МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Измерительно-вычислительные комплексы»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | |  | |  |  |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | |  | | Руководство программиста  на лабораторную работу №8  по дисциплине «Алгоритмы и структуры данных»  Тема «Разработка компьютерной игры PVP Arena» | | | | | | |
|  | | |  | |  | | | |
|  | | Исполнитель  студент гр. ИСТбд-21  Казаров Д.С.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | | |
|  | | 2024 | | | | | | |

**1. Введение**

Это руководство предназначено для программистов, которые будут поддерживать или расширять функциональность пошаговой игры, разработанной с использованием Python, Pygame и Tkinter. В руководстве описываются основные архитектурные решения, структура проекта и порядок его запуска.

**2. Среда разработки**

Для работы с проектом требуется установить следующие инструменты:

* **Python 3.x**: основной язык программирования.
* **Pygame**: библиотека для разработки игр. Устанавливается с помощью pip install pygame.
* **Tkinter**: библиотека для создания графических окон. Обычно включена в дистрибутив Python.

Для запуска игры достаточно поддерживать актуальные версии Python и указанных библиотек.

**3. Структура проекта**

Проект разделен на несколько файлов и классов:

1. **main\_menu.py**: отвечает за главное меню игры. Здесь реализованы функции для отображения кнопок, таких как "Начать игру", "Инструкции" и "Выход".
2. **game.py**: содержит основной игровой цикл, логику персонажей, их движение, атаки, а также взаимодействие с интерфейсом.
3. **character**: содержит классы персонажей (например, Warrior, Mage, Archer), описывающие их свойства и методы.
4. **enemy\_ai**: содержит логику искусственного интеллекта для врагов, реализуя управление их ходами.

**4. Описание основных классов и модулей**

1. **main\_menu.py**
   * **Описание:** Этот файл управляет главной точкой входа в игру. После завершения игры с помощью оконных уведомлений возвращает игрока в главное меню.
   * **Основные функции:**
     + create\_main\_menu(): создает главное меню с кнопками.
     + start\_game(): запускает игровую сессию.
     + show\_instructions(): отображает инструкции для игрока.
2. **game.py**
   * **Описание:** Основной файл с игровым процессом. Содержит главные функции управления персонажами, их атаками, взаимодействие с картой, а также интерфейс для отображения информации.
   * **Основные функции:**
     + initialize\_game(): инициализация игрового поля, спавн персонажей и препятствий.
     + game\_loop(): основной игровой цикл, управляющий ходами игрока и противника.
     + handle\_player\_move(): обработка перемещений игрока.
     + handle\_enemy\_move(): логика ИИ для врагов.
     + check\_win\_conditions(): проверяет условия победы или поражения.
3. **character**
   * **Описание:** Основные классы для игровых персонажей. Каждый персонаж имеет свои атрибуты, такие как здоровье, урон, радиус атаки и передвижения.
   * **Классы:**
     + Character: базовый класс персонажей, от которого наследуются все остальные классы.
     + Warrior: персонаж ближнего боя с высокой выносливостью.
     + Mage: персонаж дальнего боя с магическими атаками.
     + Archer: персонаж дальнего боя с высокой скоростью атаки.
4. **enemy\_ai**
   * **Описание:** Файл с логикой искусственного интеллекта. ИИ контролирует врагов, управляя их движением и атаками в зависимости от положения игрока на карте.
   * **Основные функции:**
     + enemy\_move(): выбирает ближайшего персонажа игрока и перемещает врагов в его сторону.
     + enemy\_attack(): если игрок находится в пределах радиуса атаки, враг наносит удар.

**5. Детали реализации**

1. **Отрисовка персонажей и игрового поля**
   * Отрисовка клеток поля и персонажей реализована через Pygame. Для каждого персонажа на поле создается объект, который может перемещаться и взаимодействовать с другими объектами.
   * Здоровье персонажей отображается как HP-бар над их фигурой.
2. **Логика перемещения и атаки**
   * Персонажи могут перемещаться по сетке поля, количество шагов ограничено для каждого типа персонажа. Радиус атаки зависит от типа персонажа и отображается в информационной панели.
3. **Интерфейс с использованием Tkinter**
   * Важные уведомления, такие как завершение игры или вывод инструкций, реализованы через tkinter.
   * После завершения игры окно уведомления позволяет вернуться в главное меню.

**6. Расширение функциональности**

1. **Добавление новых персонажей** Чтобы добавить нового персонажа, необходимо:
   * Создать новый класс, наследующий от Character, и определить его характеристики (HP, урон, радиус атаки и т.д.).
   * Добавить новый персонаж в список доступных для спавна в game.py.
2. **Модификация логики ИИ** Для изменения поведения ИИ можно корректировать алгоритмы в файле enemy\_ai, например, добавить более сложные стратегии для атаки и обороны.
3. **Добавление новых препятствий** Для изменения карты поля (например, добавления новых препятствий) нужно обновить функцию генерации карты в game.py, чтобы новые объекты правильно отображались на игровом поле.

**7. Тестирование и отладка**

1. **Основные тесты:**
   * Тестирование правильности перемещения персонажей.
   * Проверка логики атаки в зависимости от радиуса.
   * Проверка отображения победы/поражения.
2. **Отладка:** Для отладки можно использовать стандартные средства Python (print() для логирования) и инструменты IDE (например, PyCharm или VS Code), такие как точки останова (breakpoints) и пошаговое выполнение кода.

**8. Заключение**

Это руководство описывает основные аспекты проекта и помогает новым программистам понять структуру кода, его функциональность и методы расширения.