CS 319

Object-Oriented Software Engineering

Spring 2019

Analysis Report

Road Blocker: Alpha

Section 1 / Group J

Mert Özerdem
Onur Mermer
Cemil Şişman
Burak Alaydın
Doruk Altan

Table of contents

1. Introduction

The game focuses around Catching escaping thief by using police cars and blocking thief's road to escape. Player has to position his or her police cars to zero in on the suspect red car. While environment (placement of the building blocks) changing at each level. The game requires management of limited resources with a given environment. We inspired this game from the physical board game named as “RoadBlock”[1].

The main difference of our game with the original one is digital additions we made to the game such as time and scorekeepers and also expendable map option which were not a thing in the physical board game.

Scorekeeping and timekeeping improvements to the game done due to improving competitive gameplay of the game. The expendable map change is also creating a competitive environment but the main aspect of this improvement is to increase playability by increasing map count and creating bigger and better, harder maps to complete.

2. Overview

Gameplay of “Road Blocker: Alpha” is very simple, first the level that designed to beat by player is prepared which includes a placed red car and some building blocks as obstacles. Player have to use different blocks with police cars on it to block the placed red car’s escape towards edges of the table. Player can rotate those blocks before picking and placing them according to his needs. There will be a set of numbers of blocks according to difficulty of level. Table size may change depending on the level. Also some levels may provide 2 red cars to challenge the player.

Game mainly targets 7-15 years old as audience but in order to provide a challenge for the players who seeks challenge, game includes a score system. This way different players can play and challenge to each other to beat others scores which resembles the old school Arcade games where people have to play many times to master the game and challenge other people for fun. Of course game does not force player to beat the level in limited time but score system provides an alternative for different types of players.

3. Functional Requirements

In this part functional requirements that are needed to be fulfilled going to be described.

3.0 Main Menu

When player run the game the main menu screen will appear and user can navigate through different parts of the menu via using this main menu screen. Each section will explain the related requirements about different sections of the menu.

3.1 Play the game (Level Selection Screen)

Option to play the game will be provided at the top of the main menu options. This option will lead the player to menu that player can choose between 60 different levels. Each level will be shown in a little box as an overview to player so player can choose the level according to the challenge provided before starting to play. At the initial state game will provide 10 levels to the player and as player completes the levels different levels will be provided to choose. Player has to complete 5 different levels to unlock new 10 different levels. Also, under each level option box highest score will be shown which indicates how fast the level is completed with the nickname of the player that completed that level in that PC so that same player can try to challenge himself or different players try to beat others records. Those scores will be later on explained in detail at related section. After player choose the level game starts.

3.1.1 Gameplay

As the game starts initial blocks (red car and buildings) will be placed at the table in game screen. Player will be provided with different blocks with police cars to place on table to block the red car’s escape. Player can use the mouse to move blocks or click on rotate buttons to rotate the block before picking and placing it on table. There will be exit and pause buttons in case user wants to pick another level or in case urgency. Exit button will return the player to level selection screen while pause button will pause the level but also darken the screen so player cannot abuse the provided option to beat the level faster. Timer in game screen will indicate the time that level has been played. Collisions sounds also help player to indicate if the blocks colliding or if it heard after last block placed to indicate it is not the correct solution to this level. Until player placed the blocks correctly game will continue.

3.1.2 Post-Game Screen

After the player beat the level with correct blocks game will show the time, score, used blocks to beat the level in post-game screen and provide a section to player for entering a nickname so it will be saved in High scoreboard. After player enter his nickname, he can use the return button to return level selection screen.

3.2 Highest Scores Screen

Player can use this screen to check the highest 3 scores with nicknames of the players that beat the record for each level. Player can navigate through screen to display records for different levels or just return to main menu.

3.3 How to Play Screen

This screen will explain the details about gameplay, rules, functions of different buttons and menu screens to player. This way player can understand the game in a comprehensive way. If player wants to do so he can return to main screen via button.

3.4 Options Screen

This screen will provide tick boxes for player to mute/unmute the sounds and the music of the game.

3.5 Credits Screen

This screen will provide the names of the developers of this game with some information about them to reach them via mail.

4.Non-Functional Requirements

4.1 Game Description

Many users may not know what to do when they start playing the game. To prevent confusion, the game should provide an explanation part where the purpose and the gameplay explained.

4.2 Response Time

Providing speed is as important as providing quality. About this issue, what Road Blockers could possibly suffer from is level-ending scenarios. Each time the user finishes the given puzzle, program will try to find a path which will cause him to fail. Whether it finds or not, this part must take at most 4 seconds to provide a fast application.

4.3 Level Accessibility

When a user starts the game, he might expect a challenging game. The game should be able to keep his interest by providing him the appropriate difficulty level. So the user must be able to choose any difficulty/level without completing unnecessary levels before.

4.4 Tracking Score

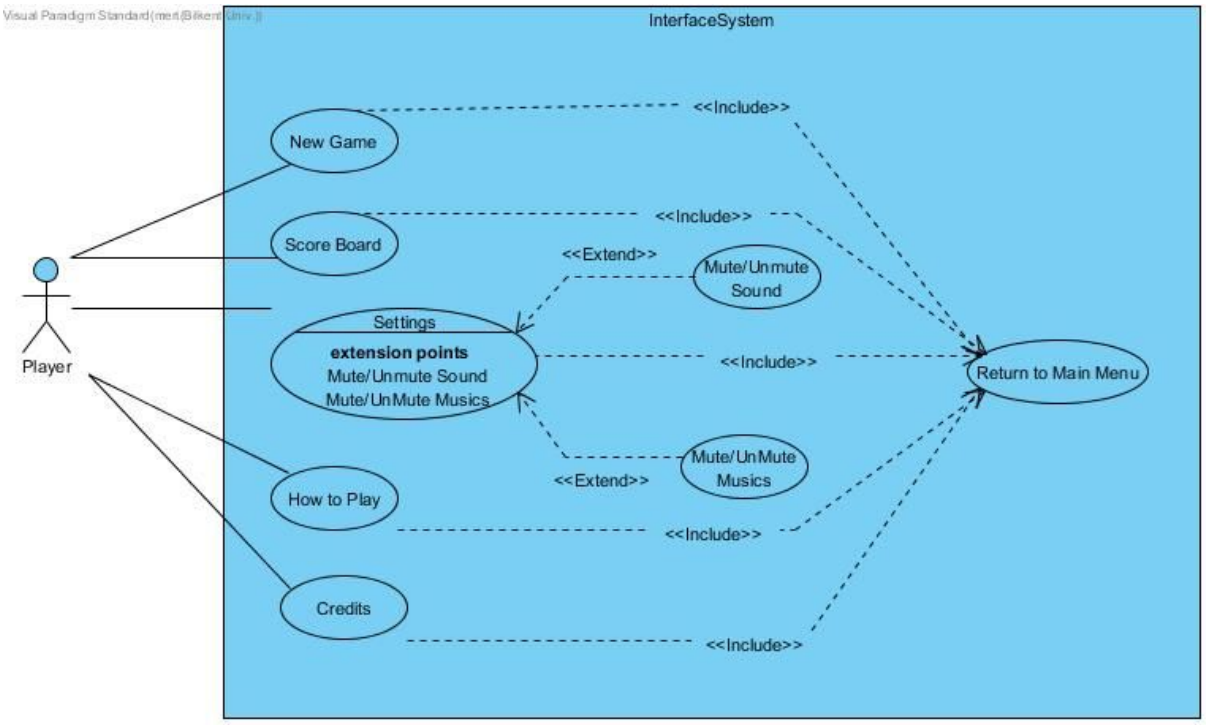
Score system is a good way to let the user check how successful he is. The user should be able to see his score after each level and he should be able to compare it with the best scores by seeing “Highest Scores” screen.

5. System Models

5.1 Use Case Models

Use case diagram represents a user’s interactions with the game which involve many interactions that user can do while playing the digital game of Road Blockers. In this part, main interactions of the game are represented with their simplest forms as use case models.

5.1.1 InterfaceSystem



5.1.1.1 New Game

Use Case name: New Game
Primary Actor: Player
Stakeholders and Interests:

1. Player wants to play a new game
2. System creates a new game

Pre-conditions:

1. System has to be bug free to create new game each time
2. Player should be in main menu

Post-conditions: -
Entry conditions:

1. Player has to choose new game button

Exit conditions:

1. Player has to choose “Return Main Menu” button

Success Scenario Event Flow:

1. Player starts the game exe
2. Player selects new game button
3. System starts a new game
4. Player exits from the game

5.1.1.2 Score Board

Use Case name: Score Board
Primary Actor: Player
Stakeholders and Interests:

1. System keeps information about recorded scores
2. Player wants to learn about his or her score placement

Pre-conditions:

1. System has to be built to keep track of the scores

Post-conditions:

1. Player cannot change scores manually

Entry conditions:

1. Player has to select “Score Board” Button

Exit conditions:

1. Player has to choose “Return Main Menu” button

Success Scenario Event Flow:

1. Player chooses “Score Board” button
2. System creates a table according to previous information that are kept
3. Player exits.

5.1.1.3 Settings

Use Case name: Settings
Primary Actor: Player
Stakeholders and Interests:

1. System has to be flexible to change music state
2. System has to be flexible to change sound state
3. Player wants to change music or sound state

Pre-conditions:

1. Music has to be on
2. Sound has to be on
3. Music has to be off
4. Sound has to be off

Post-conditions:

1. Music shuts down
2. Sound shuts down
3. Music starts
4. Sound stars

Entry conditions:

1. Player has to select “Settings” Button

Exit conditions:

1. Player has to choose “Return Main Menu” button

Success Scenario Event Flow:

1. Player pushes “Settings” button
2. Player changes Sound to off
3. Player changes Music to off
4. Player returns to main menu

5.1.1.4 How to Play

Use Case name: How to Play
Primary Actor: Player
Stakeholders and Interests:

1. Develepors inserts instructions to a how to play file
2. Player wants to know rules of the game

Pre-conditions:

1. System needs a system of rules to display on screen

Post-conditions: -
Entry conditions:

1. Player pushes “How to Play” button

Exit conditions:

1. Player has to choose “Return Main Menu” button

Success Scenario Event Flow:

1. Player pushes “How to Play” button
2. System displays instructions on the screen
3. Player returns to main menu

5.1.1.5 Credits

Use Case name: Credits
Primary Actor: Player
Stakeholders and Interests:

1. Player wants to know contributors of the game
2. System needs credits to display

Pre-conditions:

1. Developers has to put some credits text in to game to display on screen

Post-conditions: -
Entry conditions:

1. Player pushes “Credits” button

Exit conditions:

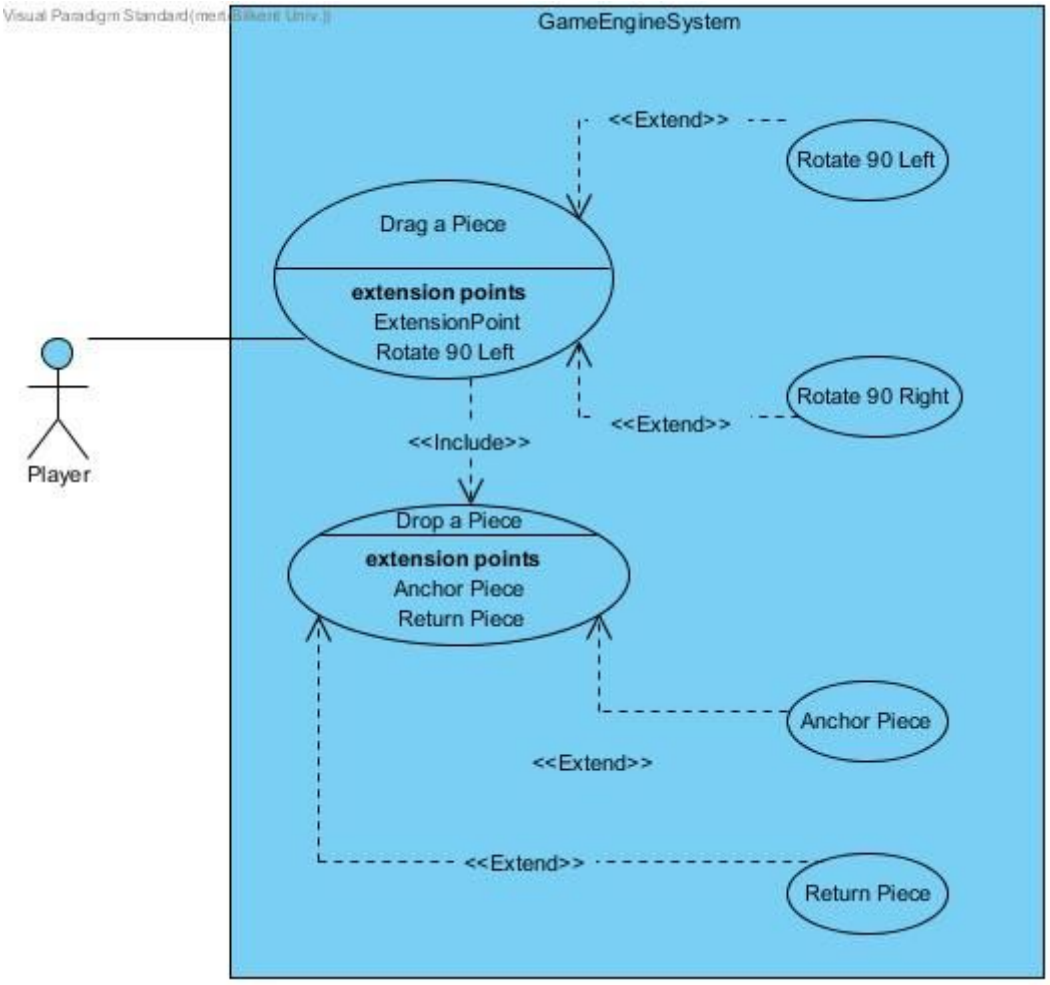
1. Player has to choose “Return Main Menu” button

Success Scenario Event Flow:

1. Player clicks on “Credits” button

2. Player views contributors
3. Player returns to main menu

5.1.2 GameEngineSystem



5.1.2.1 Drag a Piece

Use Case name: Drag a Piece
Primary Actor: Player
Stakeholders and Interests:

1. Pieces have to be movable
2. Player wants to move the piece

Pre-conditions:

1. The game needs to be running
2. Pieces are movable

Post-conditions: -
Entry conditions:

1. Player click on a piece
2. Player holds the left mouse button after clicking the piece

Exit conditions:

1. Player lifts his or her hand on the left mouse button

Success Scenario Event Flow:

1. Player Starts game
2. Player clicks and holds a piece on the screen
3. Player rotates piece 90 degrees right
4. Player lifts his or her hand from the left mouse buttom
5. Piece anchors to place

Alternate Event Flow:

1. Player Starts game
2. Player clicks and holds a piece on the screen
3. Player rotates piece 90 degrees left
4. Player lifts his or her hand from the left mouse buttom
5. Piece returns to its original place

5.1.2.2 Drop a Piece

Use Case name: Drop a Piece
Primary Actor: Player
Stakeholders and Interests:

1. System needs to figure out whether or not the piece is suitable to the dropped location
2. Player wants to place piece on a place on the active board

Pre-conditions:

1. System is programmed to putting pieces with playability in mind

Post-conditions:

1. Piece cannot put anywhere on the board so system need to relocate piece if some error occurs after dropping the piece

Entry conditions:

1. Some piece has to be already in drag mode

Exit conditions: -
Success Scenario Event Flow:

1. Player drops the piece on active board
2. Piece anchors to the available space on the board

Alternate Event Flow:

1. Player drops the piece on active board
2. Piece cannot anchor a place on the board.
3. Piece returns to original place

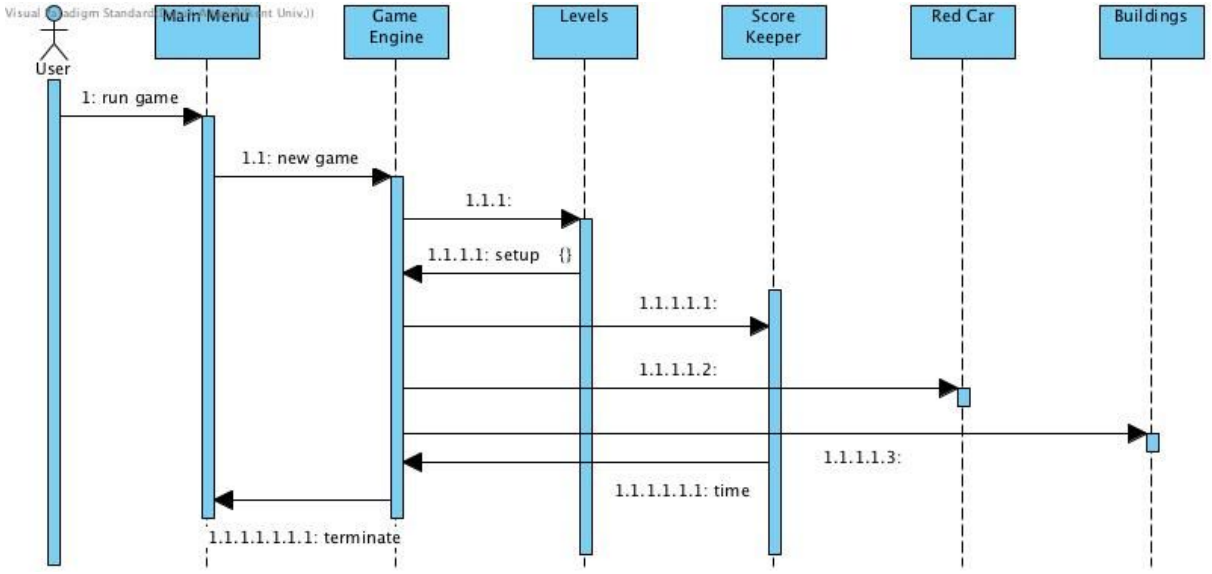
5.2 Dynamic Models

5.2.1 Sequence Diagrams

Sequence diagrams describe the dynamic behavior between objects of the system. The following diagrams aim to show the interactions of objects during important and common operations of the game. In order to make the diagram more understandable, only related steps will be included.

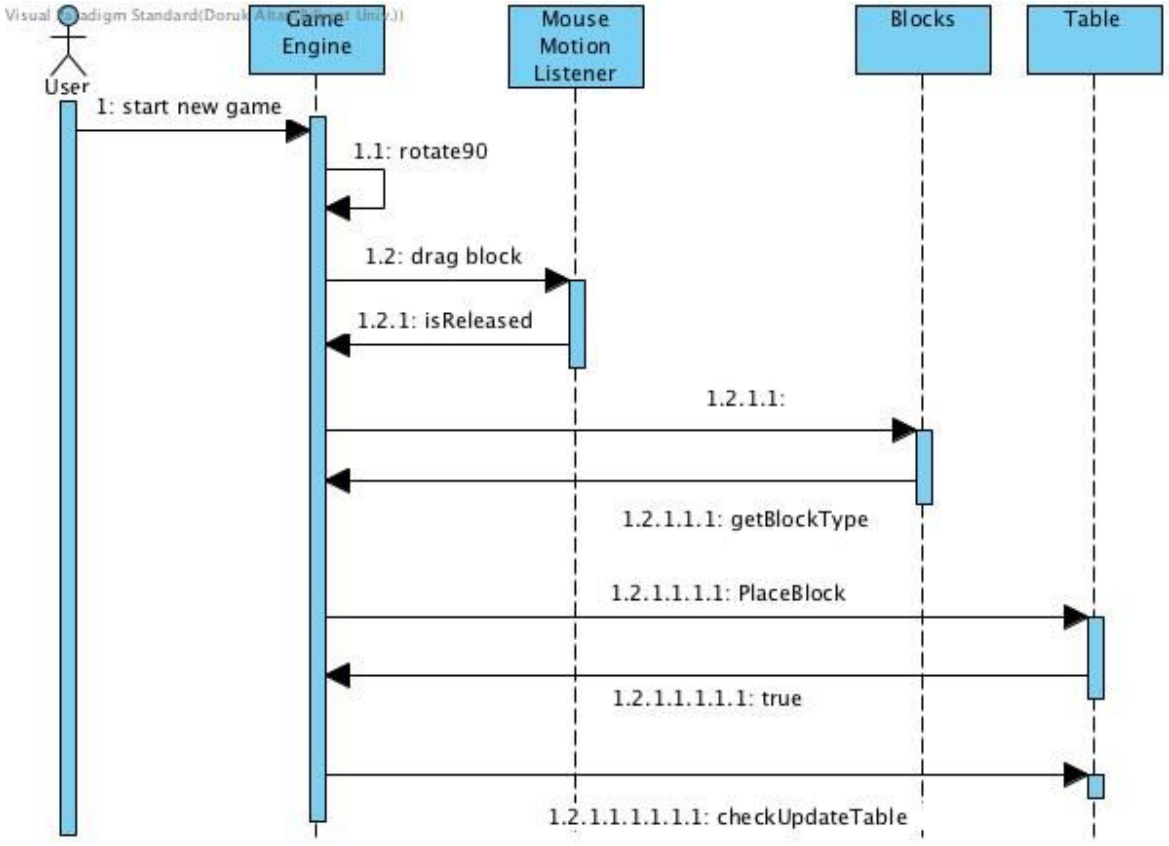
5.2.1.1 Starting a new game

In the scenario below, user chooses the play option in the main menu. Game engine generates the layout and creates the necessary objects according to the level.



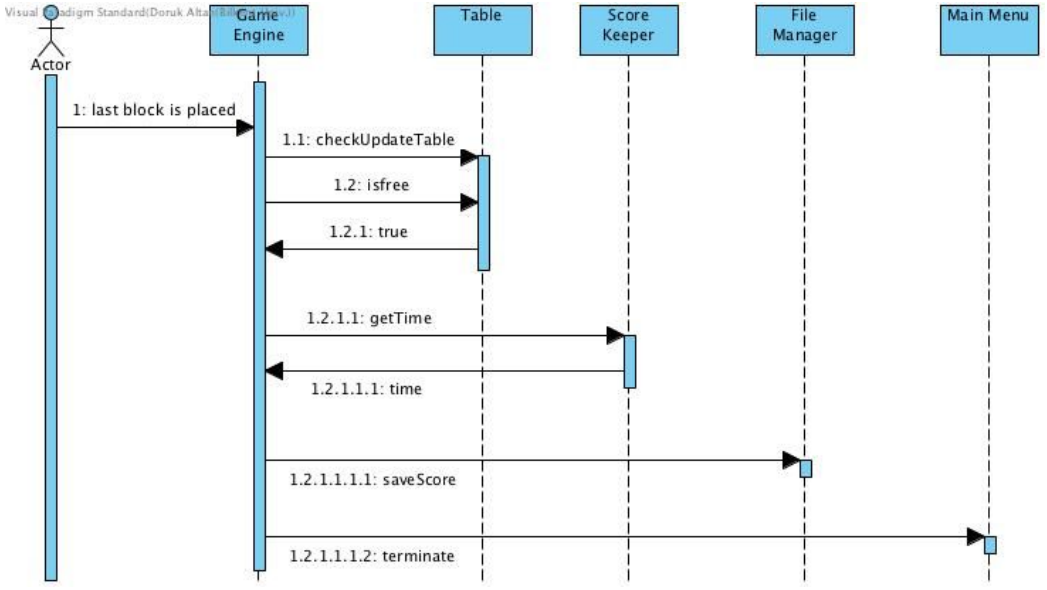
5.2.1.2 Placing a block

In this scenario, user first clicks on one of the rotation buttons for a block. Then drags and drops a block. Game engine then checks for any collisions and if the move is valid, updates the table accordingly.



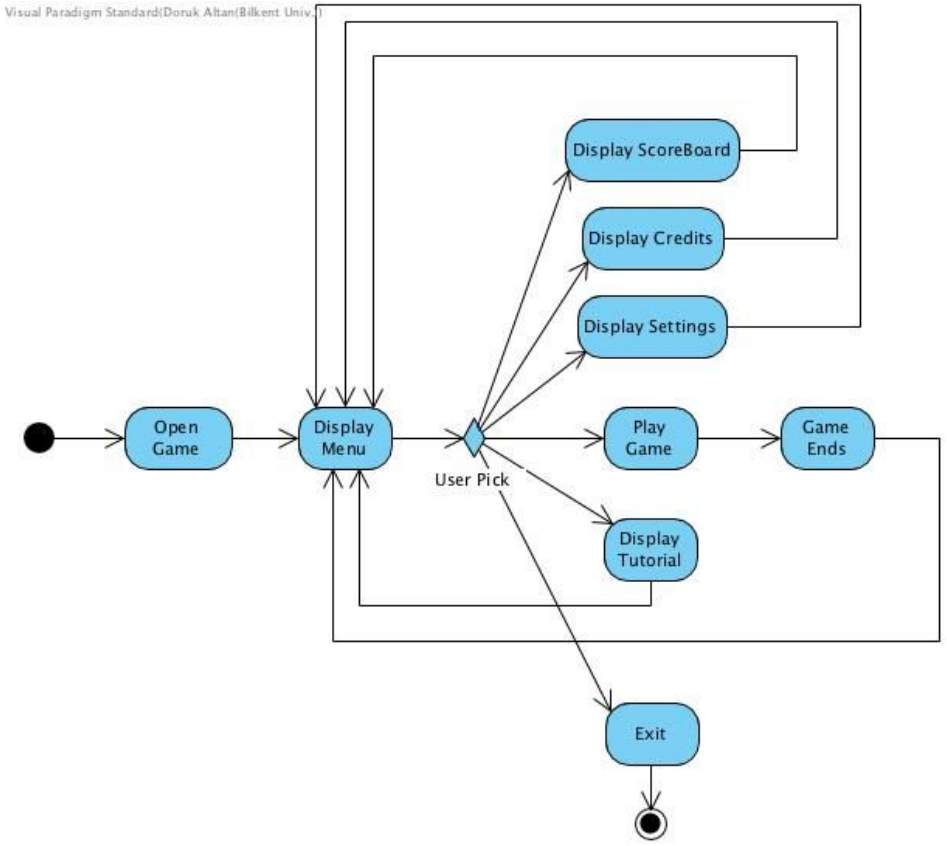
5.2.1.3 Finishing a level

In this next scenario, user places the last block. After the final update of the table is done, game engine calls isfree() to find if there are any possible escape routes for the red car and so determining whether the user's solution is valid or not. If the solution is indeed correct, relative data like time is saved.

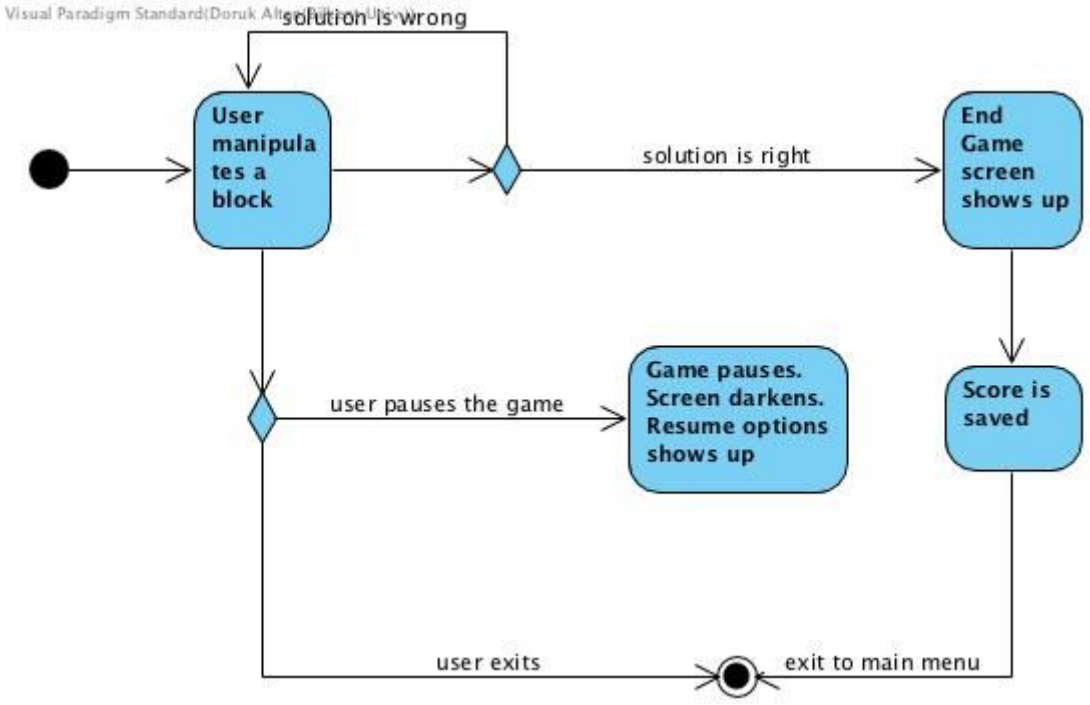


5.2.2 Activity Diagrams

5.2.2.1 Main Menu Activity Diagram

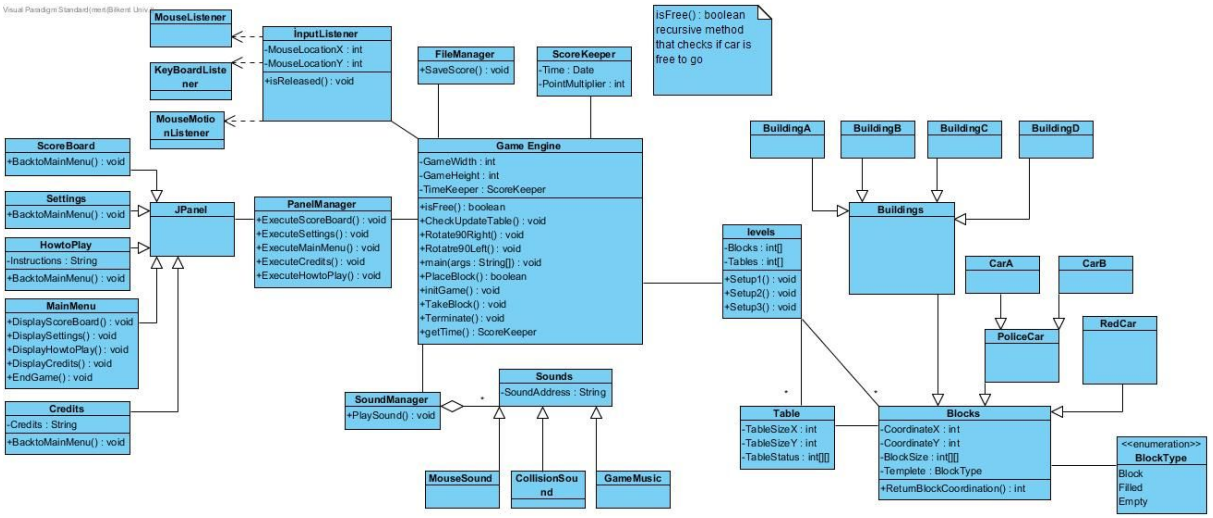


5.2.2.2 Play Game Activity Diagram



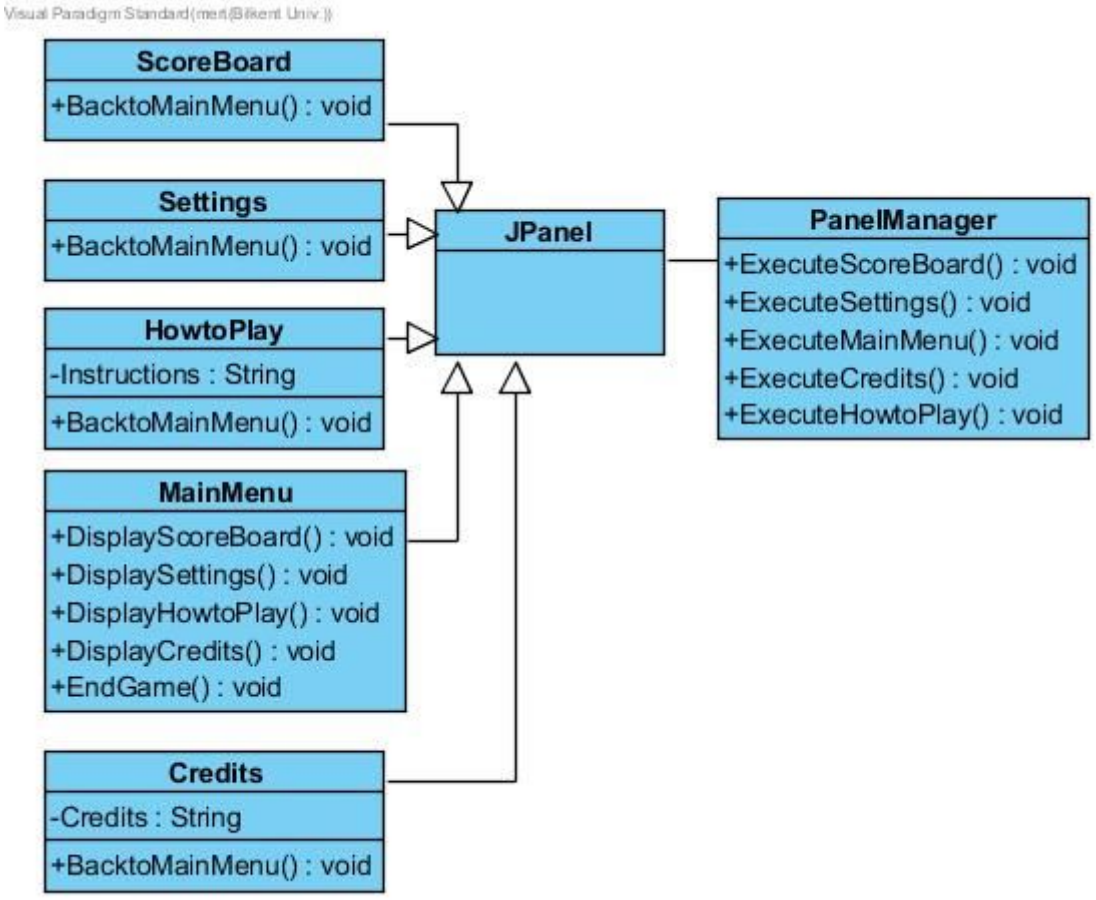
5.3 Object and class model

This diagram shows how the inner workings of the game system are designed. This system we constructed 4 main engine systems that have different responsibilities for their own and their communication is the essential part to game to work.



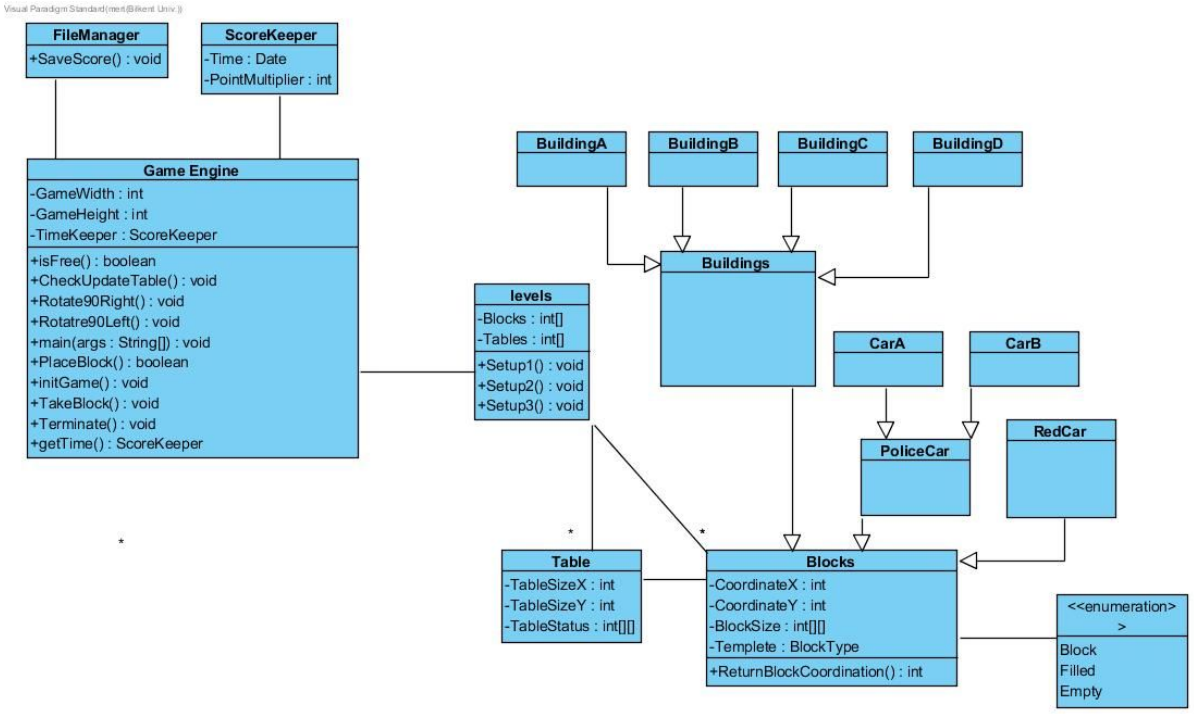
5.3.1 User Interface Component

User reacts and navigates through this component to play the game.



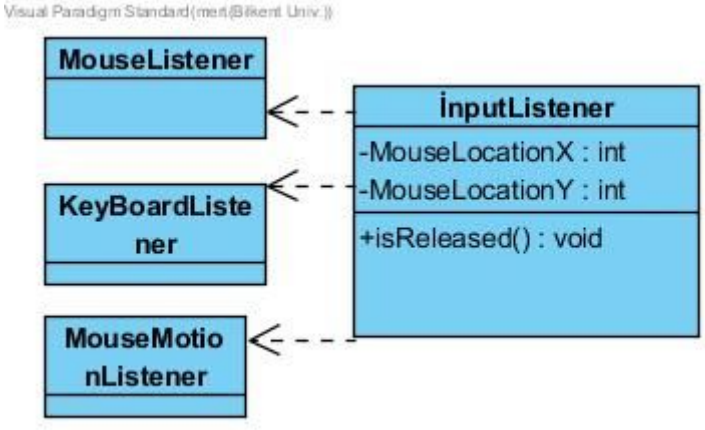
5.3.2 Game Engine Component

The game engine system is the core part of the game which includes information and algorithms for the game to run in the back-end.

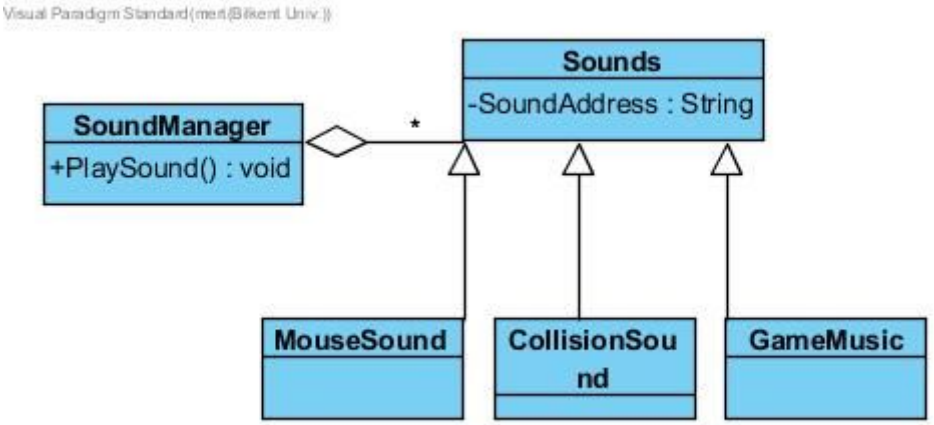


5.3.3 Input Listeners

Input listener takes mouse and keyboard interactions of the user and behaves according to those inputs.

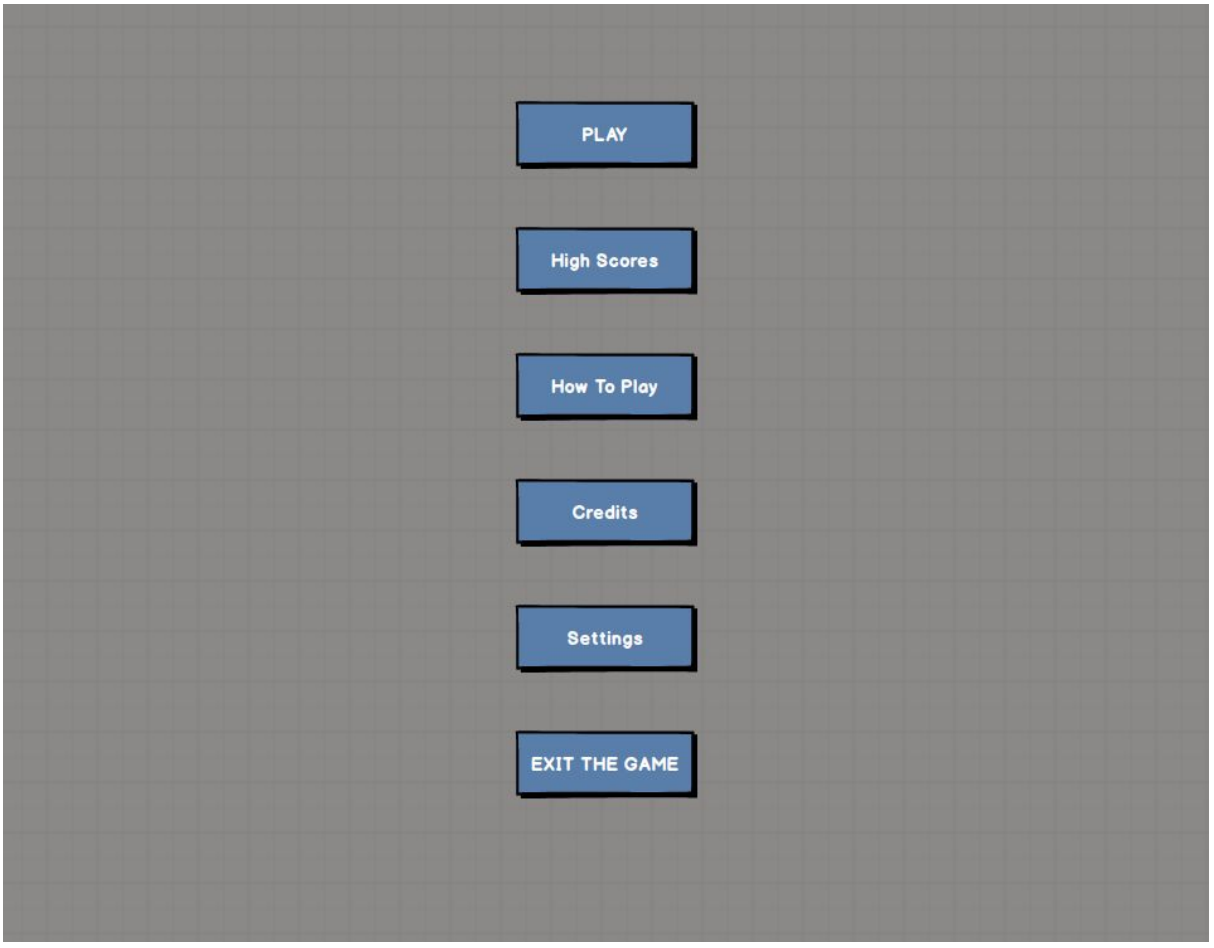


5.3.4 Sound Manager



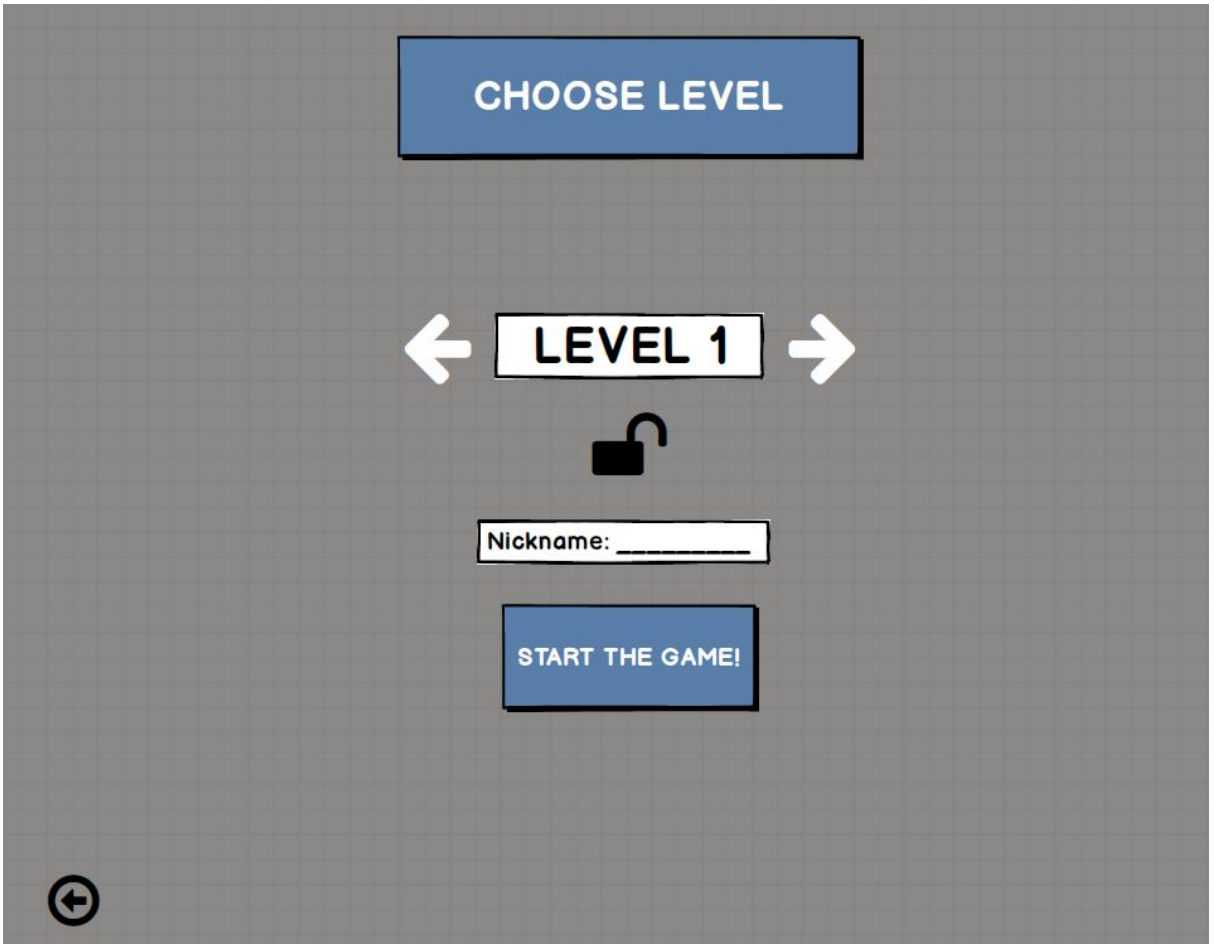
5.4 User Interface Mockups

Main Menu



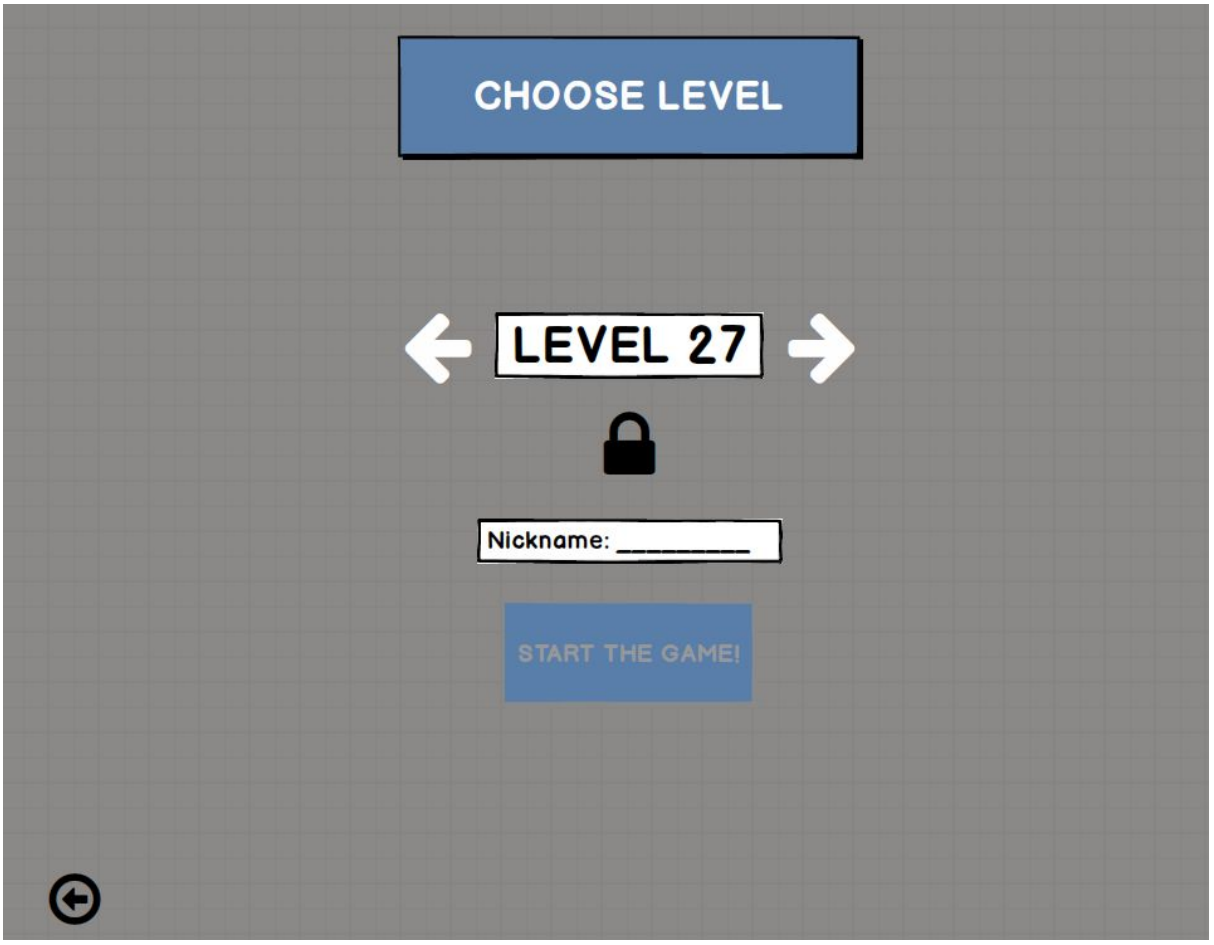
When user clicks the Play button, Choose Level 1 screen is displayed.

Choose Level: 1



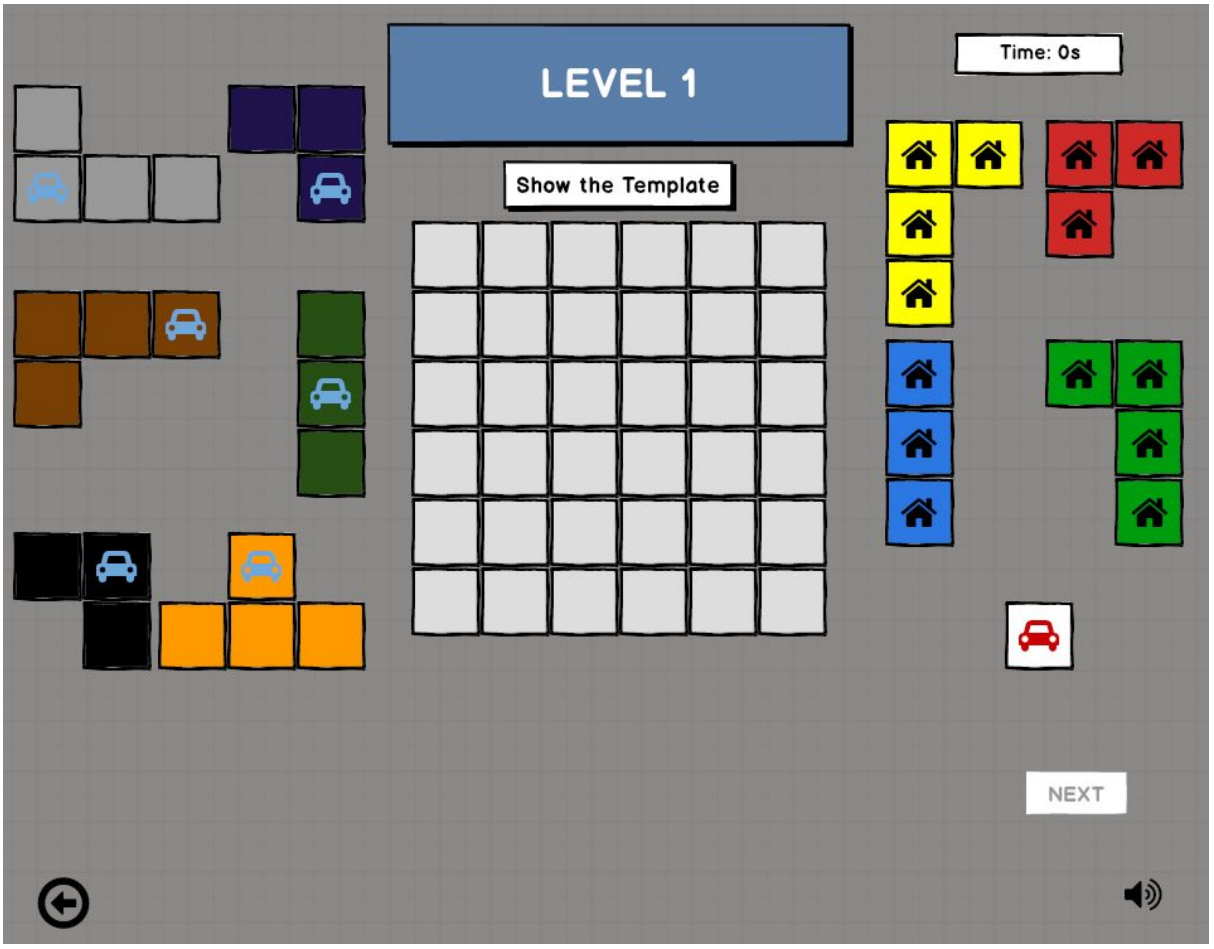
Level 1 is unlocked, user needs to enter his name and press Start the Game button. If the user wants to play another level, he needs to click the side arrows to roll through levels.Level 27 for example:

Choose Level: 27



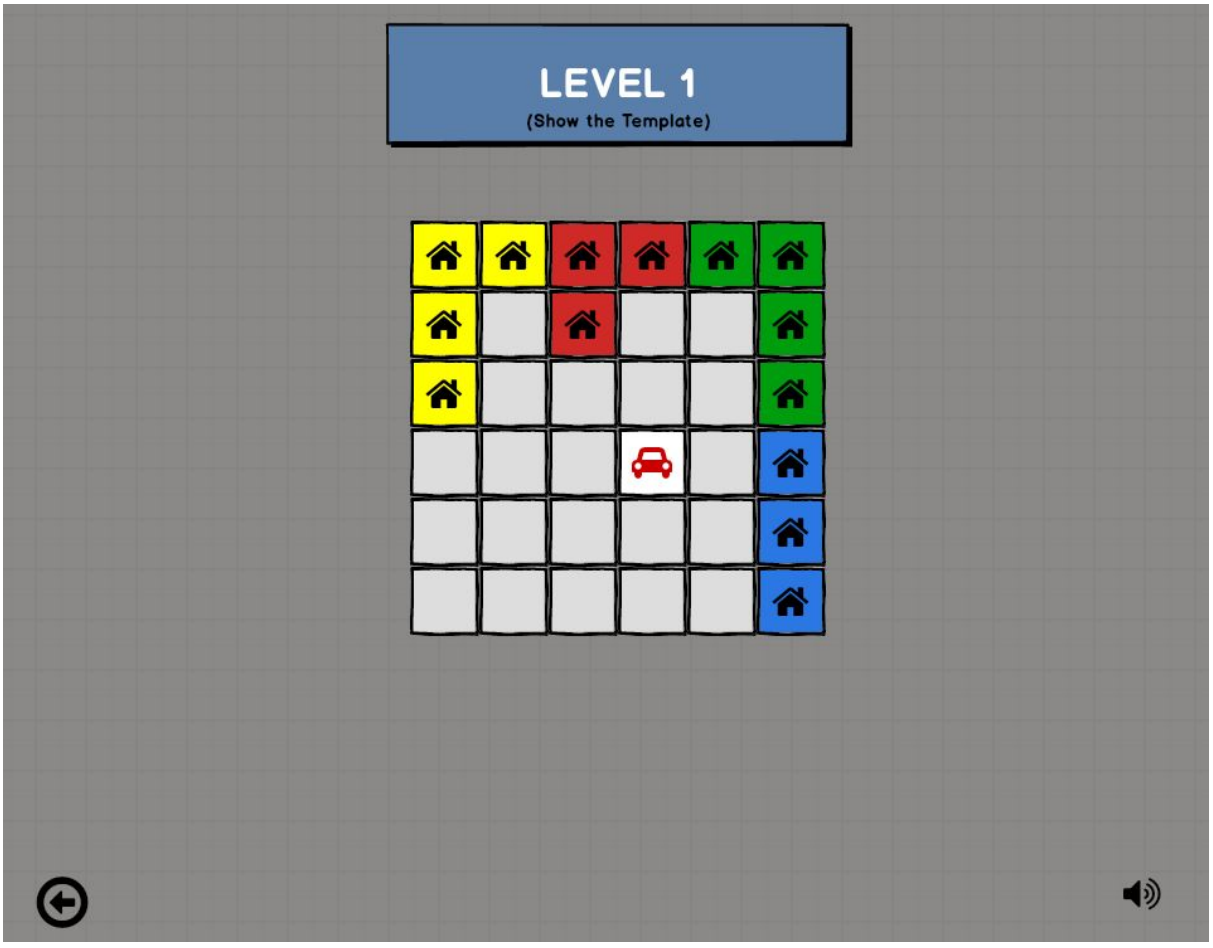
Level 27 is locked because user didn’t unlock the levels needed to unlock the level 27.

Start the Game



This is the beginning of the gameplay screen. When user clicks Show the Template button, the next screen will be displayed.

Show the Template

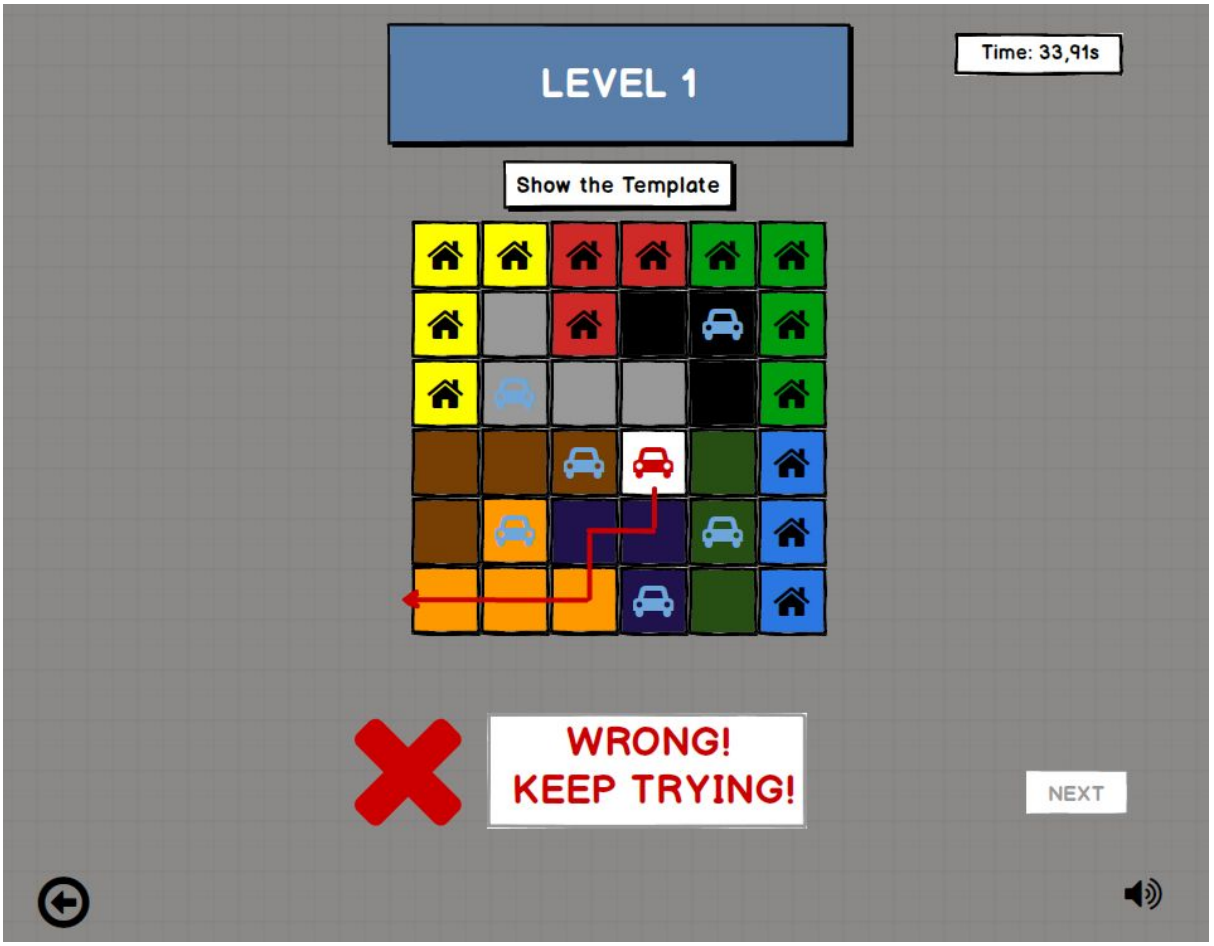


This the screen that shows the template of the game board.

Playing the Game…

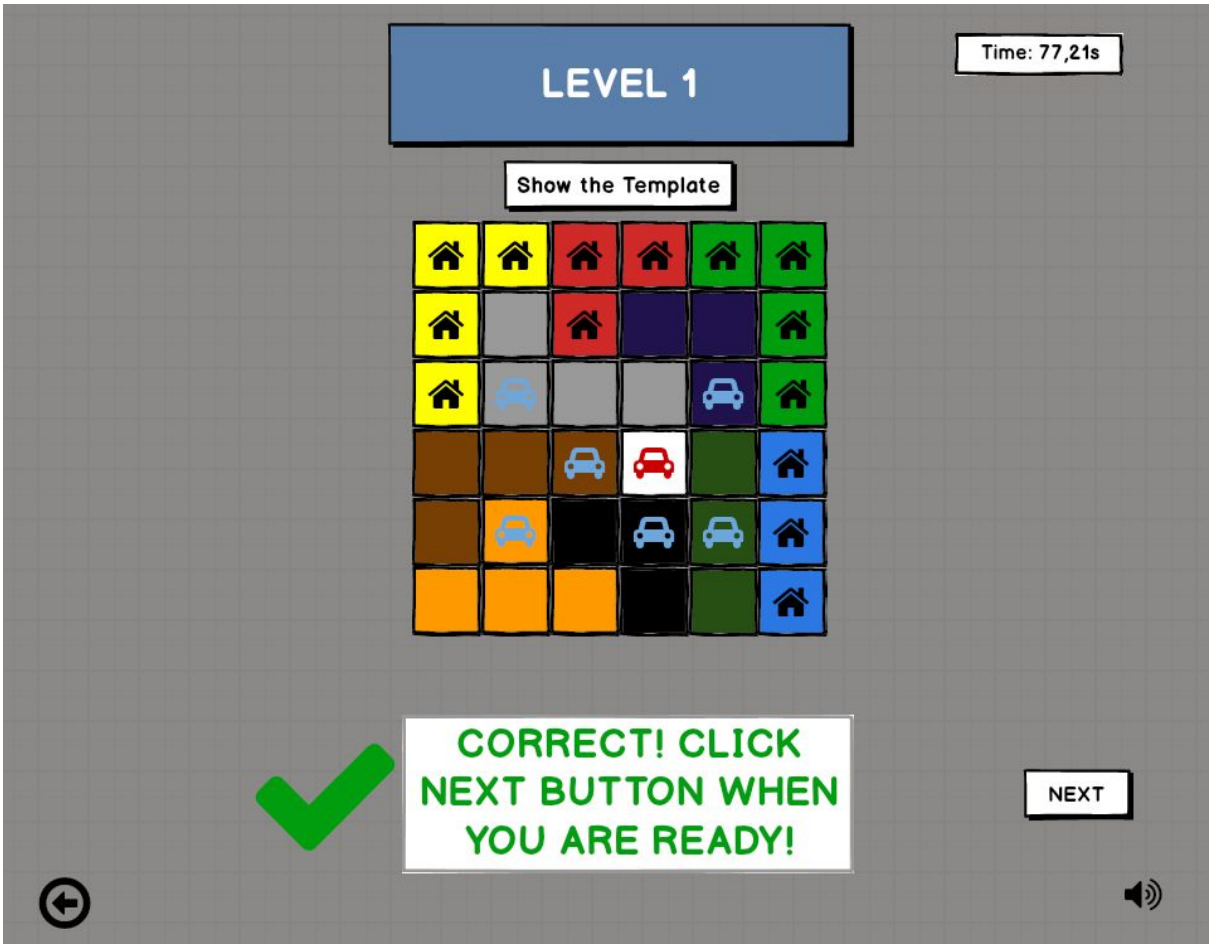
User is playing the game…

User completes the board incorrectly



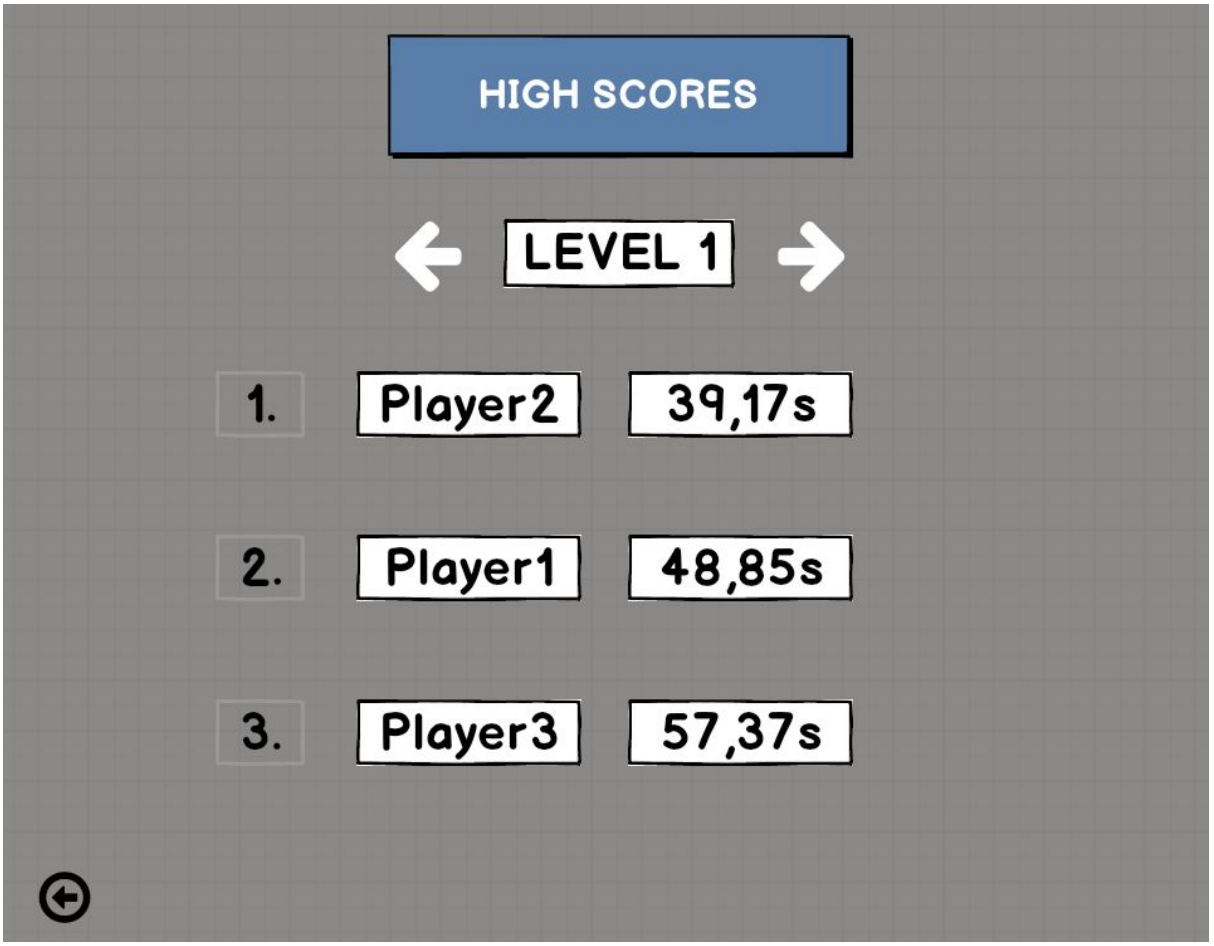
This is an example wrong solution, the user needs to move the blocks out and keep trying or he can go back to main menu.

User completes the game correctly



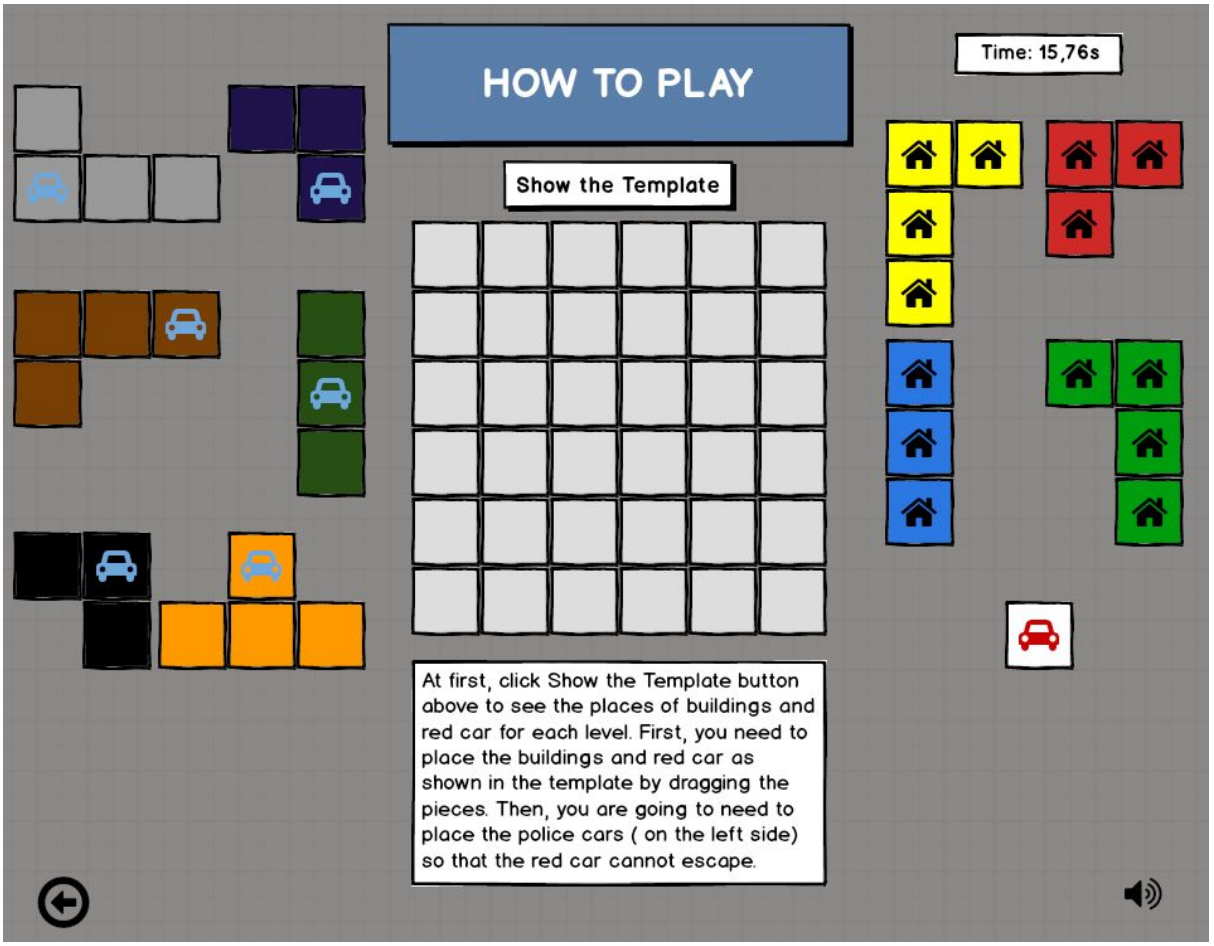
User completes the board correctly and NEXT button is available if the user wants to play the next level.

High Scores



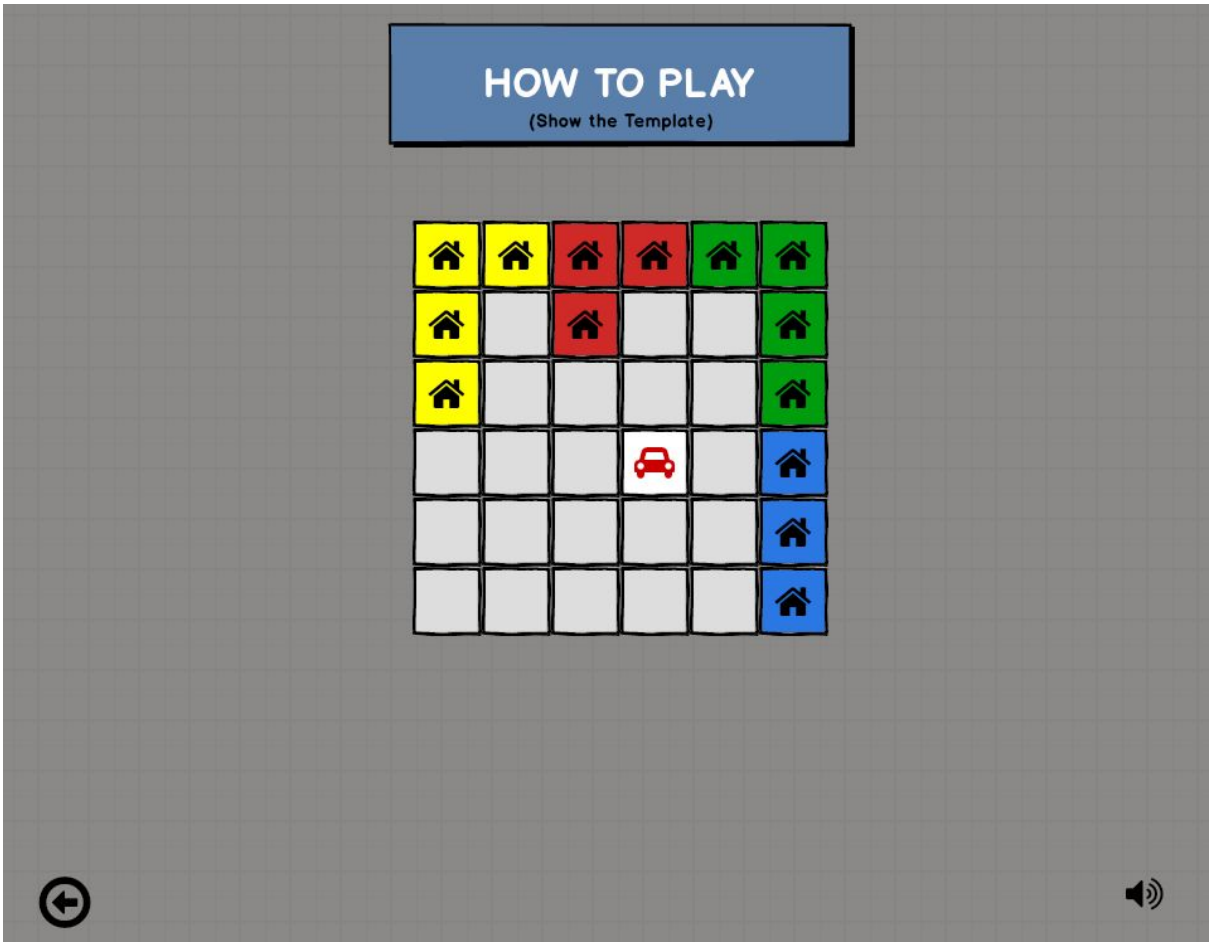
If user clicks the High Scores button from the main menu, top three high scores will be displayed for each level with nicknames.

How to Play



If the user clicks How to Play button from the main menu, how to play screen will appear, with a brief explanation of the gameplay. User can also click Show the Template button to display an example of templates.

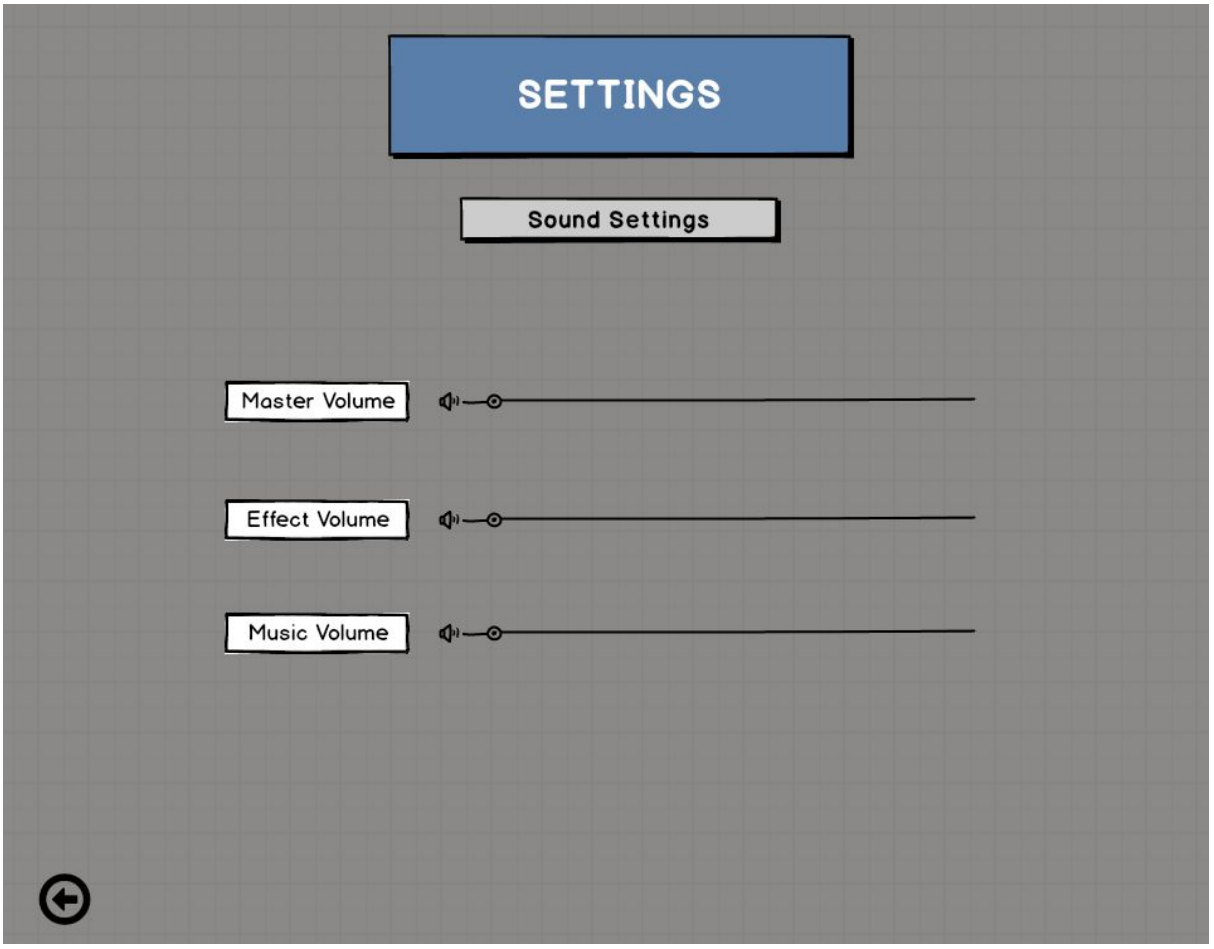
How to Play (Show the Template)



Credits

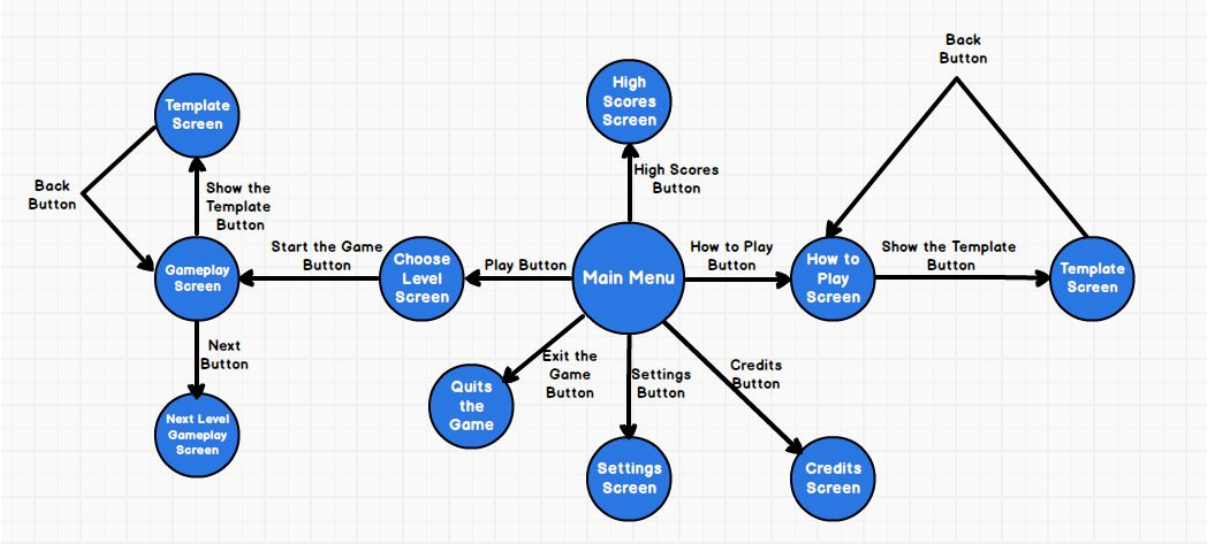
Credits screen will be displayed if the user clicks the Credits button from the main menu.

Settings



Settings menu will be displayed if the user clicks the Settings button from the main menu. User can increase/decrease the sounds from this screen.

Navigational Path



6. References

[1] https://www.smartgames.eu/uk/one-player-games/roadblock