ReadMe

Ducument for JAVA GUI Advanture Game Project

This is a JAVA program to generate a GUI base advanture game object according to the requirements of the project description. The detail of the object will be listed in following section of this document.

Features

- This is a GUIt base advanture game by exploring unknown dungeon.
- The game will conduct in a 2-dimension dungeon. There is will be arrow, treasure(Dianmond, Ruby, Sapphire) and monster (Otyugh) randomly distrisbute within the dungeon.
- The state of player and location will be display on the west panel of the GUI when entering a new location. The information including type of location, smell(to indicate the distance of a monster), treasures, arrow and available directions.
- The available direction and operation for the player will display on the east side panel.
- The state of the dungeon will display on the central panel of GUI. Player can use mouse to click on the neighbor grid or click on the direction buttons on the east panel to move.
- Where there is treasure in the place, play need to click the Pick button to pickup stuff. If there is monster nearby, use the Shoot button to launch arrow to kill the mosnter.
- The game will finish when player arrive the destination or eaten by monster. Player can also use the mune to start a new game, resume the same game or quit the game.

Design/Model Changes

There were some major changes while implementing the project:

- [Use 2 models] - the previous model has been adjusted into A ViewOnlyModel and normal Model.

  ```
        The ViewOnlyModel can acquire the state of game, but can't change the
  state. The operation will accomplish
        by the controller and normal Model.
  ```

- [Add listener] - In order to achieve multiple function for the GUI base game, the program use some Listener

  ```
        function for panel, button and menubar.
  ```

- [SWING] - the program use the Java Swing library to achieve the visual functionalities.

How To Run

To start a game by this program, user can run the "main" method in "Main.class" in src folder. In the command line, input the arguments for a new game as below:

```
java Main.class 10 10 30 0.50f 0.25f false
```

The argument are: rows, columns, Interconnectivity, treasure percentange, difficulty, wrapping setting.

Another way to start the program is to execute the ".jar" file in the res folder in a console. like

```
java Pro05.jar 10 10 30 0.50f 0.25f false
```

How to Use

Check out the code or class in ralated package/folder.

Function	Package/Folder
Start Game	Driver.class / Main.class
Function of Dungeon and Player	dungeon
Function of Controller	controller
Function of Model	model
View	view
Function of Helper class	Helper

Assumptions

Below assumptions are made to implement the project:

1. A player can only play in one dungeon at a time;
2. The parameters about a game can be input in the Main or Driver class;
3. Player can click on the menu or close the frame to quit the game;
4. Player can click on neighbour cell or use direction button to nevigate;
5. it will take two arrow to kill a monster, player has 50% to escape if monster is wounded.

Description of Examples

At the last page of this document, there are some screen shots of the program running.

Limitation

The major challenge of this program is it's very hard to design a perfect GUI for me. Since I am a bigenner of Java, I need to spend a lot of time to learn how to use the SWING and all kinds of components or Layout Manager. Although the current layout is the best I can do for now, but it sitll has a lot of room to be improved. Another

limitation is that the program can't accept keyboard control (like use arrow button on keyboard). The only occation to accept keyboard input is to take the parameter for shooting. I hope I can improve this in the net version of program.

Citations

The Java™ Tutorials. A Visual Guide to Layout Managers. Retrieve from https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html

The Java™ Tutorials. How to Write an Action Listener. Retrieve from https://docs.oracle.com/javase/tutorial/uiswing/events/actionlistener.html

Noah Patullo. Jan 8, 2018. Kruskal's Algorithm Minimum Spanning Tree (Graph MST). Retrieve from https://github.com/SleekPanther/kruskals-algorithm-minimum-spanning tree-mst