Written Report

The first step in our project was to create the python window, using all the necessary codes to set the size, color, screen caption, and the pygame.QUIT event, used to close the pygame window. Then, we used pygame.draw.rect command to draw the rectangular tiles that represents the player 1 and 2’s racket. In the process of making the game, we found out that an AI(or CPU) was impossible for me and Miguel to make. Therefore, we chose to make the game multiplayer instead of single player against the CPU. Then we set both tiles to the middle of each side(player 1 to the left, player 2 to the right). We programmed the movements for each tile. Then we created the ball that could move; we defined a variable, center, as the starting point of the ball. When the ball bounces around and collides with the tile, we used the colliderect command; when the ball collides the tile, we used another code to let the ball bounce back(-x value). To make the game better, we added that when the ball collides with the tile, the ball travels 0.05 times faster than the original, but it returns to the original when a point is scored. After this, we added the scoring system. We programmed so that when the ball passes either baseline, it would count as a score. After each score, the ball will appear in front of the loser's tile and will be served automatically from this player’s tile; we made it an automatic serve instead of having the player to left click the mouse to serve. Then, we added the scoreboard by setting variables for both players and letting it to be displayed on the screen. The source of error for this section is that we were not able to make the score disappear after it is shown; so instead, we made the score to appear on the screen until the game ends. After creating the scoring system, we programmed to let the score stop at 7 and display the winner for 6 minutes. The game will then restart automatically after 6 seconds. To make the game better, we added different music that would be played randomly in different difficulties. Finally, we created a menu that would allow the player to choose the difficulty they want; each increase in difficulty represents an increase in the ball’s movement speed. Furthermore, we added more features to the game that would make it better, Firstly, when either player presses the M button on the keyboard, the game will restart. Secondly, the game includes a smash shot; when either tile’s top or bottom corner hits the ball, it acts like a smash in tennis and gets a very fast speed boost.

If we could do the project again, we would plan ahead of time to reduce the amount of sources of error. Otherwise, we would try to create an AI if we would have another chance to do this project again. Finally, we would make add more features to the game and make the game even better.