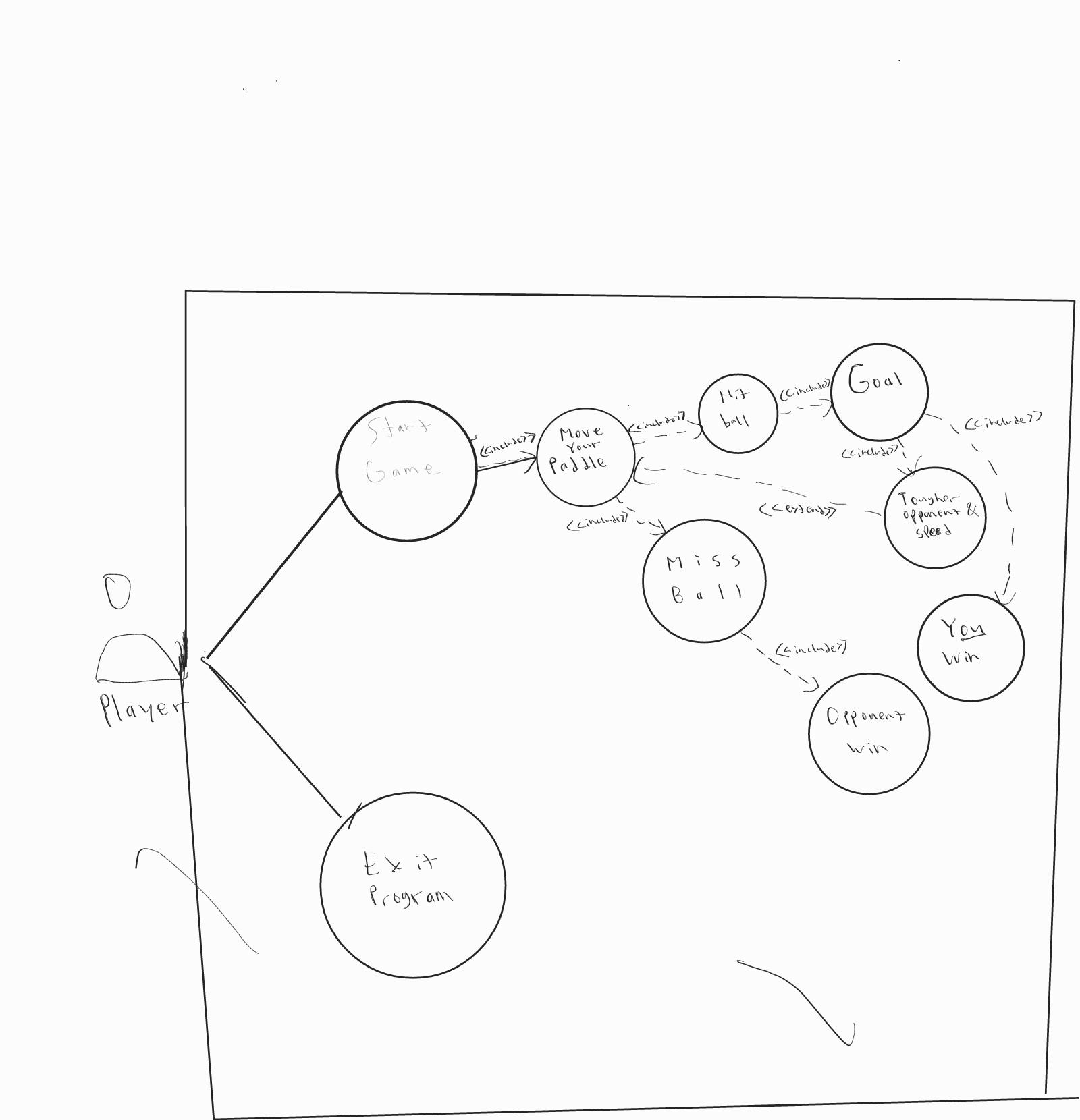
1. Brief Description

My project is a classic pong game, in which the gameplay consists of a ball bouncing around the screen with 2 paddles (one for the player and the other for the AI) moving across the edge of the screen to keep the ball from touching their goal side. The code is relatively simpler than other games, since it is comprised of less than 10 objects. However, as you score more, the game gets tougher until you reach the winning score, which is 10 points.

1. Use-case Diagram

The following is the use-case Diagram for the game:



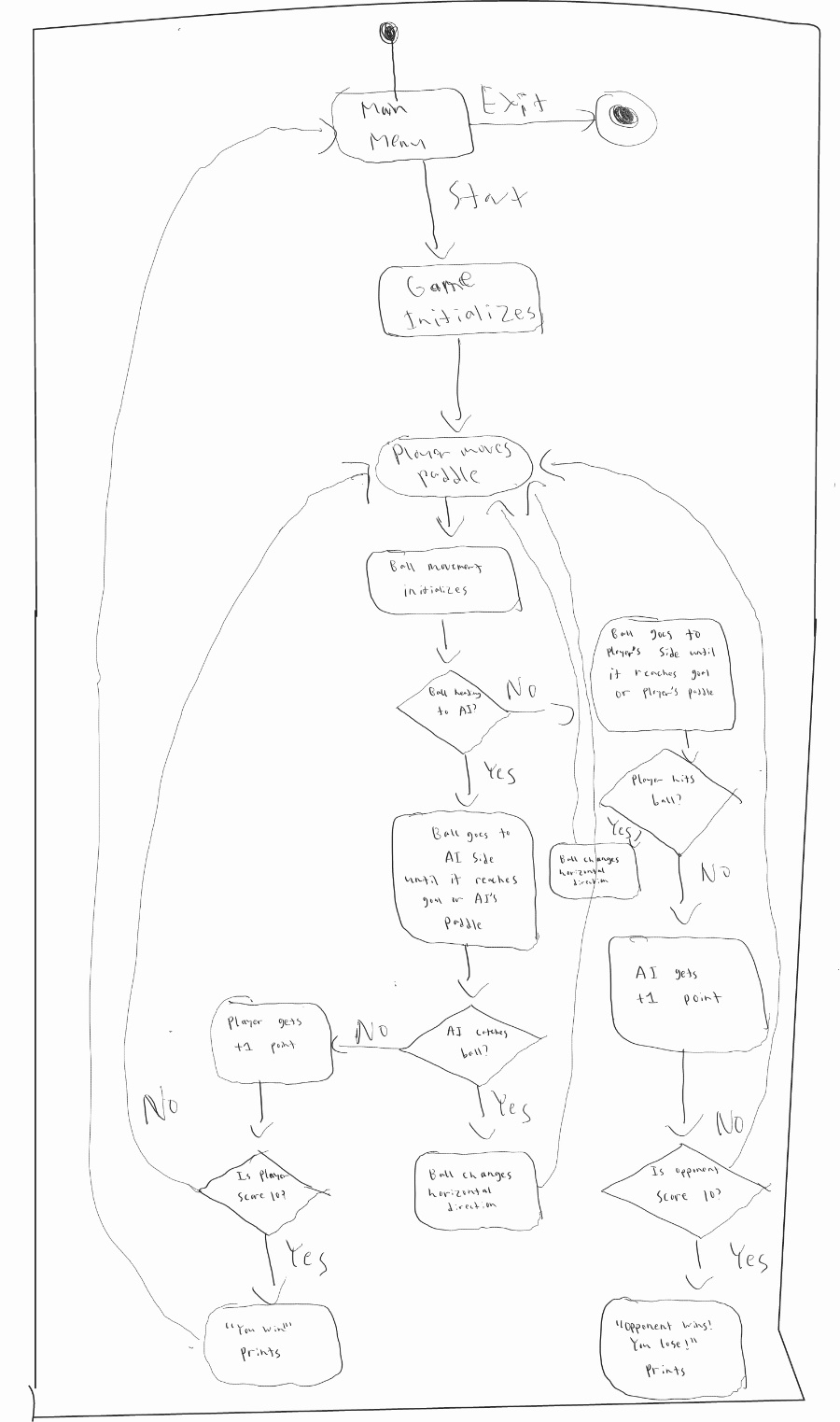
As shown in the diagram, there are only 2 options in the menu, which are “start game” and “end game”. You can start the game by pressing the “A” key. When the game is started, you will see 2 paddles, one for you, and another one for the AI. The ball will either go to your side, or the AI’s side. If you catch the ball, it will be bounced to the AI’s side. The AI will try to catch it. If the AI catches the ball, it will be bounced back to your side. Else if the AI fails to do so, the game will pause and give one point to you. Keep in mind that the more you score, the opponent and the ball will be faster, making the game tougher. However, if you fail to catch the ball, the AI will score one point.

If you or the AI score 10 points, then you will win the game, with the text “You win!” appearing on screen and the text “Opponent wins!” if the opponent wins, after either, you will be redirected to the main menu.

If you choose to exit the program by pressing the “X” key, the program will terminate.

1. Activity Diagram

Here is the activity diagram as shown below:



The diagram explains the process in the algorithm. We initially go to the main menu after starting. If you choose to start the game, the gameplay initializes and prompts you to move your pong paddle. It also initializes the ball’s direction and speed. The system detects whether the ball is heading to the AI’s side or otherwise, and it will keep moving until the ball hits the AI or the goal. If the AI catches the ball, the ball will bounce to the player’s side. If not, the player will score a point. Then, it checks whether the player has reached 10 points. If it does, the player wins the game and prints the victory message, then the system redirects you to the main menu. Otherwise, the game starts the new round with the paddles and the ball reset. The process repeats until one of the actors won.

After the case where the AI catches the ball, the ball will move to the player’s direction and will keep going until it reaches the goal or the player. If the player successfully hits the ball, it will be bounced back to the AI’s side and the process repeats. But if the player misses the ball, the AI will score a point. Later, it checks whether the AI has scored 10 points. If it does, the opponent wins and the game print the defeat message, and the game redirects you to the main menu. Otherwise, a new round starts each goal until one of the sides win.

1. Class Diagram

Apparently, the game does not use classes, yet it still works.

1. Modules

These are the modules I used for the game:

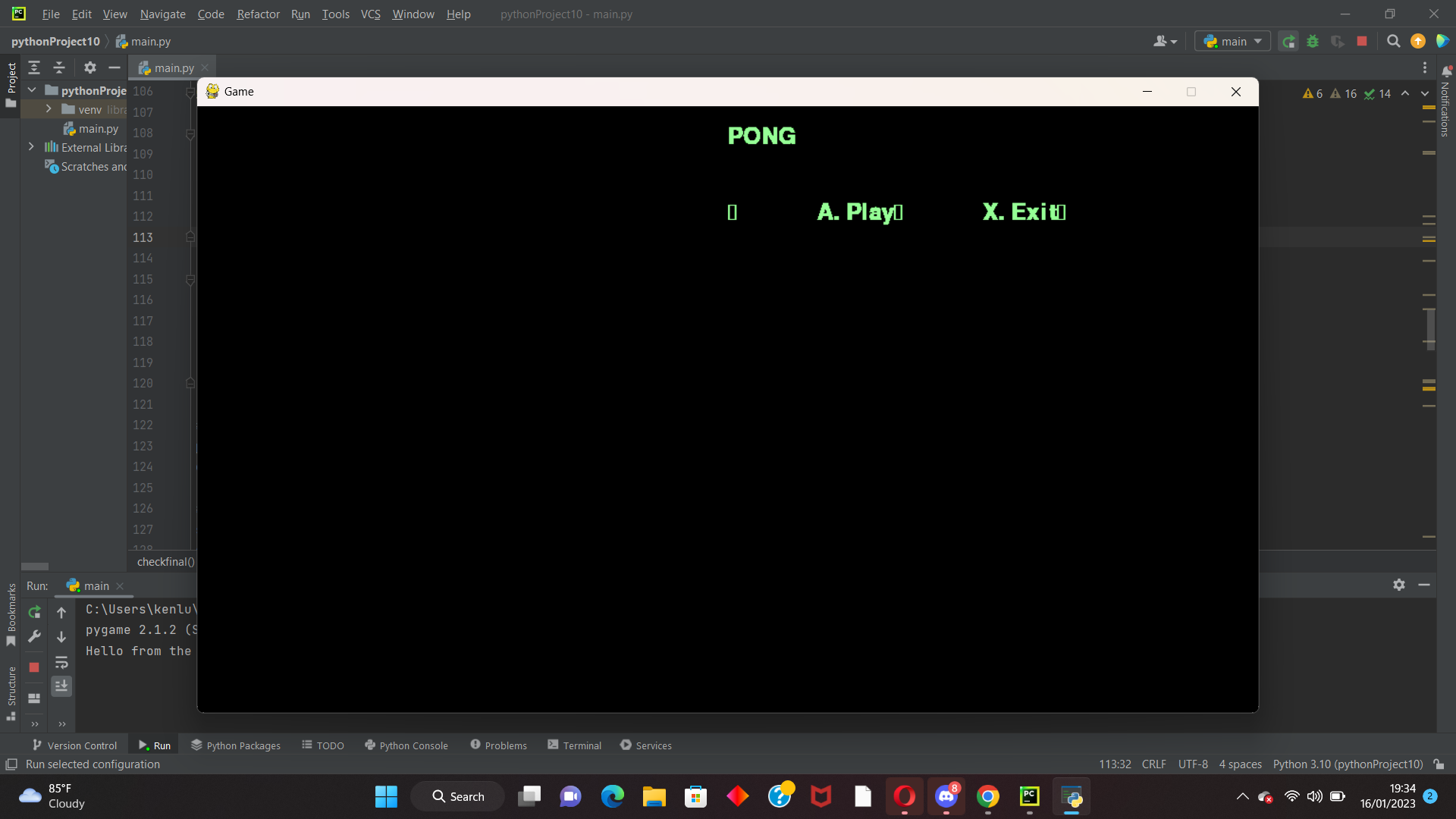
* pygame
* Sys
* random
* threading
* Time

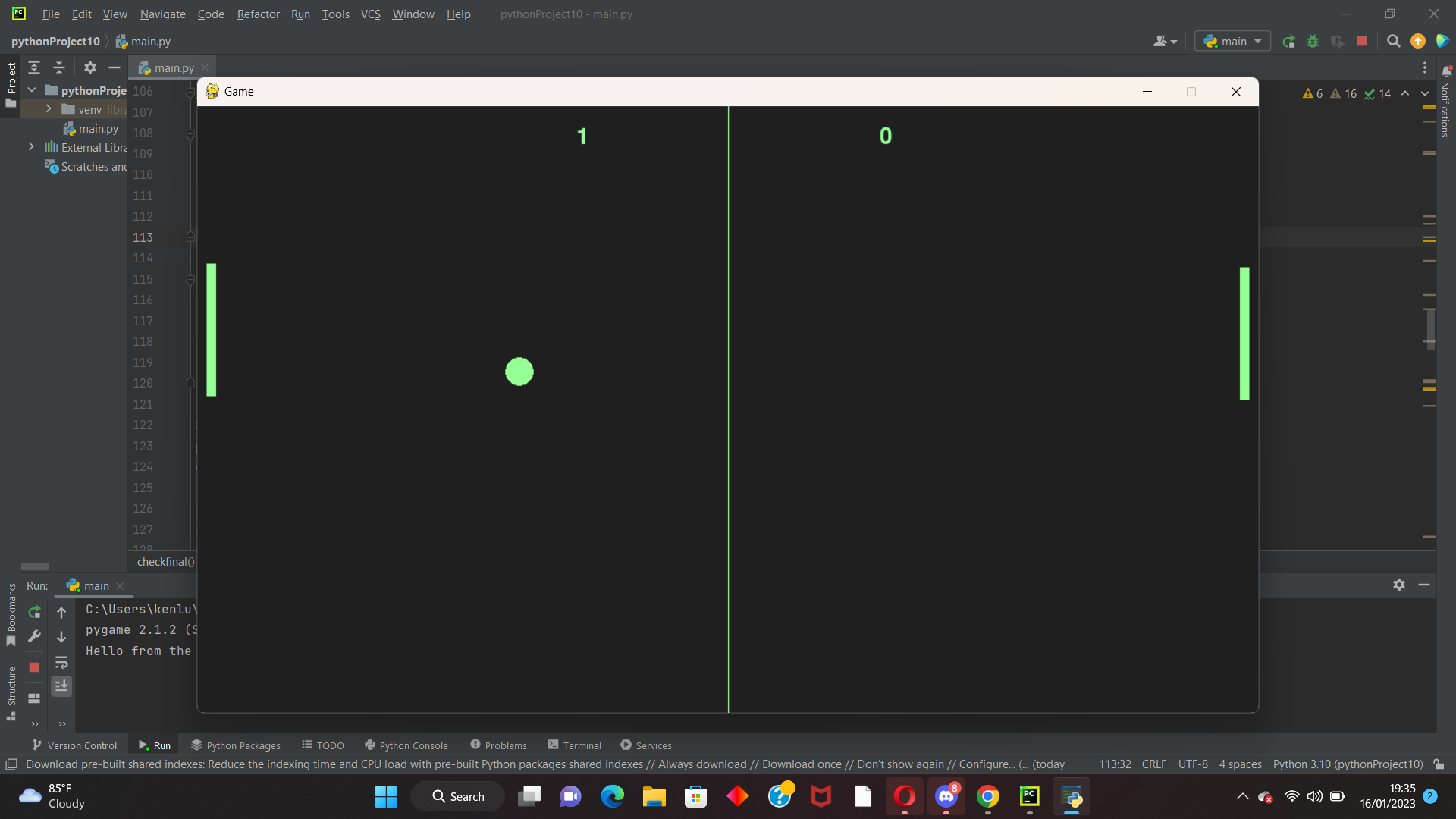
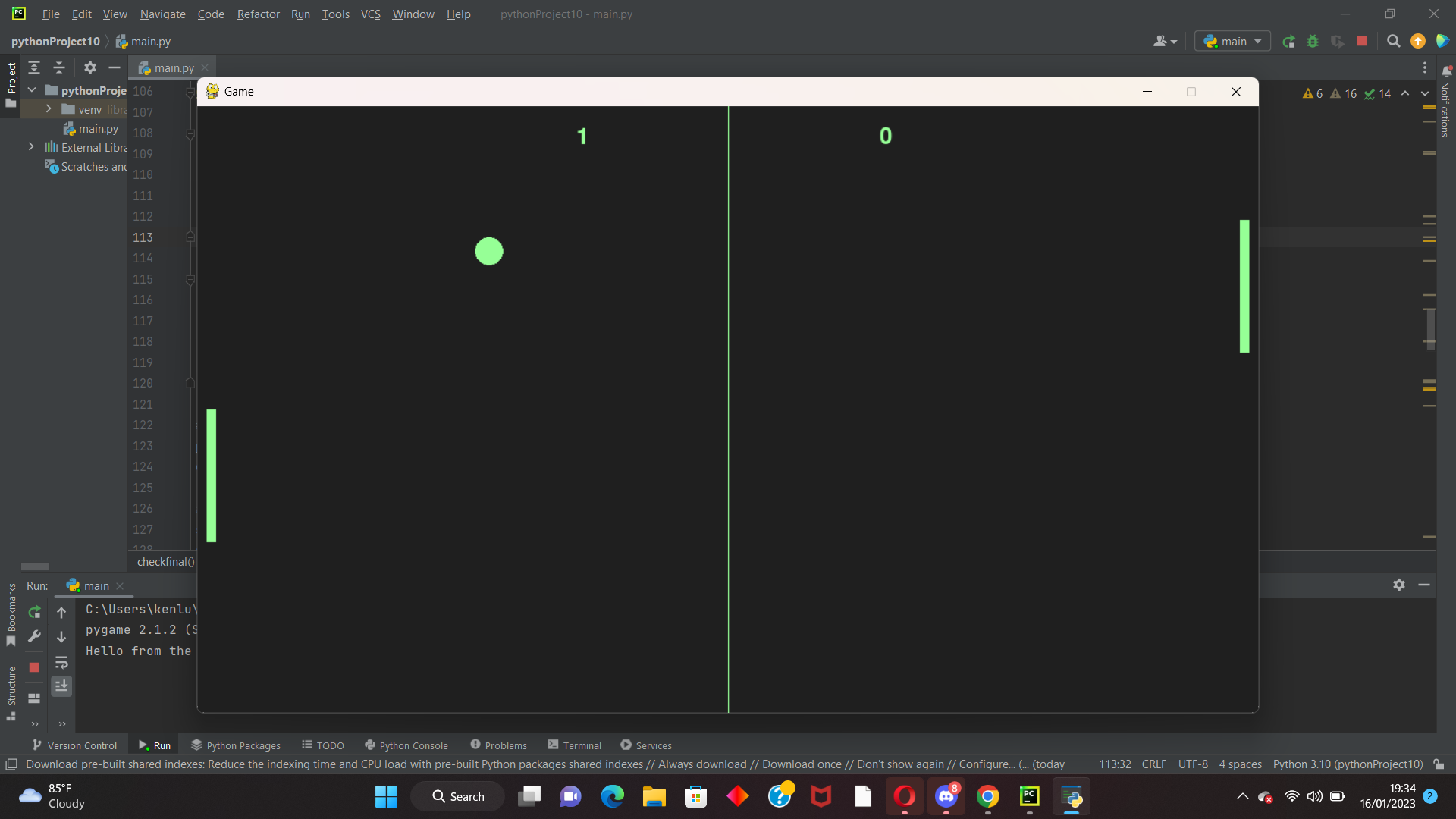
1. Essential Algorithms

The essential algorithms are the basic screen, texts, the main loop to run the game and some predetermined functions to represent game objects and its interactions. Notably, the 2 most important algorithms are ball animations and the opponent’s AI. In order to make a pong game, a ball makes up most of the gameplay quality (without it, it is not a proper pong game). Therefore, I created a function for the ball animation. Secondly, the AI is needed in order to fill in the opponent’s role. If the ball is above the AI, it moves up, and if the ball is below the AI, it moves down. Basically, the AI tries its best to follow the ball given a specific speed. As the player scores higher, the AI gradually gets faster. But the AI will never go faster than the ball in order to make the match beatable.

1. Screenshots of Application

Here are the screenshots of my application:





H. Lessons Learned / Application

From here, I learned a couple of things. First is learning how to make video games. Second, and the most important one, is to learn how to be disciplined, to plan and execute tasks properly. I need a major improvement in time management and discipline. I learned the lesson that I should hustle for the assignments effectively from the beginning of the assignment. In short, I need to be disciplined for the next time.