3D Tic Tac Toe

Jessie Wang (yw696) , Chun Yang (cky23)

Description

A tradition Tic-Tac-Toe is a board game played on a 3 by 3 grid by 2 players. The players take turn marking the grid with (O) and (X). The first player to reach three marks in horizontal, vertical, or diagonal is the winner. The game is not very complex and often ends in a draw. Therefore, it is not an interesting subject of an artificial intelligent system implementation.

In comparison, a 3D tic-tac-toe game, also known as Qubic, is played on a cubical array of cells, usually 4 by 4 by 4. The result is that there are 76 possible winning cases.This significantly increases the complexity of the game.

Our project is the implementation of an artificial intelligent system that plays 3D tic-tac-toe. The implementation utilizes minimax algorithm to pick out the best moves in the worst case scenario. It also utilizes alpha-beta pruning to make the implementation more efficient.

Approach

We will use minmax algorithm with alpha-beta pruning to evaluate a best move. When it’s our turn to play, we will perform adversarial search to choose the one move that generates the max payoff, which is evaluated based on evaluator E. E is described as follows. For a move that creates 1 in a row, we simply add 1 to the existing E result. If a move creates 2 in a row, then we would add 76 to the current result. (Since we have 76 different ways to win 3D tic tac toe, 10 * 4 for winning with a single level tic tac toe, 16 for winning with a vertical column, 10 * 2 for winning with 3D diagonal, we want to make sure that getting 2 in a row is better than getting lots of 1 in a row. Using 76 for 2 in a row guarantees when we get 2 in a row, the result is better than getting lots of unconnected pieces since the total number of cells is only 64). After picking the best move, we will then run the check function to see if we already reached some goal state. (One of the 76 winning conditions.) If so then we simply stop the game since either we or the opponent wins the game; otherwise keep iterating until the entire board is filled then we have a draw.

Plan

To evaluate the move of your 3D tic tac toe system, we will evaluate on both the result, if our player wins the game, and the number of moves. If our player wins the game, then it’s a decent indication of the success of minmax algorithm. If the number of moves is relatively small, then it shows alpha-beta pruning eliminated reasonable number of branches. We will further evaluate the system on average running time and average branching factors.

We will also collect some statistical data including the percentage of wins of playing first versus playing second.

Timeline

11/1/2017 first implementation of the algorithm described above.
11/7/2017 finish collecting data (winning percentage and runtime efficiency); introspection and plan of improvement
11/15/2017 second implementation based on data collected
11/24/2017 second round of data collection
11/30 further improvement of implementation if needed; UI improvements