

PyGame Pong Tutorial - Part 6

Contents

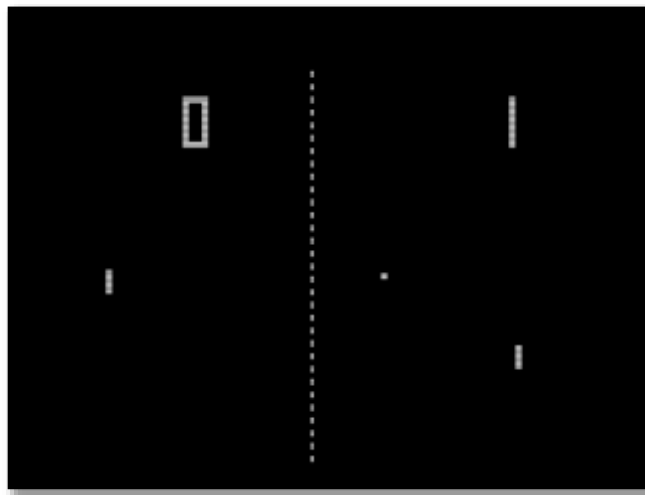
PyGame Pong Tutorial - Part 6	1
Preview of the Game	1
Scoring Time	1
Check Collision.....	3
Game Loop.....	4
Assignment Submission.....	5

Time required: 30 minutes

Preview of the Game

Atari. - the year: 1973 - the date: - November 29th - The game is Pong.

[Pong Demo Video](#)



Scoring Time

Taking names keeping score.

1. Save **pong_5.py** as **pong_6.py**
2. Add the following code to setup the score font and player score tracking.

```
39     # Set up player paddles
40     self.player = Paddle(
41         5, # x coordinate for player paddler
42         (HEIGHT - 100) // 2, # y coordinate
43     )
44
45     self.computer = Paddle(
46         WIDTH - 15, # x coordinate for computer paddle
47         (HEIGHT - 100) // 2, # y coordinate
48     )
49     self.computer_speed = 3
50
51     self.score_font = pygame.font.SysFont("freesansbold", 18)
52     self.player_score = 0
53     self.computer_score = 0
```

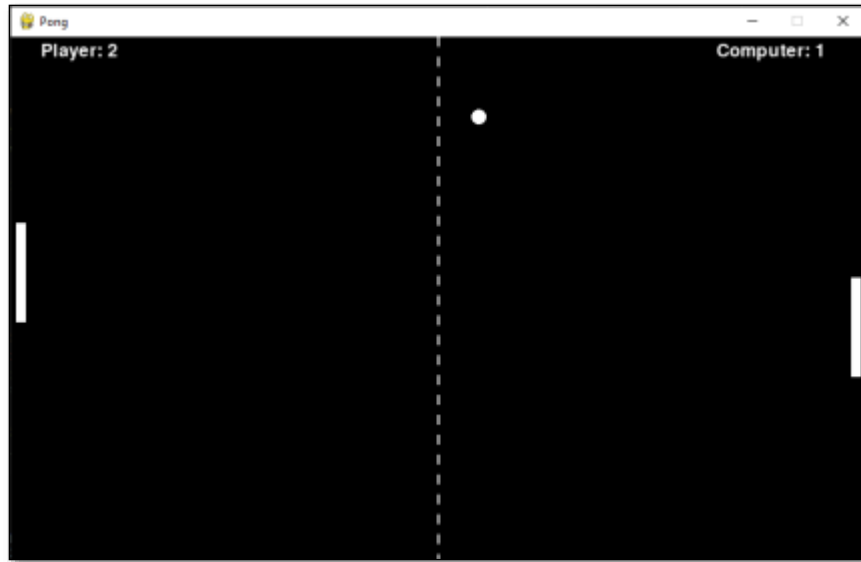
Check Collision

```
118 # ----- CHECK COLLISION ----- #
119 def check_collision(self):
120     """Check for all collisions"""
121     # Check for collision with left or right wall
122     # Subtract ball radius to bounce off the edge of the ball
123     if self.ball.left < 0 or self.ball.right >= WIDTH:
124
125         # Reverse y direction multiply by -1
126         self.ball_speed_x = self.ball_speed_x * -1
127
128     # Check for collision with top or bottom wall
129     if self.ball.top < 0 or self.ball.bottom >= HEIGHT:
130
131         # Reverse y direction multiply by -1
132         self.ball_speed_y = self.ball_speed_y * -1
133
134     # Ball collision with paddles
135     if self.ball.colliderect(self.player):
136         # Reverse ball direction
137         self.ball_speed_x *= -1
138         self.player_score += 1
139
140     elif self.ball.colliderect(self.computer):
141         # Reverse ball direction
142         self.ball_speed_x *= -1
143         self.computer_score += 1
```

Game Loop

```
222     # Draw a rectangle for the computer's paddle
223     # on the screen using Pygame's draw function
224     pygame.draw.rect(
225         self.surface, # Surface to draw on
226         BALL_COLOR, # Color to draw with
227         self.computer, # rect image object to draw
228     )
229
230     # Render the player's score text using the specified font,
231     # color, and score value
232     player_score = self.score_font.render(
233         "Player: " + str(self.player_score), True, BALL_COLOR
234     )
235
236     # Render the computer's score text using the specified font,
237     # color, and score value
238     computer_score = self.score_font.render(
239         "Computer: " + str(self.computer_score), True, BALL_COLOR
240     )
241
242     # Display the player's score text on the game surface
243     # at the specified position
244     self.surface.blit(player_score, (30, 5))
245     # Display the computer's score text on the game surface
246     # at the specified position
247     self.surface.blit(computer_score, (WIDTH - 150, 5))
248
```

Example run:



The game works! We need some sound effects and a game over menu.
Coming up next.

Assignment Submission

1. Attach a screenshot showing the operation of the program.
2. Zip up the program files folder and submit in Blackboard.