Final Project Report

Student Name: Jihun Kim

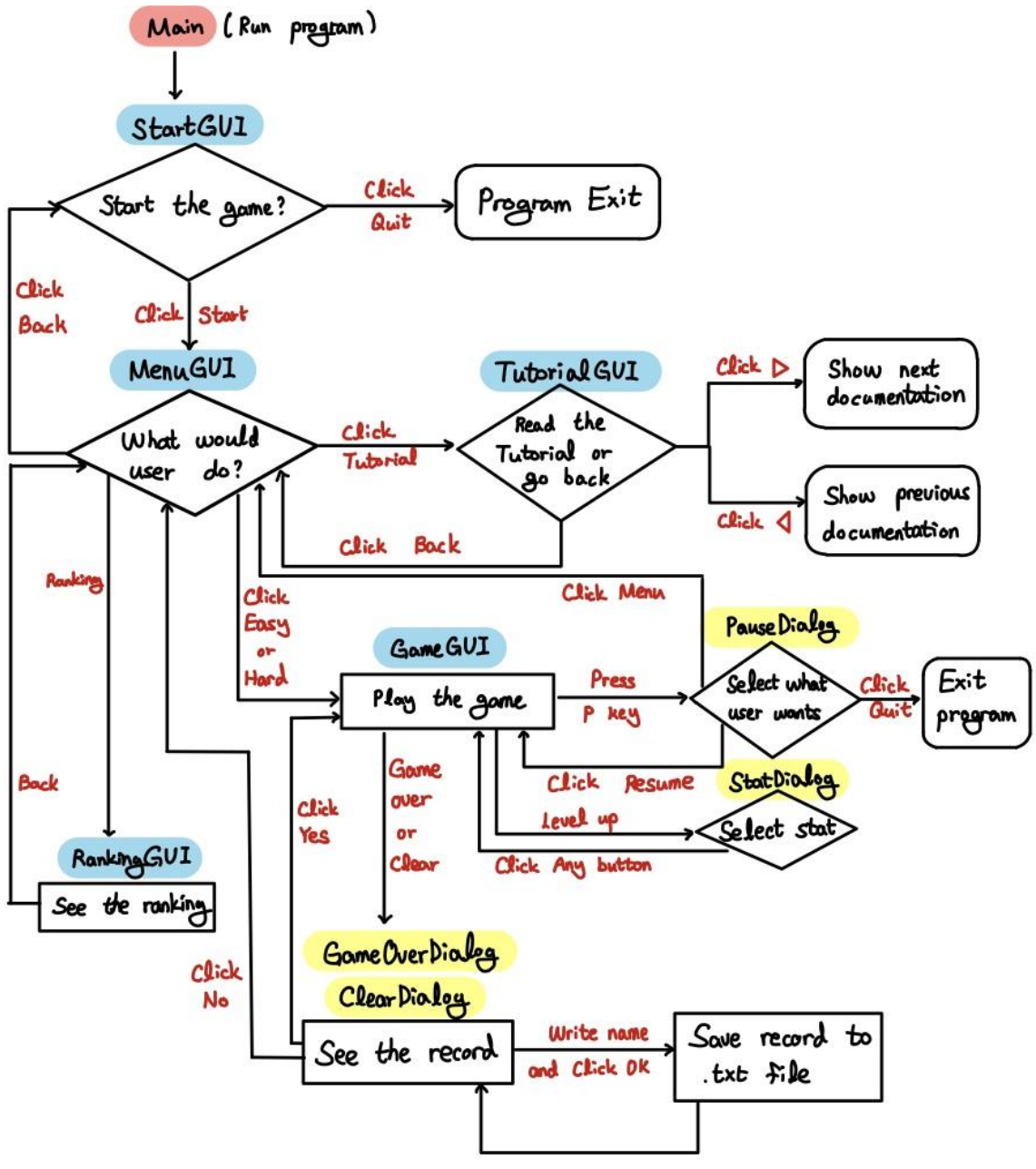
Student ID: 2019313611

1. Briefly describe the project purpose:

It is a simple shooting game to relieve the stress of modern people living a busy life. It reminds me of the 1945 game that most adult men would have played when they were young.

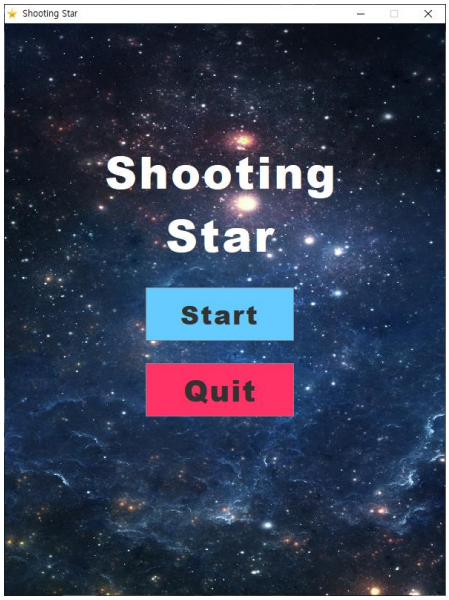
The title of the game made in this project is "Shooting Star". The name is cute and the graphics are also cute. The operation method is not difficult, so people of all ages can enjoy it easily, and there is also a Ranking system that allows you to bet with your friends.

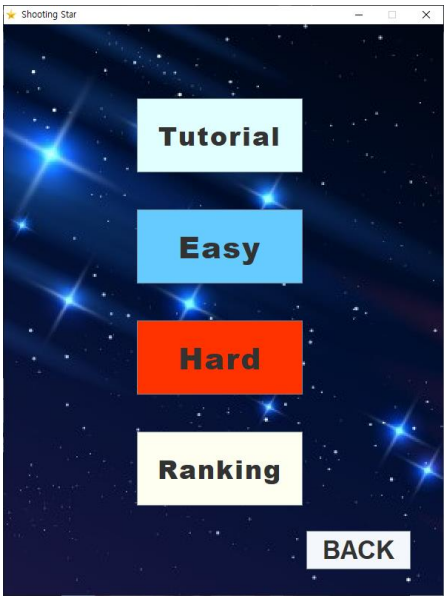
2. Draw the logic flow of the program (with flowchart):



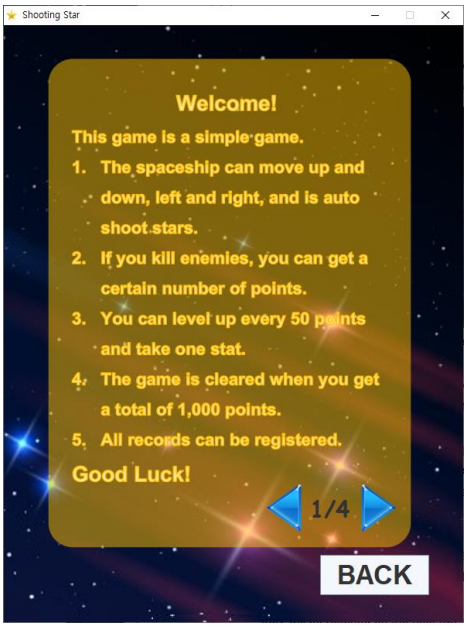
3. Provide screenshots for each screen with brief description:

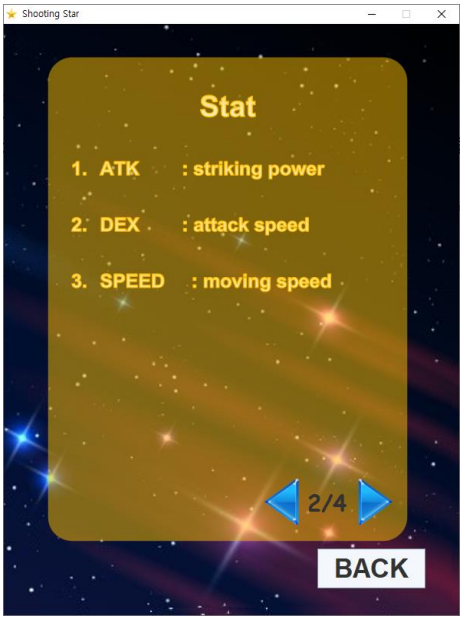
When you start the game, Tutorial, Easy, Hard, and Ranking buttons are displayed.

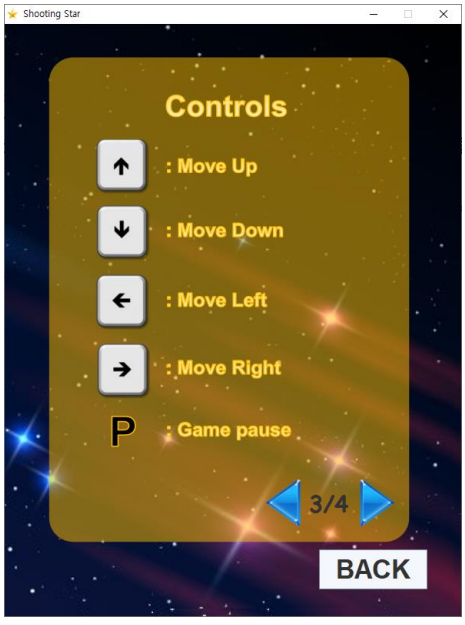


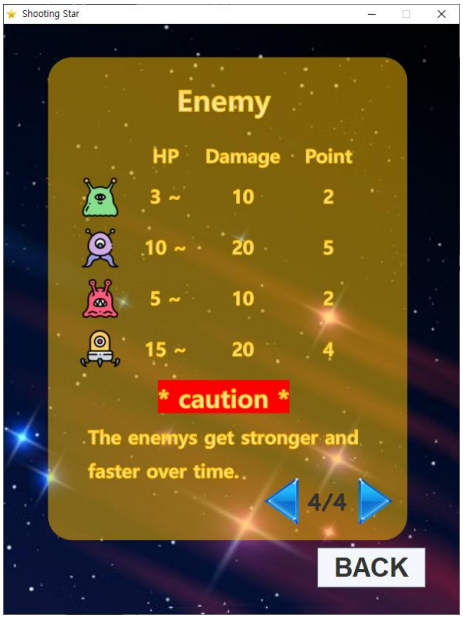


In the tutorial, you can see the game description, how to control it, and more.

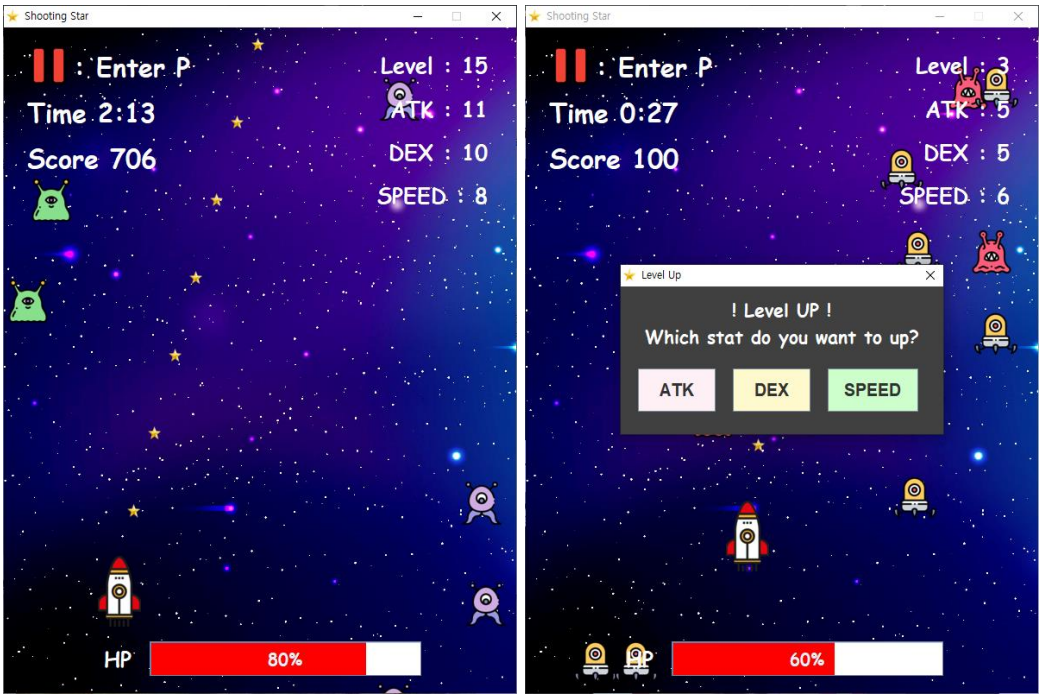




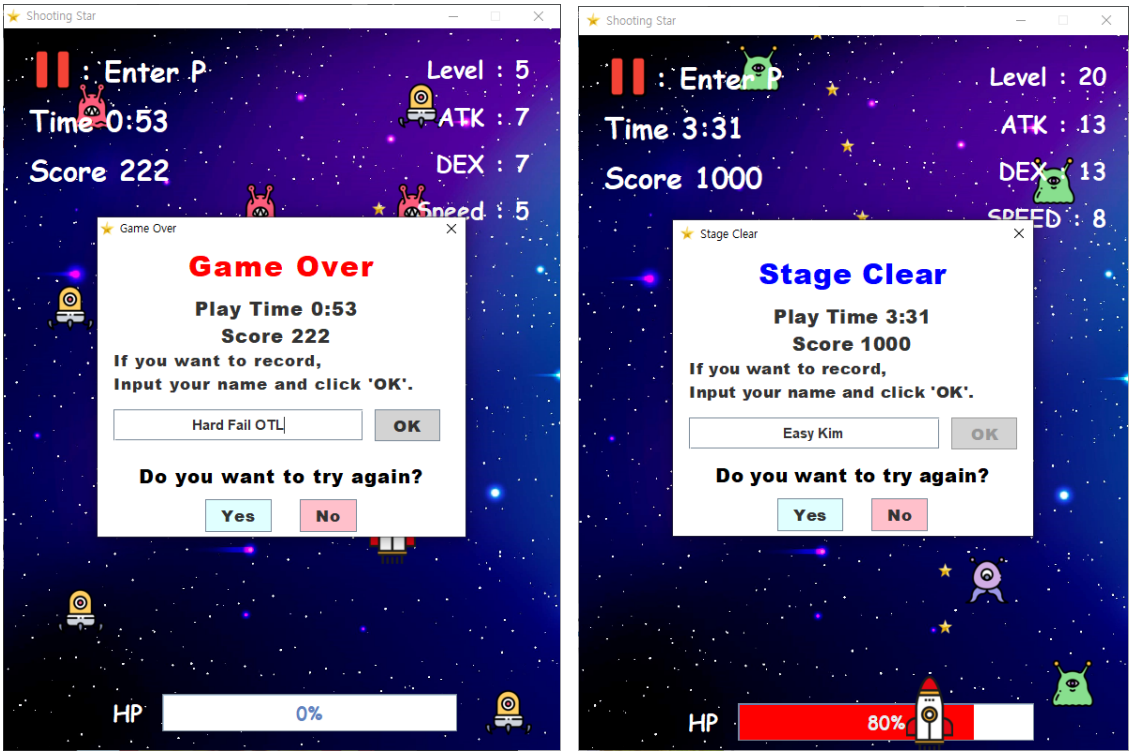




In the game, the user can ride a spaceship and shoot stars to kill enemies. The user can level up every 50 points, and can increase attack power, attack speed, and movement speed stats one by one.

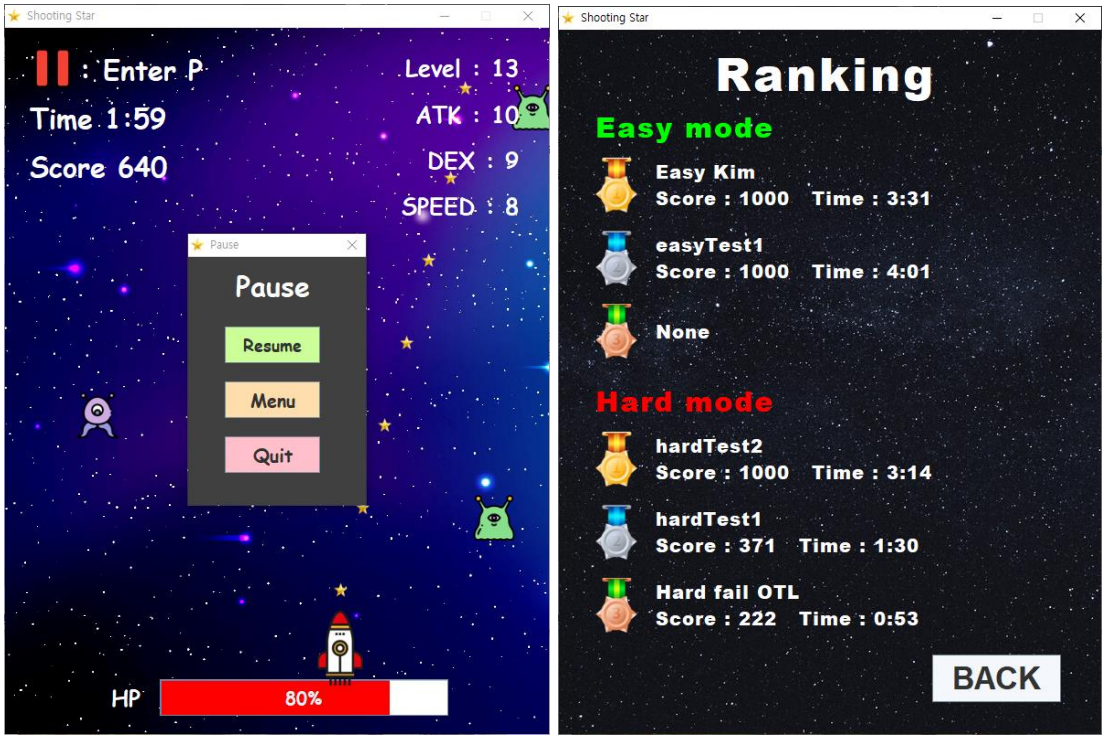


Enemies have different image and hp for each type, and the points user can get are also different. If user collide with an enemy, user will receive a certain amount of damage, and if the user's HP becomes to 0, it's game over. If the user gets 1,000 points, the game is clear.



If something urgent happens while playing the game, user can pause it by pressing the P key. In pause, user can resume the game, go to the menu, or exit the game.

All records can be left, and if you get a good score, it will be registered in Ranking. Ranking can be checked in the main menu. These records are stored in a text file, so they will be retained on the next run!



4. Explain the code of the main functionalities

In class GameGUI

[fireStar] : Method to create star

[moveStars] : Method to move all stars

[enemyCreate] : Method to create enemy

[moveEnemys] : Method to move all enemys

[keyControl] : Method that moves the user based on left, right, up, and down

[crashCheck] : Method to check if star and enemy collided or if user and enemy collided

[checkScore] : Method to check whether the current score has exceeded 1000 points or whether it is time to level up

[Pause] : When the user presses the p key, the game is paused.

In class RankingGUI

[sort] : Sort the record objects based on the compareTo method in the Record class.

[FileInputStream] : Read String from text file.

In class ClearDialog and GameOverDialog

[formatException] : Exception raised when there is a special character in the user name textField

[FileOutputStream] : Write String from text file.

In class all Dialog

[showDialog] : Method that sets the dialog visible and returns which one was selected

5. Explain what is included in your project and why it is used (Polymorphism, Inheritance, File I/O, etc)

① Inheritance

To use the GUI [Swing], JFrame and JDialog were inherited from each GUI and Dialog class. In addition, Exception was inherited from formatException class to handle special cases (special characters) as exceptions.

② Polymorphism

In order to sort objects, Comparable was implemented in the Record class. After that, we overrided the compareTo method.

Runnable was implemented to use multi-threading in GameGUI, and KeyListener was implemented to receive user's key input. After that, I overrided the run method, keyPressed method, and keyReleased method.

③ Exception Handling

Because I used methods that can cause exceptions, such as timer, fileInputStream, and fileOutputStream, exception handling was done using try-catch.

In addition, there is a line that throws and catches formatException so that special characters are not entered when receiving user names in ClearDialog and GameOverDialog.

④ Thread

If one thread is executing a long process, it does not make the entire application wait for it to finish. Therefore, Thread was used to smoothly progress the game.

⑤ File I/O

In ClearDialog and GameOverDialog, the user's record is saved in each text file according to the level of difficulty. In RankingGUI shows the ranking by importing information from each text file.

File I/O was used so that the ranking would not be lost even if the program was re-executed.