|  |
| --- |
| 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](#_Toc184903896)

[1 ПОСТАНОВКА ЗАДАЧИ 3](#_Toc184903897)

[2 ОТЧЁТ О РАЗРАБОТКЕ 5](#_Toc184903898)

[3 РУКОВОДСТВО ПО СБОРКЕ И ЗАПУСКУ 7](#_Toc184903899)

[4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 8](#_Toc184903900)

[ЗАКЛЮЧЕНИЕ 12](#_Toc184903901)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 13](#_Toc184903902)

# 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 изучить библиотеку SFML и принцип разработки игр в ней.

# 1 ПОСТАНОВКА ЗАДАЧИ

Создаваемая игра будет включать несколько ключевых элементов. Игрок управляет космическим шахтером от третьего лица, используя клавиатуру и мышь. Основной задачей станет выживание как можно дольше, сражаясь с волнами врагов, добывая ресурсы и улучшая снаряжение.

Каждая волна будет состоять из определённого количества врагов, которые появляются на карте в случайных местах. Враги преследуют игрока, нанося ему урон при столкновении. На пятой волне возникнет босс, значительно сложнее обычных врагов.

На карте размещены месторождения руды, которые игрок может добывать, взаимодействуя с ними. Собранные ресурсы позволят улучшать оружие, броню и другие элементы экипировки. Игроку нужно будет найти баланс между временем, потраченным на добычу, и необходимостью защищаться от врагов.

Каждый новый уровень откроет возможность улучшить оружие и броню с использованием собранных ресурсов. Улучшения будут включать повышение урона оружия, увеличение скорострельности или мощности атак, рост запаса здоровья и добавление новых модификаций, таких как патроны, проходящие сквозь врагов, или очереди выстрелов.

Игра начнётся на небольшой карте, представляющей собой подземную шахту. Локация включает различные открытые пространства, где игрок может найти ресурсы и встретить врагов. Каждая волна врагов начнётся через определённый промежуток времени, в течение которого можно собирать ресурсы и готовиться к атаке. После победы над волной уровень игры изменится, а сложность врагов и боссов увеличится.

Пользовательский интерфейс включает главное меню с кнопками «Играть» и «Выход», а также игровой интерфейс с индикаторами здоровья игрока, его опыта и собранной руды. Дополнительно предусмотрено меню улучшений, где отображаются доступные для апгрейда элементы экипировки и количество требуемых ресурсов.

Игра будет сопровождаться звуками и фоновыми музыкальными композициями на протяжении всего игрового процесса. Также реализованы анимации движений персонажа, таких как бег и атака, а также анимации передвижений и атак врагов.

Игра будет иметь несколько уникальных особенностей. Местоположение залежей руды и начальные точки появления врагов генерируются случайным образом на каждом уровне. Каждая пятая волна будет содержать уникального босса, который станет настоящим вызовом для игрока. Сложность игры будет увеличиваться с каждым уровнем: враги станут сильнее, а боссы — опаснее.

# 2 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 задаются через .ini файл.

# 3 РУКОВОДСТВО ПО СБОРКЕ И ЗАПУСКУ

Требования:

Visual Studio 2022 с Platform Toolset v143,

библиотека SFML.

Чтобы собрать и локально запустить проект необходимо:

1. Убедитесь, что у вас есть все исходные файлы проекта.
2. Убедитесь, что установлены необходимые компоненты
3. Откройте файл DriversGame.sln в Visual Studio (проверено на версии 2022).
4. В Visual Studio откройте свойства проекта (правый клик на проект, Properties).
5. В разделе VC++ Directories добавьте пути к папке include (для заголовочных файлов) и lib (для библиотек) SFML.
6. В разделе Linker, Input добавьте необходимые файлы SFML для линковки (например, sfml-graphics-d.lib, sfml-window-d.lib, sfml-system-d.lib для Debug, или sfml-graphics.lib, sfml-window.lib, sfml-system.lib для Release).
7. Убедитесь, что файлы DLL библиотеки SFML находятся в той же папке, что и исполнимый файл, или в системном пути.
8. В Configuration Manager выберите конфигурацию Debug x64.
9. Запустите проект с использованием Отладчика Windows.

# 4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

При входе в игру пользователь видит главное меню (Рисунок 1), 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 1 – 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 2 – 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 3 – 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 4 – Игровой интерфейс



Рисунок 5 – 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 со временем.

Также для удобного добавления нового контента в игры был реализован функционал работы с .ini файлами. Он позволяет добавлять в файл новые элементы игры, прописывая их характеристики, с минимальными вмешательствами в код лишь для того, чтобы прописать позицию пикселей для правильной отрисовки анимаций.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

Официальный сайт библиотеки SFML – URL: <https://www.sfml-dev.org/> (дата обращения 12.12.2024)

Разработка игры на C++/SFML: Начало – URL: <https://habr.com/ru/articles/800691/> (дата обращения 12.12.2024)

Игровое меню SFML C++ – URL: <https://habr.com/ru/articles/710100/> (дата обращения 12.12.2024)

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