Bilkent University

CS 319 Term Project

Analysis Report

Section 3

Team Name: JOKERS Team

Project Name: IQ Puzzler Pro

Project Group Members:

- Burak Erdem Varol
- Ravan Aliyev
- Subhan Ibrahimli
- Ismayil Mammadov
- Mahmud Sami Aydin
- Cihan Erkan

Table of Contents

1. INTRODUCTİON 4

2. OVERVİEW 5

2.1 GAMEPLAY 5
2.2 BOARD 5
2.3 GAME MODES 5
2.4 CUSTOM LEVEL CREATOR 6
2.5 2-PLAYER MODE 6
2.6 SETTİNGS 6

3. FUNCTİONAL REQUİREMENTS 7

3.1 PLAY 7
3.1.1 Single Player 7
3.1.1.1 2D mode 7
3.1.1.1.1 Normal mode 7
3.1.1.1.2 Time mode 8
3.1.1.2 3D mode 8
3.1.1.2.1 Normal mode 8
3.1.1.2.2 Time mode 8
3.1.2 Multi (Two) Player 8
3.1.3 Level 9
3.2 CREATE LEVEL 9
3.3 HIGHSCORES 9
3.4 HOW TO PLAY 9
3.5 SETTİNGS 10
3.6 CREDİTS 10
3.7 QUİT 10

4. NON-FUNCTİONAL REQUİREMENTS 11

4.1 USABİLİTY 11
4.2 CAPACİTY 11
4.3 GAME PERFORMANCE 11
4.4 EXTENDİBİLİTY 11
4.4.1 Player Mode 11
4.4.2 Game Finish part 12
4.4.3 Bonuses 12

5. SYSTEM MODELS 13

5.1 USE CASE DIAGRAM 13
5.1.1 Play 1-Player 2D Time Mode 13
5.1.2 Play 1-Player 2D Normal Mode 15
5.1.3 Play 1-Player 3D Time Mode 17
5.1.4 Play 1-Player 3D Normal Mode 19
5.1.5 Play 2-Player Mode 21
5.1.6 Play Choose Levels 23
5.1.7 High Scores/Achievements High Scores 25
5.1.8 Credits 26
5.1.9 HowToPlay 27
5.1.10 Settings 28
5.1.11 Quit 30
5.1.12 High Score/Achievements Achievements 31

5.1.13 Create Level *32*
5.2 DYNAMİC MODELS 34
5.2.1 Sequence Diagrams *34*
5.2.1.1 Start Game 34
5.2.1.2 Moving Single Piece 35
5.2.1.3 Settings 36
5.2.1.4 Custom Level 37
5.2.2 Activity Diagram *38*
5.3 OBJECT AND CLASS MODEL 40
5.3.1 Pieces Package *41*
5.3.1.1 Node Class 41
5.3.1.2 Polymino Class 41
5.3.1.3 FixedPolyminos Class 42
5.3.1.4 LocatablePolymino Class 42
5.3.1.4.1 Mutable Interface 42
5.3.1.4.2 Locatable Interface 42
5.3.1.5 PolyminoSet Class 42
5.3.2 Game Package *42*
5.3.2.1 Map Class 43
5.3.2.2 Board Class 43
5.3.2.3 Player Class 44
5.3.2.4 GameEngine Class 44
5.3.3 LevelUtilities Package *44*
5.3.3.1 Level Class 45
5.3.3.2 LevelLoader Class 45
5.3.3.3 LevelCreator Class 45
5.3.3.4 FillingCreator Class 45
5.3.3.5 PieceCreator Class 45
5.3.4 UserInterface Package *45*
5.3.4.1 MainMenu Class 46
5.3.4.2 GameMenu Class 46
5.3.4.3 SettingMenu Class 46
5.3.4.4 PauseMenu Class 47
5.3.5 SettingAndUtilities Package *47*
5.3.5.1 SettingManager Class 47
5.3.5.2 AchievementAndHighScoreShower Class 47
5.3.5.3 AchievementAndHighScoreManager Class 47
5.3.5.4 Languages Class 48
5.3.5.5 UserPreferences Class 48
5.4 SCREEN MOCKUPS 48
5.4.1 Main Menu *48*
5.4.2 Play Button *49*
5.4.3 Single Player *50*
5.4.4 Levels *51*
5.4.5 Settings *52*
5.4.6 Achievements/High Scores *53*
5.4.7 Game Interface *54*
5.4.8 Create Level Screen *55*
5.4.9 How To Play Screen *56*
5.4.10 Pause Screen *57*
5.4.11 Credits Screen *58*

6. CONCLUSİON 59

7. REFERENCES 60

1. Introduction

Implementing basic puzzle game IQ Puzzler Pro assigned to us as CS 319 term project. First of all, we purchased the physical version of the game and have played it together in order to get an idea on how we can implement and the game and to improve that even further. After playing and examining the game we realized that this game is very suitable for an object oriented design because it is possible to divide the game into many objects such as pieces, board, and spots that pieces fit.

We also discussed how we can improve the game and what should be our motivation to improve this game. We decided that the main issues of the game are feeling of repetition and danger for children caused by small pieces. We believe that implementing the game digitally will remove the danger for children and adding new features such as creating custom levels will decrease the sense of repetitiveness.

Our main aim for the project is to exercise software engineering principles that we learn in our class by approaching the game as if it was a real life project. Besides that, we know that, we are going to put a lot of time and effort for this project. Therefore, we are also aiming to produce a game that is worthy of our efforts and satisfies us at the end of semester.

2. Overview

2.1 Gameplay

IQ Puzzler Pro is a both 2 and 3-dimensional puzzle game. The game is originally designed to play on its board but we are going to create the game in a digital environment. There are two main elements of the game: Game board and pieces. Game board is consisting of spots that pieces can fit in. There are also 12 different pieces with different shapes. Aim of the player is placing those pieces in game board without leaving a single spot empty for 2-dimensional case. For 3-dimensional case aim of the player is building a pyramid. Moreover, user will be able to rotate and flip those pieces by clicking on them in order to place those pieces to board. In the beginning the game places some of the pieces to board and ask user to fill rest of the board.

2.2 Board

There are 3 different boards that the game can be played. 2 of them are 2 dimensional and one of them is 3 dimensional. General mechanics of both 2-dimensional boards are similar, only difference is the relative place of spots. However, 3-dimensional case is quite different. User uses a 5x5 square to build the pyramid.

2.3 Game Modes

There will be 2 game modes. Story mode and Custom Game mode. For story mode, user can play one of the classical challenges of IQ Puzzler Pro. For Custom Game mode, players can play one of their own custom designed levels or they can request the game to create a new game by using the original pieces of IQ Puzzler Pro.

2.4 Custom Level Creator

Users can create new levels and pieces with any of the game modes (2-D and 3-D). To do that user will be provided an empty board of their choice, and he/she will be able to fill it with his/her custom-made pieces. Those pieces could have any size and shape and it will be completely up to user. However, game will treat those piece as classic pieces, so during the gameplay, user can rotate and place those pieces to board.

2.5 2-Player Mode

In 2 player mode players can compete against each other. In this mode players will place piece one by one. If one of the players tries to place a piece to an incorrect spot, the game will not allow user to place that piece to that spot and other player's turn will begin. After the game is concluded, points will be calculated depending on the time they spent and number of correct moves they do. After computation of points winner of the game will be declared.

2.6 Settings

In settings page, user can adjust the volume of the game and they can change the language of the game.

3. Functional Requirements

3.1 Play

The player may access this screen from the Main Manu. This screen will consist Levels and two game modes: Single Player and Two Player.

3.1.1 Single Player

This screen will be accessible from Play screen. There will be "2D or 3D" and "Normal or Against Time" choices to choose the game mode to play.

3.1.1.1 2D mode

Player can play the game as two dimensional by selecting this mode. The player will play the game by placing the puzzle pieces to fill the empty places. Player can select the wanted piece by clicking the left button of the mouse. Selected piece can be rotated and flipped by the help of the appropriate keys of keyboard. After rotating and flipping the piece player can drag and drop that piece to fill the game board. Since the some of the pieces will be fixed on the board at the beginning of the level, there will be only one solution for that level. There will be two modes: Normal and Time.

3.1.1.1.1 Normal mode

In Normal mode there will not be any time limit and score will be calculated according to the time needed to pass the level. At the end of the game user can enter name in order to display his/her score in high scores list.

3.1.1.1.2 Time mode

In Time mode, time will be limited to make the game more challenging. If player fails to pass the level at a given amount of time will have to start the game from first level.

3.1.1.2 3D mode

In this mode game will be 3 dimensional and only a specific part of the board will be used. Goal of this mode is making pyramid by filling empty places using the puzzle pieces. Just like Two-Dimensional mode, this mode has two modes too. These modes are Normal and Time.

3.1.1.2.1 Normal mode

This mode will not have any time limit.

3.1.1.2.2 Time mode

Time limit of this mode makes it more challenging to finish the game.

3.1.2 Multi (Two) Player

By entering this screen, the players can play the game in two player mode. In this mode players play the game against each other by turns. Each player has a limited time to place a puzzle piece on empty places. If the placement/guess is not correct that player does not get any point and the next player gets turn. If the placement is done correctly then that player gets more times for next puzzle pieces and gets point. At the end player with higher score becomes the winner.

3.1.3 Level

In this screen user will be able to choose any level to play unlike the single player mode. In the single player mode player has to pass the level to unlock the next one, but in this mode all levels are available to play. All levels of 2D and 3D modes and new created levels by user in Create Level mode will be displayed under according title (2D levels,3D levels and New Created Levels).

3.2 Create Level

In this screen user will be able to create a new level and pieces. User has to select one of three modes to create a new level. These modes are Flat, Diagonal and 3D. Flat and Diagonal modes are parts of 2D. After selecting wanted mode, chosen board type will be shown accordingly. Then user can choose colors and fill the empty areas, to create the pieces. Then user checks empty areas, if it is fully filled then he/she can choose which places will be displayed as filled to the player.

3.3 HighScores

The user can see ten highest scores by entering to this screen. Names and time needed to complete the game of each player will be displayed here. Players completing the levels in a shorter amount of a time will be ranked higher. Also, achievements will be displayed here too. User can see achievements like “Balanced time using” and “by five bonus points”

3.4 How to Play

This screen is accessible from Main Menu. In this screen user can get information about rules, modes and explanations how to play the game.

3.5 Settings

Users can access to this screen from Main Menu. In this screen users can

- adjust the volume
- mute and unmute
- change the language.

3.6 Credits

This screen is accessible to the users from Main Menu. By the help of this screen users can get information about the identity of the creators of this game and their contact details.

3.7 Quit

Users can exit the game by selecting this function.

4. Non-functional Requirements

4.1 Usability

In contrast to the real IQ Puzzle game, parents and their children can play this game without any fear. Because IQ puzzle has little pieces and little children can swallow it which can lead to deaths, physical version of this game a more usable and safety. Apart from that, you should pay money for virtual one. But this PC version of th game will be free for everyone.

4.2 Capacity

IQ Puzzle game is useful for development of children and also one of the best occupations for leisure time. In order to make this game available for all social classes this game will take less place in PCs.

4.3 Game Performance

To attract people attention, we are planning different melodies for menu, and player mode. For instance, when player win or lose the game there will be melodies according to these cases. In menu there will be melody which will not irritate user in contrast to some other games.

4.4 Extendibility

4.4.1 Player Mode

In player mode there will be Single. Multi (2) players and Levels. In single player mode, there will be 4 choices: 2D or 3D and “Against time” or “Normal”. In Multiplayer part, 2 players can play the game same time by choosing 2D or 3D options. Levels option will be paid. In this level, player can level all levels without passing them. Apart from that, there will be Create part. In this part, user can make his/her own level.

4.4.2 Game Finish part

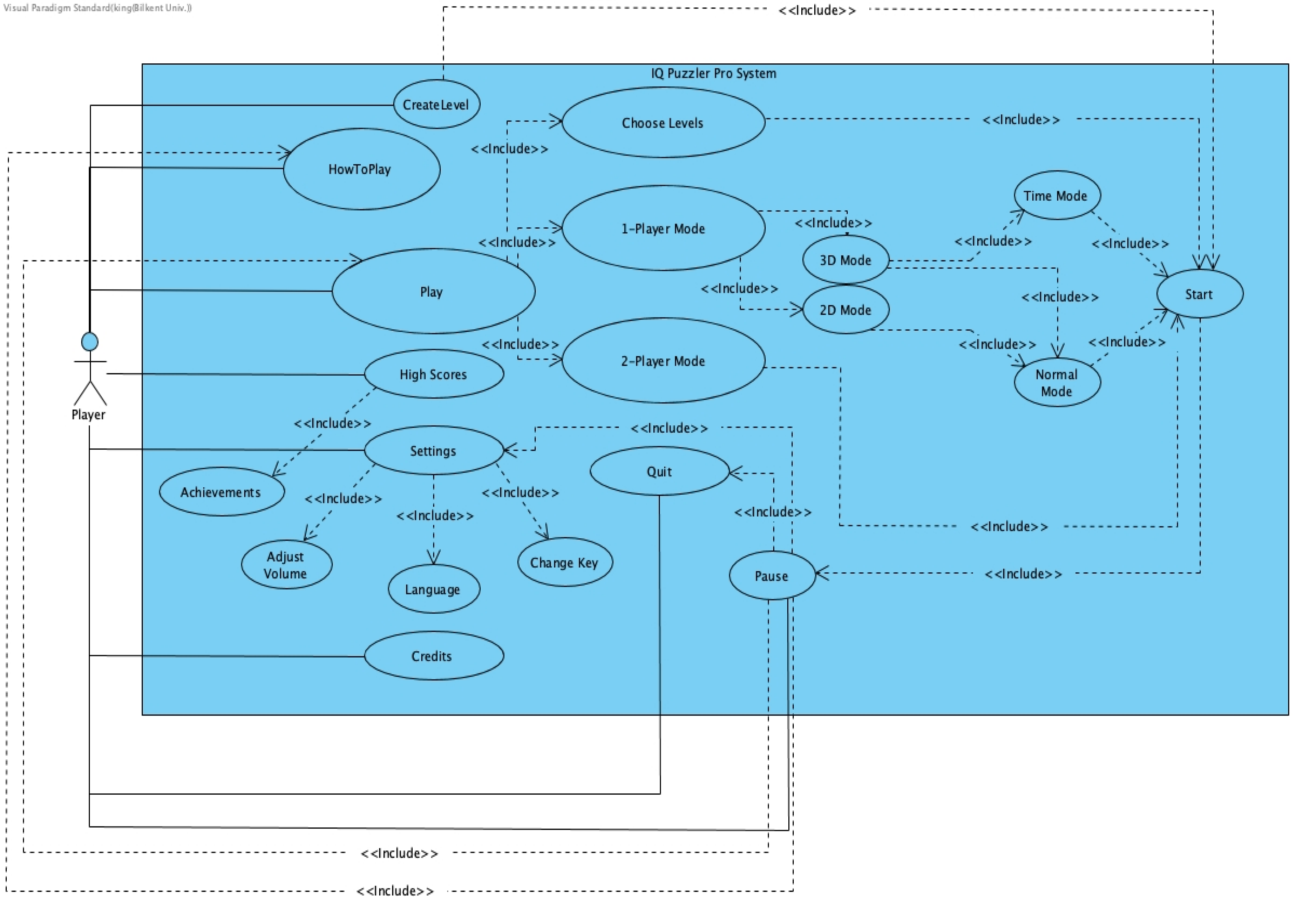
In terms of stimulation real IQ puzzle game in PC, the game send lose/win message at the end of game. User can change place of any part which putted in board. At the end, he should click “Check” button. Game will check the board and if every place is filled, it sends win message.

4.4.3 Bonuses

There will be bonuses at the end of every win. If player cannot find right place for the figure, he can use these bonuses for this difficulty.

5. System Models

5.1 Use Case Diagram



5.1.1 Play 1-Player 2D Time Mode

Use Case: Play 1-Player 2D Time Mode

Primary Actor: Player

Stakeholders and Interests:

- Player selects play
- Player selects 1-Player Mode
- Player selects 2D Mode
- Player selects Time Mode

- System creates the game table
- System starts the game.

Pre-conditions:

- Player must be in the main menu.

Post-conditions:

Entry-conditions:

- Player should choose the 1-Player Mode
- Player should choose the 2D Mode
- Player should choose the Time Mode

Exit conditions:

- Player selects “Quit” button from the Pause Menu.
- Player successfully finished the game by completing level in a given amount of time.

Success Scenario Event Flow:

1. Player chooses “1-Player Mode”.
2. Player chooses "2D Mode".
3. Player chooses "Time Mode".
4. System starts game and timer.
5. Player selects the piece.
6. Player rotates or flips the piece.
7. Player drags the piece on the game board.
8. Player drops the piece on the board.
9. System decides the piece location correct or incorrect.

10. System updates the view of game board.
11. Player tries to fill all spaces on the game board.
12. Player fills the game board with given pieces in a given amount of time.
13. Player wins the game.
14. System finishes the level and initializes the next level.

Alternative Event Flows:

1. If Player wants to return main menu:
    a. Player pauses the game by pressing Pause Button from game screen.
    b. System displays Pause Menu
    c. Player selects "Quit"
    d. System asks confirmation
    e. Player selects "Yes".
    f. System exits the game.
    g. System show the Main Menu.

5.1.2 Play 1-Player 2D Normal Mode

Use Case: Play 1-Player 2D Normal Mode

Primary Actor: Player

Stakeholders and Interests:

- Player selects play
- Player selects 1-Player Mode
- Player selects 2D Mode

- Player selects Normal Mode
- System creates the game table
- System starts the game.

Pre-conditions:

- Player must be in the main menu.

Post-conditions:

Entry-conditions:

- Player should choose the 1-Player Mode
- Player should choose the 2D Mode
- Player should choose the Normal Mode

Exit conditions:

- Player selects “Quit” button from the Pause Menu.
- Player successfully finished the game by completing level.

Success Scenario Event Flow:

1. Player chooses “1-Player Mode”.
2. Player chooses "2D Mode".
3. Player chooses "Normal Mode".
4. System starts game
5. Player selects the piece.

6. Player rotates or flips the piece.
7. Player drags the piece on the game board.
8. Player drops the piece on the board.
9. System decides the piece location correct or incorrect.
10. System updates the view of game board.
11. Player tries to fill all spaces on the game board.
12. Player fills the game board with given pieces.
13. Player wins the game.
14. System finishes the level and initializes the next level.

Alternative Event Flows:

1. If Player wants to return main menu:

    a) Player pauses the game by pressing Pause Button from game screen.

    b) System displays Pause Menu

    c) Player selects "Quit"

    d) System asks confirmation

    e) Player selects "Yes".

    f) System exits the game.

    g) System show the Main Menu.

5.1.3 Play 1-Player 3D Time Mode

Use Case: Play 1-Player 3D Time Mode

Primary Actor: Player

Stakeholders and Interests:

- Player selects play
- Player selects 1-Player Mode
- Player selects 3D Mode
- Player selects Time Mode
- System creates the game table
- System starts the game.

Pre-conditions:

- Player must be in the main menu.

Post-conditions:

Entry-conditions:

- Player should choose the 1-Player Mode
- Player should choose the 3D Mode
- Player should choose the Time Mode

Exit conditions:

- Player selects “Quit” button from the Pause Menu.
- Player successfully finished the game by completing level.

Success Scenario Event Flow:

1. Player chooses “1-Player Mode”.
2. Player chooses "3D Mode".
3. Player chooses "Time Mode".
4. System starts game
5. Player selects the piece.

6. Player rotates or flips the piece.
7. Player drags the piece on the game board.
8. Player drops the piece on the board.
9. System decides the piece location correct or incorrect.
10. System updates the view of game board.
11. Player fills the given spaces and has to make a pyramid on the game board in a given amount of time.
12. Player wins the game.
13. System finishes the level and initializes the next level.

Alternative Event Flows:

1. If Player wants to return main menu:
    a. Player pauses the game by pressing Pause Button from game screen.
    b. System displays Pause Menu
    c. Player selects "Quit"
    d. System asks confirmation
    e. Player selects "Yes".
    f. System exits the game.
    g. System show the Main Menu.

5.1.4 Play 1-Player 3D Normal Mode

Use Case: Play 1-Player 3D Normal Mode

Primary Actor: Player

Stakeholders and Interests:

- Player selects play

- Player selects 1-Player Mode
- Player selects 3D Mode
- Player selects Normal Mode
- System creates the game table
- System starts the game.

Pre-conditions:

- Player must be in the main menu.

Post-conditions:

Entry-conditions:

- Player should choose the 1-Player Mode
- Player should choose the 3D Mode
- Player should choose the Normal Mode

Exit conditions:

- Player selects “Quit” button from the Pause Menu.
- Player successfully finished the game by completing level.

Success Scenario Event Flow:

1. Player chooses “1-Player Mode”.
2. Player chooses "3D Mode".
3. Player chooses "Normal Mode".
4. System starts game
5. Player selects the piece.
6. Player rotates or flips the piece.
7. Player drags the piece on the game board.

8. Player drops the piece on the board.

9. System decides the piece location correct or incorrect.

10. System updates the view of game board.

11. Player fills the given spaces and has to make a pyramid on the game board.

12. Player wins the game.

13. System finishes the level and initializes the next level.

Alternative Event Flows:

1. If Player wants to return main menu:

   a. Player pauses the game by pressing Pause Button from game screen.

   b. System displays Pause Menu

   c. Player selects "Quit"

   d. System asks confirmation

   e. Player selects "Yes".

   f. System exits the game.

   g. System show the Main Menu.

5.1.5 Play 2-Player Mode

Use Case: Play 2-Player Mode

Primary Actor: Player

Stakeholders and Interests:

- Player selects play

- Player selects 2-Player Mode

- System creates the game table

- System starts the game.

Pre-conditions:

- Player must be in the main menu.

Post-conditions:

Entry-conditions:

- Player should choose the 2-Player Mode

Exit conditions:

- Player selects “Quit” button from the Pause Menu.
- One Player successfully finished the game by completing level.

Success Scenario Event Flow:

1. Player chooses “2-Player Mode”.
2. System starts game
3. Player1 selects the piece.
4. Player1 rotates or flips the piece.
5. Player1 drags the piece on the game board.
6. Player1 drops the piece on the board.
7. System decides the piece location correct or incorrect.
8. System updates the view of game board.
9. If Player1 decision corrects in a given amount of time, Player1 will continue.
10. System updates point of Player1.
11. System updates a given amount of time.
12. If Player1 decision incorrects and time exceeded, Player2 will continue.
13. System updates a given amount of time.

14. If game board fills by Player1 or Player2, system decides which player wins the game according to players points.
15. System finished the game.
16. System asks to user rematch or back to the main menu.

Alternative Event Flows:

1. If Player wants to return main menu:
    a. Player pauses the game by pressing Pause Button from game screen.
    b. System displays Pause Menu
    c. Player selects "Quit"
    d. System asks confirmation
    e. Player selects "Yes".
    f. System exits the game.
    g. System show the Main Menu.

5.1.6 Play Choose Levels

Use Case: Play Choose Levels

Primary Actor: Player

Stakeholders and Interests:

- Player selects play
- Player selects Choose Levels
- Player selects a level
- System creates the game table
- System starts the game.

Pre-conditions:

- Player must be in the main menu.

Post-conditions:

Entry-conditions:

- Player should choose the Choose Levels

Exit conditions:

- Player selects “Quit” button from the Pause Menu.

Success Scenario Event Flow:

1. Player chooses “Choose Levels”.
2. System starts game
3. Player selects the piece.
4. Player rotates or flips the piece.
5. Player drags the piece on the game board.
6. Player drops the piece on the board.
7. System decides the piece location correct or incorrect.
8. System updates the view of game board.
9. Player fills the game board with given pieces.
10. System finished the game.
11. System goes back to the main menu.

Alternative Event Flows:

1. If Player wants to return main menu:
    a. Player pauses the game by pressing Pause Button from game screen.
    b. System displays Pause Menu
    c. Player selects "Quit"

d. System asks confirmation

e. Player selects "Yes".

f. System exits the game.

g. System show the Main Menu.

5.1.7 High Scores/Achievements High Scores

Use Case: High Scores/Achievements High Scores

Primary Actor: Player

Stakeholders and Interests:

- Player selects High Scores/Achievements
- Player selects High Scores.
- System creates the high score table
- System shows the high score table.

Pre-conditions:

- Player must be in the main menu.

Post-conditions:

Entry-conditions:

- Player should choose the High Score/Achievements

Exit conditions:

- Player selects “Back” button from the High Score/Achievements Screen.

Success Scenario Event Flow:

1. Player chooses “High Scores/Achievements”.
2. System creates the high scores table from finished games history.
3. System shows the high scores table.

4. Player clicks the back button on the High Scores/Achievements Screen.

5. System goes back to the main menu.

Alternative Event Flows:

- No Alternative flows

5.1.8 Credits

Use Case: Credits

Primary Actor: Player

Stakeholders and Interests:

- Player selects Credits

- System shows the Credits.

Pre-conditions:

- Player must be in the main menu.

Post-conditions:

Entry-conditions:

- Player should choose the Credits.

Exit conditions:

- Player selects “Back” button from the Credits screen.

Success Scenario Event Flow:

1. Player chooses “Credits”.

2. System creates the Credits.

3. System shows the Credits.

4. Player clicks the back button on the Credits screen.

5. System goes back to the main menu.

Alternative Event Flows:

1. If Player wants to return to main menu:
    a. Player selects "Back"
    b. System exits Credits screen.
    c. System show the Main Menu.

5.1.9 HowToPlay

Use Case: HowToPlay

Primary Actor: Player

Stakeholders and Interests:

- Player selects HowToPlay
- System shows the HowToPlay.

Pre-conditions:

- Player must be in the main menu or clicks to Pause.

Post-conditions:

Entry-conditions:

- Player should choose the HowToPlay.

Exit conditions:

- Player selects “Back” button from the HowToPlay screen.

Success Scenario Event Flow:

1. Player chooses “HowToPlay”.
2. System shows the HowToPlay.
3. Player clicks the back button on the HowToPlay screen.
4. System goes back to the main menu.

Alternative Event Flows:

1. If Player wants to return main menu:
    a. Player selects "Back"
    b. System exits Credits screen.
    c. System show the Main Menu.
2. If Player wants to return a game:
    a. Player selects "Back"
    b. System exits HowToPlay screen.
    c. Player returns Pause Menu.
    d. Player selects "Back"
    e. System exits Pause Menu.
    f. Player returns a game.

5.1.10 Settings

Use Case: Settings

Primary Actor: Player

Stakeholders and Interests:

- Player selects Settings
- System shows the Settings.

Pre-conditions:

- Player must be in the Main menu or Pause menu.

Post-conditions:

Entry-conditions:

- Player should choose the Settings.

Exit conditions:

- Player selects “Back” button from the Settings screen.

Success Scenario Event Flow:

1. Player chooses “Settings”.
2. System shows the Settings.
3. Player clicks the back button on the Settings screen.
4. System goes back to the main menu.

Alternative Event Flows:

1. If Player wants to return main menu:
    a. Player selects "Back"
    b. System exits Settings screen.
    c. System show the Main Menu.
2. If Player wants to return a game:
    a. Player selects "Back"
    b. System exits Settings screen.
    c. Player returns Pause Menu.
    d. Player selects "Back"
    e. System exits Pause Menu.
    f. Player returns a game.
3. If Player wants to change volume:
    a. Player selects Adjust Volume
    b. Player decides Volume Level.

4. If Player wants to change language:
    a. Player selects Language
    b. Player changes Language
    c. System updates the game language.
5. If Player wants to change key:
    a. Player can selects every different key
    b. Player changes the button selected by Player
    c. System updates the game play keys.

5.1.11 Quit

Use Case: Quit

Primary Actor: Player

Stakeholders and Interests:

- Player selects Quit
- System shows the Confirmation Screen.

Pre-conditions:

- Player must be in the Main menu.

Post-conditions:

Entry-conditions:

- Player should choose the Quit.

Exit conditions:

Success Scenario Event Flow:

1. Player chooses “Quit”.

2. System exits the game.

Alternative Event Flows:

- No alternative events.

5.1.12 High Score/Achievements Achievements

Use Case: High Score/Achievements Achievements

Primary Actor: Player

Stakeholders and Interests:

- Player selects HighScore
- Player selects Achievements
- System shows the Achievements.

Pre-conditions:

- Player must select High Scores/Achievements.
- Player must select Achievements in the High Score/Achievements Screen.

Post-conditions:

Entry-conditions:

- Player should choose the High Score/Achievements.
- Player should choose the Achievements

Exit conditions:

- Player should choose the "Back"

Success Scenario Event Flow:

1. Player chooses “High Score/Achievements”.
2. Player chooses "Achievements".

3. System shows Achievements Screen.

Alternative Event Flows:

- No alternative event flows.

5.1.13 Create Level

Use Case: Create Level

Primary Actor: Player

Stakeholders and Interests:

- Player selects Create Level
- Player selects Flat/Diagonal/3D
- Player selects color for every pieces
- Player selects which piece will be removed on the game board.

Pre-conditions:

- Player must select Create Level in the main menu.
- Player must select Flat/Diagonal/3D.
- Player must select at least one piece remove from the game board.

Post-conditions:

Entry-conditions:

- Player should choose the Create Level.
- Player should choose "Done"

Exit conditions:

- Player should choose the "Back"

Success Scenario Event Flow:

1. Player chooses “Create Level” in the main menu.

2. Player chooses "Done" in the create level screen.

3. System creates and saved level.

Alternative Event Flows:

1. If Player wants to return to main menu:

    a. Player chooses create level in the main menu

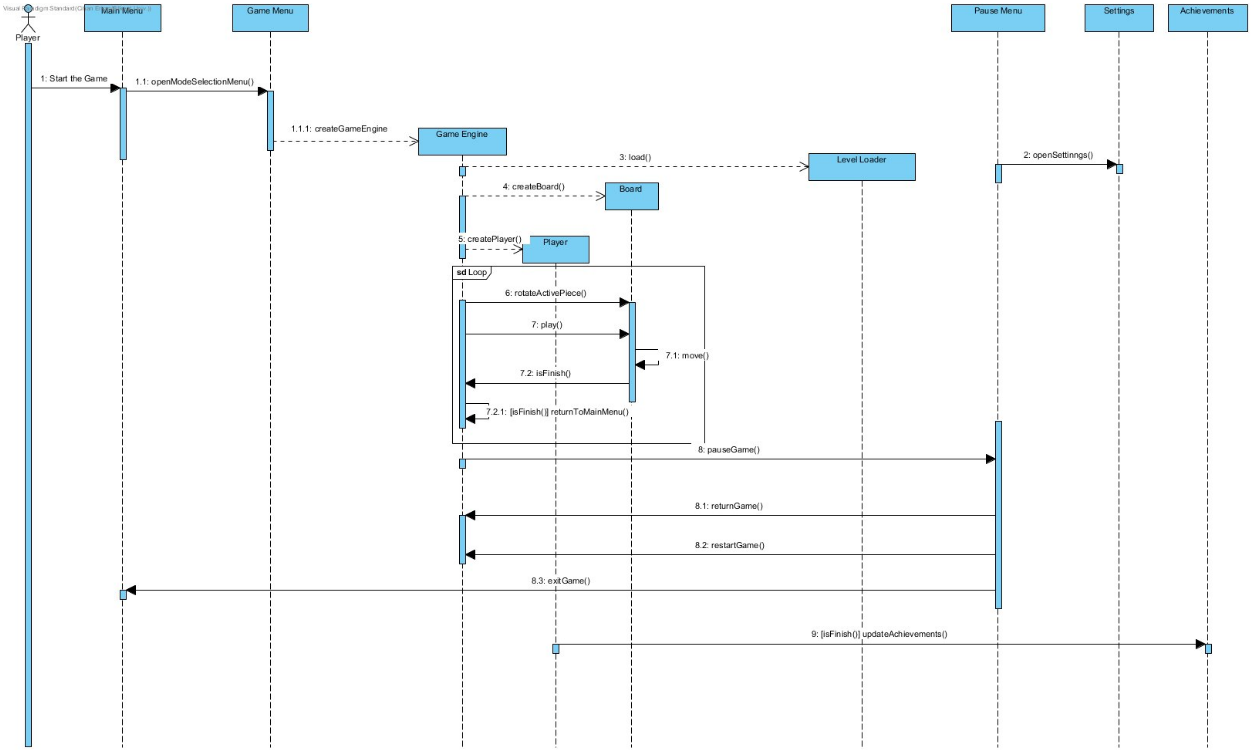
    b. Player chooses back in the create level screen.

    c. Player return to the main menu.

5.2 Dynamic Models

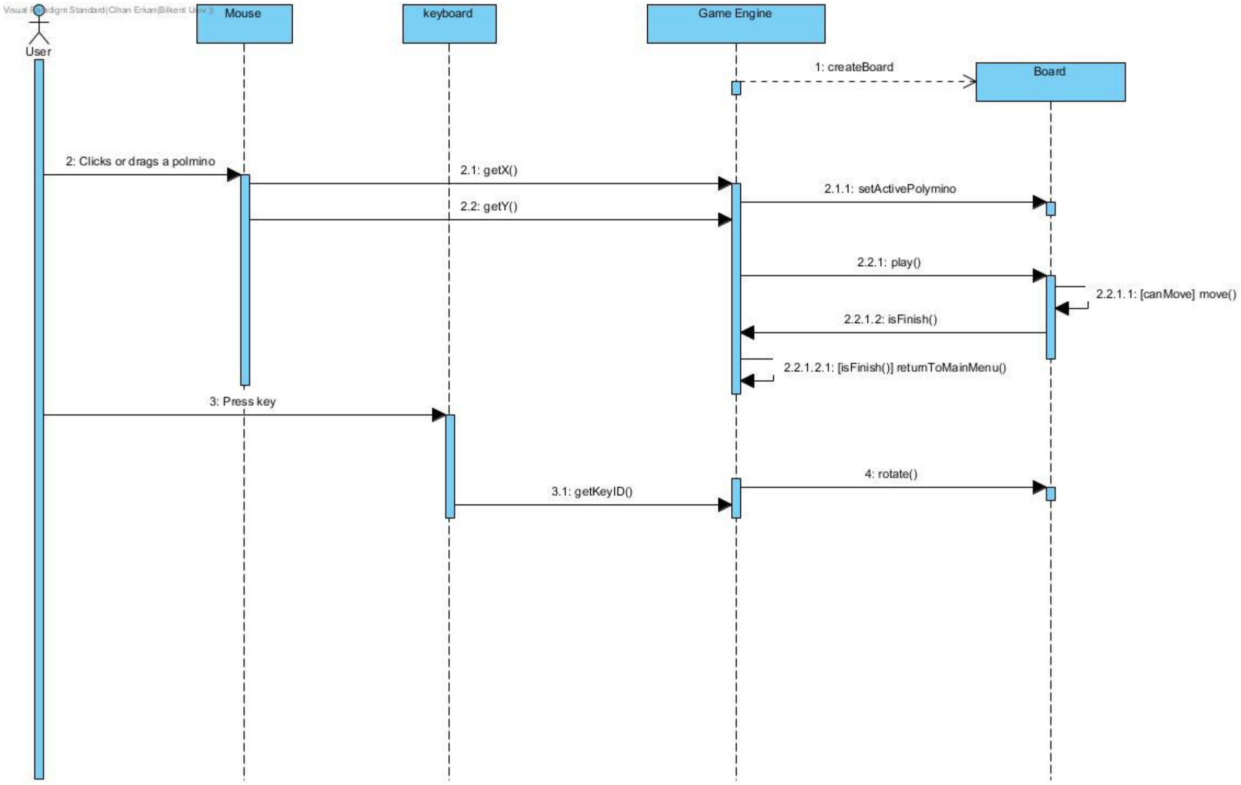
5.2.1 Sequence Diagrams

5.2.1.1 Start Game



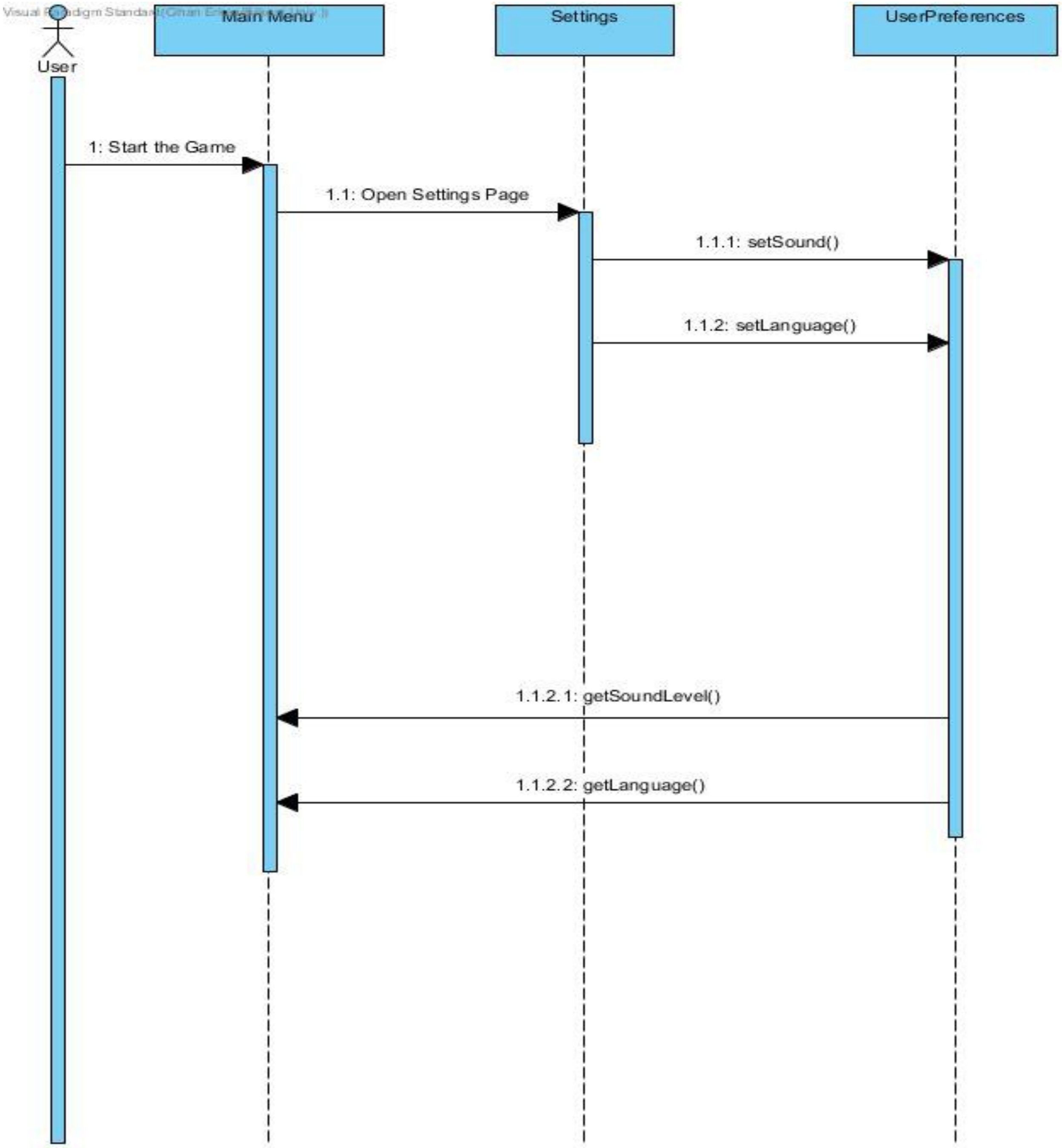
To start game, user runs the application of the game and presses play button and then, specifies what kind of game he/she wants to play by selecting menu options. After that, Game Engine will be created depending on the choices of the user. Game Engine creates instances of Player and Board classes and fills Board by using load() function of LevelLoader class. After this point game will begin. User can move or rotate any of the available pieces by using play() and rotateActivePiece functions. If the move that user wants make is allowed, Game Engine will call tell the board to make move and this continues until all of the empty spots in the board filled. After every move Board will call isFinish() function and if all empty spots are filled, this function will return true. Then Game Engine will update Achievements and then it will return to Main Menu.

5.2.1.2 Moving Single Piece



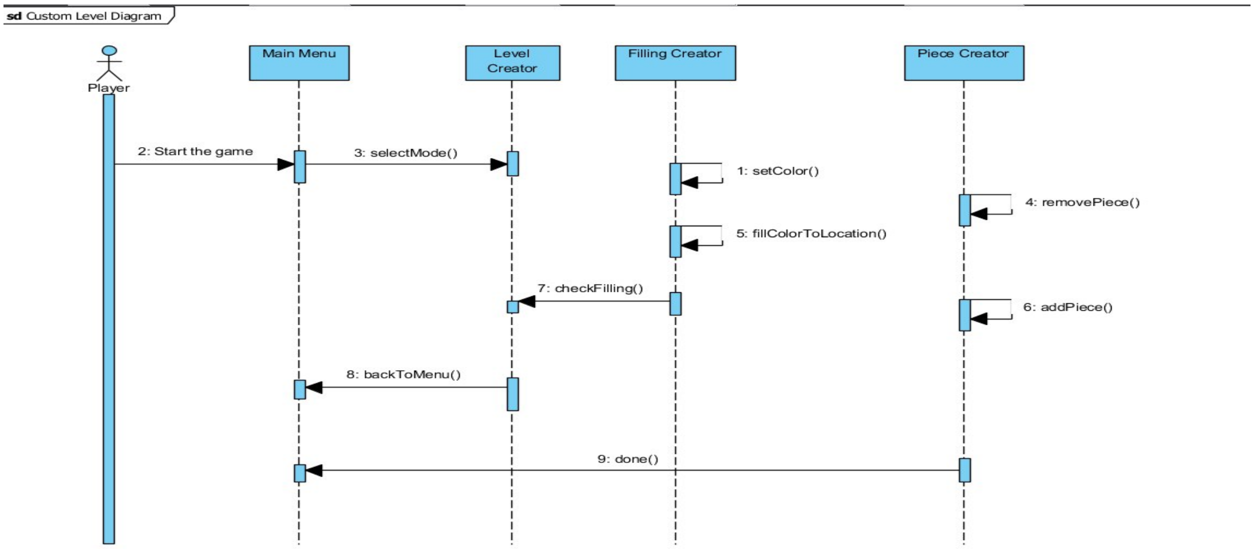
When game starts, user can click on one of the available pieces to activate them by using setActivePolymino() function and user also allowed to rotate selected piece by using keyboard. After that, user drags the piece to the desired location by using play() function of Game Engine. Then play() function calls move() function of the Board and if move is allowed Board move the piece to desired location with move() function. After that, Board checks if the game is finished and if it’s not, allows user to make another play.

5.2.1.3 Settings



User accesses settings page through Main Menu or Pause Menu with openSettingsPage() function. In settings page, user can set sound level and language by using setSound() and setLanguage() functions.

5.2.1.4 Custom Level

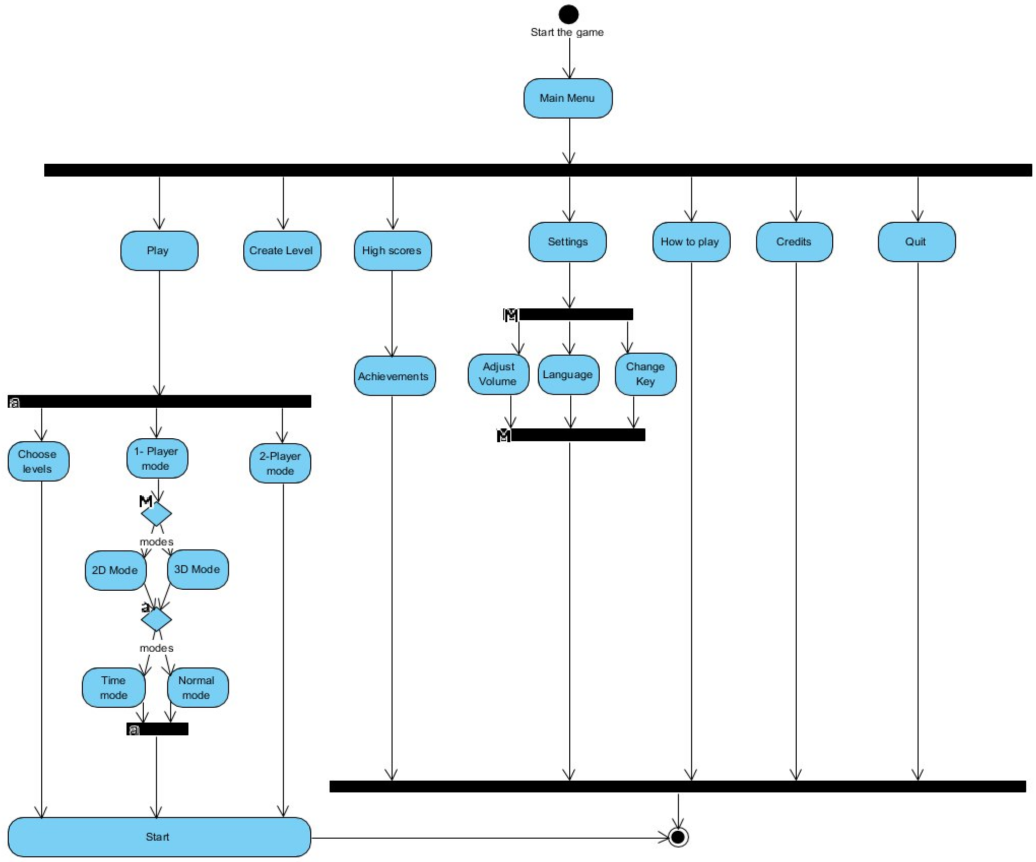


To make a custom level, it first selects the desirable mode by player. After the mode selection, the player is presented with the board interface with empty holes and the colours to fill certain holes by the given buttons.

After setting colour and filling certain, the player clicks the check button to see if the conditions are satisfied according to the game rules.

Another presented option is to remove filled pieces by remove button. If the player decides to cancel the level creation, there is a 'back to menu' on the bottom of the screen available

5.2.2 Activity Diagram

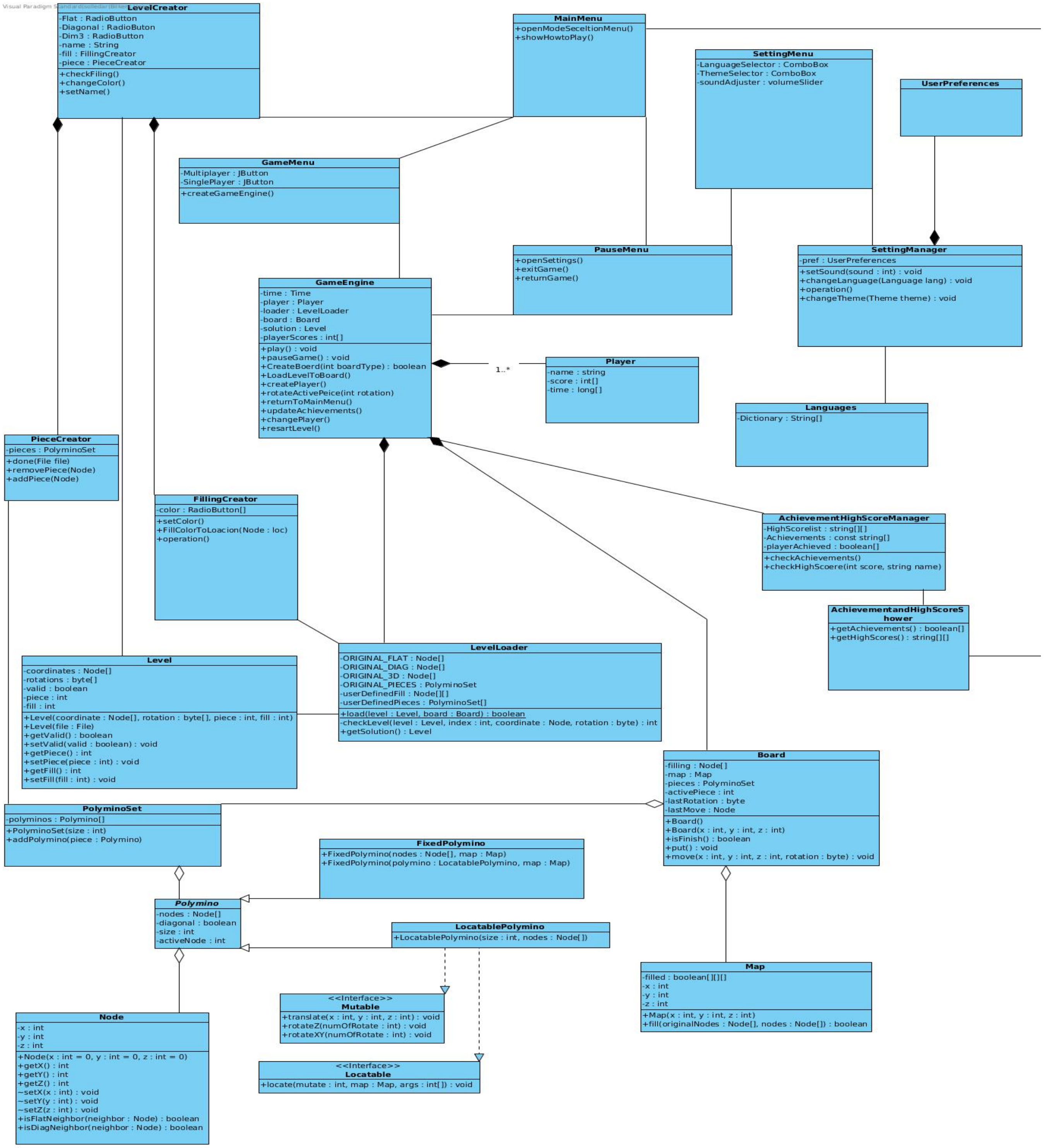


The activity diagram demonstrates the step by step actions and how the system runs the game. While starting the game, it brings the player to choose options in Main Menu in which there are 7 actions such as Play, Create Level, High Scores, Settings, How to play, Credits and Quit.There are also sub-actions accordingly to let the player choose different mode in game, to change the language of the game, to adjust volume and to change key of the keyboard.

When the player choose to play, 3 types of game options are given such as choosing available levels, or the player mode in which either 1 or 2 player can play at the same to compete themselves. In single player mode, there are also other modes such as 2D mode or 3D mode which refers to game mode and difficulty level of the game. In addition there is also Time mode option served for the player in the case they want to challenge themselves.

There are also other options in Settings part. The player will be able to adjust the volume of the game sound, change the language and alternate the keys for the play.

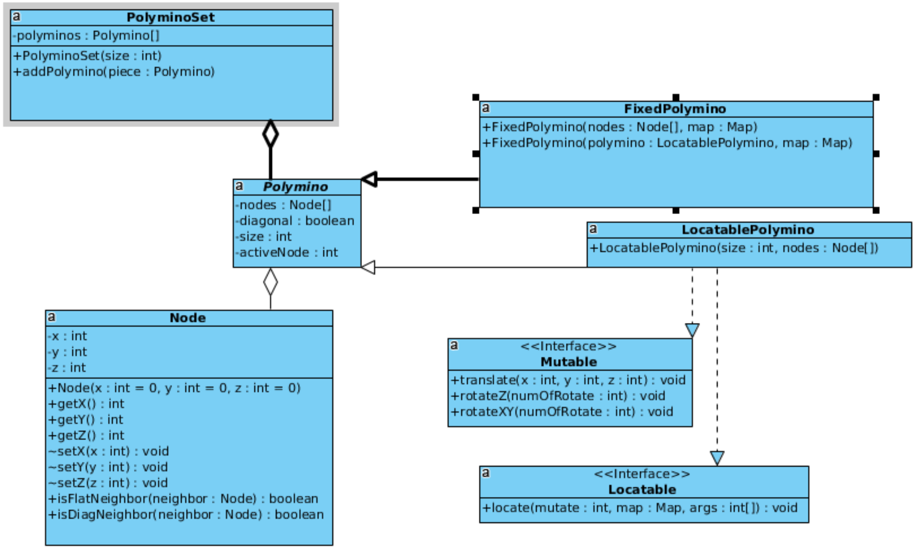
5.3 Object and Class Model



There are 4 package which are named Pieces, Game, LevelUtilities, UserInterface.

5.3.1 Pieces Package

This package contains physical pieces models and determines their basic behavior.


PolyminoSet
-polyminos : Polymino[]
+PolyminoSet(size : int)
+addPolymino(piece : Polymino)
FixedPolymino
+FixedPolymino(nodes : Node[], map : Map)
+FixedPolymino(polymino : LocatablePolymino, map : Map)
Polymino
-nodes : Node[]
-diagonal : boolean
-size : int
-activeNode : int
LocatablePolymino
+LocatablePolymino(size : int, nodes : Node[])
Node
-x : int
-y : int
-z : int
+Node(x : int = 0, y : int = 0, z : int = 0)
+getX() : int
+getY() : int
+getZ() : int
~setX(x : int) : void
~setY(y : int) : void
~setZ(z : int) : void
+isFlatNeighbor(neighbor : Node) : boolean
+isDiagNeighbor(neighbor : Node) : boolean
<<Interface>>
Mutable
+translate(x : int, y : int, z : int) : void
+rotateZ(numOfRotate : int) : void
+rotateXY(numOfRotate : int) : void
<<Interface>>
Locatable
+locate(mutate : int, map : Map, args : int[]) : void

5.3.1.1 Node Class

This class represents atomic piece of the game. It has coordinates, which are x, y, z in the space. This class instances are able to check that another node is neighbour of itself in terms of x-y plane or diagonal plane. It will be used in construction of complex pieces but it can be also used for abstract checks in game.

5.3.1.2 Polymino Class

This class is an abstract class for 2D multi-node item. Because of abstractness there is no restriction to create 3D items but in generic usage it will be 2D in the child classes.

5.3.1.3 FixedPolyminos Class

It is a variant of the Polymino Class. The point is that these are cannot be moved. They are always on the same location after created. It can be converted from LocatablePolymino.

5.3.1.4 LocatablePolymino Class

This class represents polyminos, which are composition of nodes with locational and rotationally changeable.

5.3.1.4.1 Mutable Interface

This interface provides changes for nodes.

5.3.1.4.2 Locatable Interface

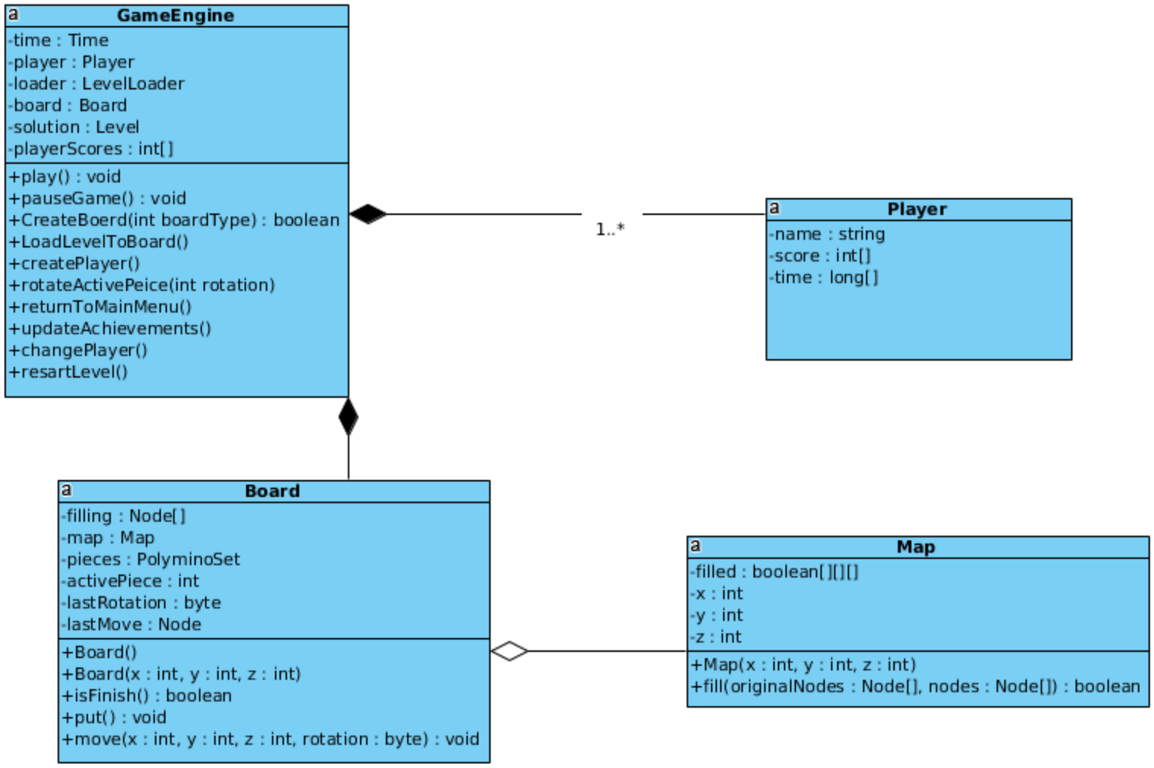
This interface provides checks for locating proper.

5.3.1.5 PolyminoSet Class

This class provide set of polyminos, which contains 12 pieces for the level creation in original game.

5.3.2 Game Package

This package provides fundamental part of the game. Actually, this package is main part as “playable”.



5.3.2.1 Map Class

Job of this class is supplying communication between pieces. It acts as 3D space in game. It stores where is filled and not. It checks whether somewhere

5.3.2.2 Board Class

This class contains a map and polymino set. It coordinates movement of pieces and checks if the game is ended or not.

5.3.2.3 Player Class

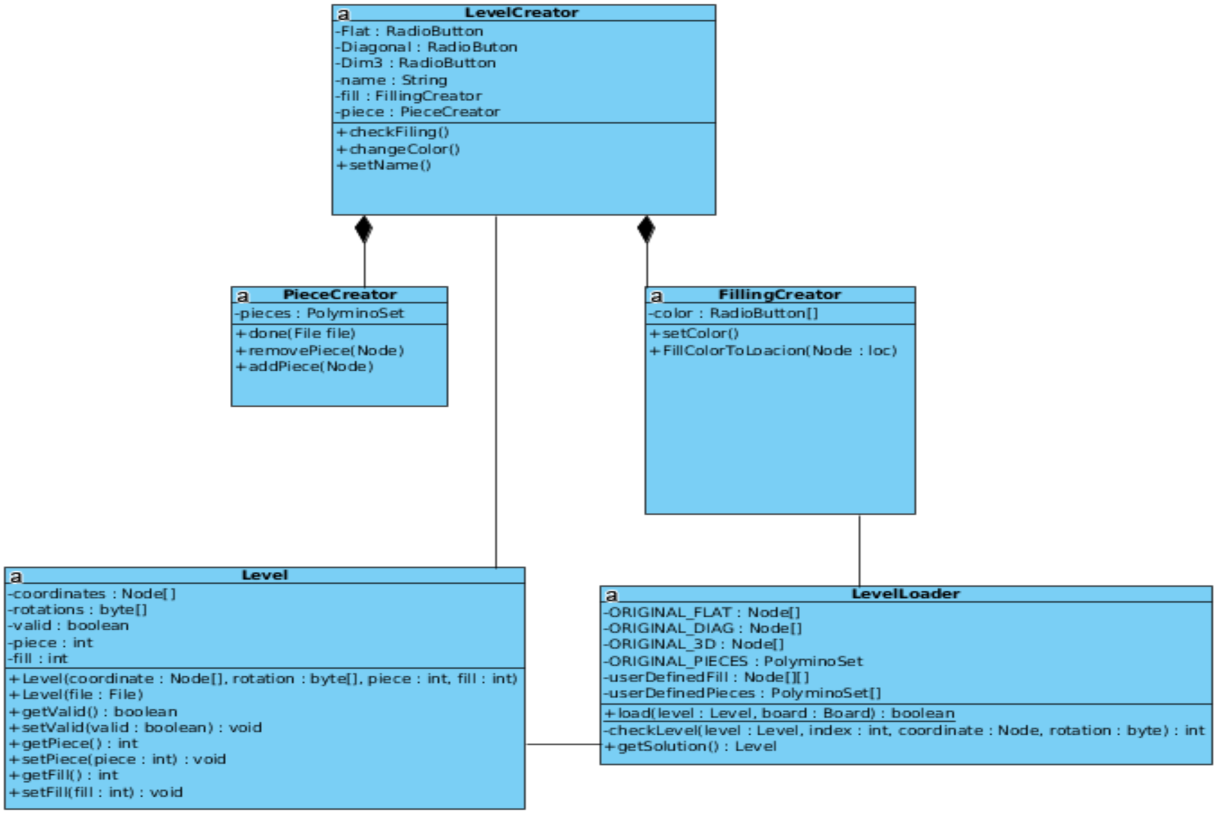
This class represents player for scores and two player utilities.

5.3.2.4 GameEngine Class

This class orients the game package components with other packages. Firstly, it creates player and boards. Secondly it ensures level is loaded. After that point it is responsible for interaction between player and board until level is completed. According to game mode (normal / time)

5.3.3 LevelUtilities Package

This package comprises classes related with level creating, storing and loading.



5.3.3.1 Level Class

It is modelling a level in terms of goal filled area, piece set and fixed pieces location.

5.3.3.2 LevelLoader Class

This class loads levels into game engine with complete information, which is partially included level class such as filling type and piece set.

5.3.3.3 LevelCreator Class

This class creates custom level by using 2 helping classes. It provides different filling areas and different piece set with original game. Creator can use original sets or add new set. It compares area and pieces to conclude that they are compatible. Creator locates fixed pieces and the class store the level in file.

5.3.3.4 FillingCreator Class

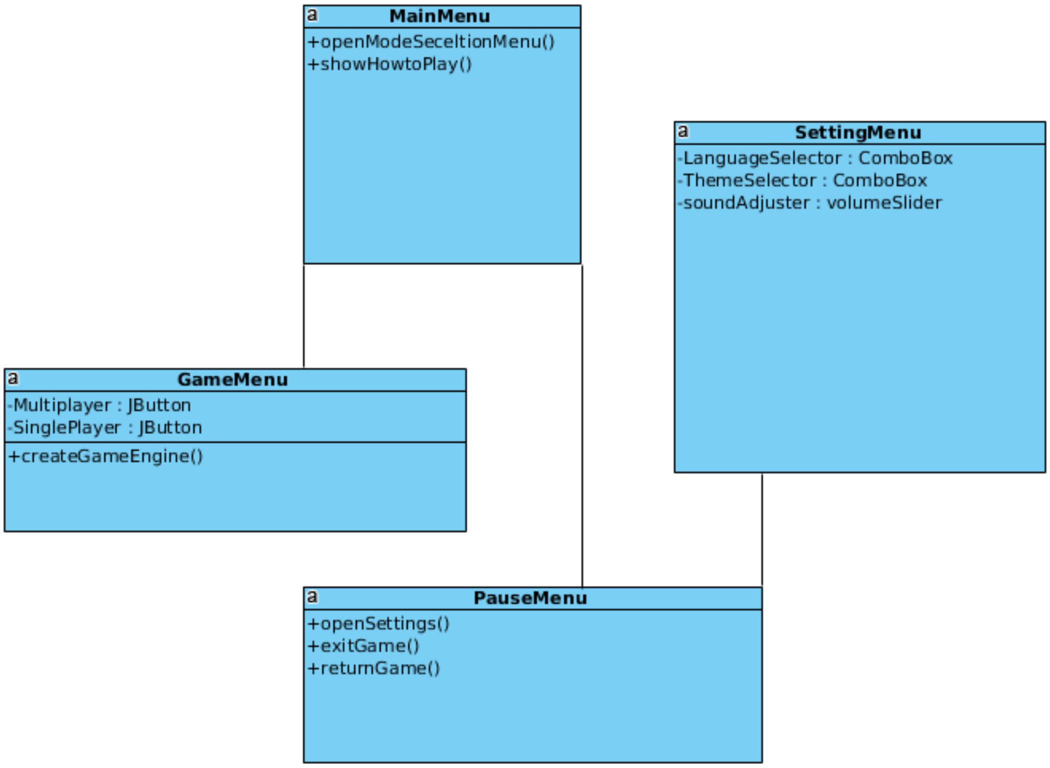
This class creates new filling area which is filled to complete level. It is one of auxiliary classes for LevelCreator.

5.3.3.5 PieceCreator Class

This class is second helper class of LevelCreator. It creates new set of pieces for the game.

5.3.4 UserInterface Package

This package contains menus to navigate the user to use many functionalities especially playing game.



5.3.4.1 MainMenu Class

This class is starting point of the program. It serves to reach game menu, high score and achievements, settings, credits, how to play and also quitting from program.

5.3.4.2 GameMenu Class

This class contains link to game modes which are Story Mode, Custom Mode and Creating Level Mode.

5.3.4.3 SettingMenu Class

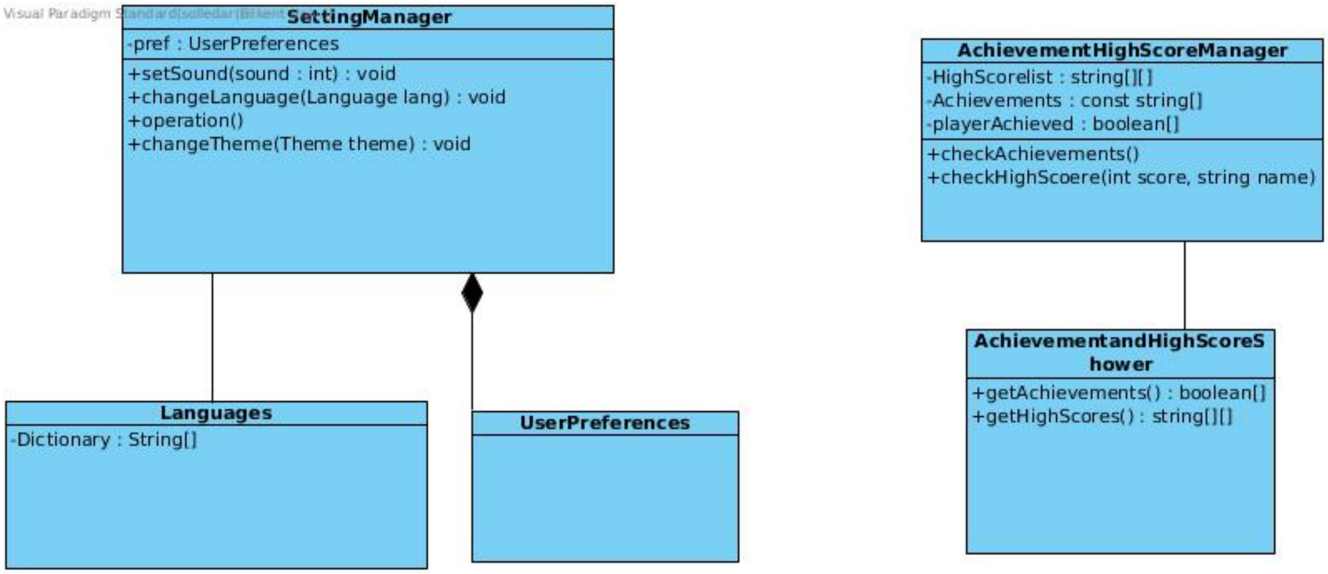
Task of this menu is changing theme and language, adjusting sound and mute/unmute the game.

5.3.4.4 PauseMenu Class

When user plays game to quit or canalize other part of the game, this menu will be accessible by pressing “P”.

5.3.5 SettingAndUtilities Package

This package contains settings, high score and achievements classes.


SettingManager
-pref : UserPreferences
+setSound(sound : int) : void
+changeLanguage(Language lang) : void
+operation()
+changeTheme(Theme theme) : void
Languages
-Dictionary : String[]
UserPreferences
AchievementHighScoreManager
-HighScorelist : string[][]
-Achievements : const string[]
-playerAchieved : boolean[]
+checkAchievements()
+checkHighScoere(int score, string name)
AchievementandHighScoreShower
+getAchievements() : boolean[]
+getHighScores() : string[][]

5.3.5.1 SettingManager Class

This class implements desires of user which comes from setting menu.

5.3.5.2 AchievementAndHighScoreShower Class

This class shows achivements which user accomplished and high scores which user can look to compare his/her rank on game.

5.3.5.3 AchievementAndHighScoreManager Class

This class updates achievements and high scores according to circumstances of completeness and time after level ends.

5.3.5.4 Languages Class

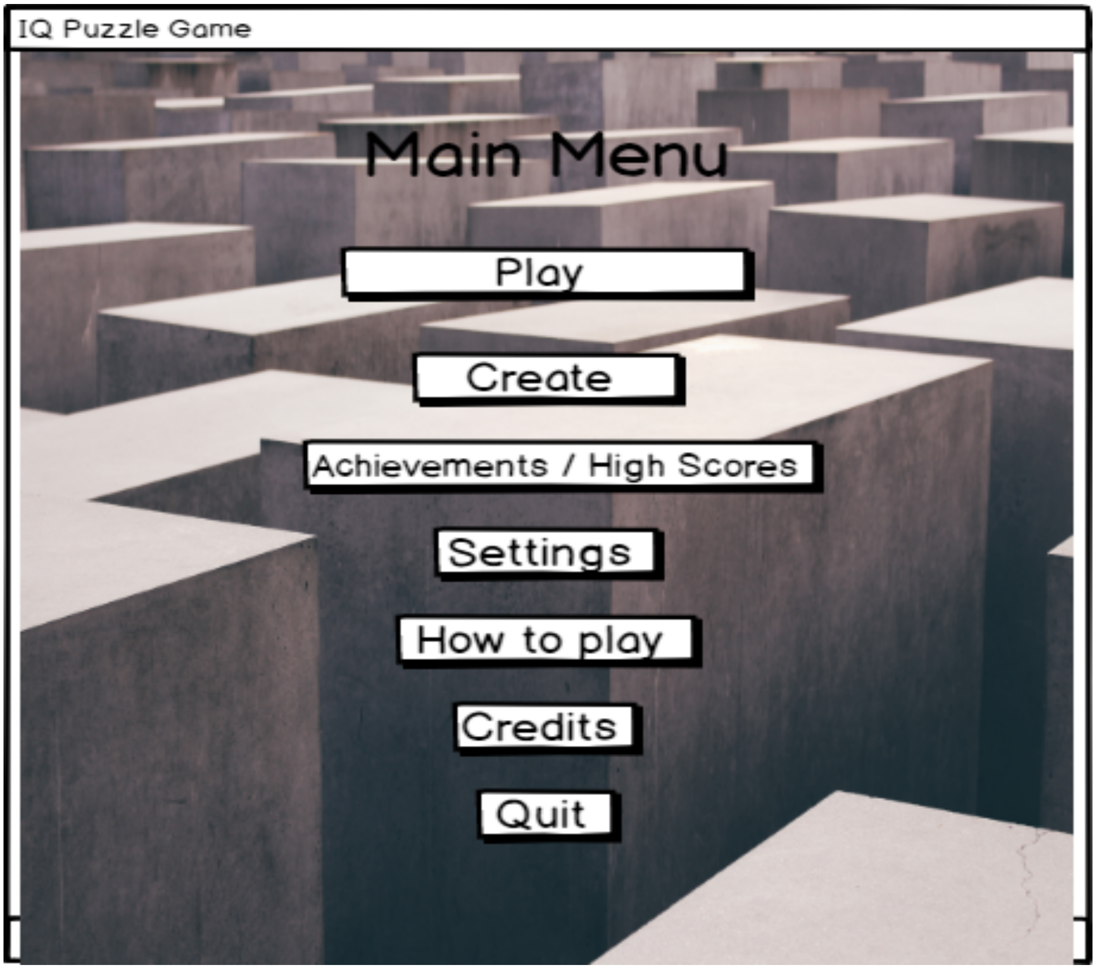
This class represents languages for menus.

5.3.5.5 UserPreferences Class

This class holds preferences of user.

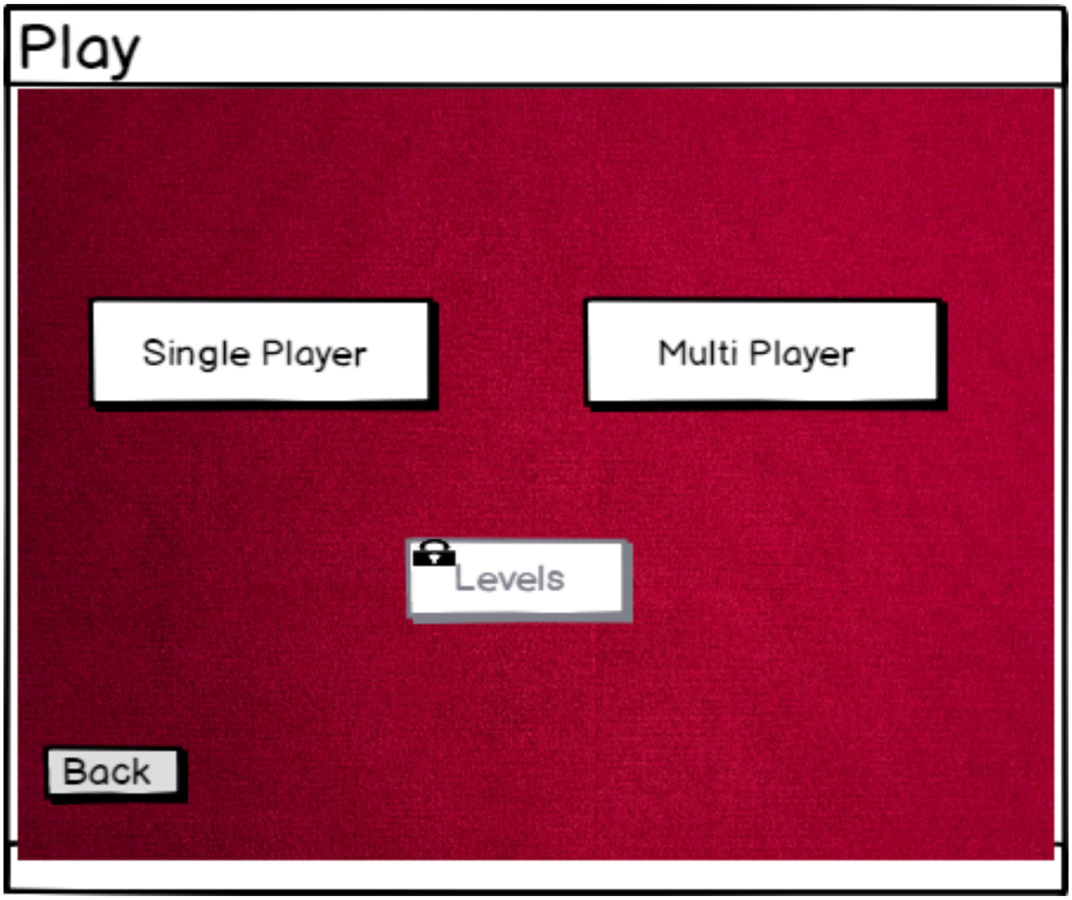
5.4 Screen Mockups

5.4.1 Main Menu



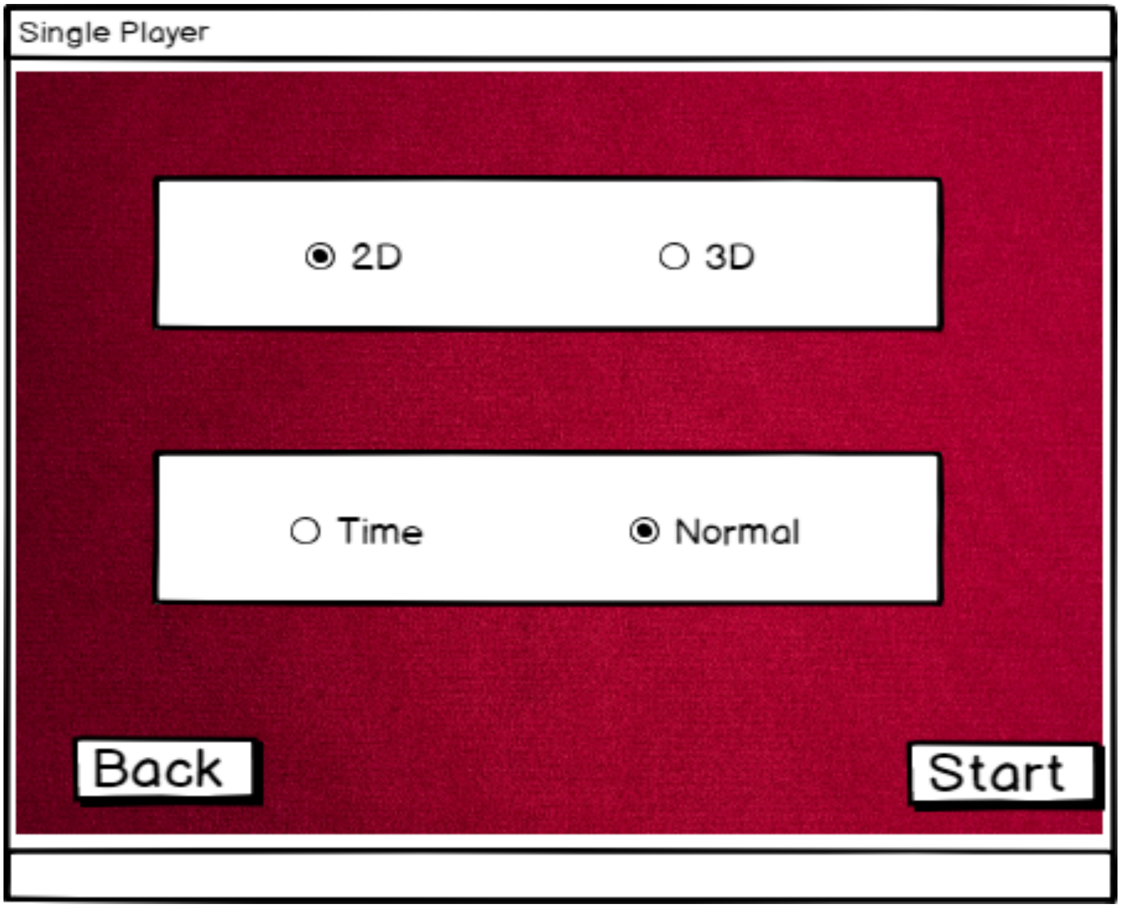
Main Menu is start page for IQ puzzle game.

5.4.2 Play Button



Player should choose Single Player, Multi Player or Levels, after clicked button. He/she can back to the main menu via Back button.

5.4.3 Single Player

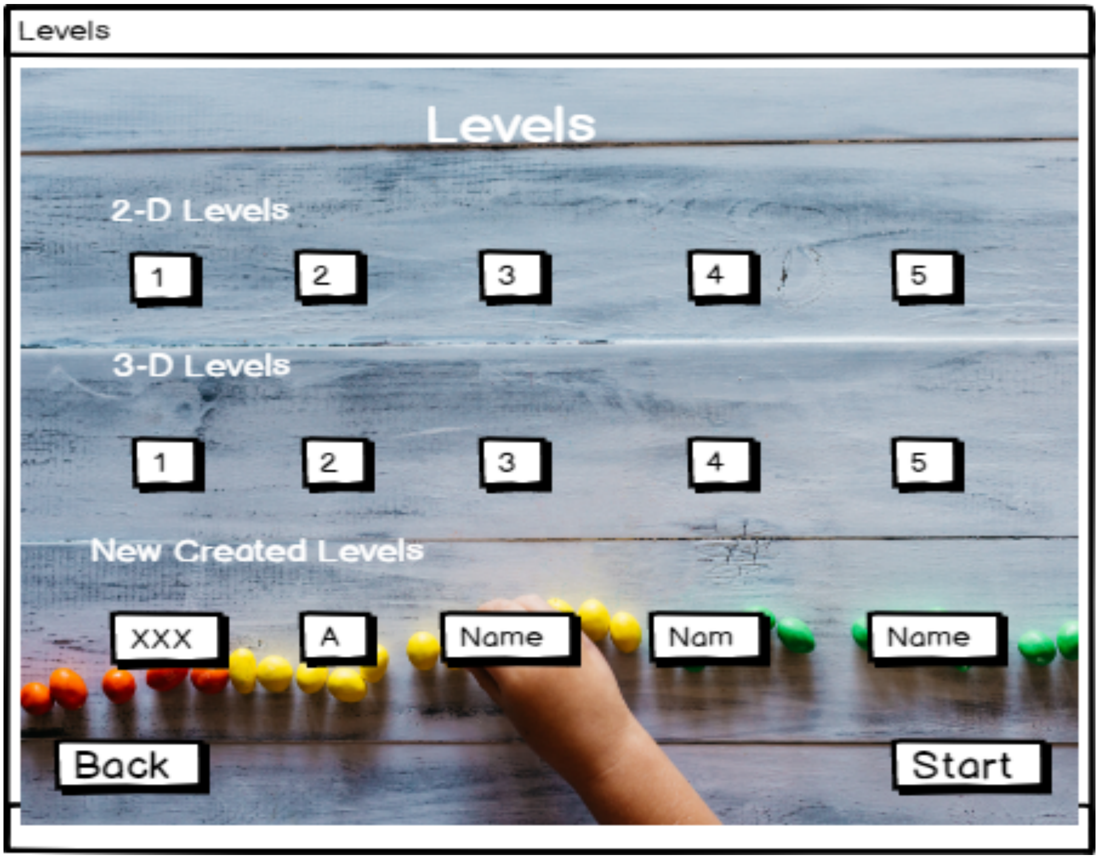


● 2D ○ 3D

○ Time ● Normal

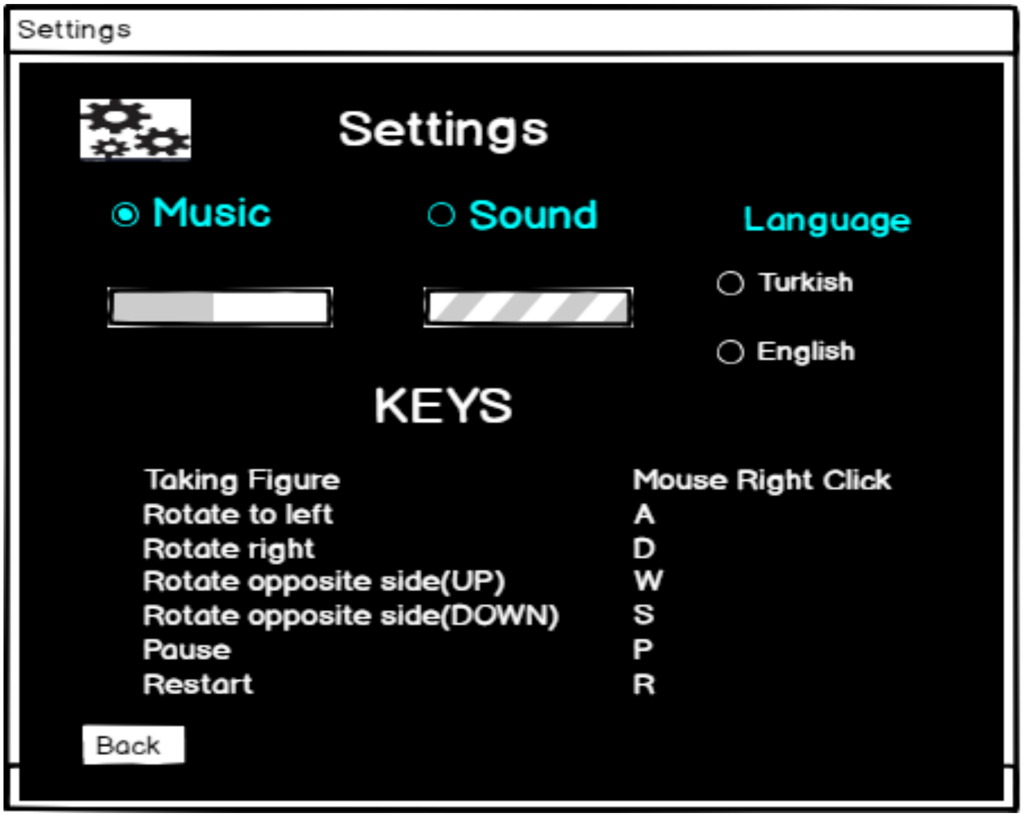
Player can choose 2D or 3D game formats and also limited/limitless time versions for the game.

5.4.4 Levels



Levels part will be paid and user can play all levels in this mode

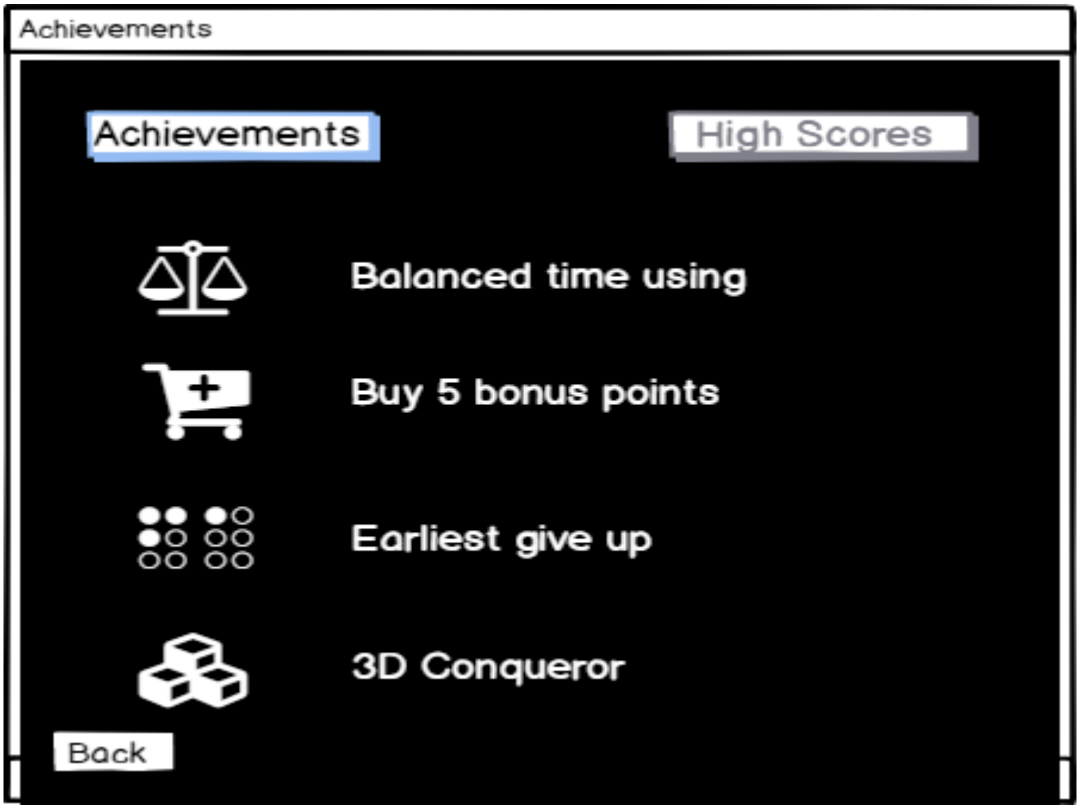
5.4.5 Settings



Taking Figure	Mouse Right Click
Rotate to left	A
Rotate right	D
Rotate opposite side(UP)	W
Rotate opposite side(DOWN)	S
Pause	P
Restart	R

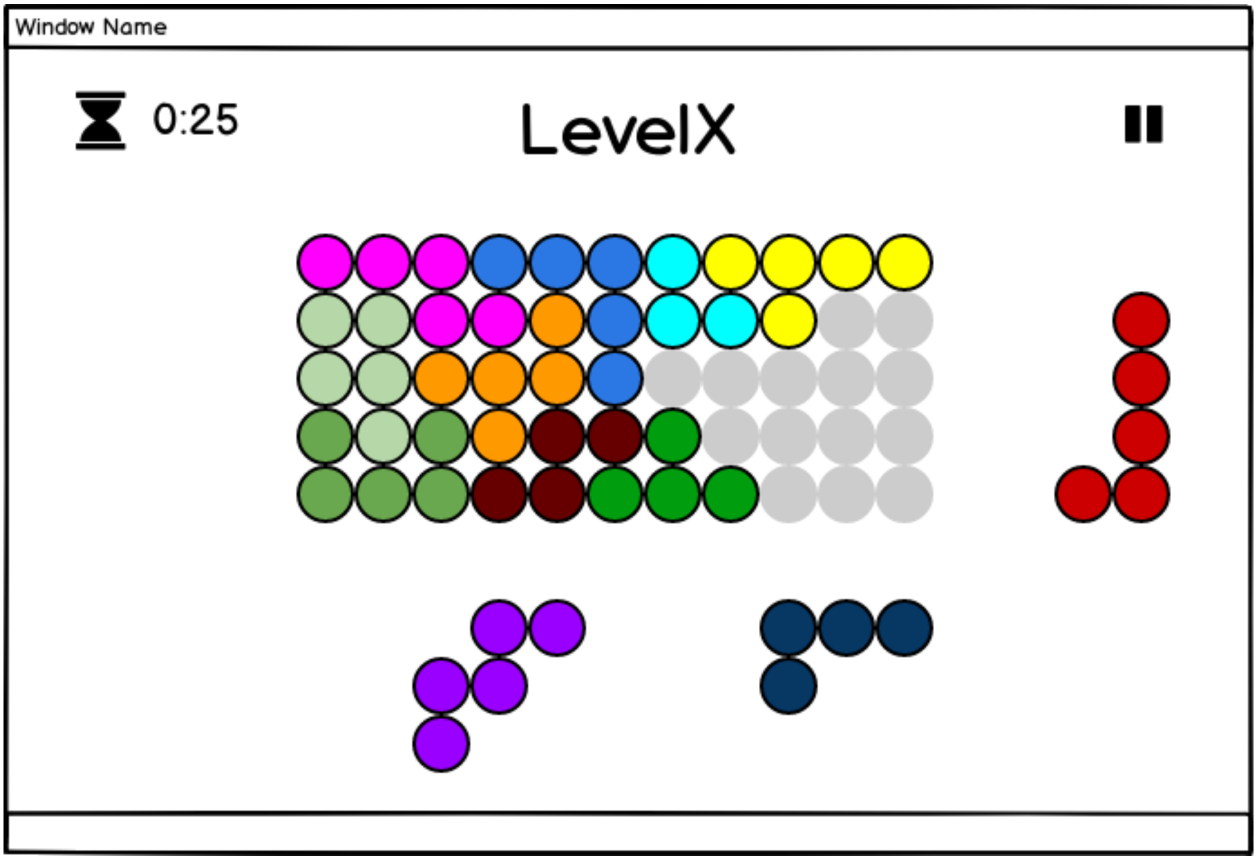
Settings part player can adjust volume and language. He/she can change keys whatever comfortable for them.

5.4.6 Achievements/High Scores



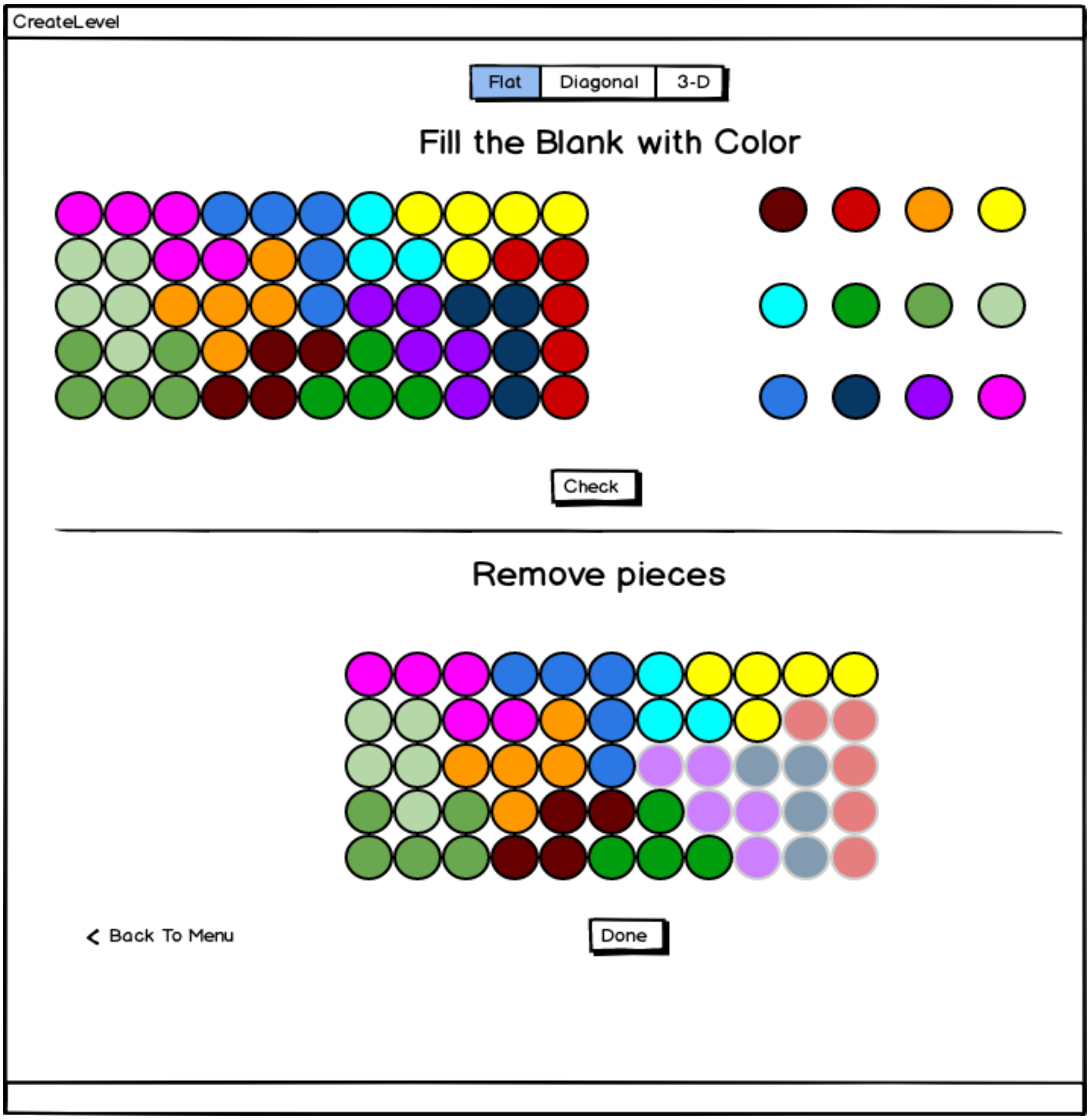
People can see the High Scores and Achievements.

5.4.7 Game Interface



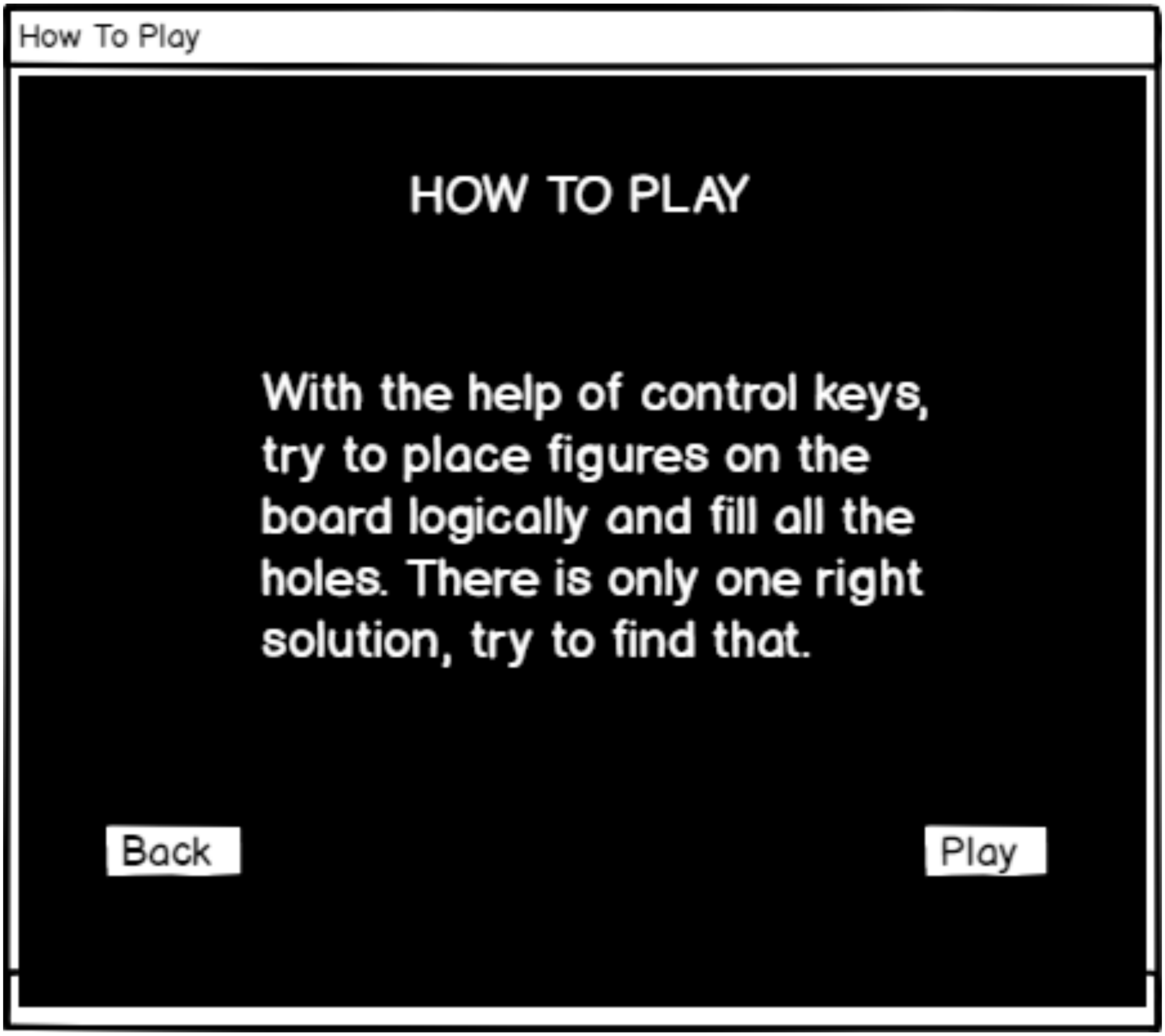
This is the screen that player plays the game.

5.4.8 Create Level Screen



In create level part, player can create his/her own level, save and play this level.

5.4.9 How To Play Screen

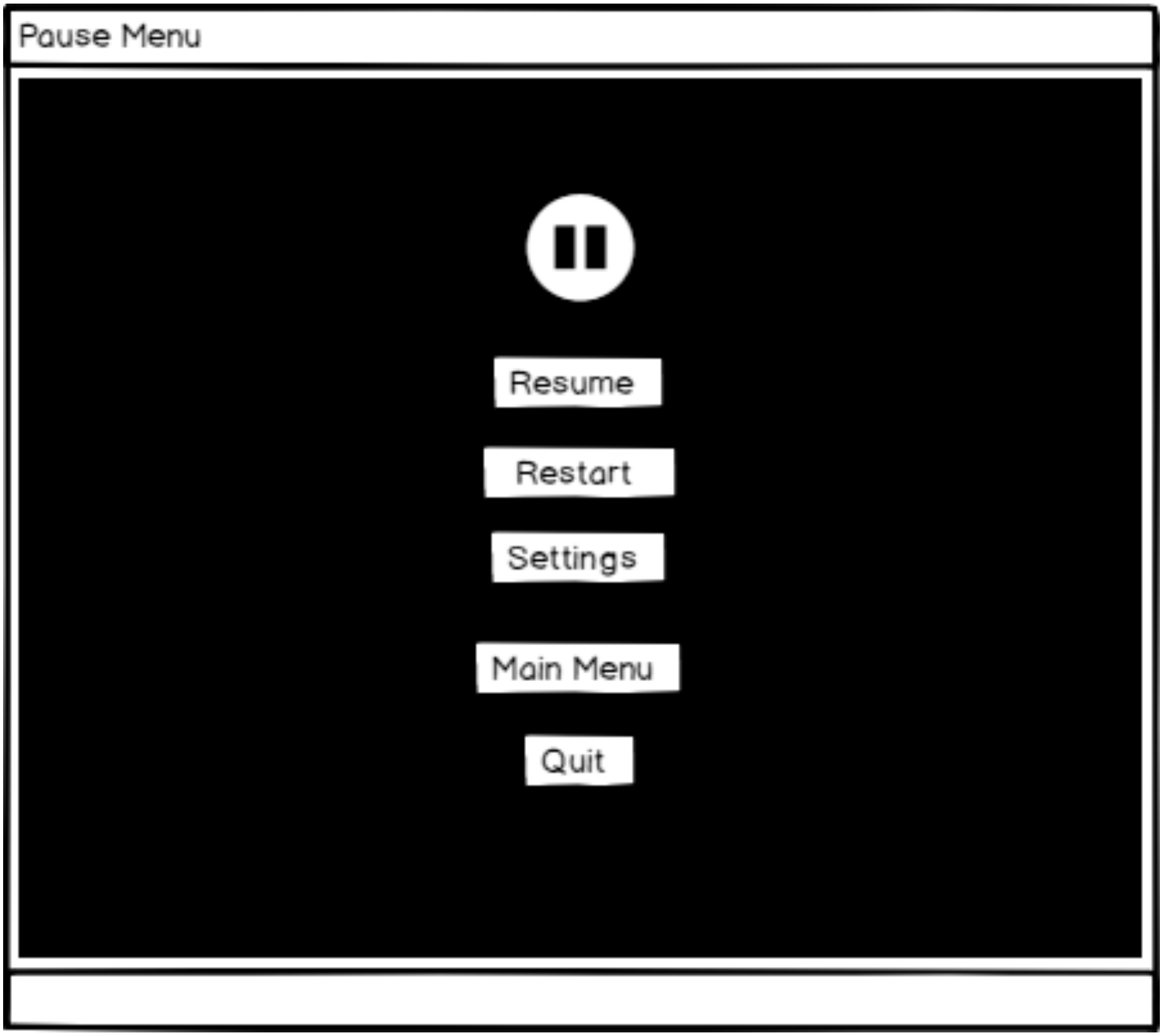


HOW TO PLAY

With the help of control keys, try to place figures on the board logically and fill all the holes. There is only one right solution, try to find that.

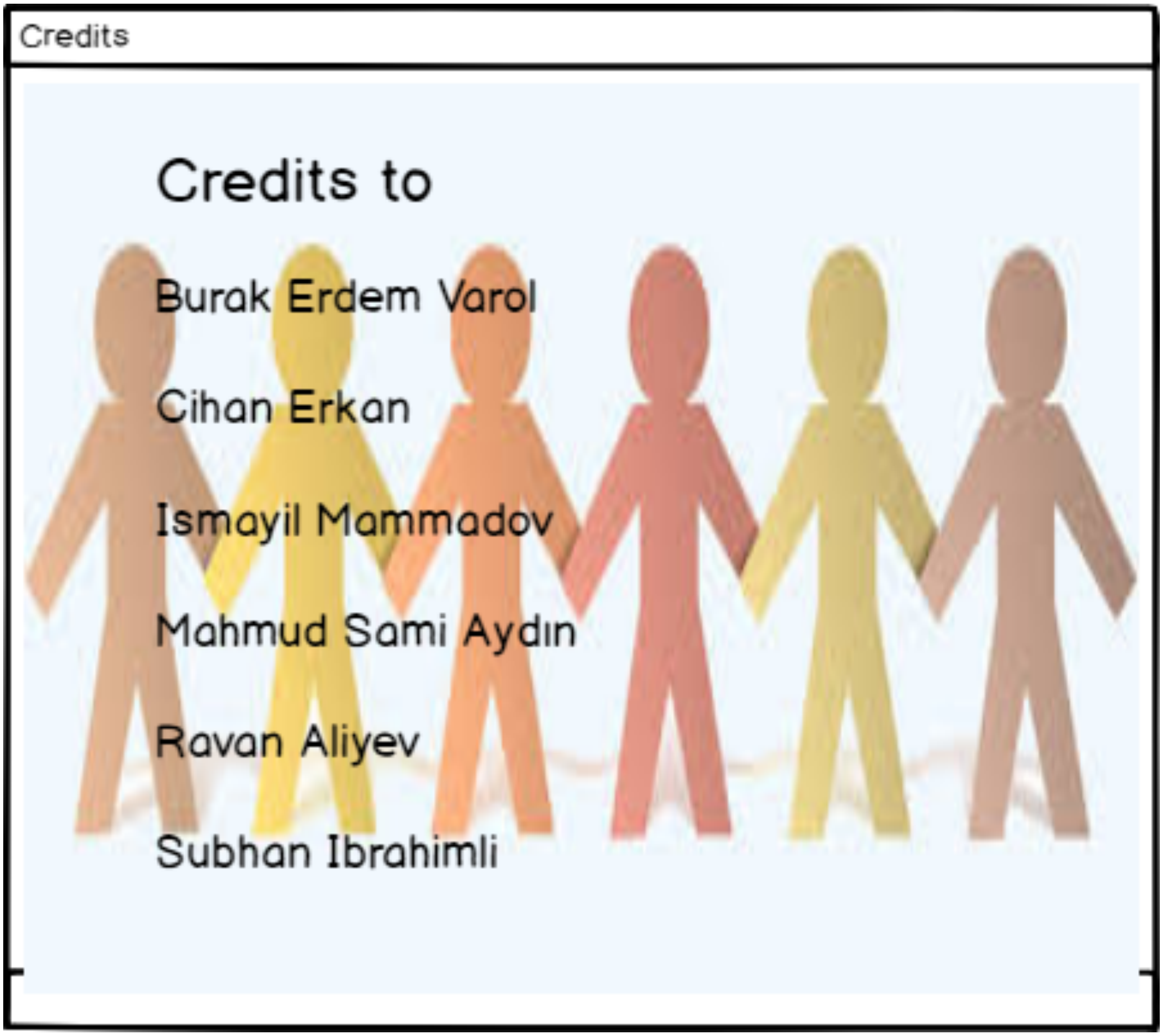
Player can get hints about how to play this game

5.4.10 Pause Screen



Player can stop the game, adjust volume or quit the game via pause game button.

5.4.11 Credits Screen



Credits

Credits to

Burak Erdem Varol

Cihan Erkan

Ismayil Mammadov

Mahmud Sami Aydın

Ravan Aliyev

Subhan Ibrahimli

Player want to know who creates the game.

6. Conclusion

By doing the project, our aim is not just to finish the project itself but to improve our skills as a team member in terms of communication, project organization, collaboration, technical writing and many so on. Providing the required report on time, being interactive by contributing creative ideas and picking up the better ones among them are one of the essence skills considered to gain during the project.

In the analysis report, we have clarified what are the essences for the project. We tried to provide all the details of each required element and tried to explain what they are for and the usage of them.

Consequently, we have demonstrated our path and aware of what we are supposed to do to satisfy required job. We believe that this report makes us to have an image about the whole report and makes the job easier in the implementation.

7. References

[1] https://engineering.purdue.edu/INSPIRE/Reviews/puzzles/iq-puzzler-pro

[2] https://vidweb.aws.marketlive.com/learningresources_vid/text/pdf/2978_EIguide.pdf