PyGame Pong Tutorial - Part 4

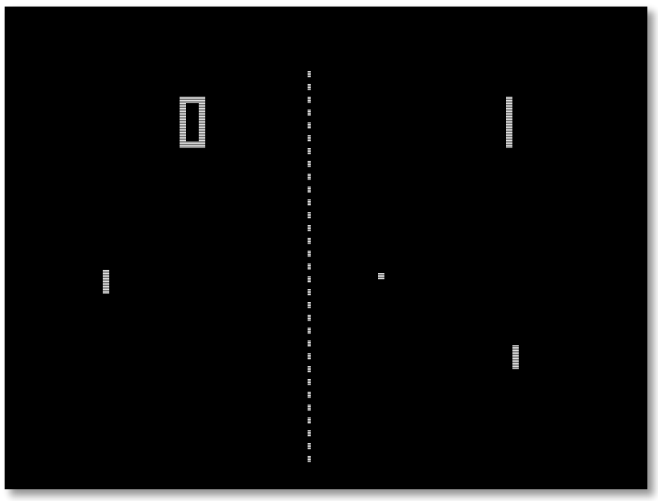
Contents

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.

```
88  # ---------------------- CHECK COLLISION ----------------------------------#
89      def check_collision(self):
90          """Check for collisions with all 4 walls"""
91          # Check for collision with left or right wall
92          if self.ball.left < 0 or self.ball.right >= config.WIDTH:
93
94              # Reverse y direction multiply by -1
95              self.ball_speed_x = self.ball_speed_x * -1
96
97          # Check for collision with top or bottom wall
98          if self.ball.top < 0 or self.ball.bottom >= config.HEIGHT:
99
100             # Reverse y direction multiply by -1
101             self.ball_speed_y = self.ball_speed_y * -1
```

Draw Net

Let's draw a net.

```
76  # ------------------------ DRAW NET ------------------------------------#
77      def draw_net(self):
78          # Define the width of the net lines
79          net_width = 2
80
81          # Loop through the height of the game screen
82          # with a step of 20 pixels
83          for i in range(0, config.HEIGHT, 20):
84
85              # Draw a rectangle representing a part of the net
86              pygame.draw.rect(
87                  self.surface,   # Surface to draw on
88                  config.WHITE,   # Color of the rectangle (white)
89                  (               # Rectangle coordinates and size
90                      # X-coordinate of the left corner of the rectangle
91                      config.WIDTH // 2 - net_width // 2,
92                      i,  # Y-coordinate of top corner of rectangle
93                      net_width,  # Width of the rectangle
94                      10          # Height of the rectangle
95                  )
96              )
```

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

```
94  # ---------------------------- GAME LOOP ---------------------------------#
95      def game_loop(self):
96          """Infinite Game Loop"""
97          while True:
98              self.check_events()
99              self.check_collision()
100
101             # ---------------- DRAW ON BACKBUFFER -------------------------#
102             # Draw everything on the backbuffer first
103             # Fill the display surface with black
104             self.surface.fill(config.BLACK)
105
106             self.draw_net()
```

Set Random Ball Direction

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

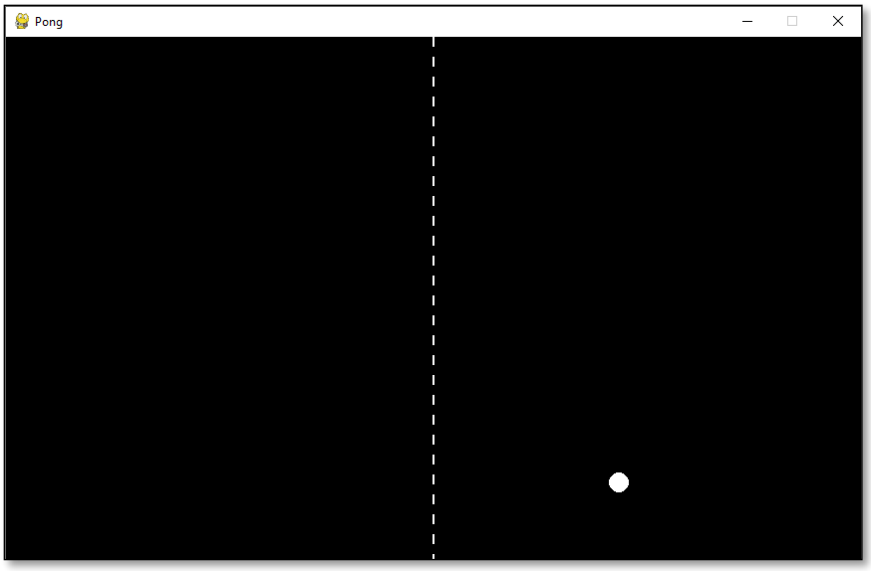
Add the following method.

```
# ------------------------ SET BALL DIRECTION -----------------------------#
    def set_ball_direction(self):
        """Set 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 library
import pygame
# Import sys.exit to cleanly exit program
from sys import exit
from random import randint
import config


class Pong:

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

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

        # Create the ball Rectangle object
        self.ball = pygame.Rect(
            config.WIDTH // 2 - config.BALL_RADIUS,     # Set x-coordinate
            config.HEIGHT // 2 - config.BALL_RADIUS,    # Set y-coordinate
            config.BALL_RADIUS,     # Set width of ball
            config.BALL_RADIUS      # Set height of 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

Zip up the program files folder and submit in Blackboard.