PyGame Pong Tutorial - Part 4

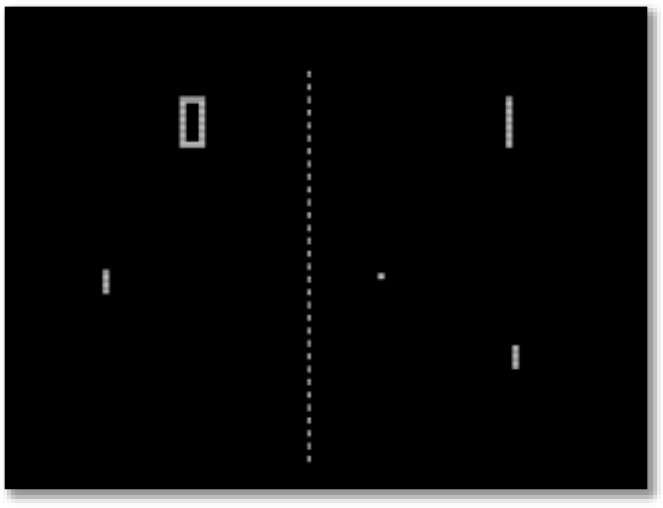
Contents

Time required: 30 minutes

Preview of the Game

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

Pong Demo Video



Yes, we are finally going to make something move. By changing the (x, y) values each time through the game loop, we animate our ball.

Bouncing Ball

1. Save **pong_3.py** as **pong_4.py**

2. Add the following code to check for collisions between the ball and the walls.

```
101     # ---------------------- CHECK COLLISION ------------------------------- #
102     def check_collision(self):
103         """Check for collisions with all 4 walls"""
104         # Check for collision with left or right wall
105         if self.ball.left < 0 or self.ball.right >= WIDTH:
106 
107             # Reverse y direction multiply by -1
108             self.ball_speed_x = self.ball_speed_x * -1
109 
110         # Check for collision with top or bottom wall
111         if self.ball.top < 0 or self.ball.bottom >= HEIGHT:
112 
113             # Reverse y direction multiply by -1
114             self.ball_speed_y = self.ball_speed_y * -1
```

When you multiply a number by -1, it changes the sign. For example, 2 * -1 = -2. This reverses the direction of the ball.

Draw Net

Let’s draw a net.

```
116     # ----------------------- DRAW NET ---------------------------------- #
117     def draw_net(self):
118         # Define the width of the net lines
119         net_width = 2
120 
121         # Loop through the height of the game screen
122         # with a step of 20 pixels
123         for i in range(0, HEIGHT, 20):
124 
125             # Draw a rectangle representing a part of the net
126             pygame.draw.rect(
127                 self.surface,  # Surface to draw on
128                 BALL_COLOR,  # Color of the rectangle (white)
129                 (  # Rectangle coordinates and size
130                     # X-coordinate of the left corner of the rectangle
131                     WIDTH // 2 - net_width // 2,
132                     i,  # Y-coordinate of top corner of rectangle
133                     net_width,  # Width of the rectangle
134                     10,  # Height of the rectangle
135                 ),
136             )
```

Call the check_collision and draw_net method each time through the game loop.

```
138     # ---------------------------- GAME LOOP ------------------------------ #
139     def game_loop(self):
140         """Infinite Game Loop"""
141         while True:
142             self.check_events()
143             self.check_collision()
144             self.update_ball()
145 
146             # ------------------- DRAW SURFACE ---------------------------- #
147             # Draw everything on the surface first
148             # Fill the display surface with a background color
149             # to clear the previous frame
150             self.surface.fill(BG_COLOR)
151 
152             self.draw_net()
153 
154             # Draw ball
155             pygame.draw.ellipse(
156                 self.surface,  # Surface to draw on
157                 BALL_COLOR,  # Color to draw with
158                 self.ball,  # Rect image object to draw
159             )
160 
161             # ------------------ UPDATE DISPLAY ---------------------------- #
162             # From surface, update Pygame display to reflect any changes
163             pygame.display.update()
164 
165             # Cap game speed at 60 frames per second
166             self.clock.tick(60)
```

Set Random Ball Direction

When the ball is initially drawn, let's randomize which direction it goes when it starts moving.

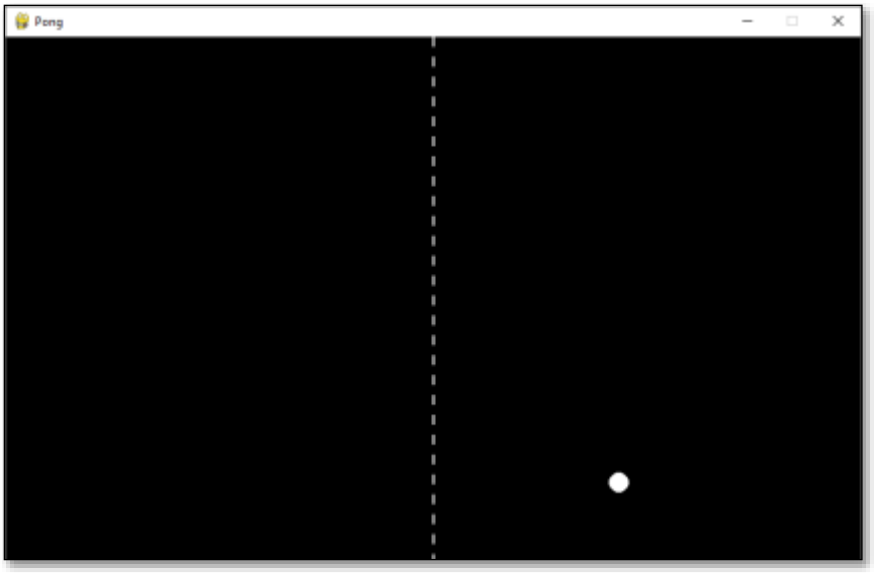
Add the following method to the pong class.

```
    # ----------------------- SET BALL DIRECTION ------------------------- #
    def set_ball_direction(self):
        """Set random initial ball direction along the x and y axis"""
        # Randomly determine the initial direction of the ball
        # along the x-axis (left or right)
        ball_direction_x = randint(0, 1)

        # If the randomly chosen direction is 0 (left),
        # set the horizontal speed of the ball to move to the right
        if ball_direction_x == 0:
            self.ball_speed_x = 3

        # If the randomly chosen direction is 1 (right),
        # set the horizontal speed of the ball to move to the left
        else:
            self.ball_speed_x = -3

        # Randomly determine the initial direction of the ball
        # along the y-axis (up or down)
        ball_direction_y = randint(0, 1)

        # If the randomly chosen direction is 0 (up),
        # set the vertical speed of the ball to move downwards
        if ball_direction_y == 0:
            self.ball_speed_y = 3

        # If the randomly chosen direction is 1 (down),
        # set the vertical speed of the ball to move upwards
        else:
            self.ball_speed_y = -3
```

Make these changes to the init method.

```python
"""
Name: pong_4.py
Author:
Date:
Purpose: Bouncing Ball
"""

# pip install pygame-ce
import pygame

# Import sys.exit to cleanly exit program
from sys import exit
from random import randint
from config import BALL_COLOR, BG_COLOR, WIDTH, HEIGHT, BALL_RADIUS


class Pong:

    def __init__(self):
        # Initialize pygame library
        pygame.init()

        # Set screen width and height as a tuple
        self.surface = pygame.display.set_mode((WIDTH, HEIGHT))

        # Set window caption
        pygame.display.set_caption("Pong")

        # Setup a computer clock object to keep the
        # game running at a constant speed regardless of computer speed
        self.clock = pygame.time.Clock()

        # Only allow these events to be captured
        # This helps optimize the game for slower computers
        pygame.event.set_allowed([pygame.QUIT, pygame.KEYDOWN])

        self.init_ball()
        self.set_ball_direction()
```

Example run:



The ball bounces all over the place, but stays in the playing field.

Paddles are next.

Assignment Submission

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