PyGame Flappy Bird Tutorial - Part 4

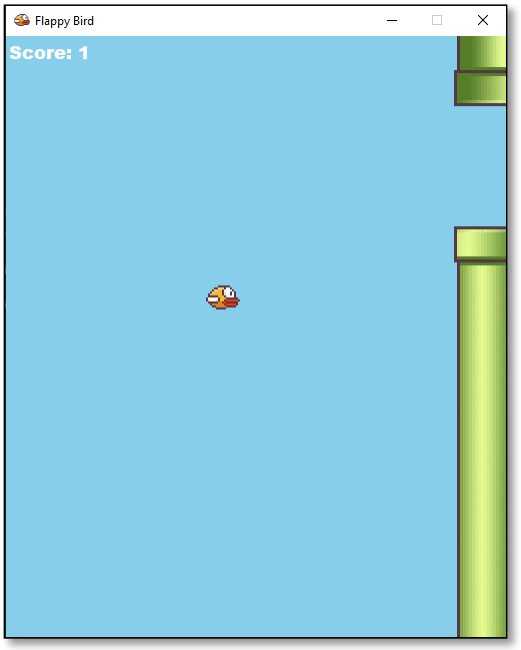
Contents

Time required: 30 minutes

Preview of the Game

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

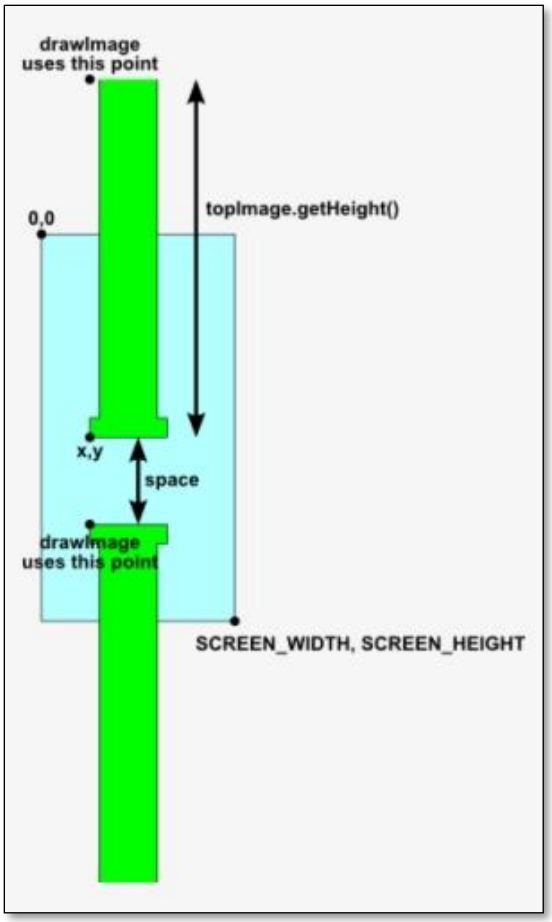
Flappy Bird Demo Video



Add the Pipes

This image gives an idea how we are going to manage the placement of the pipes. We will place the top pipe vertically by using a random integer. We will place the bottom pipe the pipe gap (space) distance away.

The pipes are the obstacles the flappy bird must fly between. We are going to load one image twice. We will use the right side up image for the bottom. We will flip the same image over to use it for the top pipe.



1. Save **flappy_bird_3.py** as **flappy_bird_4.py**
2. Modify the existing code.

```
"""
    Name: flappy_bird_4.py
    Author:
    Date:
    Purpose: Flappy Bird Clone in OOP
"""
# pip install pygame-ce
# Import pygame library
import pygame
# Import exit for a clean program shutdown
from sys import exit
from random import randint
import config
```

3. This imports the randint function. We can randomize where the pipes appear vertically.

```
class FlappyBird:
    def __init__(self):
        # Initialize pygame engine
        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("Flappy Bird")

        # Define the clock to keep the game running at a set speed
        self.clock = pygame.time.Clock()

        self.init_bird()
        self.init_pipes()
```

4. This new line calls the init pipes method. This get our pipes setup and ready to go.

```
# ------------------------- INIT PIPES ----------------------------------#
    def init_pipes(self):
        """Load pipe images, get rect, set initial positions"""
        # Set the gap between pipes
        self.pipe_gap_size = self.bird_rect.height * 5

        # How many pixels at a time the pipes move
        self.pipe_move = 4

        # Load pipe images
        self.pipe_lower = pygame.image.load(
            "./assets/pipe.png").convert_alpha()

        # Rotate upper image 180 degrees
        self.pipe_upper = pygame.transform.rotate(
            pygame.image.load("./assets/pipe.png").convert_alpha(), 180
        )

        # Get rectangles around images for easier manipulation
        self.pipe_lower_rect = self.pipe_lower.get_rect()
        self.pipe_upper_rect = self.pipe_upper.get_rect()

        # Set initial pipe location off screen to right
        self.pipe_upper_rect.left = config.WIDTH
        self.pipe_lower_rect.left = config.WIDTH

        # Initial placement of pipes vertically
        self.pipe_upper_rect.bottom = randint(
            50,                 # Stay 50 away from top
            config.HEIGHT // 2  # Upper range of random numbers
        )

        # Set lower pipe vertical location
        self.pipe_lower_rect.top = self.pipe_upper_rect.bottom \
            + self.pipe_gap_size
```

NOTE: There are 3 pipes images in the assets folder. You can pick any of them.

5. Yep, there is a lot going on here. Read the comments carefully. Notice that we use the same image, but rotate it 180 degrees to use it for the bottom pipe.

6. The \ symbol is not really part of the code. It allows us to have a single line of code on two lines for better readability.

Modify the Game Loop

1. Draw the upper and lower pipes.

2. Move the upper and lower pipes from the right to the left.

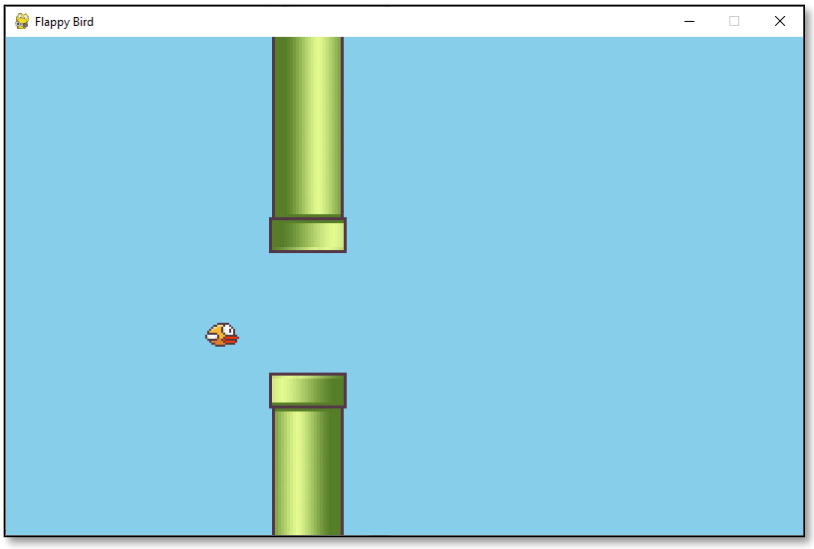
```
100   # ---------------------- GAME LOOP ------------------------------------#
101       def game_loop(self):
102           """Infinite game loop"""
103           while True:
104               self.check_events()
105               # Simulate gravity by moving the bird down
106               # unless the UP key is pressed
107               # Reset gravity to 3 each time through the loop
108               gravity = 3
109
110               # Get list of keys being pressed
111               key_input = pygame.key.get_pressed()
112
113               # If up cursor pressed, move up 5 pixels
114               if key_input[pygame.K_UP]:
115                   # Decrease gravity, the bird flies up
116                   gravity -= 5
117
118               # Move the bird by adding gravity value to y location
119               self.bird_rect.y = self.bird_rect.y + gravity
120
121               # Move pipe images from right to left
122               self.pipe_upper_rect.left = self.pipe_upper_rect.left - \
123                   self.pipe_move
124               self.pipe_lower_rect.left = self.pipe_lower_rect.left - \
125                   self.pipe_move
```

7. To make the pipes move from right to left, subtract the pipe move distance from the current location of the pipe.

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

            # Draw bird to the backbuffer
            self.surface.blit(
                self.bird,      # Source image
                self.bird_rect  # Destination location of image
            )

            # Draw pipes to the backbuffer
            self.surface.blit(
                self.pipe_lower,        # Source image
                self.pipe_lower_rect    # Destination location of image
            )
            self.surface.blit(
                self.pipe_upper,        # Source image
                self.pipe_upper_rect    # Destination location of image
            )

            # -------------- 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)


# Create flappy bird program object
flappy_bird = FlappyBird()
# Start infinite game loop
flappy_bird.game_loop()
```

8. Draw the pipes in the new location.

Example run:



You can fly your bird up and down and through the pipes.

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 all tutorials and assignments.
2. Attach screenshots showing the successful operation of each tutorial program.
3. Submit in Blackboard.