Project 2: Intelligent Tic-Tac-Toe on a General Board

Tic-tac-toe can be played on a general nrRows x nrCols board where a player wins by having nrToWin of its elements consecutively in a row, column, diagonal, or anti-diagonal. The traditional tic-tac-toe corresponds to the case where nrRows = nrCols = nrToWin = 3.

In this project, you will implement an evaluation function and minimax search with alpha-beta pruning for the general tic-tac-toe game.

The evaluation function should evaluate the board configuration from the perspective of the FIRST player, i.e., return a high positive value if the FIRST player has a better position and a low negative value if the FIRST player has a worse position. The range of values should be [-MAX_SCORE, MAX_SCORE], where MAX_SCORE is a constant defined in the support code.

The support code can be compiled similar to the support code for Project1. The name of the executable is TTT. The program can be run with

./bin/TTT nrRows nrCols nrToWin depth

from Linux/Mac and

bin\Release\TTT nrRows nrCols nrToWin depth

from Windows, where nrRows is an integer corresponding to the number of rows, nrCols corresponds to the number of columns, nrToWin corresponds to the number of consecutive elements needed for a win, and depth is an integer corresponding to the maximum depth search for the minimax search (with alpha-beta pruning).

From the graphical interface, you can use the mouse to click at a position where you would like to place the piece. Pressing the letter s on keyboard invokes the BestMove function, which uses the minimax search (with alpha-beta pruning and your evaluation function) to print out the best move for the current player. For example, starting the game with TTT 3 3 3 100 and pressing s would generate the following:

bestScore = -0.000000 bestMove = 0 0 [player = BLUE] [nrMoves = 0]

indicating that the best move for the first player (corresponding to BLUE squares) is in position (0, 0). You can then click on position (0, 0), and a BLUE square will be placed there. Pressing s again would generate

bestScore = 0.000000 bestMove = 1 1 [player = RED] [nrMoves = 1]

indicating that the best move for the second player (corresponding to RED squares) is in position (1, 1). Note that any other move would cause a loss for the second player.

How to test the quality of your evaluation function

Play on larger boards with different values for nrToWin. The program should be able to beat you most of the time. If you are not very good with Tic-tac-toe, then try to find a friend who is good and see if your program can beat your friend. It should.

Support code

You only need to look at src/Programs/TTT.hpp and src/Programs/TTT.cpp. Your implementation should be in src/Programs/TTT.cpp. See comments in these files for more information. The files in Utils directory are similar to Project1 – they provide some general functionality. The file src/Programs/RunTTT.cpp is responsible for the graphical interface of running the tic-tac-toe game.