MiniMax and Alpha-Beta Search on the Game Chomp

David Chavez

4/11/2017

The minimax and alpha-beta search algorithms are commonly used for AI. They help find the best possible outcomes for a given scenario. I decided to use this algorithm to help me with a game I recently published on the app store called "Choco Chomp" (https://itunes.apple.com/us/app/choco-chomp/id1213722025?ls=1&mt=8). It is based on similarly named game "chomp". The impartial game "chomp" is a two-player game with perfect information which means that both players can make the same moves and no information is hidden from any player. For example, chess is not an impartial game since player 2 cannot move player 1's pieces. A game without perfect information would be poker since player 1 cannot see player 2's cards.

The game chomp is played on a rectangular board with n rows and m columns. Where m, n > 1. The bottom left piece is poison. For example, consider the following board.

Initial Board

[ ][ ][ ][ ]
[ ][ ][ ][ ]
[p][ ][ ][ ]

Each player takes turns making moves until only the poison piece is remaining. A move consist of selecting any square and removing all squares above it and to the right of it. Selecting the poison square is a loss for the respective player.

First Players Turn

[ ][ ][ ][ ]
[ ][ ][1][ ]
[p][ ][ ][ ]

Second Players Turn

[ ][ ]
[ ][ ]
[p][2][ ][ ]

First Players Turn

[ ]
[1]
[p]

Second Players Turn

[p]

Second player would lose in this case.

It can be shown that for any rectangular board of chomp first player has a winning strategy, however, finding the winning strategy can be very difficult for large games of chomp. My plan is to use the minimax algorithm to add a single player mode to my game on the app store. I will be using the Swift language since that is the language for IOS development. Since the tree diagram gets exponentially bigger the larger the game board, I will also try to do use other similar algorithms like alpha-beta search to improve the search algorithm.