Spider Line 4

University of Porto

Catarina Rocha
Faculty of Engineering

Karim Kousa
Faculty of Engineering
University of Porto

***Abstract*— In this project, a game of Spider Line 4 was built. This variety of Connect 4 requires a totally different strategy to win since it offers more prospects to put pieces. The base column as well as any remaining three edges of the board "draw in" the pieces which can be, insect like, stuck to the dividers and the roof. The accompanying picture shows an illustration of a began game position and the green crosses mark squares where the player can put another piece.**

Keywords—Artificial Intelligence, Minimax

I. Introduction (*Heading 1*)

II. Specifications

Spider Line 4 has the same logic as the famous Connect 4 game, but with a little twist where a player can insert the pieces from any vertical or horizontal side of the board, this gives the player more moves since the game transformed from one dimension to two dimensions. A position becomes a possibility for the player to place his/her piece if that place is connected directly to another piece regardless of whom the piece belongs to and there is a trail of pieces to the edge of the board. And the goal is the same as Connect 4 where the player has to connect four pieces either vertically, horizontally or diagonally. The game has three modes in which a user can choose from. The first one is Player vs Player, where two users can compete against each other. The second is Player vs Computer, where the user will try to defeat a well-established algorithm. The third is Computer vs Computer, which allows us to try different levels of difficulty by choosing how deep to look into future moves.

III. Formulation of the Problem

A. State Representation

We built our game on the basis of haveing a 5 by 5 grid, so the general case is having a board with dimension N by M, filled with values from 0 to 2. Where 0 represents na empty place, 1 and 2 represents player 1 or 2.

B. Initial State

An empty board with zeros in all the places which will be changed to the numbe rof the player who puts a piece instead of the zero.

C. Objective Test

Obtaining a line of four pieces stacked vertically, horizontally or diagonally.

D. Operators

- Name: **Move**(col,row,player)
- PreConditions: **Board**[col,row]==0, **possible_pos** (col, row)
- Effects: **Board**[col,row]= player

E. Huristics/Evaluation function

Agent 1: EvalF1 = nlines4(1) - nlines4(2)
Agent 2: EvalF2 = 100 * EvalF1 + nlines3(1) - nlines3(2)
Agent 3: EvalF3 = 100 * EvalF1 + central(1) -central(2)
Agent 4: EvalF4 = 5* EvalF2 + EvalF3

nines4(x): How many lines does player x has that have the potential to be a line of four pieces.
nines3(x): How many lines does player x has with three pieces forming a line.
central(x): How many pieces does player x has in the center of the board.

IV. Implementation Details

All the places are defined with two numbers representing the column and the row in the board. The move move will try to place a piece in the coordinates passed to it by calling a function that checks if the same coordinates are a possibility to the player or computer using the edges, if the coordinates are on the edge of the board and they were not choosen before then they are always a possibility. For places other than the edges we check the four directions (up, down, left, right).

V. Approach

VI. Algorithms

The main algorithm used in Minimax algorithm with alpha and beta cuts variant. The worth determined for each line/section/corner to corner has a positive or negative sign contingent upon which hued chips are available sequentially in that line or segment.

The heuristic capacity gives an extremely high value (like INT_MAX) to 4 chips continuously in succession/section/inclining. 'Max' and 'Min' can be viewed as the two players of the game(considering two-player games just) . 'Max' will attempt to augment the heuristic capacity at each Max move and 'Min' will attempt to limit the heuristic at each Min move. MinMax (otherwise called minimax) calculation is a calculation that can be utilized to track down the most ideal move/activity for each player(Min and Max). Alpha-Beta pruning is an approach to staying away from subtrees of searches which will not be chosen.

VII. Results

Player 1 / Player 2 / Draw (Average Time)

Depth = 1	Agent1	Agent2	Agent3	Ag
Agent1	0/10/0 (0.1026)	0/10/0 (0.1110)	10/0/0 (0.0608)	10/0/0 (0.0694)
Agent2	0/10/0 (0.1285)	0/10/0 (0.1329)	10/0/0 (0.0700)	10/0/0 (0.0824)
Agent3	10/0/0 (0.0629)	10/0/0 (0.0744)	0/10/0 (0.1234)	0/10/0 (0.1288)
Agent4	10/0/0 (0.0844)	10/0/0 (0.0970)	10/0/0 (0.1417)	0/10/0 (0.0142)

Depth = 4	Agent1	Agent2	Agent3	Agent4
Agent1	0/10/0 (3.7446)	0/10/0 (5.7793)	0/0/10 (18.9145)	0/10/0
Agent2	10/0/0 (7.4682)	0/10/0 (9.3959)	10/0/0 (10.7964)	0/0/10
Agent3	10/0/0 (18.0854)	0/10/0 (23.4742)	0/10/0 (63.9549)	0/10/0
Agent4	10/0/0 (40.9636)	10/0/0 (44.4729)	0/10/0 (71.5192)	0/10/0

References