



Game UI Design and Implementation Overview

XU Yuan 24036523d

KWAN Chun Yin 24123102d

TSANG Man Ho 24028571d

TSE Hung Lap 24028633d



01 User Commands and Input Method

User Command Navigation

Number-Based Navigation

In our game design, we implemented a command-line interface (CLI) that primarily relies on numerical inputs for navigation.

This method allows users to quickly access different functionalities by simply entering numbers, streamlining the interaction process and minimizing the learning curve for new players.



Advantages and Disadvantages



Advantages of CLI

The CLI approach offers several advantages, such as faster input times for experienced users who can navigate using keys rather than a mouse. Additionally, it consumes less system resources than GUI alternatives, making it suitable for low-spec environments. This efficiency is particularly beneficial for users who prefer streamlined interactions.



Disadvantages of CLI

When compared to graphical user interfaces (GUIs), where users can often click buttons for navigation, CLI input handling requires more elaborate error management. Users may accidentally input invalid commands or misspell names, leading to potential issues in gameplay. Thus, our design emphasizes the importance of comprehensive error feedback to guide users effectively.



02 Game Status Representation

Game Status Representation

Simplified Game Status

Instead of a traditional square-shaped game board, we opted for a simplified list format to represent the game status.



Advantages and Disadvantages of the Chosen Format

Ease of Implementation

One of the primary advantages of using a list format for game status representation is the ease of implementation. Developers can quickly code the logic for displaying the game status without the need for intricate graphic design, reducing the overall development time.

1

2

Screen Size Adaptability

The simplified format is particularly beneficial for devices with varying screen sizes. Whether on a mobile phone or a tablet, players can easily view the game status without worrying about scaling issues or visual clutter, making the game more universally playable.

Loss of Game Atmosphere

Despite the practical advantages, the stripped-down version of the game status representation may lead to a loss of the Monopoly atmosphere. The visual appeal of the game board is integral to the player's experience, and its absence could make the game feel more like a traditional list-based application rather than the engaging board game it is.

3



03 Error Handling Mechanism

Understanding User Input Errors

Utilizing While Loops for Error Management

In our error handling mechanism, while loops play a crucial role in ensuring that users are prompted again if their input is incorrect. This allows for a seamless correction process, where users can refine their inputs without having to restart the entire operation. By maintaining this loop, we enhance the user experience significantly.



Leveraging Past Inputs for Better Accuracy

The Command Line Interface (CLI) retains a record of previous user inputs, which can be extremely beneficial for error correction. Users can look back at their past attempts and modify their inputs accordingly, reducing the likelihood of repeated errors. This feature not only saves time but also encourages users to learn from their mistakes.





04 Lessons Learned

Lessons from Unit Testing

Early Testing Integration

Integrating unit testing early in the development cycle showcased its benefits in catching bugs and issues sooner. This proactive approach reduced the overall time spent on debugging later in the project.

Test Coverage Importance

Maintaining high test coverage was highlighted as a key factor in ensuring the stability of the codebase. Comprehensive tests help catch edge cases and reduce the likelihood of regressions.

Automated Testing Efficiency

The use of automated testing tools proved to be efficient for repetitive testing tasks. Automation not only saves time but also ensures consistency in the testing process.

Test-Driven Development

Adopting test-driven development (TDD) practices led to better-designed code. Writing tests before implementation encouraged developers to think critically about the functionality and design of the software.



Thank You