CS319 Object-Oriented Software Engineering AnalysisReport Iteration-1

RUSH HOUR

GROUP 3.B

Musab Gelişgen
Kuluhan Binici
Yusuf Dalva
Ahmet Avcı
Melih Ünsal

1. Introduction

We, as team 3.B, preferred to implement the game “Rush Hour” as our CS319 term project. Rush hour is an old school block sliding puzzle game invented in 70’s. Normally it is a single player cardboard game in which the purpose of the player is to elude his car from the jammed traffic. However, since we are going to implement it in a digital environment, we decided to enhance the regular design by adding extra features that we believe would make the game more alluring.

Here are some of our ideas of features to the game:

- Single Player Mode
    - 3 difficulty levels for every Single Player map (easy - medium - hard)
    - Playing against time in Single Player Mode for medium & hard difficulty levels
    - Playing against a restricted number of movements in Single Player Mode for medium & hard difficulty levels
    - Predefined 10 maps for Single Player mode
    - High score tracking for every level in Single Player mode
- Multiplayer Mode (PvP)
    - Power-ups (obtained from wildcards) for multiplayer mode
- Sound options
- Different theme options for maps
- Different skin & color options for cars
- Smooth Graphics

To implement the game we will use Java and the game will be a Desktop Application. Moreover, we are planning to use LibGDX for the design/graphics part. As we implement the game, concepts we learned from CS courses, especially the ones in CS 319 are going to be used throughout the project.

2. Overview

2.1 Gameplay

Rush Hour is a 2D game. The game has a single-player mode and a multiplayer mode. Tools included in the game are blocking cars, an escape car which the player controls, a map which all the cars are placed on (in different directions) and play cards which grant the players some special movements (cards are for multiplayer only).

In single-player mode, given a predefined map of cars, the player will try to escape his car from a maze-like structure by moving the blocking cars around. For our version of the game, we plan on limiting the time and the movement that the player has for medium and hard difficulty levels. If the player succeeds in getting his car out of the maze within the restriction limits (if any), that is a victory.

In multiplayer mode, two people race their cars to the end of the map by blocking the opponent's way and opening theirs. This mode is played in turns. The first one to get his car out of the platform wins the game. In this game mode, players also draw wildcards which grant them special abilities like moving the platform for their own good or having the ability to move their car by multiple times in a round.

2.2 Modes

2.2.1 Player Modes

The game will provide both singleplayer and multiplayer options. In the single-player mode, the player will try to escape his car from the map to pass the current level or to exceed the previous high score. The high score is determined according to the time of completion.

In the multiplayer mode, 2 players will compete against each other to rescue their cars before the other player does.

2.2.2 Difficulty Level modes

There will be difficulty options in the singleplayer mode. Right now, we are planning to have 3 levels of difficulty which are: easy, medium and hard. In the easy mode, the only objective of the player is to solve the puzzle, he/she won’t face any time or movement limitations. The medium mode will have time and movement constraints. Lastly, the hard mode will have even fewer movements and less time to complete the puzzle.

2.3 Cars

There are 2 types of cars in the game: Escape cars - which the player tries to rescue from the map - and blocking cars. Blocking cars come in different sizes such

as 2x1 and 3x1 units. Escape car has a 2x1 unit size. All types of cars have different colors for a visual variance.

All the cars on the field can be moved for their direction only, meaning they cannot be moved sideways. These obstacles can be moved by any number of unit blocks as long as their path is not blocked by other cars or the borders of the map.

2.4 Maps

For single-player mode, we plan on implementing 10 predesignated square maps which have different levels of difficulties. In each of these maps, players will try to get their cars out of the maze by moving blocking cars around.

For multiplayer mode, our plan is to have 3 predesignated maps. These will be rectangular maps which are larger than the ones in the single-player mode. Multiplayer maps will consist of 3 pieces, independent of each other. These 3 separate pieces can be moved sideways (with special wildcard abilities explained in section 2.5). These map movements will make the map extend on one side and retract on the other side (i.e. change the rectangular shape of the map and make it a polygon) Player's cars will start from opposite edges of the game platform. There will be blocking cars on this map as well.

2.5 Wild Cards

There will be no cards in the single-player mode. In the multiplayer mode, there will be a pile of wild cards from which each player will draw a single card at the beginning of each turn. The players will not know which card they will receive until they draw the card as the card drawing process is completely at random.

According to the card's special effect, players will be granted a special movement. These special movements include but are not limited to sliding the platform, sliding the opponent's car by one layer backward or multiple layers sideways, moving their own car by multiple grids (by multiple layers forward or sideways).

2.6 Settings

The player is able to choose the game mode as well as its difficulty level (if the single-player mode is chosen). In the sound settings menu, the player can turn on/off the music, change the music and adjust its volume. The player will be able to choose a theme for the map and skin for his escape car.

3. Functional requirements

3.1 Main Menu

This is the interface that the user is going to experience when the application is opened. By using this screen, player may initialize the game and start playing. This screen also gives access to “Main Menu Preferences” and “How to Play” sections of the application (explained in 3.2 and 3.3). The player is going to be represented as a car (escape car in the game terminology), he will be able to move with respect to the boundaries set by the blocking cars.

In the game, the game atmosphere is adjustable and the sound changes in every level. Player can move the escape car by using the mouse, where hovering on the escape car will show the possible moves. The user can change the controls depending on his/her preferences. There are two main objectives of the game, dependent on the game modes that the user plays in. The two game modes included in the game are:

3.1.1 Single Player Mode

The user can access this interface from the Main Menu which is the first screen that the player experiences after launching the game. With entering this screen, the user can initiate the game by selecting one of the maps and difficulty levels.. The map is going to be generated from the 10 predefined maps present in the game. Difficulty levels are Easy, Medium and Hard. The default map background can be changed from the “Main Menu Preferences” (explained in 3.2). In the single player mode of the game the main objective is to successfully get the escape car out of the puzzle block where blocking cars are also located in. The level ends at the instance that the escape car gets out of the puzzle block.

In the Easy mode of the single player mode, the only aim is to get the escape car out of the puzzle block, without any constraints. The score in the game is determined only according to the time that player spends on the level. In the Medium and Hard game modes, the player must deal with the limited number of moves and the limited time. The score system keeps the completion time of the level and stores the high score to illustrate it in the single player screen.

3.1.2 Multiplayer Mode

The user can access this instance from the Main Menu. By entering this screen, the players choose one of the 3 predefined maps and directly get into the game. There is no difficulty level in this game mode which differentiates from the single player mode. As another difference to the single player mode, unlike the single piece board of single player mode; multiplayer mode consists a board composed of 3 pieces which can be shifted sideways.

In this game mode, there are two players present in the game. The game proceeds with turns. In the multiplayer mode, the moves that the players can make

are determined by the wildcards. The wildcards describe the type of move that can be done and the player decides to make a strategic move according to the wildcard. The wildcards consists of moves like shifting the board pieces, moving the escape cars (his or opponent's) and moving a blocking car. These wildcards are drawn by players one by one and in a completely random way. According to the moves done compatible to the wildcards the first player to escape the map wins the game.

3.2 Main Menu Preferences

The players can access these screens from the Main Menu. These screens consists of the features that are adjustable by the player. In these screens the player can modify some settings of the game. The settings that are adjustable by the user are:

- Map theme
- The skin of the escape car (For player 1 and player 2)
- Sound Effects (can be muted, adjusted or changed)

3.3 How to Play

Screen, consisting this section can be accessed from the main menu. The How to Play section consists of information about the game. The user accesses to a document where the gameplay details and rules are explained in detail. The preferences that player can make in settings and the different options are also introduced in this section. User will be also informed about the shortcuts present in the game. By using this page user can access information about:

- Game Rules (Both single player and multiplayer)
- Controls (dragging cars and sliding the map)
- Shortcuts present
- List of Wildcards
- List of selectable cars, information about them and available skins

With the presence of this screen, users can get into the game much more easily.

3.4 Credits

The user can access this page from the Main Menu. This page contains the license information of the game, information about the creators of the original game, brief information about the contributors and the development team itself. This page also contains the contact information of the developers engaged in the development process for users to contact them via e-mail for any feedback.

4. Nonfunctional requirements

4.1 Game Performance

In this game, we are going to not limit us to overuse the functionalities of the game engine that we are going to use for graphical usage because in this game our main concern is neither the quality nor the performance of the game but the functionalities of the game. On the other hand, it doesn't mean the performance of the game will be low because the effect of the sounds and animations will not decrease the performance. The reason is that the game we are implementing is a classical game and doesn't include so many actions so that we should be concerned with its performance.

4.2 User-Friendly Interface

The quality of user interface is one of the most important factors determining how desirable and enjoyable the game is. Therefore, we are going to use many high-resolution assets, image-based animations consisting of so many images to make the animations as smooth as possible. We are going to use customized bitmap fonts to attract the users by our own fonts and we are going to use our own style objects to create unique and enjoyable user interface elements.

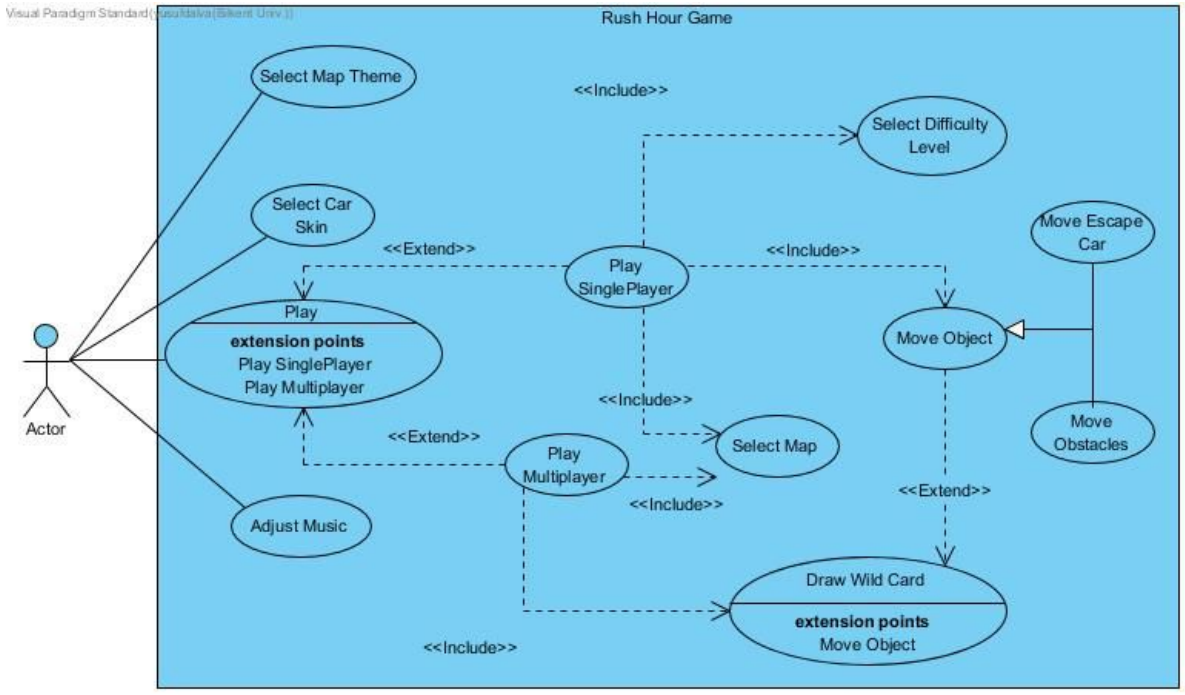
There will be also an overlay menu in the game screen to give the ability to change the game properties in the game screen such as the sound level and there will be also options to return to the main menu in the overlay menu. We will also design the high scores menu to hold the best scores of the players. There will be options menu to change the game properties such as the generic music of the game and the map theme which the user will play the game on. With all these properties, we are going to make the player enjoy the game as much as possible.

4.3 Compatible User Interface with Real World

We are going to enrich the user interface by adding different features which are taken from real life to make the game much more realistic and attract the user. The cars in the game will be taken from real life. For instance, when a car is moved by the user, it will sound like in the real life. The cars in the rush hour, horn at irregular intervals but the combination of these noises will be compatible with the real life and the horning property can be adjusted by the user in case he is disturbed.

There will be several maps not to bore the player which are familiar to us in real life. Weather condition will also be changeable for every single map to make the user understand the time passes. For some maps, the user tries to rescue a regular car but sometimes, he is going to rescue an ambulance from traffic which will increase the user's desire to rescue the car from traffic.

5. System models
5.1 Use case model



Use Case #1:

Use Case Name	Select Map Theme
Participating Actors	Initiated by Actor
Flow Of Events	1. The Actor activates the Select Map Theme option from the Main Menu 2. Rush Hour Game responds by showing the available themes to the actor 3. The Actor makes a selection among all the themes shown by the Rush Hour Game. 4. Rush Hour Game receives the selection that the Actor made. The map selection is saved in the preferences of the game
Entry Condition	The Actor is in the Selecting Map Theme Panel in Main Menu
Exit Condition	- The selection made by the Actor is applied - Rush Hour Game receives the selection made by the Actor
Quality Requirements	None

Use Case #2:

Use Case Name	Select Car Skin
Participating Actors	Initiated by Actor
Flow Of Events	1. The Actor activates the Select Car Skin option from the Main Menu 2. Rush Hour Game responds by showing the available car skins to the Actor 3. The Actor makes a selection among all the car skins supplied by the Rush Hour Game 4. Rush Hour Game receives the selection made by the Actor. The car skin selections (for player one and player two) are saved in the preferences of the game
Entry Condition	The Actor is in the Selecting Car Skin Panel in Main Menu
Exit Condition	- The selections made by the Actor are applied - Rush Hour Game receives the selections made by the Actor
Quality Requirements	None

Use Case #3:

Use Case Name	Play
Participating Actors	Initiated by Actor
Flow of Events	1. The Actor is willing to Play the game 2. Rush Hour Game is waiting for the response of the user which is either selecting Single Player Mode or Multiplayer Mode 3. The Actor selects a game mode
Entry Condition	The Actor is in the Main Menu
Exit Condition	The Actor makes a selection implying the preference of game mode
Quality Requirements	None

Use Case #4:

Use Case Name	Adjust Music
Participating Actors	Initiated by Actor
Flow Of Events	1. The Actor activates the Adjust Music option from the Main Menu 2. Rush Hour Game responds by showing the sound and music options to the Actor 3. The Actor adjusts the sound and music according to its desire 4. Rush Hour Game receives the selections that the Actor made. These sound and music selections are kept in the preferences of Rush Hour Game
Entry Condition	The Actor is in the Adjust Sound Panel in the Main Menu
Exit Condition	- The selections that the Actor made are applied from the Panel - Rush Hour Game receives the selection made by the Actor
Quality Requirements	None

Use Case #5:

Use Case Name	Play SinglePlayer
Participating Actor	Initiated by Actor
Flow Of Events	1. The Actor shows interest in playing the game 2. Rush Hour Game waits from the Actor to make a preference for the game mode 3. The Actor selects the Single Player option from the Main Menu 4. Rush Hour Game initializes the game according to the game mode preference and other preferences specified by the user
Entry Condition	This use case extends the Play use case. This use case includes “Select Map”, “Select Difficulty Level” and “Move Object”

	use cases. This use case is initiated just after the Actor makes the preference of game mode as Single Player Mode
Exit Condition	- The Actor specifies the preference of exiting the game - The level constraints make the Actor to leave the game
Quality Requirements	None

Use Case #6:

Use Case Name	Play Multiplayer
Participating Actor	Initiated by Actor
Flow Of Events	1. The Actor shows interest in playing the game 2. Rush Hour Game waits from the Actor to make a preference for the desired game mode 3. The Actor selects Multiplayer option from the Main Menu 4. Rush Hour Game initializes the game according to the game mode preference and other preferences specified by the user
Entry Condition	This use case extends the Play use case. This use case includes “Select Map” and “Draw Wild Card” use cases. This use case is initiated just after the Actor makes the preference of game mode as Multiplayer mode
Exit Condition	- At the moment that one of the users successfully completes the challenge, game ends - Actor specifically shows the interest on exiting by the exit button
Quality Requirements	None

Use Case #7:

Use Case Name	Select Difficulty Level
Participating Actor	Initiated by Actor
Flow Of Events	1. Rush Hour Game looks for the information of difficulty level in order to initialize the game 2. Rush Hour Game directs the user to difficulty level selection panel 3. Actor selects a difficulty level and apply it 4. The difficulty preference of the Actor is saved to the preferences of the game
Entry Condition	The Actor selected Single Player Mode as the play mode
Exit Condition	The difficulty level selected by the Actor is saved to the preferences of the game
Quality Requirements	None

Use Case #8:

Use Case Name	Move Object
Participating Actor	Initiated by Actor
Flow Of Events	1. Rush Hour Game indicates that the user will make a move 2. After that the user will make a specific move by moving the cars to the available spots 3. Rush Hour Game recognizes the move and processes it, by taking the move count into consideration. Each Move Object case indicates 1 move in the game 4. The turn of the player ends
Entry Condition	The playing turn is at the player, the game is still not ended
Exit Condition	The available moving object operation is recognized and processed by the Rush Hour Game. The turn ends.

Quality Requirements	The move that the user makes is valid according to the game rules.

Use Case #9:

Use Case Name	Select Map
Participating Actor	Initiated by Actor
Flow Of Events	1. After selecting the game mode (either single player or multiplayer), Rush Hour Game settles up a frame that makes the user to select a puzzle to play 2. Among all the possible options, the Actor selects a map to play. 3. If the map selection is valid, Rush Hour Game takes the selection into consideration. Else, the Actor will get a warning sound and will have to pick another map 4. Rush Hour Game gets the map selection and initiates the level according to the selection
Entry Conditions	- A game mode is selected by the Actor - If in Single Player game a difficulty level is chosen by the Actor
Exit Conditions	The Actor made a valid selection as map (among the selectable ones) and the Rush Hour Game initiated the level according to the preference of the Actor.
Quality Requirements	The map preference made by the user must be one of the selectable maps (not restricted with the lock icon)

Use Case #10:

Use Case Name	Draw Wild Card
Participating Actor	Initiated by Actor

Flow Of Events	1. Actor has the playing turn in the game 2. Rush Hour Game gives a random wild card to the Actor which contains the move that the Actor can make in any car. 3. According to the wild card, the restrictions of the player is settled again for the user to make a move
Entry Conditions	- The selected game mode is multiplayer mode - Map selection is made by the Actor - The Actor has the turn to play
Exit Conditions	The Actor receives the wild card and Rush Hour Game restricts the possible moves that can be done accordingly
Quality Requirements	None

Use Case #11:

Use Case Name	Move Escape Car
Participating Actors	Inherited from Move Object use case
Flow Of Events	1. The turn to play is at the Actor 2. The player selects to move the Escape Car 3. If the place that the Actor decides to play is compatible with the restrictions, the move is reported to the Rush Hour Game 4. If the move is valid Rush Hour Game processes the move
Entry Conditions	- The playing turn is at the Actor. - The object selected to move is the Escape Car - A game mode and map is selected
Exit Conditions	- The Actor makes a move with the escape car that is compatible with the restrictions - The move done is processed by the Rush Hour Game

Quality Requirements	The move that the Actor does is not against the restrictions defined by the game

Use Case #12:

Use Case Name	Move Obstacles
Participating Actors	Inherited from Move Object use case
Flow Of Events	1. The turn to play is at the Actor 2. The Actor selects to move one of the Obstacles 3. If the place that Actor decides to play is compatible with the restrictions, the move is reported to the Rush Hour Game 4. If the move is valid, Rush Hour Move processes the move
Entry Conditions	- The playing turn is at the Actor - The object selected to move is the Escape Car - A game mode and map is selected
Exit Conditions	- The Actor makes a move with an obstacle that is compatible with the restrictions - The move done is processed by the Rush Hour Game
Quality Requirements	- The move that the Actor does is not against the restrictions defined by the game

5.2 Dynamic models

In this section, some possible scenarios will be explained.

- Play Single Player Mode, Move Cars: Choose the single-player mode and play the game by moving the cars.
- Play Multi-Player Mode, Move Cars: Choose the multiplayer mode and play the game against each other by drawing cards and moving cars.

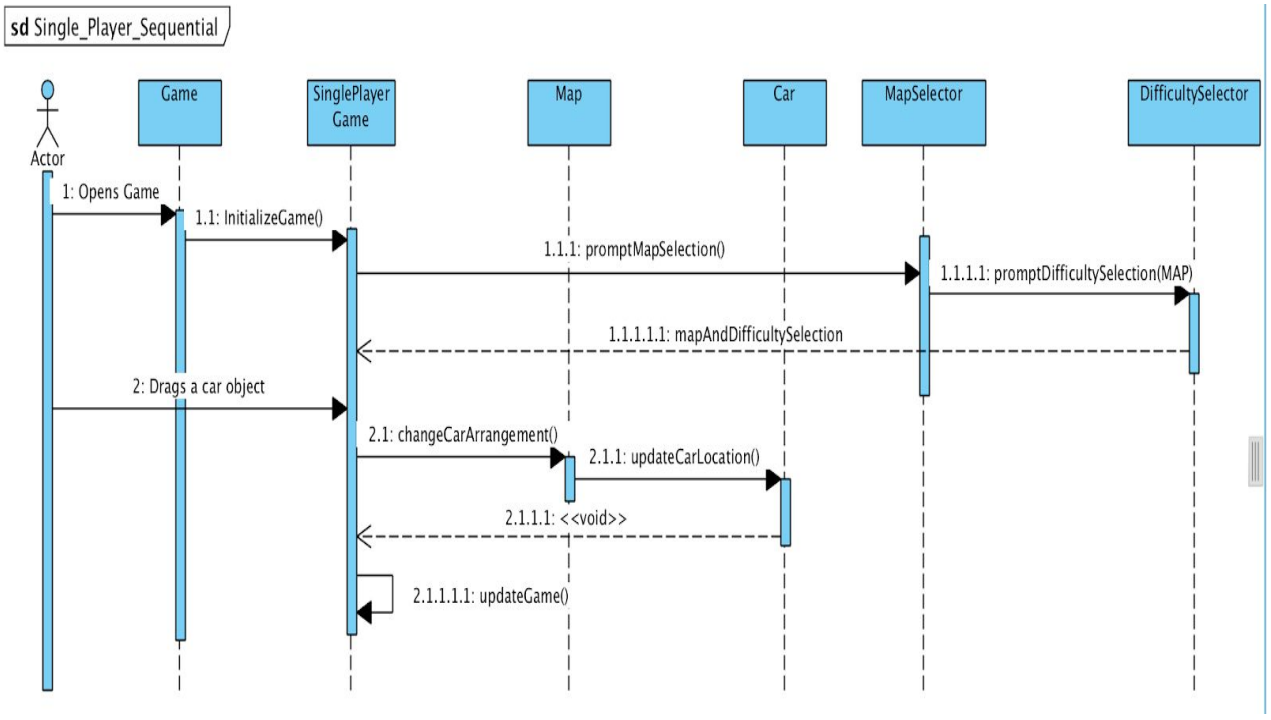
- Play Multi-Player Mode, Slide Map: Choose the multiplayer mode and play the game against each other by drawing cards and moving map.
- Adjust Settings: Open the game and adjust the game.

5.2.1 Sequence Diagram

Scenario #1:Play Single Player Mode, Move Cars

Actor: Actor

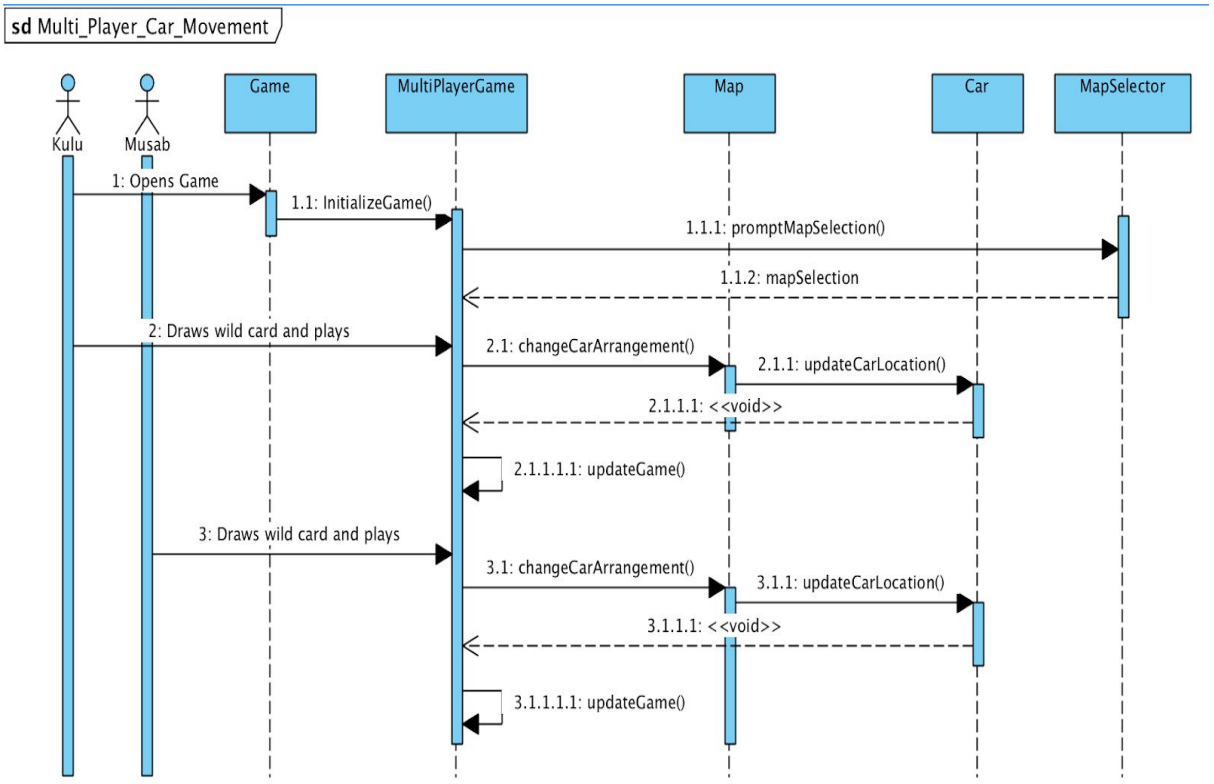
The actor opens the game and chooses the single player mode. Then, the map selection menu comes up and the actor chooses the game map. After selecting a map, the actor selects the difficulty level of the game. After making arrangements, the game starts and the actor plays the game by moving the cars.



Scenario #2:Play Multiplayer Mode, Move Cars

Actors: Kulu and Musab

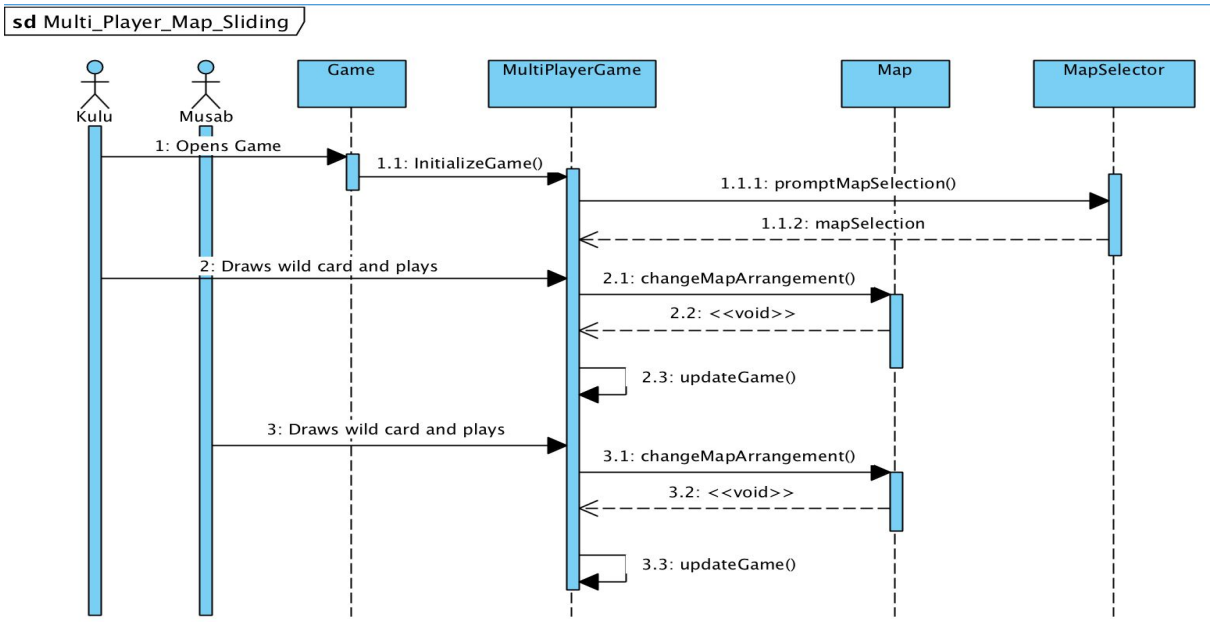
Kulu opens the game and chooses the multiplayer mode. Then, the map selection menu comes up and Kulu chooses the game map. After selecting a map, the game starts and Kulu draws a movement card and move a car according to this card. After Kulu, Musab draws a card and move a car according to the card.



Scenario #3:Play Multiplayer Mode, Slide the Map

Actors: Kulu and Musab

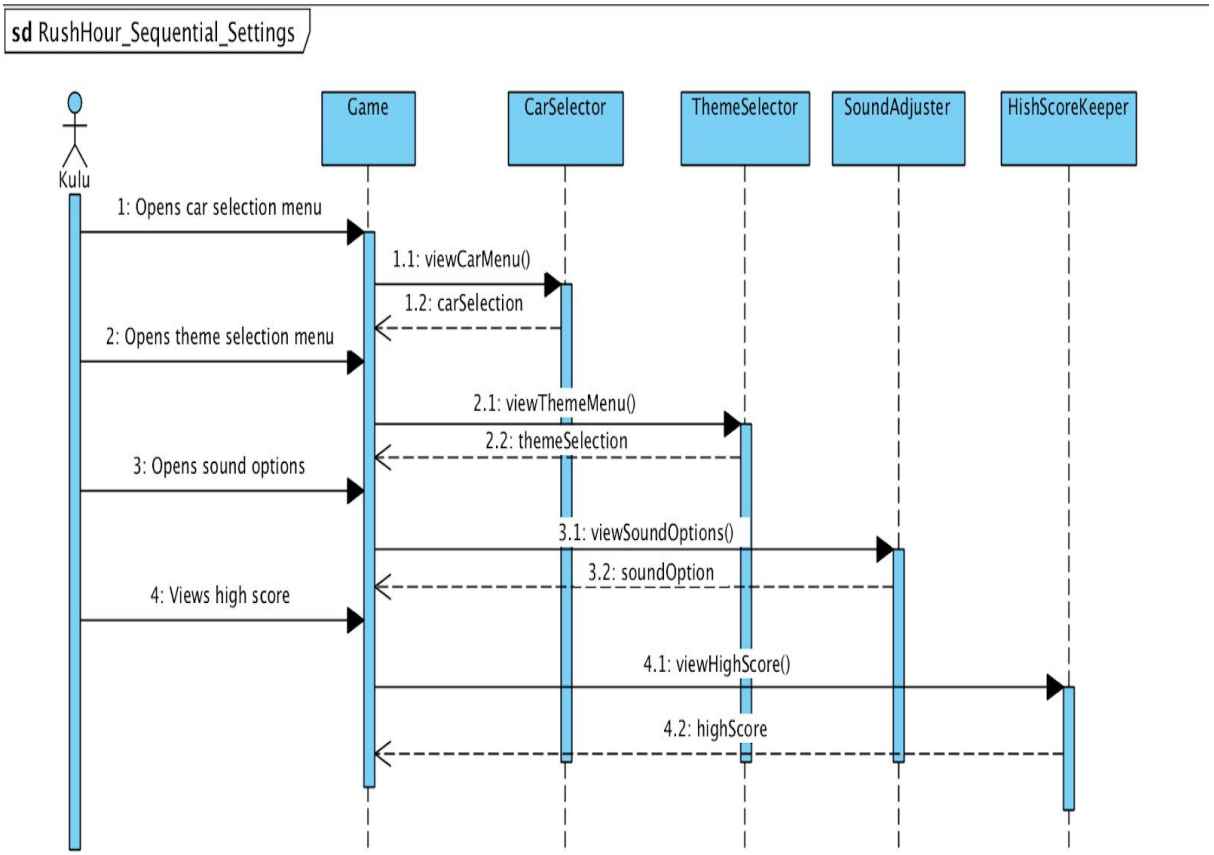
Kulu opens the game and chooses the multiplayer mode. Then, the map selection menu comes up and Kulu chooses the game map. After selecting a map, the game starts and Kulu draws a movement card and slides the one piece of the map. After Kulu, Musab draws a card and slides the one piece of the map.



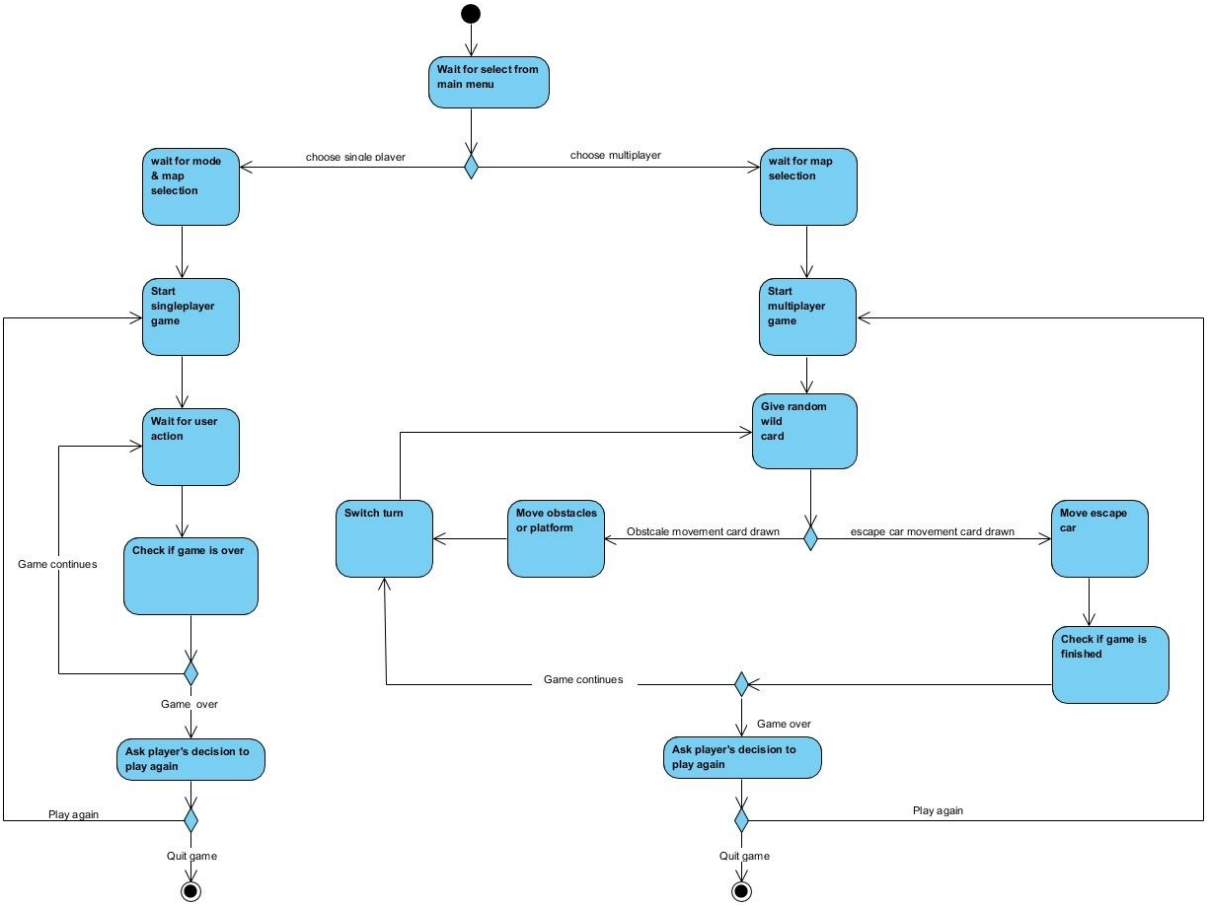
Scenario #4:Adjust Settings

Actors: Kulu

Kulu opens the game and selects the car selection menu. Then, he chooses his car and turns back to the main menu. After this, Kulu selects the theme selection menu and chooses his theme, and turns back to the main menu. Then he selects sound menu and adjusts the sounds. After that, he returns the main menu, gets into high score menu and he sees total high score records.



5.2.2 Activity Diagram



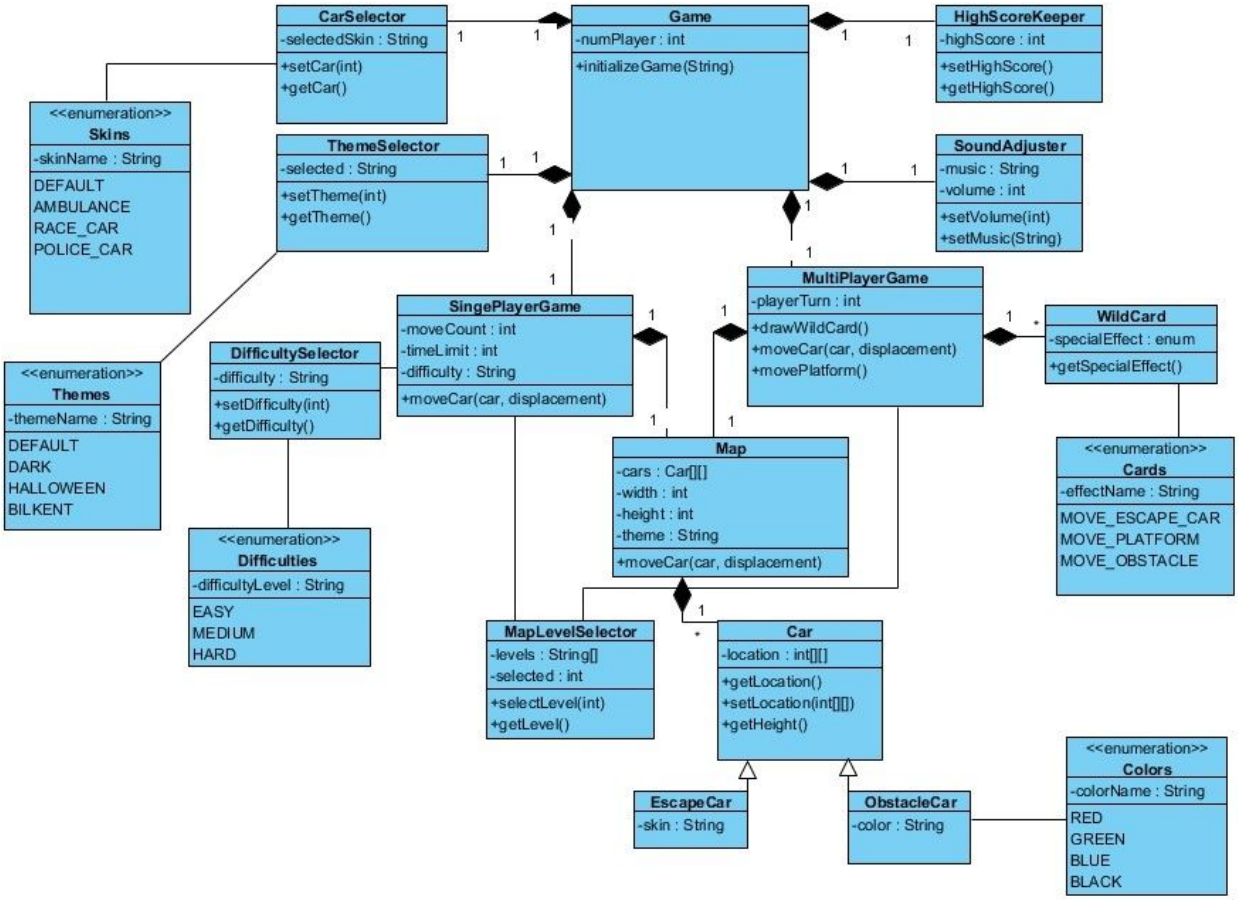
Description

After the game is started, main menu comes up and waits user to select game mode as single or multiplayer. If the users selects single player mode, the game waits for the map and difficulty level selection. Then the game starts and waits for user to move a car. At the end of each turn, the game checks whether the game is over or not. If the game is not over, again it waits for user to move a car and it goes like that until the game is over. When the game is over, the game asks for the user's decision to continue the game or quit.

If the user selects multiplayer mode, the game waits for the map selection. Then the game starts and gives a random wild card to the users in each turn starting from the first user. If the card is a escape car movement card, the users need to move the escape car. After moving the escape car, the game checks whether the game is over or not. If the game is not over, the game continues by switching the

player turn. However, if the game is over, the game asks for user to start a new game or quit. On the other hand, if the given card is a obstacle movement car, the users need to move one of the obstacle cars or the map and the game continues by switching the player turns. The game does not checks whether the game is over or not after obstacle movement car. This is because the game can be over only by reaching the finish by moving the escape car.

5.3 Object and class model

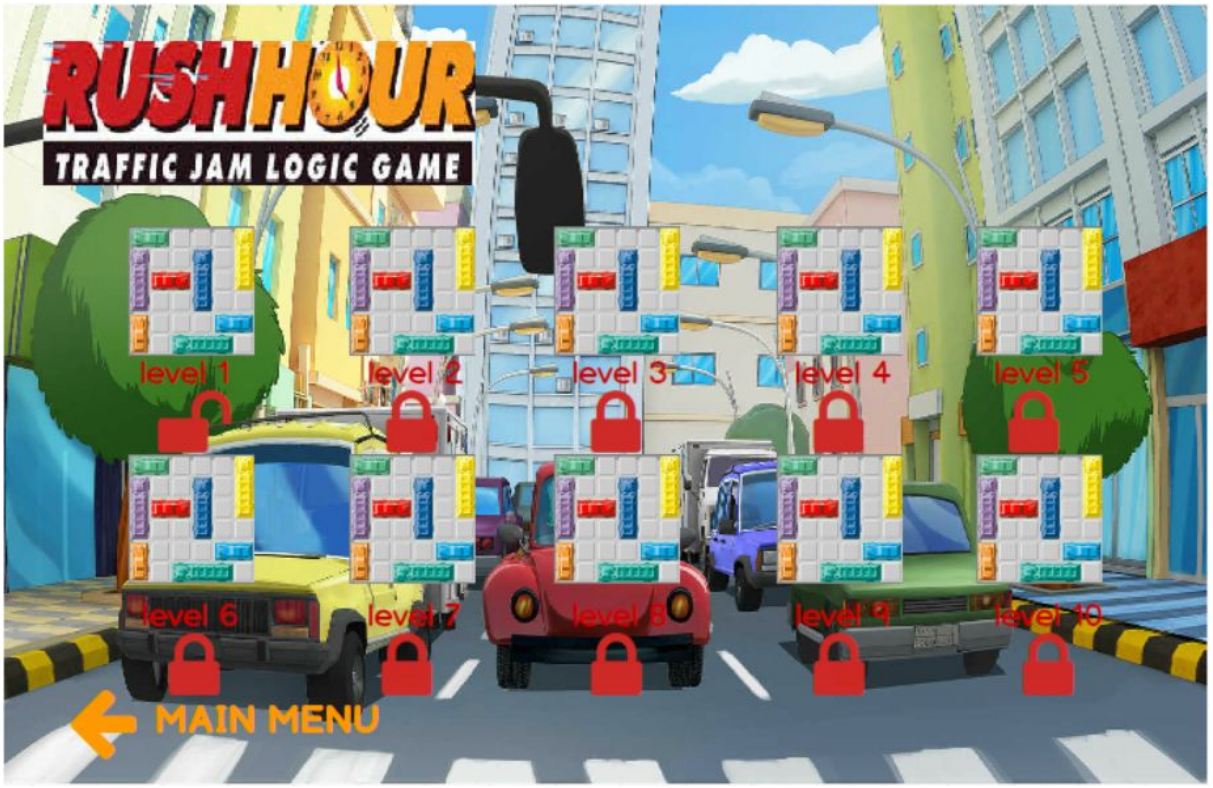


5.4 User interface - navigational paths and screen mock-ups

5.4.1 Main Menu Mockup

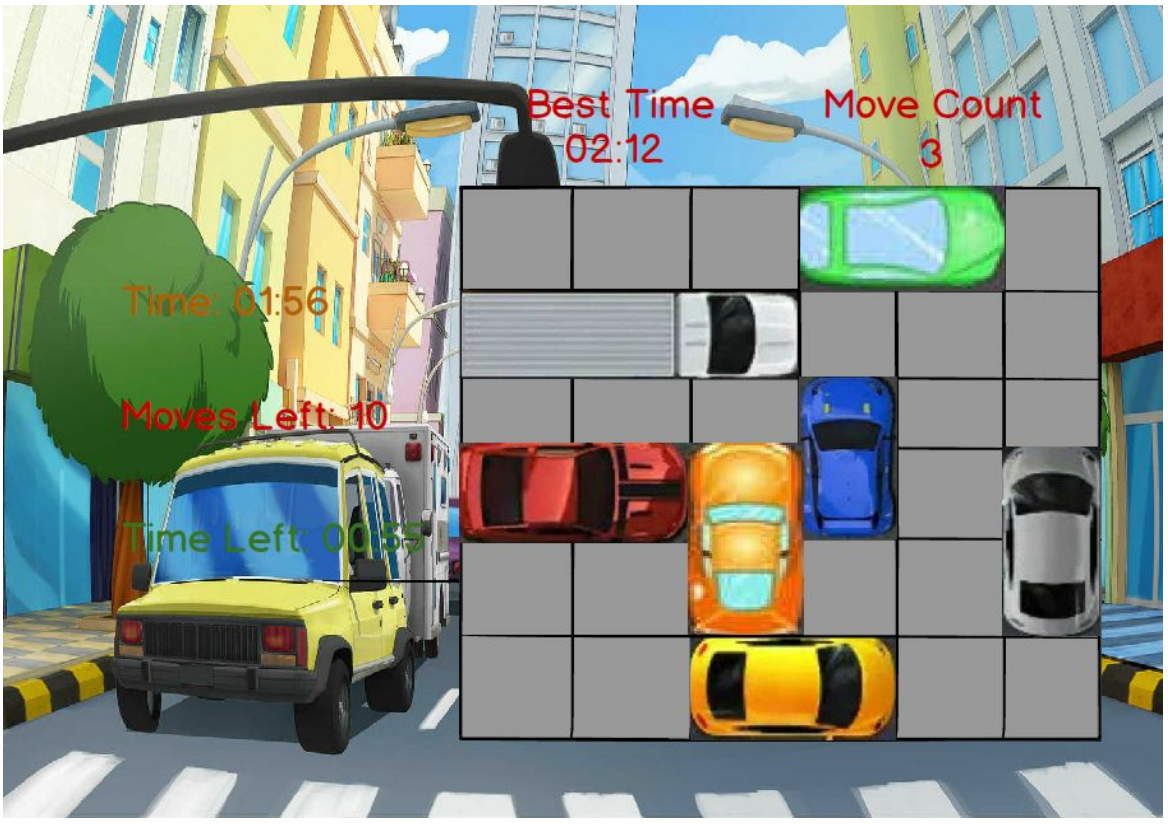
This is the main menu page that users are going to see immediately after opening the game. From this page users are able to access other parts of the game

5.4.2 Single-Player Mode Map Selection Mockup



This is the map selection menu that users are going to see after selecting the single-player mode. The difficulty level of the maps get increase according to the number of level. Also levels are going to be opened when the previous level is passed.

5.4.3 Single-Player Mode Mockup



This is an example gameplay from the Single Player Mode. Player is trying to make the car colored in red escape the map. Other than the map where the cars are located, the player can also see the best time for that level, number of moves made, time spent on the level, moves left to finish the level and time left to finish the level.

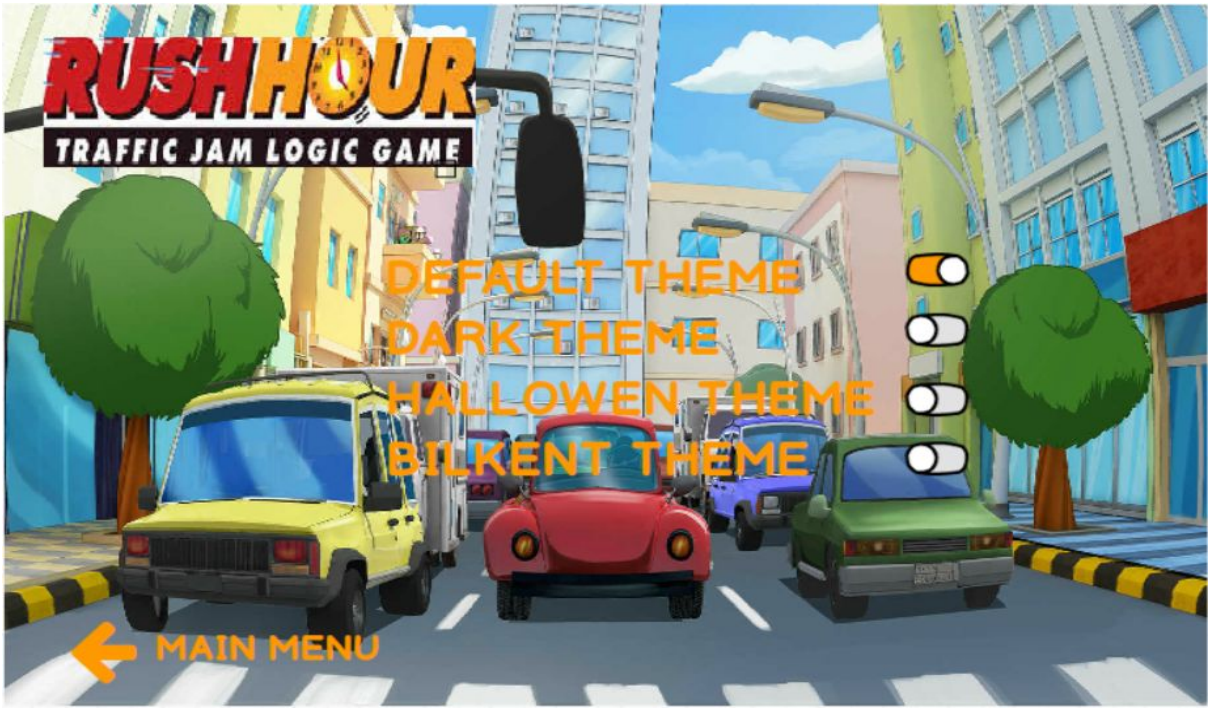
5.4.4 Multiplayer Mode Map Selection Mockup

This is the map selection menu that users are going to see after selecting the multiplayer mode. All the maps are open in this mode and there is no difficulty difference between them. Players can choose any one of them as they like.

5.4.5 Multiplayer Mode Mockup

This is an example gameplay from the Multiplayer Mode of the game where the escape cars are the yellow and white cars which have the same design pattern. The players are trying to escape the map first. The turn of which player is going to play is shown by the green arrow at the top side of the mockup. The users can play according to the wildcards dealt randomly (one by one). The turn count is shown at the top of the screen. The wildcard given to the player is also visible at the bottom of the screen.

5.4.6 Theme Selection Mockup



This is the theme selection menu. There are different themes for the game in this menu and user can select one of the as they like. The map visuals change according to the theme.

5.4.7 Car Selection Mockup

This is the car selection menu. There are different options for the car skin in the game. They are not all available besides default car. Other cars are going to be opened according to succeeded levels.

6. Conclusion

In this analysis report, we have clarified our analysis in order to design and implement an old school block sliding puzzle game called Rush Hour. We have tried to explain the most desirable properties of the game design and clearly explained the user interface in detail. Then, we have enriched the design of the game by adding extra features in comparison with the actual game.

Before we design the model, we watched some videos mentioning about how to play the game and what the rules are and also how the game is designed in real life so that we are sure about how to design the game and make improvements on it.

The meetings we organized allowed us to decide how to design the model and diagrams as efficient as possible. We set the common attributes and methods of the classes so that we construct the required relationships between classes. Thus, we are going to be able to implement the game easily later on.

In short, we have created this report to make our work easier in the implementation part because the project is not a 500-line project. So we made the report as clear as possible to benefit from it in the implementation time.

7. Glossary & References

https://www.thinkfun.com/products/rush-hour/
https://www.thinkfun.com/products/rush-hour-shift/