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

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 login. 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 here. The purpose of this mode is to provide a warm-up for the players.

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 must register to the system in the register screen. If the player has an existing account, he/she can login to the system by clicking login button to enter their personal credentials. The information of the registered users should be stored in the one centralized database.

3.2. The Game

The player should be able to drag, rotate, flip and place a puzzle piece

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.

Sandbox

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

Level Editor

Users should be able to play Level editor mode. Users will be able to create their own levels to be played in the future. Players are free to modify the level they created as they are the owners of the level. The users should also be able to save the game level they created to be viewed by other people. Everyone should be able to access the game level that was submitted.

Rush Mode

Different type of challenges are provided in the Rush Mode. Players should win the game with imposed restrictions to complete a game instance in the Rush Mode.

Volume

The user should be able to change the ingame volume or mute it altogether if they decide so using the in game volume slider.

Achievements

The user should be able to view achievements that they have collected throughout the game.

4. Non-functional Requirements

4.1. Usability

4.1.1 Color-blind Mode

Color-blind mode will be helpful for players who suffer from some sort of color-blindness. This is very critical in the game where a successful completion of the game depends on the distinguishing among 12 colorful pieces. Color-blind mode makes the game more comfortable and easier to play for players with color blindness.

4.1.2 Tutorial

The Game Tutorial is mandatory for the first time play to demonstrate how to play the game and and how the puzzles pieces are interact in game. In the tutorial, the settlement of twelve game pieces and 4x4 grid will be shown. Also, mouse clicks which are hold, drop and drag the pieces will be animated in tutorial before playing the game. Pause button for freeze the game can be provided by “P” button on the keyboard. That is, there are right and left mouse buttons and 1 button on the keyboard to interact with the game.

4.1.3 Changing Theme

My Little Quadrillion is a desktop game in the digital platform. For optional visualization while playing game can be provided by three different themes in the options part of the game. Night mode will reduce the half of the game light for providing darker screen. Also, user experiences the game more like when they play the game on a wooden table owing to wooden theme in the theme option.

4.2. Supportability

The game should be runnable on machines with Windows, Mac, and Linux operating systems.

4.3. Performance

Since it is critical for the application such as game to be responsive the requests and responses from the database should not take more than one second.

4.4. Reliability

The code should be written so that sql injections are intercepted and user input data is sanitized.

4.5. Implementation

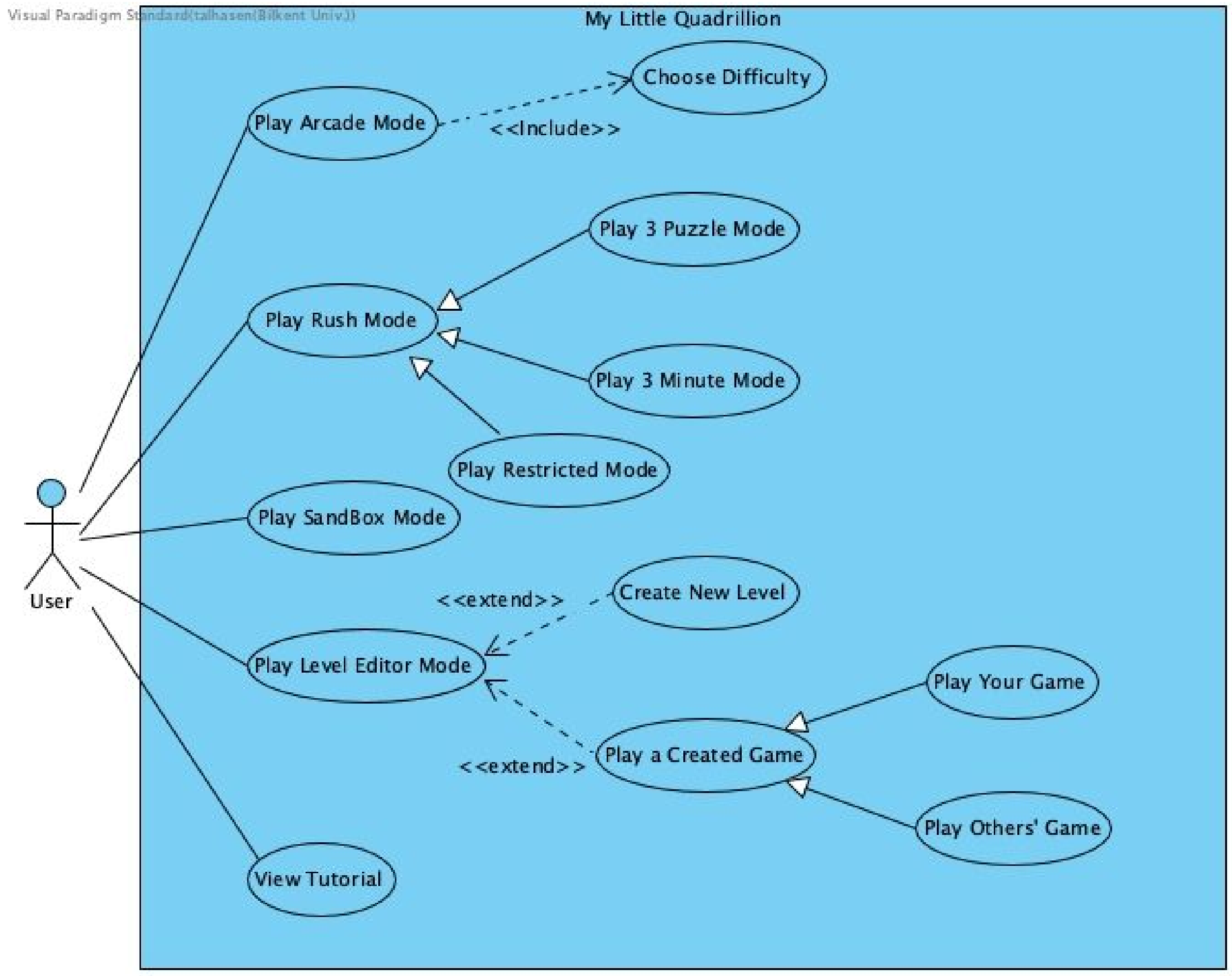
The application should be developed using Object Oriented Programming Language.

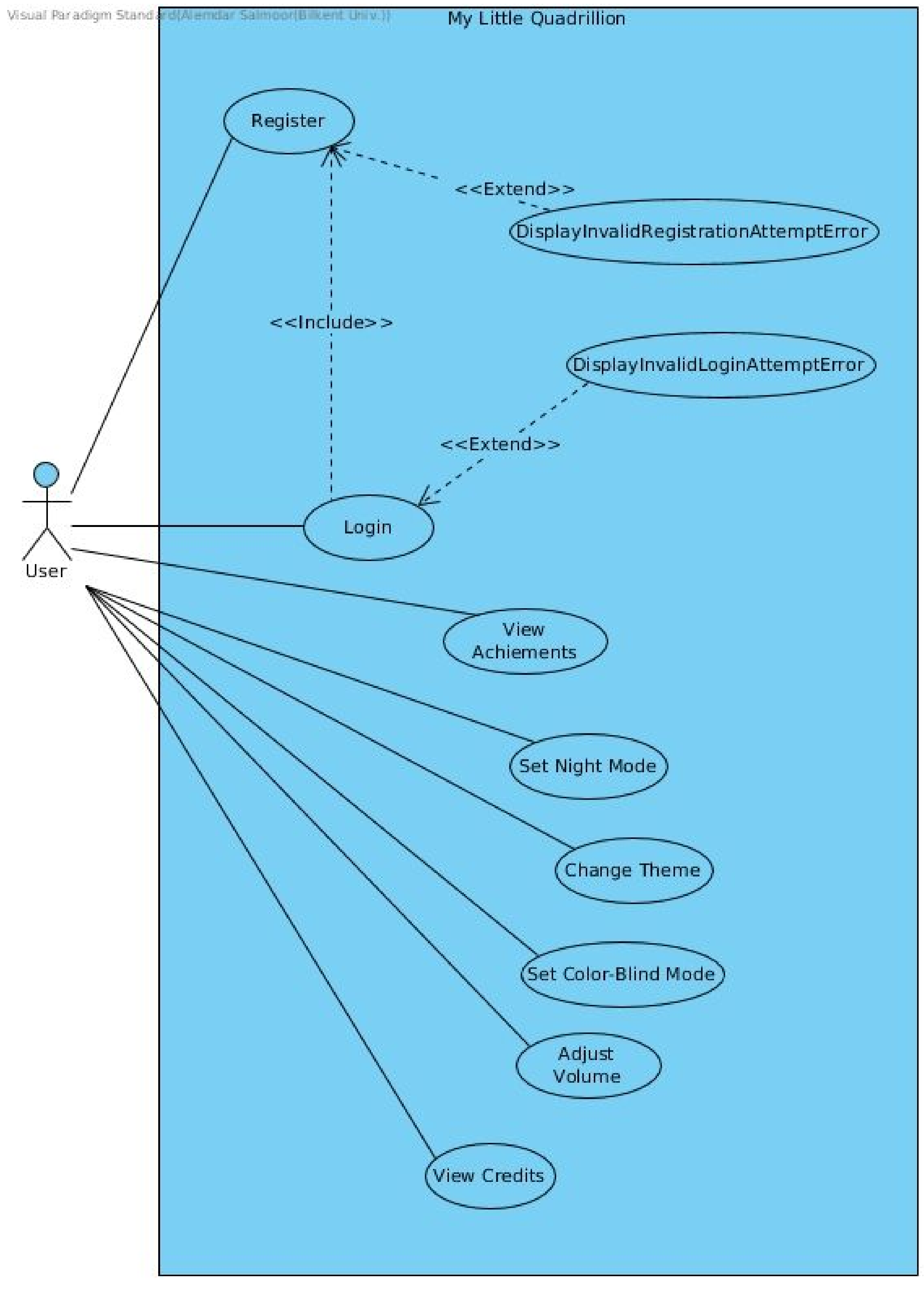
5. System Models

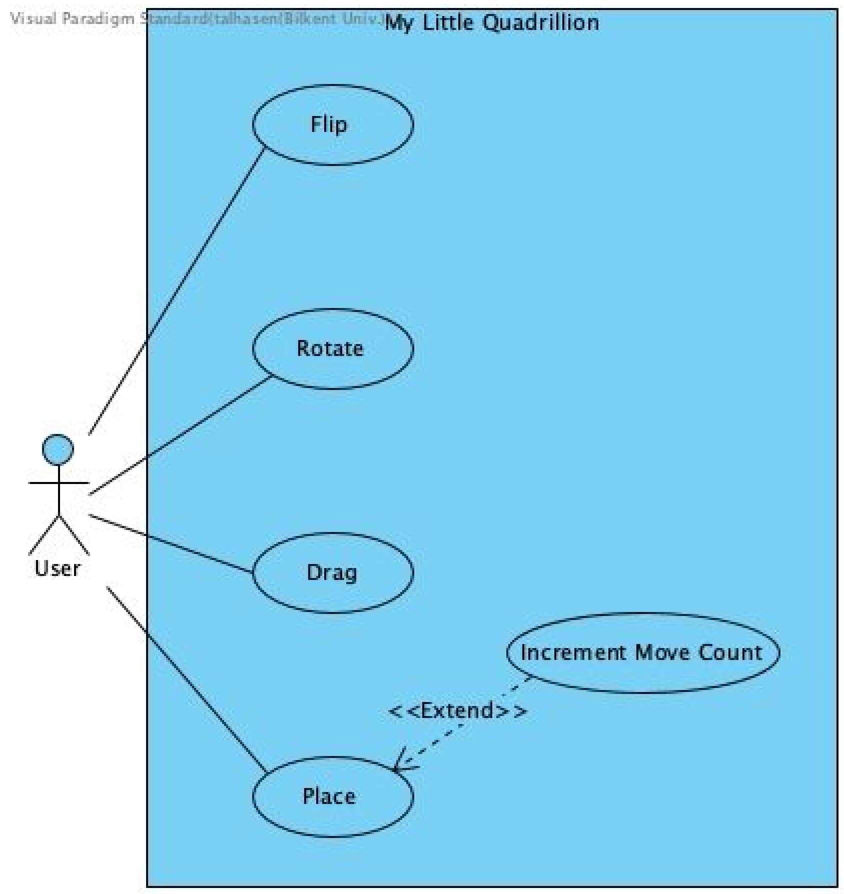
5.1. Use case model

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(Alemdar Salmoor(Bilkent Univ.))
My Little Quadrillion
Register
<<Extend>>
DisplayInvalidRegistrationAttemptError
<<Include>>
DisplayInvalidLoginAttemptError
<<Extend>>
User
Login
View Achiements
Set Night Mode
Change Theme
Set Color-Blind Mode
Adjust Volume
View Credits



Use Case Name:	Register
Participating Actor:	Invoked by User
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
Flows of events:	1. Player enters username and password. 2. System checks the username for duplicates and password for proper format

Use Case Name:	Display Invalid Registration Attempt Error
Participating Actor:	Communicates with User
Stakeholders and Interests:	User entered invalid username or password for registration
Entry condition:	This use case extends Register use case. It is initiated by the system whenever user enters duplicate username or password of incorrect format
Exit condition:	Player is warned to reenter entered values for username or password
Flows of events:	1. System displays the error message regarding the duplicate username or invalid password and prompts the user to re enter those values

Use Case Name:	Login
Participating Actor:	Invoked by User
Stakeholders and Interests:	Player wants to log in.
Entry condition:	Player should open the game and login screen.
Exit condition:	System makes the user logged in.
Flows of events:	1. Player enters username and password. 2. System checks whether the password and the username are matching or not if they are matching, the system informs the user of successful login.

Use Case Name:	Display Invalid Login Attempt Error
Participating Actor:	Communicates with User
Stakeholders and Interests:	User entered invalid username or password to login
Entry condition:	This use case extends Login use case. It is initiated by the system whenever user enters invalid username and password combination
Exit condition:	Player is prompted to re enter the username and password
Flows of events:	1. System displays the error message regarding the not matching username and password combination. The user is prompted to re-enter those values

Use Case Name:	View Achievements
Participating Actor:	Initiated by User
Stakeholders and Interests:	User wants to see his/her achievements
Entry condition:	User should be logged in,
Exit condition:	System displays user’s all achievements
Main flows of events:	1. User chooses view achievements from the main menu of the game. 2. System displays both locked and unlocked achievements.

Use Case Name:	Set Night Mode
Participating Actor:	Invoked by User
Stakeholders and Interests:	User wants to set Night Mode.
Entry condition:	User should open the game and be logged in. User should be in the settings menu of the game.
Exit condition:	System sets the theme to Night Mode
Main flows of events:	1. User toggles the night mode option 2. System enables or disables the night mode according to the user’s selection

Use Case Name:	Change Theme
Participating Actor:	Invoked by User
Stakeholders and Interests:	User wants to change the theme
Entry condition:	User should open the game and be logged in. User should be in the settings menu of the game.
Exit condition:	System changes the theme to the one chosen by the user.
Main flows of events:	1. User selects one of the themes from drop down menu. 2. System changes the theme of the game according to the selected option.

Use Case Name:	Set Color-Blind Mode
Participating Actor:	Invoked by User
Stakeholders and Interests:	User wants to set Color-Blind Mode
Entry condition:	User should open the game and be logged in. User should be in the settings menu of the game.
Exit condition:	System sets the Color-Blind Mode
Main flows of events:	1. User toggles the night mode option 2. System enables or disables the colorblind mode according to the user’s selection

Use Case Name:	Adjust Volume
Participating Actor:	Invoked by User
Stakeholders and Interests:	User wants to adjust the volume of the game.
Entry condition:	User should open the game and be logged in. User should be in the settings menu of the game.
Exit condition:	System adjusts the volume.
Main flows of events:	1. User clicks and drags the volume slider to desired intensity. 2. System adjusts volume of the game according to the position of the slider.

Use Case Name:	View Credits
Participating Actor:	Initiated by User
Stakeholders and Interests:	User wants to view credits
Entry condition:	User should be in main menu
Exit condition:	System displays credits
Main flows of events:	1. User chooses view credits from the main menu of the game. 2. System displays the information regarding the contributors of the game.

Use Case Name:	Play Arcade Mode
Participating Actor:	Initiated by User
Stakeholders and Interests:	User wants to play the Arcade Mode.
Entry condition:	User selects 3 Puzzle Mode
Exit condition:	System displays the result of the game and earned stars if the puzzle is solved.
Main flows of events:	1. User selects a level within a chosen difficulty level 2. System starts game 3. User tries to solve puzzle by moving pieces on grid 4. User may take hints 5. Game finishes when the puzzle is solved 6. System displays earned stars

Use Case Name:	Play Level Editor Mode
Participating Actor:	Initiated by User
Stakeholders and Interests:	User wants to play the Level Editor Mode.
Entry condition:	User selects Level Editor option.
Exit condition:	User selects one level from the available level list to play game.
Main flows of events:	1. Available levels are shown in screen. 2. Unlock levels are shown but not selected. 3. User select the hardship level.

Use Case Name:	Play Sandbox Mode
Participating Actor:	Initiated by User
Stakeholders and Interests:	User wants to play the Sandbox Mode.
Entry condition:	User selects Sandbox Mode option
Exit condition:	System displays the result of the game and earned stars if the puzzle is solved.
Main flows of events:	1. System starts the game 2. Player is free to add many grids and put pieces on the grid
Alternative flows of events:	If player wants to return previous menu press the back arrow icon.

Use Case Name:	Play 3 Minute Mode
Participating Actor:	Initiated by User
Stakeholders and Interests:	User selects 3 Minute Mode option.
Entry condition:	User wants to play the 3 Minute Mode.
Exit condition:	System displays the result of the game and earned stars if the puzzle is solved.
Main flows of events:	1. System starts the game 2. User selects pieces and try to place the on grid 3. System updates timer 4. User completes a puzzle and the system brings a new puzzle 5. Game finishes when time is up 6. System shows user’s score
Alternative flows of events:	If player wants to return previous menu press the back arrow icon.

Use Case Name:	Play 3 Puzzle Mode
Participating Actor:	Initiated by User
Stakeholders and Interests:	User wants to play 3 Puzzle Mode.
Entry condition:	User selects 3 Puzzle Mode.
Exit condition:	System displays the result of the game and earned stars if the puzzle is solved.
Main flows of events:	1. System starts game 2. User selects pieces and try to place them onto grid 3. System updates move count 4. User completes all puzzles one by one 5. System shows user's score
Alternative flows of events:	If player wants to return previous menu press the back arrow icon.

Use Case Name:	Play Restricted Mode
Participating Actor:	Initiated by User
Stakeholders and Interests:	User wants to play the Restricted mode.
Entry condition:	User selects Restricted Mode.
Exit condition:	System displays the result of the game and earned stars if the puzzle is solved.
Main flows of events:	1. System starts game 2. Player selects pieces and try to place them on grid 3. System updates number of moves left that can be played 4. Game finishes when no move left or puzzle is solved 5. System shows player's score

Use Case Name:	View Tutorial
Participating Actor:	Initiated by User
Stakeholders and Interests:	User wants to learn how to play game.
Entry condition:	User initiates the tutorial
Exit condition:	User finishes the tutorial.
Main flows of events:	1. System starts the tutorial. 2. System asks for showing the tutorial for a second time. 3. User selects either to go over it again or finishes the tutorial.
Alternative flows of events:	User can interrupt the tutorial any time and go back to the previous page.

Use Case Name:	Create New Level
Participating Actor:	Initiated by User
Stakeholders and Interests:	User wants to create a new level.
Entry condition:	This use case extends play Level Editor use case whenever User wants to create a new level
Exit condition:	User saves the game.
Main flows of events:	1. System starts game 2. User moves grids to create layout 3. System saves layout 4. User places pieces that will be pre-placed on the grid or skip this step 5. System saves pieces if there is any 6. User puts rest of the pieces onto grid 7. System saves the puzzle and asks a name for the level 8. Player names the level 9. System checks the name among the already created games and names the gama if that has not been used before 10. System saves the game
Alternative flows of events:	7.1. System gives a warning if the rest of the pieces are not placed properly onto grid 9.1. System gives warning for the taken name.

Use Case Name:	Play Created Game
Participating Actor:	Initiated by User
Stakeholders and Interests:	User wants to play the level that is created before.
Entry condition:	This is use case extends Play Level Editor use case whenever user wants to play a created game.
Exit condition:	System displays the result of the game and earned stars if the puzzle is solved.
Main flows of events:	1. System starts game 2. Player tries to solve puzzle by moving pieces on grid 3. Game finishes when the puzzle is solved

Use Case Name:	Play Your Game
Participating Actor:	Initiated by User
Stakeholders and Interests:	User wants to play his/her own game.
Entry condition:	Inherited from Play Created Game use case
Exit condition:	Inherited from Play Created Game use case
Main flows of events:	1. User chooses the game created by himself 2. System responds by opening the selected game instance

Use Case Name:	Play Others Game
Participating Actor:	Inherited from Play Created Game use case
Stakeholders and Interests:	User wants to play the games created by others.
Entry condition:	Inherited from Play Created Game use case
Exit condition:	Inherited from Play Created Game use case
Main flows of events:	3. User chooses the game created by others from the provided list 4. System responds by opening the selected game instance

Use Case Name:	Flip
Participating Actor:	Initiated by User
Stakeholders and Interests:	User wants to flip the piece around its axis
Entry condition:	The user should be in of the game modes, user should have rotated the piece 3 times
Exit condition:	The piece is flipped around its axis
Main flows of events:	1. The user right clicks the button on the mouse 2. The system responses by flipping the piece around its axis

Use Case Name:	Drag
Participating Actor:	Initiated by User
Stakeholders and Interests:	User wants to drag a piece on the screen
Entry condition:	The user should be in of the game modes, user should have hovered over the mouse
Exit condition:	The puzzle piece is dragged to the required place
Main flows of events:	1. The user right clicks the button on the mouse 2. The system responses by enabling the user to drag the puzzle piece around

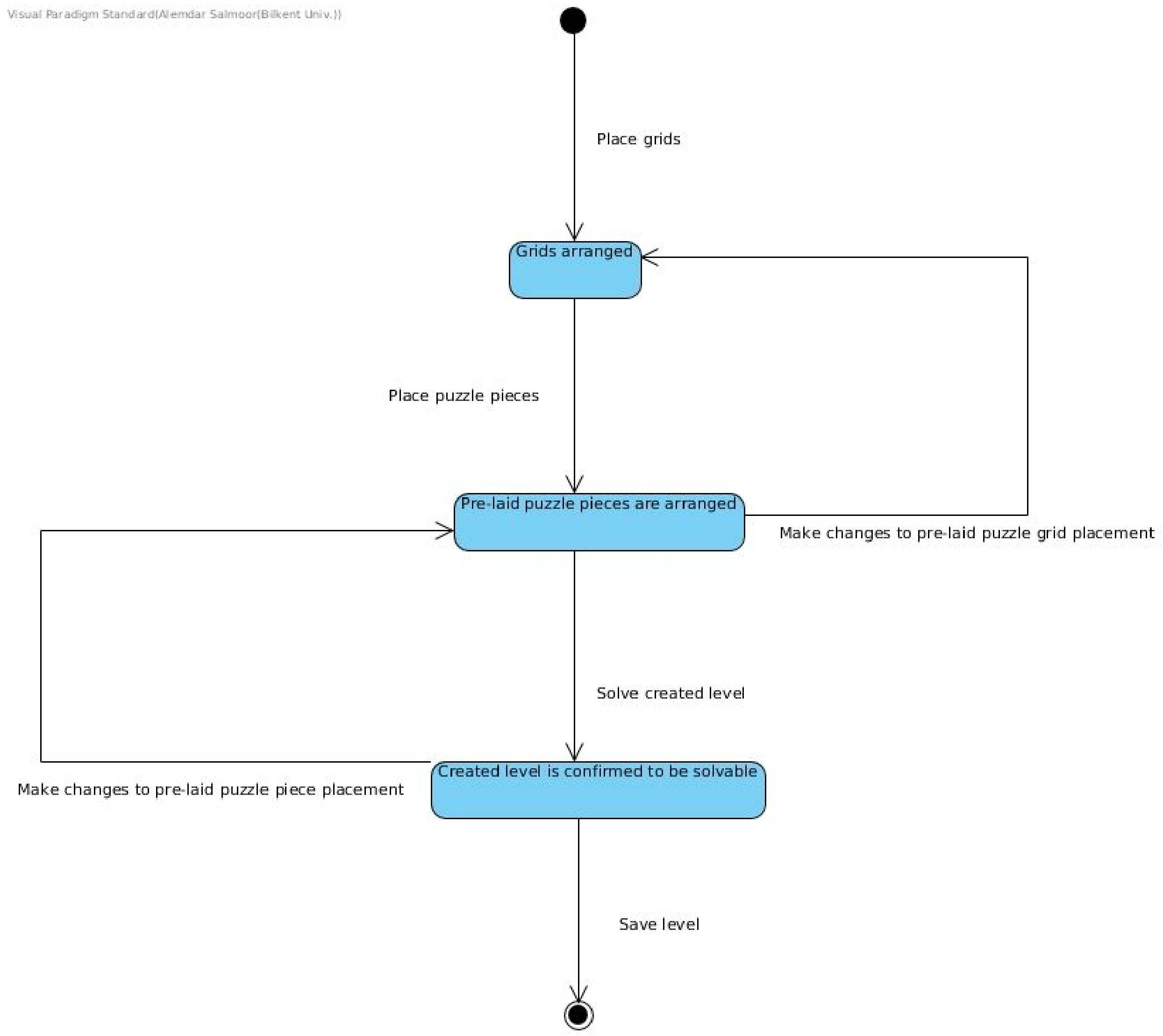
Use Case Name:	Place
Participating Actor:	Initiated by User
Stakeholders and Interests:	User wants to place a piece on a desired position
Entry condition:	The user should be in of the game modes, user should have dragged the puzzle piece over a required position
Exit condition:	The piece is placed on the required position
Main flows of events:	1. The user right clicks the button on the mouse 2. The system responses by snapping the piece to the designated space and changes color saturation of the puzzle piece to signal successful placement

Use Case Name:	Rotate
Participating Actor:	Initiated by User
Stakeholders and Interests:	User wants to rotate the puzzle piece
Entry condition:	The user should be in of the game modes, user should have dragged a piece
Exit condition:	The piece is rotated desired number of times
Main flows of events:	1. The user right clicks the button on the mouse 1 time for each rotation while dragging the piece. 2. The system responses by rotating the piece by 90 degrees for each right click.

5.2. Dynamic models

5.2.1. State diagrams

5.2.1.1. Level Editor State Diagram



Visual Paradigm Standard(Alemdar Salmoor(Bilkent Univ.))
Place grids
Grids arranged
Place puzzle pieces
Pre-laid puzzle pieces are arranged
Make changes to pre-laid puzzle grid placement
Solve created level
Created level is confirmed to be solvable
Make changes to pre-laid puzzle piece placement
Save level

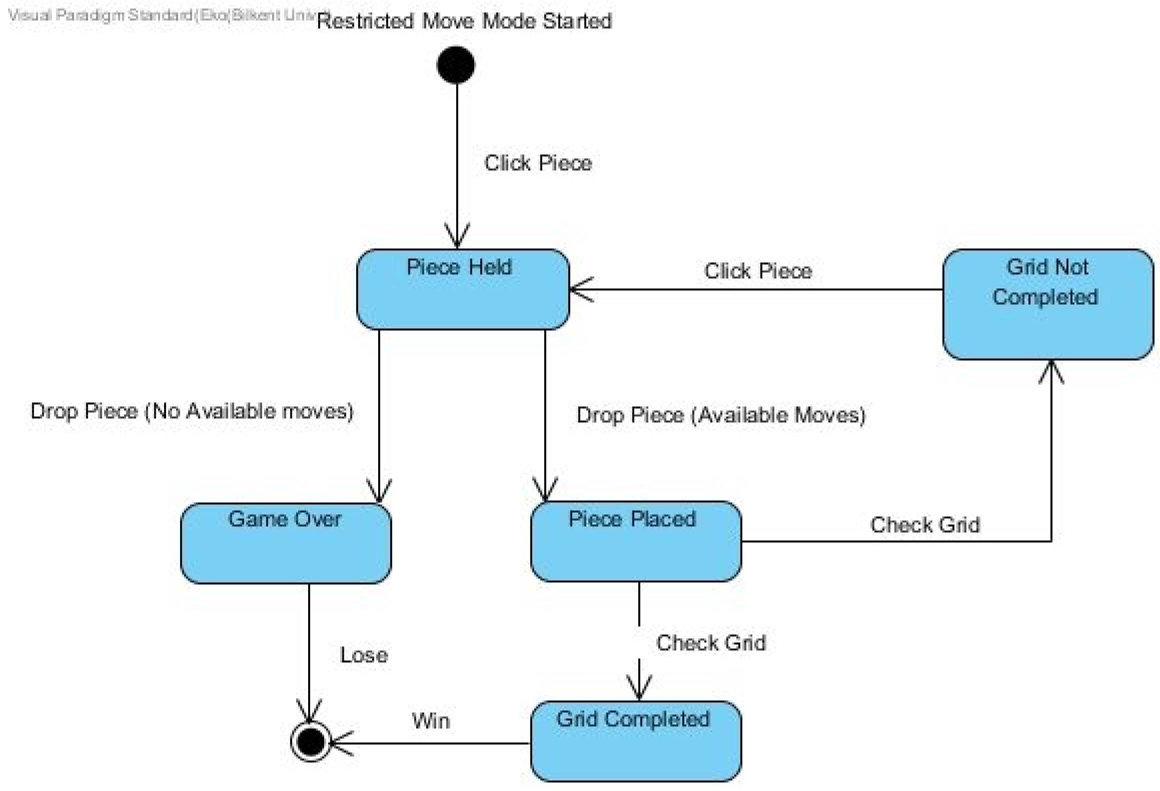
In level editor, player first decides on where will he place the grids. If he places the grids in a valid pattern, adjacent to each other, program will allow him to proceed to the next phase. In piece placement phase player decides pre-placed pieces on the board. Player can place as many pieces as he 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.

5.2.1.2. Rush Mode State Diagrams

Rush Mode is consist of three other submodes. Each mode challenges different aspect.

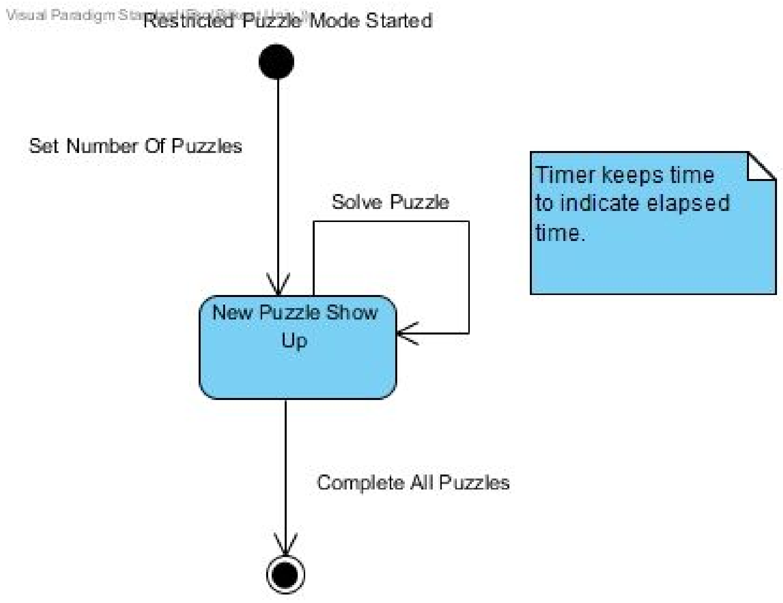
5.2.1.2.1. Restricted Move State Diagram

User has limited number of move to complete the puzzle. If user runs out of moves, s/he loses the game.


Visual Paradigm Standard(Eko(Bilkent Univ
Restricted Move Mode Started
Click Piece
Piece Held
Click Piece
Grid Not Completed
Drop Piece (No Available moves)
Drop Piece (Available Moves)
Game Over
Piece Placed
Check Grid
Lose
Check Grid
Win
Grid Completed

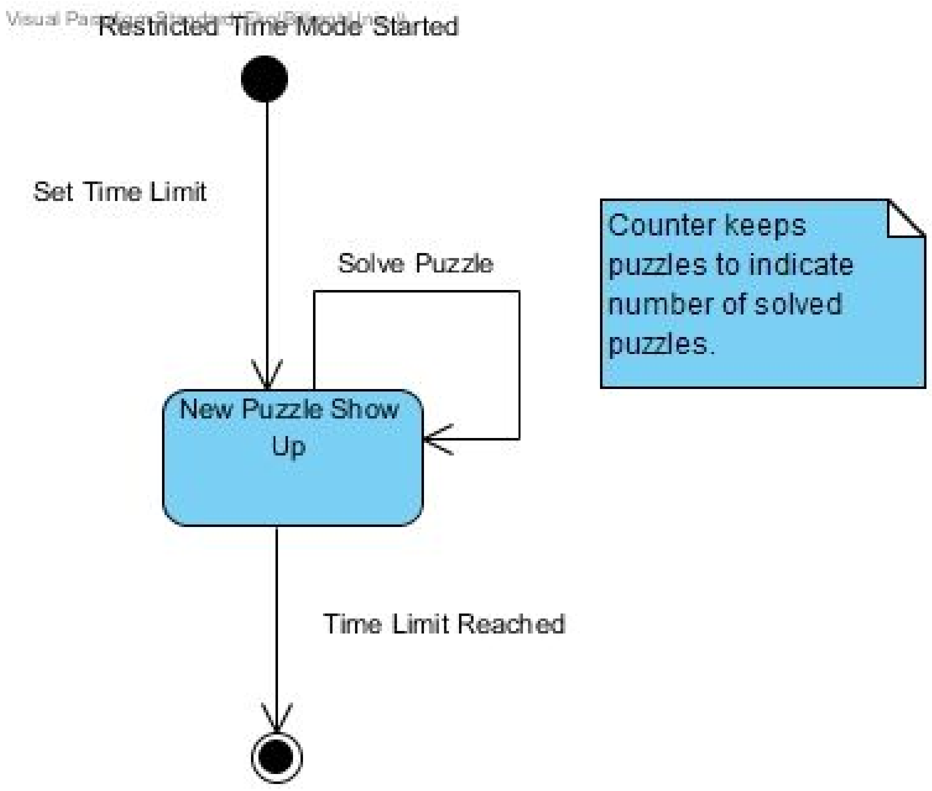
5.2.1.2.2. Restricted Puzzle State Diagram

User tries to solve specified number of puzzles and timer keeps time. When user solves a puzzle, new puzzle shows up until specified number of puzzles are solved.


Visual Paradigm Standard Edition(Bilkent Univ.))
Restricted Puzzle Mode Started
Set Number Of Puzzles
Solve Puzzle
Timer keeps time to indicate elapsed time.
New Puzzle Show Up
Complete All Puzzles

5.2.1.2.3 Restricted Time State Diagram

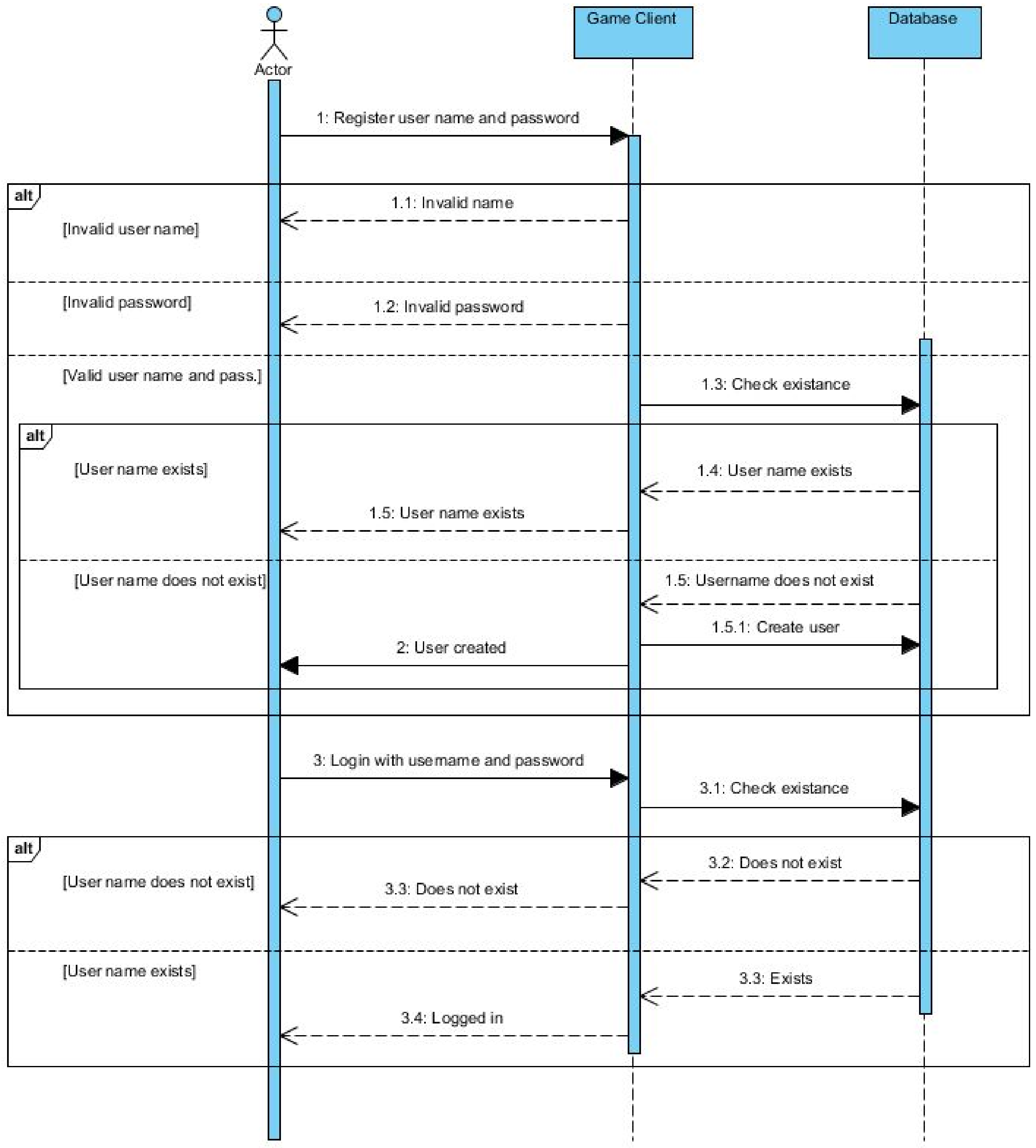
User encounters with new puzzles as long as s/he can solve each puzzle until the time limit is reached. Counter keeps track of number of solved puzzles.


Visual Paradigm Standard Edition(Bilkent Univ.))
Restricted Time Mode Started
Set Time Limit
Solve Puzzle
Counter keeps puzzles to indicate number of solved puzzles.
New Puzzle Show Up
Time Limit Reached

5.2.2. Sequence Diagrams

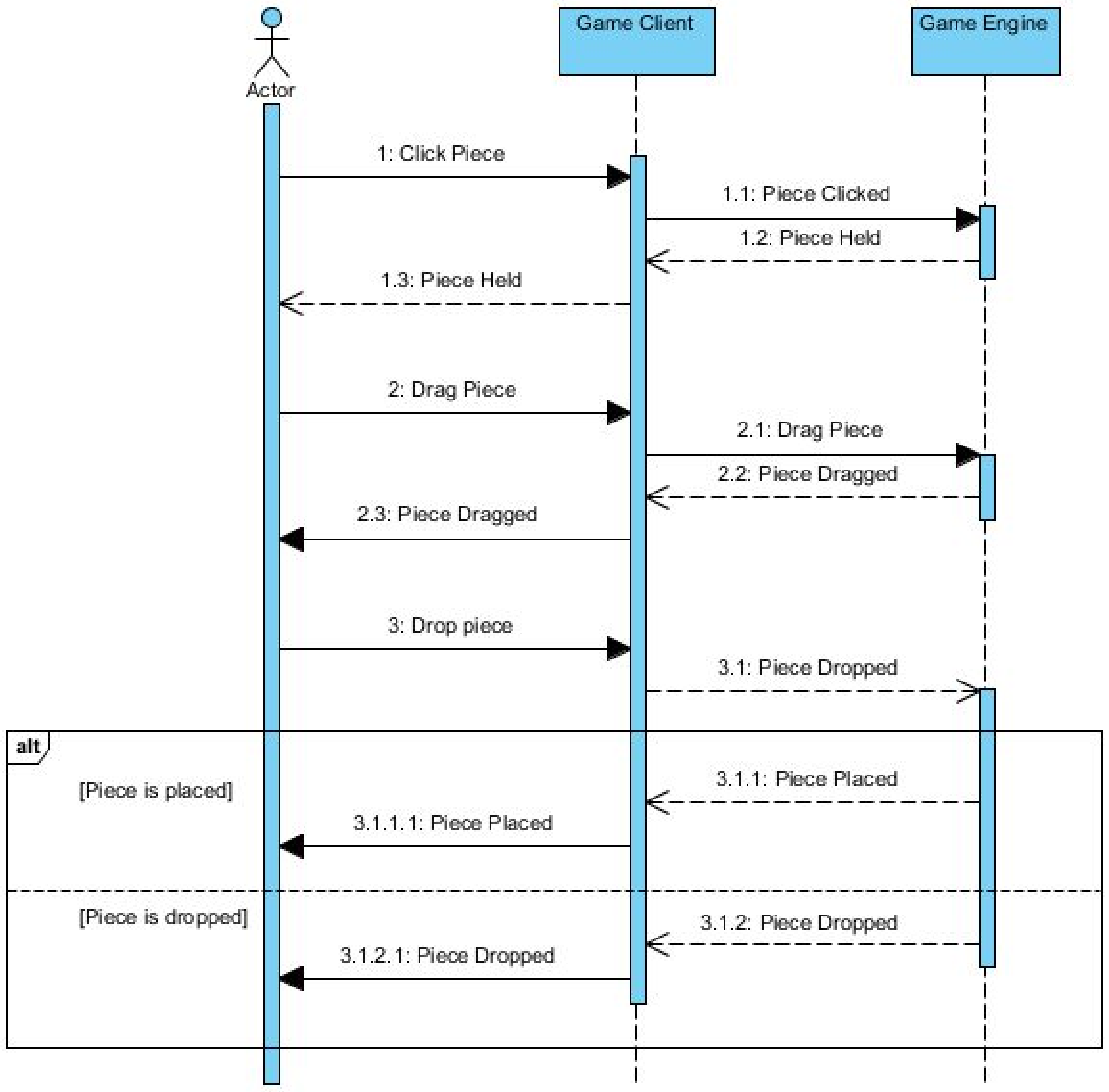
5.2.2.1. Login Sequence Diagram

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.


Actor
Game Client
Database
1: Register user name and password
alt
[Invalid user name]
1.1: Invalid name
[Invalid password]
1.2: Invalid password
[Valid user name and pass.]
1.3: Check existance
alt
[User name exists]
1.4: User name exists
1.5: User name exists
[User name does not exist]
1.5: Username does not exist
1.5.1: Create user
2: User created
3: Login with username and password
3.1: Check existance
alt
[User name does not exist]
3.2: Does not exist
3.3: Does not exist
[User name exists]
3.3: Exists
3.4: Logged in

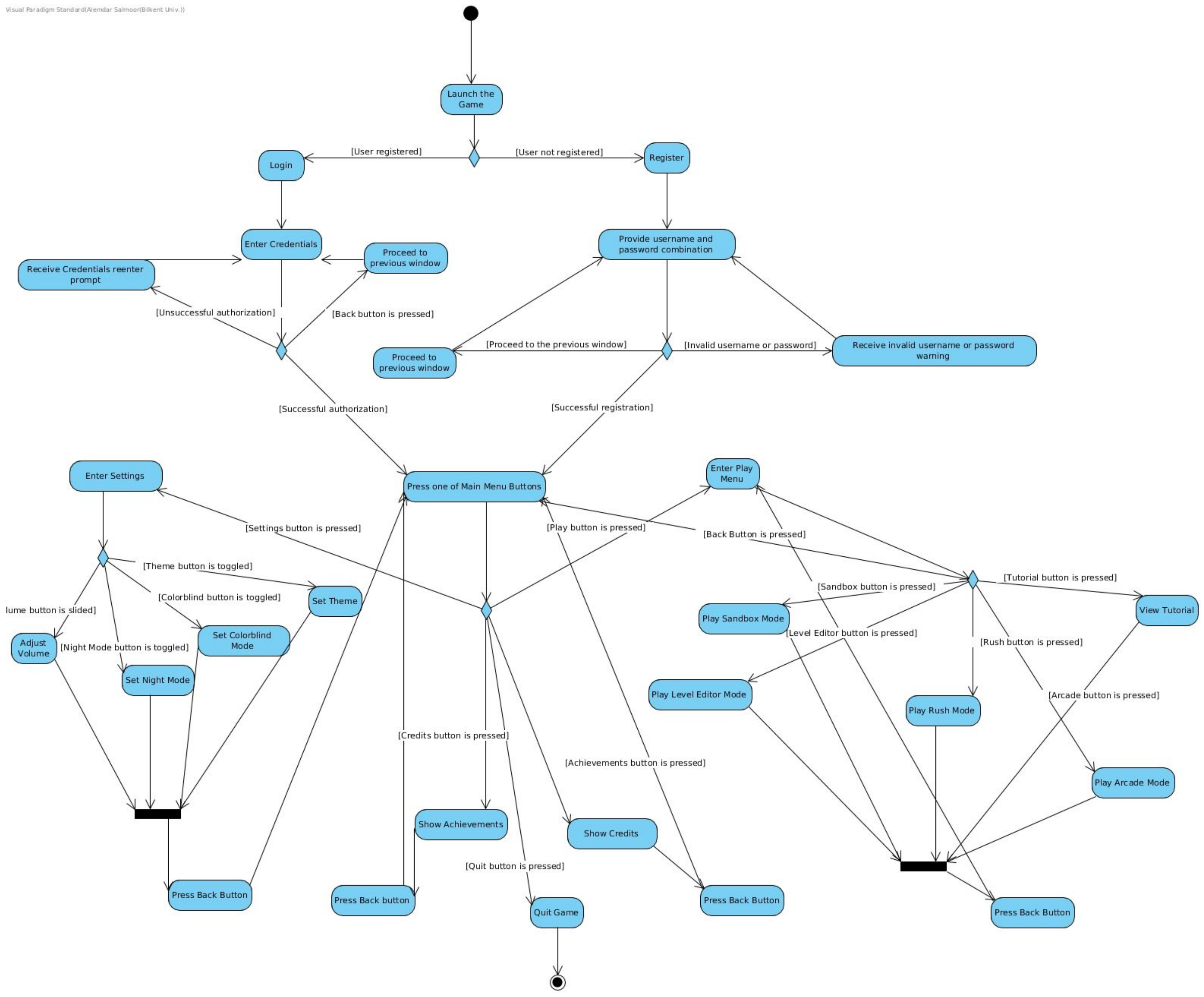
5.2.2.1 Game Play Sequence Diagram

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.



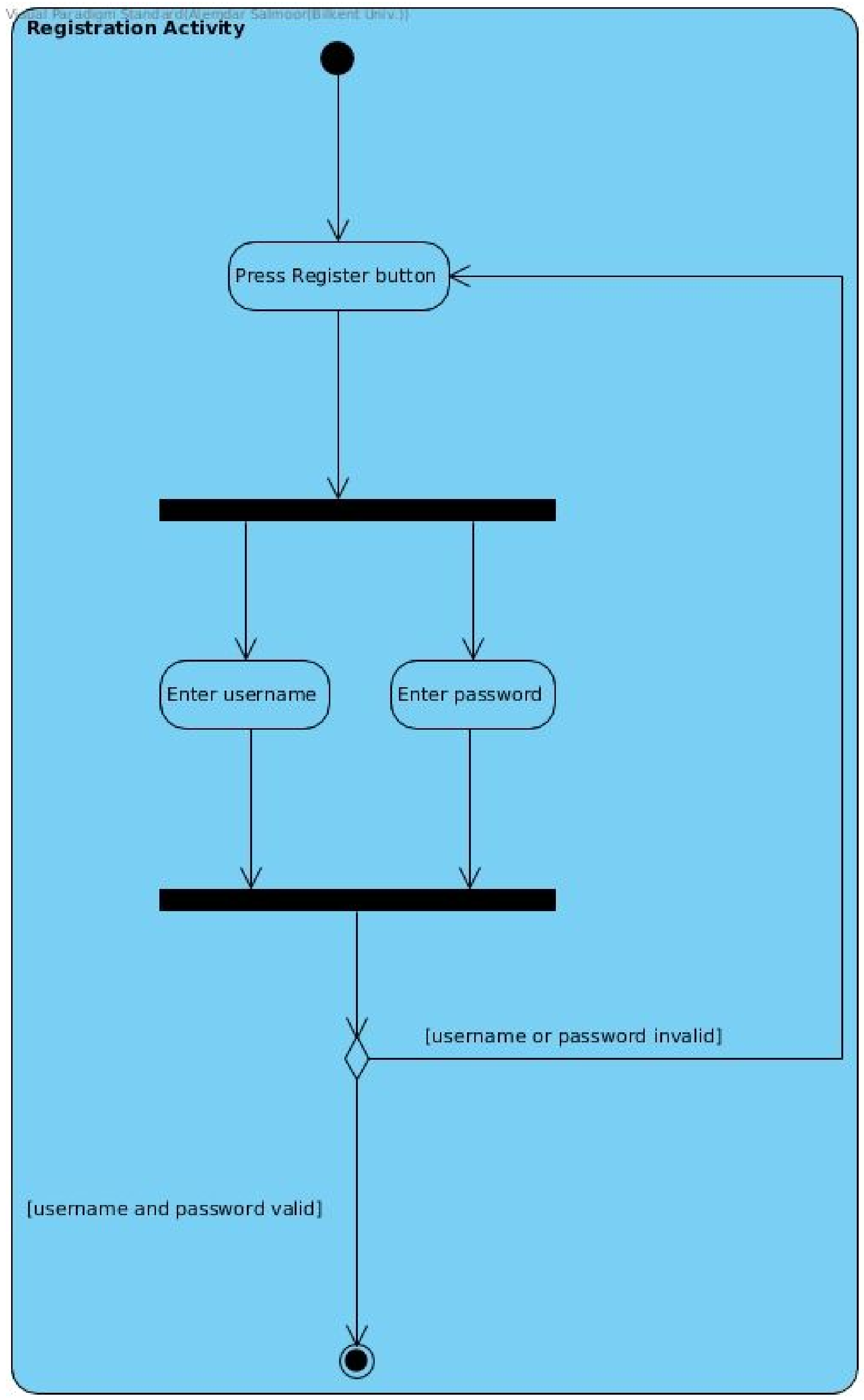
5.2.3. Activity Diagram

Below is the activity diagram of user menu navigation from perspective of the user playing My Little Quadrillion Game.



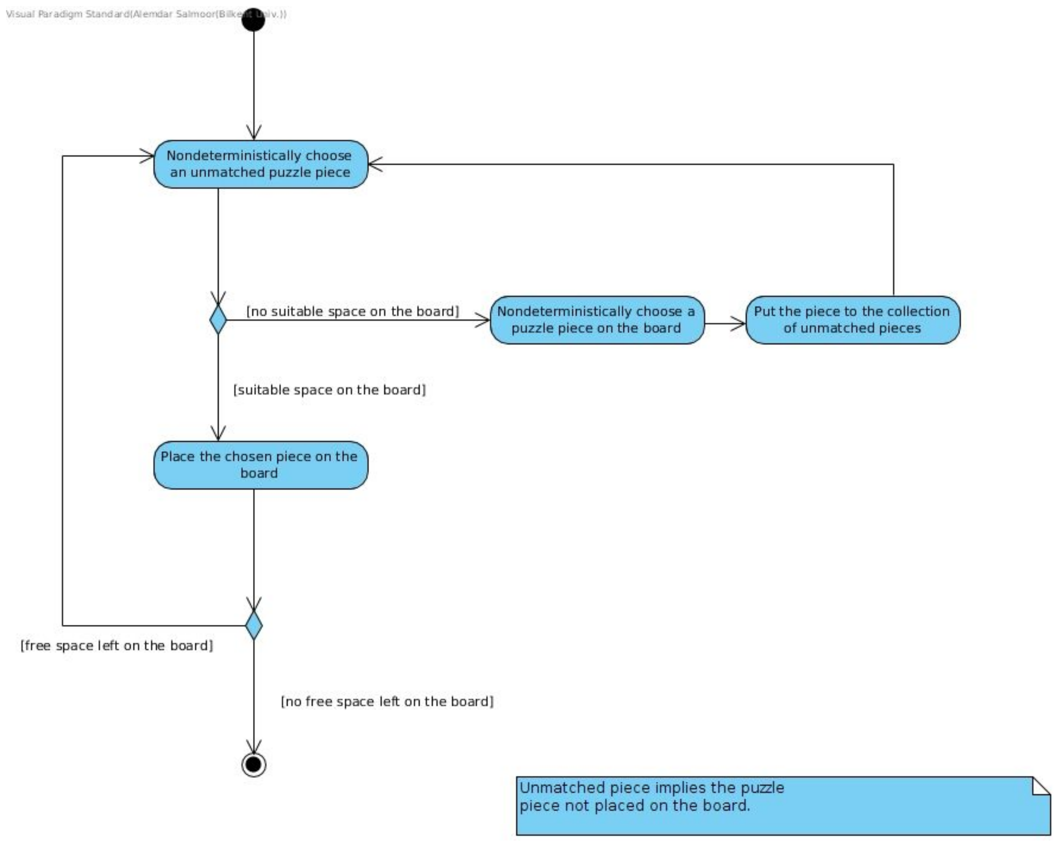
Visual Paradigm Standard(Alemdar Salmoor(Bilkent Univ.))
Launch the Game
[User registered]
[User not registered]
Login
Register
Enter Credentials
Proceed to previous window
Receive Credentials reenter prompt
[Unsuccessful authorization]
[Back button is pressed]
Provide username and password combination
[Proceed to the previous window]
[Invalid username or password]
Proceed to previous window
Receive invalid username or password warning
[Successful authorization]
[Successful registration]
Enter Settings
Press one of Main Menu Buttons
Enter Play Menu
[Settings button is pressed]
[Play button is pressed]
[Back Button is pressed]
[Theme button is toggled]
[Colorblind button is toggled]
Set Theme
lume button is slided]
Adjust Volume
[Night Mode button is toggled]
Set Colorblind Mode
Set Night Mode
[Sandbox button is pressed]
[Tutorial button is pressed]
View Tutorial
Play Sandbox Mode
[Level Editor button is pressed]
[Rush button is pressed]
Play Level Editor Mode
Play Rush Mode
[Arcade button is pressed]
[Credits button is pressed]
[Achievements button is pressed]
Play Arcade Mode
Show Achievements
Show Credits
[Quit button is pressed]
Press Back Button
Press Back button
Quit Game
Press Back Button
Press Back Button

Below is the activity diagram of login activity from the perspective of the user. The diagram depicts that username and password are entered simultaneously and if either username or password is invalid user should repeat the process.

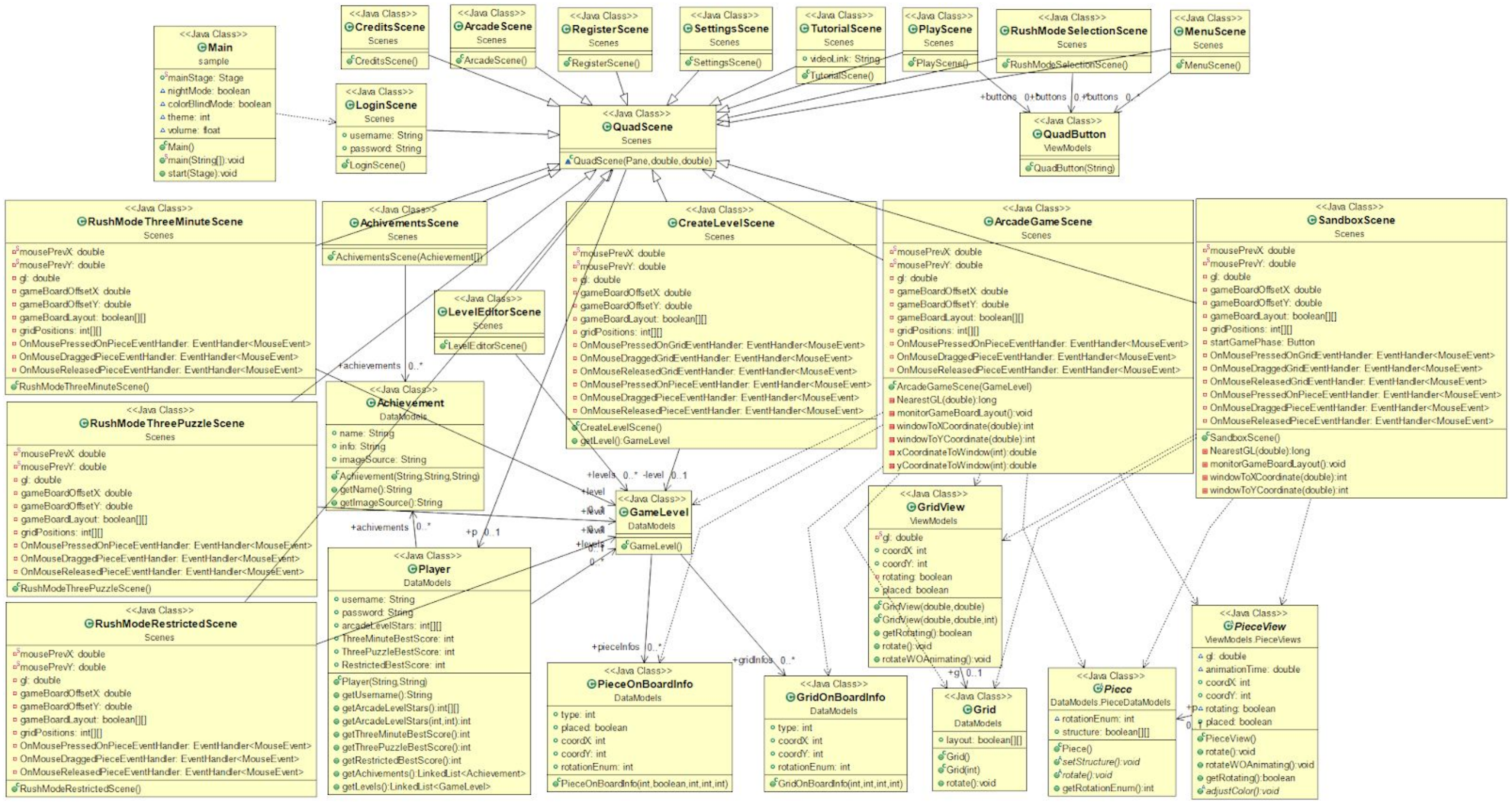

Visual Paradigm Standard(Alemdar Salmoor(Bilkent Univ.))
Registration Activity
Press Register button
Enter username
Enter password
[username or password invalid]
[username and password valid]

This is an Activity Diagram of one instance of Arcade Game Level. The word nondeterministically is used to convey the unpredictable nature of the next piece to be chosen and placed. By using the word nondeterministically we abstract ourselves from any one particular piece. The steps depicted on the Activity Diagram are as follows:

1. Choose one of the pieces to be placed on the board and look if there is a suitable place on the board(which is a collection of grids) for a chosen puzzle piece. If there is go to step 3 else go to step 2.
2. Now, since the chosen piece cannot be placed, choose a piece that is already on the board and put it to the collection of unmatched pieces. and go to step 1.
3. Place a puzzle piece and check if the board has free (unoccupied by the puzzle pieces) space left. If it does go to step 1 else the level is completed.


Visual Paradigm Standard(Alemdar Salmoor(Bilkent Univ.))
Nondeterministically choose an unmatched puzzle piece
[no suitable space on the board]
Nondeterministically choose a puzzle piece on the board
Put the piece to the collection of unmatched pieces
[suitable space on the board]
Place the chosen piece on the board
[free space left on the board]
[no free space left on the board]
Unmatched piece implies the puzzle piece not placed on the board.

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

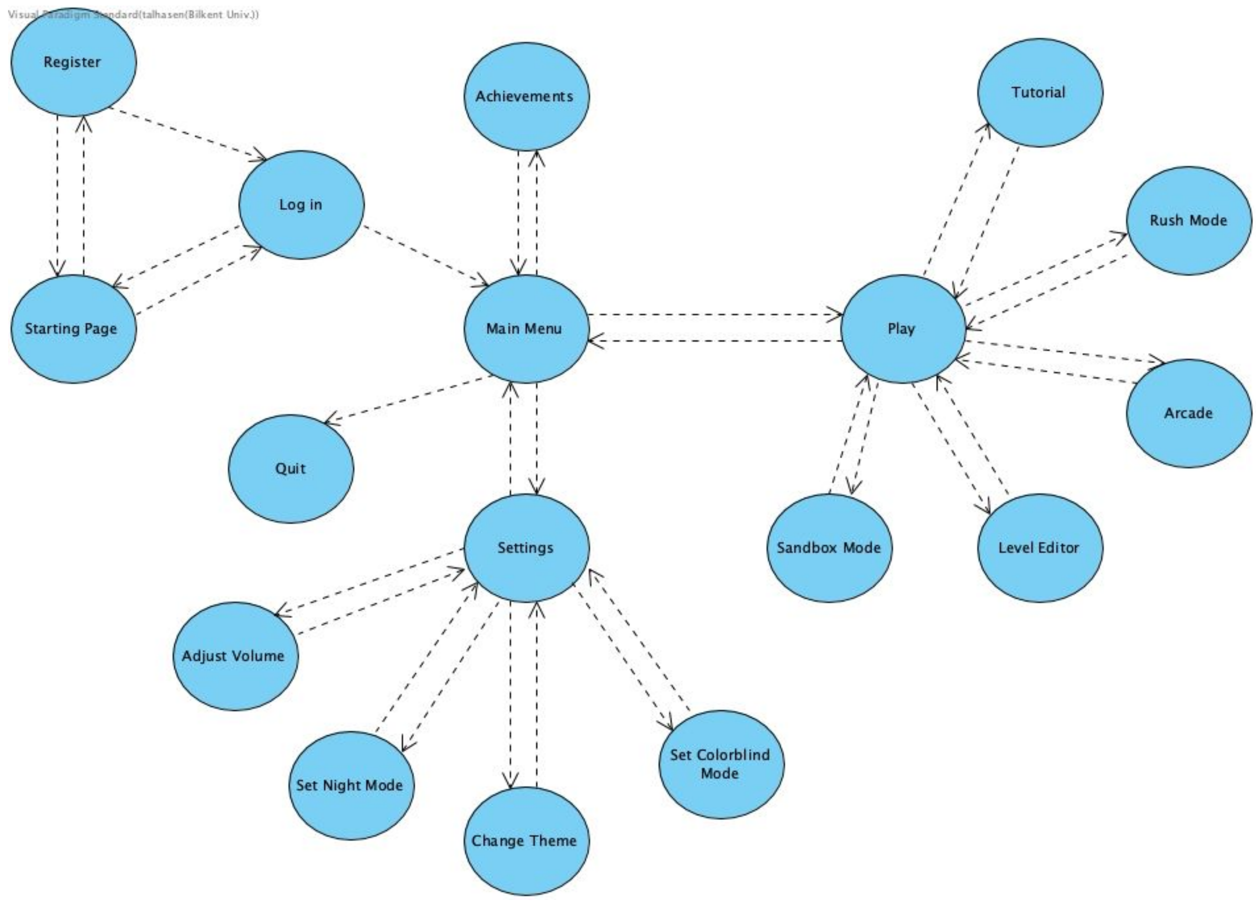
This class is a subclass of QuadScene and illustrates the how to play view before the game.

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.

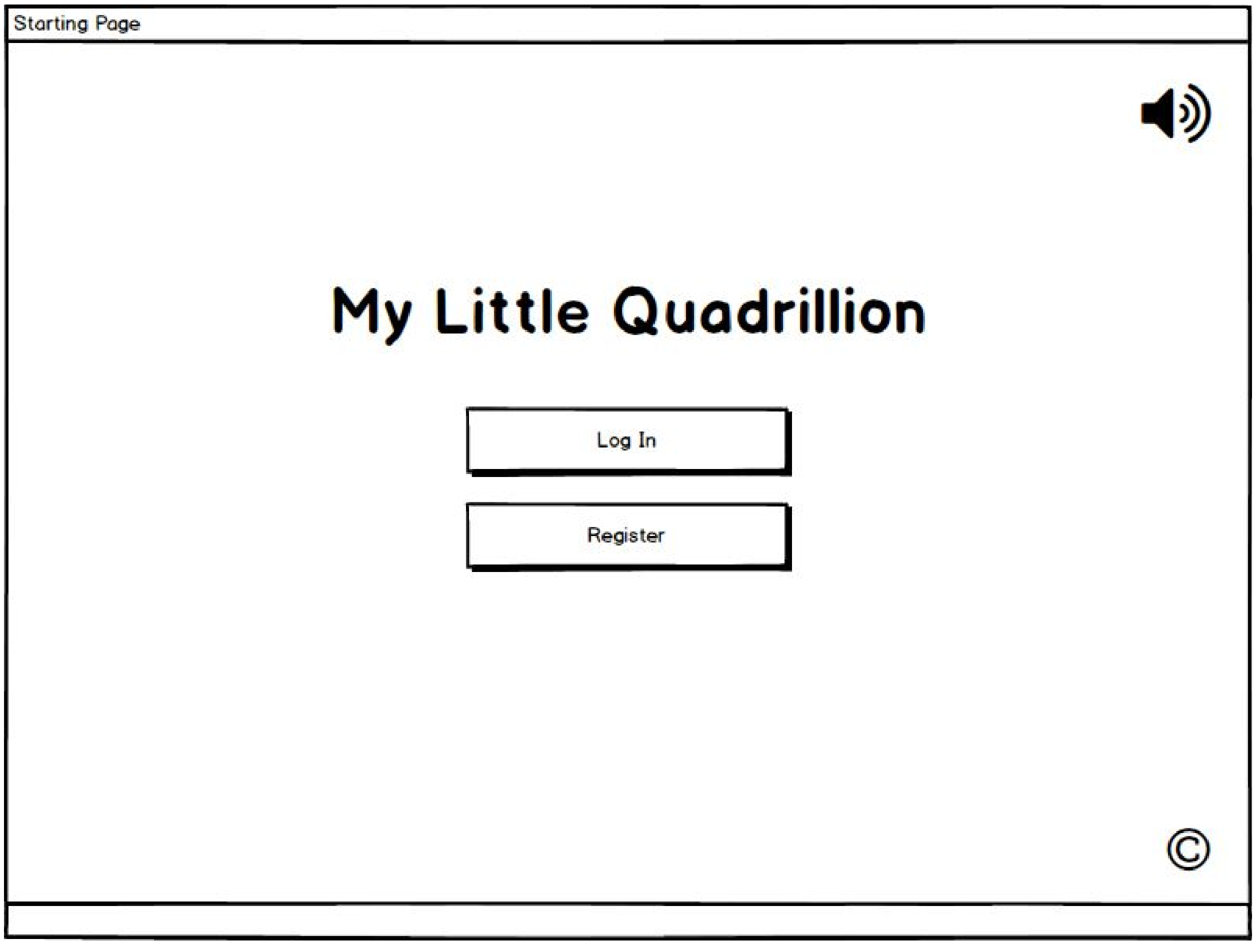
5.4. User interface

5.4.1. Navigational Path



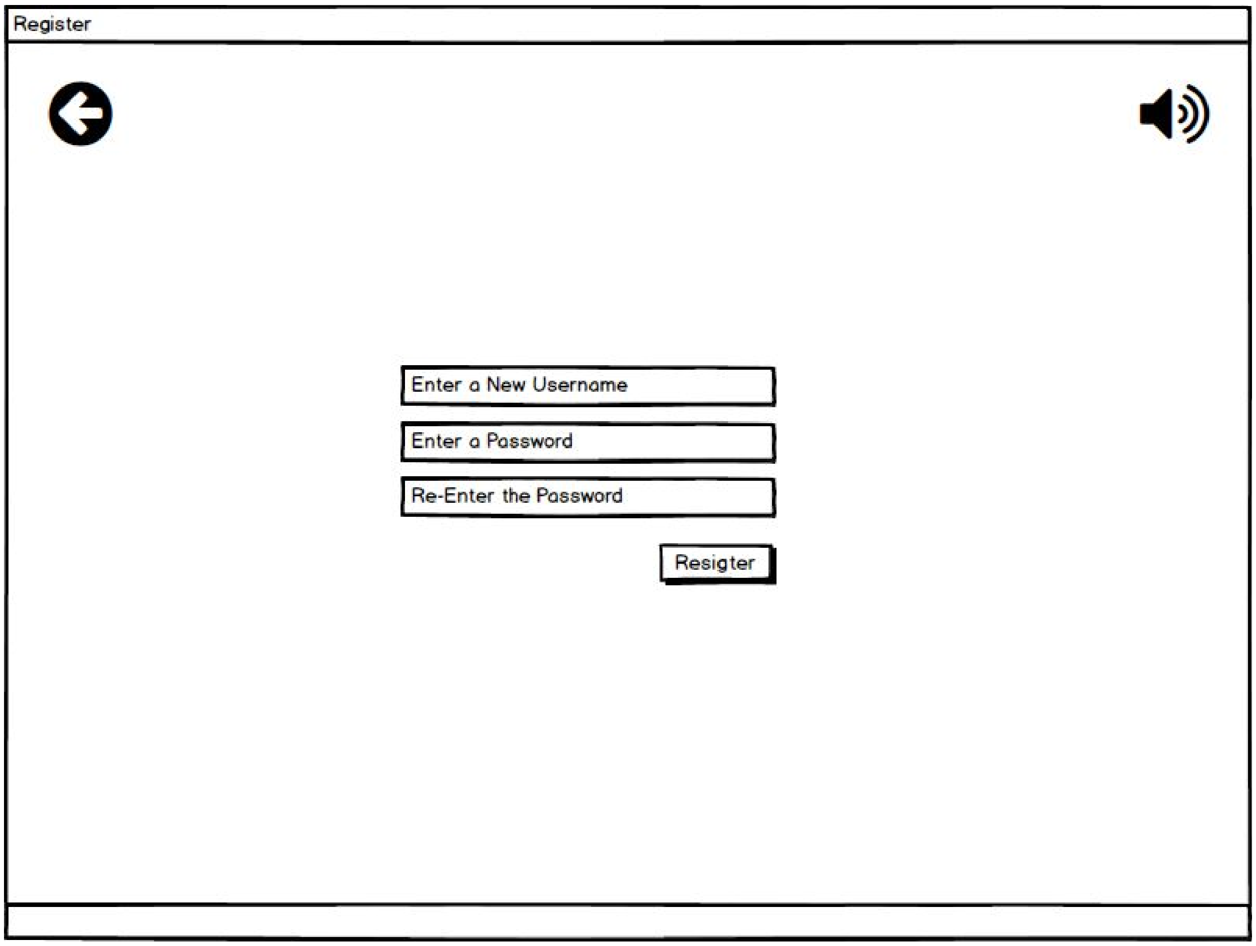
5.4.2. Screen Mock-ups

5.4.2.1. Starting Page

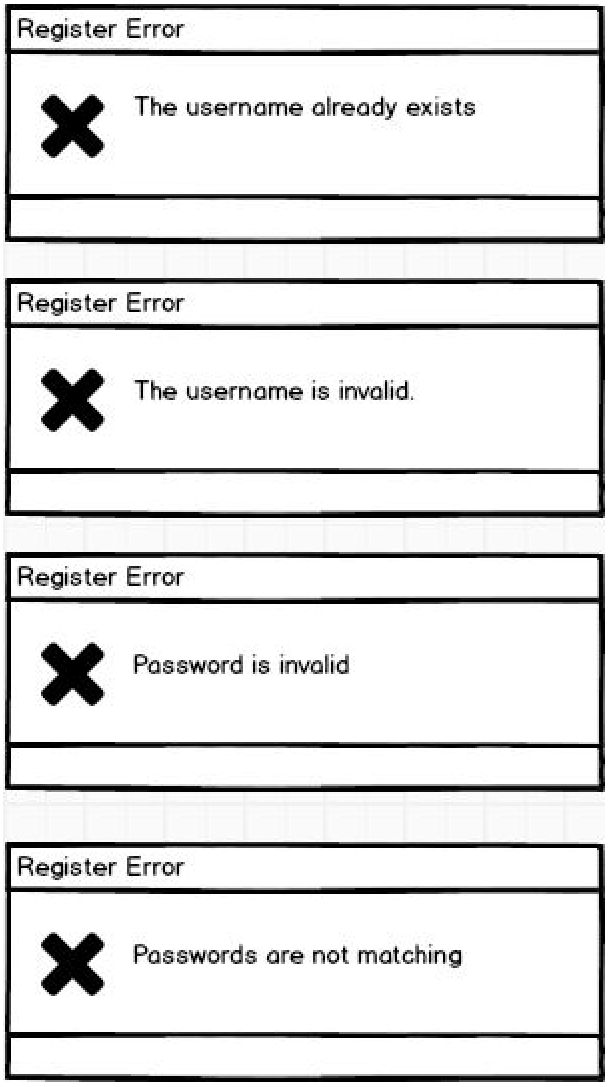


This is a starting page, it includes log in, register, sound and credits buttons. Volume button can stop or continue the sound. Credits button can give information about game and developers. Log in button for existing users and register button is for new players to enter a game.

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.


Register Error

✖ The username already exists

Register Error

✖ The username is invalid.

Register Error

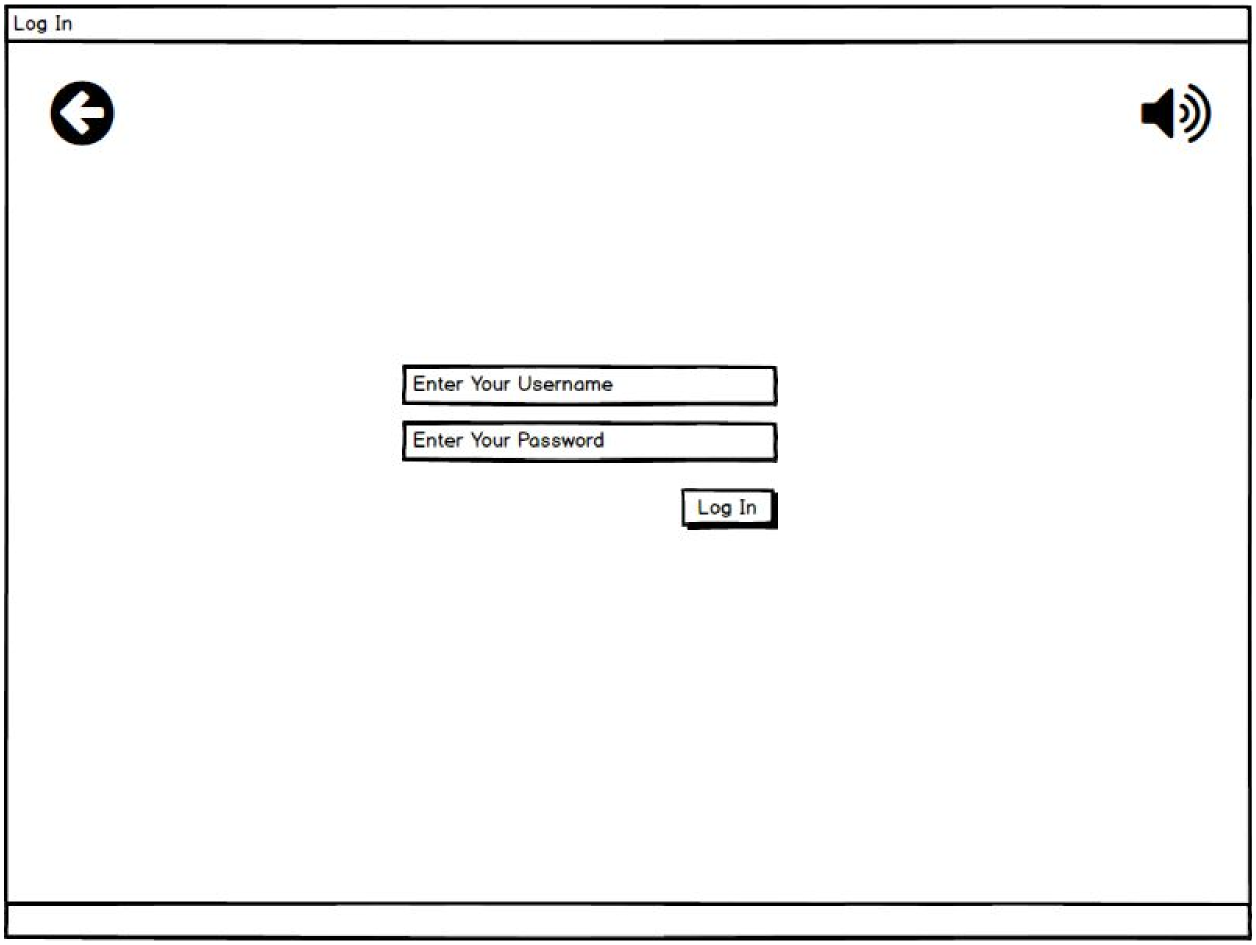
✖ Password is invalid

Register Error

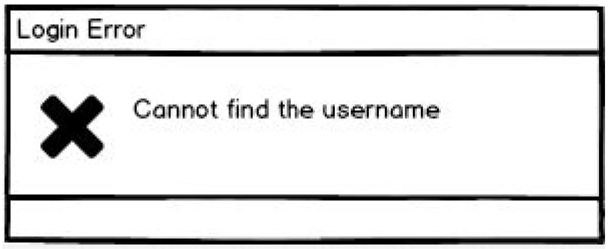
✖ Passwords are not matching

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

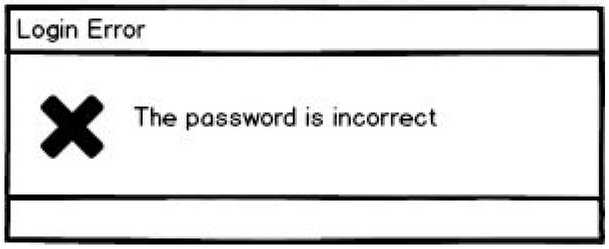


Player enters this page to log in to the game. After providing username and password player presses the login button to authorize. The page includes volume button and back button.



Login Error

Cannot find the username

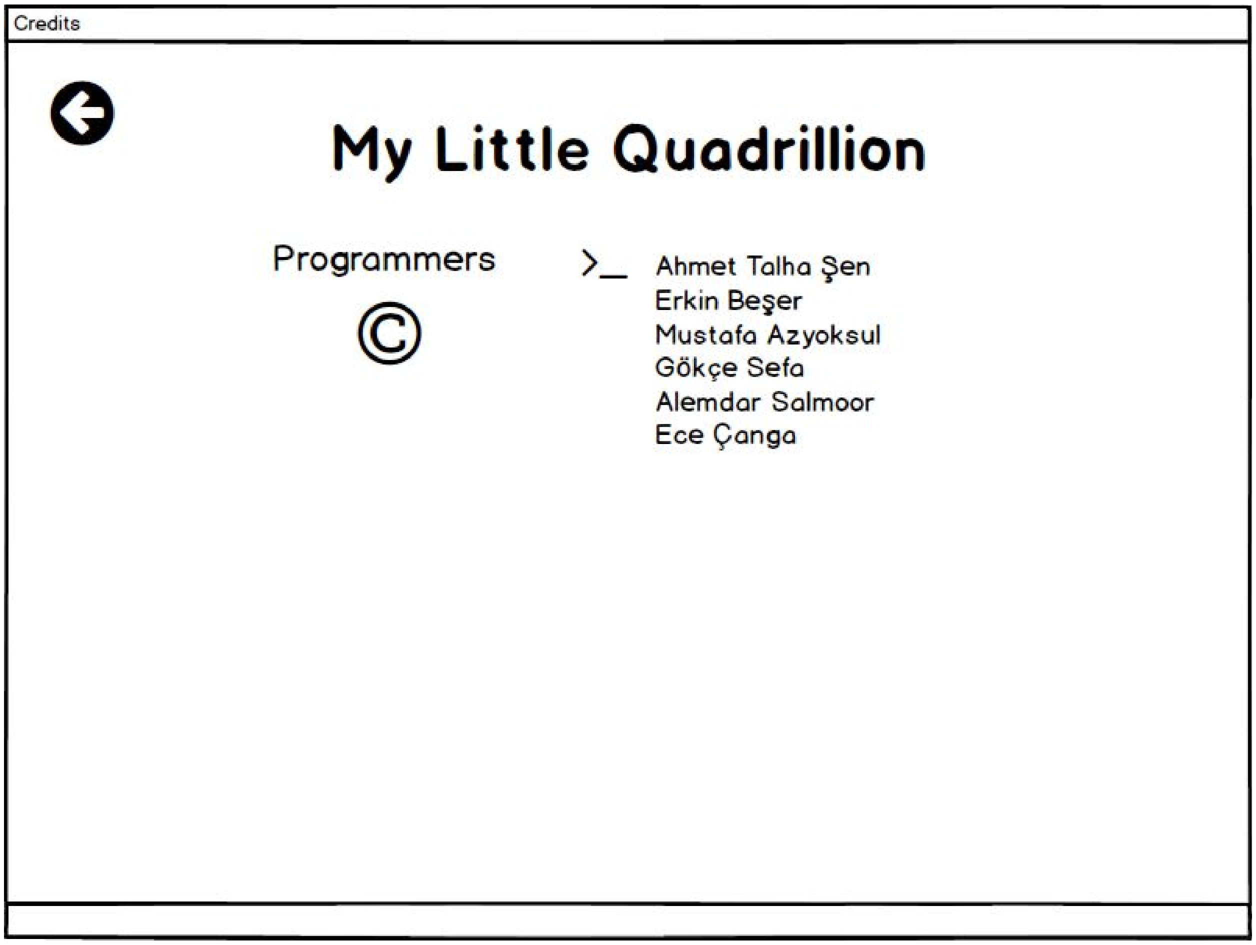


Login Error

The password is incorrect

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.

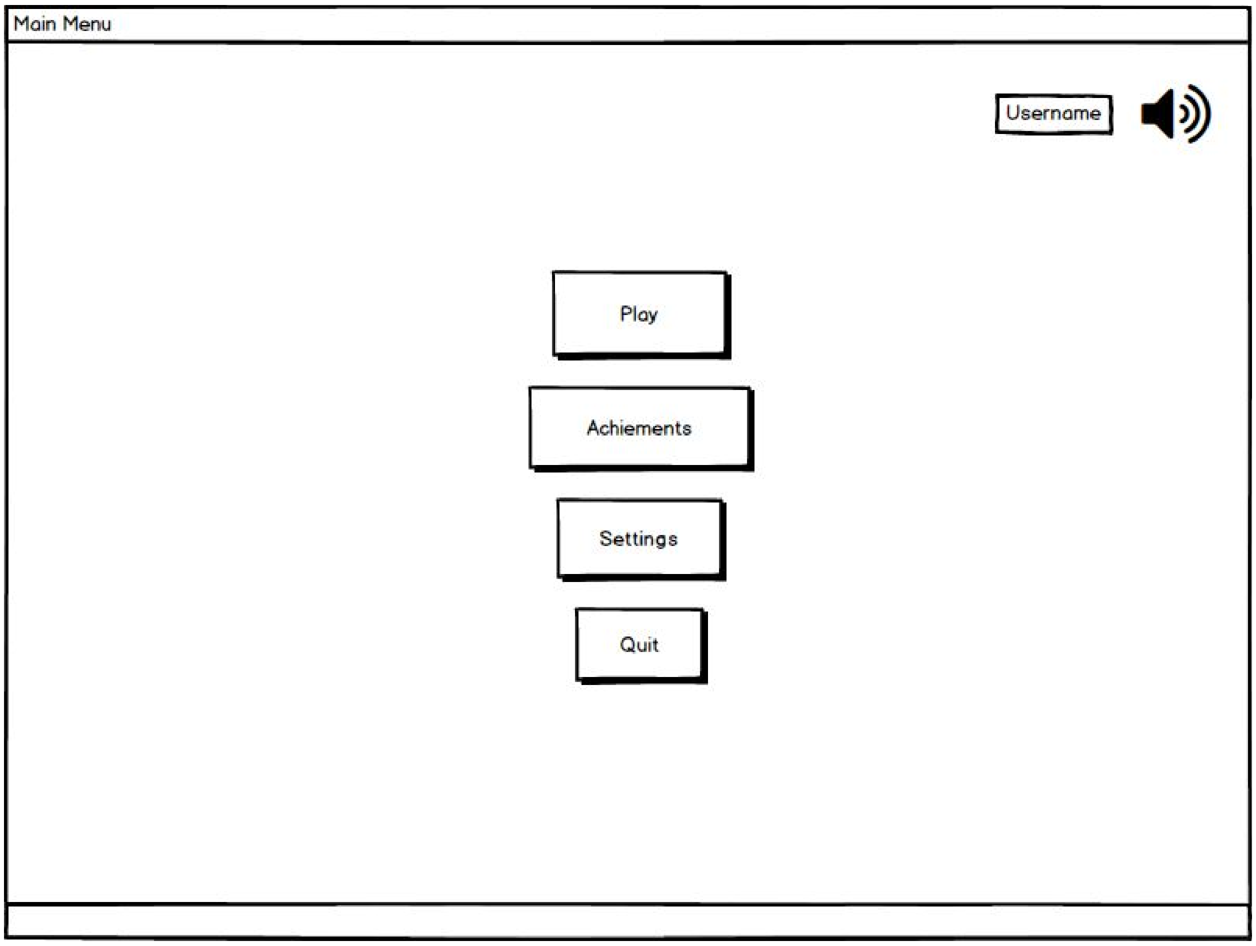
5.4.2.4 Credits Page



Ahmet Talha Şen
Erkin Beşer
Mustafa Azyoksul
Gökçe Sefa
Alemdar Salmoor
Ece Çanga

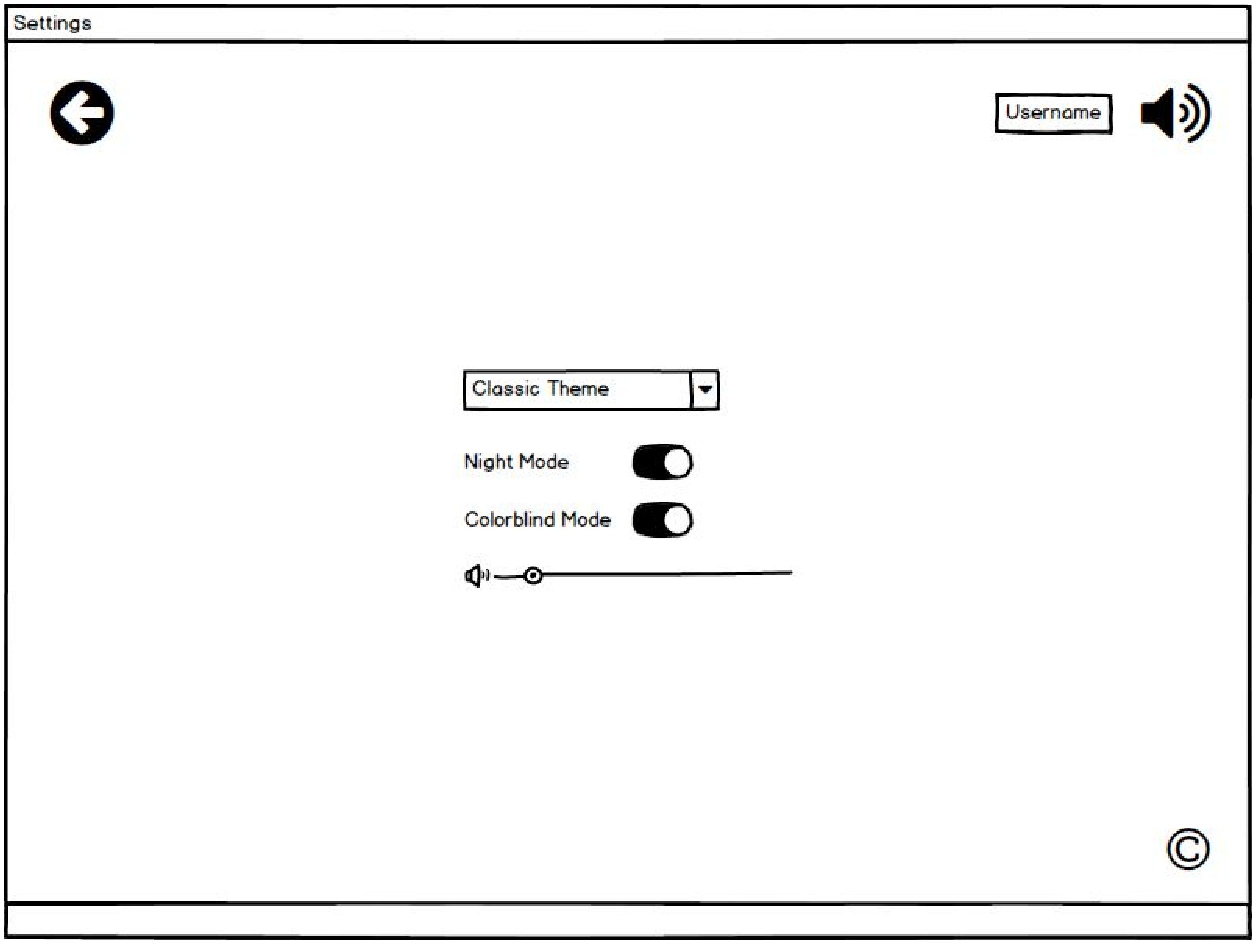
The Credits page includes information about the developers of My LIttle Quadrillion game. The page includes back button that return to the Main Menu.

5.4.2.5 Main Menu Page



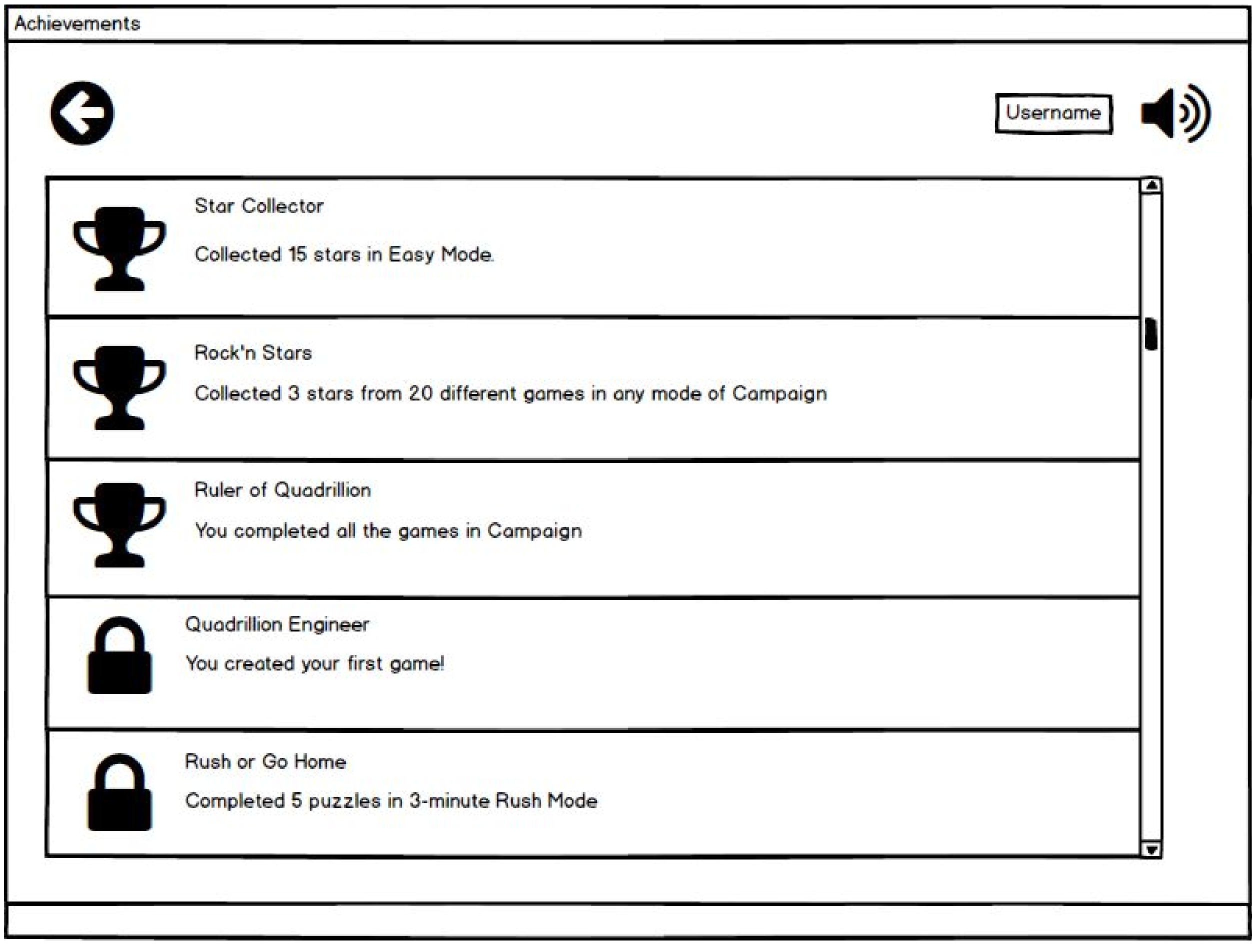
This is the Main Menu page. Player can go the Play manu, Achievements, and Settings. Player can also Quit the game if they wish to. Pressing sound button silences the sound.

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



Star Collector

Collected 15 stars in Easy Mode.

Rock'n Stars

Collected 3 stars from 20 different games in any mode of Campaign

Ruler of Quadrillion

You completed all the games in Campaign

Quadrillion Engineer

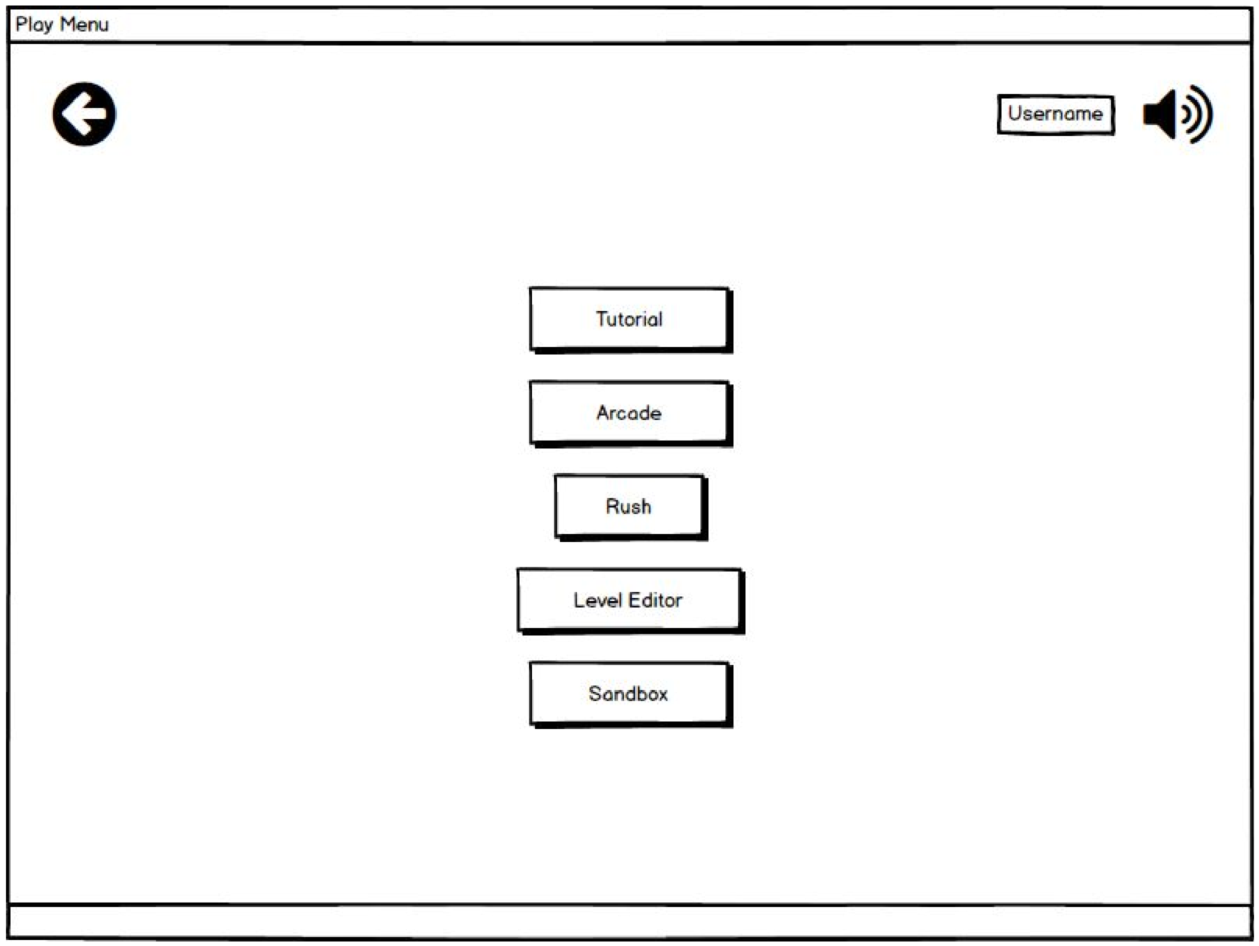
You created your first game!

Rush or Go Home

Completed 5 puzzles in 3-minute Rush Mode

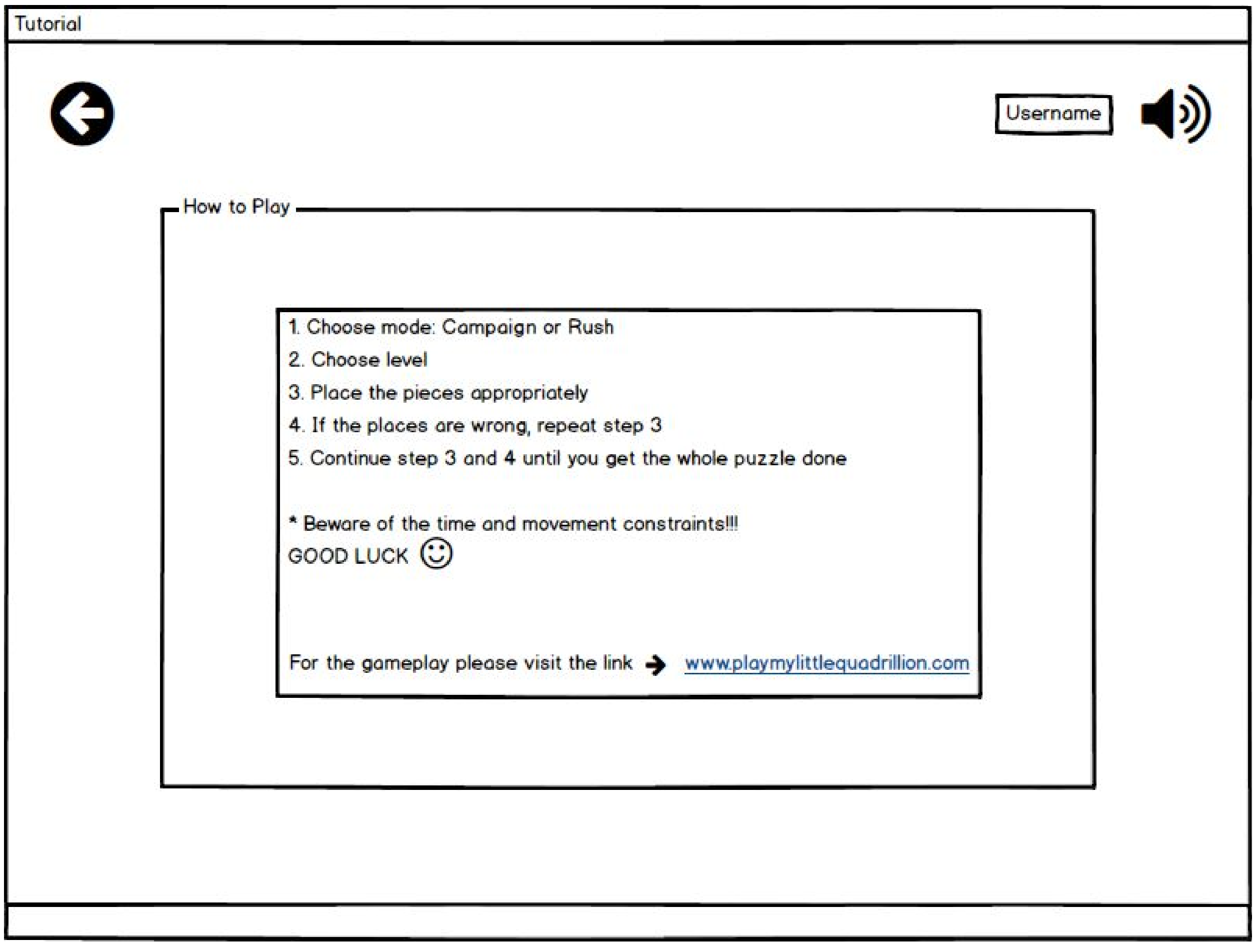
This is the page that displays the achievements of the player that they were able to collect from different modes of the game. Back button takes the player to the Main Menu.

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



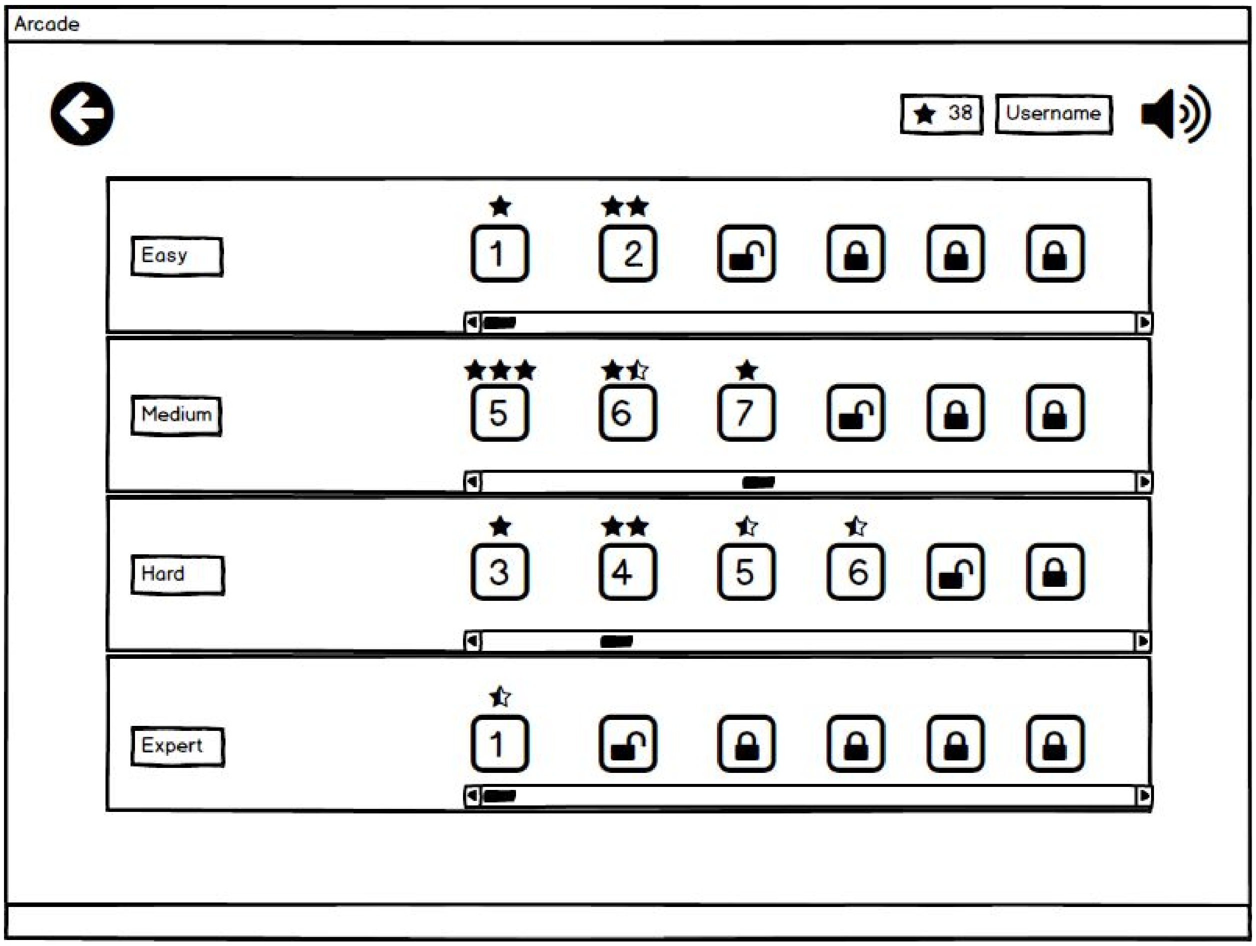
1. Choose mode: Campaign or Rush
2. Choose level
3. Place the pieces appropriately
4. If the places are wrong, repeat step 3
5. Continue step 3 and 4 until you get the whole puzzle done

* Beware of the time and movement constraints!!!
GOOD LUCK ☺

For the gameplay please visit the link ➔ www.playmylittlequadrillion.com

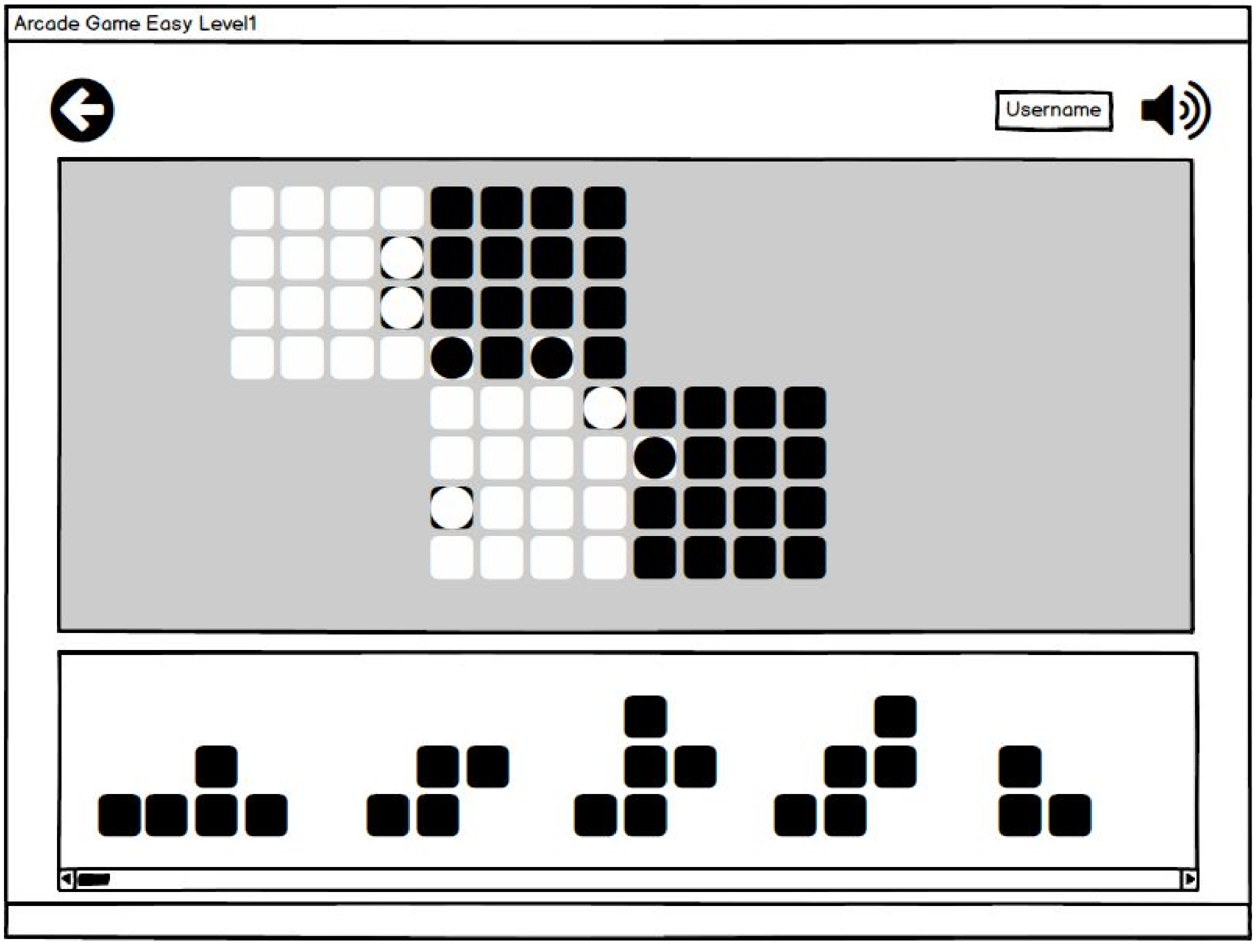
This page includes as information that can assist the player in Learning the Game. The page includes short algorithm to play the game, however if it is insufficient and player wishes to watch gameplay they can follow the link that takes them to the game demonstration. Back Button takes the user to Play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

5.4.2.10. Arcade Page



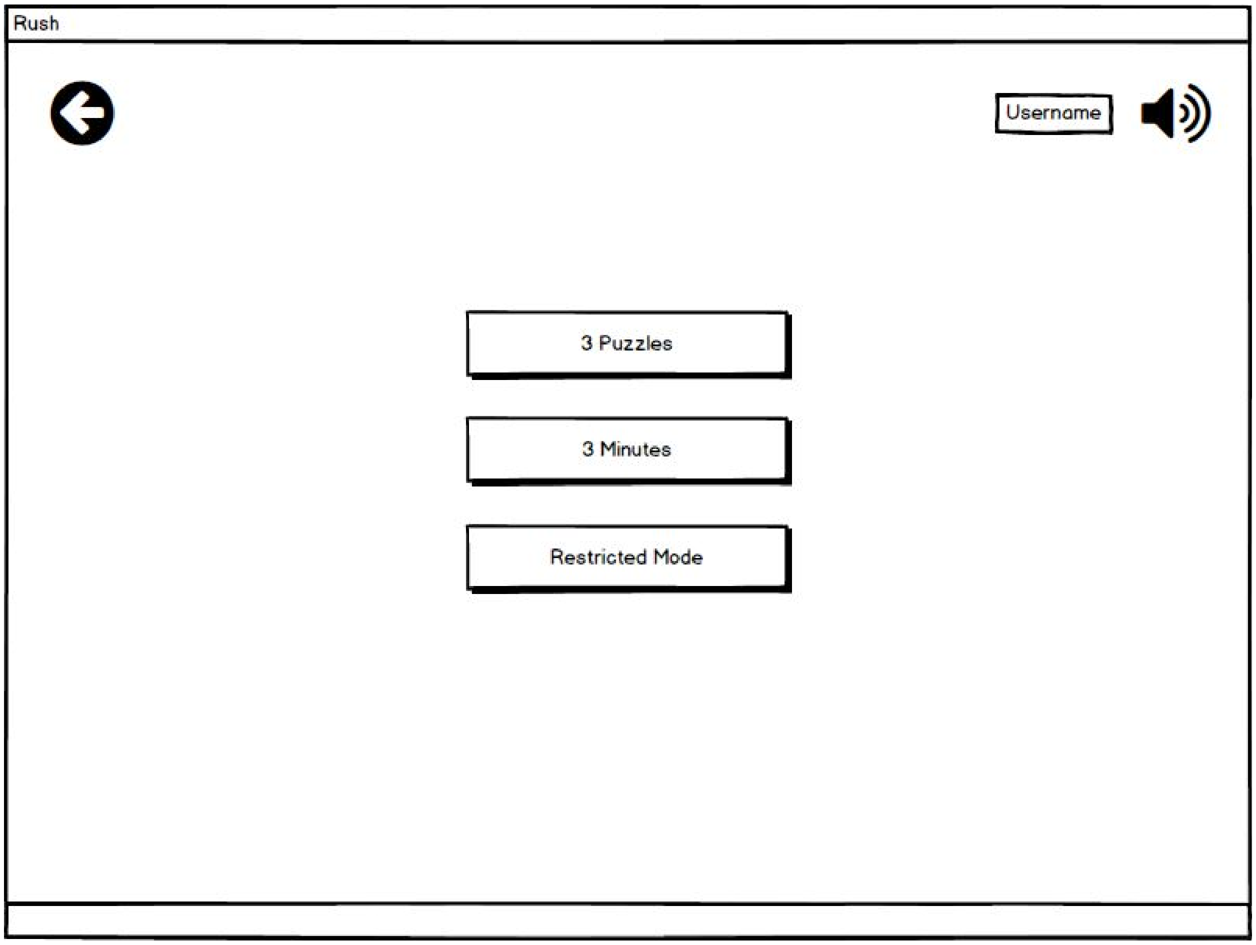
This is the page that is shown to the user when he presses Arcade from Play menu. Player can continue in any one of the difficulty levels they left off. Locked levels are indicated by locked lock sign. Unlocked and passed levels are indicated with level numbers with stars above them that correspond to 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



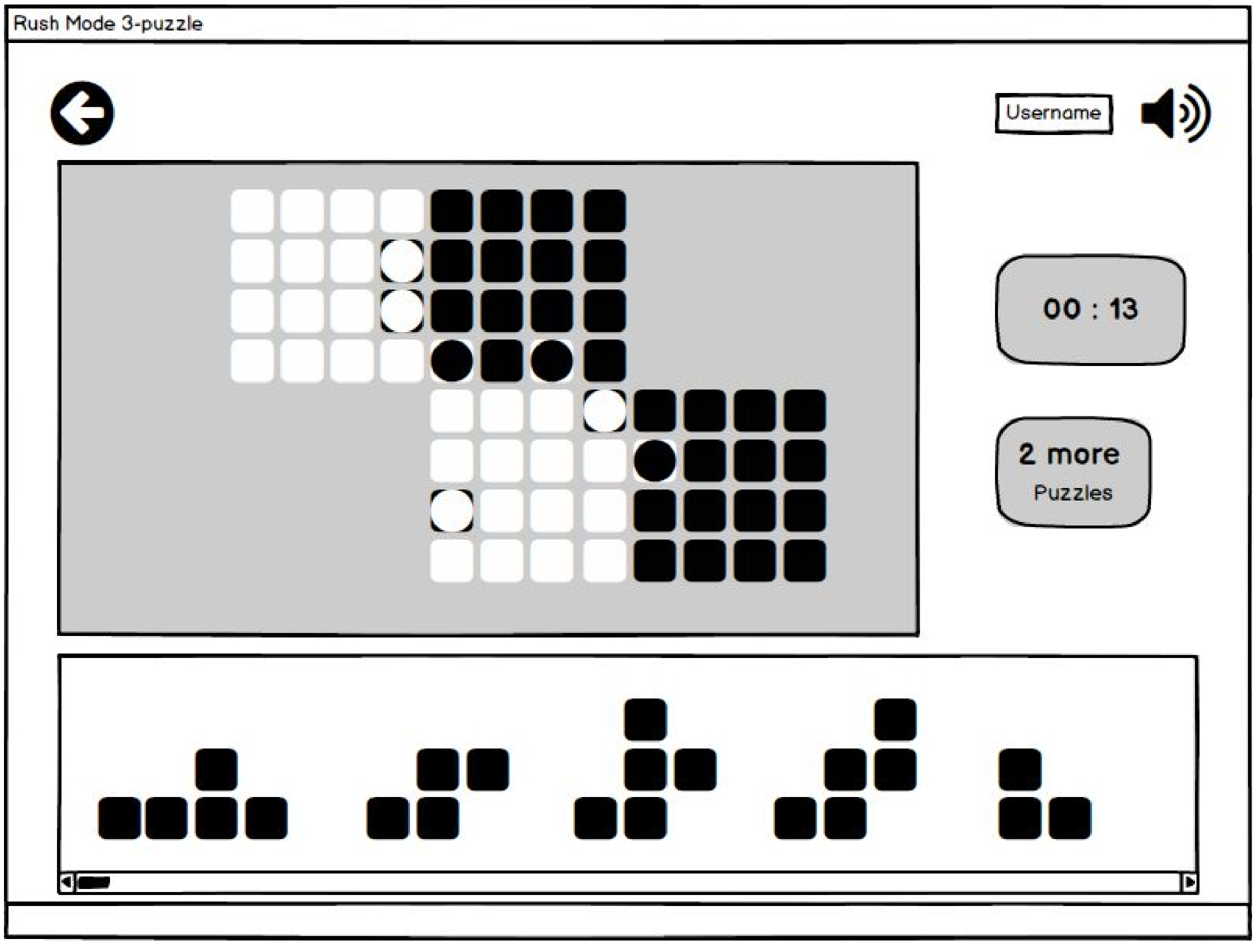
This is an example of easy Arcade Mode difficulty level. Player is challenged to solve the puzzle. If the player succeed, system gets the player back to arcade page and updates the progress and shows the stars collected by the player for this level. Player can go back by pressing left arrow key but the 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



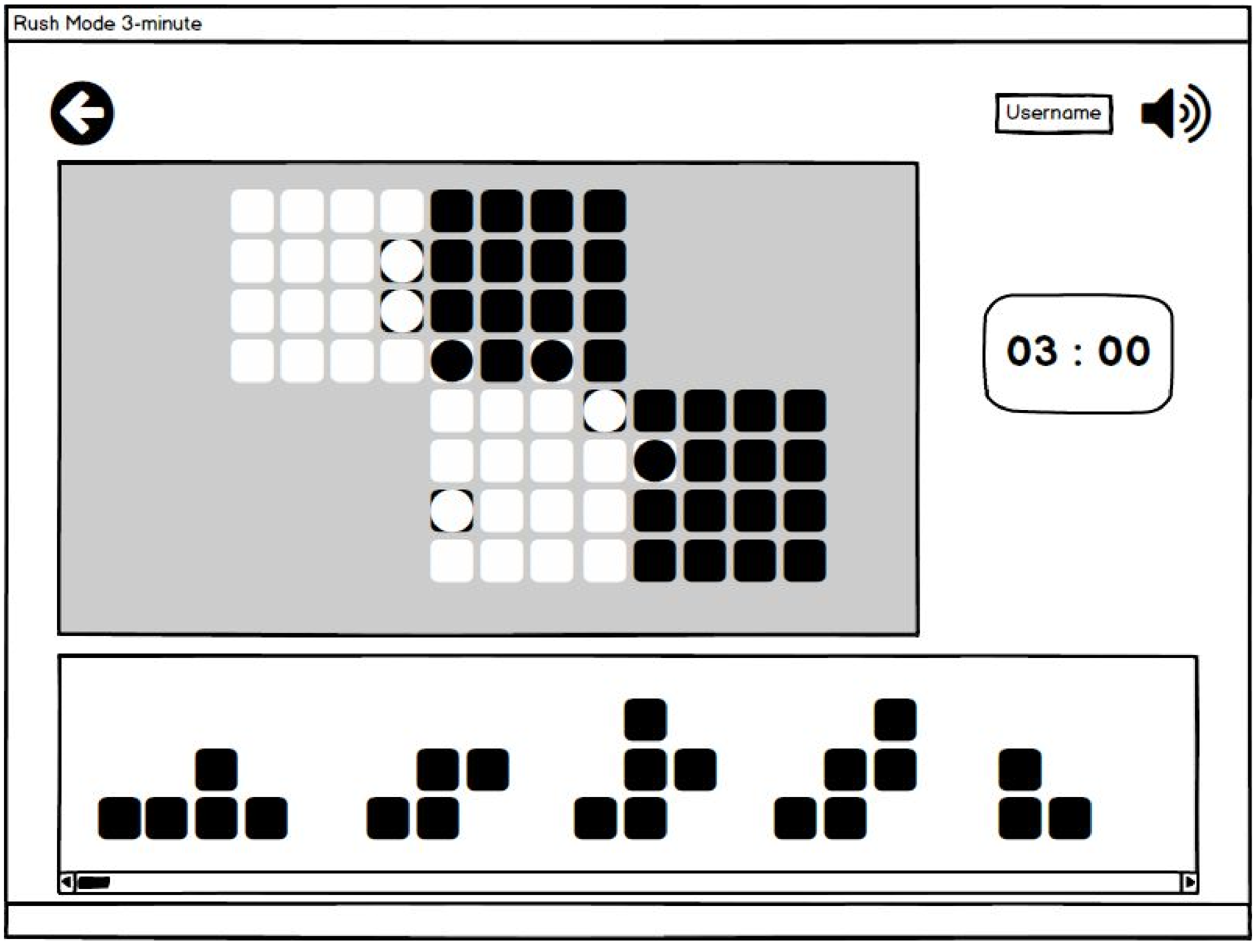
This is the Rush Mode page. User can choose which of the Rush Mode game types to play. Back button takes the user to Play Menu.

5.4.2.13. Rush Mode 3-puzzle Page



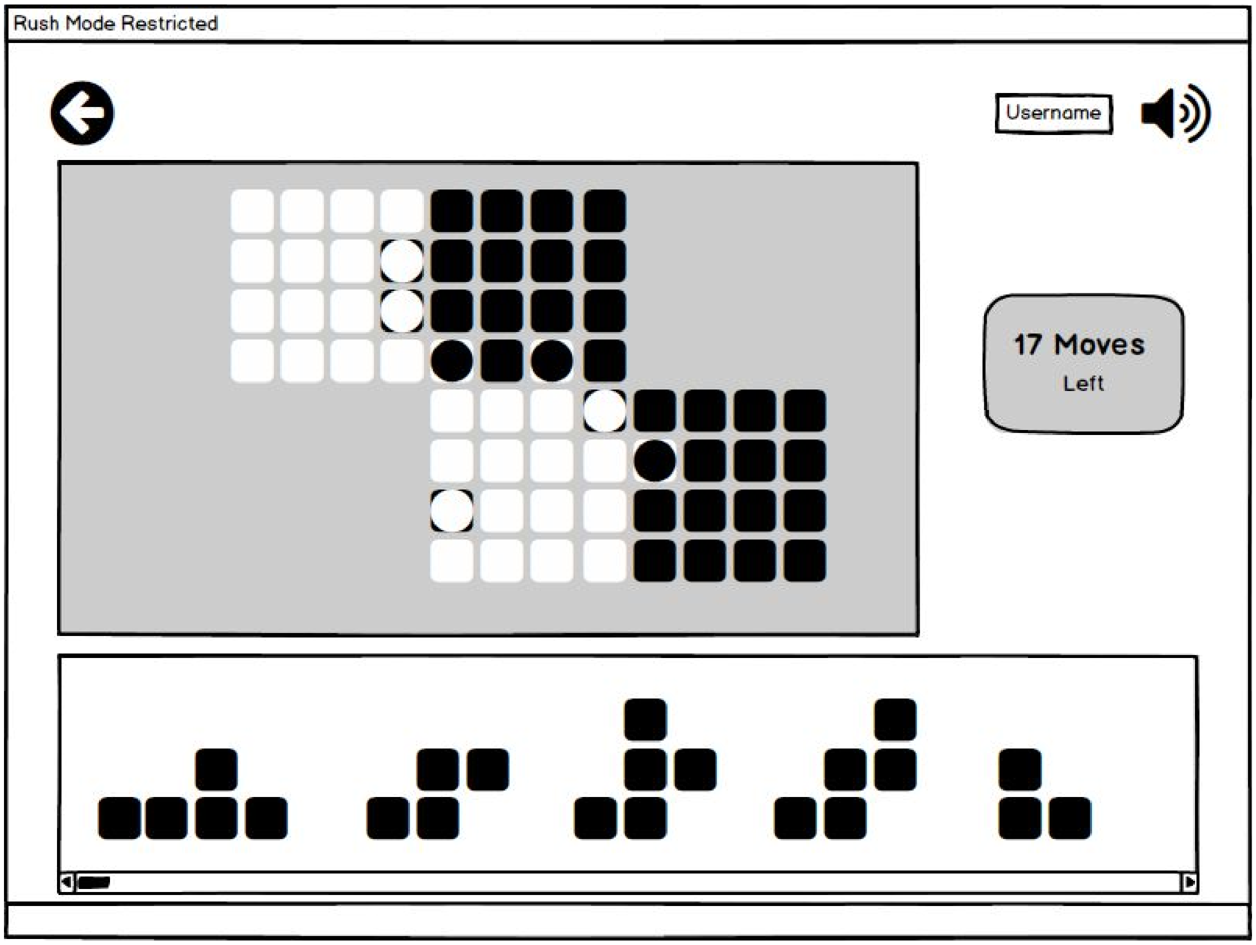
This page is shown to the user if he enters 3-puzzle Rush Mode. The page includes time that has passed from the start of the game. Additionally, player can see how many puzzles are left out of three puzzles. Back Button takes the player to Rush Mode Page where he can choose which type of Rush Mode they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

5.4.2.14. Rush Mode 3-minute Page



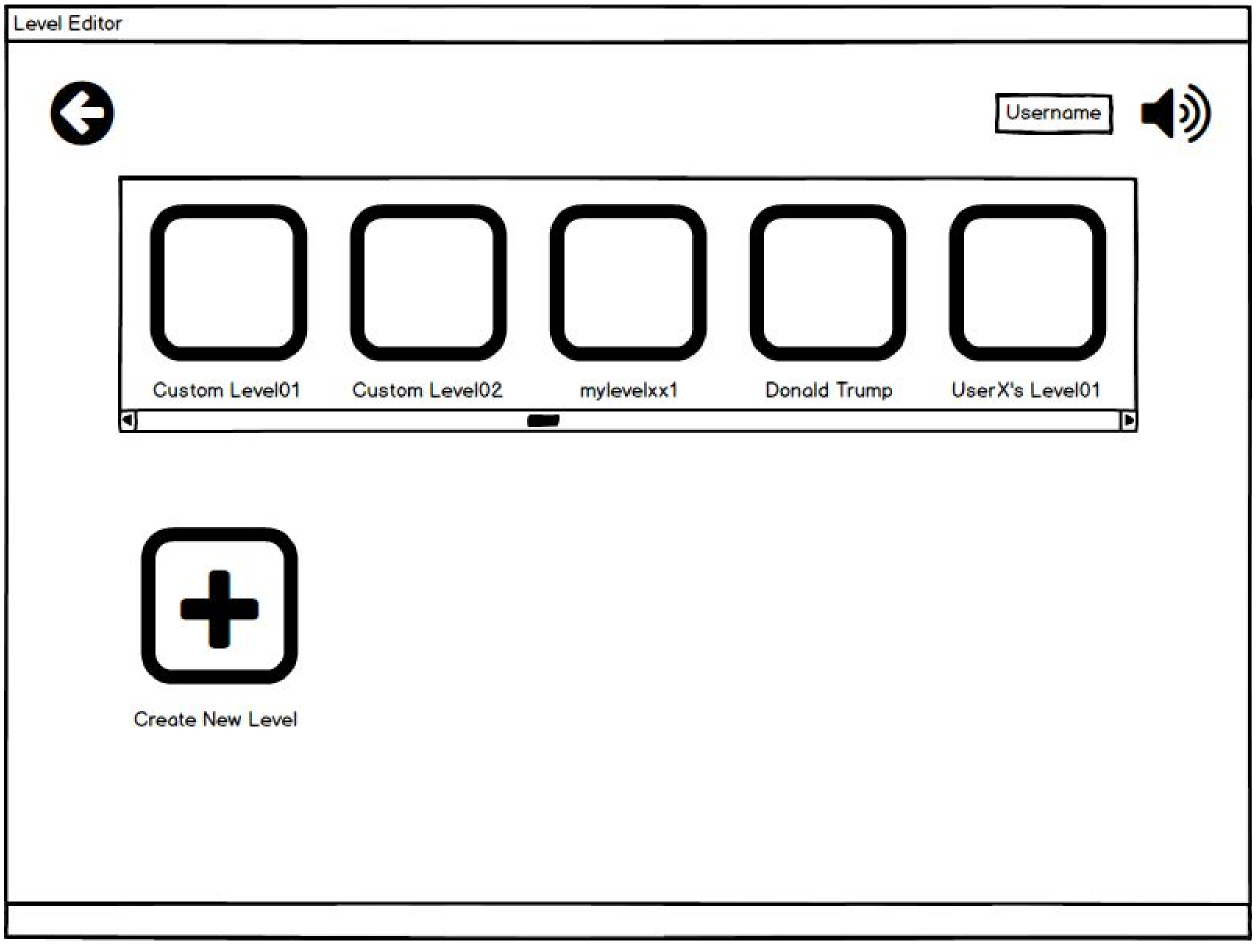
This page is shown to the user if he enters 3-minute Rush Mode. The page includes time that has left until the end of the game. Back Button takes the player to Rush Mode Page where he can choose which type of Rush Mode they would like to play.

5.4.2.15. Rush Mode Restricted Page



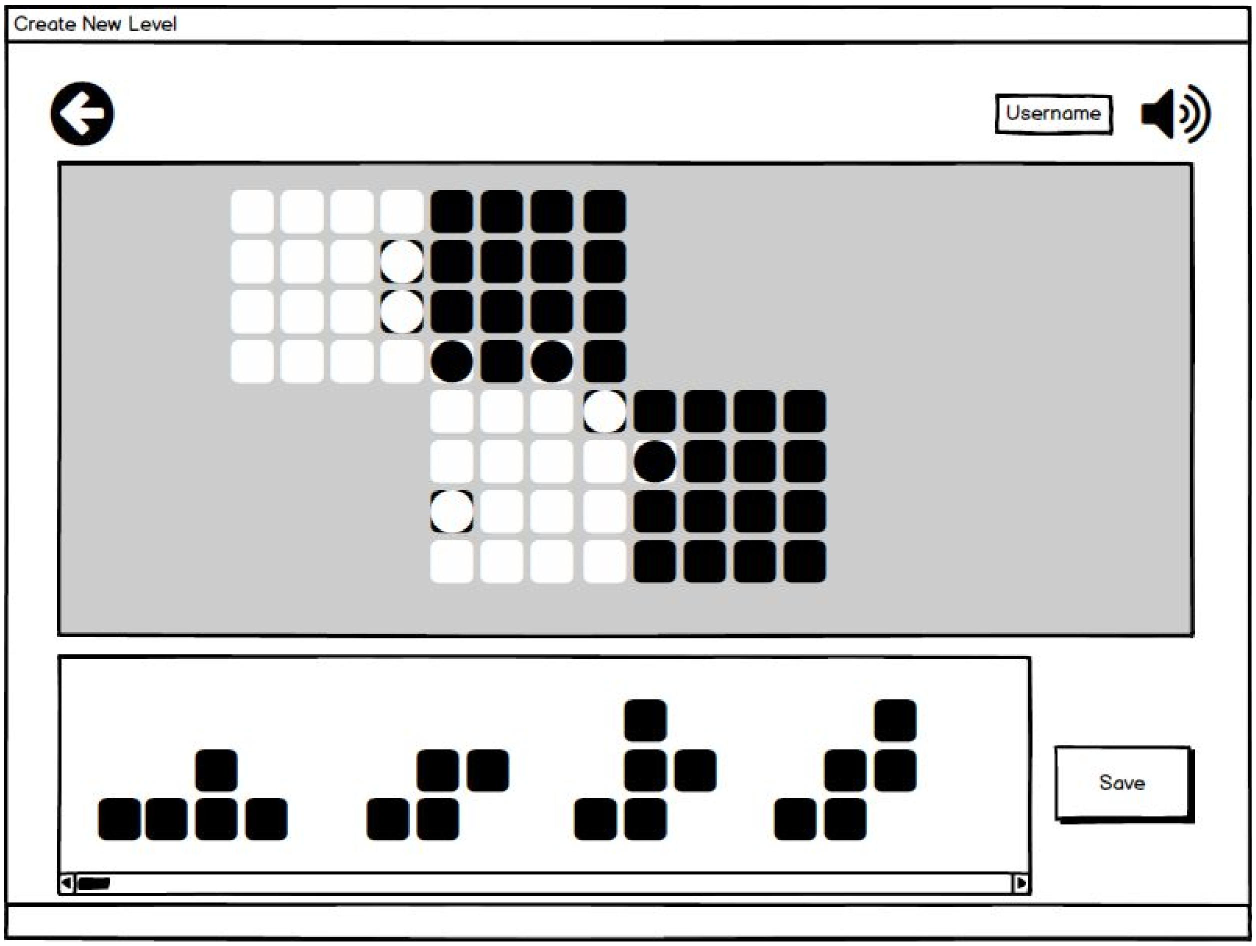
This page is shown to the user if he enters Restricted Rush Mode. Player can see how many disposable moves are left in the Restricted Mode . Back Button takes the player to Rush Mode Page where he can choose which type of Rush Mode they would like to play.

5.4.2.16. Level Editor Page



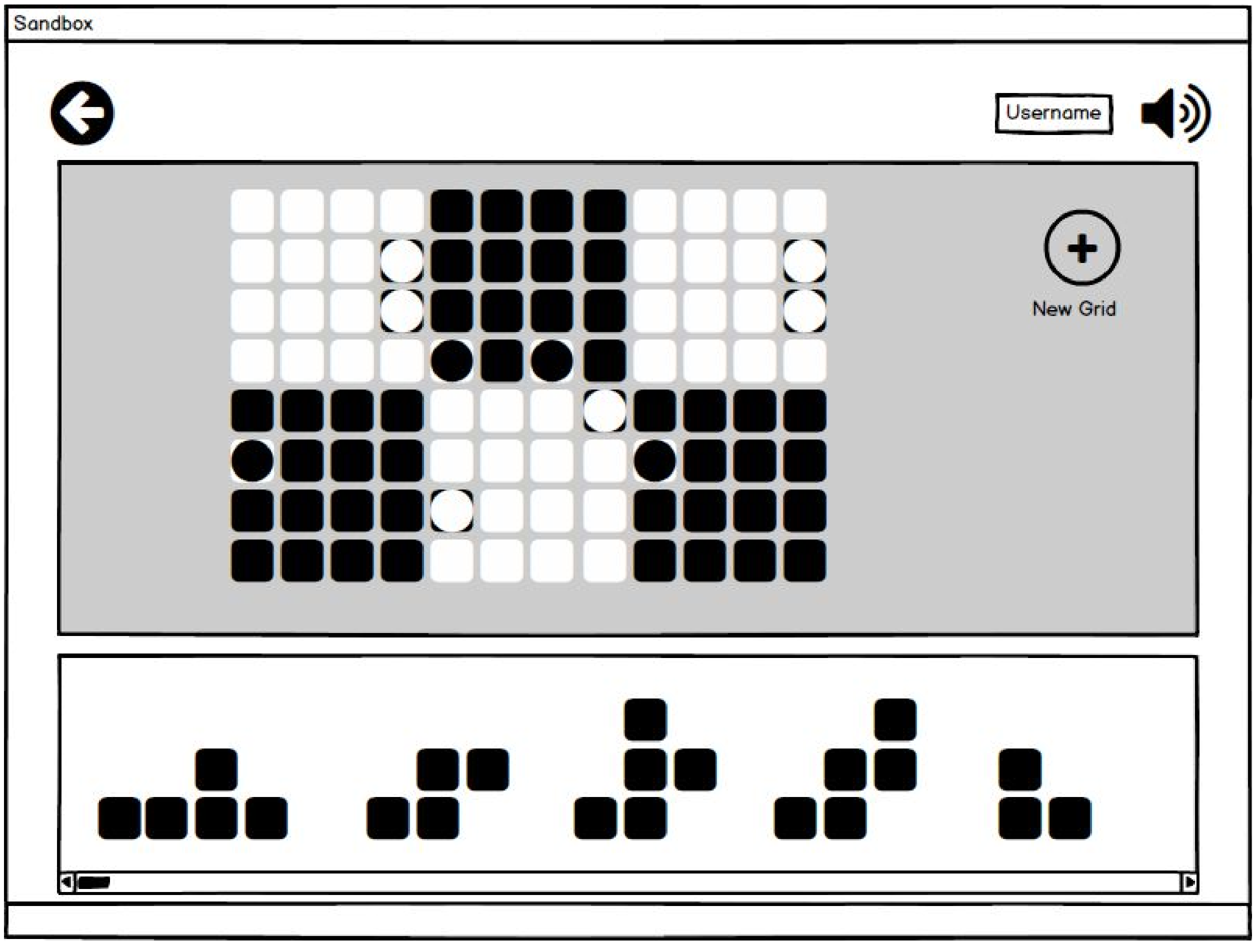
This is the page where player can see both his levels and other players' levels. In order to create a new level player presses Create New Level Button and system directs the player to the creating page.

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.

6. Improvement Summary

- Returns are added to navigational paths according to TA’s feedback
- State Diagram of Level Editor is updated according to TA’s feedback and is now consistent with State Diagram guidelines and depicts the situation with better depth.
- Use case diagram is thoroughly revised and updated. Specifically, order of execution where existed is removed. In game Quadrillion specific use cases are added. Database as an actor is removed from the use case.
- All types of login error use cases are combined into one, all types of register error use cases combined into one to be in tact with high level nature of the use case diagram.
- Activity diagram to model the arcade mode from user perspective is added.
- Activity diagram to model the the registration process from user perspective is added.
- Three more state diagrams that model Rush Mode are added.
- Nonfunctional Requirements are updated to be more testable. New non functional requirements are added. Nonfunctional Requirements are now specified under usability, reliability, performance, supportability.
- Functional requirements are updated to grasp the more of the in game functionality and Quadrillion specific functionality, like rotate, flip, drag, and place.
- General update of the Iteration 1 Analysis document with subtle additions and removals throughout the document in accordance with our better understanding of the system and what is required from after meeting with our TA and in class presentation.

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