**«Линия и квадраты»**

Прокошина Софья Андреевна

В данной реализации игры представлены 3 уровня сложности с помощью интеллектуальной стратегии компьютера, от случайных ходов на легком уровне до тщательно продуманных решений на высоком уровне.

Каждый уровень сложности по-своему меняет логику действий компьютера, чтобы создать интересный и постепенно усложняющийся опыт для игрока.

Технологии и инструменты**:**

* Pygame — основная библиотека для создания графического интерфейса и анимации игры. Она обрабатывает события, такие как щелчки мыши и нажатия клавиш, и отвечает за отрисовку элементов: кнопок, игровых линий и квадратов.
* Sys и Os — стандартные модули Python для взаимодействия с системой. Модуль Sys позволяет корректно завершать выполнение программы и передавать аргументы командной строки, а Os — управлять путями к файлам и загружать ресурсы, такие как изображения и шрифты, независимо от операционной системы.
* Random — стандартный модуль Python, используемый для создания случайных чисел. Здесь он помогает управлять случайными ходами компьютера на более низком уровне сложности.
* Tkinter — стандартный графический интерфейс Python, используется для создания дополнительных окон, таких как правила игры или окно с результатами. Tkinter облегчает создание простых интерфейсов с кнопками и текстом, что делает его хорошим выбором для создания всплывающих окон и окон с информацией.
* Threading — модуль для создания и управления потоками. Он используется для запуска функций в фоновом режиме, таких как открытие окон с правилами, не прерывая основное игровое окно. Это позволяет поддерживать плавное выполнение программы и удобное взаимодействие с пользователем.

# Игровой процесс и правила:

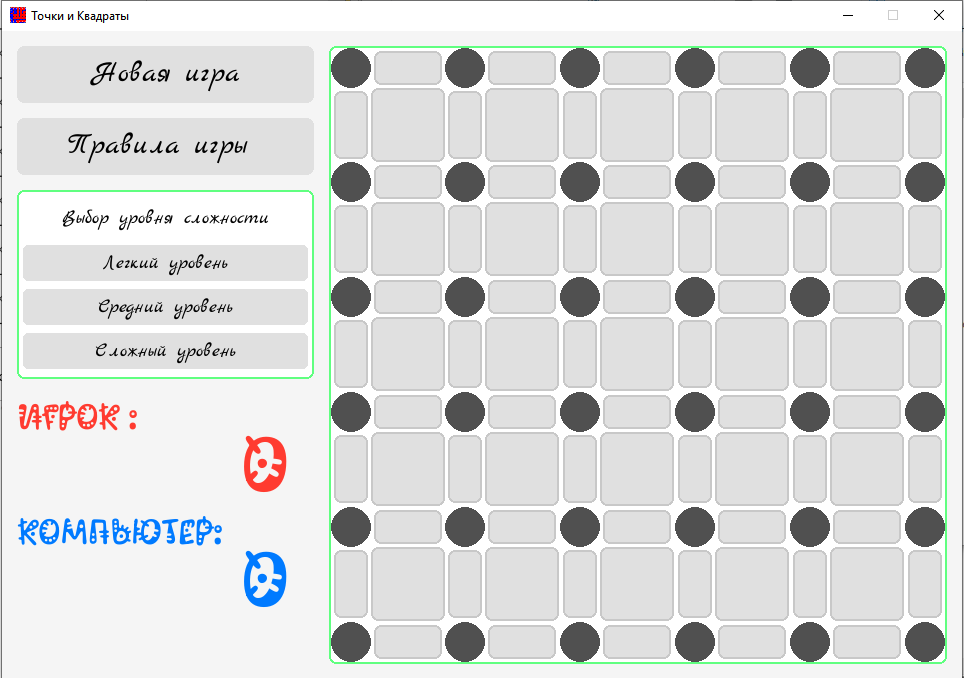


Рис. 1. Внешний вид основного окна игры

Игра начинается со стартового окна. Если пользователь нажмёт на любую кнопку, то его автоматически перенесёт на основное окно приложения.

Пользователь может выбрать уровень сложности игры с компьютером, если пользователь не выбрал какой-либо уровень по кнопке, то по умолчанию, выставлен легкий уровень.

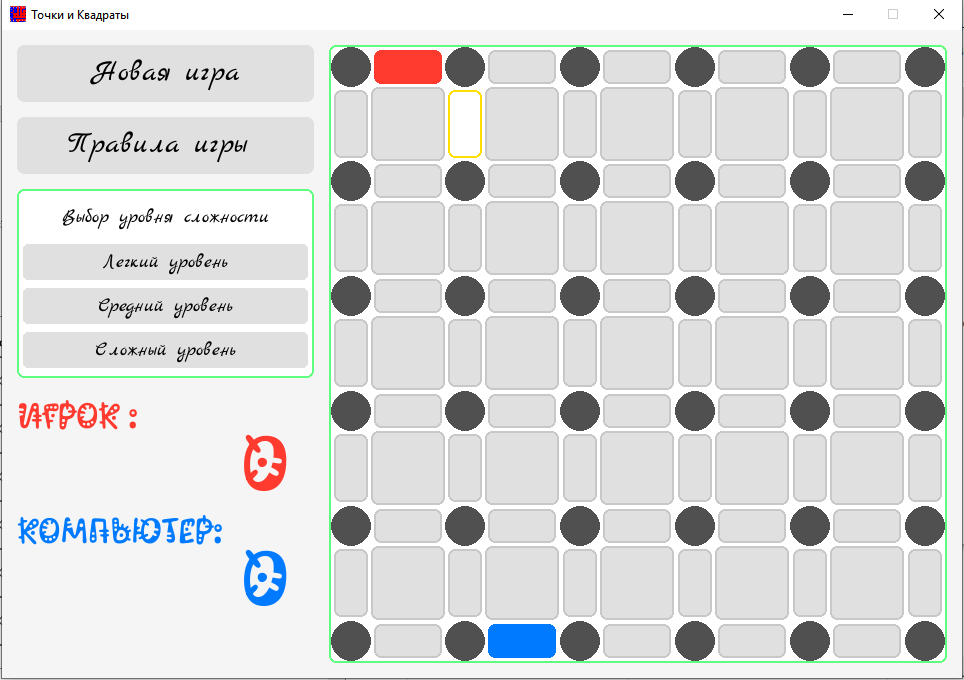


Рис. 2. Первые ходы пользователя и ИИ.

После выбора уровня пользователь делает первый ход на свободную позицию, которая подсвечивается, подсказывая возможно ли выбрать линию. Игрок закрашивает свои линии в виде прямоугольников в красный цвет, а компьютер синим. Внизу с левой стороны окна представлен счет, который определяет сколько квадратов, и кто закрасил.

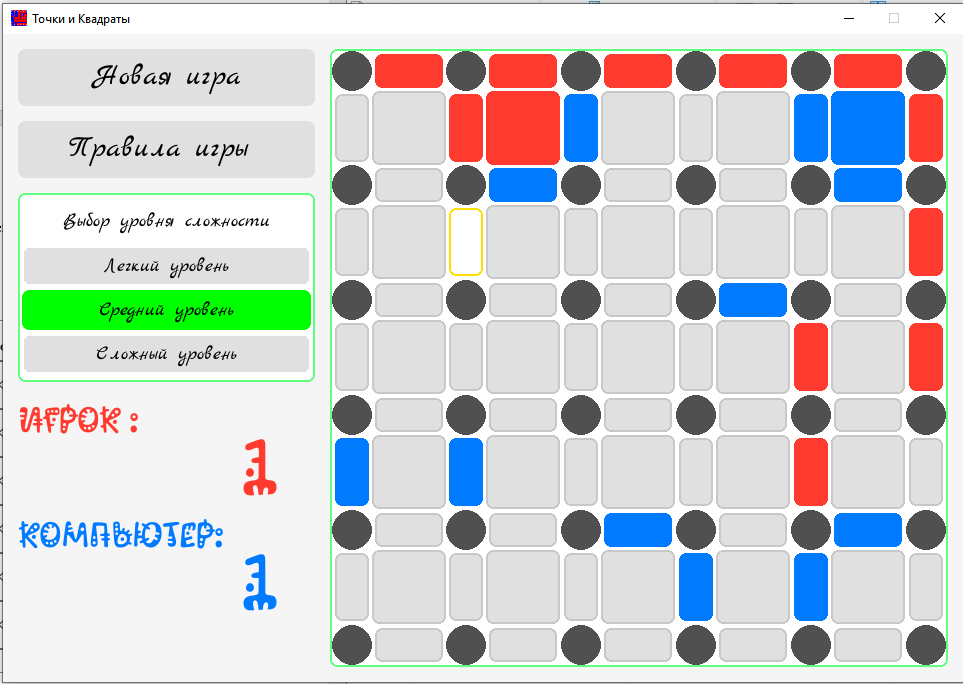


Рис.3. Игра

В конце игры подводятся итоги игры и для повторного ее запуска нужно нажать на экран.

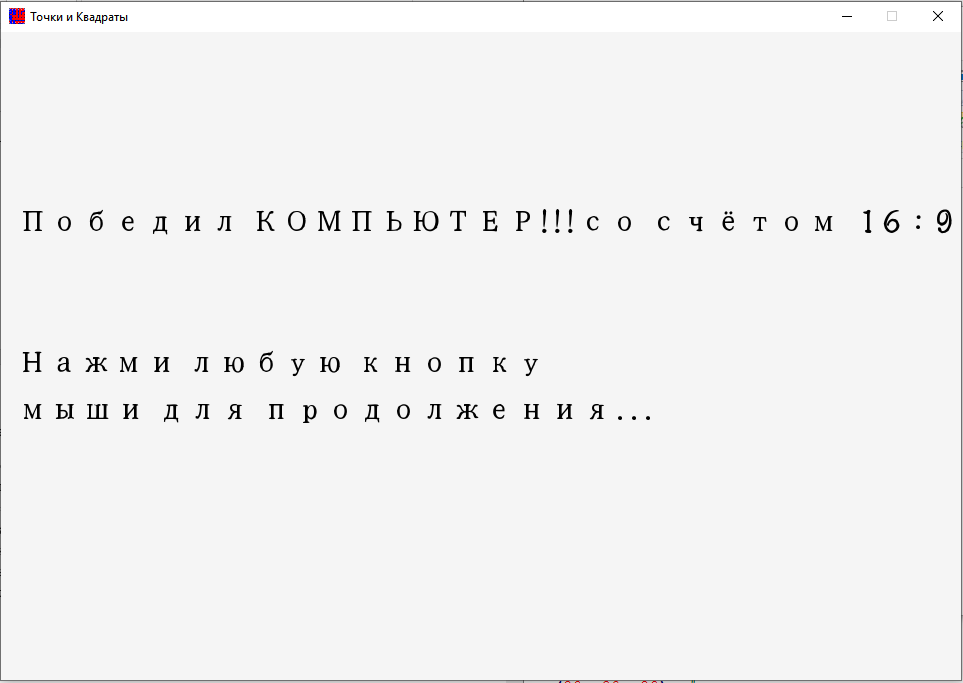


Рис.4. Итоги

Дополнительно, реализована кнопка «Правила игры», высвечивается окно с правилами, которое функционирует параллельно с основным игровым процессом, давая возможность читая правила выполнять ходы.

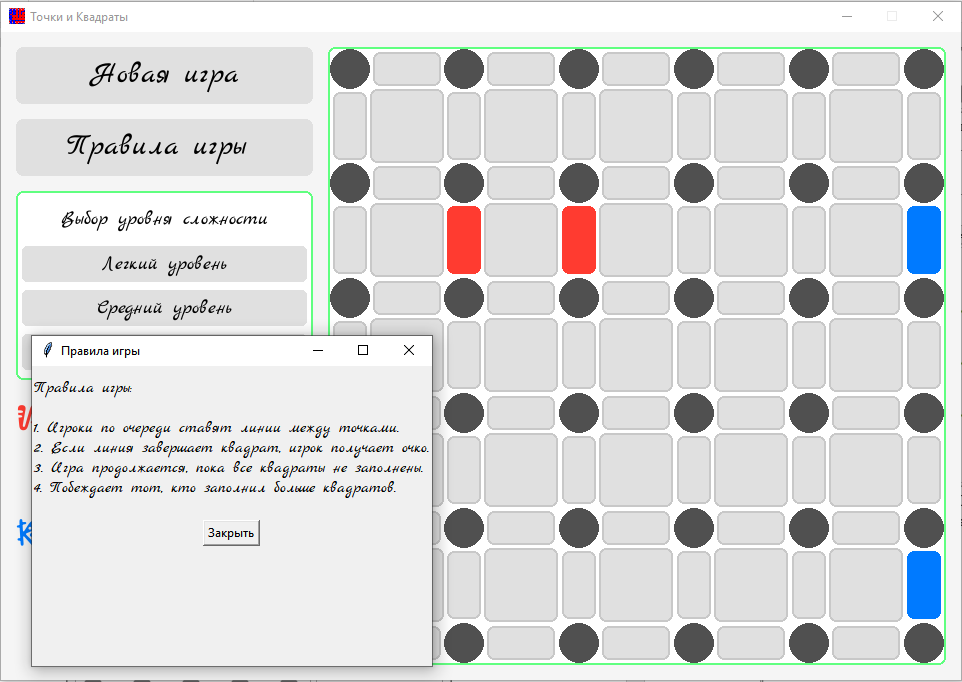


Рис.5. Правила

**Классы**:

* **class Rectangle**
* **class Square**
* **class Button**
* **class StartWindow**

**Функции обработки ходов и действий**

* **check\_squares()** функция проверяет, был ли замкнут один или несколько квадратов после хода игрока или компьютера.
* **set\_difficulty()** функция устанавливает уровень сложности игры, заданный параметром level, который может принимать значения "easy", "medium", или "hard". В зависимости от выбранного уровня сложности, функция меняет параметры, определяющие, насколько продвинутыми будут ходы компьютера.
* **easy\_move()** функция реализует логику хода компьютера на легком уровне сложности.
* **medium\_move()** функция реализует логику хода компьютера на среднем уровне сложности.
* **hard\_move()** функция реализует логику хода компьютера на сложном уровне сложности.
* **reset\_buttons()** функция сбрасывает состояние всех кнопок выбора уровня сложности, чтобы их можно было заново выбрать после перезапуска игры. Она делает кнопки активными, чтобы игрок мог выбрать новый уровень сложности, и обновляет их вид, чтобы убрать визуальную отметку о выборе.
* **game\_over()** проверяет, завершена ли игра, то есть выбраны ли все возможные линии на поле. Если игра окончена, функция подсчитывает очки и определяет победителя, выводя итоговое окно с результатами. Эта функция важна для определения конца игры и предоставления обратной связи игроку.
* **game\_loop()** основной цикл игры, который отвечает за обработку всех событий, таких как ходы игроков, ход компьютера, обновление состояния экрана и отрисовку элементов. Эта функция также отслеживает активность кнопок, проверяет завершение игры, обновляет счет и обрабатывает пользовательский интерфейс

**Функции графического интерфейса**

* **main\_field() о**твечает за отрисовку основного игрового поля на экране. В этой функции устанавливается основной макет игрового пространства, определяются его размеры и границы
* **left\_menu()** создает и отрисовывает меню, которое располагается слева от игрового поля. В этом меню размещаются кнопки и текстовые элементы, такие как выбор уровня сложности, кнопка перезапуска, кнопка для показа правил игры и информация о текущем счете.
* **draw\_point()** отвечает за отрисовку точек на игровом поле.
* **create\_grid()** создает сетку из объектов, представляющих линии и квадраты на игровом поле. Она возвращает массивы с этими объектами, которые затем используются для отрисовки и управления ходами. Функция служит для создания структуры игрового поля, которая позволяет отслеживать состояние линий и квадратов, а также управлять их изменениями в процессе игры.
* **draw\_score()** отображает текущий счет игры, показывая очки игрока и компьютера.
* **show\_rules\_window()** создает отдельное окно с правилами игры. Это позволяет игроку быстро ознакомиться с правилами, не покидая игру.
* **open\_rules\_in\_thread() з**апускает функцию show\_rules\_window() в отдельном потоке.