**Testing of Final Project**

Image 1

This image shows that two people played a game to the end and the O’s won. The first window (welcome screen) opened to the second one (name entry screen) and then onto the third one, which is the actual game. The users chose binary names, as this program accepts numbers as names as well. The game ended after O won.

Screens screenshot of a game

Description automatically generated

Image 2

This image shows that the X’s just beat the O’s. I had to learn how to add messages through the messagebox() feature in tkinter. Also, when I was initially building the game, my program could find that X’s were winners but not O’s. I used if statements to identify who was the winner (if box 1 == box 2 == box 3 == ‘X’, then X is the winner). My if statements had to check all the rows, all the columns, and the two diagonals. I realized that I could have rewritten all of those if statements to check if the O won, but I found a faster way to check. I learned that I could check both X’s and O’s if I used a box 1 in [‘X’, ‘O’] statement, which means I could write cleaner code.

Screens screenshot of a computer game

Description automatically generated

Image 3 – input validation

This image took me a while to figure out how to create. I saw on the rubric that we needed to put in fixes for input validation, including if the user left the screen blank. I didn’t know how to do that with a GUI program, but I figured out how to do it by creating a check\_for\_names() function that uses an if statement to learn if the user had entered any characters or numbers into the entry boxes. It also took me a while to figure out how to add the label below the button whenever the user left the entry fields blank.

Screens screenshot of a computer screen

Description automatically generated

Image 4 – Input validation continued

I included more input validation by adding a message box for when the user clicks on a square that has already been chose.

A screenshot of a computer

Description automatically generated