

### ***Introduction***

The purpose of the lab was to get experience in developing a basic game with python and the pygame framework. By doing this, it helped acclimate me to the different syntax quirks, conventions, and strategy when developing in python and pygame. I chose to create the game pong.

### ***Results and Discussion***

I implemented the game by organizing each aspect of the game state into its own class with a series of definitions. In particular, I had a class for Ball objects, a class for Paddle Objects, and a class for a Scoreboard Object and the game state consisted of the Ball's position & velocity, the Paddles' positions, and the scores on the scoreboard.

The game loop follows like this:

- draw background and scoreboard
- check for collisions
  - a collision is observed if the ball hits the top wall, bottom wall, or one of the paddles
  - if there is a collision then the ball's velocity is changed accordingly
  - if there is a collision it'll play the appropriate collision sound
- draw balls and paddles
- updates the display to show balls, paddles, scoreboard and background
- if a player has just scored previously then there is a short pause
- then it checks to see if anyone has scored
  - A score is observed if the ball goes beyond the game window
  - if someone has scored the score sound plays
  - if someone has scored it ticks the scoreboard and resets the ball and paddles to their starting positions
- checks for player input
  - if player is giving valid input (the w or s key) then it sets the boolean that moves paddles accordingly
    - if the w key is pressed it'll set the paddle.up boolean to true so that the game knows to move the paddle up
    - if a player releases a key it sets the according boolean to false so that the paddle will cease moving
- Next it implements the computers paddle movement
  - the paddle is set up so that every tenth of a second it checks to see whether the ball is above or below it and will set its move boolean accordingly
    - i.e. if at some tenth second interval the ball is above the computer's paddle it will set the paddle.up boolean to true
  - every two fifths of a second the paddle checks if the ball is within the bounds of the paddles y coordinates and if it is then the paddle will stop moving
    - i.e. if at some two-fifths interval the paddle and ball are at the same y coordinates then paddle.up and paddle.down will be set to false so the paddle doesn't move
- After the booleans for paddle movement are set they're positions are changed based on their velocity fields.

## ***Conclusion***

From doing this I have learned basic structure and organization of game loops and simple ways to affect the game state through objects and classes. One of the more interesting things I have discovered is the challenge in programming computer players into a game. The easiest way to of programmed an ai player would be to have the program check for the balls position each from and move the ball towards it but obviously that would make the game impossible to win and no fun. What I ended up having to do was implement a delay to all of the computer's actions so that it would essentially have a chance of being able to lose. Overall I think it works pretty well and it also has potential to make implementing difficulty levels easier (more difficult levels could just have shorter delays in computer response time).