**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«Воронежский Государственный Университет»**

КУРСОВАЯ РАБОТА

по дисциплине: «Инструментальные средства разработки ПО»

на тему:

«Разработка игры на Unity»

Выполнили:

студенты 3 курса СПО математического факультета

Мурзинова Анна Сергеевна

Сорокопуд Вадим Алексеевич

Саркисов Роман Артёмович

Преподаватель:

Зенин Кирилл Вячеславович

Воронеж 2023

Оглавление

[Введение. 3](#_Toc134975310)

[Глава 1. Основы разработки игр на платформе Unity. 4](#_Toc134975311)

[1.1. Основы Unity, возможности, особенности. 4](#_Toc134975312)

[1.2. Плюсы и минусы платформы Unity. 5](#_Toc134975313)

[1.3. Язык программирования C#. 6](#_Toc134975314)

[1.4. Диаграмма вариантов использования. 7](#_Toc134975315)

[Глава 2. Реализация проекта. 10](#_Toc134975316)

[2.1. Структура проекта. 10](#_Toc134975317)

[2.2. Реализация игрового персонажа. 12](#_Toc134975318)

[2.3. Реализация фона 13](#_Toc134975319)

[2.4. Реализация игровых объектов. 14](#_Toc134975320)

[2.5. Реализация механик игры. 14](#_Toc134975321)

[3. Тестирование. 15](#_Toc134975322)

[4. Заключение. 17](#_Toc134975323)

[5. Список литературы. 18](#_Toc134975324)

# Введение.

Современный мир развивается с неумолимой скоростью. Нынешний день уже невозможно представить без такого вида отдыха, как компьютерные игры. Индустрия компьютерный игр начала свое развитие еще в 1970 году, а сейчас уже достигла небывалого пика. С каждым годом выходят всё более культовые игры, которые в последствие запоминаются надолго.

Сейчас уже не секрет, что каждый человек при особом желании и стремлении изучать новое способен самостоятельно создать свою игру. Для этого даже не обязательно собирать большую команду, можно заниматься этим индивидуально. Будет достаточно лишь изучить вопросы, без которых создание игры будет невозможно, такое как: где и как это делать? Как осуществлять данный процесс решать только вам. Также, как и какой дизайн будет задействован, какой жанр вы захотите использовать и многое другое.

**Целью курсовой работы** является изучение самого процесса создания игры, изучения одной из популярных платформ. Также мы рассмотрим базовые элементы, которые присутствуют в играх.

**Задачи курсовой работы:**

— изучить основы создания компьютерных игр;

— рассказать о выбранной платформе для создания игры, описать её плюсы и минусы.

— описать внутренние процессы игры;

— показать процесс создания игры со стороны разработчика и со стороны дизайнера;

— протестировать созданную игру, рассмотреть её недостатки и преимущества на данном этапе разработки.

# Глава 1. Основы разработки игр на платформе Unity.

## 1.1. Основы Unity, возможности, особенности.

Unity — это не просто движок, а ведущая платформа разработки и поддержки интерактивного RT3D-контента (технология компьютерной графики, которая генерирует интерактивный контент быстрее человеческого восприятия). Разработчики игр, от программистов до художников, архитекторов, дизайнеров автомобилей, режиссеров и других специалистов, используют Unity для воплощения своих идей.

Unity позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры. Выпуск Unity состоялся в 2005 году и с того времени идёт постоянное развитие. Проект в Unity делится на сцены (Рис. 1) — 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 («пустышки»). Объекты, в свою очередь содержат наборы компонентов, с которыми и взаимодействуют скрипты.



**Рис. 1** Игровая сцена в Unity.

Также у объектов есть название (в Unity допускается наличие двух и более объектов с одинаковыми названиями в одной сцене), может быть тег (метка) и слой, на котором он должен отображаться. Так, у любого объекта на сцене обязательно присутствует компонент Transform — он хранит в себе координаты местоположения, поворота и размеров объекта по всем трём осям. Движок поддерживает множество популярных форматов.

Этот же формат используется во внутреннем магазине Unity Asset Store, в котором разработчики могут бесплатно и за деньги выкладывать в общий доступ различные элементы, нужные при создании игр.

## 1.2. Плюсы и минусы платформы Unity.

Как правило, игровой движок предоставляет множество функциональных возможностей, позволяющих их задействовать в различных играх, в которые входят моделирование физических сред, карты нормалей, динамические тени и многое другое. В отличие от многих игровых движков, у Unity имеется два основных преимущества: наличие визуальной среды разработки и межплатформенная поддержка.

Первый фактор включает не только инструментарий визуального моделирования, но и интегрированную среду, цепочку сборки, что направлено на повышение производительности разработчиков, в частности, этапов создания прототипов и тестирования. Под межплатформенной поддержкой предоставляется не только места развёртывания (установка на персональном компьютере, на мобильном устройстве, консоли и т. д.), но и наличие инструментария разработки (интегрированная среда может использоваться под Windows и Mac OS).

Вторым преимуществом называется модульная система компонентов Unity, с помощью которой происходит конструирование игровых объектов, когда последние представляют собой комбинируемые пакеты функциональных элементов. В отличие от механизмов наследования, объекты в Unity создаются посредством объединения функциональных блоков, а не помещения в узлы дерева наследования. Такой подход облегчает создание прототипов, что актуально при разработке игр.

В качестве недостатков приводятся ограничение визуального редактора при работе с многокомпонентными схемами, когда в сложных сценах визуальная работа затрудняется.

Вторым недостатком называется отсутствие поддержки Unity ссылок на внешние библиотеки, работу с которыми программистам приходится настраивать самостоятельно, и это также затрудняет командную работу.

Ещё один недостаток связан с использованием шаблонов экземпляров. С одной стороны, эта концепция Unity предлагает гибкий подход визуального редактирования объектов, но с другой стороны, редактирование таких шаблонов является сложным. Также, WebGL-версия движка, в силу специфики своей архитектуры (трансляция кода из C# в C++ и далее в JavaScript), имеет ряд нерешённых проблем с производительностью, потреблением памяти и работоспособностью на мобильных устройствах.

# 1.3. Язык программирования C#.

**C#** — объектно-ориентированный язык программирования общего назначения. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework и .NET Core. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

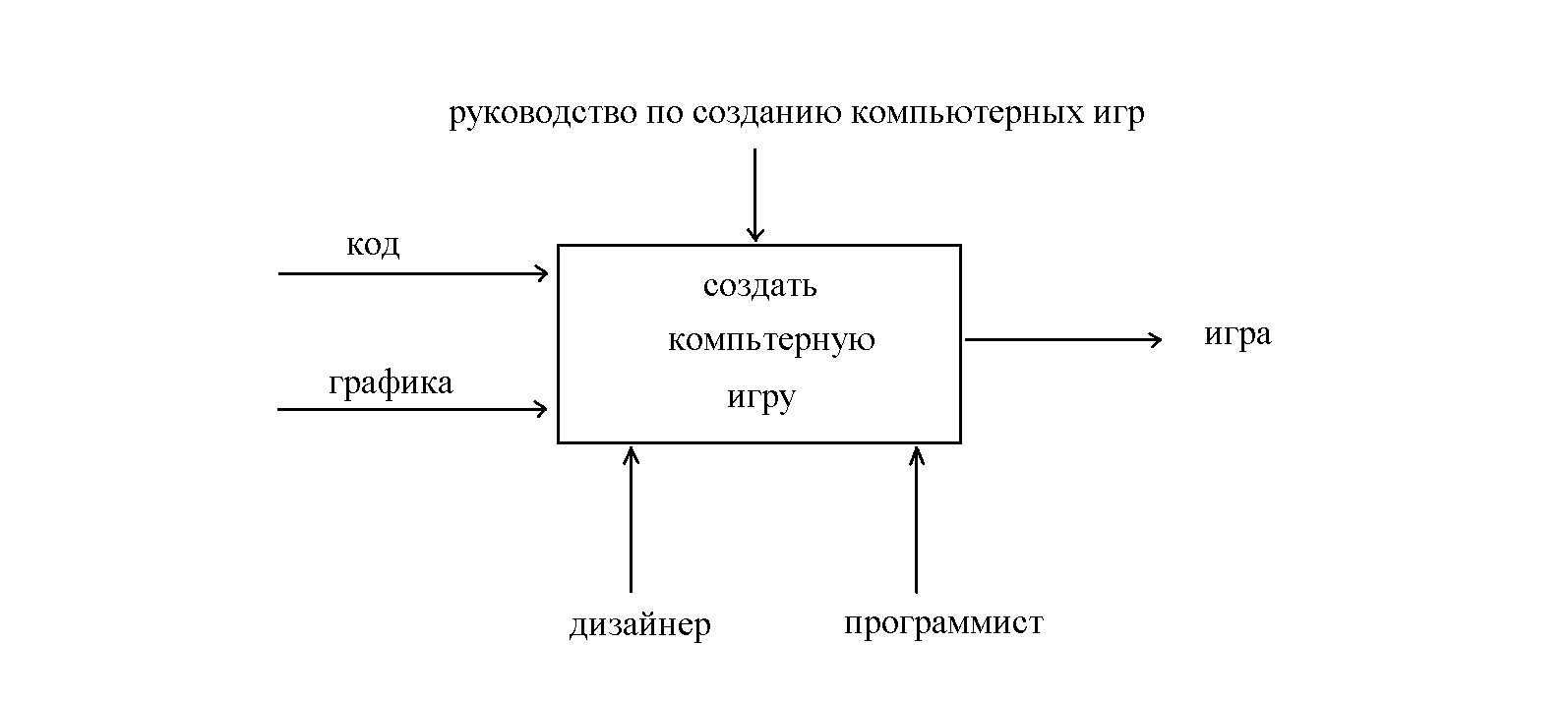
C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы.

Переняв многое от своих предшественников —языков C++, Delphi, Модула, Smalltalk и, в особенности, Java — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественная реализация интерфейсов).

## 1.4. Диаграммы вариантов использования.

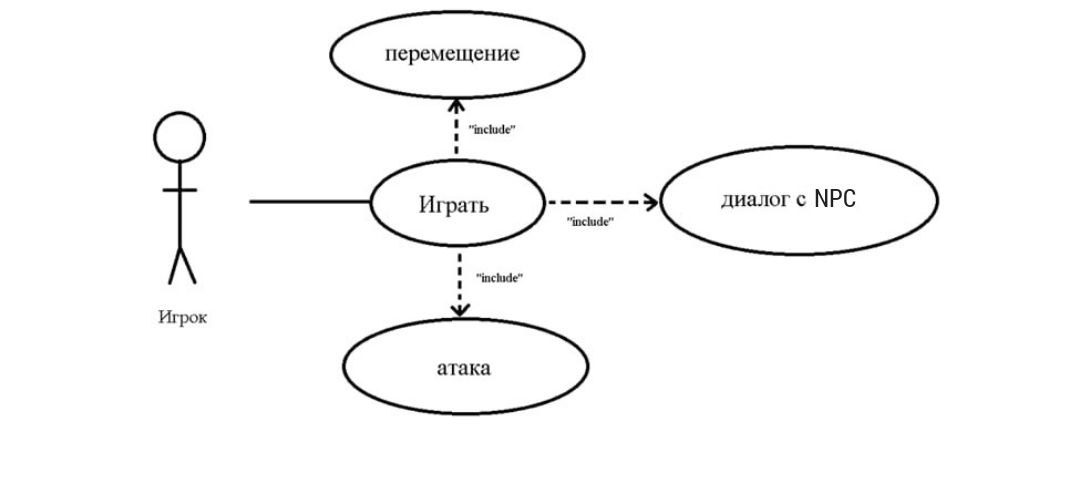
Первоначально рождается вопрос о том, с чего вообще начать создание игры. У каждого проекта на начальном уровне есть базовые исходные данные и цель, которую нужно достичь.

Для этого была задействована контекстная диаграмма (Рис. 2), на которой представлены краткие требования к созданию игры. В ней вы можете увидеть данные входа и выхода, а также требования и исполнители.



**Рис. 2** Диаграмма создания компьютерной игры.

В ходе анализа требований к игре была также разработана диаграмма, описывающая возможности игрового персонажа. (Рис. 3). Персонаж (главный герой) — объект, управляемый игроком, перемещается по игровому полю по 4 сторонам света. Перемещение осуществляется с помощью клавиатурных клавиш «WASD», клавиша «E» в нашем случае отвечает за атаку.



**Рис. 3** Диаграмма игрового персонажа.

Рассмотрим немного детальнее момент с передвижением персонажа. Как уже говорилось ранее, будет доступна некая игровая сцена, на которой и будет происходить основное действие игры. Персонаж сможет свободно перемещаться по ней, единственное, он не сможет заходить в те зоны, которые на этапе разработки будут заблокированы. Например, он не сможет пройти сквозь невидимую стену или не сможет ходить сквозь текстуры. Также в игре будет постоянная скорость передвижения, а движение будет осуществляться в разные стороны света, но также будет доступно движение по диагоналям

Из особенностей в игре не будет доступно самонаведение на вражеские объекты, поэтому задача пользователя будет правильно рассчитать расстояние и скорость попадания снаряда в противника. Снаряды также будут реализованы не как отдельные объекты, которые нужно где-то собирать или создавать самостоятельно, они будут встроены и доступны с самого начала игры.

В игре будут доступны диалоги с неигровыми персонажами (или NPC). NPC — персонаж(и) внутри игры, которые не находятся под контролем игрока. Их поведение в играх определяется программно.

Они могут быть дружественными, нейтральными или вражескими. NPC могут быть дружественными, нейтральными и враждебными (Рис. 4). Неигровые персонажи служат важным средством создания игровой атмосферы, мотивируют игроков совершать те или иные действия и являются основным источником информации об игровом мире и сюжете игры. Частным случаем NPC в современных играх является босс — враждебный персонаж (как правило, главный антагонист), которого необходимо победить, чтобы пройти всю игру или отдельный уровень.



**Рис. 4** Пример враждебного NPC

Игроку представится возможность погружения в сюжет через диалоги с неигровыми персонажами. Для диалога необходимо будет подойти к данной модельке и нажать на определенную клавишу «Q». Откроется меню диалога, однако игрок не сможет давать NPC любые варианты ответов, сюжет и диалоги прописываются автоматически.

В дальнейшем рассматривается также добавление диалоговых окон еще к обычным внутриигровым предметам. Персонаж сможет подходить к определенным объектам и нажимая указанную выше кнопку сможет узнавать какие-то сюжетные моменты.

# Глава 2. Реализация проекта.

## 2.1. Структура проекта.

Для разработки данной игры было принято решение вести разработку на языке C# в среде разработки Visual Studio.

Microsoft Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE и т.д.

Всего в игровом приложении будут две основные сцены: игровое меню и сами игровые комнаты. Или, другими словами, основное поле для игры и меню. Рассмотри эти сцены детальнее.

Игровое меню:

При входе в игру пользователь будет видеть игровое меню. При нажатии на конкретные кнопки ему будут доступны разные действия.

В главном меню игрок может выполнить следующие действия:

1. «Начать игру»: при нажатии на данную кнопку игрок сможет запустить игровой процесс.

2. «Продолжить»: данная команда возвращает игрока к моменту игры, на котором он закончил прошлый раз, другими словами, воспроизводит последние совершенные действия в игре.

3. «Настройки»: в игре предполагаются настройки звука и интерфейса, поэтому должна быть возможность их отключения. Пользователь сможет настраивать громкость звука или, например, изменить назначение клавиш со стандартных на те, которые будут удобны ему.

4. «Об Авторах»: данный раздел имеет возможность кратких сведений об авторах игры и о ней в целом. Раздел для ознакомления.

5. «Выход»: возможность игроку покинуть данный игровой процесс.

Игровая сцена:

Данная сцена будет делиться на несколько разных под сцен. Задача пользователя будет перемещаться по комнате и проходить в последующие. Задний фон будет не задействован в игровом процессе, с ним нельзя взаимодействовать, также, как и с объектами в комнатах. Они служат в роли препятствий.

Задача главного героя в том, чтобы пройти все комнаты, оставшись в живых и одолеть конечного антагониста, главного босса игры. Для этого персонажу будет доступна помощь разного формата. В каких-то местах важную роль сыграет сам «скилл» пользователя, а где-то персонаж сможет использовать некое «лечение» или, другими словами, «хил» (Рис. 5)



**Рис. 5** Пример игрового хила (восстановление HP)

## 2.2. Реализация игрового персонажа.

Игровой персонаж представлен моделькой, которая отвечает за двумерное графическое представление игрового объекта, а также обладает своим собственным набором функций (рис.6).



**Рис. 6** Модель игрового персонажа.

Персонаж может свободно перемещаться по игровому полю, взаимодействовать с внутриигровыми объектами. Также у персонажа есть встроенные анимации:

1. бег (в сторону экрана, вправо, влево, спиной к экрану)

2. статичные положения (в сторону экрана, вправо, влево, спиной к экрану)

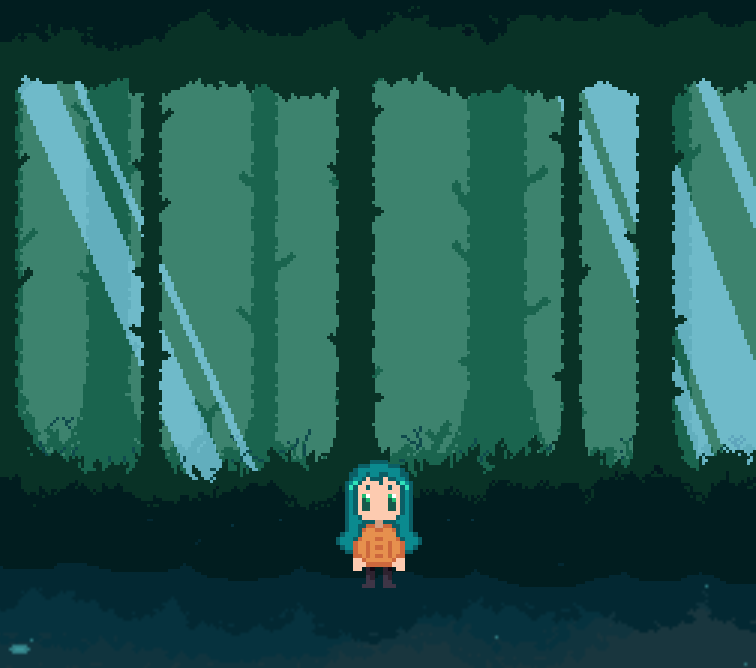
У персонажа также предполагается графа здоровья и маны. При неудачных исходах сражения с вражескими персонажами будет отбавляться значения здоровья или HP, а во время использования атак будет отбавляться запас маны.

## 2.3. Реализация фона

Задний фон отвечает за отображения фона игрового приложения. Данный скрипт отвечает за непрерывное движение фона с одинаковыми заданной скоростью и направлением. Он использует два одинаковых элемента, для применения их в качестве фона. При движении справа налево, в тот момент, когда правый край фона пересекается с правым краем камеры, создается новый объект фона, левый край которого стыкуется с правым краем предыдущего объекта. В тот момент, когда объект фона покидает пределы камеры, происходит уничтожение данного объекта. Благодаря этому, создается впечатление непрерывного движения фона в пределах экрана.

Также с фоном задействованы еще декоративные элементы игры. Они располагаются в игровом пространстве и служат невидимой областью, куда игрок не может зайти.

Ниже представлен пример фона данной игры (Рис. 7). В области, где задний фон отделяется от основной территории, где персонаж может двигаться, располагается коллайдер, предназначенная для расчёта столкновений с другими объектами.



**Рис. 7** Пример игрового фона.

## 2.4. Реализация игровых объектов.

Игровые объекты в игре — это некое дополнение к геймплею. Чаще всего они неподвижны и не могут быть задействованы пользователем. Представлены данные объекты могут совмещаться с фоном, но быть абстрагированы от него. Они создают некоторое препятствие пользователю, поскольку во время боя придется маневрировать не только между врагами, но и между объектами.

В данном случае игровые объекты это некоторые элементы локации: всевозможные препятствия для игрока, также и вражеские NPC и сам игрок. Можно сказать, что это один из базовых компонентов, из которых строится игра. Каждый физический игровой объект имеет значения поворота и положения. При этом допускается вложение игровых объектов друг в друга. Существует масса причин, по которым вам может потребоваться вложение объектов. Например, в целях структуризации будет целесообразно поместить некие объекты в родительскую сцену.

# 2.5. Реализация механик игры.

Механики игры или Механизм — способ взаимодействия с игровыми объектами в рамках установленных ограничений игры, изменяющий её состояние, влияющее на принятие игроком последующих решений. В основе механики лежит набор правил и способов, реализующий определенным образом некоторую часть интерактивного взаимодействия игрока и игры.

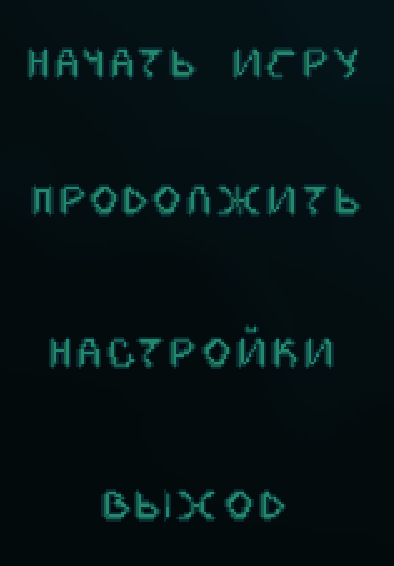
Основными механиками здесь является непосредственно передвижение и атака. В данной игре рассматривается вариант дальнего боя со стороны игрока и ближнего боя со стороны вражеских NPC. Атаки происходят по заданному сценарию: когда главный герой достигает достаточного расстояния до вражеских NPC он может использовать прописанную атаку.

# 3. Тестирование.

На данном этапе продукт находится ещё на этапе разработки. Уже сейчас всплывает множество сложностей, ошибок и идей того, что можно в последствие дополнить. Поскольку этот проект только начат, всё равно хочется выделить несколько пунктов, заслуживающих внимания.

Интерфейс данной игры проработан почти полностью. Главное меню игры и меню паузы пока что дорабатываются, в них следует немного подкорректировать дизайн, конкретно в главном меню необходимо проработать настройки. В них необходимо добавить графы звука и музыки, в дальнейшем кнопки смены управления клавиш.

Отображение текста и сам внутриигровой текст появится позднее, когда в игре будут прописаны диалоги с NPC и прочие игровые элементы. На данном этапе полностью устраивает текст главного меню. Он подходит по стилю и тематике игры, поэтому в дальнейшем видоизменён не будет (рис. 8).

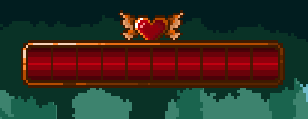


**Рис. 8** Игровое меню.

С отображением текстур следует в дальнейшем поработать, поскольку некоторые анимации не до конца закончены. Необходимо добавить базовые анимации покоя, а также проработать анимации вражеских NPC.

Что касается графики, здесь необходимо вносить большие правки. Текстура шкалы здоровья работает не стабильно, при полном убывании здоровья главный герой не умирает, ему хватает жизни еще на две атаки.

На данный момент в игру уже добавлена рабочая шкала здоровья персонажа (Рис. 9). Эта шкала отображает данное состояние персонажа, а при взаимодействии с разными объектами игры меняет свой статус. При контакте с вражескими NPC шкала будет уменьшаться на одно деление за каждое касание. Всего таких делений десять, а всего задействовано сто процентов здоровья. Когда же оно закончится персонаж скоропостижно умирает, и игра на этом заканчивается.



**Рис. 9** Шкала здоровья

Однако, если игрок использовал в процессе игры сохранения, то ему не придется начинать прохождения полностью сначала. Ему лишь нужно будет нажать «продолжить игру» и тогда он продолжит с последнего места сохранения.

# 4. Заключение.

В данной работе описан процесс создания игры на платформе Unity. Мы показали данный процесс от начала его планирования, а именно выбора платформы и языка, до его реализации, то есть создания игровых моделей, местности и прочего.

В ней были представлены также визуальные модели внутриигровых объектов. Они лишь демонстрируют один из многих вариантов того, что может создать человек с командой или в одиночку.

Конечно, работа в команде очень упрощает процесс, поскольку можно распределить разные роли и работу, для экономии сил и времени, также в команде присутствуют разные люди с разным видением и идеями. Гораздо проще развивать какие-либо задумки находясь в окружении людей. Но есть и те, кому комфортно работать в одиночку. Такая разработка игр возможна, если человек занимается разносторонним приёмом информации, умеет не только заниматься внутренней частью проекта, то есть кодом, но и обладает навыками дизайнера, возможно имеет даже некоторое музыкальное образование, для создания полноценной игровой атмосферы.

В ходе выполнения данной работы были получены следующие основные результаты:

1) выполнить анализ требований и разработать внешние спецификации;

2) выполнить проектирование игрового проекта;

3) реализовать игровой проект;

4) протестировать.

# 5. Список литературы.

1. Джереми Гибсон Бонд. Unity и C#. Геймдев от идеи до реализации. URL: [https://ru.pdfdrive.com/][1]

2. Джесси Шелл. Геймдизайн. URL: [https://ru.pdfdrive.com/ ][2]

3. Руководство Unity. URL: [https://docs.unity3d.com/][3]

4. Гамма Э., Хелм Р., Джонсон Р., Влиссидс Д. Приемы объектно-ориентированного программирования. Паттерны проектирования. – СПб.: Питер, 1994.

5. The Daily Telefrag — русскоязычный интернет-ресурс, посвящённый игровой индустрии, является ведущей русскоязычной площадкой для обмена опытом между разработчиками. [https://ru.wikipedia.org/wiki/DTF][4]

6. Введение в геймдизайн: Игровое состояние, игровые частицы, механики и другие базовые составляющие игр.: [сайт]. – URL: [https://clck.ru/][5]

7. Введение в геймдизайн: основные понятия и принципы проектирования игр.: [сайт]. – URL: [https://clck.ru/Nsk37][6]

8. Архнхейм, Р. Г. Искусство и визуальное восприятие / Р. Г. Арнхейм. – Москва: Архитектура-С, 2012. – 392 с. – ISBN 978-5-9647-0119-4.