PyGame Pong Tutorial - Part 3

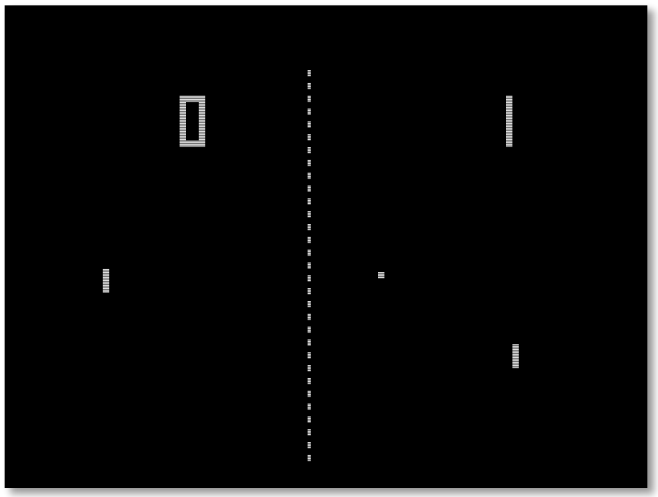
Contents

Time required: 30 minutes

Preview of the Game

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

Pong Demo Video



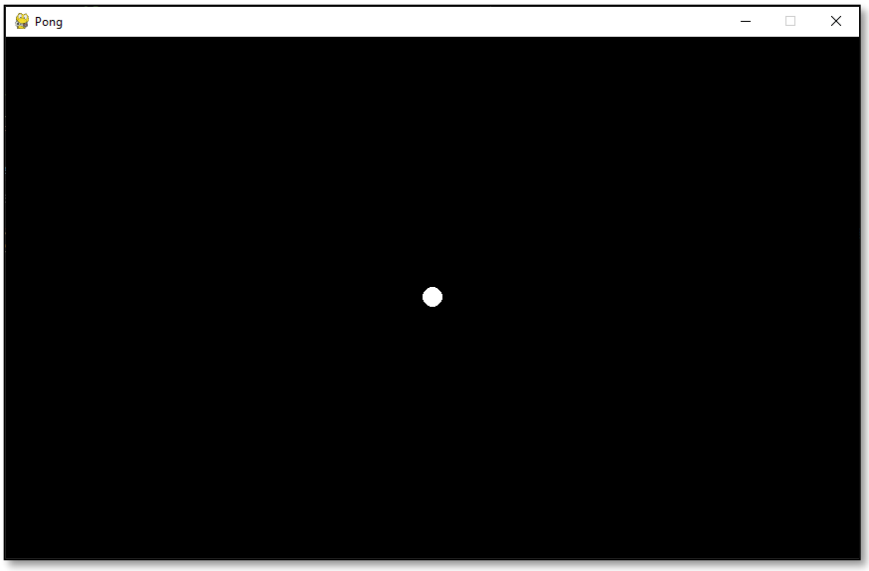
Yes, we are finally going to make something move. By changing the (x, y) values each time through the game loop, we animate our ball.

1. Save **pong_2.py** as **pong_3.py**
2. Add the following code.

```python
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
        )

        # Movement of ping pong ball in pixels
        self.ball_speed_x = 3
        self.ball_speed_y = 3
```

```python
# ---------------------------- GAME LOOP ---------------------------------#
    def game_loop(self):
        """Infinite Game Loop"""
        while True:
            self.check_events()

            # ---------------- DRAW ON BACKBUFFER -------------------------#
            # Draw everything on the backbuffer first
            # Fill the display surface with black
            self.surface.fill(config.BLACK)

            # Move the ball position every frame
            self.ball.x += self.ball_speed_x
            self.ball.y += self.ball_speed_y

            # Draw ball
            pygame.draw.ellipse(
                self.surface,   # Surface to draw on
                config.WHITE,   # Color to draw with
                self.ball       # Rect image object to draw
            )

            # -------------- UPDATE SURFACE -------------------------------#
            # From backbuffer, update Pygame display to reflect any changes
            pygame.display.update()

            # Cap game speed at 60 frames per second
            self.clock.tick(60)
```

Example run:



The ball moves . . . right off the screen.

Collisions are next.

Assignment Submission

Zip up the program files folder and submit in Blackboard.