

User Manual

for

Monopoly

Prepared by:

MAN Furui (19081789d)
LIU Sicheng (19079181d)
WANG Meng (19078543d)
XING Shiji (19079008d)

The Hong Kong Polytechnic University
Software Engineering
Course Project
Prof. Yu PEI

Hong Kong, 18 November 2021

Contents

1	Overview	2
2	Deployment	3
2.1	Environment	3
2.2	Dependencies	3
2.3	Building and Execution	3
3	Gamplay	5
3.1	Game Rules	5
3.2	User Input	5
3.3	Prompt	6
3.3.1	Color Notation	6
3.3.2	Prompt Sample	7

Chapter 1

Overview

This software is an Java implementation of the game Monopoly, supporting multi-color prompting and player interactions.

In this document, the deployment procedures and gameplay hints are provided. User are highly advised to read sections on User Input and Prompt to be aware of the software's workflow.

Also you can refer to the Software Requirements Specification Document for more details on Monopoly's game rule, and API document / API Revision document on the methodologies of the software.

Chapter 2

Deployment

2.1 Environment

This application is developed based on Java JDK 11 with only one dependency JUnit 5.7.0.

2.2 Dependencies

To deploy the project, first open the directory with IntelliJ IDEA and then install the dependency by downloading from File - Project Structure or referring to IntelliJ IDEA official guide.

An alternative that would bypass dependency installation is to `/test` folder as excluded so that the JUnit is not involved.

2.3 Building and Execution

After installing the dependency, build the project and execute the `main` method in the following Java Class:

```
hk.edu.polyu.comp.comp3211.monopoly.Main.
```

You should be able to see the following welcome message in the console.

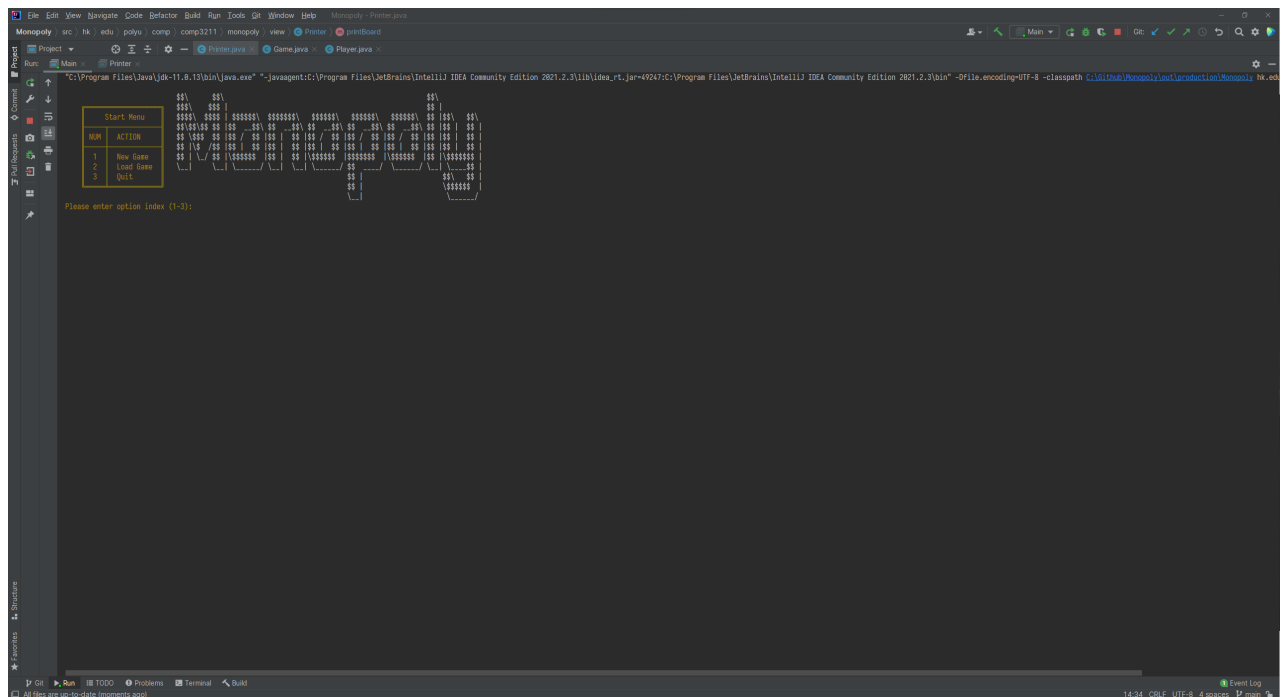


Figure 2.1: Welcome Message Demonstration

Chapter 3

Gamplay

3.1 Game Rules

The game rules are covered in both the Requirement Specification Document and the Course Project Description Document. Details will not be covered here for simplicity.

3.2 User Input

The game accepts q or Q or quit every time when it requires user input, and by doing so you interrupt from the current stage of the process and return to the previous stage. Note that no record will be saved from the interrupt and thus no un-do is supported.

Be aware that the quit command are not supported for ALL cases. For example, when player interacts with the property squares and opt to buy the not yet owned property just arrived in, quit command at this stage will not work, due to the confusing between quit from the current interactions and quit from the current game. However, the quit option is active after this turn of the player.

The game will prompt an warning (in red) to indicate an invalid input from the user, which indicates more specifically about the format of accepting input. Then it will prompt again and repeat the process infinitely until a valid input is detected.

When loading a game from a local game save, the index rather than complete game save file is used to simplify user input. For more details, observe the following loading game save demonstration.

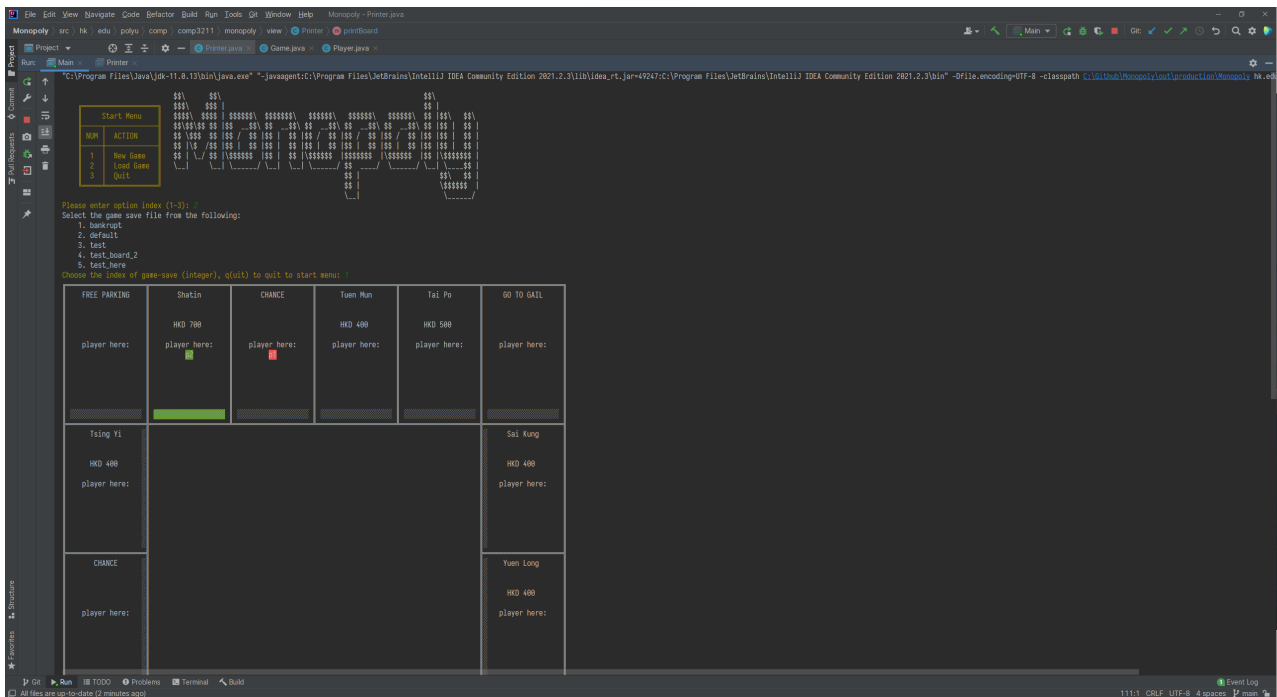


Figure 3.1: Load Game Save Demonstration

3.3 Prompt

3.3.1 Color Notation

This command-line game is colorized to provide a better user experience. Normally it conforms to the following colorization notation:

- White(Black):
The color implies text responsible for control flow, which is vital for the game to run.
- Red
This color implies warning and errors. User should attend to what the warning states and change user input by the instruction accordingly.
- Yellow
This color implies the demand for user inputs. In most cases, user input follows directly after the characters in yellow.
- Green
This color implies user input, which conforms to the default of IntelliJ IDEA settings.
- Blue
This color implies information and especially mark for those that are active at the time of execution. For example, prompts for player name will be in blue to highlight the current active player, followed by the actions taken or needed, which are colored by their use respectively.

There are of course colorization in the board. When representing a player owns a specific property, the color of which will be tuned to the same background color, where each color is distinct to the user to improve understandability.

3.3.2 Prompt Sample

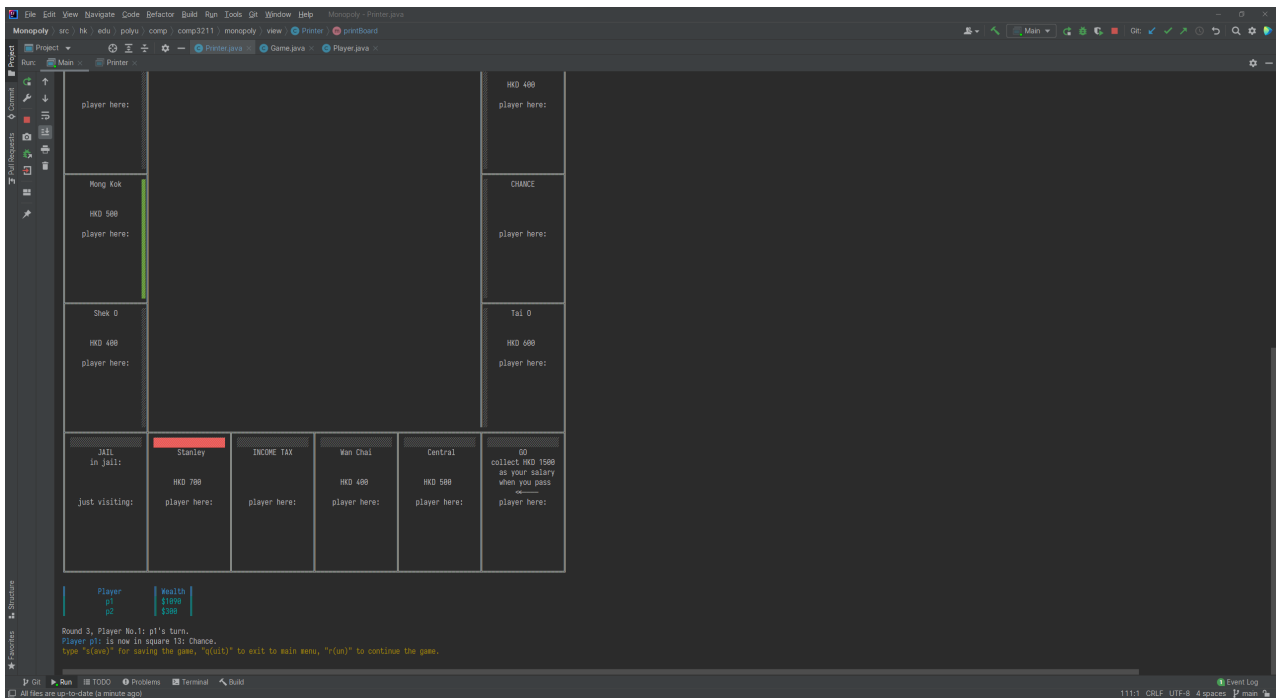


Figure 3.2: Colorization Rule Demonstration