**Game Image:**

**Text

Description automatically generated**

**Program Version**

**OS Version:** Windows 10

**Python Version:** Python 3.10.10

**Pygame Version:** Version: 2.1.2

**Motivation:** Pong is a game that was initially developed by Atari in 1972. It was the second game that Atari developed. The game is so simple and yet so fun to play that it made me curious how this game is developed. Also, since this is the first game that I'm developing I have chosen this game rather than 2D games that are fairly more complex like Tetris or pacman which would make me focus more on the game logic rather than the code design on developing games which I feel is more important during initial stages of learning. I have chosen pygame as the game engine to build this game as it is easy to implement the game using python which has a rich set of library functions making it easier to develop rather than languages such as c++. This game was originally played in arcade video games and then it was played at home in console video game and now people can play it on their own personal computers!

**Reasoning:** I have used the Model-View-Controller architecture to develop this game. The main function will start the game loop and during each iteration the position of the ball is updated by 1 in both axes in the controller. Input events are also checked during the iterations and the paddle position are updated accordingly. the keys up and down are used the move the right paddle and the keys w and s are used the move the left paddle. Logic for collision between the ball and the walls and paddles are also written in the controller. If a collision is detected the direction of the ball is reversed in the x direction if the collision is with the paddle, in the y direction if the collision is the wall. The controller updates both the paddle and ball position during each iteration in the model. The View in turn reads the parameters from the model and updates the display accordingly. As the processor speed is too fast, I have limited the game to less than or equal to 120 frames per second, so the gameplay is smooth. Meaning the display is updated 120 times or less in a second.

**Classes and Functions:**

**Diagram

Description automatically generated**

**Future work:** I have developed a simple version on the pong game. The game can be made even more difficult and challenging to play by increasing the speed of the ball by a factored each time a collision occurs. Another variation of the pong game which consists of a paddle and bricks can be made. In my code the window width should be given in multiples of 8 as the paddle will move 8 steps at a time. If the window width is not a multiple of 8 the paddle will move out of the screen. This issue needs to be addressed. Generalization: Objects of the entities paddle, and ball can be created. If we were to use some of this code in other similar games, we can use the same entities and model. The handle Event function of the controller needs to be updated accordingly. The logic behind the implementation and the view code can be configured based on the game requiremen**t.**