Документация для разработчика: Игра "Сапёр"

1. Введение

Игра "Сапёр" — это логическая компьютерная игра, в которой игрок открывает клетки на игровом поле, избегая взрывов мин. Документация для разработчика предназначена для понимания структуры и особенностей реализации игры "Сапёр".

Требования к техническим ресурсам пользователя:

* Минимальные требования: монитор разрешением 1920\*1080(Full HD), ориентация: горизонтальная, соотношение сторон 21:16, оконная.

Должна присутствовать анимация в данном окне при выборе уровня сложности => анимирование создания сетки игры

2. Структура проекта

- сапёр.py - основной файл программы, содержащий логику игры, интерфейс и управление.

3. Технологии

- Игра разработана на языке программирования Python с использованием библиотеки Tkinter для создания графического интерфейса.

- Для модуля random импорт choice

- Дополнительный import time

4.1 Установка обязательного модуля

- Установите библиотеки Tkinter и необходимые модули (random, choice, time), используя менеджер пакетов pip.

5. Использование программного продукта

- После запуска игры выберите уровень сложности, нажав на соответствующую кнопку.

- Левым кликом мыши открывайте клетки на игровом поле.

- Правым кликом мыши помечайте клетки, где, по вашему мнению, могут быть мины.

- Цель игры - открыть все клетки игрового поля без мин и избежать клеток с минами.

6. Основные функции и методы

- play(n) - обработка клика игрока на клетку, проверка условий победы/поражения.

- mark(n) - метод пометки клеток мины правым кликом мыши.

- new\_game() - начало новой игры с выбранным уровнем сложности.

- set\_level() - установка выбранного уровня сложности игры.

7. Тестирование

- Для тестирования игры необходимо использовать модульное тестирование для проверки функциональности отдельных компонентов (Каскадная модель).

Вид тестирования – белый ящик, нефункциональное, специализированное

- Необходимо использовать итеративное тестирование для проверки игры в различных сценариях и на разных уровнях сложности.

Сценарий итеративного тестирования:

1. Внесите небольшое изменение в продукт.
2. Проведите тест, чтобы увидеть, достигла ли цель изменения.
3. Соберите обратную связь и на её основе необходимо осуществить её для информирования будущих итераций.

Тестирование проводится на усмотрение специалиста по тестированию данного продукта (ручное или автоматизированное по указанным выше параметрам)

На основе A тестирования

8. Разработка и обновление

- При разработке новых функций или исправлении ошибок следуйте структуре проекта и поддерживайте однородность стиля кода.

- Для обновления игры необходимо выпускать (периодичность 3 месяца) системные пакеты с дополнительными возможностями и корректировками, а конкретно: выпуск дополнительных или новых модулей, в зависимости от отзывов или плана изменения программного продукта.