**Разработка мини-проекта: Игра 2048**

**Тематика проекта:**

Разработка игры 2048 с использованием Python3, PyQt5 и PostgreSQL для хранения рекордов игроков.

**Цель проекта:**

Создание и тестирование приложения, которое реализует игру 2048, включая создание технического задания, макета интерфейса, структуры проекта, использование Git для версионного контроля, написание тест-кейсов с помощью PyTest и документирование проекта.

**Требования:**

1. Техническое задание:

- Цели проекта: Разработка интерактивной игры 2048.

- Функциональные требования:

- Игровое поле размером 4x4.

- Возможность перемещения плиток в четырех направлениях (вверх, вниз, влево, вправо).

- Объединение плиток с одинаковыми значениями.

- Учет очков и отображение текущего рекорда.

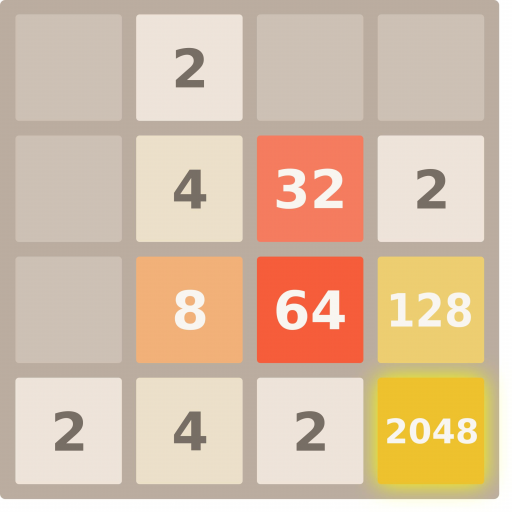
- Сохранение рекордов в базе данных.

- Нефункциональные требования:

- Высокая производительность и отзывчивость интерфейса.

- Кроссплатформенность (Windows, Linux).

**2. Макет приложения:**



- Игровое поле.

- Кнопки управления (вверх, вниз, влево, вправо).

- Отображение текущих очков и рекорда.

**3. Структура проекта:**

1. draw\_buttons()

Эта функция отвечает за отрисовку кнопок управления на экране. В игре предусмотрены две кнопки: "Рестарт" и "Отменить".

Рисование кнопок:

Используется метод pygame.draw.rect() для создания прямоугольников кнопок с серым фоном.

Добавление текста:

Текст кнопок "Рестарт" и "Отменить" создается с помощью button\_font.render() и отображается на соответствующих кнопках с помощью screen.blit().

1. draw\_over()

Функция отображает экран окончания игры.

Рисование фона:

Создается черный прямоугольник, который занимает центральную часть экрана, чтобы выделить сообщение об окончании игры.

Отображение текста:

С помощью font.render() создается текст "Игра окончена!" и выводится на экран в белом цвете.

1. take\_turn(direc, board)

Эта функция обрабатывает ход игрока в зависимости от направления движения плиток (вверх, вниз, влево или вправо).

Сохранение предыдущего состояния:

Сначала сохраняется текущее состояние игрового поля в переменной previous\_board.

Логика перемещения и объединения плиток:

В зависимости от направления (UP, DOWN, LEFT, RIGHT), плитки перемещаются и объединяются:

Для каждого направления реализована своя логика: проверка на наличие пустых клеток и возможность объединения плиток с одинаковыми значениями.

Обновляется счет, если плитки объединяются.

Возвращаемое значение:

Функция возвращает обновленное состояние игрового поля.

1. new\_pieces(board)

Функция добавляет новые плитки на игровое поле.

Проверка наличия пустых клеток:

Функция продолжает добавлять новые плитки (2 или 4) до тех пор, пока есть пустые клетки на поле.

Случайный выбор позиции:

Случайным образом выбирается позиция для новой плитки. Если клетка пуста, то в ней появляется плитка со значением 2 (в 90% случаев) или 4 (в 10% случаев).

Возвращаемое значение:

Функция возвращает обновленное игровое поле и флаг, указывающий, заполнено ли поле полностью.

1. draw\_board()

Эта функция отвечает за отрисовку игрового поля и отображение текущего счета.

Рисование фона игрового поля:

Создается прямоугольник с цветом фона для игрового поля.

Отображение счета и рекорда:

С помощью font.render() создаются текстовые строки для текущего счета и рекорда, которые затем выводятся на экран с помощью screen.blit().

Эти функции являются ключевыми для обеспечения функциональности игры 2048, обеспечивая взаимодействие пользователя с игрой через графический интерфейс и управление логикой игры.

**4. Использование Git:**

- Настроить репозиторий Git для управления версиями проекта.

- Регулярно коммитить изменения с осмысленными сообщениями (например, "Добавлена логика перемещения плиток").

**5. Разработка приложения:**

- Использовать Python3 и PyQt5 для создания графического интерфейса приложения.

- Написать логику приложения в модуле `game\_logic.py`.

- Использовать PostgreSQL для хранения рекордов игроков.

- При необходимости использовать дополнительные библиотеки (например, NumPy для обработки данных).

**6. Тестирование:**

- Написать тест-кейсы для проверки функциональности приложения:

- Проверка корректности перемещения плиток.

- Проверка объединения плиток.

- Использовать PyTest для автоматизации тестирования.

**7. Модульность:**

- Соблюдать принципы модульности при разработке, разделяя код на логические модули.

**8. Документация:**

Введение

Игра 2048 — это простая, но увлекательная логическая игра, в которой игроки перемещают плитки на квадратном поле, чтобы объединять плитки с одинаковыми значениями и достигнуть плитки с числом 2048. Эта документация описывает установку, использование и разработку игры.

Установка

Требования

Python 3.x

Библиотека Pygame

Установка Pygame

Для установки Pygame используйте pip:

pip install pygame

Запуск игры

Скачайте исходный код игры и выполните команду:

python main.py

Управление

Стрелка вверх: перемещение плиток вверх.

Стрелка вниз: перемещение плиток вниз.

Стрелка влево: перемещение плиток влево.

Стрелка вправо: перемещение плиток вправо.

Кнопка "Рестарт": сброс игры и начало новой партии.

Кнопка "Отменить": возврат к предыдущему состоянию игрового поля.

Структура проекта

Основные файлы

main.py: Точка входа в игру, основной игровой цикл.

game\_logic.py: Логика игры, включая перемещение и объединение плиток.

gui.py: Графический интерфейс пользователя, отрисовка элементов на экране.

database.py: Взаимодействие с базой данных для хранения рекордов (при необходимости).

config.py: Конфигурационные параметры игры.

Основные функции

* draw\_buttons()-отрисовывает кнопки "Рестарт" и "Отменить" на экране.
* draw\_over()-отображает сообщение об окончании игры.
* take\_turn(direc, board)-обрабатывает ход игрока в зависимости от направления движения плиток.
* new\_pieces(board)-добавляет новые плитки на игровое поле.
* draw\_board()-отрисовывает игровое поле и отображает текущий счет и рекорд.
* draw\_pieces(board)-отрисовывает плитки на игровом поле в зависимости от их значений.

Использование

Начало игры

При запуске игры вы увидите пустое игровое поле 4x4 с двумя случайно размещенными плитками (2 или 4). Цель — объединять плитки с одинаковыми значениями, перемещая их по полю.

Игра

Игроки могут использовать клавиши стрелок для перемещения плиток. При каждом перемещении новые плитки добавляются на поле, а счет обновляется. Игра продолжается до тех пор, пока не будет достигнута плитка 2048 или не останется возможных ходов.

Сохранение рекордов

Рекорды сохраняются в текстовом файле high\_score.txt, который обновляется при достижении нового рекорда.

Заключение

Игра 2048 — это не только развлечение, но и отличная возможность для улучшения логического мышления. Мы надеемся, что данная документация поможет вам разобраться в игре и насладиться процессом!

**9. Сборка проекта:**

- Собрать проект в exe файл с помощью PyInstaller для удобства распространения.

**10. Жизненный цикл программного обеспечения:**

- Описать жизненный цикл программного обеспечения в отдельном документе и добавить его в документацию на GitHub.

**11. Размещение на GitHub:**

- Загрузить проект на GitHub, включая:

- Исходный код.

- Тесты.

- Документацию.

- Собранный exe файл.