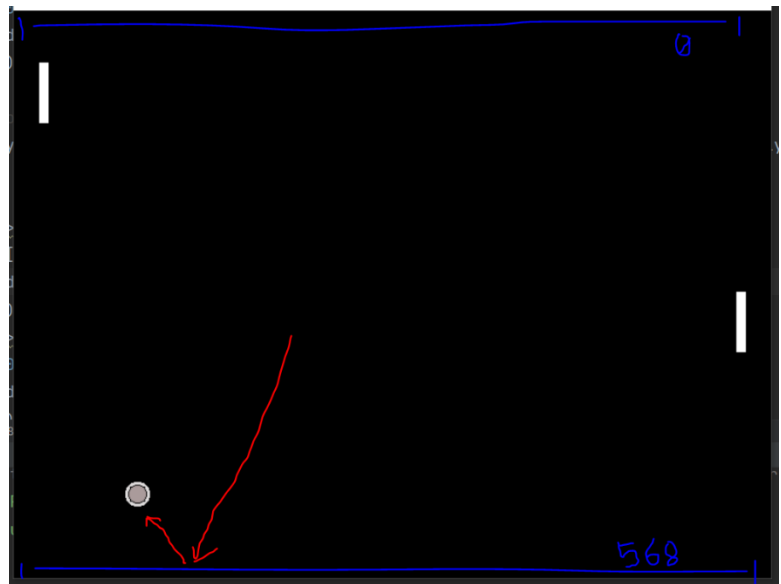


## Coding documentation

- day 1

```
32 def corner(y):
33     global ballyd
34     if y <= 0 or y >= 568:
35         ballyd *= -1
36     return ballyd
```

```
93 # display ball and its changed position #
94 ballx += ballxd
95 bally += ballyd
96 corner(bally)
97 bounce()
98 ball(ballx, bally)
```



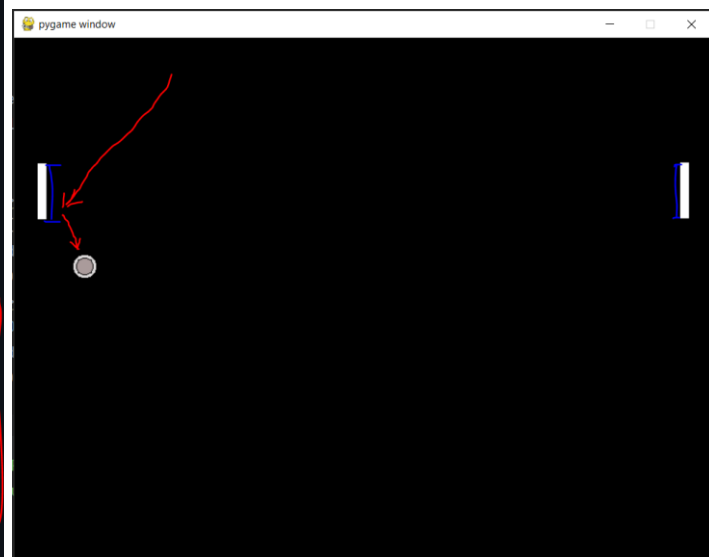
(Figure 1)

- The function “corner” is used to bounce ball between corners.

```
37 def bounce():
38     global player1x,player1y,player2x,player2y,ballx,bally,ballxd,ballyd
39     if ((ballx <= player1x + 37) and (ballx >= player1x + 28)) and ((bally+28 >= player1y) and (bally <= player1y + 64)):
40         ballx = 37
41         ballxd *= -1
42     if ((bally + 28 >= player1y + 24) and (bally <= player1y + 40)):
43         angleneer = [0.03, -0.03, 0.05, -0.05]
44         ballyd = random.choice(angleneer)
45         print(ballyd)
46     elif ((bally+28 >= player1y) and (bally <= player1y + 64)):
47         anglefar = [0.1, -0.1, 0.15, -0.15]
48         ballyd = random.choice(anglefar)
49         print(ballyd)
50
51     br = 28 # br means "bottom right"
52     if ((ballx+br <= player2x + 37) and (ballx+br >= player2x + 28)) and ((bally+28 >= player2y) and (bally <= player2y + 64)):
53         ballx = 735
54         ballxd *= -1
55     if ((bally + 28 >= player2y + 24) and (bally <= player2y + 40)):
56         angleneer = [0.03, -0.03, 0.05, -0.05]
57         ballyd = random.choice(angleneer)
58         print(ballyd)
59     elif ((bally+28 >= player2y) and (bally <= player2y + 64)):
60         anglefar = [0.1, -0.1, 0.15, -0.15]
61         ballyd = random.choice(anglefar)
62         print(ballyd)
63
64     return ballyd,ballxd
```

Player 1

Player 2



```
93 # display ball and its changed position #
94 ballx += ballxd
95 bally += ballyd
96 corner(bally)
97 bounce()
98 ball(ballx, bally)
```

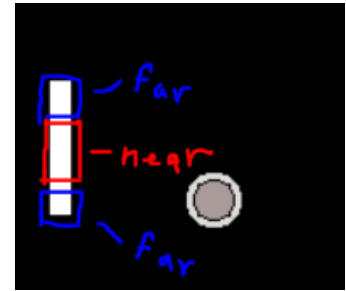
(Figure 2)

- The function “bounce” is used to bounce ball from both players.

```

42 if ((bally + 28 >= player1y + 24) and (bally <= player1y + 40)):
43     near
44     anglenear = [0.03,-0.03,0.05,-0.05]
45     ballyd = random.choice(anglenear)
46     print(ballyd)
47 elif ((bally+28 >= player1y)and(bally <= player1y + 64)):
48     far
49     anglefar = [0.1,-0.1,0.15,-0.15]
50     ballyd = random.choice(anglefar)
51     print(ballyd)

```



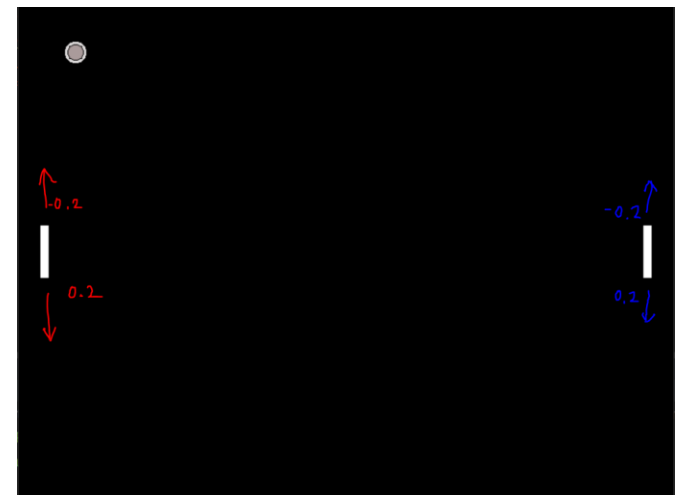
(Figure 3)

- The if and elif conditions inside the “bounce” function will change the ball movement as it hits the player’s paddle.
- The “near” part of paddle will give the ball y direction to either 0.03, -0.03, 0.05 and -0.05 randomly
- The “far” part of paddle will give the ball y direction to either 0.1, -0.1, 0.15 and -0.15 randomly

```

67 while gamerun == True:
68     for event in pygame.event.get():
69         if event.type == pygame.QUIT:
70             gamerun = False
71
72     # ===== pressed button ===== #
73     if event.type == pygame.KEYDOWN:
74         if event.key == pygame.K_w:
75             player1yd = -0.2
76         if event.key == pygame.K_UP:
77             player2yd = -0.2
78         if event.key == pygame.K_s:
79             player1yd = 0.2
80         if event.key == pygame.K_DOWN:
81             player2yd = 0.2
82
83     # ===== released button ===== #
84     if event.type == pygame.KEYUP:
85         if event.key == pygame.K_w or event.key == pygame.K_s:
86             player1yd = 0
87         if event.key == pygame.K_UP or event.key == pygame.K_DOWN:
88             player2yd = 0
89
90

```



(Figure 4)

- The keys “W”, “S”, “Up arrow” and “Down arrow” is assigned to control the player.

# that’s all of the important updates in day 1 :) #