



Bilkent University

Department of Computer Engineering

CS319-OBJECT ORIENTED SOFTWARE ENGINEERING

Group 1B: Q-Bitz

Iteration 1 - Analysis Report

Sait Aktürk, Zafer Tan Çankırı, Berkin İnan, Halil Şahiner, Abdullah Talayhan

Supervisor: Eray Tüzün

21.10.2018

Contents

| | |
|--|-----------|
| 1. Introduction | 1 |
| 2. Overview | 1 |
| 2.1 Game Modes | 2 |
| 2.2 In-Game Objects | 2 |
| 2.3 Information and Control | 3 |
| 3. Functional Requirements | 3 |
| 3.1 User related requirements | 3 |
| 3.2 Navigation | 3 |
| 3.3 Game Funcitonalties | 4 |
| 4. Nonfunctional Requirements | 5 |
| 4.1 Usability | 5 |
| 4.2 Reliability | 5 |
| 4.3 Performance | 5 |
| 4.4 Extendibility | 5 |
| 5. System Models | 6 |
| 5.1 Use case model | 6 |
| 5.2 Dynamics models | 13 |
| 5.2.1 Sequence Diagrams | 13 |
| 5.2.2 Activity Diagram | 18 |
| 5.2.3 State Diagrams | 19 |
| 5.3 Object and class model | 20 |
| 5.4 User interface | 23 |
| 5.4.1 Login | 23 |
| 5.4.2 Register | 23 |
| 5.4.3 Main Menu | 24 |
| 5.4.4 User Settings | 24 |
| 5.4.5 Room Selection in Multiplayer Mode | 25 |
| 5.4.6 Room Creation in Multiplayer Mode | 25 |
| 5.4.7 Game Lobby | 26 |
| 5.4.8 Game Screen | 26 |
| 5.4.9 Post-Game Screen | 27 |
| 5.4.10 Singleplayer Stage Selection Screen | 27 |
| 5.4.11 How to Play | 28 |
| 5.4.12 Options | 28 |
| 6. References | 29 |

Analysis Report

1. Introduction

In Q-bitz players use their special cubes to recreate patterns on the cards, gain points and win the game. Although the original game has three rounds which will be played in order to win a game[1], our game will not be round based but mode based style which will consists of those rounds. We chose Q-bitz because it has the potential to be extended with new game modes and features that we will add.

In our game we have:

- Online Q-bitz
 - Multiplayer Game Modes
- Offline Q-bitz
 - Practice
 - Single Player Game
- Options
 - Adjustable color set
 - Sound options
 - Controls
- How to play

With this structure, the game will use the benefits of being digital, meaning that a user can choose multiplayer mode to enjoy the game with friends while in single player mode users can have a taste of an arcade game without the need of any other user.

This report will give an overview for the game by describing functional and non-functional requirements for developing the game successfully. The features of the game and how these features work. It will explain how the user will interact with the game using use case diagrams. Then It will explain the flow of the game by providing sequence, activity and state diagrams and will show the object structure using the class diagram. Finally it will contain the user interface mockups of the screens that the game has for each scenario.

2. Overview

Q-bitz has both online and offline game mode. In online mode, we have multiple submodes for online play. In all the game modes, if a player successfully recreates the pattern, the player earns 10 points and a countdown starts for other players. Players who finish by the time countdown ends, earn points but fewer than the previously finished

players. Each game mode is played 3 times and the player with the highest points in the end is the winner.

2.1 Game Modes

2.1.1 Race:

In Race mode, two or more players race against each other to recreate the same pattern revealed on the card.

2.1.2 Roll to Win:

In Roll To Win mode, all cubes are rolled and players try to recreate the pattern on the card without rotating the cubes. When the player decides the cubes cannot be used for recreating the pattern more, the cubes can be rolled again without a limitation of number of rolling.

2.1.3 Memory:

In Memory mode, the pattern card is only revealed for a certain time and then players try to recreate the pattern from their memory. Each player can submit their board only 3 times to check if they are finished. After 3 failed attempts, they lose the round and earn 0 points.

2.1.4 Elimination:

This mode is only valid for multiplayer games. Since the game is digital and we can have many boards instead of four in the original one. Several players (more than 4) can join to the same game room and start a tournament. In each game, the slowest player will be eliminated.

2.1.5 Single Player Mode:

A player can use this mode to practice individual skills without time limit or to have an arcade type of game experience. It consists of back to back rounds of the game which has increasing level of difficulty.

2.2 In-Game Objects

2.2.1 Board:

Players place their cubes onto the game board. Board size can change depending on the difficulty, such as 3x3 for easy, 4x4 for normal and 5x5 for hard mode.

2.2.2 Cubes:

Cubes have 6 faces with different designs on them. Players try to recreate the pattern on the card by rotating the cubes and placing them on the board.

2.2.3 Pattern:

Cards have patterns on them for players to recreate. Like board size, card size can also change depending on the difficulty.

2.2.4 Game information:

Remaining time, pattern card and other players' progresses will be displayed on the game screen.

2.3 Information and Control

2.3.1 Options:

Players will be able to customize their controls, key bindings, color set and adjust volume in the options.

2.3.2 Controls:

Cubes have 6 faces with different designs on them. Players try to recreate the pattern on the card by rotating the cubes and placing them on the board.

2.3.3 Progress and User Level:

After completing a multiplayer or single player game, user will gain experience and after gaining enough experience, user's level will increase by one.

3. Functional Requirements

3.1 User related requirements

After launching the game, user will be directed to the login screen. On this screen, user can login to their accounts with username and password. They can also create an account if they do not have one yet or request to reset their password if they have forgotten it.

3.2 Navigation

The navigation through the game will be done by a main menu which consists of the following.

3.2.1 User Menu:

This menu is a drop down menu which will appear after selecting user symbol to navigate through the user related functionality such as user settings, go online/offline and log out from the account.

3.2.2 Play Multiplayer Button:

The players can play the Q-bitz online through that screen. If the user is offline, the multiplayer menu will be greyed out..

3.2.3 Play Single Player Button:

The player can play the Q-bitz offline through that screen. There will be many different levels with different modes and difficulty settings player can choose from.

3.2.4 How to Play Button:

This button will open How to Play Menu.

3.2.5 Options Button:

This button will open the menu for game controls and all the other settings.

3.3 Game Functionalities

3.3.1 Play Game(Multiplayer):

Q-bitz is a multiplayer game and to play with other people, the player can create a room or join a existing room from the list on the screen.

3.3.1.1 Create Room:

Users can create a room for others to join and play with them. Users can create rooms with different settings, such as different game modes, difficulty, number of players and visibility of the game room. Rooms can be set to public or private, changing its visibility on the room list. Rooms can have up to 8 players.

3.3.1.2 Join Room:

Users can join an existing room from the room list on the screen. Additionally, the user can join rooms with using “room code” that helps the users to share same room sharing this code with its codes.

3.3.2 Play Game(Singleplayer-Arcade):

Users can play the game offline by themselves in the single player mode. In single player mode there will many stages user can play any time. Each level will have rewards depending how quickly the player finishes it. The difficulties of levels will increase through to each level with adjusting the pattern card size, the pattern difficulty, memory mode and using time limitation.

3.3.3 Options:

User will be able to change several options for the game:

Music: Users can change the level of the game music playing in background

Effects: Users can change the level of the effect sounds inside the game.

Cube Color: Users can change the cube colors that will form the patterns.

Background Color: Users can change the background color inside the game.

Rotation Controls: Users can change the bindings of the keys dedicated for rotating the cube.

Navigation Controls: Users can dedicate different mouse buttons for navigating a cube along the board and selecting a cube.

3.3.4 How to Play:

Users will be able to learn how to play the game using a brief and simple tutorial provided inside the game.

4. Nonfunctional Requirements

4.1 Usability

Q-bitz is a user friendly game that will be easily playable. Easily playable means that it will not take much effort to understand how to play the game. The rules will be simple and the controls of the game will be intuitive. The aim of the project is to preserve this simplicity by providing an easy to use playing interface such that any kind of player will have a comfortable game experience.

4.2 Reliability

Since the game has multiplayer support, the game server should be able to handle the number of users such that there is no lag due to the server and prevent disconnections.

4.3 Performance

Cube rotation mechanics should be implemented such that the gameplay is smooth and there is no frame loss while intensive movement. The game should run in at least 30 frames per second

4.4 Extendibility

The architecture of the software should be designed such that new features can be added in a convenient way. For example new game modes can be added easily since the core mechanics for rotating the cubes and pattern matching will be modular and can be used when needed.

5. System Models

5.1 Use case model

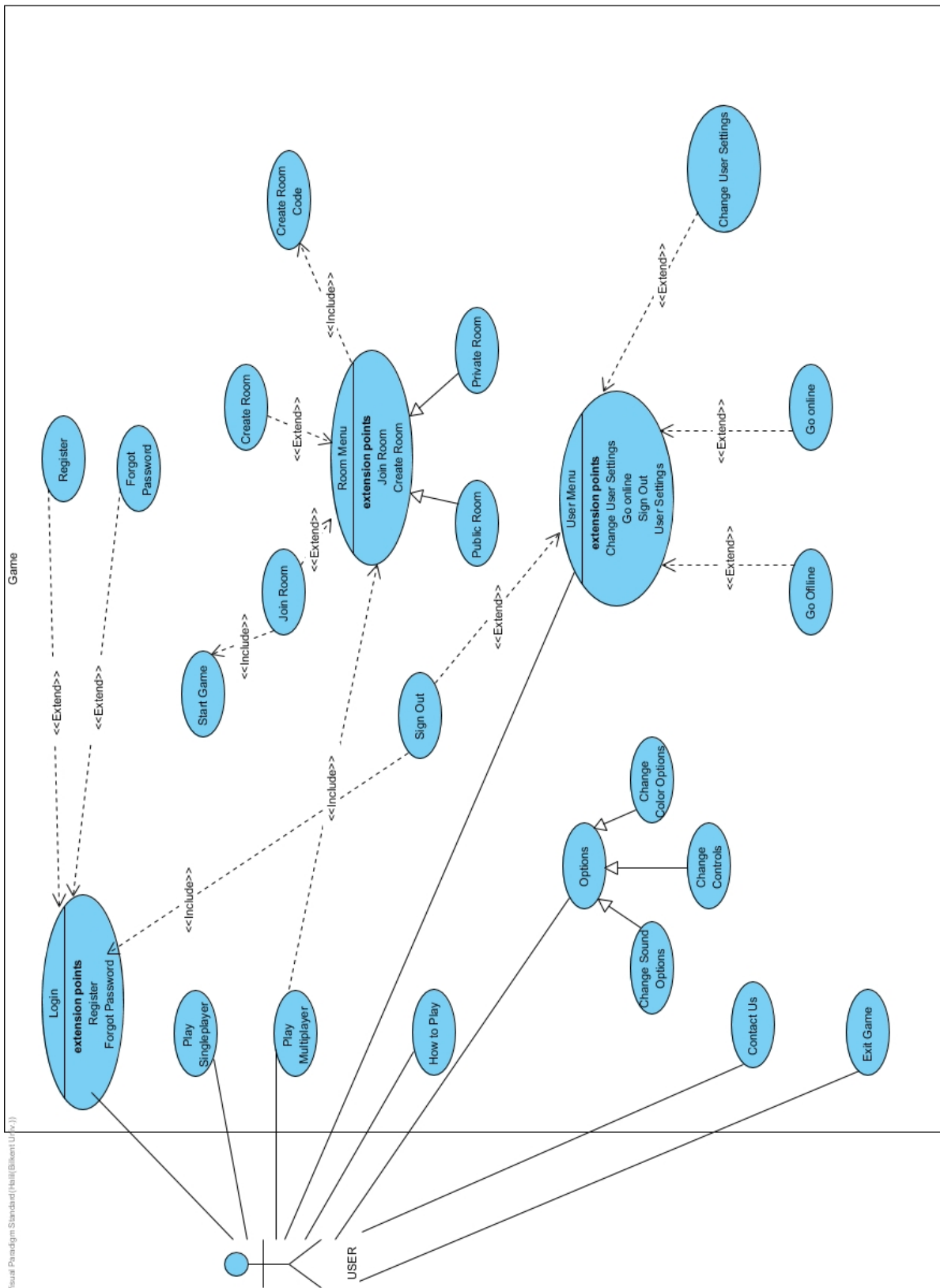


Figure 1: Use case model for Q-bitz game

5.1.1 Use Case #1

Use Case: Login

Primary Actor: User

Stakeholders and interests:

- User/users who want to login the game or register or renewal.

Pre-conditions:

- User must be online for unlogged game for the signed out or first time with game.

Post-conditions:

- No post condition

Scenario Event Flow for Login:

1. User enters the correct credentials.
2. User is directed to the main menu.

Scenario Event Flow for Forgot Password:

1. User select the "Forgot Password?" to change it.
2. User enters the e-mail which is bound to account that user wants to change.
3. User enters verification code which is sent to e-mail.
4. User enters the password to log in.

Scenario Event Flow for Register:

1. User enters the username.
2. User enters the e-mail.
3. User enters the password and confirm password.
4. If the username is unique and the passwords are matching, the account will be registered to system. If not, step 1 to 3 will repeated.

5.1.2 Use Case #2

Use Case: Multiplayer

Primary Actor: User

Stakeholders and interests:

- User/users who want to play the game with multiplayer option

Pre-conditions:

- User must be in main menu

Post conditions:

- No post condition

Entry-conditions:

- User must be online
- User should create/join a room which has different quota and mode.
- The game will be started if the room is full.

Exit-conditions:

- If the game is ended, the multiplayer room menu screen will come for users.
- User can exit from a game from "Quit" button.

Scenario Event Flow for Race Mode:

- 1.Game randomly chooses a pattern card and displays the card to the users.
- 2.User rotate and place their cubes on the board.
- 3.User recreates the pattern before other users.
- 4.User gets 10 points.
- 5.Countdown starts for other users.
- 6.Users who finish before the countdown ends, earn points. Others get 0 points.
- 7.Game starts a new round and repeats steps 1 to 6 until all three rounds end.
- 8.User with the highest points wins the game and earns rewards.

Scenario Event Flow for Roll to Win Mode:

- 1.Game randomly chooses a pattern card and displays the card to the users.
- 2.Cubes of each user will be rolled and cannot be rotated.

3. User will place the cubes on the board, trying to recreate the pattern.
4. User can re-roll all the remaining blocks any time to get different faces. User recreates the pattern before other users.
5. User gets 10 points.
6. Countdown starts for other users.
7. Users who finish before the countdown ends, earn points. Others get 0 points.
8. Game starts a new round and repeats steps 1 to 6 until all three rounds end.
9. User with the highest points wins the game and earns rewards.

Scenario Event Flow for Memory Mode:

1. Game randomly chooses a pattern card and displays the card to the users for a certain time.
2. Pattern card is flipped over so that the pattern is hidden.
3. User rotate and place their cubes on the board.
4. User recreates the pattern before other users.
5. User gets 10 points.
6. Countdown starts for other users.
7. Users who finish before the countdown ends, earn points. Others get 0 points.
8. Game starts a new round and repeats steps 1 to 6 until all three rounds end.
9. User with the highest points wins the game and earns rewards.

Scenario Event Flow for Elimination Mode:

1. Game starts with 8 players.
2. Game randomly chooses a pattern card and displays the card to the users.
3. User rotate and place their cubes on the board.
4. User recreates the pattern before other users.
5. User gets 10 points.
6. Countdown starts for other users.
7. Users who finish before the countdown ends, earn points. Others get 0 points.
8. Users with 0 points in a round gets eliminated

Alternative Event Flows:

1. If user wants to return to room menu:
 - a. User click the "Quit Game" button from game screen.
 - b. System quits the game and sends information of exiting game to server.

5.1.3 Use Case #3

Use Case: Single Player

Primary Actor: User

Stakeholders and interests:

- User/users who want to play the game with single player option

Pre-conditions:

- User must be in main menu

Post conditions:

- No post condition

Entry-conditions:

- No entry conditions

Exit-conditions:

- If the user finishes the level, they will be directed to the level selection screen.
- User can exit from a game from "Quit Game" button.

Success Scenario Event Flow for Single Player:

1. User selects a stage.
2. Game displays a pattern card to the user.
3. User rotate and place their cubes on the board.
4. User recreates the pattern.
5. User earns stars based on the time finished.

Alternative Event Flows:

1. If user wants to return to level selection screen:
 - a. User click the "Quit Game" button from game screen.
 - b. System quits the game

- c. User is directed to the stage selection screen

5.1.4 Use Case #4

Use Case: How to Play

Primary Actor: User

Stakeholders and interests:

- User wants to learn how to play Q-bitz.
- System displays how to play screen.

Pre-conditions:

- User must be in the main menu.

Post conditions:

- No post condition.

Entry-conditions:

- User selects “How to Play” button from main menu.

Exit-conditions:

- User clicks on “Home” button in the “How to Play” screen.
- User presses “Esc” from keyboard.

Event Flow for Single Player:

User clicks on “How to Play” button in the main menu.

System displays “How to Play” screen.

Alternative Event Flows:

1. If User wants to learn how to play:
 - a. User clicks on “How to Play” button in the main menu.
 - b. System displays How to Play Screen.
2. If User wants to return to main menu:
 - a. User clicks on “Home” button in the “How to Play” screen.
 - b. System displays main menu.

5.1.5 Use Case #5

Use Case: Options

Primary Actor: User

Stakeholders and interests:

- User/users who want to change options related to sound and color.

Pre-conditions:

- User must be in the main menu.

Post conditions:

- No post condition.

Entry-conditions:

- No entry conditions.

Exit-conditions:

- User can exit from options menu from "Home" button.
- User presses "Esc" from keyboard.

Alternative Event Flows:

1. If user wants to change the options while playing:
 - a. User click the "Options" button from game menu.
 - b. Options menu will pop up in game screen, then user will change options.
 - c. User ends the options menu with "Back" button to continue the game.

5.2 Dynamics models

5.2.1 Sequence Diagrams

5.2.1.1 Select Cube

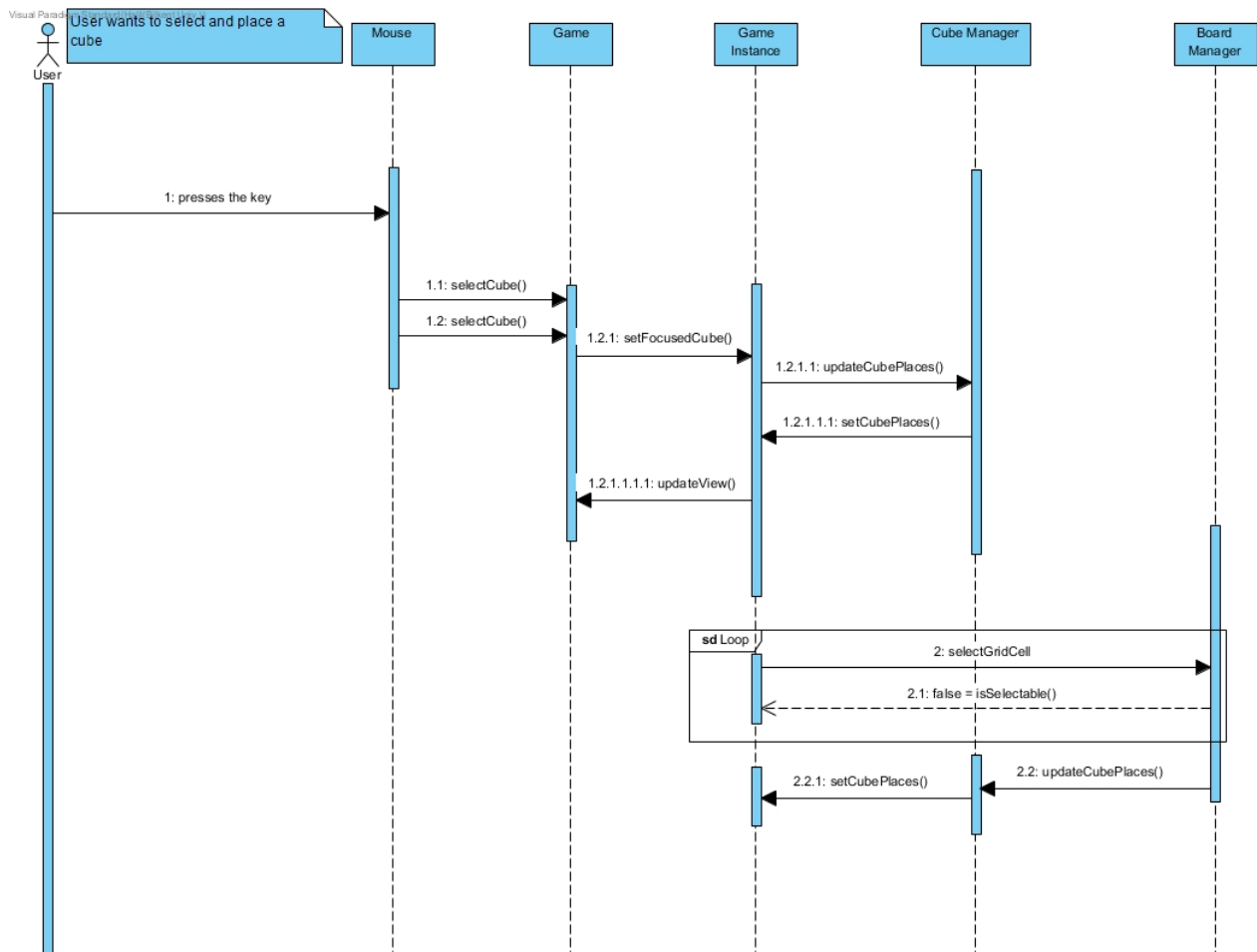


Figure 2: Sequence Diagram for Selecting a Cube

- If the user wants to select a cube and place it on the board, user must click on a cube with the mouse to select it. To focus and zoom in on the cube, user must click on the selected cube. When the selected cube is clicked, Game Instance will update the view and cube will be zoomed in for rotation. If an empty grid cell is selected, cube will be placed on it. After placing a cube on the board, Board Manager will update the cube places and set the cube on the board.

5.2.1.2 Rotate Cube

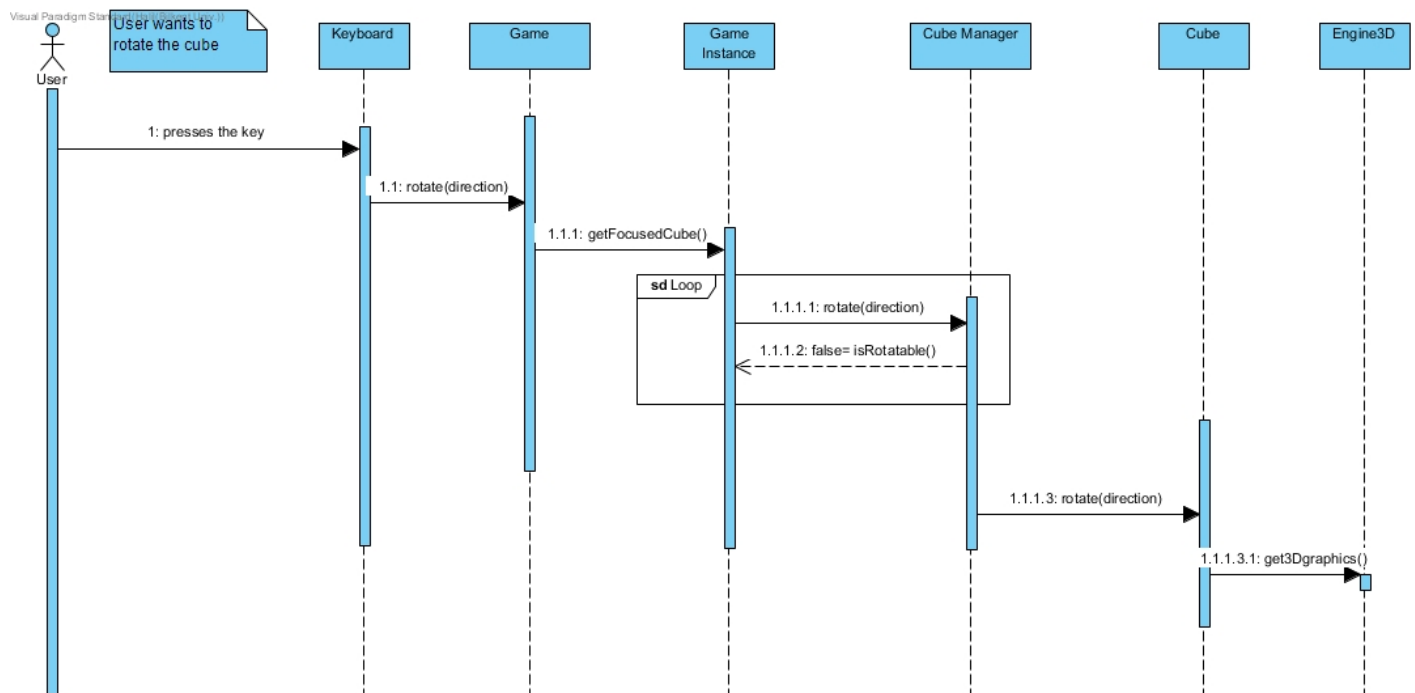


Figure 3: Sequence Diagram for Rotating a Cube

- The user will rotate the cubes by pressing some determined keys in the keyboard, this will be much more efficient than rotating with mouse in terms of time. The cubes may not be rotatable if the game is in Roll to Win mode, thus Game Instance asks for this information to Cube Manager. If the cube is rotatable, Cube class will rotate the cube by using the 3D Engine.

5.2.1.3 Singleplayer Game

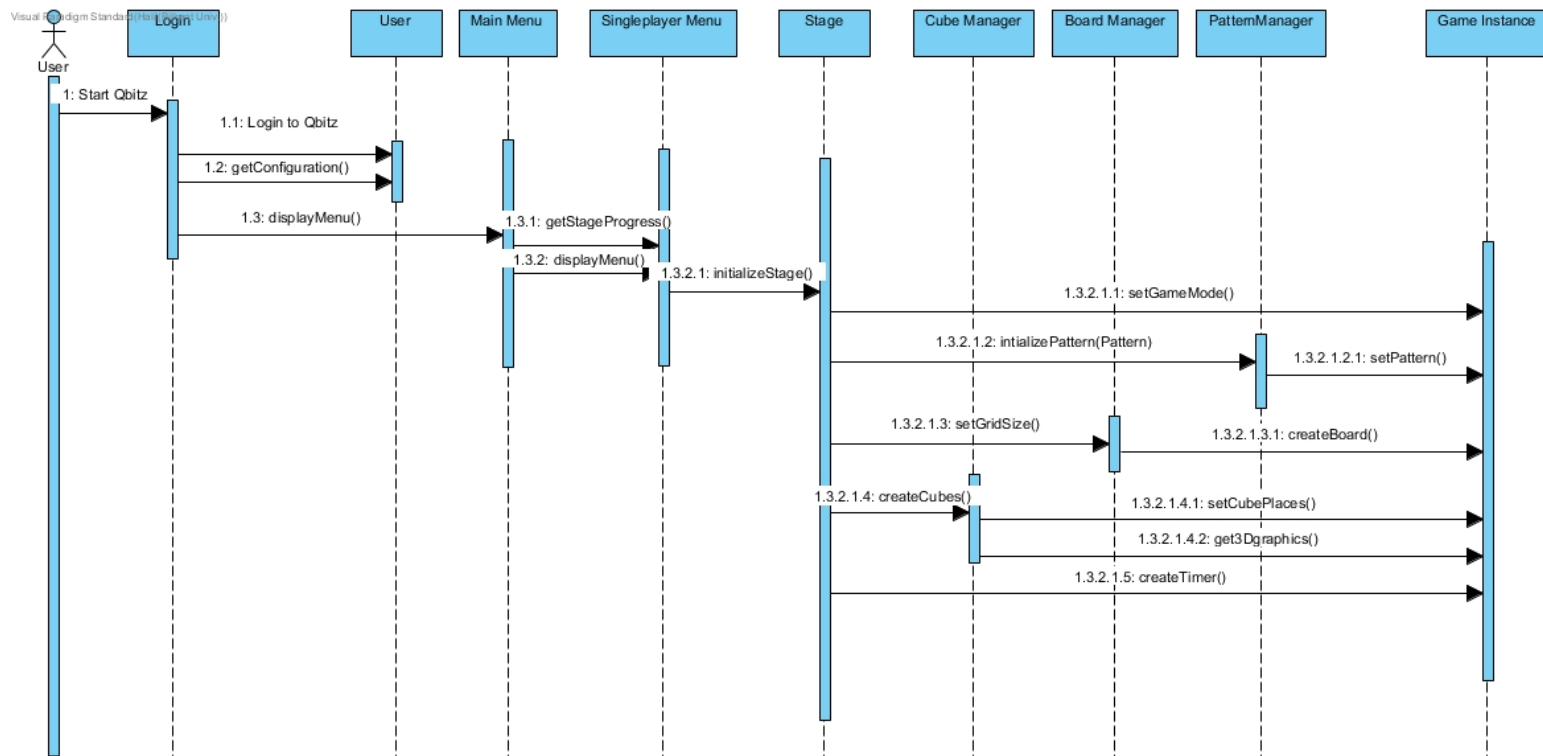


Figure 4: Sequence Diagram for a Singleplayer Game

- The user will login to the system, if the user is in singleplayer mode, the user data will be obtained from configurations, thus the login screen will automatically disappear. Then Singleplayer menu will show up to display different stages that the user can play. After selecting a stage, the game will start with several communications to the Game Instance. First the game mode will be selected. The grid size will be initialized and the pattern will be initialized which is automatically generated by the game. Then the cubes will be created and the timer will start along with the game itself.

5.2.1.4 Multiplayer Game

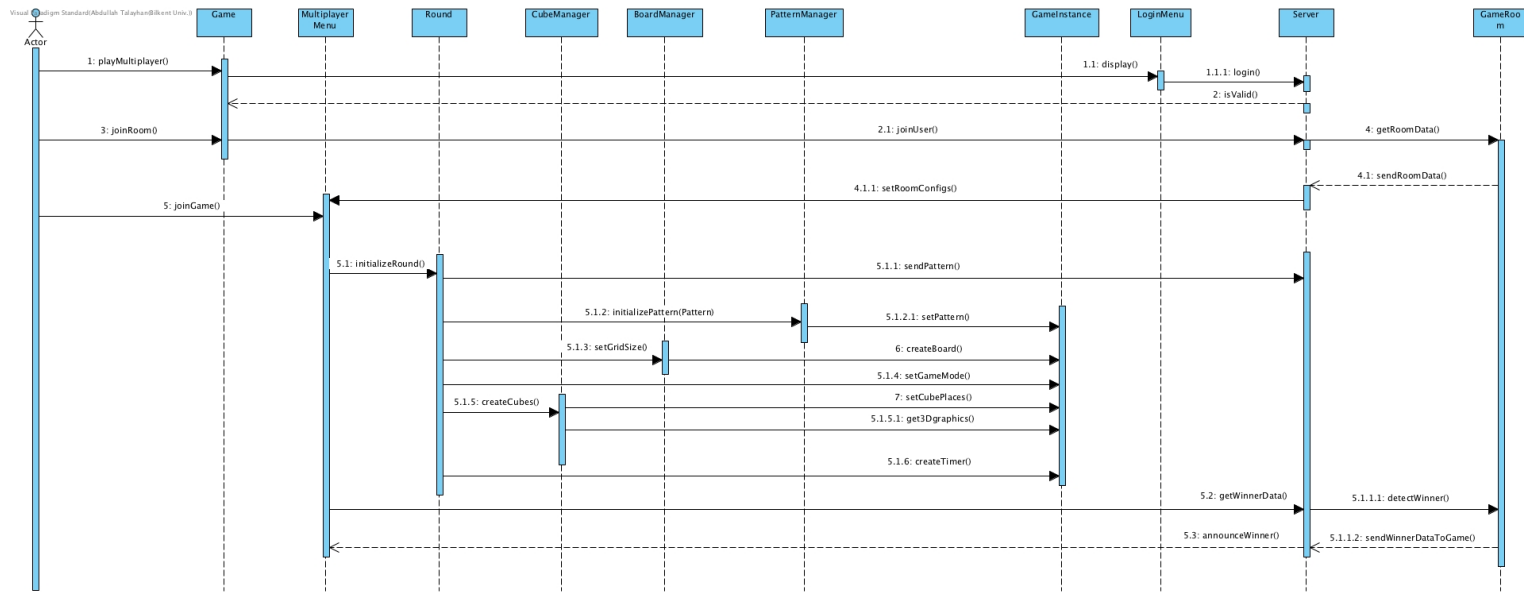


Figure 5: Sequence Diagram for a Multiplayer Game

- The user will login to the system, if the user is in multiplayer mode, the user credentials will be controlled from the server and necessary replies will be sent. After successfully logging in to the system, the user will join to a game room. The details of the room such as information related to other players will be obtained by the server. Then the user joins the game and the game procedure will be the same as the single player one. The game will start with several communications to the Game Instance. First the game mode will be selected. The grid size will be initialized and the pattern will be initialized which is automatically generated by the game. Then the cubes will be created and the timer will start along with the game itself. After the game finishes, the game winner will be detected.

5.2.1.5 Options

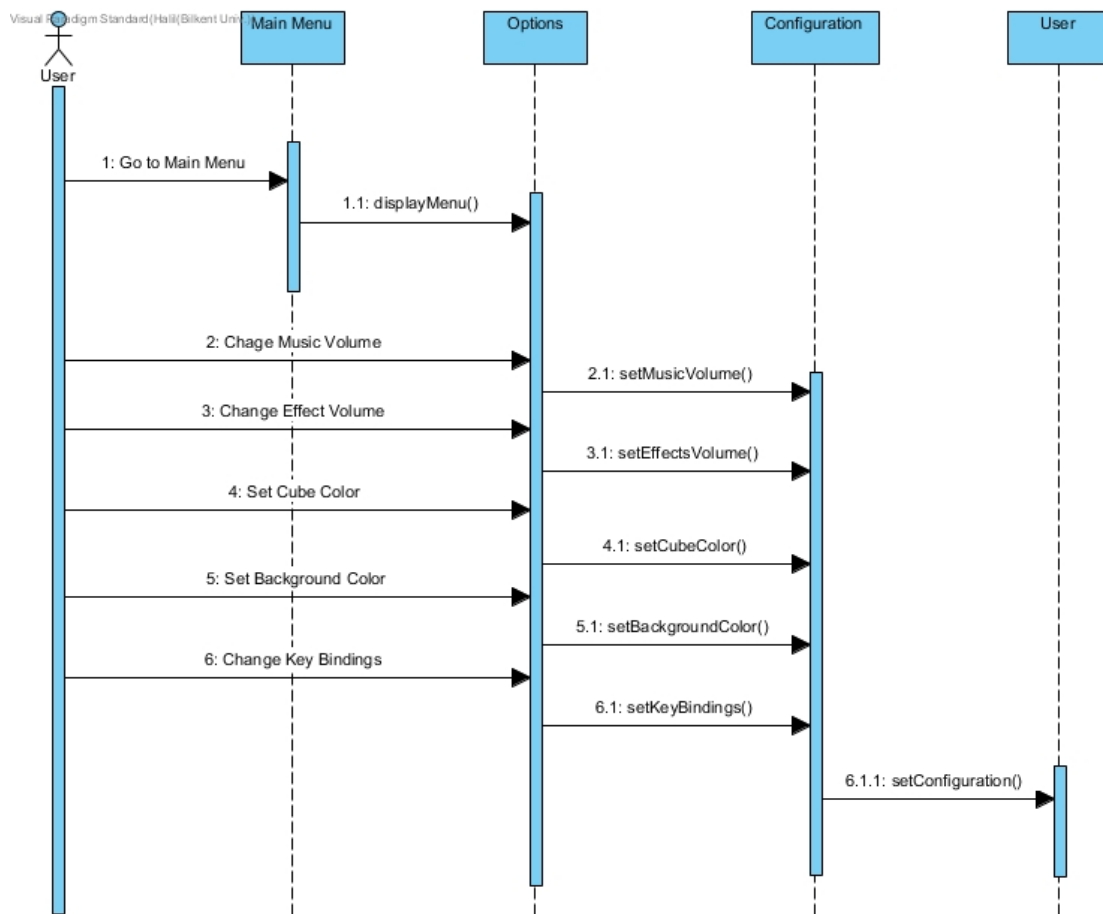


Figure 6: Sequence Diagram for Game Options

- To change the configurations of the game, user must enter the options menu from the main menu. In the options menu, music volume, effects volume, cube color, background color and the key bindings for the controls can be modified. After applying the configurations, user's old configurations will be overridden with the new modifications.

5.2.2 Activity Diagram

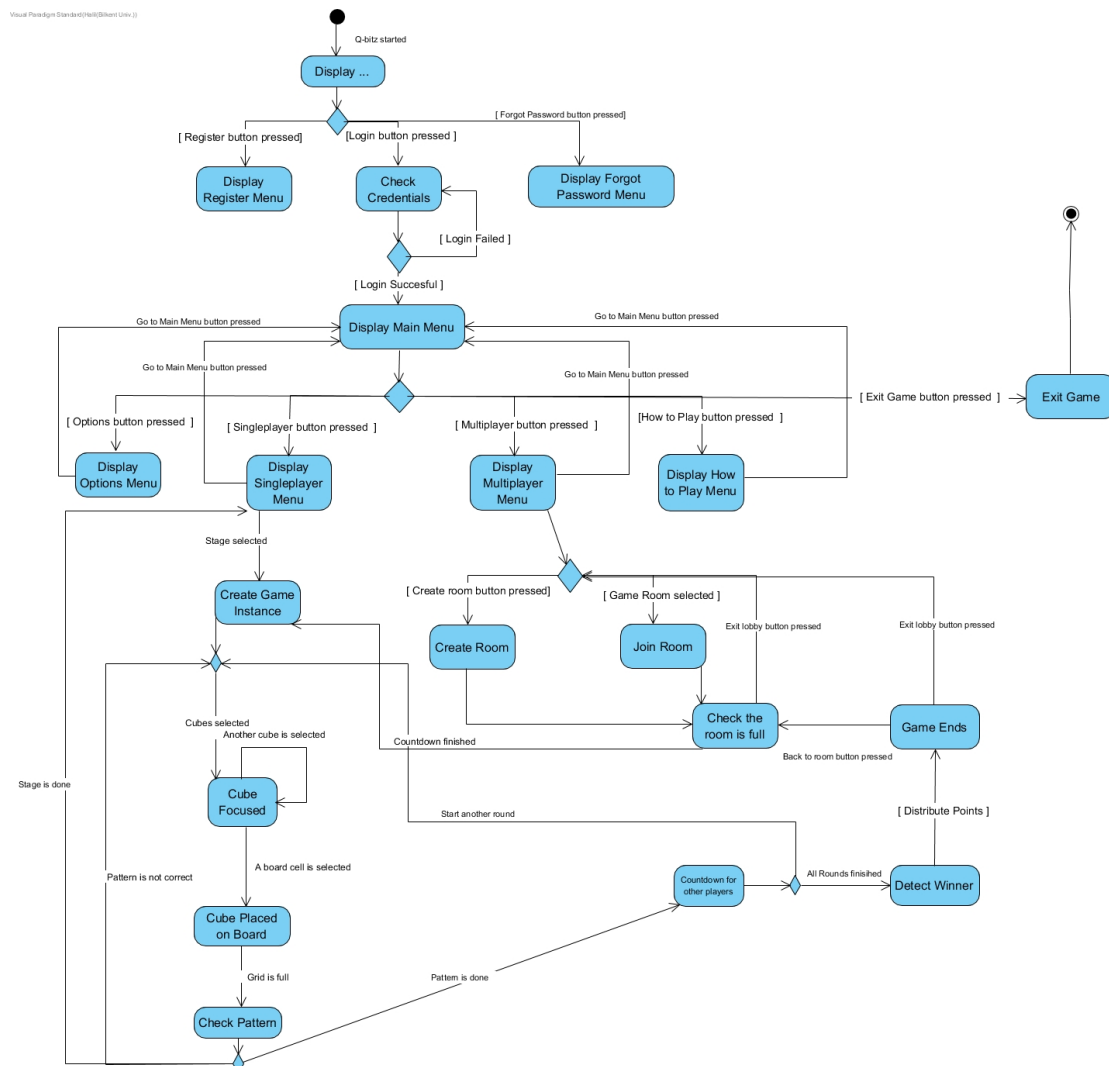


Figure 7: Activity Diagram for the Game

- The system will display the initial game screen which consists of register, login and forgot password options. Then the necessary credential checking procedures will be handled. After registering and logging in successfully, the game will display the main menu consisting of singleplayer, multiplayer, how to play, options and exit choices. If Singleplayer mode is selected, the system will create a game instance depending on the stage that is selected. If Multiplayer mode is selected, the system will add the user to a room or create a room depending on the request. After the room is full, the game instance will be created. For both playing options the system will change the focus status or orientation of the cube depending on the inputs. If the board becomes full, the system will check the pattern for correctness, then the stage menu will appear again for singleplayer while the post-game screen will appear which announces the winner for the multiplayer mode.

5.2.3 State Diagrams

5.2.3.1 Cube

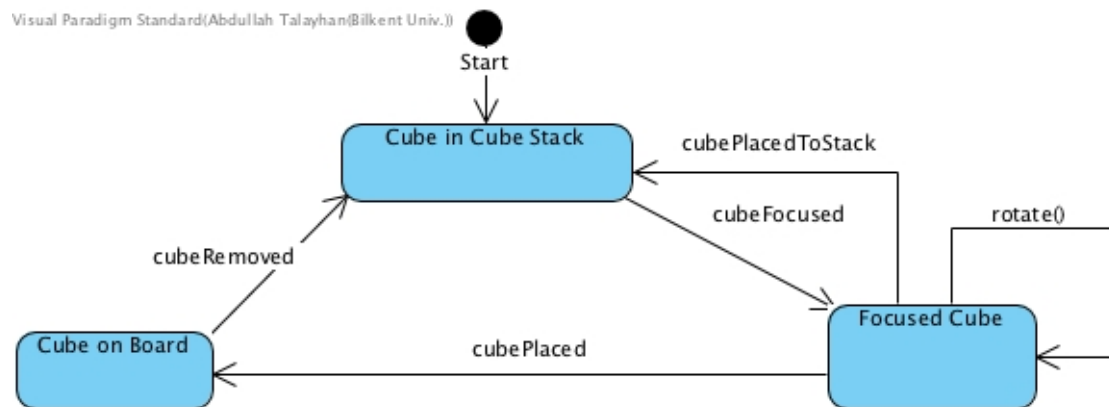


Figure 8: State Diagram for the Cube

- The cubes will be listed in a cube stack, the user can select a cube which will put the cube to a focused state. A focused cube can be rotated and then placed which will put the cube to the Cube on Board state. The user can also remove a cube from the board and this will put the cube to the Cube in Cube Stack state.

5.2.3.2 Game Room

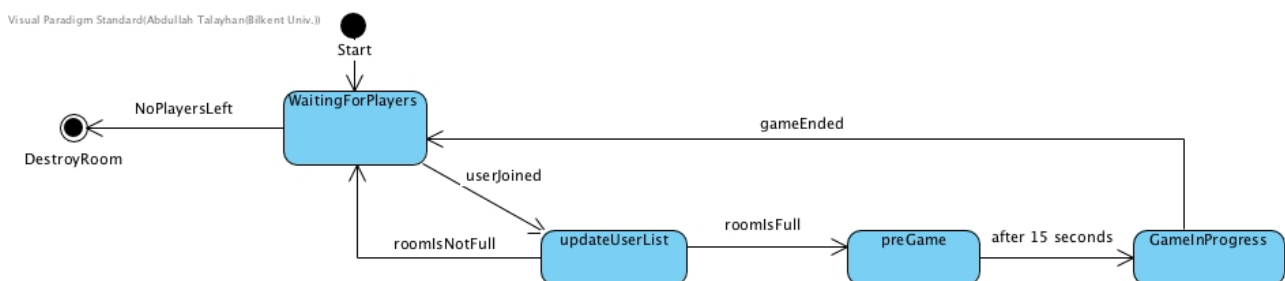


Figure 9: State Diagram for the Game Room

- A Game Room is in the “Waiting For Players” state initially. If a user enters the room the user list is updated, if the room becomes full, the room will be in pre-game state. Pre-game state automatically starts the game after 15 seconds and the room will be in “Game In Progress” state. If the room is not full or the game has ended, the room will go back to “Waiting For Players” state.

5.3 Object and class model

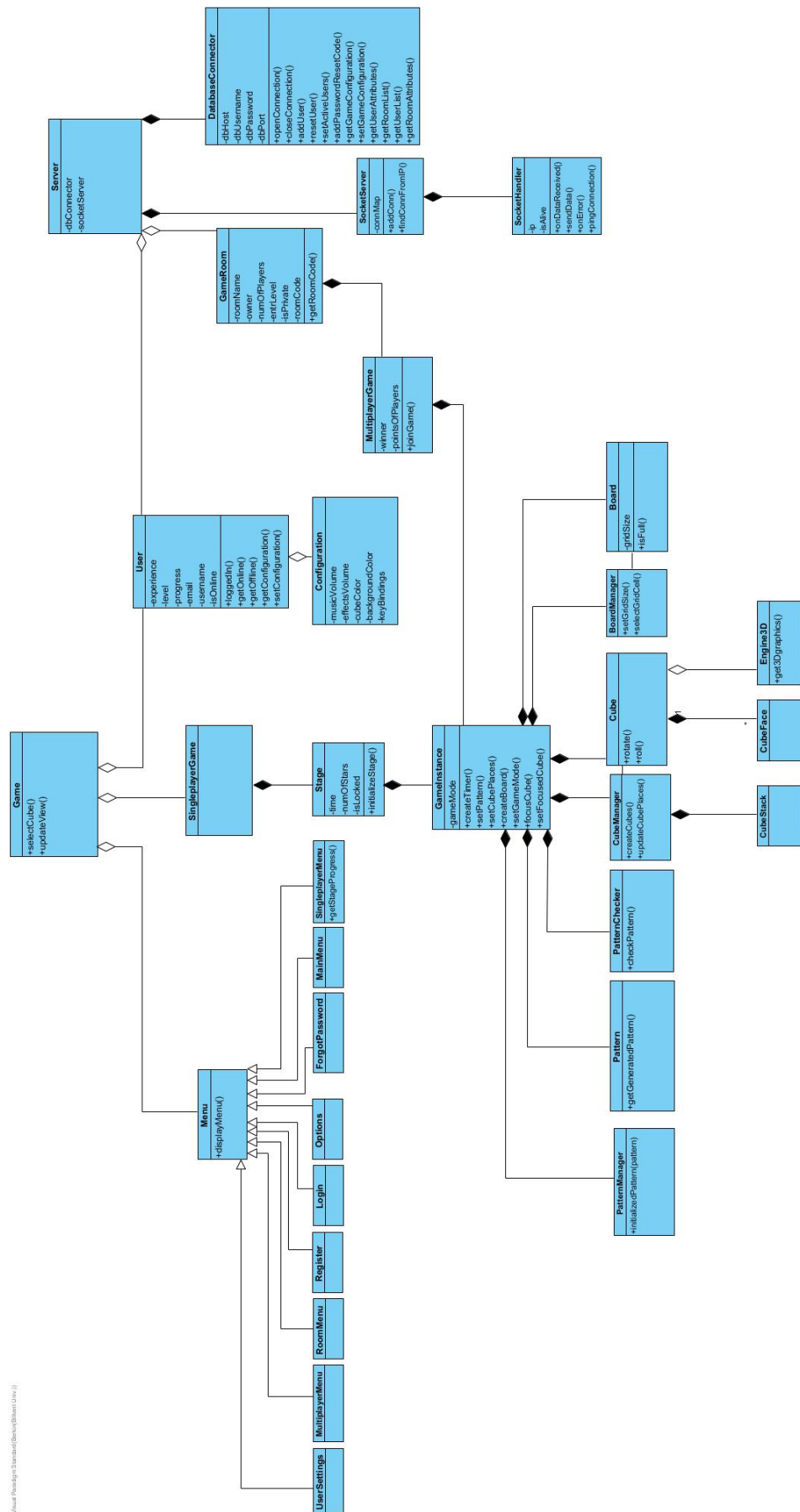


Figure 10: Class Diagram for the Game

Object and Class Model Description:

Our game has a root class "Game", handling the input listeners and the view.

- We have a "Menu" abstract class for all the navigation and non-game screens. User settings menu will display the fields for changing the email and password of the account and options for resetting the progress or deleting the account. Multiplayer menu, will display the room list and an option to create a game room. Room menu, is the menu for creating a new room and selecting its game mode, number of players and minimum required level for joining the room. On the Register menu, user can create an account and on the Login menu, user can log in with an existing account. If the user forgets their password, they can reset their password from the ForgotPassword menu. On the Options menu, user can configure the game. Main menu allows navigation between game modes and other screens such as how to play and options. In the singleplayer menu, user can select a stage to play.
- "User" class contains the information about the user such as experience, username, configuration and the online status.
- "Configuration" class contains the volume levels of music and effects, cube color scheme and keybindings for controls.
- "Stage" class contains the information about used time for that stage, number of stars that obtained from that stage and locked status of that stage. Also, "Stage" class contains the initializeStage() method to set the stage constraints for the stage object.
- "Game Instance" class is the main class that contains all the game components such as "PatternManager", "PatternChecker", "Pattern", "Cube Manager", "Cube", "Board Manager" and "Board".
- "Pattern" class is contains the pattern that will be recreated in the game with cubes.
- "PatternManager" class initializes the pattern in the "Game Instance". If it is a singleplayer game, the pattern will be pre-generated pattern stored locally and if it is a multiplayer game, game will send a request to the server and server will generate a pattern and send it to the "PatternManager".
- "Board" class contains the grid, the game is played on. If the grid is full, user can check if they recreated the pattern correctly.
- "Board Manager" class initializes the board controls the state of the board.
- "Cube" class provide the functionality for rotation and rolling of the cubes. "rotate()" function provide the rotation of cube one face at a time to a specified direction.
- "CubeStack" class is for keeping track of the cube's places on the stack.
- "CubeFace" class contains the information of the face of the cubes and order of the faces to create the cube object exactly for each creation.

- "CubeManager" class contains the functions for creating the cubes and updating the cube places for "GameInstance" class.
- "GameRoom" class contains the data of the game lobby. "GameRoom" has a name, owner, number of players in the room, minimum level to join and visibility status. "GameRoom" can also generate a room code for others to join via code.
- "MultiplayerGame" class contains the attributes for endgame of a multiplayer game such as "winner" of the game and "pointsOfPlayer" to distribute the points to users. "joinGame()" function provides to joining a game room where users can play multiplayer game with other users.

5.4 User interface

5.4.1 Login

When the game is launched, user will be presented with the login screen. On this screen, users can login or create a new account or reset their password. If the user has no account, user can create an account by “Register”.


The image shows a login menu interface. In the top-left corner, there is a small black house icon. Centered on the page are two text input fields: the first is labeled 'username' and the second is labeled 'password'. Below the 'password' field, there are two links: 'Forgot Password?' and 'Register', both in blue text. To the right of these links is a grey button with the text 'Log-in' in black.

Figure 11: Login Menu

5.4.2 Register

User can create a new account by entering an email, username and a password.

The image shows a register menu interface. In the top-left corner, there is a small black house icon. Centered on the page are four text input fields stacked vertically: 'username', 'e-mail', 'password', and 'confirm password'. Below the 'confirm password' field is a grey button with the text 'Register' in black.

Figure 12: Register Menu

5.4.3 Main Menu

Main menu provides the user with a selection of choices such as multiplayer, single player, how to play and options. User can also view their progress and other settings by clicking on the username.



Figure 13: Main Menu

5.4.4 User Settings

User can change their email, username or password on the user settings screen. User can also delete their account or reset their progress on this screen.

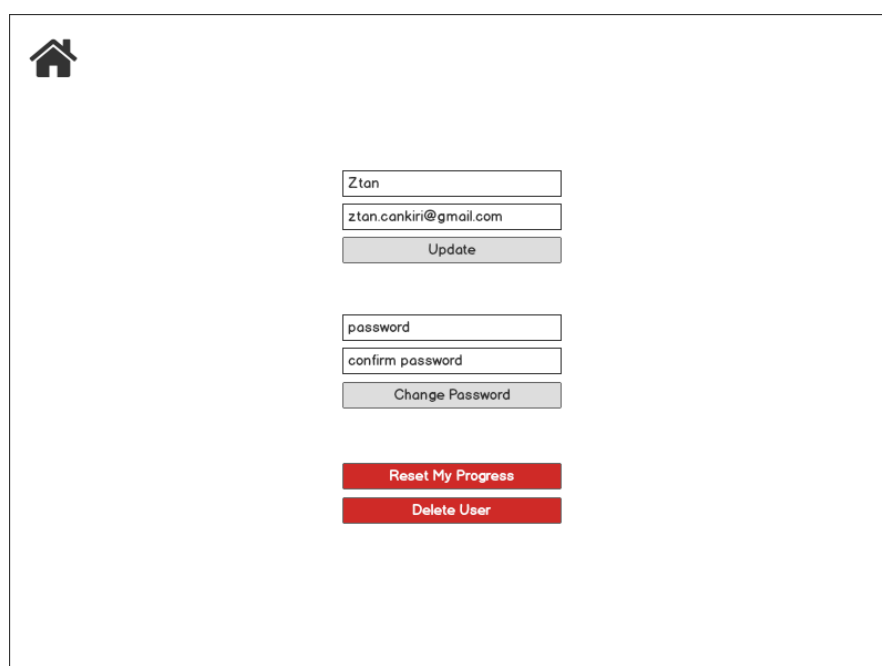


Figure 14: User Settings Menu

5.4.5 Room Selection in Multiplayer Mode

User can select a room from the list to join. User can also enter a private room by entering a room code. Rooms can be filtered with checkboxes or keywords.

The screenshot shows the 'Room Selection Menu' interface. At the top, there is a user profile section with a profile icon, 'Username', a flask icon with '635 / 1000', a star icon with 'Level 1', and an 'Online' status indicator. Below this, there are search and filter options: a 'filter rooms...' input with a magnifying glass icon, and checkboxes for 'Race', 'Roll to Win', 'Memory', and 'Tournament'. To the right, there is a 'room code' input, an 'Enter Private Room' button, and a 'Create New Room' button. The main part of the interface is a table with the following columns: 'Room Name', 'Game Mode', 'Owner', 'Players', and 'Entrance Level'. The table contains the following data:

| Room Name | Game Mode | Owner | Players | Entrance Level |
|--------------------------|-------------|-----------------|---------|----------------|
| QBits Masters | Race | Ztan | 6 / 6 | 5 |
| Last QBits Benders | Roll to Win | con456 | 2 / 5 | - |
| QBits Maestros | Memory | jejeje | 2 / 2 | 16 |
| Game of QBits | Tournament | eray_06 | 5 / 8 | 21 |
| Obsessive Mathematicians | Race | sait | 6 / 6 | 5 |
| Single Ladies Only !!! | Roll to Win | crazy_berkin | 2 / 5 | - |
| QBits Maestros | Memory | buffer4head | 2 / 2 | 16 |
| asdasd123124 | Tournament | bestPlayer4Ever | 6 / 8 | 21 |
| QBits Gentlemen | Race | david | 6 / 6 | 5 |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Figure 15: Room Selection Menu

5.4.6 Room Creation in Multiplayer Mode

User can create a room for others to join. On this screen, user enters a room name, number of players and chooses a game mode and a minimum entrance level. A room can be public or private. Public rooms will be listed on the room list, while private rooms will only be accessible via code.

The screenshot shows the 'Room Creation Menu' interface. It features the same top navigation bar as Figure 15. A 'New Room' modal dialog is open in the center, with the following fields and options:

- Room name**: A text input field.
- Room Type**: Radio buttons for 'Public Room' (selected) and 'Private Room'.
- # Players**: A text input field.
- Entrance Level**: A text input field.
- Game Mode**: A dropdown menu.
- Create Room**: A button at the bottom of the modal.

The background shows the same room list table as in Figure 15, but it is dimmed.

Figure 16: Room Creation Menu

5.4.7 Game Lobby

On the game screen, all the cubes, a grid to place the cubes, the pattern to recreate, the remaining time and other players are displayed. User can zoom on a cube by clicking on it and then rotate the cube. Cube can be placed on the the board by clicking on a tile.



Figure 16: Game Lobby

5.4.8 Game Screen

After joining a game room, user will be directed to the game lobby. In the game lobby, there will be a list of other users that have already joined the same game room and their level. Room name, game mode, the number of players, minimum level to join the room and board size will also be displayed.

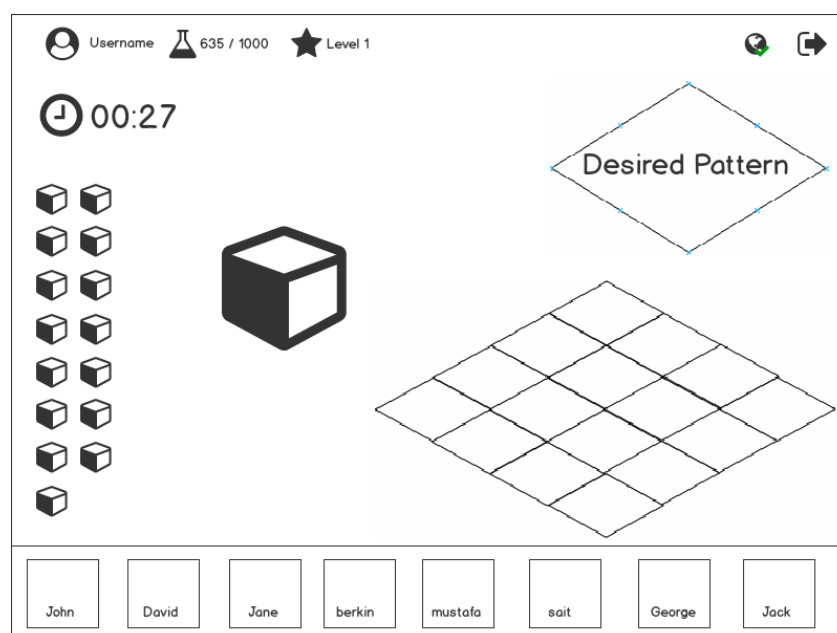


Figure 17: Game Screen

5.4.9 Post-Game Screen

After the game is finished, user will be directed to the post-game screen, where the ranking of users' and the winner will be displayed. User can choose to return to the room selection screen or join a new room with the same players who also wants to rematch.

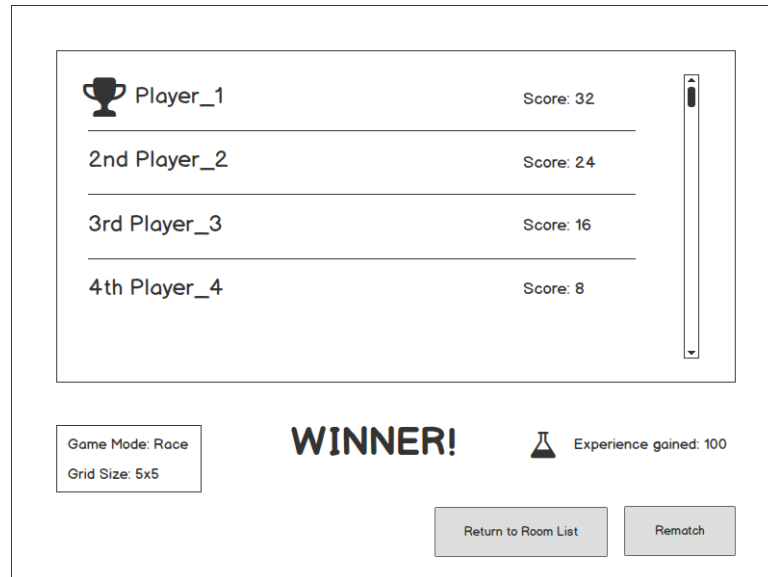


Figure 18: Post-Game Screen

5.4.10 Singleplayer Stage Selection Screen

User can play offline in single player mode. User can select a stage and play it. Stars represent the performance of player on that stage. Locked stages will be shown with lock symbol on them and can be unlocked by playing previous stages.

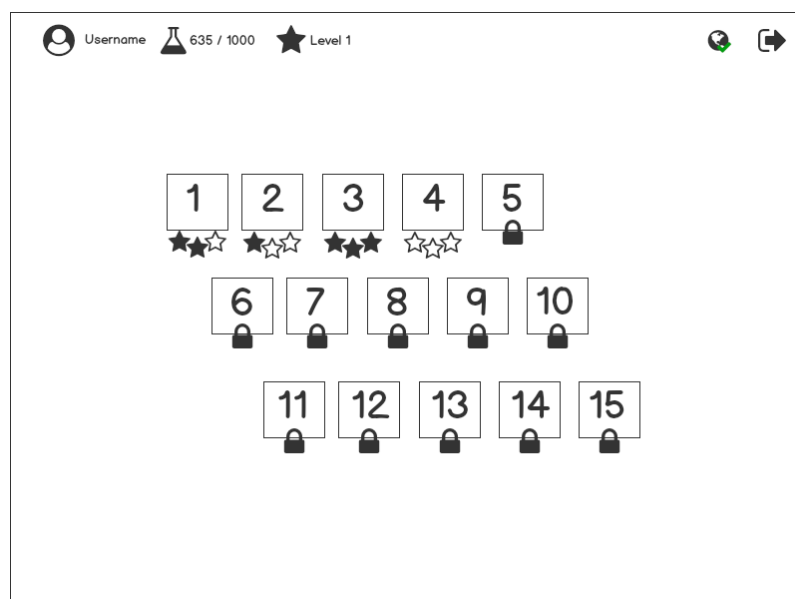


Figure 19: Singleplayer Stage Selection

5.4.11 How to Play

In how to play screen, there will be illustrations explaining the rules of all the game modes and the controls.



Figure 20: How to Play Screen

5.4.12 Options

In options menu, user can adjust the game's volume, change the color scheme of cubes and key bindings.

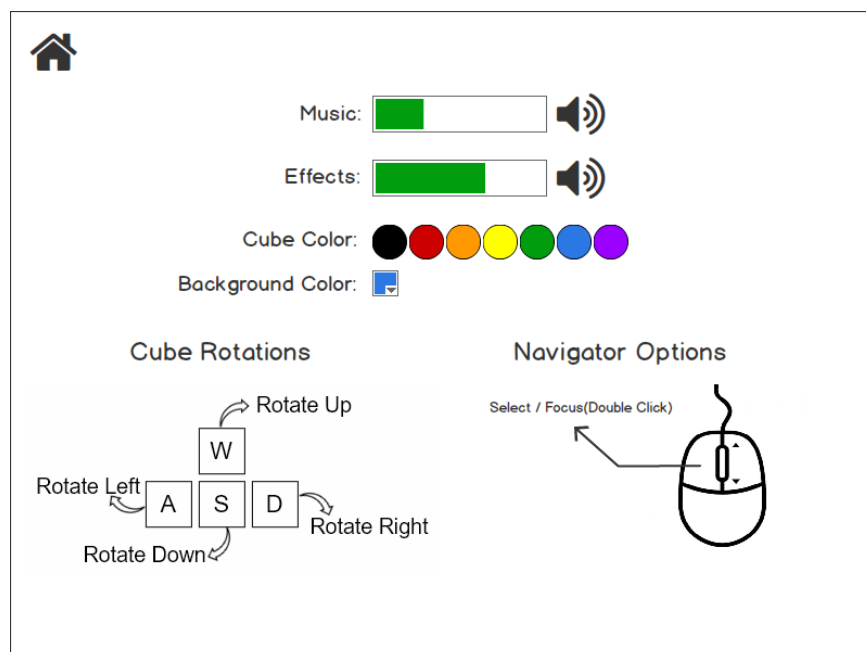


Figure 21: Options Screen

6. References

[1] QBitz Official Rules: http://www.boardgamecapital.com/game_rules/q-bitz.pdf