Project Report Template

Project Title: *Dynamic Dots and Boxes*
Submitted By: Abbas Ali Mirza [22K-4568], Lamia Rashid Jamal [22K-4524], Laiba Tabraiz Khan [21K-4620]
Course: AI
Instructor: Abdullah Yaqoob
Submission Date: 11-May-2025, Sunday

1. Executive Summary

- **Project Overview:**
  This project involves modifying the conventional Dots and Boxes game by integrating power token mechanics and advanced AI strategies. In the modified version, players (both human and AI) earn a power token upon capturing a box. When a token is used for a power-up, no additional token is awarded during that turn, even if subsequent boxes are captured until the opponent completes a turn. The AI decision-making is based on both the traditional Minimax algorithm and its optimized version using Alpha-Beta pruning. The user can select the AI algorithm at the start of the game. These modifications aim to create a more strategic and balanced gameplay experience while exploring state-space search techniques in AI.

2. Introduction

- **Background:**
  Dots and Boxes is a traditional pencil-and-paper game where two players take turns connecting adjacent dots to form boxes. The conventional game rewards extra turns when a player completes a box. This project was chosen to investigate how rule modifications (including power token usage) and sophisticated AI algorithms can impact game dynamics. The introduction of strict token management and alternative AI algorithms (Minimax and Alpha-Beta pruning) serves as a novel twist on the original gameplay.
- **Objectives of the Project:**
  - *Develop an AI model capable of playing a modified version of Dots and Boxes using state-space search techniques.*
  - *Enforce power token constraints ensuring that once a token is used in a turn, no*

new token is awarded until the opponent's turn.

• *Provide users with game options such as difficulty, and grid size*

3. Game Description

- **Original Game Rules:**
  Dots and Boxes is played on a grid of dots. Players alternate in drawing lines between adjacent dots. When a player completes the fourth side of a box, they capture that box and score a point. The conventional game awards an extra turn for every box captured.
- **Innovations and Modifications:**
  • *Integration of power token mechanics – A captured box earns a token only if no token has been used already in that turn.*
  • *New power-up functionalities such as Extra Move, Line Reversal, and Swap Token allow strategic intervention during gameplay.*
  • *The token award is limited to one per turn; once used, the token cannot be re-earned until the opponent's turn.*

4. AI Approach and Methodology

- **AI Techniques Used:**
  • *Minimax Algorithm: A traditional search method used to evaluate all possible moves and determine the optimal strategy.*
  • *Alpha-Beta Pruning: An optimization of the Minimax algorithm that reduces the number of nodes evaluated, thereby improving decision-making time.*
- **Algorithm and Heuristic Design:**
  The evaluation function calculates the difference between the number of boxes captured by the AI and the human player. The heuristics are designed to favor moves that maximize box captures and effective use of power tokens while preventing abuse by limiting token re-award within one turn.
- **AI Performance Evaluation:**
  The alpha-beta pruning was significantly faster than the min-max algorithm as it took less computational time and power to make the new move.

5. Game Mechanics and Rules

- **Modified Game Rules:**
  • *On capturing a box, a player is awarded one power token if no token has been used during that turn.*
  • *If a power token is used in any move during the turn (for an extra move, line*

reversal, or swap token), the player is not able to use the token again until the turn passes.
  - *The token usage flag resets when the opponent completes their turn.*

- **Turn-based Mechanics:**
  - *Players take turns drawing lines between dots.*
  - *Capturing a box allows the same player to continue their turn; however, if a power token is used during this extended turn, no additional token is awarded.*
  - *The game proceeds until all possible lines have been drawn, and then the captured box count determines the winner.*
- **Winning Conditions:**
  The winner is defined as the player who has captured the most boxes once the board is completely filled.

6. Implementation and Development

- **Development Process:**
  The project was implemented using Python and the Pygame library. The process involved:
  - *Developing the basic game mechanics for Dots and Boxes.*
  - *Integrating power token mechanics and modifying the turn-based logic.*
  - *Implementing two AI algorithms: traditional Minimax and the optimized Alpha-Beta pruning.*
  - *Creating user menus for selecting difficulty, grid size, and the AI algorithm.*
  - *Extensive testing to balance gameplay and AI performance.*
- **Programming Languages and Tools:**
  - Programming Language: Python
  - Libraries: Pygame
  - Tools: GitHub for version control, Visual Studio Code for code development
- **Challenges Encountered:**
  - *Ensuring that power tokens were awarded only once per turn and that using a token correctly locked additional token awards within a turn.*
  - *Balancing AI performance so that the decision-making was both competitive and efficient in various difficulty settings.*
  - *Integrating multiple menus (difficulty, grid size, AI algorithm) into a cohesive user interface.*
  - *Enhancing performance of the algorithms*

7. Team Contributions

- **Team Members and Responsibilities:**

- ○ **[Abbas Ali Mirza]:** Responsible for power tokens functionality and its corresponding UI.
- ○ **[Lamia Rashid Jamal]:** Designed the menus, boards and implemented minimax algorithm.
- ○ **[Laiba Tabraiz Khan]:** Implemented the alpha-beta pruning and end game scenario.

8. Results and Discussion

- **AI Performance:**
  The AI achieved competitive performance, winning most of the matches against human players when set to higher difficulty levels. The use of Alpha-Beta pruning reduced decision-making times significantly compared to longer times with the basic Minimax algorithm. This confirmed the effectiveness of the optimization in a turn-based, competitive game environment.

9. References

- *Pygame Documentation. Retrieved from https://www.pygame.org/docs/*

- *Online resources and tutorials on implementing the Minimax algorithm and Alpha-Beta pruning in game theory.*