CS 319

Object Oriented Software Engineering

RUSH HOUR CC

Final Report

Fatbardh Feta, Masna Ahmed, Naisila Puka, Kunduz Efronova, Talha Zeeshan

Supervisor: Eray Tüzün

December 23rd, 2018

Table of Contents

1. Implementation Process 3
1.1 System Requirements 4
1.2 Installation Guide 4
2. Changes and Improvements 5
2.1 File Management System 5
2.2 Main Menu Panel 7
2.3 Choose Level Panel 7
2.4 Game Panel 7
3. Game Manual 8
3.1 Registration 8
3.2 Main Menu 9
3.3 Choose Difficulty 10
3.4 Choose Puzzle 11
3.5 RushHour 12
3.6 Instructions 13
3.7 Settings ..... **Error! Bookmark not defined.**
3.8 Garage Panel 15
4. Implementation Reflections ..... **Error! Bookmark not defined.**

1. Implementation Process

After thoroughly building up the whole logic of game in our Design and Analysis reports and understanding main game flow, our group started implementation of Rush Hour immediately after first iteration of Design Analysis report. We chose Eclipse IDE for implementing our code since every member of our group was familiar with this particular IDE. Additionally, we used Eclipse's Windows Builder to help us out with the GUI part of our project. As is specified on the web page of our CS-319 course, we used our GitHub accounts which we opened in the beginning of semester to catalogue our project and implementation details relating to it.

Our first course of action for the implementation of our Rush-Hour was making GUI related panels and frames. At first we did not distribute workload, but rather we implemented parts of the GUI we choose ourselves and add afterwards attached basic game related logic to them for the first demo. After the second iteration of the design report, features not completed during the first iteration were divided and implemented. Some features originally added for the first demo were removed from the game and others added. The full list of changes made since the first iteration demo are listed in this report. While implementing our parts we were constantly interacting with each other to discuss our struggles and help each other out in the implementation of the project. Implementatıon of source code was made easier since we followed our design patterns detailed in our design report (2nd iteration). We have completed all our promised functionalities and even improved upon our initial promises.

1.1 System Requirements

RushHour will require a user to have the Java Virtual Machine (JVM) installed on his computer as well as the Java SDK preferably the Java SE 8. If both are installed on the user's computer, he will be able to install and run the game by following the installation guide.

1.2 Installation Guide

Users should follow this basic installation guide to be able to download, install and run our RushHour game on their systems:

1) All relevant information regarding the game can be seen on the following link:

   https://github.com/NaisilaPuka/rushHourByChainCoders

   Users can view and download the game's file as well as the required documentation that took place during the development of the game.

2) Once the user has downloaded the necessary files, he can run it either via the command prompt or by using an IDE such as Eclipse or NetBeans. Users can run the game from the command prompt by first navigating to the directory where the game files have been downloaded, and then setting a path for the java development kit (JDK) programs present on the system. The user can then use the java compiler to generate a .class file in the same directory. Once both the .class files and .java files are available in the game directory, the user will be able to successfully run the game. On windows the following commands should be used if say the files are in "RushHourCC" directory:

   - `C:\> cd RushHourCC`
   - `C:\RushHourCC> set`

```
path=%path%;C:\ProgramFiles\Java\jdk1.8.0_101\bin
```

- `C:\RushHourCC> javac *.java`
- `C:\RushHourCC> java RushHourFrame`

3) For a user playing the game for the first time, he can view the game's instructions by clicking on the "Instructions" button
4) The user can view information regarding creators of the game in the settings panel where the game credits are given.

2. Changes and Improvements

2.1 File Management System

The file management system is a project class that communicated with all panels present on the game frame and feeds them game information from a collection of files present in the game's project directories.

The file management system uses multiples files containing information about the user, system settings, user achievements, the user's inventory as well as puzzle configuration. Following is a list of the major files and collections as well as a description of what information is written in them:

1) *User text file*:

   This file contains all game-relevant information about the user such as their name, avatar, number of stars, number of coins as well as their currently selected car. Every piece of information is written on a separate line with each line separated by a semicolon.

2) Settings text file:

   This file contains information regarding the game's current settings. This includes the game's mouse sensitivity, music volume within the game, the current music theme and the game's background theme. Every piece of information is written on a separate line with each line separated by a semicolon.

3) Locked and Unlocked Cars file:

   These files contain the lists of cars that are locked and unlocked respectively. When a car is to be unlocked, the file management system simple removes that particular car's name from the locked text file and writes in the unlocked text file. The user can then choose their selected car from the list provided by the unlocked text file.

4) Puzzle text files:

   These are files that contain a particular puzzle's configuration and statistics. The first five lines of these files contain the puzzle's statistics such as the puzzle's grid size, whether it is solved or not, the user's star count, the user's coin count, the grid's minimum number of solving moves and the user's attempted number of moves. The next lines contain information about the cars that will be present on the game grid. This includes information such as the cars' position on the grid, their sizes, their orientation and their image URLs.

The file management system contains functionality to obtain the data present in these files as well as modify them for the purposes of the game. A single instance of the File Management System class is present in the game which is passed as a parameter to all game panels thereby allowing communication between them.

2.2 Main Menu Panel

Although not originally promised, our group decided that our games theme should be based the on the popular movie called “Rush Hour” starring Jackie Chan. Therefore, the backgrounds and songs that our part of the game are related to the movie mentioned above. This was done to make the game stand out more and to provide a fun twist on the game, to make it feel fresh.

(Add screenshot)

2.3 Choose Level Panel

The panel shown above allows the user to choose the level of difficulty of puzzles they would like to solve. The core functionality of this panel remains the same as was mentioned previously meaning, as was stated in our earlier reports, the beginner level is the only difficulty level initially allowed to be accessed by the user. A higher difficulty level will only be unlocked if the user has a prerequisite number of stars he/she can earn by solving puzzles. Initially, all difficulty levels included puzzles on a 6x6 grid. However, after the first iteration demo, due to feedback provided by our instructor, the hard and expert difficulty levels now include puzzles on a 7x7 grid whereas the beginner and intermediate levels still consist of puzzles on a 6x6 grid.

(Add screenshot)

2.4 Game Panel

This is the Game panel of our game where the game is actually played. Since the second iteration of the design report, the game panel has been changed noticeably. The layout of the panel remains the same as shown previously. However, changes

were made to the core gameplay systems which are reflected in this panel. The changes made to are listed below;

1) Firstly, the timer and crown scoring system was removed from the game. This was done due to the introduction of a feature we call “Explode”.

2) The explode feature allows the user to remove one and only one car from the grid in order to help them solve the puzzle. Using the “Explode” feature affects the score and reward of the user, i.e penalizes the user by decreasing the stars and coins earned. The power can only be used once per level and the scoring system discourages the user from using that feature.

Furthermore, an undo move feature was also added to the game so that the user may undo a move they had made. Finally, the movement of the car objects on the grid was improved. The cars now slide across the grid in their given directions where as previously, the cars moved in a block like animation (disappearing and reappearing).

(Add screenshot)

3. Game Manual

Information regarding the game’s installation and running has been given in section 1.2. In this section detailed information regarding in-game functionality has been given in a sequential manner.

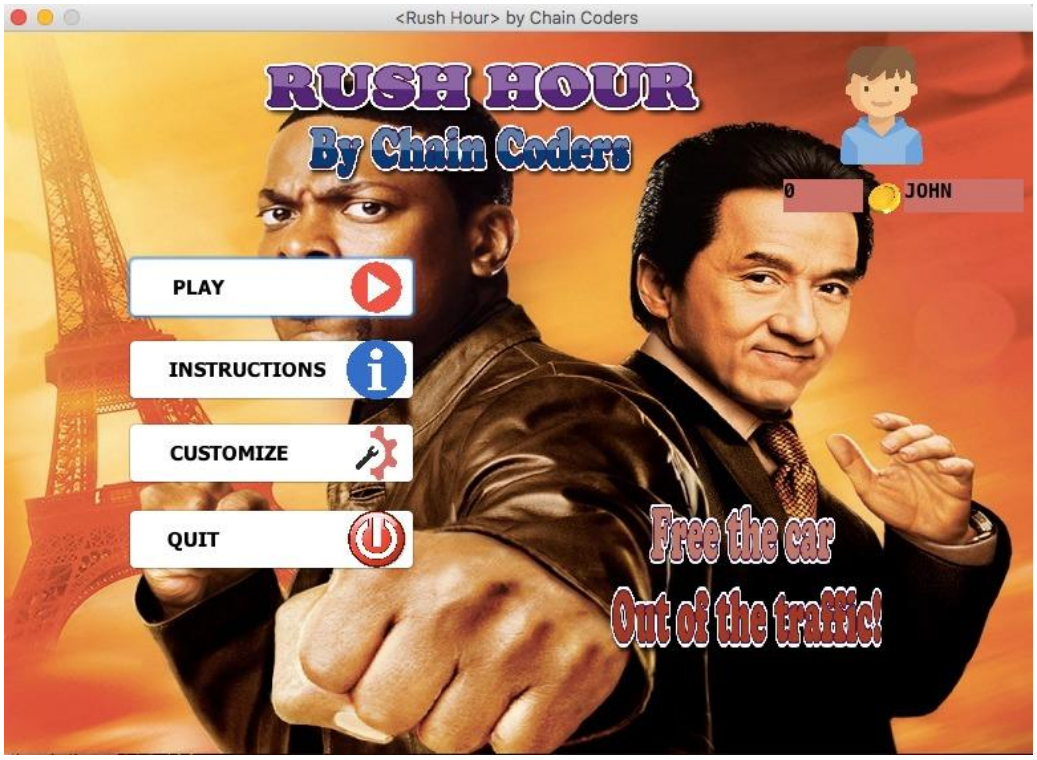
3.1 Registration

The will be the first screen that the user will see when he runs the game for the first time. It will prompt the user to enter his name and choose his avatar. This information will then be saved the user’s text file described in section 2.1 and will be used for later

screens.

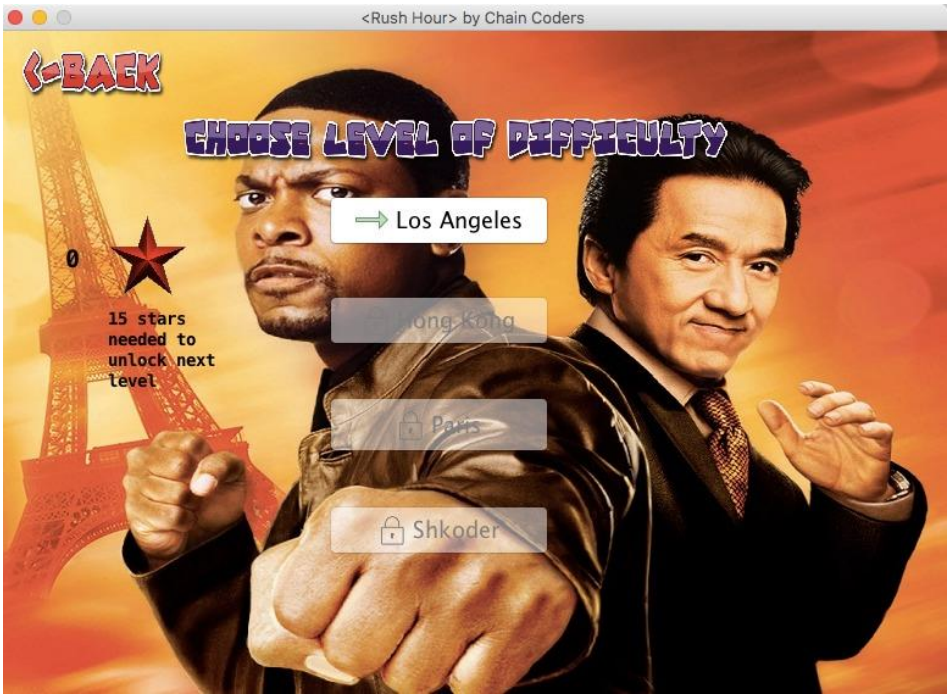
3.2 Main Menu

This menu will allow the user to either play the game, view the instruction manual, customize the game's settings and customize his inventory or quit the game. This screen also shows the user their basic information such as their name, avatar and their total number of coins.



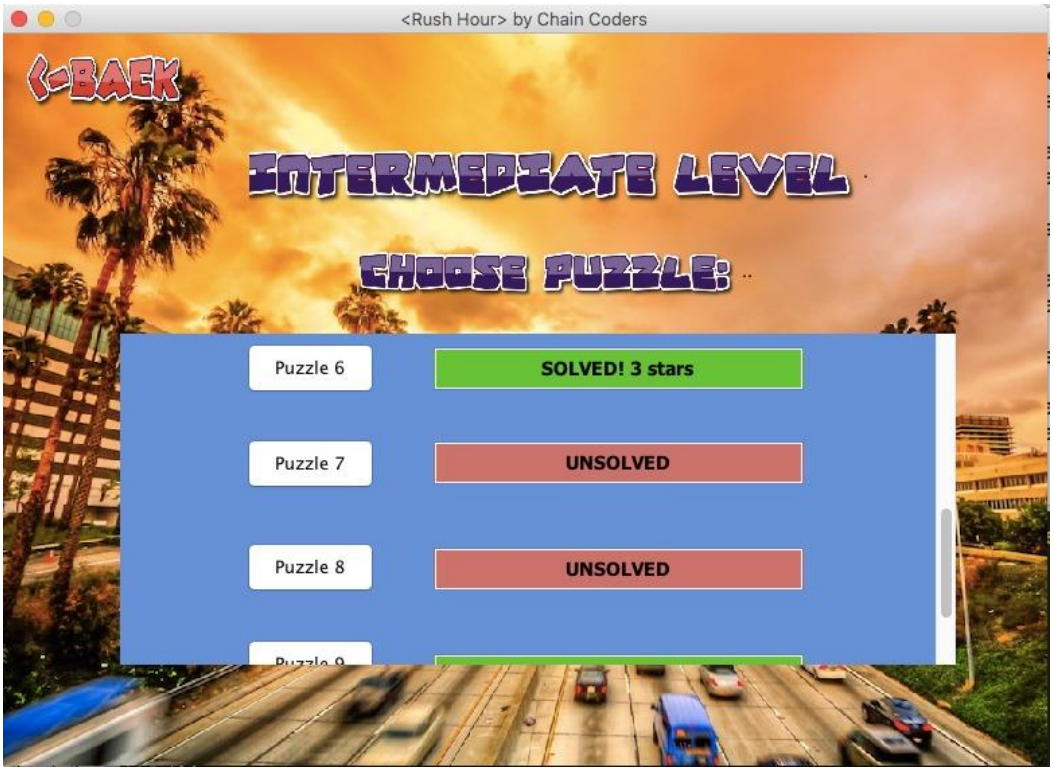
3.3 Choose Difficulty

Upon choosing the *play* option, this panel appears. It allows the user to select a particular difficulty from which to select a puzzle. The difficulties are Beginner, Intermediate, Advanced and Expert. This panel will only the user to select a difficulty if a certain threshold for the number of obtained stars is met.



3.4 Choose Puzzle

This panel will allow the user to select a particular puzzle from the previously selected difficulty. The user will have the option to select one from ten different puzzles.

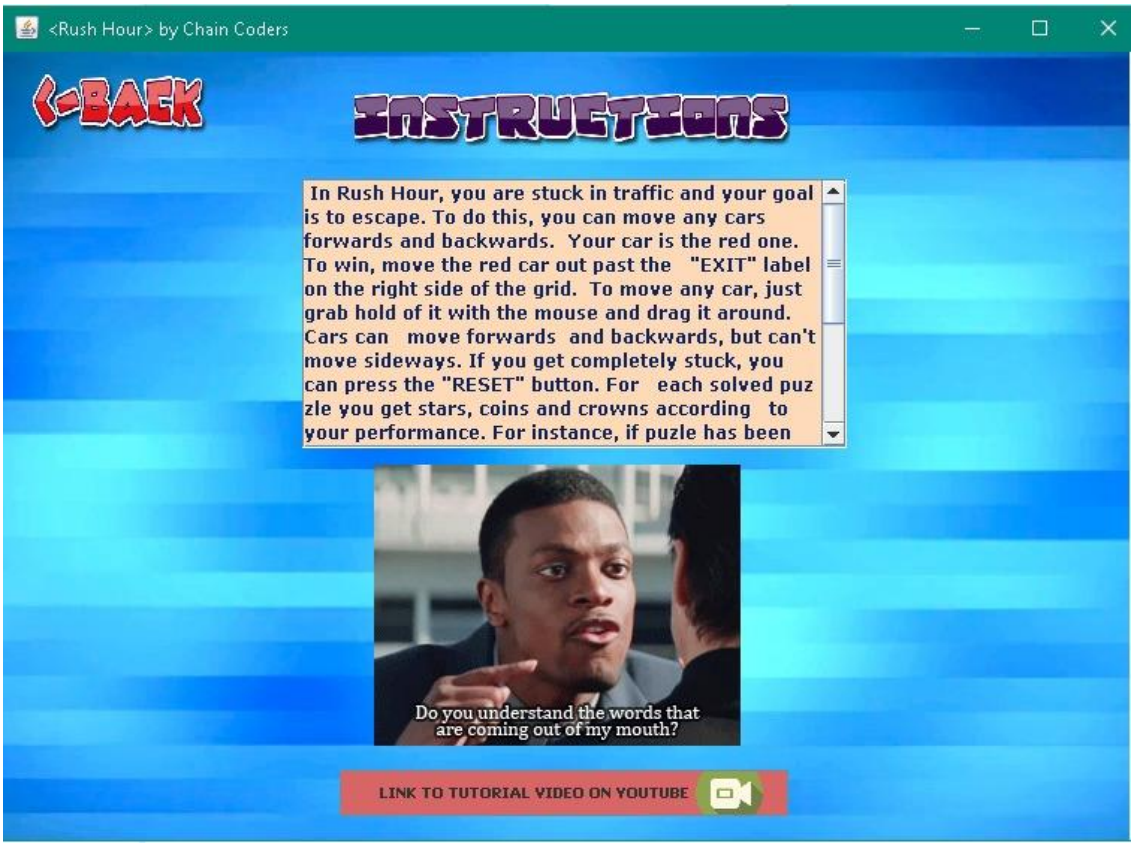


3.5 RushHour

This is the main game panel that allows the user to play the RushHour game. The user can slide the cars in any manner so that the red car reaches the exit. The user also has the ability to *explode* one car thereby making the puzzle easier to solve, however, at the cost of stars and coins. Once the user solves the puzzle, he is awarded a certain number of coins.

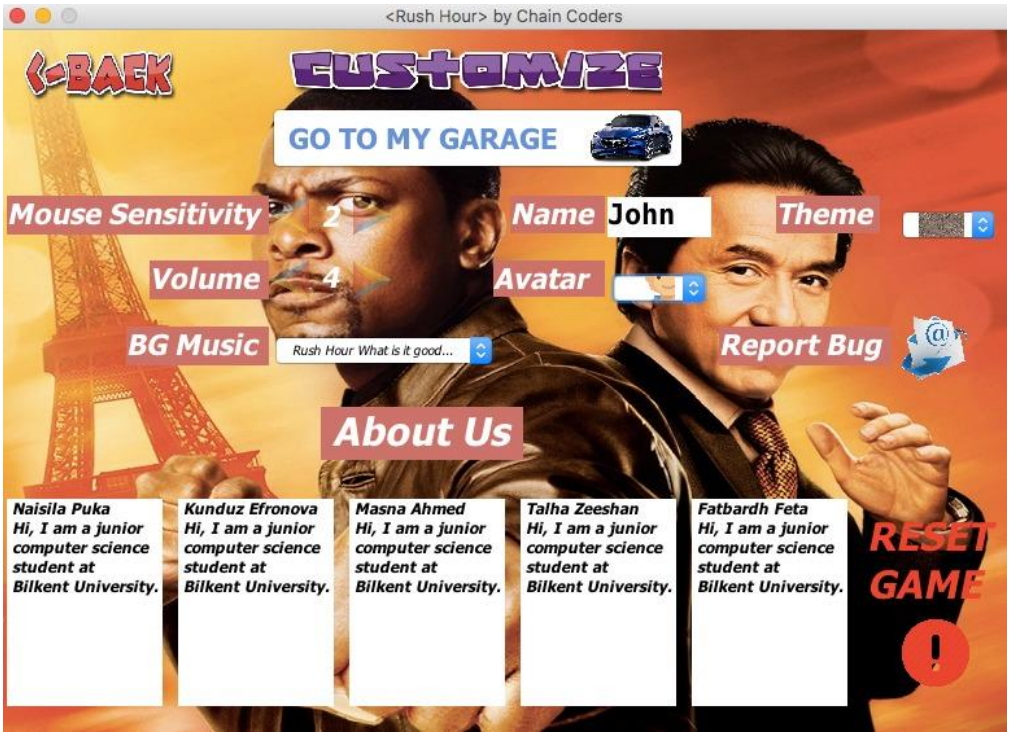
3.6 Instructions

From the main menu the user can access the instructions panel. This will contain textual instructions about how to play the game of RushHour. It also contains a link to a YouTube video containing a tutorial for the game.



3.7 Customize

From the main menu, the user can access the customize panel where he can customize the game's mouse sensitivity, the game's music volume, the game's music theme, as well as the game's background theme. He can also access the garage panel where he can unlock and select new cars as their main cars. The game credits can be seen here as well.



3.8 Garage Panel

In the garage panel which can be accessed from the settings panel, the user can choose to buy various cars if he has a certain number of coins. He can then select any bought/unlocked car as his main car.

4. Work Allocation

At the start of the semester, our aim was for everyone to have sizeable contribution toward the development of the game. In the end, everyone contributed immeasurably to the project with dedication, enthusiasm and passion.

4.1 Masna Ahmed

During the development of game's first iteration, Masna was responsible for designing the game's ChooseDifficulty and Garage Panels. He was able to design them successfully using Eclipse's Windows Builder and Java's Swing library. In the second iteration he was

tasked with developing the game's file management system was he was able to implement successfully.

4.2 Kunduz Efronova

Kunduz was responsible for designing the main menu panel during the first iteration of the project. She was able to design it successfully using Eclipse's Windows Builder and Java's swing library. In the second iteration she was tasked with developing the instructions panel, and the game's theme music functionality.

4.3 Talha Zeeshan

Talha was tasked with designing the ChoosePuzzle panel for the game during the first iteration of the game which he was able to do successfully using Eclipse's Windows Builder and Java's Swing library. In the second iteration he was tasked with developing the game's puzzles and their configurations in text files which were to be used in tandem with the file management system.

4.4 Naisila Puka

Naisila was tasked with developing the Registration panel, the Game Panel and the RushHour game frame during the first iteration. She was able to implement this successfully on the Eclipse IDE. During the second iteration she was tasked with developing the customize panel and linking all panels with the file management system.

4.5 Fatbardh Feta

Fatbardh was tasked with developing the game grid during the first iteration which he was able to successfully implement on NetBeans IDE. His task during the second iterations was to further improve and add functionality to the Game Grid. He was able to develop a fully functional game grid with additional functionality such as 7x7 grids,

exploding cars and gifs.