Bilkent University

Department of Computer Engineering

Object-Oriented Software Engineering Project

CS 319 Project: Rush Hour

Analysis Report

Project Group: Quintuple Whopper

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

Course Instructor: Eray Tüzün

Progress Report
October 21, 2018

Analysis Report

CS 319 Project : Rush Hour

1. Introduction

1.1 Purpose

This report is a description of our implementation of the block puzzle game “Rush Hour”. This report includes information about the origin and overview of the game. Furthermore, it includes UML diagrams to describe functional, non-functional requirements, use-case models and diagrams.

1.2 About the Game

“Rush Hour” was created by Nob Yoshigahara in 1970s and released to the market by the company “ThinkFun” in 1996. It is simply a sliding block puzzle that can promote logical progression, problem solving and sequential thinking skills for kids. In addition, it has high difficulty puzzles that can be challenging for adults as well. The game has 4 types of components board, puzzle cards, cars and trucks.The game is set on a 6x6 grid board with grooves to slide the cars and trucks. The difference of the trucks and cars is that they have different sizes. Cars are 2 block long, 1 block wide and trucks are 3 block long 1 block wide. The player chooses a puzzle card and places the cars/trucks according to the picture in the puzzle card. The object of the game is to drive your red car out of the playing grid moving other cars/truck on straight lines. Any kind of rotation is forbidden in the game so the cars/trucks can only move forward and backward. The base game’s puzzle cards have numbers on them that implies the difficulty of the puzzle from 1 to 4. However there are some special editions that has additional difficulties as well. For example “Rush Hour:Ultimate Collector’s Edition” has an extra difficulty called grandmaster which is more challenging than the base games’ puzzles.

1.3 Content of the Project

Our game will be for computers so it will have some additional features that are not present in the physical game.

Level Progression: Unlike the physical game, the player cannot play any game that they desire. In order to play a specific level first you need to beat all the games that came before that specific game.

Counters: There will be two kinds of counters which are time and moves made. The player will get more points if they finish the game in less time or less moves. The player will have to finish the game as soon as possible with least amount of moves. This will add a new dynamic to the game which makes it more challenging.

Special Missions: There will be special missions that will appear as the player progresses through the game. The player will have limited time and moves. If the player fails to finish the game before time runs out or they use all their move chances, they will lose. Furthermore vehicles in the game will have special designs for each stage. For example, in the time limited puzzle you will be an ambulance trying to reach the hospital before losing your patient (before the time runs out).

2. Overview

In general, Rush Hour is a puzzle game. Besides, it is beneficial for improving children's intelligence, many people like this game to spend their free time. When Rush Hour is started, users are going to see the main menu, firstly.

2.1 Main Menu

There are five buttons. Play button to start the game. For players who do not know the game, there is a guide, called “How to Play”. When it is clicked, a frame appears and teaches to play Rush Hour. There is a square area in the game, there are vehicles as obstacles and a main car to be taken out from the area. The car has to exit from the area in the given time in special levels. However, it is not that easy. There are other vehicles as obstacles. Firstly, to exit from the area, users have to open the path of exit of the car. Then, the car can go out and user can play the next level. Another button is “Credits”. Here, some information appears about who are the developers and the references. In Rush Hour, there are music and sound buttons which can be turned off and on with respect to player’s preference.

2.2 Level Selection

After pressing the play button, the game shows level page. There are different levels with different difficulties. At the beginning, levels are easier. Obstacles can be handled easily, however, the higher level user reaches, the more difficult it gets to solve. Not only puzzle is harder, but also time is shorter and user has to complete the level by less moves. Lastly, another challenge is that users cannot play next levels without solving the previous levels.

2.3 Gameplay

Game is started. User is going to start to move vehicles which are obstacles and the main car in order to reach the exit. There is finite right of moves and specific time. When user wants to move vehicles and the car they need to bring mouse to on top of vehicles or the car. Then, arrows are going to appear. All vehicles and the car can move to forward or backward. Users choose which direction they want and click the appropriate arrow. During the game, user can return to the main menu. They can replay the level or by clicking the select level they can play other levels which are unlocked.

3. Functional Requirements

- Main screen
- Instructions section
- Exit game button

- Credits button
- Play button to start game
- Level menu to choose a challenge
- Selection across different traffic jams
- Main vehicle
- Plenty of foreign vehicles
- Foreign vehicles have different sizes
- Vehicles are moveable in a line on grids
- Reset button
- Exit place for main car
- Challenges get harder with progression
- Sound button
- Special levels
- Vehicles paint with special costumes at story levels
- Background music for game
- Move counter
- Time counter

4. Nonfunctional Requirements

- Only one vehicle can be moved at a time.
- Reset button should also reset the time and move counter besides traffic jam.
- Level will skip automatically once the main car exits traffic jam.
- Music will start once the game opens and keep playing until player turns it off.
- Sound buttons can also be turned off.
- Game crash should not lead to data loss of unlocked levels and level information.
- Completed levels should unlock next level button at level selection screen.
- At level screen, best score of a certain level finished by the player will be shown when the cursor is held on the specific level button.
- Score will consist of both time and move count, but time will be the main successor of highest score.
- A vehicle shouldn’t be able to move inside other vehicles

5. System models

5.1. Use case model

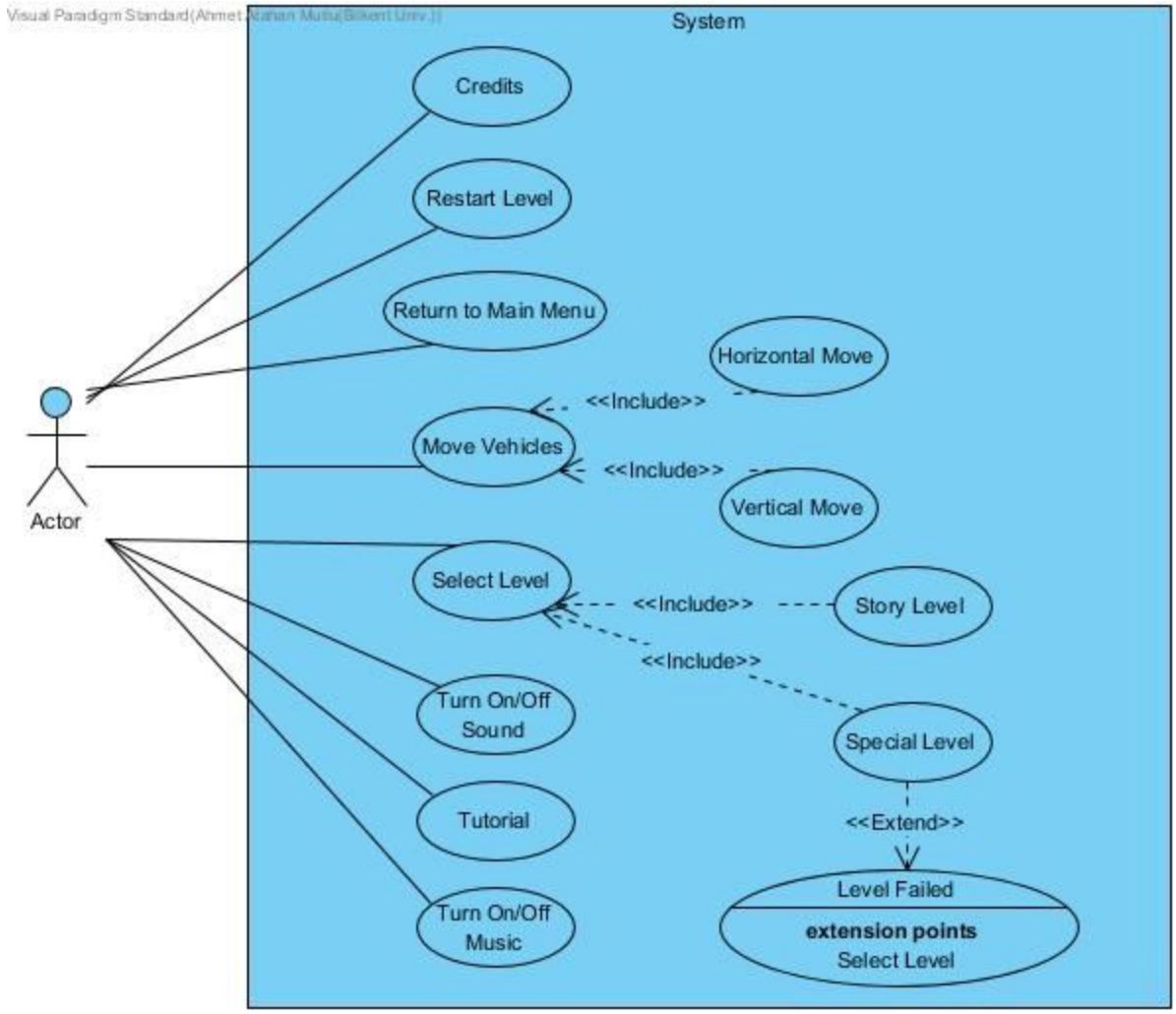


Figure 1: Use Case Diagram

Use case name: ***Select Level***

Participating actors: Player

Flow of events:
1. The user can select one of the two type game styles which contains multiple levels.
2. The system creates a map according the users selection.
3. The game starts.

Entry condition: The levels (excluding the first ones) have been unlocked by completing the previous one.

Exit condition: Player has selected a level or returned to main menu.

Use case name: ***Move Vehicles***

Participating actors: Player

Flow of events:
1. The user moves the vehicles by clicking the arrows on top of them.
2. The vehicles move horizontally or vertically depending on how they are placed inside the map.
3. The vehicles moves towards the direction the user clicks.

Entry condition: The player must have selected a level to play.

Exit condition: Player clicked return to main menu or the level failed.

Alternative Flow of events: If the path next to a vehicle is blocked by another vehicle, it does not move.

Use case name: ***Level Failed***

Participating actors: Player

Flow of events:
1. Once the time and maximum number of moves to complete the level is exceeded the level fails.

Entry condition: Special Level is selected.

Exit condition: Player restarts level or returns to main menu.

Use case name: ***Tutorial***

Participating actors: Player

Flow of events:
1. The system creates an instance of the game
2. The system pops out instructions for the player to follow.

Entry condition: None.

Exit condition: The player has completed the tutorial successfully or click return to main menu.

Use case name: ***Credits***

Participating actors: Player

Flow of events:
1. The user selects Credits option.
2. Credits will be displayed in the screen.

Entry condition: None

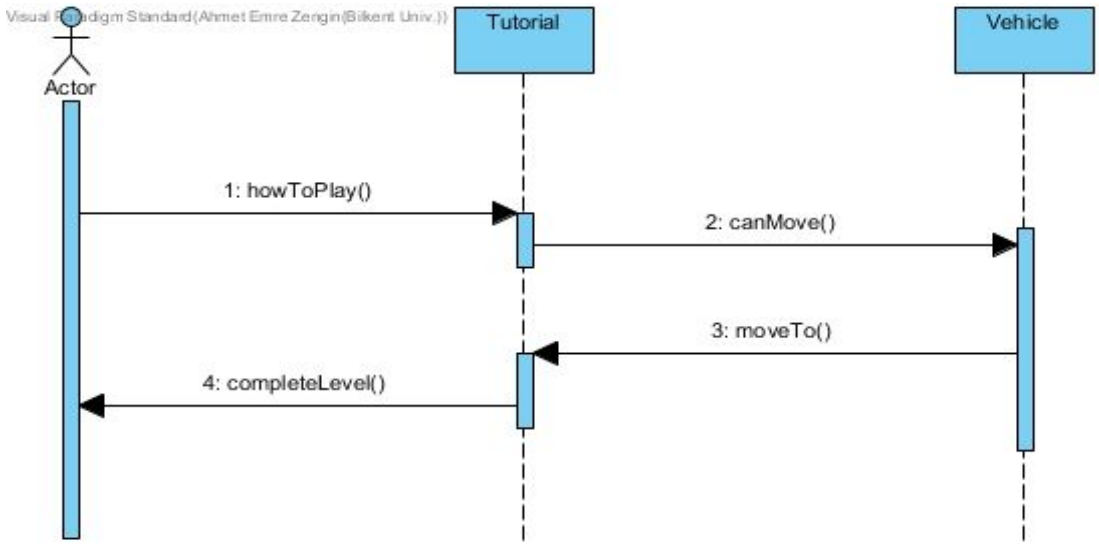
Exit condition: The player clicks return to main menu.

5.2. Dynamic models

5.2.1 Sequence Diagram

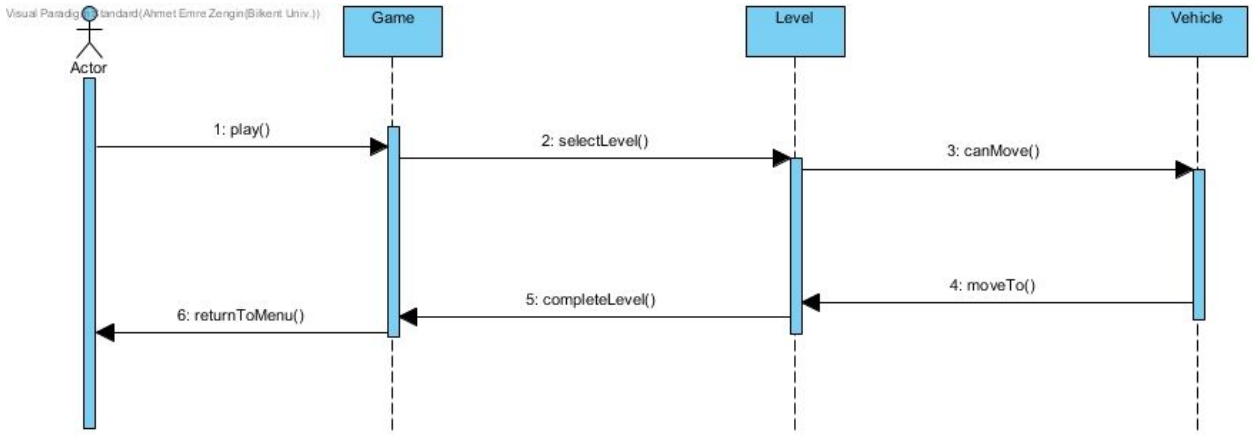
This diagram shows the process of starting and finishing a level. Also, how vehicle moves are made is shown.

5.2.1.1 Tutorial



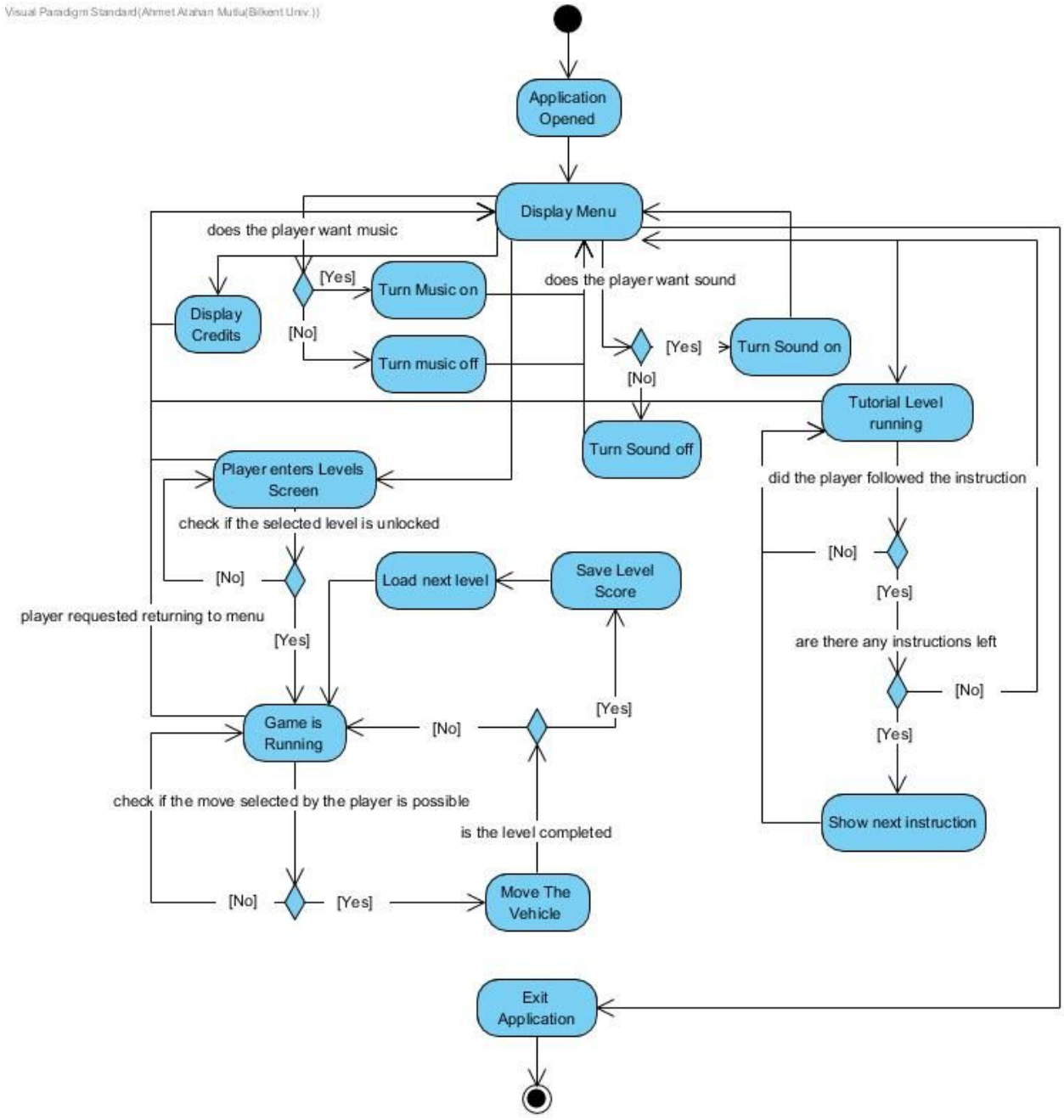
This diagram shows that what happens after pressing How To Play button.
Program opens a tutorial level which contains instructions to inform the player about how to play the game. If a user wants to move a vehicle, the possibility of movement is checked by the canMove() method. If it is possible, desired movement will happen. After solving the level, completeLevel() method takes the user to the menu.

5.2.1.2 Complete a Level



This Diagram shows what happens after pressing Play button.
The user is encountered by select level page. When a level is selected, game is going to start. The user needs to move a vehicle. To move a vehicle, canMove() method checks whether it is possible, firstly. After possible response is taken, moveTo() method makes the move of the vehicle. When the level is solved, completeLevel() returns the user to the Select Level page. By the way, if user desires, he/she can return to the menu in both selection level and during the game.

5.2.2 Activity Diagram



This activity diagram sums up the working and flow of the system. The illustration is explained as follows on the next page:

Player will open the application by clicking the icon of the game. It will open a new window which will display the main menu. The sound and music will be initially on at the beginning of the application but the player will be able to turn them off or on again from the menu. There will be 4 main options which will work to exit the application, start the game, run the tutorial, and display the credits.

Choosing credits will open a new screen where all the credits will be shown. After all the screen slides down to the end of credits the system will automatically take the player back to main menu.

If the player chooses to run tutorial the game will open up a quick tutorial level. The purpose of this level will be to teach player about the basics of the game. If the instructions aren't followed game will take the player back to the beginning of tutorial level. However, if the player follows the instructions properly the game will show the next instruction. This cycle will keep going until there aren't any instructions left. Then, game will take the player to the first level of the game.

When the player chooses to start the game directly, the system will open up the levels screen. At this screen player will have to choose a level between levels that are unlocked. If the clicked level is not unlocked the game will not start. If the selected level is unlocked the game will start running.

Once the game starts running, the player will click to vehicles to move them according to his/her planned exit route. If the move selected by player is possible the vehicle will be moved. After vehicle is moved the game will check if the level is completed. If the level is not completed game will keep running and player will keep making moves until it's completed. Once the level is completed level score of the player will be saved and next level will be loaded.

Lastly, player might select to exit the application from main menu screen. If the user chooses this option the system will close the application window.

5.3. Object and class model

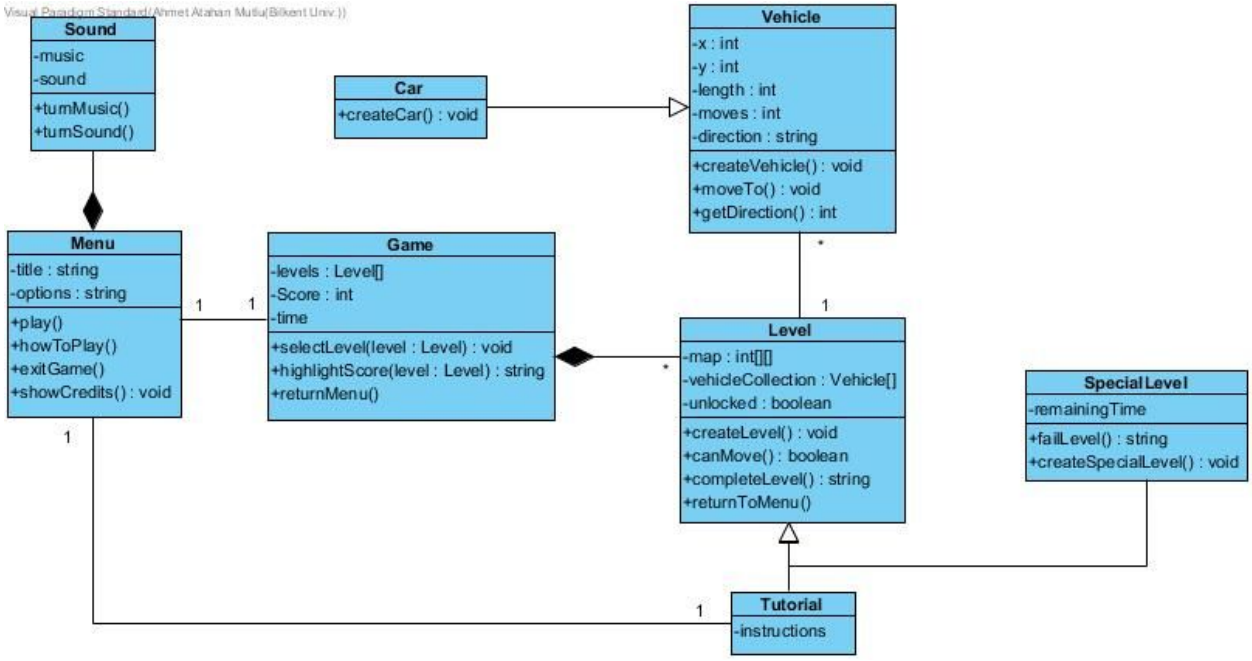
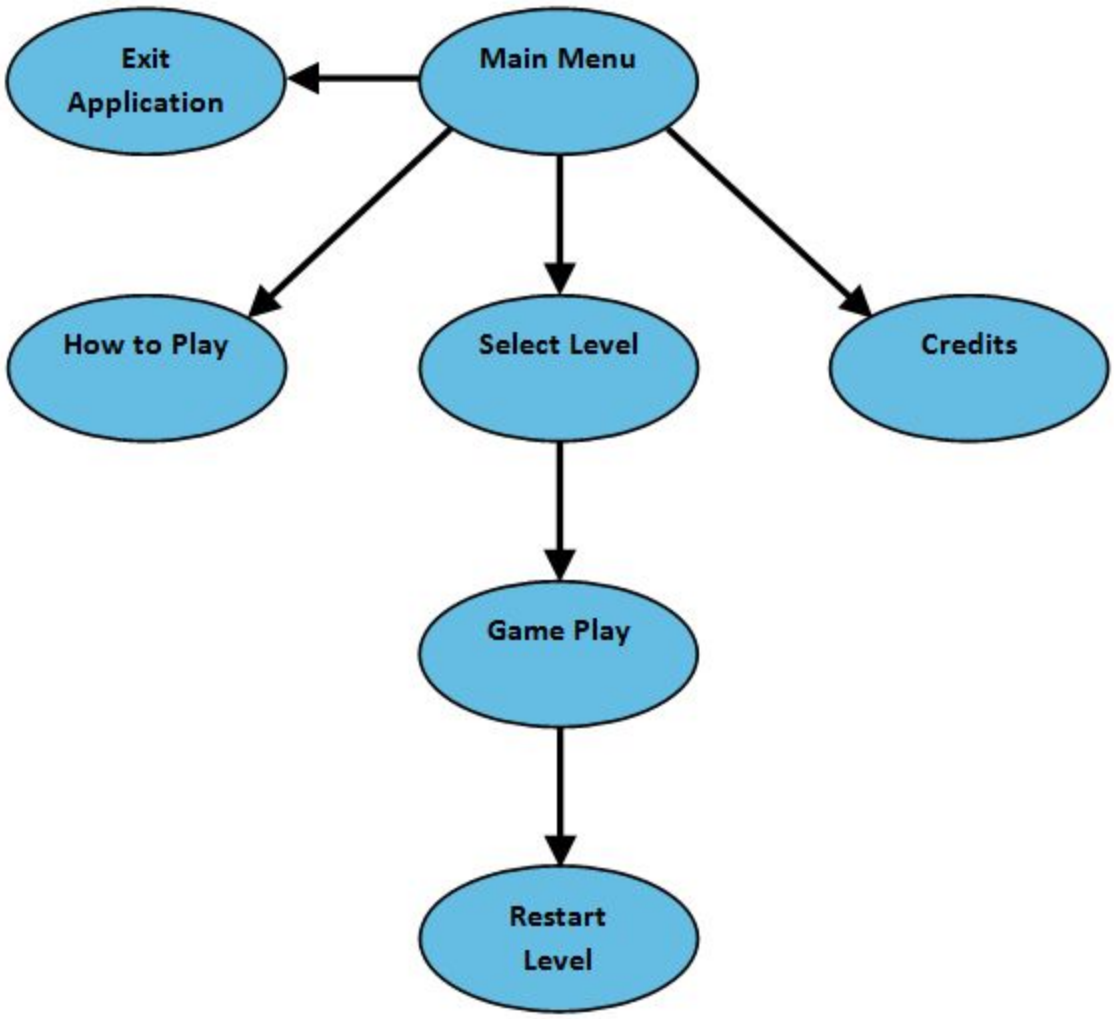


Figure 2: Object and Class Model Diagram

Figure 2 shows the structure of the system and how classes are associated with each other. Firstly, a menu greets players where they have different options. The “Menu” Class is aggregately associated with the “Sound” class where players have the ability to turn both the music and sound on and off. The “Menu” class is also 1-to-1 associated with the “Tutorial” class and the “Game” class as there can be only one instance of game running at the same time. The “Game” class is aggregately and 1-to-many associated with the “Level” class as it contains a collection of levels. The “Level” class is 1-to-many associated to the “Vehicle” class where each level is consisted of multiple vehicles and a car. This “Vehicle” class has a child class named “Car” which gives us the main goal for the game, pass the car through the exit and complete the level. The “Level” class also has two child classes named “SpecialLevel” where so called ‘themed levels’ are initialized and “Tutorial” which give a basic demonstration of how the game works.

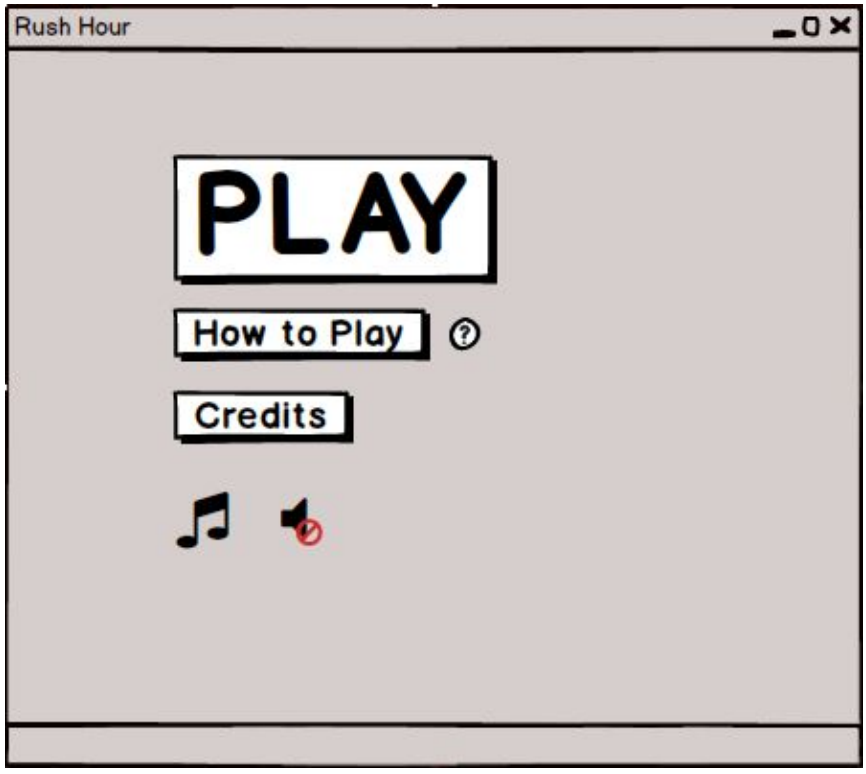
5.4. User interface - navigational paths and screen mock-ups

5.4.1 Navigational Path



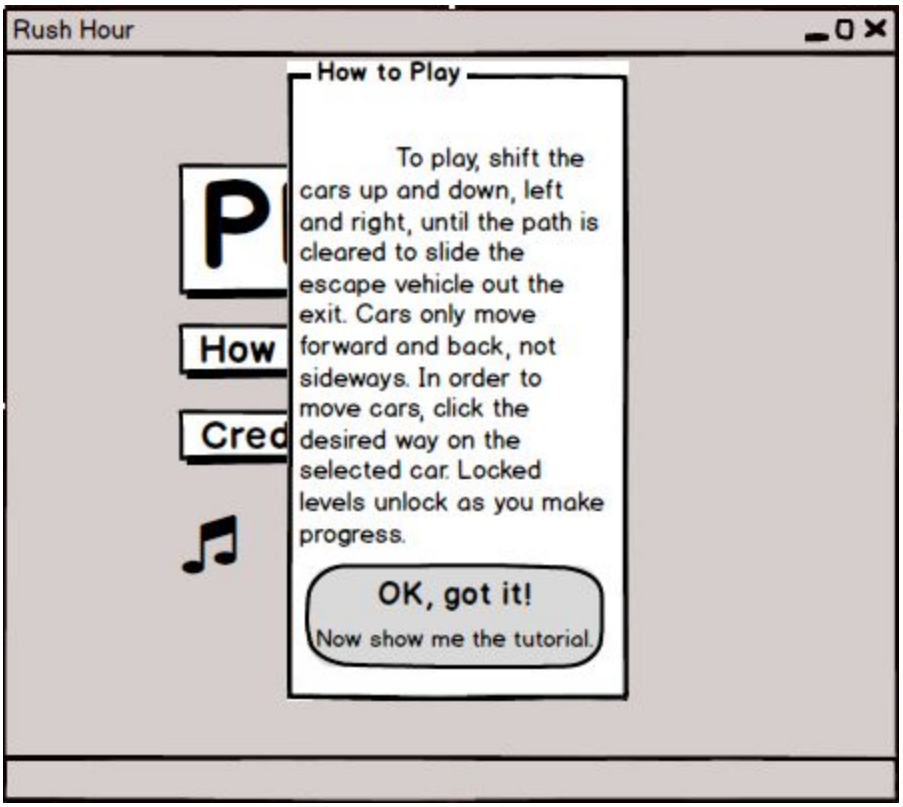
5.4.2 Screen Mockups

5.4.2.1 Opening Screen



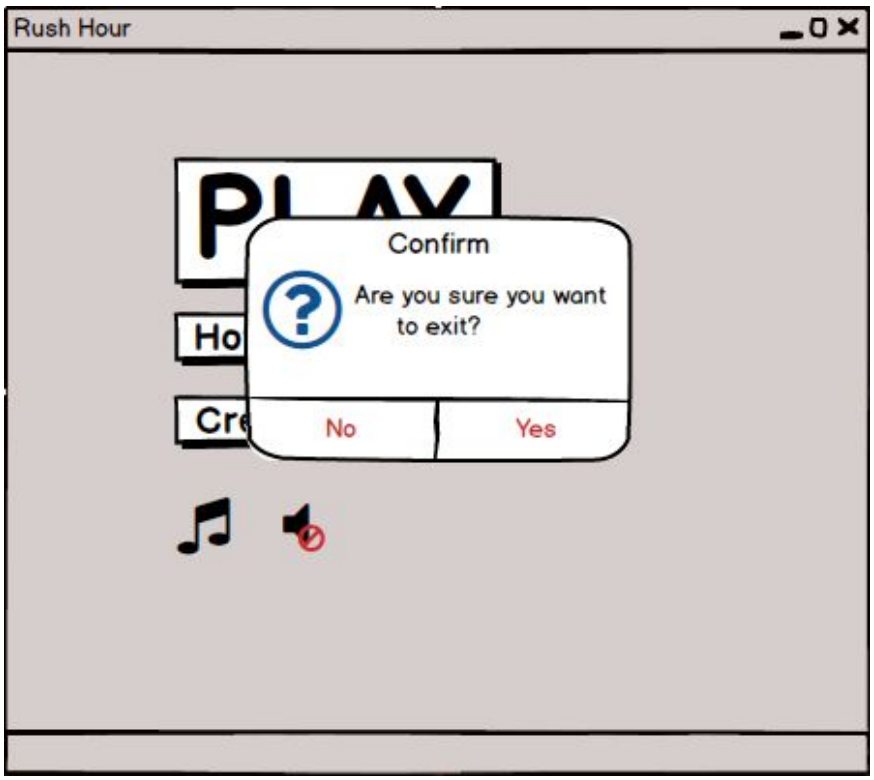
When player opens the game, player encounters this main menu. Player needs to press the “PLAY” button in order to play the game. If it is the first game of player, he/she can simply press “How to Play” button to understand how the game works. Moreover, player can turn on or turn off the music or the sound of the game by pressing note symbol. Also, player can take a look at credits in that screen by pressing “Credits” button.

5.4.2.2 How To Play Window



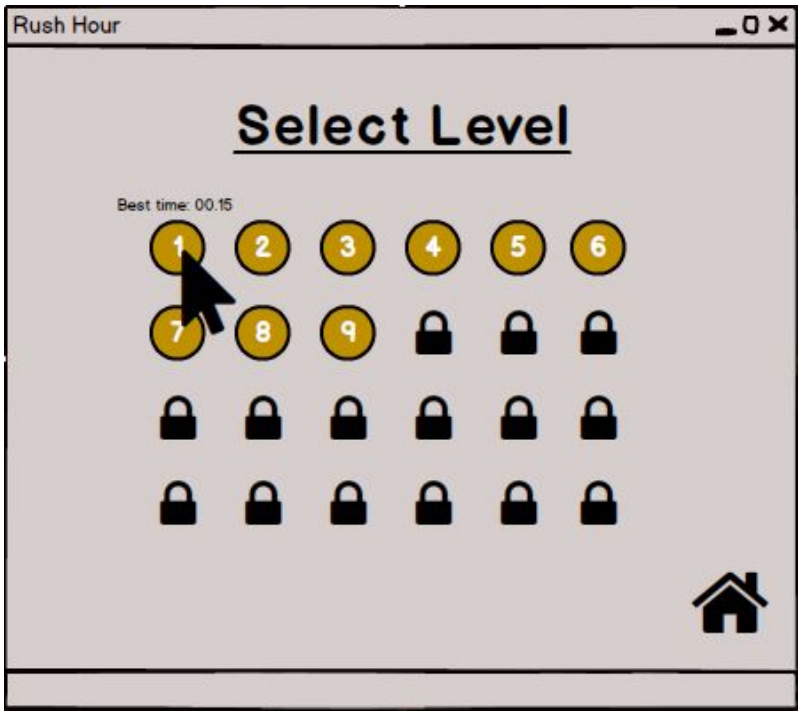
When player presses “How to Play” button in main menu, a window pops up on the screen. This window explains how the game works to the player. Then, directs the player to the tutorial.

5.4.2.3 Exit Window



If the player wants to exit, there is an exit button on the top right corner of the screen. After clicking that “X” button, a window pops up and asks to player if the player is certain about that decision. This window is created in order to eliminate pressing the exit button by mistake. If the player presses “No”, then that window closes. However, if “Yes” button is pressed, then the program is terminated.

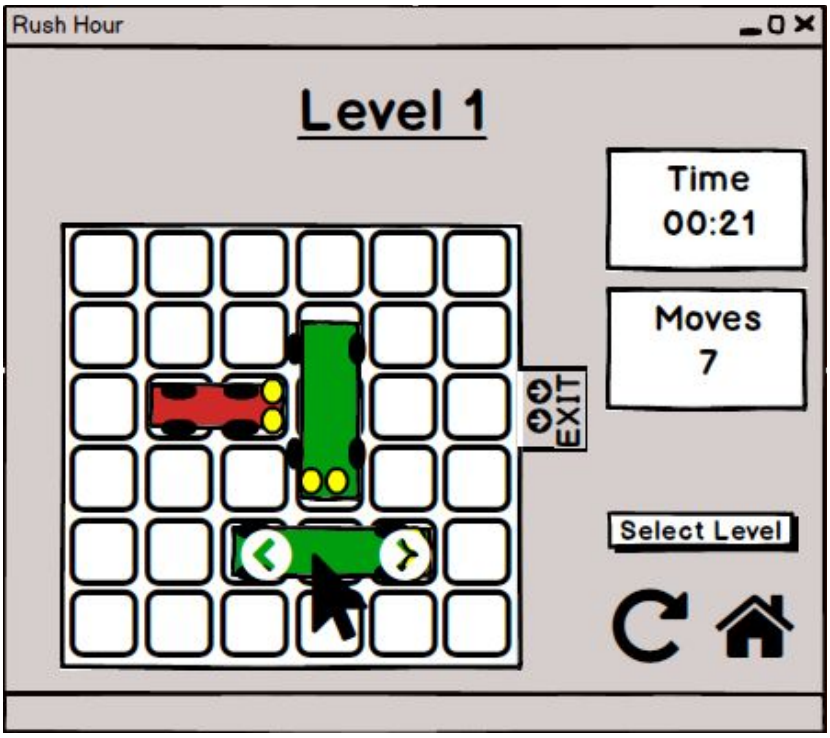
5.4.2.4 Level Selection Screen



If the “PLAY” button is pressed in the main menu, player encounters a new window which is the level selection window. In this window player chooses the level he prefers. However, the player can only play level 1, if it is the first time of player. Other levels unlock as the player completes levels. Completion of a level unlocks the next level. Furthermore, player can see his best completion time of a particular level when he moves his cursor on the level he prefers.

There is a home symbol on the bottom right corner of the screen. If the player presses this button, the player is directed to the main menu.

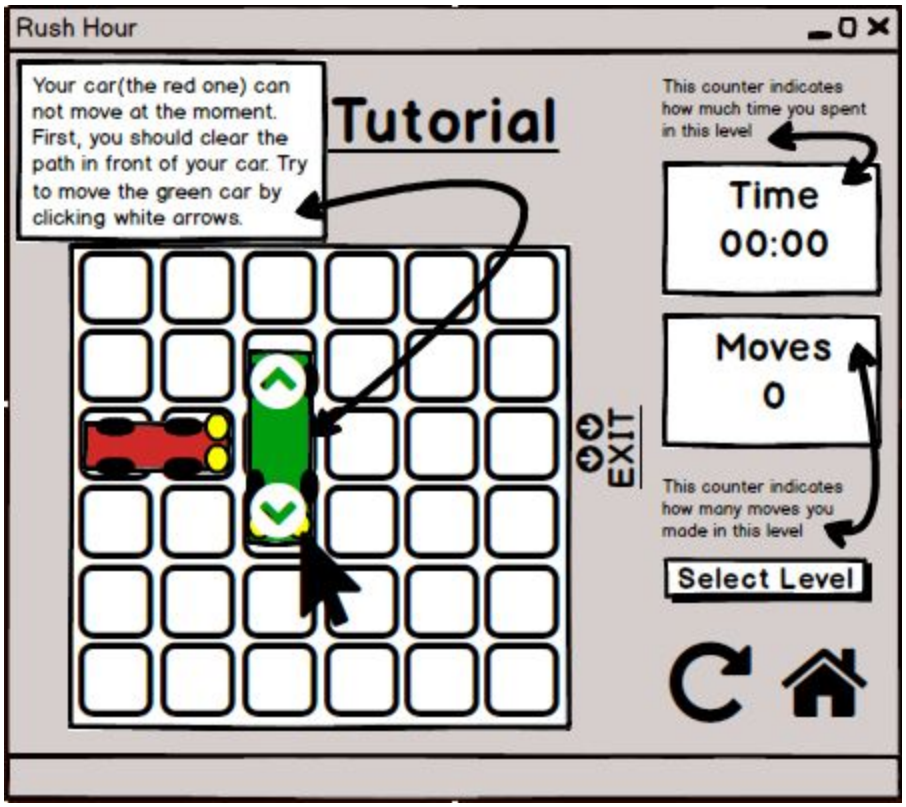
5.4.2.5 Gameplay Screen



Player encounters this screen when a level is selected in the level selection screen. In order to move a car, player should move the cursor on the car that he wants to move. Then, two white arrows appears on top of the car as it can be seen on the mockup. Car moves one unit when the player presses one of the arrows.

Also, there are time and move counters on the right side of the window. On the bottom right corner there are three buttons. One of them is “Select Level” button which directs the player to the level selection screen. The other one is the repeat symbol, it is a restart button which resets the level. The final button is a home symbol, it directs the player to the main menu.

5.4.2.6 Tutorial Screen



After the player presses the “How to Play” button in the main menu, game directs the player to this tutorial screen. In this part there is a simple level and the player receives instructions to complete the level. Furthermore, player gets information about time and move counters, home button and restart level button.

6. Glossary & references