Task: Create a Flappy Bird Clone Using Pygame and OOP

Objective

Learn the basics of Pygame and Object-Oriented Programming (OOP) by developing a Flappy Bird clone. This task covers the basic Pygame structure, including game loop, input handling, and drawing sprites.

Description

Set up Pygame and understand its basic structure.

Implement classes for the game environment, player, and pipes.

Handle user input to control the game.

Draw and animate sprites on the screen (included in the REPO under the assets folder).

Optional Code Skeleton

Suggested classes:

```
1. class Environment:
2.     def __init__(self):
3.         # TODO: Load the background image
4.         pass
5.
6.     def draw(self, screen):
7.         # TODO: Draw the background image on the screen
8.         pass
9.
```

—

```
 1. class Player(pygame.sprite.Sprite):
 2.     def __init__(self):
 3.         super().__init__()
 4.         # TODO: Load the player (bird) image, initialize position and velocity
 5.         pass
 6.
 7.     def update(self):
 8.         # TODO: Update the player position based on input and gravity
 9.         pass
10.
11.     def draw(self, screen):
12.         # TODO: Draw the player on the screen
13.         pass
14.
```

—

```python
class Pipe(pygame.sprite.Sprite):
    def __init__(self, x, y, inverted):
        super().__init__()
        # TODO: Load the pipe image and set the initial position
        #       Flip the image if it is an inverted pipe
        pass

    def update(self):
        # TODO: Move the pipe to the left
        pass

    def draw(self, screen):
        # TODO: Draw the pipe on the screen
        pass

```

```python
import pygame
import sys

class Game:
    def __init__(self):
        # TODO: Initialize Pygame
        self.clock = None  # TODO: Initialize clock
        self.screen = None  # TODO: Initialize screen with dimensions and caption
        self.environment = None  # TODO: Initialize environment object
        self.player = None  # TODO: Initialize player object
        self.pipes = None  # TODO: Initialize group of pipes
        self.all_sprites = None  # TODO: Initialize group for all sprites

    def run(self):
        # Main game loop
        while True:
            self.handle_events()
            self.update()
            self.draw()
            # TODO: Set frame rate using clock

    def handle_events(self):
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            # TODO: Handle user input for controlling the player

    def update(self):
        # TODO: Update all sprites and pipes
        pass

    def draw(self):
        # TODO: Fill the screen with sky blue color
        # TODO: Draw the environment, all sprites, and pipes
        # TODO: Update the display
        pass

```