AI Assignment 2: Connect 4

Japan Patel - s3595854

Implementation

The Agent implemented plays well against other provided AI and intermediate level human player and always beats the Monte Carlo Agent and Random Agent in less than 20 seconds with about avg runtime on Monte Carlo Agent being 11.5 seconds.

The AI uses a Min-Max algorithm with alpha-beta pruning with a depth of 4 and then evaluates the next states and is being assigned a score/reward for every state. The algorithm then gets the best state and plays it as the next move.

AI Techniques

The agent uses a few AI techniques to determine the best move and is done by the following techniques:
Min-Max: This is one of the search strategies where the max outcome is for the agent and the min outcome is for the opponent. This technique considers the current state of the game and picks a move, checks if the move is a terminal move or evaluates the picked move. The reason to use this technique is to improve efficiency and look deeper and in future of the game as generating a tree is very inefficient.

Alpha-Beta Pruning: Using Min-Max with higher depths increases the number of outcomes exponentially. So, to increase the efficiency of the min-max algorithm, alpha beta purring is also implemented. Alpha-beta pruning doesn't look at all the possible moves, it removes branches that don't influence the final outcome. So unnecessary states of the games are never expanded to increase efficiency.

Dynamic reward technique & opponent moves to check: This is techniques is used regularly in modern AI games where the player move is been awarded a reward on how well their next move is. The agent uses this technique to determine the next best move. The agent considers the opponent moves and reduces the reward so that the agent can stop the opponent from winning.

Evaluation functions

The agent uses the following evaluation functions to determine a score for all valid moves for the next state till the depth of 4. The score is determined by how well the agent is playing or how well the opponent player is playing.

Horizontal Evaluation: This evaluation checks all the valid columns and for all the rows in the valid column it calculates the score based total number moves made by the agent or the opponent. The evaluation then checks if there is a possibility to win and gets a score/reward from the rewards function to assign a "reward" to that state.

Vertical Evaluation: The vertical evaluation is like horizontal evaluation with a minor difference. The vertical evaluation checks all the valid rows and for all the columns in the valid rows, it calculates the score based on a total number of moves made by an agent or the opponent/ And then the reward is assigned.

Diagonal Evaluation: This evaluation is one of the advance evaluations where it checks the negative slope and positive slope for all the valid columns and valid rows of the next state of the board. It then assigns the reward/score for the state of the board.

Centre Column Evaluation: This evaluation checks if a preferred move with higher reward as during early stages of the game the agent places the move on the centre column to increase the chances of winning later in the game. This is done by checking all the rows in the centre column to check for agent's moves and reward the agent with a higher number.

All the evaluations mentioned above gets assigned a reward value which is determined by the type of the move, e.g. if there are 3 in a row compared to 2 then the reward will be higher for the 3 in a row. The reward function also checks if the opponent has 2 or 3 in a row and then assigns a negative reward to the agent so that the agent stops the opponent from winning.

Agent Strength

The agent performs well by evaluating the centre column as a better preference column compared to other columns as it increases the chances of winning in all directions. The Agent while this making a preference also checks if the opponent moves are future winning moves, the agent then decides the best move. The agent does this by checking till the depth of 4 of the min-max algorithm.

The best situation is when the agent is player 1 as it can start with the centre column. Even with being the second player the agent's evaluations function together with AI techniques mentioned above determines best moves.

Agent Weakness

Currently, the weakness of the agent is that it gives more preference to horizontal and vertical evaluation more than the diagonal evaluation. The agent still wins with diagonal evaluation and it even blocks the opponent winning moves on diagonal evaluation. The reason for this because the diagonal evaluations take double time calculating the reward/score compared to vertical or horizontal because it must check both positive and negative diagonal, so the reward has been lowered for this to keep the agent well below the 20s mark. This weakness only occurs and noticeable if the opponent player is making extremely strong moves and building traps where the agent will show not notice the opponent moves.

Possible future upgrades to the Agent

Implementing even-odd strategy where the agent is played as player 1 gives preference to row 1,3, 5 and player 2 to row 2, 4, 6 for the 4th and 2nd column and then play the match until player 1 plays the last row with player 2 winning on even rows and player 1 winning on odd rows. This is a very popular strategy and is guaranteed a win. It was implemented for this agent to keep it below 20s mark as this is a long game strategy and win mostly occurred in last moves of the game.

The other winning strategy that can be implemented by building traps for an opponent with even-odd strategy and another strategy to predict opponents move and then the agent will have only 1 move to win the game. Even with this strategy, the winning occurs in the last moves of the game.

There are other strategies that can be implemented including the ones discussed above in future to improve the agent's performance.

Overall the current agent has good AI techniques implemented and can win against another similar level AI.