ с элементами платформера, разработанная независимыми разработчиком Дином Додриллом и опубликованная Microsoft Studios. Данная игра так же не является чистым платформером.

Dust происходит в вымышленном мире Фаланы, населенном антропоморфными животными. Игрок управляет титулярным главным героем, Да-стом, который пытается вспомнить свое прошлое. В последствии ему предстоит узнать тайну своего появления, а так же спасти свой мир от тирании Генерала Гаяя. Да-ст обладает легендарным мечом, Лезвием Аарах. Фиджет,

Хранитель меча, выступает в качестве компаньона Даста и может использовать магические атаки. Пока игрок и его компаньон путешествуют по миру, они могут приобрести бонусы, которые постоянно будут изменять геймплей, такие как возможность двойного прыжка или вход на ранее недоступные территории. Включая в себя элементы ролевых игр, Даст может получать очки опыта, побеждая врагов, и в свою очередь повышать уровень. Они могут быть использованы, чтобы поднять различные атрибуты, такие как здоровье, сила, защита или магия.

Геймплей стандартный для платформеров – игрок передвигается по уровням, преодолевая препятствия, но стоит понимать, что в отличие от предыдущих 2х игр, здесь наиболее важным аспектом является развитие персонажа и убийство врагов. Из-за этого понятно, что данная игра скорее подвид платформера, который носит название метроидвания.

**Limbo** [13] – игра жанра пазл-платформер, созданная компанией Playdead.

Как такого сюжет здесь нет, все, что известно игроку, это цель игры – пройти до загадочной девочки. Геймплей представляет собой типичный платформер с элементами пазла – игроку нужно добраться от начала уровня к его концу избегая ловушек и врагов, попутно решая простые пространственные головоломки. Игра выполнена в стилистике минимализм, и в ней преобладают преимущественно темные тона, из-за этого ловушки не всегда заметно и игрок легко погибнуть.

Основной проблемой игры, является её собственная стилистика, которая хоть и создает нужную атмосферу, в некоторых случаях может очень сильно помешать при игре. Так же геймплей для многих людей может показаться скучным и неинтересным, что так же сказывается на отношении к самой игре. Таким образом, понятно, что игра изначально не рассчитана на широкую аудиторию.

**DuckTales: Remastered** [10] – платформер, разработанный компанией Carcom, является ремастером (переизданием) старого платформера DuckTales.

Сюжет игры достаточно прост и, по сути, является просто катализатором для начала игры. Игрок берет под свое управление селезня Скруджа МакДака, которым движет жажда завладеть сокровищами и стать самым богатым селезнем в мире.

Геймплей типичен для платформеров. Игрок должен преодолеть все препятствия и врагов, что бы добраться до конца уровня. В конце игрока встречает босс, которого необходимо победить. По уровням раскиданы различные предметы, одни из которых увеличивают общий счет, а другие способны восстановить здоровье. Стилистика игры, по сравнению с оригиналом переработана, теперь в игре только персонажи и враги выполнены в виде двумерной графики, в то время как окружение и предметы созданы в виде трехмерных моделей.

Игра выполнена хорошо, отличное графическое исполнение и интуитивное управление, понятный интерфейс. Минусом можно было бы назвать только устаревшую механику, все же за столько времени было придумано множество других интересных геймплейных особенностей.

**Wings of Vi** [25] – платформер, разработанный и выпущенный студией Grynsoft.

Сюжет незамысловат и типичен для подобных игр. Как-то Ви, с другом Руби направилась в сырую темницу, где находилось в заточении Зло. По каким то причинам темница не имела охраны, а по приходу «непосед», они обнаружили (благодаря подруге садистке), что цепи разорваны а дверь где находилось Зло выбиты. Злобная сущность вырвалась дабы сеять раздор и смуту, по всему белому свету. Героям Wings of Vi предстоит исправить выявленную ими проблему и отправиться в путь по подземельям, городам и небеса, по следу великого Зла.

Геймплейно игра не отличается от первых платформеров игроку так же необходимо преодолевать множество различных препятствий и врагов, что бы пройти уровень от его начала до конца. В конце уровня игрока встретит босс, которого нужно убить, что бы пройти дальше. Отличительной особенностью данной игры является её высокая сложность. Большинство врагов наносят большой урон персонажу, а уровни спроектированы таким образом, что бы игроку изначально было сложно пройти игру. Такого вида усложнение, однако, не делает игру хуже, а скорее даже наоборот. Графически игра выполнена в стиле пиксель-арт (стиль, при котором изображения рисуются попиксельно, для достижения эффекта «ретро» игр).

Сложность в данной игре выступает как достоинством, так и недостатком. Из-за того, что у игры высокий порог вхождения, не каждый сможет её пройти, а значит данный продукт является скорее нишевым, чем предназначенным для широких масс.

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.



## 2 РАЗРАБОТКА ПРОЕКТА

### 2.1 Описание целевой аудитории

Для разработки компьютерной игры, как и для любого другого продукта, одну из самых важных ролей играет определение целевой аудитории продукта. Когда определена целевая аудитория, разработчику намного легче определить многие аспекты разрабатываемого продукта.

Для данного разрабатываемого проекта была выделена достаточно большая целевая аудитория, а именно молодые люди от 16 до 35 лет. Выбор такой обширной категории людей обуславливается сразу несколькими причинами.

Прежде всего, выбор такой категории людей обусловлен средним возрастом активно играющих в компьютерные игры людей, который в России стабильно держится в диапазоне от 30 до 33 лет [4]. Именно из этих соображений выбрана такая верхняя грань целевой аудитории игры.

Нижняя грань была выбрана из тех соображений, что хотя средний возраст геймера и приближается к отметке 33 года, реальный игрок разрабатываемой игры, вероятно, будет значительно моложе. Это обусловлено тем, что люди начиная с 30-35 летнего возраста, начинают постепенно переходить от жанров с быстрым геймплеем к жанрам с более медленным игровым процессом.

Также не стоит забывать про то, что данная категория людей с большей вероятностью обладает достаточными средствами для покупки разрабатываемой игры, когда она будет выпущена. Обусловлено это тем, что проект относится к категории малобюджетных компьютерных игр, а значит более доступен для широких масс потенциальных покупателей.

Таким образом, можно считать, что выделенная целевая аудитория является основной для данного разрабатываемого проекта.

## **2.2 Постановка задачи проекта**

### **2.2.1 Актуальность, цель и назначение**

На сегодняшний день игровая индустрия развивается невероятно быстро, количество активных игроков растет с каждым годом, как и прибыль, которая все за 6 выросла почти вдвое и на конец 2016 года составляла больше 90 млрд долларов. В России же доходы выросли ещё больше с 73 млн долларов за 2010 год до почти 3 млрд за 2016 год. Все это указывает на то, что разработка видеоигр является одним из перспективнейших направлений в сфере разработки программного софта.

Можно точно сказать, что разрабатываемая игра будет продаваться среди выделенной целевой аудитории. Этому способствуют выбранный жанр и стилизация проекта. Жанр платформер достаточно популярен среди игроков, так как представляет собой простой и понятный, а так же интересный игровой процесс. Обладая очень низким порогом вхождения, такие игры доступны очень широкому кругу людей, в том числе тем, кто играми как средством проведения собственно досуга мало интересуется.

Так же стоит помнить, что современное общество перенасыщено крупными проектами, требующие от игрока глубокого понимания механики игрового процесса. Для их прохождения зачастую требуется довольно большое количество времени. На их фоне игра с простым и понятным геймплеем будет смотреться более предпочтительно среди людей, которым сложно уделять достаточно много времени.

На основе выше перечисленного легко выделяются цель и назначение проекта.

Цель проекта заключается в предоставлении средства для приятного времяпрепровождения.

Назначение проекта заключается в развлечении потенциального потребителя.

### **2.2.2 Функционал проекта**

Проект представляет собой компьютерную игру жанра двумерный платформер, основная цель которой заключается в развлечении и предоставлении средства для отдыха и приятного времяпрепровождения.

Прежде всего, проект должен отвечать следующим требованиям:

1. Игра должна обладать простым и понятным геймплеем.
2. Основная механика игры заключается в возможности игрока спрятаться за определенными предметами дабы избежать встречи с противником.
3. Не должно быть слишком усложнённое управление главным героем.
4. Враги обитают в определенных местах, но при обнаружении игрока они должны следовать за ним, до тех пор пока игрок не сбежит, или не будет пойман.
5. На уровне должны присутствовать собираемые объекты. Эти объекты должны увеличивать счет игрока.
6. При выигрыше или проигрыше игрока, должно выводиться небольшое меню, предлагающее либо повторить уровень, либо выйти в главное меню.
7. Меню не должно быть перегружено, но в тоже самое время должно быть информативным.

Данные требования к функциональной части являются основными, должны быть выполнены в первую очередь.

### **2.2.3 Сценарий**

Перед началом разработки проекта, необходимо написать сценарий игры, представляющий собой краткий синопсис сюжета с описанием основных персонажей, а так же описание уровней игры. На его основе в дальнейшем будут разрабатываться первые концепты персонажей и уровней.

Ясон, спустя почти 20 лет, решает вернуться в город, в котором жил до трагического пожара, в котором погибли его родители, когда ему едва перевалило за 5 лет. Видения давней трагедии все это время преследовали его в кошмарах, и он думает, что если посетит пепелище старого дома, сможет от них избавиться. Рядом с руинами бывшего особняка Ясон находит странного вида амулет, когда он берет его в руки, его посещает образ огромного черного пса. Придя в себя, Ясон видит странного старца. Старец сообщает, что давний пожар не был случаен, а амулет – старый артефакт, что позволяет заглядывать в прошлое.

Используя амулет, что бы проследить путь таинственного пса, Ясон узнаёт, что виновниками пожара являются трое старых друзей его отца: Гарри, Петр и Алекс, которые решают убить его, так как он отказывается продавать старые археологические находки, что тот находит не далеко от своего дома. Так же Ясон понимает, что черный пес – это душа его отца, которая переродилась из-за сильного чувства гнева и желания отомстить.

1 арка – Ясон, молодой человек 25-30 лет, среднего роста и довольно приятной внешности. Он ищет способ избавиться от частых ночных кошмаров, что появились у него вследствие психологической травмы из-за смерти родителей в детстве. По совету психолога, вернулся в свой родной город, что бы посетить пепелище старого дома. По сути, является не столько главным героем, сколько наблюдателем и рассказчиком истории о Черном Псе.

2 арка – Черный Пес, большой пес черного окраса, не является ни живым, ни мертвым, о чем сообщают голубые линии на его теле. В реальности, представляет собой перерожденную душу отца Ясона, что из-за сильных отрицательных чувств не смогла «уйти». Совершат основные сюжетные выборы.

## **1 Уровень**

Открывающий ролик: мы видим, как Ясон посещает пепелище своего старого дома, находит амулет и его посещает видение о пробуждении Черного Пса.

Локация – затухающий пожар старого дома и лес не далеко от него.

Цель – выйти из леса и найти Гарри.

Черный Пес обнаруживает себя рядом с огромным пожаром. Он ничего не помнит, кроме коротких отрывков прошлого и страшной жажды мести. Черный Пес бросается в лес. В лесу он встречает, странных черных Сущностей, которые пытаются его поймать. У Черного Пса получается сбежать от них, и выйти из леса, где находит Гарри. Черного Пса охватывает ярость, и он сразу убивает его [Гарри].

## **2 Уровень**

Локация – архив документов.

Цель – Исследовать документы и найти улики.

Придя в себя, Ясон видит перед собой загадочного старца, который общается ему, что амулет позволяет заглядывать в прошлое мест и предметов, а так же рассказывает, что пожар не был случайностью. Старец уходит, а Ясон принимает решение разобраться в загадке пожара. Для этого он отправляется в архив, что бы узнать как можно больше о том, что произошло.

В архиве Ясон находит документы, рассказывающие о его отце. Ясон возвращается в гостиницу и ложится спать. Во сне ему снова приходит видение о Черном Псе.

## **3 Уровень**

Локация – окраина Города.

Цель – добраться до дома Петра.

Черный Пес приближается к городу, но слышит зов о помощи, где-то не далеко. Подойдя к месту, откуда доносятся крики, черный Пес видит как стая бродящих собак, явно чем-то больных, пытается загрызть маленькую девочку, которая и зовет на помощь. Черному Псу нужно решить как поступить в данной ситуации (таблица 2), а затем направиться дальше в город.

В городе Черный Пес приходится часто прятаться, так как люди напуганы недавними нападениями бродячих собак, и попытаются убить любого подозрительного пса.

Таблица 2 — Сюжетные выборы на уровне 3

<b>ЧП решает пойти на помощь</b>	<b>ЧП решает не идти на помощь</b>
<p>ЧП спасает маленькую девочку от разъяренной стаи собак. Девочка оказывается Еленой - дочкой Петра, которому пытается отомстить ЧП. Благодаря её запаху он сразу узнает, где находится его следующая цель.</p>	<p>ЧП предстоит самостоятельно обнаружить свою следующую цель. Для этого ему предстоит обследовать большую часть домов поблизости.</p>

Черный Пес находит дом Петра.

#### **4 Уровень**

Локация – небольшой двор и сам дом Петра.

Цель – найти Петра.

Развитие сюжета на уровне 4 находятся в таблице 3.

Таблица 3 — Сюжетные выборы на уровне 4

<b>Если Черный Пес спасает Елену.</b>	<b>Если Черный Пес не приходит на помощь Елене.</b>
<p>Черному Псу нужно найти способ привлечь внимание Елены. Когда ему это удастся, девочка проводит его через пустой дом к себе в комнату.</p> <p>Через некоторое время, возвращаются родители, Петр заходит в комнату и видит Черного Пса.</p> <p>Черный Пес бросается на свою цель и пытается убить. Елена приходит на помощь отцу.</p>	<p>Черному Псу нужно проникнуть в дом. Самый верный способ – через подвал. В доме, ему предстоит прятаться от большого количества полицейских, которые приехали на звонок родителей, что не обнаружили свою дочь дома.</p> <p>Через некоторое время полицейские уходят, и у Черного Пса появляется шанс убить Петра, который находится в гостиной.</p>

<b>Черный Пес продолжает попытки убить Петра</b>	<b>Черный Пес отступает</b>	После убийства Петра, Черный Пес покидает дом.
Черный Пес убивает Петра, но в борьбе тяжело ранит Елену. Черный Пес уходит.	Елена успокаивает Петра, сообщив, что Черный Пес спас её. Черный Пес уходит.	

### **5 уровень**

Локация – Город, музей.

Цель – проникнуть в музей и найти директора.

Ясон просыпается, он помнит свой сон, в котором наблюдал за Черным Псом, поэтому решает, что стоит посетить дом Петра. Но сначала ему нужно посетить музей, так как в документах из архива говорится, что артефакты, найденные отцом Ясона, находятся там.

В музей Ясона не пускает охранник, так как сейчас там идет реставрация. Ясону нужно найти способ проникнуть внутрь.

В музее Ясон встречается с директором. Тот рассказывает Ясону о его отце, так же высказывает свои догадки на счет пожара и что считает его виновниками – Гарри, Петра и Алекса. Ясон отправляется к дому Петра

### **6 Уровень**

Локация – дом Петра.

Цель – попасть в дома Петра и найти Елену.

Ясону нужно по образу из памяти найти дом Петра. Когда он его находит, Ясон так же нужно завоевать доверие вдовы Петра, что бы войти внутрь. Вскоре ему это удастся, и вдова разрешает Ясону войти.

В доме Ясон встречает Елену (таблица 4).

Таблица 4 — Сюжетные выборы на уровне 6

<b>Елена была спасена, но убит Петр</b>	<b>Елена и Петр были спасены</b>	<b>ЧП не спасал Елену, Петр убит</b>
Елена прикована к инвалидной коляске, она практически ничего не помнит о своем детстве из-за психологической травмы. Единственное, что она может рассказать – это история об огромном Черном Псе, который убил её отца	Елена полностью здорова, но малоразговорчива из-за недавней кончины её отца. Она рассказывает Ясону, о красивом Черном Псе, что спас её от стаи собак, а так же о последних словах Петра, который кажется, догадался об истинной личности Черного Пса.	Елена прикована к инвалидной коляске, а психологическая травма, полученная в детстве, сделала её психически не уравновешенной. Она ничего не может сообщить Ясону. Вдова Петра рассказывает о странной собаке, что она видела очень давно.

Ясон возвращается в гостиницу.

### **7 Уровень**

Локация – ночной Город.

Цель – добраться до дома Алекса.

Выбравшись из дома Петра, Черный Пес отправляется к дому Алекса, по странному стечению обстоятельств – это то немногое, что он помнит из своего прошлого. Через Город теперь пробираться куда опасней, так как теперь вместо людей по улицам бродят черные Сущности. Черный Пес добирается до дома Алекса и находит способ, как проникнуть в дом через подвал.

### **8 Уровень**

Локация – дом Алекса.

Цель – найти Алекса.



Проникнув в дом Алекса, Черный Пес обнаруживает, что тот полон черных Сущностей. Черному Псу нужно включить свет в доме, а для этого нужно найти способ завести генератор в подвале.

Включение света прогоняет Сущностей, но будит хозяина дома. Алекс встречается с Черным Псом, пугается его и убегает.

### **9 Уровень**

Локация – Лес перед рассветом.

Цель – догнать Алекса и победить его.

Алекс убегает далеко вперед и Черному Псу нужно догнать его, но черные Сущности, что находятся в лесу, мешают ему в этом. Правда теперь они намного слабее, так как уже начинает вставать солнце, что даёт Черному Псу преимущество, и он догоняет Алекса. Тот долго сопротивляется, но в конечном итоге проигрывает (таблица 5).

Таблица 5 — Сюжетные выборы на уровне 9

Если Петр жив	Если Петр мертв
Черный Пес пытается убить Алекса, но в последний момент останавливается, спасение Петра Еленой сильно на него повлияло, он больше не хочет мстить. Внезапно к Черному Псу возвращается память о прошлой жизни. Черный Пес несколько секунд смотрит на Алекса и уходит.	Черный Пес убивает Алекса, его жажда мести утолена, но как только он отходит от трупа бывшего друга, он вспоминает свою прошлую жизнь. Тело Черного Пса начинает резко изменяться, и он превращается в черную Сущность.

### **10 Уровень**

Локация – дом Алекса.

Цель – пробраться в дом и исследовать его.

Ясон просыпается, теперь он знает, что произошло 20 лет назад. Он решает посетить дом Алекса, что бы удостовериться в своих догадках.

Дом Алекса кажется пустым и покинутым, Ясон решает просто поникнуть туда, но ему нужно найти способ сделать это. В конечном итоге он взламывает замок подвальных дверей (таблица 6).

Таблица 6 — Сюжетные концовки

<b>Если Алекс жив</b>		<b>Если Алекс мертв</b>
<p>Ясон поднимается наверх в гостиную, но обнаруживает, что дом только казался пустым. В гостиной незваного гостя поджидал хозяин дома с ружьем в руках. Алекс понимает, кто перед ним, опускает оружие, и, упав на колени, сознается в убийстве друга ради наживы.</p>		<p>В гостиной Ясон находит дневник хозяина дома. В нем тот рассказывает свои планы по убийству отца Ясона с целью забрать древние реликвии. Ясон забирает дневник с собой, а дом сжигает.</p>
<b>Елена спасена</b>	<b>Елена не спасена</b>	
<p>Ясон выслушивает Алекса. В гневе он бросается на него, но вовремя останавливается, так как понимает, что его убийство не исправит прошлое.</p>	<p>Ясон выслушивает Алекса, но в порыве гнева за пережитые страдания бросается на него и убивает.</p>	

#### 2.2.4 Характеристика оборудования разработки

Так как для разработки был выбран игровой движок Unity, а для написания и редактирования программного кода используется Microsoft Visual Studio 2017, то понадобится оборудование, обладающее техническими характеристиками не ниже минимально необходимых для корректно работы данных средств разработки. Поэтому разработка велась на ноутбуке со следующими параметрами:

- Процессор: Core i3-3120M 2,5ГГц;
- Видеокарта: NVidia GeForce 710M 2Гб VRAM;
- ОЗУ: 4Гб;
- ОС: Windows 10.

## **2.3 Реализация проекта**

### **2.3.1 Этап разработки графического оформления**

Стоит принять во внимание, что основные этапы реализации проекта, а именно разработки графического оформления и разработка игры, проходили параллельно друг другу. Этому способствовал тот факт, что для написания программного кода необязательно наличие готовых анимаций и прочего, а для проверки работоспособности самого кода достаточно стандартных примитивов Unity. Так же такой способ реализации проекта значительно увеличивает скорость работы.

Прежде всего, стоит выделить основные понятия, такие как тайл (Tile) и спрайт (Sprite).

Тайл (Tile) – небольших размеров повторяющийся фрагмент, который служит для постройки изображений больших размеров (Тайловая графика). Часто используется для создания уровней для двумерных игр.

Спрайт (Sprite) – графический объект, представляющий собой растровое изображение. Используется в компьютерной графике как основная единица для анимаций двумерных объектов.

Для реализации графического оформления был приглашен сторонний человек, который нарисовал все необходимые элементы для игры. А именно анимации движений для главного персонажа и противников на уровне, а так же фоновое изображение, отображаемое позади уровня. Фон и атласы спрайтов представлены на рисунках 1-3.



Рисунок 1 — Фоновое изображение

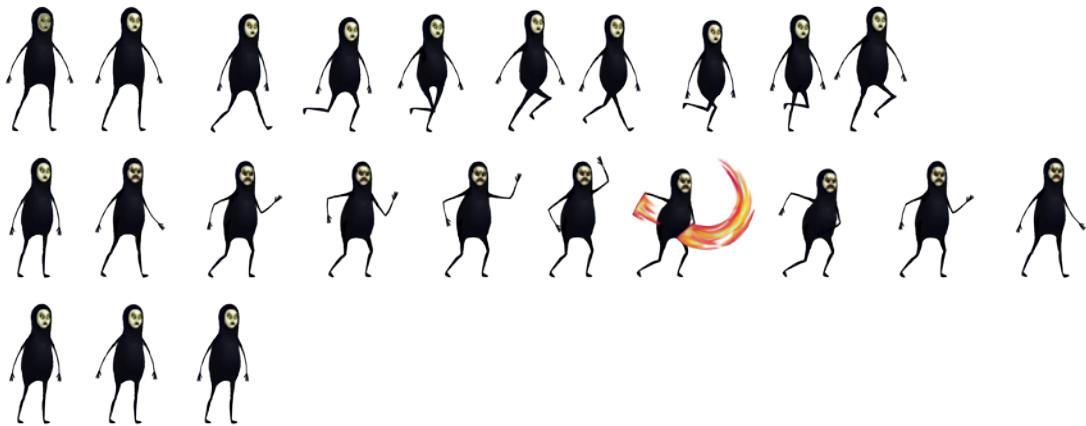


Рисунок 2 — Атлас спрайтов для врага

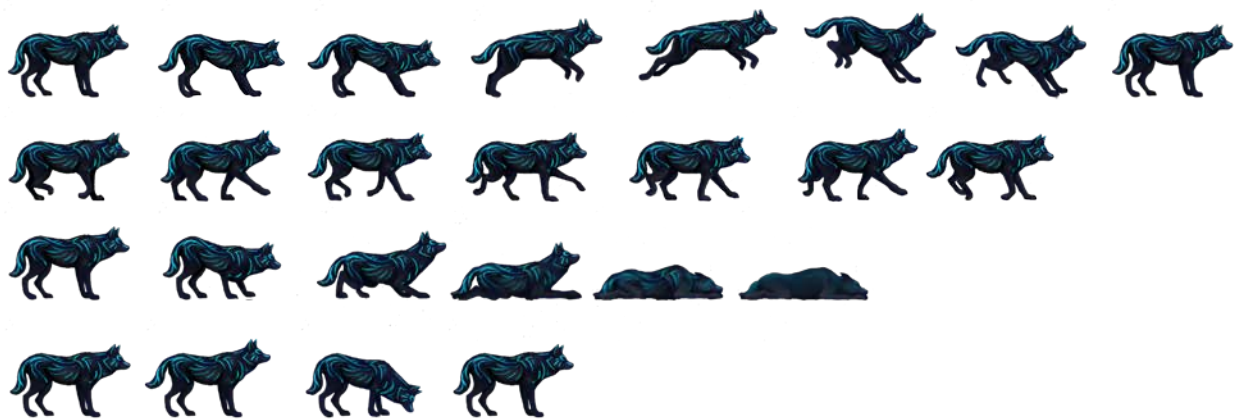


Рисунок 3 — Атлас спрайтов для Черного Пса

Для реализации графики был использован графический редактор PaintTool SAI, так как привлеченный к работе над проектом человек наиболее хорошо разбирался в данном графическом редакторе.

Для тайловой графики был использован набор тайлов, находящийся в свободном доступе.

### 2.3.2 Этап разработки игры

Для реализации компьютерной игры были выбраны игровой движок Unity и Microsoft Visual Studio 2017 для описания программного кода.

При запуске Unity мы увидим окно проекта (рисунок 4), посередине находятся окно Сцены, окно Анимации, а так же окно Игры. Первое служит для создания общей композиции уровня и добавления новых объектов, второе представляет возможность для создания и редактирования анимаций объектов, третье окно представляет вид из камеры показывая, каким образом будет выглядеть игра на данный момент.

Слева находится окно Иерархии, здесь показываются все объекты, участвующие в данной сцене. С самого начала здесь находится только камера.

Справа окно Инспектора, здесь отображаются все текущие свойства выбранного объекта.

Снизу находятся три окна: окно Проекта, окно Аниматора и Консоль. В окне Проекта отображаются все объекты, добавленные в текущую игру, в том числе скрипты, анимации и прочее. В окне Аниматора создаются связи и правила перехода между различными анимациями. Консоль служит для отображения ошибок и исключений возникающих во время работы игры.

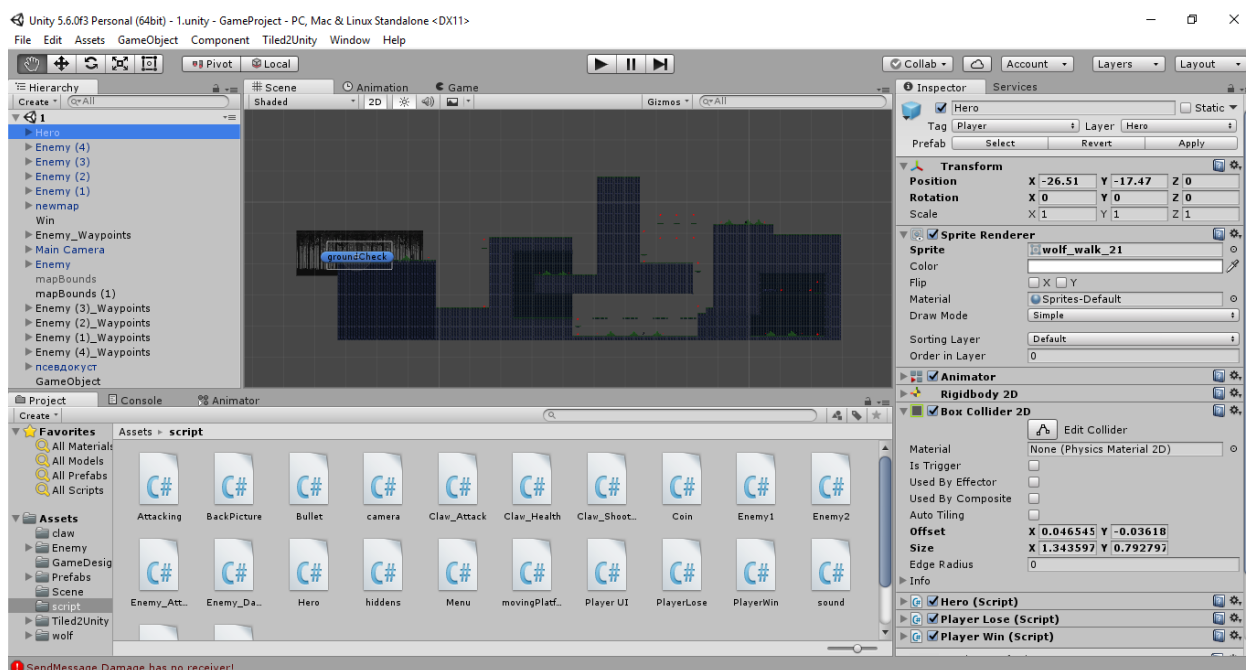


Рисунок 4 — Окно проекта Unity

Прежде всего, нам необходимо добавить в сцену платформу, по которой будет перемещаться наш герой и его враги. Без неё игрок просто будет бесконечно падать в пространстве. Сначала нам необходимо добавить сам спрайт платформы, который был создан ранее, для этого достаточно просто перетащить его из папки в окно проекта, после чего он должен появиться среди всех объектов уже добавленных в игру. После этого нам необходимо добавить нашу платформу в текущую сцену. Это можно сделать 2мя способами: перетащить из окна Проекта в окно Сцены, либо сначала в окне Иерархии создать отдельный пустой объект, к которому впоследствии и применить спрайт платформы (рисунок 5).

После этого нам необходимо добавить к объекту коллайдер. Он необходим для того, что бы движок понимал границы объекта и игрок мог с ним взаимодействовать. В Unity существует два вида коллайдеров: для трехмерных и двумерных объектов. Так как создаваемая игра двумерная, то и коллайдер нужно добавлять двумерный. Что бы добавить коллайдер к объекту, необходимо выделить его в окне Иерархии, а затем добавить компонент коллайдера в окне Инспектора. Для настройки размеров коллайдера нужно воспользоваться кнопкой Edit Collider.

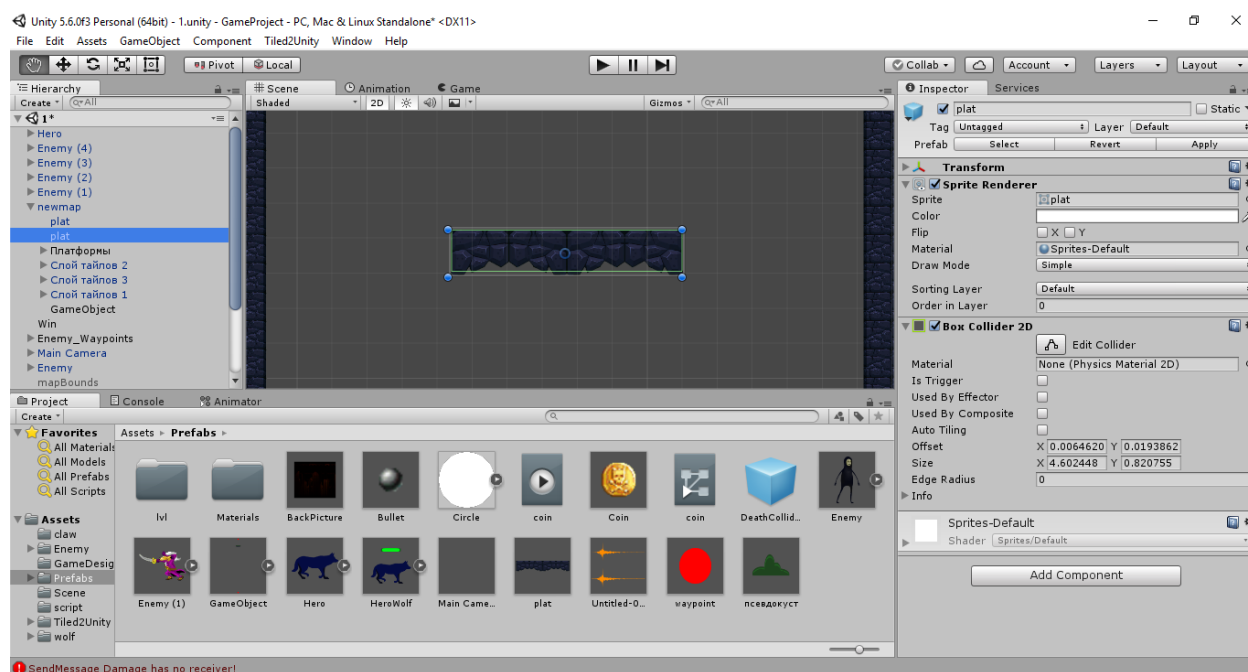


Рисунок 5 — Платформа в окне сцены и её свойства

Unity позволяет сохранять объекты, со всеми добавленными свойствами и скриптами для того, что бы потом их можно было повторно использовать, просто переместив на сцену, не создавая заново новый объект. Такой объект называется Префаб и для его создания достаточно просто переместить созданный объект из окна Иерархии в окно Проекта. Для префабов имеет смысл создать отдельную папку, дабы они не терялись среди прочих элементов игры.

Теперь создаем героя игры. Для этого так же добавляем двумерный объект, и добавляем к нему спрайт героя. Для реализации героя нам понадо-

бится коллайдер, а так же компонент Rigidbody 2D, через него настраивается физическая модель объектов.

Так как главный герой должен двигаться и совершать различные действия нам необходимо создать скрипт на языке C#. Это можно сделать несколькими способами: создать в окне Проекта, или создать сразу на необходимом объекте в окне Инспектора. Когда скрипт будет создан, открываем его и после этого должен запускаться Microsoft Visual Studio. В коде уже будут подключены основные библиотеки, а так же создан стандартный метод Update. Записываем код для движения персонажем (рисунок 6). После чего добавляем его к нашему герою.

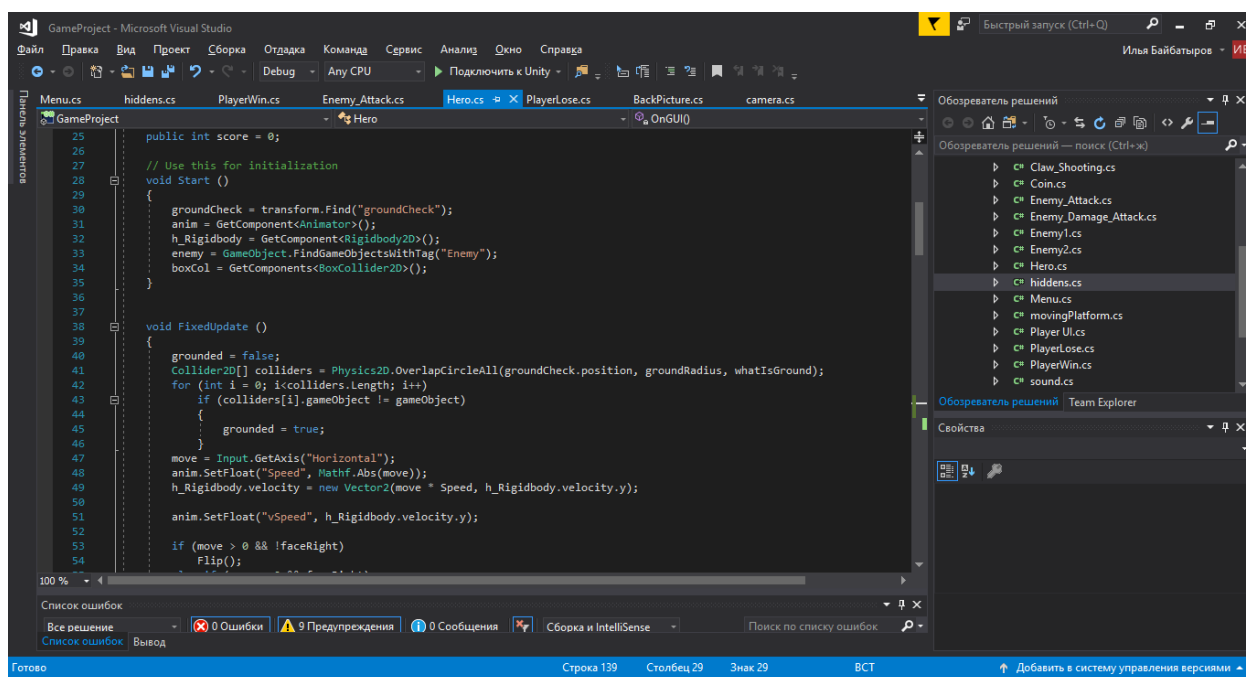


Рисунок 6 — Код для передвижения героя

Для реализации прыжков на необходимо создать у героя пустой дочерний объект. Этот пустой объект назовем groundCheck, при помощи него мы будем определять, находится персонаж на земле или нет. Это необходимо, что бы игрок не мог бесконечно подниматься вертикально вверх при постоянном нажатии клавиши прыжка. Теперь, если запустить проект, мы сможем побегать по платформе из стороны в сторону.

Точно так же создаем второй скрипт, который мы добавляем к камере и в нем прописываем условия, по которым камера будет следовать за героем по



уровню. Если мы этого не сделаем, то не сможем следить за действиями персонажа.

Теперь нам необходимо создать врагов. Враги создаются схожим способом, за тем исключением, что игрок ими управлять не может. Так же отличительной особенностью врагов является то, что в зависимости от ситуации, они сами выбирают необходимое действие доступное им по правилам игры. Так же передвижения врагов ограничены границами их мест обитания.

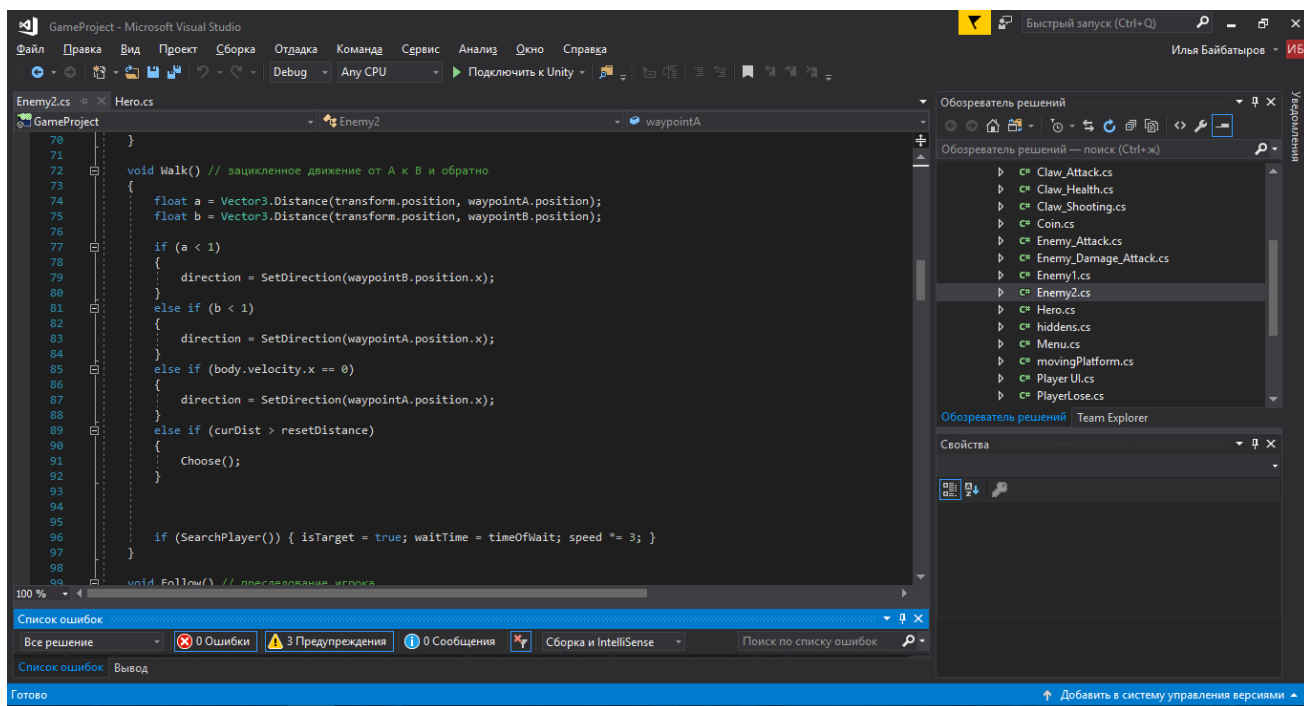


Рисунок 7 — Фрагмент кода скрипта поведения врага

После создания скриптов для главного героя и врагов нам необходимо создать для персонажа три анимации:

- анимация бега;
- анимация атаки;
- анимация прыжка.

Для этого добавляем в проект атласы скриптов всех этих анимаций. Если теперь выбрать какой – либо из файлов, станет заметно, что Unity воспринимает его как единый файл. Что бы выделить каждый кадр анимации необходимо в инспекторе в графе Sprite mod поменять значение на multiple, что даст понять игровому движку, что файл содержит несколько различных объ-

ектов. После этого переходим в Sprite editor (рисунок 8) и нарезаем кадры для анимации при помощи инструмента Slice. После чего настраиваем фреймы под размер кадров, а их центры таким образом, что бы свести «плавание кадров» к минимуму.

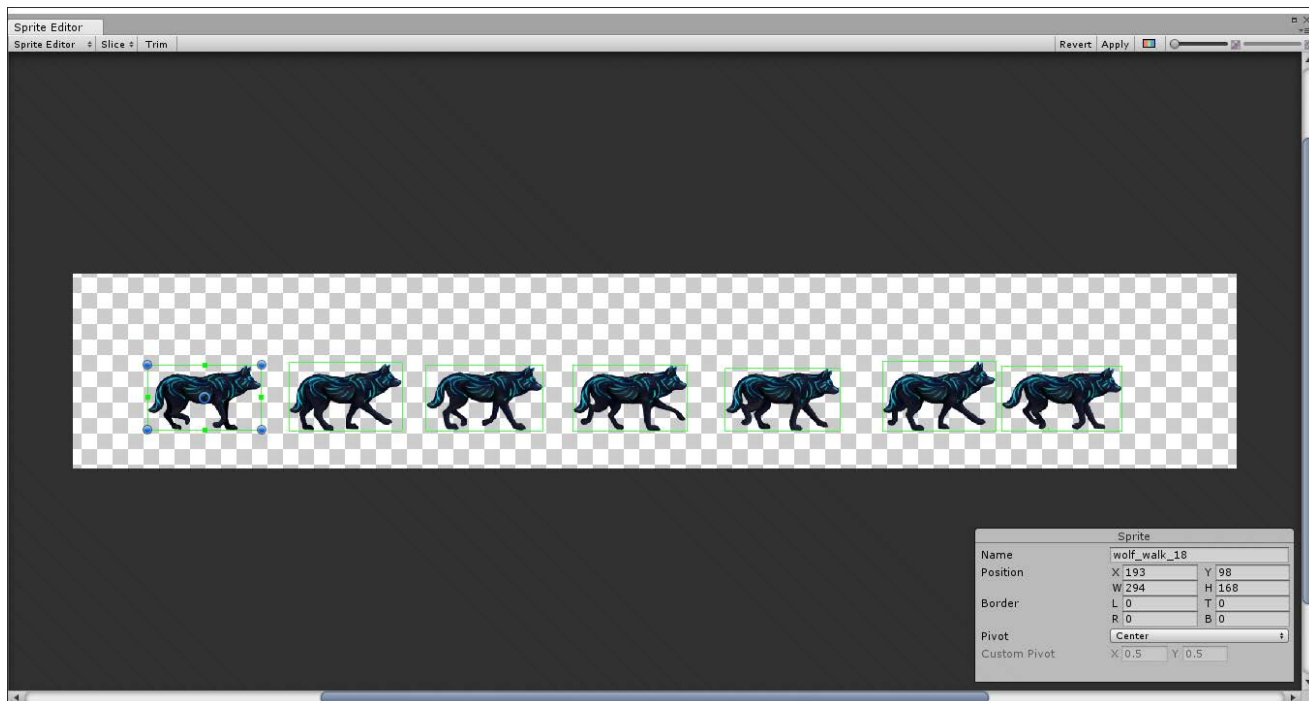


Рисунок 8 — Окно Sprite Editor

Затем создаем Animator Controller и добавляем к нашему персонажу. Он определяет правила и связи для анимаций определённого объекта.

Для создания самих анимаций, нужно в окне Animation создать новый клип анимации, перед этим выбрав нужный объект, а затем перенести все кадры необходимой анимации в данное окно в привальном порядке. Здесь настраиваем скорость анимации.

После создания всех необходимых клипов анимаций, переходим в окно Аниматора. Здесь нам необходимо создать связи и настроить условия переходов между различными анимациями, что мы уже создали ранее (рисунок 9). Для врагов делаем тоже самое.

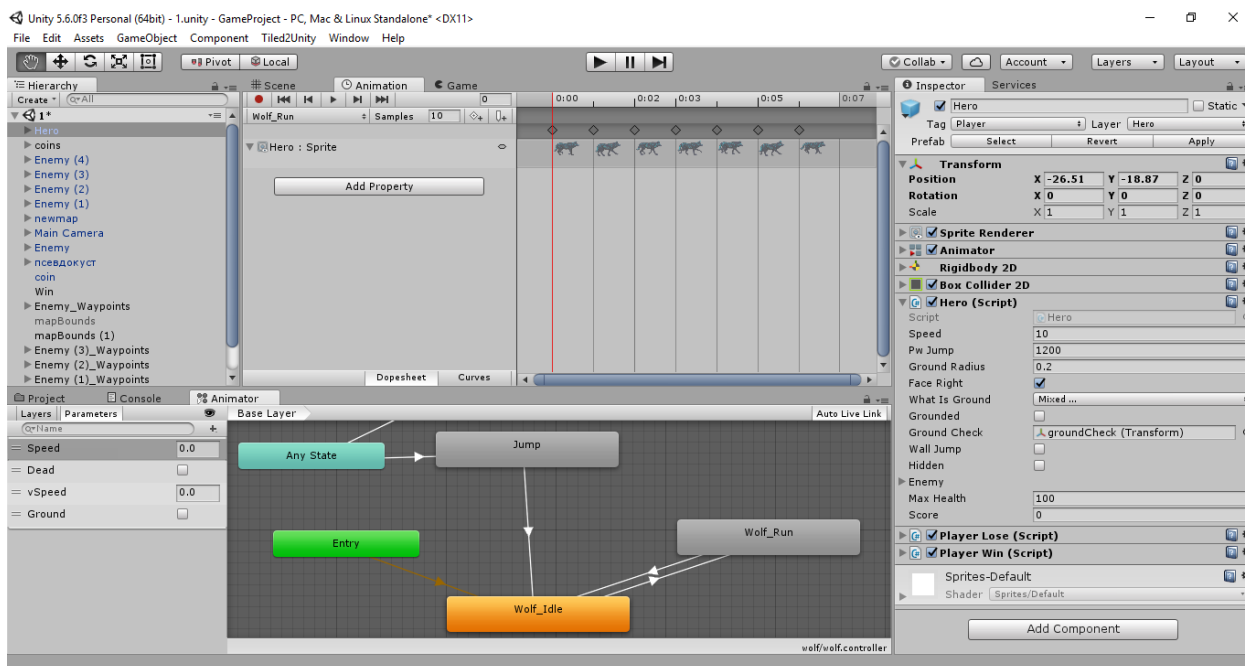


Рисунок 9 — Создание Анимаций

Если запустить уровень, мы сможем побегать в разные стороны, попрыгать, и совершать остальные доступные игроку действия.

Теперь нам необходимо создать уровень, после чего расставить на нем различные объекты и врагов. Первым делом нужно прикинуть примерный план уровня, а так же расположение врагов и объектов. После того как общий план уровня будет достаточно проработан, можно приступать к реализации самого уровня.

Уровень можно собрать прямо в окне сцены Unity, но такой способ отнимать колоссальное количество времени, так как приходится правильно позиционировать тайлы карты относительно друг друга. По этой причине, было решено использовать стороннюю программу для реализации данной задачи Tiled (рисунок 10).

Tiled распространяется бесплатно и предоставляет широкий функционал в плане реализации двумерных уровней при помощи тайловой графики. Интерфейс данной программы очень просто и понятный. При запуске программы мы видим поле, на котором будет собираться уровень, справа находится окно Слоев, окно Тайлов, а так же Миникарта, что бы легче было ори-

ентироваться в плане постройки уровня. Слева расположено окно Параметров.

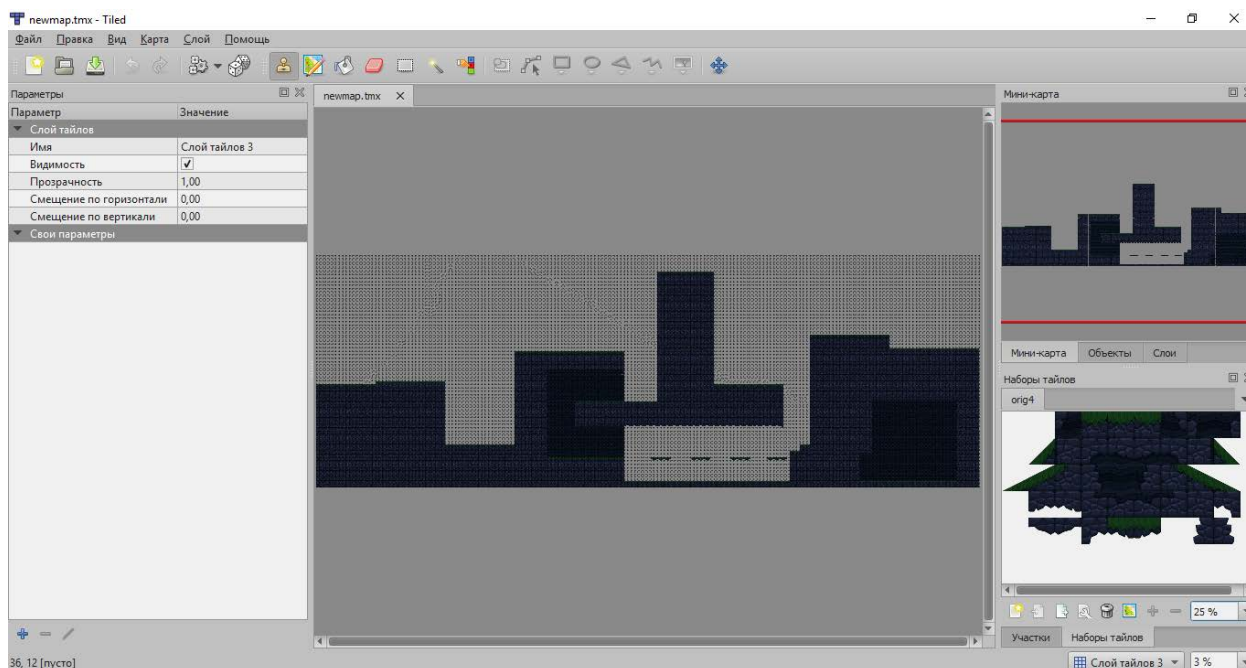


Рисунок 10 — Окно Tiled с готовой картой

Для создания карты необходимо в первую очередь импортировать нужный набор тайлов, после чего сборка уровня напоминает простую мозаику. В этой же программе можно настроить коллайдеры для каждого тайла отдельно.

К сожалению, Unity не принимает форматы Tiled напрямую, из-за чего приходится пользоваться дополнительным софтом Tiled2Unity (рисунок 11), предоставляемым сторонними людьми отдельно и так же бесплатно.

Данная программа с легкостью импортирует стандартный формат Tiled в ваш проект Unity.

После импортирования, уровень появится в вашем проекте. Теперь нам только остаётся наполнить готовый импортированный из Tiled уровень всеми необходимыми элементами, в то числе и врагами. После этого создание первого уровня разрабатываемого проекта можно считать окончанным.

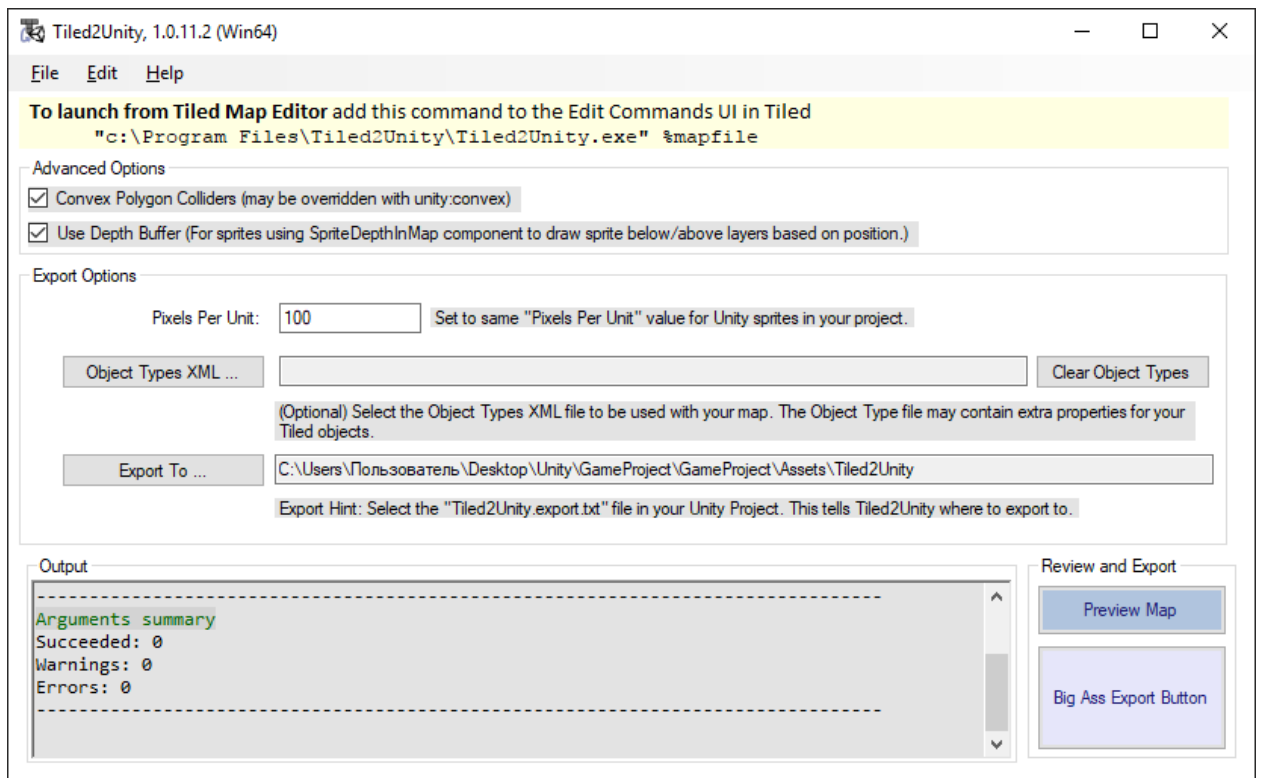


Рисунок 11 — Окно Tiled2Unity

Проверить работоспособность проекта можно прямо в окне Unity, для этого необходимо нажать кнопку запуска проекта в верхней части экрана. Рядом с ней находятся кнопки паузы и окончания теста игры. Они имеют такой же вид, как кнопки управления в любом другом аудио- или видеоплеере.

Вид запущенного проекта из окна Unity представлен на рисунках 12-14.

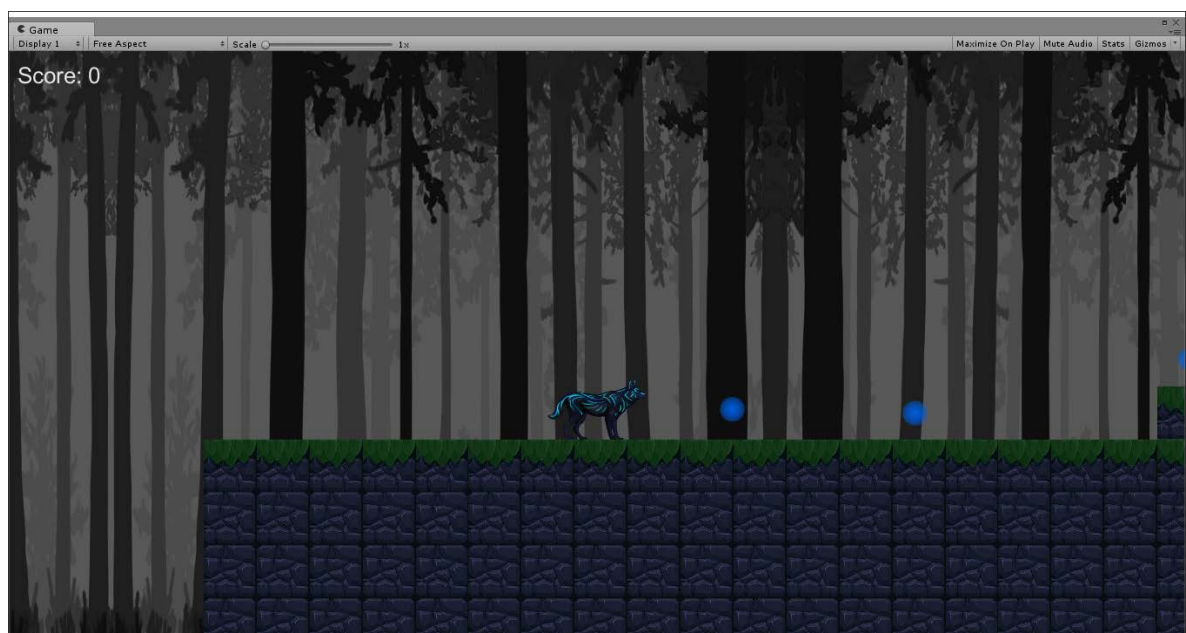


Рисунок 12 — Начало уровня



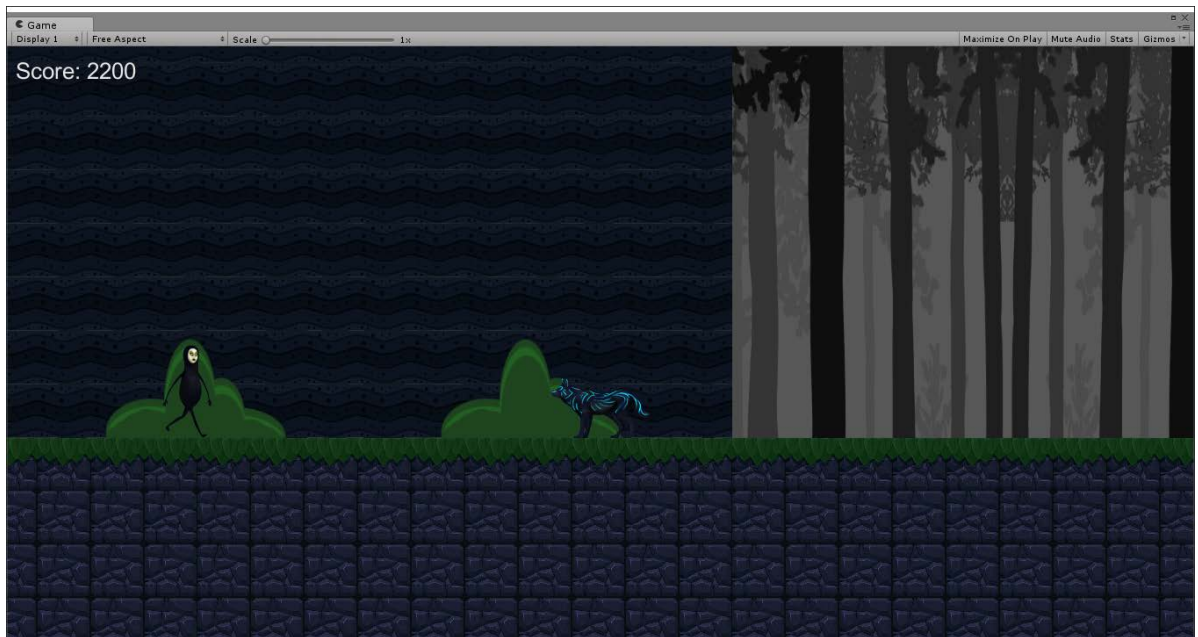


Рисунок 13 — Враг

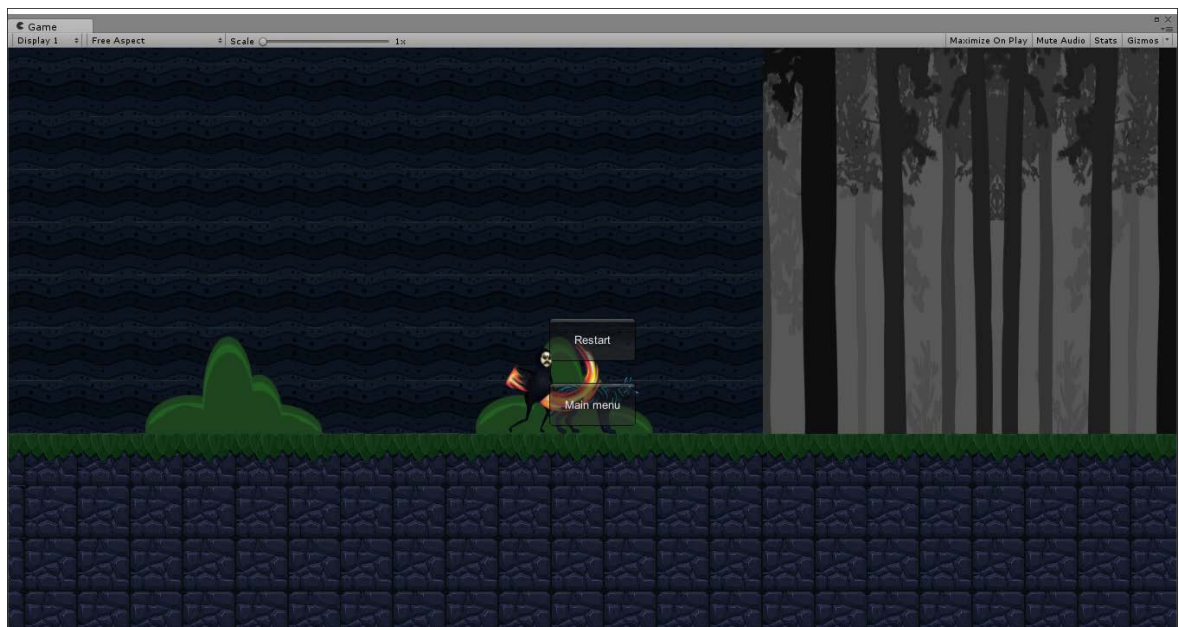


Рисунок 14 — Проигрыш игрока

### 2.3.3 Тестирование

Стоит отметить, что на данном этапе реализации проекта проводить какое-либо крупное тестирование смысла нет, так как в игре реализована лишь малая толика возможностей, что предполагалось реализовать. Именно по этой причине данное тестирование (таблица 7), является лишь небольшой проверкой на соответствие основным функциональным требованиям.

Таблица 7 — 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 установленном направлении, корректное проигрывание анимации	Персонаж передвигался в нужном направлении, корректное отображение анимации	Пройден
2	Проверка на корректный прыжок	Используя клавишу прыжка, подпрыгнуть несколько раз подряд	Игрок может сделать прыжок только с земли, корректное проигрывание анимации	Персонаж способен совершить прыжок только с земли, верное исполнение анимации	Пройден
3	Проверка на возможность персонажа прятаться	Подойдя к объекту и нажав клавишу, спрятаться, а затем выйти обратно.	Игрока не видно, враги не могут обнаружить персонажа, когда герой прячется, он не может двигаться	Игрок пропадает с экрана, и не может передвигаться, враги не обнаруживают героя.	Пройден
4	Проверка поведения врагов	Подойти к врагу на достаточно близкое расстояние, сбежать от врага.	Пока игрок находится вне поля зрения врага, патрулирует местность. Если герой попадает в поле зрения противника, враг начинает его преследовать, если во время преследования игрок сбегает (прячется, спрыгивает с большой высоты) останавливается на короткое время, затем возвращается к точке старта.	Патрулирует местность, если находится в пассивном состоянии, когда замечает игрока, преследует его, если игрок сбежал, останавливается на короткое время и затем возвращается на стартовую точку.	Пройден
5	Проверка на корректную реализацию события проигрыша игрока	Позволить противнику атаковать игрока	Когда враг атакует игрока, проигрывается анимация смерти персонажа и анимация атаки у противника. Время останавливается и появляется меню выбора действия.	Враг атакует игрока, анимация проигрывается верно, но не проигрывается анимация смерти главного героя, появляется меню выбора действия	Частично пройден

#### Окончание таблицы 7

6	Проверка на корректную реализацию события выигрыша игрока	Дойти до конца уровня	При достижении конца уровня появляется меню с выбором действия, время останавливается	При достижении конца уровня останавливается время и появляется меню	Пройден
7	Проверка на корректную работу меню проигрыша	1)Проиграть и перезапустить уровень 2)Проиграть и выйти в главное меню	Есть возможность перезапустить уровень или выйти в главное меню.	В зависимости от выбора перезапускается уровень, либо выходит в главное меню	Пройден
8	Проверка на корректную работу меню выигрыша	1)Выиграть и перезапустить уровень 2)Выиграть и выйти в главное меню	Есть возможность перезапустить уровень или выйти в главное меню.	В зависимости от выбора перезапускается уровень, либо выходит в главное меню	Пройден
9	Проверка на корректную работу главного меню	1)Запустить уровень 2)Выйти из игры	Есть возможность запустить уровень, либо выйти из игры	В зависимости от выбора запускается уровень, либо закрывается игра	Пройден
10	Проверка на корректный сбор собираемых объектов	Собрать коллекционируемые объекты	При сборе коллекционируемых объектов, собранный объект уничтожается, счет увеличивается на 100 очков	При сборе объекта, он исчезает, и счет увеличивается на 100 очков	Пройден

Данное тестирование показывает, что на данном этапе реализации проекта, игра соответствует большинству функциональных требований выделенных на этапе постановки задачи. Но все же стоит уделить особое внимание программному коду и исправить ошибку, из-за которой ситуация проигрыша игрока отрабатывается неверно.

### 2.3.4 Технические характеристики

Прежде всего, стоит понимать, что данный проект не является полностью готовой к релизу компьютерной игрой, и по своей сути является прототипом стоящим примерно на уровне ранней альфа-версии. Вызвано это было тем, что создание качественного продукта занимает колоссально количество



времени. Так, к примеру, разработка небольшой игры для мобильных телефонов с процедурно генерируемым уровнем может занимать более года, при условии, что над проектом работает не более трех человек. Увеличение количества человек может ускорить разработку, но не сильно.

На данный момент очень сложно определить какими системными требованиями будет обладать проект в будущем, так как большое количество элементов просто не реализованы. На данный момент мы имеем лишь элементы, отвечающие за основную игровую механику, а значит, в будущем технические характеристики для разрабатываемой компьютерной игры могут очень сильно измениться.

Сейчас можно с уверенностью утверждать лишь то, что для запуска данного прототипа потребуется компьютер, обладающий системными характеристиками не ниже минимально допустимых для корректной работы игрового движка Unity, для которого наиболее важным аспектом является только видеокарта. Она должна поддерживать DirectX 9 с шейдерами не ниже версии 3.0. Это означает, что в данный момент созданный прототип с большой вероятностью пойдет на большинстве компьютеров.

### **2.3.5 Калькуляция проекта**

Для реализации проекта были использованы множество различных элементов, начиная от статичных картинок и заканчивая несколькими скриптами для реализации механики игры.

Во время разработки игры было создано 2 сцены: Главное меню и Игровой уровень. В главном меню присутствуют лишь фоновое изображение и 2 кнопки, отвечающие за запуск Игрового уровня и выход из игры.

Общее количество объектов, которые были использованы или созданы во время реализации проекта, представлены в таблице 8.

Таблица 8 — Калькуляция элементов игры

Элемент	Комментарий
Фоновые изображения	2 растровых изображения
Тайлы	1 набор из 47 тайлов
Главный герой («Черный Пес»)	4 анимации, 25 спрайтов
Враг («Черная сущность»)	3 анимации, 23 спрайта
Коллекционируемые объекты	1 спрайт
Скрипты	16 скриптов с общим количеством строчек кода 1034
Прочие элементы	1 спрайт, 1 аудиозапись

В Игровом уровне присутствуют все основные элементы игры, и в общей сложности содержит более 90 объектов, как повторяющихся, так и нет. В том числе главный персонаж, 6 врагов, несколько подвижных платформ, объекты, за которыми игрок может спрятаться, а так же собираемые объекты.

## ЗАКЛЮЧЕНИЕ

Во время анализа доступных источников было проведено исследование понятия компьютерная игра, во время которого была проведена классификация компьютерных игр по 4 критериям, но из-за сравнительной молодости игровой индустрии, а так же того, что классификация компьютерных игр не была систематизирована, составить подробную классификацию не удалось. Дополнительно был составлен алгоритм разработки видеоигр.

Были проанализированы популярные средства разработки. В ходе анализа, было проведено их сравнение и выбраны наиболее актуальные средства разработки для начинающих разработчиков. Выбор приоритетных средств разработки проходил по двум критериям: доступность и функциональность.

При анализе существующих разработок, был проведено их сравнение и выделены их достоинства и недостатки. В ходе анализа стало ясно, что при разработке компьютерной игры с простой игровой механикой, стоит обратить внимание на дополнительные элементы игры, такие как сюжет и графическое оформление. Это нужно для того, что бы удержать потенциального игрока и продлить жизненный цикл разработки.

Основываясь на все полученной в ходе исследования информации, было решено разработать прототип двумерного платформера для одного игрока на игровом движке Unity. Такое решение было принято по нескольким причинам:

- 1) двумерная графика, в отличие от трехмерной легче в создании;
- 2) по игровой механике, игра жанра платформер проще реализуется;
- 3) игровой движок Unity распространяется бесплатно и позволяет разрабатывать приложения на языке программирования C#.

После выбора средств разработки было начато изучение Unity, а так же разработка самого проекта. В ходе разработки был изучен игровой движок Unity и были приобретены необходимые знания и умения, а именно:

- создание сцен;
- создание анимаций;
- создание и написание скиптов;
- настройка объектов;
- создание UI;
- компиляция проекта.

Освоение среды разработки Unity несет не маловажный характер, так как в современном мире индустрия разработки игр все сильнее распространяется в нашем обществе. Игры перестали быть лишь предметом для развлечений, и теперь используются и в других областях, например, в науке или в обучении пользователей. Поэтому развитие в данном направлении можно считать одним из самых важных в современном обществе.

В ходе реализации проекта были выполнены следующие задачи:

- 1) изучены особенности и состояние компьютерной индустрии России;
- 2) выбраны жанр, вид и платформа для компьютерной игры;
- 3) разработаны сценарий и концепция основных элементов;
- 4) выбрано и изучено средство реализации;
- 5) подготовлены необходимые для игры анимации;
- 6) реализован прототип игры.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Ильин В. Основы создания 2D персонажа в Unity 3D 4.3 [Электронный ресурс]/ Ильин Вячеслав // Хабрахабр. — Режим доступа: <https://habrahabr.ru/post/211472/> (дата обращения: 04.05.2017).
2. Компьютерные игры как искусство [Электронный ресурс]. — Режим доступа: <http://gamesisart.ru> (дата обращения: 04.05.2017).
3. Официальный интернет-портал правовой информации [Электронный ресурс]. — Режим па: <http://publication.pravo.gov.ru/Document/View/0001201704140005?index=0&rangeSize=1> (дата обращения: 04.05.2017).
4. «Средний возраст российского геймера достиг 33 лет» [Электронный ресурс] //Канобу. — Режим доступа: <http://kanobu.ru/news/srednij-voznrast-rossijskih-gejmerov-dostig-33-let-371699/> (дата обращения: 14.05.2017).
5. Торн А. Основы анимации в Unity [Текст]: учебное пособие/ Алан Торн– ред.: Д. Мовчан, переводчик: Р. Рагимов – Москва: ДМК, 2016 – 176с.
6. Хокинг Дж. Unity в действии. Мультиплатформенная разработка на C# [Текст]: учебное пособие/ Джозеф Хокинг – Санкт-Петербург: Питер, 2016 – 336с.
7. Adobe Photoshop [Электронный ресурс]. — Режим па: <http://www.adobe.com/ru/products/photoshop.html><https://unity3d.com/ru> (дата обращения: 04.05.2017).
8. CorelDRAW [Электронный ресурс]. — Режим па: <http://www.coreldraw.com/ru/><https://unity3d.com/ru> (дата обращения: 04.05.2017).
9. Create Level in Tiled, Import to Unity using Tiled2Unity, YouTube [Электронный ресурс]: видеоролик. — Режим доступа: <https://www.youtube.com/watch?v=PRZFARVoZj0> (дата обращения: 04.05.2017).

10. Duck Tales Remastered [Электронный ресурс]. — Режим доступа: [http://store.steampowered.com/app/237630/DuckTales\\_Remastered/](http://store.steampowered.com/app/237630/DuckTales_Remastered/) (дата обращения: 05.05.2017).

11. Dust: An Elysian Tail (wiki) [Электронный ресурс]. — Режим доступа: [http://ru.elysiantail.wikia.com/wiki/Dust:\\_An\\_Elysian\\_Tail\\_\(%D0%B8%D0%B3%D1%80%D0%B0\)](http://ru.elysiantail.wikia.com/wiki/Dust:_An_Elysian_Tail_(%D0%B8%D0%B3%D1%80%D0%B0)) <https://unity3d.com/ru> (дата обращения: 05.05.2017).

12. Game Maker: Studio [Электронный ресурс]. — Режим па: <https://www.yoyogames.com/gamemakerhttps://unity3d.com/ru> (Дата обращения: 04.05.2017).

13. Limbo [Электронный ресурс]. — Режим па: <http://playdead.com/games/limbo/> <https://unity3d.com/ru> (дата обращения: 05.05.2017).

14. Making a 2D Platformer in Unity (C#) - Wall Jump, YouTube [Электронный ресурс]: видеоролик. — Режим па: [https://www.youtube.com/watch?v=9QjwHsjbX\\_A](https://www.youtube.com/watch?v=9QjwHsjbX_A) (дата обращения: 04.05.2017).

15. Mark of the Ninja [Электронный ресурс]. — Режим па: <https://www.kleientertainment.com/games/mark-ninjahttps://unity3d.com/ru> (Дата обращения: 05.05.2017).

16. Ori and the Blind Forest [Электронный ресурс]. — Режим па: <http://www.oriblindforest.com/https://unity3d.com/ru> (дата обращения: 05.05.2017).

17. PaintTool SAI [Электронный ресурс]. — Режим па: <http://mypainttoolsai.ru/https://unity3d.com/ru> (дата обращения: 04.05.2017).

18. Sliding & Bouncing in 2D - Official Unity Tutorial, YouTube [Электронный ресурс]: видеоролик. — Режим па: <https://www.youtube.com/watch?v=23HEMKfMKKk> (Дата обращения: 04.05.2017).

19. Unity 5 2D Platformer Tutorial - Part 20 - Melee Attacking, YouTube [Электронный ресурс]: видеоролик —

<https://www.youtube.com/watch?v=xwPahXLpNh8> (дата обращения: 04.05.2017).

20. Unity 5 tutorial for beginners: 2D Platformer - Moving Platform, YouTube [Электронный ресурс]: видеоролик. — Режим па: [https://www.youtube.com/watch?v=4R\\_AdDK25kQ](https://www.youtube.com/watch?v=4R_AdDK25kQ) (дата обращения: 04.05.2017).

21. Unity Game Engine [Электронный ресурс]. — Режим па: <https://unity3d.com/ru> (Дата обращения: 04.05.2017).

22. Unity Manual, Unity Documentation [Электронный ресурс]: справочник. — Режим доступа: <https://docs.unity3d.com/Manual/> (дата обращения: 04.05.2017).

23. UNITY3D Мини-урок [9] - Звуки и их проигрывание - audio.Play(), YouTube [Электронный ресурс]: видеоролик. — Режим па: <https://www.youtube.com/watch?v=EGI32DEBfsM> (дата обращения: 04.05.2017).

24. Unreal Engine [Электронный ресурс]. — Режим па: <https://www.unrealengine.com><https://unity3d.com/ru> (дата обращения: 04.05.2017).

25. Wings of Vi [Электронный ресурс]. — Режим па: <http://www.grynsoft.com/wings-of-vi><https://unity3d.com/ru> (дата обращения: 05.05.2017).

# ПРИЛОЖЕНИЕ

**Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования**

**«Российский государственный профессионально-педагогический университет»**

Институт инженерно-педагогического образования  
Кафедра информационных систем и технологий  
направление подготовки 09.03.02 Информационные системы и технологии  
профилю подготовки «Информационные технологии в медиаиндустрии»

УТВЕРЖДАЮ

Заведующий кафедрой

\_\_\_\_\_ Н. С. Толстова

«\_\_\_\_\_» \_\_\_\_\_ 2017 г.

## ЗАДАНИЕ

### на выполнение выпускной квалификационной работы бакалавра

студента 4 курса, группы ИТМ-401 Байбатырова Ильи Львовича

1. Тема Компьютерная игра платформер  
утверждена распоряжением по институту от 07.02.2017 г. № 73.
2. Руководитель Толстова Наталья Сергеевна, канд. пед. наук, зав. кафедрой ИС
3. Место преддипломной практики Российский государственный профессионально-педагогический университет
4. Исходные данные к ВКР
5. Содержание текстовой части ВКР (перечень подлежащих разработке вопросов)
  - 1 Исследование понятия «Компьютерная игра» и обзор технологий разработки компьютерных игр
    - 1.1 Исследование понятия и классификация компьютерных игр
    - 1.2 Общий алгоритм реализации проекта
    - 1.3 Средства создания компьютерных игр
    - 1.4 Анализ существующих разработок
  - 2 Разработка проекта
    - 2.1 Описание целевой аудитории
    - 2.2 Постановка задачи проекта
    - 2.3 Реализация проекта
6. Перечень демонстрационных материалов  
Презентация к докладу  
Видеоролик, демонстрирующий проект



