**ЧАСТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ТЕХНИКУМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»**

**КУРСОВАЯ РАБОТА**

**ПМ.02** Разработка модулей программного обеспечения для компьютерных систем

**МДК.2.1** Разработка мобильных приложений

**ТЕМА:** Разработка игры «Monkey Jump» на Unity

**Выполнил:**

студент Гурэу Максим Сергеевич

курс IV группа 4П9

**Проверила:**

Преподаватель Алымова Ольга Леонидовна

**Работа защищена: «**\_\_\_\_**»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.**

**Оценка «\_**\_\_\_\_\_\_\_\_\_\_\_\_\_**»**

(прописью)

**\_\_\_\_\_\_\_\_\_\_\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О. преподавателя)

**г. Пятигорск**

**2023 г.**

**ЧАСТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ТЕХНИКУМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»**

**ЗАДАНИЕ**

на курсовую работу

**Студенту курса IV группы** 4П9 **специальности** 09.02.07

**Информационные системы и программирование**

**Фамилия** Гурэу

**Имя** Максиму **Отчество** Сергеевичу

**Тема** Разработка игры «Monkey Jump» на Unity

**Содержание пояснительной записки:** Понятия и цели создания игр на Unity; Основные понятия создания игр; Проведение работ по созданию игры;

**Дата выдачи задания** «14» сентября 2022 г.

**Срок представления работы к защите** «03» февраля 2023 г**.**

**Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**(дата и подпись студента)**

**СОДЕРЖАНИЕ**

# ВВЕДЕНИЕ

Сегодня мобильный и быстрый доступ к информации предпочитают все больше людей. С каждым годом процент пользователей мобильных версий сайтов растет, и, следовательно, компьютерные версии становятся все менее популярными. Статистика с сайта Cossa.ru показывает, что мобильным телефоном в России владеют 98% населения.

Множество программных продуктов в эру цифровых технологий, разрабатывается для мобильных телефонов. Большинство последних передовых моделей электронных устройств не уступают по функционалу и характеристикам громоздким мобильным телефонам и компьютерам.

Сейчас в мире практически не осталось людей, которые бы не пользовались мобильным устройством. Причинами их популярности можно назвать: функционал, доступ ко всем способам связи и, конечно же, небольшой размер устройства. Нужно заметить, что мобильные устройства практически заменили настольный компьютер в доме. Однако у этого есть и свои минусы, так как в связи с увеличением мобильной техники у населения появляются некоторые трудности, заключающиеся в отслеживании всех потоков информации на всех устройствах.

На сегодняшний день, основную классификацию мобильных телефонов делят на: BlackBerry, Symbian OS, Windows Mobile, Android, Mac OS и иные. ОС Android — является «одной из наиболее простых и одновременно комплексных платформ, вся система скачивается за один раз».

Игры – это отличный способ скоротать время в очередях, общественном транспорте или в школе на перемене.

В последнее время рынок мобильных игр растет с невероятной скоростью. Постоянно появляются новые жанры, либо совершенствуются уже имеющиеся и проверенные временем. На этом рынке часто становятся популярны игры с простой игровой механикой, которые не требуют от игрока больших познаний в игровой индустрии. В такие игры могут играть все, аудитория мобильных игр очень широка, в нее играют и дети, и взрослые.

Целью выполнения курсовой работы является разработка игры «Monkey Jump» на Unity.

Задачами курсовой работы являются:

* изучение особенностей конкретной предметной области, относящихся к теме курсового проекта;
* разработка мобильного приложения;
* анализ результатов работы мобильного приложения.

**ГЛАВА 1 АНАЛИТИЧЕСКИЙ РАЗДЕЛ**

**1.1 Мобильное приложение**

Мобильное приложение - является программным обеспечением, которое специально разработано для конкретной мобильной платформы (iOS, Android, Windows Phone и др). Предназначено для применения на смартфоне, планшете, умных часах и иных мобильных устройствах. Процесс создания мобильных приложений протекает следующим образом:

1) Написание на языке программирования высокого уровня;

2) Компиляция в машинный код операционной системы для максимальной производительности. При разработке приложений необходимы учитывать некоторые особенности: работа мобильных устройств осуществляется на батарейках и оснащены не такими мощными производительными процессорами, как у персональных компьютеров. Помимо этого, современные смартфоны и планшеты универсально имеют дополнительные устройства, как гироскопы, акселерометры и камеры, которые предоставляют уникальные возможности для расширения функциональности приложения. Как правило, продают мобильные устройства с некоторыми, заранее установленными приложениями.

Каждая организация имеет возможность выбора места размещения своей информации. Создание компьютерной-версии сайта является очевидным первым шагом, но не стоит останавливаться на этом. Разработка мобильного приложения, которое содержит всю информацию с компьютерной-версии сайта, приносит множество преимуществ, таких как возможность загрузить его один раз на мобильное устройство и использовать в офлайн режиме.

Использование приложений существенно избавляет от долгих ожиданий загрузки графики, изображения, звука и других компонентов, а также экономит время, поскольку программа уже установлена на телефон.

Разработчики приложений могут создавать мобильные приложения, которые доступны для каждой операционной системы телефона или планшета в отдельности. Сегодня мобильная игра имеет особое значение для компаний.

Компания может создать мобильную игру для различных целей:

* Привлечение новых клиентов и увеличение базы клиентов
* Увеличение прибыли и дохода от продаж игры и внутриигровых покупок
* Повышение бренда и увеличение распространения его имени
* Предложение удобного и доступного способа взаимодействия с клиентами и продвижения продуктов.

Мобильная игра может быть инструментом для достижения коммерческих и маркетинговых целей компании.

**1.2 Типы мобильных приложений**

Классификация мобильных приложений довольно сложна, поскольку многие из них являются «гибридами», то есть они предлагают пользовательские функции различных радаров. Но все же некоторые типы мобильных предложений можно категорировать однозначно. Среди пользователей мобильных приложений набрали свою популярность контент-приложения. На данный момент такие действия, как прослушивание музыки, просмотр фильмов, видеороликов, фотографий и чтение цифровых книг, настолько же доступны и удобны для каждого владельца мобильного устройства, так как это создает спрос на этот сегмент мобильных приложений.

Бизнес-приложение считается неотъемлемым инструментом для многих пользователей, который поможет им ускорить и оптимизировать работу в офисе. В настоящее время для инвесторов и бизнес-компаний, сегмент бизнес-приложений является наиболее удобным, но сложность заключается в переносе бизнес-задач на мобильные устройства.

Мобильные игры сегодня занимают значительную часть на рынке мобильных приложений. Разработчики создают новые игры или улучшают и обновляют уже выпущенные.

Мобильные социальные сети набирают все большую популярность с каждым днем, увеличивая аудиторию по всему миру. Сегодня огромное число людей пользуются социальными сетями, на что влияет еще одна тенденция, которая уже сформировалась: увеличение числа пользователей смартфонов. Можно сказать, что это самый популярный тип мобильных приложений, из-за того, что при покупке устройства там уже установлена одна или несколько мобильных программ по умолчанию.

**1.3 Обзор мобильных операционных систем**

На данный момент в мире имеется большой выбор языков программирования для того, чтобы разработать мобильные приложения. Это потому, что для разных мобильных устройств нужно применять разные языки программирования. Как правило, это зависит от того, что у мобильных устройств разные операционные системы (ОС). Я ознакомился с такими технологами, как Windows Phone SDK , iPhone SDK, Android SDK.

iPhone SDK

Для разработки под iPhone рекомендуется использовать ObjectiveC. Так же написание программы может производиться на С и на С++. Отладка приложения происходит при помощи среды хCode и эмулятора установленного в ней.

Windows Phone SDK

Последняя версия инструментария доступна в версии ReleaseCandidate Windows Phone SDK 7.1 с возможностью разработки приложений и публикации их в Windows Phone Marketplace. WindowsPhone SDK 7.1 ReleaseCandidate содержит следующие компоненты

Android SDK

Для разработки под операционную систему Android вы можете использовать среду Eclipse, которая не так давно была самой популярной средой разработки Android, ее используют многие опытные разработчики, или Android Studio - более новая среда разработки, но быстро завоевывающая популярность как более интеллектуальная и удобная. Стоит отметить, Android Studio активно развивается и поддерживается Google в качестве официальной среды разработки для Android-приложений. Основой среды Android Studio для разработки Android-приложений является проект российской компании IntelliJ IDEA.

Android-аудитория не только широкая и очень разнообразная. Закачать программы на Android могут люди разного достатка и возраста. Компании разработчики устройств Android делают свои гаджеты не только в сегменте дорогих моделей, но и предлагают свои бюджетные варианты. Разработка для OC Android имеет несколько особенностей.

Операционная среда Android за все время пережила ряд выпусков. Многие даже сейчас все еще используют устройства с версией 5.0, которая была выпущена в относительно далеком 2014 году, хотя в настоящий момент уже ведется разработка для выпуска 14 Q. Гаджеты Android характеризуются широким выбором разрешений дисплеев и физических размеров. Все производители стараются удивить своих пользователей новыми функциями, качеством камеры, мощностью батареи. Поэтому в продаже имеются устройства с размером дисплея от 4 до 10 дюймов, среди смартфонов можно найти 6-дюймовые устройства. Это еще одна особенность гаджетов Android. База пользователей Android очень большая, и, как результат, продукт, представленный в приложении, будет иметь широкую аудиторию.

После рассмотрения данных операционных систем для разработки игры я выбрал Android.

**1.4 Инструментарий разработки**

**1.4.1 Unity**

Unity – межплатформенная среда для разработки приложений трехмерных и двухмерных игр, поддерживающая свыше 25 платформ, таких как Windows, Android, OS X, IOS и т.д .

Редактор Unity имеет простой Drag&Drop интерфейс, который легко настраивать, состоящий из различных окон, благодаря чему можно производить отладку игры прямо в редакторе. Из языков программирования Unity позволяет выбрать между языками C# и JavaScript.

Любое приложение на Unity состоит из сцен, которые в свою очередь

содержат различные игровые объекты, обладающие набором поведенческих скриптов и других компонентов.

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.

Достоинства:

1) легкость использования;

2) качественная документация;

3) кроссплатформенность;

4) большое сообщество разработчиков;

5) простота освоения;

6) интегрированные сервисы для монетизации.

Недостатки:

1) закрытый исходный код;

2) ограниченный набор инструментов.  
**1.4.2 Язык программирования С#**

На сегодняшний момент язык программирования C# один из самых мощных, быстро развивающихся и востребованных языков в ИТ-отрасли. В настоящий момент на нем пишутся самые различные приложения: от небольших десктопных программок до крупных веб-порталов и веб-сервисов, обслуживающих ежедневно миллионы пользователей.

C# уже не молодой язык и как и вся платформа .NET уже прошел большой путь. Первая версия языка вышла вместе с релизом Microsoft Visual Studio .NET в феврале 2002 года. Текущей версией языка является версия C# 11, которая вышла 8 ноября 2022 года вместе с релизом .NET 7.

C# является языком с Си-подобным синтаксисом и близок в этом отношении к C++ и Java. Поэтому, если вы знакомы с одним из этих языков, то овладеть C# будет легче.

C# является объектно-ориентированным и в этом плане много перенял у Java и С++. Например, C# поддерживает полиморфизм, наследование, перегрузку операторов, статическую типизацию. Объектно-ориентированный подход позволяет решить задачи по построению крупных, но в тоже время гибких, масштабируемых и расширяемых приложений. И C# продолжает активно развиваться, и с каждой новой версией появляется все больше интересных функциональностей.

**1.4.3 Microsoft Visual Studio**

Microsoft Visual Studio - полнофункциональная интегрированная среда разработки (IDE) с поддержкой популярных языков программирования, среди которых С, C++, VB.NET, C#, F#, JavaScript, Python.

Функциональность Visual Studio охватывает все этапы разработки программного обеспечения, предоставляя современные инструменты для написания кода, проектирования графических интерфейсов, сборки, отладки и тестирования приложений. Возможности Visual Studio могут быть дополнены путем подключения необходимых расширений.

Редактор кода Visual Studio поддерживает подсветку синтаксиса, вставку фрагментов кода, отображение структуры и связанных функций. Существенно ускорить работу помогает технология IntelliSense - автозавершение кода по мере ввода.

Были выявлены преимущества мобильного приложения после анализа предметной области. Было принято решение использовать среду Unity и язык программирования C# после анализа многих похожих проектов и популярных сред для разработки. Проанализировав источники и аналогичные приложения был сделан вывод, что многие приложения не имеют четкой классификации и структуры. Я пришел к тому, что мобильное приложение улучшает восприятие информации с мобильного устройства, заметно сокращает время доступа к информации, имеет удобный интерфейс, раскрывая при этом актуальность создания мобильного приложения.

# ГЛАВА 2. РАЗРАБОТКА ИГРОВОГО ПРИЛОЖЕНИЯ

# 2.1 Требования к программной части приложения

Игра разработана с помощью интегрированной среды разработки JetBrains Rider и написан на языке C#

В качестве игрового движка был выбран Unity, т.к он больше подходит для разработки мобильных 2D-игр. Кроме того, он прост в освоении.

Требования к игре:

* Разработать игровую локацию
* Разработать главное меню
* Разработать переходы по сценам
* Разработать управление персонажем
* Разработать искусственного интеллекта противника
* Разработать взаимодействия противника с игроком

**2.2 Разработка интерфейса**

Игра состоит из двух сцен:

1) Главное меню;

2) Игровая локация.

Эскиз сцены главного меню представлен на рисунке 1, в этой сцене реализованы следующие функции:

1) Переход на игровую локацию;

2) Переход в меню;

3) Выход с игры;

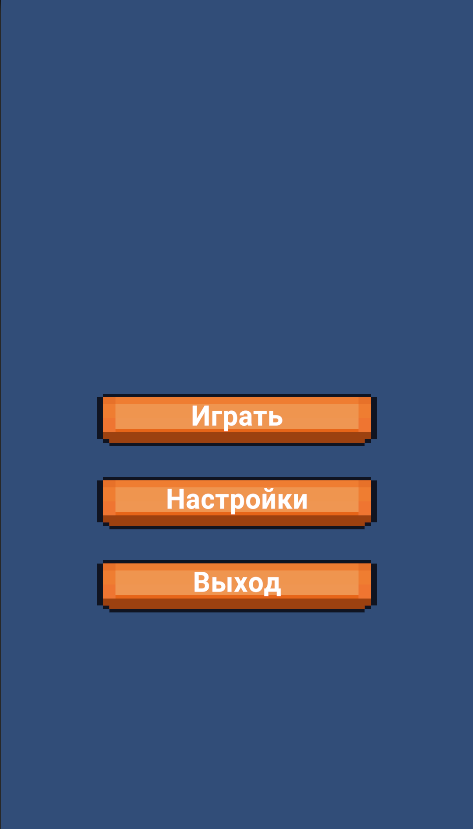


Рисунок 1 – Эскиз главного меню

Эскиз игровой сцены представлен на рисунке 2, в этой сцене реализованы следующие функции:

1) Взаимодействие пользователя с главным героем;

2) Взаимодействие противников с главным героем;

3) Отрисовка объектов на экране;

4) Подсчет поверженных противников;



Рисунок 2 - Эскиз игровой сцены

Так же, в игре должен быть реализовано окно и звуковой эффект, всплывающее в случае смерти персонажа и предлагающее вернуться в главное меню. Эскиз данного окна представлен на рисунке 3.



Рисунок 3 - Эскиз всплывающего окна

**2.3 Разработка дизайна игры**

Проанализировав рынок мобильных игр, решение пало на создание игры похожей на популярный хит Vampire Survivor, только в другой стилистике. Для поиска моделей я воспользовался встроенным магазином Unity - AssetStoreUnity. Используемые в проекте модели главного героя и его анимации представлены на рисунке 4



Рисунок 4 – Образ главного героя

Помимо образа персонажа, был разработан дизайн уровня:



Рисунок 5 – Зеленая поляна

Уровнь были спроектированы в Unity с помощью встроенного компонента Tilemap.

После разработки всех визуальных компонентов приложения, была создана картинка интерфейса и игрового процесса, которые показаны на рисунке 6.

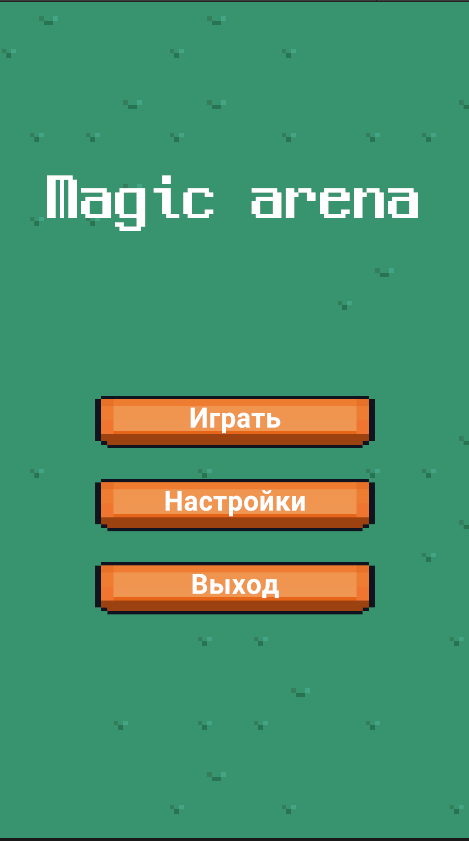


Рисунок 6 – Визуальная картинка интерфейса главного меню

Главное меню было создано в формате 2D. Все кнопки находятся на элементе Canvas.

Кнопка «Играть» перенаправляет на игровую сцену.

Кнопка «Документация» скрывает игровой объект игрового меню и открывает объект документации.

Кнопка «Выход» закрывает и выключает процессы приложения.

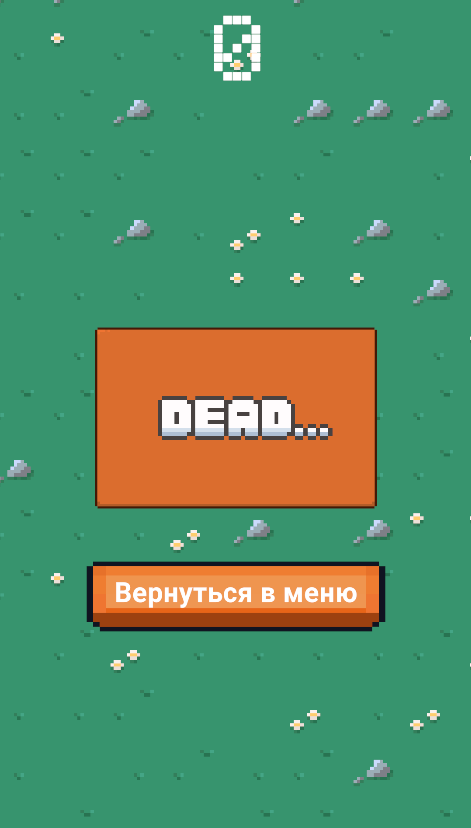


Рисунок 10 – Визуальная картинка всплывающего окна

Визуальная картинка всплывающего окна после смерти персонажа реализована с помощью объекта Canvas при котором после столкновения персонажа с противником скрытое окно «Lose Panel».

Кнопка «Вернуться меню» перемещает игрока в главное меню приложения.

**2.4 Разработка алгоритмов**

Для реализации бесконечного уровня была реализована генерация платформ с компонентом Collider, позволяющим обрабатывать столкновения с объектами. Платформы хранятся пуле объектов и в определенный момент помещаются на сцену или удаляются со сцены. Блок-схема работы данного алгоритма представлена на рисунке 11.

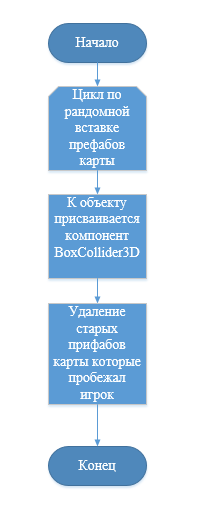


Рисунок 11 – Блок-схема алгоритма генерации платформ

Управление персонажем реализовано следующим образом. При старте сцены запускается таймер, по прошествии 3-х секунд персонаж начинает двигаться в определенном направлении. При нажатии на кнопку прыжка производится проверка положения персонаж, если он касается земли, то совершает прыжок. Блок-схема работы данного алгоритма представлена на рисунке 12.

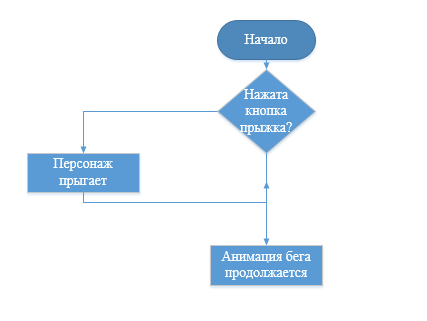


Рисунок 12 – Блок-схема алгоритма управления персонажа

**2.5 Разработка игровой сцены**

Для того, чтобы настроить анимацию в среде Unity, был использован инструмент Animator, встроенный в среду. Данный инструмент представлен на рисунке 13.

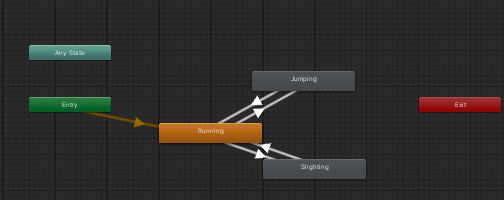


Рисунок 13 – Инструмент Animator

В данном инструменте для перехода между анимационными клипами используются параметры. Например, для перехода из анимации бега в анимацию прыжка Animator отслеживает состояние параметра Jumping. Если Running становится false, то происходит переход к анимации прыжка, когда становится Jumping true – переход из анимации прыжка к анимации бега.

В среде Unity объект, управляемый игроком – персонаж, состоит из родительского объекта Player и дочерних ему объектов Char и Punch.

Управление осуществляется над объектом Player. Следовательно, этому объекту добавлены следующие компоненты:

1) Rigidbody 2D. Придает объекту физические свойства.

2) CapsuleCollider 2D. Компонент, способный отслеживать столкновение и пересечение с другими объектами сцены.

3) PlayerControler. Скрипт, дающий пользователю возможность управления персонажем. Листинг скрипта представлен в [приложении 1](#_heading=h.1fob9te).

Полный список компонентов объекта Player и их параметры показан на рисунке 14.

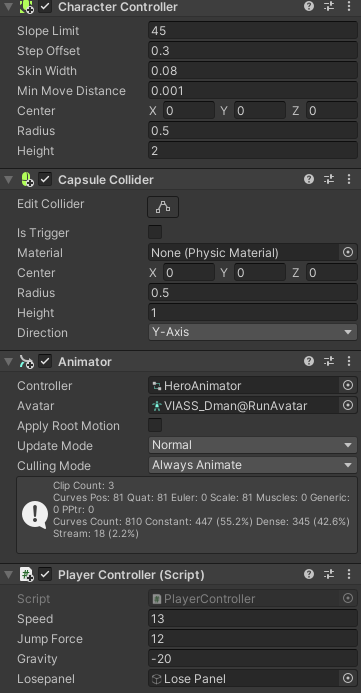


Рисунок 14 – Компоненты объекта Player

Объект Char используется для проигрывания анимации. Имеет следующие компоненты:

1) Sprite Renderer;

2) Animator. Механизм воспроизведения анимации.

**2.6 Разработка меню и всплывающих окон**

Главное меню состоит из текста с названием игры и двух кнопок:

1) Кнопка старта. Запускает игровую сцену.

2) Кнопка выхода в главное меню.

Все элементы сцены главного меню управляются скриптом MenuScript.cs, листинг которого представлен в [приложении 2](#_heading=h.3znysh7).

Всплывающее окно, оповещающее о смерти главного героя, состоит из текстового счетчика очков и двух кнопок, одна из которых отвечает выполняет функцию возврата в главное меню, а другая выполняет перезапуск игровой сцены. Всплывающее окно представлено на рисунке 15.

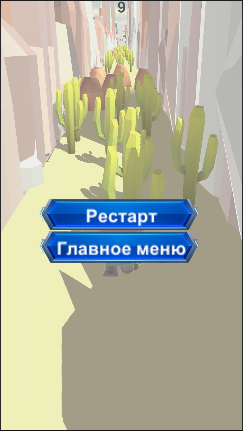


Рисунок 15 – Всплывающее окно

**2.7 Тестирование**

Тестирование продукта включает в себя этап модульного тестирования:

1) Модульное тестирование

Данный вид тестирования представляет собой проверку отдельных модулей продукта.

По прохождении этапа модульного тестирования можно утверждать, что каждый модуль работает правильно.

Необходимость модульного тестирования обусловлена тем, что для корректной работы всего приложения в целом нужно сначала провести проверку каждого модуля отдельно, что ускорит отладку компонентов в случае, если были допущены ошибки, а так же уменьшит количество ошибок при функциональном тестировании.

Тестирование каждой сцены проводится отдельно, так как они независимы друг от друга.

Функциональное тестирование приведено в таблице 1.

Таблица 1 – Функциональное тестирование

| Функция | Результат |
| --- | --- |
| Бег | При движении персонаж двигается в одном направлении с увеличивающейся скоростью. |
| Прыжок | При нажатии на кнопку, персонаж прыгает, с условием, если он находится на платформе. |
| Взаимодействие с платформой | Персонаж и платформа имеют коллизию, в связи с этим они не проходят через друг друга. |
| Взаимодействие с объектом DieSpace | Если персонаж касается объекта, то выпадает меню смерти персонажа. |
| Переход на сцену главного меню | Кнопка возврата в главное меню выполняет свою функцию |
| Переход между экранами главного меню | Кнопки главного меню выполняют свою функцию |
| Переход на игровую сцену | Кнопка перехода на игровую сцену выполняет свою функцию |
| Генерация объектов | Объекты генерируются на протяжении всего сеанса игры на рабочей области |
| Уничтожение объектов | Объекты уничтожаются за пределами видимости пользователя. |