PyGame Pong Tutorial - Part 3

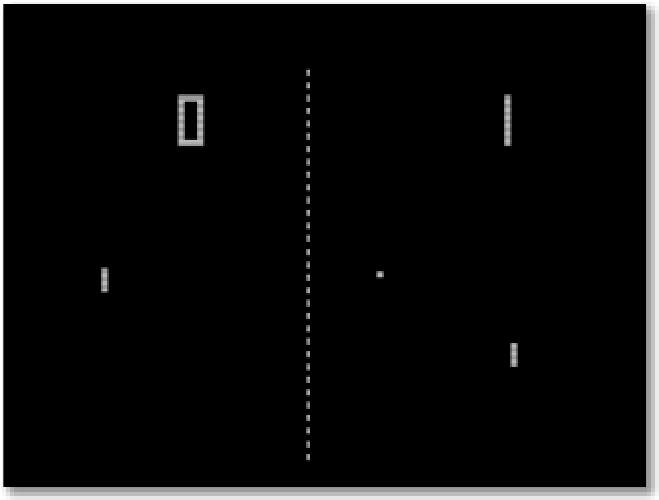
Contents

Time required: 30 minutes

Preview of the Game

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

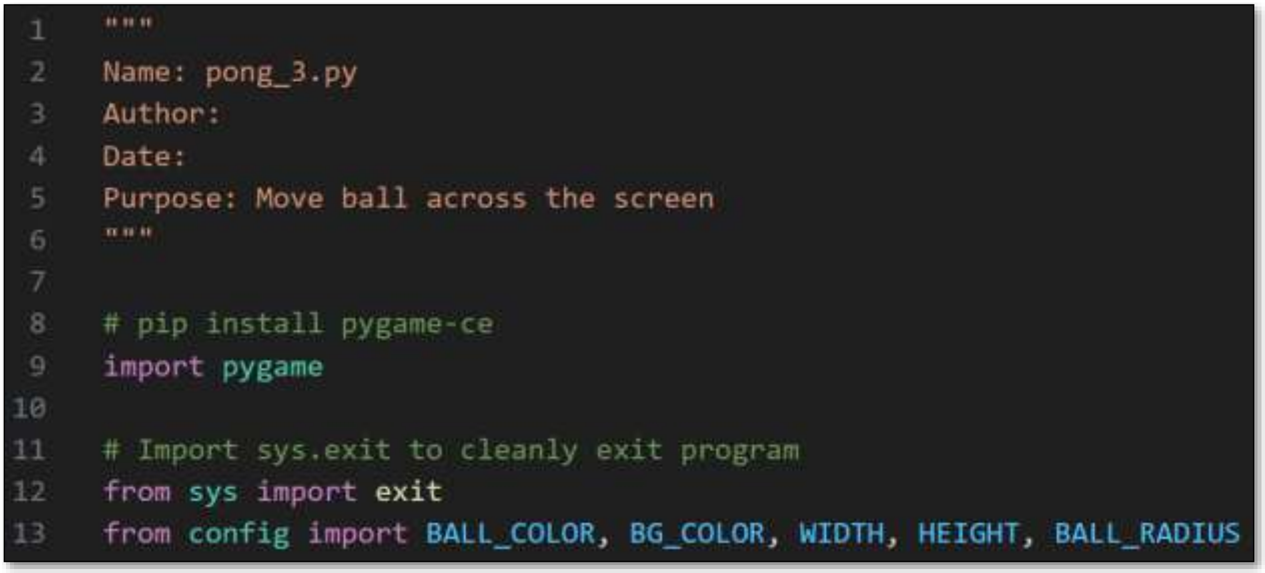
Pong Demo Video



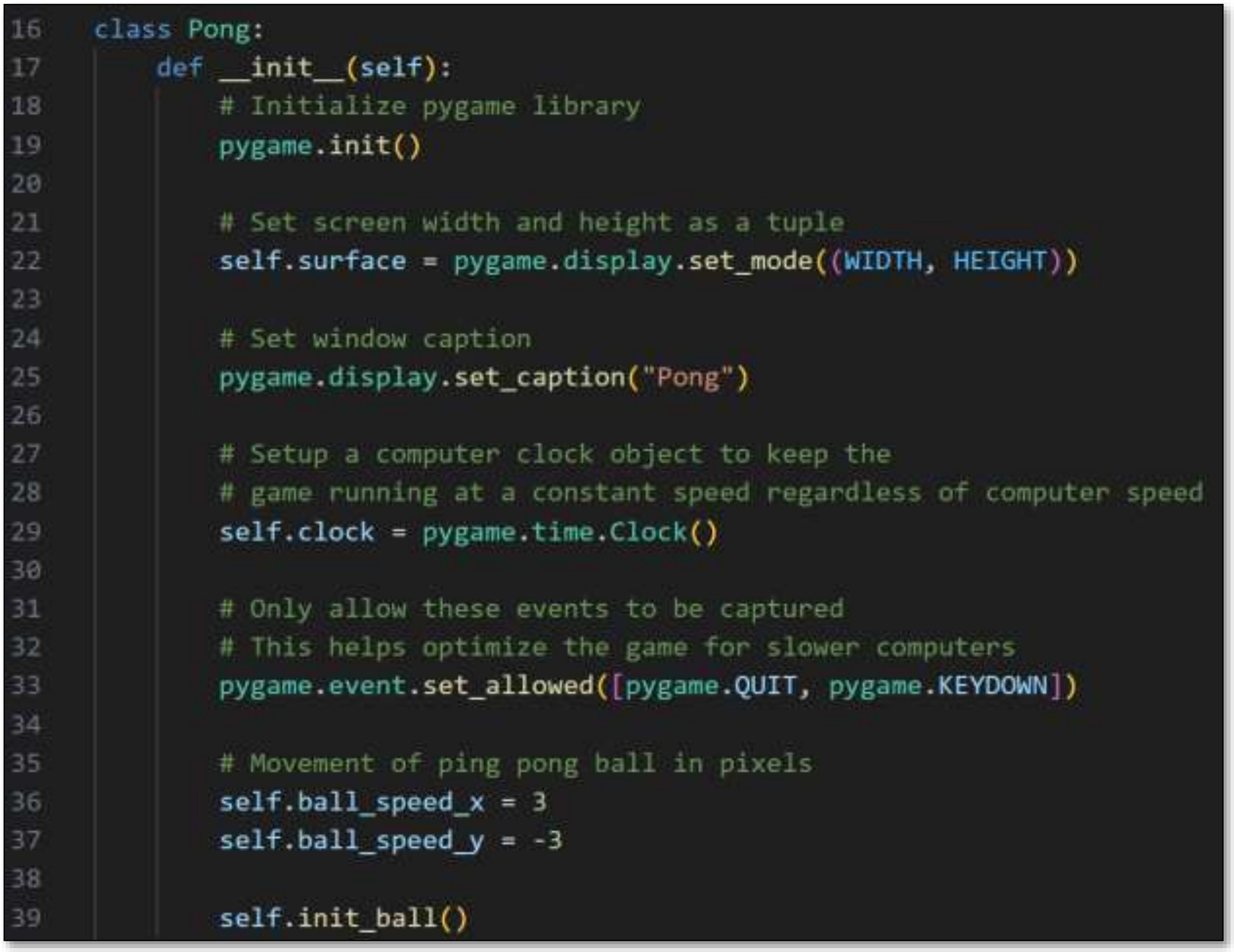
Yes, we are finally going to make something move. By changing the (x, y) values each time through the game loop, we animate our ball.

1. Save **pong_2.py** as **pong_3.py**

2. Modify the following code.



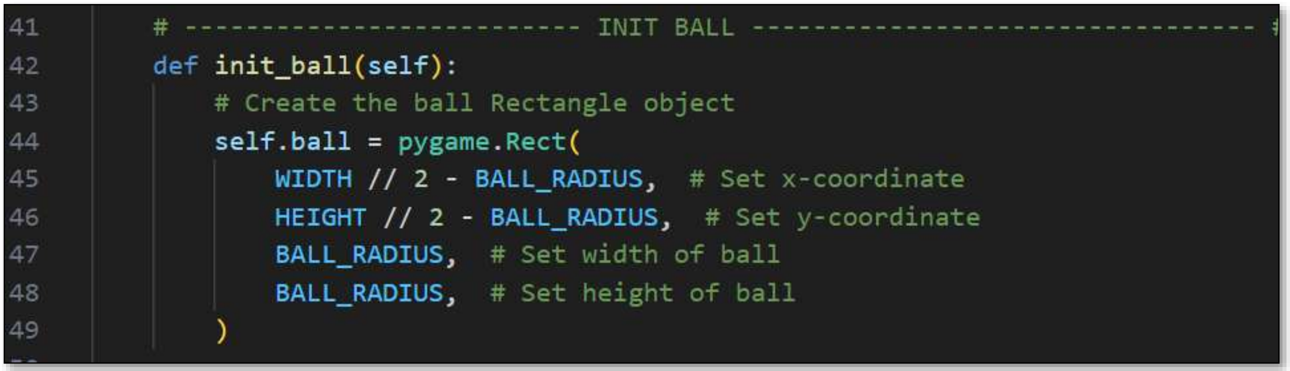
```python
"""
Name: pong_3.py
Author:
Date:
Purpose: Move ball across the screen
"""

# pip install pygame-ce
import pygame

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



```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((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])

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

        self.init_ball()
```

Init Ball

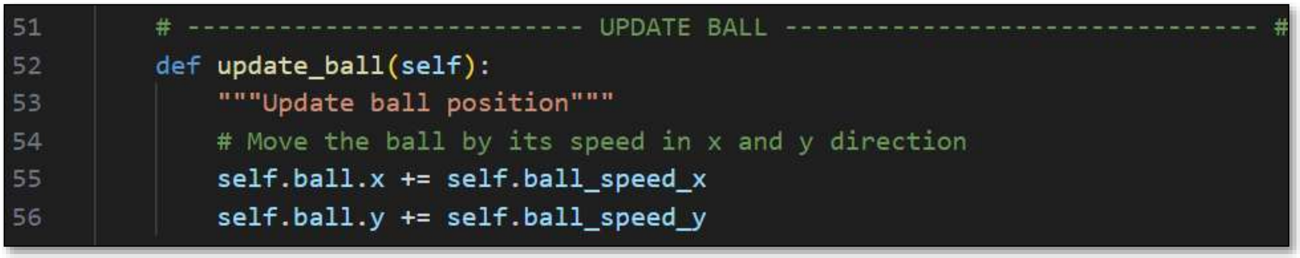
Let’s create a method to initialize the ball to make our code more modular.



```python
    # -------------------------- INIT BALL --------------------------------- #
    def init_ball(self):
        # Create the ball Rectangle object
        self.ball = pygame.Rect(
            WIDTH // 2 - BALL_RADIUS,  # Set x-coordinate
            HEIGHT // 2 - BALL_RADIUS,  # Set y-coordinate
            BALL_RADIUS,  # Set width of ball
            BALL_RADIUS,  # Set height of ball
        )
```

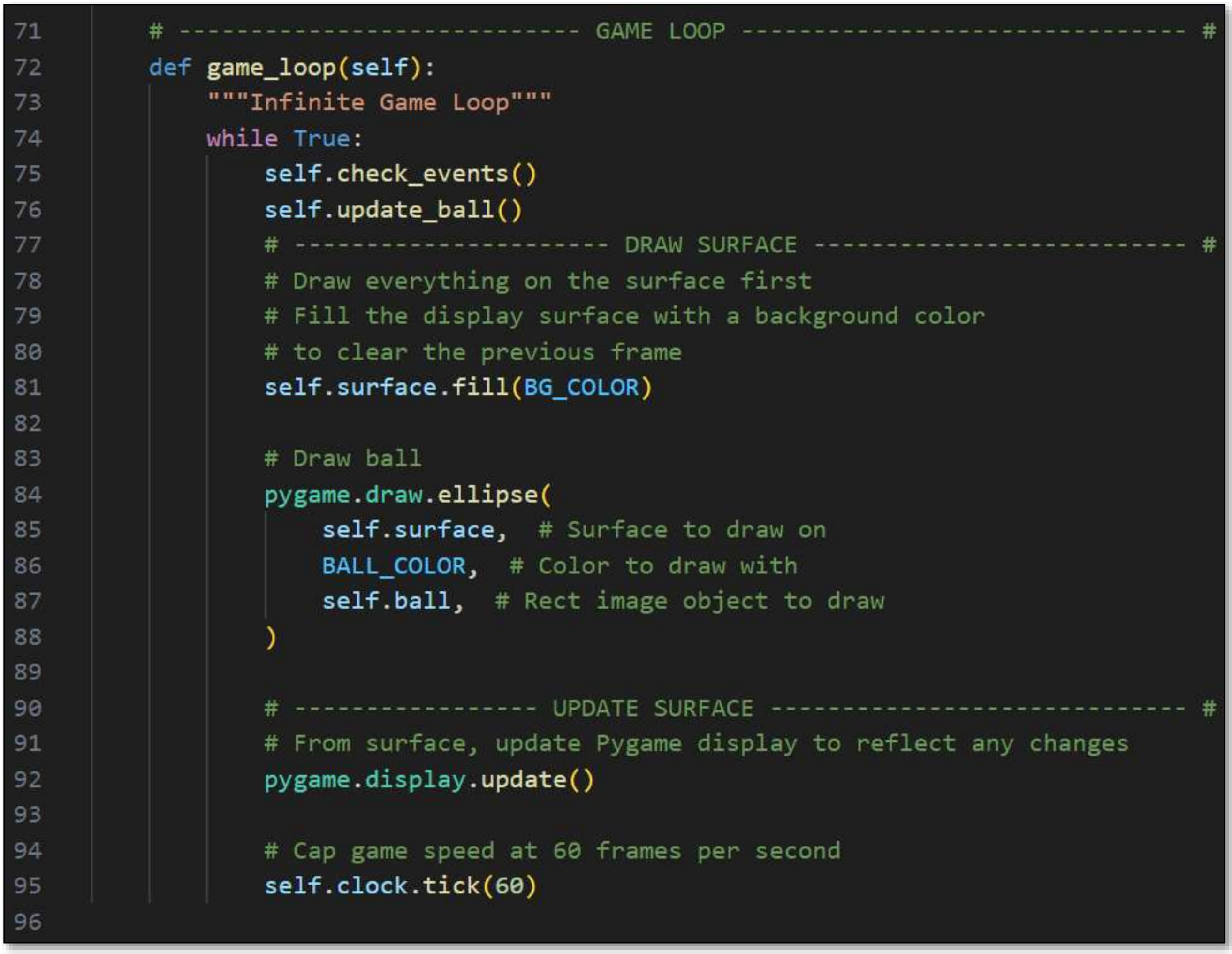
Update Ball

Let’s move our update ball code to a method.



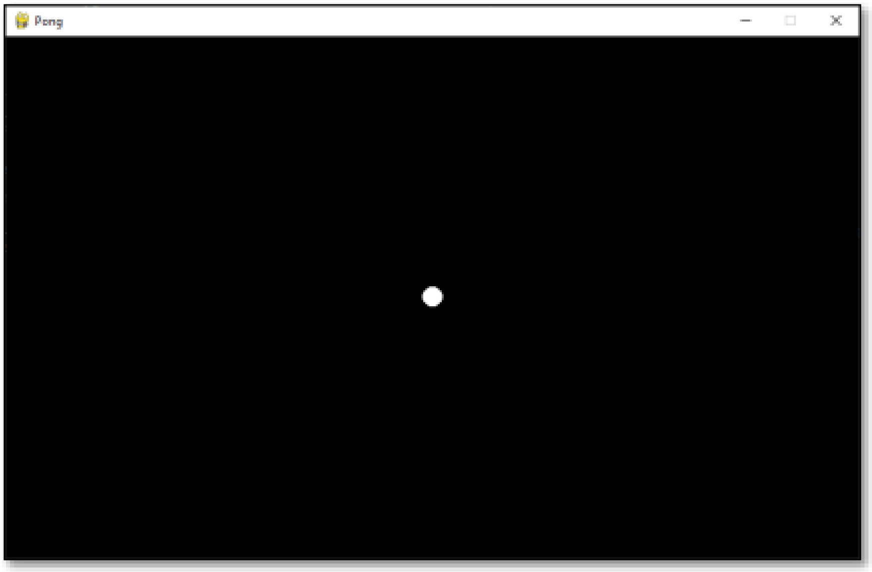
```python
    # -------------------------- UPDATE BALL ------------------------------- #
    def update_ball(self):
        """Update ball position"""
        # Move the ball by its speed in x and y direction
        self.ball.x += self.ball_speed_x
        self.ball.y += self.ball_speed_y
```

Game Loop



```python
    # --------------------------- GAME LOOP ------------------------------ #
    def game_loop(self):
        """Infinite Game Loop"""
        while True:
            self.check_events()
            self.update_ball()
            # --------------------- DRAW SURFACE ------------------------- #
            # Draw everything on the surface first
            # Fill the display surface with a background color
            # to clear the previous frame
            self.surface.fill(BG_COLOR)

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

            # ---------------- UPDATE SURFACE ---------------------------- #
            # From surface, 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.

Creative

- Change the colors to different RGB colors.

Assignment Submission

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