

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра системного программирования**

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,  
профессор

\_\_\_\_\_ Л.Б. Соколинский

«\_\_\_» \_\_\_\_\_ 2024 г.

## **Разработка компьютерной игры 2D Платформер на Unity**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ЮУрГУ – 09.03.04.2024.308-568.ВКР

Научный руководитель,  
профессор кафедры СП, д.ф.-м.н.,  
доцент

\_\_\_\_\_ Р.Ж. Алеев

Автор работы,  
студент группы КЭ-403

\_\_\_\_\_ Е.И. Смирнов

Ученый секретарь  
(нормоконтролер)

\_\_\_\_\_ И.Д. Володченко

«\_\_\_» \_\_\_\_\_ 2024 г.

Челябинск, 2024 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет  
(национальный исследовательский университет)»**  
Высшая школа электроники и компьютерных наук  
Кафедра системного программирования

УТВЕРЖДАЮ

Зав. кафедрой СП

\_\_\_\_\_ Л.Б. Соколинский

29.01.2024 г.

### **ЗАДАНИЕ**

**на выполнение выпускной квалификационной работы бакалавра**  
студенту группы КЭ-403  
Смирнову Егору Игоревичу,  
обучающемуся по направлению  
09.03.04 «Программная инженерия»

- 1. Тема работы** (утверждена приказом ректора от 22.04.2024 г. № 764-13/12)  
Разработка компьютерной игры 2D Платформер на Unity.
- 2. Срок сдачи студентом законченной работы:** 03.06.2024 г.
- 3. Исходные данные к работе**
  - 3.1. Buttfield-Addison P., Manning J., Nugent T. Unity Game Development Cookbook. – O'Reilly Media, 2019. – С. 216–223.
  - 3.2. Hocking J. Unity in Action: Multiplatform game development in C# 2nd Edition. – Manning, 2018. – 175 с.
- 4. Перечень подлежащих разработке вопросов**
  - 4.1. Провести обзор аналогов.
  - 4.2. Спроектировать приложение.
  - 4.3. Реализовать и протестировать приложение.
- 5. Дата выдачи задания:** 29.01.2024 г.

**Научный руководитель,**  
профессор кафедры СП, д.ф.-м.н., доцент

Р.Ж. Алеев

**Задание принял к исполнению**

Е.И. Смирнов

## **ОГЛАВЛЕНИЕ**

ВВЕДЕНИЕ .....	4
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	6
1.1. Предметная область проекта.....	6
1.2. Анализ аналогичных проектов и существующих решений .....	6
1.3. Обоснование выбора средств реализации.....	10
2. АНАЛИЗ ТРЕБОВАНИЙ К ПРОЕКТИРУЕМОЙ СИСТЕМЕ .....	11
2.1. Функциональные и нефункциональные требования .....	11
2.2. Диаграмма вариантов использования.....	11
3. АРХИТЕКТУРА СИСТЕМЫ .....	13
3.1. Общее описание архитектуры системы .....	13
3.2. Диаграмма компонентов .....	14
3.3. Представление пользовательского интерфейса .....	15
4. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ СИСТЕМЫ .....	18
4.1. Компоненты системы .....	18
4.2. Функциональное и юзабилити тестирования.....	23
ЗАКЛЮЧЕНИЕ .....	25
ЛИТЕРАТУРА.....	26
ПРИЛОЖЕНИЯ.....	28
Приложение А. Спецификация вариантов использования .....	28
Приложение Б. Скриншоты итоговой версии игры .....	30

## **ВВЕДЕНИЕ**

### **Актуальность**

Несмотря на проблемы текущего времени и экономические потрясения, игровая индустрия продолжает расти и развиваться, что привлекает новых инвесторов. Игровые компании выпускают более качественный игровой продукт, улучшают свои навыки в продвижении собственного продукта, что привлекает все больше пользователей. Выручка от реализации игровых продуктов по всему миру показывает стабильный рост, начиная с 2012 года, и по сей день продолжает расти. По прогнозам, доход от игровых продуктов в 2024 году составит около 200 миллиардов долларов.

Помимо прочего, можно утверждать, что наиболее динамично растущим сектором становится мобильный гейминг. Развитие смартфонов позволило разработчикам мобильных игр шагнуть на несколько шагов вперед и выпускать интересные продукты. В последнее время именно данное направление показывает наибольший рост на рынке игровой индустрии.

### **Постановка задачи**

Целью выпускной квалификационной работы является разработка компьютерной игры 2D Платформер на Unity. Для достижения поставленной цели необходимо решить следующие задачи:

- 1) провести обзор аналогов в данной области;
- 2) спроектировать игровое приложение;
- 3) реализовать и протестировать итоговое приложение.

### **Структура и содержание работы**

Работа состоит из введения, четырех глав, заключения и списка литературы. Объем работы составляет 31 страницу, объем списка литературы – 15 источников.

В первой главе «Анализ предметной области и обзор аналогов», описываются особенности игровых жанров, обоснование выбора данной

тематики и средств реализации, где будут расставлены акценты на преимущества схожих проектов и их недостатки по сравнению с конкурентами.

Во втором разделе «Анализ требований к программной системе» приведены функциональные и нефункциональные требования к системе, построена диаграмма вариантов использования и спецификация вариантов использования. Рассмотрена предметная область проекта, особенности жанра и их сочетаний.

В третьей главе «Архитектура системы» описывается общая архитектура системы и составляющие ее компоненты. Выполнены работы по созданию набросков игрового меню, меню паузы и пользовательского интерфейса.

В четвертой главе «Реализация и тестирование системы» представлено описание реализации игрового приложения, особенности составляющих его компонентов, их реализация, а также результаты функционального и юзабилити тестирования, по результатам тестирований будут сделаны выводы о достижении поставленных целей.

В приложении А содержатся таблицы, описывающие основные варианты использования (ВИ) разрабатываемого игрового приложения.

В приложении Б содержатся скриншоты финальной версии игры, демонстрирующие меню и игровой процесс.

# **1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

## **1.1. Предметная область проекта**

Платформер – это жанр видеоигр и поджанр экшн – игр, основной целью которых является перемещение персонажа игрока между точками в рендеринг – среде. Обычно данный процесс сопровождается перемещением по игровому уровню, прыжками по платформам, сбором предметов, набором очков или баллов, решением головоломок. Данный вид игр обычно обладает довольно простой рисованной графикой.

В то же время игры – платформеры очень обширное понятие и как таковых, однозначных правил в них нет. Бесспорно, присутствуют общие черты, но разнообразие правил и игровых механик поражает воображение. Для примера, одной из самых знаменательных игр в данном жанре стала «Super Mario Bros». В итоговом игровом приложении главной целью игрока будет, преодолев препятствия, победить босса.

Разрабатываемое игровое приложение будет сочетать в себе такие жанры как «Головоломка», «Казуальная игра», «2D – Платформер».

«Казуальная игра» – это игры с короткими и интенсивными игровыми сессиями, адресованные неискушенным игрокам, с очень красивыми, приятными визуальными эффектами, минимумом текста. Игры данного жанра легко дадутся пользователям практически любого возраста.

«Головоломка» – жанр компьютерных игр, целью которых является решение логических задач, требующих от игрока задействования логики, стратегии и интуиции.

## **1.2. Анализ аналогичных проектов и существующих решений**

В ходе выполнения выпускной квалификационной работы были изучены компьютерные игры в жанре 2D платформер. Проанализированы их сильные и слабые стороны, особенности и механики геймплея. Ниже приведен список, почему для анализа были выбраны именно данные игровые проекты:

- 1) игровой процесс проходит в 2D плоскости;
- 2) цель игрока пройти игровые локации до конца;
- 3) проекты стали наиболее популярными;
- 4) проекты появились относительно недавно;
- 5) для игрока есть препятствия или противники.

### **Ori and the Blind Forest**

Данный проект сильно полюбился геймерам и получил широкую огласку. Игровой процесс представлен на рисунке 1.



Рисунок 1 – Игра «Ori and the Blind Forest»

Причинами на то послужил увлекательный сюжет и потрясающее графическое оформление. По мере прохождения игры у героя появляются новые способности, которые приходится умело применять.

Сильные стороны игры:

- 1) разнообразие грамотно реализованных механик;
- 2) живописный мир;
- 3) простой, но сказочный сюжет;
- 4) исследование мира гармонично вписано в прохождение сюжета.

К слабой стороне игры относится то, что довольно часто игроку приходится быстро реагировать на действия противников и разные ловушки, а также резко управлять персонажем, что может быть весьма

утомительным.

### **Rayman Legends**

Игра, которая в отличие от своего предшественника серии «Rayman Origins» стала куда более любимой и популярной. Игровой процесс показан на рисунке 2.



Рисунок 2 – Игра «Rayman Legends»

Удалось ей это благодаря собственному уникальному стилю и очень интенсивным игровым процессом, в который можно погрузиться на некоторое время.

Игра очень умело комбинирует все возможные и известные механики, так что все уровни увлекательны и динамичны.

Сильные стороны:

- 1) отличный дизайн и красивая двумерная графика;
- 2) музыкальное сопровождение;
- 3) огромное количество дополнительных развлечений.

Для прохождения более высоких уровней потребуются не одна попытка, хорошая реакция и способность мгновенно адаптироваться к новым препятствиям и испытаниям, что может сильно повлиять на игру в кооперативе. Это может испортить впечатление пользователя об игре и перекроет все ранее полученные положительные эмоции и уникальность



данного продукта.

## **HOLLOW KNIGHT**

На первый взгляд, ничем не примечательная игра, которая стала популярной среди игроков. Игровой процесс изображен на рисунке 3.



Рисунок 3 – Игра «HOLLOW KNIGHT»

После знакомства с основными игровыми механиками, игрок оказывается в запутанной локации. Примечательно то, что те вещи, которые доступны в прочих платформерах изначально, будь то карта или точка сохранения, здесь придется буквально делать самому.

Сильные стороны:

- 1) огромный мир с непохожими друг на друга локациями;
- 2) множество типов врагов;
- 3) большое количество секретов, предметов и тайных проходов;
- 4) идея и сюжет;
- 5) отличный дизайн.

Сложность игровой карты, а также запутанные игровые локации доставляют дискомфорт и неудобство в игровом процессе. Что может неблагоприятно сказаться на конечном впечатлении пользователя об игре и перекроет ее уникальность и положительные эмоции, которые были ранее

получены.

### **1.3. Обоснование выбора средств реализации**

Для реализации будет использоваться один из популярнейших игровых движков Unity. Он обрел свою популярность при помощи впечатляющей мощности, которой достаточно для создания как 2D, так и 3D игр. Прочие именитые игровые движки все же больше ориентированы на 3D проекты. Кроме впечатляющей мощности, Unity обладает удобным и ориентированным на пользователя интерфейсом. Также к данному игровому движку огромное количество обучающих курсов, пособий, видео, сайтов, а также документация, которые упростят создание качественного проекта. Пользоваться им можно абсолютно бесплатно, при этом известные игровые проекты, о которых было изложено выше, а именно: «Ori and the Blind Forest» и «HOLLOW KNIGHT» созданы на Unity. Что говорит о том, что на данном игровом движке можно создавать хорошие 2D платформеры, пользующиеся популярностью на рынке.

#### **Вывод по первой главе**

В результате анализа предметной области были рассмотрены аналоги игровых продуктов, которые успешно закрепились на мировом рынке. Проведен анализ преимуществ и недостатков каждого из аналогов и взяты во внимания. А также выбран наиболее подходящий для проекта движок, на котором будет реализован 2D платформер.

## **2. АНАЛИЗ ТРЕБОВАНИЙ К ПРОЕКТИРУЕМОЙ СИСТЕМЕ**

### **2.1. Функциональные и нефункциональные требования**

Функциональные требования – это требования, которые определяют действия, которые должна выполнять система, без учета ограничений, связанных с ее реализацией, то есть определяют поведение системы в процессе обработки информации.

Функциональные требования к проектируемому игровому проекту:

- 1) система должна позволять управлять персонажем;
- 2) должна быть реализована механика взаимодействия персонажа с препятствиями;
- 3) должна быть реализована способность атаки у персонажа;
- 4) должна быть реализована система жизней персонажа;
- 5) должна быть реализована система появления персонажа при потере всех жизней.

Нефункциональные требования – требования, определяющие качественные характеристики проектируемого игрового приложения.

Нефункциональные требования к проектируемой системе:

- 1) система должна работать на операционной системе Windows 7 и новее;
- 2) система должна быть создана с использованием платформы Unity;
- 3) система должна быть написана на языке программирования C#.

### **2.2. Диаграмма вариантов использования**

Для проектирования системы был использован язык графического описания UML. В соответствии с требованиями была построена диаграмма вариантов использования (рисунок 4), отражающая модель взаимодействия актера «Игрок» с разрабатываемым игровым приложением. С системой взаимодействует только один основной актер – «Игрок», использующий игровое приложение. Запустив уровень, задача игрока – пройти его до конца.

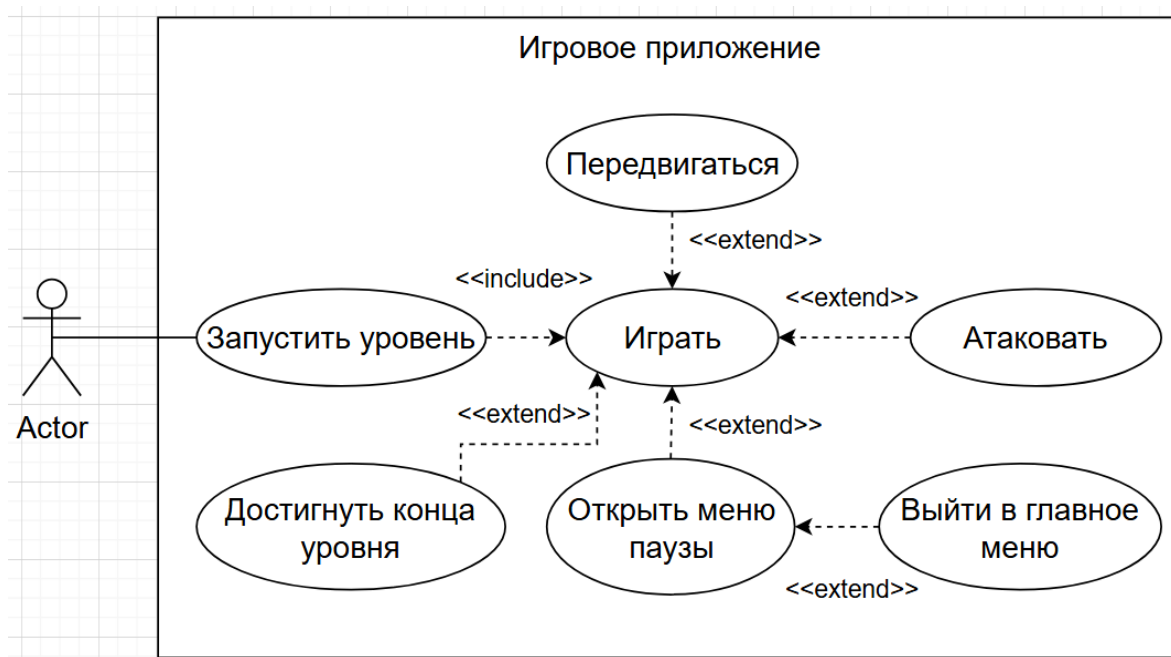


Рисунок 4 – Диаграмма вариантов использования

В ходе игры игрок может передвигаться, открыть меню паузы и выйти в главное меню.

Спецификации основных вариантов использования представлены в таблицах 1–5 приложения А.

### **Вывод по второй главе**

Во второй главе были определены функциональные требования к системе. На основе этих требований была построена диаграмма вариантов использования, определены основные актеры, взаимодействующие с системой, а также приведено краткое описание и спецификация вариантов использования.

### 3. АРХИТЕКТУРА СИСТЕМЫ

#### 3.1. Общее описание архитектуры системы

Архитектурно система состоит из сцен главного меню, меню паузы и игровых уровней.

При запуске игрового приложения первой сценой загружается «Главное меню». В главном меню присутствует пользовательский интерфейс, состоящий из кнопок «Играть», «Выйти» и слайдера громкости музыки. При нажатии кнопки «Играть», загружается сцена первого игрового уровня. При нажатии кнопки «Выйти» игровое приложение завершает работу. При использовании слайдера влево громкость музыкального сопровождения будет уменьшаться.

При этом любая сцена игрового уровня позволяет игроку перейти в игровое меню нажатием кнопки «Esc» на клавиатуре. В предложенном меню есть две кнопки «Продолжить», «Меню». Первая кнопка позволяет выйти из меню паузы и продолжить игру, вторая переносит игрока в сцену главного меню, игровой уровень закрывается.

Файловая система игры представлена ниже:

- 1) animations – содержит в себе анимации игровых объектов и их множественные параметры;
- 2) audio – содержит в себе аудиофайлы, используемые игровым приложением, а также микшеры звука, что позволяет регулировать громкость звуков или музыки;
- 3) levels – содержит в себе сцены главного меню и уровней;
- 4) material2D – содержит материалы и их свойства для поверхностей и объектов игры;
- 5) prefabs – содержит в себе готовые префабы противников, ловушек, персонажа, предметов;
- 6) scripts – содержит в себе скрипты, описывающие поведение и свойства игровых объектов;
- 7) sprites – содержит изображения (спрайты) игровых объектов.

### 3.2. Диаграмма компонентов

На рисунке 5 представлена диаграмма компонентов игрового приложения. Данные компоненты содержат в себе классы, описывающие поведение, свойства, параметры и зависимости игровых объектов.

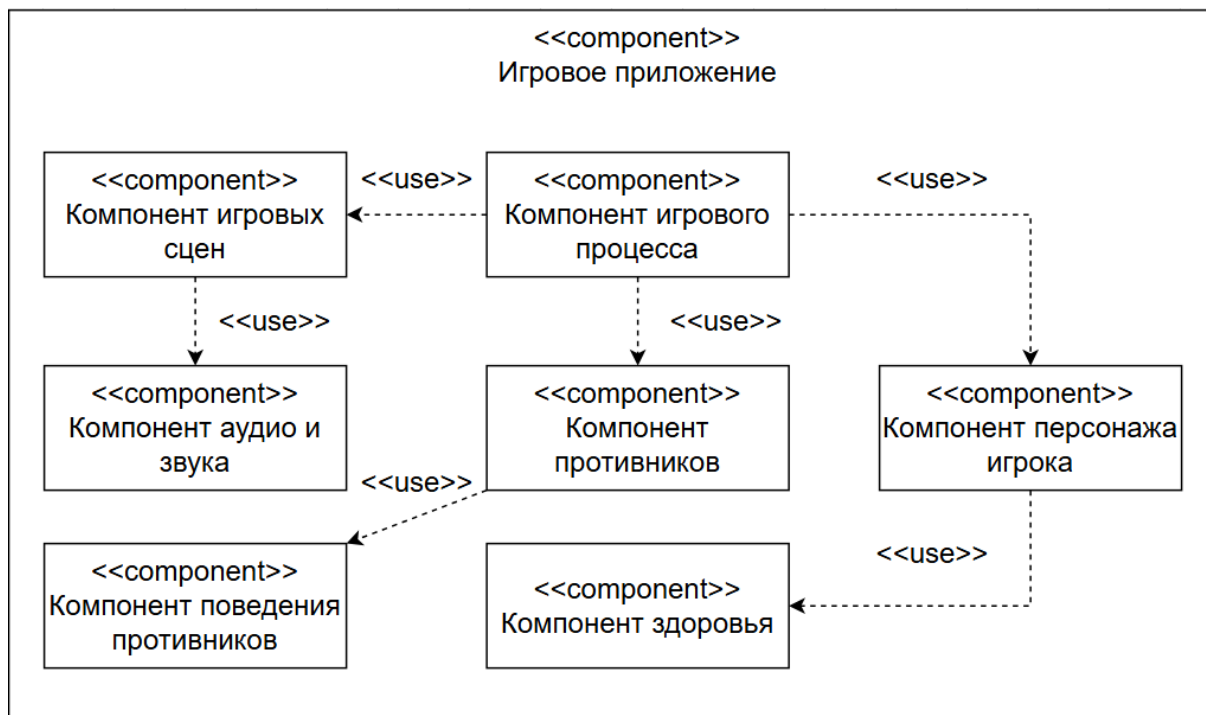


Рисунок 5 – Диаграмма компонентов

Описание компонентов, показанных на рисунке 5, представлено ниже:

- 1) компонент игровых сцен – содержит набор классов, реализующих главное меню, меню паузы, переключение между данными меню;
- 2) компонент игрового процесса – содержит набор классов, реализующих игровой процесс системы;
- 3) компонент аудио и звука – позволяет в главном меню изменять громкость звука и музыки;
- 4) компонент противников – содержит набор классов и параметров, хранящих данные о противниках, ловушках, определяющих свойства, логику и их поведение;
- 5) компонент персонажа игрока – содержит набор классов и параметров, хранящих и позволяющих настраивать главного персонажа, управляемого игроком;

б) компонент поведения противников – содержит набор классов, описывающих поведение противников;

7) компонент здоровья – содержит класс, описывающий и задающий параметры здоровья, а также вычитания здоровья при получении урона.

### 3.3. Представление пользовательского интерфейса

В данном разделе будут представлены макеты пользовательского интерфейса игрового приложения. Для расположения элементов пользовательского интерфейса игрового приложения выбиралось наиболее удобное расположение для пользователя и не нарушающее общепринятые стандарты игрового дизайна.

В главном меню игроку будут доступны и необходимы следующие элементы:

- 1) играть – запуск игры, запуск первого игрового уровня;
- 2) выход – завершить работу игрового приложения;
- 3) слайдер – изменить громкость музыки.

На рисунке 6 представлен макет главного меню игрового приложения.



Рисунок 6 – Макет главного меню

В меню паузы игроку будут доступны и необходимы следующие элементы пользовательского интерфейса:

- 1) продолжить – пауза прекращается, игрок возвращается к игре;
- 2) меню – возвращает игрока в главное меню.

На рисунке 7 представлен макет интерфейса меню паузы.

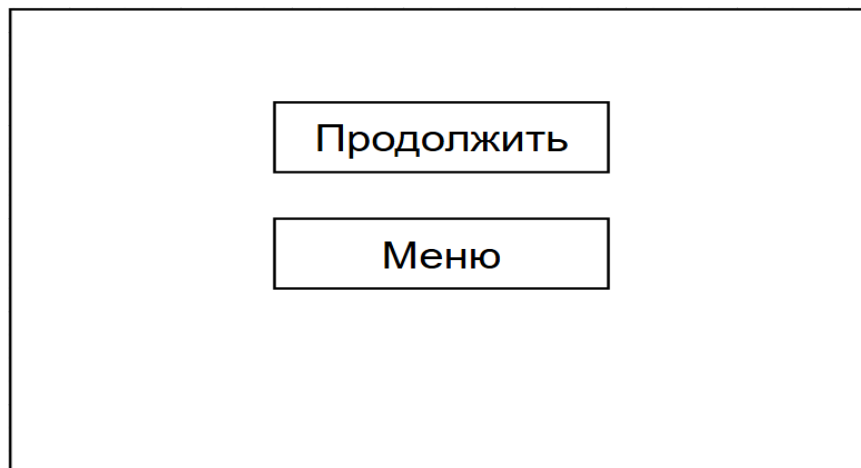


Рисунок 7 – Макет интерфейса меню паузы

Во время игрового процесса, когда игрок проходит игровые уровни, также есть элементы, которые необходимо отображать для игрока, например, количество здоровья. Таким образом игрок будет знать, сколько здоровья осталось у его персонажа, а также поможет игроку понимать, сколько вреда наносит ему то или иное препятствие или противник, чтобы осторожней обращаться с различными противниками и препятствиями. Расположение количества здоровья должно быть общепринятым и удобным для пользователя. Макет интерфейса игрового процесса отображен на рисунке 8.

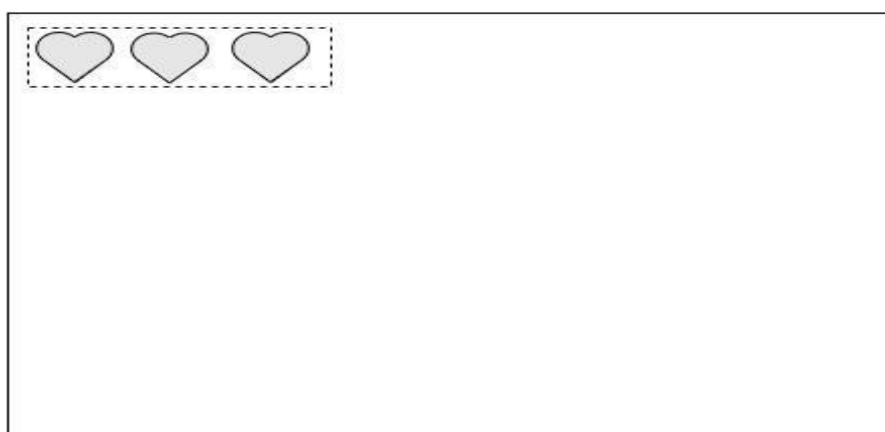


Рисунок 8 – Отображение здоровья персонажа во время игры

При победе над боссом игрок будет автоматически перенесен на сцену поздравления, после чего он сможет через меню паузы выйти в главное



меню.

### **Вывод по третьей главе**

В третьей главе была спроектирована архитектура приложения, а также приведено ее краткое описание. Для отображения динамического аспекта системы была построена диаграмма деятельности. Эта диаграмма демонстрирует поток действий пользователя и системы в различных сценах, таких как запуск игры, приостановка игры и выход из игры.

Для описания структурного аспекта системы была построена диаграмма компонентов. Эта диаграмма иллюстрирует различные компоненты приложения, их зависимости и взаимодействия. Она включает в себя компоненты игрового движка, управление графикой, управление аудио и пользовательского интерфейса.

Помимо архитектуры приложения, также были разработаны и представлены макеты пользовательского интерфейса для различных аспектов игры, таких как отображение количества здоровья у персонажа. Все макеты были оптимизированы для удобства использования и соответствия общепринятым стандартам игрового дизайна.

Описанная архитектура приложения и пользовательский интерфейс обеспечат надежную и удобную основу для дальнейшего развития и реализации игры.

## **4. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ СИСТЕМЫ**

Была выбрана платформа Unity для создания 2D-Платформера. Unity обладает всеми необходимыми инструментарием для создания 2D игр, а также поддержкой обширного круга платформ, подробной и официальной документацией.

Также для поиска графических элементов, спрайтов, персонажей и фонов использован официальный магазин Unity – «Unity Asset Store», в котором взяты бесплатные графические элементы, которые могут быть свободно использованы и распространены:

- 1) инструмент «Animation» – для анимации спрайтов, назначения и выполнения программного кода при воспроизведении анимации;
- 2) физика «Box2D» – работа с физикой для твердых 2D тел, определения столкновения между телами;
- 3) «physics materials 2D» – задание поведение и свойств материалам на основе физики;
- 4) инструмент «Animator» – для воспроизведения необходимой анимации в зависимости от действий пользователя, взаимодействие с анимацией через программный код.

### **4.1. Компоненты системы**

#### **Реализация компонента игровых сцен**

В игре реализовано два игровых меню, это меню паузы и главное меню, а также переход из главного меню в сцену игрового уровня и обратно. В главном меню игрок может осуществить выход из приложения, получить доступ к настройке громкости звука и музыки. Из главного меню игрок может начать игру, перейдя на первый игровой уровень. На рисунке 9 представлена диаграмма классов, отображающая реализацию главного меню и меню паузы.

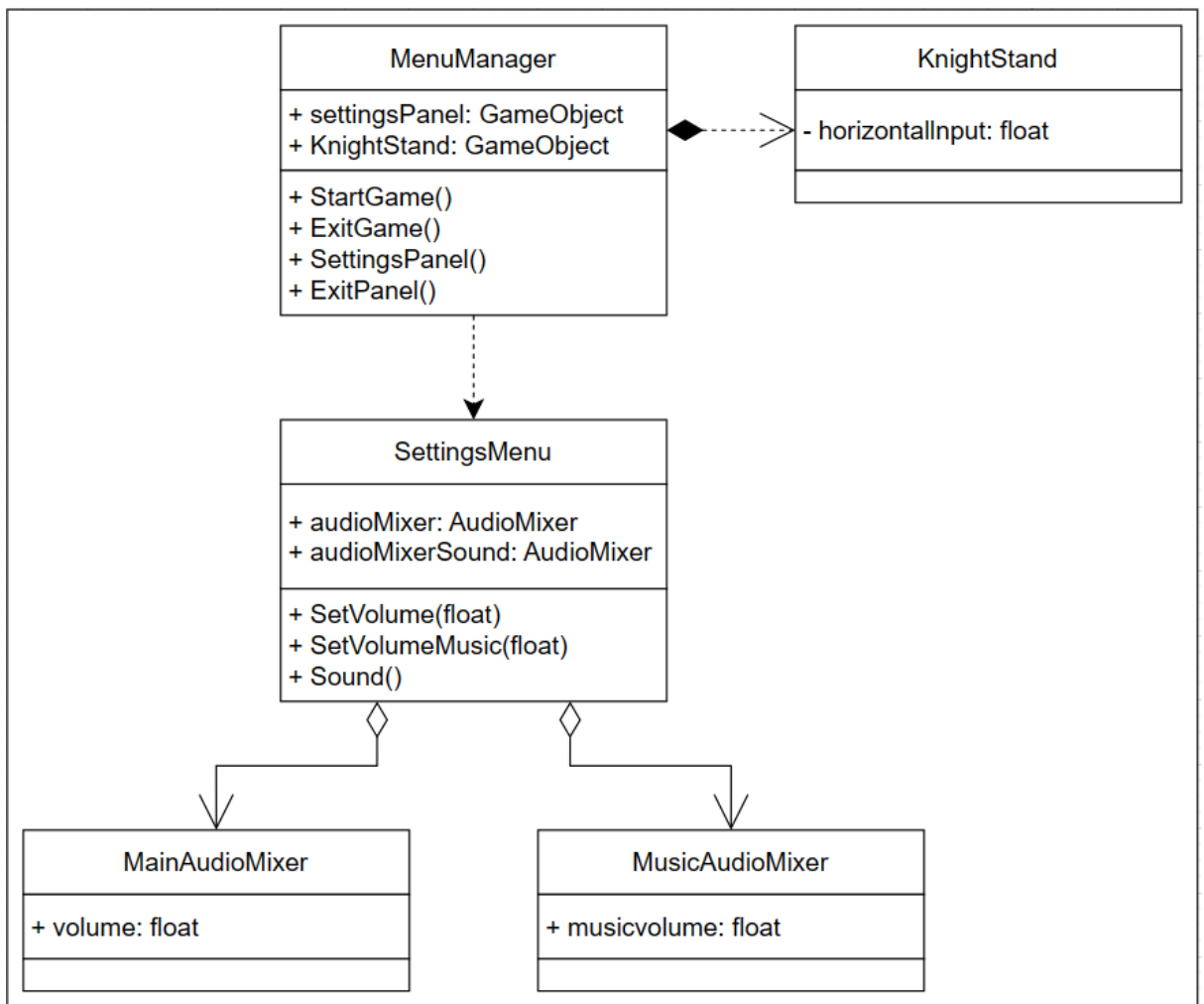


Рисунок 9 – Диаграмма классов «Главного меню» и «Меню паузы»

Основным классом является «MenuManager», благодаря которому и реализована функция запуска игрового уровня, выход из игрового приложения и переход в меню настроек.

Класс «MenuKnight» является частью класса «MenuManager» и не может быть использован без него, поскольку нарушается его функциональность.

Класс «SettingsMenu» используется для изменения параметров звуков, используя слайдер.

«MainAudioMixer» отвечает за звуки в игре, а «MusicAudioMixer» за музыкальное сопровождение.

## Реализация компонента аудио и звука

Для реализации воспроизведения аудио и звука был использован встроенный в Unity компонент «AudioSource». При этом было задействовано два компонента «AudioSource», один из которых отвечает за воспроизведение, а другой за регулировку громкости.

## Реализация компонента игрового процесса

В данном компоненте реализовано отображение пользовательского интерфейса (здоровье персонажа), а также логика и свойства предметов, с которыми может взаимодействовать персонаж.

Например, есть объекты, благодаря которым можно потерять определенное количество здоровья. Листинг класса Entity для данного объекта представлен на рисунке 10.

```
public class Entity: MonoBehaviour
{
    protected int lives;
    public virtual void GetDamage()
    {
        lives--;
        if (lives < 1)
            Die();
            SceneManager.LoadScene(3);
    }
}
```

Рисунок 10 – Листинг класса

Также присутствует разделитель между комнатами, который не дает противникам из одной комнаты передвинуться во вторую, для того чтобы у игрока была возможность таким образом спастись. На рисунке 11 представлен скрипт класса Portal, который не даст противнику пройти через портал.

```
public class Portal: MonoBehaviour
{
    private void OnTriggerEnter2D(Collider2D collision)
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex+1)
    }
}
```

Рисунок 11 – Листинг класса

## Реализация компонента ловушек

В данном компоненте реализована система классов для ловушек и их параметров. Классы ловушек, присутствующие в игровом приложении представлены ниже:

- 1) `spikes` – данный класс позволяет устанавливать ловушки, которые наносят урон игровому персонажу при соприкосновении;
- 2) `obstacle` – данный класс является основным для всех ловушек и обеспечивает нанесение урона игроку при соприкосновении.

На рисунке 12 представлен листинг класса `Obstacle`.

```
public class Obstacle: MonoBehaviour
{
    private void OnCollisionEnter2D(Collision2D collision)
    {
        if (collision.gameObject == Hero.Instance.gameObject)
        {
            Hero.Instance.GetDamage();
        }
    }
}
```

Рисунок 12 – Листинг класса

## Реализация компонента противников

Данный компонент состоит из класса босса и определяет его параметры, такие как скорость передвижения, наносимый урон, а также зона, в которой противник опознает игрока и наносит ему урон. Босс будет атаковать персонажа в небольшой зоне перед собой. Параметры босса описывает класс `Ninja`.

## Реализация компонента поведения противников

Данный компонент определяет поведение противника, передвижения по игровой локации. Этот компонент содержит в себе класс `EnemyPatrol`, который позволяет назначать границы патрулирования с помощью специальных меток на объектах. При столкновении с объектами, на которых находятся специальные метки, противник поменяет свое направление на противоположное.

## Реализация компонента здоровья

Данный компонент содержит в себе класс `Health`. Благодаря данному компоненту назначается значение единиц жизни игровым объектам, а также обеспечивается возможность понижать значение здоровья. Помимо этого, данный класс обеспечивает удаление объекта (уничтожение объекта) при потере всех жизней. На рисунке 13 представлен листинг функции, обеспечивающий удаление персонажа игрока при получении достаточного количества урона.

```
public override void GetDamage()
{
    Health -= 1;
    If (health == 0)
    {
        foreach(var h in hearts)
            h.sprite = deadHeart;
        Die();
        Exit();
    }
}
```

Рисунок 13 – Функция удаления объекта

## Реализация компонента персонажа игрока

Данный компонент содержит набор классов, описывающих поведение, свойства и характеристики игрового персонажа.

Класс `PlayerAttack` описывает параметры атаки игрока, ее скорости, а также значение наносимого урона.

Класс `PlayerRespawn` – сохраняет координаты контрольной точки, на которой в последующем возродится при потере всех жизней.

Класс `PlayerMovement` описывает параметры передвижения игрока, такие как скорость, сила прыжка, находится ли игрок в данный момент на земле. На рисунке 14 ниже представлен листинг функции, определяющей, находится ли игрок в данный момент на земле, что необходимо для того, чтобы знать, можно ли в данный момент выполнить прыжок или атаку.

```

private void CheckingGround()
{
    onGround = Physics2D.OverlapCircle(GroundCheck.position, check-
Radius, Ground);
}

```

Рисунок 14 – Функция определения нахождения игрока на земле

#### 4.2. Функциональное и юзабилити тестирования

В ходе данного тестирования проверялось соответствие игрового приложения предъявленным функциональным требованиям. В таблице 1 ниже представлены результаты проведенных тестирований. Данные функциональные тесты проверяют работу основных и базовых механик игры и ее логики, данные функции должны работать бесперебойно.

Таблица 1 – Функциональное тестирование игрового приложения

№	Название теста	Действия	Ожидаемый результат	Тест пройден?
1.	Работоспособность главного меню.	В главном меню выбрать пункт «Играть».	Запускается первый уровень.	Да
2.	Проверка управления нажатием кнопки «А».	Во время прохождения уровня нажать «А» на клавиатуре.	При нажатии клавиши персонаж движется влево.	Да
3.	Проверка управления нажатием кнопки «D».	Во время прохождения уровня нажать «D» на клавиатуре.	При нажатии клавиши персонаж движется вправо.	Да
4.	Проверка вызова меню паузы.	Во время прохождения уровня нажать клавишу «ESC» на клавиатуре.	Игра приостанавливается, появляется меню паузы.	Да
5.	Проверка работоспособности атаки.	Во время прохождения игрового уровня нажать левую кнопку мыши.	Персонаж атакует мечем.	Да
6.	Уничтожить противника мечем.	Лишить противника всех жизней.	Противник уничтожен.	Да
7.	Возрождение игрока.	Потерять все жизни.	Игрок переносится в главное меню.	Да
8.	Получение урона от атаки противника.	Подойти к противнику достаточно близко.	Игрок получает урон.	Да

На рисунках 1–4 приложения Б представлены скриншоты итогового игрового проекта.

Также было проведено юзабилити тестирование. Цель тестирования

заключалась в том, чтобы оценить удобства использования игрового приложения и выявить потенциальные проблемы, которые могут возникнуть у пользователей. В тестировании принимали участие шесть друзей с различным уровнем игры. Это позволило получить более полную информацию использования игрового приложения и выявить нюансы, которые могли остаться незамеченными при тестировании только одним типом пользователей. В ходе тестирования пользователям предлагалось запустить игру и победить босса.

В рамках тестирования было выявлено малое количества здоровья босса, что делало сражение с ним слишком коротким и неинтересным. В результате было принято решение увеличить количество здоровья босса до трех единиц, что позволило сделать сражение более увлекательным и затяжным. Такое изменение сделало игровой процесс более динамичным и позволило пользователям почувствовать реалистичность сражения.

#### **Вывод по четвертой главе**

В четвертой главе были приведены методы реализации разрабатываемой игры, описана реализация основных классов и объектов, которые отвечают за игровой процесс и взаимодействие с пользователем. Подробно описаны основные механизмы, отвечающие за функционирование игрового приложения. Кроме того, были представлены результаты тестирований, которые позволили внести необходимые корректировки в разработку.



## **ЗАКЛЮЧЕНИЕ**

В рамках данной работы была разработана компьютерная игра в жанре 2D-Платформер на игровом движке Unity.

Разработка игры была проведена в несколько этапов, каждый из которых был направлен на решение определенных задач.

Основные задачи, которые удалось решить:

1) проведен обзор аналогов в рамках предметной области и выявлены сильные и слабые стороны проектов, а также основные критерии успеха;

2) проведен анализ функциональных и нефункциональных требований, исходя из анализа требований был выбран игровой движок Unity, а также обоснован выбор именно данной платформы;

3) спроектирована архитектура игрового приложения, описаны компоненты, макеты меню и интерфейса;

4) создано игровое приложение.

В результате работы был извлечен опыт работы с игровым движком Unity. Процесс разработки позволил получить глубокие знания о возможностях платформы, изучить документацию и освоить инструменты для создания игр. Нарботки и опыт создания данного игрового приложения могут стать основой для создания будущих игровых проектов или улучшения текущего. Полученные знания и навыки позволят разрабатывать более сложные и интересные проекты, реализовывать новые идеи и создавать уникальный игровой опыт для пользователей.

## ЛИТЕРАТУРА

1. Unity User Manual. [Электронный ресурс] URL: <https://docs.unity3d.com/ru/530/Manual/Prefabs.html> (дата обращения: 25.01.2024 г.).
2. Unity User Manual. [Электронный ресурс] URL: <https://unity.com/ru/prefabs> (дата обращения: 25.01.2024 г.).
3. Unity User Manual. [Электронный ресурс] URL: <https://docs.unity3d.com/ru/530/Manual/Sprites.html> (дата обращения: 25.01.2024 г.).
4. Unity. [Электронный ресурс] URL: <https://unity.com/ru/pages/more-than-an-engine> (дата обращения: 25.01.2024 г.).
5. Gamersgate – Платформеры. [Электронный ресурс] URL: <https://gamersgate.ru/reviews/zhanr-kompyuternykh-igr-platfo/> (дата обращения: 25.01.2024 г.).
6. DTF – My.Games. [Электронный ресурс] URL: <https://dtf.ru/gameindustry/1012173-2024-god-dlya-mobilnogo-geyminga-i-netolko-chego-zhdai-k-chemu-gotovitsya> (дата обращения: 15.01.2024 г.).
7. Unity 2D Manual. [Электронный ресурс] URL: <https://docs.unity3d.com/Manual/Unity2D.html> (дата обращения: 10.01.2024 г.).
8. Unity 2D. [Электронный ресурс] URL: <https://unity.com/ru/solutions/2d> (дата обращения: 25.01.2024 г.).
9. Etudesite. [Электронный ресурс] URL: <http://www.etudesite.ru> (дата обращения: 10.02.2024 г.).
10. Unity Asset Store. [Электронный ресурс] URL: <https://assetstore.unity.com/> (дата обращения: 25.01.2024 г.).
11. Unity User Manual. [Электронный ресурс] URL: <https://docs.unity3d.com/Manual/2D-Platformer-Tutorial.html> (дата обращения: 25.01.2024 г.).

12. Unity User Manual. [Электронный ресурс] URL:  
<https://docs.unity3d.com/Manual/class-Rigidbody2D.html> (дата обращения:  
25.01.2024 г.).

13. Unity User Manual. [Электронный ресурс] URL:  
<https://docs.unity3d.com/Manual/class-Animation.html> (дата обращения:  
25.01.2024 г.).

14. Unity User Manual. [Электронный ресурс] URL:  
<https://docs.unity3d.com/Manual/Audio.html> (дата обращения: 25.01.2024 г.).

15. Unity User Manual. [Электронный ресурс] URL:  
<https://docs.unity3d.com/Manual/UnityOverview.html> (дата обращения:  
25.01.2024 г.).

## ПРИЛОЖЕНИЯ

### Приложение А. Спецификация вариантов использования

Спецификация вариантов использования (ВИ) системы приведена в таблицах 1–5.

Таблица 1 – Спецификация ВИ «Запустить уровень»

Прецедент: запустить уровень
ID: 1
Краткое описание: запуск уровня позволяет начать игру и проходить игровой уровень
Главные актеры: игрок
Второстепенные актеры: отсутствуют
Предусловия: 1. Успешный запуск игрового приложения. 2. Игрок должен находиться в главном меню.
Основной поток: 1. Вариант использования начинается, когда игрок нажимает кнопку «Играть». 2. Приложение загружает игровой уровень и начинается игра.
Постусловия: отсутствуют
Альтернативные потоки: отсутствуют

Таблица 2 – Спецификация ВИ «Играть»

Прецедент: играть
ID: 2
Краткое описание: управление персонажем и прохождение игровых уровней
Главные актеры: игрок
Второстепенные актеры: отсутствуют
Предусловия: 1. Успешный запуск игрового уровня
Основной поток: 1. Вариант использования начинается, когда завершается вариант использования 1. 2. Игрок управляет игровым персонажем.
Постусловия: отсутствуют
Альтернативные потоки: отсутствуют

Таблица 3 – Спецификация ВИ «Атаковать»

Прецедент: атаковать
ID: 3
Краткое описание: персонаж атакует мечем, который наносит урон противникам.
Главные актеры: игрок
Второстепенные актеры: отсутствуют
Предусловия: игрок проходит игровой уровень
Основной поток: 1. Вариант использования начинается, когда игрок нажимает левую кнопку мыши. 2. Игровой персонаж атакует мечем.
Постусловия: отсутствуют
Альтернативные потоки: отсутствуют

Таблица 4 – Спецификация ВИ «Достигнуть конца уровня»

Прецедент: достигнуть конца уровня
ID: 4
Краткое описание: игрок достигает конца уровня и переходит на следующий
Главные актеры: игрок
Второстепенные актеры: отсутствуют
Предусловия: отсутствуют
Основной поток: 1. Вариант использования начинается, когда игрок переходит в портал. 2. Загружается следующий уровень
Постусловия: отсутствуют
Альтернативные потоки: отсутствуют

Таблица 5 – Спецификация ВИ «Выйти в главное меню»

Прецедент: выйти в главное меню
ID: 5
Краткое описание: выход в главное меню игрового приложения
Главные актеры: игрок
Второстепенные актеры: отсутствуют
Предусловия: игрок находится в меню паузы
Основной поток: 1. Вариант использования начинается, когда игрок нажимает на кнопку возврата в главное меню. 2. Приложение открывает главное меню.
Постусловия: отсутствуют
Альтернативные потоки: отсутствуют

## ПРИЛОЖЕНИЕ Б. Скриншоты предварительной версии игры

На рисунках 1–4 представлены скриншоты предварительного проекта.

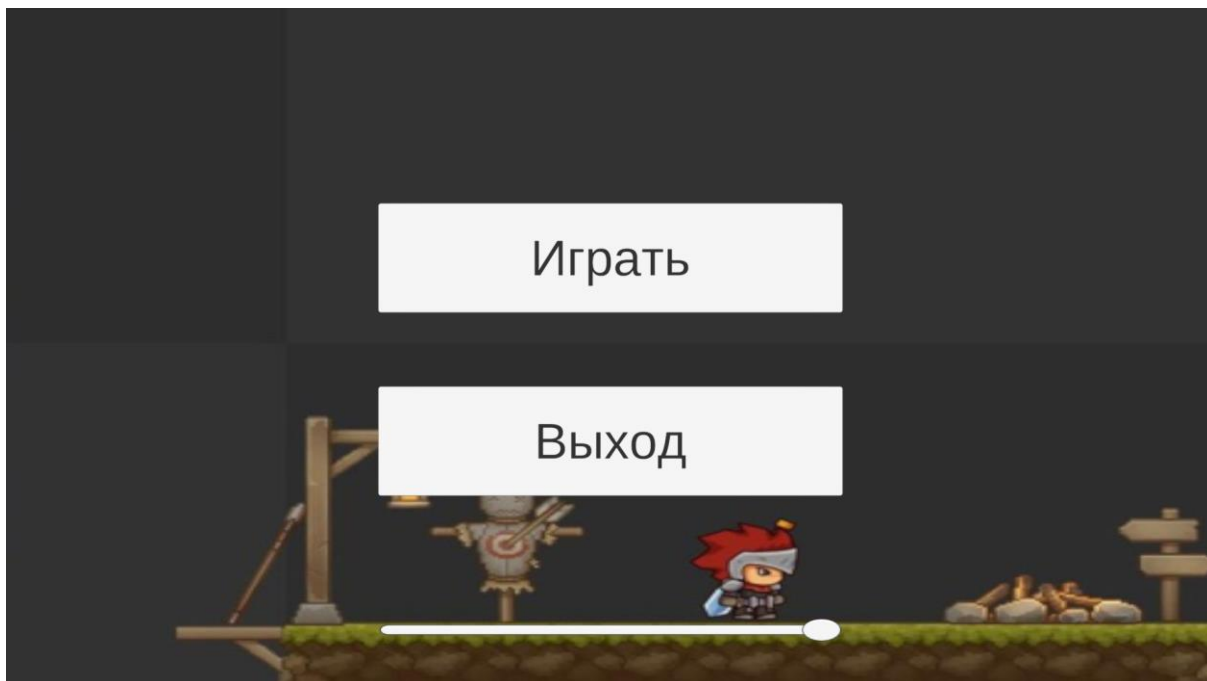


Рисунок 1 – Главное меню игры

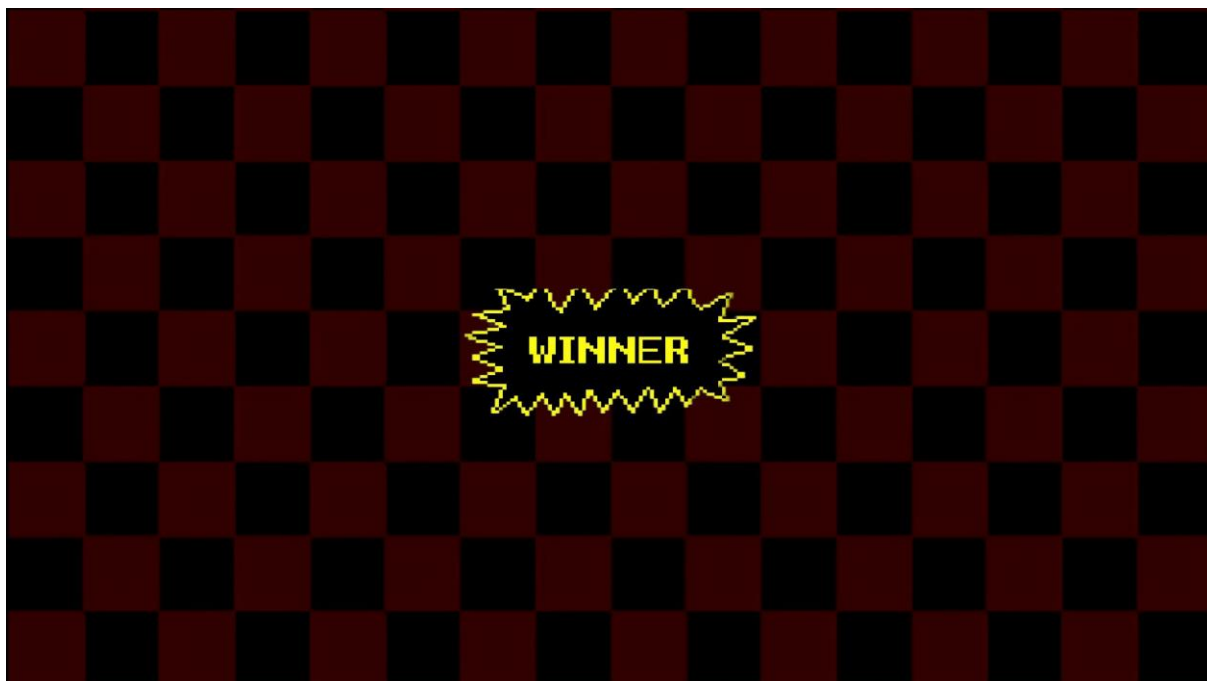


Рисунок 2 – Победа в игре

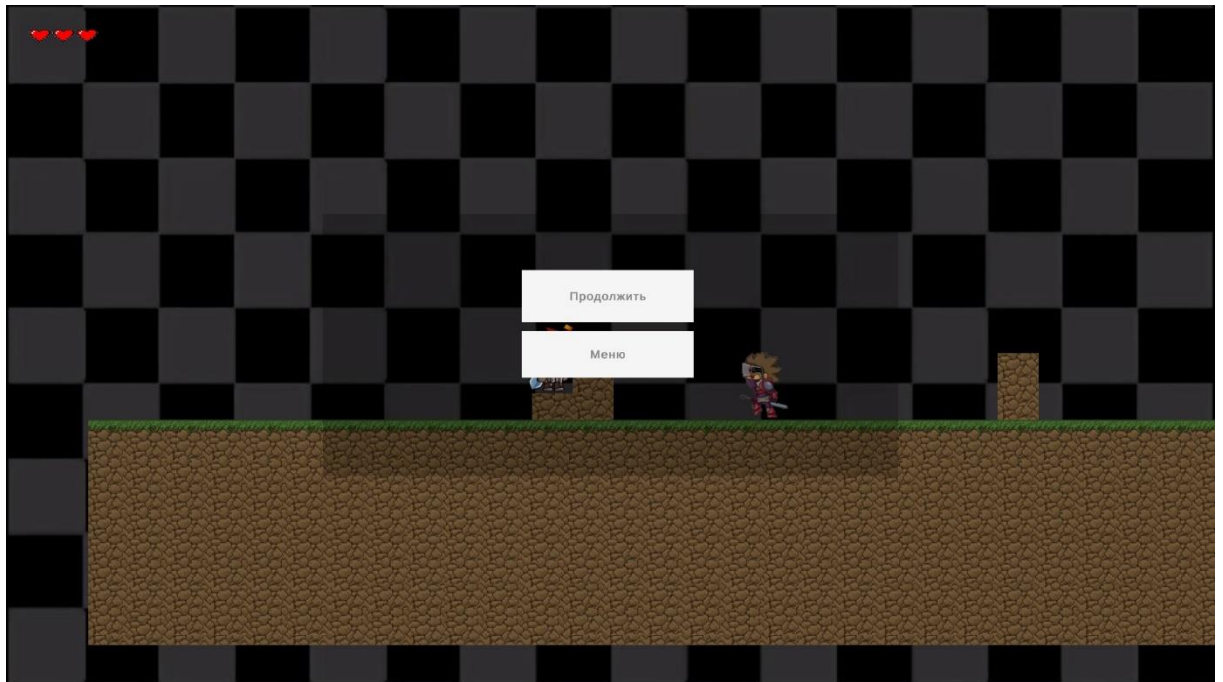


Рисунок 3 – Меню паузы



Рисунок 4 – Игровой процесс