

**Project Title:** "HEXXAGON: A Multi-Dimensional Board Game"

**Submitted By:**

Hareem Farooqui 22k-5094

Tooba Jatoi 22k-6010

Rameen Saleem 22k-5117

**Course:** Artificial Intelligence  
**Instructor:** Almas Ayesha  
**Submission Date:** 26/3/2025

**1. Project Overview**

* **Project Topic:**  
  Hexxagon is a hybrid of Hex and Checkers, played on a hexagonal grid where players aim to connect opposite sides of the board (Hex objective) while capturing opponent pieces (Checkers mechanic). Innovations include piece splitting, multi-directional movement, and dynamic board states.
* **Objective:**  
  Develop a strategic AI using **Minimax with Alpha-Beta Pruning** to handle the game’s high branching factor, and design heuristics to evaluate board control and piece advantage.

**2. Game Description**

* **Original Game Background:**
  + Hex: Players connect opposite sides of a hexagonal grid.
  + Checkers: Pieces jump to capture opponents; kings gain mobility.
* **Innovations Introduced:**
  1. **Hexagonal Grid**: Enables 6-directional movement (vs. Checkers’ 4).
  2. **Splitting Mechanic**: Pieces can duplicate themselves, increasing board complexity.
  3. **Jump Captures**: Like Checkers, but with Hex’s adjacency rules.
  4. **Dynamic Goals**: Win by either connecting sides or eliminating all opponent pieces.
* **Impact on Complexity:**
  1. Splitting/jumping increases branching factor.
  2. Dual win conditions demand adaptive AI strategies.

**3. AI Approach and Methodology**

* **AI Techniques:**
  + **Minimax Algorithm**: Adapted for hexagonal movement and splitting actions.
  + **Alpha-Beta Pruning**: Critical to handle the high branching factor (~50% reduction in nodes).
  + **Heuristic Design**:
    - Board Control: Percentage of connected paths to goal edges.
    - Piece Advantage: Weighted sum of pieces (splits = +1, jumps = +2).
    - Threat Detection: Opponent’s proximity to winning paths.
* **Complexity Analysis:**
  + **Time Complexity**: ~O(b^d) where b = branching factor (estimated 15-20), d = depth (optimized to 4-5 with pruning).
  + **Challenges**: Balancing heuristic accuracy vs. computational speed.

**4. Game Rules and Mechanics**

* **Modified Rules:**
  1. **Movement**: Pieces move 1 hexagon or split/jump 2 hexagons.
  2. **Splitting**: Sacrifice a turn to duplicate a piece (max 3 splits/game).
  3. **Captures**: Jump over opponent pieces to remove them.
  4. **Turn Order**: Players alternate; splits count as a full turn.
* **Winning Conditions:**
  1. Connect two opposite edges or eliminate all opponent pieces.
* **Turn Sequence**:
  1. Player 1 → AI → Player 2 (if testing 3-player mode).

**5. Implementation Plan**

* **Programming Language**: Python (Pygame for GUI, NumPy for grid logic).
* **Libraries**:
  + Pygame (visualization),
  + NumPy (board state matrices),
  + Timeit (performance testing).
* **Milestones**:
  + **Weeks 1-2**: Finalize hexagonal grid logic and splitting mechanics.
  + **Weeks 3-4**: Implement Minimax + Alpha-Beta Pruning.
  + **Weeks 5-6**: Code heuristics and integrate GUI.
  + **Weeks 7-8**: Test AI vs. human players, optimize depth/heuristics.

**6. References**

1. Browne, C. Hex Strategy: Making the Right Connections. 2000.
2. Russell, S. & Norvig, P. Artificial Intelligence: A Modern Approach (Minimax theory).
3. Pygame Documentation.