**GROUP MEMBERS:**

1. **Malaika(k22-5059)**
2. **Muhammad Asim (k22-5066)**

**Project Report: 3-Player Checkers Game with AI Bot**

**1. Introduction**

This report documents the design, development, and functionality of the "3-Player Checkers Game" built using Python and Tkinter. The game introduces an exciting twist to traditional checkers by adding a third player controlled by an AI bot, making the gameplay more dynamic and challenging.

**2. Project Objectives**

* Develop a 3-player version of Checkers.
* Implement an AI bot using the Minimax algorithm with alpha-beta pruning.
* Design an intuitive graphical interface using Tkinter.
* Ensure robust game logic including movement, capturing, and winning detection.

**3. System Requirements**

* Python 3.x
* Tkinter library (pre-installed with Python)

**4. Project Components**

**4.1 Game Board**

* 15x15 grid.
* Red and Blue players start from bottom corners.
* Green (AI) player starts from top center.
* Empty cells represented by '.' character.

**4.2 GUI Implementation**

* Tkinter Canvas used to draw the board and pieces.
* Pieces drawn as colored circles.
* Highlight for selected piece.
* Display message when the game ends.

**4.3 Game Mechanics**

* Players take turns based on a turn counter (Red -> Blue -> Green).
* Valid moves include moving diagonally to adjacent cells and capturing opponent pieces.
* Captures involve jumping over an opponent's piece.
* Turn management ensures fair play among all three players.

**4.4 AI Implementation**

* Green player is controlled by a bot using the Minimax algorithm.
* Depth of 2 used for search.
* Alpha-beta pruning to improve performance.
* Evaluation function favors states where Green has more pieces.

**5. Game Flow**

1. Game initializes the board.
2. Red player's turn (manual move).
3. Blue player's turn (manual move).
4. Green bot's turn (automatic move).
5. After every move, check if a player has won.
6. Repeat steps 2-5 until a winner is declared.

**6. Challenges Faced**

* Balancing the AI difficulty while keeping computation time low.
* Designing fair starting positions for three players.
* Managing player turns efficiently.
* Validating all types of moves, including captures.

**7. Testing**

* Manual play throughs to ensure the correctness of movement and capturing.
* Simulated moves to verify AI responses.
* Edge cases tested, such as no available moves, multiple captures, and win condition accuracy.

**8. Results**

The final version successfully:

* Allows two human players and one AI to compete.
* Provides a smooth and responsive GUI.
* Ends the game correctly when only one player remains.
* Demonstrates AI decision-making within practical time limits.

**9. Future Work**

* Implement piece promotion (kings).
* Add adjustable AI difficulty.
* Create networked multiplayer mode.
* Enhance graphics with animations and sound effects.

**10. Conclusion**

The 3-Player Checkers project effectively combines GUI development, game logic, and AI algorithms into an enjoyable and functional board game. It achieves its goals of delivering a unique gameplay experience and demonstrating key programming concepts in Python.

✨ End of Report ✨