



Bilkent University

Department of Computer Engineering

---

# **CS 319 - Object-Oriented Software Engineering Project**

## **IQ Puzzler Analysis Report**

**Group Name: Violet**

### **Group Members**

Ege Akın  
Safa Aşkın  
Betim Doğan  
Alp Ertürk  
Elif Gülşah Kaşdoğan

# **Contents**

## **1. Introduction**

## **2. Overview**

### **2.1 Game Play**

### **2.2 Board**

### **2.3. Puzzle Pieces**

### **2.4 3D Shapes**

### **2.5 Settings**

## **3. Functional Requirements**

### **3.1 New Game**

#### **3.1.1 2D Mode**

#### **3.1.2 3D Mode**

### **3.2 Tutorial**

### **3.3 Load Game**

### **3.4 Sign In**

### **3.5 Sign Up**

### **3.6 Leaderboard**

## **4. Nonfunctional Requirements**

### **4.1 Game Performance**

### **4.2 User Interface**

### **4.3 Extendibility**

## **5. System Models**

### **5.1 Use Case Models**

### **5.2 Object and Class Diagram**

### **5.3 Sequence Diagrams**

### **5.4 Dynamic Diagrams**

#### **5.4.1 Activity Diagram**

#### **5.4.2 State Chart Diagram**

### **5.5 Screen Mockups and Icons**

### **5.6 Navigational Path**

## **5. Conclusion**

## **6. Glossary & References**

## **1. Introduction**

IQ Puzzler Pro is a brain-teasing board game for children and adults which support the development of cognitive skills. It is a puzzle which has geometric pieces of different sizes and colors; the main purpose is correctly placing these pieces to board or creating a specific shape with the given pieces. Game starts with 2D which corresponds to filling the board and continues with 3D shapes.

For our course project, we will implement a different version of this game. IQ Puzzler is a desktop game which allows users to enjoy both the experience of classical board game and new features. Since IQ Puzzler is a suitable game for object oriented design we were able to add some additional features as we desire.

We will use Java platform to implement this game. We are planning to use Libgdx or Slick 2D for gaming framework, JavaFX for GUI design. We will apply object oriented design principles we will learn in CS319 during the project.

## **2. Overview**

### **2.1 Game Play**

IQ Puzzler is a single player game. It has two and three dimensional versions. Player has some puzzle parts with different sizes and shapes and a board to fill or a 3D pattern to satisfy to finish the game successfully. Pieces will be given such that there will be one unique solution for each level. Level is successfully finished if the given shape is created by given pieces. Players will obtain a score for each successful level. Time and count of moves are not important to finish the game but it will be important for score.

### **2.2 Score and Time**

We will add timer and score a system based to help users see their improvement also they can play this game in a competitive manner thanks to this feature. Score will be

calculated depending on time spent on finishing the level and number of moves made.

### **2.3 Puzzle Pieces**

Pieces are shaped as 4 to 6 balls are connected with different combinations. Each type of piece has a specific color so it can be identified based on color or shape independently. Their structure allows them to form expected 3D shapes.

### **2.4 Board**

Pieces will be inserted to the board until board is completely full for 2D. For 3D version, board will be used to form the base level of the 3D object.

### **2.5 3D shapes**

Expected 3D shape will be given to player and player will try to obtain this shape by using combination of these pieces.

### **2.6 Settings**

User will be able to alter the board or color of background. Also he/she will be able to change the color of pieces

## **3. Functional Requirements**

### **3.1 New Game**

There will be 2 game modes which are 2D game mode and 3D game mode. If user starts a new game, if there exists any saved game in current mode (2d or 3d), the saved game in current mode will be destroyed. So user will only be able to save 1 game for 2d game mode and 1 game for 3d game mode.

### **3.1.1 2D Game Mode**

When user log in to the game, he/she will see a menu which is composed of play game, load game, leaderboard and tutorial. If user click the button new game or load game there, user will see game mode selection screen. If user clicks 2D game mode, he/she will see the initialized board and pieces. In this game mode the difficulty of levels will be increased. The maps of all levels are generated by the system. The score of user will be increased if he/she passes the levels and after that there will be a leaderboard according to these scores.

### **3.1.2 3D Game Mode**

If user clicks 3D game mode in game mode selection screen 3D game will be initialized. User will be able to see the board and puzzle from various perspectives. Calculating the user's score in 3D game mode will be different than 2D since it is harder than 2D game mode. User will get more points in a 3D level than 2D level. The difficulty of levels in 3D game mode will also be increased.

## **3.2 Tutorial**

Player will see the tutorial button in the main menu when he/she login. Tutorial will help user about the buttons and how to play the game. Tutorial will also give a video description which informs the user about game more.

## **3.3 Load Game**

Player will be able to load game that he/she saved before. Load Game button is in the main menu. When user pushes the load game button he/she will get a 2d or 3d game mode selection window to choose which game mode that user will load from. If user pushes load game button he/she will see the successfully passed levels and count of stars of a level. There are maximum 3 stars in each level. If user gets 3 stars from one level, it means that he/she finished puzzle very quick.

And if he/she gets 1 star it means that he/she finished puzzle scarcely. So in load game feature user will be able to play passed levels again and for example he/she can increase the count of stars of a level 2 to 3. Load game will also stores the last unfinished puzzle and user will be able to see put pieces in last level that is played.

### **3.4 Sign In**

When user starts the game, firstly he/she will see log in screen. If user had an account he/she will be able to enter the username and password and immediately start game. In sign in screen, there will be check for validity of username and password. If user does not have an account, game system will not allow user to start game immediately.

### **3.5 Sign Up**

When user starts the game, in initial screen there will be a sign up button. If user does not have an account he/she will pick the sign up button for creating a new account to play the game. In sign up screen, user will create a new username and password. Game system will check if username is taken before, after a successful sign up user will be able to start the game.

### **3.6 Leaderboard**

Leaderboard button is in the main menu. When user presses leaderboard button, he or she will be directed to leaderboard screen. In leaderboard screen user will be able to see his/her ranking among the Iq Puzzler Pro players. Leaderboard also gives the list of best players. The ranking system calculates all rankings with respect to each player's score. Also in leaderboard user can type another username to the search field in order to find a specific user's ranking.

## **4. Nonfunctional Requirements**

### **4.1 Game Performance**

Due to make a game which works in high performance we are planning to have components which will not need much system requirement. In order to increase the game performance we stay away from using high resolution graphics. We are aiming that our game will not decrease the computer's performance.

### **4.2 User Interface**

User interface is one of the most important points of a game because users have their first interaction with user interface rather than game play or implementation. In order to have a nice user interface we are planning to have easy to use menus and screens and have good quality graphics.

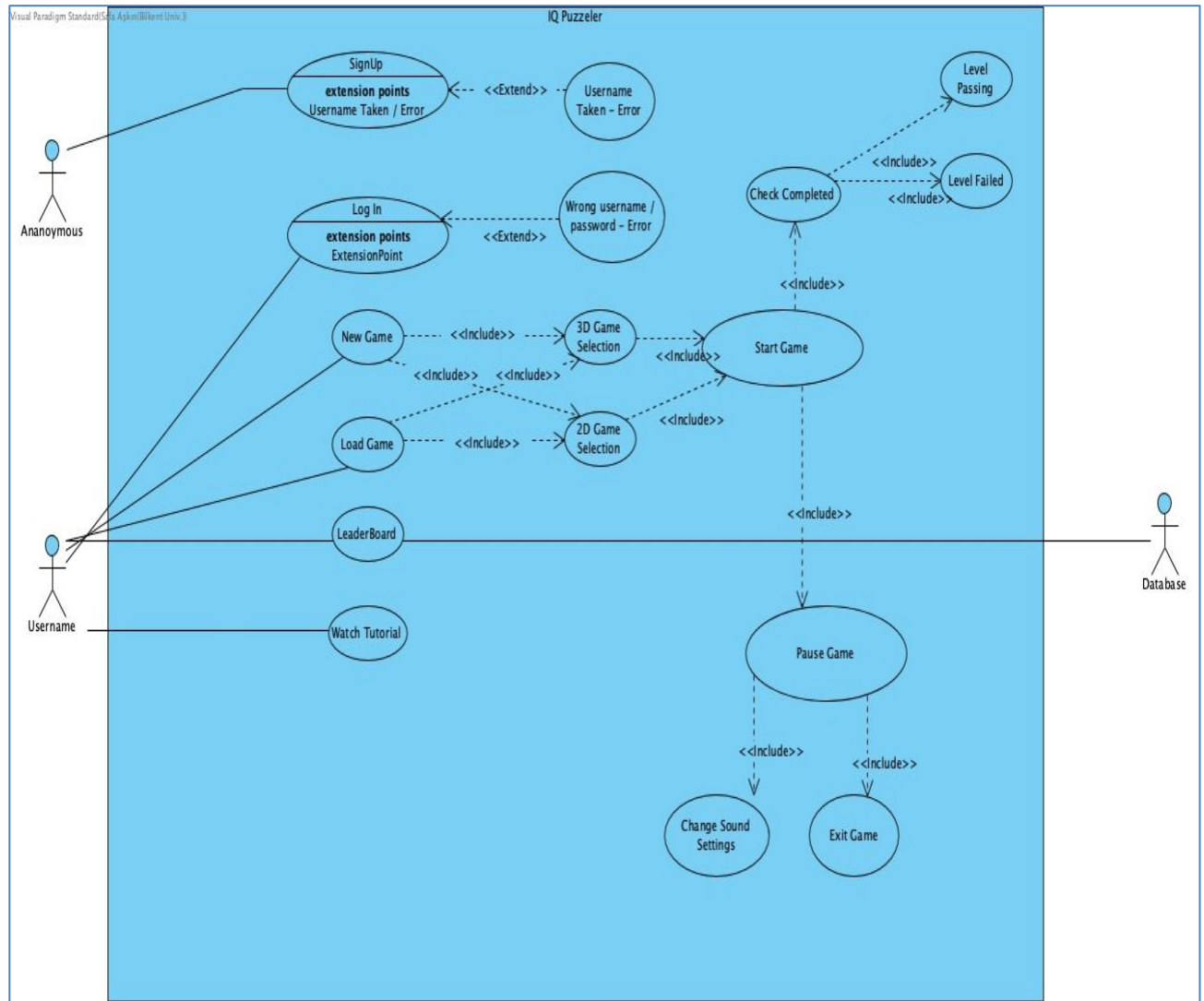
### **4.3 Extendibility**

According to object oriented design we are planing to have a game which will not need too much change when new and additional features will be added or bugs will be fixed.



## 5. System Models

### 5.1 Use Case Model



### Use Case Descriptions

#### 1) Use Case Name: Sign Up

**Participating Actor:** Player

**Stakeholders and Interests:** Player may create an user ID for saving his/her scores.

**Pre-condition:** User must be in the main screen.

**Post-condition:** -

**Entry condition:** Player clicks on sign up button in main screen.

**Exit condition:**

- Player clicks on back button in sign up screen.

**Main flows of events:**

- Player chooses an appropriate user name.
- Player chooses an appropriate password.
- System saves the user name and password and creates the new user ID.
- Player is directed into main screen.

**Alternative flows of events:**

- Player selects a username that is taken before and gets error.
- Player selects an inappropriate password and gets error.

**2) Use Case Name:** Log In

**Participating Actor:** Player

**Stakeholders and Interests:** Player may want to log in to play the game with his/her account.

**Pre-condition:**

- User must be in the main screen.
- User must have an account he/she created before.

**Post-condition:** -

**Entry condition:** Player clicks on log in button in main screen.

**Exit condition:**

- Player clicks on back button in log in screen.

**Main flows of events:**

- Player enters the true username and password.
- System checks whether the username and the password matches or not.
- Player is directed into main screen.

**Alternative flows of events:**

- Player enters the wrong username and password combination and system directs he/she to an empty log in screen.

**3) Use Case Name: New Game****Participating Actor: Player****Stakeholders and Interests:**

- Player may want to create a new iq puzzler game.
- System creates the game and starts the game.

**Pre-condition:**

- User must be in the main menu.

**Post-conditions: -**

**Entry conditions:** Player opens game and starts game by clicking relevant section.

**Exit condition:**

- Player clicks on back button in game screen, OR
- Player exceeds the determined game time, OR
- Player wins all of the levels.

**Main flows of events:**

- Player enters a new nickname and starts the game.
- System loads default settings and starts the game.
- Player places the pieces at unique spots correctly.

- Player completes the one of the levels.
- The obtained score is demonstrated on screen at the end of game.

**Alternative flows of events:**

- Player does not place the pieces correctly.
  - Player exceeds the time and game is over.
- Player gets bored and may want to close the game.
  - The game instructions are loaded and whenever the user loads the game, player continues from the position where he/she left.

**4) Use Case Name:** Pause Game

**Participating Actor:** Player

**Stakeholders and Interests:** Player may wants to pause the game.

**Pre-condition:** User must be playing the game.

**Post-condition:** -

**Entry condition:** Player clicks the “Pause” button in game screen.

**Exit condition:**

- Player clicks on back button in game screen, OR

**Main flows of events:**

- Player clicks the “Pause” button in the game screen.
- System displays pause menu.
- Player places the pieces at unique spots correctly.
- Player completes the one of the levels.
- The obtained score is demonstrated on screen at the end of game.

**Alternative flows of events:**

- If the player wants to continue the game,
  - Player presses “Continue” button in pause menu

- Game continues.

## **5) Use Case Name:** Load Game

**Participating Actor:** Player

**Stakeholders and Interests:** Player may want to continue previous uncompleted game.

**Pre-condition:** User must have record that he/she plays this game before and not completed yet.

**Post-condition:** -

**Entry condition:** Player clicks on load game button in main screen.

**Exit condition:**

- Player clicks on back button in 2D/3D mode selection screen.

**Main flows of events:**

- Player chooses the 2D or 3D mode in mode selection screen.
- Player starts the game in recorded level.
- System loads default settings and starts the game.
- Player places the pieces at unique spots correctly.
- Player completes the one of the levels.
- The obtained score is demonstrated on screen at the end of game.

**Alternative flows of events:**

- Player does not place the pieces correctly.
- Player exceeds the time and game is over, game instructions are loaded again and waiting for next connection of player.

## **6) Use Case Name: LeaderBoard**

**Participating Actor:** Player

**Stakeholders and Interests:** Player may want to view the scoreboard and indicate competition between users.

**Pre-condition:** Player has to be in main menu.

**Post-condition:** -

**Entry condition:** Player clicks on “LeaderBoard” button in main menu.

**Exit condition:** Player clicks on back button in leaderboard screen.

### **Main flows of events:**

- Player observes the scores of different users and the leaderboard of the game.
- Player can search for a particular user name to see his/hers score.

### **Alternative flows of events:**

- Player may want to return to main menu.
  - Player clicks on back button on screen.
  - System renders main menu screen.

## **7) Use Case Name: Watch Tutorial**

**Participating Actor:** Player

**Stakeholders and Interests:** Player may want to watch the tutorial of the iq puzzler game.

### **Pre-condition:**

- Player has to be in main menu.
- Player must pause the game.

**Post-condition:** -

**Entry condition:**

- Player clicks on “Watch Tutorial” button in the main menu.
- Player clicks on “Watch Tutorial” button in the pause menu

**Exit condition:** Player clicks on back button in watch tutorial screen.

**Main flows of events:**

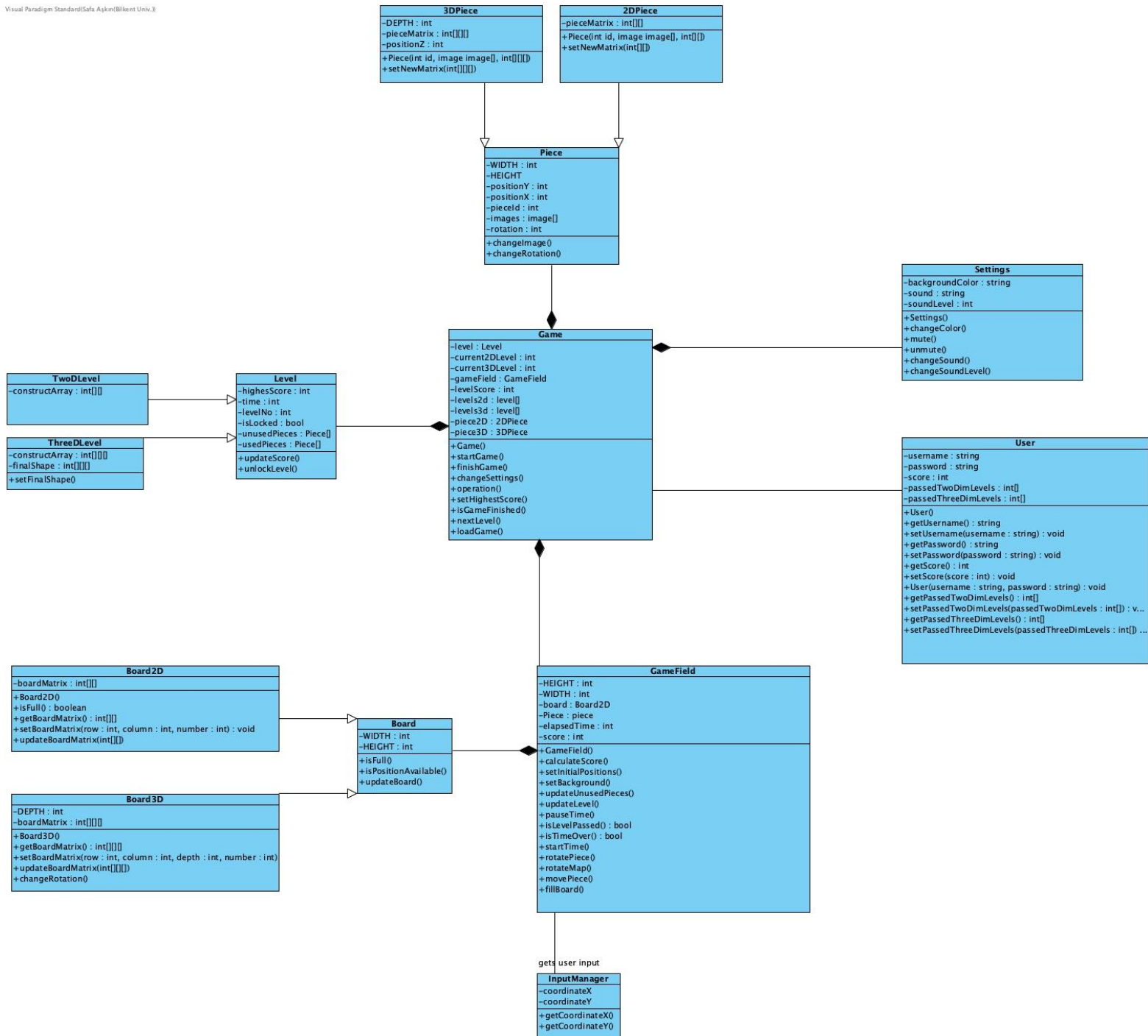
- Player watches the video tutorial and observes the gameplay.

**Alternative flows of events:**

- Player may want to return to main menu.
  - Player clicks on back button on screen.
  - System renders main menu screen.
  - System renders pause screen.

## 5.2 Object and Class Diagram

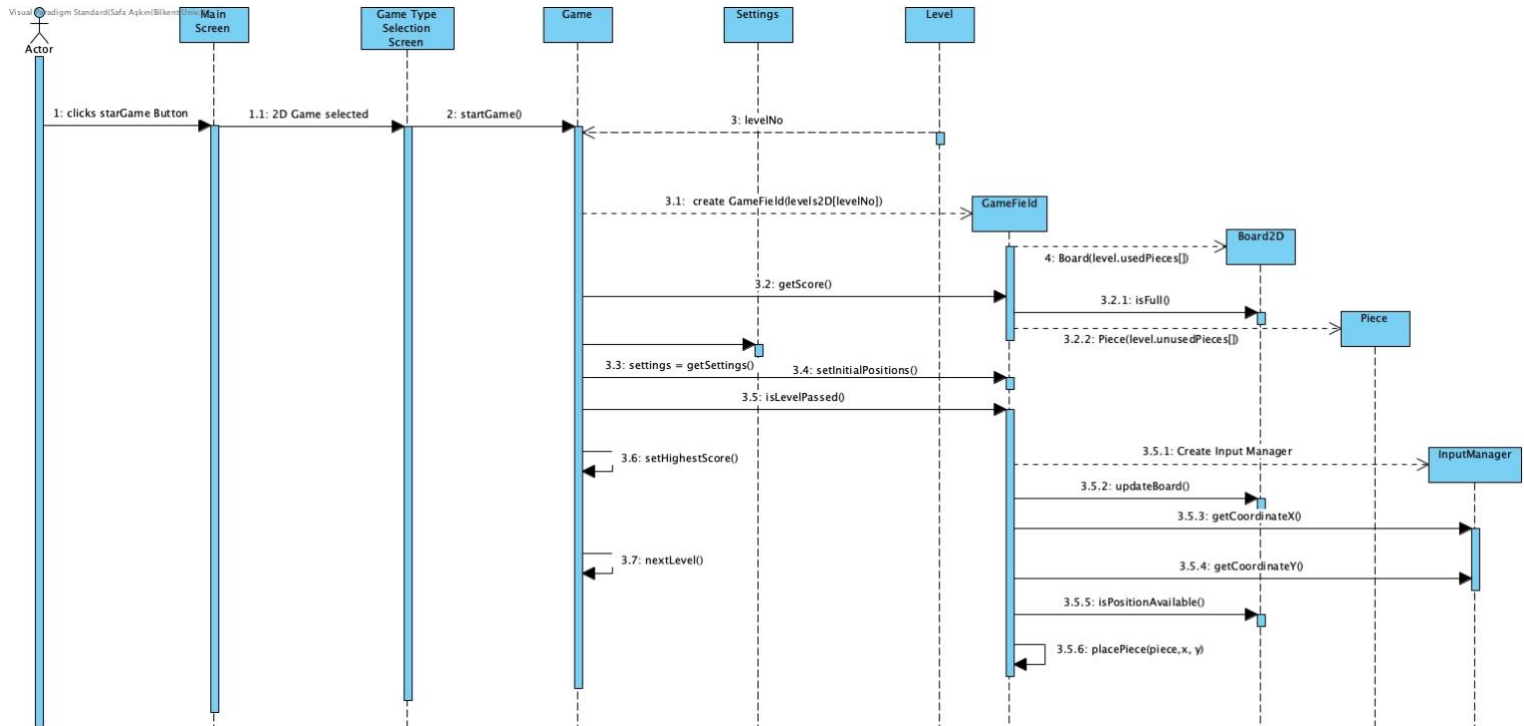
Visual Paradigm Standard(Safa Askin@Bilkent Univ.)





## 5.3 Sequence Diagrams

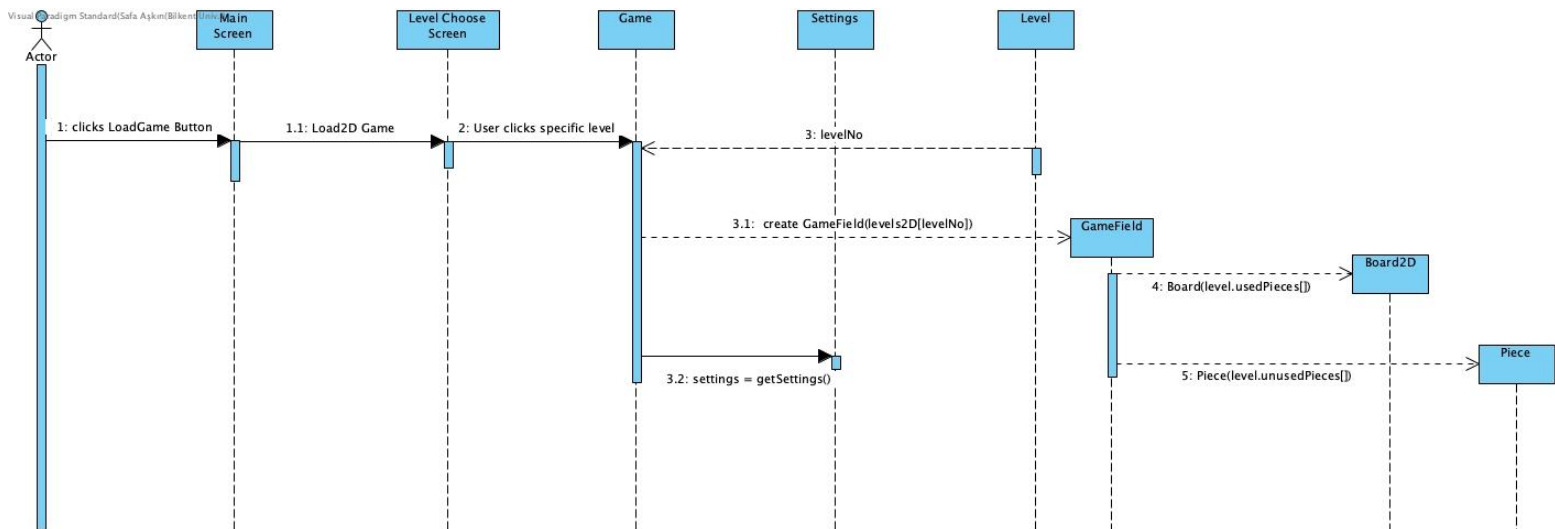
### 5.3.1 Play Game



#### Scenario: User plays the game

User wants to play the game. He enters the Menu and pushes the Start Game button. The Game class initializes all distinct Levels and Pieces. It gets the initial settings from Settings. After that it creates the GameField with proper Level. GameField initializes the game arena by using Level instance's usedPieces and unusedPieces arrays. After placing all Pieces and Board into proper position at scene the game starts. User takes pieces by clicking them and puts them to Board. Game field takes the x and y coordinates and decides which unused Piece is selected. GameField checks whether the game finished or not and it turns the score back to Game when games finishes.

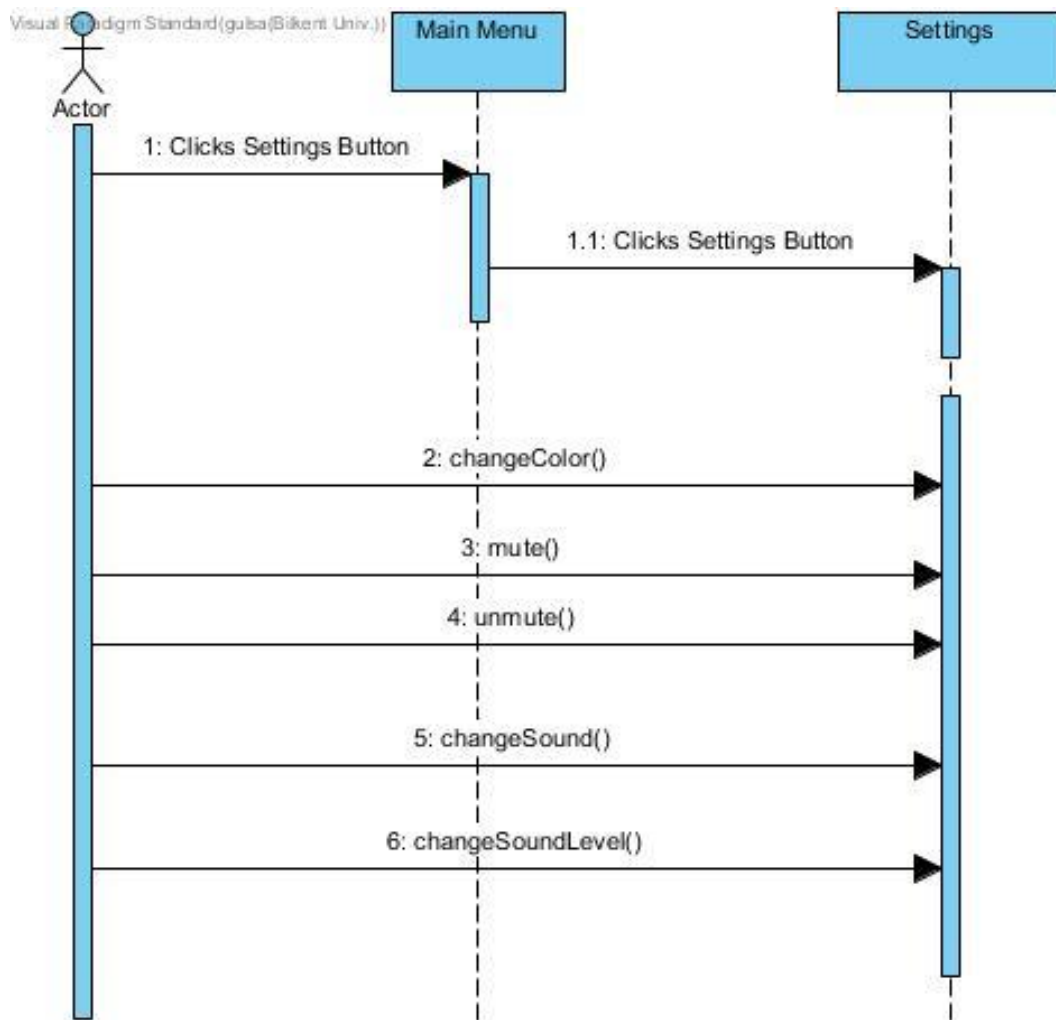
## 5.3.2 Load Game



**Scenario:** Users Loads a game previously played

When user wants to load a saved game, he/she presses the load button in main menu. When he/she presses the load game menu, system directs user to game mode selection menu. In this sub menu, system asks user whether he/she wants to load a 3d game or 2d game. According to his/her preference the level selection screen opens for given game mode and user picks the level that he/she wants to continue. After that, according to given level, Game class creates GameField object to load all views of current level. After game is loaded user can start to play game.

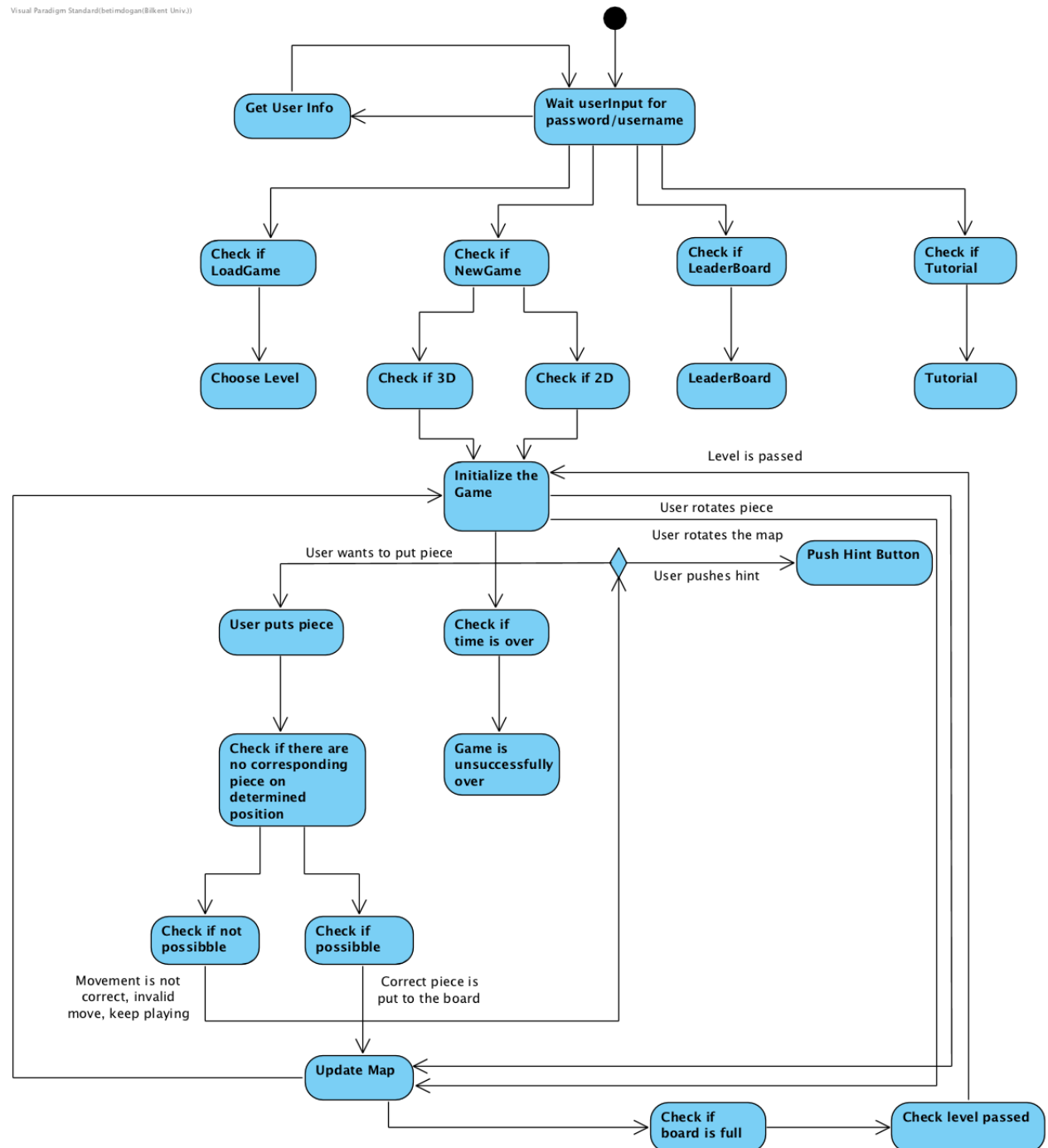
### 5.3.3 Settings



**Scenario:** User wants to change the background and sound  
User wants to change the background color, sound level or sound itself.  
Settings class manages this alterations and provides a better experience to users.

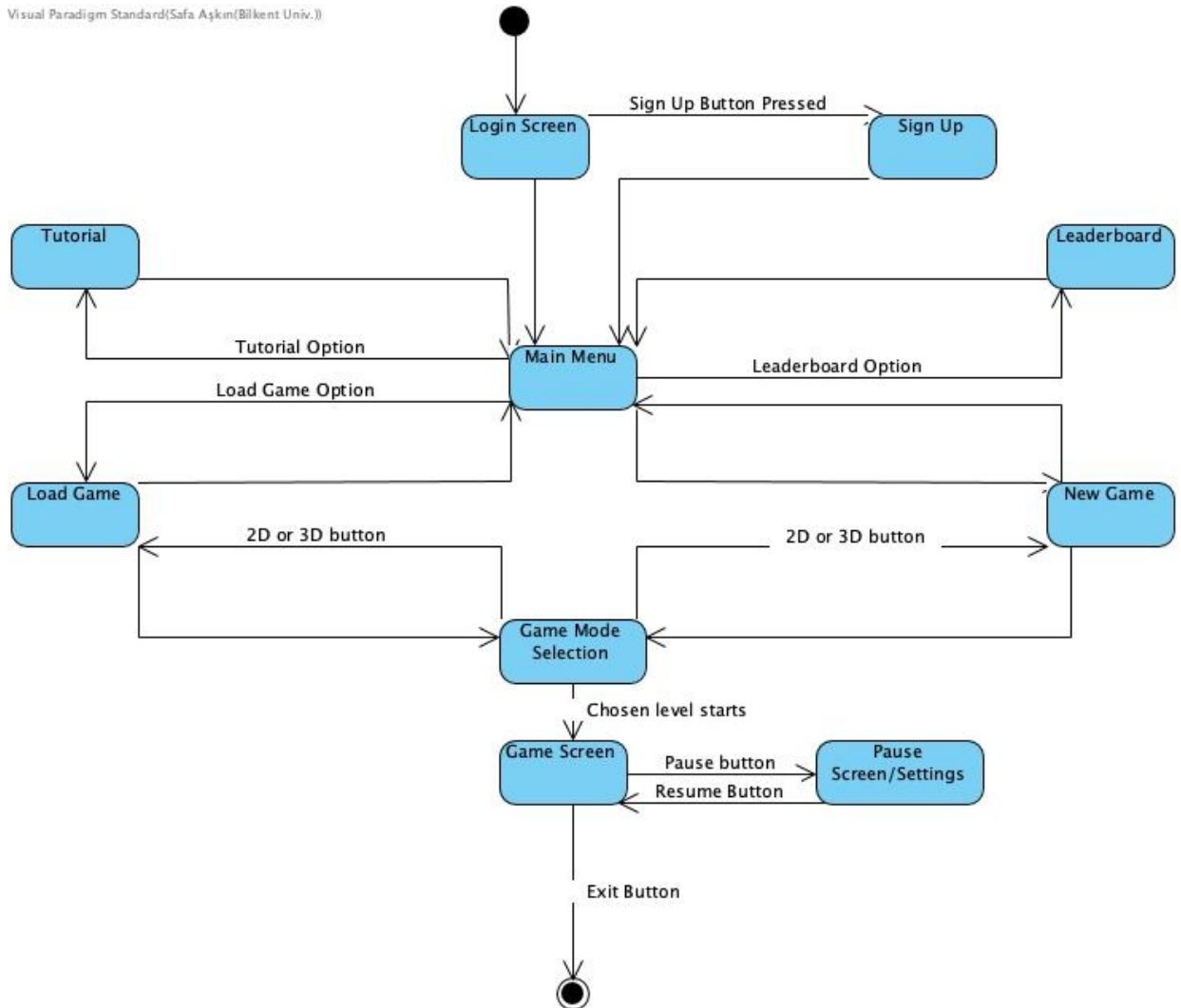
## 5.4 Dynamic Diagrams

### 5.4.1 Activity Diagram



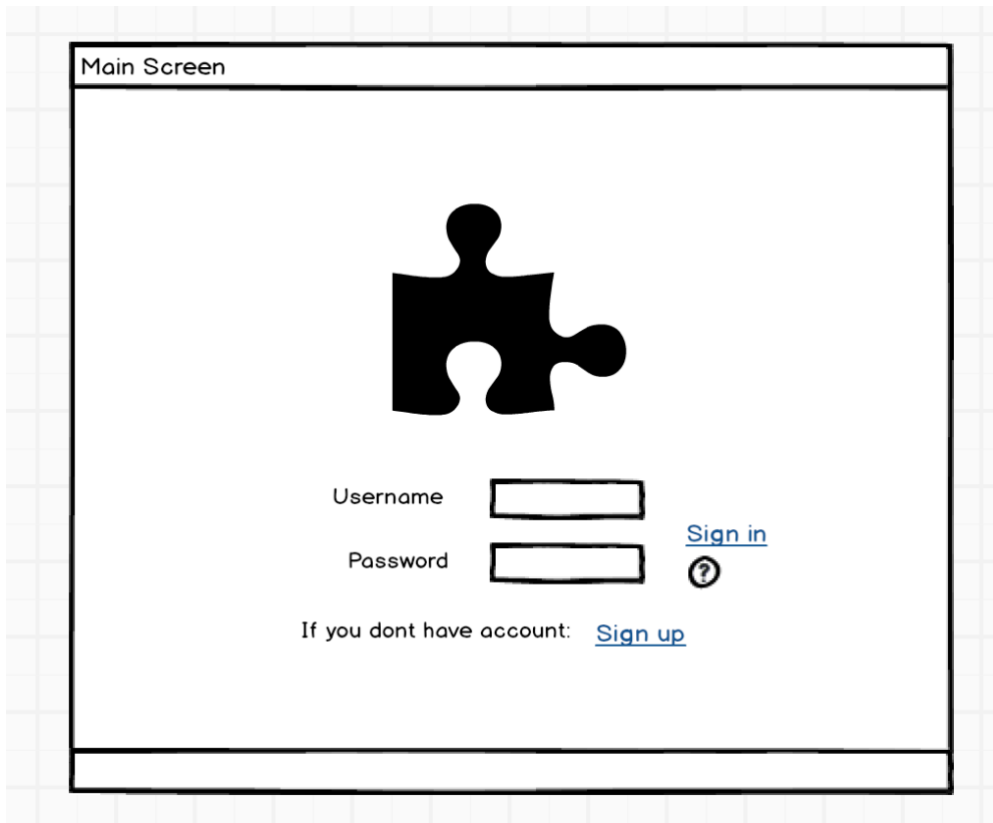
## 5.4.2 State Chart Diagram

Visual Paradigm Standard(Safa Aşkın(Bilkent Univ.))

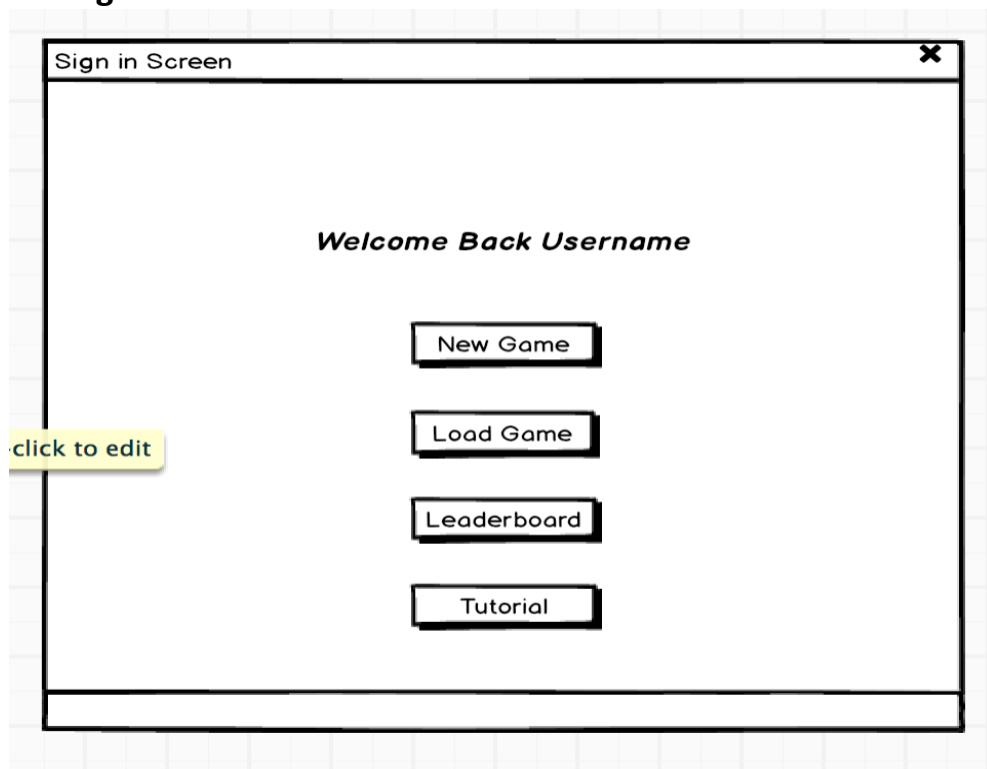


## 5.5. Game Mockup

### 5.5.1 Main Screen




### 5.5.2 Sign In Screen



### 5.5.3 Sign Up Screen

Sign Up Screen



Create an username

Create a password

Create a password

Sign Up

### 5.5.4 Game Mode Selection Screen

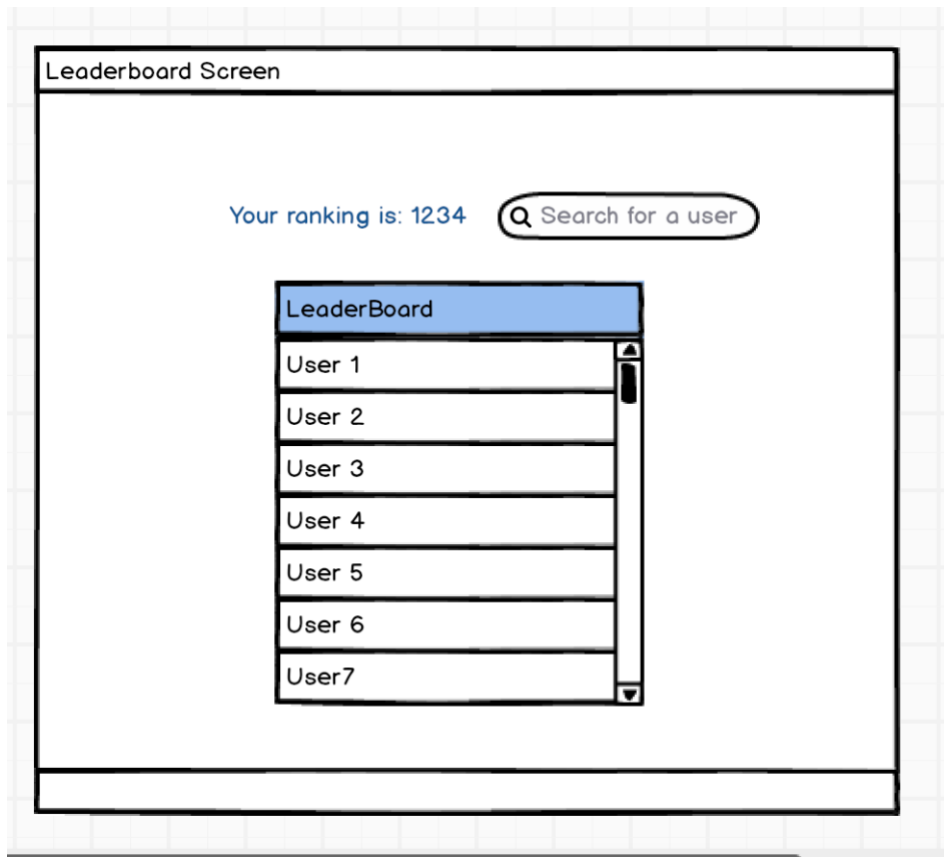
New Game Screen

Choose game mode

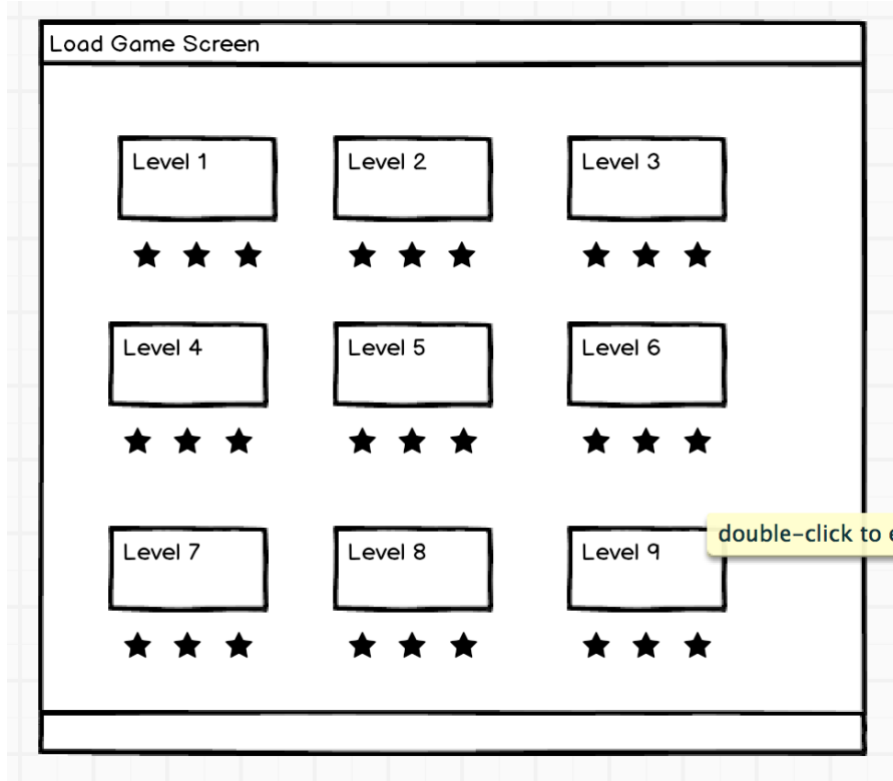
☐ 2D

☐ 3D

### 5.5.5 Leaderboard Screen

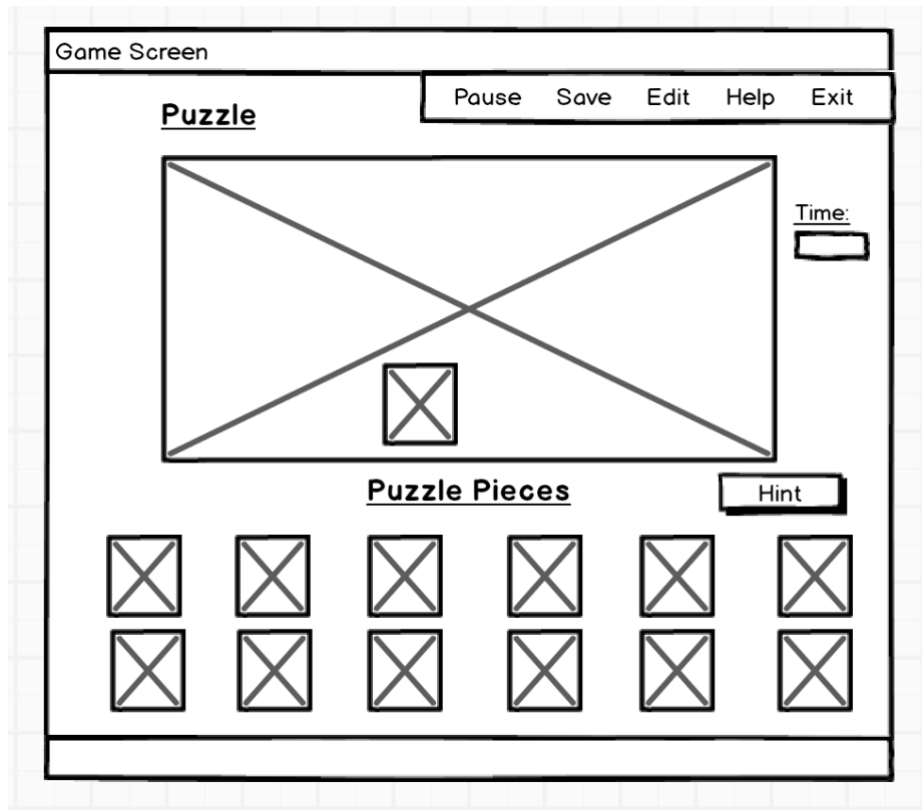


### 5.5.6 Load Game Screen



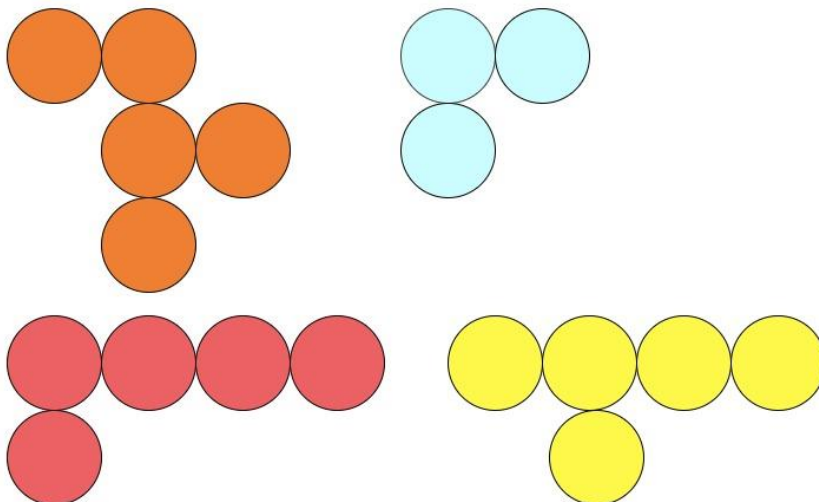


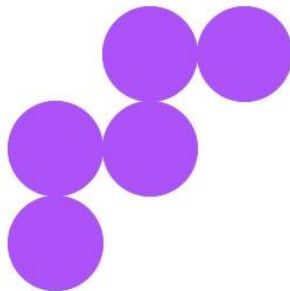
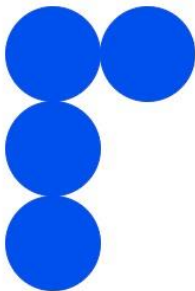
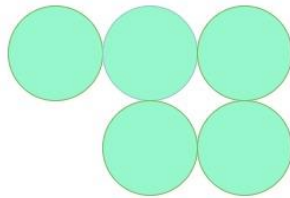
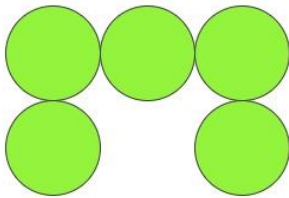
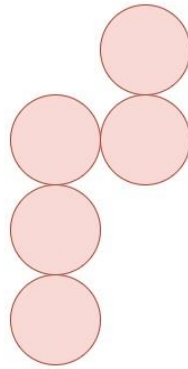
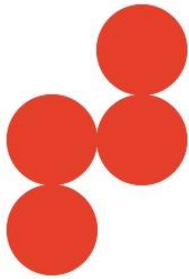
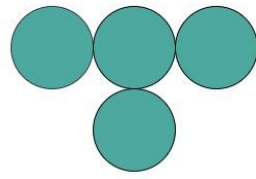
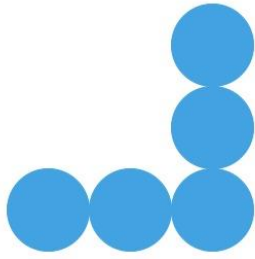
### 5.5.7 Game Screen



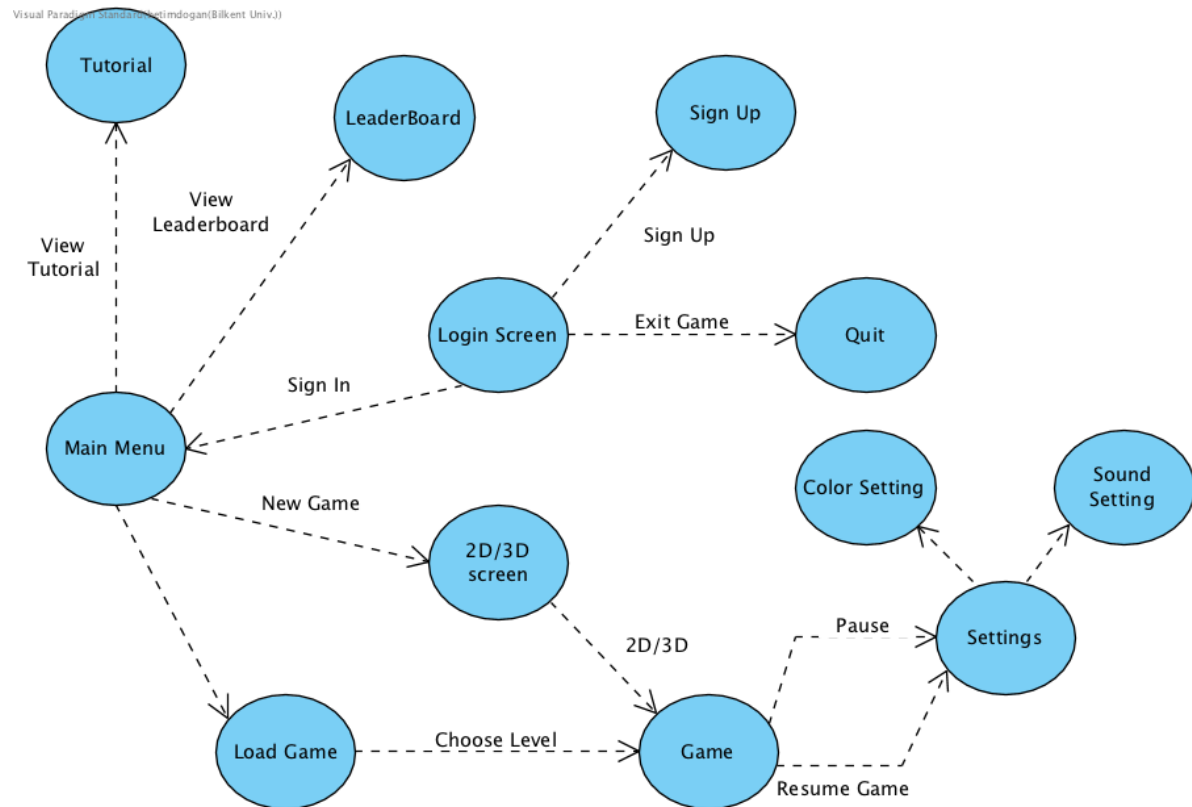
### Icons

Pieces:





## 5.6 Navigational Path



## 6. Conclusion

In this report we analyzed the system to develop to implement a fully functioning, well designed game. We explained the necessary functionalities, requirements and system models of the game. We tried to explain functionalities player may perform in “Functional Requirements” section (Section 3.1); we tried to explain additional features offered to players in “Non-Functional Requirements” section (Section 3.2). We designed system models to express the design of our game from different perspectives. Also we designed screen mockups and attached icons to show the interface of the game. We tried to make our class and object models as clear as possible so that we can implement the game efficiently and well-organized.

## 7. Glossary & References

1. <https://www.smartgames.eu/uk/one-player-games/iq-puzzler-pro>