Bilkent University

Department of Computer Engineering

Object-Oriented Software Engineering Project

CS 319 Project: Rush Hour

Final Report

Project Group: Quintuple Whopper

Group Members: Doğa Acar, Tunç Zerener, Ahmet Atahan Mutlu, Sarp Tekin, Ahmet Emre Zengin

Course Instructor: Eray Tüzün

December 23, 2018

Final Report

CS 319 Project : Rush Hour

1. Introduction

We have successfully finished our project. Several levels have been added including special levels. We have added appropriate GUI to the game. Background music and sound effects are working neatly. Now, our report consists of a tutorial on how to set up our game.

2. Design Changes

Generally, we kept the previous design we have mentioned earlier, but some of the methods and variables are changed in the implementation process. We also had to add some methods that we could not think before.

We changed the design of our storage. We used to store levels in arrays, but now we're storing them in arraylist. In the previous implementation of our game there were some bugs such as background music turning itself off after playing for a while. We have successfully fixed these problems so sounds and background music are working properly. Background picture has been updated so that it matches with atmosphere of the game.

3. Lessons Learnt

At each part of the project we have learned new lessons which can be pretty useful in the future. Firstly, we have learned that task sharing is crucial for our project. All members of the group have different skill sets, and it is ideal to make sure that everyone uses their strong sides for a better end result. But this also means that everyone will have different opinions about the functioning of the game, thus we have to compromise and reach a common point. Since we are adapting a board game to computers, we had the analyze the board game properly to meet the expected requirements from a computer version. In this analysis part we have experienced some divergence on game mechanism and we dissolved this by exchanging ideas and not just obsessing on our own opinions but also asking what a user would expect from the game.

At the design process we had to work together to form a stronger perspective of a final object design. Distributing roles at this part could result in a weaker design. In this part we have learned more about creating a useful architecture and managing subsystems to form a sensible system and became aware of how critical the design will be for the implementation process.

Architecture of our system came in very handy at the implementation process. There weren't any vital changes needed thanks to the efficient architecture. We have added much to our knowledge on Java, and we have learned how to use GitHub more efficiently for the project.

We have realized that we attached less importance to GUI than we should have and that GUI is one of the most important parts of a project. Finally, we have successfully created a fully compatible GUI. Thanks to our experiences in this project it will be easier for us to overcome any problem in the feature regarding object oriented software programming.

4. User's Guide

4.1 System Requirements & Installation

Our game Rush Hour does not require high system requirements. Any computer that has java in it will be able to run it. The game can be downloaded from our GitHub page. (https://github.com/aatahanm/1C_CS319Project_RushHour). The game can be played clicking the jar file of the game. Furthermore, if the user wishes to see the code and make some changes on it. They can download a java IDE(NetBeans, Eclipse, IntelliJ etc.) open the game as a project and change the code or review it as they wish. After compiling and running the game in IDE they will be able to play it.

Windows: To install our game firstly you need to clone our repository to your computer from GITHUB. Then, open your IDE and click the import project button in the file segment. A window will appear and you need to show the class path of the project. To do this, you need to select RushHour class and press open button. When your IDE ask you about the SDK version, you need to select "Java\jdk1.8.0_131". Now, the game is ready to play just click Run and play.

4.2 How to Use

When a player opens the game, the main menu shows up first. It contains 3 options "Play", "Tutorial" and "Credits". Furthermore there are 2 icons on the bottom; one for sound and the other for music. The user is able to click on them to toggle them on or off. Music is for the theme music and sound is for the game sound effects such as clicking or moving the cars. If the user clicks on the "Play" button it takes the user to a level select screen where the unlocked and locked levels are shown. If the user clicks on an unlocked level, that level will start. The game screen contains a board where the cars will be placed. On the right side of the board there will be a time and move counter displayer. Which shows how much time passed and how many moves the player made. In special levels there isn't a time counter, instead there is a countdown. Time spent and how many move a player made affects player's score. Below the counters there are 2 buttons next to each other. One takes the user back to main screen and the other restarts the level. While playing the game, the player has to move the cursor to the vehicle that he/she wants to move. If the player presses left click on a vehicle, vehicle goes to the right or downwards.

Opposite for the right click. When the main car reaches the exit, a notification window appears; saying the player passed the level. Then, score of the player and the high score will be shown. There are 2 options after finishing a level: “Select Level” and “Next Level”. As their name implies “Select Level“ takes the player to “Select Level” screen and “Next Level” will initialise the next level for the board. In the main menu, if the user clicks on “Show Credit”, name of our project group members are shown. If the button “Tutorial” is clicked, there will be several instructions and notifications shown. Furthermore, there is a house shaped button on the right bottom corner on every screen which will take the player back to the main menu.

5. Work Allocation

Ahmet Atahan Mutlu: contributed to all reports worked on the core and interface of the game
Doğa Acar: contributions on Design & Analysis & Final reports worked on Level designs and generations as well as game graphics.
Tunç Zerener: contributed on all reports, worked on graphic design and sound effects in the game.
Ahmet Emre Zengin: contributions on Design & Analysis & Final reports helped with reading Level from files.
Sarp Tekin: contributed on all reports dealt with running the game in cross platforms (Ubuntu).