**Игра-Тетрис «Block Breakers» на языке программирования python с использованием pygame**

**Парахин Александр и Соколович Никита**

* **Идея проекта:**
  + Создание классической игры "Тетрис" с использованием библиотеки Pygame.
  + Цель: разработка игрового приложения для развлечения и изучения основ программирования на Python и библиотеки pygame.
* **Задачи проекта:**
  + Реализация игровой логики (падение фигур, управление, подсчет очков).
  + Создание интуитивно понятного интерфейса.
  + Добавление дополнительных функций (пауза, правила игры, уровни сложности).
* **Геймплей:**
  + Управление фигурами с помощью клавиатуры (стрелки, пробел, Enter).
  + Динамическое увеличение сложности с ростом уровня.
  + Подсчет очков и отображение следующей фигуры.
  + Сохранение и просмотр рекордов игр
* **Визуальные элементы:**
  + Цветовая палитра для фигур и фона, скины и фон с изображением.
  + Эффекты 3D для текста и блоков.
  + Анимация падения фигур.
* **Дополнительные функции:**
  + Пауза и возобновление игры.
  + Адаптивный интерфейс
  + Отображение правил игры.
  + Отображение рекордов, их сохранение и сброс
  + Возможность включения музыки
  + Возможность мгновенного падения фигуры.
* **Выводы по работе:**
  + Проект успешно реализован, все основные функции работают корректно.
  + Игра обладает интуитивно понятным интерфейсом и увлекательным геймплеем.
  + Проект демонстрирует возможности Python и Pygame для создания 2D-игр.
* **Возможности для доработки:**
  + Добавление новых типов фигур и уровней сложности.
  + Реализация мультиплеера (игра на двоих).
  + Интеграция системы рекордов и онлайн-лидерборда.
  + Улучшение графики (анимации, эффекты, звуки).

Пояснение

**Пояснительная записка к проекту "Block Breakers"**

**Введение**

В данной пояснительной записке представлена информация о проекте "Block Breakers", который реализует классическую игру "Тетрис" на языке программирования Python с использованием библиотеки Pygame. Проект разработан студентами Парахиным Александром и Соколовичем Никитой с целью создания развлекательного игрового приложения и изучения основ программирования в контексте разработки игр.

**Цели и задачи проекта**

**Цель проекта**

Основной целью проекта является создание интуитивно понятной и увлекательной игры "Тетрис", которая позволит пользователям не только развлекаться, но и изучать основы программирования на Python и работу с библиотекой Pygame.

**Задачи проекта**

Для достижения поставленной цели были определены следующие задачи:

1. Реализация игровой логики:
   * Падение фигур.
   * Управление фигурами.
   * Подсчет очков.
2. Создание интуитивно понятного интерфейса для пользователя.
3. Добавление дополнительных функций, таких как:
   * Пауза.
   * Правила игры.
   * Уровни сложности.
   * Звуки и скины
   * Таблица лидеров
   * Ввод имени игрока

**Описание функций**

**Основные функции**

**Эти функции отвечают за ключевые аспекты игры: управление игровым процессом, отрисовку интерфейса, обработку событий и логику игры.**

1. **main()**
   * **Главная функция, которая инициализирует игру, загружает ресурсы и запускает основной игровой цикл.**
2. **playTetris()**
   * **Основной игровой цикл, управляющий логикой игры: движение фигур, проверка столкновений, очистка линий, обновление счета и уровня.**
3. **showInitialText(text)**
   * **Отображает начальный текст (например, название игры или сообщение о паузе) и ждет нажатия клавиши для продолжения.**
4. **showGameRules()**
   * **Отображает правила игры на экране.**
5. **show\_score\_table()**
   * **Отображает таблицу рекордов.**
6. **game\_over(score)**
   * **Обрабатывает завершение игры: добавляет результат в таблицу рекордов и сортирует её.**
7. **start\_game()**
   * **Запрашивает имя игрока перед началом игры.**
8. **get\_player\_name()**
   * **Отображает поле ввода для имени игрока и возвращает введенное имя.**
9. **draw\_reset\_scores\_button(is\_clicked=False)**
   * **Рисует кнопку для сброса таблицы рекордов.**
10. **draw\_score\_button(is\_clicked=False)**
    * **Рисует кнопку для открытия таблицы рекордов.**
11. **draw\_sound\_button(is\_playing=False)**
    * **Рисует кнопку для переключения звука.**
12. **draw\_save\_button(is\_clicked=False)**
    * **Рисует кнопку для сохранения результата игры.**
13. **save\_score(name, score)**
    * **Сохраняет имя игрока и его балл в файл.**
14. **renderTitle()**
    * **Отображает заголовок игры на экране.**
15. **renderInfo(points, level)**
    * **Отображает информацию о текущем счете и уровне.**
16. **renderNextFigure(fig)**
    * **Отображает следующую фигуру, которая появится на поле.**
17. **drawCup(cup)**
    * **Рисует игровое поле (стакан) на экране.**
18. **renderFigure(fig, pixel\_x=None, pixel\_y=None)**
    * **Рисует текущую фигуру на экране.**
19. **animate\_fall(fig)**
    * **Анимирует падение фигуры.**
20. **flash\_block(block\_x, block\_y, color)**
    * **Мигает блоком на экране (используется для визуального эффекта при очистке линии).**

**Вспомогательные функции**

**Эти функции выполняют вспомогательные задачи: помогают в отрисовке, обработке событий, проверке условий и других технических аспектах.**

1. **reset\_scores()**
   * **Очищает таблицу рекордов.**
2. **displayPauseScreen()**
   * **Отображает экран паузы с полупрозрачным фоном.**
3. **render3DText(text, font, color, x, y)**
   * **Рисует текст с эффектом 3D (тенью).**
4. **load\_sprites()**
   * **Загружает изображения и звуки для игры.**
5. **createTextObject(text, font, color)**
   * **Создает объект текста для отображения на экране.**
6. **terminateGame()**
   * **Завершает игру и закрывает окно.**
7. **checkForKeyPress()**
   * **Проверяет, была ли нажата какая-либо клавиша.**
8. **handleQuit()**
   * **Обрабатывает события выхода из игры (например, нажатие ESC или закрытие окна).**
9. **calculateSpeed(points)**
   * **Рассчитывает скорость падения фигур в зависимости от набранных очков.**
10. **generateNewFigure()**
    * **Генерирует новую случайную фигуру для игры.**
11. **addFigureToCup(cup, fig)**
    * **Добавляет фигуру в игровое поле.**
12. **createEmptyCup()**
    * **Создает пустое игровое поле.**
13. **isInCup(x, y)**
    * **Проверяет, находится ли координата (x, y) в пределах игрового поля.**
14. **isPositionValid(cup, fig, adjX=0, adjY=0)**
    * **Проверяет, может ли фигура находиться в текущей позиции или в позиции с учетом смещения.**
15. **isLineCompleted(cup, y)**
    * **Проверяет, заполнена ли линия на игровом поле.**
16. **clearFullLines(cup)**
    * **Очищает заполненные линии на игровом поле и возвращает количество очищенных линий.**
17. **convertBlockCoords(block\_x, block\_y)**
    * **Преобразует координаты блока в пиксельные координаты на экране.**
18. **drawBlockOnScreen(block\_x, block\_y, color, pixel\_x=None, pixel\_y=None)**
    * **Рисует блок на экране в указанных координатах.**
19. **drawBackground()**
    * **Рисует фон игры на экране.**

**Геймплей**

Игра управляется с помощью клавиатуры, где игрок может перемещать фигуры, вращать их и ускорять падение. Динамическое увеличение сложности происходит с ростом уровня, что делает игру более увлекательной. Подсчет очков осуществляется на основе удаленных линий, и игроку отображается следующая фигура для планирования ходов.

**Визуальные элементы**

Визуальные элементы включают:

1. **Фигуры**:
   * Фигуры в игре имеют различные формы и цвета(а также скин после падения), что помогает игрокам легко различать их. Каждая фигура может быть представлена в виде набора блоков, которые могут иметь разные текстуры или границы.
   * Использование 3D-эффектов при отрисовке фигур добавляет глубину и делает игру более динамичной.
2. **Анимация**:
   * Эффекты, такие как исчезновение заполненных линий, могут быть дополнены анимацией, что делает игровой процесс более увлекательным.
3. **Текст**:
   * Все текстовые элементы, такие как названия, подсказки и очки, оформлены с использованием шрифтов, которые хорошо читаются и соответствуют общей стилистике игры.
   * Эффекты 3D для текста, такие как тени, добавляют визуальной привлекательности и делают интерфейс более интересным.

**Интерфейс пользователя**

Интерфейс игры разработан с учетом удобства и интуитивности. Включает в себя:

1. **Игровое поле**:
   * Основное игровое поле четко выделено, с границами, чтобы игроки могли легко увидеть, где находятся фигуры и где они могут падать.
2. **Панели**:
   * Боковая панель может содержать информацию о текущей игре, такую как количество очков, уровень сложности и следующая фигура, кнопки таблицы рекордов и сохранения результата, кнопки управлением музыкой и сброса результатов. Это помогает игрокам планировать свои действия.
3. **Кнопки управления**:
   * Управление игрой осуществляется с помощью клавиатуры, и игроки могут видеть подсказки о том, какие клавиши использовать для управления фигурами.
4. **Меню**:
   * Игра включает меню для начала новой игры, просмотра правил, просмотра рекордов и выхода из игры, меню таблицы рекордов, меню ввода имени. Это меню интуитивно понятно и легко доступно.

**Дополнительные функции**

**Пауза и возобновление игры**

* Игра включает возможность паузы, что позволяет игрокам сделать перерыв, не теряя прогресс. При активации паузы на экране отображается соответствующий текст, и игра останавливается до тех пор, пока игрок не решит продолжить.

**Правила игры**

* В проекте предусмотрено отображение правил игры, что помогает новым игрокам быстро ознакомиться с основами. Правила могут быть представлены в виде текста на экране, доступного через меню.

**Таблица рекордов**

* В проекте предусмотрено отображение таблицы рекордов, что помогает новым игрокам быстро просмотреть все сохраненные рекорды..

**Ввод имени игрока**

* После нажатия любой кнопки при старте игры, запрашивается имя игрока, которое он может ввести в строку ввода на экране..

**Управление музыкой**

* На боковой панели можно включить или отключить музыку..

**Мгновенное падение фигур**

* Возможность мгновенного падения фигур позволяет игрокам ускорять процесс, что может быть полезно в напряженных ситуациях, когда необходимо быстро освободить пространство на игровом поле.

**Выводы**

Проект "Block Breakers" успешно реализован, и все основные функции работают корректно. Игра обладает интуитивно понятным интерфейсом и увлекательным геймплеем, что позволяет пользователям наслаждаться процессом. Реализация игры демонстрирует возможности языка Python и библиотеки Pygame для создания 2D-игр.

**Возможности для доработки**

Несмотря на успешную реализацию, проект имеет потенциал для дальнейшего развития. Возможные направления доработки включают:

1. **Добавление новых типов фигур и уровней сложности**:
   * Введение уникальных фигур с особыми свойствами или механиками может разнообразить игровой процесс. Уровни сложности могут включать разные скорости падения фигур и новые правила.
2. **Реализация мультиплеера**:
   * Возможность играть с друзьями или другими игроками онлайн может значительно увеличить интерес к игре.
3. **Интеграция системы рекордов и онлайн-лидерборда**:
   * Введение системы рекордов позволит игрокам соперничать друг с другом, что добавит элемент соревнования и мотивации.
4. **Улучшение графики**:
   * Работы над графикой могут включать создание более сложных анимаций, улучшение текстур и добавление звуковых эффектов, что сделает игру более захватывающей.

**Заключение**

Данная пояснительная записка призвана предоставить полное представление о проекте "Block Breakers", его целях, задачах и возможностях для дальнейшего развития. Проект не только демонстрирует технические навыки разработчиков, но и предлагает увлекательный опыт для пользователей. Если у вас есть дополнительные вопросы или нужно больше информации о конкретных аспектах проекта, пожалуйста, дайте знать!