МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Разработка игры «Рэндзю»

(наименование темы проекта или работы)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к выпускной работе

ОБУЧАЮЩИЙСЯ:

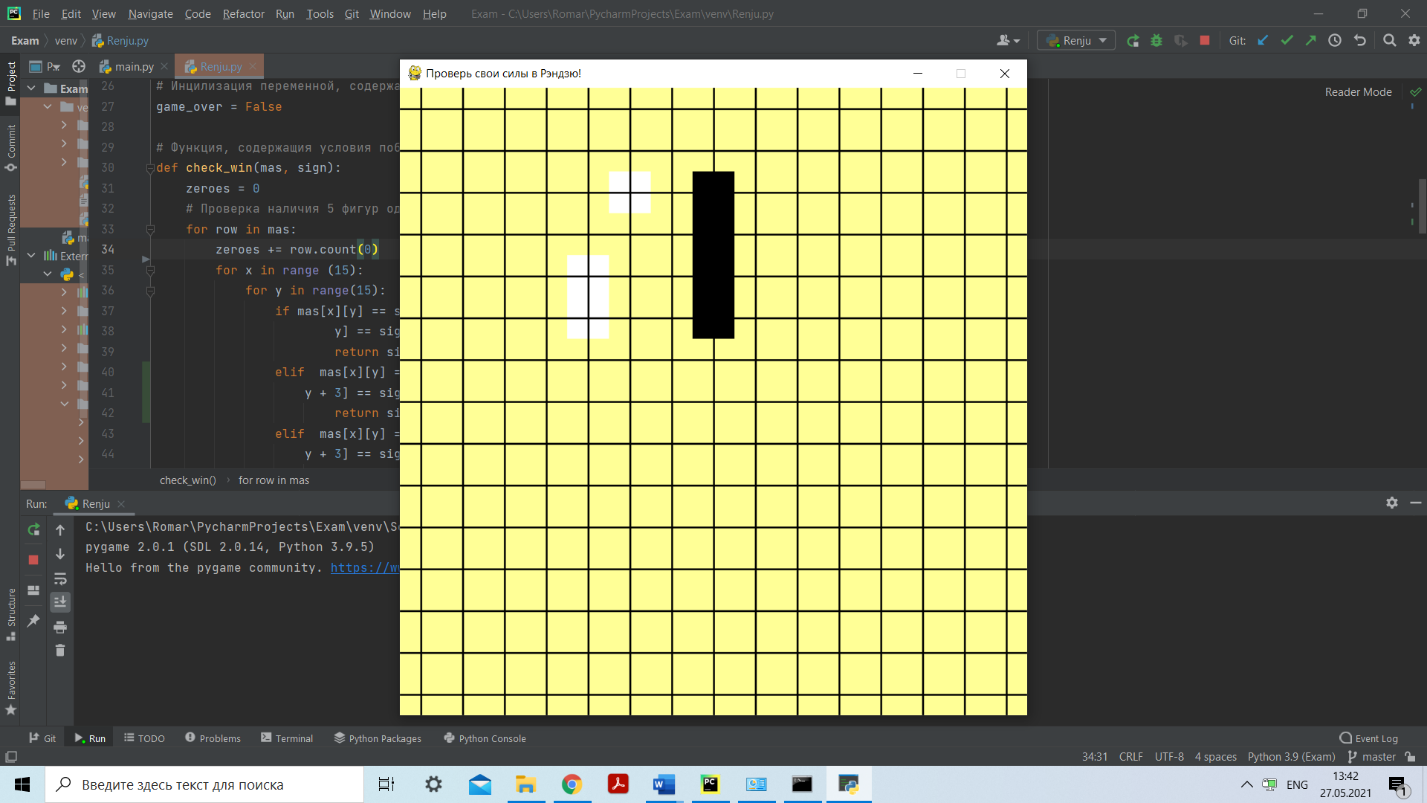
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Пох Д.С.

(подпись) (фамилия, и.,о.)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

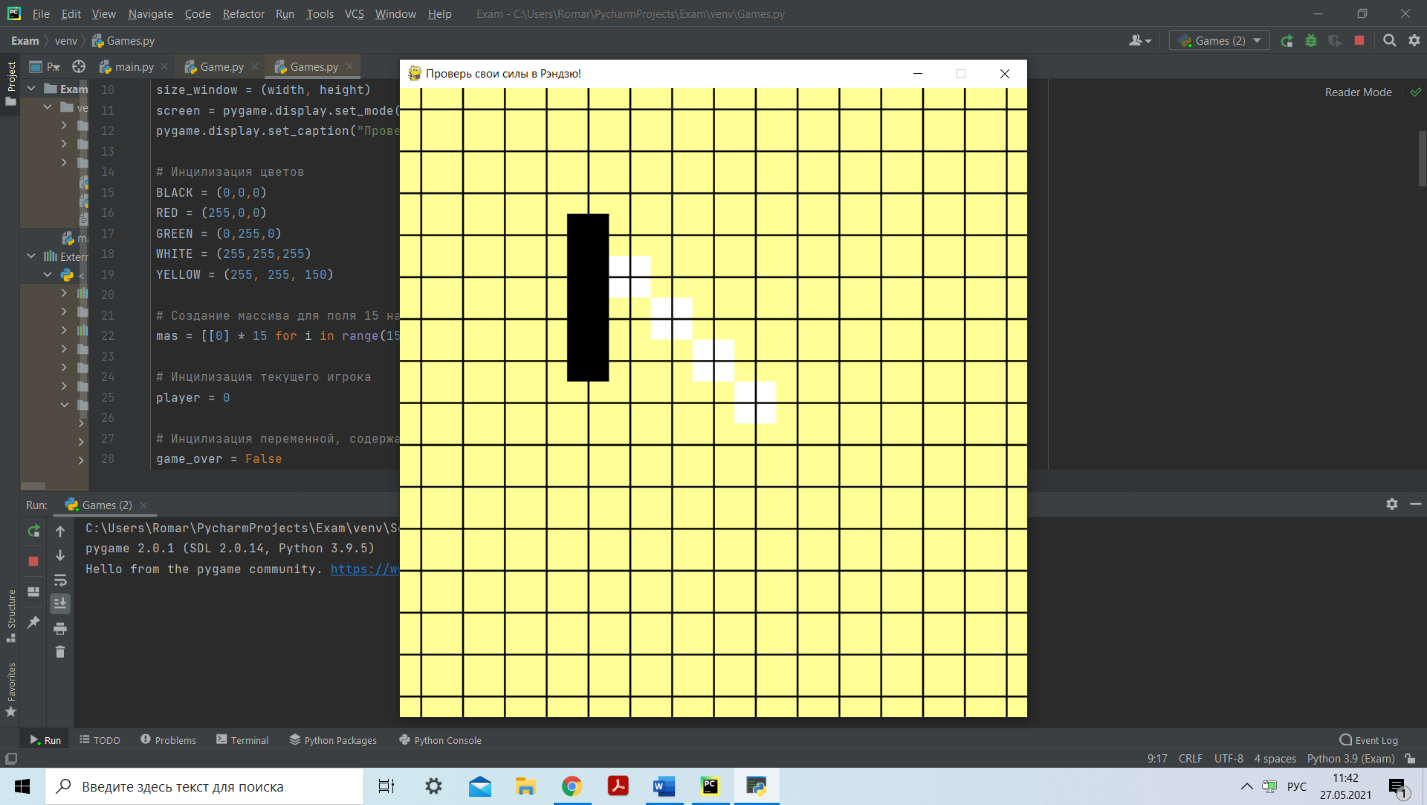
Нижний Новгород 2021

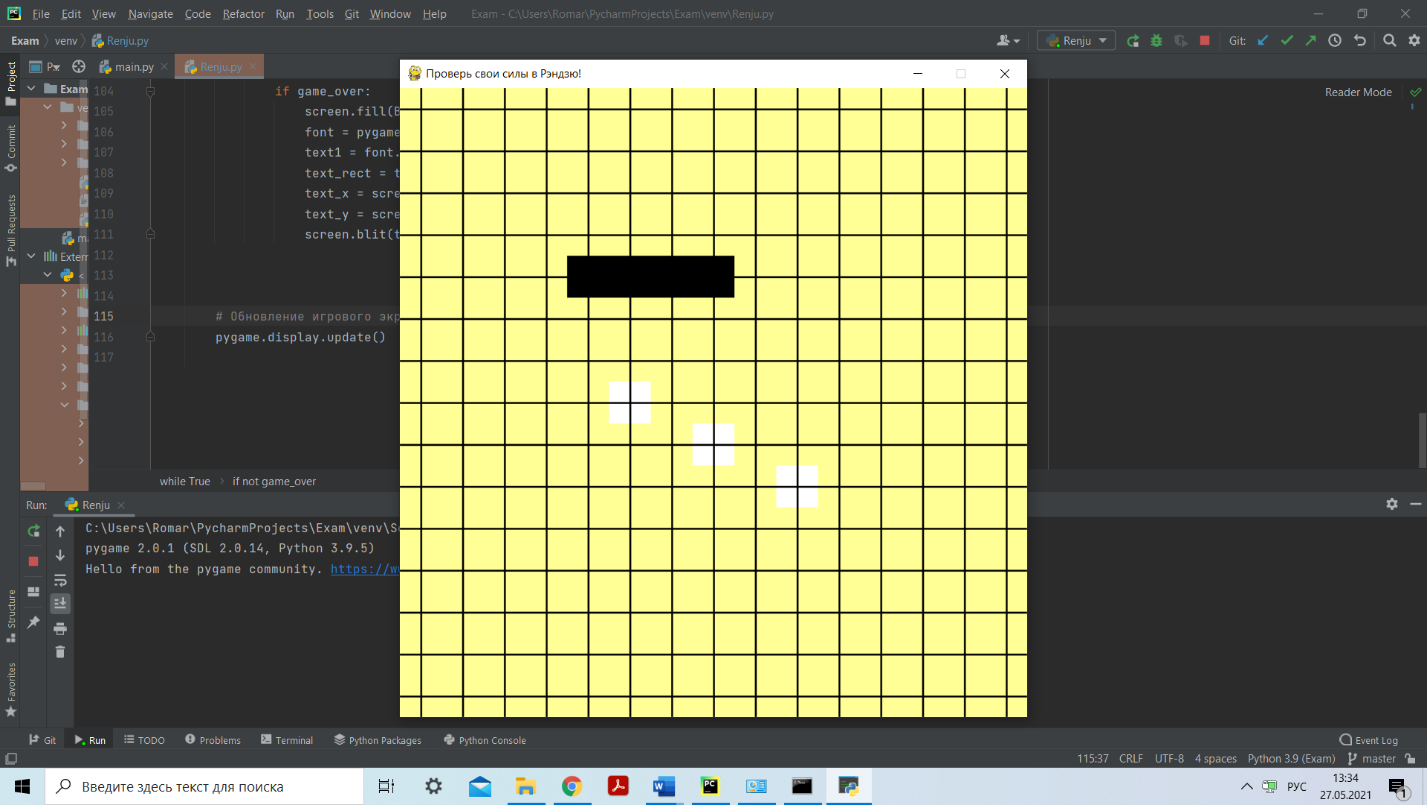
**Задание 13**. Игра «Рэндзю». Поле игры представляет собой квадрат 15 на 15. Игроки делают ходы по очереди. Первыми ходят чёрные. Каждым ходом игрок выставляет на доску в любое свободное пересечение линий доски один камень своего цвета. Побеждает тот, кто сможет первым построить непрерывный ряд из пяти камней своего цвета — по горизонтали, по вертикали или в диагональном направлении.

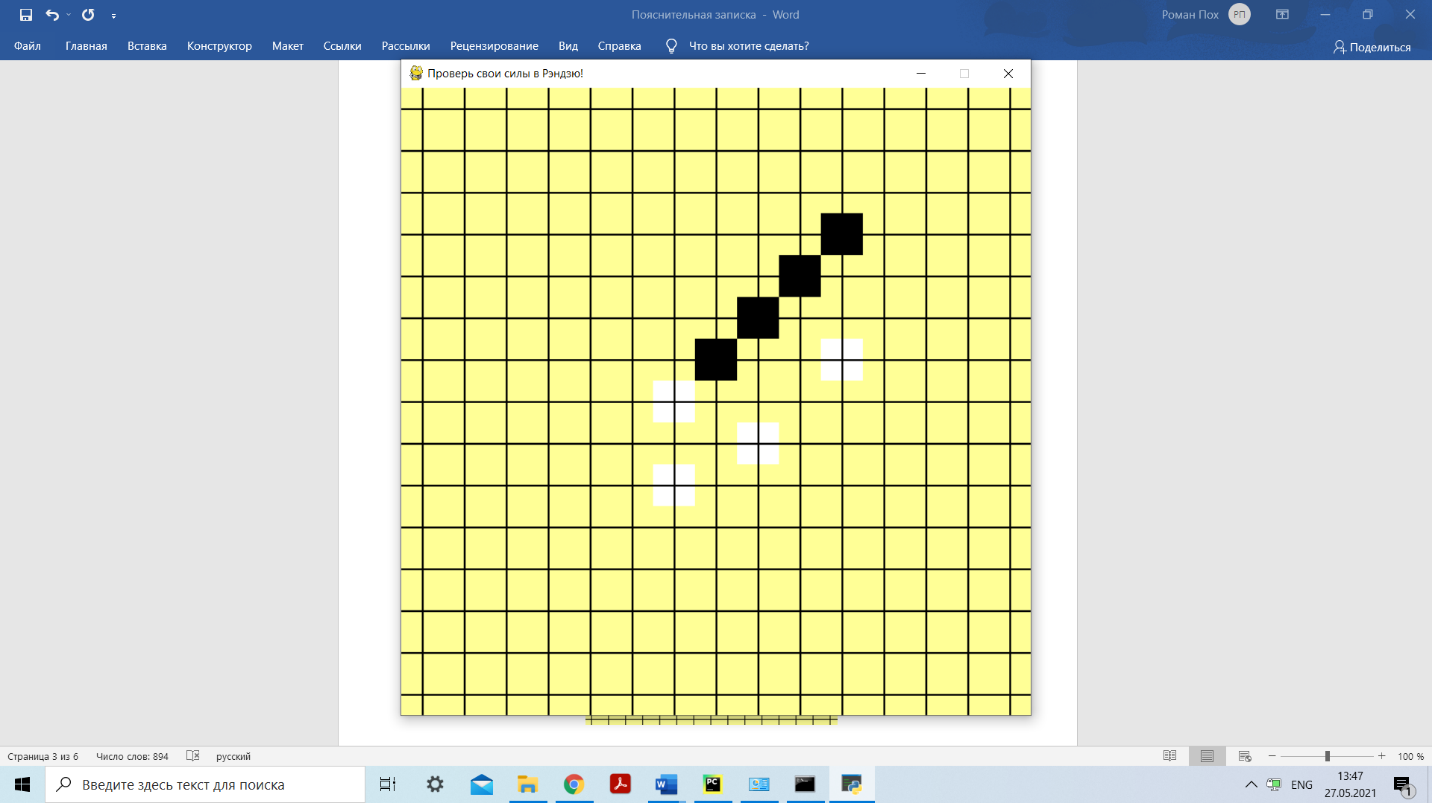
**Логика игры**

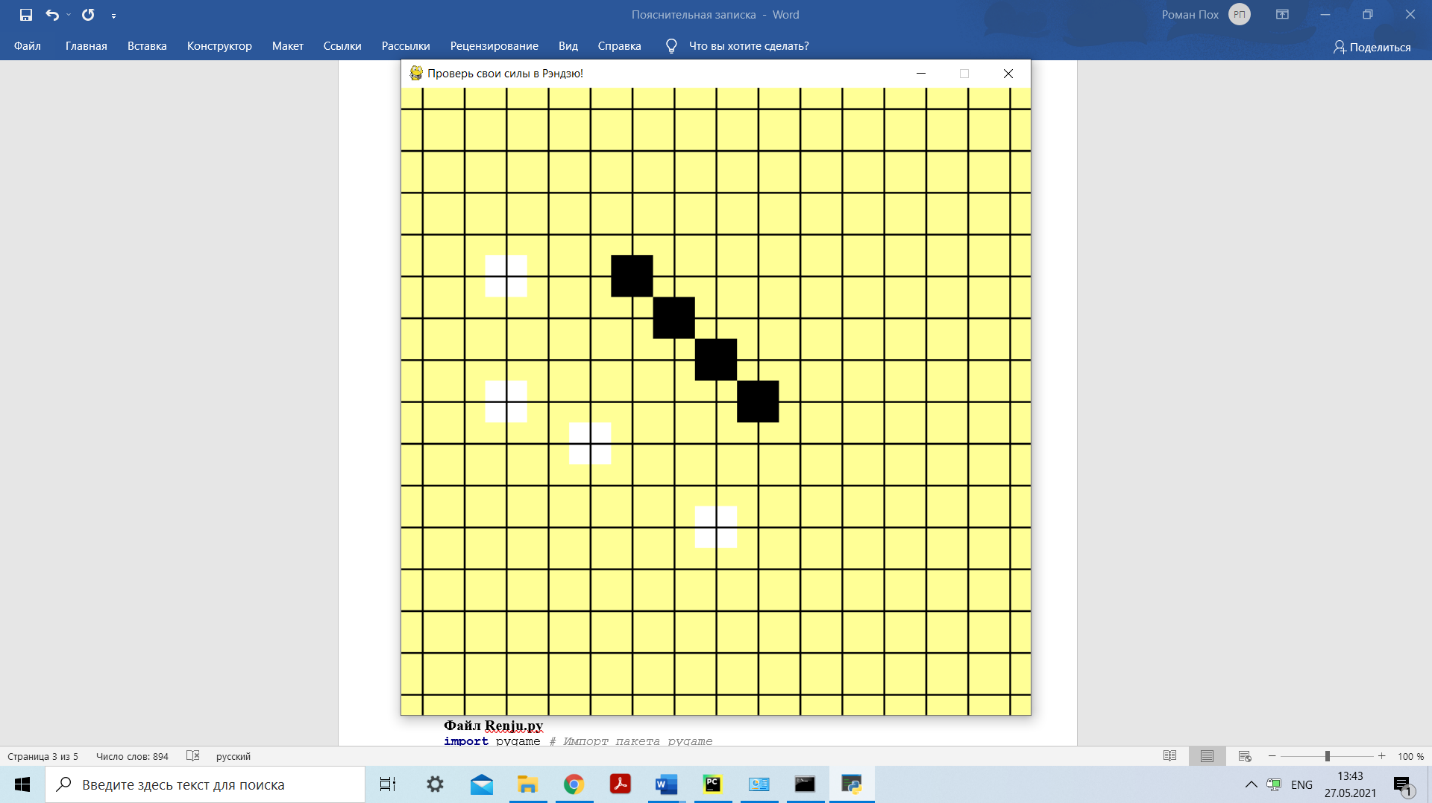
Поле полностью состоит из игровых фигур. Условно строится по осям X и Y. Один из игроков побеждает, когда раньше другого соберет линию их пяти фигур по следующим координатам:

* А(x, y), B(x+1, y), C(x+2, y), D(x+3, y), E(x+4, y) – Вертикальная линия



* А(x, y), B(x, y+1), C(x, y+2), D(x, y+3), E(x, y+4) – Горизонтальная линия
* А(x, y), B(x+1, y+1), C(x+2, y+2), D(x+3, y+3), E(x+4, y+4) – Диагональная линия



* А(x, y), B(x+1, y-1), C(x+2, y-2), D(x+3, y-3), E(x+4, y-4) – Диагональная линия

**Программный код**

* Среда разработки PyCharm;
* Модули: pygame, sys;
* Файл Renju.py содержит основной код программы, отвечает за хранение информации о текущем состоянии игры и определение допустимых ходов, а также за ведение журнала перемещений.

**Файл Renju.py**

**import** pygame *# Импорт пакета pygame***import** sys *# Импорт пакета sys*pygame.init()  
size\_block = 45 *# Размер фигуры, которой делается ход*width = height = size\_block \* 15 *# количество блоков  
  
# Настройка окна*size\_window = (width, height)  
screen = pygame.display.set\_mode(size\_window)  
pygame.display.set\_caption(**"Проверь свои силы в Рэндзю!"**)  
  
*# Инцилизация цветов*BLACK = (0,0,0)  
RED = (255,0,0)  
WHITE = (255,255,255)  
YELLOW = (255, 255, 150)  
  
*# Создание массива для поля 15 на 15*mas = [[0] \* 15 **for** i **in** range(15)]  
  
*# Инцилизация текущего игрока*player = 0  
  
*# Инцилизация переменной, содержащий условие окончания игры*game\_over = **False***# Функция, содержащия условия победы и ничьи***def** check\_win(mas, sign):  
 zeroes = 0**for** row **in** mas:  
 zeroes += row.count(0)  
 **for** x **in** range (15):  
 **for** y **in** range(15):  
 **if** mas[x][y] == sign **and** mas[x+1][y] == sign **and** mas[x + 2][y] == sign **and** mas[x + 3][  
 y] == sign **and** mas[x + 4][y] == sign:  
 **return** sign *# Проверка наличия 5 фигур одного цвета, стоящих в одной вертикальной линии* **elif** mas[x][y] == sign **and** mas[x][y + 1] == sign **and** mas[x][y + 2] == sign **and** mas[x][  
 y + 3] == sign **and** mas[x][y + 4] == sign:  
 **return** sign  
 **elif** mas[x][y] == sign **and** mas[x - 1][y + 1] == sign **and** mas[x - 2][y + 2] == sign **and** mas[x - 3][  
 y + 3] == sign **and** mas[x - 4][y + 4] == sign:  
 **return** sign *# Проверка наличия 5 фигур одного цвета, стоящих в одной диагональной линии, где значения по оси x убывают, а по y возрастают* **elif** mas[x][y] == sign **and** mas[x + 1][y + 1] == sign **and** mas[x + 2][y + 2] == sign **and** mas[x + 3][  
 y + 3] == sign **and** mas[x + 4][y + 4] == sign:  
 **return** sign *# Проверка наличия 5 фигур одного цвета, стоящих в одной диагональной линии, где значения по осям x и y возрастают* **if** zeroes == 0:  
 **return "Победила дружба!"** *# Условие, выполняемое при заполнении всего поля, но отсутствии победных комбинаций* **else**:  
 **return False***# Цикл игры***while True**:  
 **for** event **in** pygame.event.get():  
 **if** event.type == pygame.QUIT:  
 pygame.quit()  
 sys.exit(0)  
  
 *# Условия управления мышкой, выбора блока на поле* **elif** event.type == pygame.MOUSEBUTTONUP **and not** game\_over:  
 x\_mouse, y\_mouse = pygame.mouse.get\_pos()  
 col = x\_mouse // size\_block  
 row = y\_mouse // size\_block  
 **if** mas[row][col] == 0:  
 **if** player % 2 == 0:  
 mas[row][col] = **'Победа черных!'** *# Ход черной фигурой* **else**:  
 mas[row][col] = **'Победа белых!'** *# Ход белой фигурой* player += 1  
 *# Условия выполняется при победе одного из игроков. Новая игра включается при нажатии пробела.* **elif** event.type == pygame.KEYDOWN **and** event.key == pygame.K\_SPACE:  
 game\_over = **False** mas = [[0] \* 15 **for** i **in** range(15)]  
 player = 0  
 screen.fill(BLACK)  
  
 *#Условия продолжения игры* **if not** game\_over:  
 **for** row **in** range(15):  
 **for** col **in** range(15):  
 **if** mas[row][col] == **'Победа черных!'**:  
 color = BLACK *# Отметка на поле чероной фигуры* **elif** mas[row][col] == **'Победа белых!'**:  
 color = WHITE *# Отметка на поле белой фигуры* **else**:  
 color = YELLOW *# Цвет свободных фигур* x = col \* size\_block  
 y = row \* size\_block  
 pygame.draw.rect(screen,color, (x, y, size\_block, size\_block))  
 pygame.draw.line(screen, BLACK, (x + 22.5, 0), (x + 22.5, 690), 2) *# Прорисовка вертикальный линий на поле* pygame.draw.line(screen, BLACK, (0, y + 22.5), (690, y + 22.5), 2) *# Прорисовка горизонтальный линий на поле  
  
 # Условия выбора победителя* **if** (player-1)%2 == 0:  
 game\_over = check\_win(mas, **'Победа черных!'**)  
 **else**:  
 game\_over = check\_win(mas, **'Победа белых!'**)  
  
 *# Прописаны условия, выполняемые при окончании игры* **if** game\_over:  
 screen.fill(BLACK)  
 font = pygame.font.SysFont(**'stxingkai'**, 80)  
 text1 = font.render(game\_over, **True**, WHITE)  
 text\_rect = text1.get\_rect()  
 text\_x = screen.get\_width() / 2 - text\_rect.width / 2  
 text\_y = screen.get\_height() / 2 - text\_rect.height / 2  
 screen.blit(text1, [text\_x, text\_y])  
  
  
  
 *# Обновление игрового экрана* pygame.display.update()