PyGame Flappy Bird Tutorial - Part 6

Contents
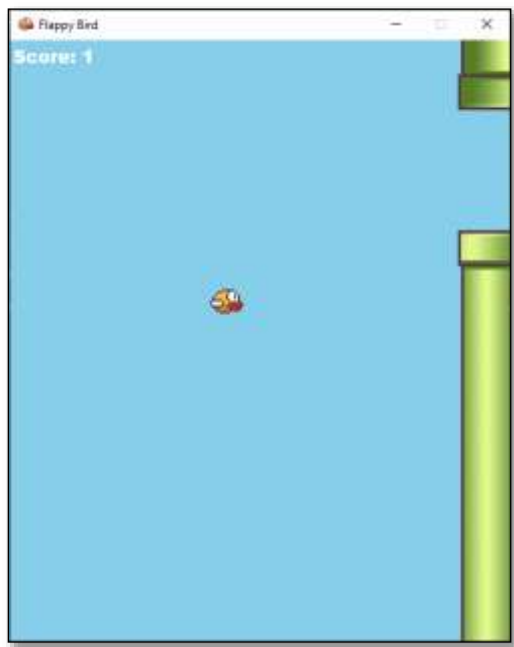
Time required: 30 minutes

Preview of the Game

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

[Flappy Bird Demo Video](#)

Collisions @%**(((&&*

Time to add collisions and a game over display. We are going to use another library called PyGame Menu.

1. Install PyGame Menu.

```
# Windows
pip install pygame-menu-ce
```

2. Save **flappy_bird_5.py** as **flappy_bird_6.py**

3. Modify the existing code.

```
1    """
2    Name: flappy_bird_6.py
3    Author:
4    Date:
5    Purpose: Add collisions and a game over screen
6    """
7
8    # https://pypi.org/project/pygame-ce
9    # pip install pygame-ce
10   import pygame
11
12   # https://pypi.org/project/pygame-menu-ce/
13   # pip install pygame-menu-ce
14   import pygame_menu as pm
15
16   # Import exit for a clean program shutdown
17   from sys import exit
18   from random import randint
19   from config import WIDTH, HEIGHT, BIRD_X, BIRD_Y
```

4. Let's use the PyGame Menu library to create a Game Over screen.

```
141        # ---------------------- DISPLAY GAME OVER ---------------------------- #
142    def display_game_over(self):
143        """Display game over menu using the Pygame Menu library"""
144        # Define a menu object for the game over screen
145        game_over = pm.Menu(
146            title="Game over",  # Set title menu to "Game over"
147            width=config.WIDTH,  # Set to width of game surface
148            height=config.HEIGHT,  # Set to height of game surface
149            # Set the theme of the menu to an orange color scheme
150            theme=pm.themes.THEME_BLUE,
151        )
152
153        # Add a button to the game over menu for exiting the game
154        game_over.add.button(title="Play Again?", action=main)
155
156        # Add a button to the game over menu for exiting the game
157        game_over.add.button(
158            title="Exit",  # Button text
159            action=pm.events.EXIT,  # Exit the game when clicked
160        )
161
162        # Run the main loop of the game over menu on the specified surface
163        game_over.mainloop(self.surface)
```

There are different themes you can choose for the game_over object. This example uses THEME_BLUE. You can use any of the following themes to customize your menu.

```
THEME_BLUE
THEME_DARK
HEME_DEFAULT
THEME_GREEN
THEME_ORANGE
THEME_SOLARIZED'
```

Detect Collision

The display game over menu gets triggered by the bird hitting the bottom or top of the screen or running into a pipe.

```
165          # ----------------------- DETECT COLLISION ---------------------------- #
166          def detect_collision(self):
167              # If the bird hits the top or bottom of screen, game over
168              if self.bird_rect.bottom > config.HEIGHT or self.bird_rect.top < 0:
169                  self.display_game_over()
170
171              # The bird is between the pipes
172              if (
173                  self.bird_rect.right > self.pipe_upper_rect.left
174                  and self.bird_rect.right < self.pipe_upper_rect.right
175              ):
176
177                  # If the bird runs into a pipe, game over
178                  if (
179                      self.bird_rect.top < self.pipe_upper_rect.bottom
180                      or self.bird_rect.bottom > self.pipe_lower_rect.top
181                  ):
182                      self.display_game_over()
```

Detect collision runs each time through the game loop.

```
229          # --------------------------- GAME LOOP -------------------------------- #
230          def game_loop(self):
231              """Infinite game loop"""
232              while True:
233                  self.check_events()
234                  self.detect_collision()
235                  self.update_bird()
236                  self.update_pipes()
237                  self.update_background()
```

Modify and add a main program definition at the end of the program.

```
219                   # Cap game speed at 60 frames per second
220                   self.clock.tick(60)
221
222
223    # Program entry point, main function
224    def main():
225         # Create flappy bird program object
226         flappy_bird = FlappyBird()
227         # Start infinite game loop
228         flappy_bird.game_loop()
229
230
231    # Start the program
232    main()
```

Example run:

You can fly your bird up and down, collide with pipes and end the game.

Time to add some scoring and some sounds. Yes, it is time to finish up our Flappy Bird project!

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.