PyGame Pong Tutorial - Part 5

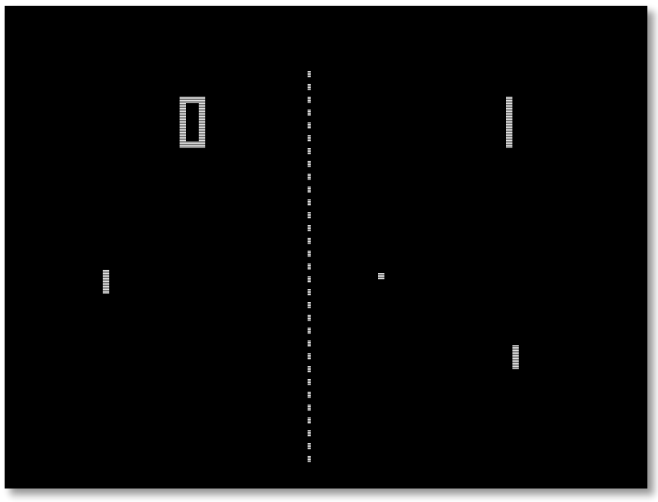
Contents

Time required: 30 minutes

Preview of the Game

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

Pong Demo Video



Paddles

It is time to get our paddle on. We will have a player paddle and a computer paddle with AI. Not a very smart AI, but it does move up and down.

1. Create a new Python file named: **paddle.py**
2. Add the following code.

```python
"""
    Name: paddle.py
    Author:
    Date:
    Purpose: Define a paddle's methods and attributes
"""
import config
import pygame


class Paddle:
    # Constructor method to initialize the paddle's attributes
    def __init__(self, x, y):
        # Initialize the x-coordinate of the paddle
        self.x = x
        # Initialize the y-coordinate of the paddle
        self.y = y
        # Set the width of the paddle
        self.width = 10
        # Set the height of the paddle
        self.height = 100

        # Create a rectangle object for paddles
        self.rect = pygame.Rect(
            self.x,  # x-coordinate of the top-left corner of the rectangle
            self.y,  # y-coordinate of the top-left corner of the rectangle
            self.width,     # width of the rectangle
            self.height     # height of the rectangle
        )

        # Set the speed at which the paddle moves
        self.speed = 5
```

```python
# ---------------------- MOVE PADDLE UP ----------------------------------#
    def move_up(self):
        """Move the paddle up"""
        # Check if the y-coordinate of the paddle is greater than 0
        if self.rect.y > 0:
            # Decrease the y-coordinate of the paddle by the speed value
            # which moves the paddle upwards
            self.rect.y = self.rect.y - self.speed
```

This method will be tied to the up cursor key to move the player paddle up.

```
43  # ---------------------- MOVE PADDLE DOWN --------------------------------#
44      def move_down(self):
45          """Move the paddle down"""
46          # Check if the y-coordinate of the paddle is less than
47          # the screen height minus the paddle's height
48          if self.rect.y < config.HEIGHT - self.height:
49              # Increase the y-coordinate of the paddle by the speed value
50              # which moves the paddle downwards
51              self.rect.y = self.rect.y + self.speed
```

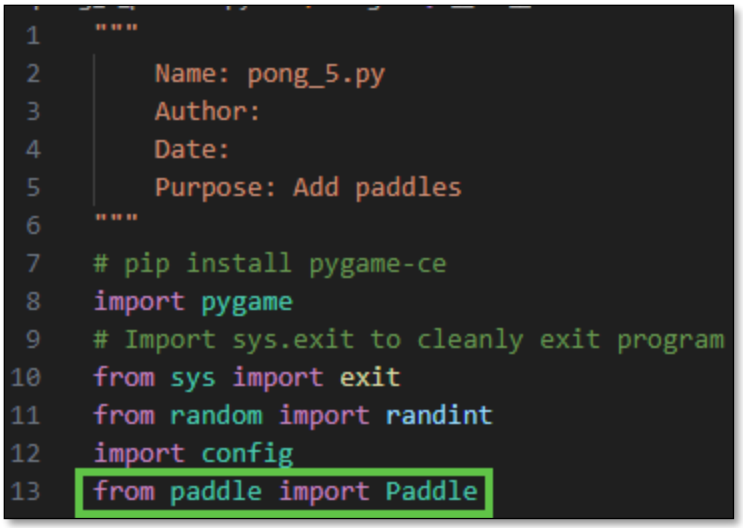
This method moves the player paddle down the screen.

```
53  # ---------------------- MOVE COMPUTER PADDLE ----------------------------#
54      def move_computer_paddle(self):
55          """Move computer paddle up and down"""
56          # If the computer paddle is inside the top and bottom border
57          # keep moving in the same direction
58          if self.rect.top + self.speed > 20 and \
59                  self.rect.bottom + self.speed < config.HEIGHT - 20:
60
61              # Move computer paddle in the current direction
62              self.rect.y += self.speed
63
64          else:
65              # Reverse paddle direction multiply by -1
66              self.speed = self.speed * -1
```

The computer paddle AI moves the paddle up and down the screen.

PONG 5

1. Save **pong_4.py** as **pong_5.py**
2. Import the Paddle class.



3. Create a player and a computer paddle object.

```python
class Pong:

    def __init__(self):
        """Initialize the Pong class"""
        # Initialize pygame
        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()

        # Set up player paddles
        self.player = Paddle(
            5,                          # x coordinate
            (config.HEIGHT - 100) // 2  # y coordinate
        )

        self.computer = Paddle(
            config.WIDTH - 15,          # x coordinate
            (config.HEIGHT - 100) // 2  # y coordinate
        )
        self.computer_speed = 3
```

4. Get the up and down arrow key pressed by the player.

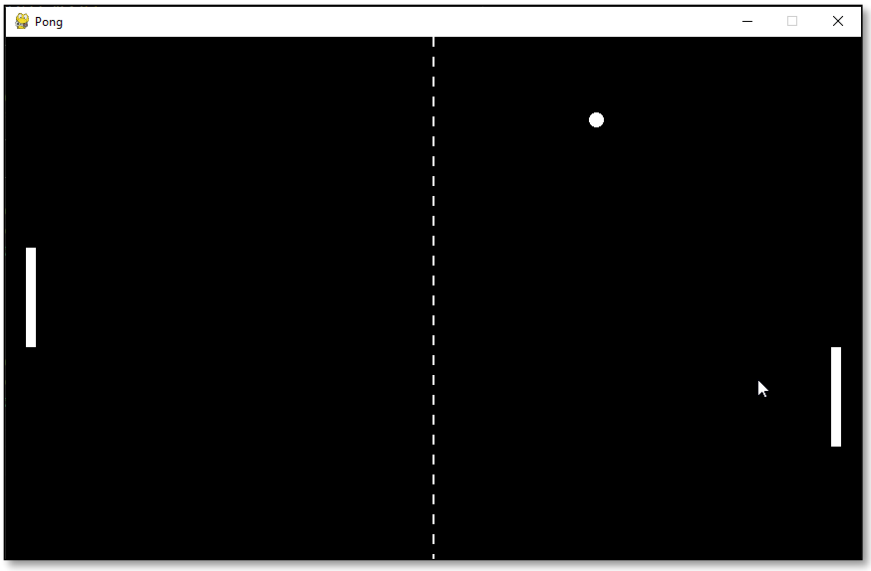
```
113  # --------------------------- GET KEYS ---------------------------------#
114      def get_keys(self):
115          # Update player paddle position
116          # Get the state of all keyboard keys pressed at the moment.
117          keys = pygame.key.get_pressed()
118
119          # Check if the UP arrow key is pressed.
120          if keys[pygame.K_UP]:
121              # If the UP arrow key is pressed, move the player up
122              self.player.move_up()
123
124          # Check if the DOWN arrow key is pressed
125          if keys[pygame.K_DOWN]:
126              # If the DOWN arrow key is pressed, move the player down
127              self.player.move_down()
128
129          # The Esc key will quit the game
130          if keys[pygame.K_ESCAPE]:
131              # Quit Pygame
132              pygame.quit()
133              # Exit Python
134              exit()
```

5. Modify the Game Loop.

```
151  # --------------------------- GAME LOOP ---------------------------------#
152      def game_loop(self):
153          """Infinite Game Loop"""
154          while True:
155              self.check_events()
156
157              self.computer.move_computer_paddle()
158
159              self.get_keys()
160
161              self.check_collision()
162
163              # ---------------- DRAW ON BACKBUFFER -------------------------#
164              # Draw everything on the backbuffer first
165              # Fill the display surface with black
166              self.surface.fill(config.BLACK)
167
168              self.draw_net()
```

```
170             # Draw a rectangle for the player's paddle
171             # on the screen using Pygame's draw function
172             pygame.draw.rect(
173                 self.surface,   # Surface to draw on
174                 config.WHITE,   # Color to draw with
175                 self.player     # rect image object to draw
176             )
177
178             # Draw a rectangle for the computer's paddle
179             # on the screen using Pygame's draw function
180             pygame.draw.rect(
181                 self.surface,   # Surface to draw on
182                 config.WHITE,   # Color to draw with
183                 self.computer   # rect image object to draw
184             )
185
186             # Move the ball position every frame
187             self.ball.x += self.ball_speed_x
188             self.ball.y += self.ball_speed_y
189
190             # Draw ball
191             pygame.draw.ellipse(
192                 self.surface,   # Surface to draw on
193                 config.WHITE,   # Color to draw with
194                 self.ball       # Rect image object to draw
195             )
196
197             # Redraw the display surface object
198             pygame.display.update()
199
200             # Set the frame rate
201             self.clock.tick(60)
```

Example run:



The ball moves . . . the paddles move . . .

Scoring is next.

Assignment Submission

Zip up the program files folder and submit in Blackboard.