

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ

КАФЕДРА ПРОГРАММНАЯ ИНЖЕНЕРИЯ

ОТЧЁТ О ПРОДЕЛАННОЙ РАБОТЕ

«информатика и основы программирования»

Создание текстовой RPG-игры

Студент

гр. БИН-25-1 _____ Д.В.Потылицин

Преподаватель _____ А.Д.Шнейдер

Владивосток 2025

Содержание

Содержание	Ошибка! Закладка не определена.
Введение	Ошибка! Закладка не определена.
1 Описание концепции	Ошибка! Закладка не определена.
1.1 Игровой мир и сюжет	Ошибка! Закладка не определена.
1.2 Основные механики	Ошибка! Закладка не определена.
1.3 Баланс	Ошибка! Закладка не определена.
2 Разработка	Ошибка! Закладка не определена.
2.1 Система классов	Ошибка! Закладка не определена.
2.2 Класс <i>player</i>	Ошибка! Закладка не определена.

Введение

Цель проекта — разработка полностью функциональной текстовой RPG-игры на языке Python, реализующей основные механики жанра: создание персонажа, систему боя, повышения уровня, инвентарь и исследование мира.

1 Описание концепции

1.1 Игровой мир и сюжет

Действие игры происходит в фэнтезийном мире, наполненном опасностями и приключениями. Игрок принимает роль безымянного героя, который путешествует по различным локациям, сражается с врагами, собирает предметы и повышает свой уровень. Хотя игра не имеет развернутого сюжета, её геймплей построен вокруг цикла "исследование-бой-прокачка", который мотивирует игрока к дальнейшему прохождению.

Конечная цель – достижение 5-го уровня и победа над одним из финальных боссов: "Старостой Давидом" или "Фенриром". Каждый босс имеет уникальные характеристики, требующей от игрока стратегического подхода и правильного использования накопленных ресурсов. Например, староста Давид имеет всего 5 единиц здоровья, но 99% защиту.

1.2 Основные механики

Игрок начинает игру с создания персонажа, основные характеристики которого — здоровье, урон, броня и уклонение — генерируются случайным образом в заданных пределах. Это позволяет каждому новому прохождению быть уникальным. При неудовлетворённости начальными значениями игрок имеет возможность перераспределить их повторно до начала приключения.

Основной игровой цикл строится вокруг исследования мира, в ходе которого случайным образом происходят различные события. Игрок может встретить врага, вступить с ним в бой, найти сундук с ценными предметами или же безопасную локацию для отдыха, восстанавливающую здоровье. Система боя реализована пошагово: в свой ход игрок выбирает одно из доступных действий — атаковать противника, использовать предмет из инвентаря или попытаться уклониться от следующей вражеской атаки. Важно отметить, что использование предмета является самостоятельным действием и отменяет возможность атаки в текущем ходу, что добавляет стратегической глубины принятию решений.

После успешной победы над врагом персонаж не только получает опыт, но и частично восстанавливает здоровье. Накопление опыта ведёт к повышению уровня игрока, что является ключевым элементом прогрессии. Инвентарь персонажа может содержать разнообразные предметы, такие как оружие, боеприпасы, лечебные средства и специальные артефакты. Их использование предоставляет временные бонусы к характеристикам или накладывает постоянные эффекты, напрямую влияющие на игровой процесс и открывающие дополнительные тактические возможности в бою.

1.3 Баланс

В игре очень развитый баланс. Сложность врагов на высоких уровнях компенсируется бесконечным инвентарём и вероятностью получить любое количество сундуков между боями. Это достигается благодаря уникальной механике: в начале игры или после каждой победой над врагом следующая локация выбирается абсолютно случайно, может попасться как несколько сундуков или отдых, так и новый враг. Это позволяет делать уровень сложности более динамическим: то легко, то сложно.

2 Разработка

2.1 Система классов

Игра изначально разрабатывалась с объектно-ориентированным программированием (классы). Это сделано для того, чтобы было проще обращаться к переменным в разных частях кода. В игре всего 2 класса: это *Player* (Игрок) и *Enemy* (Враг). Класс *Player* используется как для хранения данных об игроке, так и для основы игры. Класс *Enemy* используется для хранения информации о враге.

2.2 Класс *Player*

Класс *Player* включает следующие функции:

- `__init__` – инициализация игрока с заданием характеристик и начальных предметов;
- `show_status` – вывод текущих параметров персонажа; `walk` – перемещение по случайным локациям с обработкой событий (сундук, враг, отдых);
- `battle` – управление пошаговым боем с выбором действия;
- `attack` – атака врага с учётом брони и эффектов;
- `use_item` – использование предметов из инвентаря с применением соответствующих эффектов;
- `avoid` – попытка уклонения от атаки врага;
- `get_damage` – расчёт итогового урона с учётом всех модификаторов;
- `death` – обработка поражения игрока;
- `victory` – завершение игры при победе над финальным боссом.

2.3 Класс *Enemy*

Класс *Enemy* представляет игрового противника и отвечает за хранение его характеристик. При создании объекта класса задаётся имя врага, на основе которого

определяются его основные параметры: здоровье, урон, броня, опыт за победу и особые эффекты.

2.4 Внеклассовая часть кода

Помимо классов в игре содержатся следующие части кода:

- Импорт библиотеки для определения случайных чисел или выбора;
- Константы с названиями локаций, предметов и врагов на разных уровнях;
- Функция *bar*, помогающая сделать красиво оформленную шкалу (для опыта, здоровье и т.д.);
- Запуск самой игры путём вызова основного класса *Player*.

Заключение

В результате работы была создана текстовая RPG-игра на Python. В ней есть основные элементы: создание персонажа, бои с врагами, сбор предметов и повышение уровня. Игра использует объектно-ориентированное программирование – вся логика разделена на классы *Player* и *Enemy*.

Все запланированные функции реализованы: случайные события, пошаговые бои, инвентарь с эффектами и система опыта. Баланс игры построен так, что сложность меняется в зависимости от удачи игрока. Проект полностью работоспособен и может быть легко расширен новыми врагами, предметами или механиками. Цель – написать полноценную текстовую RPG – достигнута.