CS 319 Object-Oriented Software Engineering

Spring 2019

Analysis Report

My Little Quadrillion - Group 1C

- Alemdar Salmoor 21500430
- Ece Çanga 21600851
- Erkin Beşer 21400961
- Gökçe Sefa 21400596
- Mustafa Azyoksul 21501426
- Talha Şen 21502663

Instructor - Eray Tüzün

TABLE OF CONTENTS

1. Introduction 4

2. Overview 5
   2.1. Authentication 5
   2.2. Navigation 5
   2.3. Gameplay 5
      2.3.1. Arcade Mode 6
      2.3.2. Rush Mode 6
      2.3.3. Level Editor 6
      2.3.4. Sandbox Mode 6
   2.4 Data Management 7

3. Functional Requirements 7
   3.1. Login & Register 7
   3.2. Play Game 7
      3.2.1 Arcade Mode 7
      3.2.2 Sandbox 8
      3.2.3 Level Editor 8
      3.2.4 Rush Mode 8
   3.3. Settings 8
      3.3.1 Night Mode 9
      3.3.2 Color-blind Mode 9
      3.3.3 Theme Combobox 9
      3.3.4 Volume Slider 9

4. Non-functional Requirements 9
   4.1. Performance 9
   4.2. User Friendly Interface 10
   4.3. Compatible Interface with Real World 10
   4.4. Extendibility 10
      4.4.1. Achievements 11
      4.4.2. Challenges 11

5. System Models 11
   5.1. Use case model 12
   5.2. Dynamic models 20
      5.2.1. State diagrams 20

5.2.1.1. Level Editor State Diagram 20
5.2.2. Sequence Diagrams 20
5.2.2.1. Login Sequence Diagram 20
5.2.2.1 Game Play Sequence Diagram 22
5.2.3. Activity Diagram 23
5.3. Object and class model 24
5.3.1. Achievement 24
5.3.2. GameLevel 24
5.4. User interface 26
5.4.1. Navigational Path 26
5.4.2. Screen Mock-ups 27
5.4.2.1. Starting Page 27
5.4.2.2. Register Page 28
5.4.2.3 Login Page 32
5.4.2.4 Credits Page 34
5.4.2.5 Main Menu Page 35
5.4.2.6 Settings Page 36
5.4.2.7 Achievements Page 37
5.4.2.8. Play Menu Page 38
5.4.2.9 Tutorial Page 39
5.4.2.10. Arcade Page 40
5.4.2.11. Arcade Game Easy Level 1 Page 41
5.4.2.12. Rush Game Menu Page 42
5.4.2.13. Rush Mode 3- puzzle Page 43
5.4.2.14. Rush Mode 3-minute Page 44
5.4.2.15. Rush Mode Restricted Page 45
5.4.2.16.Level Editor Page 46
5.4.2.17. Create New Level Page 47
5.4.2.19. Sandbox Page 48
7. Glossary & References 49

1. Introduction

As group 1C, we have chosen the game “Quadrillion” by Smart Games to implement. The story and the design of this game is more minimalistic compared to the other games. That is why Quadrillion is much more open to major innovations. According to the experts, “Quadrillion” focuses on five main skills that include concentration, logic, problem solving, spatial insight and visual perception [1] which obviously indicates the qualifications of both computer engineers and computer scientists.

Since this project is a great opportunity to express each other’s creativity and improve our teamwork skills, every single team member is motivated and ambitious enough to produce our brand new version called “My Little Quadrillion”. My Little Quadrillion is a digital, single-player, puzzle, board game. The game is inspired by the physical board game called Quadrillion by Smart Games which is mentioned above [1]. The game pieces include four two-sided 4x4 grids and ten pentomino, one tetromino and one trimino pieces. Purpose of the game is to place all the given pieces to available locations with different restrictions. In arcade mode, grids and some pieces are pre-laid and the player tries to place the remaining pieces to the spaces available on the grids. Two-sided boards have unique obstacles on them and there are many different board layout combinations. Harder the level gets, fewer pieces are pre-laid on the board.

My Little Quadrillion has four different game modes:

- Arcade mode
- Rush mode
- Level editor
- Sandbox mode

It is a desktop application. Gameplay requires mouse and keyboard. Implementation will be coded in Java, using JavaFX 11.

2. Overview

2.1. Authentication

Immediately after the game launch, the opening screen requires the user to either register or log in. In order to register, the user has to enter a valid username and password. Login is based on entering the registered username and the corresponding password. The password and username combinations are kept in the database. Moreover, user credentials are stored for game progress tracking.

2.2. Navigation

After logging in, game brings user at main menu. Here and across all screens in the game, on the top bar, user can see his/her username, logout button, volume setting and help button. On main menu, there exists play button, achievements button, options button and quit button. Play button takes the user to the play menu which has five different buttons for five different game modes. Achievements button takes the user to the achievements page where it can browsed through the unlocked and locked achievements. Options button takes the user to the options page where the user can set different visual themes, activate night theme and change sound level.

2.3. Gameplay

My Little Quadrillion features four different game modes. Briefly, the purpose of the game is to place the pieces on the given grids. User selects a piece and drags it to the desired location one at a time in order to fill all the empty and available places on game board. After completing a game, depending on game mode, user collects stars based on performance on that level. Moreover, by satisfying various quests, user collects trophies that can be seen on achievements section.

2.3.1. Arcade Mode

In arcade mode, there are four difficulty levels: easy, medium, hard, and expert. Each difficulty include ten different puzzles. All levels are locked by default, except first levels of easy and medium. By completing a puzzle, players unlock the next puzzle in that difficulty level. Players also get up to three stars in each level based on the time elapsed and moves made. Player unlocks first level of hard by collecting 60% of the available stars, and first level of expert by collecting 80% stars from hard level puzzles. Personal progress is saved in database per each user.

2.3.2. Rush Mode

In rush mode, there are three sub-categories: limited time challenge, sprint challenge, and limited move challenge. In limited time challenge, player tries to solve as much puzzles as possible in a given amount of time. In sprint challenge, player tries to solve set number of puzzles as fast as possible. In limited move challenge, player tries to solve as many puzzles as possible with given number of moves. Points earned are, depending on challenge, based on number of puzzles solved, time elapsed and moves made.

2.3.3. Level Editor

In level editor, players are allowed to create their own levels, save them and play them later or share them with others. Level creation consists of three stages. In the first stage, player can rotate, move, and place 4x4 grids to the game board. In the following stage, player sets up the puzzle by placing some pieces to the board. In last stage, player solves the puzzle so that it is guaranteed to have at least one solution. After solving, player can submit and save the puzzle.

2.3.4. Sandbox Mode

In sandbox mode, players can place as many grids and pieces to the game area they want. Pieces can be duplicated. There is not much constraint or purpose here.

2.4 Data Management

Game related data is stored in the database. The stored information in the database are: username, password, each game mode progress information and achievements.

3. Functional Requirements

3.1. Login & Register

In the first screen of the game, if the player has not signed up yet he/she can register to the system in the register screen. If the player has an existing account, he/she can choose a login button to enter their personal account.

3.2. Play Game

Play game has four sub-modes:

- Arcade Mode
- Sandbox Mode
- Rush Mode
- Level Editor Mode

3.2.1 Arcade Mode

Arcade Mode has different difficulties which are Easy, Medium, Hard and Expert levels. Each difficulty has ten different levels to demonstrate the entire difficulty. However, each level provide at most three stars according to player performance. Also, by the time the earned star count increases, total star number automatically increases too, in the main arcade mode screen, and the next level will be unlocked in the same difficulty level.

3.2.2 Sandbox

Sandbox level is a free mode without any hardship level, stricted time or movement number in this game. Sandbox provides users unlimited number of grids and puzzle pieces and locations. There is no rule restrictions for the random game to be played.

3.2.3 Level Editor

Level editor can enable users to create their own levels in different shapes, hardship by introducing a unique impression. Players are able to modify the game as they are the owners of the game.

3.2.4 Rush Mode

Different type of challenges are provided by Rush Mode in play menu in a game. This mode creates the difference among the other versions of “Quadrillion”. Player can succeed from different challenges.

3.3. Settings

This functional part of Settings have four sub categories:

- Night Mode
- Color-blind Mode
- comboBox Theme Menu
- Volume Slider

3.3.1 Night Mode

Night mode lowers the level of brightness by shifting into a darker theme. By this way user experience in evening light is aimed to be better.

3.3.2 Color-blind Mode

Since everyone has a different way of perceiving the colors in the environment, My Little Quadrillion provides a wide variety of colour palettes. These palettes include plentiful options that a color-blind person can improve his/her vision.

3.3.3 Theme Combobox

My Little Quadrillion improves the user experience with the availability of themes that can be related to his/her mood or choices. This feature makes the gaming atmosphere more appealing. In addition, this feature leads to a more realistic playground.

3.3.4 Volume Slider

The sound in the game can either be supportive or disincentive. For this reason, this choice depends on the player. Hence, there is a volume slider.

4. Non-functional Requirements

4.1. Performance

Performance is one of the crucial requirements in games. Especially, delay and loss of view should not be allowed while playing a game. Also, there is a direct proportion between performance and response in a game. If the response runs fast, performance can also be fast and users can play with pleasure. To avoid from ordinariness, some of components such as theme, sound, level etc. will be added while developing a game. These components should not decrease the speed of the game response, otherwise practicability may also decrease for users.

4.2. User Friendly Interface

My Little Quadrillion game can be understandable and played easily because the path of usage will be designed for people of all ages. Simpler design can make game more easy. Also, “Tutorial” option will be added to get rid of getting lost in the game. Therefore, providing enough information to users makes My Little Quadrillion more user-friendly. There is no need to use many control keys, only mouse can be adequate for playing game. Although My Little Quadrillion seems easy at a first glance, it is an unfamiliar mind game which needs consideration. We will give a brief information about game history and who developed this game in “Credits” options. In other words, user interface is designed according to user comfortability and curiosity.

4.3. Compatible Interface with Real World

My Little Quadrillion is a box game normally and we convert this box game into the digital platform. We are going to make various options about theme of game to adapt in different places. To illustrate, night mode can be comfortable when playing dark places, and wooden table theme fells user like playing a game on the real table.

Furthermore, dragging pieces and putting desired place into the grid are similar to real life like playing with hands.

4.4. Extendibility

Extendibility is one of the important parts in software engineering to design the game modified later in future. Also, modifications can helpful to avoid boringness because if the game always have same levels or features, users can get bored very easily. That is because we are going to add more extensions for variety. Fundamentally, extendibility can provide the game sustainability.

4.4.1. Achievements

Achievements part is like a report card of player's experience and this part boosts user to obtain new medals or degree in a game. This part similar to prerequisites of lectures in real life. Players should pass desired levels to access further achievements.

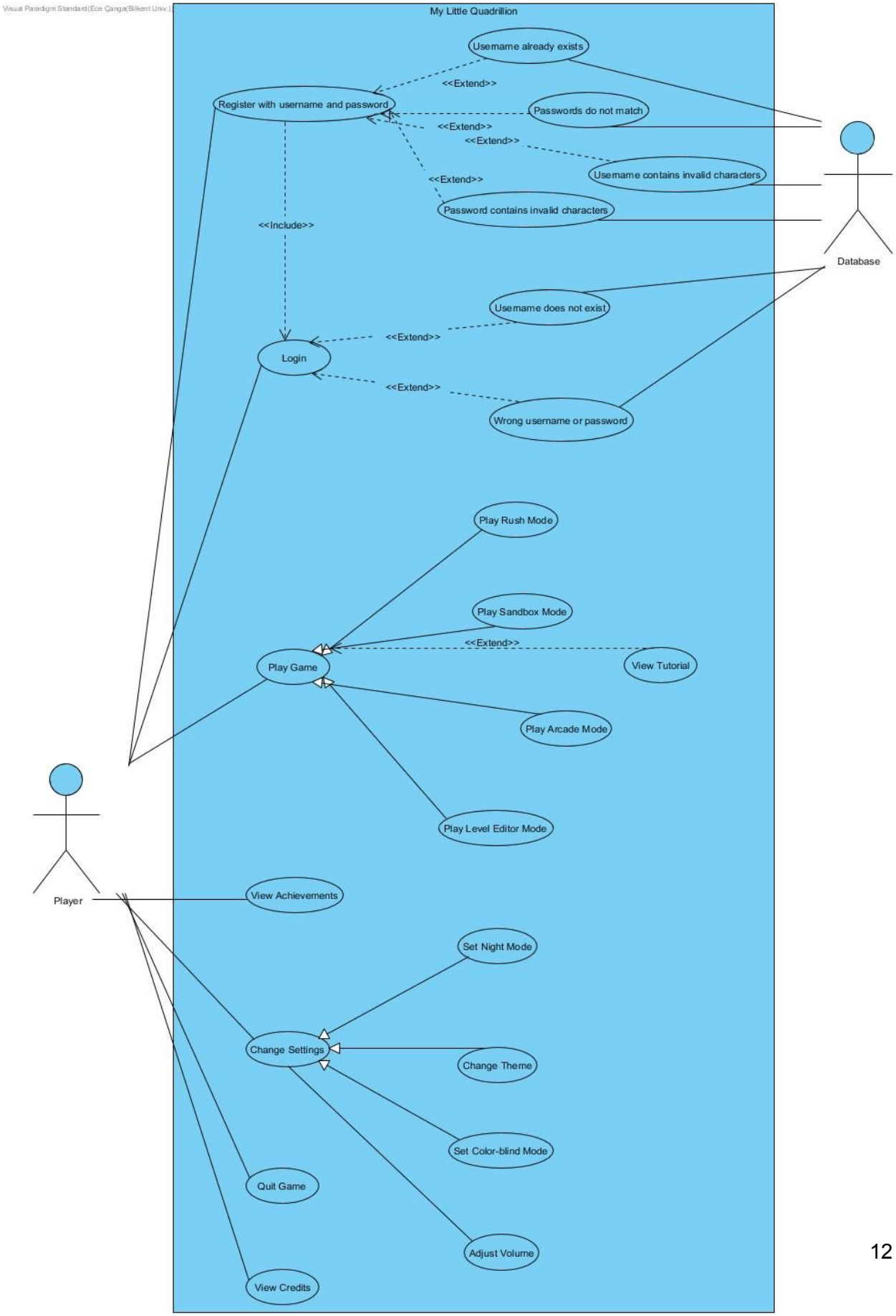
4.4.2. Challenges

Challenges can also be helpful to get rid of boringness because players have a tendency to enhance their game skills to achieve new challenges. To illustrate, players should think carefully with high concentration in the "Limited mode" and "3 Puzzles" mode. Also players should be quick and careful when the "3 Minutes" mode in Rush Mode. Therefore, different type of challenges would provide players to improve their different skills when they are playing. Being quick and think cleverly can be gained with experiences.

5. System Models

5.1. Use case model

The following page includes the Use Case Diagram followed by the Description of the Use case.
Remark about the Diagram. In the Diagram all of the use cases apart from Register Use Case have include relationship to “Login” Use Case.


Visual Paradigm Standard (Ece Çanga(Bilkent Univ.))
My Little Quadrillion
Username already exists
<<Extend>>
Register with username and password
Passwords do not match
<<Extend>>
<<Extend>>
Username contains invalid characters
<<Extend>>
Password contains invalid characters
<<Include>>
Database
Username does not exist
<<Extend>>
Login
<<Extend>>
Wrong username or password
Play Rush Mode
Play Sandbox Mode
<<Extend>>
View Tutorial
Play Game
Play Arcade Mode
Play Level Editor Mode
Player
View Achievements
Set Night Mode
Change Settings
Change Theme
Set Color-blind Mode
Quit Game
Adjust Volume
View Credits

Use Case 1

Use Case Name:	Register
Participating Actor:	Invoked by Player and Communicates with Database
Stakeholders and Interests:	Player wants to register System checks whether the password and the username are valid or not.
Entry condition:	Player should open the game and register screen.
Exit condition:	System makes the player registered and then logged in
Main flows of events:	Player enters username and password.
Alternative flows of events:	If player wants to return previous menu press the back arrow icon.

Use Case 2

Use Case Name:	Login
Participating Actor:	Invoked by Player and Communicates with Database
Stakeholders and Interests:	Player wants to log in. System checks whether the password and the username are matching or not.
Entry condition:	Player should open the game and login screen.
Exit condition:	System makes the player logged in.
Main flows of events:	Player enters username and password.
Alternative flows of events:	If player wants to return previous menu press the back arrow icon.

Use Case 3

Use Case Name:	Play Game
Participating Actor:	Invoked by Player
Stakeholders and Interests:	Player selects play game.
Entry condition:	Player should open the game and logged in.
Exit condition:	System displays game modes.
Main flows of events:	If player chooses rush mode game splits three categories: ● 3 Puzzle mode ○ System starts game ○ Player selects pieces and try to place them on grid ○ System updates timer ○ Player completes all puzzles one by one ○ System shows player's score ● 3 Minute mode ○ System starts game ○ Player selects pieces and try to place the on grid ○ System updates timer ○ Player completes a puzzle and the system brings a new puzzle ○ Game finishes when time is up ○ System shows player's score ● Restricted mode ○ System starts game ○ Player selects pieces and try to place them on grid ○ System updates number of moves left that can be played ○ Game finishes when no move left or puzzle is solved

	○ System shows player's score If player chooses sandbox mode: ● System starts game ● Player is free to add many grids and put pieces on the grid If player chooses arcade mode, game enables player to select available levels from different difficulties : ● Player selects a level ● System starts game ● Player tries to solve puzzle by moving pieces on grid ● Player may take hints ● Game finishes when the puzzle is solved ● System displays earned stars If player chooses level editor, game enables player to either play created puzzles or create a new one: ● Playing created puzzles ○ System starts game ○ Player tries to solve puzzle by moving pieces on grid ○ Player may take hints ○ Game finishes when the puzzle is solved ● Creating a new puzzle ○ System starts game ○ Player moves grids to create layout ○ System saves layout ○ Player places pieces that will be pre-placed on the grid or skip this step ○ System saves pieces if there is any ○ Player puts rest of the pieces onto grid ○ System saves puzzle and asks a name for the level ○ Player names the level

Alternative flows of events:	If player wants to return previous menu press the back arrow icon. If player wants to learn how to play game, press tutorial button. If player wants to pause the game, simply presses the “P” from keyboard.

Use Case 4

Use Case Name:	Achievements
Participating Actor:	Initiated by Player
Stakeholders and Interests:	Player wants to see his/her achievements
Entry condition:	Player should be logged in
Exit condition:	System displays player’s all achievements
Main flows of events:	Player can see both locked and unlocked achievements by scrolling mouse wheel.
Alternative flows of events:	If player wants to return previous menu press the back arrow icon.

Use Case 5

Use Case Name:	Change Settings
Participating Actor:	Invoked by Player
Stakeholders and Interests:	Player wants to change settings
Entry condition:	Player should open the game and logged in.
Exit condition:	System displays game settings
Main flows of events:	Player can change themes: ● Player clicks combo box ● Player selects one of the themes ● System changes the theme Player can activate night mode: ● Player turn on/off night mode ● System updates dark mode Player can activate colorblind mode: ● Player turn on/off colorblind mode ● System updates colors of the game Player can adjust volume of the game ● Player clicks and drags the volume bar ● System adjusts volume of the game
Alternative flows of events:	If player wants to return previous menu press the back arrow icon.

Use Case 6

Use Case Name:	Quit
Participating Actor:	Initiated by Player
Stakeholders and Interests:	Player wants to Quit the game.
Entry condition:	Player should be in game.
Exit condition:	Player quits the game
Main flows of events:	Game closes
Alternative flows of events:	-

Use Case 7

Use Case Name:	View Credits
Participating Actor:	Initiated by Player
Stakeholders and Interests:	Player wants to view credits
Entry condition:	Player should be in main menu
Exit condition:	System displays credits
Main flows of events:	-
Alternative flows of events:	If player wants to return previous menu press the back arrow icon.

5.2. Dynamic models

5.2.1. State diagrams

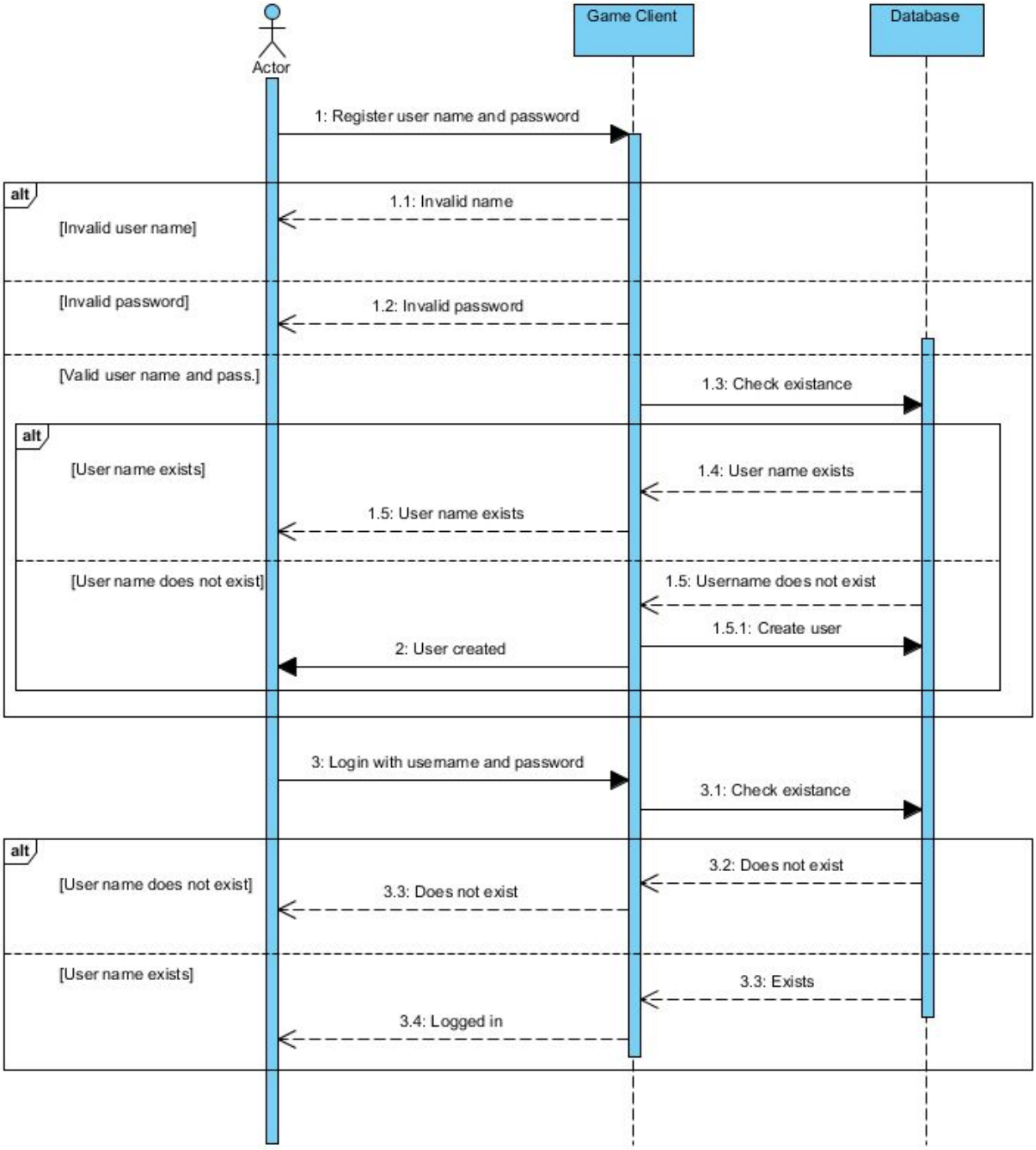
5.2.1.1. Level Editor State Diagram

In level editor, player first decides on where will he place the grids. If he places the grids in a valid pattern, adjacent to each other, program will allow him to proceed to the next phase. In piece placement phase player decides pre-placed pieces on the board. Player can place as many pieces as he wants to the game board. Then player advances to the next phase. In last phase user needs to solve the game. If he can, he is then allowed to save the level.

5.2.2. Sequence Diagrams

5.2.2.1. Login Sequence Diagram

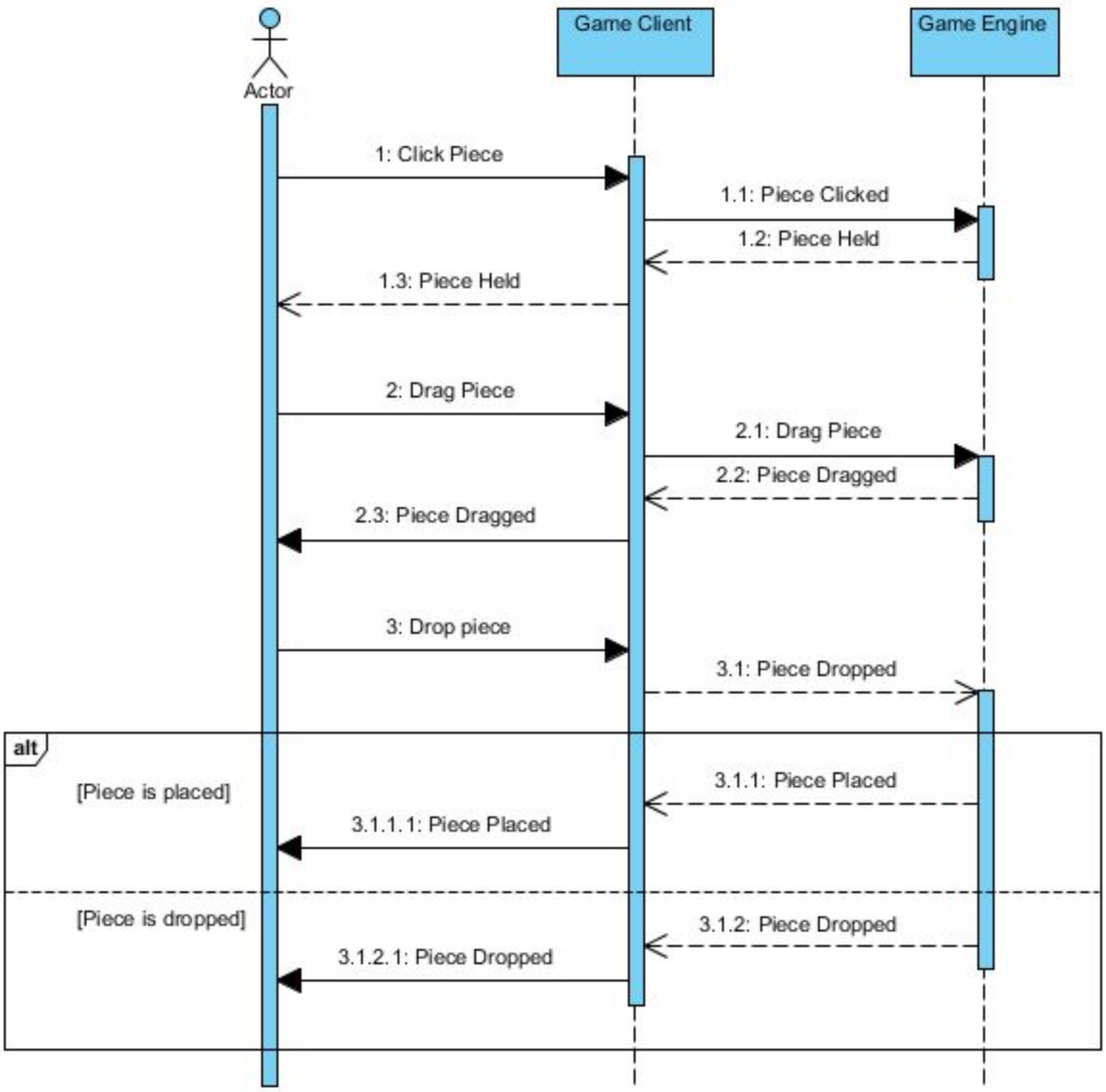
User registers to the game with a username and password. Then player can login to the game with these credentials. Register fails if an invalid username or password are entered. Login fails if an inexistant username and password combination is entered. When these errors occur, there will be a pop-up error message on the screen. If the valid name and password are entered in login or register page, user enters the game.



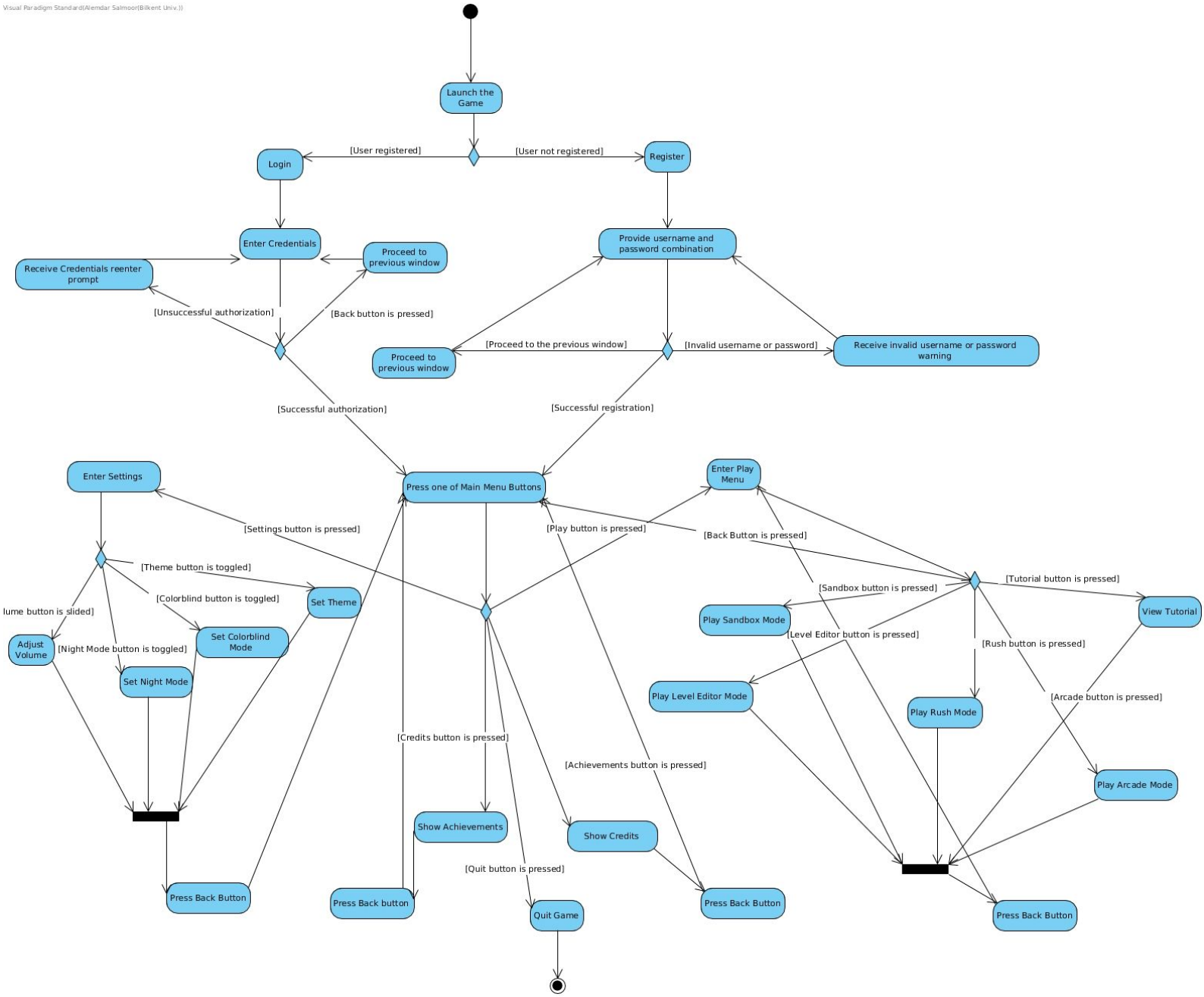
5.2.2.1 Game Play Sequence Diagram

In the playing game part, there are four different options which are Arcade Mode, Rush Mode, Level Editor and Sandbox are available but actions of game are same for all options. The basis of game is same for the template of game in all options. When

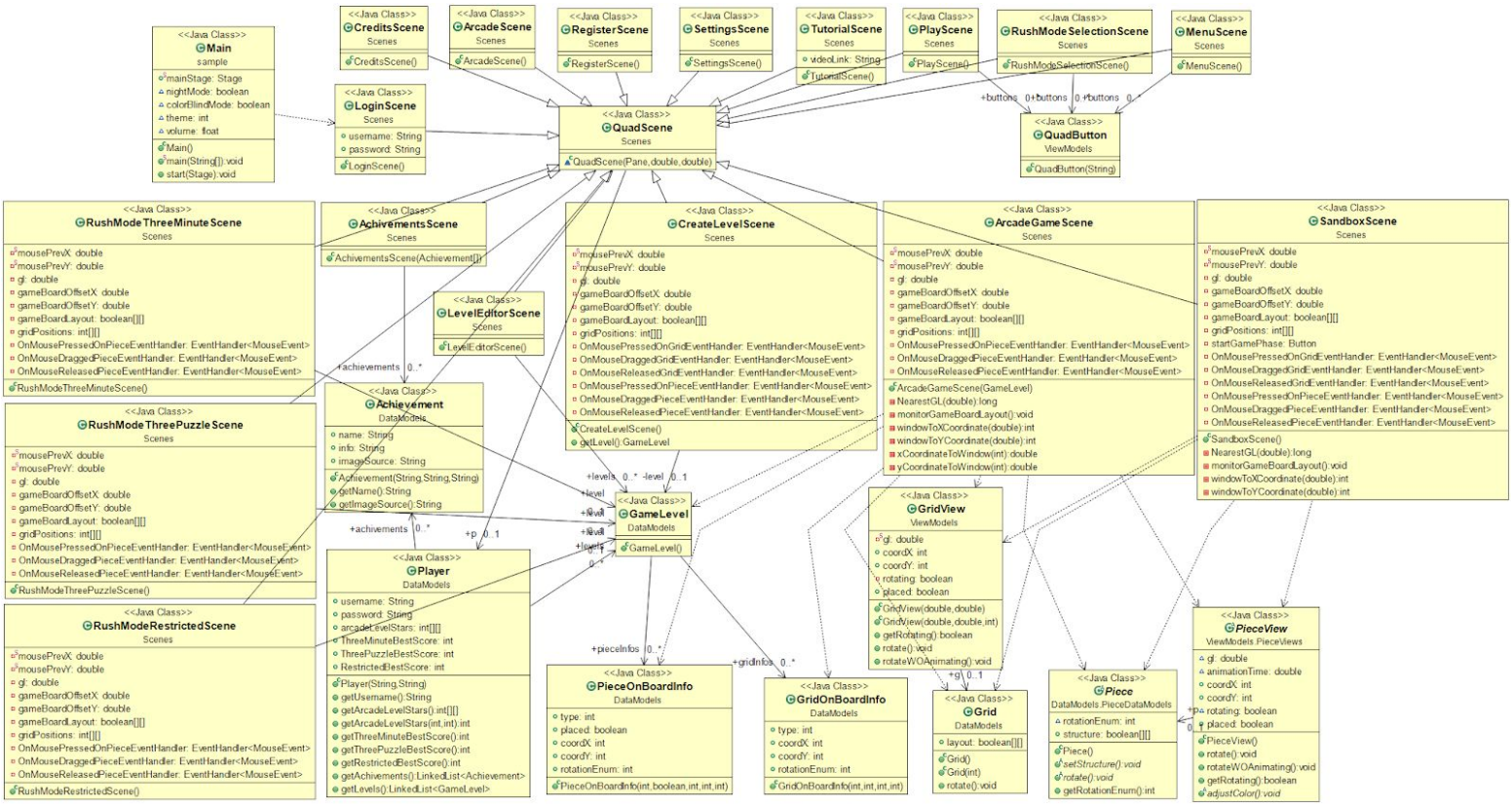
playing game users click pieces with mouse to hold and drag to the grids. If the grid spaces which is chosen are not available to put, piece automatically dropped in old place under the grid.



5.2.3. Activity Diagram



5.3. Object and class model



Class diagram is also added to docs root in the github

My Little Quadrillion currently has 29 classes as shown above diagram.

5.3.1. Achievement

Achievement is a java class which has users' degree and medals about game progress.

5.3.2. GameLevel

Game Level demonstrates the data about level number, name and models. It also includes gameLevel() method to shows all game levels.

5.3.3. Grid

Grid class contains 4x4 grid layout and rotate() method specifies how grid layout actually rotate.

5.3.4. GridView

GridView class contains design structure of the grid.

5.3.5. GridOnBoardInfo

GridOnBoardInfo class includes coordinates of the grid on the screen and also rotationEnum stores how many possible rotation moves can be applied on grid.

5.3.6. Player

Player class has game information for each different users. It has user's name, score and level data. Player class has this informations with getter methods and Player object.

5.3.7. Piece

Piece class defines pieces in the game and it contains number of possible rotations of the pieces also structure of the piece with boolean 2d array.

5.3.8. PieceView

PieceView class contains design structure of the piece.

5.3.9. PieceOnBoardInfo

PieceOnBoardInfo class includes coordinates of the piece on the screen and also rotationEnum stores how many possible rotation moves can be applied on piece. Boolean placed determines whether piece is placed on the grid or not.

5.3.10. QuadScene

This class has a responsibility about all view of the game because it includes all game view subclasses and make a relationship between all views during the playing. QuadScene class is subclass of Scene class of javafx.

5.3.11. AchivementsScene

AchievementScene has method which has Achievement Array as a parameter to illustrate the achievement list in this view.

5.3.12. ArcadeGameScene

This class is a subclass of QuadScene and shows the inside of the game view and includes actions and move features and coordinates as a method. Also this class has ArcadeGameScene method which has GameLevel parameter.

5.3.13. ArcadeScene

This class is a subclass of QuadScene and demonstrates the arcade menu view.

5.3.14.CreateLevelScene

This class is a subclass of QuadScene and has methods to create and make directions about using mouse listeners. This class enable game to move all pieces grid and directions by users. Also for creating a unique level, createLevelScene() and getLevel() are included.

5.3.15. CreditsScene

This class is a subclass of QuadScene and shows the credit page.

5.3.16. LevelEditorScene

This class is a sub-class of QuadScene and shows the created games list by user.

5.3.17. LoginScene:

This class is a sub-class of QuadScene and it contains username and password that is used by user to login to the game.

5.3.18. MenuScene

This class is a subclass of QuadScene and illustrates the menu lists with menuScene().

5.3.19. PlayScene

This class is a subclass of QuadScene and it demonstrates the scene that contains possible game modes of the game.

5.3.20. RegisterScene

This class is a subclass of QuadScene and contains registration scene of the game.

5.3.21. RushModeRestrictedScene

This class is a subclass of QuadScene and includes the restricted scene games view. It has coordinates, mouse actions and RushModeRestrictedScene() method for game view.

5.3.22. RushModeSelectionScene

This class is a subclass of QuadScene and make a view about rush mode game lists.

5.3.23. RushModeThreeMinuteScene

This class is a subclass of QuadScene and creates a view about three minutes game view and features.

5.4.24. RushModeThreePuzzleScene

This class is a subclass of QuadScene and creates a view about three puzzle game view and features.

5.3.25. SandboxScene

This class is a subclass of QuadScene and creates a view about sandbox game view and features.

5.3.26. SettingsScene

This class is a subclass of QuadScene and it contains theme, volume, colorblind and light options.

5.3.27. TutorialScene

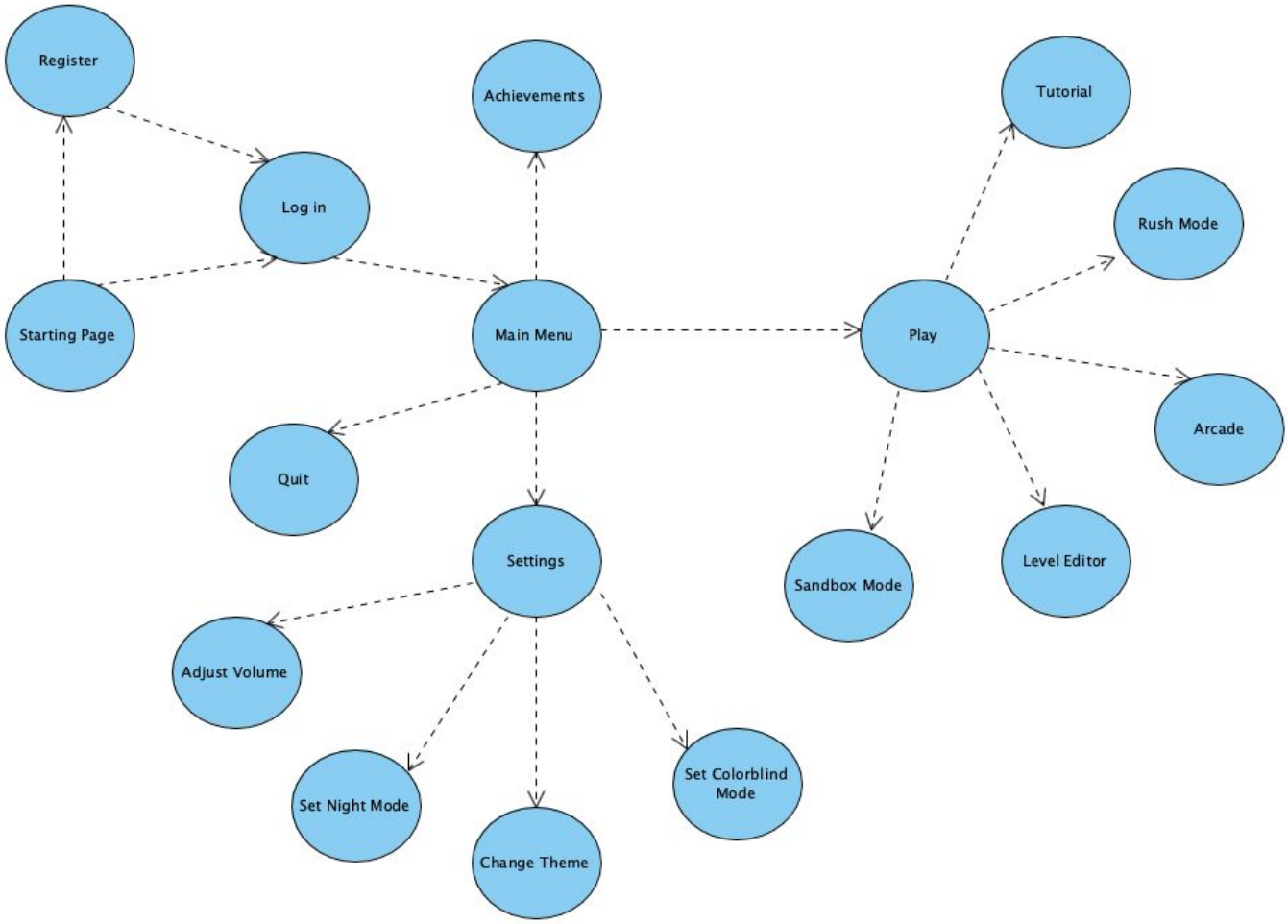
This class is a subclass of QuadScene and illustrates the how to play view before the game.

5.3.28. QuadButton

This class contains functionalities of the all buttons in the game, and QuadButton is subclass of the Button class of javafx.

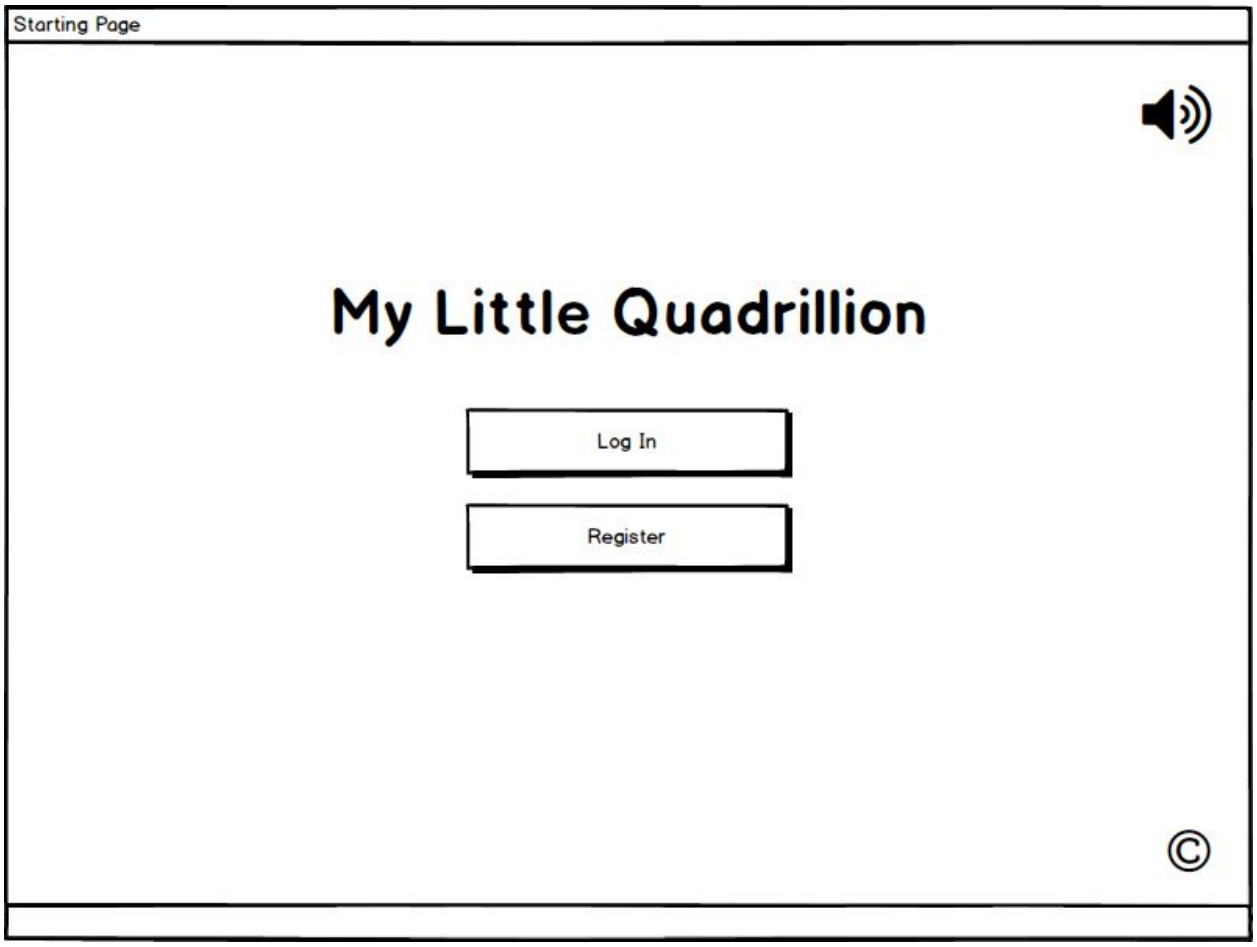
5.4. User interface

5.4.1. Navigational Path



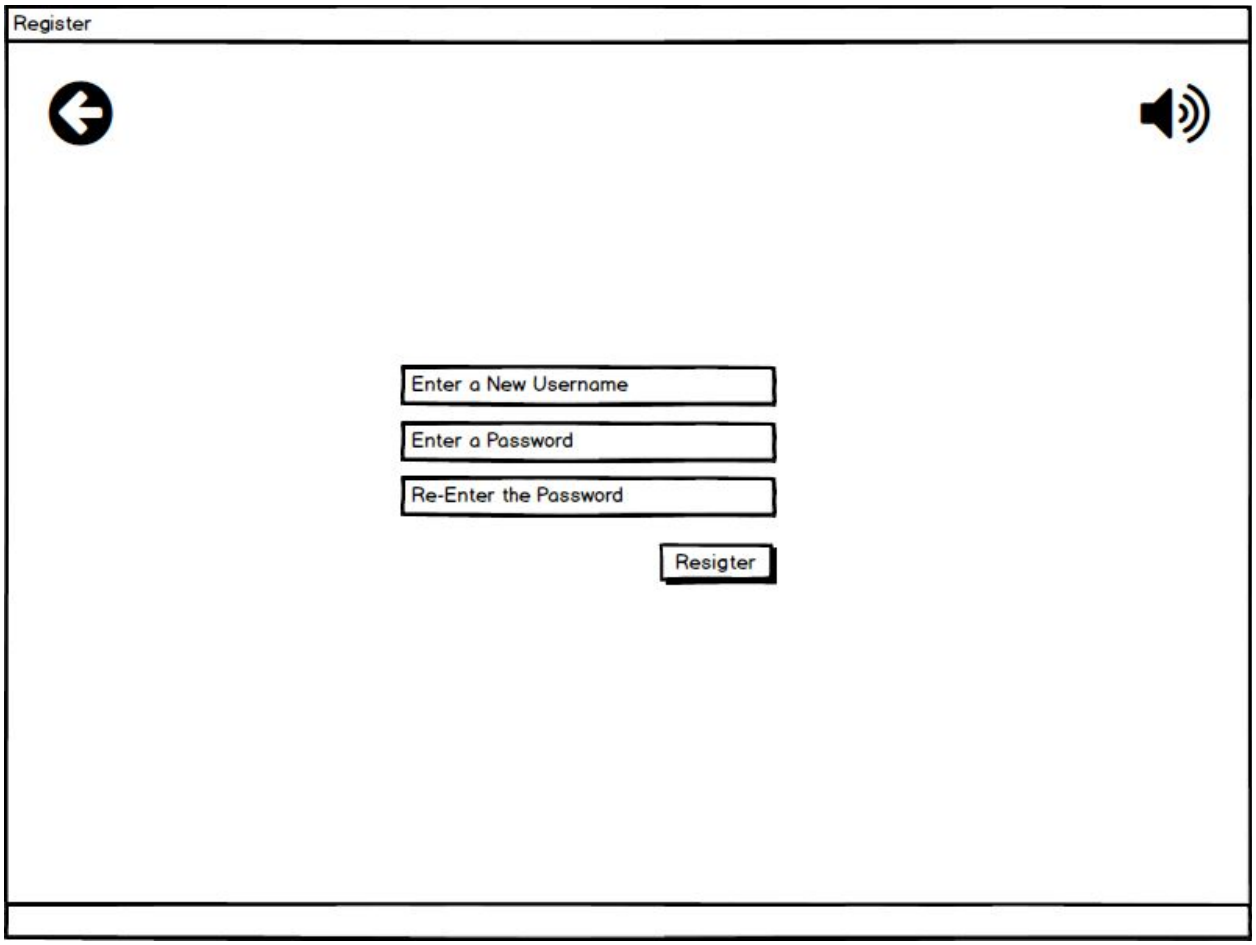
5.4.2. Screen Mock-ups

5.4.2.1. Starting Page

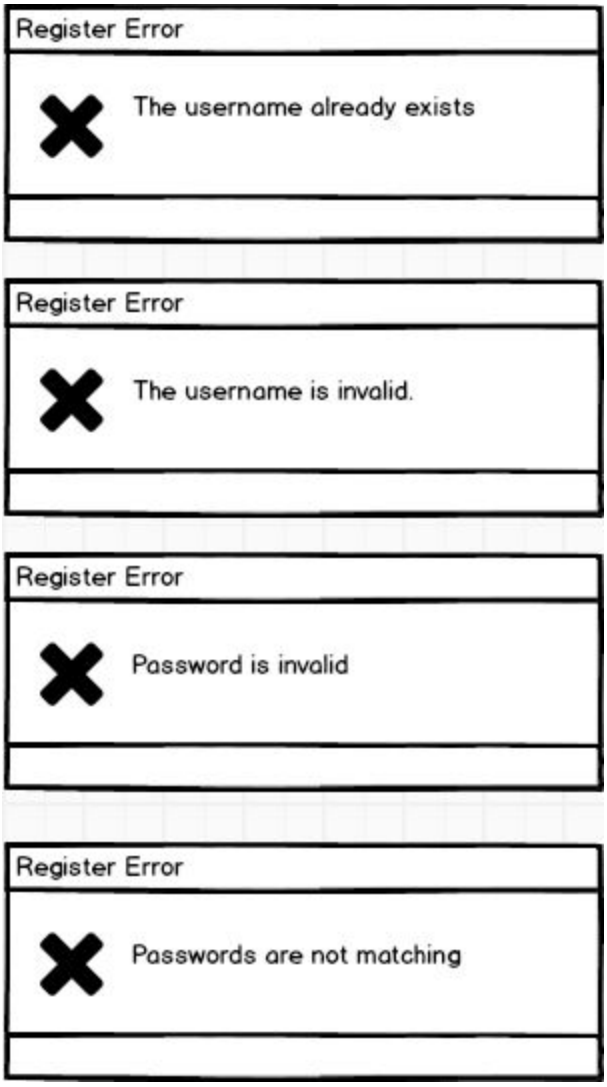


This is a starting page, it includes log in, register, sound and credits buttons. Volume button can stop or continue the sound. Credits button can give information about game and developers. Log in button for existing users and register button is for new players to enter a game.

5.4.2.2. Register Page

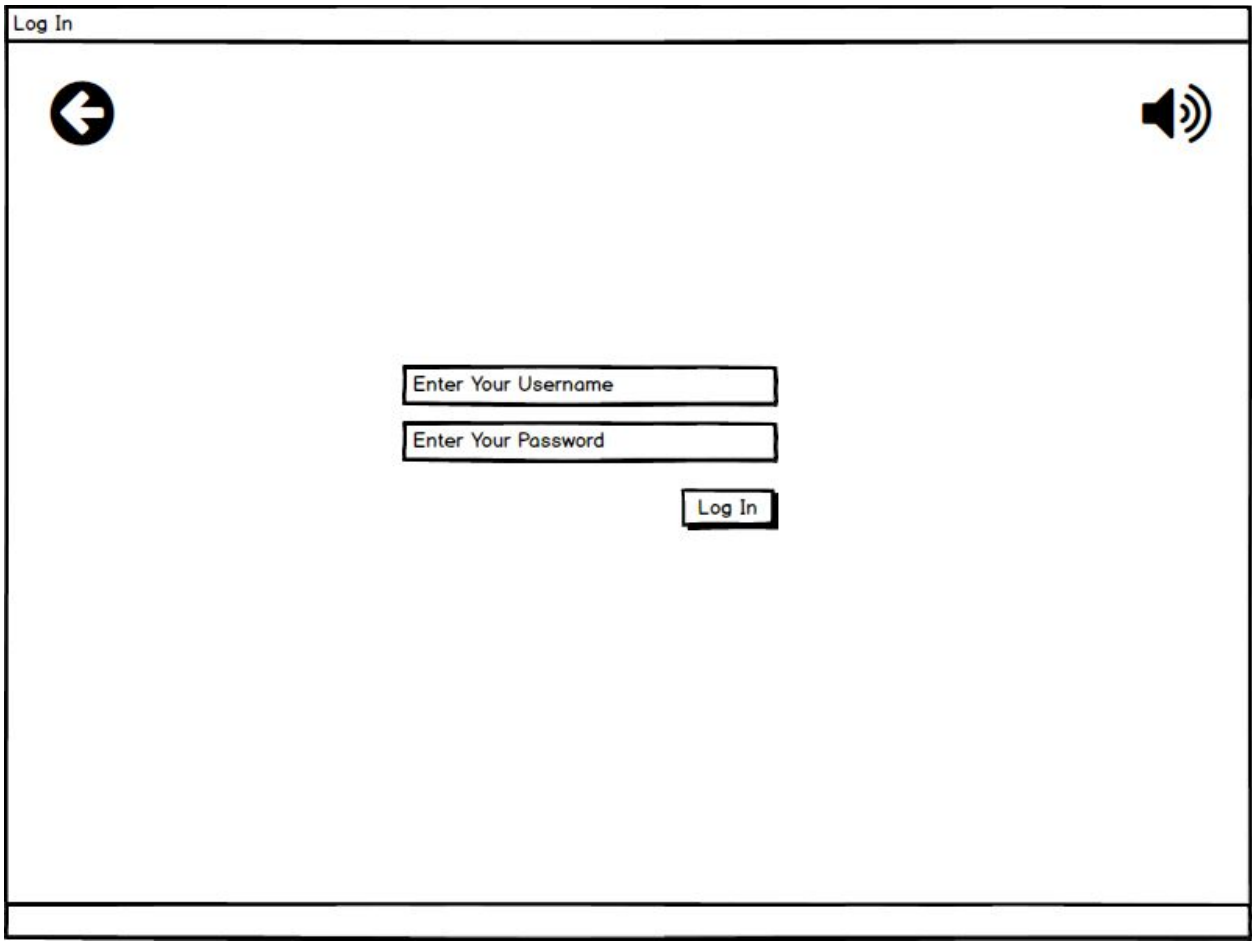


This is a Registration Page of the Game. User can register using the provided boxes for username password combination. To ensure unintentional wrong password entry, the page includes two fields for password, after the completion of provided fields unique user can be created by pressing the register button. The page includes volume button and back button.

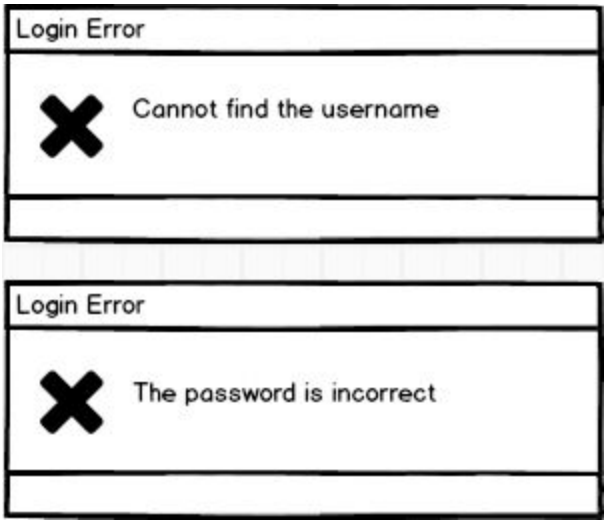


These are the error warning windows that can be generated on the register page. First error is shown if the entered username already exists. Second error is generated if username entered contains invalid characters. Third error is generated if password contains invalid characters. Last error is generated if the re-entered password does not match with the original password.

5.4.2.3 Login Page

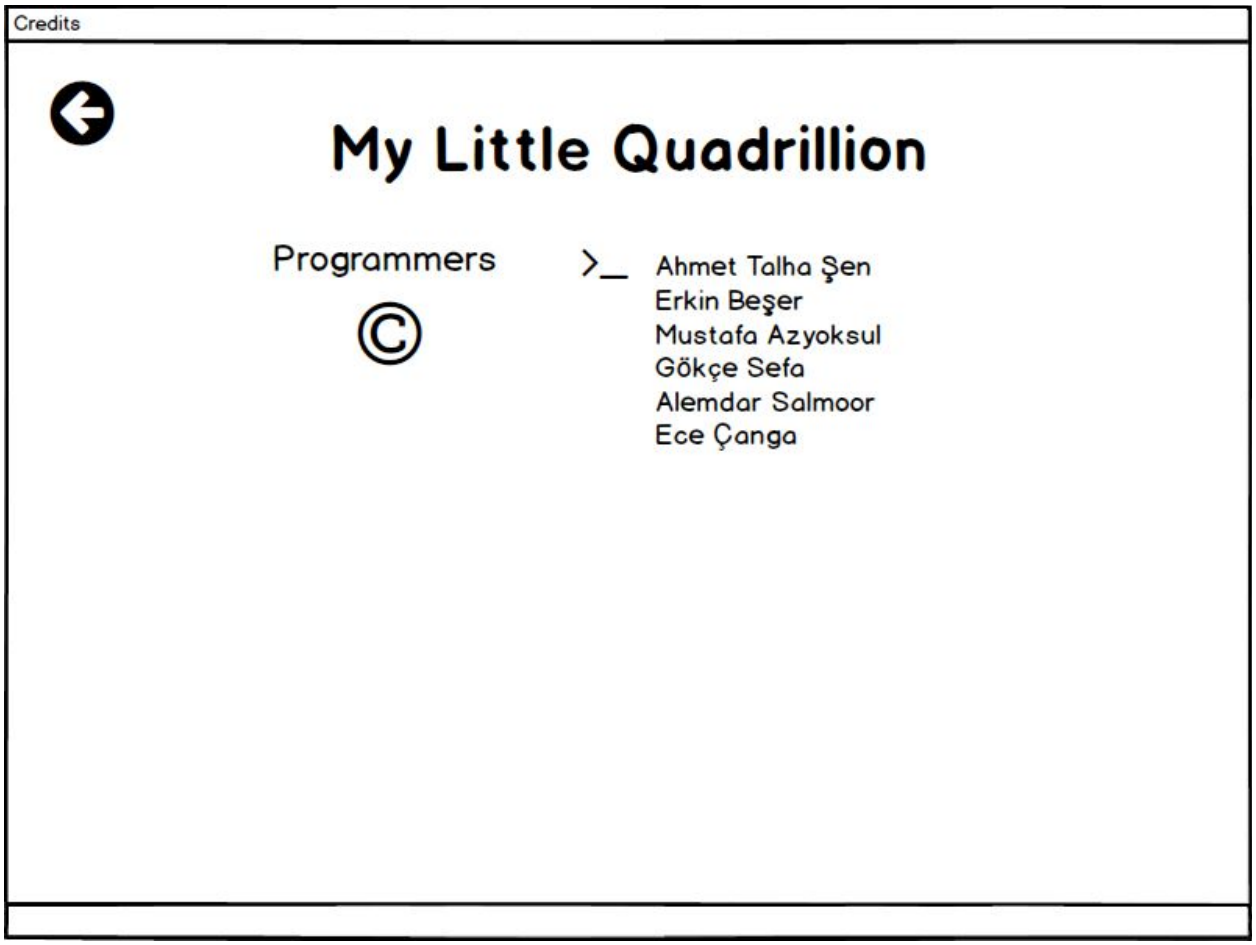


Player enters this page to log in to the game. After providing username and password player presses the login button to authorize. The page includes volume button and back button.



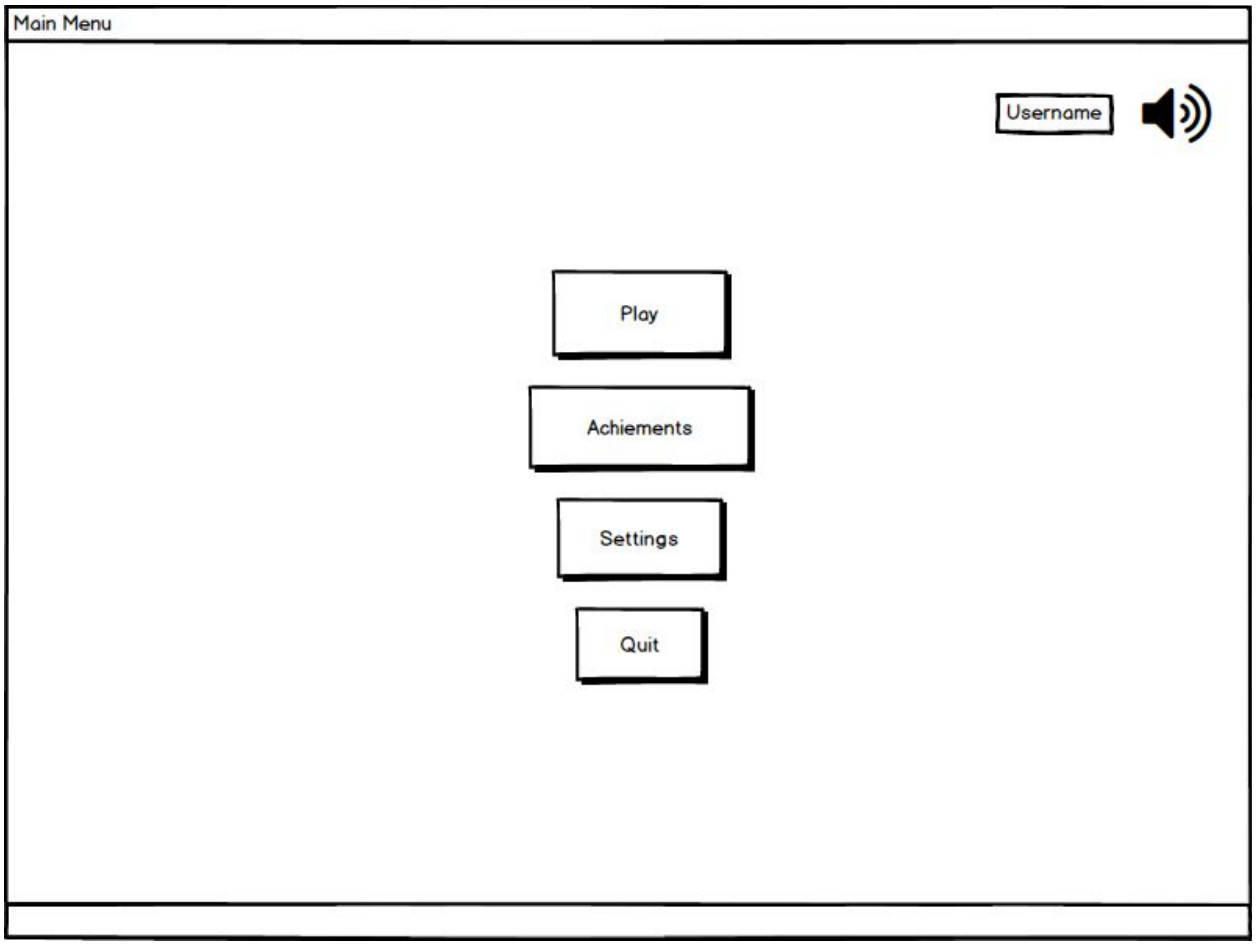
The first error window is shown when the entered username is not found in the database. The second error is shown if the username is found but the provided password does not match the username.

5.4.2.4 Credits Page



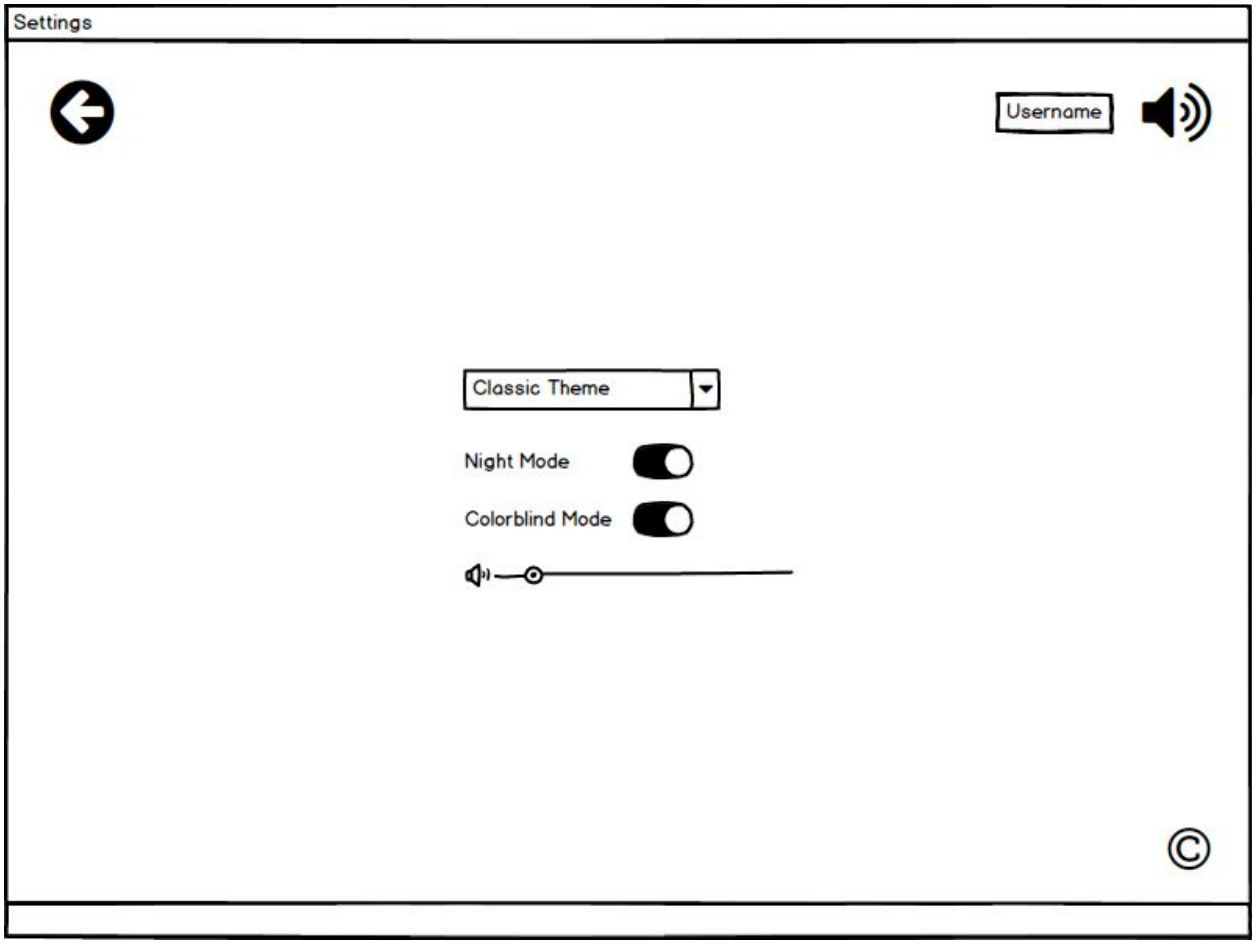
The Credits page includes information about the developers of My LIttle Quadrillion game.
The page includes back button that return to the Main Menu.

5.4.2.5 Main Menu Page



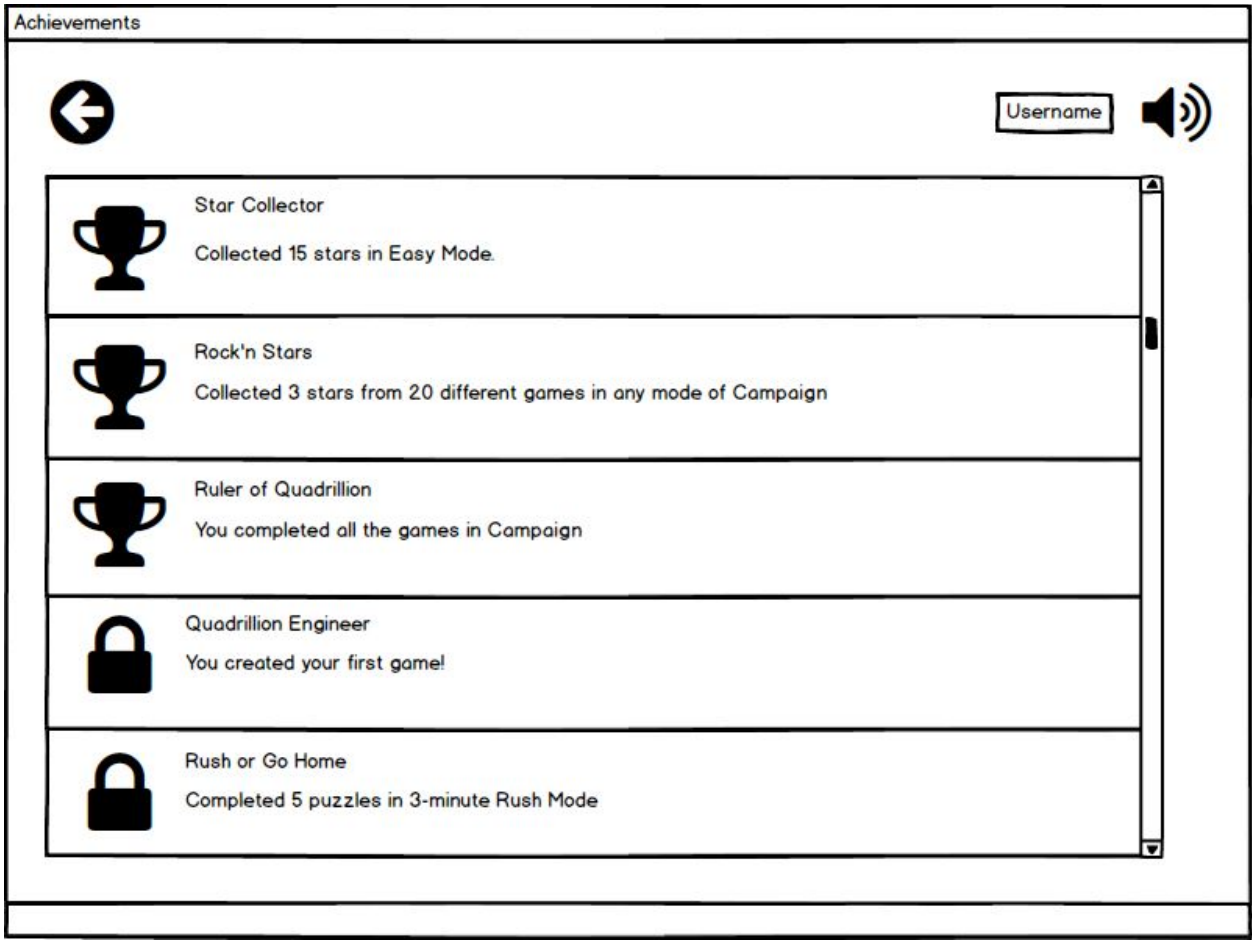
This is the Main Menu page. Player can go the Play manu, Achievements, and Settings. Player can also Quit the game if they wish to. Pressing sound button silences the sound.

5.4.2.6 Settings Page



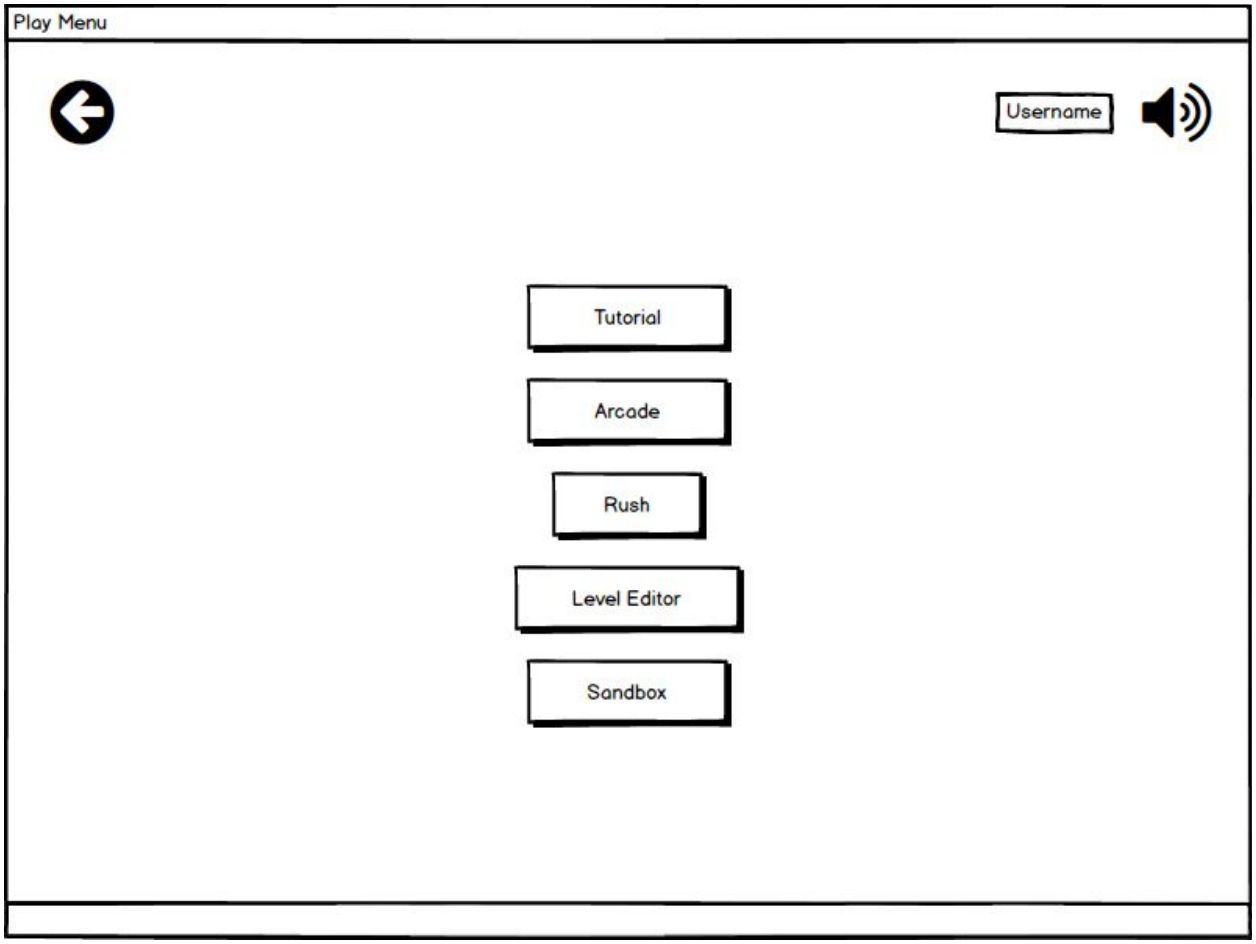
This is the Settings Page. Player can change the theme of the game such as Wooden, Christmas, Summer. Player can toggle Night Mode state of the app that provides comfortable color palette for night time use of the app. Player can also enable Colorblind Mode. The back button takes the Use to the Main Menu.

5.4.2.7 Achievements Page



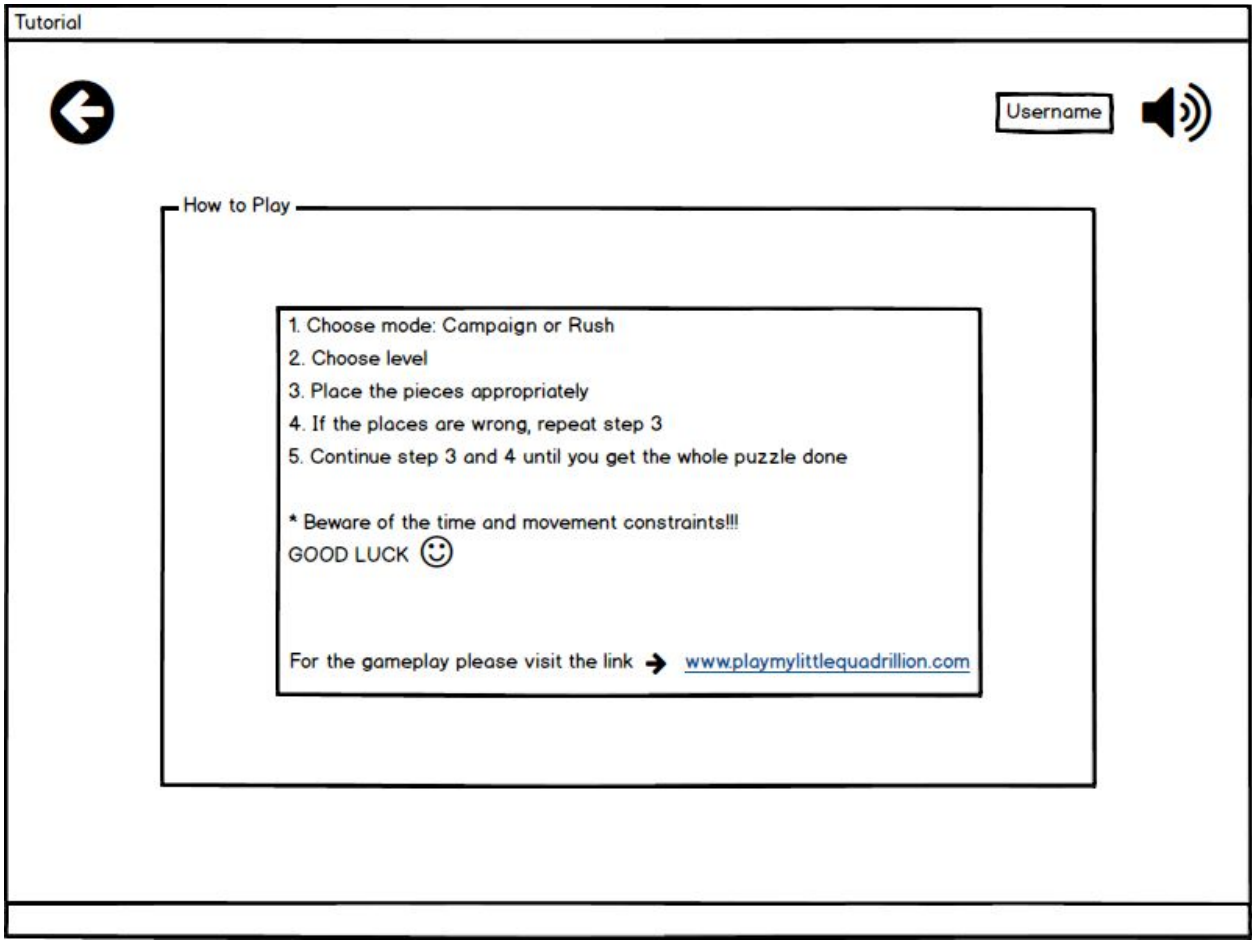
This is the page that displays the achievements of the player that they were able to collect from different modes of the game. Back button takes the player to the Main Menu.

5.4.2.8. Play Menu Page



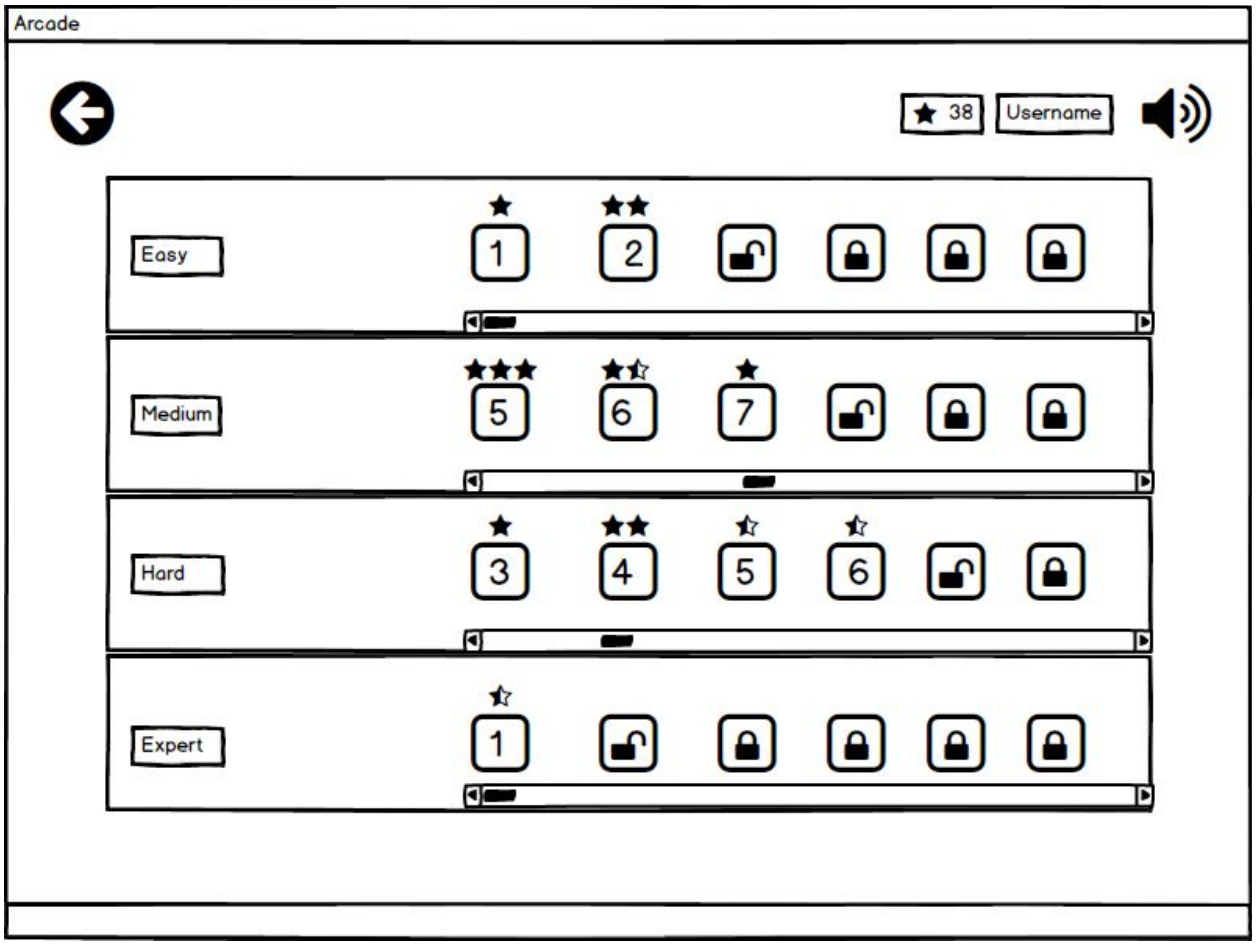
This is the Play Menu. Player proceeds here from Main Menu when he presses the Play button.
Player can initiate different game modes from this page such as Arcade, Rush, Level Editor, Sandbox. User can also View Tutorial from this Menu. Back Button takes the player to Main Menu.

5.4.2.9 Tutorial Page



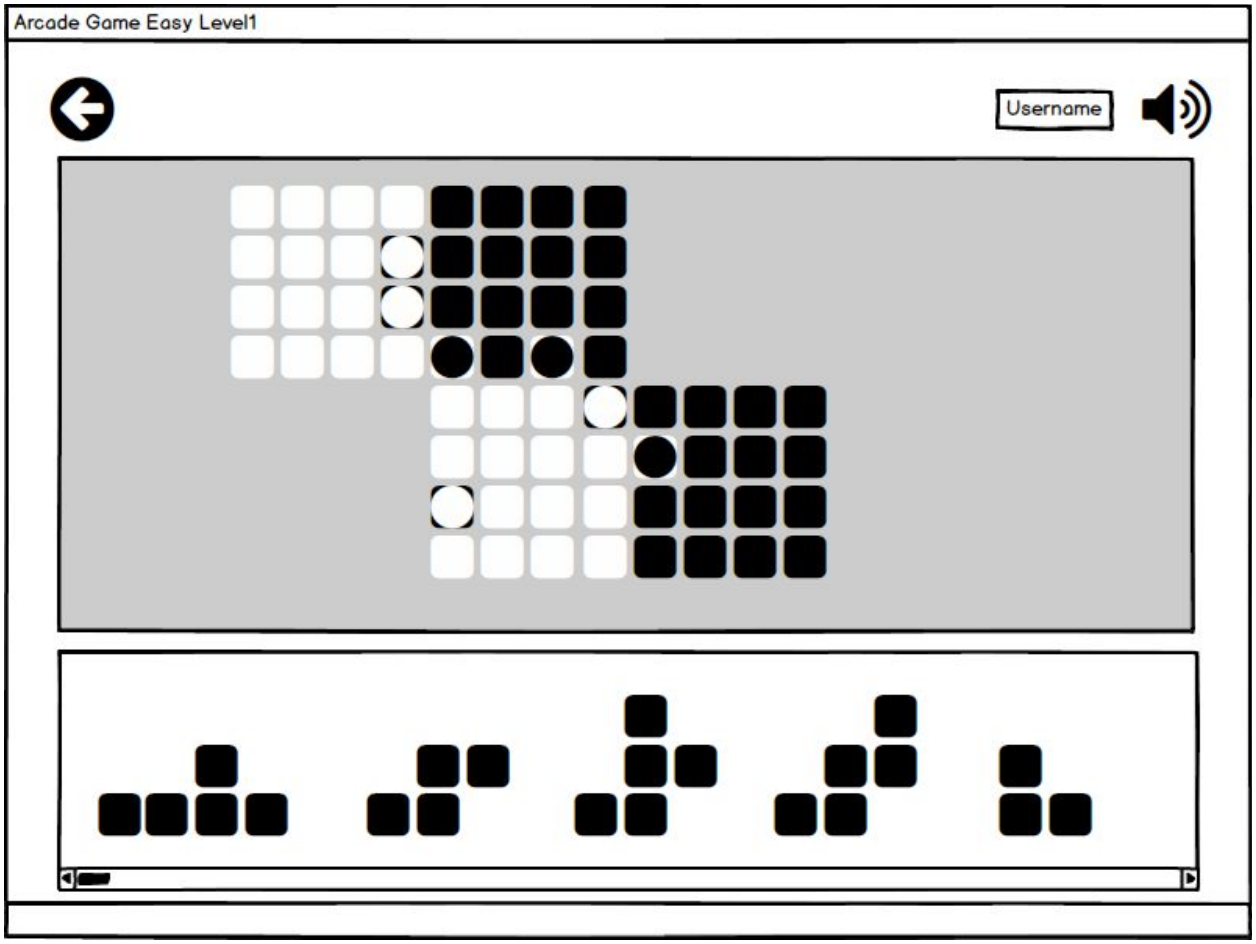
This page includes as information that can assist the player in Learning the Game. The page includes short algorithm to play the game, however if it is insufficient and player wishes to watch gameplay they can follow the link that takes them to the game demonstration. Back Button takes the user to Play Menu.

5.4.2.10. Arcade Page



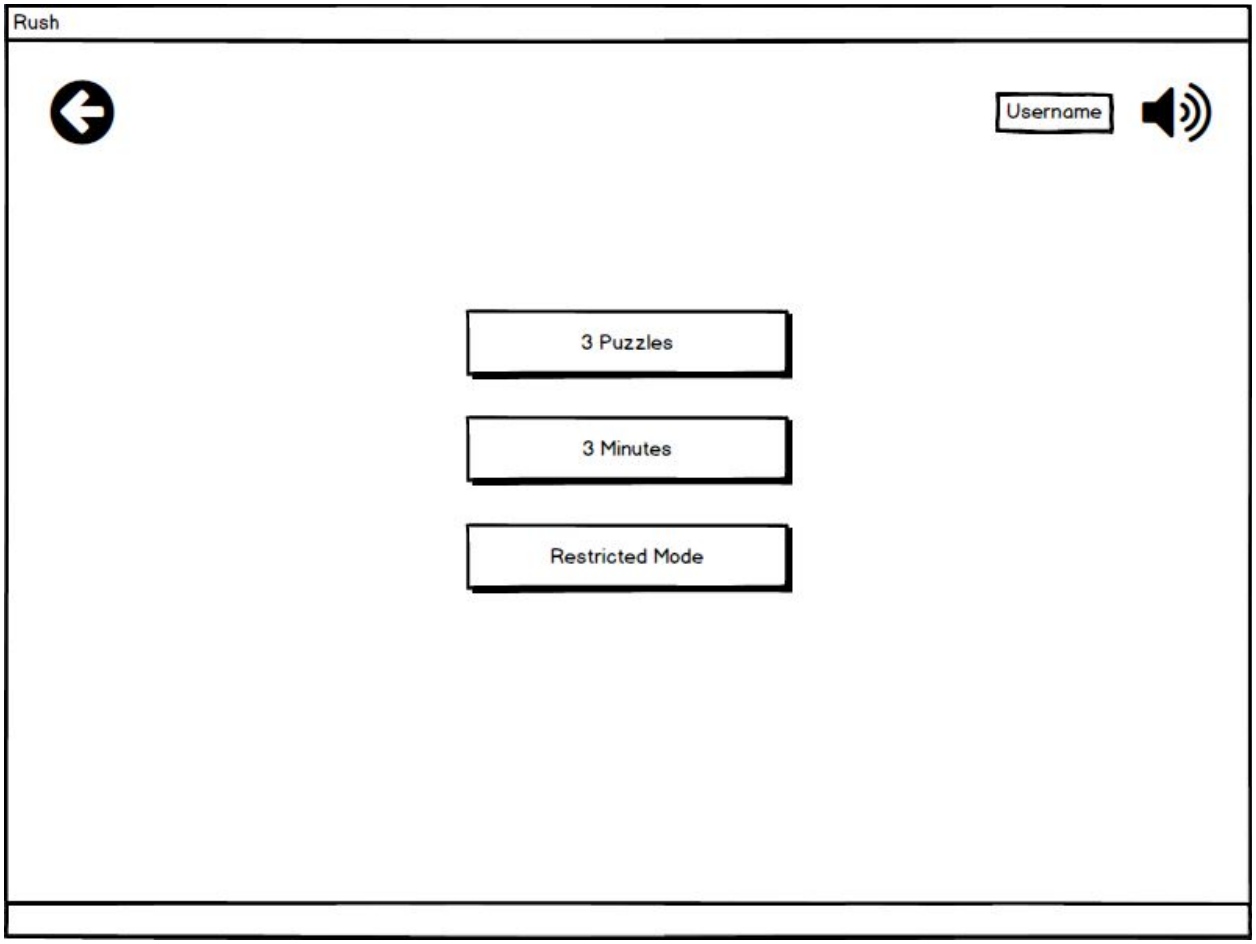
This is the page that is shown to the user when he presses Arcade from Play menu. Player can continue in any one of the difficulty levels they left off. Locked levels are indicated by locked lock sign. Unlocked and passed levels are indicated with level numbers with stars above them that correspond to the success of the player at this level. Unlocked and untried levels are indicated by unlocked lock sign. The page also includes the stars the user was able to collect in Arcade Mode. Back button takes the player to Play menu.

5.4.2.11. Arcade Game Easy Level 1 Page



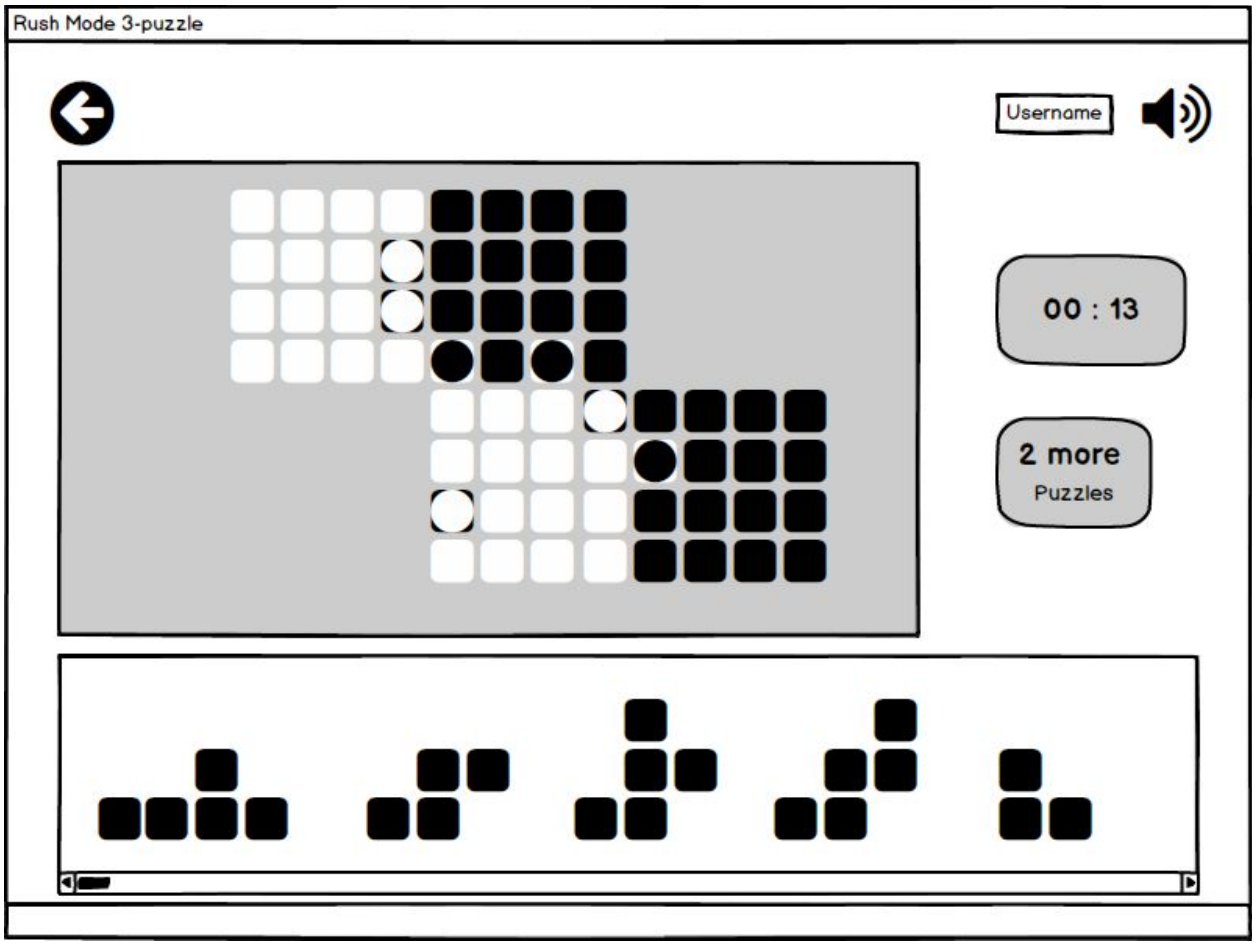
This is an example of easy Arcade Mode difficulty level. Player is challenged to solve the puzzle. If the player succeed, system gets the player back to arcade page and updates the progress and shows the stars collected by the player for this level. Player can go back by pressing left arrow key but the game is not saved. The Back Button takes the user to Arcade Mode where he can choose difficulty level of the puzzle.

5.4.2.12. Rush Game Menu Page



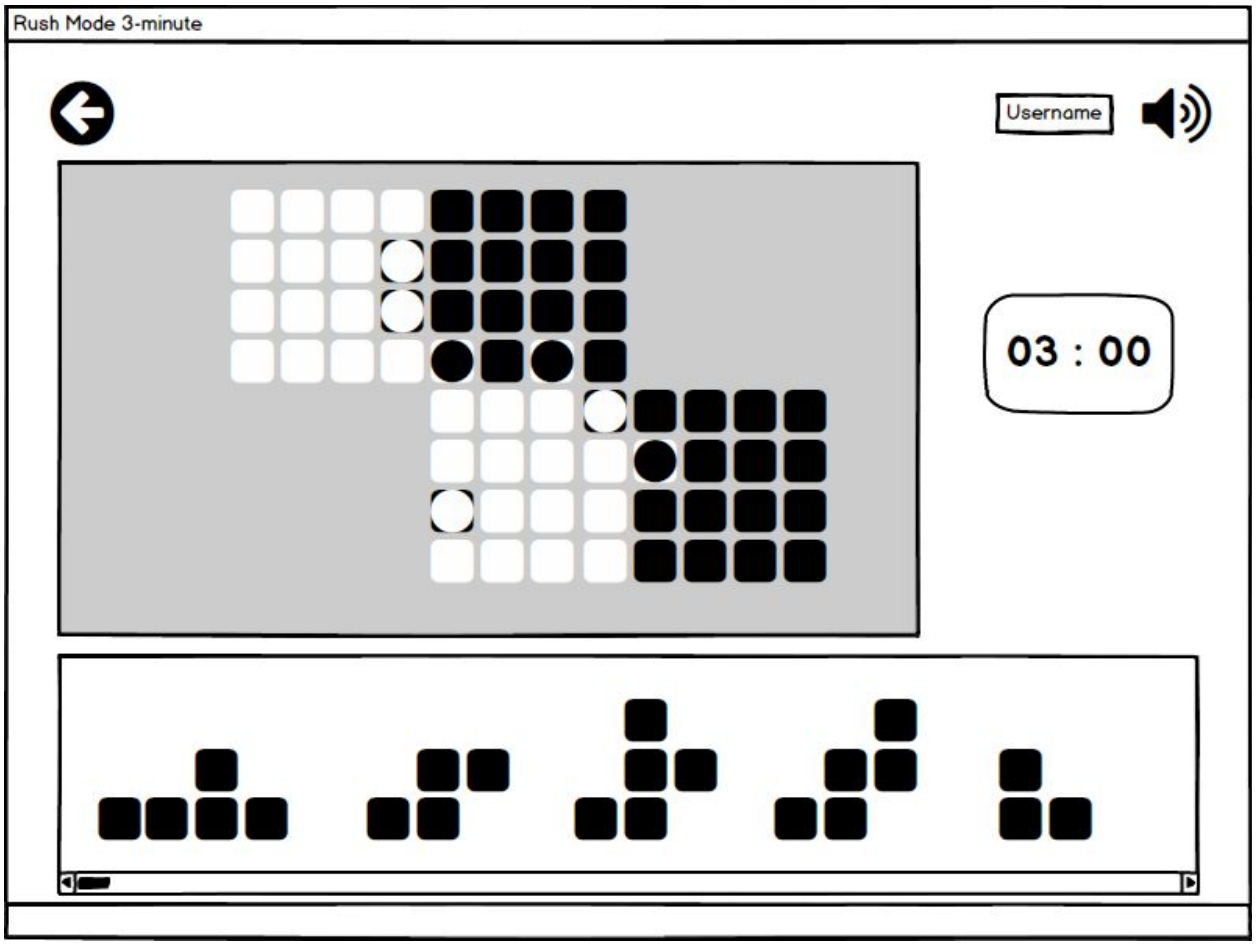
This is the Rush Mode page. User can choose which of the Rush Mode game types to play. Back button takes the user to Play Menu.

5.4.2.13. Rush Mode 3-puzzle Page



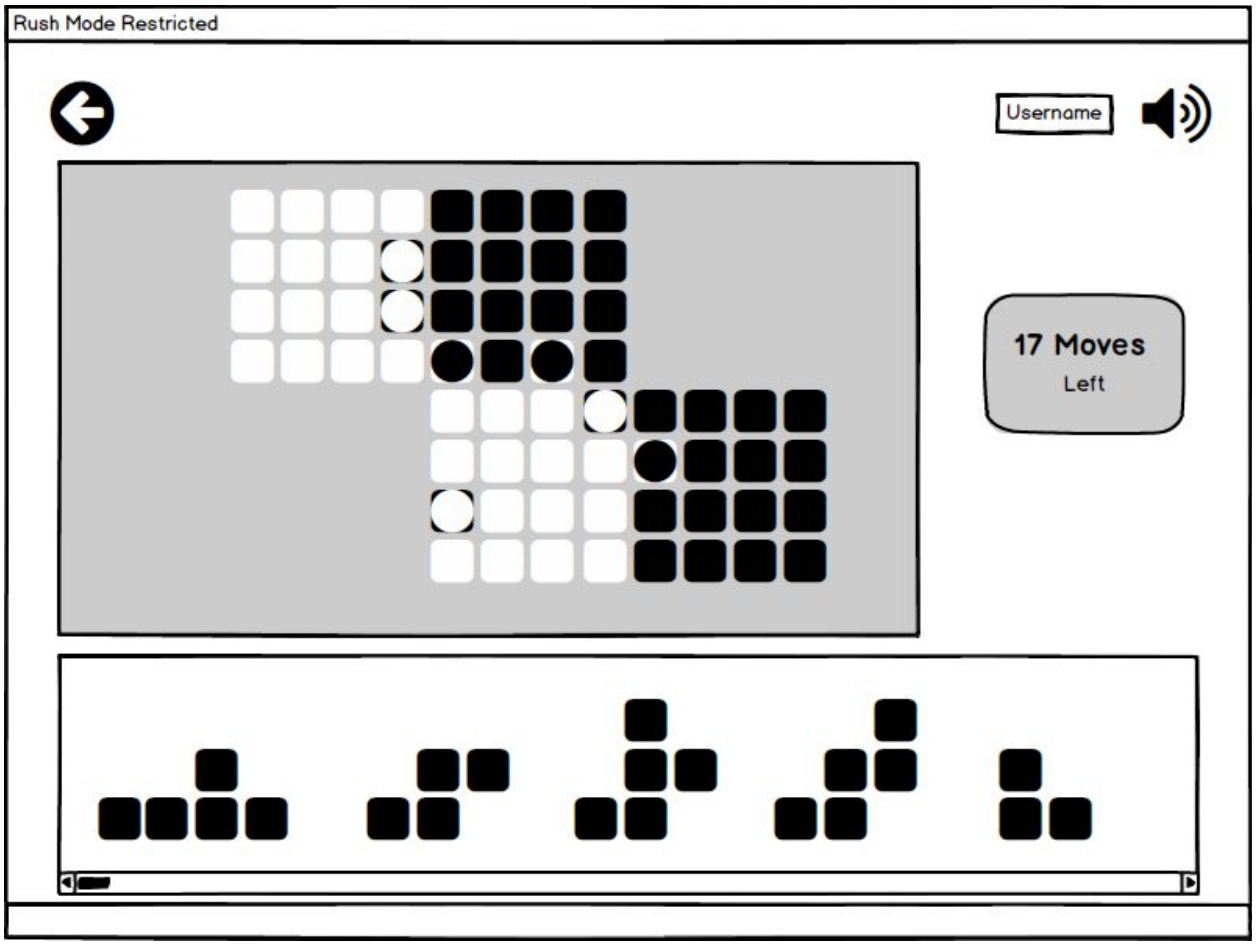
This page is shown to the user if he enters 3-puzzle Rush Mode. The page includes time that has passed from the start of the game. Additionally, player can see how many puzzles are left out of three puzzles. Back Button takes the player to Rush Mode Page where he can choose which type of Rush Mode they would like to play.

5.4.2.14. Rush Mode 3-minute Page



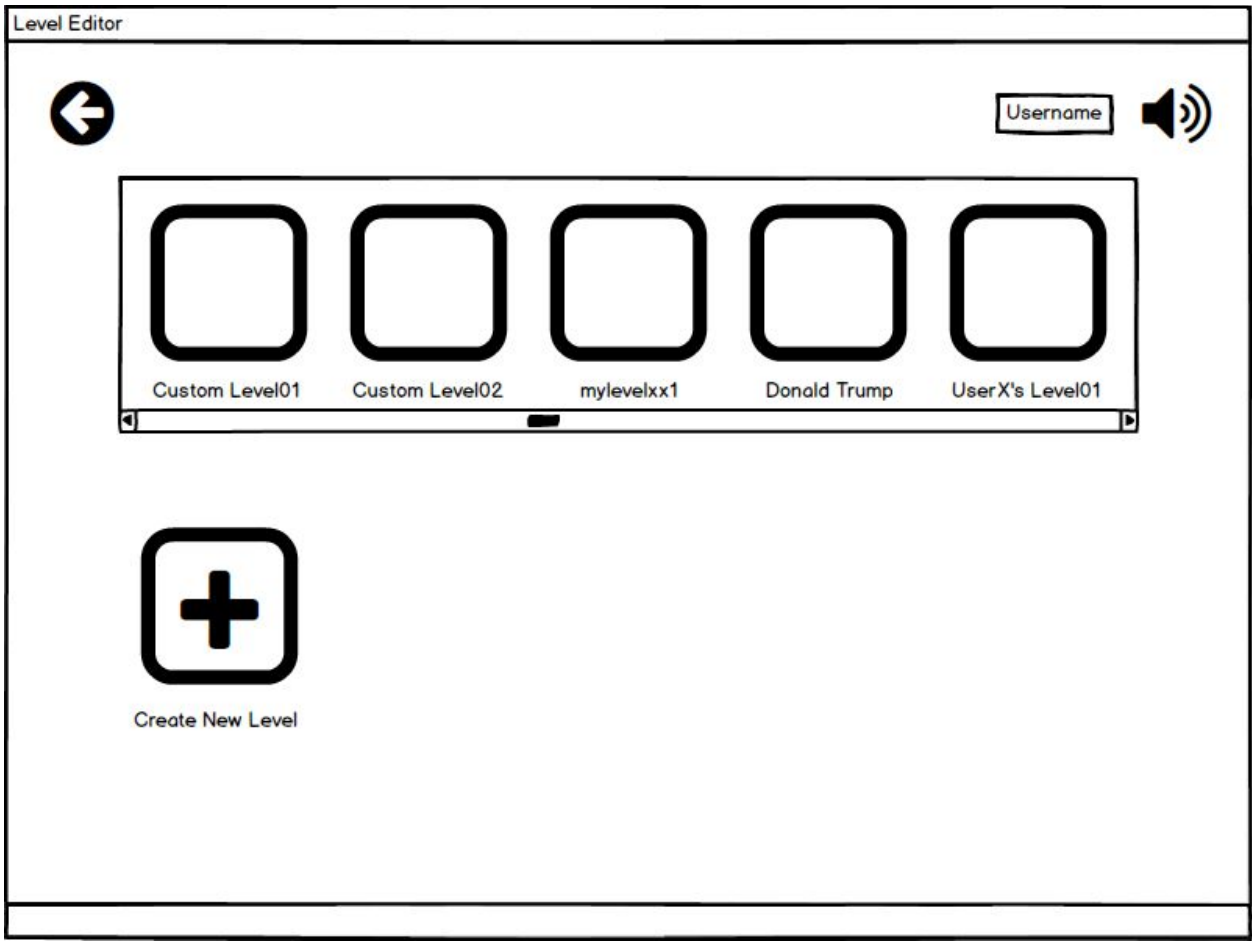
This page is shown to the user if he enters 3-minute Rush Mode. The page includes time that has left until the end of the game. Back Button takes the player to Rush Mode Page where he can choose which type of Rush Mode they would like to play.

5.4.2.15. Rush Mode Restricted Page



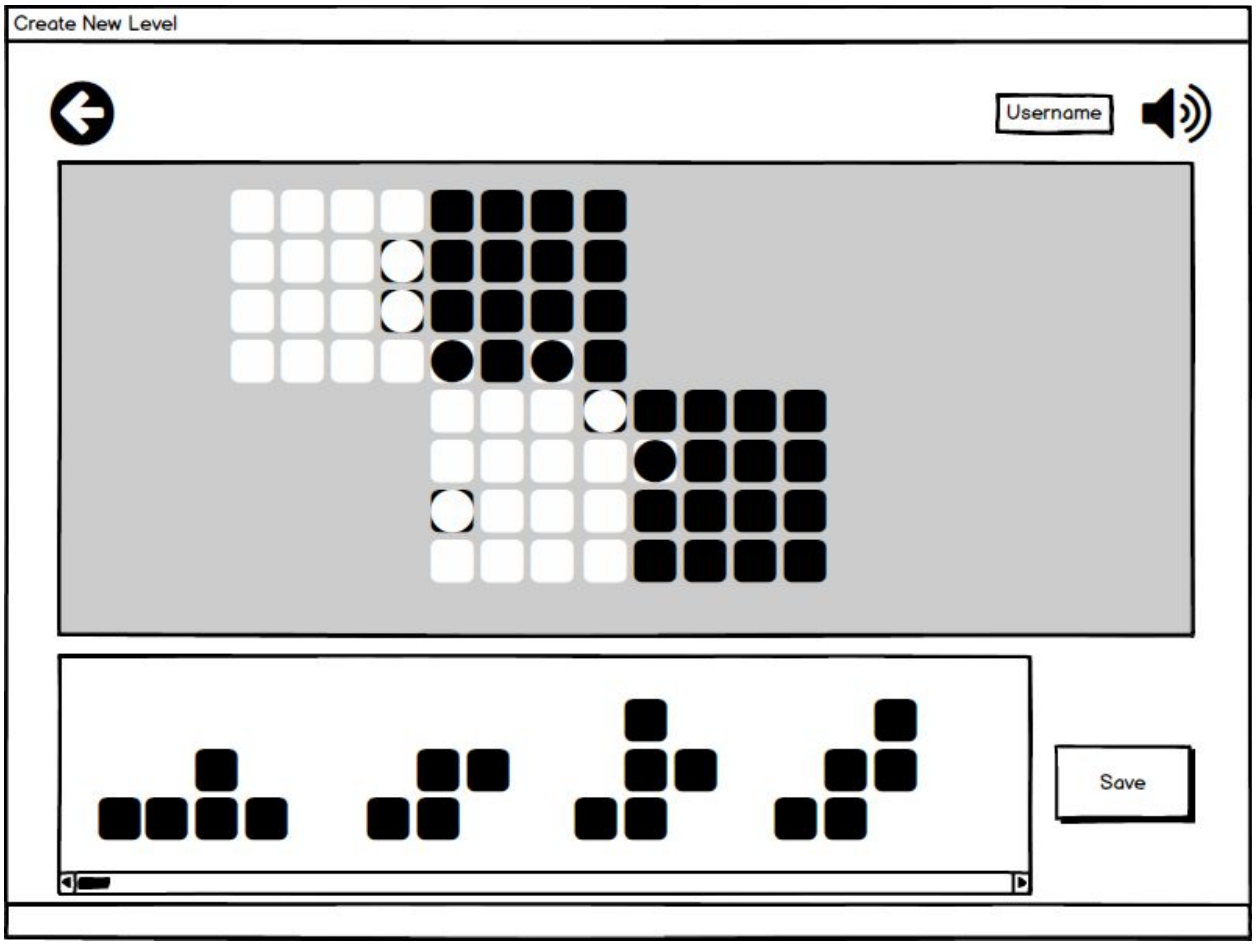
This page is shown to the user if he enters Restricted Rush Mode. Player can see how many disposable moves are left in the Restricted Mode . Back Button takes the player to Rush Mode Page where he can choose which type of Rush Mode they would like to play.

5.4.2.16. Level Editor Page



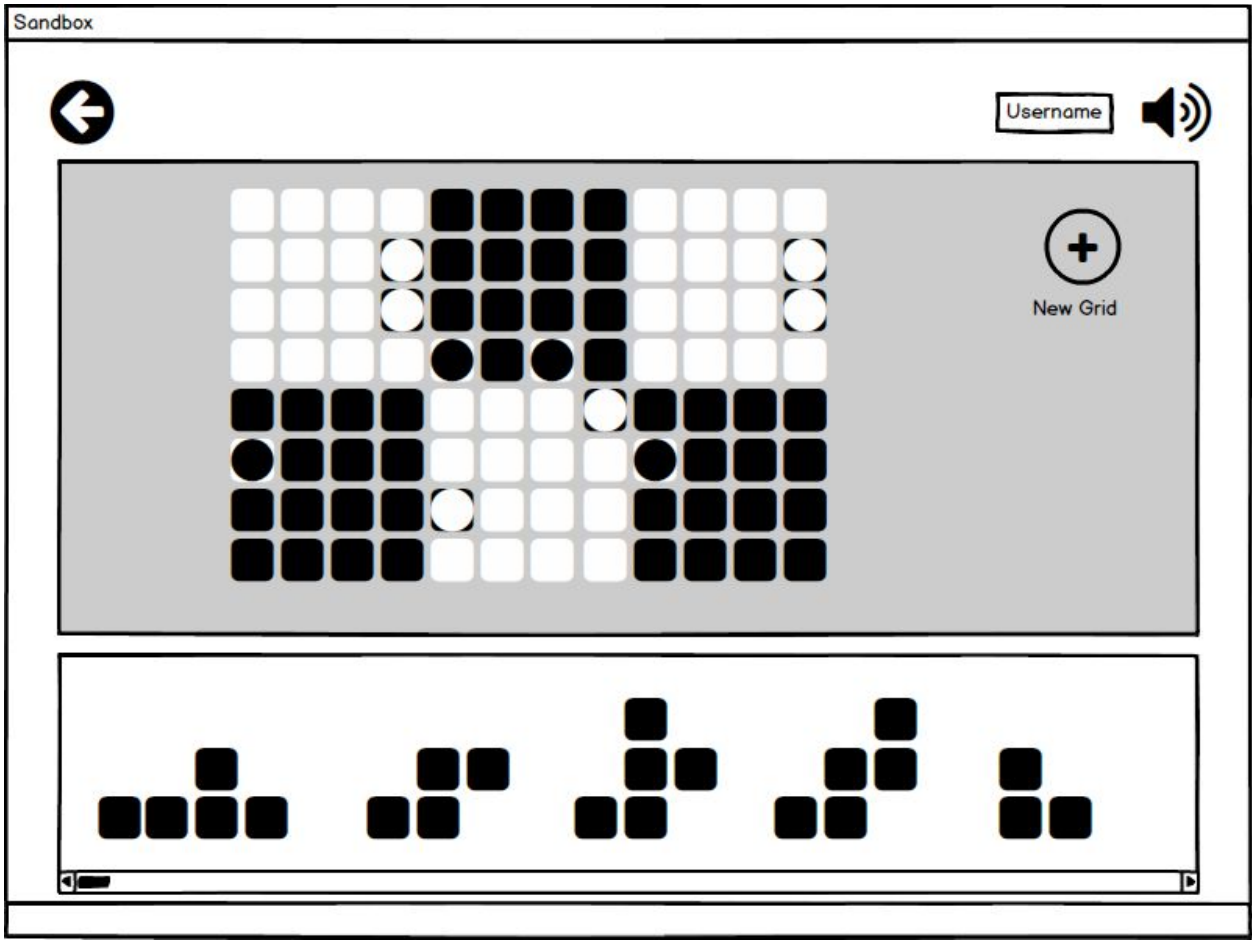
This is the page where player can see both his levels and other players' levels. In order to create a new level player presses Create New Level Button and system directs the player to the creating page.

5.4.2.17. Create New Level Page



This is the page where user can create a new level. Firstly, player decides the placement of grids and saves it. Then, player may place some of the pieces onto grid which will be pre-placed for this level. After system saves the last step, player has to solve the rest of the puzzle in order to save the whole puzzle as a game. In the case of success, system saves the game as a new level for all the users and returns the previous page.

5.4.2.19. Sandbox Page



This page is the sandbox game mode that enables player to add grids more than four and as many pieces as they wish. This is simply the freedom of the player. Player can move any piece regardless of the restrictions. By clicking new grid button on the right upper corner, a grid appears on the screen which can be deleted by pressing backspace. Player can go back to play menu by pressing back arrow button.

7. References

[1] “Quadrillion - SmartGames”, [Online] Available: https://www.smartgames.eu/uk/one-player-games/quadrillion , Accessed March 09, 2019
[2] “user1444_Quadrillion_ CS 319 Term Project - Spring 2019”, [Online] Available: https://github.com/user1444/Quadrillion , Accessed March 10, 2019
[3] B. Bruegge, A. H. Dutoit, Object-Oriented Software Engineering 3rd Edition, Using UML, Patterns, and Java, Prentice-Hall, 2010
[4] “Ideal Modeling & Diagramming Tool for Agile Team Collaboration”, [Online] Available: https://www.visual-paradigm.com/ , Accessed March 10, 2019
[5] “Balsamiq. Rapid, effective and fun wireframing software.”, [Online] Available: https://balsamiq.com/ , Accessed March 10, 2019
[6] “Enabling Open Innovation & Collaboration _ The Eclipse Foundation”, [Online] Available: https://www.eclipse.org/ , Accessed March 10, 2019
[7] “IntelliJ IDEA_ The Java IDE for Professional Developers by JetBrains”, [Online] Available: https://www.jetbrains.com/idea/ , Accessed March 10, 2019