Department of Computer Engineering

Bilkent University

CS 319 Term Project

Section 1
Grup Şurup

Analysis Report

Project Group Members

1. Ata Coşkun
2. Muhammet Said Demir
3. Asuman Aydın
4. Zeynep Nur Öztürk
5. Tarık Emin Kaplan

Contents

Introduction

The cars and the challenges of them are some of the passionate of humankind in a sense of speed and control over something. Especially, the car is one of the daily life items that we use for everything mostly. In the game Rush Hour, as we decided to implement as being Group Şurup, this control and car passions can be felt in a minimalistic matter. A car that tries to get out of a parking lot where there are different types of car as being obstacles. The gamer has the control over cars and he/she has to use a logic and strategy to take the specific car out of the challenging level. One of the glimmering specialties of this game is nicely presented and enjoyable interface. There are cars in different types. As it can be implemented, there will be other types of obstacles to inhibit the car to get out and make the game more interesting in a way of challenging. The game also aims to entertain the user with different themes. While choosing which game we should implement, this kind of factors was effective and it can be concluded that it was a good decision to do this game because of the characteristics that we mentioned.

To give more sight of the specialties of the game; there are some parts we should mention.

First of all, single player and multiplayer mode give an opportunity to choose what the gamer wants. The single-player mode is for entertaining the lonely and self-care times to spend. On the other hand, Multiplayer mode lets the user socialize and bound with the other person by having fun while playing and also enjoying being the winner. Secondly, different themes and having music with it make the game more

interesting and easy to focus on the game. However, some people may not find this musical and thematically environment, So they can easily change or remove it from the settings in both the main menu and in-game menu. Another matter is help functionality of the game. There are two subjective over this property. One is Help option and the other one is getting a hint. The details will be given in below chapters but to say as final words in this chapter, Rush Hour is a game for the fun time and developing strategic aspect for anything and everything.

2. Overview

2.1 Game Engine

Rush Hour can be played in both Single Player Mode and Multiplayer Mode. The user will be controlling many cars in the parking lot and try to take the specific car out by using the directions up/down/right/left. Although taking the car out makes the level done, being successful in fewer amounts of time and steps is something considered in the game. There are stars to be filled according to the success of the gamer.

2.2 Board

The game will be played on the board that includes the cars, the obstacles. Also; levels get difficult within every board. Another thing about the levels is changeable obstacles. For example, if there is one type of obstacle in earlier levels, there would be more than one and more difficult obstacles in later levels because of challenge increases.

2.3 Settings

In game settings and main menu settings, the sound is depending on the player's request. So this makes the game more compatible with user interface qualities. More importantly, the levels do not have to be in the same shape. The option to change theme is enjoyable for the user. There are different themes such as space, forest or specific locations in our country and in the space.

2.4 Player

In the beginning, the player takes a card and the card will include the level. Each level, cards' content will get difficult. There will be no unlocking the level. Also, he/she can continue the game where it is left before.

2.5 Cards

This specialty gives some difference and enjoys the ability to the game. Different levels, different cards... So levels will not be directly given and it depends on the chance of the player. However, as mentioned before, level difficulty changes so cards change.

2.6 Cars

In different shapes and colors, there are many cars on one board. Combinations of setting the cars on the board change by levels. There is one specific car which player will able to differentiate from others. That car will be the car that player has to take out of the parking lot.

3. Functional Requirements

3.1 Play Game

The game will be the android game and controlled by users screen touch and drag. Entering the screen, users encounter the board which is prepared in advance. Boards have the own coordinates like nodes in the grid, cars and obstacles are put in these grid nodes. In the beginning, the board will have obstacles and cars and players' cars are put in initial positions. The game is based on movements of cars and board. Players have own car and try to overcome the obstacles and other cars and reach to exit. All cars will be able to move according to their positions. Obstacles will not be moved. The player may move cars vertically or horizontally according to their directions by dragging the car objects. Players also may move boards up and down if he gets the appropriate card in the multiplayer game. The game has 2 different modes:

3.1.1 Single Player Game

To play with single mode, users will select players in the main menu and choose the single mode from 2 different options. After that, they will encounter the level screen. The level screen has 20 different levels according to their difficulties. These levels will be hardcoded in game. Base data of these levels are taken from similar games. Players try to reach the exit by using the minimum number of moves. All levels have stars according to their difficulties. Each of the player moves will decrease the score that will affect the number of stars that the player gains from that level. At the single mode, players also may race against time, with the same logic of the number of movements, so that time spend will decrease their score. The time will start when the user makes the first movement. Also, some challenges might have time

limitations or number of moves limitations as challenger mode to further increase the difficulty of the game.

3.1.2 Multiplayer Game

Multiplayer mode is another type of game. The user will select players in the main menu and choose the multiplayer mode. At this mode, the fundamental logic remains the same. However, the game has some different properties. The game will be played with 2 player on the same board and one cellphone. Firstly, players select a card and the first card will determine the initial board. For each step, the player draws a card and decides their move according to contents of cards. These cards include the movement of boards and cars. Players might move the left/middle/right parts of the boards if they draw appropriate card by dragging. In this mode, players who reach the exit first will win.

3.2 How to Play

In the main menu, there will be a tutorial section. Players select the tutorial option to get information about the game. Tutorial section informs players about:

. Rules

. How they drag the cars and board.

. How they draw cards

. Stars, number of moves, time limitations

. Levels

The tutorial menu will also explain the game with video. Video demonstrates all actions to the player visually.

3.3 Settings

3.3.1 Sound settings

There will be background music in the game. If players want to increase or decrease volume, they go to settings option from the main menu or choose change settings during the game. Players also have mute and unmute options to adjust sounds.

3.3.2 Theme Settings

There will be different themes in the game such as ocean theme. These themes can be changed by settings from the main menu or player can change by using change settings option during the game. Therefore, themes are able to change before the game and during the game by the players.

4. Nonfunctional Requirements

4.1. Game Performance

The game will have high performance due to a different approach that we use. The data of each map (car coordinates, obstacle coordinates etc.) won't be stored in an array, instead, we'll store it at the inside of if statements that will hold the place at level selection part. There is no need to hold every map's information; instead, it would be sufficient to hold the information of only the map that the player would choose. For example, when the player will choose the 19th map, it will go to the if (level==19) part and at that place, we'll fill the board with the required information's. With this way of solution, we increased the performance and efficiency at the data storage part by n times (n is the number of maps). Other from that, the game will not have high system requirements since it does not have superior graphical elements. But it will have some animations that can be seen when the car is moving and stop moving.

4.2 User-Friendly Interface

The interface of the game is composed of elementary units so that everybody between 5-200 IQS will find their way through the interface. The main page contains play, settings, and a tutorial. Play page contains the option of multi and single and single contains levels. And if the user would face some confusion inside of levels, they can always go to the main page to get tutorial or they can get help by hints (at single version) inside of the game. At tutorial, every element inside of game will be explained clearly. At settings, the level of sound and music will be differentiated from each other with the same way of every mobile game does, the sound will contain a sound symbol and music a music symbol.

4.3 Compatible Interface with Real World

Since the game will hold car elements that will be drawn from the real world, the users will find it compatible with the real world. But they can always go and choose other fun themes that are not so compatible with the real world, for example, space theme which contains spaceships that will make the user feel as they were inside of Star Wars or Star Trek universe. If we go back to the normal theme, there will be obstacles just like in the real world that will block the user. There will be real life elements like trees, grasses outside of the game area which again makes it more like the real world.

4.4 Extendibility

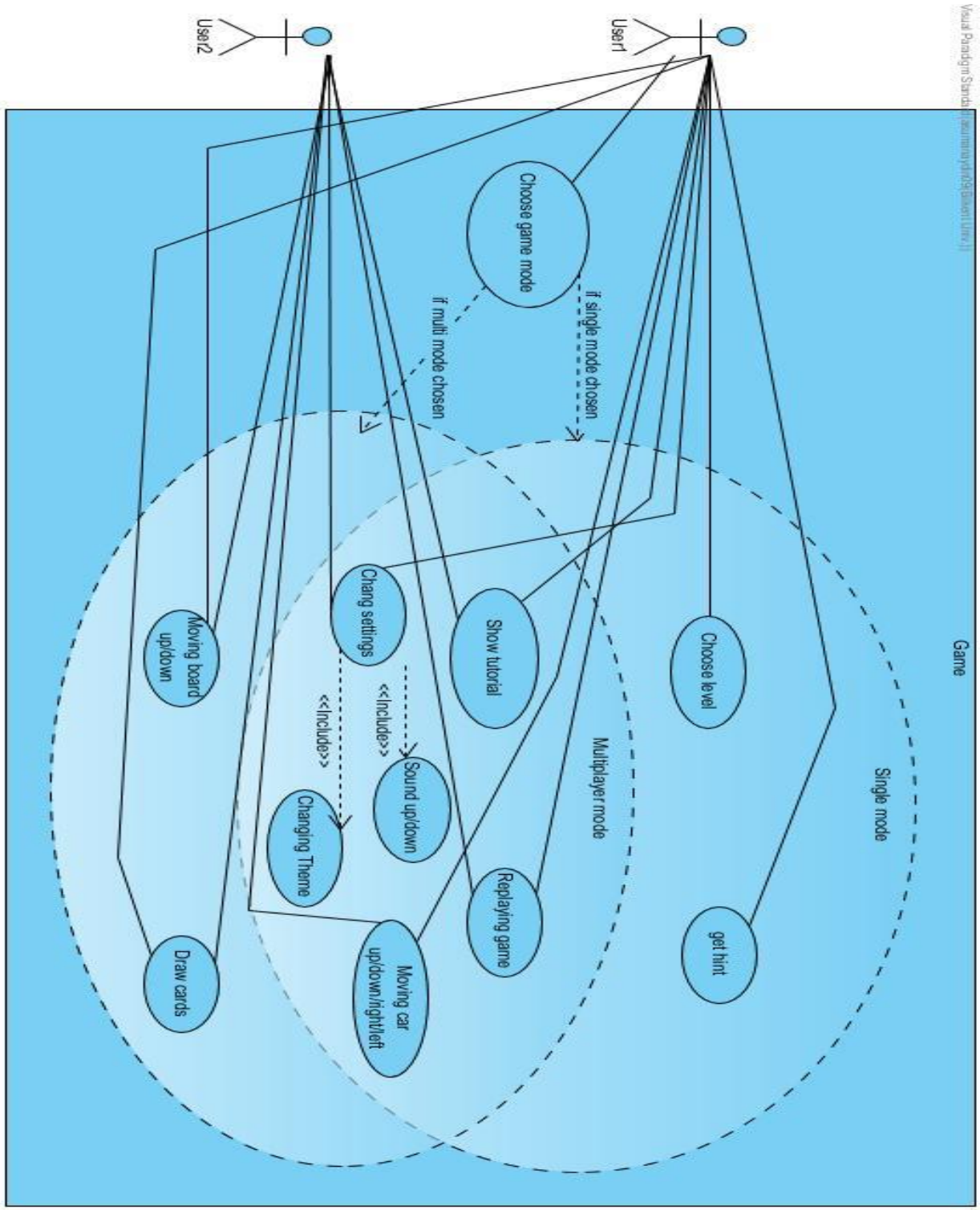
There will be 2 modes that the users can choose, one is single game mode and the other is the multiplayer game mode that requires 2 players. Each of them will contain different kinds of elements and game logic so that users will not be stuck within the same mode through the game and get bored. Multiplayer mode can be extended to 3 or more players which are one of our plans if we'll have time. Also, the single mode can be extended to 2 players which they will see each other's board at the right top of the page and they will compete with their scores. They will face 3 maps consecutively and the player with the better score all through the 3 games would win. This mode is different from the others since the users will not play it on the same phone, instead, they will connect each other through the internet.

4.4.1 Player Mode

The multiplayer version can be extended so that a player can play against an artificial intelligence. The AI will calculate and determine the best move with the card that it draw and play it. This can be extended to a user can play with more than one AI bots or more than one users can play against AI bots but again, if we have time.

5. System Models

5.1 Use case model



5.1.1 Play Game

Use case 1: The user chooses the single player mode:

1. The player starts to play the game after deciding the level and the difficulty to play. The main idea is the player moves the cars strategically and takes the specific (red) car out of the parking lot.
2. Each move that the player does decreases the score of the level, same goes for time. Longer the player plays, lower the player gets points. After finishing the level (which means getting the red car out) start will be given according to score.
3. The player can always restart the game of undo the last action
4. If the situation is even worse than that, the player can choose to get hint. But each time the player hits the hint, score will decrease. Hint will hold a queue that possesses right moves from start to end. Each time the player makes a move, their last movement is enqueued to the queue. Each time they hit the hint, queue will pop one item

Use case 2: The user chooses the multiplayer mode:

1. The very fundamental logic will be the same on the multiplayer mode which is to get the red car to the exit. But this time, each player has a red car and they have to move to their own exits.
2. Each turn, the players will be able to draw a card which contains the number of moves that the players can do, or they can move the board in order for them to get the exit or to block the other player/s

3. 5.1.2 Pause

Use case 1: The users/ user chose to stop the game:

They are in the game but something out of play happened, they press the pause button and the game will be saved till the state they have been. At the main menu, there might be a continue button for them to continue on their current game or they can reselect that specific level to continue

5.1.3 How to Play

Use case 1: The user is new in the game:

Newly user opens the game and in the main screen there will be seen the tutorial option. They can press and learn how the game is played. They can redo this action limitless time

Use case 2: The user is stuck within the game:

In case of the user does not have any idea about how to finish the level, they can press the hint button which is explained above.

5.1.4 Settings

Use case 1: In game settings chosen

Use case 1.1: Change theme

The player got bored from the current theme in the game or it caused the player to lose the attention, so they want to change the theme. In the game, they press the settings icon and the settings screen comes on the screen. With choosing the change theme option, the theme flow will appear and passing either left or right the player will be able to choose the theme again. This action can be done inside or outside of the map which means main menu or within levels.

Use case 1.2: The sound settings

The theme includes a sound that has a melodic rhythm. However, not everybody has to like this music and listen. So when they press the sound settings, they can either change the volume or mute the sound.

Use case 2: In main menu settings chosen:

Use case 2.1: Change theme

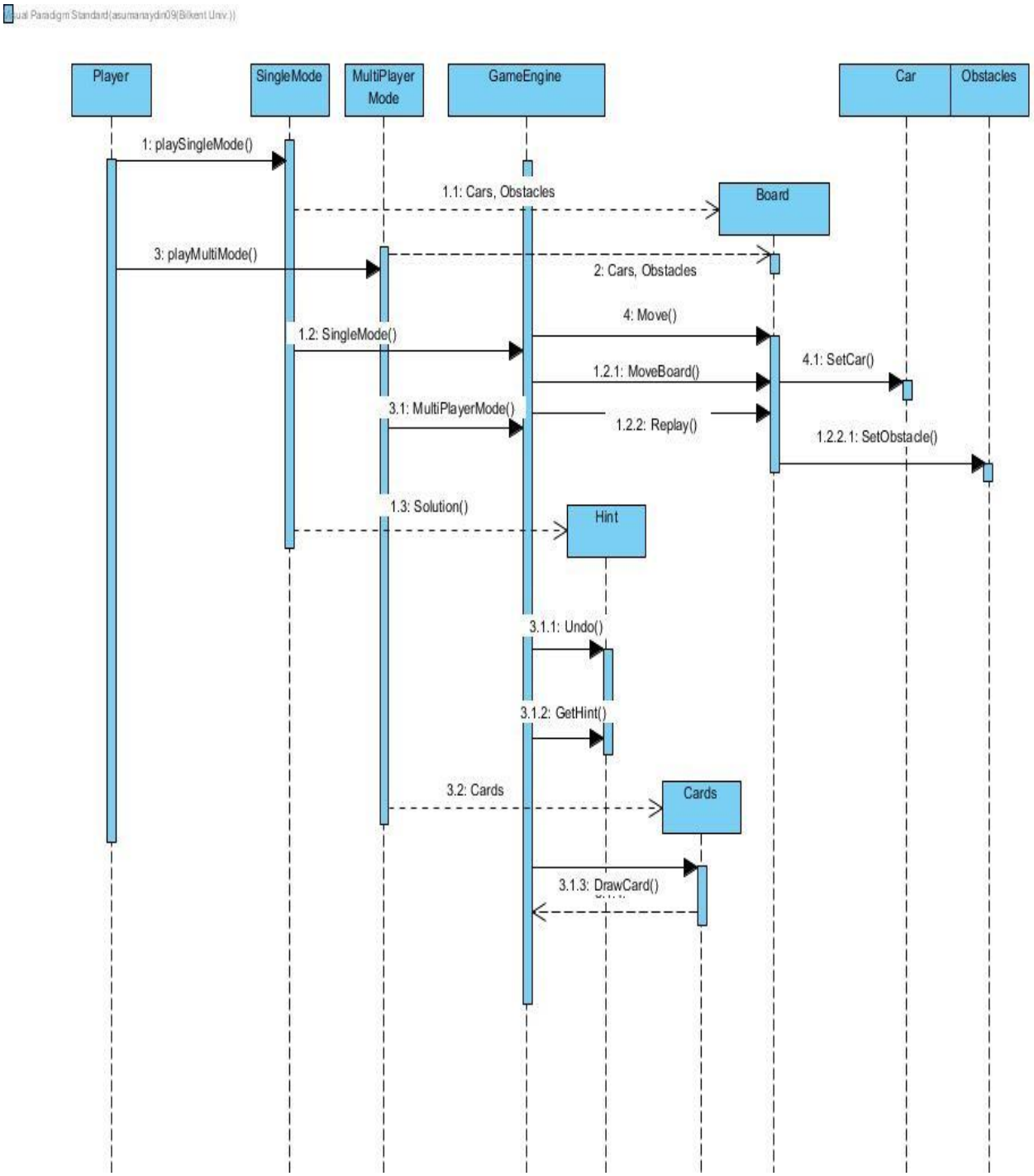
The player has favorite theme in the game and enjoys the game with the specific theme, so they want to set the theme before starting. The settings icon and the settings screen come on the screen. With choosing the change theme option, the theme flow will appear and passing either left or right the player will be able to choose the theme again.

Use case 2.2: The sound settings

The situation in Use Case 1.2 is applied also here. Only difference is the player sets the sound from the beginning.

5.2 Dynamic models

5.2.1 Sequence Diagrams



5.2.1.1 Start Game

When the player starts the game, they either choose single player mode or multiplayer mode. The game will be started by the game engine and the cars and the obstacles will be set. There will be hint option at the single game. In multiplayer mode, board, the car and the obstacles will be set. Within the game players will draw card each turn which determines their number of action limits.

5.2.1.2 Settings

Settings depend on the Rush Hour object and changes will be driven by on the subject. So the steps are inside of the object instances. As Rush Hour object has the theme and sound of this theme, changes will be on this class.

5.2.1.3 Help

Help is not changing according to any action on the game. So it is not considered in the sequence diagram. However, the hint is depending on the other functions and also the queue that the user's moves are kept. The moves are stored in the single mode and when the user wants to get the hint, the solution and the hint will be compared. As a result of this comparison, the next move's hint will be given. The function will send the result back the player.

5.2.1.4 Resume

The whole idea of playing game is also not starting from the beginning every time. So continuing from where you left is a good option. But it is not something changeable with the function's output. It is saved while the game is closed by the user and it will just push the last version of the board.

5.2.1.5 Game Engine

The game is all controlled over the game engine. Before and after game engine there is set up and the game. So when the user wants to change something on the game, the game engine is buffer between the board and the place where there are levels, modes, and themes. The moves, car directions and obstacles are set in the subclasses of the game engine and the game engine controls all of them.

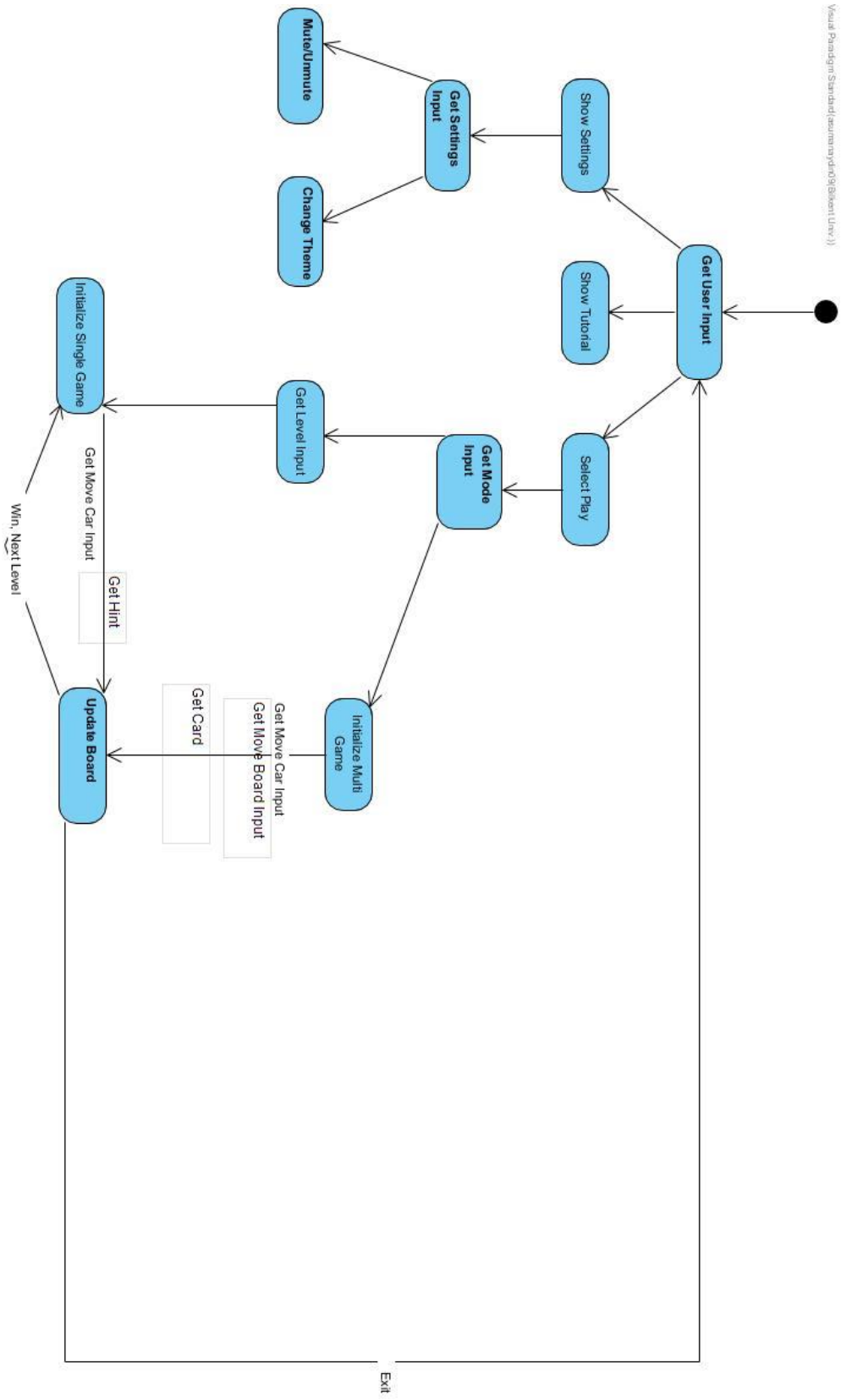
5.2.1.5 Board

Board gets the information from the levels section. When the user decides which level he/she wants to play, the information according to that level will be loaded into the board and then, that board will be loaded to the game engine. All screens (GUI) will be set by the game engine, however; changes will be driven by the board. For example, the user moved one car to the right, the update of x position will be connected to the board and game engine will update the GUI. Also, the user wanted to replay the game, the board will be updated with taking the level information from the levels section once again. The car, obstacle positions will be updated the same in the beginning. The score and the time will vanish to zero.

5.2.1.6 Cards

Cards will be a feature for multiplayer version only. Each turn the players will be able to draw a card that determines their number of action limit. Player will draw it randomly.

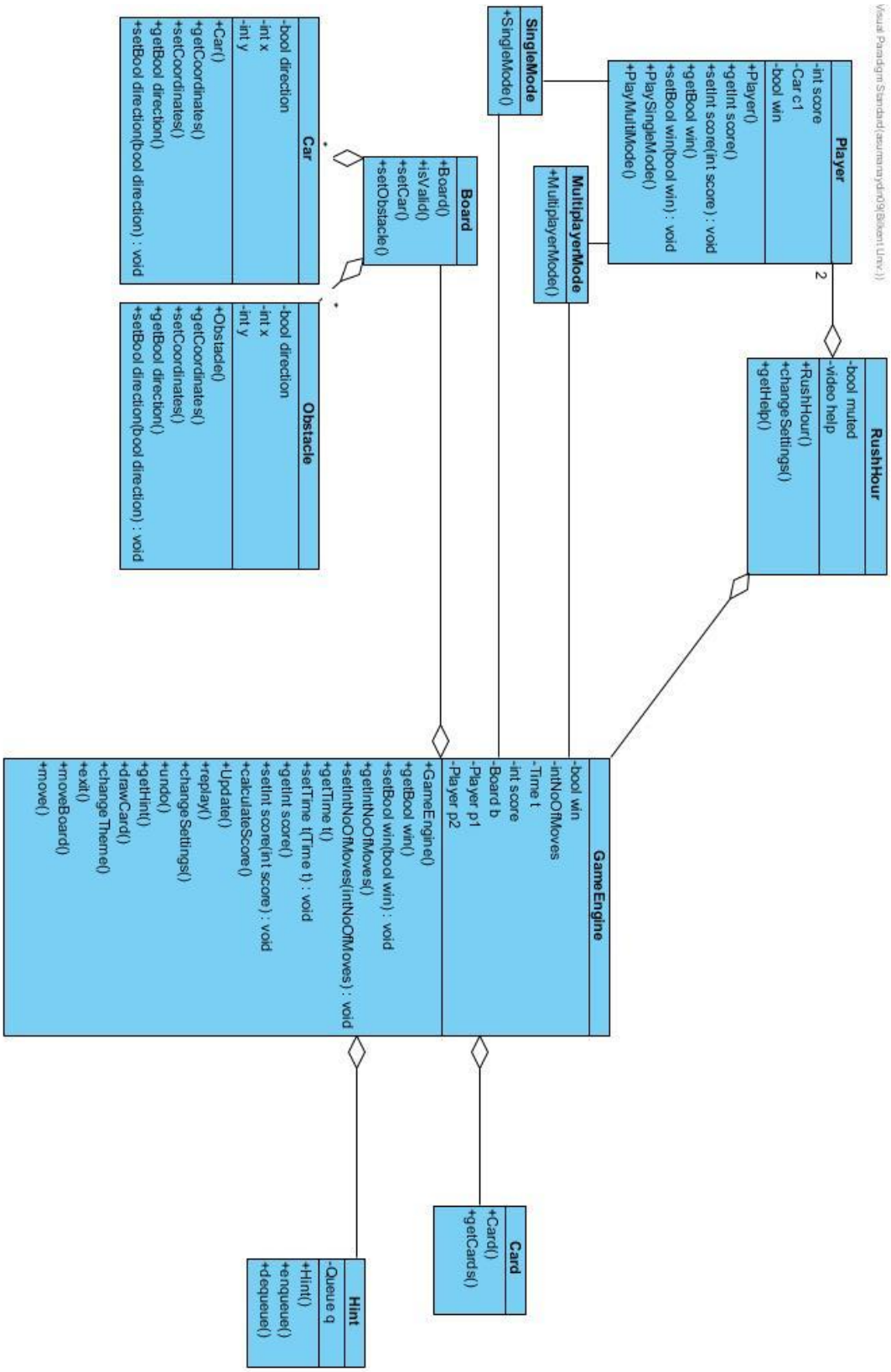
5.2.2 Activity Diagram


Get User Input
Show Settings
Show Tutorial
Select Play
Get Settings Input
Mute/Unmute
Change Theme
Get Mode Input
Get Level Input
Initialize Single Game
Initialize Multi Game
Get Move Car Input
Get Hint
Win, Next Level
Update Board
Get Move Car Input
Get Move Board Input
Get Card
Exit

Activity diagram demonstrate the steps according to user input. There are different orientations:

- **Showing setting**: The user wants to change the settings. Then user input differentiates to two either sound changing or theme changing. When the user chooses the sound settings, input adjustment will be given to rush hour object and output will be either muted or unmuted. When the user chooses the theme change setting, the flow of different themes will be taken from the stored data. While the player is choosing, he/she can also see sample display and decide. When the input theme is entered, all game will be set up.
- **Show tutorial**: There is tutorial to show according to this input. There is video that includes everything about the game. Output will be this video in this action.
- **Select play**: The player chooses to play. Then there are two options. The player can either choose Single Player Mode or Multi-Player Mode. When he/she chooses the single mode, the level comes according to where the player left the game in a sense of state. Then the player starts to play and depending on the directions of cars, the number of steps is kept and the time will decrease while playing. When the exit is seen, the score will be displayed and according to score, the stars will be given. When he/she chooses the multiplayer mode, the game will start with the default level. Each turn the players will be able to draw a card that determines their number of action limit. Player will draw it randomly. They will be able to move the board as well. And the rest is the same with the single-player mode.

5.3 Object and class model


Visual Paradigm Standard(asumanyid-09@ilkent.unv.t)
Player
-int score
-Car c1
-bool win
+Player()
+getInt score()
+setInt score(int score) : void
+getBool win()
+setBool win(bool win) : void
+PlaySingleMode()
+PlayMultiMode()
RushHour
-bool muted
-video help
+RushHour()
+changeSettings()
+getHelp()
2
SingleMode
+SingleMode()
MultiplayerMode
+MultiplayerMode()
GameEngine
-bool win
-intNoOfMoves
-Time t
-int score
-Board b
-Player p1
-Player p2
+GameEngine()
+getBool win()
+setBool win(bool win) : void
+getIntNoOfMoves()
+setIntNoOfMoves(intNoOfMoves) : void
+getTime t()
+setTime t(Time t) : void
+getInt score()
+setInt score(int score) : void
+calculateScore()
+Update()
+replay()
+changeSettings()
+undo()
+getHint()
+drawCard()
+changeTheme()
+exit()
+moveBoard()
+move()
Board
+Board()
+isValid()
+setCar()
+setObstacle()
Car
-bool direction
-int x
-int y
+Car()
+getCoordinates()
+setCoordinates()
+getBool direction()
+setBool direction(bool direction) : void
Obstacle
-bool direction
-int x
-int y
+Obstacle()
+getCoordinates()
+setCoordinates()
+getBool direction()
+setBool direction(bool direction) : void
Card
+Card()
+getCards()
Hint
-Queue q
+Hint()
+enqueue()
+dequeue()

5.3.1 Rush Hour Class:

This class includes game settings, tutorials. It has a player and a game engine. The main menu starts with this class

5.3.2 Player Class:

This class represents players. There is a one player in single mode and there are multiple players in multiplayer mode. Player class reach to Single Mode class or Multiplayer Mode class.

5.3.3 Single Mode Class:

This class keeps the board data. It works as our database for single mode. We get the initial data from table game or similar online game according to their difficulties. After user chooses single mode, he/she choose a level from 1 to 20. After that a board is initialized in this class and sent to game engine.

5.3.4 Multiplayer Mode Class:

This class keeps the board data for multiplayer game. We get the initial data from the default level description at levels part. Each turn the players will be able to draw a card that determines their number of action limit. Player will draw it randomly. Or they can move the board.

5.3.5 Board Class:

Board class will be the base system the game is played on. Cars and obstacles hosted in this class. Game engine uses valid positions or not.

5.3.6 Car Class:

This class represents players' cars and other cars. All cars will have a coordinate on board. In addition, all cars will have a direction horizontally or vertically.

5.3.7 Obstacle Class:

Obstacles will have a direction and coordinate on the board. Players will not be able to move these obstacles.

5.3.8 Card Class:

Cards includes the information about actions in the multiplayer game. Cards consists moves of cars, obstacles or board. For example, one card may include 2

moves up for cars or 3 moves left for obstacle or 2 moves change of boards. In addition, first card will determine initial board before the game starts.

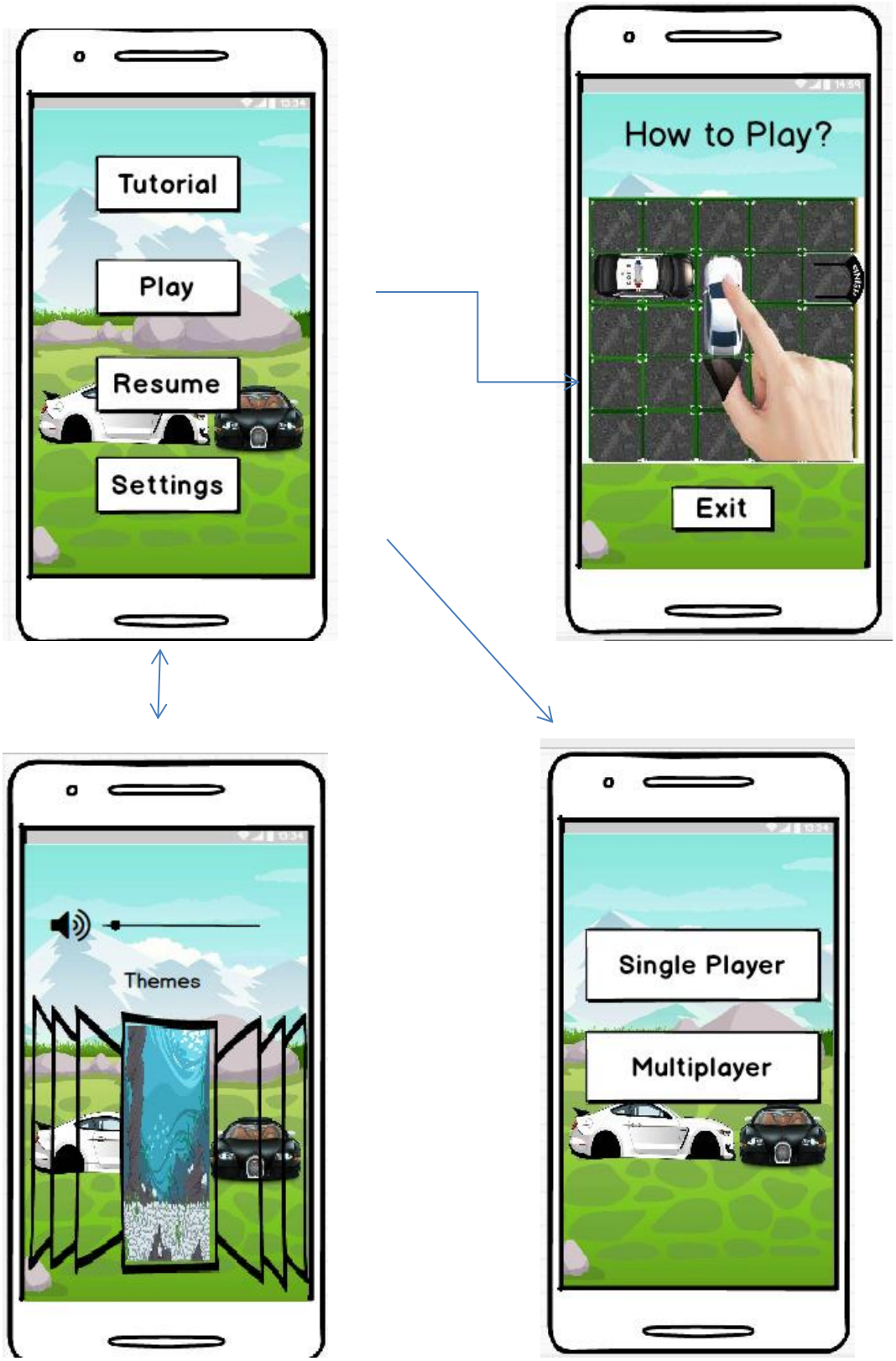
5.3.9 Hint Class:

This class keeps the moves of players in queue. When the player wants to help, this queue gives some hint to player.

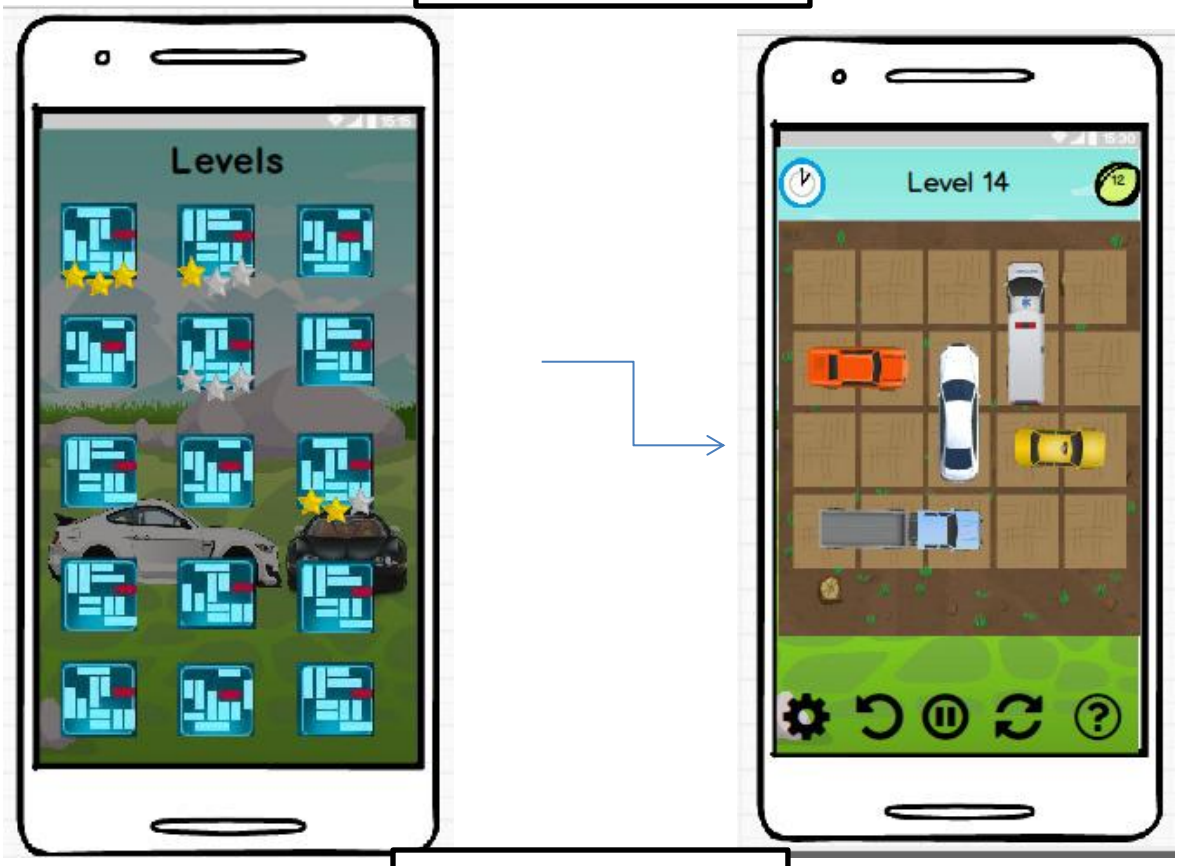
5.3.10 GameEngine Class:

GameEngine class is the fundamental class of all actions in the game. This class manage the all game changes. It has a game board and changes the board according to players' input. The variables like number of movements, time, scores are stored in GameEngine class. All actions in the game such as draw cards, move cars, move boards are handled in GameEngine. Moreover, players might change the sound settings and theme setting during the game. GameEngine produces and refresh **all GUI parts and instances in the game with update method.**

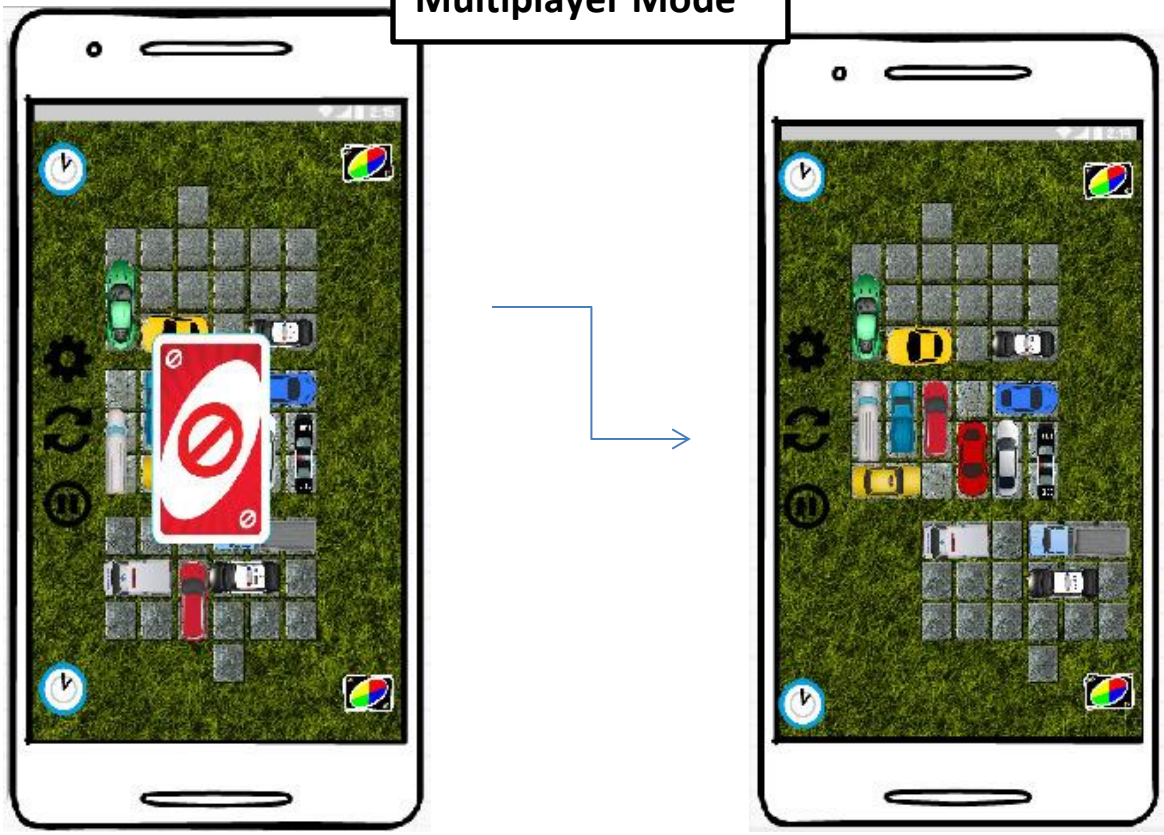
5.4 User interface – navigational paths and screen mockups



Single Player Mode



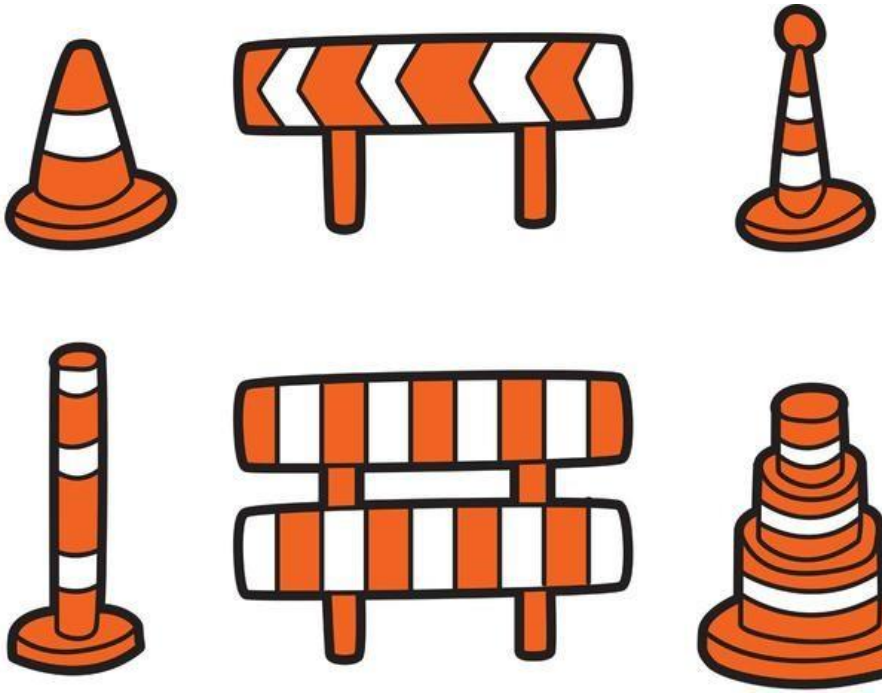
Multiplayer Mode



Theme Example: Space

Theme Example: Forest

Car examples



Obstacle Examples

6. Conclusion

As a result, Rush Hour is a game that can be played in both single player mode and multiplayer mode. There are levels of different difficulties. Also, in the game in any level, there are different obstacles such as stop signs, barriers and etc. The player will be able to see the score while playing and if they want to replay, they will be able to start again. One of the fun parts of Rush Hour is random played levels. Taking one game card and playing it append more joy into the game. Another matter is that the player can continue from where he/she left the level by using the resume option. Also, by getting a hint in the game, the game may not be boring for the people who do not like struggling in a sense that they just play a game. Hence, these are the things make Rush Hour separable from other games.

7. Glossary & references

https://www.vecteezy.com/vector-art/87351-road-traffic-cartoon-icons-vector

https://tr.pinterest.com/pin/301107925061187211/?lp=true

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

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