

Preparation for Pong Tutorial

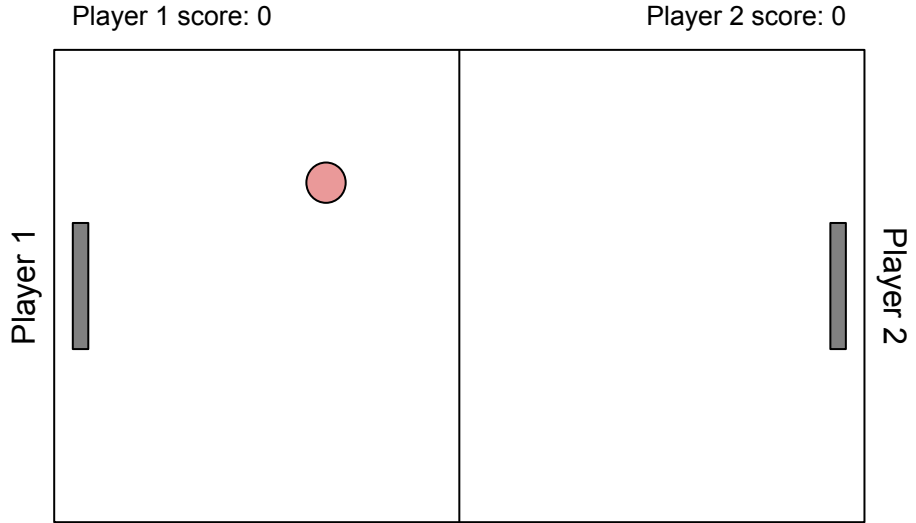
Dienstag, 30.08.22

16:00-17:15 Uhr



Spielregeln

- Berührt der Ball den rechten/ linken Rand, so verliert der Spieler 1/2 einen Punkt.
- Die Spieler können die Paddle bewegen um den Ball daran zu hindern, die Linie zu berühren.
- Der Ball wird jeder Mal schneller, wenn er ein Paddle berührt.
- Das Spiel endet, wenn ein festgelegter Score erreicht wird.



Welche Klassen werden benötigt?

→ Gemeinsames Brainstorming an der Tafel/ Whiteboard

Now let's walk through the code!

Öffne den folgenden Link um auf das Repository zuzugreifen:

<LINK TO REPO>

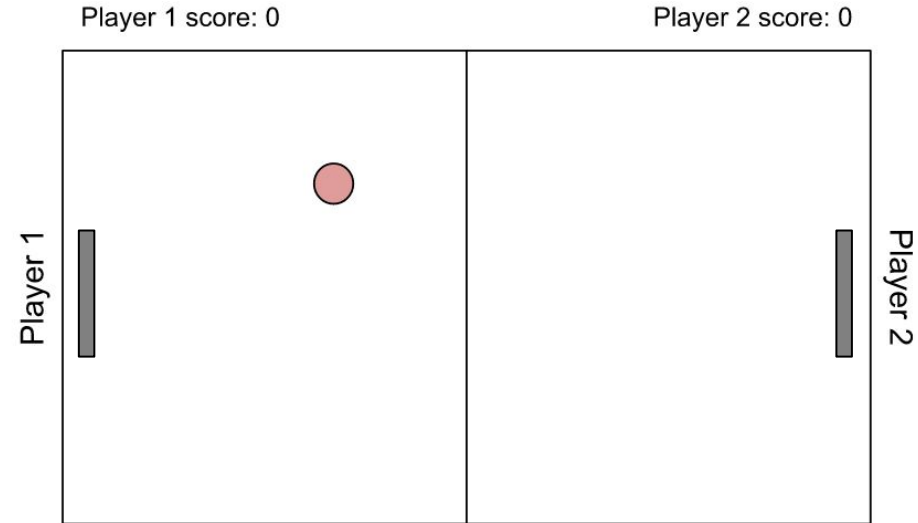
Clone the repo to your directory:

```
git clone <LINK TO REPO>
```

Spielumgebung

Für die Spielumgebung, müssen folgende Objekte erstellt werden:

- background
- ball
- middle line
- scores
- paddles



Spielumgebung - Blick in den Code!

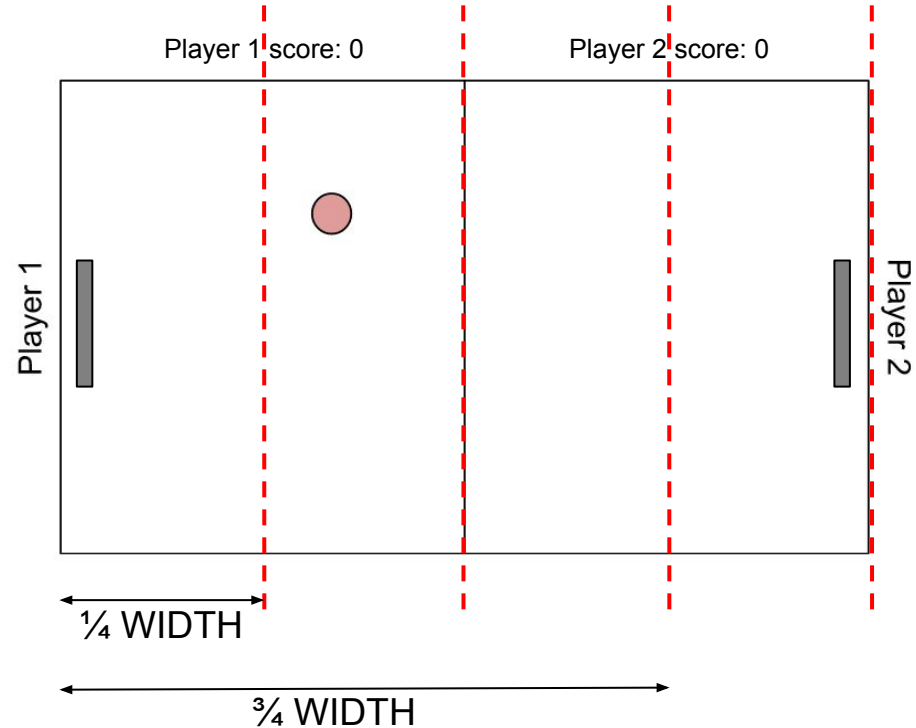
```
32     def draw(self, win, score_font):
33         win.fill(BLACK) # draw background
34
35         self.ball.draw(win) # draw ball
36
37         for i in range(10, HEIGHT, HEIGHT//20): # drawing the dots in the
            middle which are small rectangles
38             if i % 2 == 1:
39                 continue
40             pygame.draw.rect(win, WHITE, (WIDTH//2 - 5, i, 10, HEIGHT//20))
            # width of the rect is 5, we start it not right in the middle,
            but minus 5
```

Spielumgebung

Für die Spielumgebung, müssen folgende Objekte erstellt werden:

- background
- ball
- middle line
- **scores**
- paddles

Um die Scores zu positionieren, können Sie den Hintergrund in 4 Teile aufteilen:

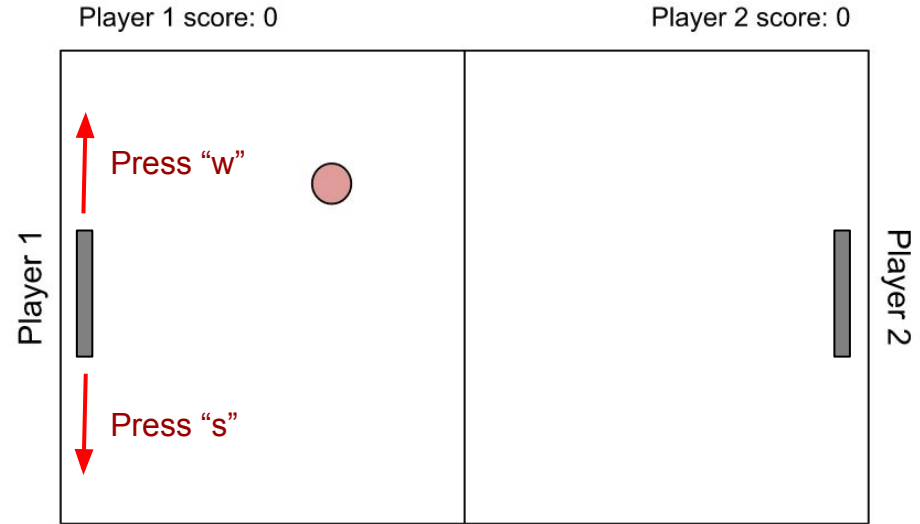


Spielumgebung - Blick in den Code!

```
42     left_score_text = score_font.render(f"{self.left_score}", 1, WHITE)
43     win.blit(left_score_text, (WIDTH * (1/4) - left_score_text.get_width
44                                ())/2, 20))
45
46     """
47     TODO: Draw the right score
48     Hint: This is similar to the left score, but be careful of the
49     coordinates
50     """
51
52     """
53     TODO: Go to the function draw_paddles() and draw the paddles
54     """
55     self.draw_paddles(win)
```

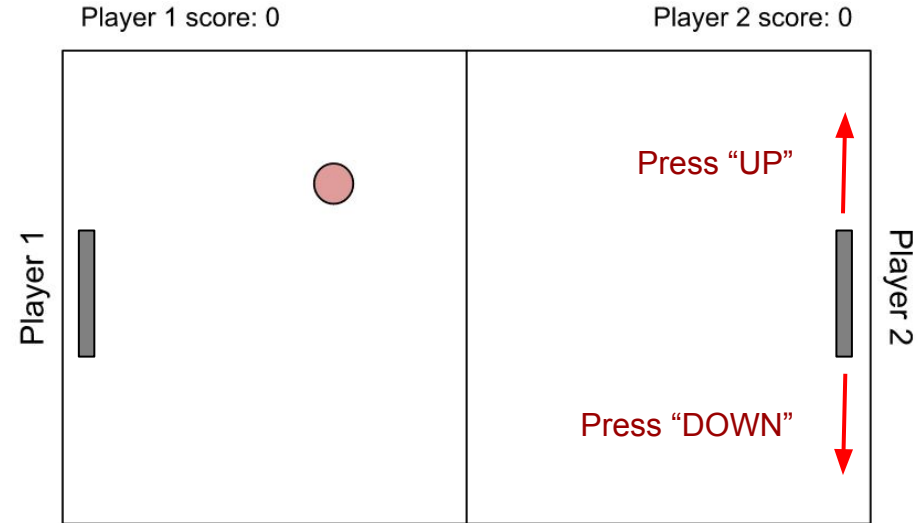

Paddle Bewegung

Spieler 1 (links) benutzt die Tasten "w" und "s", um das Paddel auf und ab zu bewegen



Paddle Bewegung

Spieler 2 (rechts) benutzt die Tasten "UP" und "DOWN", um das Paddel nach oben und unten zu bewegen



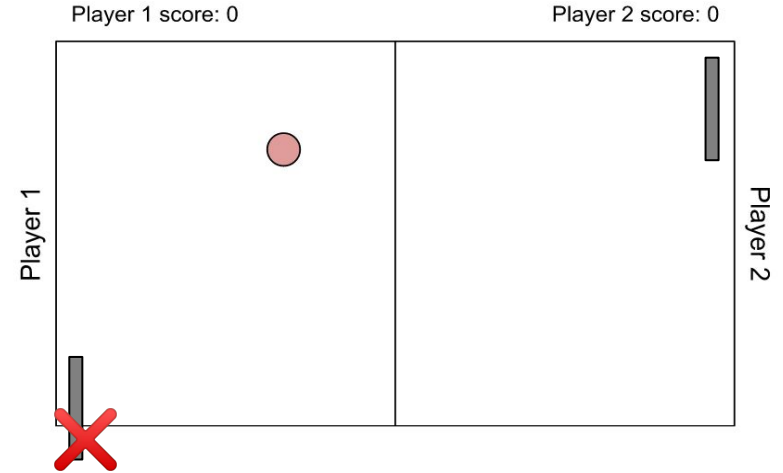
Paddle Bewegung

Die Paddles können nicht außerhalb der Grenzen bewegt werden!

Für Player 1:

```
if (key_w is pressed) and  
(paddle_position + paddle_velocity >= 0):  
    paddle_moves_up()
```

```
if (key_s is pressed) and  
(paddle_position + pad_h +  
paddle_velocity <= background_height):  
    paddle_moves_down()
```



pad_y_pos



pad_y_pos + pad_h

Paddle Bewegung - Blick in den Code!

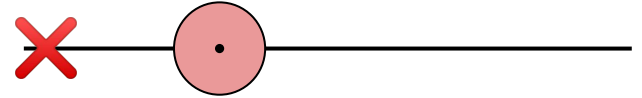
```
14 def handle_paddle_movement(keys, game: Game):  
15     if keys[pygame.K_w] and game.left_paddle.y - game.left_paddle.VEL >=  
    0 : # check if you reached the borders  
16         game.left_paddle.move(up=True)  
17     if keys[pygame.K_s] and game.left_paddle.y + game.left_paddle.VEL +  
    game.left_paddle.height <= HEIGHT: # take into account paddle height  
    for the amount of movement  
18         game.left_paddle.move(up=False)  
19     """"  
20     TODO: Make the right paddle move.  
21     Add here the conditions for the right paddle, moving with arrow keys  
    up and down  
22     """"
```

Kollision mit der Außenkante

Der Ball kollidiert mit der Außenkante, wenn sein Rand die Kante berührt

```
if (ball_y_pos - ball_r <=
    top_border_position):
    ball_collides = True
```

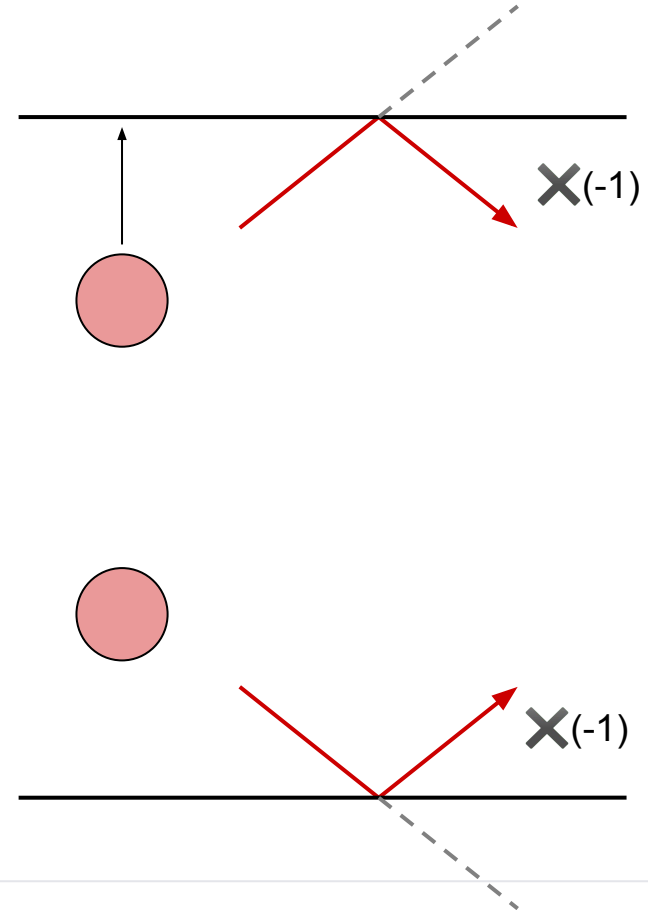
```
if (ball_y_pos + ball_r >=
    bottom_border_position):
    ball_collides = True
```



Kollision mit der Außenkante

Wenn der Ball auf die obere (oder untere) Grenze trifft:

- Die Geschwindigkeit bleibt gleich
- Aber sie bewegt sich auf der y-Achse in die entgegengesetzte Richtung



Kollision mit dem Paddel

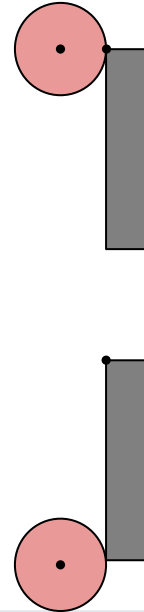
(gleich für links und rechts)

y-Achse:

```
ball_y_pos >= pad_y_pos
```

Und

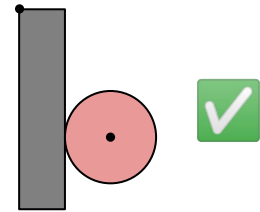
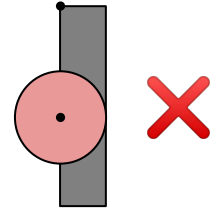
```
ball_y_pos <= pad_y_pos + pad_h
```



Kollision mit dem linken Paddel

x-Achse

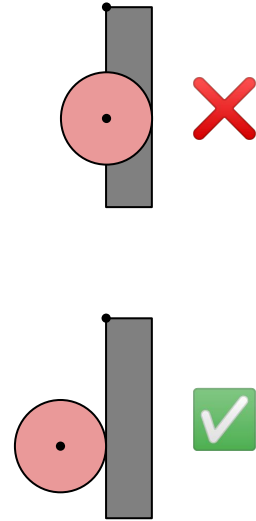
```
ball_x_pos - ball_r >= pad_x_pos + pad_w
```



Kollision mit dem rechten Paddel

x-Achse:

```
ball_x_pos + ball_r <= pad_x_pos
```



Geschwindigkeit des Balls

(gleich für links und rechts)

x-Achse:

```
ball_x_velocity *= -1
```

y-Achse:

```
reduction_factor = (pad_h / 2) / | ball_x_velocity |  
y_vel = y_difference / reduction_factor  
ball_y_velocity = -1 * y_vel
```

