

## 1 Generalized Tic-tac-toe

Generalized tic-tac-toe, (or  $n$ -in-a-row) is a game where the players alternate placing pieces (X's and O's) on a  $m \times m$  board. The game is won when a player gets  $n$  pieces in a row. If neither player achieves this before the board is full the result is a draw.

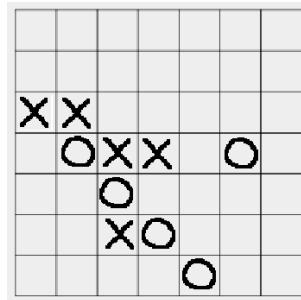


Figure 1: 4-in-a-row on a 7 x 7 board won by 'O'.

## 2 The Assignment

Working in groups of up to two, students are to implement a generalized tic-tac-toe player. The player will take the board size ( $m$ ) and the numbers of pieces to be connected in a row ( $n$ ) as input parameters as well as the cut-of-depth for the game tree. The player should be able to move first (be 'X') or move second (be 'O').

We recommend that you implement game-tree search with alpha-beta pruning but other approaches are welcome as long as they produce an acceptable result (i.e they can beat Mike or Johan for some reasonable settings of  $m$  and  $n$ ). You may use any programming language of your choice.

## 3 Extra Credit 33pts (optional)

- Make the game interruptible, that is, make it possible to at any given point halt the computer evaluation and force it to play (hint: iterative deepening...). We shall hold a tournament at the end of the course to determine the smartest computer-player.

## 4 Report

You should hand in a complete and well written report in the box marked 5DV019. Be sure to document your board evaluation function. Take special note to show (via a log file) that the program beats a reasonable human player on a 4-in-a-row on a 7 x 7 board. Also provide a link to your source code, residing in your edu folder, along with runnable files (compiled binaries or scripts) that can be easily run on the departments Linux systems.