PyGame Flappy Bird Tutorial - Part 5

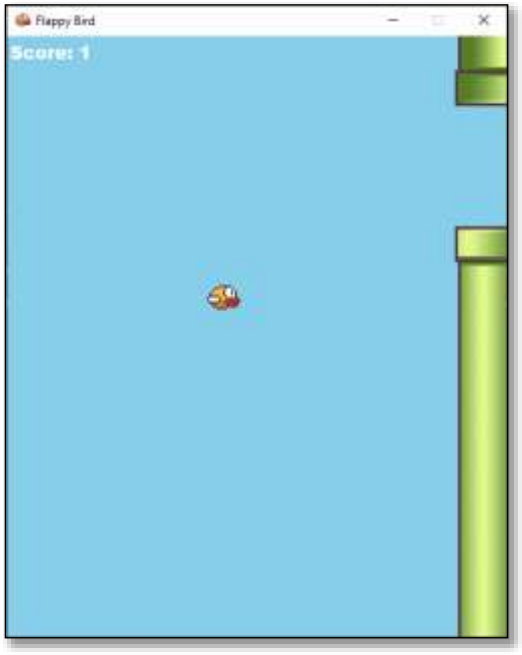
Contents

Time required: 30 minutes

Preview of the Game

Here's a sneak peak of the game that we are going to work on.

Flappy Bird Demo Video



Update Background

Save **flappy_bird_4.py** as **flappy_bird_5.py**

Time to get our background moving. This will help with the illusion of movement.

Add self.background_speed to the class.

```
35
36          # Set the gravity to 3
37          # This is how fast the bird falls
38          # The higher the number, the faster the bird falls
39          # The lower the number, the slower the bird falls
40          self.gravity = 3
41          self.pipes_speed = 4  # Pipes move faster than background
42          self.background_speed = 2  # Background moves slower than pipes
43
44          self.load_background()
45          self.init_bird()
46          self.init_pipes()
```

Modify the **load_background()** method.

```
48      # ---------------------- LOAD BACKGROUND ------------------------------ #
49      def load_background(self):
50          """Load background image"""
51          # Load image from file into a variable
52          background = pygame.image.load("./assets/background.png")
53          # Convert the image to a PyGame surface
54          # This is done to speed up the game
55          self.background = background.convert_alpha()
56
57          # Get rectangle around background for easier game manipulation
58          self.background_rect = self.background.get_rect()
59          # Set initial position of background image
60          self.background_rect.left = 0
61          self.background_rect.top = 0
```

Add the **update_background()** method.

```
59      # ---------------------- UPDATE BACKGROUND ----------------------------- #
60      def update_background(self):
61          """Move background image seamlessly"""
62          # Move the background image to the left by self.background_speed
63          self.background_rect.left -= self.background_speed
64
65          # If the first background image is off the left side of the screen
66          if self.background_rect.right <= 0:
67              # Reset it to the right side of the screen
68              self.background_rect.left = 0
```

Reset the Pipes

We only get one set of pipes. Let's reset the pipes to have more pipes traveling across.

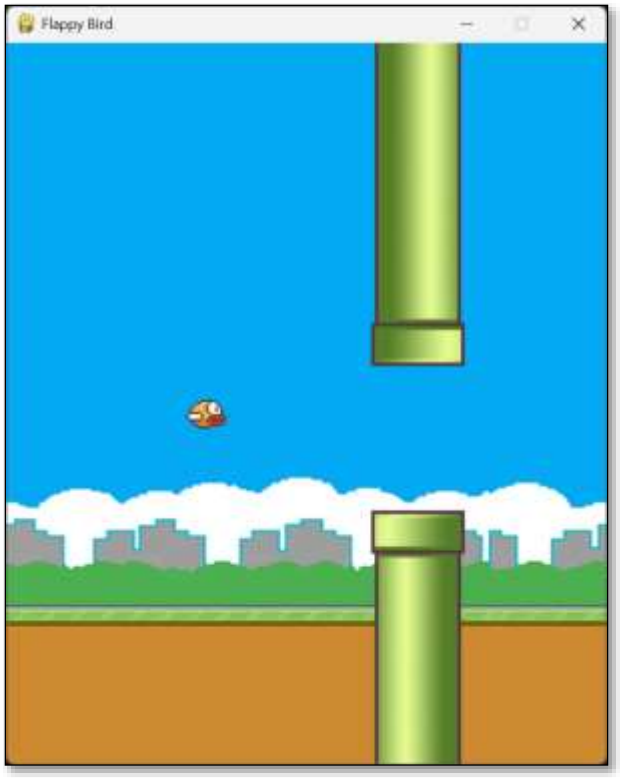
Let's add a **reset_pipes()** method.

```
87      # -------------------- RESET PIPES ----------------------------------#
88      def reset_pipes(self):
89          """Reset pipes every time they leave the screen"""
90          # Pick a random height for the bottom of the top pipe
91          self.pipe_upper_rect.bottom = randint(
92              50,                 # Set minimum random number to 50
93              config.HEIGHT // 2  # Set maximum to half the surface height
94          )
95
96          # Set lower pipe top to upper pipe bottom plus pipe gap
97          self.pipe_lower_rect.top = self.pipe_upper_rect.bottom \
98              + self.pipe_gap_size
99
100         # Set initial X off screen to right
101         self.pipe_upper_rect.left = config.WIDTH
102         self.pipe_lower_rect.left = config.WIDTH
```

Update Game Loop

```
182     # ---------------------------- GAME LOOP ------------------------------ #
183     def game_loop(self):
184         """Infinite game loop"""
185         while True:
186             self.check_events()
187             self.update_bird()
188             self.update_pipes()
189             self.update_background()
190 
191             # If the pipes are off the screen, reset them
192             if self.pipe_upper_rect.right < 0:
193                 self.reset_pipes()
194             # ------------------- DRAW SURFACE ---------------------------- #
195             # Filling the surface with the background image
196             # clears the previous frame
197             # Draw the background
198             self.surface.blit(self.background, self.background_rect)
199             self.surface.blit(self.background, (self.background_rect.right, 0))
200 
201             # Draw bird to the surface
202             self.surface.blit(self.bird, self.bird_rect)
203 
204             # Draw pipes to the surface
205             self.surface.blit(
206                 self.pipe_lower,  # Source image
207                 self.pipe_lower_rect,  # Destination location of image
208             )
209             self.surface.blit(
210                 self.pipe_upper,  # Source image
211                 self.pipe_upper_rect,  # Destination location of image
212             )
213 
214             # ------------------ UPDATE DISPLAY --------------------------- #
215             # From surface, update Pygame display to reflect any changes
216             pygame.display.update()
217             # Cap game speed at 60 frames per second
218             self.clock.tick(60)
```

Example run:



You can fly your bird up and down and through the pipes. The pipes keep coming!

There are a few issues. The bird can fall off the screen or fly up to the sun. There aren't any collisions or score keeping.

Coming right up!

Assignment Submission

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