Homework #5 **Due: 11/8**

1. [100 points] This homework will test your abilities to implement an algorithm in Python and to build a user interface for it using `Tkinter`.

   Use `Tkinter` to build `tictactoe`, an script that plays an interactive Tic-Tac-Toe game of you against the computer. Tic-tac-toe with humans dates back to Ancient Egypt. (Thanks, Wikipedia!)

   The template for `tictactoe` (a script) is:

```
#! /usr/bin/env python3

from tkinter import Tk

# Insert code for TicTacToeBoard or import it from a separate module.
# Your choice.

root = Tk()
root.title("Tic␣Tac␣Toe")
board = TicTacToeBoard(root)
board.grid()

root.mainloop()
```

   You should use this. It will be available from the course web page as `tictactoe_tplt` (rename it to `tictactoe` when you add your code).

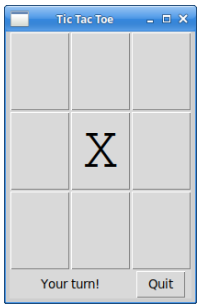
   The `TTicTacToeBoard` class (which you are to write) is a child class of `tkinter.Frame` that is made up of 3×3 "cells". (You'll have to pick the best `Tkinter` widget to represent a cell here.)

   As you can see, the computer always goes first and selects "X"s, while the player selects "O"s. This means that the computer cannot lose. (It's not sporting, but it makes the algorithm below much simpler!)

Requirements

These are the requirements:

(a) The window initially resembles this:



small differences are not penalized.

(b) The window is titled "Tic Tac Toe".

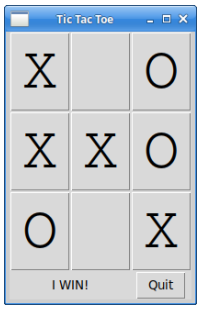
(c) There is a `TicTacToeBoard` class as specified above.

(d) The player clicks with the left mouse button on any blank cell to convert it to an "O" and the computer automatically selects the next cell according to the algorithm.

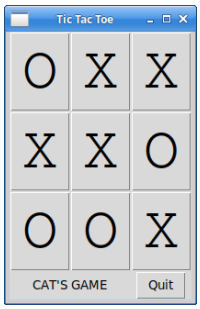
(e) The fonts for the "X"s and "O"s are bigger than other text. (A Courier font is suggested, but not required.)

(f) The message on the lower left tells game status. It is either the player's turn, the computer has won, or it's a "cat's game" (a tie).

(g) A computer win looks like this:



(h) A cat's game looks like this:



(i) After the game is over (win or a cat's game), the board does not change.

(j) The "Quit" button on the lower right ends the game and exits the script.

The Tic-Tac-Toe Algorithm

The algorithm to play an interactive game of tic-tac-toe against the computer looks like this (in pseudocode):

```
set the computer move "c" to the center cell
let m = 1 (the move counter)
while m < 9:
    if m == 1 (on the first move)
        the player selects any blank cell "p"
        set the computer move "c" to the next cell clockwise from "p"
    else
        the player selects any blank cell "p"
        let "o" be the cell opposite "c"
        if "o" is not "p"
             set the computer move to "o"
             computer wins! (game ends)
        else
             set the computer move "c" to the next cell clockwise from "o"
    m = m + 2
it's a cat's game (game ends)
```

The challenge here is to integrate this into the `Board` class, ideally in the event handler for the left mouse button. Your code will probably not look like the algorithm, but the game play should be identical.