# Архитектурное описание рогалика FedyaGame

|  |  |
| --- | --- |
| Кудрявцев Федор | [koufesser@gmail.com](mailto:koufesser@gmail.com) |
| Попов Егор |  |
| Красов Владислав |  |
| Мосин Александр |  |

Оглавление

[Архитектурное описание рогалика FedyaGame 1](#_Toc164108986)

[Общие сведения о системе 1](#_Toc164108987)

[Architectural Drivers 1](#_Toc164108988)

[Роли и случаи использования 1](#_Toc164108989)

[Описание типичного пользователя 1](#_Toc164108990)

[Композиция (диаграмма компонентов) 1](#_Toc164108991)

[Логическая структура (диаграмма классов) 1](#_Toc164108992)

[Взаимодействия и состояния (диаграммы последовательностей и конечных автоматов) 1](#_Toc164108993)

### Общие сведения о системе

### Общие сведения о системе:

### Название игры: "FedyaGame"

### Жанр: Roguelike (игра-рогалик)

### Платформы: Windows, macOS, Linux

### Режимы игры: Одиночный (офлайн)

### Краткое описание: "FedyaGame" - это классический рогалик с простой консольной графикой, активным использованием случайной генерации и перманентной смертью персонажа. Игрок управляет персонажем, который перемещается по сгенерированной карте, сражается с монстрами, собирает вещи и улучшает свои характеристики. Игра характеризуется высокой свободой действий персонажа и чрезвычайно развитым набором игровых правил.

### Целевая аудитория: Любители рогаликов и RPG, игроки от 12 лет

### Ключевые особенности:

### Простая консольная графика

### Активное использование случайной генерации

### Перманентная смерть персонажа и невозможность загрузить предыдущее сохранение

### Чрезвычайно развитый набор игровых правил

### Высокая свобода действий персонажа ("игры-песочницы")

### Управление персонажем с клавиатуры

### Карта, генерируемая или загружаемая из файла

### Характеристики персонажа (здоровье, сила атаки и т.д.)

### Инвентарь персонажа, состоящий из вещей, которые он носит с собой

### Вещи, которые можно надеть и снять, влияющие на характеристики персонажа

### Вещи, которые можно поднять с карты и добавить в инвентарь

### Снятые вещи, которые можно надеть в дальнейшем

### Игровой движок: Собственная реализация на языке программирования C++

### Язык программирования: C++

### Графика: консольная

### Звук и музыка: Минималистичный звук и музыка, созданные специально для игры

### Локализация: Английский, русский и другие (в зависимости от спроса)

### Разработчик: "Aboba Studios"

### Издатель: "Indie Games Inc."

### 15. Дата выхода: Ожидается в 2025 году

### Architectural Drivers

1. Простота и гибкость: Архитектура должна быть простой и гибкой, чтобы обеспечить легкое изменение и расширение игровых правил, а также поддерживать активное использование случайной генерации.
2. Надежность: Обеспечение стабильности и надежности игры, предотвращение ошибок и сбоев. Для этого необходимо использовать отказоустойчивые архитектурные решения и проводить регулярные тестирования.
3. Высокая производительность: Игра должна обеспечивать высокую производительность и плавность работы на различных устройствах и конфигурациях, несмотря на простую тайловую или консольную графику.
4. Расширяемость: Архитектура должна предоставлять возможности для расширения и модификации игрового контента, таких как добавление новых вещей, врагов и уровней.
5. Удобство использования: Игра должна иметь интуитивно понятный и удобный интерфейс, а также предоставлять ясные инструкции и подсказки для новичков.
6. Простота поддержки: Архитектура должна обеспечивать простоту поддержки и обновления игры, а также предоставлять надежные инструменты и фреймворки для отладки и тестирования.
7. Безопасность: Защита игровых данных и личной информации пользователей от несанкционированного доступа, мошенничества и хакерских атак. Для этого необходимо использовать современные методы шифрования, аутентификации и авторизации, а также регулярно обновлять системы безопасности.
8. Переносимость: Игра должна быть легко переносимой на различные платформы и устройства, а также поддерживать различные интерфейсы взаимодействия (клавиатура/мышь, контроллер). Для этого необходимо использовать кросс-платформенные технологии и инструменты разработки, а также обеспечить гибкую архитектуру и адаптивный дизайн.
9. Совместимость: Игра должна быть совместима с различными версиями операционных систем, драйверами и сторонним программным обеспечением. Для этого необходимо проводить регулярные тестирования и вносить необходимые изменения в код и конфигурацию.

### Роли и случаи использования

Роли:

1. Игрок - человек, который управляет персонажем в игре.

#### Типичный пользователь:

Типичный пользователь игры "FedyaGame" - это любитель рогаликов и RPG, который предпочитает игры с простой графикой и высоким уровнем сложности. Ему нравится активное использование случайной генерации и перманентная смерть персонажа, которые делают игру уникальной и непредсказуемой. Он ценит высокую свободу действий персонажа и развитый набор игровых правил, которые позволяют ему экспериментировать с различными стратегиями и тактиками. Типичный пользователь также предпочитает игры, которые можно играть в одиночку и офлайн, без необходимости подключения к интернету или взаимодействия с другими игроками.

Возраст: 12 – 35 лет

Место проживания: подвал родительского дома

Background: СНГ

### Композиция (диаграмма компонентов)

Вот примерная диаграмма компонентов для игры "Labyrinth of Shadows":

Изображение выглядит как текст, снимок экрана, диаграмма

Автоматически созданное описание

Описание компонентов:

1. Game Engine - основной игровой движок, отвечающий за обработку графики, звука, ввода/вывода и других низкоуровневых задач.

2. Game Logic - компонент, отвечающий за основную логику игры, включая перемещение персонажа, взаимодействие с объектами на карте, бои и т.д.

3. Character Controller - компонент, отвечающий за управление персонажем, включая его движение, атаки и другие действия.

4. Map Loader & Generator - компонент, отвечающий за загрузку и генерацию уровней игры.

5. Inventory Manager - компонент, отвечающий за управление инвентарем персонажа, включая добавление, удаление и использование предметов.

6. Item Manager - компонент, отвечающий за управление предметами в игре, включая их характеристики и эффекты.

7. Save & Load Manager - компонент, отвечающий за сохранение и загрузку игры, включая сохранение прогресса персонажа и состояния игры.

8. UI Manager - компонент, отвечающий за отображение графического интерфейса пользователя, включая меню, диалоги и другие элементы интерфейса.

### Логическая структура (диаграмма классов)

Изображение выглядит как текст, снимок экрана, диаграмма

Автоматически созданное описание

Вот примерная диаграмма классов для игры "FedyaGame":

Описание классов:

1. Game - основной класс игры, отвечающий за запуск, обновление и рендеринг игры.

2. Character - класс персонажа, отвечающий за его характеристики, инвентарь и перемещение по карте.

3. Map - класс карты, отвечающий за генерацию и загрузку уровней, а также за хранение информации о плитках и их свойствах.

4. Inventory - класс инвентаря, отвечающий за хранение и управление предметами, которые носит персонаж.

5. Item - класс предмета, отвечающий за хранение информации о названии, описании, эффекте и других характеристиках предмета.

6. Tile - класс плитки, отвечающий за хранение информации о типе плитки (например, трава, стена, дверь) и о том, можно ли по ней пройти.

7. UI - класс графического интерфейса пользователя, отвечающий за отображение кнопок, надписей и других элементов интерфейса.

8. GameManager - класс менеджера игры, отвечающий за управление игрой, включая запуск, паузу, сохранение и загрузку игры.

### Взаимодействия и состояния (диаграммы последовательностей и конечных автоматов)

### Изображение выглядит как текст, диаграмма, Параллельный, линия Автоматически созданное описание

Изображение выглядит как текст, диаграмма, линия, снимок экрана

Автоматически созданное описание