



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

November 28, 2018

<i>Object-Oriented Software Engineering Project</i>	<i>0</i>
<i>1. Introduction</i>	<i>2</i>
<i>1.1 Purpose</i>	<i>2</i>
<i>1.2 About the Game</i>	<i>2</i>
<i>1.3 Content of the Project</i>	<i>3</i>
<i>2. Overview</i>	<i>3</i>
<i>2.1 Main Menu</i>	<i>3</i>
<i>2.2 Level Selection</i>	<i>4</i>
<i>2.3 Gameplay</i>	<i>4</i>
<i>3. Functional Requirements</i>	<i>4</i>
<i>4. Nonfunctional Requirements</i>	<i>5</i>
<i>4.1 Supportability</i>	<i>5</i>
<i>4.2 Usability</i>	<i>5</i>
<i>4.3 Reliability</i>	<i>5</i>
<i>4.4 Performance</i>	<i>6</i>
<i>5. System models</i>	<i>7</i>
<i>5.1. Use case model</i>	<i>7</i>
<i>5.2. Dynamic models</i>	<i>11</i>
<i>5.2.1 Sequence Diagram</i>	<i>11</i>
<i>5.2.1.1 Tutorial</i>	<i>11</i>
<i>5.2.1.2 Complete a Level</i>	<i>12</i>
<i>5.2.2 Activity Diagram</i>	<i>13</i>
<i>5.3. Object and class model</i>	<i>15</i>
<i>5.4. User interface - navigational paths and screen mock-ups</i>	<i>17</i>
<i>5.4.1 Navigational Path</i>	<i>17</i>
<i>5.4.2 Screen Mockups</i>	<i>18</i>
<i>5.4.2.1 Opening Screen</i>	<i>18</i>
<i>5.4.2.2 How To Play Window</i>	<i>19</i>
<i>5.4.2.3 Exit Window</i>	<i>20</i>
<i>5.4.2.4 Level Selection Screen</i>	<i>21</i>
<i>5.4.2.6 Tutorial Screen</i>	<i>23</i>
<i>6. Glossary & references</i>	<i>24</i>

Analysis Report

CS 319 Project : Rush Hour

1. Introduction

1.1 Purpose

This analysis 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. [4] It is simply a sliding block puzzle that can promote logical progression, problem solving and sequential thinking skills for kids. In addition, it has difficult 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 blocks 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 played on computers so it will have additional features that are not in the physical game.

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

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 fewer 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 Levels: There will be special levels that will appear as the player progresses through the game. The player will have limited time or moves. If the player fails to finish the game before time runs out or they use all their move chances, they will not pass that level. Furthermore, vehicles in the game will have special designs for each stage. For example, in the time limited puzzle there will be an ambulance trying to reach the hospital before losing your patient (before the time runs out). Moreover, some special levels will be movie themed. For instance, the main car will be Batmobile or in an another special level all cars will be cars from Transformers film.

2. Overview

Rush Hour is a puzzle game which is beneficial to improve children's versatile thinking ability while amusing them. Not only children, but also adults really enjoy playing this game in their free time. When Rush hour is opened, users are going to see the main menu firstly.

2.1 Main Menu

There are five buttons in the main menu. These are "Play", "How To Play", "Credits", "Music" and "Sound". To start the game, user must click the "Play" button. But if the player doesn't know how the game works, s/he can click "How to Play" button to learn the movements

and mechanics of the game. Another button is “Credits” which will include information about the developers of the game. Remaining two buttons are “Music” and “Sound” buttons so that music and sound can be turned off and on with respect to player’s preference. Moreover, the player can exit from the main menu by clicking (X) button on the top right corner.

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. Also players cannot play next levels without solving the previous levels.

2.3 Gameplay

When the game is started, a level screen is going to be opened. Levels are composed of a square grid which includes vehicles. One of those vehicles is our main car. The purpose of the game is to move our main car forward through exit path by moving other vehicles out of the way. However, it is not that easy. A vehicle you move might block the path of another vehicle you need to get out of the way. Thus, you have to think carefully to complete a level in the best time possible. When player wants to move vehicles it is needed to move cursor to on top of vehicles and click according to the direction player wants. Vehicles can only move forward or backward. Left click is used to move the vehicle right or down side and right click is used to move the vehicles left or up side. When the main car reaches the exit point, player can play the next level. During the game, user can return to the main menu. They can replay the level or by clicking the “Select Level” button they can play other levels which are unlocked.

3. Functional Requirements

- Main screen should enable players to run game functions
- Tutorial section should teach players to play the game
- Exit button should enable players to exit the game
- Credits button should show the creators of the game
- Level menu should enable players to select the level they want to play
- Selection across different traffic jams
- Main vehicle should have a distinguishable appearance
- Foreign vehicles should block the main car
- Vehicles should be moveable
- Reset button should enable players to restart the level

- There should be an exit place for player to complete the game
- Sound button should be included to enable players to have a preference
- Special levels should not change the proper system of normal levels
- Vehicles at special levels should also be moveable
- Players should have the button to turn on and off the background music for the game
- Move counter should count the moves of the player
- Time counter should calculate the time player has spent

4. Nonfunctional Requirements

4.1 Supportability

- Additional features should be added without changing the main features.

4.2 Usability

- Players must be able to understand how the game works with the help of tutorial.
- User interface should have an attractive design for the users with respect to the feedback from our survey done by target audience.
- The volume should be at less than 53 dB noise level to not be perceived as annoying by players.[1]
- A vehicle shouldn't be able to move inside other vehicles.
- Completed levels should unlock next level button at level selection screen.
- Level should be skipped automatically once the main car exits traffic jam.

4.3 Reliability

- System crash should not result in data loss.
- Highscore should be saved and displayed when the player moves the cursor on a level in the level selection screen.

4.4 Performance

- Transitions between screens should take 230-250 ms on average, which is the mean eye movement time for humans.[2]
- Operations should consume little amount of time
- The system should execute functions of the system in less than 62.5 milliseconds for 16 FPS.[3]

5. System models

5.1. Use case model

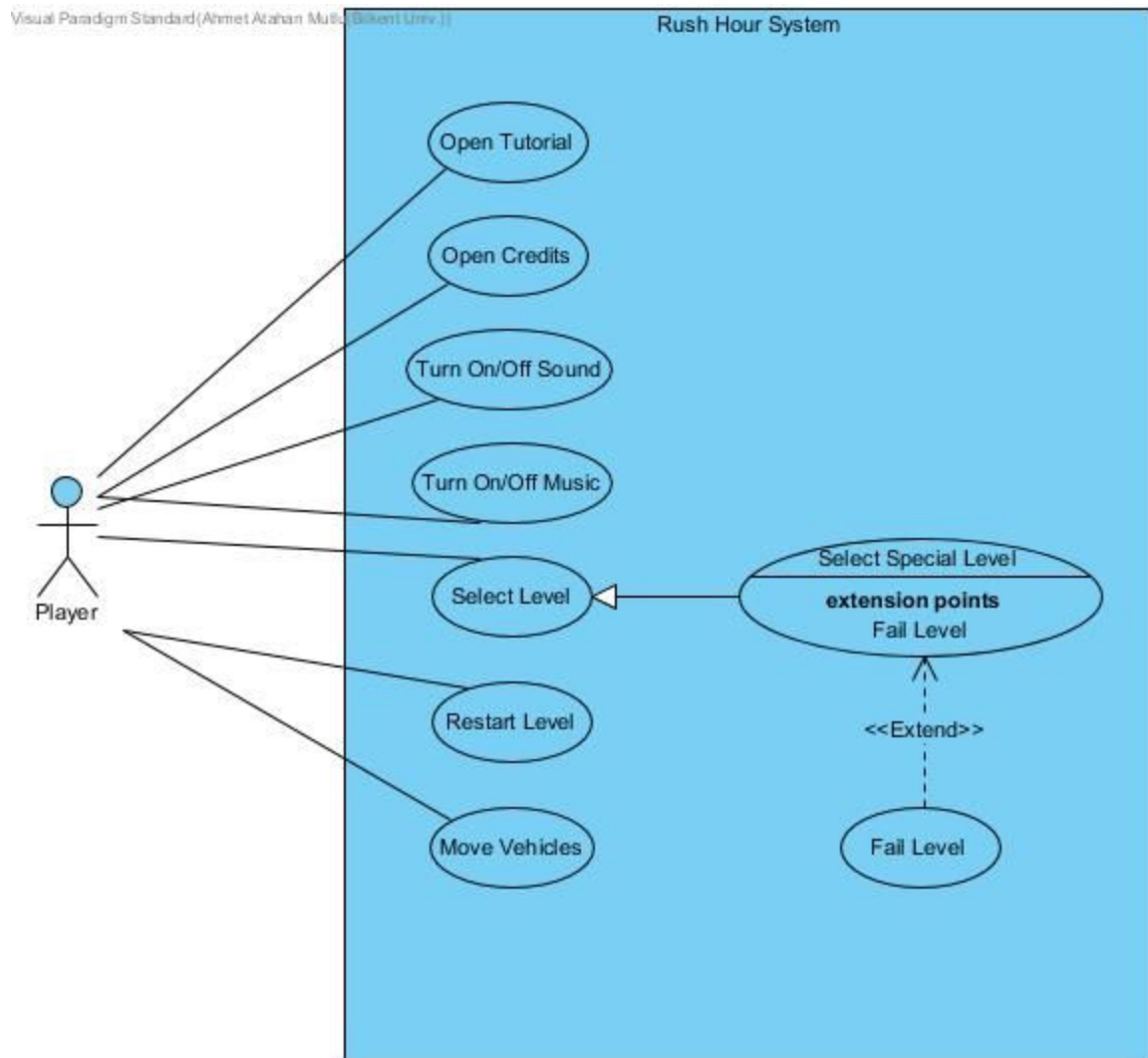


Figure 1: Use Case Diagram

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: *Open 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: *Open 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

Our sequence, activity diagrams and object model are listed and explained as follows.

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

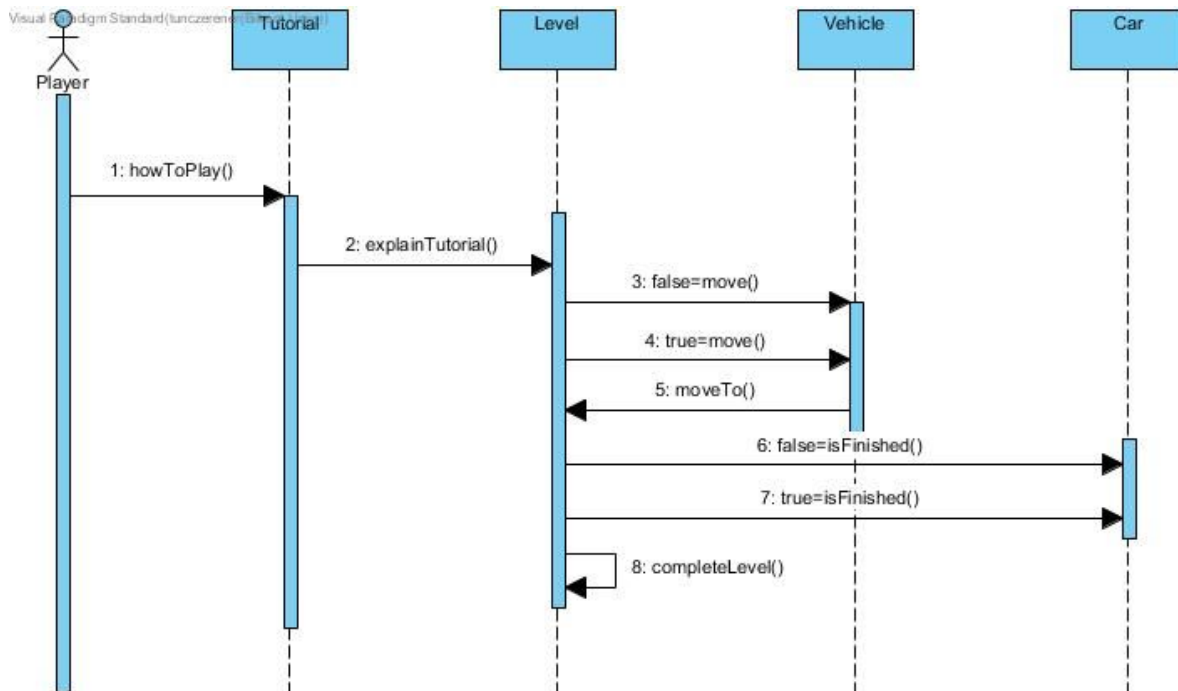


Figure 2: Tutorial Sequence Diagram

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

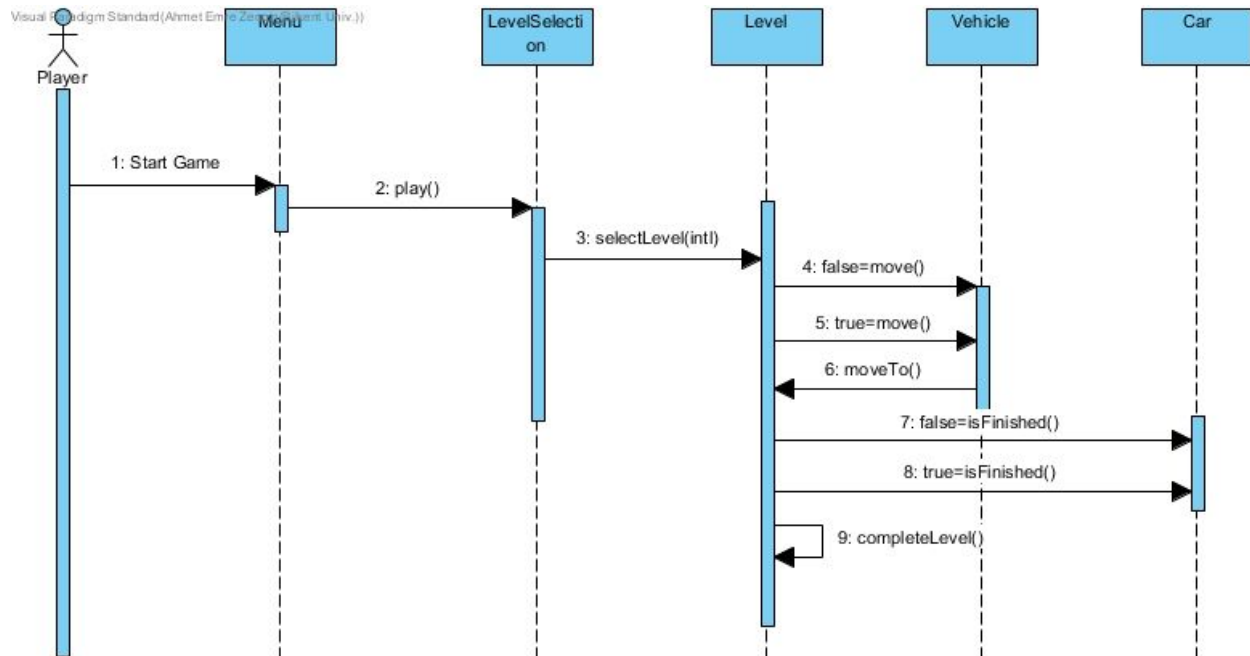


Figure 3: Level Sequence Diagram

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

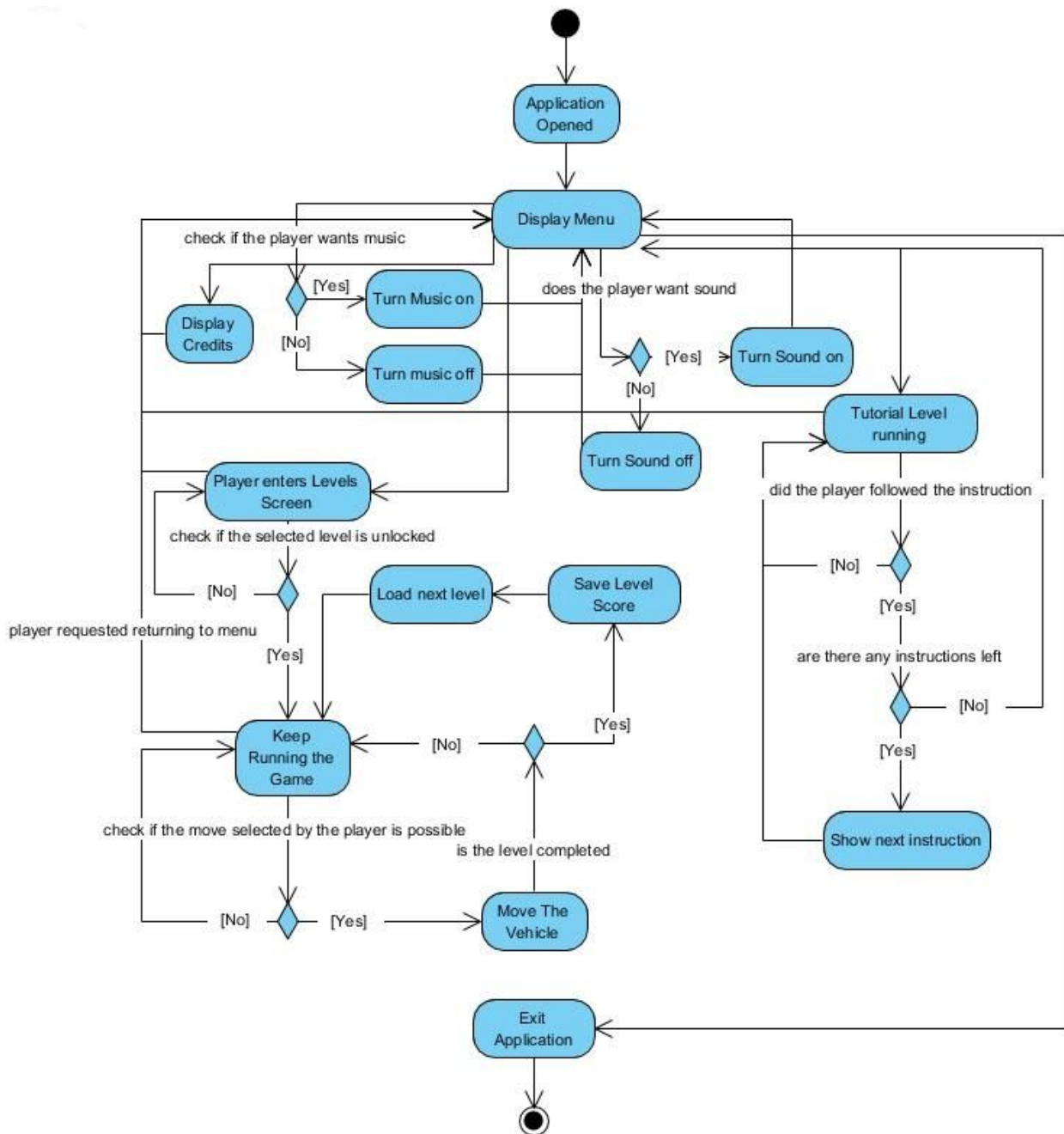


Figure 4: Activity Diagram

This activity diagram sums up the working and flow of the system. How the activity diagram works is explained 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

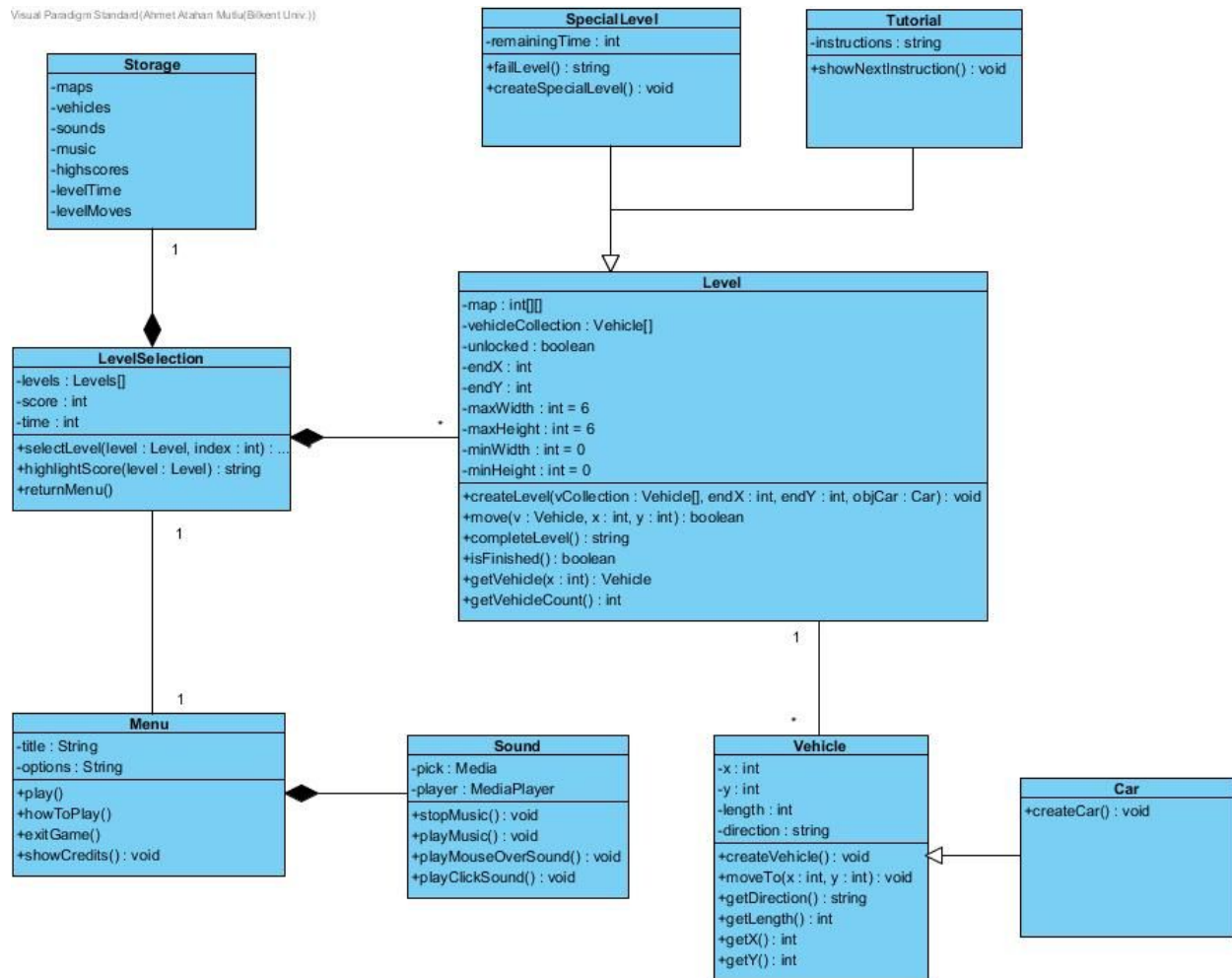


Figure 5: Object and Class Model Diagram

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 “LevelSelection” class as there can be only one instance of game running at the same time. The “LevelSelection” class is aggregately and 1-to-many associated with the “Level” class as it contains a collection of levels. It is also aggregately associated with the “Storage” class where

the data of each level is kept. 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

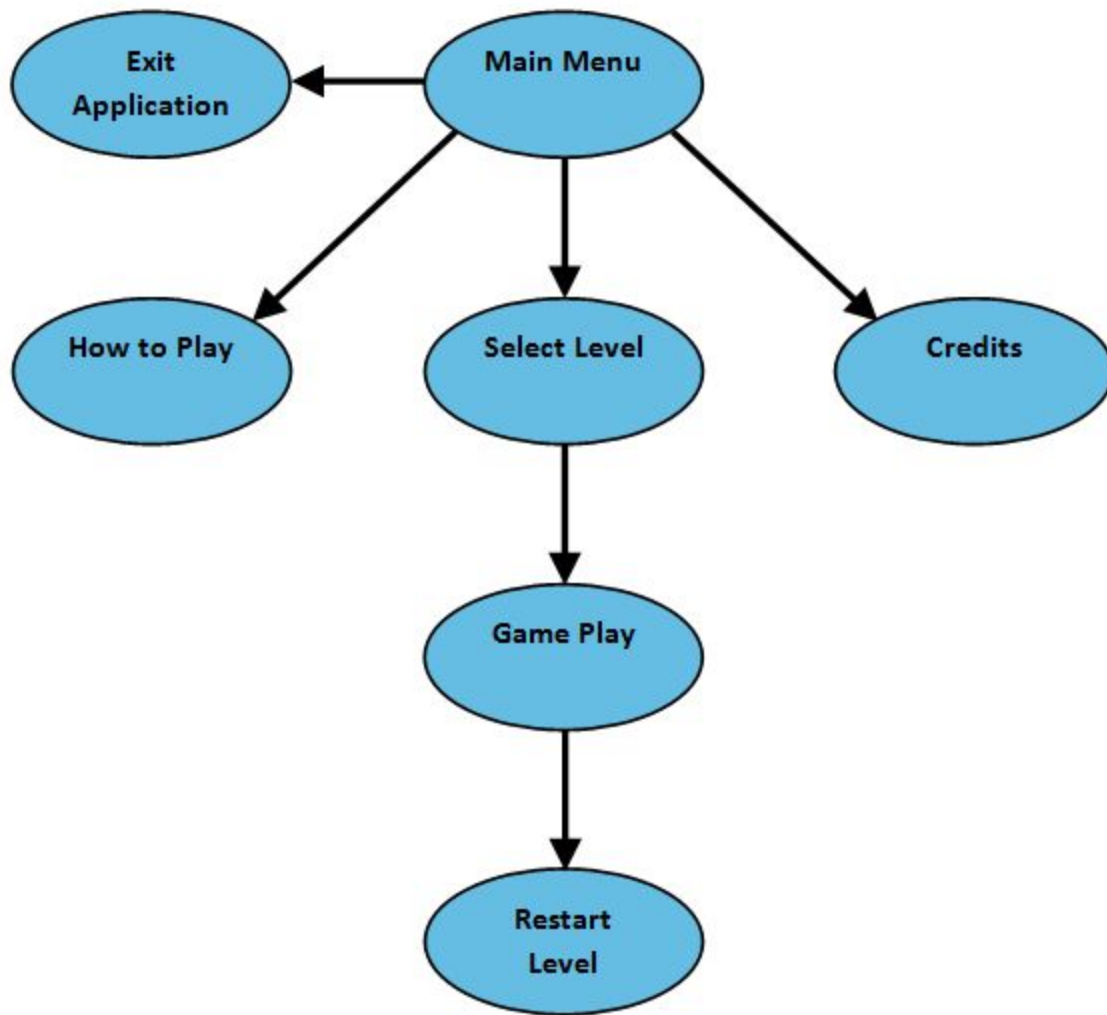


Figure 6: Navigational Path

5.4.2 Screen Mockups

The following screen mockups are just illustrations on how our game will be seen by the players. The final version of the game's user interface won't be like those mockups but similar.

5.4.2.1 Opening Screen



Figure 7: Mockup of Main Menu

When the 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

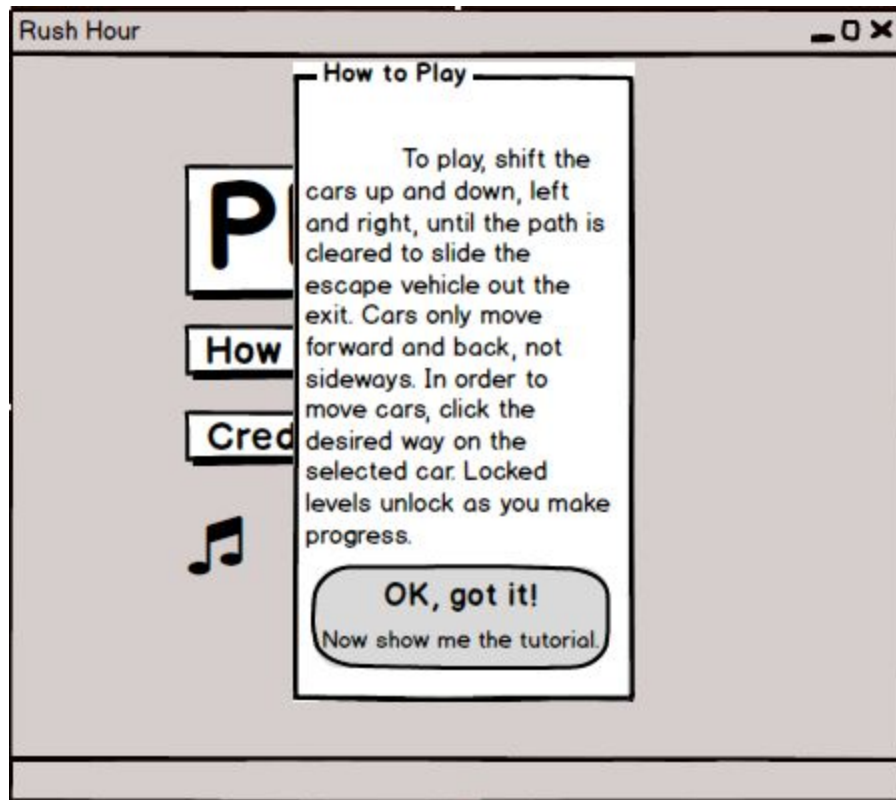


Figure 8: Mockup of How to Play Screen

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

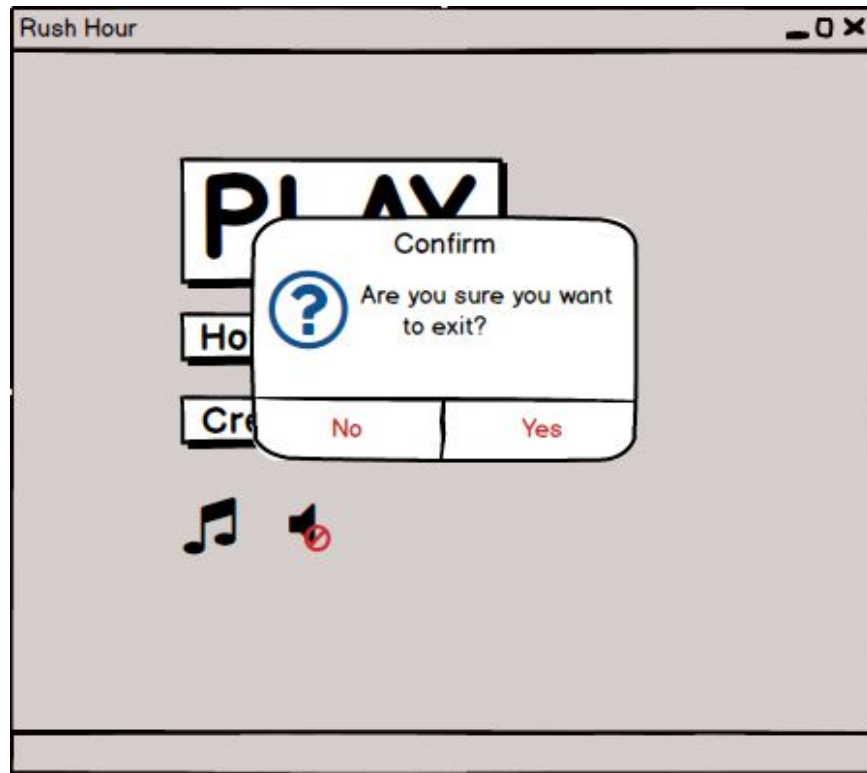


Figure 9: Mockup of Exit Window

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

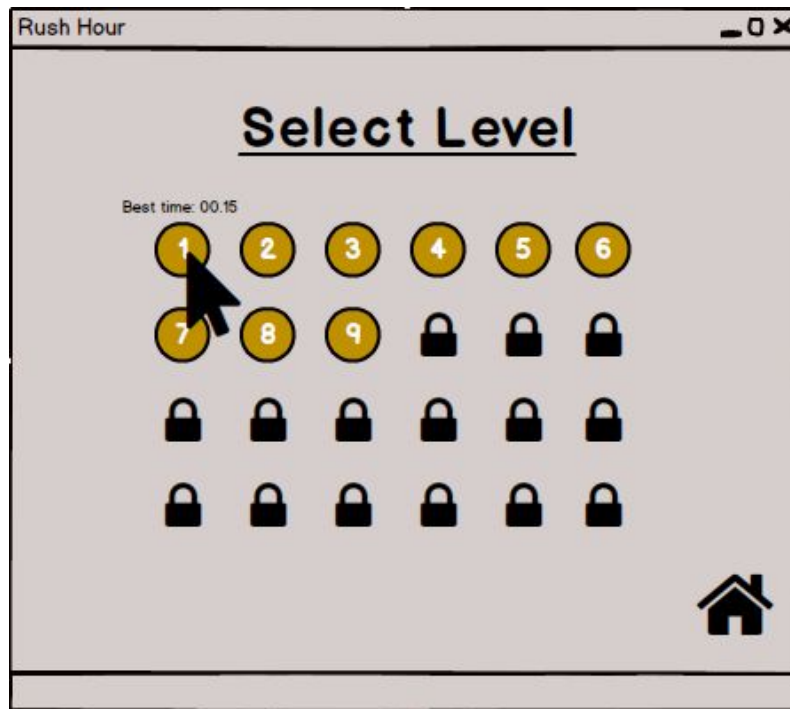


Figure 10: Mockup of Level Selection Screen

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

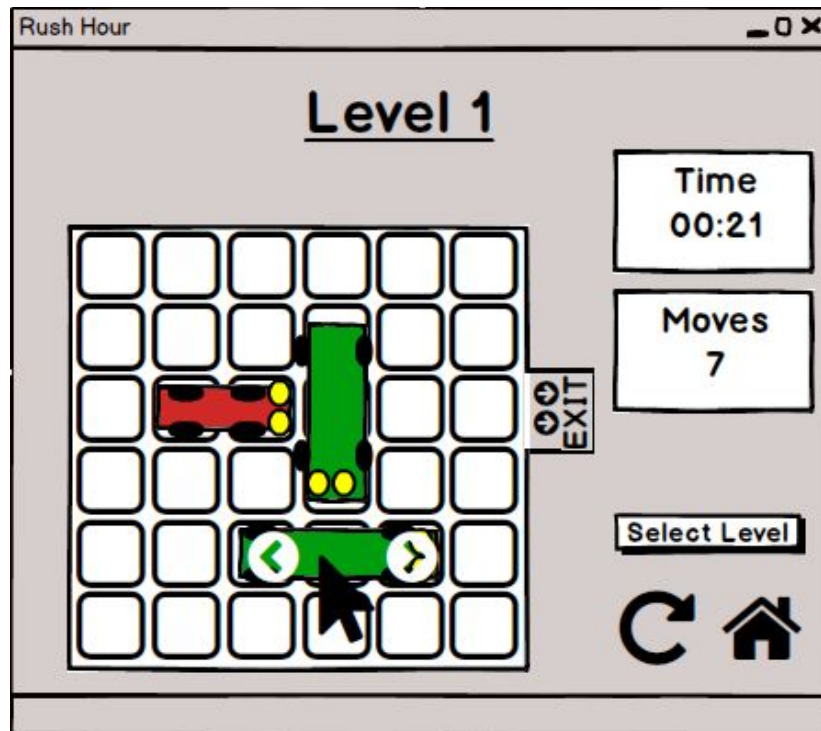


Figure 11: Mockup of Gameplay Screen

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

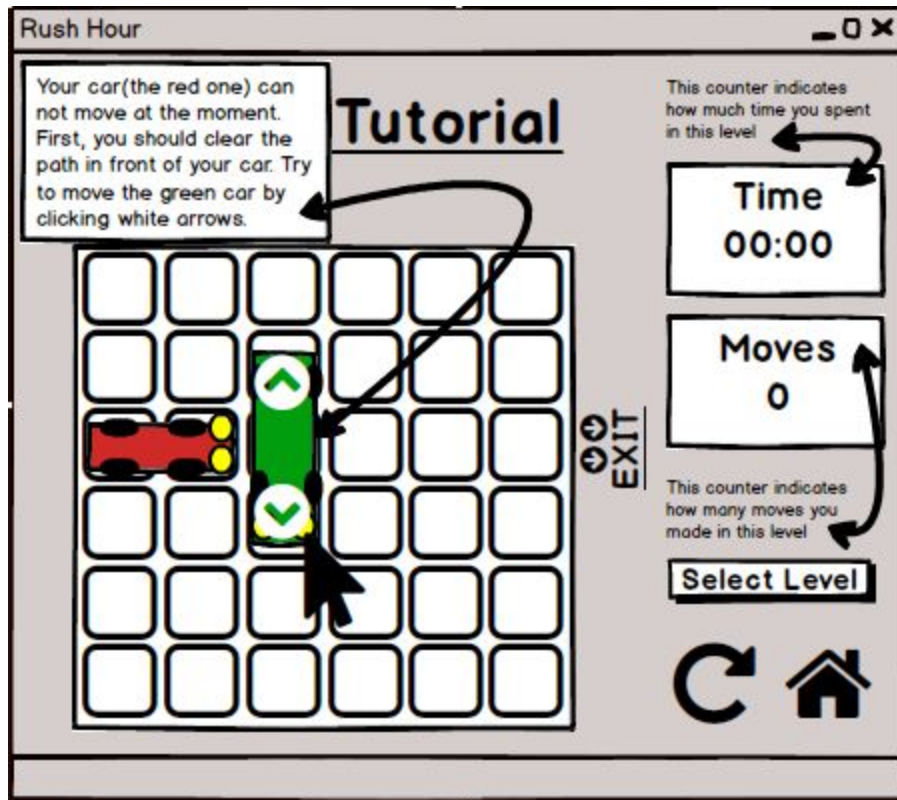


Figure 12: Mockup of Tutorial Level

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

- [1] “Loudness and annoyance of disturbing sounds – perception by normal hearing subjects”, Taylor & Francis Online. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/14992027.2017.1321790>. [Accessed 25 November 2018]
- [2] “Human Processor Model”, Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Human_processor_model. [Accessed 20 November 2018]
- [3] “Micro Java Game Development”, Google Books. [Online]. Available: https://books.google.com.tr/books?id=tCvX60J8OAC&pg=PA202&lpg=PA202&dq=maximum+execution+time+for+java+game&source=bl&ots=ixQMIGXoWK&sig=PCaAuCgE_UZs9f1yHhv3A44m2Lg&hl=en&sa=X&ved=2ahUKEwj6J7cgfLeAhULqIsKHV7SAkIQ6AEwAXoECAMQAQ#v=onepage&q=maximum%20execution%20time%20for%20java%20game&f=false. [Accessed 18 November 2018]
- [4] “Rush Hour (Puzzle)”, Wikipedia. [Online]. Available: [https://en.wikipedia.org/wiki/Rush_Hour_\(puzzle\)](https://en.wikipedia.org/wiki/Rush_Hour_(puzzle)). [Accessed 27 November 2018]