|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |
| Институт искусственного интеллекта |
| Кафедра программного обеспечения систем радиоэлектронной аппаратуры |

КУРСОВАЯ РАБОТА

по дисциплине «Методы и стандарты программирования»

на тему: «Создание игры Deep Rock Galactic Survivor»

|  |  |  |  |
| --- | --- | --- | --- |
| Обучающийся | \_\_\_\_\_\_\_\_\_\_\_\_\_ | Карпов Дмитрий Александрович | |
|  | *Подпись* |
| Шифр |  | 23К0029 |  |
| Группа |  | КМБО-02-23 |  |
|  |  |  |  |
| Руководитель  работы | \_\_\_\_\_\_\_\_\_\_\_\_\_ | Черноусов Игорь Дмитриевич | |
|  | *Подпись* |  | |

Москва 2024

Оглавление

[ВВЕДЕНИЕ 2](#_Toc184563829)

[2 ПОСТАНОВКА ЗАДАЧИ 3](#_Toc184563830)

[3 ОТЧЁТ О РАЗРАБОТКЕ 5](#_Toc184563831)

[4 РУКОВОДСТВО ПО СБОРКЕ И ЗАПУСКУ 7](#_Toc184563832)

[5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 8](#_Toc184563833)

[ЗАКЛЮЧЕНИЕ 12](#_Toc184563834)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 13](#_Toc184563835)

# ВВЕДЕНИЕ

Целью курсовой работы является создание копии игры, аналогичной “Deep Rock Galactic: Survivor”, представляющей собой экшен в жанре выживания с элементами добычи ресурсов. Игроку предстоит управлять космическим шахтером, сражаться с волнами врагов и боссами, добывать руду и улучшать свое оружие.

Для осуществления проекта необходимо было изучить библиотеку SFML и принцип разработки игр в ней.

# 2 ПОСТАНОВКА ЗАДАЧИ

Создаваемая игра включает в себя несколько ключевых элементов. Игрок управляет космическим шахтером от третьего лица, используя клавиатуру и мышь. Основная задача игрока — выжить как можно дольше, сражаясь с волнами врагов, добывая ресурсы и улучшая свое снаряжение.

Каждая волна содержит определённое количество врагов, которые появляются на карте в случайных местах. Враги преследуют игрока, нанося ему урон при столкновении. На каждой пятой волне появляется босс, который представляет повышенную сложность.

На карте размещены месторождения руды, которые игрок может добывать, взаимодействуя с ними. Собранная руда используется для улучшения оружия, брони и других предметов. Игроку необходимо находить баланс между временем, потраченным на добычу руды, и выживанием во время атак врагов.

По достижении нового уровня игрок получает возможность улучшить своё оружие и броню, используя собранные ресурсы. Улучшения могут включать повышение урона оружия, увеличение скорострельности или мощности атак, увеличение запаса здоровья, а также добавление новых модификаций к оружию, например, проходящие сквозь врагов патроны или очереди выстрелов.

Игра начинается на небольшой карте, представляющей собой подземную шахту. Локация включает различные открытые пространства, где могут находиться ресурсы и появляться враги. Каждая новая волна врагов начинается через определённый промежуток времени, в течение которого игрок может собирать ресурсы и готовиться к нападению. После того, как игрок проходит волну, уровень игры меняется, и сложность врагов и боссов увеличивается.

Пользовательский интерфейс игры включает в себя главное меню с кнопками «Играть» и «Выход», а также игровой HUD с индикаторами здоровья игрока, его опыта и количеством собранной руды. Также предусмотрено меню улучшений, которое показывает доступные для улучшения элементы экипировки и количество ресурсов, необходимых для этого.

В игре предусмотрено звуковое сопровождение с фоновой музыкой на протяжении всего игрового процесса. Также реализованы анимации движений персонажа, такие как бег и атака, анимации передвижений и атак врагов.

Игра обладает рядом особенностей. Местоположение рудных залежей и начальные точки появления врагов случайным образом генерируются на каждом уровне. Каждая пятая волна содержит уникального босса, который представляет повышенную сложность для игрока. Сложность игры постепенно увеличивается: враги становятся сильнее, а боссы — более опасными.

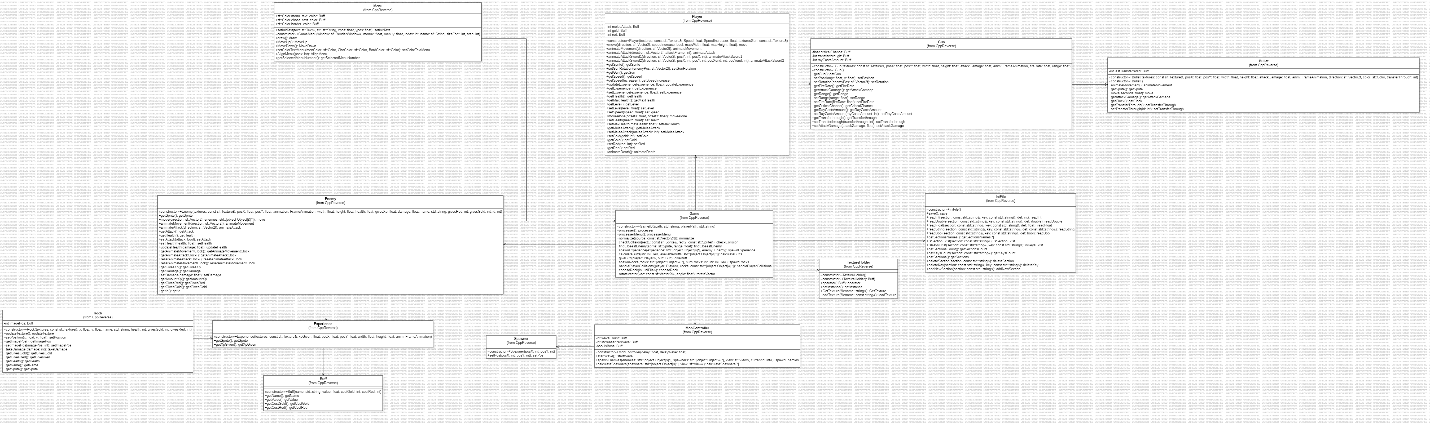
# 3 ОТЧЁТ О РАЗРАБОТКЕ

Архитектура программы состоит из классов, которые можно разбить на 3 вида:

1. Классы-контроллеры – отвечают за процесс игры и интерфейс, с которым взаимодействует пользователь (game.h, gamemenu.h)
2. Классы-модели – хранят информацию о внутриигровых предметах, такую как количество здоровья, урон, текущая отображающаяся картинка. Благодаря им можно легко добавлять новые предметы, персонажи и улучшения без значительного изменения кода. (enemy.h, gun.h, bullet.h и т.д.)
3. Вспомогательные классы – хранят функции, к которым можно обратиться из любого вышеописанного класса. Реализовано два таких класса: textureholder.h – singleton-класс, который хранит в себе std::map со всеми текстурами, которые используются в игре, чтобы при каждой новой отрисовке не приходилось открывать файл, содержащий текстуру; inifile.h – представляет интерфейс для взаимодействия с .ini файлами, оттуда берутся данные о ресурсах, которые добывает игрок и улучшениях, которые он может получить.

В игре также реализован алгоритм поиска врагами кратчайшего пути с учётом всех остальных врагов. Ни один враг не может задеть другого или зайти в него. В игре есть 4 позиции, откуда могут начать появляться враги, они определяются случайным образом, исключая те, которые входят в область видимость игрока. Также был добавлен алгоритм случайной генерации ресурсов, характеристики которых через .ini файл.

На рисунке 1 приведена диаграмма всех классов игры, а также методов, которые они включают.

Рисунок 1 – Диаграмма классов

# 4 РУКОВОДСТВО ПО СБОРКЕ И ЗАПУСКУ

Чтобы собрать и локально запустить проект необходимо:

Клонировать репозиторий <https://github.com/Esternit/DiversGame>

Убедиться в наличии Visual Studio версии не ниже 2022

Открыть через Visual Studio файл DriversGame.sln

Запустить Отладчик Windows с конфигурацией Debug x64.

# 5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

При входе в игру пользователь видит главное меню (Рисунок 2), состоящее из кнопки играть (Play) и выйти (Exit). Навигация в меню реализована через стрелки «Вверх» и «Вниз». Активная кнопка отображается красным цветом, а нажать на активную можно с помощью клавиши «Enter».

Рисунок 2 – Главное меню игры

Во время игры пользователь может управлять персонажем с помощью клавиш «WASD» и добывать руду, подбежав к ней, с помощью клавиши «E». На рисунке 3 приведён пример игрового процесса в самом начале игры.

Рисунок 3 – Игровой процесс

Во время игрового процесса на игрока волнами наступают враги (Рисунок 4). После каждого убийства из врага выпадает опыт (Рисунок 4), который можно собрать, подбежав к нему.



Рисунок 4 – Враги и опыт, выпадающий с них

Опыт увеличивает уровень игрока, отображающийся вместе со здоровьем и показателями ресурсов в левом верхнем углу (Рисунок 5). По достижении нового уровня игроку даётся выбор между случайными 3-мя характеристиками, которые он может купить за накопленные ресурсы (Рисунок 6). Чтобы выбрать улучшение надо кликнуть на желаемое улучшение.



Рисунок 5 – Игровой HUD



Рисунок 6 – Меню выбора улучшений

# ЗАКЛЮЧЕНИЕ

В результате курсовой работы получилось реализовать копию игры Deep Rock Galactic Survivor. Основная часть механик из игры была добавлена в финальный проект. Однако вместо механики со сменой локации после определённой волны была заменена на бесконечные волны с боссами раз в несколько волн и постепенно увеличивающейся сложностью врагов.

Была изучена библиотека для разработки игры на C++ SFML и основные принципы работы с ней, такие как:

Создание окна

Главный цикл приложения

Обработка событий

Работа с графическими объектами

Работа с видом игрока (View)

Анимация фреймов персонажей

Работа со звуком

Модульная структура

Работа с временем.

Также был реализован главный каркас игры, который позволяет легко добавлять новых персонажей, камни, улучшения. Для этого был сделан интерфейс для работы с .ini файлами, которые находятся в корневой папке проекта вместе с исходными данными.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

Официальный сайт библиотеки SFML – URL: <https://www.sfml-dev.org/>

Разработка игры на C++/SFML: Начало – URL: <https://habr.com/ru/articles/800691/>

Игровое меню SFML C++ – URL: <https://habr.com/ru/articles/710100/>

Artur Moreira SFML Game Development ­– Packt publishing, 2013. – 296с.