

**BINUS UNIVERSITY**



**BINUS INTERNATIONAL**



**Assignment Cover Letter**

**(Individual Work** **)**

**Student Information :**

**Surname**  **Given Names Student ID Number**

Avatara  Ida Bagus Ryogassa (Ryo) 2440100323

**Course Name :** Program Design Methods

**Name of Lecturer(s) :** Jude Joseph Lamug Martinez

**Course Code**  **:** COMP6056

**Class**  **:** L1AC

**Major**   **:** Computer Science

**Title of Assignment** **:** Pong Game (with PyGame)

**Type of Assignment** **:** Final Project

**Submission Pattern :**

**Due Date**  **:** 13th January 2021   **Submission Date**  **:**  13th January 2021

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer’s instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

**Plagiarism/Cheating**

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

**Declaration of Originality**

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student:

Ida Bagus Ryogassa Avatara

**Program Description:**

The game is made with python 3.9 with implementation of the newest pygame update. The game is a recreation of one of the most popular games in the 70s-80s era of video games, known as “pong”. Pong is an arcade game based on the sports table tennis made in 2D.

**How to play the game?**

The game is really made to be simple and accessible for everyone. It is also built mainly for 2 players, meaning that you could battle your way to victory against someone else. To play the game, you must press the “p” key on the keyboard to start. Use your “w” and “s” key to move up and down for player 1 (left paddle) and “up” and “down” key to move up and down for player 2 (right paddle). The ball will start in the middle. Either player must score a goal of the other player in order for you to win. However, there is no limit to how many goals you can score, it is a matter of who gets the most point. If you want to restart the game, meaning all the scores are all reset to 0, you can press the “r” key on the keyboard.

**Library Used:**

Only 2 libraries are used to create the game itself, which are

* **Pygame:**

This allows me to use all the pygame functions on python.

* **Sys:**

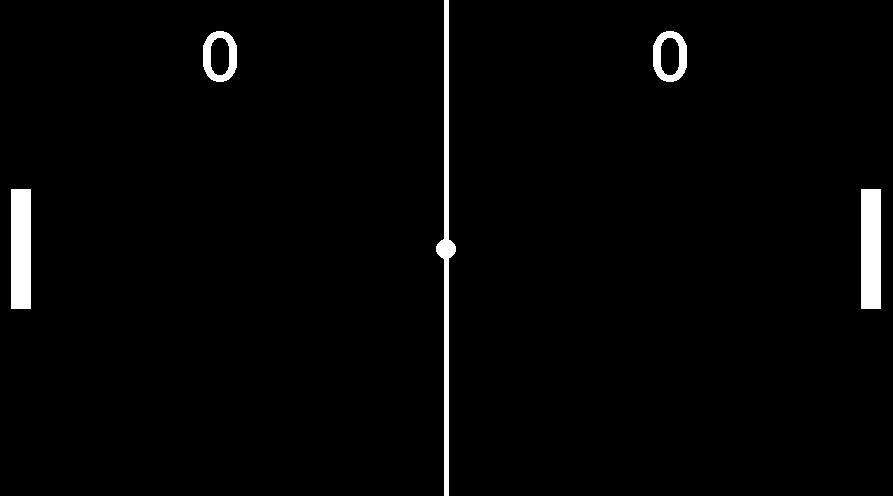
This allows me to use constants, methods, functions on python.

* **Winsound:**

This allows me to use sound machinery

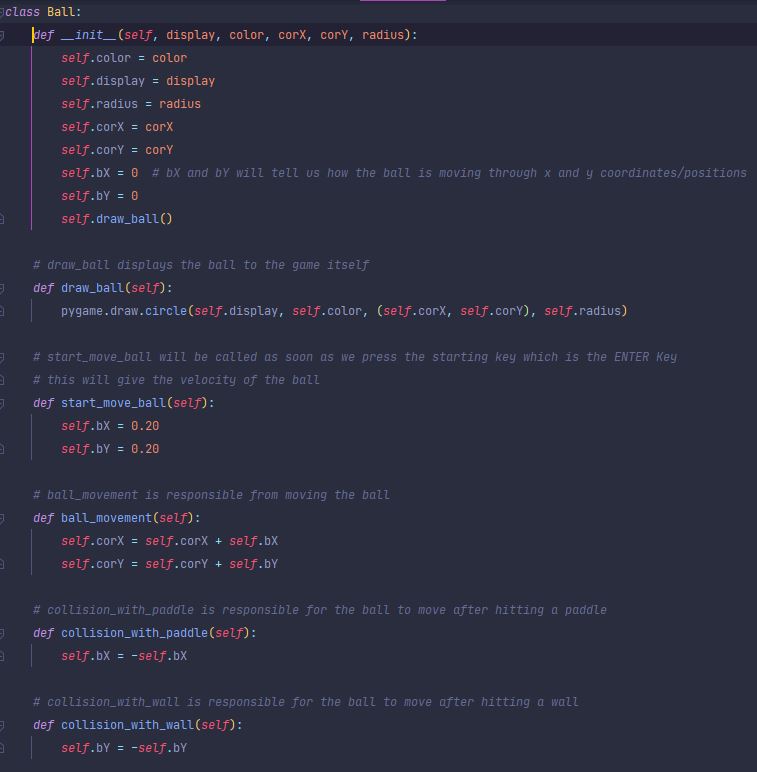
**What I have learned:**

This is actually my first project for python and it is also my first time learning python. When creating a game, I also had to learn pygame in my own time, and it is really easy to understand and implement it into the codes. Learning python, however, was actually harder than I thought it would be, even though it was still on the basics side of it. I actually wanted to make a website with Django, but I was not really interested in it anymore. Therefore, I decided to create childhood game, pong.

****

**This is the pong game board. Code explanation down below.**

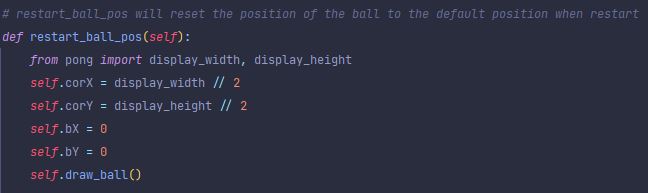
**Code explanation:**

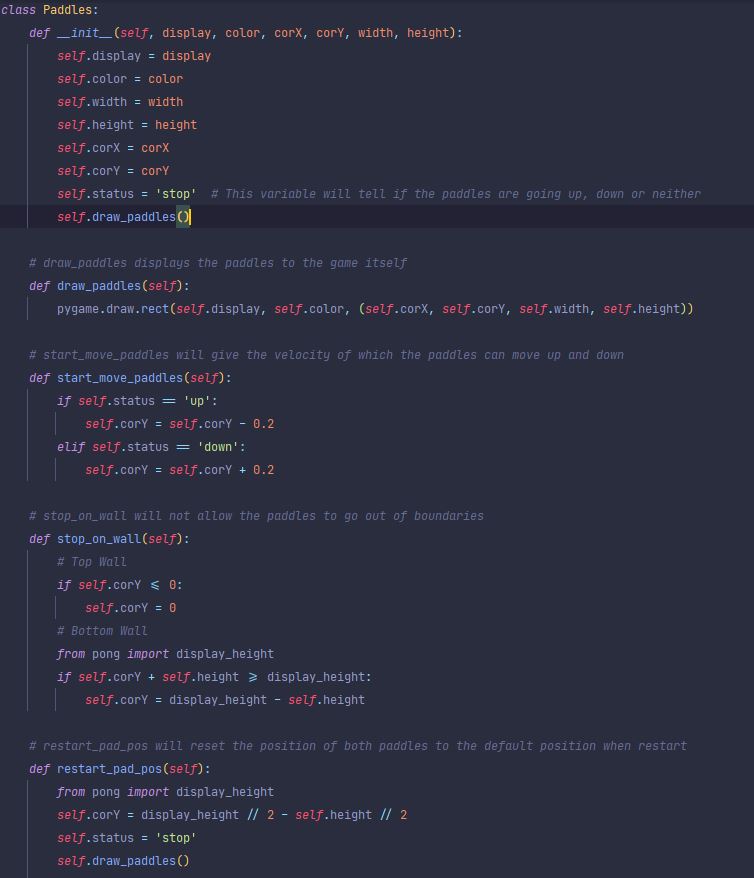
**ball.py (ball class)**

This is the ball class. Ball receive display, color, corX (position/coordinate x), corY(position/coordinate y), and radius as parameters. Draw\_ball function will display the the ball to the game itself. Start\_move\_ball will determine the velocity/speed of the ball.

Ball\_movement is responsible for moving the ball throughout the x and y coordinates.

Collision with paddle/wall is responsible for the ball to move to the opposite direction after hitting a paddle or a wall. There was a bit of calculations here to find how the ball will react as soon as it hits a barrier.

****This function will reset the position of the ball to the default position which is in the middle. This will be called when the player press restart or after a goal.

**Paddle.py (paddle class)**

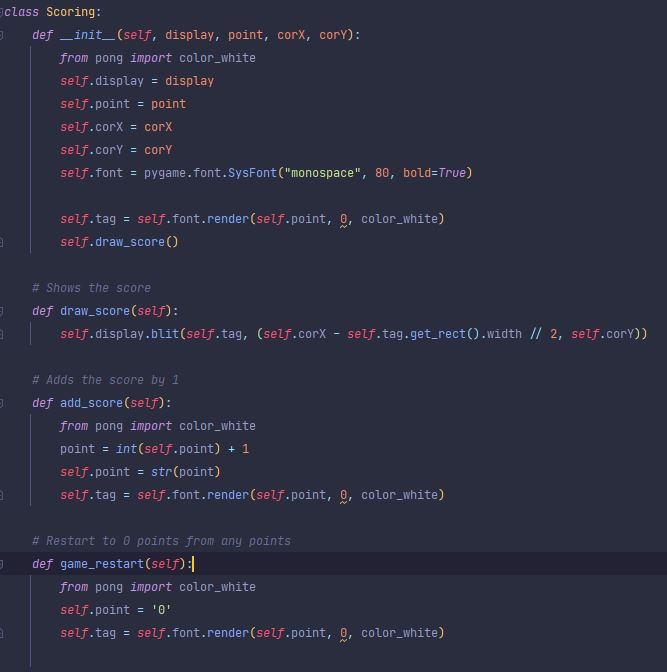
This is the paddle class. This is class to determine the characteristics and functions of the both paddles.

**Draw\_paddles** will display the paddle to the game.

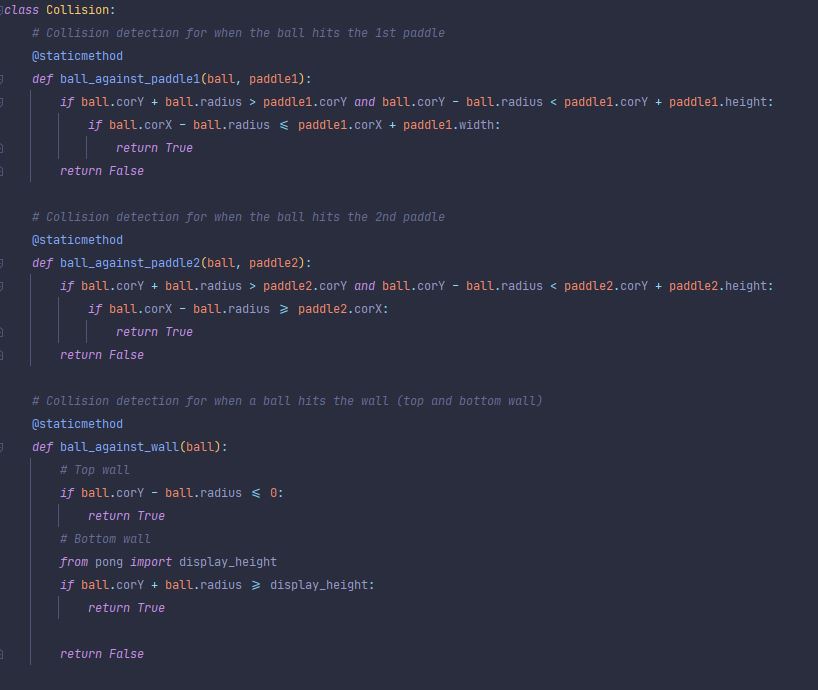
**Start\_move\_paddles** will give the velocity/speed of how the player moves up and down.

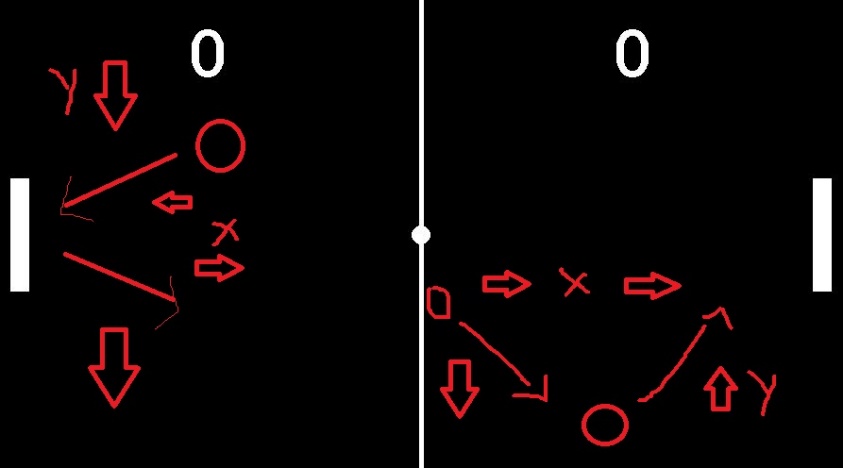
**Stop\_on\_wall** will not allow the paddles to go out of boundaries. Without this function, the paddles can go through the top and bottom wall.

Like the ball class, **restart\_pad\_pos** will reset the position of both paddles to the default position when restart or when a goal is scored.

**Scoring.py (scoring class)**

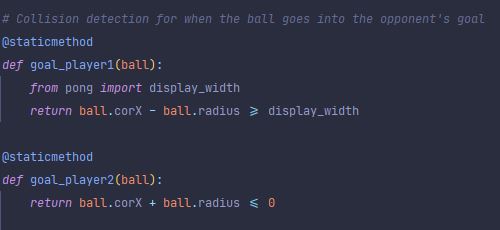
This is the scoring class. This is responsible to update the score in the game. **Self.font** is used to determine what font you want to use for the scores. Add\_score will add the score by 1 if either player scores a goal. Game\_restart will restart the score to 0.

**Collision.py**

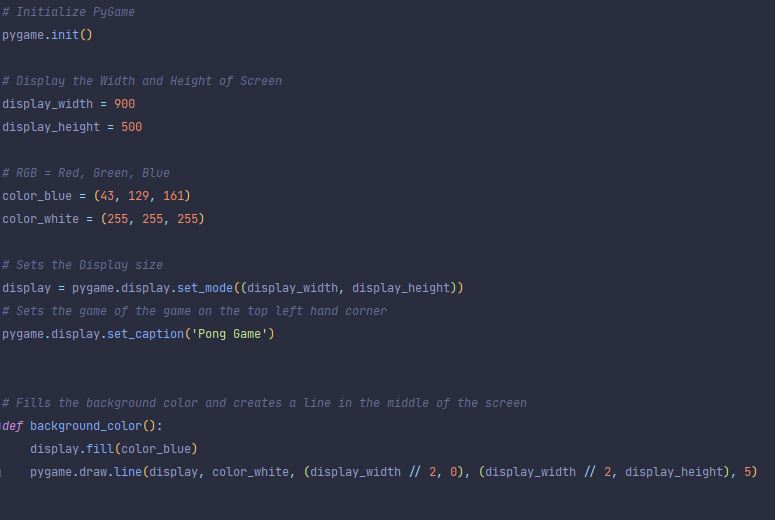
This is the collision class that is responsible for checking the collisions of the ball when it hits the paddle or the wall. Ball\_against\_paddle1/2 checks if the ball hits paddle 1 (player 1 right) or paddle 2 (player 2 left). Ball\_against\_wall, however, checks if the ball hits the top wall or the bottom wall because that is the only walls that ball can bounce on.

In this picture, it explains

This picture explains how the ball bounces of the wall. On the illustration on the left, if the ball is coming towards the paddle, the y coordinate doesn’t change because it starts from going down and finishes going down as well. But the x coordinate changes because it has to go to the opposite x direction after colliding with the paddle. Same goes with the wall where the x coordinate doesn’t change while the y coordinate changes to the opposite y direction (right illustration)



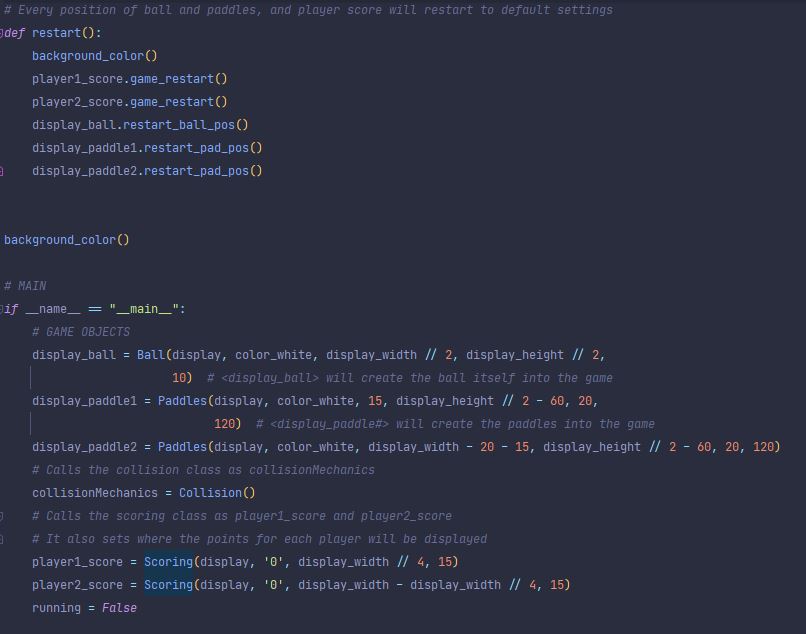
goal\_player1/2 checks if the ball goes through the opponents goal.

**Pong.py (main)**

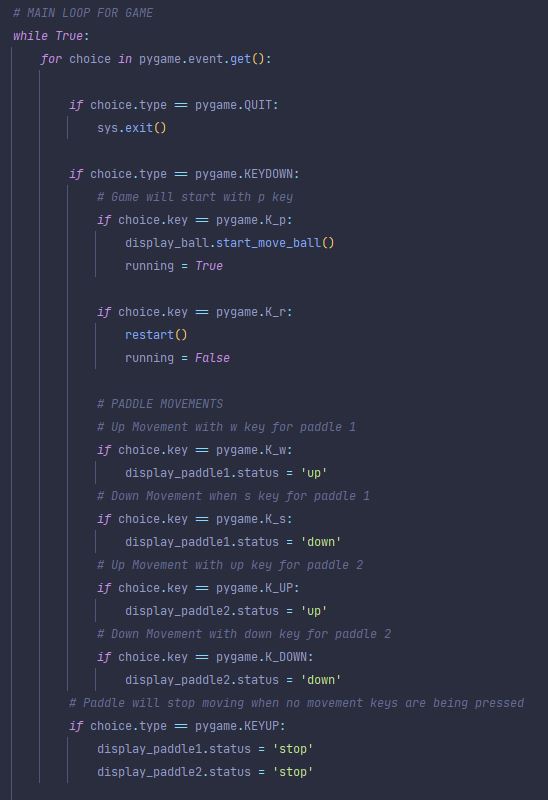
**Pygame.init()** was used to import all the modules and functions of pygame.

**Display\_width/height** was to create the size of the display/screen of the game. Like the picture of the game above. **Color\_blue/white** variable was used to set the color of the game later when it is being called.**Display** variable is to set the size of the display/screen of the game depending on the width and height which was 900x500. Below that was the set the name of the game as ‘pong game’ displayed on the top left of the window. **Background\_color** function was to set the color of the background as blue and **draw.line** was to create a line in the middle of the display. To create the line, I had to do some basic calculations because the measurements of each side (width and height) had to be considered. The function will be called below.

**pygame.draw.line(Surface, color, starting position, end position, width of the line)**

****

**restart** function is used to restart the game to the default settings. The score will be set to 0 and the starting positions of both paddles and the ball will be set to its default positions. Below it is the main, which will control the flow of the game. **Display\_ball** will display the ball to the middle of the screen. **Display\_paddle1/2** will display each paddles to both sides but also placed in the middle. Both of these variables was a bit difficult because of the calculations I had to do. Considering that both paddles and ball had to be placed in the exact position. I had a difficult time for this part and had help because of the use of calculations. **CollisionMechanics** will call the Collision class to work. **Player\_score1/2** will display the score on the desired position.

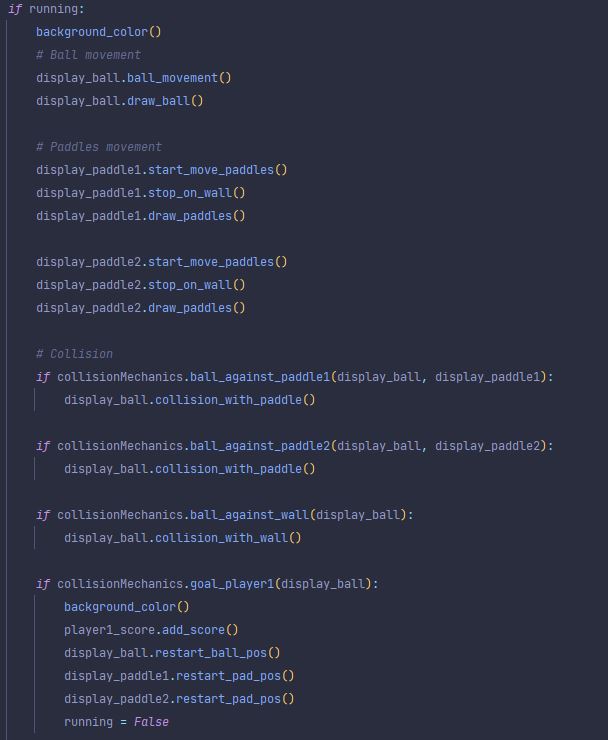


Continuing to the main, there is the main loop, which will control and loop the game properly. Basically the main loop will call all functions on each classes, depending on the user’s input/choice. Therefore, it is easier to understand and not be cluttered with numbers and what not.

**Pygame.event.get** will receive all the registered events from the player from a queue.

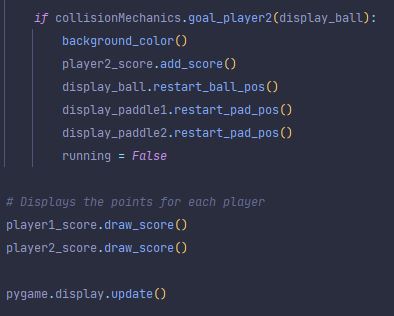
Basically, everything has already been explained in the actual codes with comments. As you can see in the photo on the left, we can already see the code’s functionality.

Also added sound functionality to the game to add more of the classic feel to the game.



Running starts with False, it will return true when the game is started with pressing the starting key ‘p’. After that, if running will start doing its thing. Everything variables and functions are being called in the loop. Ball and paddle movement is basically calling variables/functions inside each class. Collision is also doing the same thing, but it must check first either the ball hits each paddle, walls, and goals.

**Goal\_player1** will basically restart the game to its default settings, but the score will be added. **Goal\_player2** can not be seen in the picture but it has the same contents as goal\_player1.

**Player1/2\_score.draw\_score()** will update the score on the screen to display if goals were made.

**Pygame.display.update()** will update the whole window.

**What could have I done better**? Well, adding a computer-controlled paddle would be nice because not everyone can get someone to play with. But with limited knowledge of how to create it was really difficult and time consuming. More features that is different to the classic pong. Maybe add a game mode with more balls.

**References:**

* <https://www.youtube.com/watch?v=i6xMBig-pP4> (Pygame tutorial till end)
* <https://www.youtube.com/watch?v=C6jJg9Zan7w> (to get the gist of how pong works)