|  |
| --- |
| 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»

на тему:

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

Москва 2024

Оглавление

[ВВЕДЕНИЕ 3](#_Toc184485404)

[2 ПОСТАНОВКА ЗАДАЧИ 4](#_Toc184485405)

[3 ОТЧЁТ О РАЗРАБОТКЕ 6](#_Toc184485406)

# 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

# 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.