PyGame Flappy Bird Tutorial - Part 6

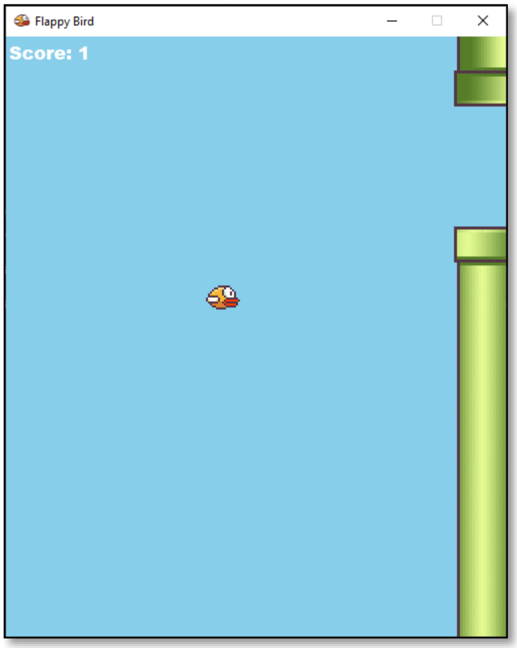
Contents

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 @%**(((&&*

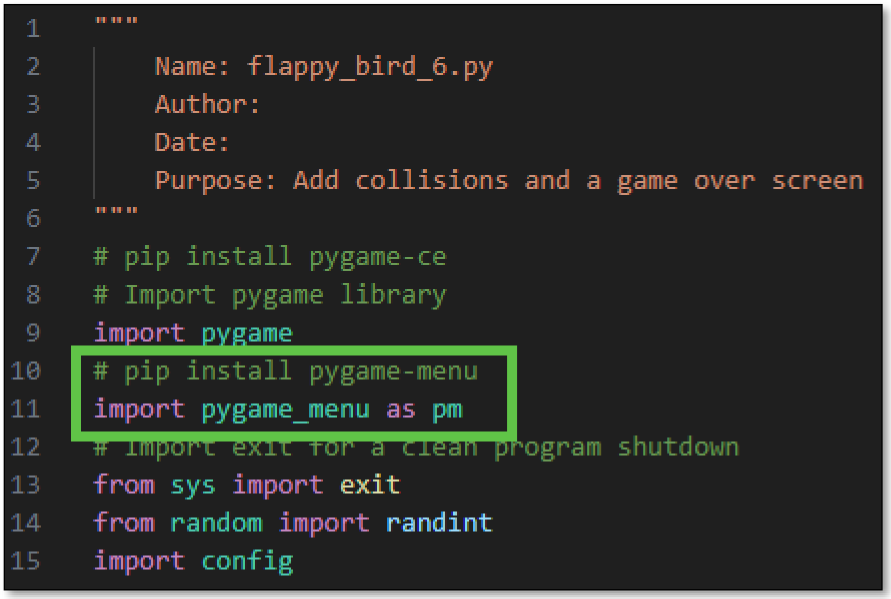
Time to add collisions and a game over display. We are going to use another library called PyGame Menu.

1. Install PyGame Menu.

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

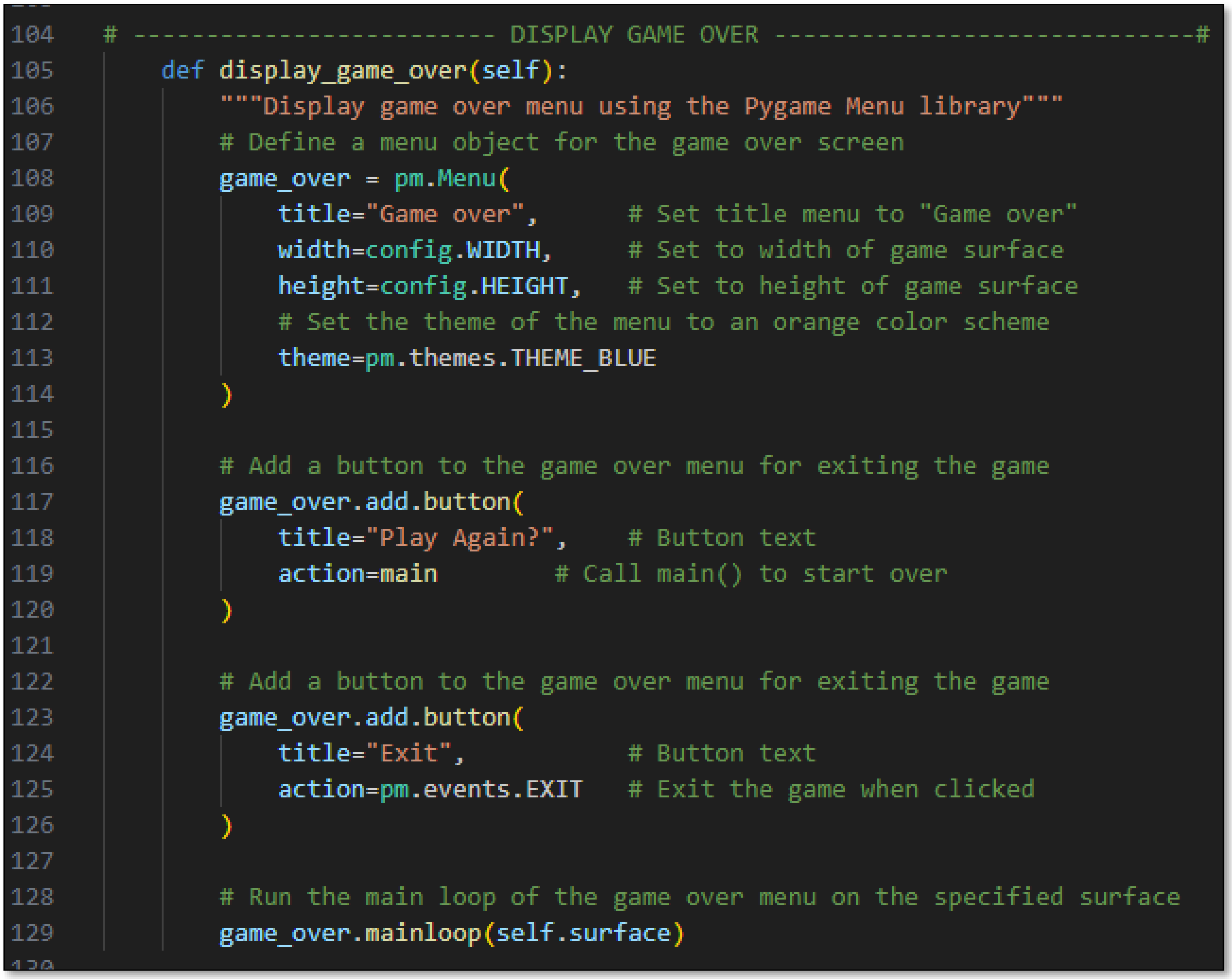
2. Save **flappy_bird_5.py** as **flappy_bird_6.py**

3. Modify the existing code.



```python
"""
    Name: flappy_bird_6.py
    Author:
    Date:
    Purpose: Add collisions and a game over screen
"""
# pip install pygame-ce
# Import pygame library
import pygame
# pip install pygame-menu
import pygame_menu as pm
# Import exit for a clean program shutdown
from sys import exit
from random import randint
import config
```

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

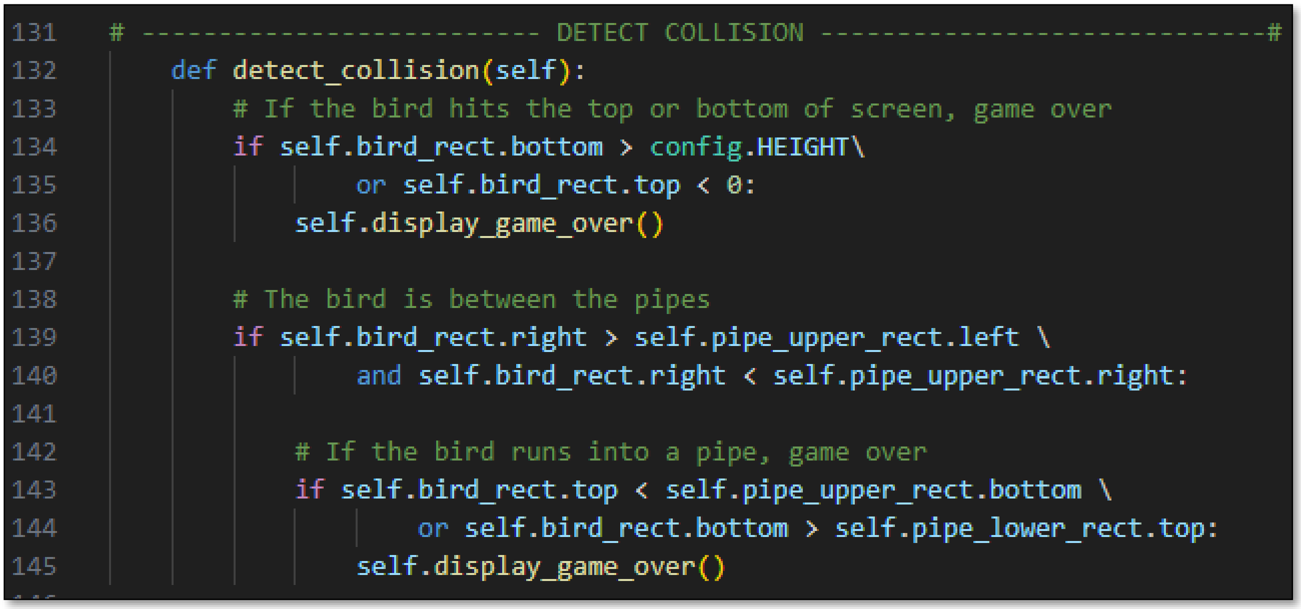


```python
# ------------------------- DISPLAY GAME OVER -----------------------------#
    def display_game_over(self):
        """Display game over menu using the Pygame Menu library"""
        # Define a menu object for the game over screen
        game_over = pm.Menu(
            title="Game over",      # Set title menu to "Game over"
            width=config.WIDTH,     # Set to width of game surface
            height=config.HEIGHT,   # Set to height of game surface
            # Set the theme of the menu to an orange color scheme
            theme=pm.themes.THEME_BLUE
        )

        # Add a button to the game over menu for exiting the game
        game_over.add.button(
            title="Play Again?",    # Button text
            action=main        # Call main() to start over
        )

        # Add a button to the game over menu for exiting the game
        game_over.add.button(
            title="Exit",           # Button text
            action=pm.events.EXIT   # Exit the game when clicked
        )

        # Run the main loop of the game over menu on the specified surface
        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 to customize your menu.

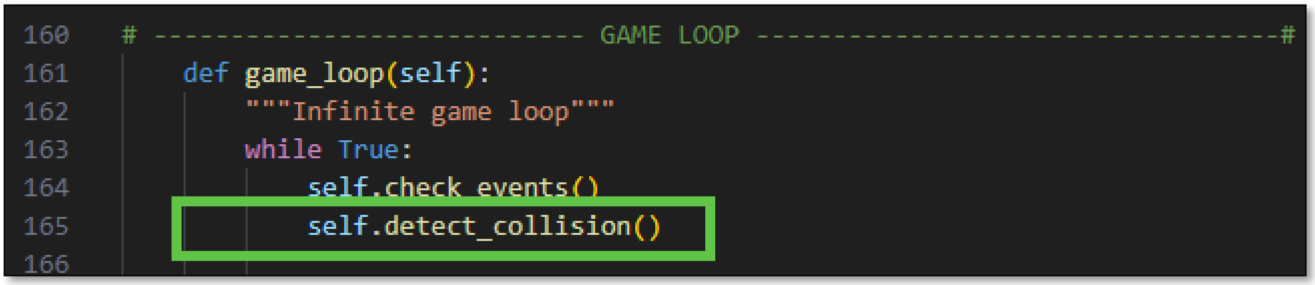
```
THEME_BLUE
THEME_DARK
HEME_DEFAULT
THEME_GREEN
THEME_ORANGE
THEME_SOLARIZED'
```

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



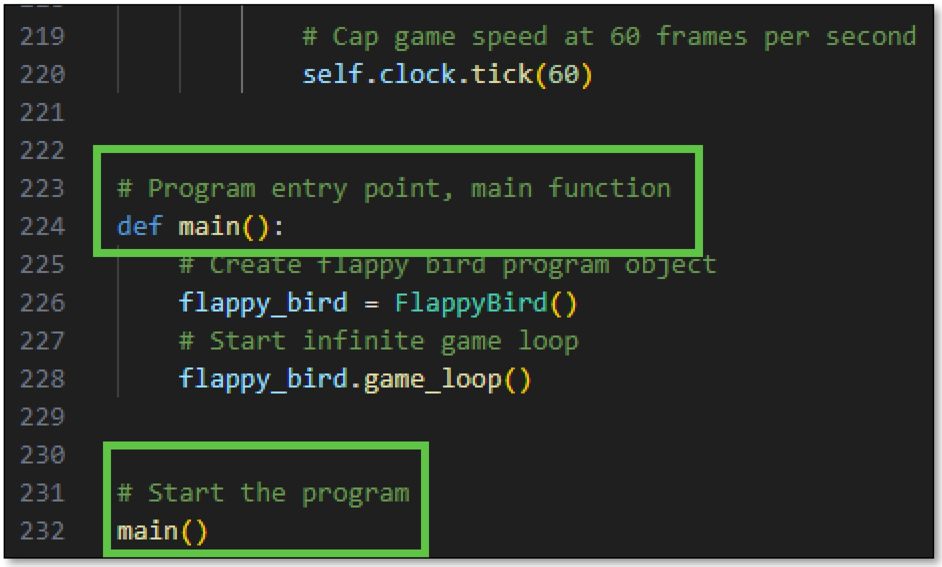
```python
# --------------------------- DETECT COLLISION ------------------------------#
    def detect_collision(self):
        # If the bird hits the top or bottom of screen, game over
        if self.bird_rect.bottom > config.HEIGHT\
                or self.bird_rect.top < 0:
            self.display_game_over()

        # The bird is between the pipes
        if self.bird_rect.right > self.pipe_upper_rect.left \
                and self.bird_rect.right < self.pipe_upper_rect.right:

            # If the bird runs into a pipe, game over
            if self.bird_rect.top < self.pipe_upper_rect.bottom \
                    or self.bird_rect.bottom > self.pipe_lower_rect.top:
                self.display_game_over()
```

Detect collision runs each time through the game loop.



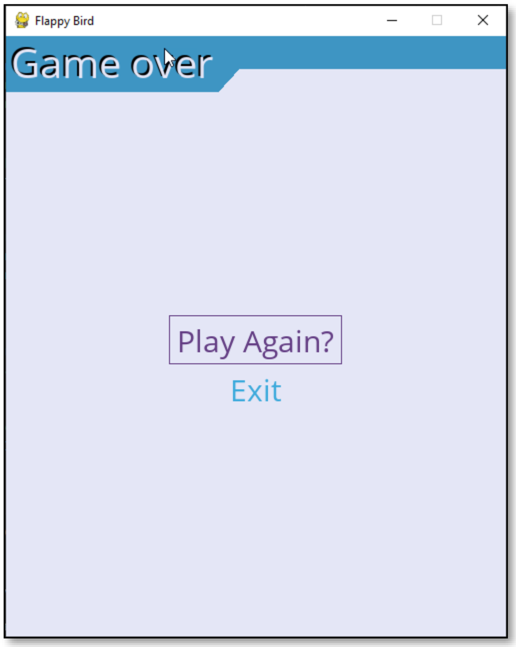
```python
# ---------------------------- GAME LOOP ----------------------------------#
    def game_loop(self):
        """Infinite game loop"""
        while True:
            self.check_events()
            self.detect_collision()
```

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



```python
            # Cap game speed at 60 frames per second
            self.clock.tick(60)


# Program entry point, main function
def main():
    # Create flappy bird program object
    flappy_bird = FlappyBird()
    # Start infinite game loop
    flappy_bird.game_loop()


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