PyGame Flappy Bird Tutorial - Part 4

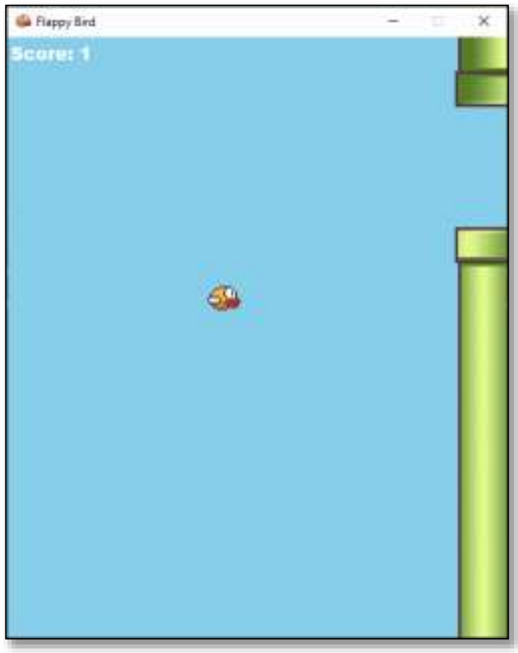
Contents

Time required: 30 minutes

Preview of the Game

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

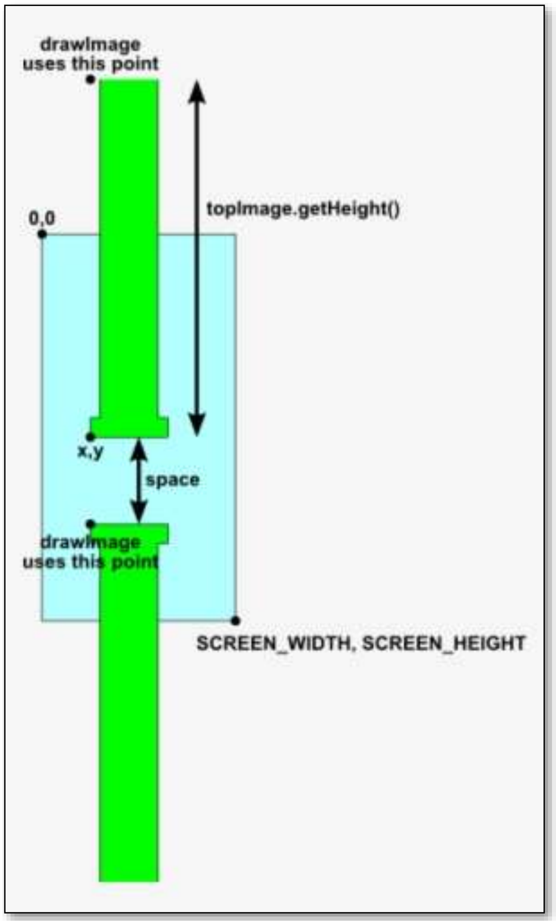
[Flappy Bird Demo Video]



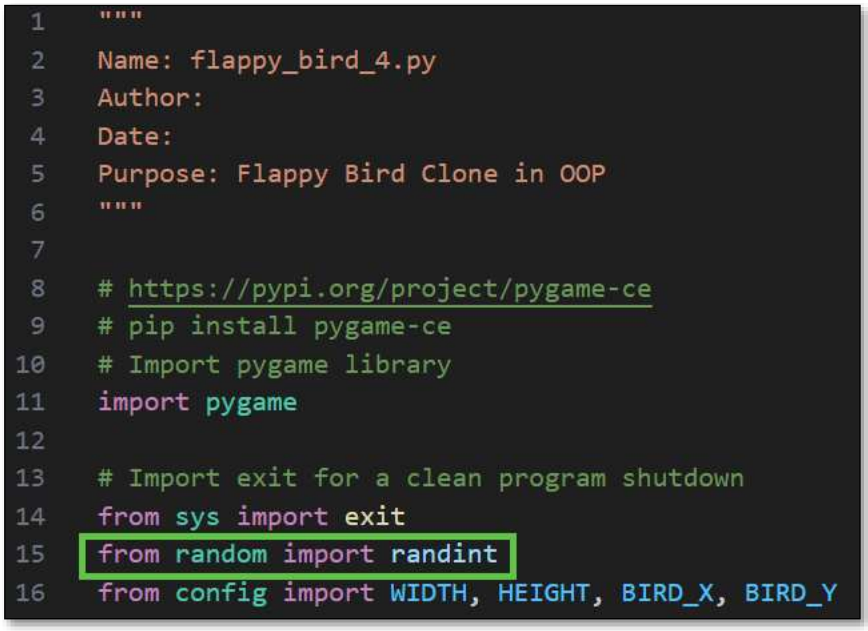
Add the Pipes

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.

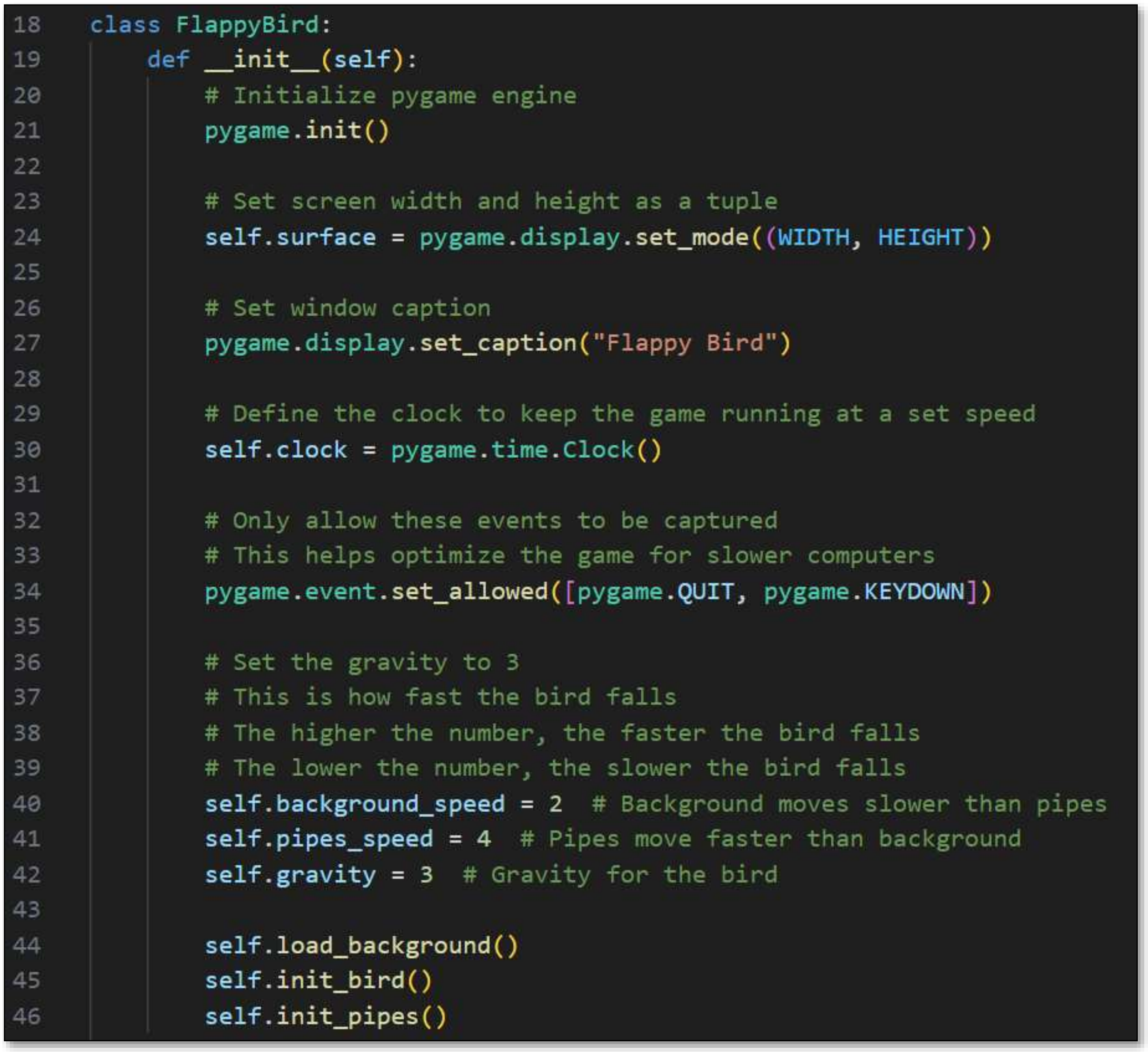

drawImage uses this point
topImage.getHeight()
0,0
x,y
space
drawImage uses this point
SCREEN_WIDTH, SCREEN_HEIGHT

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



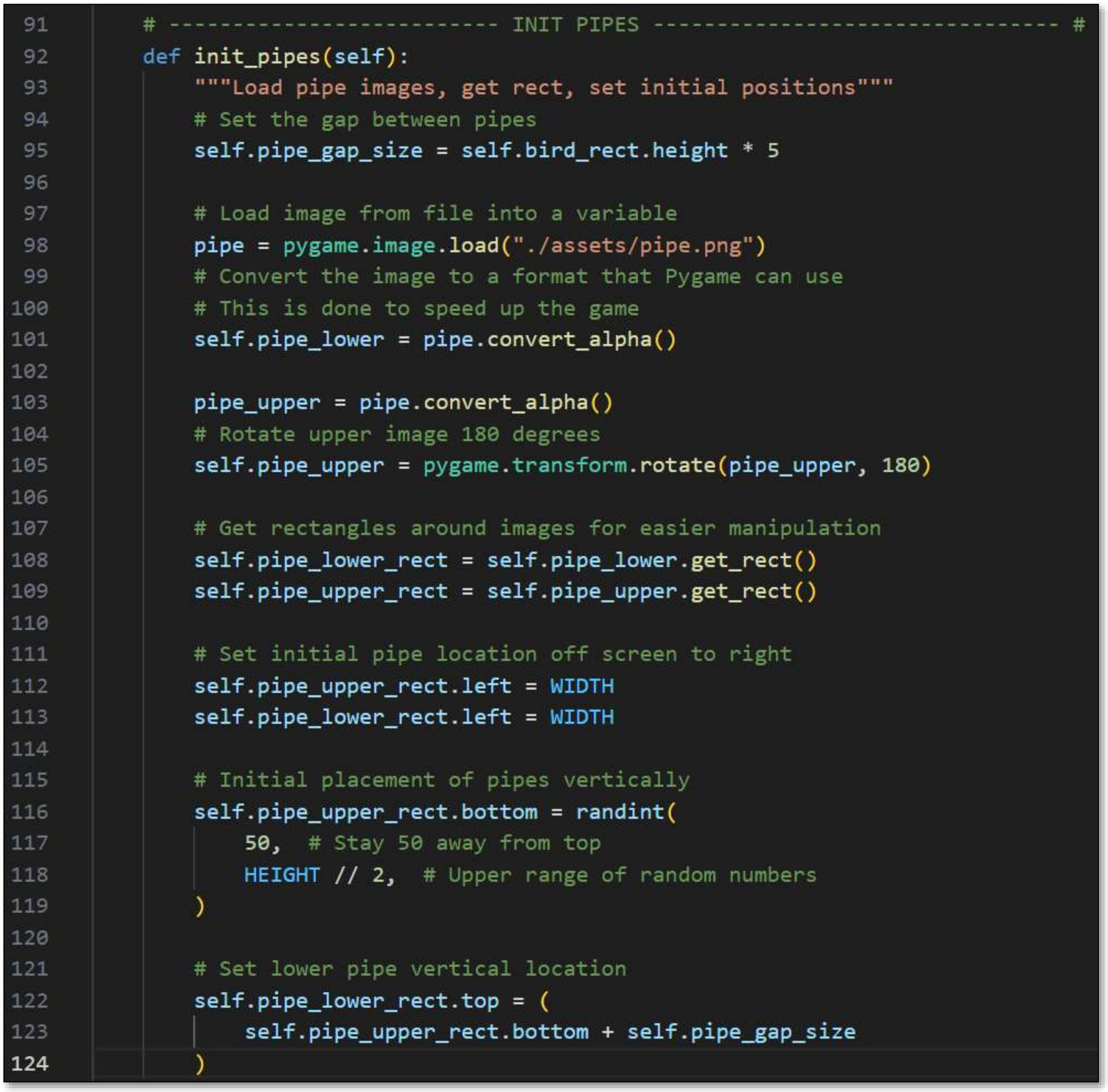
```python
"""
Name: flappy_bird_4.py
Author:
Date:
Purpose: Flappy Bird Clone in OOP
"""

# https://pypi.org/project/pygame-ce
# pip install pygame-ce
# Import pygame library
import pygame

# Import exit for a clean program shutdown
from sys import exit
from random import randint
from config import WIDTH, HEIGHT, BIRD_X, BIRD_Y
```

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



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

        # Define the clock to keep the game running at a set 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])

        # Set the gravity to 3
        # This is how fast the bird falls
        # The higher the number, the faster the bird falls
        # The lower the number, the slower the bird falls
        self.background_speed = 2  # Background moves slower than pipes
        self.pipes_speed = 4  # Pipes move faster than background
        self.gravity = 3  # Gravity for the bird

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

4. This new line calls the init pipes method. This will get our pipes set up 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

        # Load image from file into a variable
        pipe = pygame.image.load("./assets/pipe.png")
        # Convert the image to a format that Pygame can use
        # This is done to speed up the game
        self.pipe_lower = pipe.convert_alpha()

        pipe_upper = pipe.convert_alpha()
        # Rotate upper image 180 degrees
        self.pipe_upper = pygame.transform.rotate(pipe_upper, 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 = WIDTH
        self.pipe_lower_rect.left = WIDTH

        # Initial placement of pipes vertically
        self.pipe_upper_rect.bottom = randint(
            50,  # Stay 50 away from top
            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 pipe 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.

Update Pipes

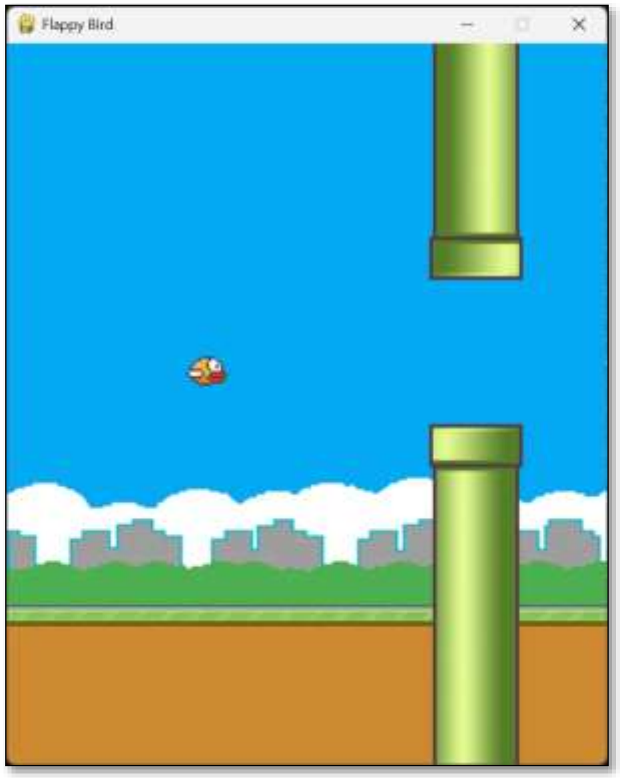
To make the pipes move from right to left, subtract the pipe move distance from the current location of the pipe.

```python
    # ---------------------- UPDATE PIPES --------------------------------- #
    def update_pipes(self):
        """Update pipe positions"""
        # Move pipe images from right to left
        self.pipe_upper_rect.left -= self.pipes_speed
        self.pipe_lower_rect.left -= self.pipes_speed
```

Modify the Game Loop

```python
    # ---------------------- GAME LOOP ------------------------------------ #
    def game_loop(self):
        """Infinite game loop"""
        while True:
            self.check_events()
            self.update_bird()
            self.update_pipes()
            # ------------------- DRAW SURFACE ------------------------------ #
            # Filling the surface with the background image
            # clears the previous frame
            self.surface.blit(self.background, (0, 0))

            # Draw bird to the surface
            self.surface.blit(self.bird, self.bird_rect)

            # Draw pipes to the surface
            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 DISPLAY ------------------------------ #
            # 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:



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