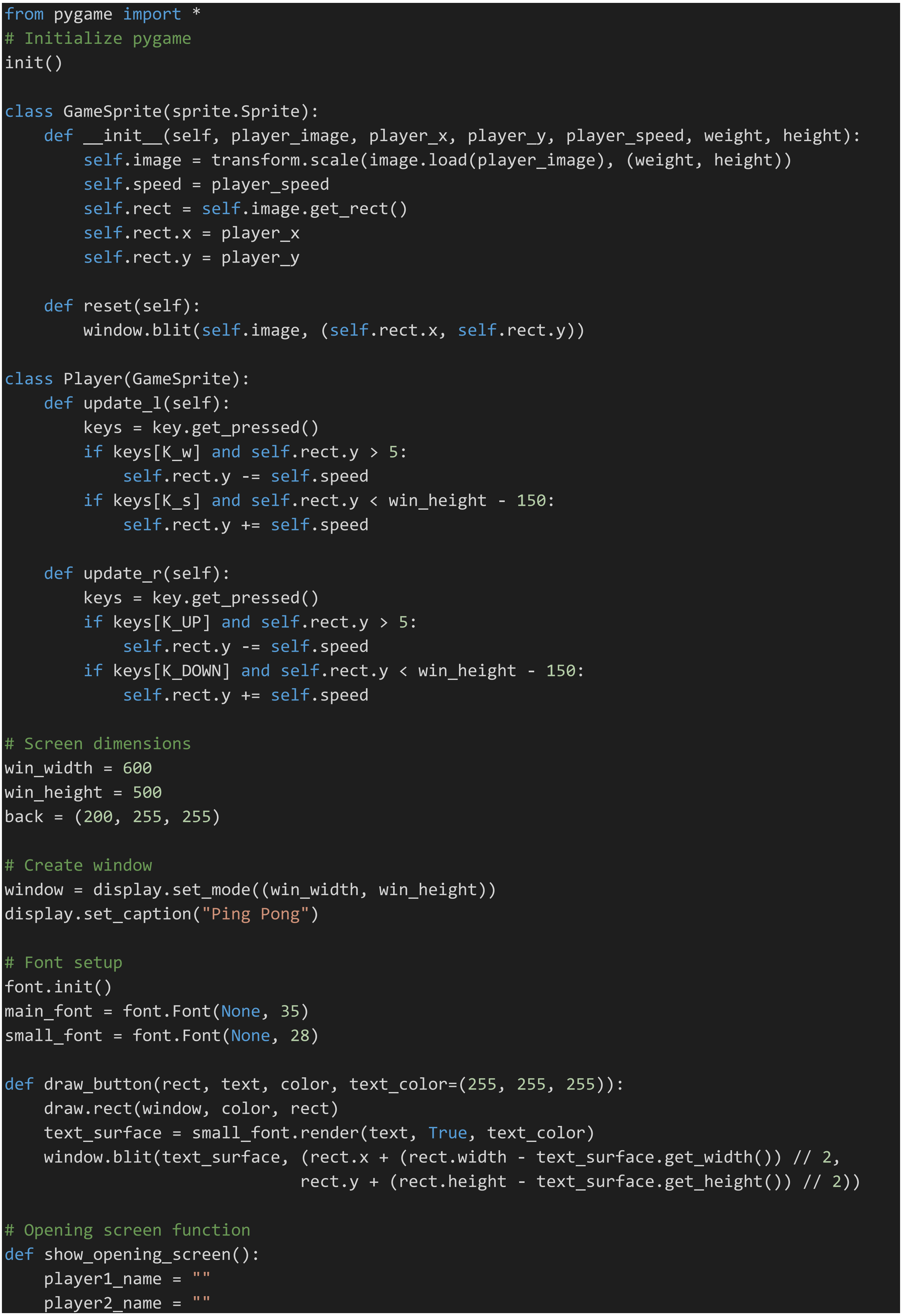
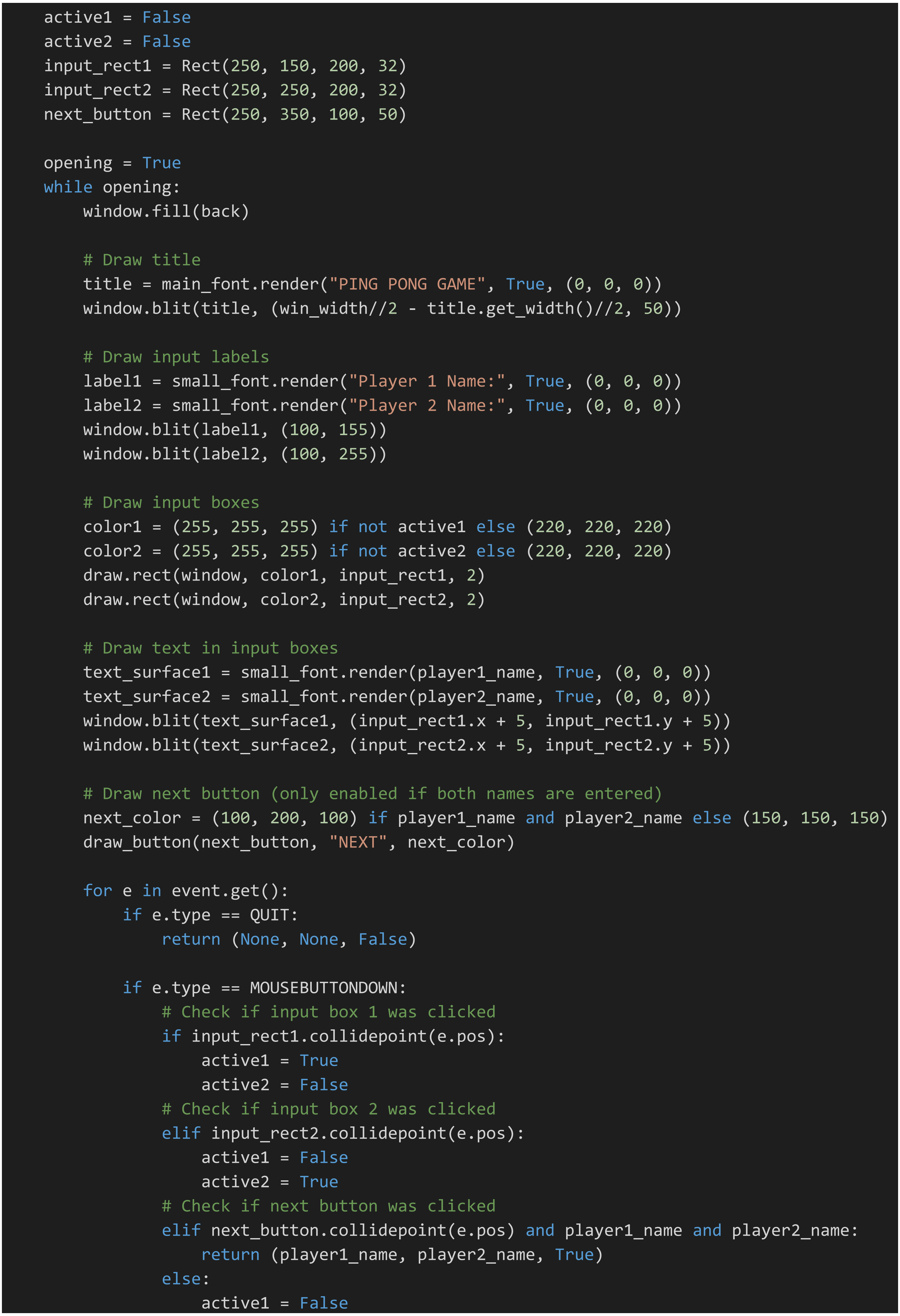


```python
from pygame import *
# Initialize pygame
init()

class GameSprite(sprite.Sprite):
    def __init__(self, player_image, player_x, player_y, player_speed, weight, height):
        self.image = transform.scale(image.load(player_image), (weight, height))
        self.speed = player_speed
        self.rect = self.image.get_rect()
        self.rect.x = player_x
        self.rect.y = player_y

    def reset(self):
        window.blit(self.image, (self.rect.x, self.rect.y))

class Player(GameSprite):
    def update_l(self):
        keys = key.get_pressed()
        if keys[K_w] and self.rect.y > 5:
            self.rect.y -= self.speed
        if keys[K_s] and self.rect.y < win_height - 150:
            self.rect.y += self.speed

    def update_r(self):
        keys = key.get_pressed()
        if keys[K_UP] and self.rect.y > 5:
            self.rect.y -= self.speed
        if keys[K_DOWN] and self.rect.y < win_height - 150:
            self.rect.y += self.speed

# Screen dimensions
win_width = 600
win_height = 500
back = (200, 255, 255)

# Create window
window = display.set_mode((win_width, win_height))
display.set_caption("Ping Pong")

# Font setup
font.init()
main_font = font.Font(None, 35)
small_font = font.Font(None, 28)

def draw_button(rect, text, color, text_color=(255, 255, 255)):
    draw.rect(window, color, rect)
    text_surface = small_font.render(text, True, text_color)
    window.blit(text_surface, (rect.x + (rect.width - text_surface.get_width()) // 2,
                               rect.y + (rect.height - text_surface.get_height()) // 2))

# Opening screen function
def show_opening_screen():
    player1_name = ""
    player2_name = ""
```



```
    active1 = False
    active2 = False
    input_rect1 = Rect(250, 150, 200, 32)
    input_rect2 = Rect(250, 250, 200, 32)
    next_button = Rect(250, 350, 100, 50)

    opening = True
    while opening:
        window.fill(back)

        # Draw title
        title = main_font.render("PING PONG GAME", True, (0, 0, 0))
        window.blit(title, (win_width//2 - title.get_width()//2, 50))

        # Draw input labels
        label1 = small_font.render("Player 1 Name:", True, (0, 0, 0))
        label2 = small_font.render("Player 2 Name:", True, (0, 0, 0))
        window.blit(label1, (100, 155))
        window.blit(label2, (100, 255))

        # Draw input boxes
        color1 = (255, 255, 255) if not active1 else (220, 220, 220)
        color2 = (255, 255, 255) if not active2 else (220, 220, 220)
        draw.rect(window, color1, input_rect1, 2)
        draw.rect(window, color2, input_rect2, 2)

        # Draw text in input boxes
        text_surface1 = small_font.render(player1_name, True, (0, 0, 0))
        text_surface2 = small_font.render(player2_name, True, (0, 0, 0))
        window.blit(text_surface1, (input_rect1.x + 5, input_rect1.y + 5))
        window.blit(text_surface2, (input_rect2.x + 5, input_rect2.y + 5))

        # Draw next button (only enabled if both names are entered)
        next_color = (100, 200, 100) if player1_name and player2_name else (150, 150, 150)
        draw_button(next_button, "NEXT", next_color)

        for e in event.get():
            if e.type == QUIT:
                return (None, None, False)

            if e.type == MOUSEBUTTONDOWN:
                # Check if input box 1 was clicked
                if input_rect1.collidepoint(e.pos):
                    active1 = True
                    active2 = False
                # Check if input box 2 was clicked
                elif input_rect2.collidepoint(e.pos):
                    active1 = False
                    active2 = True
                # Check if next button was clicked
                elif next_button.collidepoint(e.pos) and player1_name and player2_name:
                    return (player1_name, player2_name, True)
                else:
                    active1 = False
```

```python
                    active2 = False

            if e.type == KEYDOWN:
                if active1:
                    if e.key == K_BACKSPACE:
                        player1_name = player1_name[:-1]
                    else:
                        player1_name += e.unicode
                elif active2:
                    if e.key == K_BACKSPACE:
                        player2_name = player2_name[:-1]
                    else:
                        player2_name += e.unicode

        display.update()
        clock.tick(30)
from random import randint
# Main game function
def main_game(player1_name, player2_name):
    # Create game objects
    racket1 = Player('racket.png', 30, 200, 4, 50, 150)
    racket2 = Player('racket.png', 520, 200, 4, 50, 150)
    ball = GameSprite('tenis_ball.png', 200, 200, 4, 50, 50)

    game = True
    finish = False
    clock = time.Clock()
    FPS = 60

    lose1 = main_font.render(f'{player1_name} Lose!', True, (180, 0, 0))
    lose2 = main_font.render(f'{player2_name} Lose!', True, (180, 0, 0))

    # Player name displays
    player1_text = small_font.render(player1_name, True, (0, 0, 0))
    player2_text = small_font.render(player2_name, True, (0, 0, 0))

    speed_x = 3
    speed_y = 3

    while game:
        for e in event.get():
            if e.type == QUIT:
                return False  # Return False to quit entirely

        if not finish:
            window.fill(back)

            # Draw player names
            window.blit(player1_text, (30, 20))
            window.blit(player2_text, (win_width - 30 - player2_text.get_width(), 20))

            racket1.update_l()
            racket2.update_r()
            ball.rect.x += speed_x
            ball.rect.y += speed_y

            if sprite.collide_rect(racket1, ball) or sprite.collide_rect(racket2, ball):
                speed_x *= -1
                speed_y *= randint(-1,1)

            if ball.rect.y > win_height - 50 or ball.rect.y < 0:
                speed_y *= -1

            if ball.rect.x < 0:
                finish = True
                window.blit(lose1, (200, 200))

            if ball.rect.x > win_width:
                finish = True
                window.blit(lose2, (200, 200))

            racket1.reset()
            racket2.reset()
            ball.reset()
        else:
            # Show game over screen with option to return to main menu
            return_button = Rect(win_width//2 - 100, 300, 200, 50)
            draw_button(return_button, "Main Menu", (100, 150, 200))

            # Check for click on return button
            mouse_pos = mouse.get_pos()
            mouse_click = mouse.get_pressed()

            if mouse_click[0] and return_button.collidepoint(mouse_pos):
                return True  # Return True to go back to main menu

            # Reset mouse click state to prevent immediate return
            mouse.get_pressed()

        display.update()
        clock.tick(FPS)


# Run the game
clock = time.Clock()
running = True
while running:
    # Show opening screen
    player1, player2, start_game = show_opening_screen()

    if not start_game:
        running = False
        break

    # Start the game
    return_to_menu = main_game(player1, player2)

    # If return_to_menu is False, quit the game
    if not return_to_menu:
        running = False
quit()
```