solution

November 25, 2022

[ ]:
```
import pygame
pygame.init()


WIDTH, HEIGHT = 1200, 768
WIN = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Pong")

FPS = 60

WHITE = (255, 255, 255)
BLACK = (0, 0, 0)

PADDLE_WIDTH, PADDLE_HEIGHT = 20, 100
BALL_RADIUS = 7

SCORE_FONT = pygame.font.SysFont("comicsans", 50)
WINNING_SCORE = 10


class Paddle:
    COLOR = WHITE
    VEL = 4

    def __init__(self, x, y, width, height):
        self.x = self.original_x = x
        self.y = self.original_y = y
        self.width = width
        self.height = height

    def draw(self, win):
        pygame.draw.rect(
            win, self.COLOR, (self.x, self.y, self.width, self.height))

    def move(self, up=True):
        if up:
            self.y -= self.VEL
```

```
        else:
            self.y += self.VEL

    def reset(self):
        self.x = self.original_x
        self.y = self.original_y


class Ball:
    MAX_VEL = 10
    COLOR = WHITE

    def __init__(self, x, y, radius):
        self.x = self.original_x = x
        self.y = self.original_y = y
        self.radius = radius
        self.x_vel = self.MAX_VEL
        self.y_vel = 0

    def draw(self, win):
        pygame.draw.circle(win, self.COLOR, (self.x, self.y), self.radius)

    def move(self):
        self.x += self.x_vel
        self.y += self.y_vel

    def reset(self):
        self.x = self.original_x
        self.y = self.original_y
        self.y_vel = 0
        self.x_vel *= -1


def draw(win, paddles, ball, left_score, right_score):
    win.fill(BLACK)

    left_score_text = SCORE_FONT.render(f"{left_score}", True, WHITE)
    right_score_text = SCORE_FONT.render(f"{right_score}", True, WHITE)
    win.blit(left_score_text, (WIDTH//4 - left_score_text.get_width()//2, 20))
    win.blit(right_score_text, (WIDTH * (3/4) -
                                right_score_text.get_width()//2, 20))

    for paddle in paddles:
        paddle.draw(win)

    for i in range(10, HEIGHT, HEIGHT//20):
        if i % 2 == 1:
```

```
            continue
        pygame.draw.rect(win, WHITE, (WIDTH//2 - 5, i, 10, HEIGHT//20))

    ball.draw(win)
    pygame.display.update()


def handle_collision(ball, left_paddle, right_paddle):
    if ball.y + ball.radius >= HEIGHT:
        ball.y_vel *= -1
    elif ball.y - ball.radius <= 0:
        ball.y_vel *= -1

    if ball.x_vel < 0:
        if ball.y >= left_paddle.y and ball.y <= left_paddle.y + left_paddle.
 ↪height:
            if ball.x - ball.radius <= left_paddle.x + left_paddle.width:
                ball.x_vel *= -1

                middle_y = left_paddle.y + left_paddle.height / 2
                difference_in_y = middle_y - ball.y
                reduction_factor = (left_paddle.height / 2) / ball.MAX_VEL
                y_vel = difference_in_y / reduction_factor
                ball.y_vel = -1 * y_vel

    else:
        if ball.y >= right_paddle.y and ball.y <= right_paddle.y + right_paddle.
 ↪height:
            if ball.x + ball.radius >= right_paddle.x:
                ball.x_vel *= -1

                middle_y = right_paddle.y + right_paddle.height / 2
                difference_in_y = middle_y - ball.y
                reduction_factor = (right_paddle.height / 2) / ball.MAX_VEL
                y_vel = difference_in_y / reduction_factor
                ball.y_vel = -1 * y_vel


def handle_paddle_movement(keys, left_paddle, right_paddle):
    if keys[pygame.K_w] and left_paddle.y - left_paddle.VEL >= 0:
        left_paddle.move(up=True)
    if keys[pygame.K_s] and left_paddle.y + left_paddle.VEL + left_paddle.
 ↪height <= HEIGHT:
        left_paddle.move(up=False)

    if keys[pygame.K_UP] and right_paddle.y - right_paddle.VEL >= 0:
        right_paddle.move(up=True)
```

```
    if keys[pygame.K_DOWN] and right_paddle.y + right_paddle.VEL + right_paddle.
 ↪height <= HEIGHT:
        right_paddle.move(up=False)


def main():
    run = True
    clock = pygame.time.Clock()

    left_paddle = Paddle(10, HEIGHT//2 - PADDLE_HEIGHT //
                         2, PADDLE_WIDTH, PADDLE_HEIGHT)
    right_paddle = Paddle(WIDTH - 10 - PADDLE_WIDTH, HEIGHT //
                          2 - PADDLE_HEIGHT//2, PADDLE_WIDTH, PADDLE_HEIGHT)
    ball = Ball(WIDTH // 2, HEIGHT // 2, BALL_RADIUS)

    left_score = 0
    right_score = 0

    while run:
        clock.tick(FPS)
        draw(WIN, [left_paddle, right_paddle], ball, left_score, right_score)

        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                run = False
                break

        keys = pygame.key.get_pressed()
        handle_paddle_movement(keys, left_paddle, right_paddle)

        ball.move()
        handle_collision(ball, left_paddle, right_paddle)

        if ball.x < 0:
            right_score += 1
            ball.reset()
        elif ball.x > WIDTH:
            left_score += 1
            ball.reset()

        won = False
        if left_score >= WINNING_SCORE:
            won = True
            win_text = "Left Player Won!"
        elif right_score >= WINNING_SCORE:
            won = True
            win_text = "Right Player Won!"
```

```
        if won:
            text = SCORE_FONT.render(win_text, 1, WHITE)
            WIN.blit(text, (WIDTH//2 - text.get_width() //
                            2, HEIGHT//2 - text.get_height()//2))
            pygame.display.update()
            pygame.time.delay(5000)
            ball.reset()
            left_paddle.reset()
            right_paddle.reset()
            left_score = 0
            right_score = 0

    pygame.quit()


if __name__ == '__main__':
    main()
```