CS 1410: Pong

In computer science, classes and object-oriented programming are very commonly used when creating programs that involve computer graphics and graphical user interfaces. To display something visually on the screen, like a button, a class is used to represent a Button, and multiple instances of the Button class can be created to display multiple buttons on the screen at one time. This is the foundation on which all modern applications are created, whether it’s on your computer, phone, or television.

Pong was one of the first arcade games released to the public. In this assignment, you will create a one player version of the game. You can watch the two player version in many places.

Assignment

Your assignment is to create a program using Python and Pygame that allows a user to play a one player version of Pong.

At a minimum, your program should include:

- A paddle object that the user can control vertical motion using the keyboard.
- A ball object that bounces from the top, bottom and one side wall. The ball object must also bounce from the paddle. The ball should be served about 1/3 of the way from the left. The serve should have randomized speeds to the right, and up or down. If the ball goes past the paddle, it should be re-served.
- A game score. The game scores points for the user every time the ball bounces from the paddle. The game scores points for the computer every time the ball goes past the paddle.
- A game object that contains the paddle, ball, and other game information.
- A collision detection mechanism to recognize when the ball bounces off of the paddle.
- A mechanism to detect when the ball passes the paddle.

You must use classes and objects to construct this program. There must be a top level class that represents the game, and classes and objects created for all of the elements in the game.

Start by downloading the provided Pygame starter kit by clicking here.

Code Requirements

- Each element of the game must be an object of a class.
- Each class implementation must be in its own source file.

Extra Challenges

- Add a second player.
- Allow a game restart.
- Define an end game (e.g. first to 10 points wins).
- Add a start screen, and allow the user to start the game.
- When a reserve is necessary, let the user serve with a key press.
- Add a restart option to the game so the player doesn’t have to exit the application and start it again to restart game play.
- Add a high score as well as the current score. This requires that the restart option is available.
- Add a high score file so that the game can remember the best score ever, not just the best score since the program was launched.
- Add sound.
Hints

- The ball may be round or square. Your choice.
- Build the program one piece at a time. For example, start with a picture that has an unmoving ball and an unmoving paddle. Proceed to have the ball move. Next, have the ball bounce off of the walls. Then, allow the user to move the paddle up or down with the keyboard. And so forth.
- Refer to the Pygame documentation to understand which parameters are necessary when calling each of the Pygame draw methods. Specifically, you should be interested in `pygame.draw` and `pygame.Rect`.
- When creating colors, use a helpful tool to determine the RGB values. Here are two good options: color.adobe.com and colorpicker.com