

Игра «Тетрис»

Автор: Горбачева Зоя Евгеньевна

Введение

Идея проекта: создание своей версии игры «Тетрис», используя библиотеку pygame.

Задачи:

- Изучить предметную область, т. е. правила игры «Тетрис»;
- Выделить необходимый функционал;
- Реализовать выделенный функционал;
- Протестировать полученное приложение.

Правила игры «Тетрис»

- Существует прямоугольная область.
- На экране вверху этой области появляется фигура, которая состоит из одинаковых квадратов.
- Фигура периодически опускается вертикально в области.
- Игрок управляет фигурой: он может поворачивать ее и двигать по горизонтали.
- Когда фигура касается низа области или доходит до границы ранее установленной фигуры, ее положение фиксируется.
- Если фиксация положения новой фигуры заполнила полностью горизонтальную линию в области, то игрок получает очки, а линия исчезает из области, остальные линии смещаются.
- Если фиксация положения невозможна из-за того, что места для фигуры нет, то игра заканчивается.
- Появляется новая фигура и так далее, пока игра не закончится.

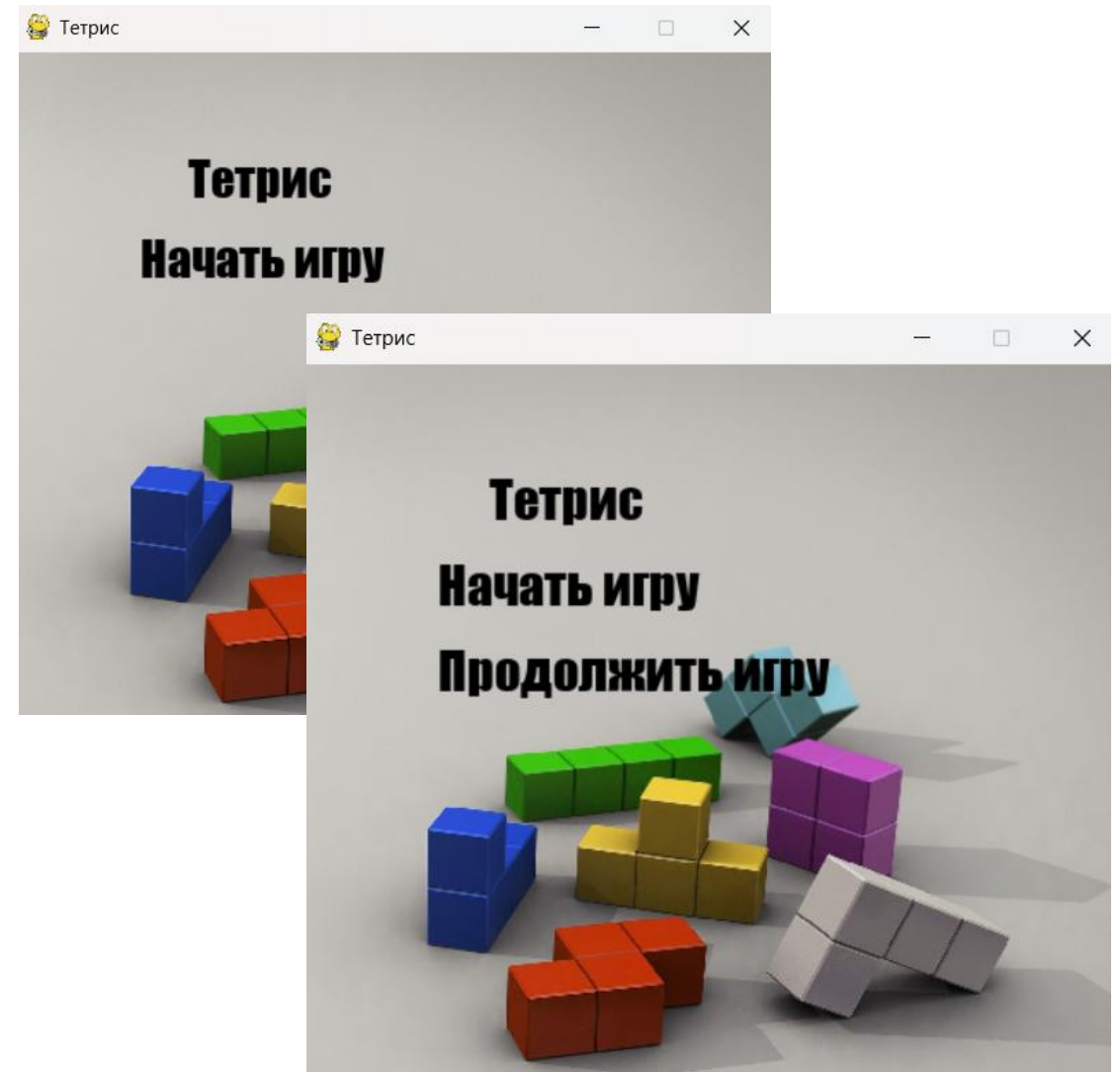
Необходимый функционал

- Выделение прямоугольной области на экране, выделение места для дополнительной информации.
- Создание фигуры со случайной формой и случайным цветом.
- Движение фигуры.
- Обработка действий игрока.
- Обработка заполнения линии, подсчет очков.
- Обработка завершения игры.
- Дополнительный функционал для удобства игрока: пауза, начальный экран, возможность рестарта, сохранение игры и т. п.

Реализация. Начальный экран

При открытии приложения пользователь видит стартовый экран с тематическим изображением, названием игры и надписью «Начать игру». Для начала игры нужно нажать на соответствующую надпись. Если есть файл сохранения, то на экране также присутствует надпись «Продолжить игру», ее можно нажать для загрузки сохранения.

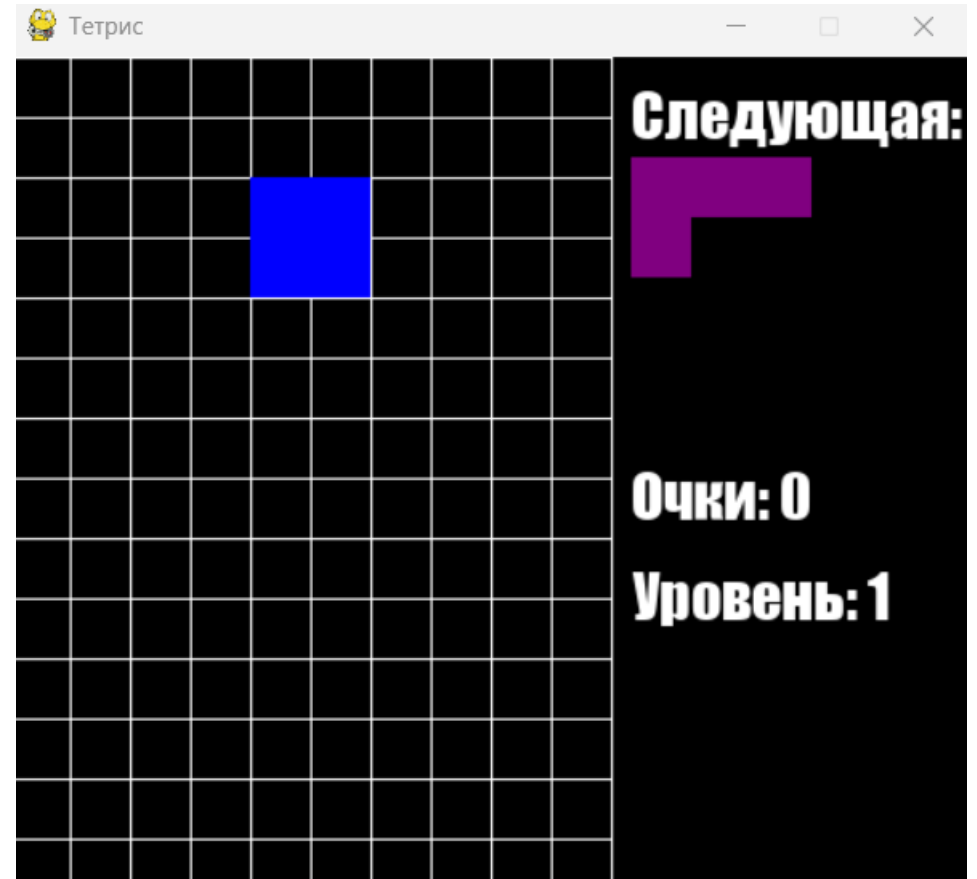
Для реализации была написана функция `start_screen`.



Реализация. Выделение областей

Была выделена прямоугольная область, где непосредственно происходит игра, а также область для дополнительной информации: показываем текущий уровень сложности, количество набранных очков и следующую фигуру, которая попадет в область.

Для реализации была написана функция `draw_grid` и часть кода из основной функции игры `game`.

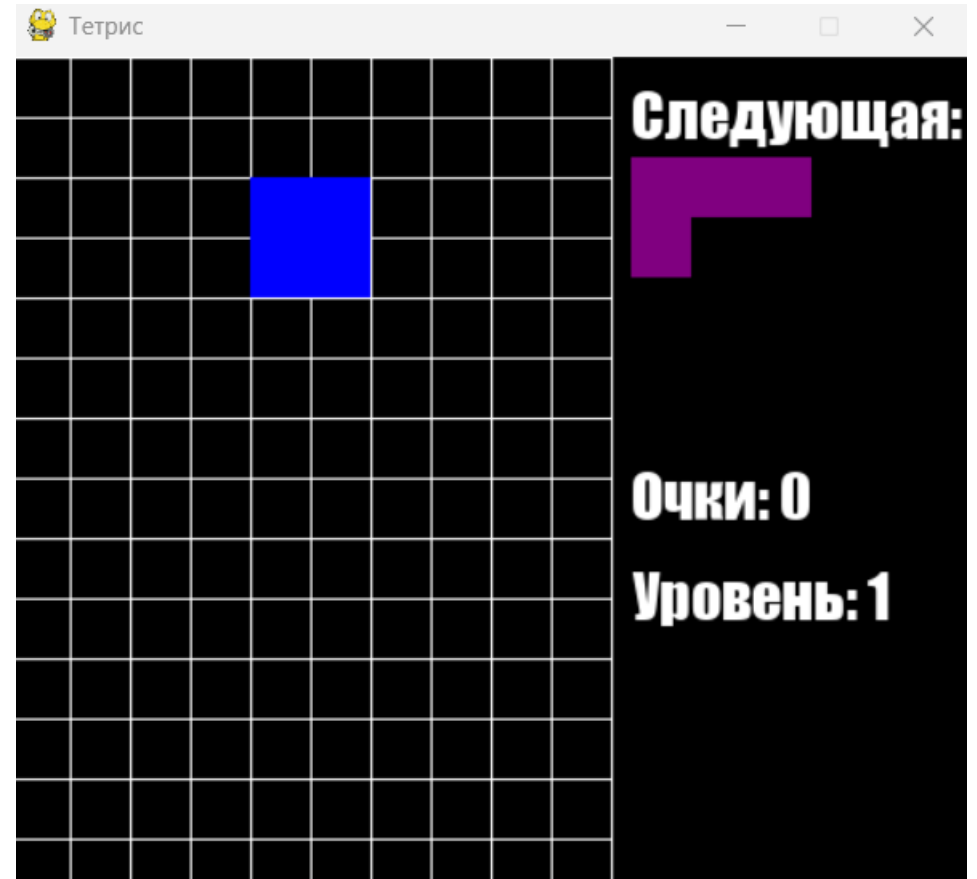


Реализация. Создание фигуры

Был создан список возможных форм и цветов для фигуры. Создаем фигуру, выбрав случайным образом цвет и форму. Добавляем ее на поле. Создаем фигуру, которая будет считаться «следующей».

Когда игрок зафиксировал текущую фигуру, «следующая» сменит ее и появится новая «следующая» в выделенном месте экрана. Повторяем до конца игры.

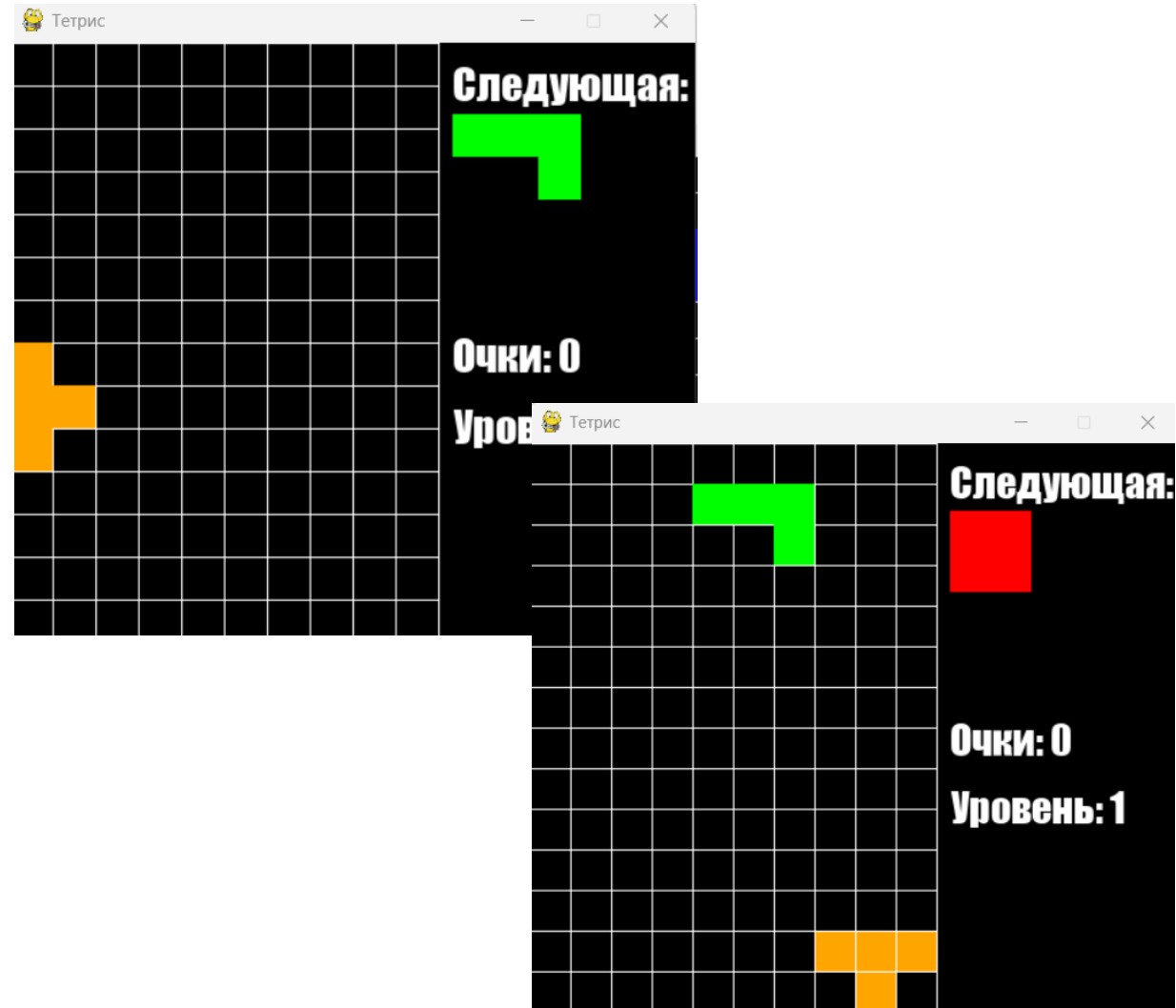
Для реализации был создан класс Piece и функция `add_to_grid`, также часть кода находится в основной функции `game`.



Реализация. Движение фигуры

Была реализована возможность поворота фигуры с помощью стрелки вверх. Есть проверка, что поворот фигуры не передвинет ее за пределы области. Также есть возможность двигать фигуру по горизонтали с помощью стрелок влево и вправо. Можно немного ускорять падение с помощью стрелки вниз.

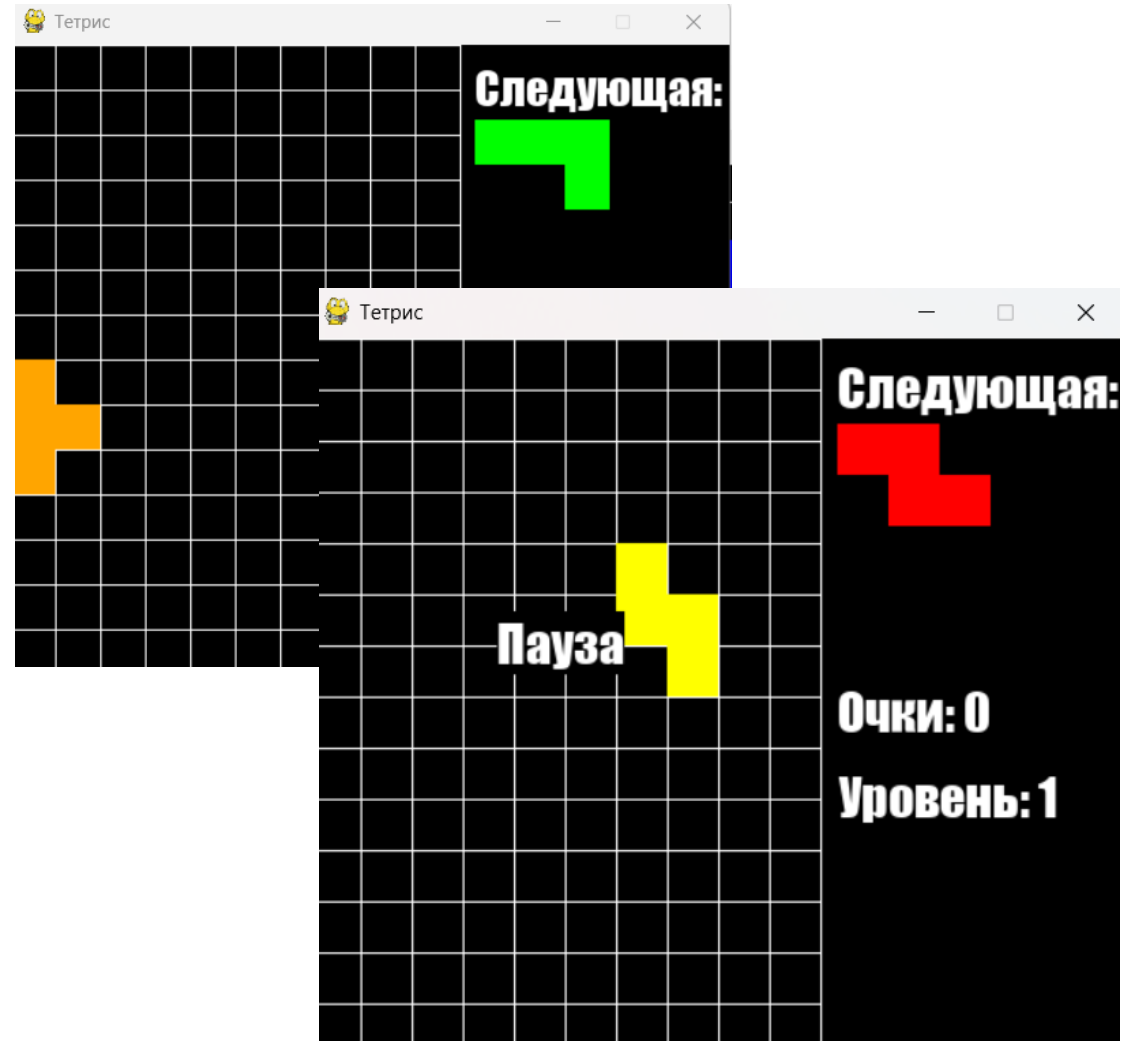
Для реализации был написан метод класса Piece rotate, часть кода находится в основной функции игры game.



Реализация. Обработка действий игрока

Игрок может нажимать на клавиши-стрелочки для движения фигуры. Нажатие на клавишу R(англ.) поставит игру на паузу. После проигрыша игрок может нажать на клавишу R для рестарта игры.

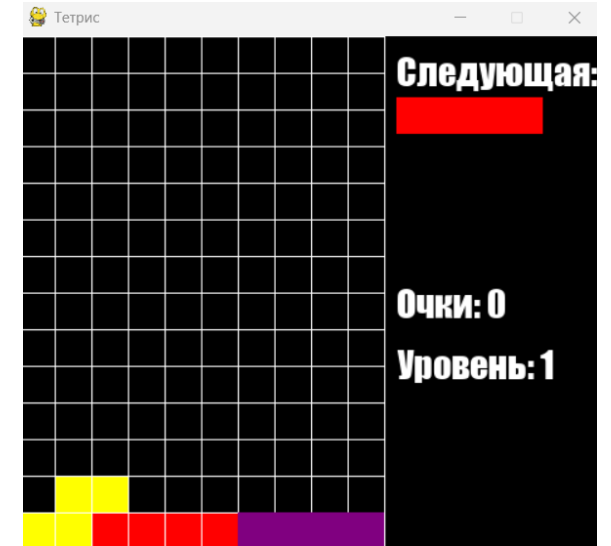
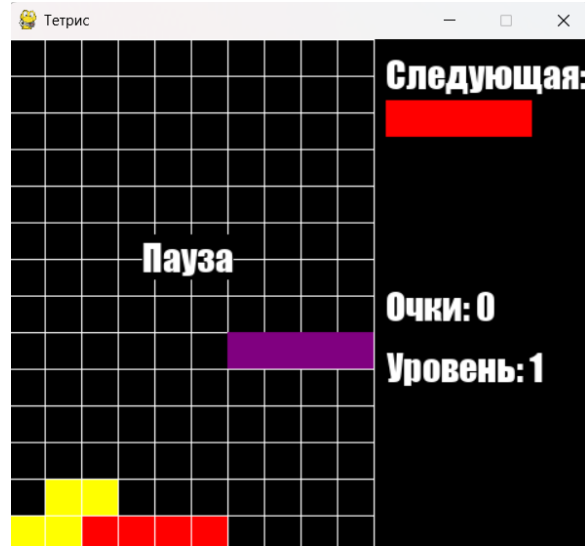
Код находится в основной функции игры game.



Реализация. Обработка заполнения линии, подсчет очков

После фиксации фигуры происходит проверка на заполнение горизонтальных линий области. Если линия заполнена, то она подсвечивается, а потом исчезает из области. Остальные линии смещаются вниз. К очкам добавляется 100 за каждую заполненную линию. К уровню добавляется единица.

Для реализации были написаны функции `clear_lines` и `is_collision`. Часть кода находится в основной функции игры `game`.

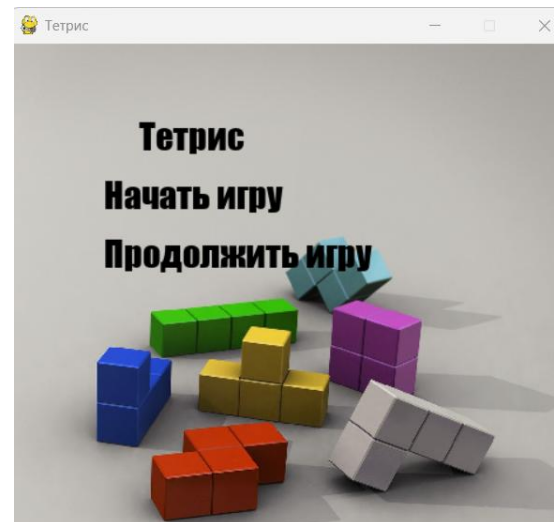
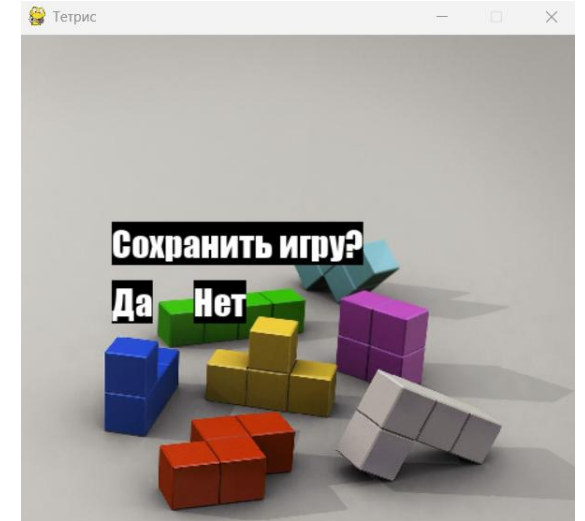
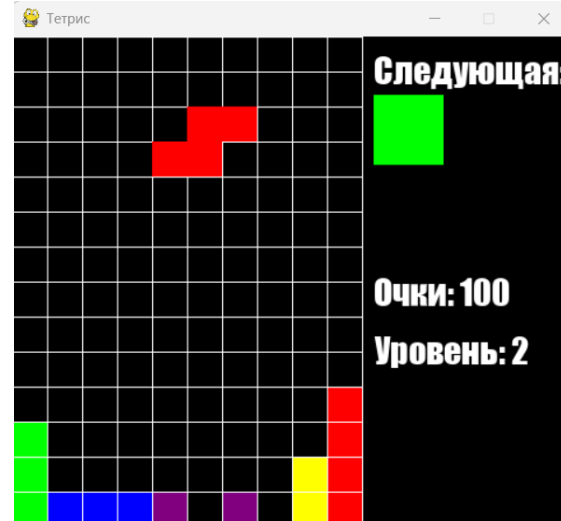


Реализация. Сохранение игры

При выходе из игры (нажатие на крестик) пользователю предлагается сохранить игру. При нажатии на надпись «Да» в текстовый файл будет сохранена вся информация о текущей игровой ситуации и игра закрывается. При нажатии на надпись «Нет» игра закрывается.

При новом запуске игры на стартовом экране можно нажать на надпись «Продолжить игру». Информация о сохраненной игровой ситуации будет взята из текстового файла. Игра продолжается по обычным правилам.

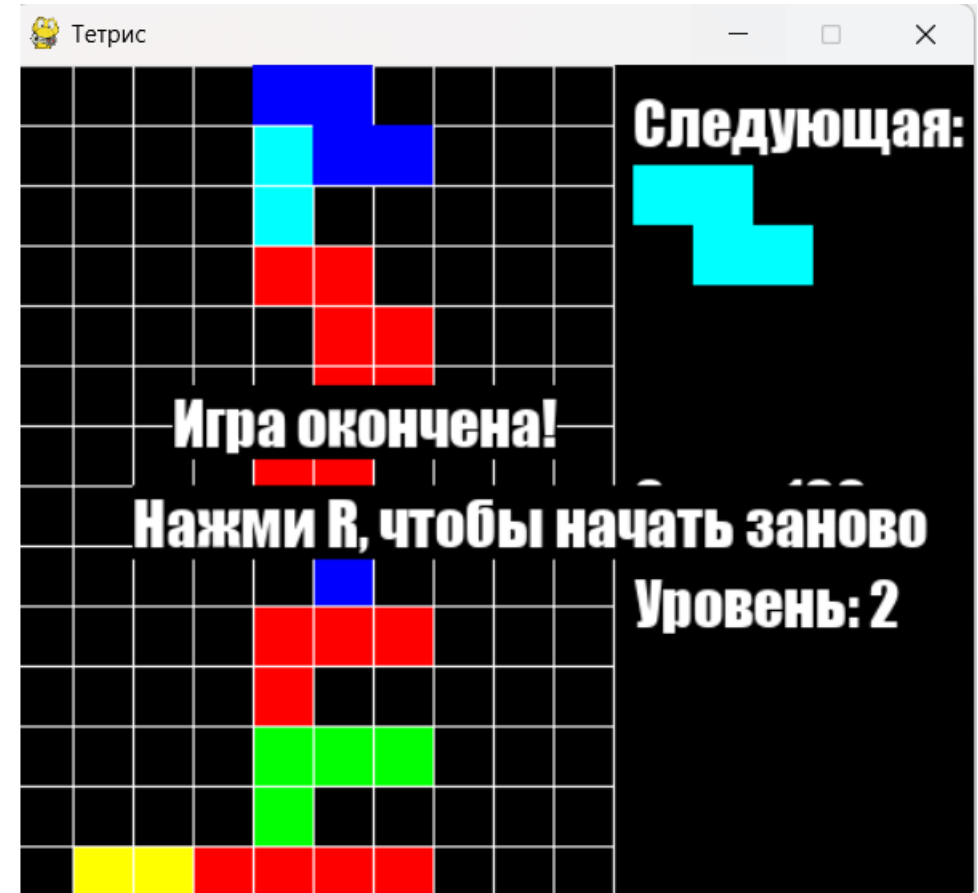
Для реализации были написаны функции `save_game` и `load_game`.



Реализация. Завершение игры

Если место в области, на которое игрок хочет поставить фигуру недостаточно (фигура начинает выходить за пределы экрана сверху), то игра завершается. Игроку можно начать игру заново, нажав клавишу R.

Для реализации была написана функция `is_collision`. Часть кода находится в основной функции игры `game`.



Тестирование

Пример работы всего приложения можно посмотреть на видео, представленном в отдельном файле.

Заключение

- Результатом проекта стала моя версия игры Тетрис.
- У проекта есть различные варианты развития как аудиовизуальных эффектов (например, добавление звука при заполнение линии), так и функционала (таблица рекордов, выбор уровня сложности, усложнение метода подсчета очков).