William Parente

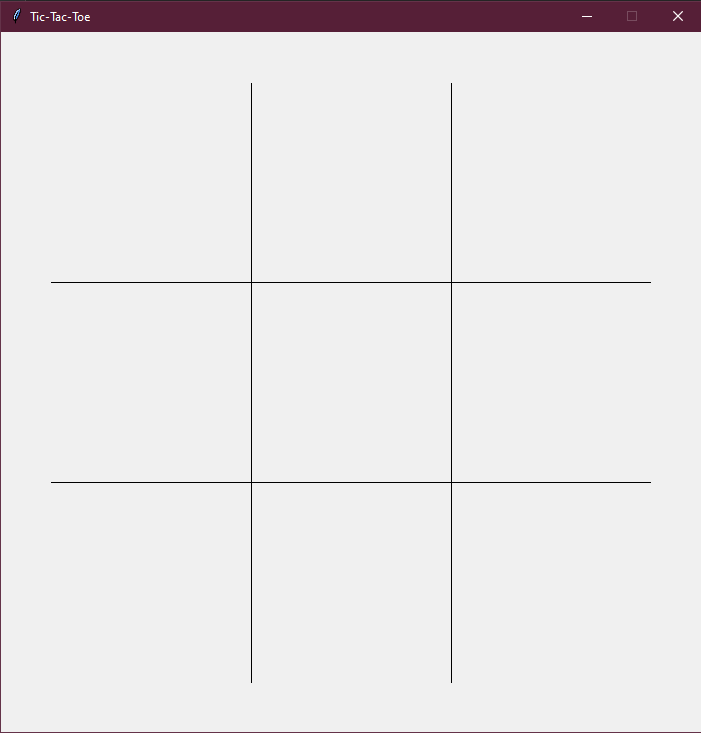
Professor Kevin Ryan

CS-110

14 December 2020

CS-110 Final Project Report

## Section One: Project Overview and Summary

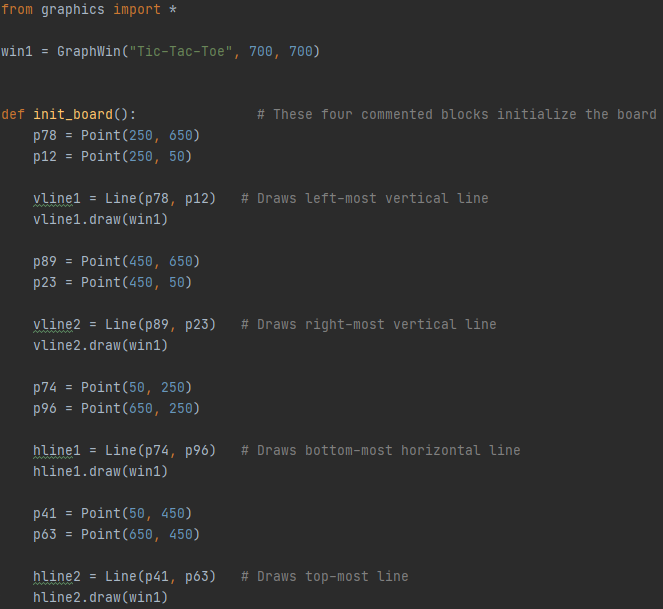
 The purpose of this program is to function as a two-player game of Tic-Tac-Toe. To achieve this, the program utilizes John Zelle’s graphics.py package to provide a functional user interface. In a game of Tic-Tac-Toe, the object is to achieve three in a row of a respective symbol horizontally, vertically, or diagonally. To play, the program creates a graphics window and generates a 3x3 grid via four line segments. Nine rectangles, one for each square of the grid, are generated but not displayed to function as buttons.

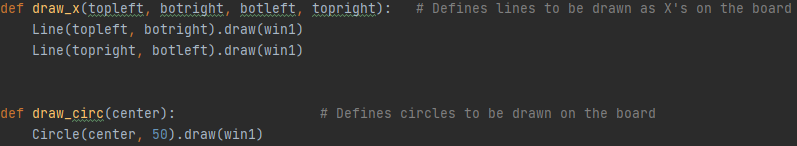
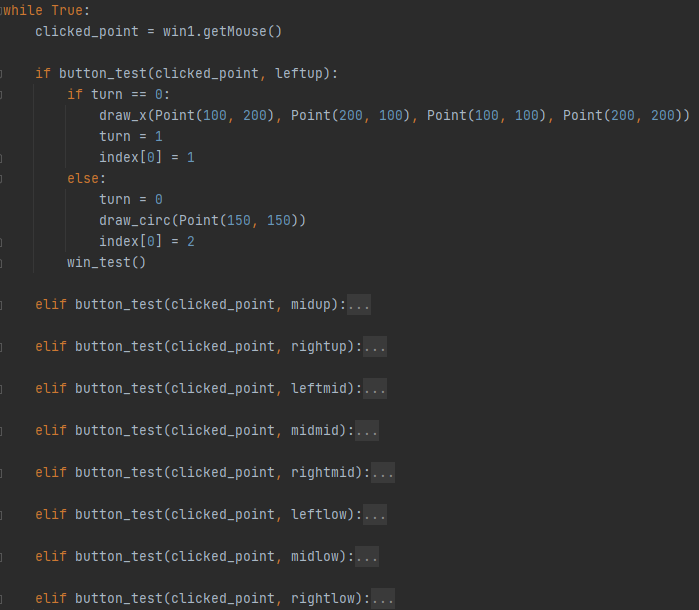
When the user clicks on any section of the grid, a uniform X or O is displayed in the center of that section, depending on which turn it is. Once there is three in a row of one of the symbols, a line segment is drawn connecting the three sections and a message is displayed in the center of the window indicating the match has been won.

## Section Two: Program’s Target Audience

Due to this program functioning simply as a game, it is applicable to anyone to simply enjoy. In a practical sense, the program could be used as a teaching tool. It has the capability of being used to teach young children basic geometric shapes, fundamental decision-making skills, as well as how to use a computer mouse to point and click.

## Section Three: Programming Techniques Employed

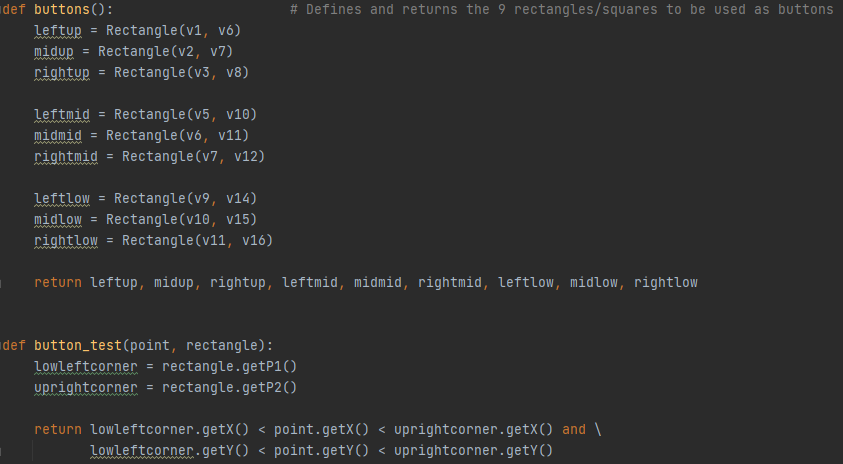
 The core of the program is the graphics.py package, a simplified version of python’s built-in tkinter library. The package allows for a usable graphics interface to be rendered and interacted with rather than utilizing complicated print statements to display the game state in the python output area. In terms of the graphics.py package, most of its capabilities are employed. Generating and displaying windows, points, lines, and shapes, as well as collecting data from user mouse inputs are all vital functions of the game.

To achieve the same dynamic as a game of Tic-Tac-Toe, various functions had to be defined, some of the most important being those which draw the symbols on the proper space on the board, test if a winning arrangement is present, and generate the window and grid in the first place. The only data this program uses is the x-y position of the mouse on the window at the time of a selection. After all the functions of the game are defined, the program enters a loop which allows it to place a new shape as well as check for a win upon every mouse click.

The loop contains nested if/elif/else statements which guide it to placing the correct symbol on the correct spot depending on the mouse input from the user. Additionally, the program uses variables and lists which are changed upon every mouse click as condition checkers for various functions; their uses vary from determining the shape to place to the text to display.

## Section Four: Challenges Encountered

The main challenge encountered during the creation of this program was in the nature of the GraphWin() function. When this function creates a window bounded by an x-y plane, it is not a standard x-y plane. Instead, it is flipped vertically. This results in the planed arrangement of y-coordinates being flipped at every point they were interacted with. This issue was more tedious than anything, as every user interaction with the top or bottom regions of grid would cause the program to interact with the opposite side. It was fixed relatively easily by simply inverting the y-coordinates of every Point(x, y) function within the program.

The other significant challenge was the manual creation of interactable buttons. The graphics.py package does not include a built-in function to create clickable buttons, and user interaction was essential for this program. Two functions had to be created, one which defines and returns nine separate rectangles in the play spaces, and one which tests of the clicked point is within the x and y coordinates of the bounds of each defined rectangle.

## Section Five: Future Extensions

Being that Tic-Tac-Toe is an old and well-defined game, there is not much room for gameplay expansion. However, there are two other areas which could be expanded upon. The first is simply aesthetics. Adding win counters, turn indicators, and reset buttons could all increase player enjoyment and visual appeal. On the aforementioned topic of education, much more could be done to equip the program for that task. Giving the user wider choice of shape/symbol to use during their game could help teach shape differentiation, and the same could be done for color. By implementing warnings of when an opponent has two-in-a-row, it could reinforce decision making skills as well as socially competitive behavior.