AI Project Proposal Submission

1. **Project Title**
   Nostalgic Snake Game
2. **Problem Statement**
   The problem at hand is playing the snake game most efficiently. On old phones, when we used to play the game, we usually got a high score and then tried to beat that score in the next *x* tries. However, beating that score took time, and we often made inefficient moves, rarely playing the game to its theoretical limit. Our model will be important as it will play the same game while staying within the same rule boundaries but making the most efficient moves to achieve the maximum possible score in a single run.
3. **Proposed Solution**
   First, we will define the rule boundaries of the game. Then, we will break down the problem using AI techniques and apply various models to find the most efficient movement strategy. Our approach is unique because it integrates multiple AI methods to optimize gameplay while learning dynamically. Unlike human players, who rely on experience and trial-and-error, our model will continuously improve and adapt to maximize the score in every attempt.
4. **AI Algorithm(s) to be Used**
   - **Particle Swarm Optimization (PSO):** A potential method to optimize movement strategies based on a swarm intelligence approach.
   - **Genetic Algorithm:** To evolve the best strategies over multiple iterations.

   These algorithms are suitable because they provide adaptive learning, efficient pathfinding, and optimization techniques that align with the game's objective.

5. **Group Members details and Work Distribution**
   Saad Thaplawala-27172 Sharjeel Chandna-24384.
6. **Relevant Research Papers**
   - *Deep Q-Learning for Snake Game Optimization* –link
   - *AI-Based Game Strategies for Classic Snake Game-link* – IEEE Conference Paper
   - *Teaching AI to play snake game-link*
7. **Input Data**
   - **Game environment data:** The game grid, snake position, fruit position, and movement constraints.
   - **Real-time state updates:** The AI will learn based on continuous state transitions.
   - **No external dataset required:** The game itself generates all required data dynamically.

8. **Expected Output**
    - A model capable of playing the Snake game optimally.
    - Best moves calculated based on real-time game states.
    - Maximum possible score achieved efficiently in one run.
9. **Reference Examples (Existing Projects or Implementations)**
    - **https://github.com/shohan-pherones/snake-game-self-played**
    - *https://github.com/maw157/ME384-kilobots*
10. **Feasibility & Challenges**

- **Challenges:**
    - AI model selection may require adjustments during implementation.
    - Reinforcement learning might need extensive training to generalize strategies effectively.
    - Optimizing for real-time decisions while maintaining efficiency.
- **Tools and Resources:**
    - IDE: **VS Code**
    - Version Control: **GitHub**
    - Libraries: **TensorFlow/PyTorch (for RL), NumPy, OpenCV, Pygame**
    - AI Tools: **LLMs, AI chatbots for research and debugging**