Project Report

Group 5

COMP2021 Object Oriented Programming Fall 2017

CHENG Yiran(16098521d)

WANG Bokang(16097234d)

JIANG Yuxin(16096336d)

26-11-2017

1. Introduction

This document describes the design and implementation of the Monopoly game of group 5. The project is part of the course COMP2021 Object-Oriented Programming at PolyU. The following sections describe the requirements that were implemented and the design decisions taken. The last section describes the available commands in the game.

2. The Monopoly Game

Monopoly is a game where players roll two six-sided dice to move around the game-board buying and trading properties. Players collect rent from their opponents, with the goal being to drive other player into bankruptcy or having the most money at the end of the game. Money can also be gained or lost through Chance and tax squares; players can end up in jail, which they cannot move from until they have throwed doubles or paid fine. The game ends after 100 rounds, or when there is only one player left at the end of a round. The winner is the player with the most money when the game ends. Ties (multiple winners) are possible.

2.1 Requirements

Req-1. This game provides a command line user interface for displaying game information and asking for user input.

Software elements:

Type	Name	Description
Interface	UserInterface	The connection between game process class (***class** Game*) and user interface class *(**class** CLI* and ***class** GUI)*. Game process display or ask for user input though this interface.
Class	CLI (implements UserInterface)	This class is the main class of implementing CLI (command-line interface) input and output.
Method	methods in CLI	The methods inside CLI implements the game functions which need IO operations. (e.g. *showPlayerInfo*, *askNumberOfPlayers*, *displayMessage*).
Method	Game.report	This method implements the function of report in command-line since there is no need to report when the game is run in GUI.
Variable	Game.UI	This variable records the user interface of the current game.

Req-2. The game supports both human players and computer players. After game starts, user can set any player to auto mode by choose the option.

Software elements:

Type	Name	Description
Method	Player.setAuto	This method set the player to an auto player
Method	Player.isAuto	This method returns a Boolean value representing whether the player is a computer player.
Method	Methods in class Game	The game operation methods (e.g. *askPayFine*, *runGame*) supports the rules for auto player. These methods will call get random number methods instead of asking for user input when it checks that the current player is an auto player.

Req-3. At the beginning of a player's turn, the game will ask the player whether she wants to continue, report, auto, retire, save, or load. Moreover, in GUI mode, it will ask whether the user wants to continue, auto or retire at the beginning of each turn. Save and load are available during the entire process and report is unneeded for GUI.

Software elements:

Type	Name	Description
Method	UserInterface. step1ChooseOperation (implements in CLI & GUI)	This method ask for user input in CLI or check user operation in GUI at the beginning of each player's turn and returns the choice of the player.
Variable	RoundStep (in Game.runGame)	It saves the current step of the turn. It begins with 0 when a turn begins. Method can determine whether it is the beginning of the turn by this variable.
Method	Game.runGame	This method implements the main process of the game. In each turn, it will check the value of current round step (roundStep), if it is the beginning of this turn, it will call setp1ChooseOperation method.

Req-4. Upon continue, the player takes her turn following the rules of the game.

Software elements:

Type	Name	Description
Method	Game.runGame	The rule of the game is basically controlled by this method. If it received continue command value, it will continue to next step (mostly rolling dices).
Variable	RoundStep (in Game.runGame)	It saves the current step of the turn. Method can control the game stage by checking this variable.

Req-5. The game should ask for the human player's input, if there are multiple actions from which the player can choose;

Software elements:

Type	Name	Description
Method	askPayFine (in Game)	This method checks whether the current player is a computer player and call different methods to get choice of pay fine immediately or not.
Method	askForBuying (in UserInterface)	This method asks for and returns user's choice (buy or not buy) when he/she landed on an empty property.

Method	askPayFine (in UserInterface)	This method asks for and returns user's choice (pay fine immediately or not) when he/she is in jail.
Method	askNewOrLoadGame (in UserInterface)	This method asks for and returns user's choice (new game or load game) at the very beginning of the game.
Method	askNumberOfPlayers (in UserInterface)	This method asks for and returns the number of players when it begins a new game.
Method	askPlayerName (in UserInterface)	This method asks for and returns the name of a new player at the beginning of a new game.

Req-6. Upon report, the game prints out the game board and each player's location on the board.

Software elements:

Type	Name	Description
Method	Game.report	This method prints out the game board with player information on it. It can display most information of the game.

Req-7. Upon auto, the game should replace a human player with a computer player. A computer player always continues until the end of the game and makes other decisions randomly.

Software elements:

Type	Name	Description
Method	Player.randomChoice	This method returns a random Boolean value represent a random choice of the computer player.
Method	Player.setAuto	This method set the player to an auto player
Method	Player.isAuto	This method returns a Boolean value representing whether the player is a computer player.
Method	Player.askPayFine	This method checks whether the current player is a computer player. If it is, it returns random choice by calling random choice method.
Method	Player.runGame	This method supports the rules for auto player. And will call different methods from human player's methods when it is an auto player.

Req-8. Upon retire, the game should remove the player from the list of players and make all her property unowned.

Software elements:

Type	Name	Description
Method	Player.quitGame	This method simply sets a player to be offline / retired.
Method	Game.checkPlayerOut	This method checks whether a player is out of game. If so, it will check his/her properties and set them to free.
Method	Game.setNextPlayer	This method can set the current player to next player after the current player retired and prepare for next turn.

Req-9. Upon choosing save option, the game will save the current status of the game by storing compulsory variables in Game class to a specified file with its file path.

Software elements:

Type	Name	Description
Method	Game.saveGame	This method will call user interface to ask user for a file path and save the relative objects of game to the file. It will return true if success.
Method	SaveGame (in UserInterface)	This method ask user for a file path and returns a file object. It works differently in GUI and CLI.

Req-10. Upon load and a path to a file, a previously saved game will loaded from the specified file which user inputs its path and continued.

Software elements:

Type	Name	Description
Method	Game.loadGame	This method will call user interface to ask user for a file path and load the relative objects of the game from the file to the current game object. It will return true if success.
Method	loadGame (in UserInterface)	This method ask user for a file path and returns a file object.

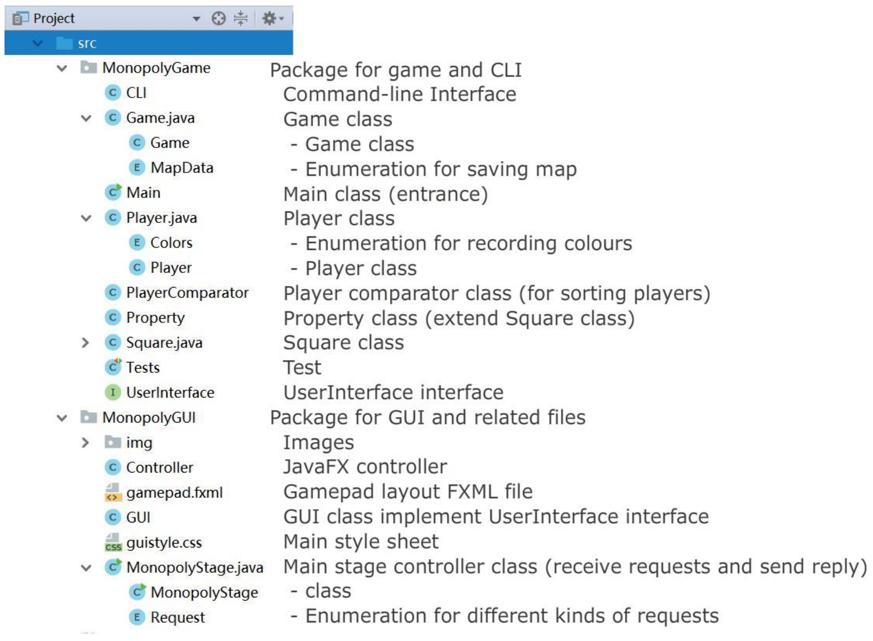
Bon-2. Graphical user interface, with full support to all user interactions.

Software elements:

Type	Name	Description
Interface	UserInterface	GUI class is based on UserInterface interface. This can make sure the GUI is compatible with the Game class.
Class	GUI (implements UserInterface)	This class implements the interface UserInterface. Game can get connected with GUI through this class.
Class	MonopolyStage	This class loaded the pre-designed interface from fxml file and initialized the root stage. It can also check the request of the game through GUI class and get user operation through Controller class.
Class	Controller	This class is for controlling the widgets of GUI and run specific methods when user operates.
Array	requests (in MonopolyStage)	This array is for recording requests from game. When game needs any inputs or user operations, it will send data to request. When receiving requests, methods can conduct different operations on GUI.
Array	replies (in GUI)	This array records the replies form the GUI. When GUI replies, data will be set in this array for returning to game to use.
Method	checkRequests (in MonopolyStage)	This method checks the requests list regularly. If there are any requests, it will call specific method to handle and send reply.

2.2 Design

Give an overview of the design of the game and describe in general terms how different components fit together. Feel free to elaborate on design patterns used, class diagrams, or anything else that might help others understand the design.



```
Project
  src
    MonopolyGame              Package for game and CLI
      CLI                       Command-line Interface
      Game.java                 Game class
        Game                     - Game class
        MapData                  - Enumeration for saving map
      Main                      Main class (entrance)
      Player.java               Player class
        Colors                   - Enumeration for recording colours
        Player                   - Player class
      PlayerComparator          Player comparator class (for sorting players)
      Property                  Property class (extend Square class)
      Square.java               Square class
      Tests                     Test
      UserInterface             UserInterface interface
    MonopolyGUI               Package for GUI and related files
      img                       Images
      Controller                JavaFX controller
      gamepad.fxml              Gamepad layout FXML file
      GUI                       GUI class implement UserInterface interface
      guistyle.css              Main style sheet
      MonopolyStage.java        Main stage controller class (receive requests and send reply)
        MonopolyStage            - class
        Request                  - Enumeration for different kinds of requests
```

Image - File structure reference

Components

According to the overview of file structure, this project can be separated into some components.

1. **Game components**

   The whole game component is mainly controlled by Game class.

   1.1 Game rules:

   The main game rules and process are controlled by runGame() method.

   1.2 Game data:

   The game data is initialized by starting a new game or loading a game. It includes squares, players and some status and variables.

   Squares record basic information (e.g. ascription, position, type) of each square and provide modification methods to their information.

   Players record each player's static information (e.g. name, number) and dynamic status (e.g. money, in jail). And class Player provide methods to modify the player information.

   1.3 Other operations:

   Other operations such as save and display report. They are controlled by game class.

   The relationship between them are like the diagram below.

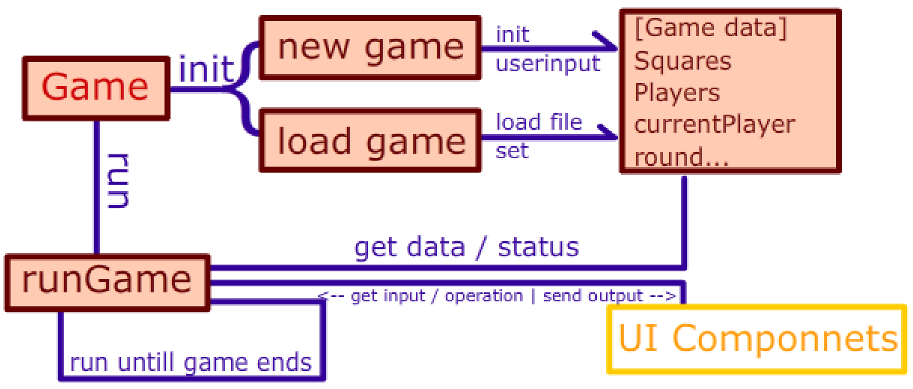


Image - Game components

2. UI Components

The UI components include user input / operation methods and the game input request and output display methods.

2.1 Command-line interface (CLI) component

The output of CLI can simply implemented by *System.out.print*. The input of CLI is provided by the scanner of given input stream. These data can be send in from / out of game part through UserInterface interface.

2.2 Graphical user interface component

The GUI part is a little bit more complicated. In terms of sending data, it is same as CLI. Differently, aside from GUI implements UserInterface interface, it has a controller and a main JavaFX processor. The diagram below displays how they work together.

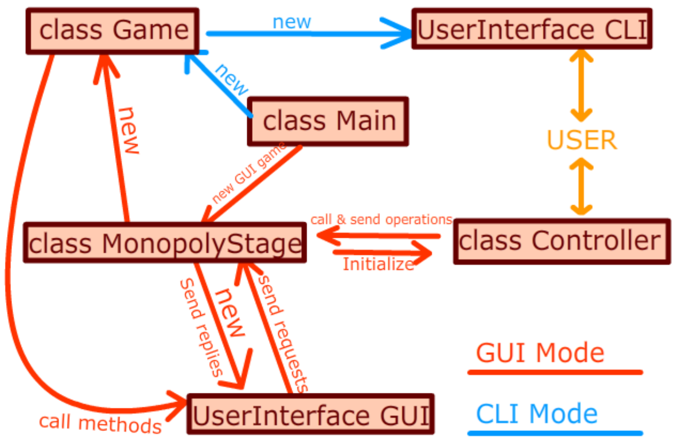


Image – UI & Game cooperation

Concurrency

Overview

As the project needs both GUI and background game process, we design the Game class implements runnable. For CLI mode, there is no need of concurrency supplied. For GUI mode, we set the JavaFX as the main thread and start a new game thread.

Connections

1. **Overview**

   The project needs connections between input – manage, manage – output, UI – input, UI – output.

2. **Connect front UI and background game**

   To connect 2 threads (game and GUI), we design a request-reply method to make sure the process of game and GUI are on the same step. Game call interface to send request to JavaFX GUI and wait. Upon user operates, GUI replies the operation value or data to interface. After that, interface gets reply and returns it to game and game thread continues. The diagram below shows the process of our request-reply method between interface and JavaFX processor.

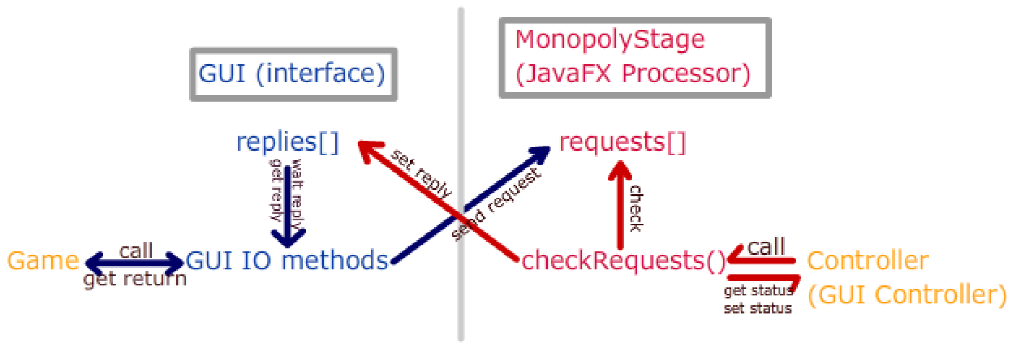


Image – Request-reply connection method

GUI Design tools

1. Design tools:

   Layout: SceneBuilder

   Style: JavaFX CSS

   Image design: Flash

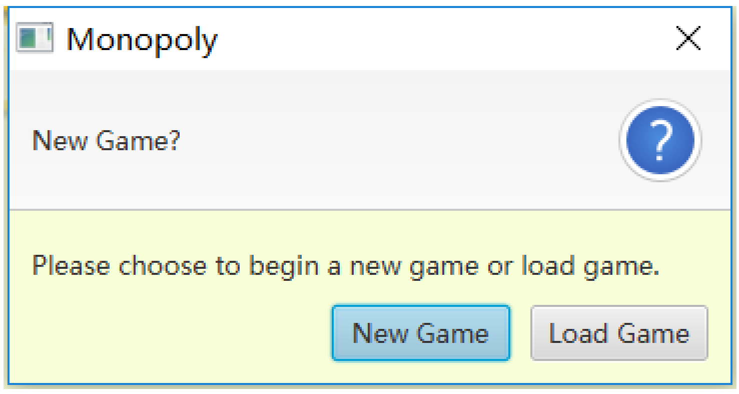
3. Quick Start Guide

3.1 Graphical User Interface

First, after running the program, choose 1 for GUI.

```
Please choose the program mode:
1. Graphical User Interface (GUI)
2. Command-line Interface (CLI)
1
```

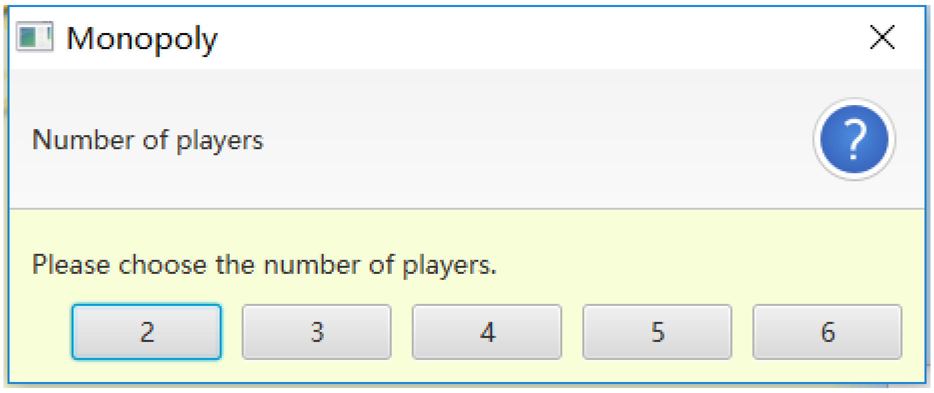
Then, you will see a window, choose "New Game" for starting a new game, choose "Load Game" for loading a game.


Monopoly

New Game?

Please choose to begin a new game or load game.

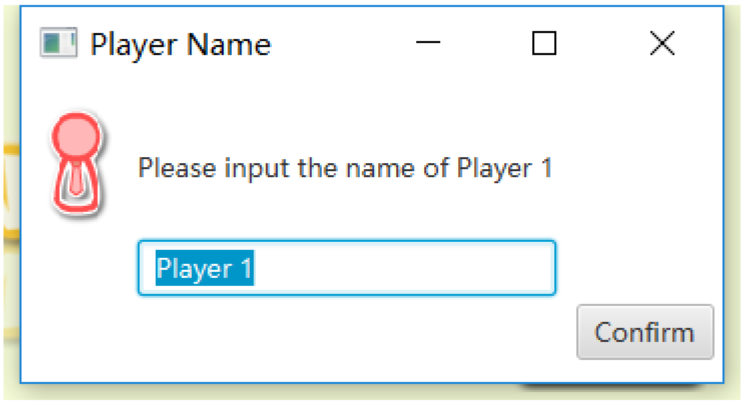
If you choose" New Game", then you will see another window for choosing the number of players


Monopoly

Number of players

Please choose the number of players.

Then, input each player's name.

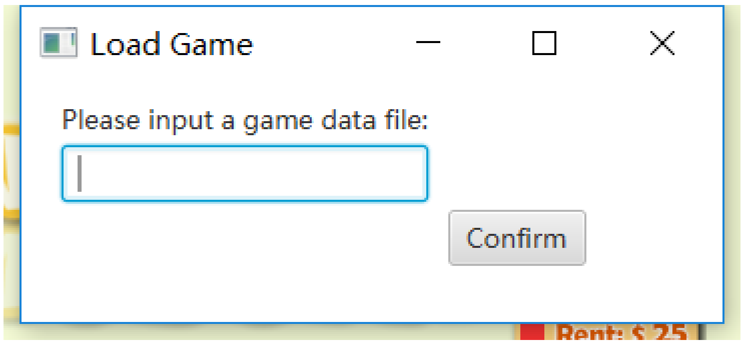


Player Name

Please input the name of Player 1

Player 1

If you click load game, you should input the correct file path to load file.

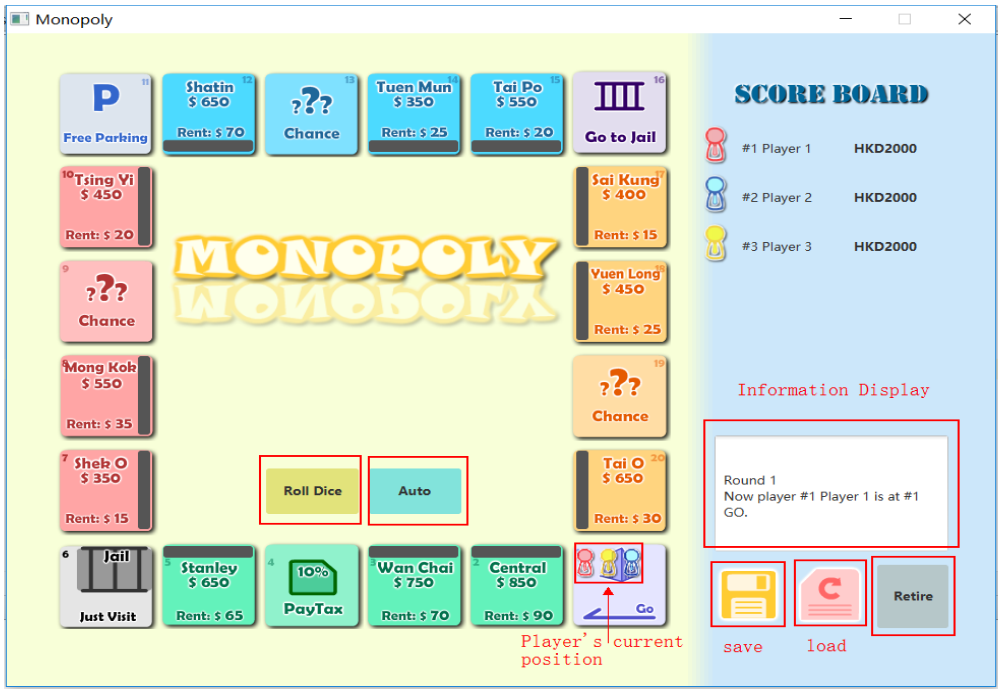


Load Game

Please input a game data file:

Then, you will see the Game Board.

As shown in the below screenshot, the game board lies on the left of screen, there are two buttons: “Roll dice” and “Auto”. There is a score board on the right which show the rank of each player and the color of each player, the game information will be displayed at lower right corner. There are also three buttons: save, load and retire at lower right corner.



Monopoly

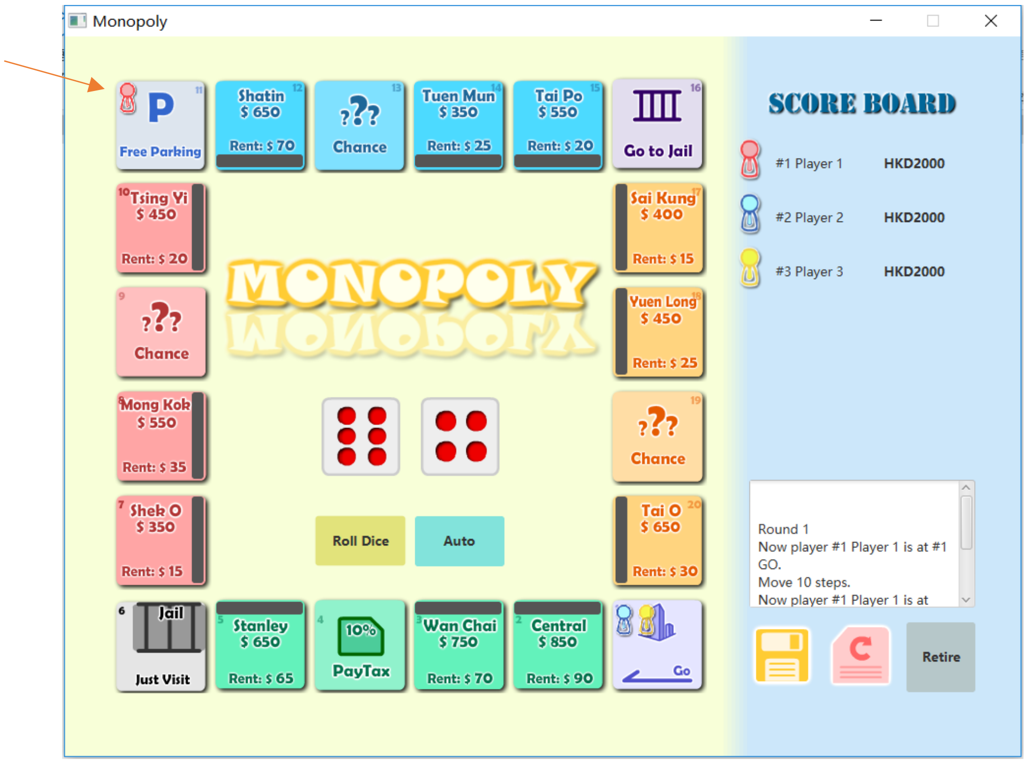
SCORE BOARD

#1 Player 1	HKD2000
#2 Player 2	HKD2000
#3 Player 3	HKD2000

Information Display

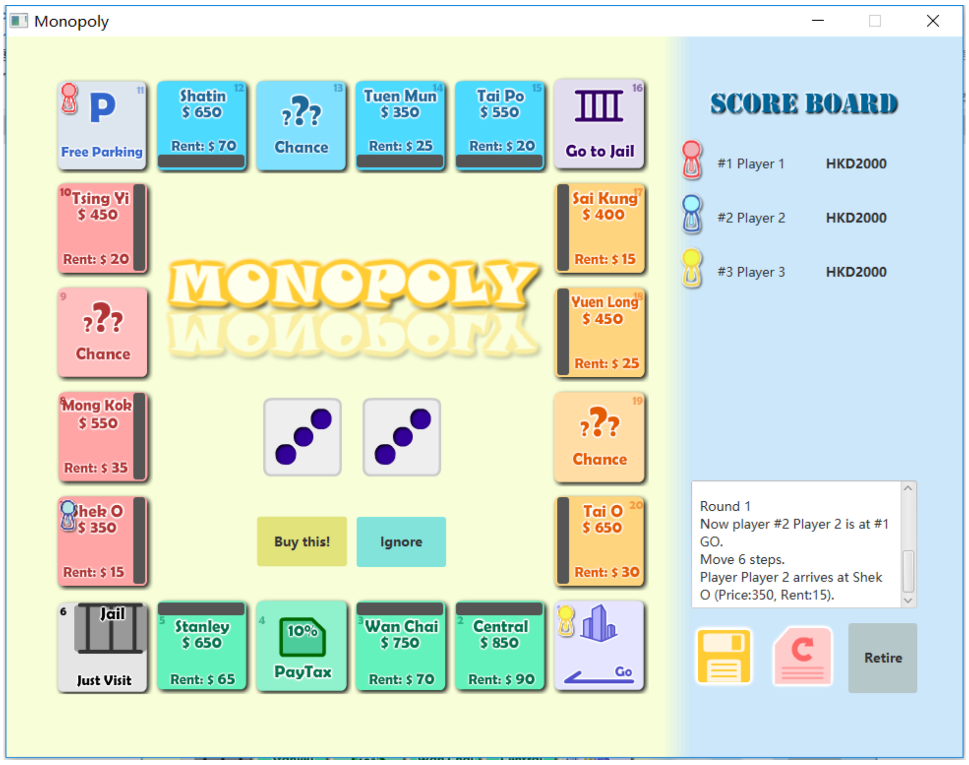
Round 1
Now player #1 Player 1 is at #1 GO.

If you click “Roll Dice”, you will roll two dices, the result will be showed on screen, you player will also move to the corresponding position. (Just as the arrow in the picture refers to)



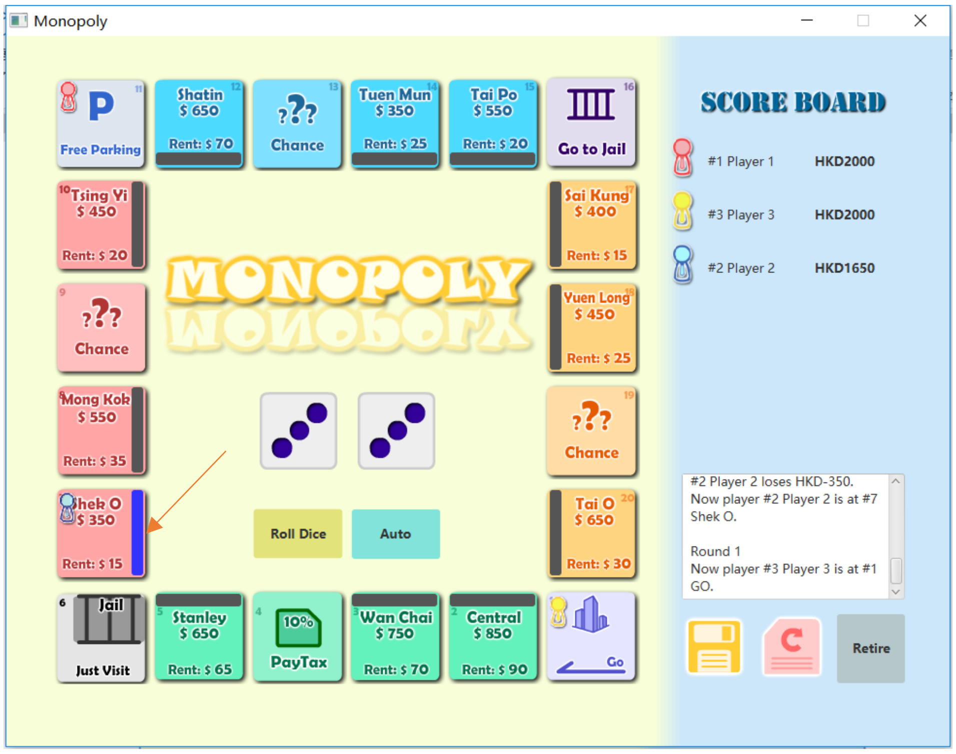
Round 1
Now player #1 Player 1 is at #1 GO.
Move 10 steps.
Now player #1 Player 1 is at

If you go to an unowned property, you can choose “Buy this” or “Ignore”



Round 1
Now player #2 Player 2 is at #1 GO.
Move 6 steps.
Player Player 2 arrives at Shek O (Price:350, Rent:15).

If you buy this place successfully, the color of this place will change to the corresponding color which represent the owner's color. (Just as the arrow in the picture refers to)

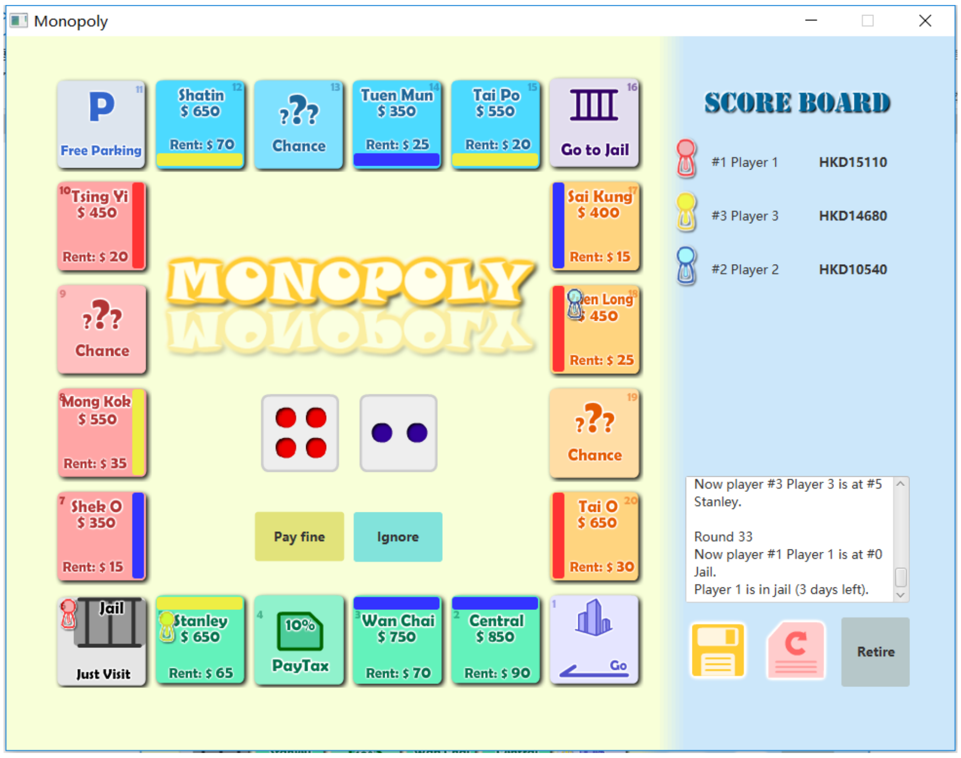


#1 Player 1	HKD2000
#3 Player 3	HKD2000
#2 Player 2	HKD1650

#2 Player 2 loses HKD-350.
Now player #2 Player 2 is at #7 Shek O.

Round 1
Now player #3 Player 3 is at #1 GO.

If you are in Jail, you can choose "Pay fine" to get out of jail immediately or choose "Ignore"

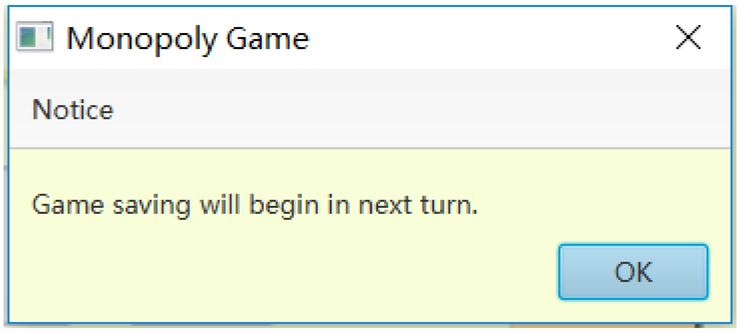


#1 Player 1	HKD15110
#3 Player 3	HKD14680
#2 Player 2	HKD10540

Now player #3 Player 3 is at #5 Stanley.

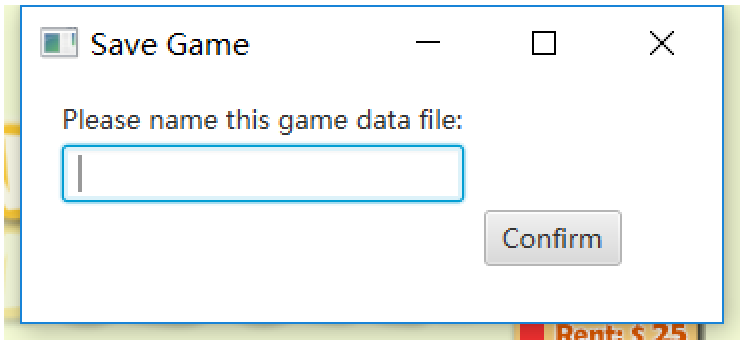
Round 33
Now player #1 Player 1 is at #0 Jail.
Player 1 is in jail (3 days left).

If you click save game, you should input the correct file path for saving game. (The file can be any type, like save.dat)


Monopoly Game

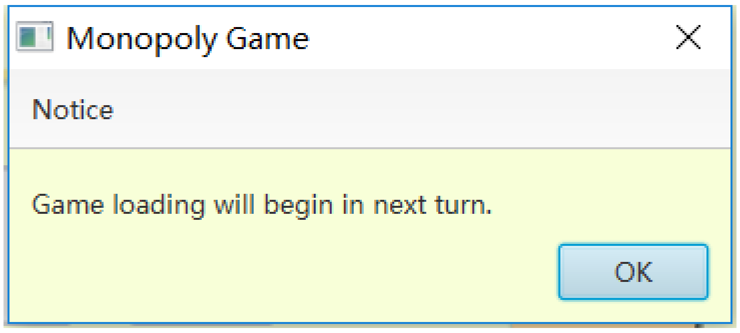
Notice

Game saving will begin in next turn.


Save Game

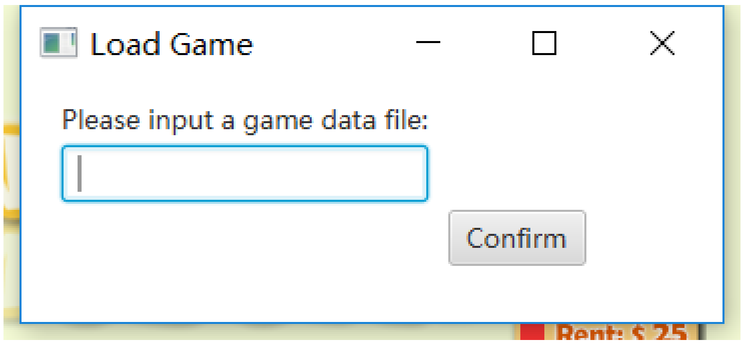
Please name this game data file:

If you click load game, you should input the correct file path for loading game.


Monopoly Game

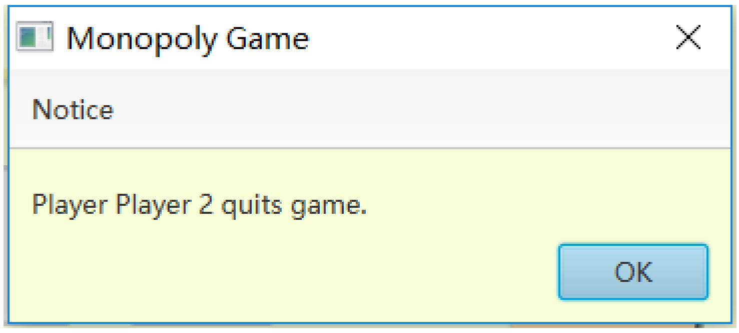
Notice

Game loading will begin in next turn.


Load Game

Please input a game data file:

If you click “Retire”, the game will remove the player from the list of players and make all his property unowned.



Monopoly Game

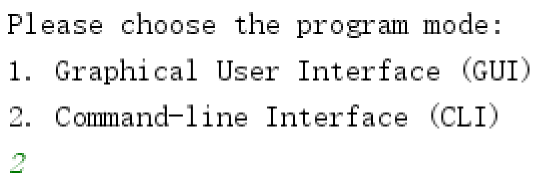
Notice

Player Player 2 quits game.

The game ends after 100 rounds, or when there is only one player left at the end of a round. The winner is the player with the most money when the game ends. Ties (multiple winners) are possible.

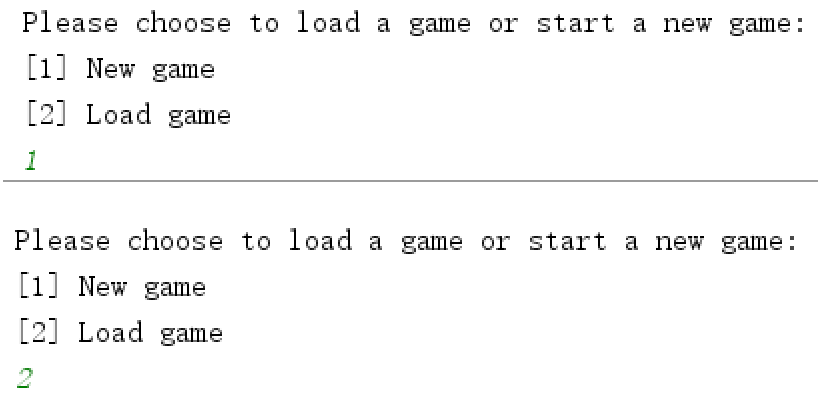
3.2 Command Line Interface:

First, after running the program, choose 2 for Command-line Interface.



```
Please choose the program mode:
1. Graphical User Interface (GUI)
2. Command-line Interface (CLI)
2
```

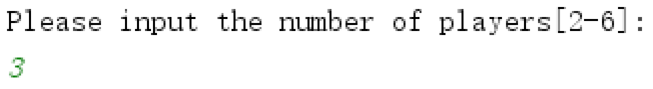
Choose 1 for new game, 2 for load game



```
Please choose to load a game or start a new game:
[1] New game
[2] Load game
1
```

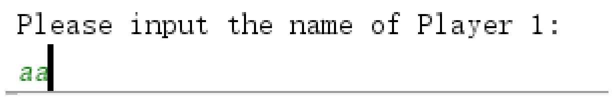
```
Please choose to load a game or start a new game:
[1] New game
[2] Load game
2
```

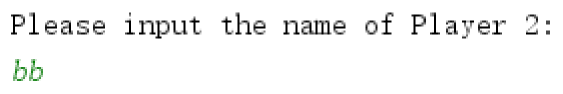
If you choose to begin a new game ,you should input the number of players(between 2 to 6)

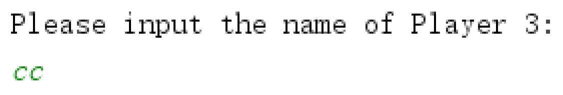


```
Please input the number of players[2-6]:
3
```

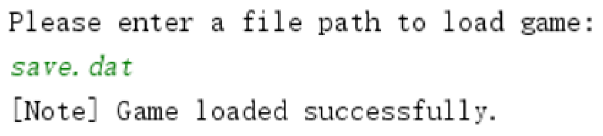
Then input the player's name respectively


```
Please input the name of Player 1:
aa
```


```
Please input the name of Player 2:
bb
```

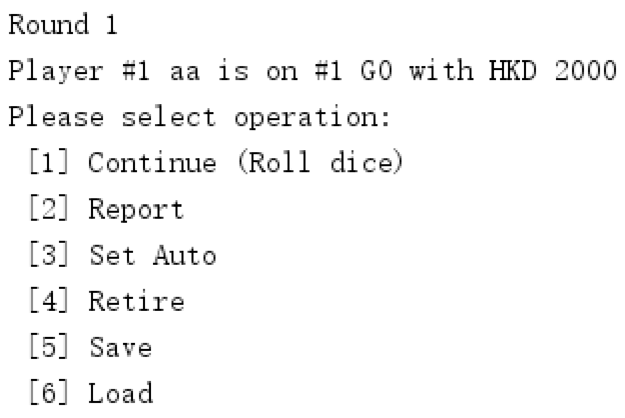

```
Please input the name of Player 3:
cc
```

If you choose load game, you should input the correct file path.

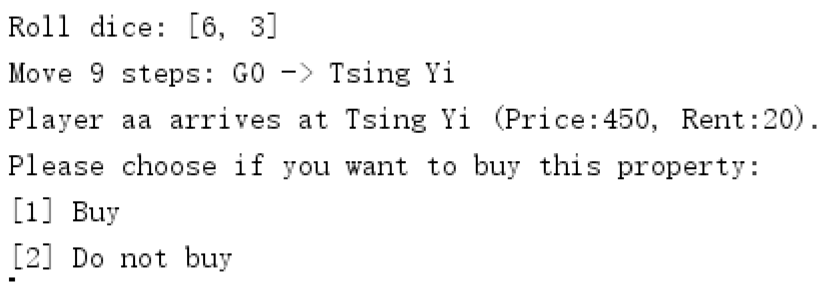

```
Please enter a file path to load game:
save.dat
[Note] Game loaded successfully.
```

For each round, you have 6 choices,

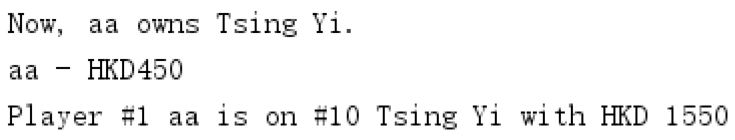
[1] Continue (Roll dice) [2] Report [3] Set Auto [4] Retire [5] Save [6] Load


```
Round 1
Player #1 aa is on #1 GO with HKD 2000
Please select operation:
 [1] Continue (Roll dice)
 [2] Report
 [3] Set Auto
 [4] Retire
 [5] Save
 [6] Load
```

If you choose [1] Continue (Roll dice) and after rolling dice you are in a property with no owner, you can choose whether to buy this property


```
Roll dice: [6, 3]
Move 9 steps: GO -> Tsing Yi
Player aa arrives at Tsing Yi (Price:450, Rent:20).
Please choose if you want to buy this property:
[1] Buy
[2] Do not buy
```

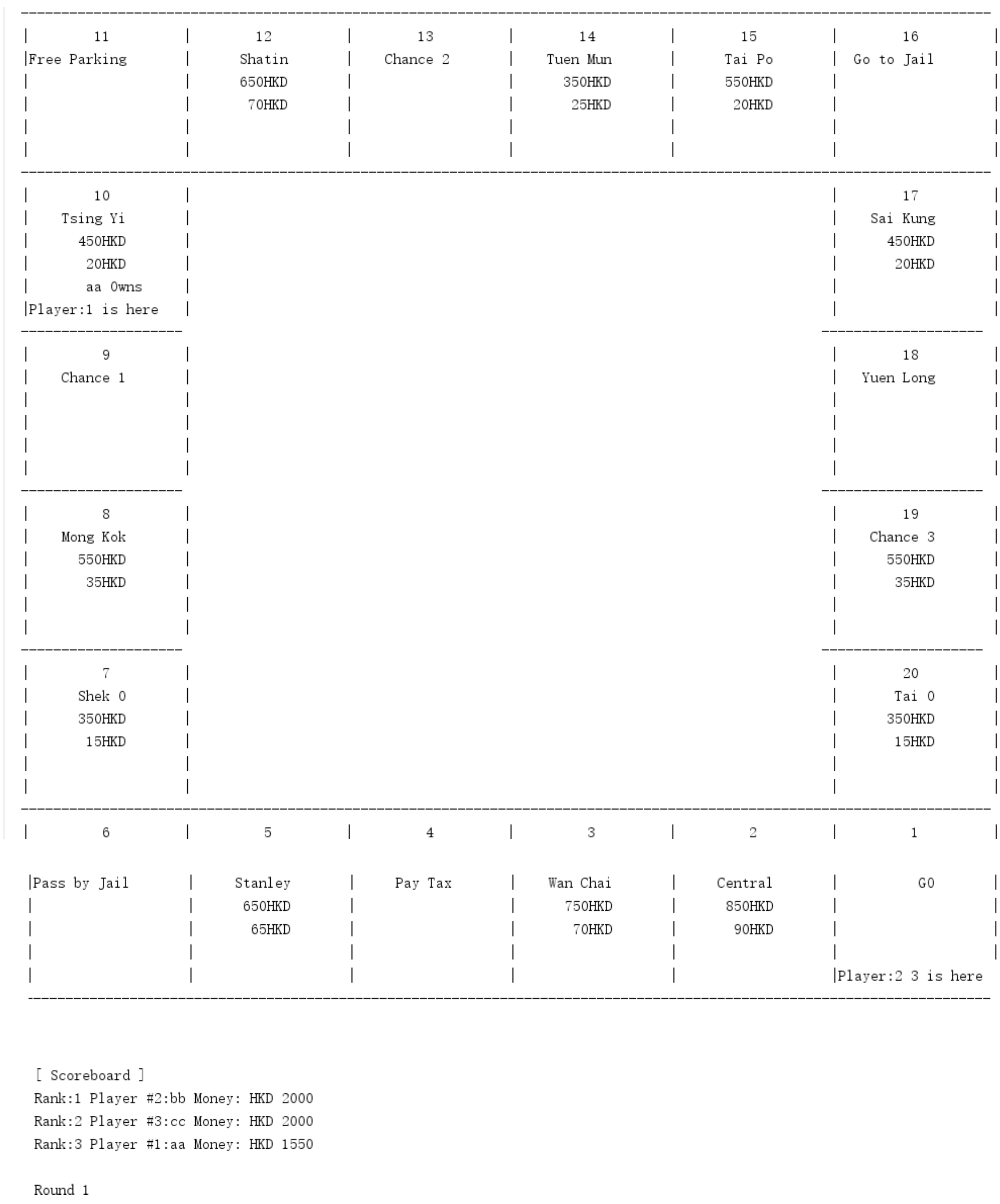
If you choose [1] buy, you will see the picture below, if you choose [2] Do not buy, it will become next player's round.



```
Now, aa owns Tsing Yi.
aa - HKD450
Player #1 aa is on #10 Tsing Yi with HKD 1550
```

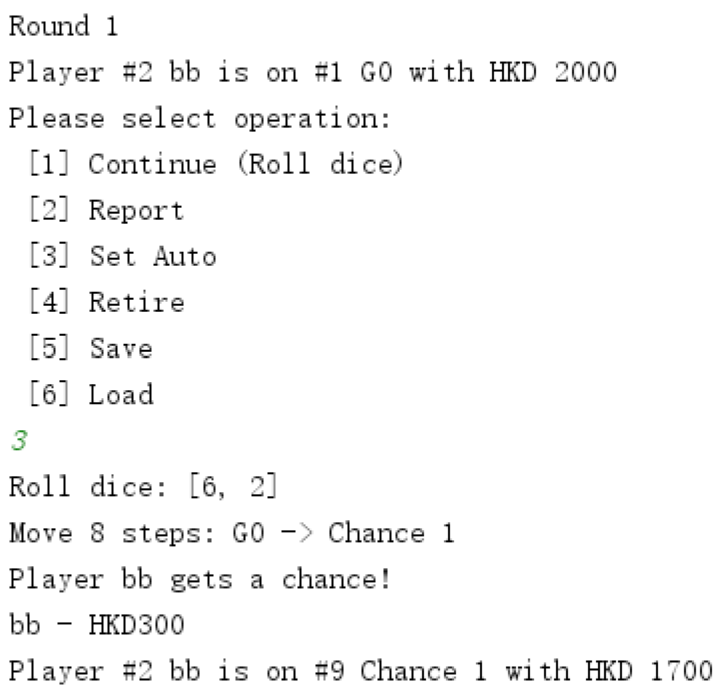
If you choose [2] Report, you will see the whole game board. For each place, the first line represents its number, the second line shows its name, the third line shows its price, the forth line shows its rent, the fifth line shows the ascription of this place, the sixth line shows which player is in this place.

You will also see a scoreboard.



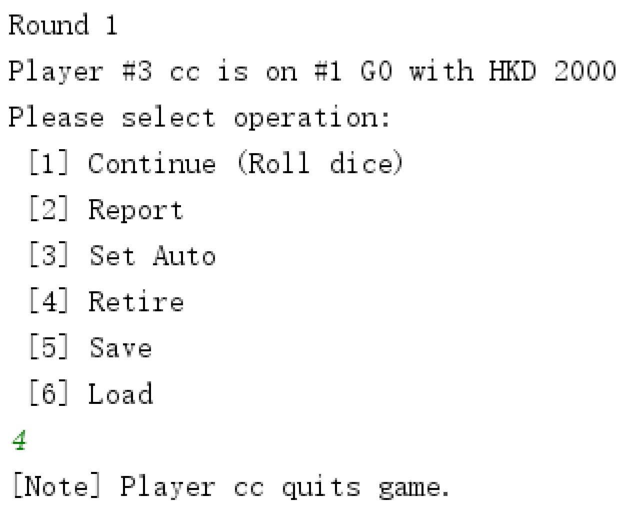
```
------------------------------------------------------------------------------------------------------------------------
|       11         |        12         |       13         |        14        |        15        |        16        |
|Free Parking      |      Shatin       |    Chance 2      |     Tuen Mun     |      Tai Po      |   Go to Jail     |
|                  |      650HKD       |                  |      350HKD      |      550HKD      |                  |
|                  |       70HKD       |                  |       25HKD      |       20HKD      |                  |
|                  |                   |                  |                  |                  |                  |
|                  |                   |                  |                  |                  |                  |
------------------------------------------------------------------------------------------------------------------------
|       10         |                                                                              |        17        |
|    Tsing Yi      |                                                                              |     Sai Kung     |
|     450HKD       |                                                                              |      450HKD      |
|      20HKD       |                                                                              |       20HKD      |
|     aa Owns      |                                                                              |                  |
|Player:1 is here  |                                                                              |                  |
--------------------                                                                              --------------------
|        9         |                                                                              |        18        |
|    Chance 1      |                                                                              |    Yuen Long     |
|                  |                                                                              |                  |
|                  |                                                                              |                  |
|                  |                                                                              |                  |
|                  |                                                                              |                  |
--------------------                                                                              --------------------
|        8         |                                                                              |        19        |
|    Mong Kok      |                                                                              |     Chance 3     |
|     550HKD       |                                                                              |      550HKD      |
|      35HKD       |                                                                              |       35HKD      |
|                  |                                                                              |                  |
|                  |                                                                              |                  |
--------------------                                                                              --------------------
|        7         |                                                                              |        20        |
|     Shek O       |                                                                              |      Tai O       |
|     350HKD       |                                                                              |      350HKD      |
|      15HKD       |                                                                              |       15HKD      |
|                  |                                                                              |                  |
|                  |                                                                              |                  |
------------------------------------------------------------------------------------------------------------------------
|        6         |         5         |        4         |         3        |         2        |         1        |

|Pass by Jail      |      Stanley      |     Pay Tax      |     Wan Chai     |      Central     |        GO        |
|                  |      650HKD       |                  |      750HKD      |      850HKD      |                  |
|                  |       65HKD       |                  |       70HKD      |       90HKD      |                  |
|                  |                   |                  |                  |                  |                  |
|                  |                   |                  |                  |                  |Player:2 3 is here
------------------------------------------------------------------------------------------------------------------------


[ Scoreboard ]
Rank:1 Player #2:bb Money: HKD 2000
Rank:2 Player #3:cc Money: HKD 2000
Rank:3 Player #1:aa Money: HKD 1550

Round 1
```

If you choose [3] Set Auto, the game will replace a human player with a computer player. The computer player always continues until the end of the game and makes other decisions randomly.



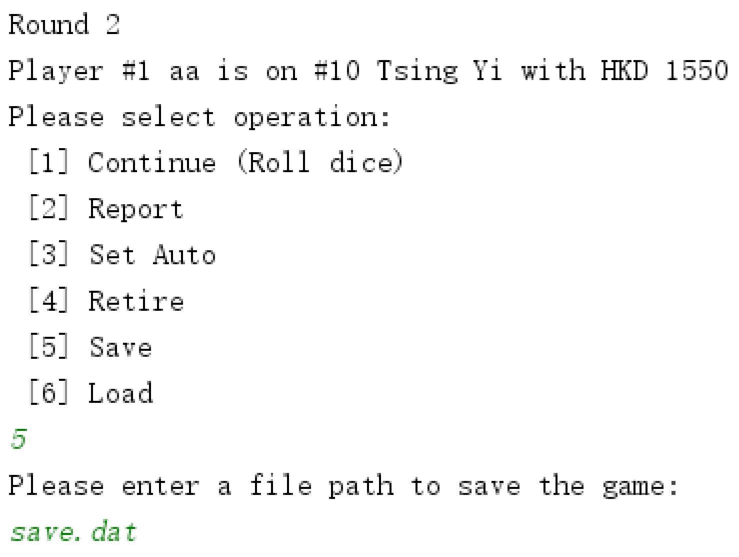
```
Round 1
Player #2 bb is on #1 GO with HKD 2000
Please select operation:
 [1] Continue (Roll dice)
 [2] Report
 [3] Set Auto
 [4] Retire
 [5] Save
 [6] Load
3
Roll dice: [6, 2]
Move 8 steps: GO -> Chance 1
Player bb gets a chance!
bb - HKD300
Player #2 bb is on #9 Chance 1 with HKD 1700
```

If you choose [4] Retire, the game will remove the player from the list of players and make all his property unowned.



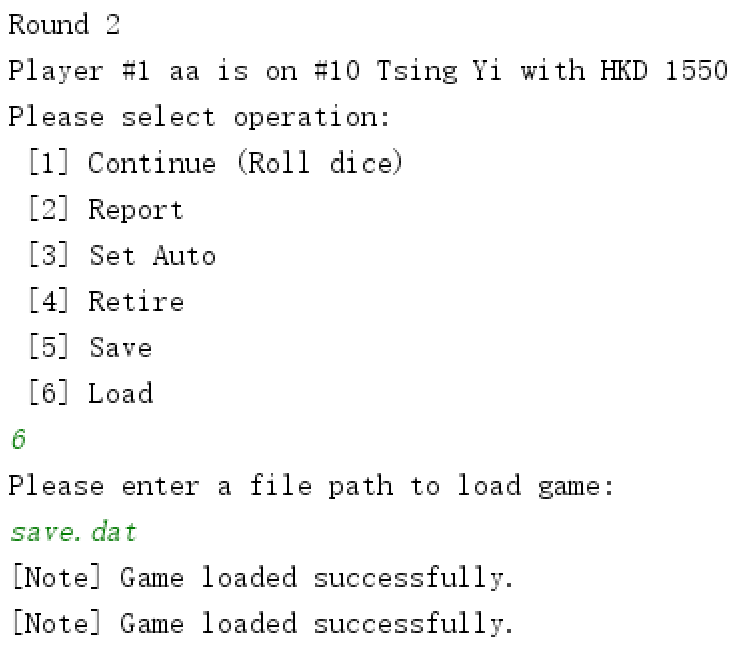
```
Round 1
Player #3 cc is on #1 GO with HKD 2000
Please select operation:
 [1] Continue (Roll dice)
 [2] Report
 [3] Set Auto
 [4] Retire
 [5] Save
 [6] Load
4
[Note] Player cc quits game.
```

If you choose [5] Save, you should input the file path you want to save. Then the game will be saved.(The save file can be any type, like save.dat)



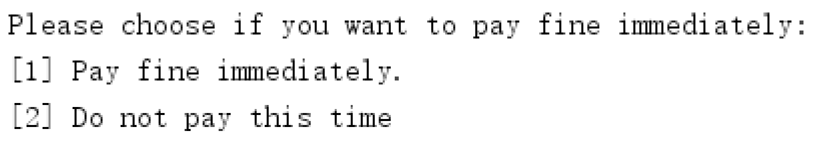
```
Round 2
Player #1 aa is on #10 Tsing Yi with HKD 1550
Please select operation:
 [1] Continue (Roll dice)
 [2] Report
 [3] Set Auto
 [4] Retire
 [5] Save
 [6] Load
5
Please enter a file path to save the game:
save.dat
```

If you choose [6] Load, you should input the correct file path. Then the game will be loaded



```
Round 2
Player #1 aa is on #10 Tsing Yi with HKD 1550
Please select operation:
 [1] Continue (Roll dice)
 [2] Report
 [3] Set Auto
 [4] Retire
 [5] Save
 [6] Load
6
Please enter a file path to load game:
save.dat
[Note] Game loaded successfully.
[Note] Game loaded successfully.
```

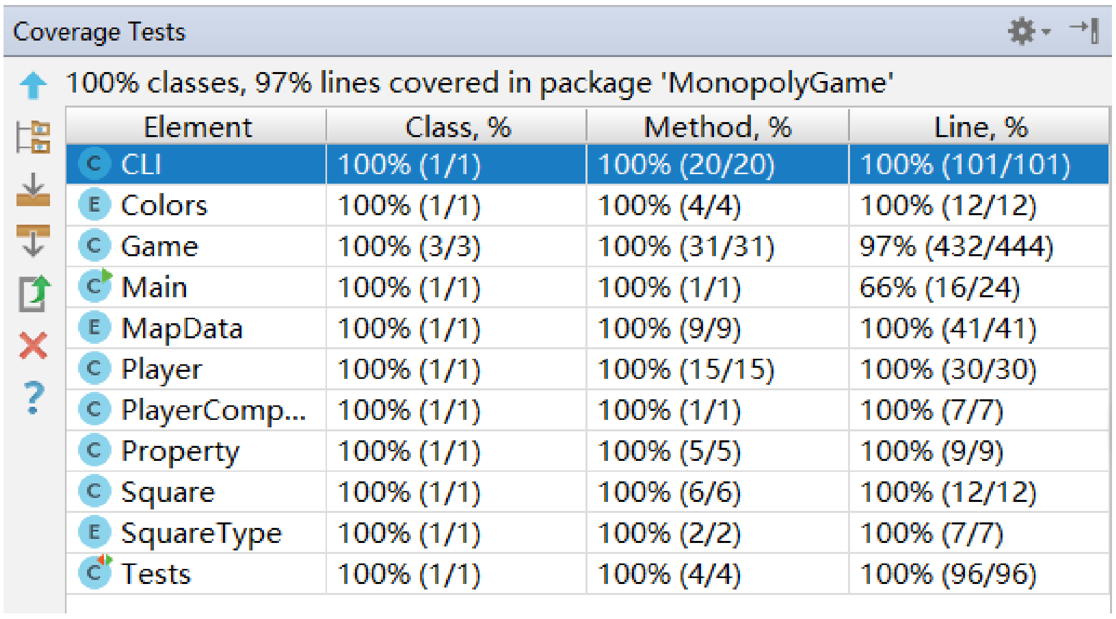
If you are in jail, you can choose [1] for paying fine immediately or choose [2] for not paying the fine. However, if it is your third day in jail, you must pay.



```
Please choose if you want to pay fine immediately:
[1] Pay fine immediately.
[2] Do not pay this time
```

The game ends after 100 rounds, or when there is only one player left at the end of a round. The winner is the player with the most money when the game ends. Ties (multiple winners) are possible.

Appendix: test coverage



Coverage Tests

100% classes, 97% lines covered in package 'MonopolyGame'

Element	Class, %	Method, %	Line, %
CLI	100% (1/1)	100% (20/20)	100% (101/101)
Colors	100% (1/1)	100% (4/4)	100% (12/12)
Game	100% (3/3)	100% (31/31)	97% (432/444)
Main	100% (1/1)	100% (1/1)	66% (16/24)
MapData	100% (1/1)	100% (9/9)	100% (41/41)
Player	100% (1/1)	100% (15/15)	100% (30/30)
PlayerComp...	100% (1/1)	100% (1/1)	100% (7/7)
Property	100% (1/1)	100% (5/5)	100% (9/9)
Square	100% (1/1)	100% (6/6)	100% (12/12)
SquareType	100% (1/1)	100% (2/2)	100% (7/7)
Tests	100% (1/1)	100% (4/4)	100% (96/96)