

## Lab: Tic Tac Toe with Graphics

In the game of tic tac toe, players alternate placing Xs and Os on a board until one player gets three of their symbol in a row, column, or diagonal. We will store the tic tac toe board as a 2D list of *integers*, where a zero means a space on the board is empty, a 1 means it is filled with an X, and a -1 means it is filled with a O. This will make it easier later on to detect wins.

Fill in the code for tic-tac-toe in this order, assuming a board is stored as a 3 by 3 grid of numbers.

1. Find the `draw_board` function. This function serves a similar purpose as the `print_board` function, in that they both are in charge of displaying the tic-tac-toe board in a format that the players can read. The difference is that `print_board` displays the board using text, and `draw_board` uses simplegraphics to draw the board as a picture.

Fill in the `draw_board` code. The code should draw a tic-tac-toe board using X's and O's. Remember that the board has 0/1/-1 in each square, so use nested loops & if statements to print the correct pieces. Use `draw_line` and `draw_circle`.

2. Find in the `main()` function where it says to wait for a mouse click. We will put some code here to let the user click the square in the board in which they want to move, rather than typing it in from the keyboard. Follow these steps:
  - a. Comment out the row/col input statements.
  - b. Call the `wait_for_click()` function [see simplegraphics reference].
  - c. Call `get_last_click_x()` and `get_last_click_y()` to capture the x/y coordinates of the mouse click.
  - d. Write code to convert the x/y coordinates into row/col coordinates. Remember, x and y will be between 0 and 300, while row and col need to be between 0 and 2, so what mathematical operation will do the conversion?
3. Find in the `main()` function where it says to use input validation. Write this code so someone can't move on top of someone else's piece. The user should be re-prompted as many times as possible to choose a different square until they choose an open one.
4. Use simplegraphics to improve the look of the game or add other enhancements. Here are some suggestions:
  - a. Change the line thickness so the pieces are thicker, but the lines of the board itself remain thin (or vice-versa).
  - b. Make the X and O pieces slightly smaller so they don't touch the sides of their squares.
  - c. Change colors so the X and O pieces are differing colors from each other.
  - d. Add artificial intelligence so the user can play against the computer. The computer should not just play randomly, but play like a human would.
  - e. Let the user play multiple games. Keep track of how often each player wins.
  - f. Let the user pick the board size.