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*** |  | |  | | **Пояснительная записка**  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 | | | | | | |

**Введение**

Разрабатываемое игровое приложение называется **"PvP Arena"** и представляет собой пошаговую стратегию, в которой игрок управляет командой персонажей с различными характеристиками, целью игры является уничтожение вражеской команды, управляемой компьютером (AI). Приложение разработано с использованием библиотеки Pygame и языка программирования Python.

Выбор данной технологии обусловлен широкими возможностями Pygame для реализации 2D-графики, а также гибкостью Python, что позволяет реализовать сложные игровые механики и алгоритмы. Приложение демонстрирует основные навыки программирования, разработки пользовательских интерфейсов и организации взаимодействия между персонажами в игровом пространстве.

**1. Проектная часть**

**1.1. Постановка задачи на разработку приложения**

Определяется заданием на курсовую работу. Детализируется в разработанном техническом задании (приложение 1).

**1.2. Математические методы**

Математическая модель, используемая в приложении, основывается на системе координат для отображения игрового поля и расчётов расстояний между персонажами. Для перемещения персонажей и определения возможности атак используются стандартные методы работы с векторной геометрией. Данный подход выбран ввиду простоты реализации в двумерной плоскости и возможности легко расширять функционал игры. Также модель основана на методах теории графов для представления поля в виде сетки с ограничениями на движение.

**1.3. Архитектура и алгоритмы**

**1.3.1 Архитектура**

Архитектура приложения включает следующие основные компоненты:

* **Игровое поле** – сетка, представляющая собой игровую карту, с координатами для размещения персонажей и препятствий.
* **Персонажи** – объекты, которые содержат информацию о текущем состоянии (HP, атака, радиус атаки, тип атаки).
* **Обработка событий** – основной цикл игры, в котором отслеживаются действия игрока и обновляются состояния объектов.
* **AI** – простая модель искусственного интеллекта для управления действиями команды противника.
* **Информационная панель** – отображает данные о выбранных персонажах, их характеристиках и доступных действиях.

**1.3.2 Алгоритм движения персонажей**

Алгоритм отвечает за перемещение персонажей по игровому полю:

1. Игрок выбирает персонажа.
2. Проверяется количество оставшихся шагов.
3. Проверяется, не занята ли целевая клетка другим персонажем или препятствием.
4. Если условия соблюдены, персонаж перемещается, а число оставшихся шагов уменьшается.
5. Обновляется визуальное отображение персонажа на игровом поле.

**1.3.3 Алгоритм атаки**

1. Игрок выбирает цель для атаки.
2. Определяется расстояние между атакующим и целью.
3. Если цель находится в пределах радиуса атаки, происходит расчет урона.
4. Если урон больше 0, у цели уменьшается здоровье (HP).
5. Проверяется состояние цели – если здоровье достигло 0, цель удаляется с поля.

**1.4. Тестирование**

**1.4.1 Описание отчета о тестировании**

Отчёт о тестировании включает результаты проверки всех функций игры, включая корректную работу перемещения, атак, отображения данных и взаимодействие с искусственным интеллектом.

**1.4.2 Цель тестирования**

Цель тестирования – проверка корректности работы игровых механик, соответствие игры заявленным требованиям, а также поиск и исправление ошибок, которые могут возникнуть в ходе игрового процесса.

**1.4.3 Методика тестирования**

Тестирование проводилось методом черного ящика, где проверялись функциональные возможности игры и их соответствие техническому заданию. Каждая функция тестировалась на различных игровых сценариях и действиях игрока.

**1.4.4 Проведенные тесты**

1. Тестирование перемещения персонажей: проверка возможности хода на корректные и некорректные клетки.
2. Тестирование атак: проверка выполнения атак в пределах радиуса и невозможности атаки за его пределами.
3. Тестирование взаимодействия с препятствиями: проверка невозможности пересечения препятствий персонажами.
4. Тестирование работы AI: проверка корректности выполнения действий компьютером.

**1.4.5 Выводы**

По итогам тестирования выявлены и исправлены следующие проблемы:

* Ошибки в логике перемещения персонажей и их взаимодействии с препятствиями.
* Проблемы с отображением информации на панели характеристик после смерти персонажей.

Рекомендации:

* Оптимизация кода для улучшения производительности.
* Добавление более сложных моделей поведения AI.

**2. Источники, использованные при разработке**

1. Пирогов, А.А. Python для начинающих. Изучаем программирование с нуля. – М.: Издательство "БХВ-Петербург", 2020.
2. "Документация Pygame" [Электронный ресурс] // Официальный сайт библиотеки Pygame: https://www.pygame.org/docs/ – Режим доступа: свободный.
3. ГОСТ 19.701-90 «Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения». – М.: Издательство стандартов, 1990.
4. В.В. Шишкин, Д.С. Афонин РАЗРАБОТКА ЛОГИЧЕСКИХ КОМПЬЮТЕРНЫХ ИГР С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ В СРЕДЕ ПИТОН