Tic-Tac-Toe and Functions
Lab 4

Prepared by Matt Bennett
for ELEC 3150
Professor Carpenter - Fall 2016

I - Introduction

The goal of this lab was to place the best move in a tic-tac-toe game board given a board in a three by three character array. This lab required both algorithmic thinking to determine the best move of a given board and knowledge of pass by reference C++ functions.

II - Procedure and Assumptions

The best X placement algorithm is based on five different basic tic-tac-toe board conditions. The case of an empty board, the case of a board with a possible winning placement, the case of a board without a winning placement where the opponents winning placement must be blocked, the case of available corner positions, and the case of only ‘edge’ (non-corner or center) positions. Given an empty board, X is placed in the center of the board to maximize the chances of winning. In the cases of a winning board or a non-winning board where the opponent must be blocked, the respective placements of winning and preventing a loss are the best placements and so these positions are chosen.

In the remaining cases where there is neither a winning move or need to block, corner positions are prioritized because each corner position is required for three different winning conditions while the edges are required for two different winning positions, making those positions generally less useful for winning the game.

The algorithm assumes that the input array is a valid board, not a board of a complete game, since a complete game does not have a possible best resulting board.

III - Results

Note: The following figures display the tic-tac-toe board prior to placement on the left and following placement on the right.

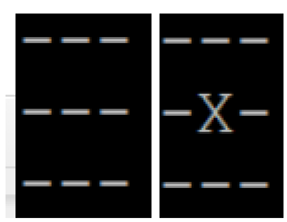

Figure 1: Verification of empty board center placement behavior

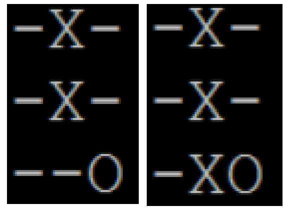

Figure 2: Verification of winning placement in column

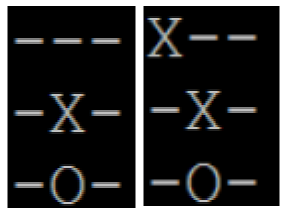

Figure 3: Verification of general placement

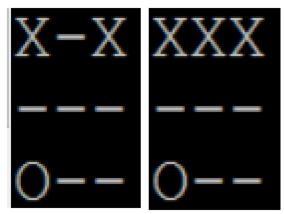

Figure 4: Verification of winning placement in row

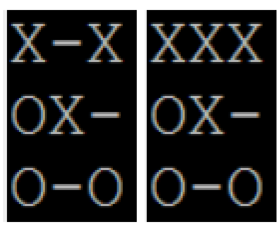

Figure 5: Verification of win priority over loss prevention

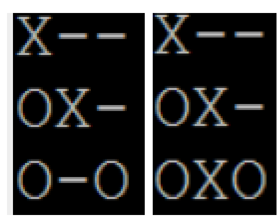

Figure 6: Verification of loss prevention

IV - Analysis

The X placement functioned as expected, placing winning moves when possible, blocking losses when wins were not possible, placing in center given empty board, and prioritizing corner positions over edge positions. These cases and their prioritization were each tested in Figures 1 through 6 and the resulting boards matched the desired program functionality.

V - Conclusions

The algorithm functions as expected, placing winning moves, blocking the opponent, and the case of an empty board. The algorithm could be further improved in the future to better respond to O placement on the boards edge and guarantee a win in these cases.