



CS 319 - Object-Oriented Software Engineering Analysis Report

Q-Bitz

Group 1F

Burak Yaşar
Görkem Yılmaz
Hikmet Demir
Mert Duman
Mert Alp Taytak

Table of Contents

1. Introduction	3
2. Overview	4
2.1 Menu	4
2.2 Game Options	4
2.3 Settings	5
2.4 Level Setup	5
2.5 Gameplay	6
3. Functional Requirements	7
3.1 Game Modes	7
3.1.1 Pattern Matching	7
3.1.2 Racing & Rolling (Online Multiplayer Mode)	7
3.1.3 From Memory in 10 Seconds	7
3.1.4 Maximum Pattern Succeed in 2 Minutes	7
3.1.5 Time Constraint Mode	8
3.1.6 Painting Puzzle	8
3.1.7 Different Cubes	8
3.2 Controls	8
4. Nonfunctional Requirements	10
4.1 Game Performance	10
4.2 Non-Complex User Interface	10
4.3 Tile Options (HD Tiles/Shadowing Tiles/Animation Tiles)	10
5. System Models	11
5.1 Activity Diagram	11
5.2 State Diagram	12
5.3 Use Case Diagram	13
5.3.1 Play Online	13
5.3.2 Play Singleplayer	14
5.3.3 Change Board Type	14
5.3.4 Change Game Mode	14
5.3.5 Quit	14
5.4 Class Diagram	15
5.5 Custom Game Sequence Diagram	16
5.6 Select Level Sequence Diagram	18
5.7 Winning Scenario Sequence Diagram	20
5.8 Losing Scenario Sequence Diagram	21
6. Screen Mockups	22
7. Conclusion	27
8. References	27

1. Introduction

We the members of the group 1F chose the game Q-bitz for the term project. This is an appropriate game to implement object-oriented programming principles. We will be implementing our game in Java since we thought that we can best express the object-oriented programming principles with Java and we are all comfortable with it. We are going to use Java Swing library since it will provide us to design User Interface (UI) and graphics. Our main goal by implementing this game is to get used to using the principles that we will learn in CS 319 course and to work with the object-oriented design concepts.

We can define the game as the following. Q-bitz is a magnificent visual agility game that will never lose its fascination. With 80 pattern cards and 16 cubes, players recreate the patterns as quickly as possible. From matching the card shown to having ten seconds to study a card and then remaking the pattern from memory, each variation of these fast-paced rounds requires a different set of visual and cerebral skills. But Q-bitz does not have to be played as a game. The cards and cubes can also be used as an exceptional thinking skill challenge for children or adults with short-term memory loss.

Q-bitz is a multiplayer game about building a pattern. The patterns are chosen randomly from a pile of cards and players try to match the pattern using cubes. Cubes are identical to each other but each face of a cube is different from its other faces. Depending on the specifics of the current round the player who builds the correct pattern fastest or the player who gets closest to the correct pattern wins the round. The player with the most rounds wins the game.

Our aim in this project is to make a Q-bitz game that has the same rules as the tabletop version. However, we are going to focus on user experience, features and try to polish the end product as our attempt at making our project something more than the tabletop version.

2. Overview

2.1 Menu

Main menu consists of several buttons. Functions of these buttons are explained in the following,

- Play: Leads to a game options menu. After desired options are selected the player can play the game in single player.
- Select Level: Leads to a campaign mode where player can select from pre-designed levels. For the levels, default game options and game mode are used. The player success is measured and given a rating upon level completion.
- Multiplayer: Leads to a game options menu. After desired options are selected the player can play the game in multiplayer with other players.
- How to Play: Opens the screen where game controls are shown and an explanation about how to play the game is written.
- Settings: Opens the screen where settings that are not directly related to the gameplay such as board size and game mode are displayed.
- Credits: Opens the screen where developers of this project are displayed.
- Exit: Closes the game.

2.2 Game Options

In the game options menu there are three settings,

- Board Size: The player can select to play the game in 3x3, 4x4 (default) or 5x5 cubes sized boards.
- Number of Players: The player can select to play the game with 1, 2, 3 or 4 players.

Where 1 player is single player and the rest are multiplayer.

- Game Mode: The player can select from a list of game modes. These game modes are explained in more detail in functional requirements.

2.3 Settings

The list of settings are as follows,

- Window Options: The player can select window size.
- Sound Options: The player can select the turn on/off the sound and turn on/off the background music.

2.4 Level Setup

The game is played in rounds. For each round there is a level. A level consists of players, cards, cubes, trays and boards. Following is the description for the previous level parts:

- Players: In each level there are 1 to 4 players. All of these are humans, there are no computer controlled players. Each player is given a board, a tray and cubes.
- Cards: The game is played by matching a pattern. The patterns are printed on cards. In each level, depending on the game mode, there can be many cards.
- Cubes: The players match the patterns on the cards using cubes, each face of a cube has a different pattern painted on it. Every player has a set of cubes. The amount of cubes depends on the board size settings. Also depending on the settings each cube in a set can be identical or have faces different from one another.
- Trays: Trays are where a set of cubes are stored before they are used. Each player has a tray.
- Boards: Boards are where players place the cubes to match the pattern given. Depending on the game options board size can be 3x3, 4x4 or 5x5 cubes. This will let the

more advanced players challenge themselves with the 5x5 board and let the beginners get used to the game playing in the 3x3 board.

2.5 Gameplay

Gameplay takes place between level start and level end. There will be multiple game modes implemented. Each game mode is different from each other gameplay and goals wise. However, the fundamentals do not change from game mode to game mode. There will be more information about game modes in the requirements section.

The fundamental gameplay consists of constraints, rounds, controls and goals. In each game mode there are different types of constraints. For example, in some modes patterns are visible for the whole round and in some modes they are hidden after a certain amount of time passes. The game is played in rounds and each round has one or more winners. After multiple rounds the player with the most points wins. Controls work as explained in the relevant functional requirements section. The goals also depend on the game mode. In some game modes finishing the pattern in time is enough but in other game modes the player has to be the first one to finish the pattern to win the round.

3. Functional Requirements

3.1 Game Modes

3.1.1 Pattern Matching

Mode one asks players to use their set of cubes to match the pattern shown on the card. The first one to complete the pattern wins the card.

3.1.2 Racing & Rolling (Online Multiplayer Mode)

Mode two asks the players to roll the cubes like dice and use as many of the cubes as possible to duplicate the pattern shown on the new card. Race to re-roll the cubes until the pattern is complete. First one done wins the card. (We have found it more exciting to re-roll each turn together instead of doing a free-for-all.)

3.1.3 From Memory in 10 Seconds

In mode three, players have 10 seconds to study a new card then remake the pattern from memory. This can be unbelievably difficult. The one who finishes first, or the one who has the most blocks in the correct position, wins the card. At the end of these modes, the player with the most cards is the winner.

3.1.4 Maximum Pattern Succeed in 2 Minutes

Finally, in round four, player race against time which is given 2 minutes. Player should achieve the maximum number of patterns as best he/she can. In this mode, the player with the most cards is the winner as well.

3.1.5 Time Constraint Mode

In this mode, both players will have a time constraint on them and they will try to finish within this time limit. The players will earn points based on how much spare time they have left when they finish.

3.1.6 Painting Puzzle

The original game has 6 different patterns for each face of the cube. We will add to this and add some original cubes as well. For example, the players will try to complete a painting that has its pieces on each face of the cubes.

3.1.7 Different Cubes

In this mode, different cubes will have different patterns as opposed to the original game, where each cube has the same patterns on each face. This will add the challenge of finding the correct cube on top of finding the correct face.

3.2 Controls

The game we are implementing is dexterity based in real life. Because of that, the controls are very important to give the user a satisfactory experience. Our implementation uses mouse and keyboard. The mouse is used to pick up cubes from the tray and place them on the board. The keyboard is used to rotate the cubes to reveal other faces. To make it easier for the user, when a cube is picked up a small window will open to show all of the faces of that cube.

The mouse will also be used for camera controls. Spinning the mouse wheel will control zoom level, pressing right click and dragging will control camera rotation and pressing and dragging the third mouse button, the mouse wheel, will translate the camera.

Finally, the keyboard will have various shortcuts. Along with previously mentioned cube controls, there will be sound controls, shortcuts for switching between the views we implement and more controls as we need in the future.

4. Nonfunctional Requirements

4.1 Game Performance

We are planning to add some animations into our game and we are going to make sure that these animations will not cause any performance issues in the gameplay. The standard FPS rate is usually 60 in most computers and we are planning to pass this limit or at least stay in that limit.

4.2 Non-Complex User Interface

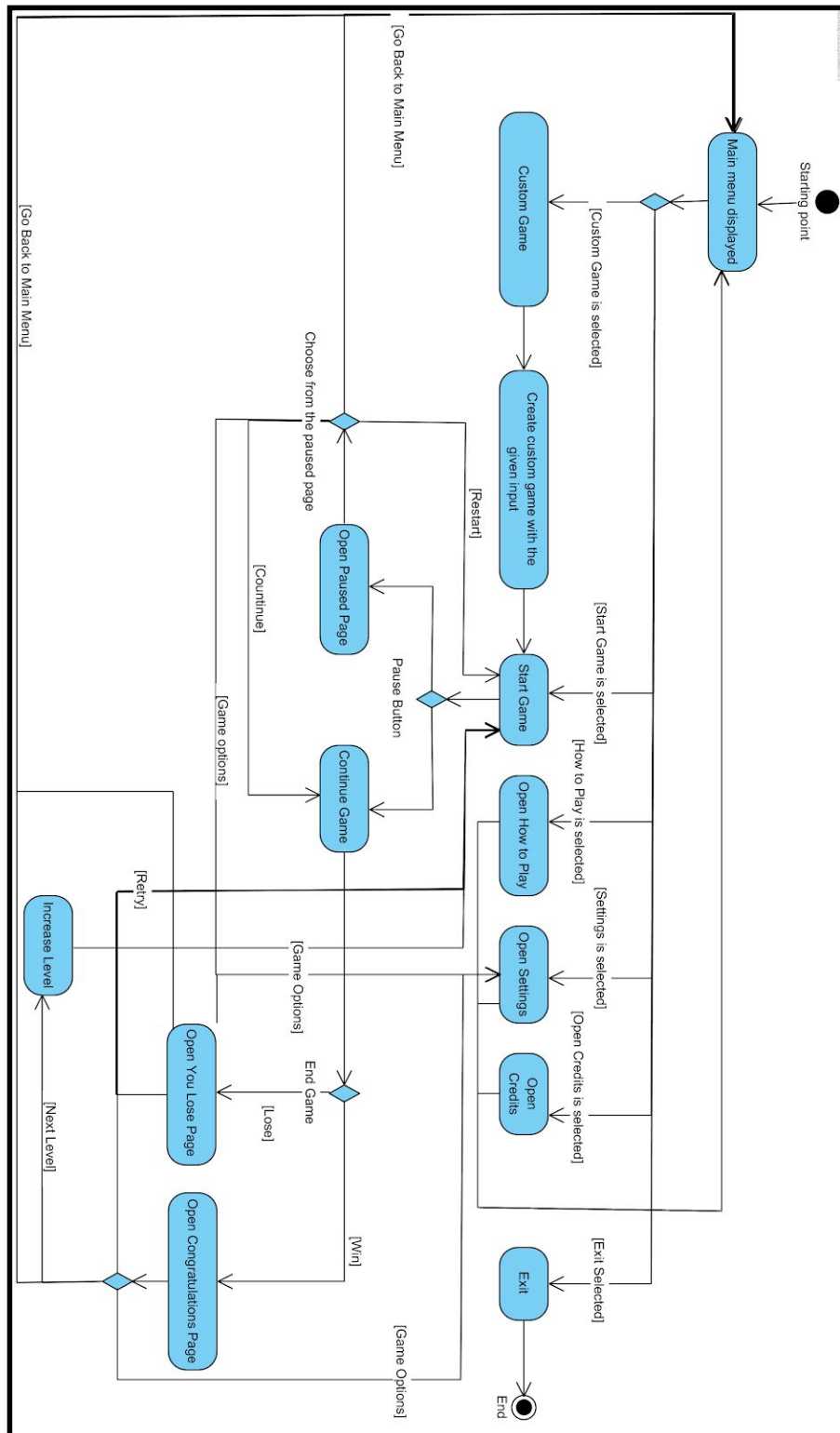
A complex user interface could easily make a player feel uncomfortable and confused. Therefore we are planning to create a user friendly user interface which is easy to understand. There will not be lots of confusing buttons everywhere and the mechanics will be smooth and flexible.

4.3 Tile Options (HD Tiles/Shadowing Tiles/Animation Tiles)

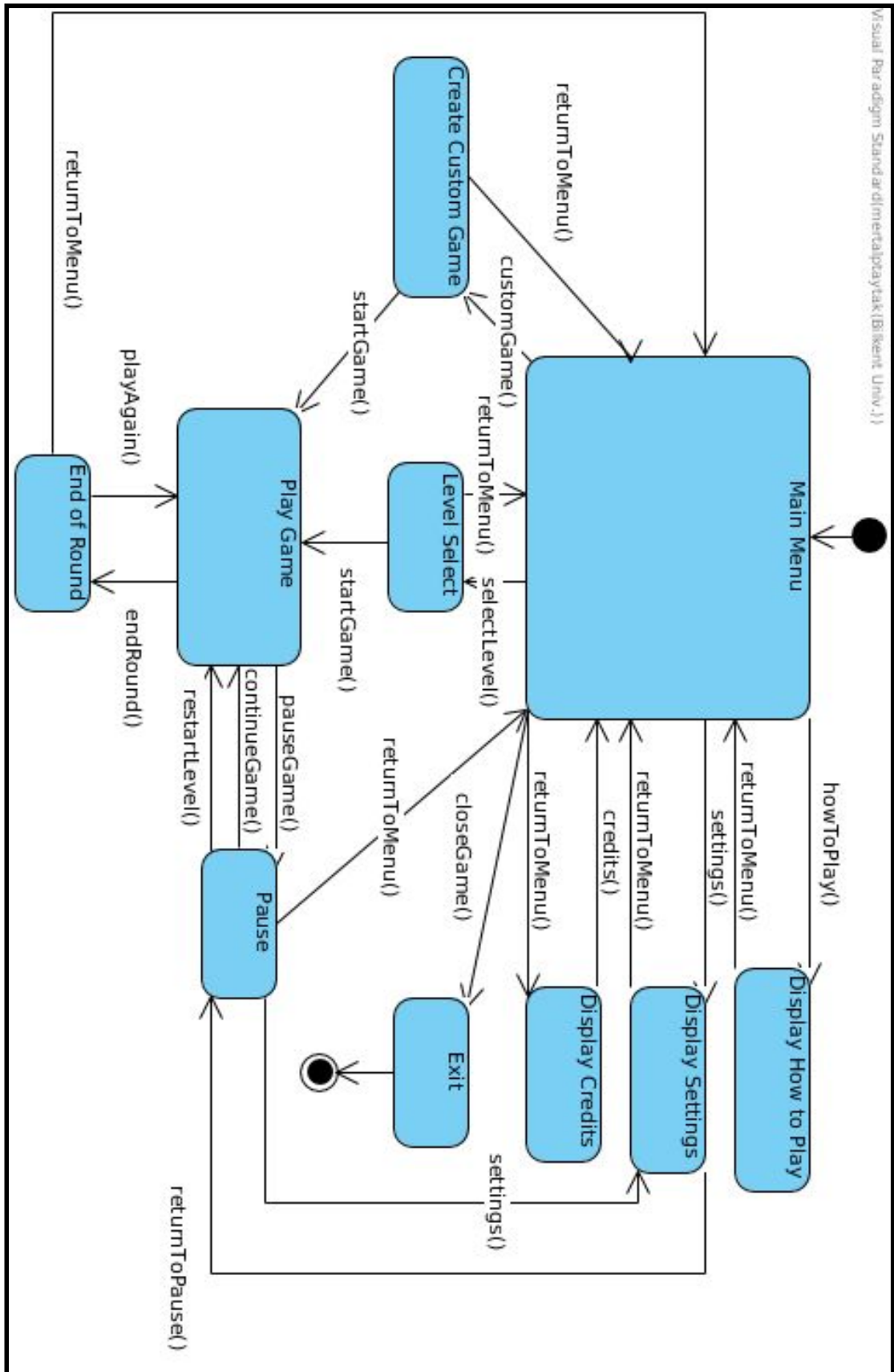
In our game we are thinking of using high-quality images to create game pieces and for backgrounds. The animation tiles and the shadowing tiles that we are planning to use will make this game more than a simple board game in the eye of the player.

5. System Models

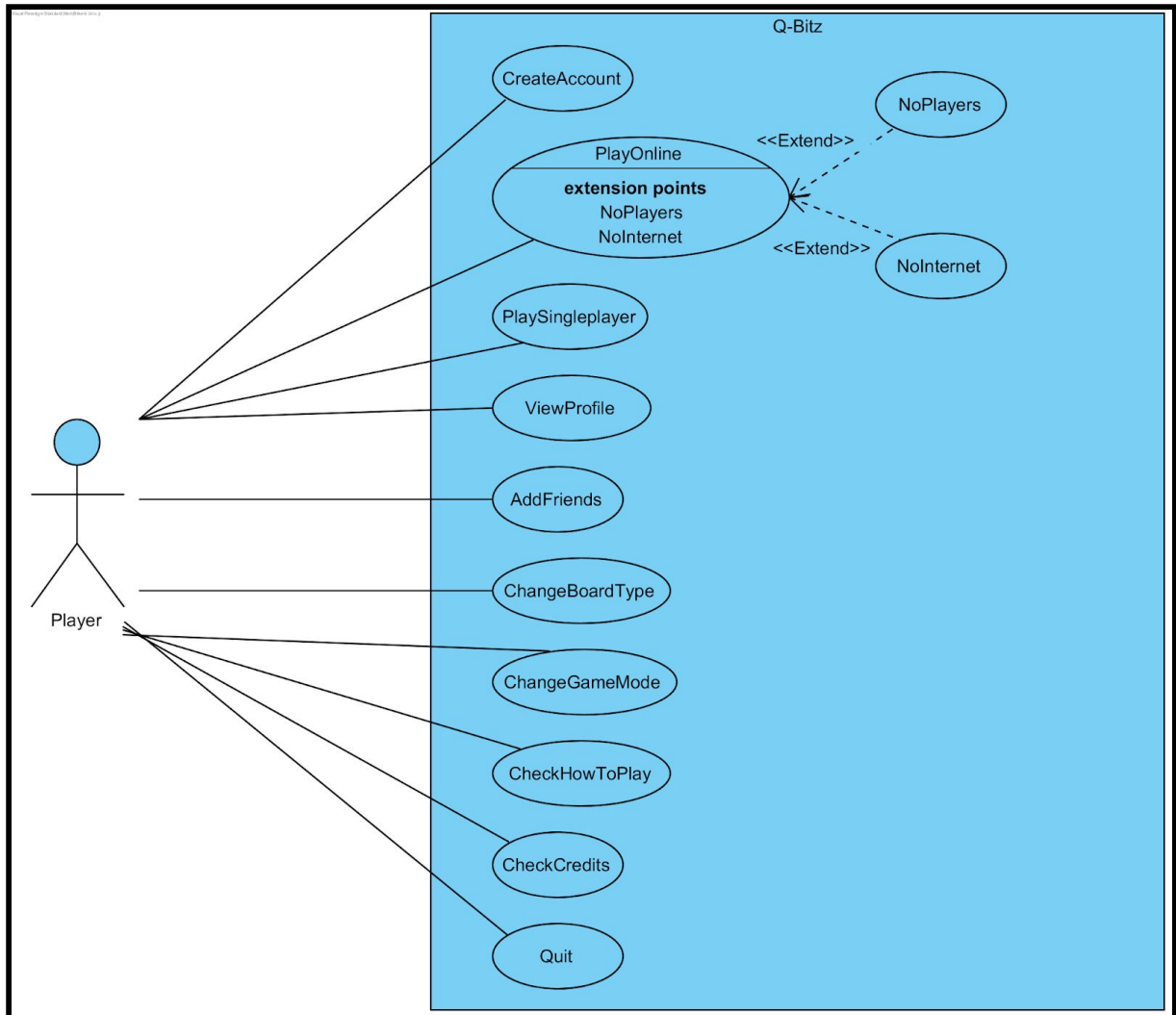
5.1 Activity Diagram



5.2 State Diagram



5.3 Use Case Diagram



5.3.1 Play Online

Use Case Name:	PlayOnline
Participating Actor:	Player
Entry Condition:	Player has internet connection. "Multiplayer" button is pressed.
Exit Condition:	Game is over. Exit button is selected.
Flow of Events:	Player connects to the internet. Player presses on multiplayer from the menu. Player selects the desired game mode and board type. The game starts. The game ends or the player quits the game.

5.3.2 Play Singleplayer

Use Case Name:	PlaySingleplayer
Participating Actor:	Player
Entry Condition:	“Play” or “Select Level” button is pressed.
Exit Condition:	Game is over. Exit button is pressed.
Flow of Events:	Player presses on “Play” or “Select Level”. Player chooses the game options if “Play” is selected. The game starts. The game ends or the player quits the game.

5.3.3 Change Board Type

Use Case Name:	ChangeBoardType
Participating Actor:	Player
Entry Condition:	“Game Options” button is clicked.
Exit Condition:	“Back to Menu” button is clicked within in Game Options.
Flow of Events:	Player clicks on the “Game Options” button. Game options screen is displayed. Player chooses the desired board type. Player clicks on the “Back to Menu” button.

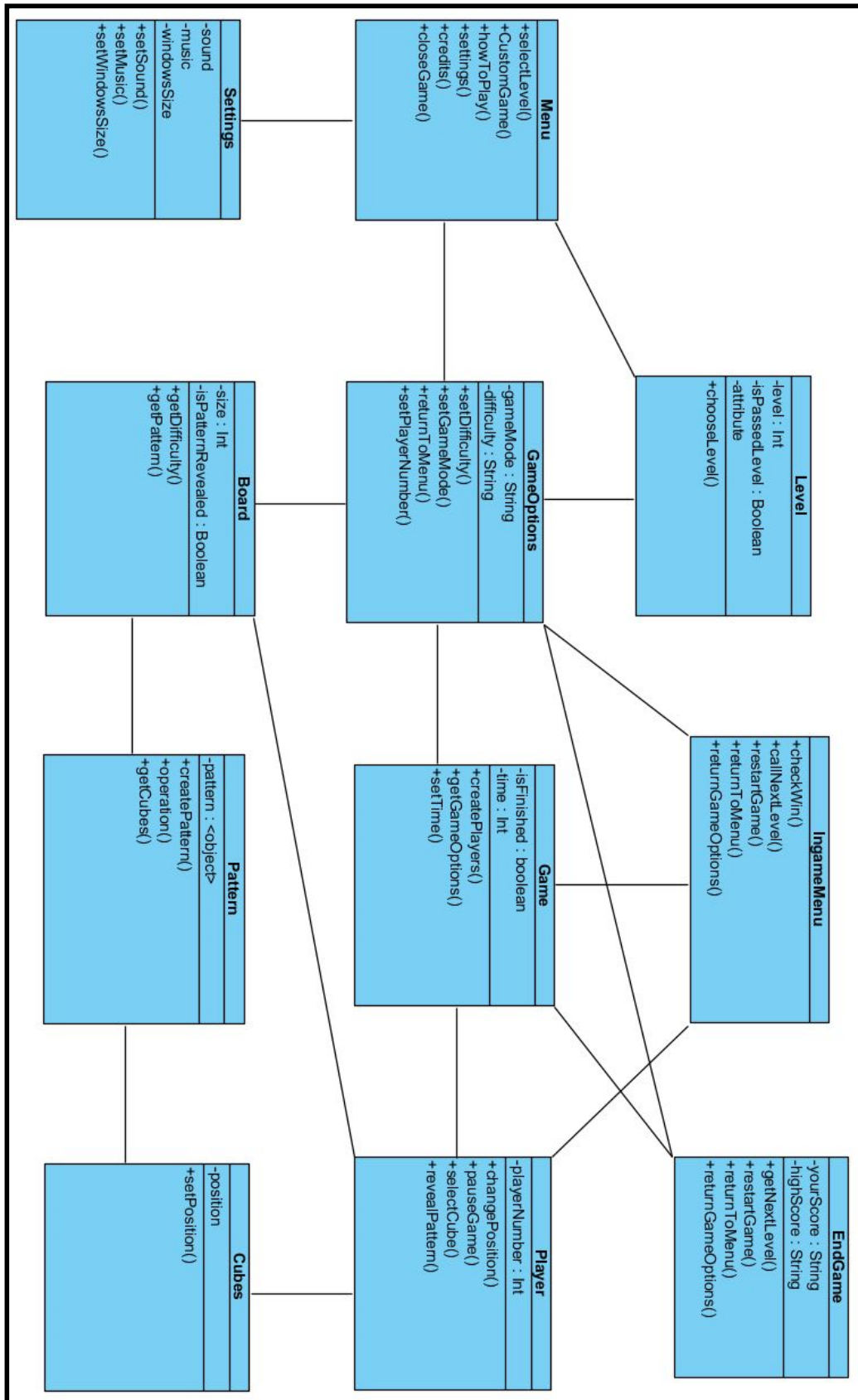
5.3.4 Change Game Mode

Use Case Name:	ChangeGameMode
Participating Actor:	Player
Entry Condition:	“Game Options” button is clicked.
Exit Condition:	“Back to Menu” button is clicked within in Game Options.
Flow of Events:	Player clicks on the “Game Options” button. Game options screen is displayed. Player chooses the desired game mode. Player clicks on the “Back to Menu” button.

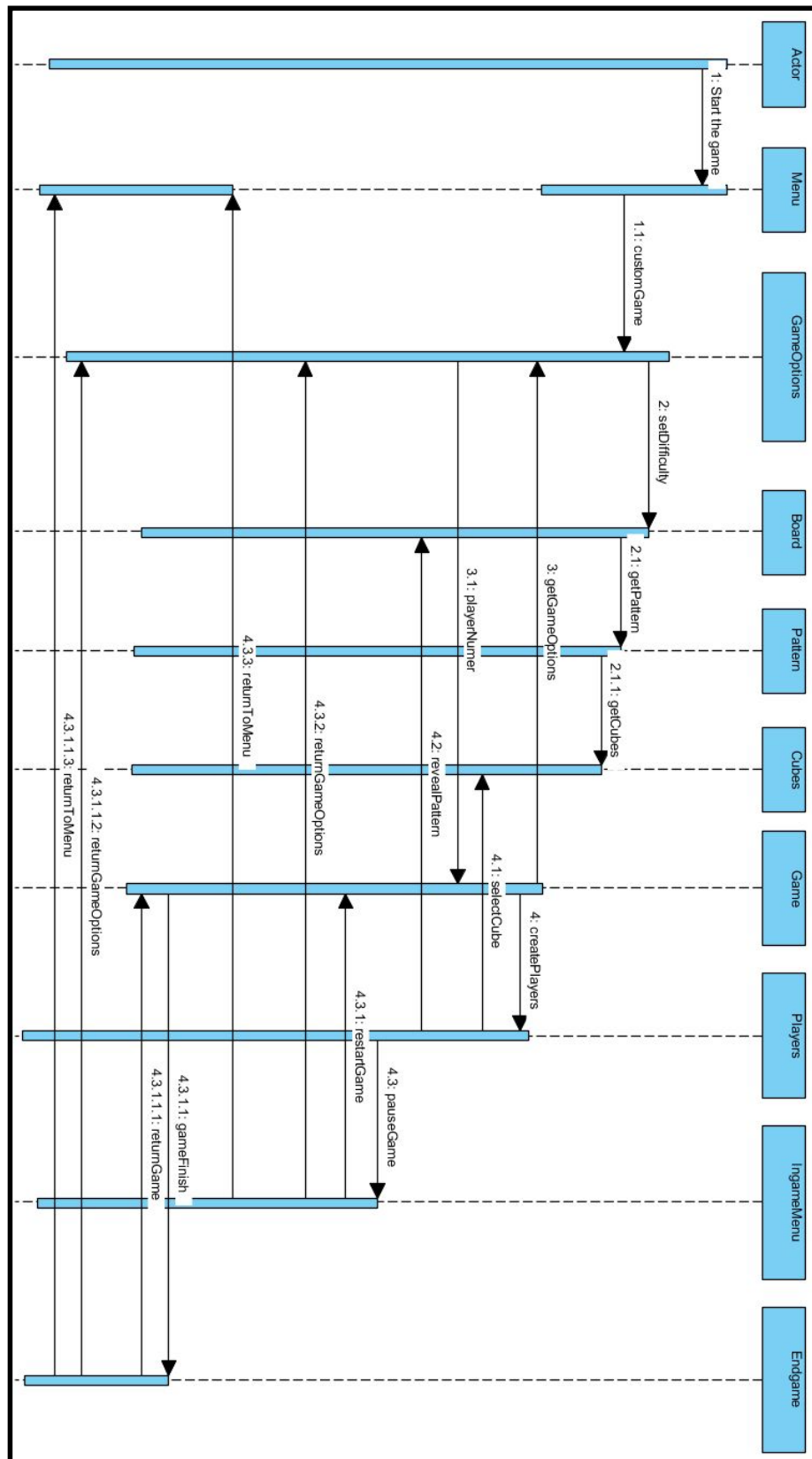
5.3.5 Quit

Use Case Name:	Quit
Participating Actor:	Player
Entry Condition:	“Exit” button is clicked from the menu.
Exit Condition:	The game is closed.
Flow of Events:	Player clicks on the “Exit” button from the menu. The game is closed.

5.4 Class Diagram



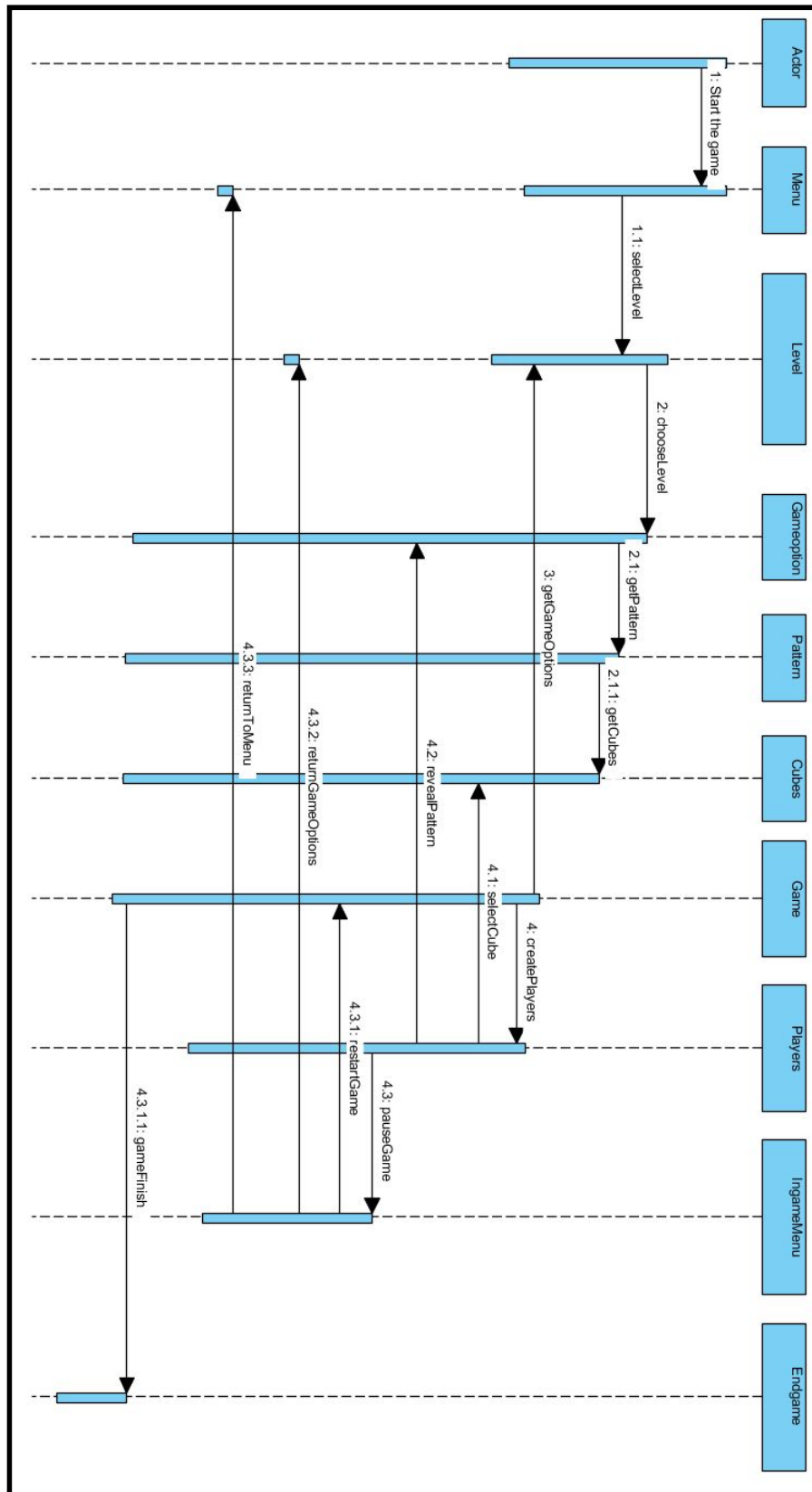
5.5 Custom Game Sequence Diagram



Scenario: User starts the game

Player enters the game and encounters with the main menu. When the player press the 'Play' button, game options menu pops up where player customize the game as he/she wants. Depending on which difficulty he/she selects, the board is set up and the pattern is loaded according to board size. Cubes are given to the user with the pattern as well then the game starts. After the game starts, the player can select one cube at a time and turn it in order to fit it into the pattern. Also, the player can pause the game and encounters with pause screen where he/she can turn back to the game options menu to change the difficulty of the game or turn back to the main menu. The player may restart the same game as well.

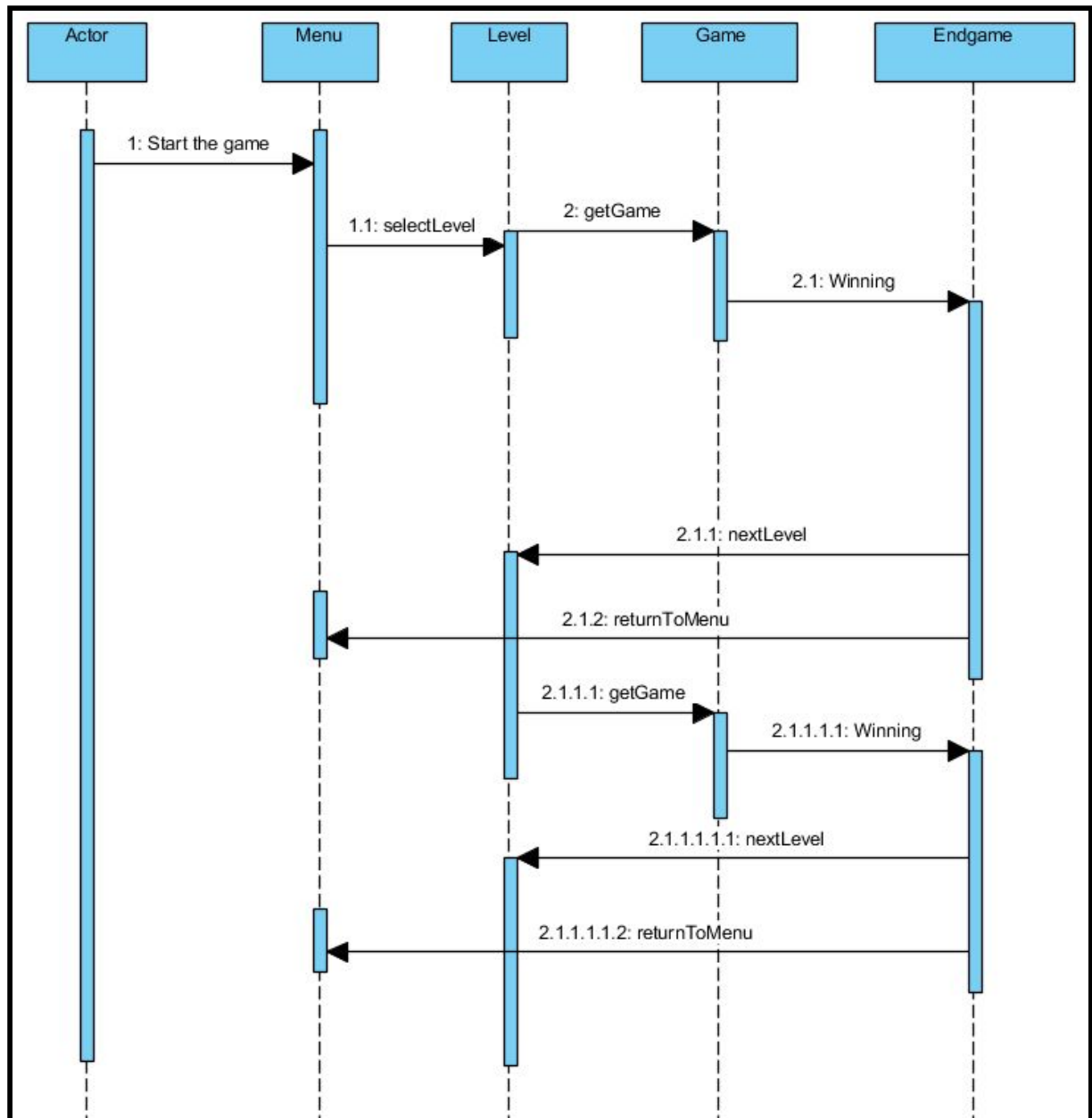
5.6 Select Level Sequence Diagram



Scenario: User selects the level

Player enters the game and encounters with the main menu. When the player press the 'Select Level' button, level selection menu pops up where player can choose which level he/she wants to play. Depending on which level he/she selects, the board is set up, cubes are given to the user and game starts. After the game starts, player can select one cube at a time and turn it in order to fit it into the pattern. Also, player can pause the game and encounters with pause screen where he/she can turn back to the game options menu to change the level of the game or turn back to the main menu. The player may restart the same game level as well.

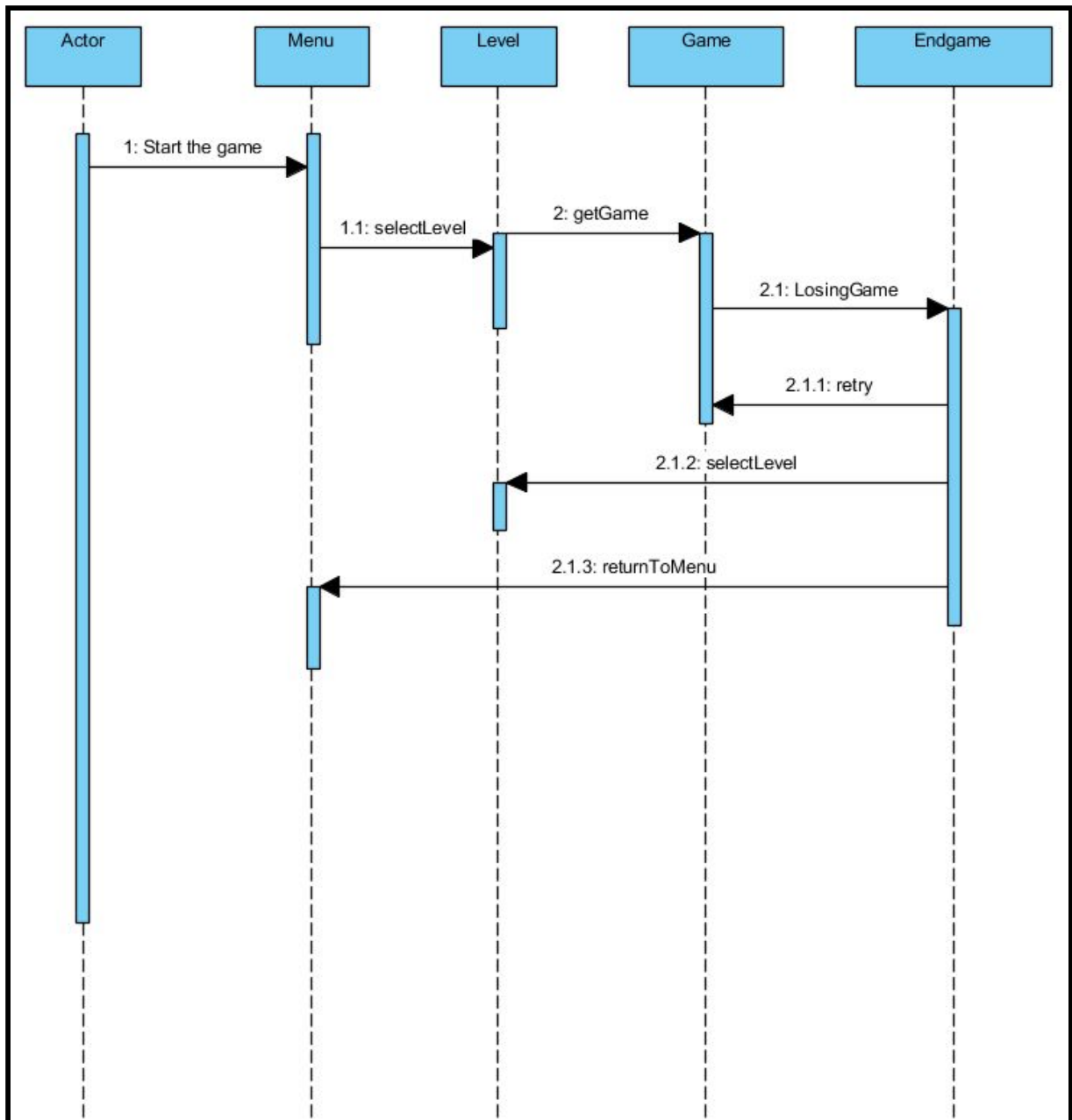
5.7 Winning Scenario Sequence Diagram



Scenario: Plays wins the game

Player chooses select level option from main menu and select one level which is not locked from the levels and the game starts with default features (4x4 grid). In this case player wins the game and the game is over, congratulations pop up screen appears and the player can go to the next level or can go back to main menu.

5.8 Losing Scenario Sequence Diagram



Scenario: Plays loses the game

Player chooses select level option from main menu and select one level which is not locked from the levels and the game starts with default features (4x4 grid). In this case player loses the game and the game is over, you lose pop up screen appears and the player can retry the same level or can go back to main menu.

6. Screen Mockups

Q-Bitz

Play

Select Level

Multiplayer

How to Play

Settings

Credits

Exit

Game Options

Difficulty

3 x 3

4 x 4

5 x 5

Player Count

1

2

3

4

Game Mode

Pattern Matching

▼

Back to Menu

LEVELS

1

2

3

4

5

SETTINGS

Music ☒

Sound ☒

Window Size ▼

Back to Menu

Developer Team

Burak Yaşar
Görkem Yılmaz
Hikmet Demir
Mert Duman
Mert Alp Taytak

Back to Menu

Game Options

Difficulty

3 x 3

4 x 4

5 x 5

Player Count

1

2

3

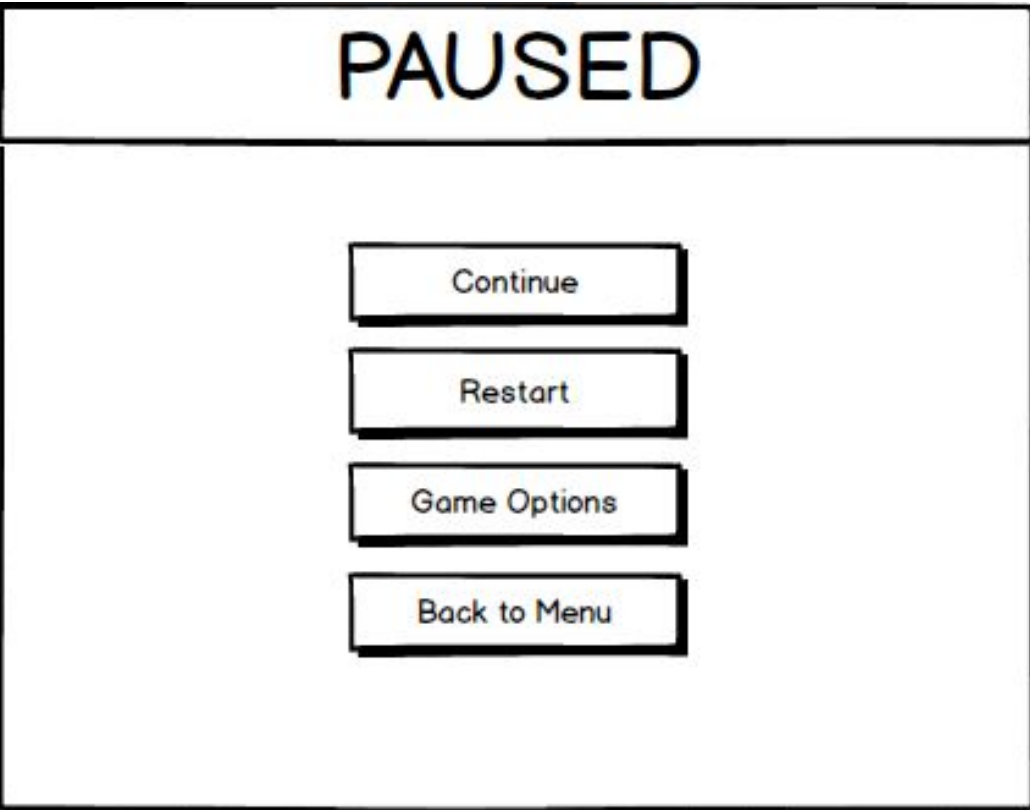
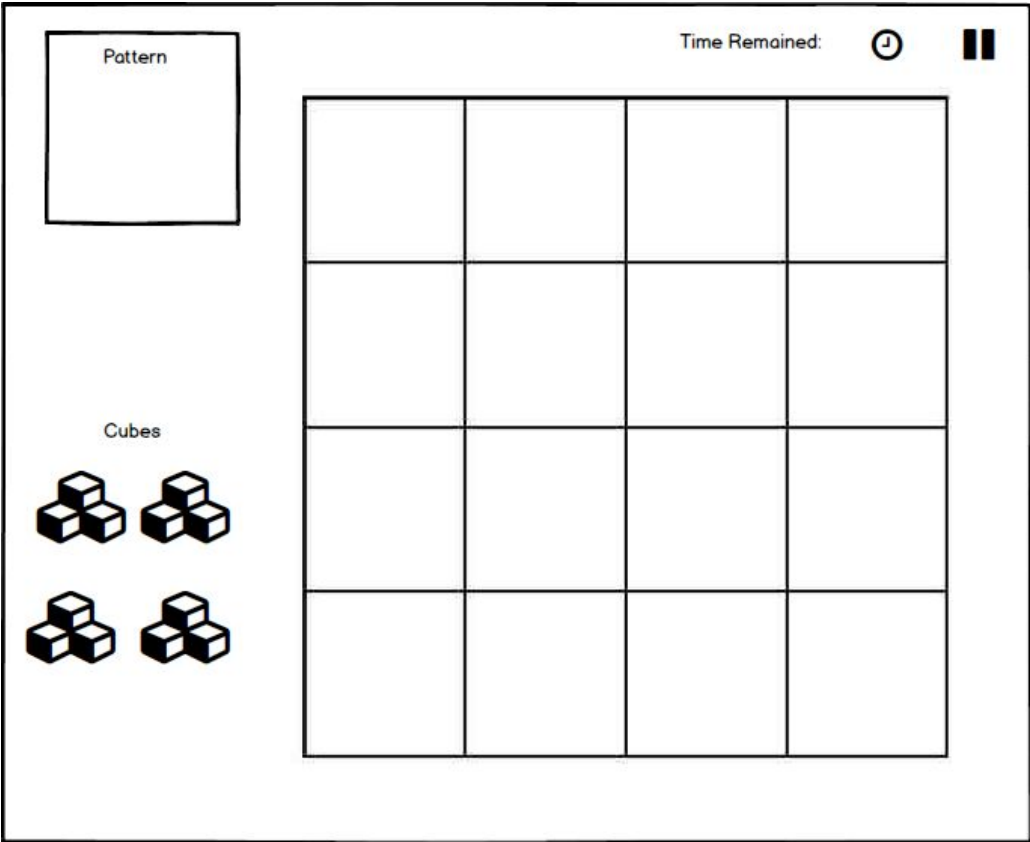
4

Game Mode

Pattern Matching



Back to Menu



YOU LOSE

Your Score :

Retry

Game Options

Back to Menu

CONGRATULATIONS

Your Score :

Highest Score :

Next Level

Retry

Game Options

Back to Menu

7. Conclusion

In this report, the detailed analysis of how we will implement and design our game “Q-Bitz” have been reported. We have presented our report into 3 main sections. In the first part, we have provided an overview to explain the game features. After that, the requirements of this project including 2 subsections functional and non-functional have been discussed. Generally, what will be implemented is discussed in functional requirements part while how they will be implemented is discussed in non-functional requirements. Furthermore, we have drawn stated some system models for our game. The game has 5 different type of diagrams (models) which are the use case, sequence, state, activity and class diagram. We have stated these diagrams along with their purposes for being used.

To sum up, the main purpose of this analysis report is to represent the game providing better visualization so that both the design and the implementation will be understandable expressed better for potential customers and us to use these for future reports.

8. References

“Q-Bitz.” *Timberdoodle Co*, <https://timberdoodle.com/products/q-bitz>.

“Q-Bitz.” *Amazon.co.uk: Toys & Games*, 7 May 2010, <https://www.amazon.co.uk/Green-Board-Games-44002-Q-Bitz/dp/B0031P91LK>.

“Q-Bitz Jr.” *TOYTAG*, <https://www.toytag.com/products/q-bitz-jr>.

TimeToPlayMag. “Q-Bitz from MindWare.” *YouTube*, YouTube, 3 Sept. 2013, <https://www.youtube.com/watch?v=j4PEAbkT780>.