### **Project Proposal: Two-Player Pac-Man Game with AI-Driven Search Algorithms**

#### **1. Project Overview**

The goal of this project is to develop an AI-driven two-player Pac-Man game that incorporates search algorithms to simulate an interactive gameplay experience. The game will feature two primary characters: Pac-Man (the chaser) and the Ghost (the target). The player controlling Pac-Man must chase the Ghost within a given time frame while attempting to eat food scattered across the board. If Pac-Man fails to eat all the food within the time limit, the roles of the characters will switch, and Pac-Man will become the pursued target while the Ghost chases Pac-Man. The game will end when either Pac-Man or the Ghost catches the other or when Pac-Man successfully eats all the food.

This project will explore the application of searching algorithms such as A\*, Minimax, and other AI strategies to manage the behavior of the Pac-Man and Ghost entities. The project will also focus on optimizing search space expansion and improving algorithm performance for an efficient and engaging game experience.



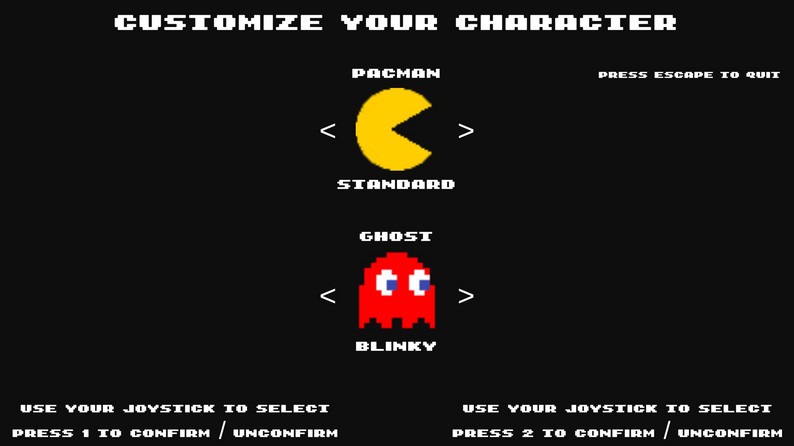
#### **2. Objectives**

* Develop a dynamic, AI-driven two-player Pac-Man game.
* Use search algorithms to manage the movement and strategy of Pac-Man and the Ghost.
* Implement a user-friendly interface to facilitate an enjoyable gaming experience.
* Analyze and compare the algorithmic complexity, search space optimization, and performance of different algorithms.

#### **3. Game Concept and Mechanics**

The game features two primary agents:

1. **Pac-Man (Chaser)**: The player controlling Pac-Man must catch the Ghost within a time limit while also attempting to eat all the food items scattered throughout the maze.
2. **Ghost (Target)**: The Ghost attempts to avoid being caught by Pac-Man while also trying to eat food items. When Pac-Man fails to eat all the food in time, the Ghost becomes the chaser, and Pac-Man becomes the pursued target.



Key mechanics:

* **Time Limit**: Pac-Man has a limited time to eat all the food or catch the Ghost.
* **Role Reversal**: If Pac-Man fails to catch the Ghost or eat all the food in time, the roles reverse, and Pac-Man becomes the pursued character.
* **Endgame Condition**: The game ends when either Pac-Man catches the Ghost, the Ghost catches Pac-Man, or all the food is consumed by Pac-Man.

#### **4. Search Algorithms and AI Integration**

The project will utilize the following search algorithms for AI decision-making:

* **A**\*: For pathfinding, allowing Pac-Man and the Ghost to efficiently find the shortest paths to their targets.
* **Minimax Algorithm**: Used for decision-making in adversarial environments. This will enable the Ghost to predict and counter the Pac-Man's movements, while Pac-Man can use it to predict and evade the Ghost.
* **Alpha-Beta Pruning**: To optimize the Minimax algorithm by reducing the number of nodes evaluated, enhancing performance during gameplay.

Each algorithm will be implemented to manage the movements and actions of Pac-Man and the Ghost, ensuring an AI-driven interactive game environment.

#### **5. Game Board and UI Design**

* **Game Board**: A grid-like layout representing the maze, with food items scattered across the board. Pac-Man and the Ghost move through the maze based on algorithmically determined paths.
* **User Interface**: A simple, engaging UI where the game status (e.g., time left, remaining food items, current roles) is visible. Users will interact with the game using keyboard controls, and the game will display real-time updates.
* **Intuitive Interaction**: The interface will be designed to be easy for the user to understand, with clear visual indicators of player roles, remaining food, and time.



#### **6. Search Space Expansion**

We will expand the search space by:

* Adding more complex levels with different layouts, increasing the search space for both Pac-Man and the Ghost.
* Dynamically adjusting the AI difficulty, with the Ghost adapting its strategy based on Pac-Man’s performance.

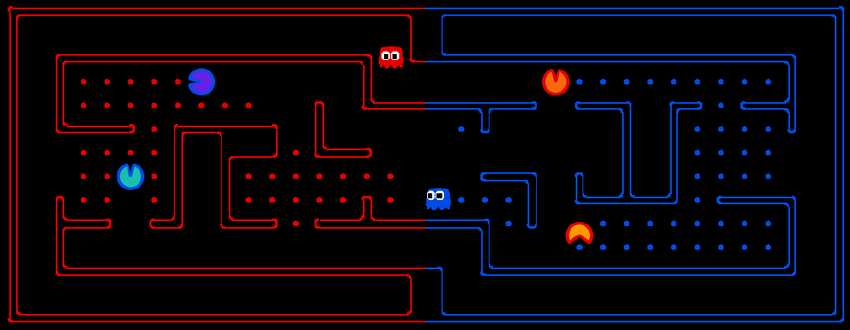
#### **7. Complexity Analysis and Comparative Evaluation**

The project will include:

* A comparison of the algorithmic complexity of A\*, Minimax, and Alpha-Beta Pruning.
* Analysis of the number of searches performed, search space efficiency, and optimization.
* Dry-run demonstrations will showcase the performance of each algorithm in various scenarios.

#### **8. Methodology**

1. **Game Design and Prototyping**:  
   * Develop the game board, implement the basic Pac-Man and Ghost mechanics, and create a simple prototype.
2. **Search Algorithm Integration**:  
   * Implement A\*, Minimax, and Alpha-Beta Pruning for pathfinding and decision-making.
3. **UI Design**:  
   * Design a clean, user-friendly interface for easy interaction.
4. **Testing and Optimization**:  
   * Perform rigorous testing of the search algorithms, optimize performance, and evaluate the game’s AI behavior.
5. **Comparative Analysis**:  
   * Perform a comparative analysis of algorithmic performance, search space optimization, and user experience.

  
.

#### **9. Deliverables**

* **Code**: Fully functional game with search algorithm integration and UI.
* **Report**: A detailed report discussing the methodology, algorithms used, comparative analysis, and performance evaluation.
* **Dry-Run Demonstration**: A demonstration showing the algorithm execution and gameplay.