

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по ДЗ

Выполнил:

студент группы ИУ5-35Б

Пермяков Семён

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Юрий Евгеньевич

Подпись и дата:

Постановка задачи

Создать приложение Сапер с использованием pygame.

Текст программы

```
import pygame, random, os, sys

pygame.init()
running = True
size_start = 1100, 800
screen = pygame.display.set_mode(size_start)
screen.fill((0, 0, 0))
flag_first = True
mine_coords = []

#выход из начальных и конечных экранов
def terminate():
    pygame.quit()
    sys.exit()

#обработка начальный экран
def start_screen():
    intro_text = ["
                                САПЁР",
                    "Правила игры",
                    "Цель игры: обнаружить все мины и при этом постараться не",
                    "подорваться на них. После клика (ЛКМ) на ячейку, на ней",
                    "появляется цифра. Она обозначает, сколько мин заложено по",
                    "соседству. Если ячейка пустая, то мин по соседству нет.",
                    "Анализируя эти цифры, можно обнаружить конкретную",
                    "заминированную ячейку. Ее можно пометить флажком (ПКМ)",
                    "чтобы случайно не подорваться. Таким образом, нужно",
                    "обнаружить все мины.",
                    "Введите параметры поля через пробел: ширина, длина",
                    "количество мин",
                    "",
                    "Чтобы начать, нажмите ЛКМ."]
    fon = pygame.transform.scale(load_image('fon.jpg'), (1100, 800))
    screen.blit(fon, (0, 0))
    font = pygame.font.Font(None, 50)
    text_coord = 50
    for line in intro_text:
        string_rendered = font.render(line, 1, pygame.Color('black'))
        intro_rect = string_rendered.get_rect()
        text_coord += 10
        intro_rect.top = text_coord
        intro_rect.x = 10
        text_coord += intro_rect.height
        screen.blit(string_rendered, intro_rect)
    arguments = ''
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                terminate()
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_1:
                    arguments = arguments + '1'
                if event.key == pygame.K_2:
                    arguments = arguments + '2'
                if event.key == pygame.K_3:
                    arguments = arguments + '3'
                if event.key == pygame.K_4:
                    arguments = arguments + '4'
                if event.key == pygame.K_5:
                    arguments = arguments + '5'
                if event.key == pygame.K_6:
                    arguments = arguments + '6'
```

```

        if event.key == pygame.K_7:
            arguments = arguments + '7'
        if event.key == pygame.K_8:
            arguments = arguments + '8'
        if event.key == pygame.K_9:
            arguments = arguments + '9'
        if event.key == pygame.K_0:
            arguments = arguments + '0'
        if event.key == pygame.K_SPACE:
            arguments = arguments + ' '
    if event.type == pygame.MOUSEBUTTONDOWN and event.button == 1 and
arguments != '':
        return arguments
    pygame.display.flip()

#загрузка спрайтов
def load_image(name, colorkey=None):
    fullname = os.path.join('data', name)
    image = pygame.image.load(fullname).convert()
    if colorkey is not None:
        if colorkey == -1:
            color_key = image.get_at((0, 0))
            image.set_colorkey(color_key)
        else:
            image = image.convert_alpha()
    return image

#класс спрайта бомбы
class Bomb(pygame.sprite.Sprite):
    image = load_image("bomb.png")
    image = pygame.transform.scale(image, (65, 65))

    #инициализация класса
    def __init__(self, group):
        super().__init__(group)
        self.image_bomb = Bomb.image
        self.rect = self.image.get_rect()
        self.rect.x = 175
        self.rect.y = 10

#инициализация экрана, выводимого в случае победы игрока
def win():
    intro_text = ["        ВЫ ВЫИГРАЛИ!!!"]
    font = pygame.font.Font(None, 50)
    text_coord = 80
    for line in intro_text:
        string_rendered = font.render(line, 1, pygame.Color('white'))
        intro_rect = string_rendered.get_rect()
        text_coord += 10
        intro_rect.top = text_coord
        intro_rect.x = 10
        text_coord += intro_rect.height
        screen.blit(string_rendered, intro_rect)
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                terminate()
        pygame.display.flip()

#инициализация экрана, выводимого в случае поражения игрока
def lose():
    intro_text = ["        ВЫ ПРОИГРАЛИ!!!"]
    font = pygame.font.Font(None, 50)
    text_coord = 80
    all_sprites = pygame.sprite.Group()
    Bomb(all_sprites)
    for line in intro_text:

```

```

        string_rendered = font.render(line, 1, pygame.Color('red'))
        intro_rect = string_rendered.get_rect()
        text_coord += 10
        intro_rect.top = text_coord
        intro_rect.x = 10
        text_coord += intro_rect.height
        screen.blit(string_rendered, intro_rect)
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                terminate()
        all_sprites.draw(screen)
        pygame.display.flip()

#класс работы с игровым полем
class Board:
    #инициализация класса
    def __init__(self, width, height):
        self.n_mine = n_mine
        self.width = width
        self.height = height
        self.board = [[0] * width for _ in range(height)]
        self.left = 10
        self.top = 70
        self.cell_size = 50
        self.i_first_cell = 0
        self.j_first_cell = 0
        self.mine_coords = mine_coords
        self.flag_first = True
        self.status_end = 'no results'
        self.k = 0

    #размеры поля
    def set_view(self, left, top, cell_size):
        self.left = left
        self.top = top
        self.cell_size = cell_size

    #обработка поля
    def render(self):
        for i in range(self.width):
            for j in range(self.height):
                pygame.draw.rect(screen, pygame.Color('blue'), (self.left + (i *
self.cell_size), self.top + (j * self.cell_size), self.cell_size, self.cell_size),
1)
                pygame.draw.rect(screen, pygame.Color('grey'), (self.left + (i *
self.cell_size), self.top + (j * self.cell_size), self.cell_size - 1,
self.cell_size - 1))

    #добавление мин на поле
    def get_mine(self):
        while self.board[self.i_first_cell][self.j_first_cell] != 0:
            self.board = [[0] * a for _ in range(b)]
            copy = self.n_mine
            while copy != 0:
                i = random.randint(0, len(self.board) - 1)
                j = random.randint(0, len(self.board[0]) - 1)
                if self.board[i][j] != 10:
                    self.board[i][j] = 10
                    copy = copy - 1
            if self.board[0][0] != 10:
                self.board[0][0] = (int(self.board[0][1]) + int(self.board[1][0]) +
int(self.board[1][1])) // 10
            if self.board[0][len(self.board[0]) - 1] != 10:
                self.board[0][len(self.board[0]) - 1] =
(int(self.board[0][len(self.board[0]) - 2]) + int(self.board[1][len(self.board[0])
- 1]) + int(self.board[1][len(self.board[0]) - 2])) // 10

```

[illegible]

```

elif self.board[i + 1][j + 1] == 4:
    self.board[i + 1][j + 1] = 44
elif self.board[i + 1][j + 1] == 5:
    self.board[i + 1][j + 1] = 55
if self.board[i + 1][j - 1] == 1:
    self.board[i + 1][j - 1] = 11
elif self.board[i + 1][j - 1] == 0:
    self.board[i + 1][j - 1] = 100
elif self.board[i + 1][j - 1] == 2:
    self.board[i + 1][j - 1] = 22
elif self.board[i + 1][j - 1] == 3:
    self.board[i + 1][j - 1] = 33
elif self.board[i + 1][j - 1] == 4:
    self.board[i + 1][j - 1] = 44
elif self.board[i + 1][j - 1] == 5:
    self.board[i + 1][j - 1] = 55
if self.board[i + 1][j] == 1:
    self.board[i + 1][j] = 11
elif self.board[i + 1][j] == 0:
    self.board[i + 1][j] = 100
elif self.board[i + 1][j] == 2:
    self.board[i + 1][j] = 22
elif self.board[i + 1][j] == 3:
    self.board[i + 1][j] = 33
elif i == len(self.board) - 1 and j !=
len(self.board[0]) - 1 and j != 0:
    if self.board[i][j - 1] == 1:
        self.board[i][j - 1] = 11
    elif self.board[i][j - 1] == 0:
        self.board[i][j - 1] = 100
    elif self.board[i][j - 1] == 2:
        self.board[i][j - 1] = 22
    if self.board[i][j + 1] == 1:
        self.board[i][j + 1] = 11
    elif self.board[i][j + 1] == 0:
        self.board[i][j + 1] = 100
    elif self.board[i][j + 1] == 2:
        self.board[i][j + 1] = 22
    if self.board[i - 1][j + 1] == 1:
        self.board[i - 1][j + 1] = 11
    elif self.board[i - 1][j + 1] == 0:
        self.board[i - 1][j + 1] = 100
    elif self.board[i - 1][j + 1] == 2:
        self.board[i - 1][j + 1] = 22
    elif self.board[i - 1][j + 1] == 3:
        self.board[i - 1][j + 1] = 33
    elif self.board[i - 1][j + 1] == 4:
        self.board[i - 1][j + 1] = 44
    elif self.board[i - 1][j + 1] == 5:
        self.board[i - 1][j + 1] = 55
    if self.board[i - 1][j - 1] == 1:
        self.board[i - 1][j - 1] = 11
    elif self.board[i - 1][j - 1] == 0:
        self.board[i - 1][j - 1] = 100
    elif self.board[i - 1][j - 1] == 2:
        self.board[i - 1][j - 1] = 22
    elif self.board[i - 1][j - 1] == 3:
        self.board[i - 1][j - 1] = 33
    elif self.board[i - 1][j - 1] == 4:
        self.board[i - 1][j - 1] = 44
    elif self.board[i - 1][j - 1] == 5:
        self.board[i - 1][j - 1] = 55
    if self.board[i - 1][j] == 1:
        self.board[i - 1][j] = 11
    elif self.board[i - 1][j] == 0:
        self.board[i - 1][j] = 100
    elif self.board[i - 1][j] == 2:

```

```

        self.board[i - 1][j] = 22
    elif self.board[i - 1][j] == 3:
        self.board[i - 1][j] = 33
elif j == 0 and i != 0 and i != len(self.board) - 1:
    if self.board[i - 1][j] == 1:
        self.board[i - 1][j] = 11
    elif self.board[i - 1][j] == 0:
        self.board[i - 1][j] = 100
    elif self.board[i - 1][j] == 2:
        self.board[i - 1][j] = 22
    if self.board[i + 1][j] == 1:
        self.board[i + 1][j] = 11
    elif self.board[i + 1][j] == 0:
        self.board[i + 1][j] = 100
    elif self.board[i + 1][j] == 2:
        self.board[i + 1][j] = 22
    if self.board[i - 1][j + 1] == 1:
        self.board[i - 1][j + 1] = 11
    elif self.board[i - 1][j + 1] == 0:
        self.board[i - 1][j + 1] = 100
    elif self.board[i - 1][j + 1] == 2:
        self.board[i - 1][j + 1] = 22
    elif self.board[i - 1][j + 1] == 3:
        self.board[i - 1][j + 1] = 33
    elif self.board[i - 1][j + 1] == 4:
        self.board[i - 1][j + 1] = 44
    elif self.board[i - 1][j + 1] == 5:
        self.board[i - 1][j + 1] = 55
    if self.board[i + 1][j + 1] == 1:
        self.board[i + 1][j + 1] = 11
    elif self.board[i + 1][j + 1] == 0:
        self.board[i + 1][j + 1] = 100
    elif self.board[i + 1][j + 1] == 2:
        self.board[i + 1][j + 1] = 22
    elif self.board[i + 1][j + 1] == 3:
        self.board[i + 1][j + 1] = 33
    elif self.board[i + 1][j + 1] == 4:
        self.board[i + 1][j + 1] = 44
    elif self.board[i + 1][j + 1] == 5:
        self.board[i + 1][j + 1] = 55
    if self.board[i][j + 1] == 1:
        self.board[i][j + 1] = 11
    elif self.board[i][j + 1] == 0:
        self.board[i][j + 1] = 100
    elif self.board[i][j + 1] == 2:
        self.board[i][j + 1] = 22
    elif self.board[i][j + 1] == 3:
        self.board[i][j + 1] = 33
elif j == len(self.board[0]) - 1 and i !=
len(self.board) - 1 and i != 0:
    if self.board[i - 1][j] == 1:
        self.board[i - 1][j] = 11
    elif self.board[i - 1][j] == 0:
        self.board[i - 1][j] = 100
    elif self.board[i - 1][j] == 2:
        self.board[i - 1][j] = 22
    if self.board[i + 1][j] == 1:
        self.board[i + 1][j] = 11
    elif self.board[i + 1][j] == 0:
        self.board[i + 1][j] = 100
    elif self.board[i + 1][j] == 2:
        self.board[i + 1][j] = 22
    if self.board[i - 1][j - 1] == 1:
        self.board[i - 1][j - 1] = 11
    elif self.board[i - 1][j - 1] == 0:
        self.board[i - 1][j - 1] = 100
    elif self.board[i - 1][j - 1] == 2:

```

```

        self.board[i - 1][j - 1] = 22
    elif self.board[i - 1][j - 1] == 3:
        self.board[i - 1][j - 1] = 33
    elif self.board[i - 1][j - 1] == 4:
        self.board[i - 1][j - 1] = 44
    elif self.board[i - 1][j - 1] == 5:
        self.board[i - 1][j - 1] = 55
    if self.board[i + 1][j - 1] == 1:
        self.board[i + 1][j - 1] = 11
    elif self.board[i + 1][j - 1] == 0:
        self.board[i + 1][j - 1] = 100
    elif self.board[i + 1][j - 1] == 2:
        self.board[i + 1][j - 1] = 22
    elif self.board[i + 1][j - 1] == 3:
        self.board[i + 1][j - 1] = 33
    elif self.board[i + 1][j - 1] == 4:
        self.board[i + 1][j - 1] = 44
    elif self.board[i + 1][j - 1] == 5:
        self.board[i + 1][j - 1] = 55
    if self.board[i][j - 1] == 1:
        self.board[i][j - 1] = 11
    elif self.board[i][j - 1] == 0:
        self.board[i][j - 1] = 100
    elif self.board[i][j - 1] == 2:
        self.board[i][j - 1] = 22
    elif self.board[i][j - 1] == 3:
        self.board[i][j - 1] = 33
elif i == 0 and j == 0:
    if self.board[i + 1][j] == 1:
        self.board[i + 1][j] = 11
    elif self.board[i + 1][j] == 0:
        self.board[i + 1][j] = 100
    elif self.board[i + 1][j] == 2:
        self.board[i + 1][j] = 22
    if self.board[i][j + 1] == 1:
        self.board[i][j + 1] = 11
    elif self.board[i][j + 1] == 0:
        self.board[i][j + 1] = 100
    elif self.board[i][j + 1] == 2:
        self.board[i][j + 1] = 22
    if self.board[i + 1][j + 1] == 1:
        self.board[i + 1][j + 1] = 11
    elif self.board[i + 1][j + 1] == 0:
        self.board[i + 1][j + 1] = 100
    elif self.board[i + 1][j + 1] == 2:
        self.board[i + 1][j + 1] = 22
    elif self.board[i + 1][j + 1] == 3:
        self.board[i + 1][j + 1] = 33
    elif self.board[i + 1][j + 1] == 4:
        self.board[i + 1][j + 1] = 44
    elif self.board[i + 1][j + 1] == 5:
        self.board[i + 1][j + 1] = 55
elif i == 0 and j == len(self.board[0]) - 1:
    if self.board[i + 1][j] == 1:
        self.board[i + 1][j] = 11
    elif self.board[i + 1][j] == 0:
        self.board[i + 1][j] = 100
    elif self.board[i + 1][j] == 2:
        self.board[i + 1][j] = 22
    if self.board[i][j - 1] == 1:
        self.board[i][j - 1] = 11
    elif self.board[i][j - 1] == 0:
        self.board[i][j - 1] = 100
    elif self.board[i][j - 1] == 2:
        self.board[i][j - 1] = 22
    if self.board[i + 1][j - 1] == 1:
        self.board[i + 1][j - 1] = 11

```



```

        elif self.board[i + 1][j - 1] == 0:
            self.board[i + 1][j - 1] = 100
        elif self.board[i + 1][j - 1] == 2:
            self.board[i + 1][j - 1] = 22
        elif self.board[i + 1][j - 1] == 3:
            self.board[i + 1][j - 1] = 33
        elif self.board[i + 1][j - 1] == 4:
            self.board[i + 1][j - 1] = 44
        elif self.board[i + 1][j - 1] == 5:
            self.board[i + 1][j - 1] = 55
    elif i == len(self.board) - 1 and j == 0:
        if self.board[i - 1][j] == 1:
            self.board[i - 1][j] = 11
        elif self.board[i - 1][j] == 0:
            self.board[i - 1][j] = 100
        elif self.board[i - 1][j] == 2:
            self.board[i - 1][j] = 22
        if self.board[i][j + 1] == 1:
            self.board[i][j + 1] = 11
        elif self.board[i][j + 1] == 0:
            self.board[i][j + 1] = 100
        elif self.board[i][j + 1] == 2:
            self.board[i][j + 1] = 22
        if self.board[i - 1][j + 1] == 1:
            self.board[i - 1][j + 1] = 11
        elif self.board[i - 1][j + 1] == 0:
            self.board[i - 1][j + 1] = 100
        elif self.board[i - 1][j + 1] == 2:
            self.board[i - 1][j + 1] = 22
        elif self.board[i - 1][j + 1] == 3:
            self.board[i - 1][j + 1] = 33
        elif self.board[i - 1][j + 1] == 4:
            self.board[i - 1][j + 1] = 44
        elif self.board[i - 1][j + 1] == 5:
            self.board[i - 1][j + 1] = 55
    elif i == len(self.board) - 1 and j ==
len(self.board[0]) - 1:

        if self.board[i - 1][j] == 1:
            self.board[i - 1][j] = 11
        elif self.board[i - 1][j] == 0:
            self.board[i - 1][j] = 100
        elif self.board[i - 1][j] == 2:
            self.board[i - 1][j] = 22
        if self.board[i][j - 1] == 1:
            self.board[i][j - 1] = 11
        elif self.board[i][j - 1] == 0:
            self.board[i][j - 1] = 100
        elif self.board[i][j - 1] == 2:
            self.board[i][j - 1] = 22
        if self.board[i - 1][j - 1] == 1:
            self.board[i - 1][j - 1] = 11
        elif self.board[i - 1][j - 1] == 0:
            self.board[i - 1][j - 1] = 100
        elif self.board[i - 1][j - 1] == 2:
            self.board[i - 1][j - 1] = 22
        elif self.board[i - 1][j - 1] == 3:
            self.board[i - 1][j - 1] = 33
        elif self.board[i - 1][j - 1] == 4:
            self.board[i - 1][j - 1] = 44
        elif self.board[i - 1][j - 1] == 5:
            self.board[i - 1][j - 1] = 55
    else:
        if self.board[i + 1][j - 1] == 1:
            self.board[i + 1][j - 1] = 11
        elif self.board[i + 1][j - 1] == 0:
            self.board[i + 1][j - 1] = 100
        elif self.board[i + 1][j - 1] == 2:

```

```
        self.board[i + 1][j - 1] = 22
    elif self.board[i + 1][j - 1] == 3:
        self.board[i + 1][j - 1] = 33
    elif self.board[i + 1][j - 1] == 4:
        self.board[i + 1][j - 1] = 44
    elif self.board[i + 1][j - 1] == 5:
        self.board[i + 1][j - 1] = 55
    if self.board[i + 1][j + 1] == 1:
        self.board[i + 1][j + 1] = 11
    elif self.board[i + 1][j + 1] == 0:
        self.board[i + 1][j + 1] = 100
    elif self.board[i + 1][j + 1] == 2:
        self.board[i + 1][j + 1] = 22
    elif self.board[i + 1][j + 1] == 3:
        self.board[i + 1][j + 1] = 33
    elif self.board[i + 1][j + 1] == 4:
        self.board[i + 1][j + 1] = 44
    elif self.board[i + 1][j + 1] == 5:
        self.board[i + 1][j + 1] = 55
    if self.board[i - 1][j - 1] == 1:
        self.board[i - 1][j - 1] = 11
    elif self.board[i - 1][j - 1] == 0:
        self.board[i - 1][j - 1] = 100
    elif self.board[i - 1][j - 1] == 2:
        self.board[i - 1][j - 1] = 22
    elif self.board[i - 1][j - 1] == 3:
        self.board[i - 1][j - 1] = 33
    elif self.board[i - 1][j - 1] == 4:
        self.board[i - 1][j - 1] = 44
    elif self.board[i - 1][j - 1] == 5:
        self.board[i - 1][j - 1] = 55
    if self.board[i - 1][j + 1] == 1:
        self.board[i - 1][j + 1] = 11
    elif self.board[i - 1][j + 1] == 0:
        self.board[i - 1][j + 1] = 100
    elif self.board[i - 1][j + 1] == 2:
        self.board[i - 1][j + 1] = 22
    elif self.board[i - 1][j + 1] == 3:
        self.board[i - 1][j + 1] = 33
    elif self.board[i - 1][j + 1] == 4:
        self.board[i - 1][j + 1] = 44
    elif self.board[i - 1][j + 1] == 5:
        self.board[i - 1][j + 1] = 55
    if self.board[i][j - 1] == 1:
        self.board[i][j - 1] = 11
    elif self.board[i][j - 1] == 0:
        self.board[i][j - 1] = 100
    elif self.board[i][j - 1] == 2:
        self.board[i][j - 1] = 22
    elif self.board[i][j - 1] == 3:
        self.board[i][j - 1] = 33
    if self.board[i][j + 1] == 1:
        self.board[i][j + 1] = 11
    elif self.board[i][j + 1] == 0:
        self.board[i][j + 1] = 100
    elif self.board[i][j + 1] == 2:
        self.board[i][j + 1] = 22
    elif self.board[i][j + 1] == 3:
        self.board[i][j + 1] = 33
    if self.board[i - 1][j] == 1:
        self.board[i - 1][j] = 11
    elif self.board[i - 1][j] == 0:
        self.board[i - 1][j] = 100
    elif self.board[i - 1][j] == 2:
        self.board[i - 1][j] = 22
    elif self.board[i - 1][j] == 3:
        self.board[i - 1][j] = 33
```

```

        if self.board[i + 1][j] == 1:
            self.board[i + 1][j] = 11
        elif self.board[i + 1][j] == 0:
            self.board[i + 1][j] = 100
        elif self.board[i + 1][j] == 2:
            self.board[i + 1][j] = 22
        elif self.board[i + 1][j] == 3:
            self.board[i + 1][j] = 33

#обработка мин на поле
def render_mine(self):
    for i in range(len(self.board)):
        for j in range(len(self.board[0])):
            if self.board[i][j] == 1010:
                pygame.draw.rect(screen, pygame.Color('red'), (self.left + (j *
self.cell_size), self.top + (i * self.cell_size), self.cell_size - 1,
self.cell_size - 1))
            elif self.board[i][j] == 100:
                pygame.draw.rect(screen, pygame.Color('white'), (self.left + (j
* self.cell_size), self.top + (i * self.cell_size), self.cell_size - 1,
self.cell_size - 1))
            elif self.board[i][j] == 11:
                pygame.draw.rect(screen, pygame.Color('white'), (self.left + (j
* self.cell_size), self.top + (i * self.cell_size), self.cell_size - 1,
self.cell_size - 1))
                pygame.draw.line(screen, pygame.Color('dark green'),
(int(self.left + (j + 0.5) * self.cell_size), int(self.top + (i + 0.1) *
self.cell_size)), (int(self.left + (j + 0.5) * self.cell_size), int(self.top + (i +
0.9) * self.cell_size)), 5)
            elif self.board[i][j] == 22:
                pygame.draw.rect(screen, pygame.Color('white'), (self.left + (j
* self.cell_size), self.top + (i * self.cell_size), self.cell_size - 1,
self.cell_size - 1))
                pygame.draw.line(screen, pygame.Color('dark green'),
(int(self.left + (j + 0.6) * self.cell_size), int(self.top + (i + 0.1) *
self.cell_size)), (int(self.left + (j + 0.6) * self.cell_size), int(self.top + (i +
0.5) * self.cell_size)), 5)
                pygame.draw.line(screen, pygame.Color('dark green'),
(int(self.left + (j + 0.2) * self.cell_size), int(self.top + (i + 0.5) *
self.cell_size)), (int(self.left + (j + 0.2) * self.cell_size), int(self.top + (i +
0.9) * self.cell_size)), 5)
                pygame.draw.line(screen, pygame.Color('dark green'),
(int(self.left + (j + 0.2) * self.cell_size), int(self.top + (i + 0.1) *
self.cell_size)), (int(self.left + (j + 0.6) * self.cell_size), int(self.top + (i +
0.1) * self.cell_size)), 5)
                pygame.draw.line(screen, pygame.Color('dark green'),
(int(self.left + (j + 0.2) * self.cell_size), int(self.top + (i + 0.5) *
self.cell_size)), (int(self.left + (j + 0.6) * self.cell_size), int(self.top + (i +
0.5) * self.cell_size)), 5)
                pygame.draw.line(screen, pygame.Color('dark green'),
(int(self.left + (j + 0.2) * self.cell_size), int(self.top + (i + 0.9) *
self.cell_size)), (int(self.left + (j + 0.6) * self.cell_size), int(self.top + (i +
0.9) * self.cell_size)), 5)
            elif self.board[i][j] == 33:
                pygame.draw.rect(screen, pygame.Color('white'), (self.left + (j
* self.cell_size), self.top + (i * self.cell_size), self.cell_size - 1,
self.cell_size - 1))
                pygame.draw.line(screen, pygame.Color('dark green'),
(int(self.left + (j + 0.6) * self.cell_size), int(self.top + (i + 0.1) *
self.cell_size)), (int(self.left + (j + 0.6) * self.cell_size), int(self.top + (i +
0.5) * self.cell_size)), 5)
                pygame.draw.line(screen, pygame.Color('dark green'),
(int(self.left + (j + 0.6) * self.cell_size), int(self.top + (i + 0.5) *
self.cell_size)), (int(self.left + (j + 0.6) * self.cell_size), int(self.top + (i +
0.9) * self.cell_size)), 5)
                pygame.draw.line(screen, pygame.Color('dark green'),
(int(self.left + (j + 0.2) * self.cell_size), int(self.top + (i + 0.1) *

```

[illegible]

[illegible]

[illegible]

```

        else:
            return self.status_end

    #проверка клика ЛКМ на клетку (проверка клетки на бомбу, посредством её
открытия)
    def check(self, event):
        pos = pygame.mouse.get_pos()
        if (pos[0] <= self.cell_size * a + self.left) and (pos[0] >= self.left) and
(pos[1] <= self.cell_size * b + self.top) and (pos[1] >= self.top):
            x_cell = (pos[0] - self.left) // self.cell_size
            y_cell = (pos[1] - self.top) // self.cell_size
            self.i_first_cell = y_cell
            self.j_first_cell = x_cell
            if self.board[y_cell][x_cell] == 1:
                self.board[y_cell][x_cell] = 11
            elif self.board[y_cell][x_cell] == 0:
                self.board[y_cell][x_cell] = 100
            elif self.board[y_cell][x_cell] == 10:
                self.board[y_cell][x_cell] = 1010
                self.status_end = 'lose'
            elif self.board[y_cell][x_cell] == 2:
                self.board[y_cell][x_cell] = 22
            elif self.board[y_cell][x_cell] == 3:
                self.board[y_cell][x_cell] = 33
            elif self.board[y_cell][x_cell] == 4:
                self.board[y_cell][x_cell] = 44
            elif self.board[y_cell][x_cell] == 5:
                self.board[y_cell][x_cell] = 55
            elif self.board[y_cell][x_cell] == 6:
                self.board[y_cell][x_cell] = 66
            elif self.board[y_cell][x_cell] == 7:
                self.board[y_cell][x_cell] = 77
            elif self.board[y_cell][x_cell] == 8:
                self.board[y_cell][x_cell] = 88

    #проверка клика ПКМ на клетку (проверка клетки на бомбу, посредством установки
флажка)
    def flag(self, event):
        pos = pygame.mouse.get_pos()
        if (pos[0] <= self.cell_size * a + self.left) and (pos[0] >= self.left) and
(pos[1] <= self.cell_size * b + self.top) and (pos[1] >= self.top):
            x_cell = (pos[0] - self.left) // self.cell_size
            y_cell = (pos[1] - self.top) // self.cell_size
            if self.board[y_cell][x_cell] == 1:
                self.board[y_cell][x_cell] = 91
            elif self.board[y_cell][x_cell] == 0:
                self.board[y_cell][x_cell] = 90
            elif self.board[y_cell][x_cell] == 10:
                self.board[y_cell][x_cell] = 910
                self.k = self.k + 1
            elif self.board[y_cell][x_cell] == 2:
                self.board[y_cell][x_cell] = 92
            elif self.board[y_cell][x_cell] == 3:
                self.board[y_cell][x_cell] = 93
            elif self.board[y_cell][x_cell] == 4:
                self.board[y_cell][x_cell] = 94
            elif self.board[y_cell][x_cell] == 5:
                self.board[y_cell][x_cell] = 95
            elif self.board[y_cell][x_cell] == 6:
                self.board[y_cell][x_cell] = 96
            elif self.board[y_cell][x_cell] == 7:
                self.board[y_cell][x_cell] = 97
            elif self.board[y_cell][x_cell] == 8:
                self.board[y_cell][x_cell] = 98
            elif self.board[y_cell][x_cell] == 91:
                self.board[y_cell][x_cell] = 1
            elif self.board[y_cell][x_cell] == 90:

```

```

        self.board[y_cell][x_cell] = 0
    elif self.board[y_cell][x_cell] == 910:
        self.board[y_cell][x_cell] = 10
    elif self.board[y_cell][x_cell] == 92:
        self.board[y_cell][x_cell] = 2
    elif self.board[y_cell][x_cell] == 93:
        self.board[y_cell][x_cell] = 3
    elif self.board[y_cell][x_cell] == 94:
        self.board[y_cell][x_cell] = 4
    elif self.board[y_cell][x_cell] == 95:
        self.board[y_cell][x_cell] = 5
    elif self.board[y_cell][x_cell] == 96:
        self.board[y_cell][x_cell] = 6
    elif self.board[y_cell][x_cell] == 97:
        self.board[y_cell][x_cell] = 7
    elif self.board[y_cell][x_cell] == 98:
        self.board[y_cell][x_cell] = 8

start_settings = start_screen()
start_list = start_settings.split()
a = int(start_list[0])
b = int(start_list[1])
n_mine = int(start_list[2])
if n_mine >= (a * b) - 9:
    n_mine = (a * b) - 9
print(n_mine)
size = width, height = (20 + a * 50), (170 + b * 50)
screen = pygame.display.set_mode(size)
screen.fill((0, 0, 0))
board = Board(a, b)
screen.fill((0, 0, 0))
board.render()
pygame.display.flip()
quit = False
flag = True
running = True
#основной игровой цикл
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
            quit = True
        if event.type == pygame.MOUSEBUTTONDOWN and event.button == 1:
            board.check(event)
            if flag:
                board.get_mine()
                flag = False
        if event.type == pygame.MOUSEBUTTONDOWN and event.button == 3:
            board.flag(event)
    screen.fill((0, 0, 0))
    board.render()
    if board.render_mine() == 'win':
        running = False
    elif board.render_mine() == 'lose':
        running = False
    pygame.display.flip()
#вывод экранов конца игры и выход
if quit == True:
    pygame.quit()
if board.render_mine() == 'win':
    size_finish = 400, 200
    screen = pygame.display.set_mode(size_finish)
    screen.fill((0, 191, 255))
    win()
elif board.render_mine() == 'lose':
    size_finish = 400, 200
    screen = pygame.display.set_mode(size_finish)

```



```
screen.fill((255, 255, 255))  
lose()
```

Анализ результатов



