Project Title: *HexBots – A Grid Domination Game*

Submitted By: [Muhammad Mustafa Shahzad 22k-4166
Muhammad Alyan 22k-4582
Nawfal Hussain 22k-4372]

Course: AI
Instructor: [Maam Alina Arshad]
Submission Date: [22/03/25]

1. Project Overview

- **Project Topic:**
  We are developing **HexBots**, a two-player grid-based strategy game where bots attempt to dominate a hexagonal board by capturing as much territory as possible. The game will incorporate **goal-based agents** and **adversarial search techniques** to simulate intelligent behaviour and strategic planning.

- **Objective:**

The primary goal is to build a strategic AI for the **HexBots** game using **Minimax with Alpha-Beta Pruning and goal-based agent architecture**. Each bot will act as an autonomous agent contributing to an overall game plan.
The objective is to create an engaging and intelligent system capable of competing with a human or another AI agent.

2. Game Description

- **Original Game Background:**
  HexBots is an original game concept and not based on any existing game.
  It is inspired by grid-based turn strategy games **where units (bots)** control territory through movement and interaction with opponents.

- **Innovations Introduced:**

  - Introduction of **goal-based AI agents** representing individual bots with specific roles (e.g., expansionist, blocker, controller).
  - **Use of adversarial search (Minimax + Alpha-Beta)** to evaluate optimal moves in each turn.
  - Integration of **optional Generative AI modules** to explain bot decisions or suggest strategies.
  - **A new and unique hexagonal grid** layout to introduce spatial depth and control logic.

These innovations increase the game's strategic complexity and demonstrate real AI decision-making under adversarial conditions.

3. AI Approach and Methodology

- **AI Techniques to be Used:**
  - Minimax Algorithm (with adjustments for multi-agent decision flow)
  - Alpha-Beta Pruning
  - Agents (Goal based Agent)
  - Reinforcement Learning (Optional, not core to this project)
  - Other Techniques (Generative AI for optional strategy suggestions/explanations)

- **Heuristic Design:**

  Our evaluation function will score game states based on:

  - Number of cells controlled
  - Distance to high-value targets
  - Blocking potential of opponent
  - Safety from being captured

- **Complexity Analysis:**
  - The AI implementation for HexBots involves **adversarial search using the Minimax algorithm** with **Alpha-Beta Pruning**, which explores possible future game states to choose optimal actions.

  - Assuming an average branching factor **b** (possible actions per bot per turn) and a search depth **d**, the time complexity is:

    Time complexity(Min max) = O(b^d)

    - And with Alpha-beta pruning = O(b^d/2)

  - In a turn, if each player controls **n bots** and the board allows **m possible legal moves per bot**, then:

    B = n x m

  - This leads to exponential growth of game states as the number of bots or depth increases.

Challenges:

1. **Search Space Explosion:** With multiple bots per team, the branching factor increases rapidly, making deeper lookahead computationally expensive.

2. **Heuristic Tuning:** Designing a good evaluation function that balances territory control, defense, and blocking is non-trivial.

3. **Simultaneous Multi-Agent Planning:** Although the game is turn-based, each bot can move independently, requiring **multi-agent coordination** per move.

4. **Streamlit Limitations:** Real-time game animation is limited, so rendering and turn update speed may need optimization.

5. **Optional GenAI Integration:** If used, real-time natural language generation must not slow down the core gameplay loop.

4. Game Rules and Mechanics

Modified Rules:

- Each player controls a team of bots on a hexagonal grid.
- Bots can move to adjacent cells and convert neutral or enemy tiles.
- The game proceeds in alternating turns, with each bot allowed one action per turn.

Winning Conditions:

- The player who controls the majority of the board after a fixed number of turns wins.
- Tie-breaker: Number of bots remaining or proximity to central tiles.

Turn Sequence:

- Turn-based: Each player alternates turns.
- On a player's turn, all their bots are allowed to take one move/action sequentially.

5. Implementation Plan

- **Programming Language:** *(Python)*
- **Libraries and Tools:**

Components	**Tool/Library**
Language	Python
UI	Streamlit
Board/Matrix logic	NumPy
AI Logic	Custom Minmax + Alpha-Beta
Agent Architecture	Python classes + Goal Logic
GenAi(optional)	OpenAi/Hugging Face Api's
Visualisation(optional)	Matplotib

- **Milestones and Timeline:**

  ○ **Week 1-2:** Game design and rule finalisation

  ○ **Week 3-4:** AI Development (goal based agent + Minmax + heuristic)

  ○ **Week 5-6:** Coding core mechanics and full gameplay loop

  ○ **Week 7:** AI integration, testing and debugging

  ○ **Week 8:** Final testing, polishing and report /demo preparation

6. References

- Russell, S. J., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*.
- OpenAI Documentation (for optional LLM integration)
- Streamlit Documentation
- Research papers on adversarial search and heuristic optimization