Short Description

The Pong game from Atari is one of the earliest arcade games. In this lab, you will create a graphical two-player Pong game. The left player controls the left paddle by pressing and holding down the `q` key to move the paddle up and the `a` key to move the paddle down. The right player controls the right paddle by pressing and holding down the `p` key to move the paddle up and the `l` key to move the paddle down. These keys were chosen since `q` and `a` are on the left side of the keyboard and `p` and `l` are on the right side so both players can share a single keyboard and not get in each other's way.

Learning Outcomes

- Practice basic programming skills
- Discover new language features[1]
- Learn how to organize your software according to classes and functions
- Use pygame to create graphical game programs

Detailed Explanation and Things To Do

- Study the following video to understand how the game Pong is supposed to look like and work:
    - Video of the "Pong" game with two more game start clips at the end of the video.
- Design and write code for a program that implements Pong as shown in the above video.
- Your code must be based on our final pre-poke framework, which imports pygame, and implements the game framework in its own class. You are not allowed to import any additional modules, other than modules that are in the Python Standard Library, if needed.
- In addition to the Game class, your code must contain a Ball class, representing the ball, and a Paddle class, representing the paddles, in the Pong game. To represent the shape and location of a paddle you can use an object of the Pygame type Rect in your Paddle class.

[1] To create this game, you may need to use a few programming language features that have not been used yet in class. Discovering new language features and how to use them is an integral part of problem-solving in computing science and an essential skill that you should learn. Think about what you need to do, search the web for python3 programming examples, and/or use the Python documentation and pygame documentation to help you find the programming constructs that you need. If you get stuck, ask your TA for help/hints about the programming constructs you need to use.

Hints

- To determine whether the ball has collided with the paddle, it is sufficient for you to check whether the centre of the ball is inside the paddle using the method collidepoint() in the Rect class from the pygame module.

- When moving a paddle, you will need to know the y coordinates of the top and bottom of the paddle's Rect so that you don't move it out of the window. Every Rect has a set of attributes listed in the documentation for the Rect type. For example, if you had a variable called paddle that was bound to one of the paddle's Rect, you could compute the y coordinate of the top of the rectangle, using the attribute reference expression: **paddle.top**

- To allow both players to move their paddle at the same time, you need to detect whether two keys are being pressed and held down at the same time, and compute the corresponding paddle movement accordingly. There are a number of options to achieve this. We recommend the following, which is probably the most elegant option for this game, given our framework:

  Listen for KEYDOWN and KEYUP events. When a KEYDOWN event for a player occurs, change the corresponding paddle's velocity similar to what you do with a dot or Pong ball. When a KEYUP event for a player occurs, set the corresponding paddle's velocity/speed to zero. Move paddles in the game's update method according to their velocities.

- Make sure you are following code quality standards outlined in the Software Quality Requirements for code with classes.

Resources

- Your code must be based on our pre-poke framework.

Submission Information

- You will be asked to submit the solution for this mini-project along with other lab exercises at a later point in time for a Lab Checkpoint. For submission purpose, the file with your code should be named:

  `pong.py`