CS 319

Object Oriented Software Engineering

RUSH HOUR CC

Final Report

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

Supervisor: Eray Tüzün

November 18, 2018

Table of Contents

1. Implementation Process

After thoroughly building up the whole logic of game in Design Report and understanding main gameflow, we came to conclusion that we were ready to start implementing code, and , therefour, our group started implementation of Rush Hour immediately after first iteration of Design Analysis report. We chose Eclipse IDE for implementing code since every member of our group was familiar with this particular IDE. However, we are still deciding between Eclipse and NetBeans. Additionally, we used Eclipse's Windows Builder to help us out with GUI part of our project. As specified on web page of CS 319 course, we used our GitHub accounts which we opened in the beginning of semester to push and pull our part of code in order to merge them.

We started implementation of Rush Hour with some GUI parts of our project. At first we did not distribute workload, but rather implement parts of GUI we choose ourselves. For instance, Masna chose to implement ChooseLevel screen, Naisila chose to implement Game screen and My Achievements screen, Fatbardh chose to implement Main Menu screen, Talha chose to implement Choose Puzzle screen and Settings screen, Kunduz chose to implement Registration screen and My Garage screen and so on. While implementing our parts we were constantly interacting with each other to discuss our struggles and help each other out in this long and hard project. Implementatıon of source code was neither extremely hard nor confusıng sınce we have designed everything related to low level in Design Analysis stage. Talha and Masna focused on Controller part of MVC, Naisila, Kunduz and Fatbardh focused on Model part of MVC. However, as mention earlier we constantly ask each other for advice and help with code.

In spite of the fact that we have our design report we were not able to complete implementation of source code. However, I think we have done enough for first iteration of final report and first demo. We believe we will be able to complete game implementation before second iteration of final report and second demo. This report contains part of our project we were able to finish before first demo deadline.

1.1 Main Menu Panel

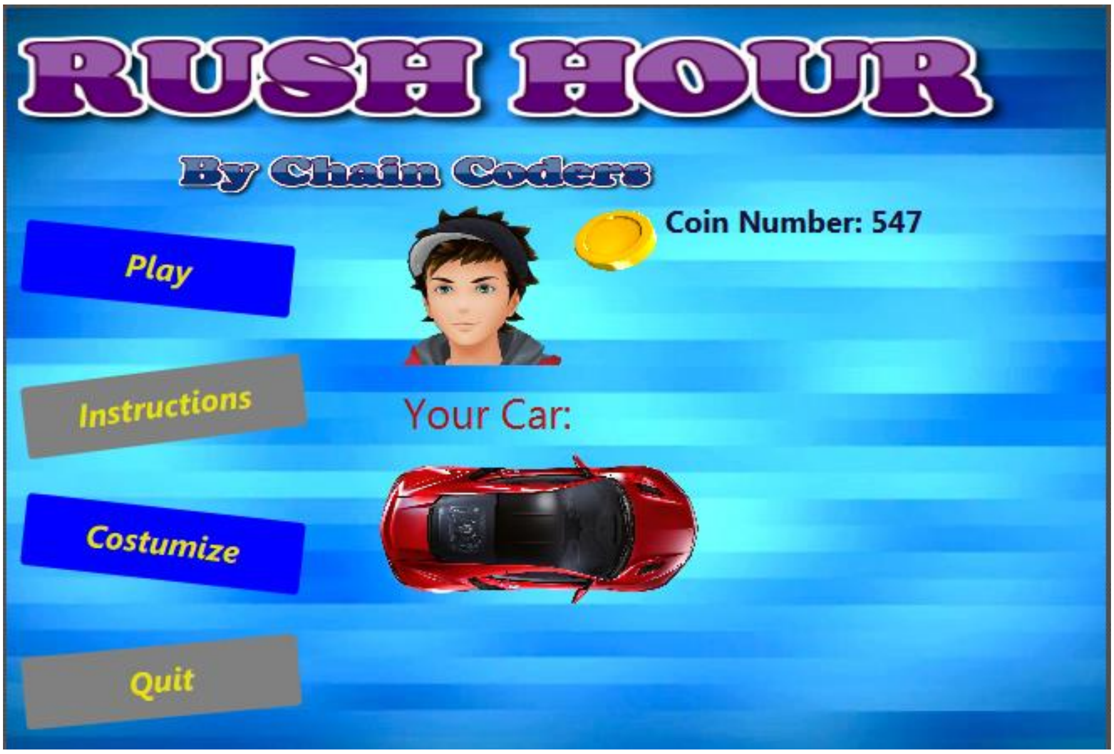


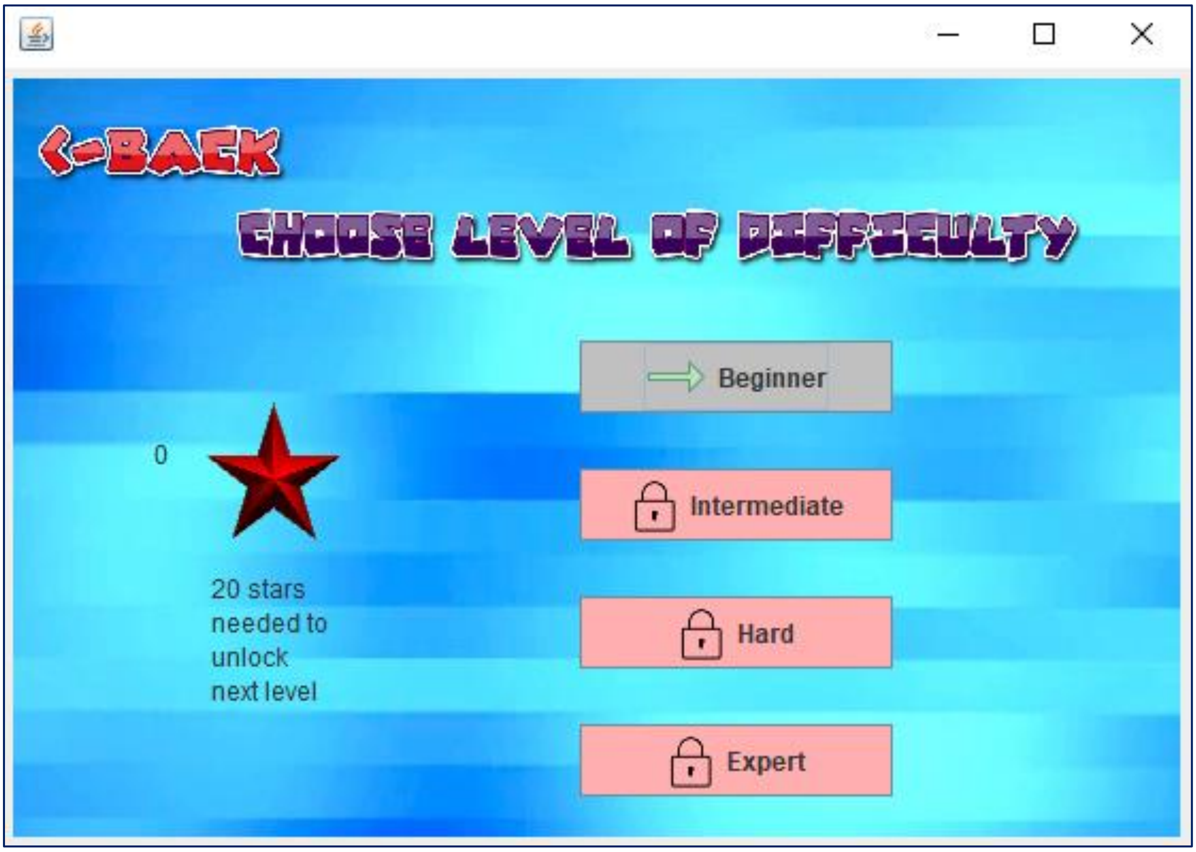
Figure 1 Main Menu Panel

This is the Main Menu panel of our game. In this panel we allow the user to navigate to the LevelSelection panel, Instruction Panel, CustomizePanel and to quit the game. MainMenu panel also shows the player coin number , avatar and the car that the user is using. The player can change any of those properties in the Customize menu.

This panel differs from the other panels that we have because it was build using Java Fxml. For this panel we wanted to offer a more appealing view to the user. We decided to build it with JavaFx and used the SceneBuilder2.0 tool to construct the GUI. To build this panel I firstly built the fxml document using SceneBuilder2.0 and then connect the GUI components to the controller of this panel.

We changed the implementation method for this panel in order to test if fxml implementation was better than the normal implementation which we did for the other panel.

1.2 Choose Level Panel



<-BACK
CHOOSE LEVEL OF DIFFICULTY
Beginner
Intermediate
Hard
Expert
0
20 stars needed to unlock next level

Figure 2 Choose Level of Difficulty Panel

When the user presses the play button, this is the first screen that appears that allows the user to choose a particular level of difficulty for the puzzles that will appear in the subsequent screens. This is the ChooseDifficulty panel containing a label named "Back" that will allow the user to return to the previous screen. The textpanes with text '0' and '20' stars needed to unlock next level' will change based on the user's progress as the game advances. Currently, only the first difficulty level is unlocked however as the user obtains more stars, the other levels will unlock automatically. Choosing a level of difficulty will open the ChoosePuzzle panel whose functionality is explained later.

1.3 Choose Puzzle Panel

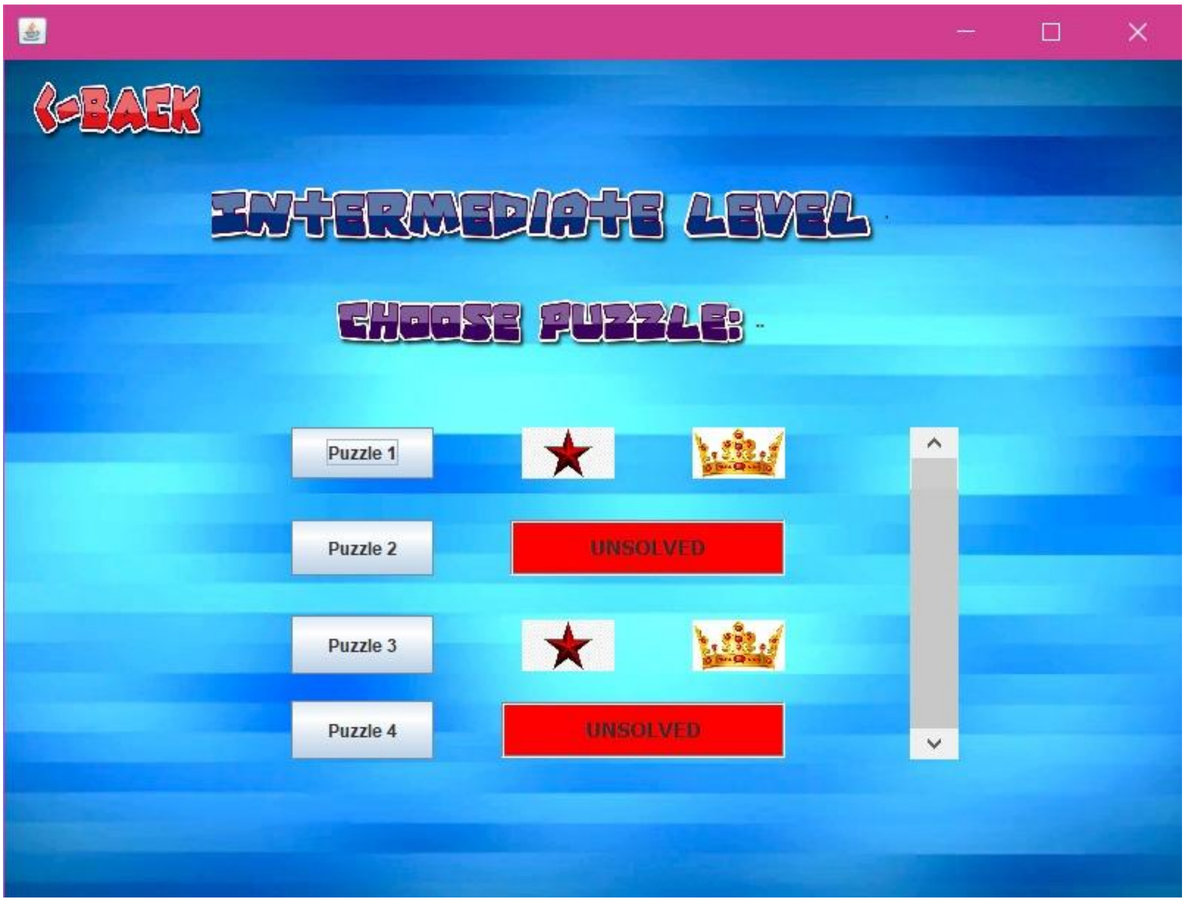


Figure 3 Choose Puzzle Panel

When the user has chosen the difficulty level he is wishes to play at (is comfortable with), the “Choose Puzzle” screen shown above is presented to them. The puzzles corresponding to the shown difficulty level are listed for the player to choose. Unlike the difficulty levels which are locked and require that the player has a certain number of stars to proceed further, the puzzles are not locked and the player can choose any puzzle he wishes. The player is also allowed to replay any puzzle he wishes to. Information regarding the performance of the player (number of stars and crowns they were rewarded) on a certain puzzle is displayed next to said puzzle. If the player replays the puzzle and gets a better result, the information is updated. Puzzles not attempted by the player have "UNSOLVED" written next to them so that the player can easily find and complete puzzles he/she has not attempted.

1.4 Game Panel

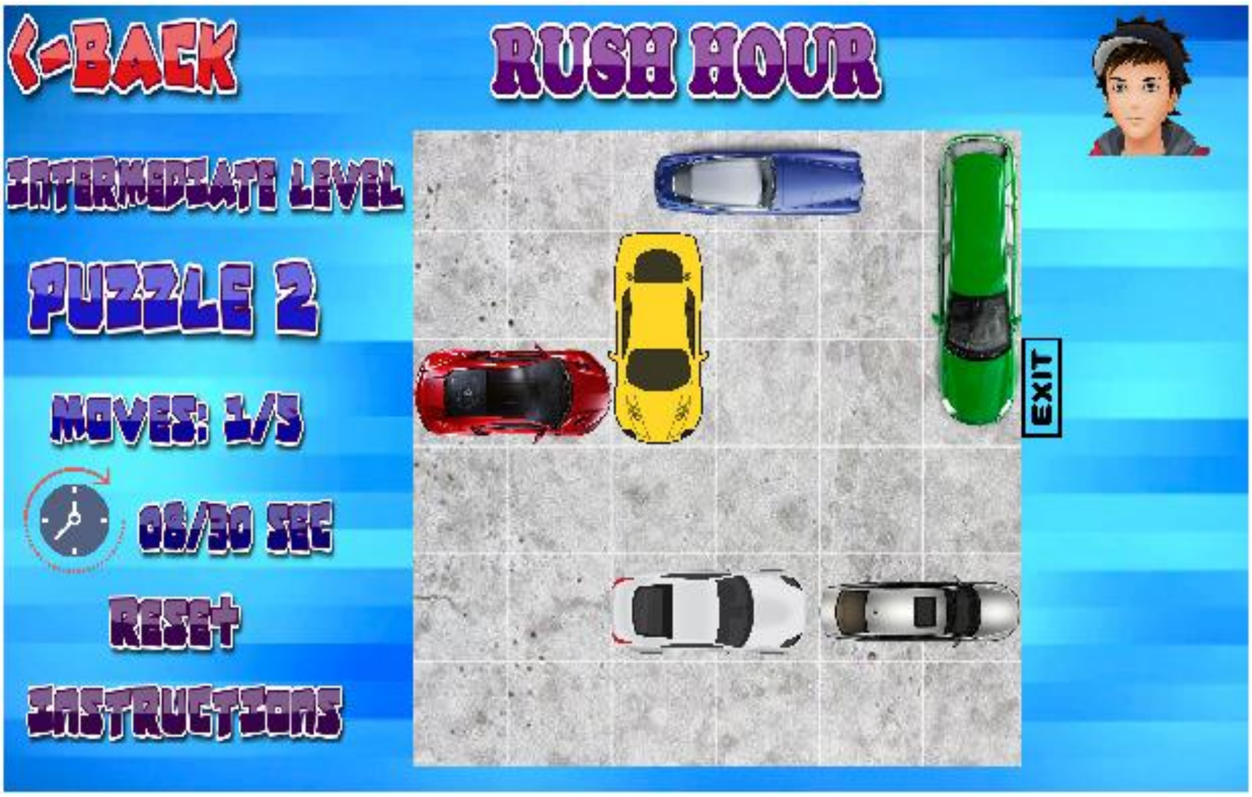


Figure 4 Game Panel

This is the Game panel of our game where actual game starts. At the top user can see main label with name of our project and right next to it avatar of user. Game panel also contains label Back to go back to Choose Puzzle screen, texts to display level and puzzle number, label to display number of moves used and number of moves allowed to use, label to show time spent and time allowed to spend, label Reset to reset puzzle, label instructions to go to instructions screen. The most important part of this panel is obviously game grid containing cars and exit. This panel is not fully functional right now because by the time of writing this report we have not fully implemented source code essential to functionality of GUI.

2. Implementation Reflections

We could not finish the game and some user interface details are missing, however we have learned how to work with Eclipse & NetBeans (we are still elaborating on the right framework to use for this project) GitHub, javafx, css etc. Our plan is to complete implementation before final demo.