Hex Game – A Strategic Board Game with AI Opponents

1. Project Overview

This project aims to develop an interactive version of the Hex Game, a two-player abstract strategy board game, using Python. The game will feature a graphical user interface built using either Tkinter or Pygame, and will include two AI opponents using classic and modern algorithms: Minimax with Alpha-Beta Pruning and Monte Carlo Tree Search (MCTS). The game will allow a human player to compete against the AI, offering different levels of challenge and showcasing AI decision-making techniques in a strategic game environment.

2. Objectives

- Implement the Hex Game logic for any NxN grid (starting with 11x11).
- Create an intuitive and interactive GUI using Tkinter or Pygame.
- Implement two AI strategies:
  - Minimax with Alpha-Beta Pruning for medium difficulty
  - Monte Carlo Tree Search (MCTS) for hard difficulty
- Allow Human vs AI gameplay with dynamic difficulty selection.
- Ensure efficient performance and responsive user interactions.

3. Technologies Used

- Language: Python 3.x
- GUI Library: Tkinter or Pygame (final choice to be confirmed after prototyping)
- Algorithms:
  - Minimax with Alpha-Beta Pruning
  - Monte Carlo Tree Search (MCTS)
- Tools:
  - Git (for version control)
  - Google Colab/Jupyter (for algorithm testing)
  - IDE: VSCode / PyCharm

4. Game Rules Summary

- The Hex board is a rhombus-shaped NxN grid of hexagonal cells.
- One player connects top to bottom (Blue), the other connects left to right (Red).
- Players alternate turns placing their colored piece on an empty cell.
- The first to form an unbroken connection from their sides wins.
- No draws are possible in Hex.

5. System Architecture

- Frontend Layer: Game rendering and player interaction (Tkinter/Pygame)
- Backend Logic: Game rules, move validation, winner detection
- AI Layer: Decision-making logic for the two AI strategies

6. AI Strategy Details

- Minimax + Alpha-Beta Pruning:
  - Depth-limited search tree to evaluate game states
  - Heuristic evaluation function based on path connectivity
- Monte Carlo Tree Search:
  - Random simulations to estimate win probabilities
  - Selection, Expansion, Simulation, Backpropagation phases

7. Project Milestones & Timeline

Phase	Tasks	Estimated Duration
Planning & Design	Game rules, UI sketch, algorithm research	1 week
Core Game Logic	Grid, move validation, win detection	1 week
GUI Implementation	Interactive board, game flow, player input	1–2 weeks
AI Integration	Minimax with pruning, MCTS integration	2 weeks
Testing & Optimization	Game balancing, bug fixing, performance tuning	1 week
Final Presentation & Report	Documentation, demo video, cleanup	1 week

8. Challenges & Risks

- Ensuring AI performs well on large boards (state space explosion)
- Maintaining smooth GUI performance during AI computation
- Efficient win detection on hexagonal paths
- Balancing AI difficulty to be fair but challenging

10. Future Enhancements (Post-MVP)

- Human vs Human multiplayer mode (local and online)
- Adjustable board sizes (e.g., 7x7 to 11x11)
- AI vs AI mode for demonstration
- Animated move hints or path visualizations
- Save/load game functionality

11. Conclusion

The Hex Game project will provide a complete strategic board game with a polished interface and intelligent AI opponents. It offers a platform for understanding game theory, algorithm efficiency, and GUI design in Python. With both classic and modern AI implementations, it serves as an engaging and educational game for players and developers alike.