|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.03 Прикладная информатика**

**Отчет**

|  |
| --- |
| **по проектной работе** |

**Дисциплина:** Разработка приложений на языке С#

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-74Б |  |  | Голчина О.С. |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  | Али Мусса Адум |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |
| Преподаватель |  |  |  | Минитаева А.М. |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

# СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc120732649)

[1 Анализ требований и уточнение спецификаций 5](#_Toc120732650)

[1.1 Анализ задания и выбор технологии, языка и среды разработки 5](#_Toc120732651)

[1.2 Разработка диаграммы вариантов использования 5](#_Toc120732652)

[1.3 Разработка концептуальной модели предметной области 8](#_Toc120732653)

[2 Проектирование структуры и компонентов программного продукта 9](#_Toc120732654)

[2.1 Разработка интерфейса пользователя 9](#_Toc120732655)

[2.1.1 Средства разработки интерфейса 9](#_Toc120732656)

[2.1.2 Построение графа состояний интерфейса 9](#_Toc120732657)

[2.1.3 Формы интерфейса 11](#_Toc120732658)

[2.2 Разработка диаграмм классов 14](#_Toc120732659)

[3 Тестирование программного продукта 22](#_Toc120732660)

[ЗАКЛЮЧЕНИЕ 26](#_Toc120732661)

**ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ**

ENC– настольное приложение «ENCOURAGERS»;

ЛКМ – левая кнопка мыши;

Префаб - класс в редакторе;

# ВВЕДЕНИЕ

ENC является простым и гибким настольным приложениям по сравнению с аналогами. Уникальность данного приложения заключается в комбинировании механик головоломки и интерактивного сражения. Актуальность разработки обусловлена созданием более доступной игровой системы. Такую игровую систему сможет опробовать каждый желающий без дополнительных знаний и руководств.

Цель работы – создание настольного приложения, интерактивной системы в реальном времени, которая является видеоигрой с видом сверху, позволяющей:

* выбрать одного из двух предложенных игровых персонажей,
* управлять одним из двух предложенных игровых персонажей, а конкретно:
  + управлять перемещением персонажа
  + с помощью физического движка влиять на предметы окружения
  + использовать боевую систему для сражения с противниками
    - создавать боевые снаряды и импульс снаряда (в зависимости от персонажа)
    - использовать способности персонажей
  + взаимодействовать с используемыми предметами,
  + взаимодействовать со статичными предметами окружения и локацией,

# Анализ требований и уточнение спецификаций

## Анализ задания и выбор технологии, языка и среды разработки

При выборе технологии предпочтение было отдано объектно- ориентированному подходу, поскольку с помощью данного подхода удобнее моделировать игровой мир и данный подход обеспечивает максимальную эффективность разработки.

В качестве технологии разработки был выбран движок Unity, так как это простой и удобный инструмент для создания игр в двумерном или трехмерном пространстве. Важным фактором при выборе движка являлось сообщество Unity, которое активно продвигает, поддерживает технологию и способствует обучению новых разработчиков. Аналоги движка были рассмотрены, но не подошли, так как представляли более специфичный функционал. Unreal Engine, например, позволяет работать только с 3D графикой. Godot engine и остальные 2D движки больше подходят для разработки небольших систем, в которых не делается упор на экономию ресурсов и эффективность приложения. Основным языком в Unity является C#. Язык является полностью объектно-ориентированным, имеет С-подобный синтаксис и является крайне эффективным при проектировании систем разной степени нагруженности.

В качестве среды разработки использовался Unity Editor, являющийся частью системы движка, в комбинации с текстовым редактором Visual Studio Code. Среда Unity Editor является полнофункциональной средой разработки интерактивных систем к которой прилагается объемная документация на сайте разработчика[6]. С помощью данной среды можно редактировать все сущности проекта, создавать и обновлять все файлы проекта, запускать и отлаживать собранную версию проекта.

## Разработка диаграммы вариантов использования

Диаграмма вариантов использования ENC приведена на рисунке 1. На данной диаграмме отображены основные функции, которые доступны пользователям.

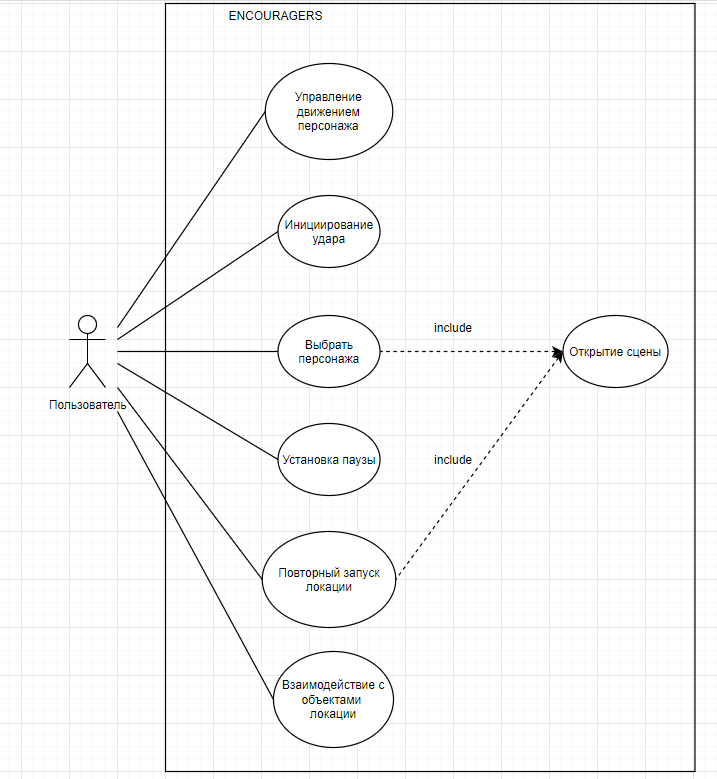


Рисунок 1 – Диаграмма вариантов использования

Краткое описание вариантов использования представлено в таблице 1.

Таблица 1 – Краткое описание вариантов использования

|  |  |
| --- | --- |
| Название варианта  Цель  Действующие лица  Краткое описание  Тип варианта | ***Управление движением персонажа***  *Управлять перемещением персонажа*  *Пользователь*  *Пользователь нажимает на клавиши «w,a,s,d» для управления перемещением персонажа*  *Основной* |

Продолжение таблицы 1

|  |  |
| --- | --- |
| Название варианта  Цель  Действующие лица  Краткое описание  Тип варианта | ***Инициирование удара***  *Нанести урон*  *Пользователь*  *Пользователь нажимает на левую кнопку мыши для нанесения урона*  *Основной* |
| Название варианта  Цель  Действующие лица  Краткое описание  Тип варианта | ***Меню выбора персонажа***  *Выбрать игрового персонажа*  *Пользователь*  *Пользователь может выбрать предпочитаемого персонажа*  *Основной* |
| Название варианта  Цель  Действующие лица  Краткое описание  Тип варианта | ***Установка паузы***  *Поставить приложения на паузу*  *Пользователь*  *Пользователь может поставить игру на паузу*  *Основной* |
| Название варианта  Цель  Действующие лица  Краткое описание  Тип варианта | ***Перезапуск локации***  *Перезапустить локацию после поражения*  *Пользователь*  *Пользователь перезапускает локацию после поражения*  *Основной* |

## Разработка концептуальной модели предметной области

В настоящем приложении цель основного варианта использования –достижение поставленных игроком и игрой целей.

В рассматриваемой предметной области можно выделить основные классы-понятия:

* пользователь;
* персонаж;
* противник;
* интерактивный предмет;
* интерактивное окружение;
* пассивный предмет окружения;
* локация

В соответствии с описанием предметной области составлена концептуальная диаграмма классов, представленная на рисунке 2.



Рисунок 2 – Концептуальная диаграмма классов

# Проектирование структуры и компонентов программного продукта

## Разработка интерфейса пользователя

### Средства разработки интерфейса

Создание правильно построенной визуальной части интерфейса для пользователя является достаточно трудоемкой задачей, при которой необходимо учитывать психофизические особенности человека, связанные с запоминанием и обработкой информации. При проектировании интерфейса необходимо учитывать такие особенности пользователя как: уровень подготовки в предметной области разрабатываемого программного обеспечения, уровень подготовки в области владения устройством, интуитивными моделями выполнения операций в этой предметной области [1]. Интерфейс ENC выполнен при помощи редактора сцен Unity [4] и функционала элемента Canvas [5], которые встроены в среду разработки Unity Editor. Редактор упрощает разработку пользовательского интерфейса, так как для его создания не нужно писать программный код. Также благодаря режиму предварительного просмотра можно быстро посмотреть, как выглядит интерфейс на различных устройствах без необходимости запускать приложение, что значительно ускоряет процесс проектирования. В игровых приложениях пользователи привыкли пользоваться графическим пользовательским интерфейсом, поэтому было принято решение создавать именно такой интерфейс.

### Построение графа состояний интерфейса

На основе анализа основных функций приложения и вариантов использования был построен граф состояний интерфейса, который представлен на рисунке 3.

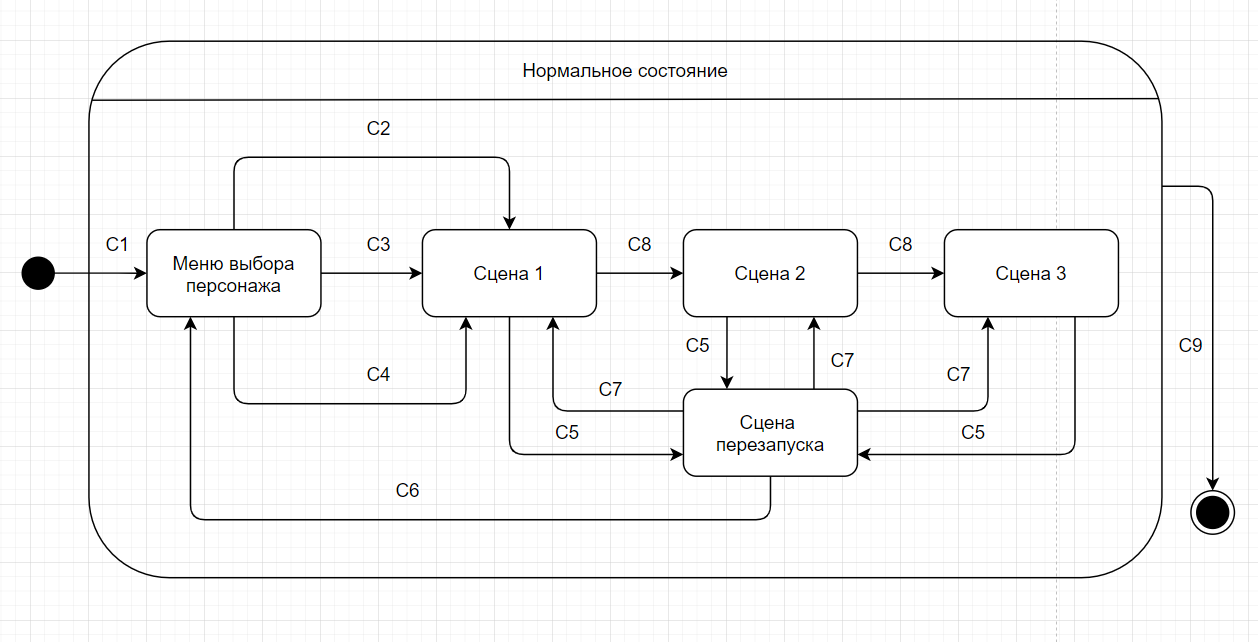


Рисунок 3 – Граф состояний интерфейса

На рисунке 3 обозначены следующие переходы:

* С1 – Запуск приложения;
* С2 – Выбор персонажа 1 путем нажатия ЛКМ;
* С3 – Выбор персонажа 2 путем нажатия ЛКМ;
* С4 – Выбор случайного персонажа путем нажатия клавиши «Space»;
* С5 – Смерть персонажа;
* С6 – Нажатие клавиши «Escape»
* С7 – Нажатие клавиши «R»
* С8 – Подобрать необходимый предмет;
* С9 – Выход из приложения

### Формы интерфейса

На рисунке 4 представлено меню выбора персонажа:



Рисунок 4 – Меню выбора персонажа

На рисунке 5 представлена сцена первая:

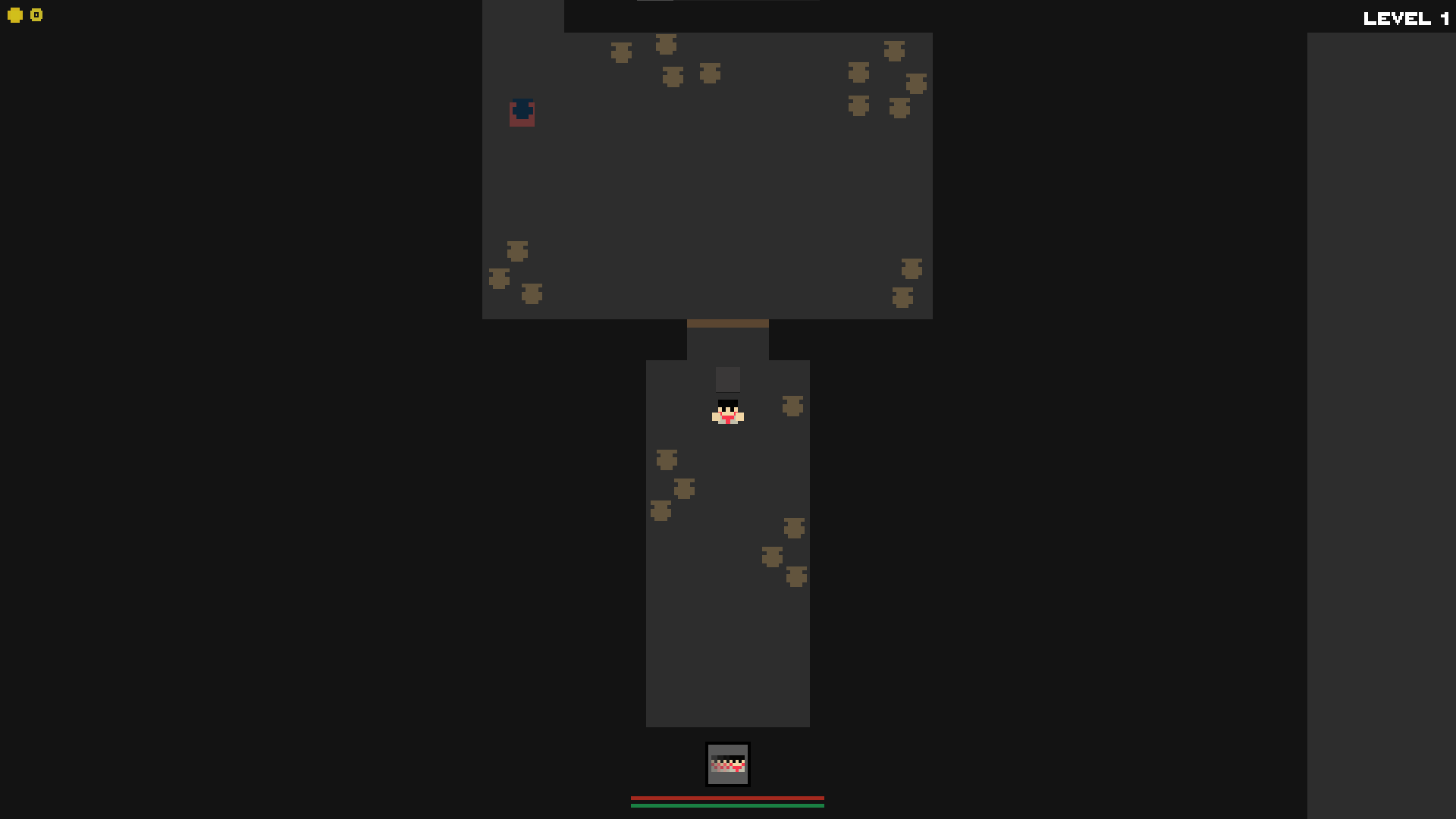


Рисунок 5 – Сцена первая

На рисунке 6 представлена сцена вторая:



Рисунок 6 – Сцена вторая

На рисунке 7 представлена сцена третья:



Рисунок 7 – Сцена третья

На рисунке 8 представлена сцена перезапуска уровня:

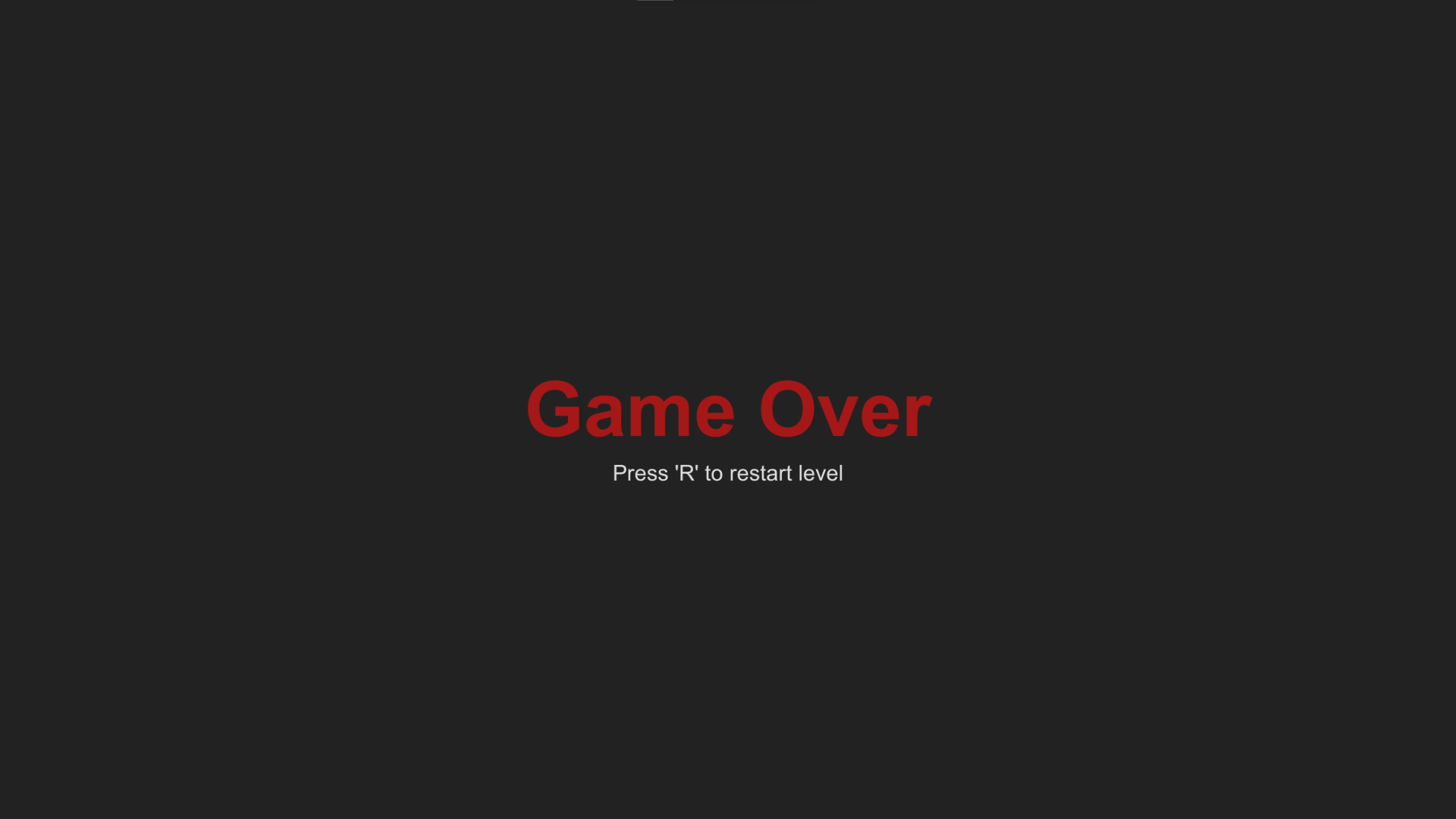


Рисунок 8 – Сцена перезапуска уровня

## Разработка диаграмм классов

На рисунке 9 представлена диаграмма классов внутреннего устройства движка, которая используется в приложении

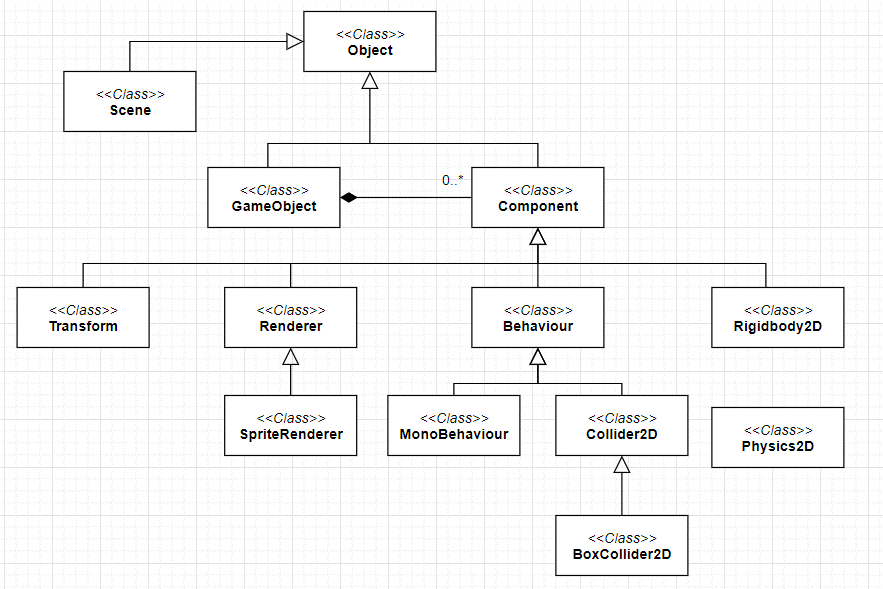


Рисунок 9 –Диаграмма классов внутреннего устройства движка, которые используются в приложении

Ниже описано назначение каждого из классов:

1. UnityEngine.CoreModule – модуль Ядра реализует базовые классы, необходимые для работы Unity.
2. UnityEngine.Physics2DModule – модуль Physics2d реализует 2D-физику в Unity
3. Object – базовый класс для всех объектов, на которые может ссылаться Unity.
4. Scene – класс для размещения любых сущностей в Unity.
5. GameObject – базовый класс для всех сущностей в сценах Unity.
6. Component – базовый класс для всего, что прикреплено к GameObject.
7. Behaviour – это компонент, который можно включать или отключать.
8. MonoBehaviour – базовый класс для задания поведения любому объекту. Это базовый класс, от которого наследуется каждый скрипт Unity.
9. Renderer – это то, что заставляет объект появляться на экране. Отвечает за визуализацию объекта на экране.
10. SpriteRenderer –визуализирует спрайт для 2D-графики.
11. Collider2D – родительский класс для типов коллайдеров, используемых в 2D-игре.
12. BoxCollider2D – коллайдер для 2D-физики, представляющий выровненный по оси прямоугольник.
13. Physics2D – глобальные настройки и помощники для 2D-физики.
14. Rigidbody2D – компонент физики твердого тела для 2D-спрайтов. Добавление компонента Rigidbody2D к спрайту переводит его под контроль физического движка. Само по себе это означает, что на спрайт будет действовать сила тяжести, и им можно будет управлять из скриптов с помощью сил. Добавляя соответствующий компонент коллайдера, спрайт также будет реагировать на столкновения с другими спрайтами. Такое поведение полностью исходит из физической системы Unity.

На рисунке 10 изображена диаграмма классов «живых объектов», на рисунке 11 представлена диаграмма классов предметов окружения.

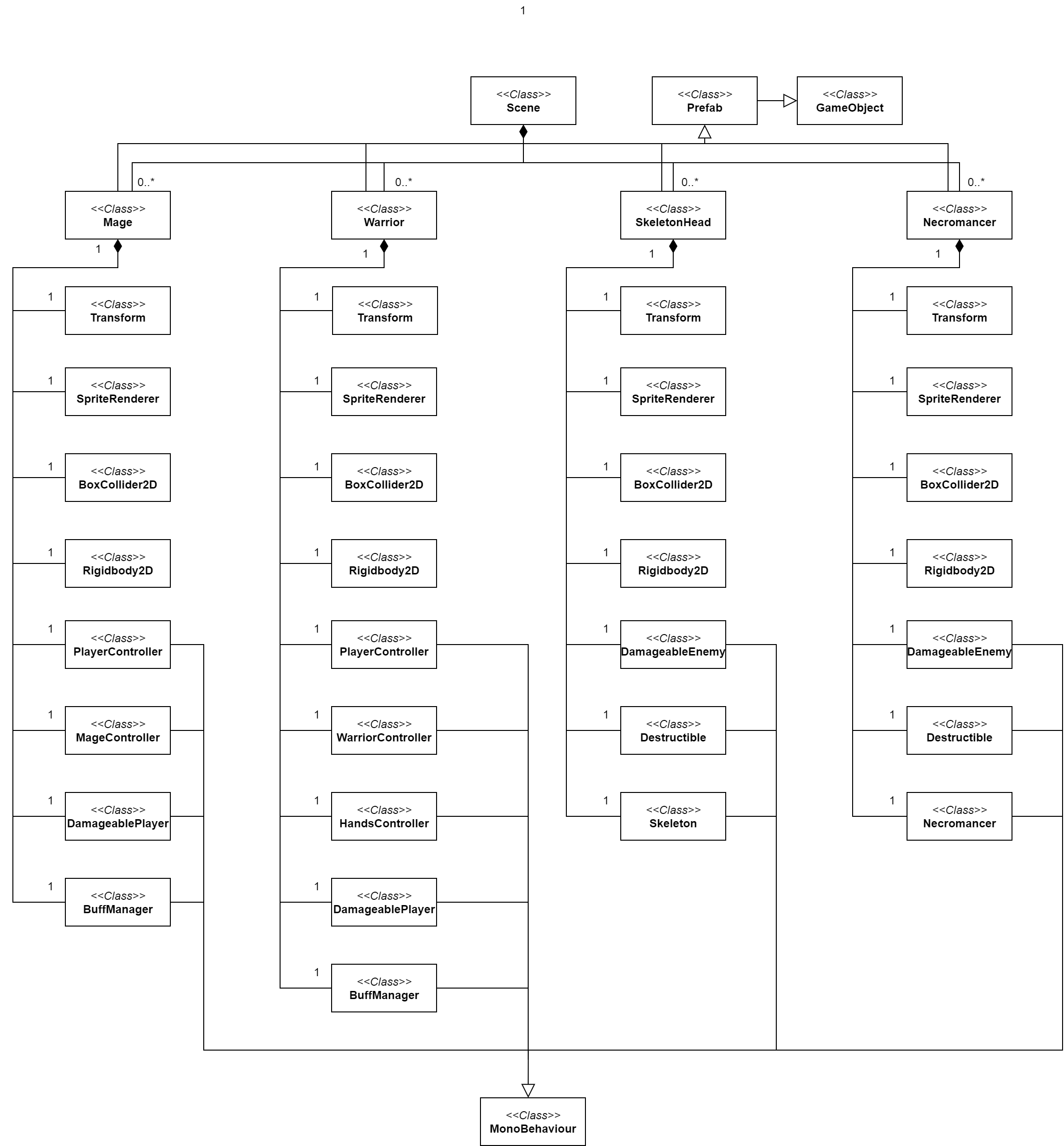


Рисунок 10 – Диаграмма классов «живых объектов»

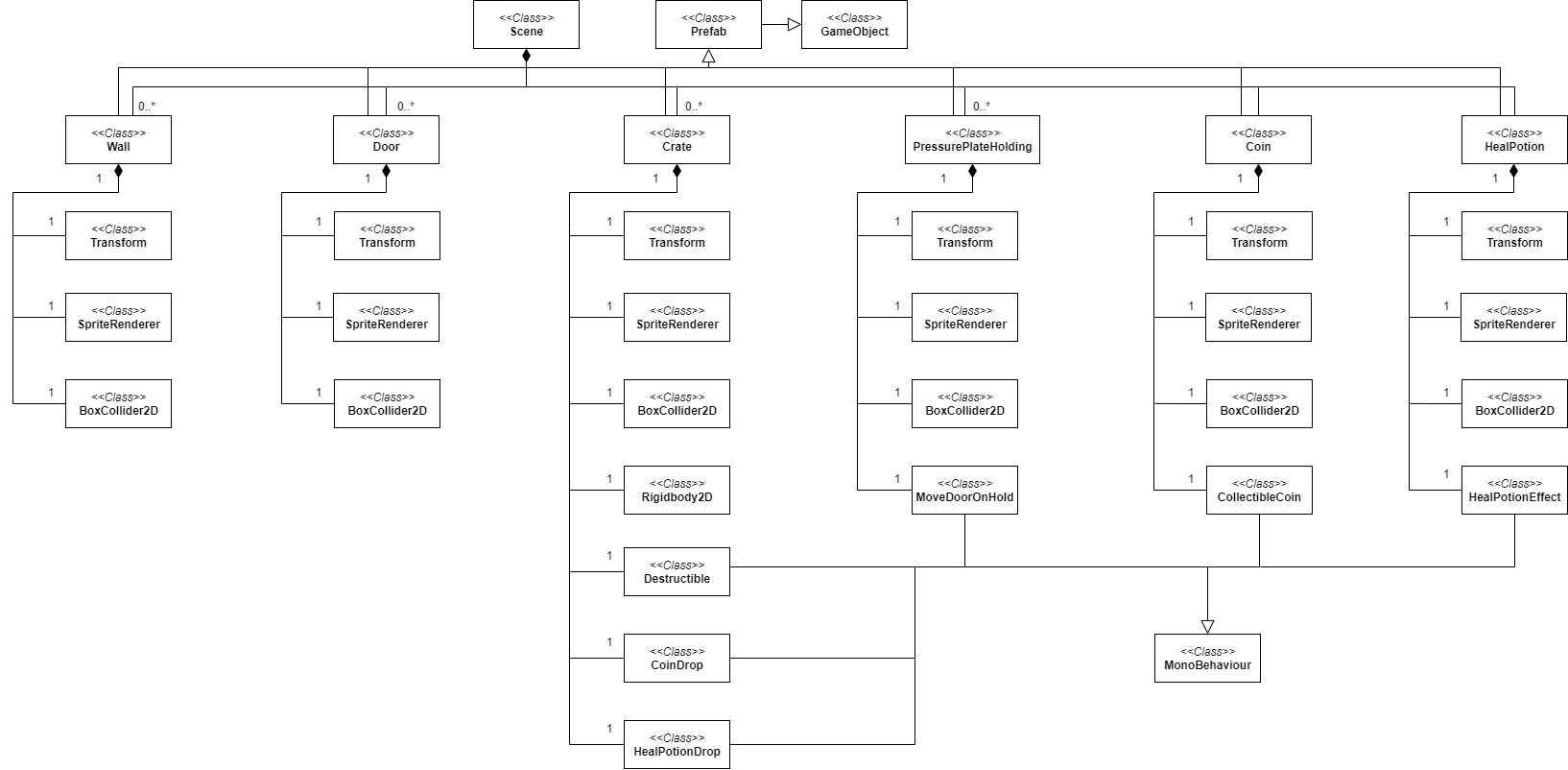


Рисунок 11 – Диаграмма классов предметов окружения

Более детализирующие диаграммы классов представлены ниже.

Диаграмма классов Mage класса представлена на рисунке 12:

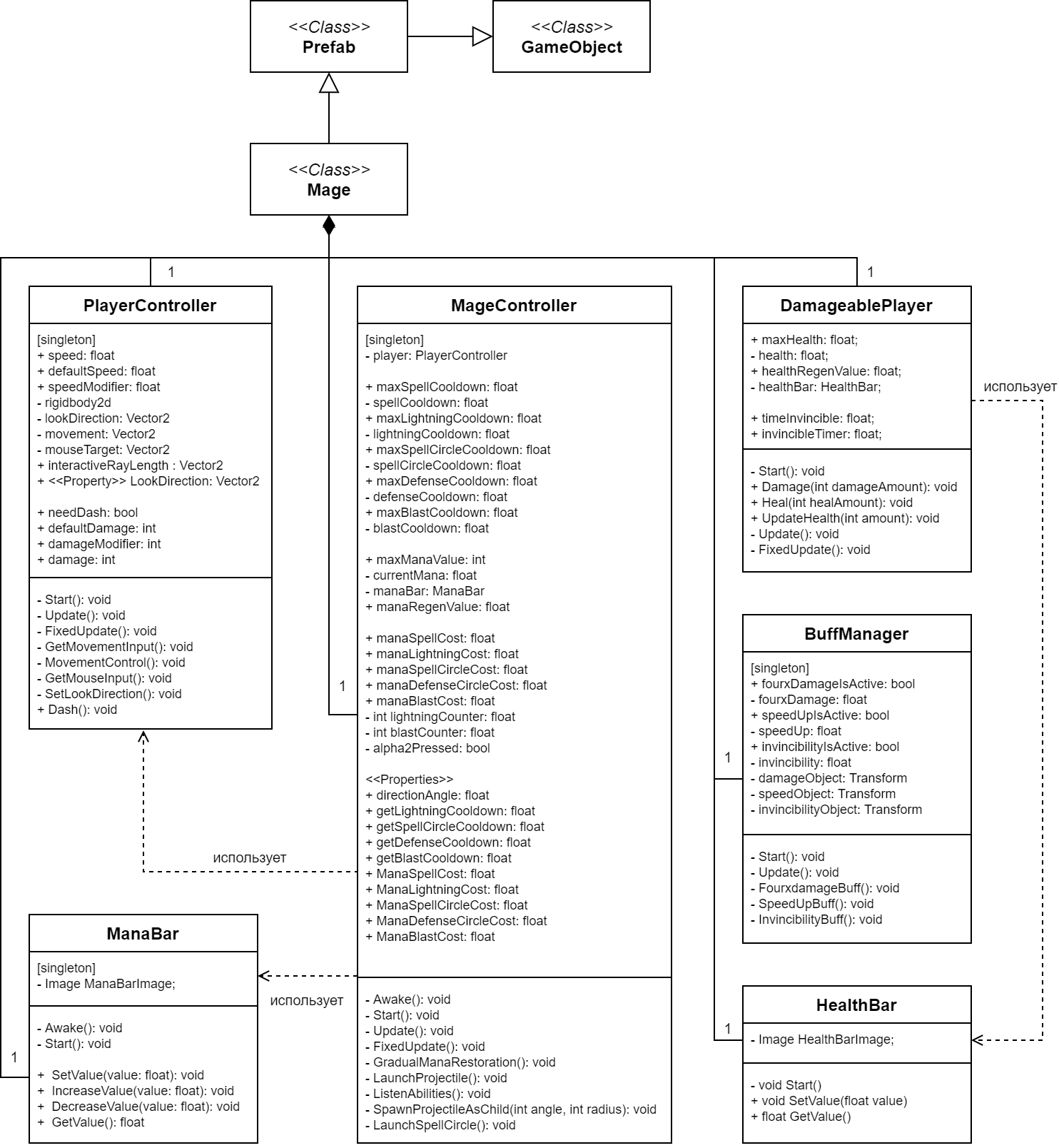


Рисунок 12 – Диаграмма классов Mage класса

Диаграмма классов Warrior класса представлена на рисунке 13:

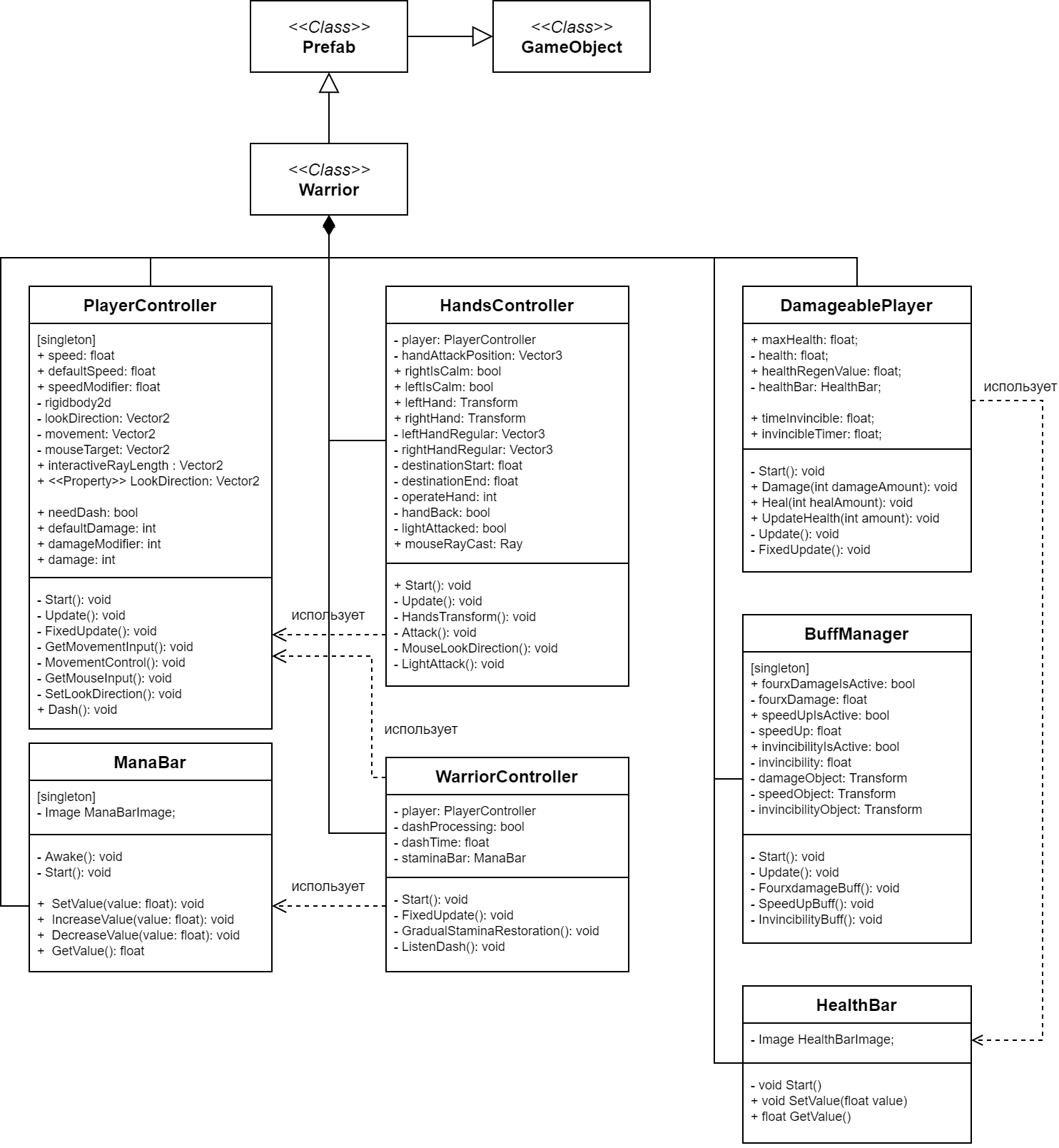


Рисунок 13 – Диаграмма классов Warrior класса

Диаграмма классов SkeletonHead класса представлена на рисунке 14:

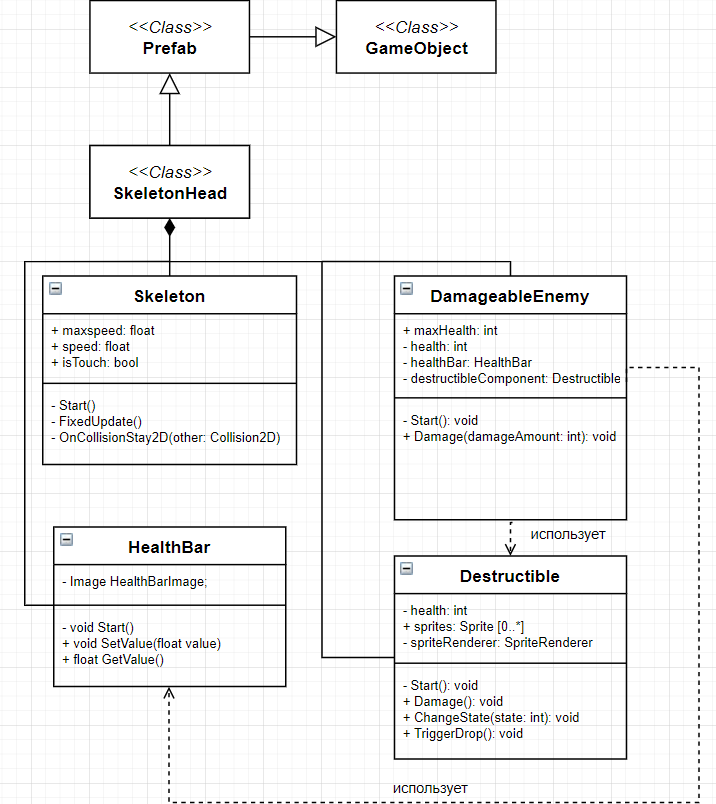


Рисунок 14- Диаграмма классов SkeletonHead класса

Диаграмма классов Necromancer класса представлена на рисунке 15:

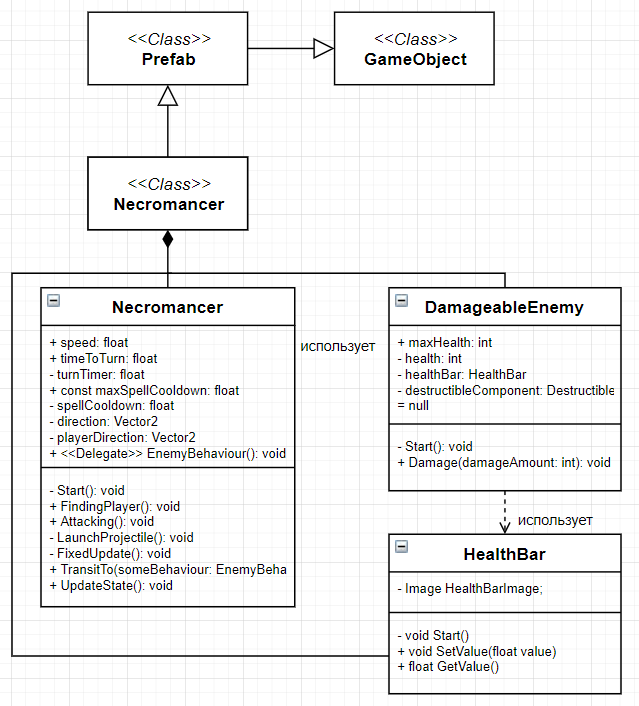


Рисунок 15– Диаграмма классов Necromancer класса

Классы - игровые объекты представляют из себя структуры, которые создаются в реальном времени на основе Префаба.

Ниже описано назначение нескольких ключевых методов класса MonoBehaviour:

* Start() вызывается в первом кадре визуализации объекта, когда скрипт активирован и непосредственно перед первым вызовом любого из методов Update.
* Update() вызывается каждый визуализированный кадр, если MonoBehaviour включен.
* FixedUpdate() – вызывается каждую дельту времени кадра для активного MonoBehaviour,

# Тестирование программного продукта

Для того чтобы удостовериться, что приложение работает корректно в соответствии с заявленным функционалом, необходимо провести тестирование. Для тестирования была выбрана стратегия по принципу «черного ящика». В этом случае программа рассматривается как «черный ящик», и целью тестирования является выяснение обстоятельств, в которых поведение программы не соответствует спецификации. Данная техника тестирования основана на работе исключительно с внешними интерфейсами тестируемого программного продукта. В таблице 2 приведены тесты по методу «предположение об ошибке».

Таблица 2 – Результаты тестирования по методу «Предположение об ошибке»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тест | Значение исходных данных | Ожидаемый результат | Полученный  результат | Вывод |
| Управление передвижением персонажа | Клавиши:  w – вперед  s – назад  a – влево  d - вправо | Передвижение персонажа по указанному игроком направлению | Передвижение персонажа по указанному игроком направлению | Тест успешно пройден |
| Управление снарядами персонажа “Mage” | Клавиши:  ЛКМ - выстрел | Инициирование снаряда и его импульса | Инициирование снаряда и его импульса | Тест успешно пройден |
| Управление первой способностью персонажа “Mage” | Клавиша:  Альфа 1 | Создание молнии в направлении мыши | Создание молнии | Тест успешно пройден |
| Управление второй способностью персонажа “Mage” | Клавиша:  Альфа 2 | Создание круга снарядов и его импульса в направлении мыши | Создание круга снарядов и его импульса | Тест успешно пройден |

Продолжение таблицы 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Управление третьей способностью персонажа “Mage” | Клавиши:  Альфа 3 | Создание энергетического щита | Создание энергетического щита | Тест успешно пройден |
| Управление четвертой способностью персонажа “Mage” | Клавиши:  Альфа 4 | Создание волны из снарядов в направлении мыши | Создание волны из снарядов | Тест успешно пройден |
| Расход маны первой способностью | Текущее количество маны | Расход маны на 7 пунктов | Расход маны на 7 пунктов | Тест успешно пройден |
| Расход маны второй способностью | Текущее количество маны | Расход маны на 15 пунктов | Расход маны на 15 пунктов | Тест успешно пройден |
| Расход маны третьей способностью | Текущее количество маны | Расход маны на 25 пунктов | Расход маны на 25 пунктов | Тест успешно пройден |
| Расход маны четвертой способностью | Текущее количество маны | Расход маны на 40 пунктов | Расход маны на 40 пунктов | Тест успешно пройден |
| Столкновение объекта “Projectile” и объекта “Wall” | Коллизия физических тел “Projectile” и “Wall” | Уничтожение объекта “Projectile” | Уничтожение объекта “Projectile” | Тест успешно пройден |
| Столкновение объекта “Projectile” и объекта “Crate” | Коллизия физических тел “Projectile” и “Crate” | Уничтожение объекта “Projectile”, нанесение повреждений объекту “Crate” | Уничтожение объекта “Projectile”, нанесение повреждений объекту “Crate” | Тест успешно пройден |
| Столкновение объекта “Projectile” и объекта “SkeletonHead” | Коллизия физических тел “Projectile” и “SkeletonHead” | Уничтожение объекта “Projectile”, нанесение повреждений объекту “SkeletonHead” на 1 пункт\* | Уничтожение объекта “Projectile”, нанесение повреждений объекту “SkeletonHead” на 1 пункт\* | Тест успешно пройден |

Продолжение Таблицы 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Столкновение объекта “Projectile” и объекта “Necromancer” | Коллизия физических тел “Projectile” и “Necromancer” | Уничтожение объекта “Projectile”, нанесение повреждений объекту “Necromancer” на 1 пункт\* | Уничтожение объекта “Projectile”, нанесение повреждений объекту “Necromancer” на 1 пункт\* | Тест успешно пройден |
| Столкновение объекта “Mage” и объекта “Necromancer” | Коллизия физических тел “Mage” и “Necromancer” | Нанесение повреждений объекту “Mage” на 35 пунктов | Нанесение повреждений объекту “Mage” на 35 пунктов | Тест успешно пройден |
| Столкновение объекта “Mage” и объекта “SkeletonHead” | Коллизия физических тел “Mage” и “SkeletonHead” | Нанесение повреждений объекту “Mage” на 35 пунктов | Нанесение повреждений объекту “Mage” на 35 пунктов | Тест успешно пройден |
| Столкновение объекта “Mage” и объекта “Crate” | Коллизия физических тел “Mage” и “Crate” | Смещение объекта “Crate” в сторону движения объекта “Mage” | Смещение объекта “Crate” в сторону движения объекта “Mage” | Тест успешно пройден |
| Столкновение объекта “Mage” и объекта “Wall” | Коллизия физических тел “Mage” и “Wall” | Невозможность движения объекта “Mage” в сторону нахождения объекта “Wall” | Невозможность движения объекта “Mage” в сторону нахождения объекта “Wall” | Тест успешно пройден |
| Управление снарядами персонажа “Warrior” | Клавиши:  ЛКМ - удар | Воспроизведение удара рукой | Воспроизведение удара рукой | Тест успешно пройден |
| Управление способностью персонажа “Warrior” | Клавиша:  ПКМ - рывок | Ускоренное перемещение персонажа в направлении мыши | Ускоренное перемещение персонажа в направлении мыши | Тест успешно пройден |

Продолжение Таблицы 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Столкновение объекта “Hand” и объекта “Crate” | Коллизия физических тел “Hand” и “Crate” | Нанесение повреждений объекту “Crate” | Нанесение повреждений объекту “Crate” | Тест успешно пройден |
| Столкновение объекта “Hand” и объекта “SkeletonHead” | Коллизия физических тел “Hand” и “SkeletonHead” | Нанесение повреждений объекту “SkeletonHead” на 1 пункт\* | Нанесение повреждений объекту “SkeletonHead” на 1 пункт\* | Тест успешно пройден |
| Столкновение объекта “Hand” и объекта “Necromancer” | Коллизия физических тел “Hand” и “Necromancer” | Нанесение повреждений объекту “Necromancer” на 1 пункт\* | Нанесение повреждений объекту “Necromancer” на 1 пункт\* | Тест успешно пройден |
| Столкновение объекта “Warrior” и объекта “Necromancer” | Коллизия физических тел “Warrior” и “Necromancer” | Нанесение повреждений объекту “Warrior” на 35 пунктов | Нанесение повреждений объекту “Warrior” на 35 пунктов | Тест успешно пройден |
| Столкновение объекта “Warrior” и объекта “SkeletonHead” | Коллизия физических тел “Warrior” и “SkeletonHead” | Нанесение повреждений объекту “Warrior” на 35 пунктов | Нанесение повреждений объекту “Warrior” на 35 пунктов | Тест успешно пройден |
| Столкновение объекта “Warrior” и объекта “Crate” | Коллизия физических тел “Warrior” и “Crate” | Смещение объекта “Crate” в сторону движения объекта “Warrior” | Смещение объекта “Crate” в сторону движения объекта “Warrior” | Тест успешно пройден |
| Столкновение объекта “Warrior” и объекта “Wall” | Коллизия физических тел “Warrior” и “Wall” | Невозможность движения объекта “Warrior” в сторону нахождения объекта “Wall” | Невозможность движения объекта “Warrior” в сторону нахождения объекта “Wall” | Тест успешно пройден |

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы разработано настольное приложение «ENCOURAGERS», основным предназначением которого является интерактивное компьютерное развлечение пользователя. Были разработаны боевая система, поведение примитивного искусственного интеллекта на основе машины состояний, взаимодействие с используемыми предметами и предметами окружения, пользовательский интерфейс и «Холст» игры в реальном времени.

В качестве средства реализации был выбран Unity. Была изучена работа с движком Unity, а конкретно работа с моделированием сцен, заданием скриптов, пользовательским интерфейсом, камерой сцены, физической системы, визуальными эффектами, аудио слушателем.

Заявленная функциональность настольного приложения реализована в соответствии с требованиями технического задания, однако данный проект можно развивать в различных направлениях, например, добавление функционала случайно генерируемых локаций, добавление новых классов и добавление экономической системы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Иванова Г.С. Технология программирования: учебник / Г.С. Иванова. – 3- е изд., стер. – М.: КНОРУС, 2016. – 334 с.

2. Г. С. Иванова, Т. Н. Ничушкина, Е.К. Пугачев, Р.С. Самарев. Методические указания по выполнению курсовой работы по дисциплине «Технология разработки программных систем». МГТУ имени Н.Э. Баумана. 2019. 41 c.

3. ГОСТ 7.32-2017 СИБИД. Отчет о научно-исследовательской работе. Структура и правила оформления (с Поправкой)

4. Редактирование сцены в Unity: https://docs.unity3d.com/Manual/CreatingScenes.html

5. Canvas элемент: https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/UICanvas.html

6. Unity Scripting API: https://docs.unity3d.com/ScriptReference/index.html