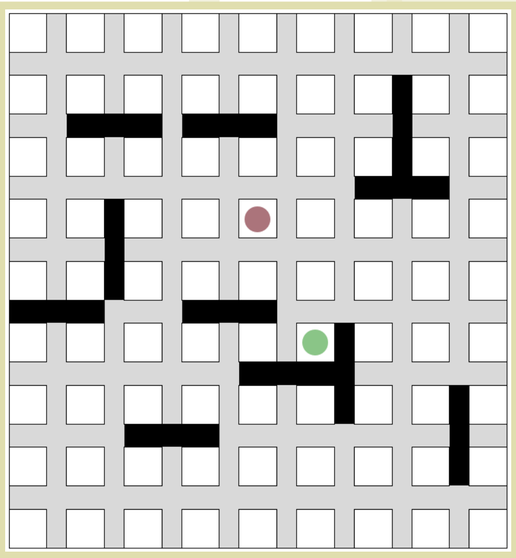
University of Witwatersrand, Johannesburg
Zagesh Parshotam, 1845347
Supervisors: Branden Ingram, Pravesh Ranchod

MIN-MAX VERSUS MONTE CARLO TREE SEARCH FOR QUORIDOR

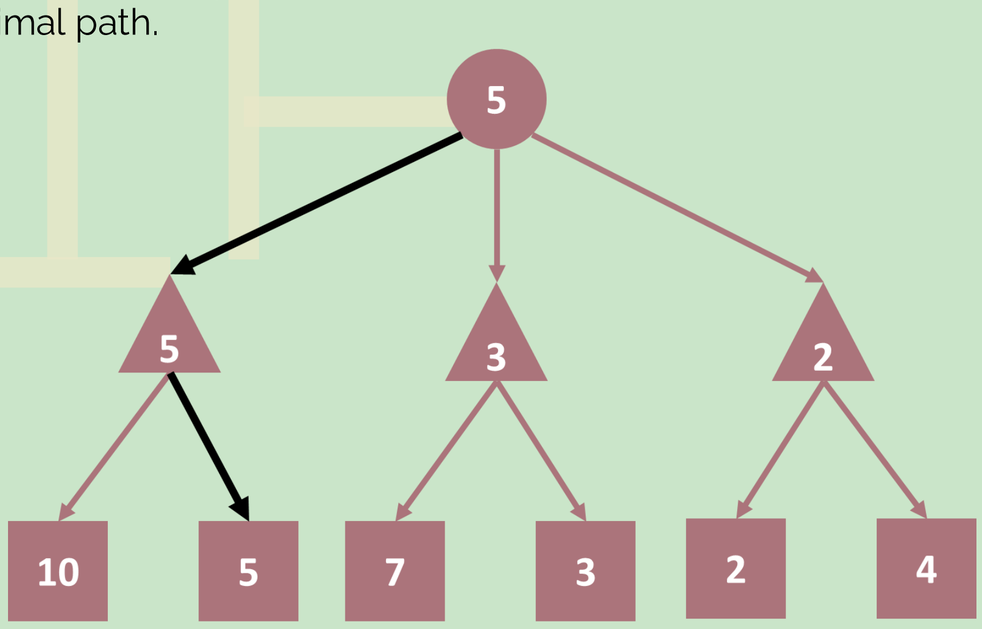


Motivation

- Great significance of artificial intelligence in game theory, optimizing decision-making processes and tactics in competitive environments.
- The objective of this paper was to compare **Min-Max** and **Monte Carlo Tree Search** (MCTS).
- By providing insights into these algorithms, we aim to contribute to the advancement of game theory and potentially facilitate their application in real-world scenarios.

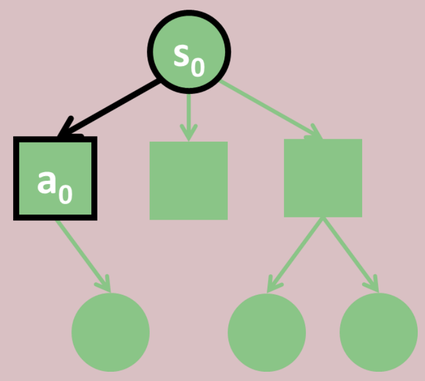
Min-Max:

The main idea is for the player to choose the optimal move for themselves at each step.
Max will choose the maximum score and **Min** will choose the minimum score when it is their turn.
The figure below is an example game tree where circle nodes represent **Max** moves and triangle nodes represent **Min** moves. The square nodes represent terminal nodes. The bold arrows show the optimal path.

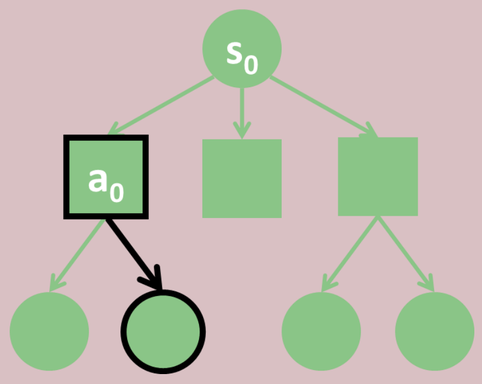


We used **Alpha-Beta Pruning** in our Min-Max implementation.
When we know the current score will not be affected by a child node, we prune that branch.
The use of this pruning allowed this agent to select moves faster.

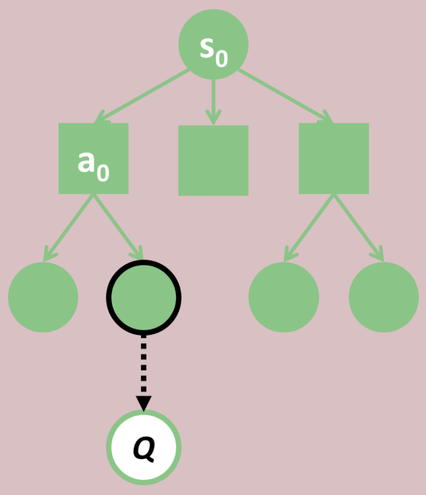
MCTS:



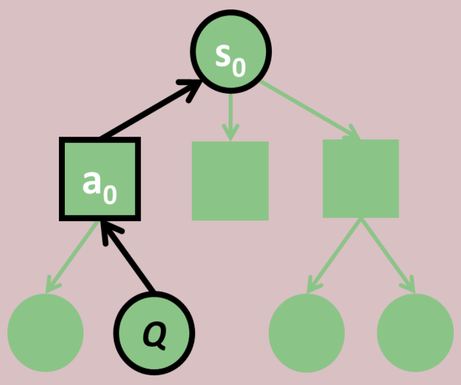
Selection phase is where one of the children of the current node is chosen. The main idea being to select the node that finds the best result. An Upper Confidence bound (UCT) was used to select the node with the best result.



Expansion phase is where we decide if a leaf node will be expanded. The children of this node are stored. The first child that has not been evaluated is the first node to be expanded.



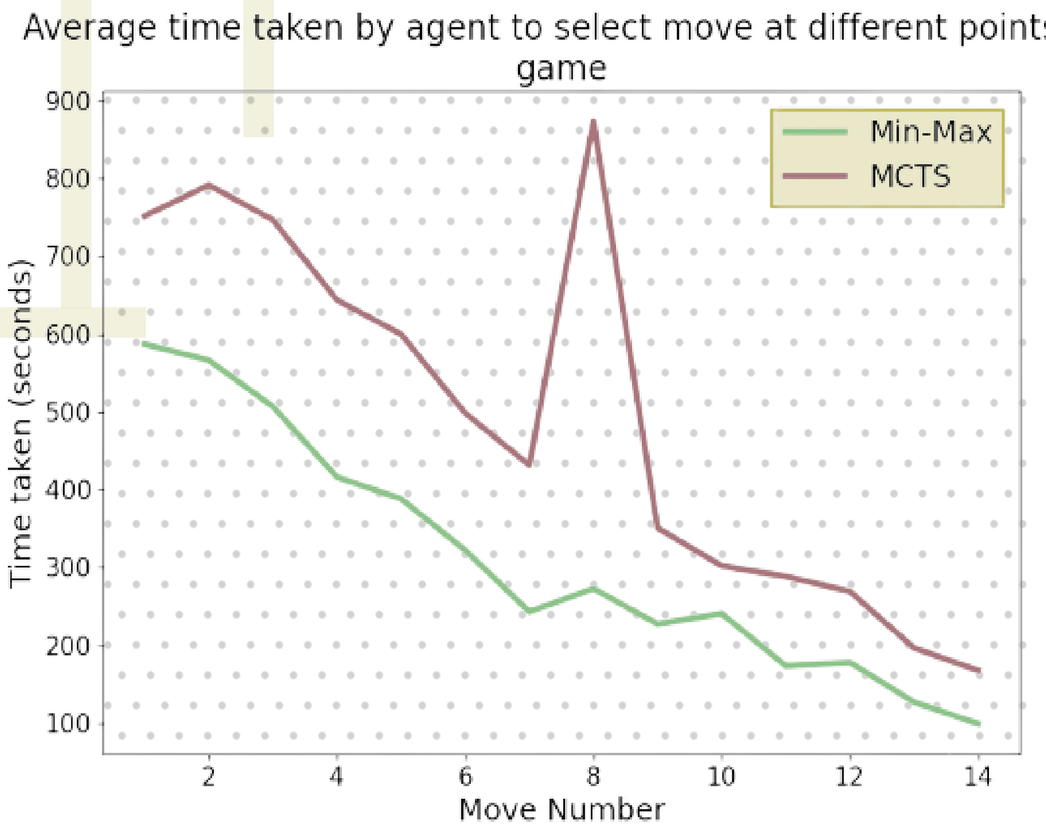
Simulation/Rollout phase plays moves until the end of a game is reached. The score from the terminal node is saved. We used a random strategy to select moves.



Back-propagation phase is where the result of the simulation phase is taken back up the tree. The score is moved up to the root and every node's new value is calculated.

Results:

- Both agents shown to take longer to choose a move at the start of the game than towards the end.
- Min-Max, proved to be superior at all points in the game.
- MCTS has to explore every possible move while Min-Max is able to prune the inferior nodes.



- MCTS took longer to select moves in each game.
- Min-Max has a much better time complexity for selecting moves.
- Alpha-Beta pruning removed inferior moves.

Number of wins:
MIN-MAX : 6
MCTS : 0