GamesCrafters

Tic Tac Toe

Dan Garcia (Dev)

Documentation for Developers

Fall 2007: Michael So, Moonway Lin

Date	Version	Notes
2007.10.14	1.0.0	First version created
2007.11.26	1.0.1	First revision
2007.12.4	1.0.2	Second revision

Table of Contents

Gamescrafters Tic Tac Toe Documentation

Game Overview

Tic-tac-toe is an ancient two-player game played on a three-by-three board. One player has pieces that are X's and the other has pieces that are O's. Each player takes turn putting their pieces on the board. The player's goal is to get three pieces in a row horizontally, vertically, or diagonally.

It is non-partisan, non-loopy, and ties are possible

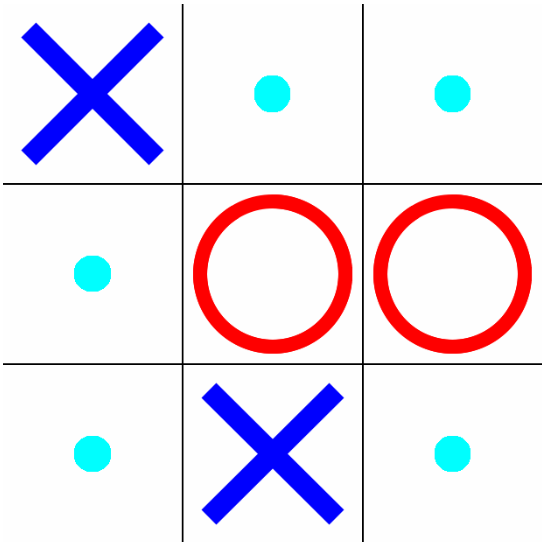

Figure 1 The Tic Tac Toe Game Board.

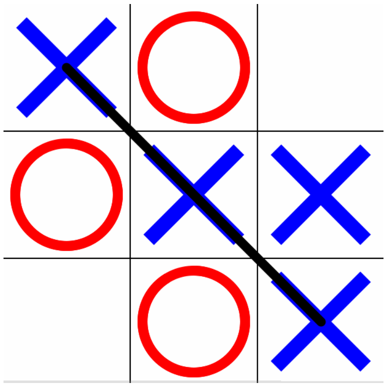

Figure 2 A sample winning configuration for X (3 X's in a row diagonally)

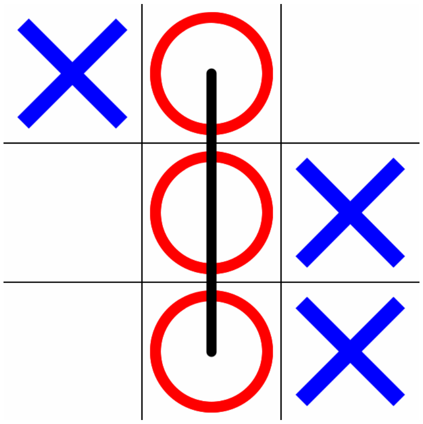

Figure . A sample losing configuration for X (3 O's in a column)

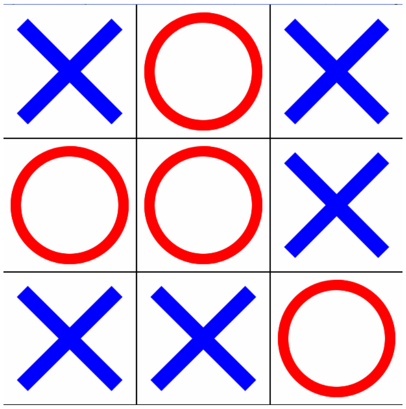

Figure 4 A sample tie configuration. All nine spaces must be occupied in a tie. Note that neither X nor O has three in a row.

Design Overview

We simplify game boards that are effectively identical to greatly reduce the size of the problem: the total number of possible games is on the order of 10^5, yet after simplifying the boards, the total number of possible unique games is on the order of 10^2. This makes game analysis much simpler.

There are two kinds of symmetries: flips and rotations. There is one kind of flip: a horizontal flip. An example of a horizontal flip would be

```
0 1 2 -> 2 1 0
3 4 5 -> 5 4 3
6 7 8 -> 8 7 6
```

In other words, the entire game board is "flipped" on its horizontal (east-west) or vertical (north-south) axes.

Rotations are equivalences that involve rotating the board by 90 degrees.
For example,
0 1 2 is equivalent to 6 3 0
3 4 5 is equivalent to 7 4 1
6 7 8 is equivalent to 8 5 2

We implement both types of symmetries using three arrays:
qSymmetryMatrix[NUMSYMMETRIES][BOARDSIZE]
gFlipNewPosition[] = {2,1,0,5,4,3,8,7,6};
gRotate90CWNewPosition[] = {6,3,9,7,4,1,8,5,2};

In the initialization function initializeGame(), we convert all possible boards to their "canonical positions" (basic positions, i.e. have the smallest board representation number) through a series of flips and rotations until the boards are in their canonical positions.

In the helper function GetCanonicalPosition(), we go through all of the positions that are symmetrically equivalent and return the smallest one.

The other helper function, DoSymmetry(), performs the symmetry operation specified by the input (flip or rotate) and returns the new position.

Data Structures

The only data structure is the board, which is arranged as follows

```
Tile 0, Tile 1, Tile 2
Tile 3, Tile 4, Tile 5
Tile 6, Tile 7, Tile 8
```

OX#s: Empty = 0, O = 1, X = 2

The board's position is stored as an integer as follows: SUM from i = 0 to 8: 3^i * OX#[i]

Example:

Figure 1’s board position = 3^0*2 + 3^1*0 + 3^2*0 + 3^3*0 + 3^4*1 + 3^5*1 +
3^6*0 + 3^7*2 + 3^8*0 = 4700

The main struct, gPosition, consists of the board, the next player to move (O or

X), and the number of pieces placed.

C Game Checklist

- ✓ All help strings are done.
- ✓ Debug turned off
- ✓ There are no variants.
- ✓ No memory leaks
- ✓ PrintPosition done
- ✓ getoption, setoption, and option stuff is done
- ✓ Placed in Makefile
- ✓ Clean/commented code
- ✓ Implementation of MoveToString() is done
- ✓ Calls the common functions from the Game function libraries
    - GenerateMoves calls library function CreateMovelistNode().
    - DoMove calls library function whosTurn().
    - PositionToBlankOX calls library function BadElse().
- ✓ symmetries is used. *See Design Overview regarding symmetry.*
- ✓ Uses GPS

Complexity Analysis

Because each of the nine squares has three possibilities: empty, O, and X, there are a total of 3^9 = 19,683 positions.

Of these positions, 18917 are undecided positions (not all tiles have been occupied), 224 are losing positions (by either player), 390 are winning positions, and 151 are tie positions. The rest are impossible positions.

Without eliminating board rotations and reflections, there are 255168 possible games (sequences of moves). Of these, 131184 games can be won by the player going first, 77904 by the player going second, and 46080 games ending in a draw.

After eliminating rotations and reflections, there are only 138 unique games, among which 91 can be won by the first player to go, 44 by the second to go, and 3 possible draws.

Tic-Tac-Toe, due to its simplistic nature and small game tree, has been strongly solved.

Optimal Strategy

Gamescrafters Tic Tac Toe Documentation

The optimal strategy for playing Tic-Tac-Toe involves a sequence of moves with descending priority:

1. (Highest) If there is a tile that allows you to get three in a row, place your piece on that tile.
2. If there is a tile that, if left unoccupied, your opponent can use to get three in a row, place your piece on that tile.
3. If there is a tile that allows you to "fork," i.e. win with two possible three-in-a-rows, place your piece there.
4. If your opponent has set up a fork, block that fork.
5. Occupy the center square.
6. Occupy a side square
7. (Lowest) Occupy a corner square.

(Note: This sequence is a "perfect" strategy and ensures victory or a tie. If both players play under this optimal strategy, the game will always tie.)