

GamesCrafters

Connect 4

Michael Thon(Dev)

Documentation for Developers

Fall 2007: Michael So

Date	Version	Notes
2007.10.14	1.0.0	First version created
2007.12.4	1.0.1	First revision

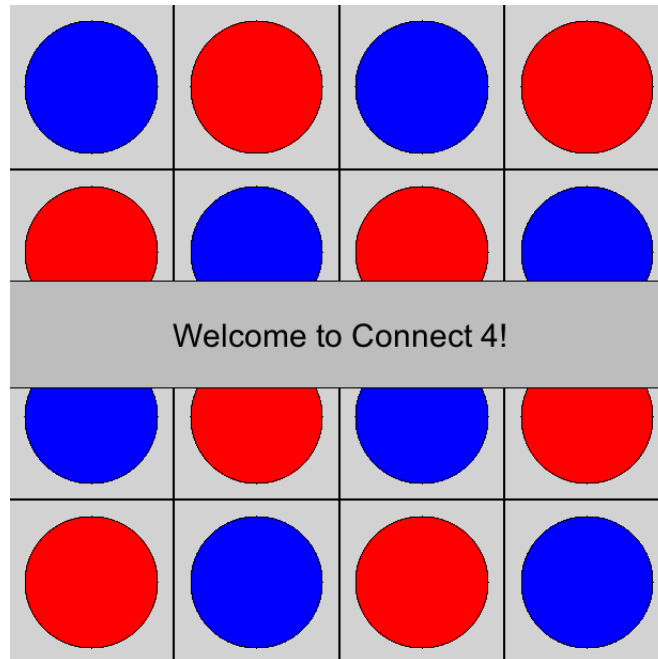
Table of Contents

<u>Game overview</u>	3
<u>Design overview</u>	3
<u>Data structures</u>	4
<u>C Game Checklist</u>	4
<u>Complexity Analysis</u>	5
<u>Optimal Strategy</u>	5

Gamescrafters Connect 4 Documentation

Game Overview

Connect 4 is a two-player board game where the players take turns in dropping their own colored pieces into a 4x4 grid. The player's goal is to be the first to connect 4 of their pieces in a row either vertically, horizontally, or diagonally.



The game is partisan, non-loopy, and a tie is possible.

Design Overview

The board uses tierification. *See document for Tierification.*

To generate moves, check for each column if the highest space is occupied. If it is not, generate a move list for that column. If it is occupied, no more possible moves can be made in that column.

When initialized, all the spaces in the board are in an empty state (also known as Blank).

Data Structures

Gamescrafters Connect 4 Documentation

The board is defined as a 2-D array.

The possible board pieces are encoded as:

```
typedef enum possibleBoardPieces {  
    x, o, Blank  
} XOBlank;
```

Positions are made with bit strings. For example, consider a 5X4 board:
10100 10010 11010 01110

Each segment is a column going up. 1 means "O", 0 means "X" unless either is followed by all 0's, in which case the bit is the beginning of "on pieces" part of the column.

When converting a position to binary, it is checked if the position is divisible by 2, then that space on the board is '0' (zero). Otherwise, that space on the board is '1' (one). The position is halved and this process is repeated for all the spaces on the board. *Refer to the positionToBinary() function in mwin4.c for more detail.*

C Game Checklist

- ✓ All help strings are done.
- ✓ Debug turned off
- ✓ There are no variants.
- ✓ No memory leaks
- ✓ printPosition done
- ✓ getopt, 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().
 - Primitive calls library function PositionToBoard().
 - GetAndPrintPlayersMove calls library function ValidMove().
- ✓ symmetries is not used.
- ✓ Uses GPS

Complexity Analysis

Gamescrafters Connect 4 Documentation

Possible board pieces are X's, O's, and empty. If a column contains an empty space, then all spaces above it also contain an empty. If the total number of occupied spaces is odd, the number of X's is one more than the number of O's. If the total number of occupied spaces is even, the number of X's equals the number of O's. If a position contains four connected X's, or four connected O's, then the game ends. Since that position contains four connected pieces, at least one of the four spaces in the connected group must be in the highest filled square in its column. If one player has more than one connected group, this position is only legal if these groups share a space, which contains the player's piece.

Optimal Strategy

Placing pieces in the middle columns are more valuable than those placed on the sides because players have more chances of creating four in a row with them.

For all boards, which consist of at most 6 columns and an even number of rows, the following strategy will ensure the player going second either a win or a tie.

For columns 1, 2, 5 and 6 the following rule is used:

(1) The second player places a piece in this column if and only if the first player has just played in this column.

For columns 3 and 4 the following rule is used:

(2) If the first player plays for the first time in one of these columns, the second player places a piece in the other column. Otherwise if the second player can place a piece in the column the first player just placed a piece in, the second player does so; otherwise the second player places a piece in the other column.

The strategy can work for a 4X4 board. (1) applies to columns 1 and 4 and (2) applies to columns 2 and 3.