Введение

Игровая разработка является одним из наиболее перспективных направлений в области IT. Данный проект направлен на создание двухмерной игры с использованием языка программирования Python и библиотеки Pygame. Проект объединяет современные подходы к проектированию графического интерфейса, работы с мультимедиа и интерактивным взаимодействием с пользователем.

Pygame — это библиотека для разработки игр, которая предоставляет широкий спектр инструментов для работы с графикой, звуком и пользовательскими событиями. В рамках проекта особое внимание было уделено адаптивности интерфейса, взаимодействию с конфигурационными файлами и управлению игровыми событиями.

Цель проекта Цель проекта — разработка функциональной и адаптивной двухмерной игры с использованием библиотеки Pygame, которая будет включать в себя графический интерфейс, управление событиями и настройку параметров игры через конфигурационные файлы.

Основные задачи

1. Реализовать динамический игровой процесс, включая анимацию и взаимодействие объектов.
2. Разработать адаптивный интерфейс, учитывающий изменение размеров окна игры.
3. Внедрить управление через конфигурационные файлы JSON.
4. Интегрировать музыкальное сопровождение и звуковые эффекты.

Используемые технологии

* Python — основной язык разработки.
* Pygame — библиотека для работы с графикой и звуком.
* JSON — для хранения настроек игры.
* Библиотека datetime — для работы со временем и временными метками.

Глава 1. ХОД РАБОТЫ

1.1 Разработка функционала игры

Основной функционал включает:

* Реализацию игрового цикла с частотой обновления кадров 30 FPS.
* Настройку масштабируемого фона, учитывающего размеры окна.
* Обработку пользовательских событий (нажатие кнопок, движения мыши и т.д.).

1.2 Настройка графического интерфейса

Графический интерфейс был разработан с учётом адаптивности:

* Фоновые изображения автоматически изменяются под разрешение окна.
* Используются масштабируемые шрифты и кнопки, сохраняющие пропорции при изменении размеров окна.
* Основное меню игры включает элементы управления: кнопки «Новая игра», «Продолжить» и «Настройки».

1.3 Работа с событиями и управлением

События в игре обрабатываются через цикл, работающий с библиотекой Pygame. Примеры событий:

* Нажатие левой кнопки мыши вызывает проигрывание звукового эффекта.
* Изменение размеров окна автоматически обновляет размеры всех графических элементов.
* Перемещение мыши по экрану вызывает динамическое смещение фона, создавая эффект глубины.

1.4 Взаимодействие с конфигурационными файлами

Для хранения параметров игры используется файл config.json. В нём хранятся:

* Размеры окна.
* Уровень игры.
* Параметры графического интерфейса.

Пример структуры файла config.json:

{

"settings": {

"size": [800, 600],

"game": "1"

}

}

Конфигурация загружается и сохраняется через методы load\_config() и save\_config(), что позволяет сохранять состояние игры между сессиями.

Глава 2. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

2.1 Сравнение проекта с аналогами

Для оценки качества проекта были изучены аналогичные решения. Критерии сравнения:

* Адаптивность интерфейса.
* Качество обработки событий.
* Возможности кастомизации.

Вывод: проект выделяется удобством адаптации к разным разрешениям экрана и интуитивным управлением через JSON.

2.2 Потенциальные улучшения проекта

1. Добавить поддержку нескольких языков интерфейса.
2. Реализовать сохранение прогресса игры.
3. Улучшить графическую составляющую, включая анимацию персонажей.
4. Добавить сетевую функциональность для многопользовательской игры.

Заключение

Разработанный проект демонстрирует основные возможности библиотеки Pygame и Python для создания интерактивных приложений. Использование конфигурационных файлов и адаптивного интерфейса делает приложение универсальным и удобным для дальнейшего расширения.