Chutes & Ladders

Final Report

Team Triple Threat

(Manny Uzoma, William Garlington, Steve Deibert)

Course: CPSC 224 - Software Development

I. Introduction and Project Description

This document serves as a summary of our project's progress and technical details. It aims to provide stakeholders, team members, and other interested parties with a comprehensive overview of our Chutes and Ladders game implementation. From describing the purpose of the project to detailing the key features and technical aspects, this document aims to encapsulate our engineering efforts and progress throughout the development process. Whether for reference, documentation, or review purposes, this document offers valuable insights into our project's journey and achievements.

II. Team Members - Bios and Project Roles

William Garlington studies computer science at Gonzaga University, with an interest in data analytics, machine learning, and database systems. His previous projects have included a calorie tracking program utilizing a persistent user database, an agent based simulation of plant growth, and an exploration into the connection between tobacco use and happiness. William's skills include C++, Java, Python, Assembly, and NetLogo. In this project, his contributions included designing and implementing the player scoreboard, early prototypes for the game board, the dice value display, as well as some improvements to the player rendering system made by Steve Deibert.

Steve Deibert is a computer science student interested in application development, data analytics, and game design. His prior projects have included Apple Health data analytics, a customized version of farkle, and multiple mods for games. Steve's skills include C/C++, Python, Lua, Java, Rust, and Unreal. For this project his responsibilities included developing the chute and ladder generation and their graphical representation, game start and end, and some of the gameboard generation, and representation.

Manny Uzoma is a computer science student interested in application development as well. His prior project has been data analysis on spotify data and weather, and checking for correlation and also some UI/UX design in other applications. Manny's skills include C/C++, and Python. For this project, his main responsibilities were creating documentation, presentation, and creating unit tests.

III. Project Requirements

This section outlines the specific requirements for our Chutes and Ladders game project. These requirements serve as the foundation for our design and development efforts, guiding the implementation of key features and functionalities. From gameplay mechanics to user interface elements, each requirement contributes to the overall success of the project. Below are the detailed requirements categorized into functional and non-functional aspects:

Functional Requirements:

1. **Gameplay Mechanics:**
    - Players must be able to roll dice to determine their moves.
    - Movement on the game board should be based on the outcome of the dice roll.

    - Players should move up ladders and down chutes based on their position.
2. **Player Interaction:**
    - Multiple players should be supported, with each taking turns sequentially.
    - Players must be able to interact with the game through a user-friendly interface.
    - The game should provide feedback and prompts to guide players through their turns.
3. **Win Condition:**
    - The game should detect when a player reaches the last square (100) and declare them the winner.
    - Upon reaching the last square, the game should prompt players to restart or exit.

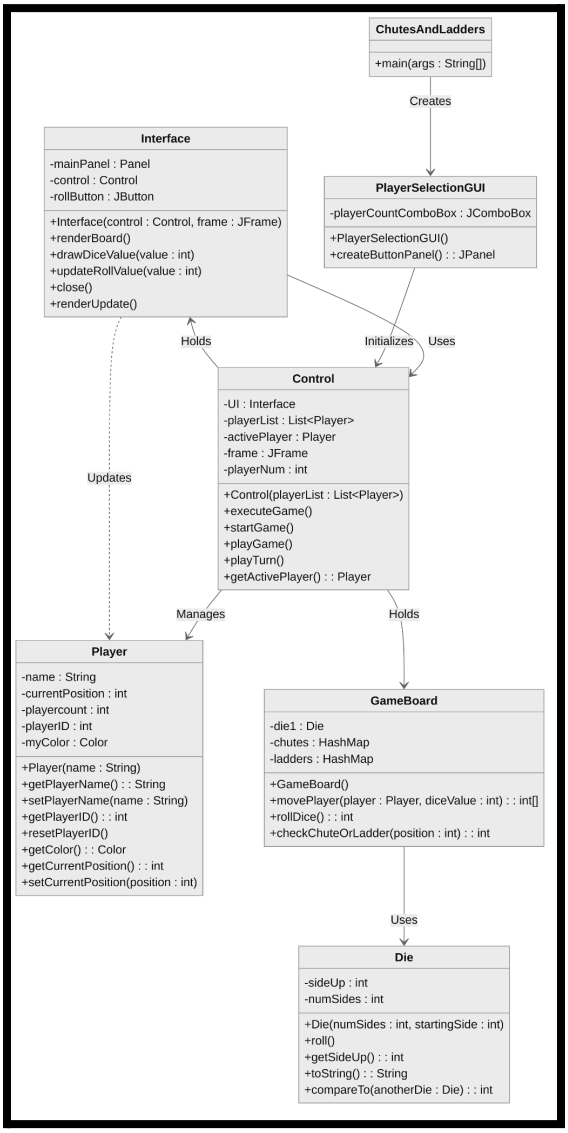
Non-functional Requirements:

1. **Performance:**
    - The game interface should respond promptly to user inputs, ensuring smooth gameplay.
    - Resource usage (CPU, memory) should be optimized to run efficiently on various devices.
2. **User Interface:**
    - The user interface should be intuitive and visually appealing, enhancing the overall gaming experience.
    - UI elements should be responsive and adapt to different screen sizes and resolutions.
3. **Reliability:**
    - The game should be stable and reliable, minimizing crashes or unexpected behavior during gameplay.
    - Error handling mechanisms should be in place to gracefully handle unforeseen situations.

By adhering to these requirements, we aim to deliver a high-quality Chutes and Ladders game that meets the expectations of our stakeholders and provides an enjoyable gaming experience for players of all ages.

IV. Solution Approach

Our solution comprises several major components that work together seamlessly to create an engaging gaming experience. At its core is the Model-View-Controller (MVC) architecture, which forms the backbone of our system. The Control component serves as the central hub, managing the game flow and coordinating player actions. The GameBoard component embodies the virtual game space, complete with chutes and ladders, providing the foundation for gameplay mechanics. The Interface component is responsible for the graphical user interface (GUI), rendering the game board and player representations, ensuring an immersive visual experience. Meanwhile, the Player component encapsulates the attributes and behaviors of each player participating in the game.



In terms of functionality, our solution boasts several compelling Game/UI features. It offers a graphical representation of the game board, vividly illustrating the chutes and ladders that players encounter throughout their journey. Players' positions are dynamically rendered on the board, enabling them to visually track their progress and strategize accordingly. A roll button facilitates player turns, injecting an element of chance and anticipation into each move. Moreover, the system seamlessly handles movement along chutes and ladders, enhancing the excitement and unpredictability of the gameplay experience. Together, these components and features form a cohesive and entertaining solution that promises hours of enjoyment for players of all ages.

V. Test Plan

Our solution's validation process involves a comprehensive test plan implemented with JUnit tests, ensuring compliance with both functional and non-functional requirements. Functionally, tests confirm that players start at position one, die rolls fall within the valid range, and movement remains constrained within the game board limits. While the direct testing of the roll button functionality is absent, its validity is indirectly implied through die roll tests. Non-functional aspects such as intuitive UI design, responsiveness, flexibility for modifications, clear instructions, and accessibility considerations are acknowledged but not directly tested within the provided code. Overall, the tests assure the robustness and alignment of our solution with project objectives, with potential for additional testing to evaluate UI design, responsiveness, and accessibility in greater depth.

VI. Project Implementation Description

Our team has made significant progress in implementing an MVC system for our Chutes & Ladders game. Here's an overview of the current progress at each stage of the implementation:

1. **Game Mechanics:**
    - Implemented the core game logic, including dice rolling, player movement, and interaction with the chutes and ladders.
    - Completed the win condition detection to determine when a player reaches the last square and declare them the winner.
2. **Player Interaction:**
    - Developed a user-friendly interface to facilitate player interaction with the game.
    - Implemented turn-based gameplay mechanics to allow multiple players to take turns sequentially.
3. **User Interface:**
    - Designed and implemented the GUI using Java Swing components.
    - Created UI elements such as the game board, player tokens, dice rolling button, and a display for the rolled dice value.
    - Created system to visually indicate which player is currently active.
4. **Restart and Exit Functionality:**
    - Implemented functionality to prompt players to restart or exit the game upon reaching the win condition. This is accomplished using a pop-up window created after the main window is destroyed.
5. **Testing and Debugging:**
    - Conducted thorough debugging at all stages in development using a combination of console outputs and manual verification of function.
6. **User Feedback and Refinement:**
    - Incorporated user suggestions and made iterative improvements to enhance the overall gaming experience.

Github

VII. Future Work

When considering future work, several systems could be streamlined and many more ideas could be implemented. One issue that affected development was the structure of the program being unfavorable for thorough unit testing. Altering the program structure to remedy

this problem would be a major help to all future development and maintenance. Several features thought of during the design process like changing the difficulty of the game by altering chute/ladder generation or implementing consumable items to add a sense of surprise to the gameplay experience would drastically improve the replay value if put in. Another direction of future work would be improving the UI system to support multiple window sizes and potentially dynamic resizing. In this vein, the rendering system could also be overhauled to move away from pixel specific drawing.