Checkers AI
Report

Description:

Goal:

Search through game space and choose most suitable move at each turn in checkers. With a hopeful outcome of the program playing checkers well.

How:

Find all possible moves. Find possible moves after these possible moves and link back to original move. Do this for a few steps (9).
A score is assigned to each possible move, based on favourability of the board to the AI player.

Scores are chained back from final move in branch of moves in such a way to quantify a more accurate score of each original possible move.

A branch's score is chained back by finding the max score if it is AI's turn in that branch, or finding the min score if it is the human's turn in that branch.Then assign this score to the root of that branch. Keep doing this until the all original possible moves scores are updated.

IncreaseDepth:

Goal:

Add extra depth of moves to last moves in branchMoves.
Calculate score for this extra depth.
Chain score back through moves in branchMoves.

MakeAIMove:

Goal:

If first move: Get first possible moves and use increaseDepth to create a branch of possible board connected by moves. So branchMoves is size depth if possible by time constraints.
IncreaseDepth will update scores of first possible moves.
Make move of highest score of first possible moves.

If not first move: Still have branchMoves data from previous move.
Find the board that represents the current bored in branchMoves
Redefine this board as root of branchMoves.
IncreaseDepth of branchMoves to depth if possible by time constraints.
Make move of highest score of first possible moves.

Results:

Shows rational moves.
Is capable of ‘planning’ moves a few moves ahead, as expected.

Is too slow to search large enough ahead to always beat me or other AIs online.
At depth of 9 moves uses about 2gb of ram.