Amir Mohammad Ganjizade – 402243093

AI Project

Core Project Algorithm

1. Minimax

I used the Minimax search algorithm as the foundation of my agent and improved it with additional heuristics and techniques.

2. Alpha-Beta Pruning

Integrated alpha-beta pruning into the Minimax search to eliminate branches that cannot produce a better result, improving efficiency.

3. Move Ordering

At each node in the Minimax tree, I search the possible moves in a strategic order. This improves pruning efficiency and ensures that if time runs out, the agent will have found at least one strong move.

4. Evaluation Function

When reaching a leaf node—or in situations where the quality of a game state must be determined—the evaluation function maps the state to a numerical score representing how good or bad it is. This function considers four key factors:

1. Number of filled cells – How many cells are already occupied.

2. Mobility – The number of possible moves available.

3. Threats – The number of lines that are almost complete for a player.

4. Positional Weight – Each cell has a weight indicating its importance on the board.

5. Caching with Hashing

Since the same game state may appear multiple times during the search, I implemented state caching using hashing. This stores previously evaluated states in a hash table, avoiding redundant calculations and improving speed.

Testing

I tested my agent against the sample_agent and achieved 100% win rate.