3D Tic-Tac-Toe

Group 19

PROJECT REPORT

ITIS 6150: Intelligent Systems

Dr. ZBIGNIEW W. RAS

Team Members

Aishwarya Bathina - 801328057

Pavan Gopi Praneeth Gidda - 801328702

Harshini Baswapuram - 801329511

Sri Datta Kiran Avasarala - 801318007

ABSTARCT

This project presents a reimagined 3D version of Tic-Tac-Toe, implemented in Java Swing, offering an engaging and challenging gaming experience. The user interface employs a CardLayout, seamlessly transitioning between the title screen, game options, and the 3D game board. Customization features, accessible through a well-organized options panel, allow players to tailor their experience, choosing difficulty levels, game pieces, and starting players.

At the core of the application is a 3D game board represented by a four-by-four-by-four grid, introducing a new layer of complexity to the traditional game. Gameplay challenges players to think strategically in three dimensions, with winning conditions evaluated across horizontal, vertical, and diagonal lines. An intelligent AI opponent, employing a minimax algorithm with alpha-beta pruning on higher difficulty settings, adds an intriguing element, enhancing both single-player and AI enthusiasts' experiences.

This project not only elevates the classic Tic-Tac-Toe to new dimensions but also serves as an educational tool. Through the combination of a user-friendly interface, customizable gameplay, and a sophisticated AI opponent, it showcases the potential of Java Swing in creating engaging applications. This 3D Tic-Tac-Toe application stands as an exemplary project for students and enthusiasts in computer science, offering insights into advanced programming concepts, AI implementation, and UI design.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER - 1

INTRODUCTION

Tic-Tac-Toe, a timeless classic celebrated for its simplicity, has woven itself into the fabric of childhood and casual gaming experiences. Typically confined to a two-dimensional 3x3 grid, our project, titled "3D Tic-Tac-Toe: An Enhanced Java Swing Application," embarks on a transformative journey by introducing a third dimension. This evolution seeks to provide players with a more intricate and engaging gaming experience, challenging them to strategize in three dimensions. The introduction delves into the motivation behind this project, emphasizing the aspiration to elevate the cognitive demands of the traditional game, while also serving as a practical application of advanced Java programming concepts, including GUI development with Java Swing and artificial intelligence (AI) integration.

Motivated by the desire to enhance strategic thinking and spatial awareness, the project's objectives are threefold: to develop an interactive 3D Tic-Tac-Toe game through Java Swing, offering an immersive user experience; to implement varying difficulty levels by integrating an AI opponent with a minimax algorithm and alpha-beta pruning; and to serve as an educational tool, exemplifying the practical application of advanced programming concepts and AI in the realm of game development. The subsequent sections will delve into the methodologies, technologies employed, and the comprehensive development process of this innovative and educational gaming project.

CHAPTER - 2

PROJECT DESCRIPTION

The "3D Tic-Tac-Toe: An Enhanced Java Swing Application" project revolutionizes the classic Tic-Tac-Toe game by transitioning from the traditional 2D board to an engaging 3D grid. Developed using Java, the primary objective of this project is to add a new level of strategic depth and challenge to the familiar game. The use of Java Swing for the graphical user interface (GUI) ensures a visually appealing and user-friendly experience. This project aims to provide an innovative gaming experience while also serving as an educational tool for those interested in learning about GUI development, game logic programming, and artificial intelligence (AI) in Java.

The game's interface is meticulously crafted using Java Swing, offering various interactive screens for settings and gameplay. The centerpiece is a 4x4x4 grid, expanding the classic gameplay into a three-dimensional challenge. This multidimensional approach not only enhances the complexity but also requires players to think and strategize in a 3D space. The game logic is robust, incorporating the ability to check for winning combinations across the 3D grid, including horizontal, vertical, and diagonal lines within and across the layers. An integral component of the game is the AI opponent, designed to challenge players at different skill levels. The AI uses a minimax algorithm with alpha-beta pruning, especially effective at higher difficulty settings, thus making the game engaging and challenging for both novice and seasoned players.

CHAPTER - 3

METHODOLOGY

Initialization and GUI Development:

The project begins with the creation of a graphical user interface (GUI) using Java Swing. This interface includes radio buttons for players to select the difficulty level, their game piece (X or O), and who gets the first move. A critical function of the GUI is to update button labels based on player moves, enhancing the interactive experience. The GUI is designed to be user-friendly and intuitive, providing a seamless transition between different game states and options.

Initializing the Game Board:

The core of the game is a 4x4x4 three-dimensional array, representing the game board. Each cell within this array is initialized to a blank state, symbolizing an unoccupied position. This setup is pivotal in facilitating the 3D aspect of the game, providing a more complex and engaging playing field compared to traditional 2D Tic-Tac-Toe.

Player Turns and Input Validation:

The game logic is structured to prompt players for their moves, requiring them to input coordinates (x, y, z). A validation mechanism is implemented to ensure that the chosen square is unoccupied and within the game board's bounds. The game board's data structure is updated accordingly to reflect each player's move.

Alternate Turns and Game Continuation:

A loop structure is employed to alternate turns between players. This loop continues until a winning condition is met or a tie is declared. After each game's conclusion, options are provided for restarting the game or exiting the application, with the GUI reflecting the current state of the game board throughout the process.

Checking Winning and Tie Conditions:

Algorithms are developed to check for winning conditions after each move. The game logic includes functions to determine a winner by checking for four consecutive pieces in a row, either horizontally, vertically, or diagonally, across each 2D slice of the 3D board. Additionally, a tie condition is recognized when all squares are filled without a winner, prompting a tie message in the GUI.

AI Implementation:

An AI opponent, utilizing a minimax algorithm with alpha-beta pruning, is integrated for higher difficulty levels. This AI analyzes potential moves and outcomes, providing a challenging experience for players. The AI's difficulty is adjustable, catering to both beginners and advanced players.

MinMax Algorithm

The MinMax algorithm is a decision-making algorithm, commonly used in two-player games such as Tic Tac Toe. The algorithm searches through the game tree of possible moves, evaluating each position to eventually choose the move that maximizes the player's chances of winning while minimizing the opponent's chances.

Alpha-Beta Pruning

To optimize the MinMax algorithm, alpha-beta pruning is used. This technique reduces the number of nodes evaluated in the search tree, effectively pruning away branches that won't influence the final decision.

Alpha represents the minimum score that the maximizing player (computer) is assured of. Beta represents the maximum score that the minimizing player (human) is assured of.

As the search progresses, if the minimax value of a position is worse than the current alpha for the maximizer, or the beta for the minimizer, that branch is pruned as it won’t affect the final decision.

Application in Code

Computer Move Evaluation: The computerMove() method is responsible for initiating the MinMax process. It looks for the optimal move by iterating through all possible moves and evaluating them using the MinMax algorithm.

MinMax Function: The findBestMove() function implements the MinMax logic. It alternates between maximizing and minimizing turns, evaluating each move's potential and recursively calling itself to explore deeper levels of the game tree.

Alpha-Beta Pruning: Within the findBestMove() function, alpha and beta parameters are utilized to implement pruning. The function checks if the current board score is worse than the alpha or beta values and prunes the branches accordingly.

Heuristic Function: The heuristic() method is used to evaluate the game board positions. This evaluation is based on the number of potential winning combinations available to each player. It helps in deciding the most promising moves when the depth limit is reached.

Challenges and Considerations

Performance: Due to the 3D nature of the game and the increased complexity of the game tree, performance and speed of the algorithm are crucial. Alpha-beta pruning significantly helps in this aspect by reducing the number of evaluated nodes.

Depth of Search: The depth to which the algorithm searches the game tree is controlled to balance between decision quality and computation time. Deeper searches yield better decisions but require more processing time.

Winning Move Detection: The algorithm prioritizes immediate winning moves, both for the computer and against the human opponent, to ensure the best immediate outcome.

In conclusion, this methodology outlines a structured approach to developing a 3D Tic-Tac-Toe game in Java, ensuring clarity and ease of implementation. The process encompasses all aspects of game development, from GUI creation and game logic to AI integration and application testing, culminating in a sophisticated and engaging gaming application.

CHAPTER - 4

SYSTEM FLOW DIAGRAM

1. Initialization and Turn Management:

- The game starts by initializing a 4x4x4 board, setting each cell to an unoccupied state.
- Players take turns to input coordinates (x, y, z) for their moves. The system validates these inputs for occupancy and board boundaries.
- The game board is updated to reflect each player's move, alternating between Player 1 and Player 2.

2. Win and Tie Checks:

- After each move, the system checks for a win condition: four pieces in a row horizontally, vertically, or diagonally in any direction on the 3D board.
- If a player achieves this condition, a victory message is displayed, and the game ends.
- Simultaneously, the system checks for a tie—if all squares are filled without a winner, it declares a tie, displays a corresponding message, and ends the game.

3. Game Progression and Conclusion:

- The game continues in a loop, alternating between players until a win or tie condition is met.
- Upon conclusion, the game offers options to restart or exit.

- The GUI is dynamically updated throughout the game to reflect the current state and player interactions.

This streamlined design efficiently encapsulates the core aspects of the 3D Tic-Tac-Toe game, ensuring a clear and logical gameplay progression from start to finish. It covers the critical phases of play, including setup, turn-taking, condition checking, and game conclusion, all within a cohesive and user-friendly framework.

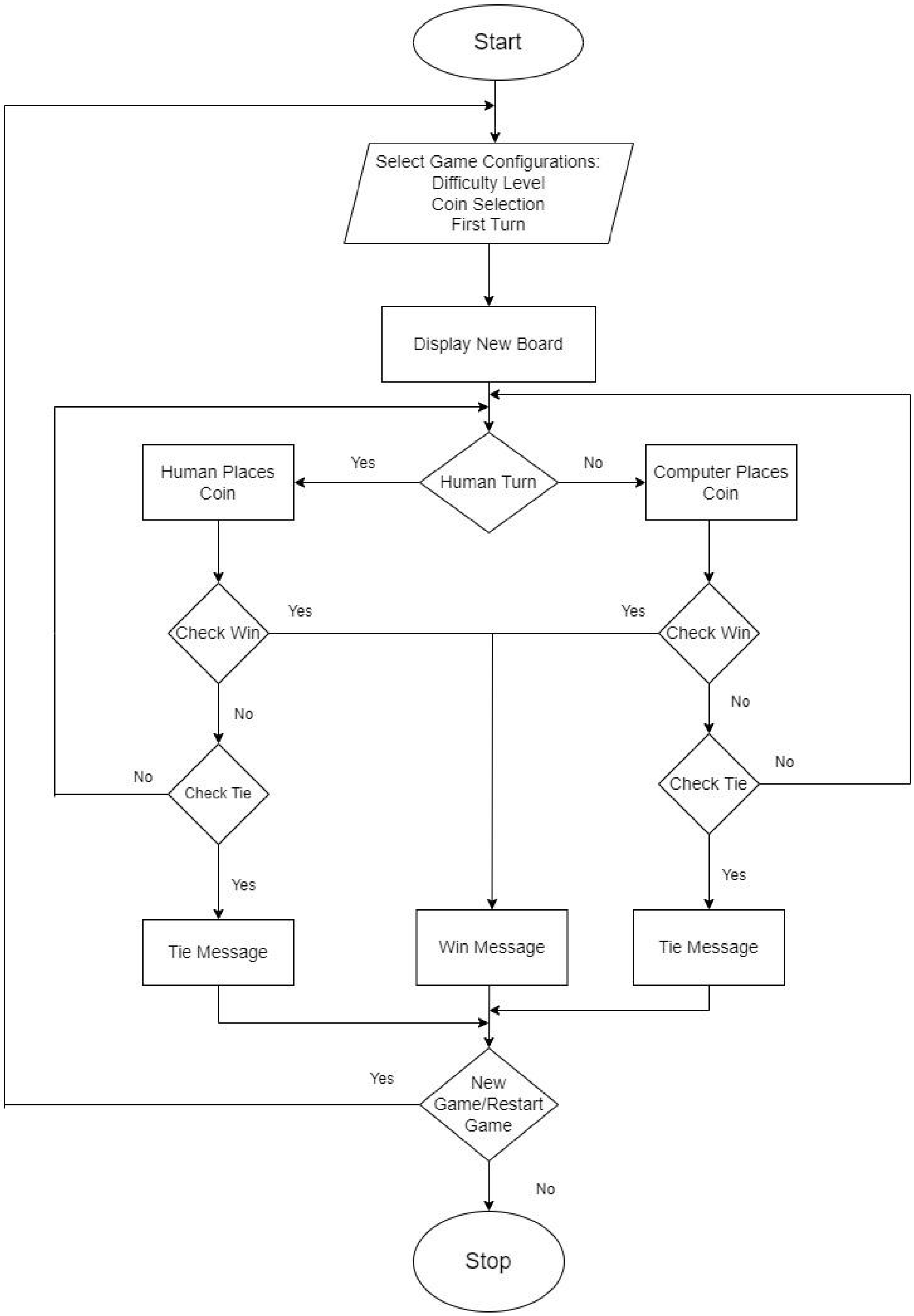

Figure 4.1: System Flow Diagram for 3D Tic-Tac-Toe.

CHAPTER - 5

DATA FLOW DIAGRAM

The Data Flow Diagram (DFD) for the 3D Tic-Tac-Toe system serves as a comprehensive visual representation depicting the intricate flow of data throughout the gaming process. Comprising various essential components, the DFD outlines key processes, inputs, outputs, and data storage mechanisms.

The system initiates by obtaining player inputs, encompassing coordinates, difficulty levels, and game piece selections. Subsequently, the processes unfold: updating the game board based on player moves, checking for win or tie conditions, and displaying the current game status to the players. The critical data stores include the Game Board State, encapsulating the evolving configuration of the 4x4x4 game board, and the Move History, recording players' moves for win or tie evaluations. External entities, such as Players and the User Interface, interact with the system by providing inputs and facilitating communication with players. The interconnectedness of these elements is visualized through arrows representing data flows, emphasizing the dynamic exchange of information.

In summary, the DFD serves as a powerful tool for comprehending the data dynamics within the 3D Tic-Tac-Toe system, offering clarity on how information is processed and communicated, enriching the understanding of the game's inner workings.

Diagram 1

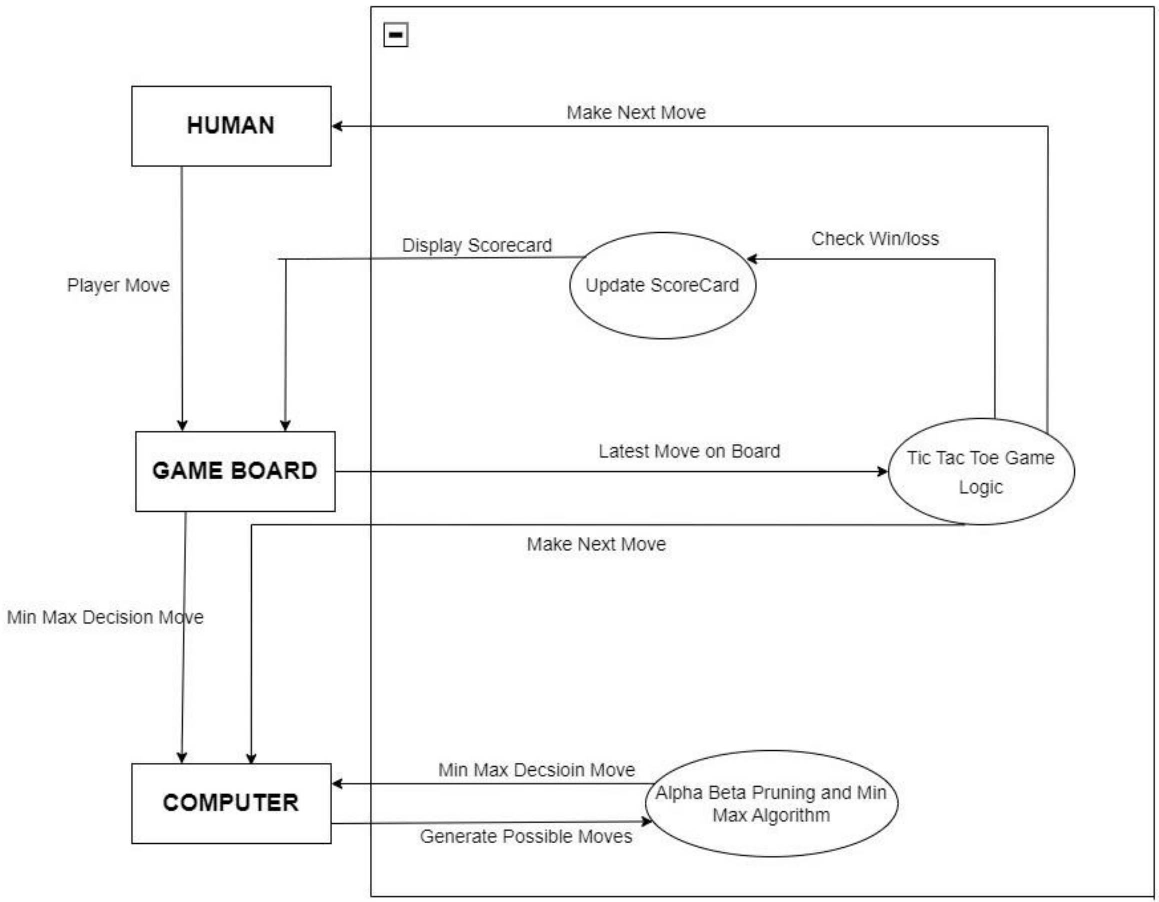

Figure 5.1: Data Flow Diagram

Here is an example of how these components could be connected in a data flow diagram for a 3D tic tac toe game:

In a data flow diagram for a 3D tic-tac-toe game, the interconnected components facilitate the smooth flow of data throughout the system. Players initiate the process by providing input through their moves on the game board. The game logic, a crucial processing element, takes charge of interpreting and processing these moves, subsequently updating the game board to reflect the evolving state of the game. The updated game board then communicates the current game state to the user interface. The user interface serves as the bridge between the digital realm and players, visually presenting the game board and its current configuration.

If the game involves a computer player, it contributes to the data flow by providing input to the game board based on predetermined rules or algorithms. This introduces an additional layer of complexity to the data flow, especially when an automated player is part of the gameplay.

Importantly, this representation is just one possible design for a data flow diagram in a 3D tic-tac-toe game. The specifics of the diagram might vary based on the unique implementation of the game. Throughout the game's progression, the game logic continually processes players' moves, ensuring a dynamic and engaging experience, and it constantly checks for win conditions until the game reaches its conclusion.

UML Diagram

The UML Use-Case Diagram for the 3D Tic-Tac-Toe game graphically represents the system's functionalities from the user's perspective, detailing the interactions between the users (players) and the different use cases within the system. The diagram encompasses various actors and their respective actions, illustrating how they interact with the game's core functionalities.

Actors:

1. **Human Player:** Represents the real-world user who interacts directly with the game.
2. **Computer:** Acts as the automated opponent within the game, controlled by the AI.

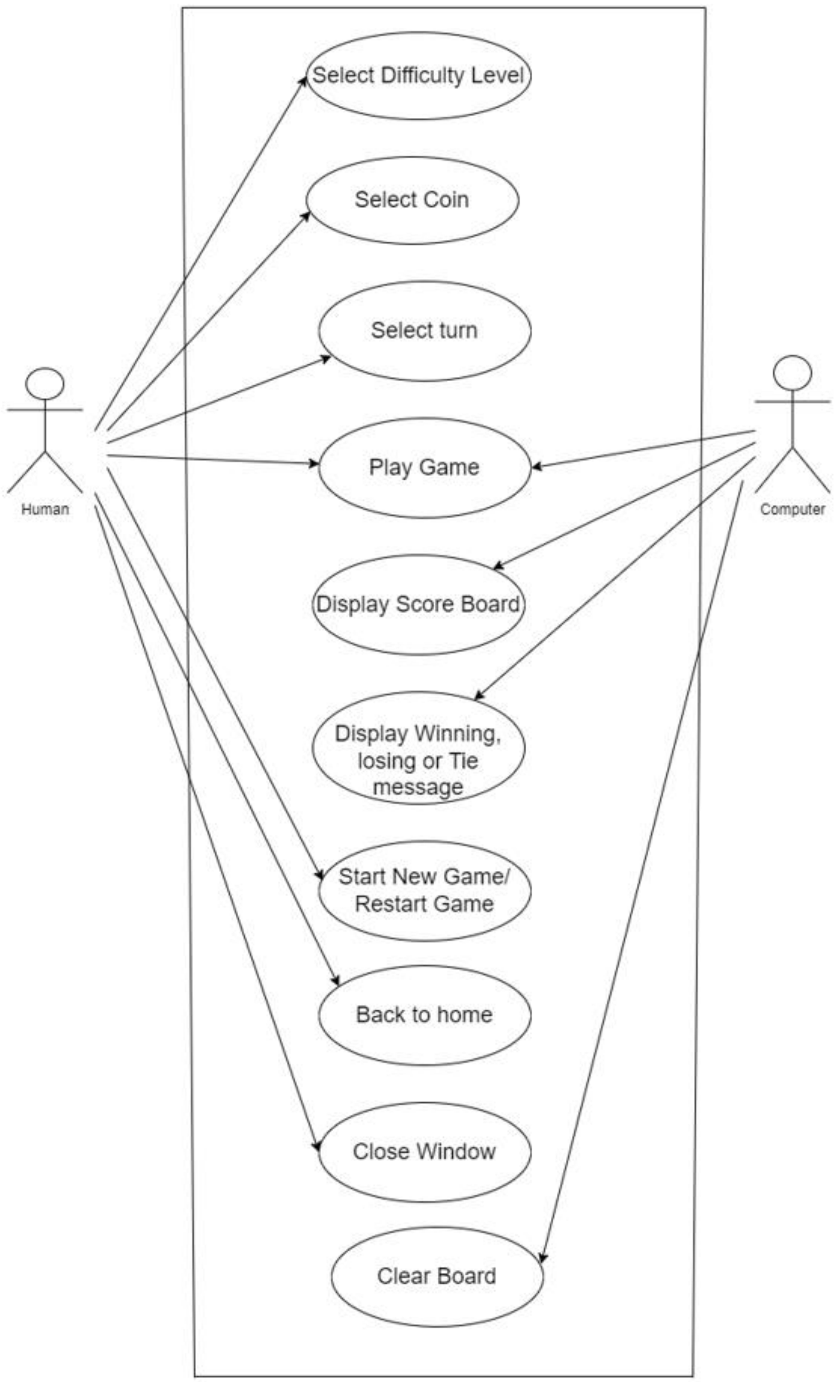

Figure 5.2: Use Case Diagram for 3D Tic-Tac-Toe.

Use Cases:

1. **Select Difficulty Level:** The human player can choose the level of difficulty at which they wish to play against the computer, with options ranging from Easy to Insane.
2. **Select Coin:** The player selects their preferred game piece symbol (X or O), which will be used throughout the game.
3. **Select Turn:** The human player decides who will take the first turn in the game, either themselves or the computer.
4. **Play Game:** This central use case is where the actual gameplay occurs. The human player and the computer alternate turns, inputting moves until the game concludes.
5. **Display Score Board:** The system displays the ongoing score, reflecting the outcomes of each game played during the session.
6. **Display Winning, Losing or Tie Message:** Upon the completion of each game, the system will display a message indicating whether the human player has won, lost, or if the game has ended in a tie.
7. **Start New Game/Restart Game:** The human player can initiate a new game or restart the current game, resetting the board and starting afresh.
8. **Back to Home:** The player has the option to return to the main menu or home screen of the application.
9. **Close Window:** The human player can close the game window, effectively exiting the game.
10. **Clear Board:** This use case involves clearing the game board of all moves, preparing it for a new game.

System's Behavior:

1. The diagram defines a clear pathway for user interactions, starting from the initial game settings to the eventual conclusion of the game.
2. It outlines the dependencies between actions, such as the need to select difficulty, coin, and turn before commencing gameplay.
3. The system is designed to respond to the user's actions, such as displaying the score or concluding the game, based on the moves made by the human player and the computer.

This UML Use-Case Diagram serves as a blueprint for understanding the various ways a user can interact with the 3D Tic-Tac-Toe game. It provides a structured view of the game's requirements and functionalities, essential for the development and implementation of the system. Each use case represents a fundamental building block of the game's design, ensuring that the system caters to all user actions and scenarios within the gameplay environment.

CHAPTER - 6
INPUT DESIGN

The 3D Tic Tac Toe game, presented in a JFrame titled "Group-19" and sized at 600x800 pixels, offers an engaging user interface designed for ease of use and optimal player interaction. The game utilizes a CardLayout to seamlessly switch between different panels — the Title Panel, Options Panel, and Game Board Panel. The Title Panel serves as the welcoming screen, featuring a stylized game title, an image placeholder, likely for a game logo, and a "Play Game" button that transitions the user to the Options Panel. This panel is crucial for setting up the game, offering players a variety of choices through radio buttons, including difficulty levels (Easy, Difficult, Insane), piece selection (X or O), and the option to decide who starts the game first (Human or CPU).

The core gameplay takes place in the Game Board Panel, where a 4x4x4 grid of custom JButton elements forms the interactive game board. Each button represents a potential move, and their functionality is intricately tied to the game's logic through ActionListeners. Enhancing the player's experience, this panel also includes a text panel with a dynamic game board title and a real-time scorecard. Additional control buttons for starting new games, restarting, and returning to settings are thoughtfully included for comprehensive game control. Custom classes like BoardButton and BoardPanel add a unique touch, with the latter responsible for graphically rendering the 3D game board, including features like lines and winning combination highlights.

CHAPTER - 7

OUTPUT DESIGN AND RESULT ANALYSIS

The home page is simple yet effective, with a welcoming title, an image that probably represents Tic Tac Toe or related imagery, and a clear call to action with the "Play Game" button. The user is immediately presented with the game theme and has a straightforward option to start playing, which is good for user engagement.

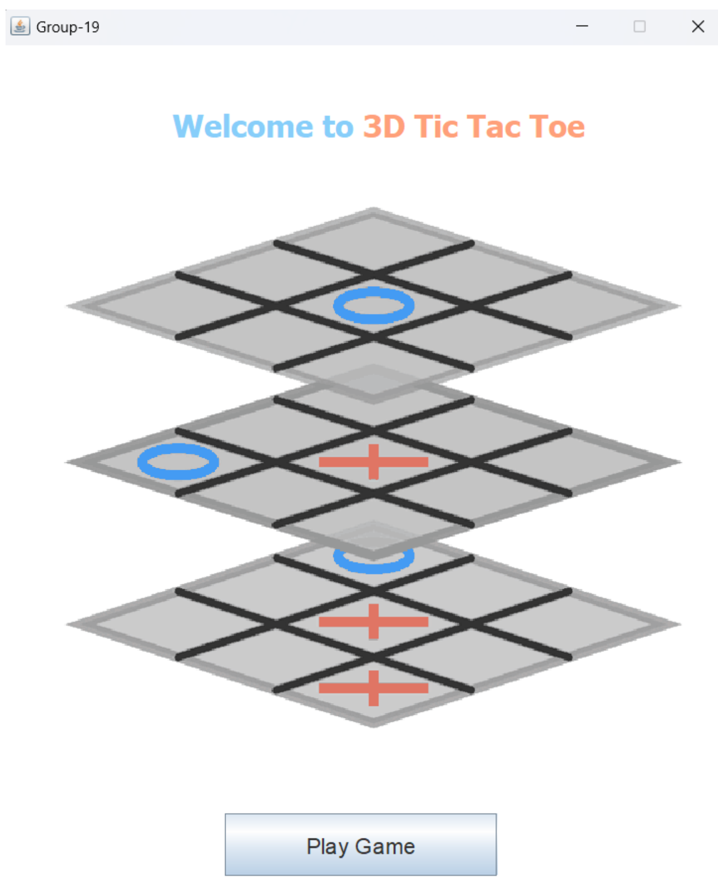

Figure 7.1: Home Screen Interface

The ‘Game Settings’ page appears after the user clicks the "Play Game" button on the home page. The settings page appears to be functional and straightforward, offering essential customization options without overwhelming the user. The clear categorization of settings and the final call to action with the "Start Game" button sets a smooth transition into the actual gameplay.

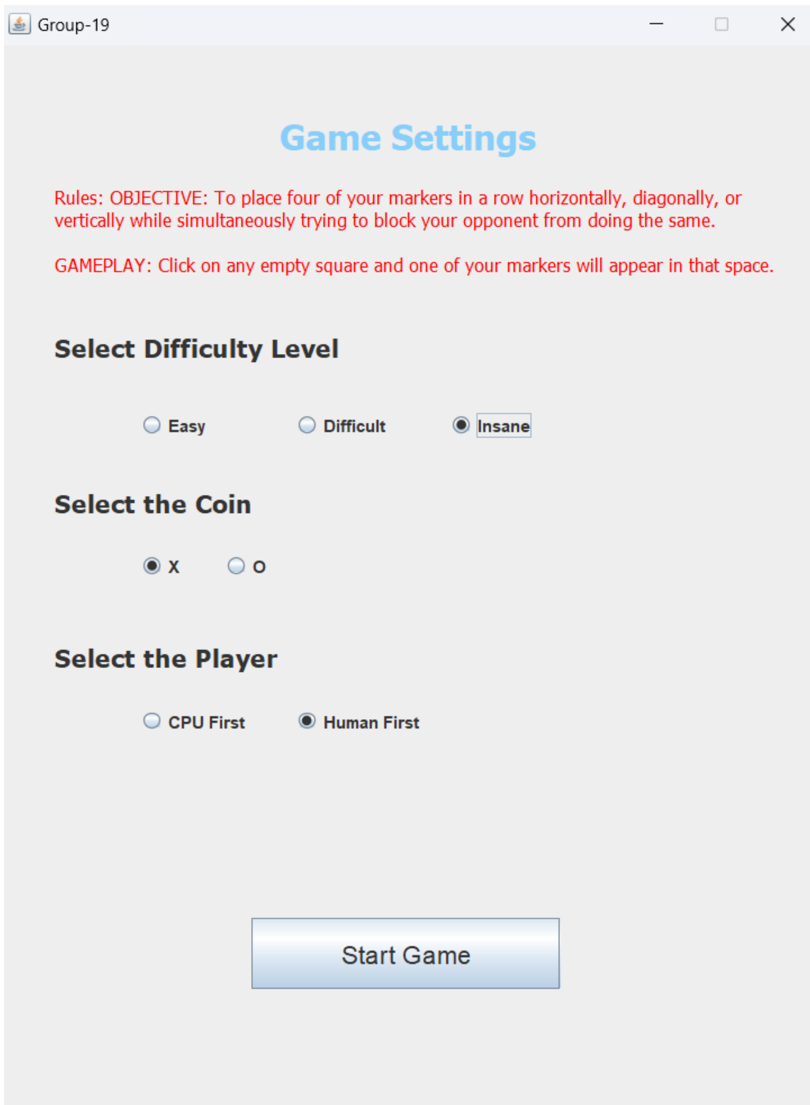

Figure 7.2: Game Settings Interface

The game board page likely features a 3D Tic Tac Toe grid, which is the central element of this page. This 3D aspect of the grid adds a unique twist to the traditional Tic Tac Toe game, making it visually appealing and intriguing. Additional buttons or options are available for actions like starting a new game or returning to the settings, ensuring the player can easily navigate through different stages of the game.

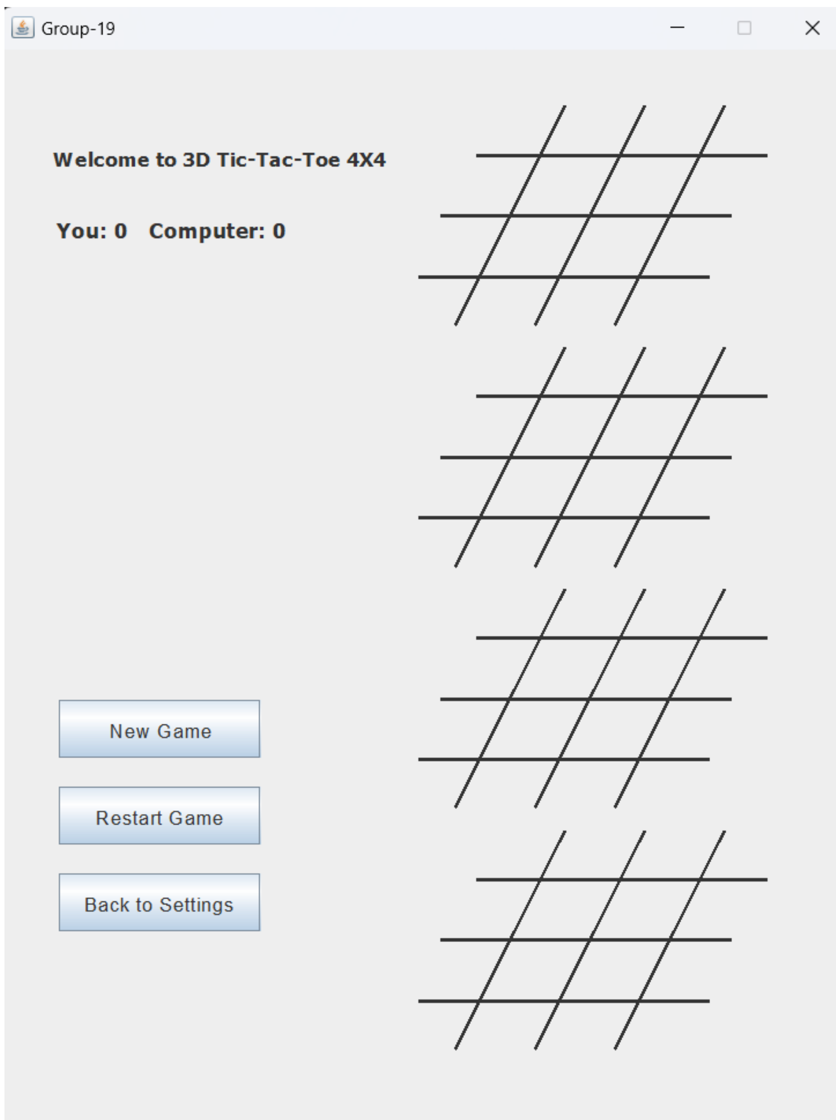

Figure 7.3: 3D Tic Tac Toe Game Board Interface

When a player or the computer wins, the game immediately communicates this outcome through a popup message. This straightforward approach ensures that players are clearly informed about the game's result without any confusion.

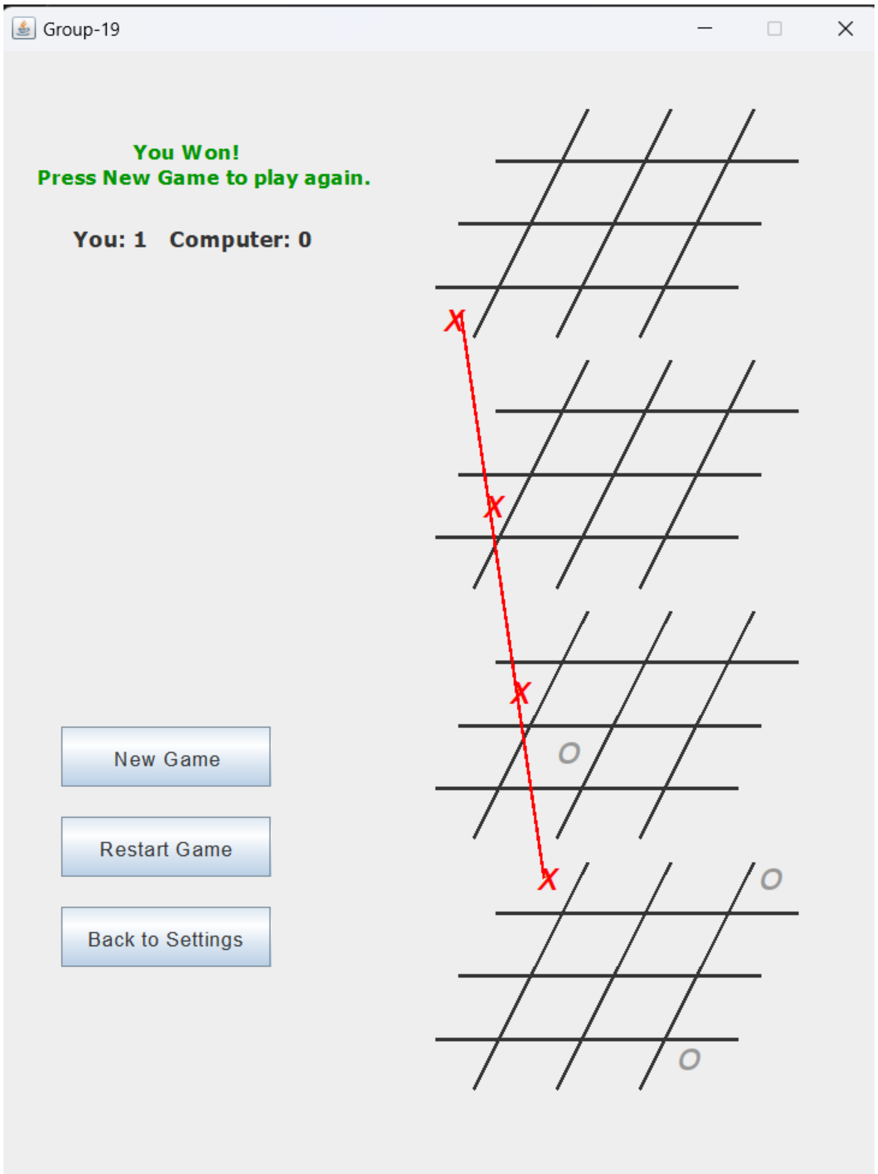

Figure 7.4 (a): Scorecard and Wining message Interface

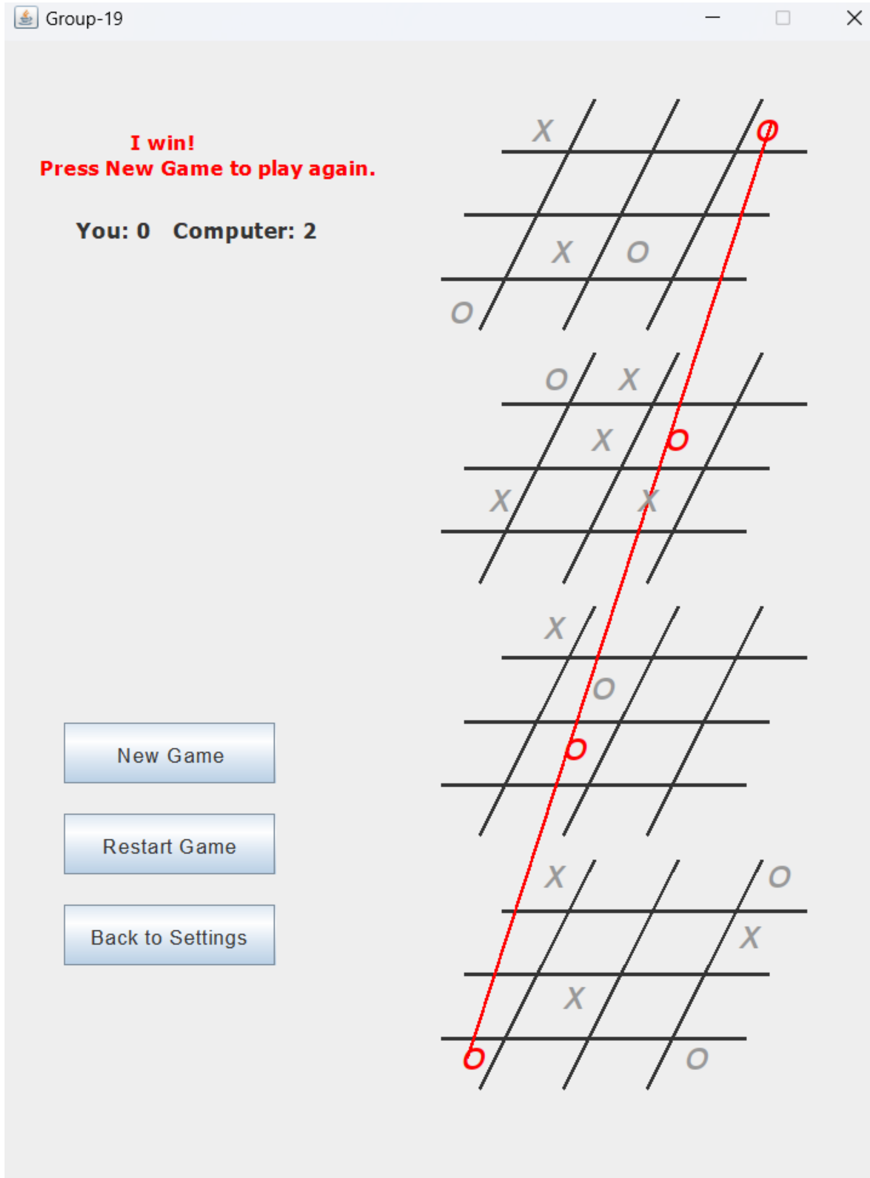

Figure 7.4 (b): Scorecard and Wining message Interface

When the "Restart" button is clicked in the 3D Tic Tac Toe game, it likely offers a prompt for confirmation, preventing accidental restarts and ensuring a thoughtful user experience. Restarting the game provides a fresh start, allowing players to try new strategies or correct previous mistakes, enhancing the replay value of the game.

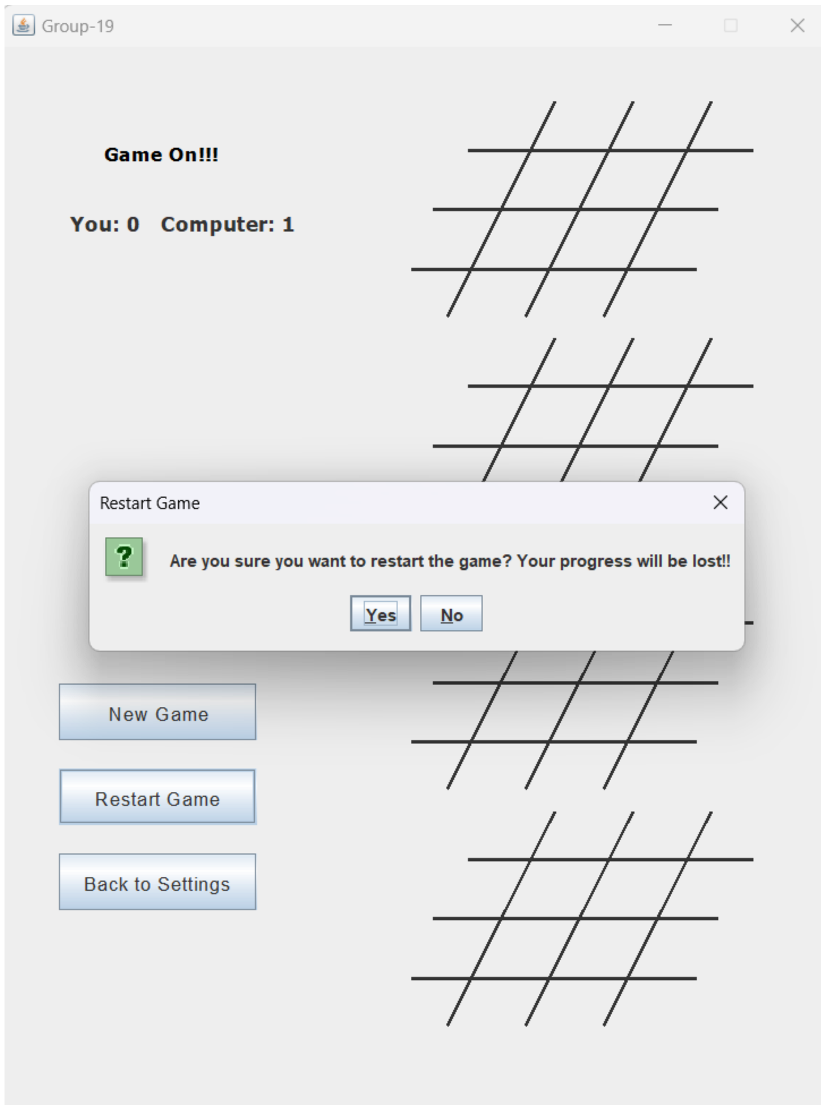

Figure 7.5: Dialogue prompt Interface

CHAPTER - 8

CONCLUSION

In conclusion, the development of a 3D Tic Tac Toe game involves a meticulously planned and executed project, integrating diverse elements of game design, programming, and user interface considerations. The project's foundation lies in the initialization of a 4x4x4 3D game board, where players engage in strategic moves, guided by a well-defined set of rules. The alternating turns, validated inputs, and win/tie conditions establish a robust gameplay structure.

In essence, the 3D Tic Tac Toe project encapsulates a comprehensive approach to game development, blending technical precision with user-centric design. The project's success hinges on the harmonious integration of programming logic, graphical representation, and user engagement, resulting in an entertaining and immersive gaming experience.