

**Проектная практика:  
Разработка игры на C++  
“Красный лёд”**



**Архитектура игры**

Команда: Дамы с Севера  
Евдокимова Анна Ивановна  
Биктагирова Диана Рафиковна

**ИТМО**

Санкт-Петербург, 2025

*Игра "Красный лёд" является вымышленной текстовой RPG и создана исключительно в развлекательных целях. Она содержит упоминания и элементы, связанные с оружием, наркотиками и криминальной деятельностью, которые представлены в рамках игрового сюжета и сеттинга. Авторы игры не пропагандируют запрещенные на территории РФ психотропные вещества и призывы к вооруженному противостоянию.*

## 1. Игровые механики

### 1.1. Неформальное описание:

- **Механика боя:** Игрок выбирает тип оружия (пистолет, граната, РПГ) и количество использованных боеприпасов. Урон рассчитывается на основе типа оружия и количества боеприпасов. Например, пистолет наносит 5 НР за патрон, граната — 10 НР за штуку, РПГ — 20 НР за выстрел.
- **Ловушки:** Наносят фиксированный урон (20 НР) врагу.
- **Переговоры:** Игрок может попытаться договориться с врагом, потратив деньги. Успех переговоров зависит от суммы и влияет на уровень влияния игрока.
- **Магазин:** Позволяет покупать оружие и боеприпасы. Доступные предметы зависят от текущего уровня игры.
- **Влияние и НР:** Влияние и НР игрока динамически изменяются в зависимости от действий. Если влияние падает ниже 30% или НР достигает 0, игрок проигрывает.

### 1.2. Формальное описание:

#### • Урон от оружия:

Формула:

$$\text{totalDamage} = \text{damagePerUnit} * \text{ammoSpent}$$

Где:

- **damagePerUnit** зависит от типа оружия:

- Пистолет: 5

- Граната: 10

- РПГ: 20

- **ammoSpent** — количество использованных боеприпасов.

#### • Ловушки:

Формула:

enemyHP -= 20

Если enemyHP < 0, то enemyHP = 0.

**•Переговоры:**

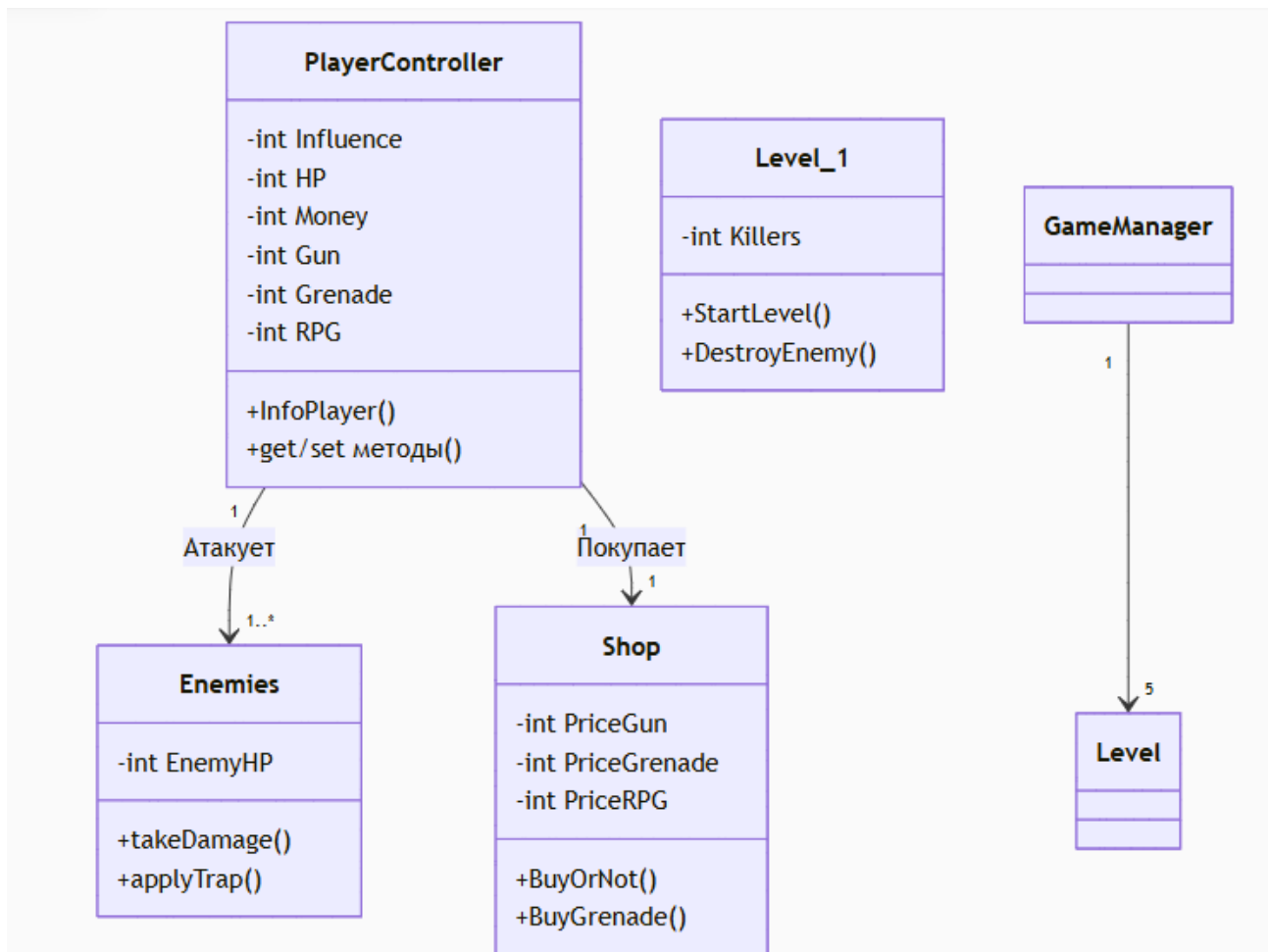
Успех переговоров определяется условиями:

- Если игрок соглашается на предложение врага, он теряет деньги и влияние.
- Если игрок торгуется, он может сохранить часть ресурсов, но рискует вызвать агрессию врага.

**Диаграмма состояний:**



**Диаграмма классов:**

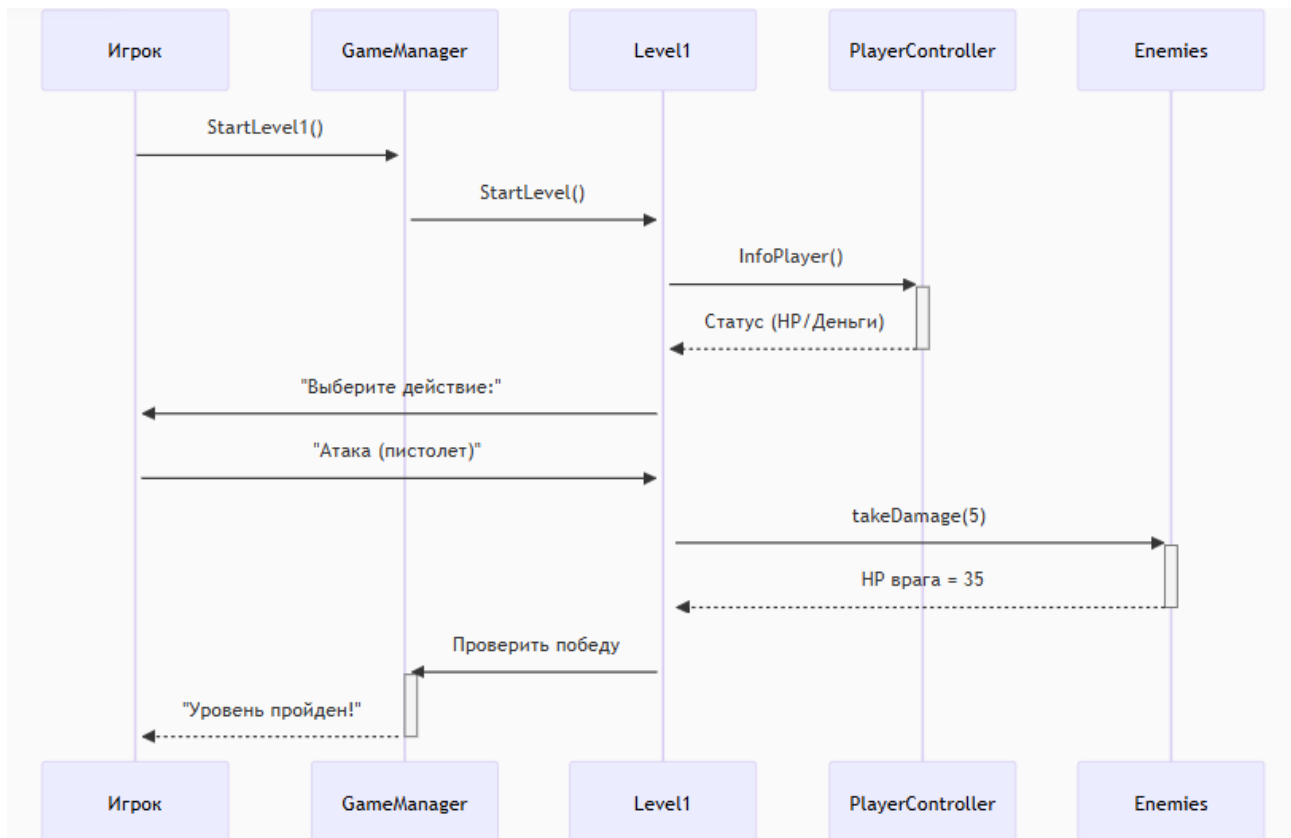


## 2. Общая архитектура игры

### 2.1. Неформальное описание:

- **Уровни (Levels):** Каждый уровень представляет собой отдельный сценарий с уникальными событиями и врагами. Уровни наследуются от базового класса и реализуют свои методы (например, StartLevel, Attack).
- **Управление игроком (PlayerController):** Хранит и управляет состоянием игрока (HP, влияние, деньги, инвентарь).
- **Враги (Enemies):** Отвечает за HP врагов и обработку урона.
- **Магазин (Shop):** Позволяет игроку покупать оружие и боеприпасы.
- **Менеджер игры (GameManager):** Управляет переходами между уровнями и проверяет условия победы/поражения.

### 2.2. Диаграмма последовательности (UML Sequence Diagram):



### 3. Физическая организация

Таблица распределения по файлам:

Файл	Описание
Enemies.h/cpp	Класс для управления врагами (HP, урон).
PlayerController.h/cpp	Управление состоянием игрока (HP, влияние, инвентарь).
Shop.h/cpp	Магазин для покупки оружия и боеприпасов.
Level_1.h/cpp	Первый уровень игры, реализует сценарии боя и переговоров.
Level_2.h/cpp	Второй уровень с расширенными механиками (ловушки, атаки).
GameManager.h/cpp	Управление переходами между уровнями и проверка условий победы.
Text.h/cpp	Текстовые описания событий и диалогов для уровней.
RedIce2.cpp	Главный файл, запускающий игру.

