

PLAYING CONNECT-6 GAME USING α - β SEARCH

THE GAME :

In this assignment, you will implement the playing of a variation of connect-6 game using the α - β pruning techniques.



Connect-6 is two-person game. The rules of the game is as indicated below.

(Begin quote from Wikepedia site)

The rules of Connect6 are very simple

- Players and stones: There are two players named Black and White. Black plays first and White second. Each player plays with a collection of appropriate color of stones.
- Game board: Connect6 is played on a square board made up of orthogonal lines, with each intersection capable of holding one stone. In theory, the game board can be any finite size from 1×1 up (integers only), or it could be of infinite size. However, boards that are too small may lack strategy (boards smaller than 6×6 are automatic draws), and extremely large or infinite boards are of little practical use. 19×19 Go boards might be the most convenient.
- Game moves: Black plays first, putting one black stone on one intersection. Subsequently, White and Black take turns, placing two stones on two different unoccupied spaces each turn.
- Winner: The player who is the first to get six or more stones in a row (horizontally, vertically, or diagonally) wins

(End quote from Wikepedia site, the site also points to some research groups that are trying to find algorithms that can play this game. The game is not yet solved... which means it is a very difficult game...)

REQUIREMENTS AND ASSUMPTIONS:

We will do the following variations to this game.

Variation 1) We will vary the board size so that game ends in reasonable time. We will use odd numbers for the board size. We will try on board sizes 7, 9, 11, 13 etc.

Variation 2) We will allow placement of stones only in such a way that it is adjacent to a stone already placed (adjacent may mean left, right, up, down, or immediate diagonal position). This will reduce the number of moves possible in the initial stages.

You need to write a program that can be either Black player or White player to play the connect-6 game. You should develop a utility function for each move that can be made and then use Min-Max search algorithm with α - β pruning to decide which move to make. The depth of the Min-Max tree should be a constant so that it can be changed.

INTERNAL REQUIREMENTS

All teams will implement the same connect6Driver that is given. You may change any other classes as you wish. The only things you must use or expect from the opponent would be BlackPlayer(WhitePlayer) class with constructor as given and getMove() method and update() method. Note that your code should be written in such a way that it will compile when the class for BlackPlayer or WhitePlayer is supplied by another team. So you must make sure your BlackPlayer and WhitePlayer classes have the specified constructor and methods.

The grading will be based on design of the code, user interface for the game play and game play itself. 10% of the grade will be assigned for number of wins or losses suffered. If an illegal move is made the team that made the illegal move is the loser. If the program crashes when the opponent makes a legal move, the team whose interface is used is considered to have lost. We will play 5 games using each interface and decide the points on the basis of number of wins and losses. Each team will play as Blackplayer same number of times.