

Practical 2: Torus Checkers

Id: 150028460

Description:

Torus Checkers is identical to normal draughts with the notable rule changes being the continuance of the board on the side and end boundaries where they wrap around in place. The game itself is implemented using vectors containing the positional references given from the standard input along with a basic counter to check whose turn it was. Calculating possible moves for each side was a series of 4 different cases for movement, and 6 cases for attempting to take a piece. When attempting to perform a simple move the cases were those where were offset from the side as opposed from those that were not, and those which would cross the boundary on the left or right. I used modular arithmetic to attempt to avoid laboriously describing every pieces rules. When attempting to take a piece the case structure expands to accept those that might cross the torus boundaries on the take. To account for moves and takes that might overcede the end edge of the board the pieces are wrapped with more modular arithmetic, specifically the size of board. Captures and moves themselves are made simply by adding or subtracting a constant amount based upon their position on the board.

The AI itself is based upon a Alpha-beta pruning minimax tree to avoid searching the entire search space of checkers. In correspondence with the rules of draughts the AI does not make simple moves when a capture is available. The Alpha Beta tree evaluates nodes and cuts off sub-trees who are shown to be non-optimal moves from the max nodes and vice-versa for the min nodes. When a leaf node or a psuedo leaf node is reached the value found is propagated back and the algorithm cuts more sub-trees.

Testing

The testing was performed in multiple stages. As command line input was added testing was simply printing out the basic data structures that held the relevant information. After was perfected the `printBoard()` method was added to more easily the programs output for all possible moves as well as its ability to take pieces. This testing was done largely by hand simply entering in different game states and comparing them to the results calculated via pencil and paper. Edge cases such as takes that would cross the board or edges were specifically focused during this endeavor, and went through gradual corrections leading to the current version.

After the completion of the rules portion of draughts testing was less thorough but more broad-ended on the actual AI. Scenarios were created in order to test if it were making a move I would recognize as overly beneficial, as well as making moves that would end the game, such as described in the end of the practical pdf.

Playing games against the AI, to put it colloquially, was like playing against a very intelligent child. It makes several intelligent moves, especially when it comes to the late game, however it consistently makes moves that baffle and seem counter-intuitive. The addition to the base heuristic that was added seemed to remedy this slightly, as control of the center is very key to early play.