



Bilkent University

Department of Computer Engineering

CS319 Term Project

Katamino

Analysis Report

Group 1H

Ege Özcan, Mustafa Bayraktar, Simge Tabak, Yağız Mertol, Zeynep Gözel

Supervisor: Eray Tüzün

Progress Report
October 21, 2018

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Term Project course CS319.

Analysis Report

Katamino

Introduction

Overview

Functional Requirements

Nonfunctional Requirements

System Models

5.1. Use case model

5.2. Dynamic models

5.3. Object and class model

5.4. User interface - navigational paths and screen mock-ups

Glossary & References

1. Introduction

In our project for CS319, we will implement the geometrical mind game Katamino. Katamino is a singleplayer 2D mind game, where the player tries to fill the board with the given blocks in different colours and geometrical shapes. The classic katamino game is setup by dividing the board into two parts. After that both players are given five blocks and 4 filler blocks. The first player to fill their section with the given blocks is the winner of the game.¹ The main objective of this project is to make this a desktop game using the fundamentals of object oriented programming. We will use Java to implement this real life puzzle game on desktop.

2. Overview

At the beginning of the game, player chooses the game level and category from the main screen and begins the game. Different boards for easy, medium and hard levels will appear according to the player's selections. Player should fill the board with the given blocks. Board designs will feature pixel art that will give an extra joy to the main target audience that are children and also adults. The users can add their own pixel art as an additional feature as well. Mouse is used to drag the blocks from the given blocks onto the board. After the block is chosen by using left click, it can be rotated ninety degrees clockwise by using the right click.

In addition, a time counter will keep count of the time from the beginning of the level. A move counter will keep count of the moves done by the user in the current level. These will be used to determine the score of the player at the end of the game. An online leaderboard will allow the user to upload their

¹ "Katamino." Game Rules, www.thegamerules.com/en/board-games/family/katamino-907-detail.

score to an online database and compare it with other users. Moreover, we will implement a hint system, which will guide the user towards a solution.

3. Functional Requirements

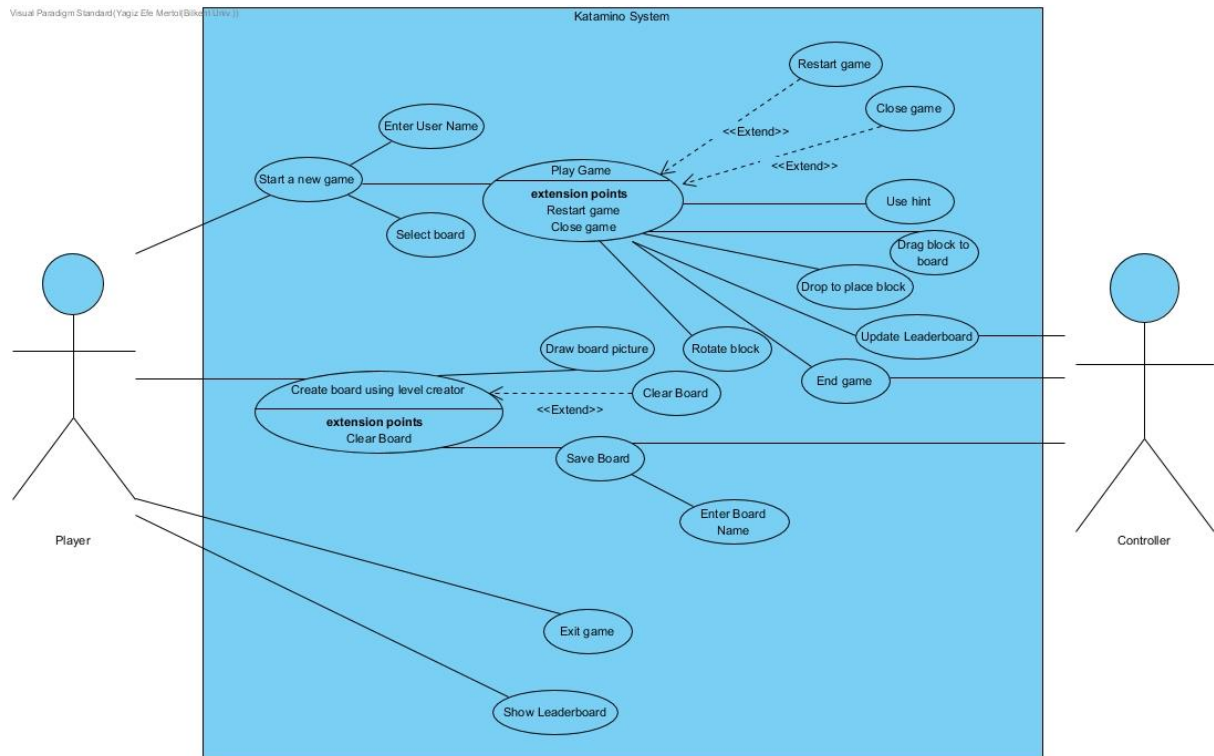
- Start a new game
- Enter user name
- Create board using level creator
- Draw board picture
- Select board
- Rotate block
- Drag block to board
- Drop to place block
- Use hint button
- Create board using level creator

4. Nonfunctional Requirements

- All user inputs should be acknowledged within 1 second
- A system crash should not result in data loss
- Keep high scores and update online leaderboard
- Keep time count from the beginning of the game
- Keep move count from the beginning of the game
- End the game when all blocks are placed correctly
- Give the hint when the user presses hint button

5. System Models

5.1. Use case model



Use Case Scenario 1

Name: Start a new game

Participating Actor: Player

Entry Condition: Player must be in the main menu

Exit Condition:

Player clicks exit button

Player finishes the game successfully

Flow of Events:

1. Player clicks "Play Katamino" button from the main menu.
2. Player enters name.
3. Player selects a board.

4. Player drags a block to the board.
5. Player drops the block to replace the block.
6. Player repeats steps 4-5.
7. Player finishes the game.
8. Controller ends the game

Special Requirements: None

Use Case Scenario 2

Name: Create board using level creator

Participating Actor: Player

Entry Condition: Player must be in the main menu

Exit Condition:

Player clicks exit button.

Player click save board button.

Flow of Events:

1. Player clicks "Create Your Board" button from the main menu.
2. Player draws a board.
3. Player tests the board.
4. Player saves the board.
5. Player enters board name.

Special Requirements: None

Use Case Scenario 3

Name: Show Leaderboard

Participating Actor: Player

Entry Condition:

Player must be in the main menu

Game must be over

Exit Condition:

Player clicks exit button.

Flow of Events:

1. Player clicks "Show Leaderboard" button from the main menu.
2. Leaderboard is shown.

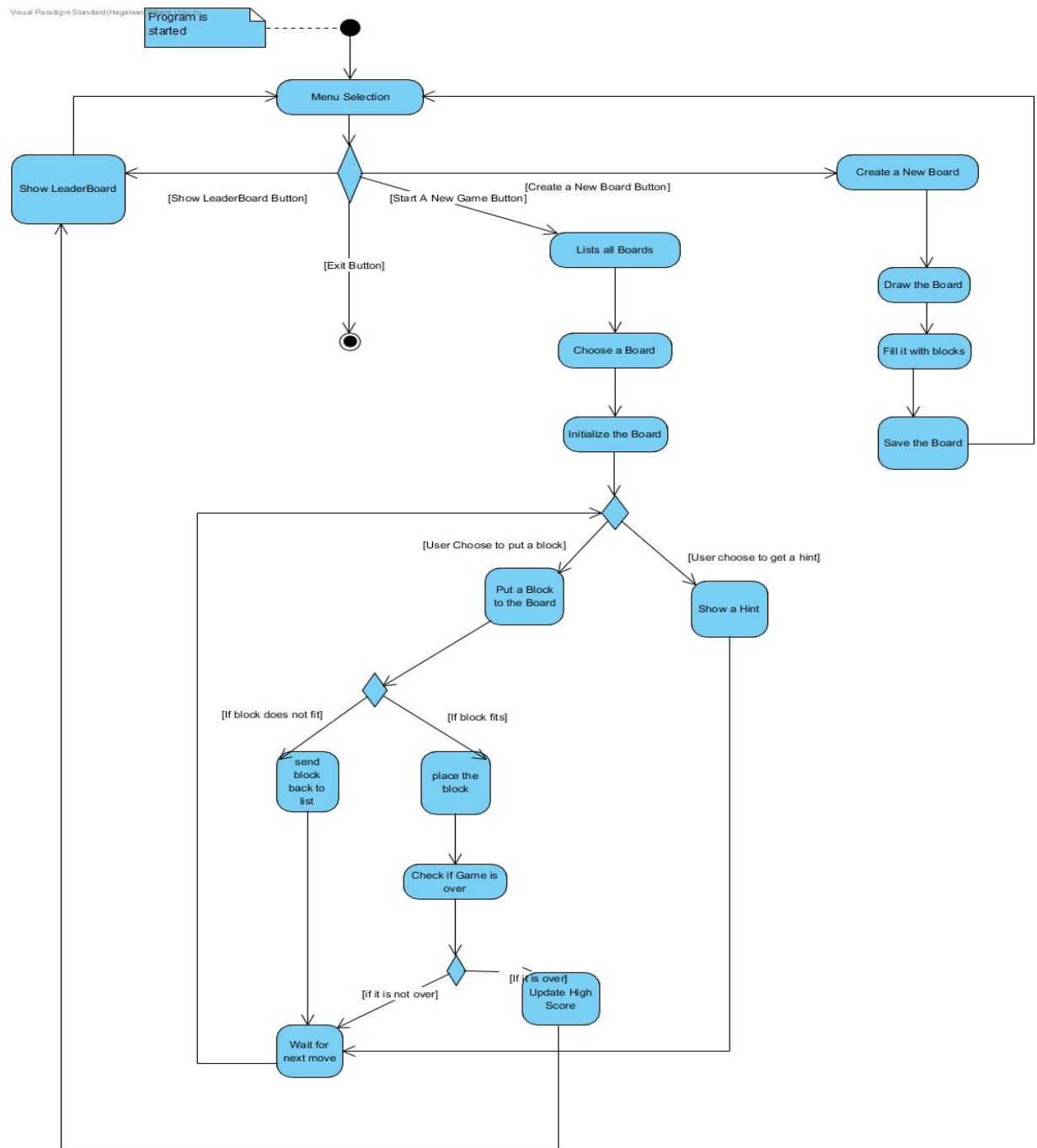
Alternative Flow of Events:

1. The game ends successfully.
2. Leaderboard is shown.

Special Requirements: None

5.2. Dynamic models

Activity Diagram



At the beginning, the program displays a menu which includes start a new game, create a new board, show leaderboard and exit buttons. Then it waits for an input from the user. There are different cases at that point. At each button selection, the user will be faced with different scenarios.

If the user chooses to start a new game, the program lists all playable boards to the user and waits for a selection. After the user selects a board from the list, the program initializes the board. This board will be the game which includes the blocks and the shape of the board, hint button and a timer.

Now the user may either try to put a block to the board or get a hint.

- If the user decides to put a block to the board, the program checks if it is suitable for the place where the user wanted to put.
 - If the block fits, it is placed on the board and then program checks if the game is over or not.
 - If it is over program updates the leaderboard with the data of the game and shows it to the user. This leaderboard sends the user directly to the main menu.
 - If it is not over, the program waits for a new move.
 - If the block does not fit, the program sends the block back to the block list, and then waits for a new move.
- If the user decides to get a hint, the program shows a hint to the user and then starts to wait for a new move.

Another case is the user choosing to create a new board. After the user clicks on the create a new board button, the program displays a new empty board to the user to

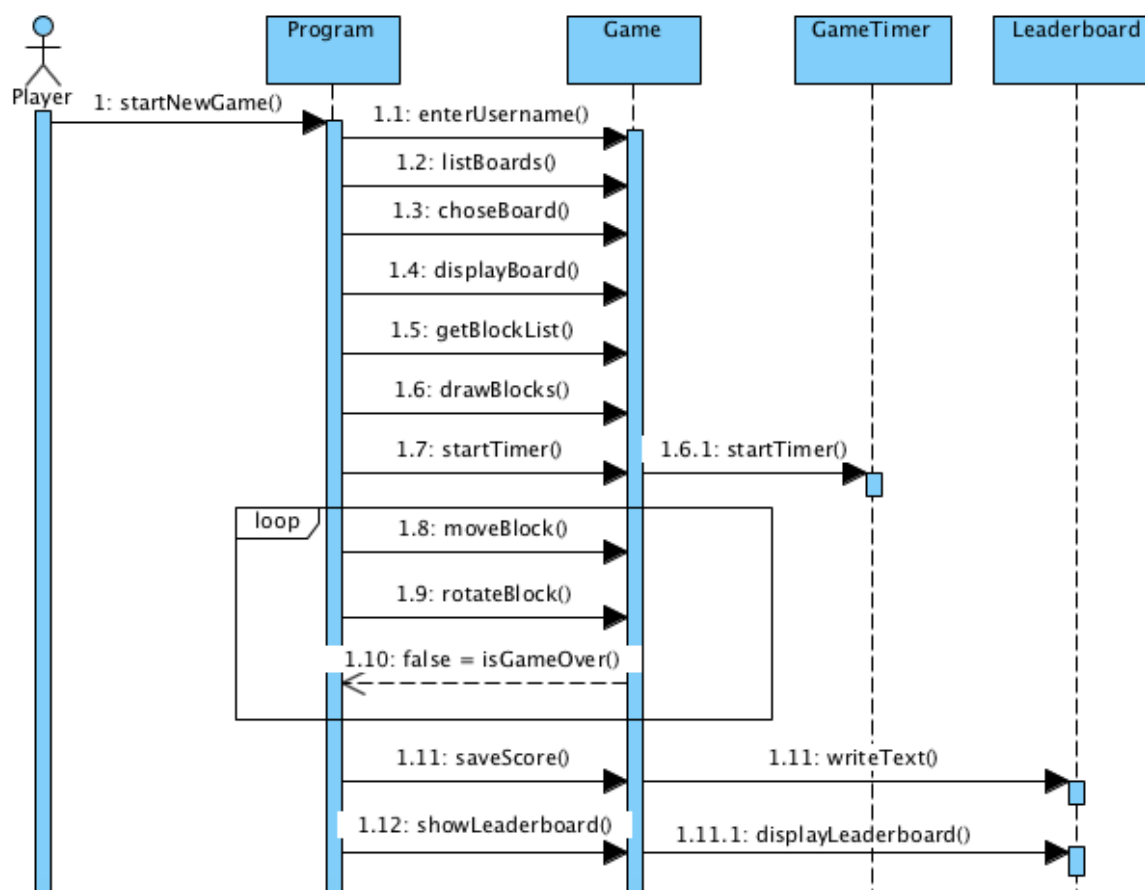
draw a new shape. After the user draws the board, this board is checked with blocks.

At the end, the program saves this board.

If the user clicks on the show leaderboard button, the program displays the leaderboard to the user and then it will send the user to main menu.

The last case is clicking on the exit button. If the user clicks on the exit button, program will shut down.

Sequence Diagram 1

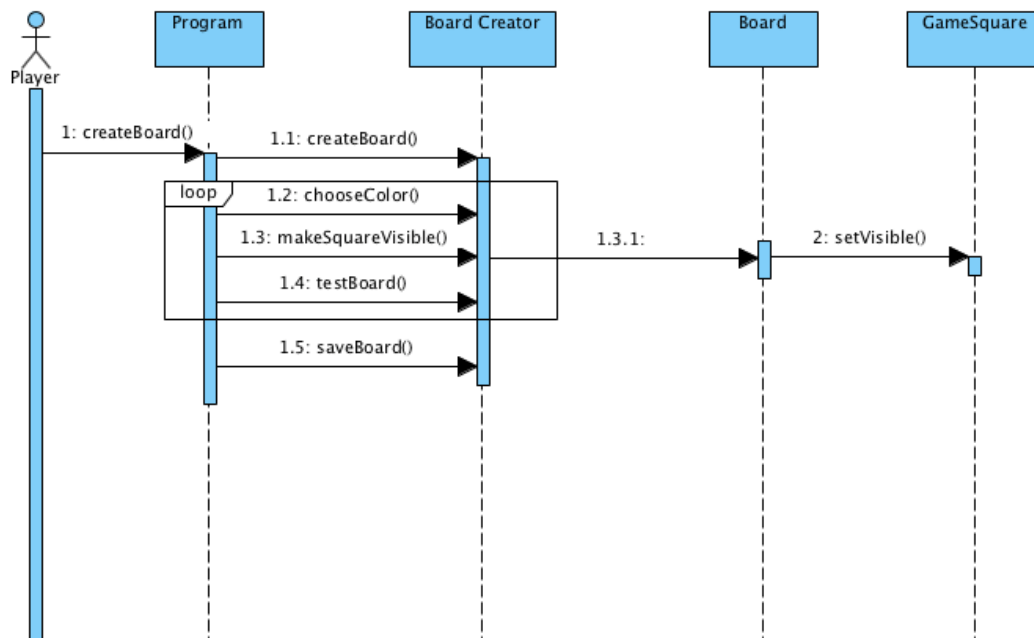


Scenario: User starts the game

The user wants to play the game so he/she presses the play game button on the main menu. Then the user is shown a list of different boards and then chooses one of these boards. The game class gets the block list of the chosen board and draws these blocks

to the block pool. And then the timer is started using the GameTimer class. After this the user can move and rotate blocks as they wish until the game is over. When the game is over the user is given an option to save their score and/or see the leaderboard.

Sequence Diagram 2

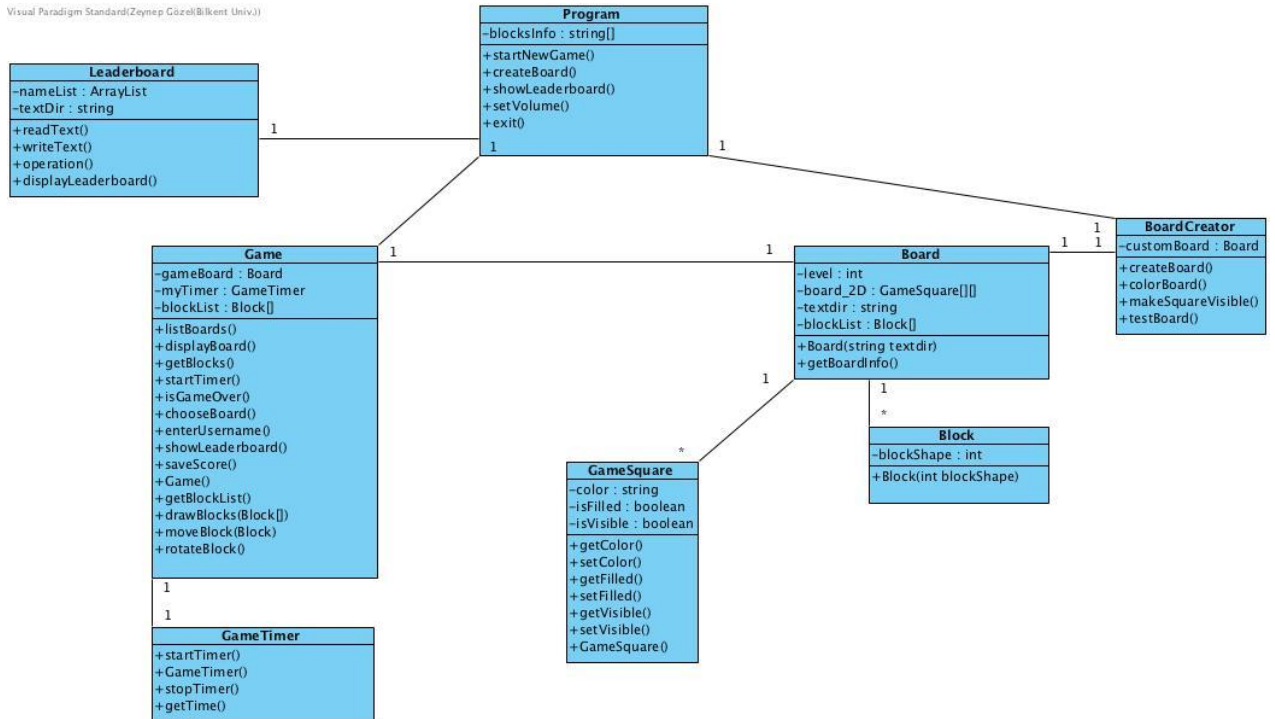


Scenario: User wants to create a custom board

The user wants to create a custom board so they press the create your board button on the main menu. Then the user is given an empty grid with game squares that they can fill and color. After the user fills the squares that they desire they can test the board by using infinitely many blocks. After the testing is complete the user can either save the board or edit the board further.

5.3. Object and class model

Visual Paradigm Standard(Zeynep Güzel/Bilkent Univ.)



Program Class

As shown above, the class model of Katamino has eight classes in total. Program class is the main class that allows the player either to start a new game, or create a board or show leaderboard at the beginning of the game. In addition, the player can also set the volume or exit the game.

Game Class

Mainly, Game Class allows the user to play the game and checks whether the game is over or not. It starts the timer when the game begins and controls the moves and rotations of the blocks.

Board Class

The Board class contains the shape of the board as a 2 dimensional game square array. It also has the block information of a board. More spesifically, it contains which blocks the board is made up of.

BoardCreator Class

BoardCreator class is used for controlling the creation of custom user boards. The player fills the squares by choosing a color and this class makes the specified squares colored for the user.

Block Class

Block class has one to many association with Board class, meaning that every board contains many blocks. We generate different shaped blocks by using this class and numerate these blocks to distinguish them (e.g. Block "1" refers to simple 1x1 block).

GameSquare Class

GameSquare Class is used to make a whole board from squares. Every square can be thought of as a pixel in a pixelart. The specific shape of each board is obtained by setting certain parts of a 15x15 square grid visible. This class allows us to understand which square can be filled or not.

GameTimer Class

GameTimer class is used to control the time of the game. The timer starts when the game screen appears and ends when the player finishes the game successfully. We use getTime() function to determine the point and ranking of the player.

Leaderboard Class

The leaderboard class reads the leaderboard from a text file found in textDir and copies the leaderboard into an arraylist called namelist. This process is done using the readText method. After the startup of the game every change to the leaderboard is done on the arraylist. Before the termination of the program the latest leaderboard is copied from the namelist to the text file using the writeText function. Every time the user wants to see the leaderboard, the displayLeaderboard function of this class is called.

5.4. User interface - navigational paths and screen mock-ups

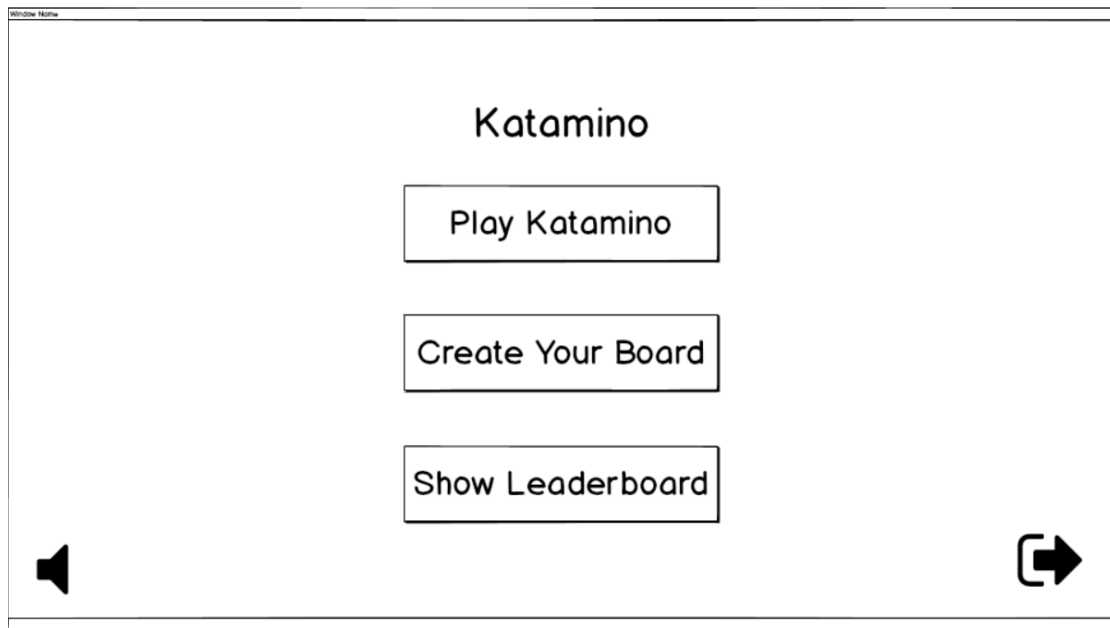


Figure 1: First screen of Katamino

Figure 1 shows the first menu screen of Katamino. There are three buttons for the user to go on. First: "Play Katamino", second: "Create Your Own Board", third: "Show LeaderBoard". The user can set the sound with the sound set button shown in bottom left. To quit the game, the user clicks on the icon shown in bottom right.

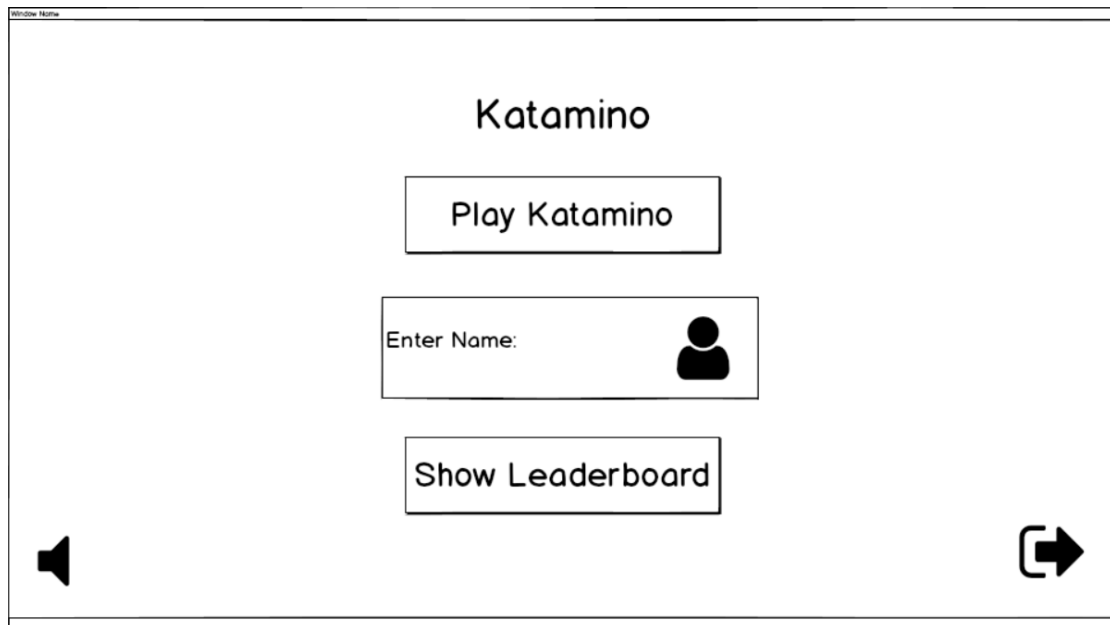


Figure 2: Name request

When the user presses “Play Katamino” button in figure 1, a text area pops up for the user to enter his/her name.

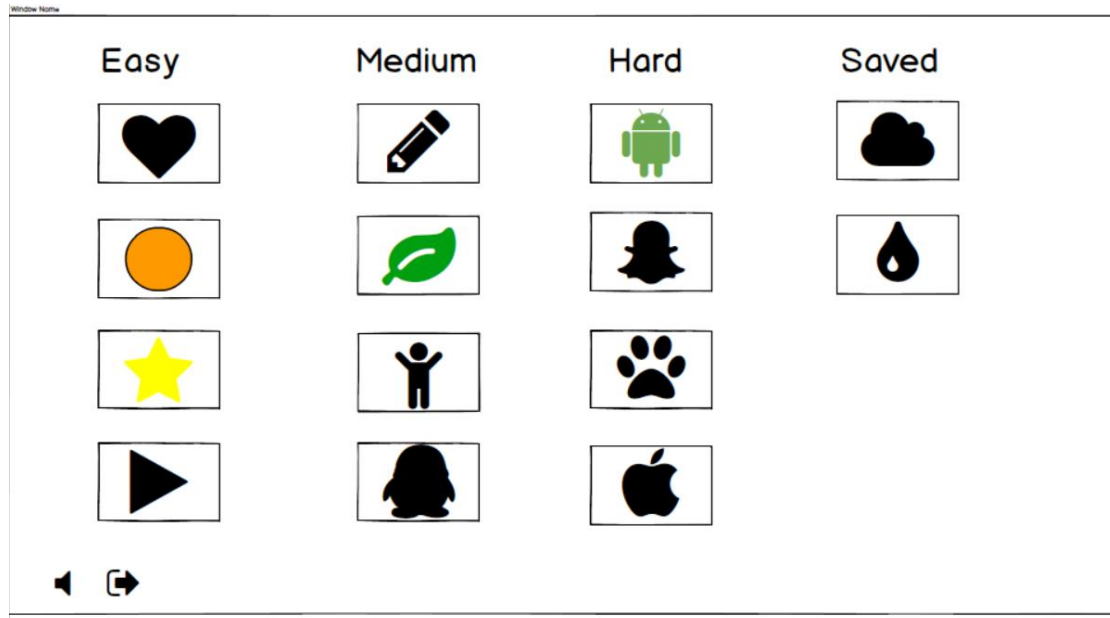


Figure 3: Level screen of Katamino

Figure 3 shows the level screen of Katamino. This screen appears after the user enters his/her name. There are easy, medium and hard levels, each showing the pattern of the Katamino board to be filled in. "Saved" shows the boards created by the user.

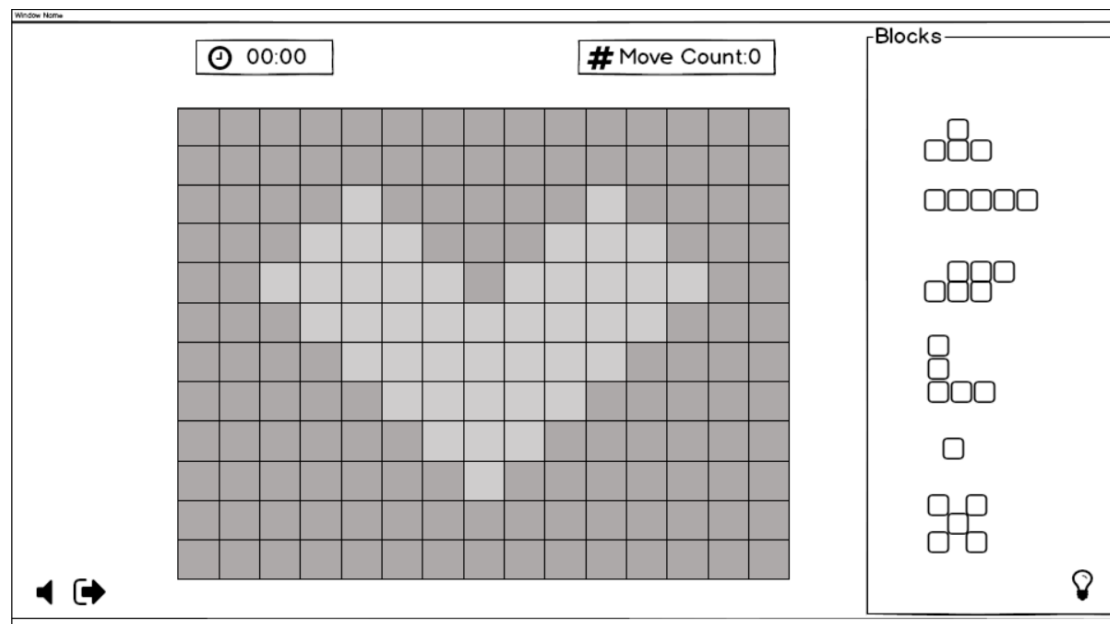


Figure 4: Beginning of the game

Figure 4 shows the very beginning of the Katamino game, after a level is chosen from the choices shown in figure 3. The timer shown in upper left begins instantly when this screen appears, meaning the game is started. At the beginning of the game move count shown in upper right is zero. Each time the user places a block to the board, it increases by one. The fair grey part of the board is in a special shape and only that part can be filled in using the blocks from the "Blocks" section in right. The bulb shown in bottom right is the hint button.

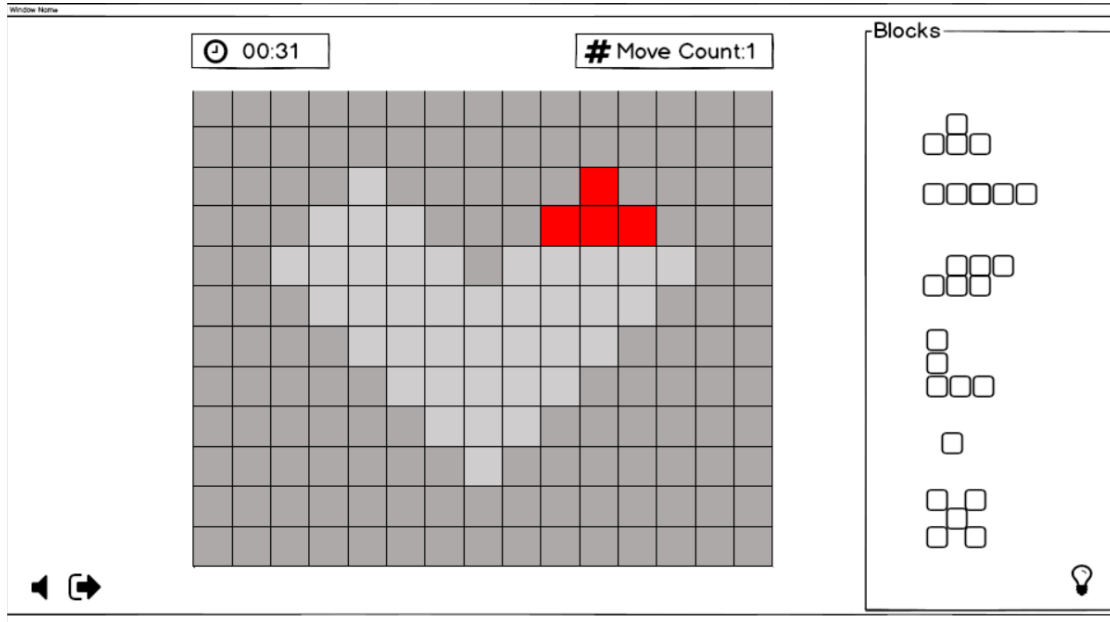


Figure 5: Playing the game

Figure 5 shows the ongoing game. The fair grey board gets colored when the user places a block on it.

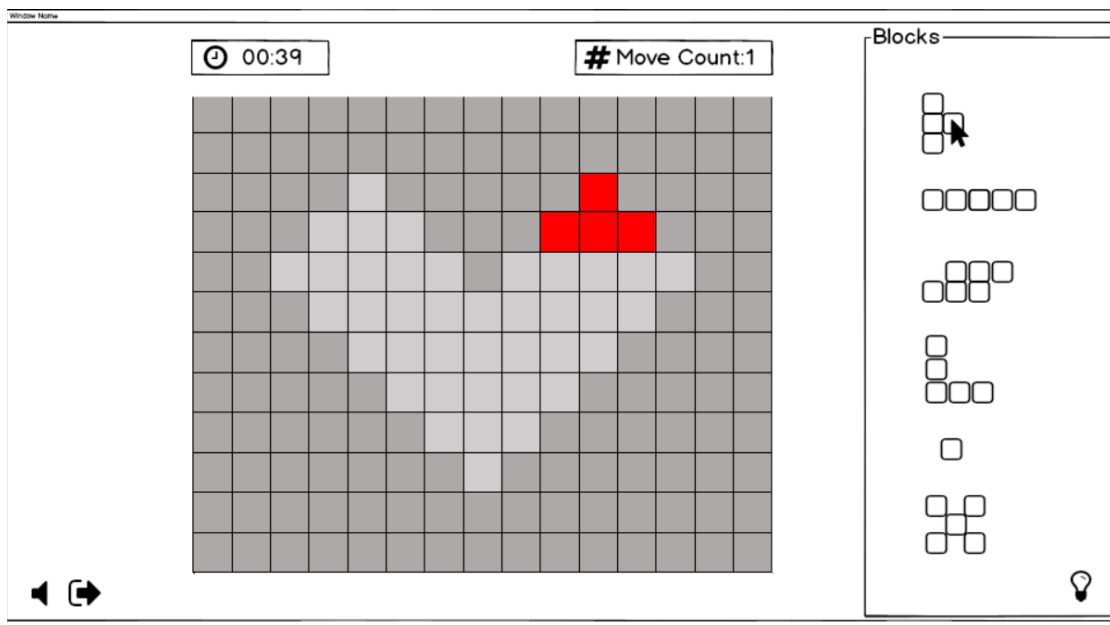


Figure 6: Rotating a block

Figure 6 shows a block rotated clockwise for 90° after the user has firstly left clicked on it to choose it and second time right clicked to rotate it.

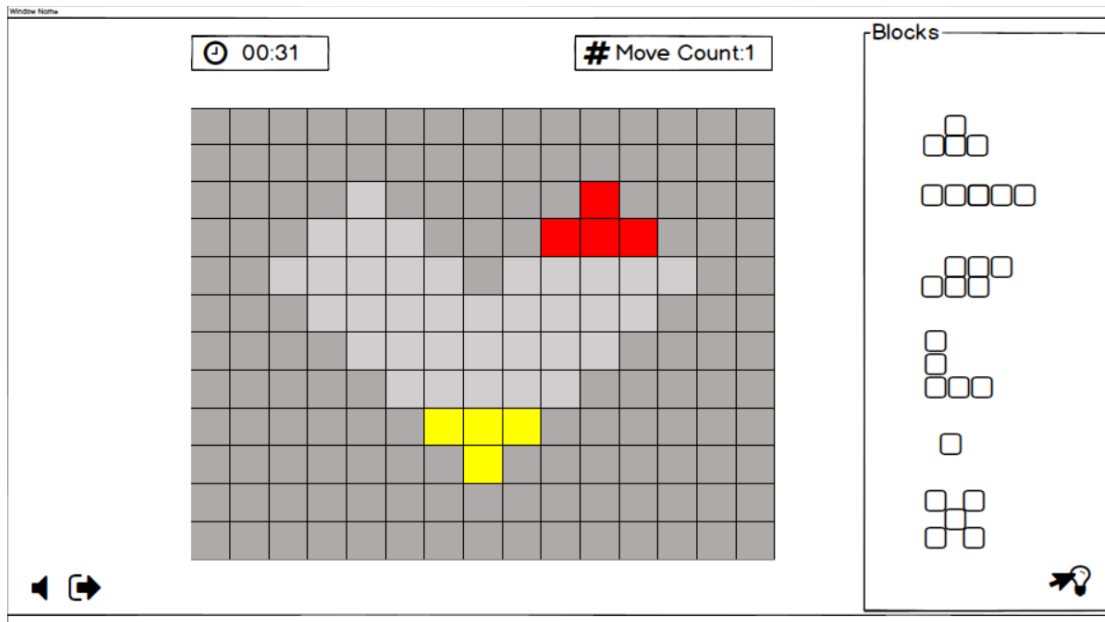


Figure 7: Highlighted hint

Figure 7 shows a part of the board highlighted as a hint after the user clicks on the bulb icon.

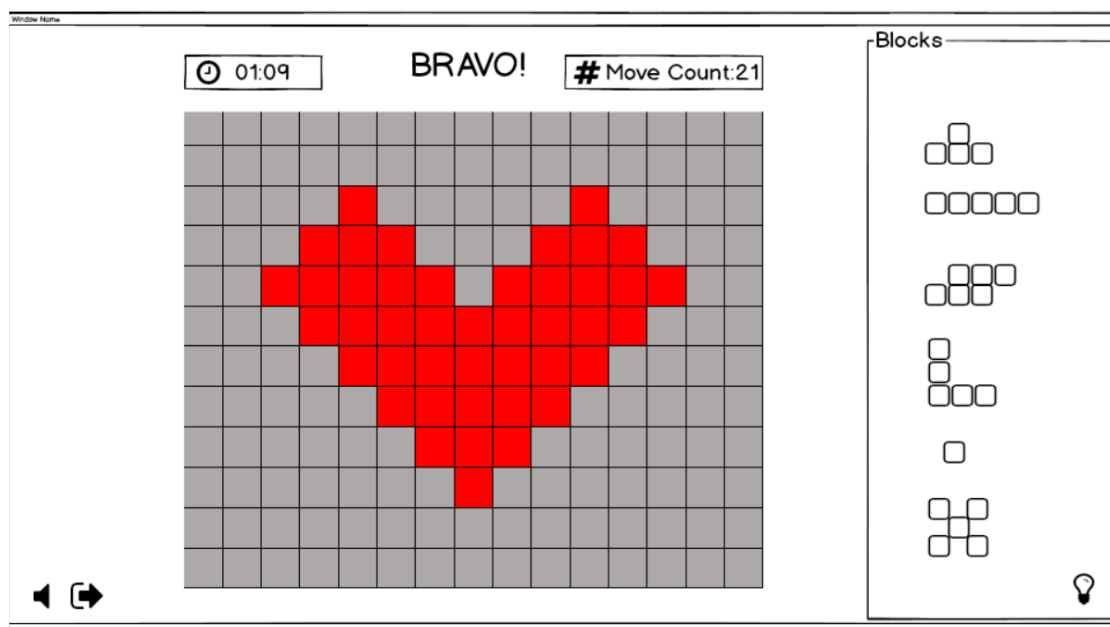


Figure 8: Completed game

Figure 8 shows the completed Katamino Game. The fair grey part of the board gets completely colored when the blocks are placed correctly and fill all the board. The timer shows the time passed until the beginning of the game. Move count shows the total move counts.

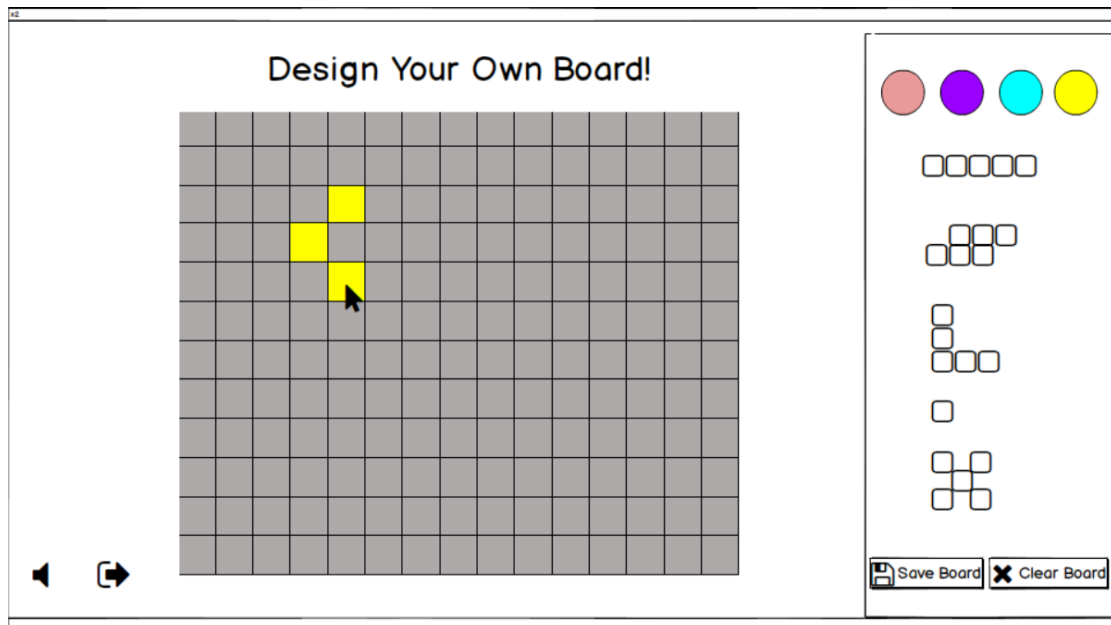


Figure 9: Player's Own Board Design

The board design screen shown in figure 9 appears when the user presses "Create Your Board" button on the first screen shown in figure 1. The user chooses a color from the right and draws his/her own board by clicking the grids one by one. The user clicks the save board button in bottom right to save the board. To start from scratch, the user clicks on "clear board" button.

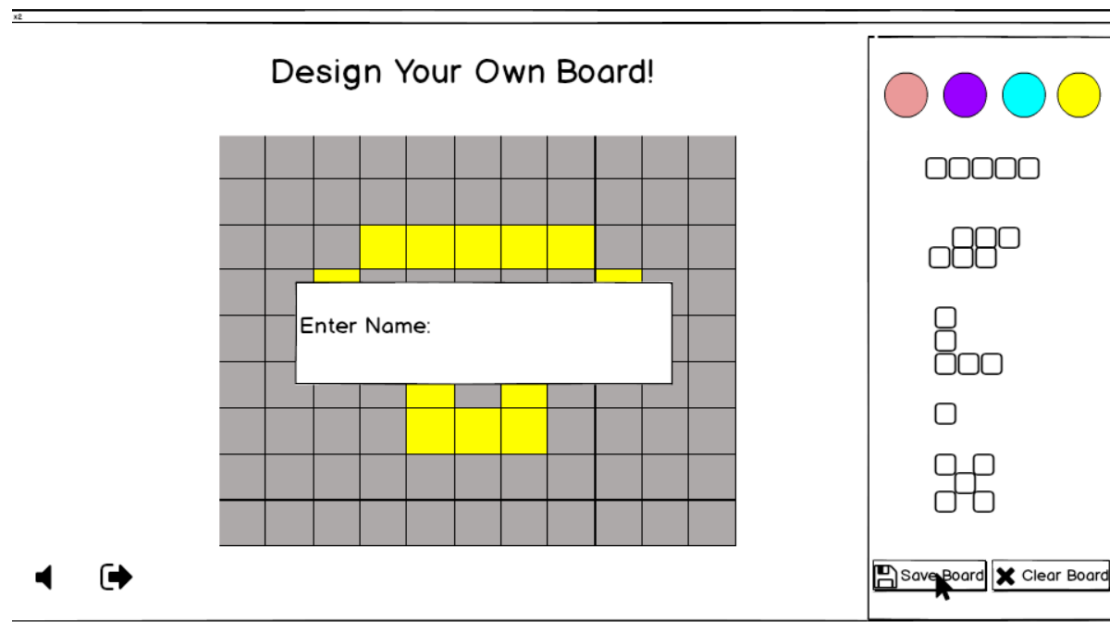


Figure 10: Entering board's name

Screen in figure 10 appears after the user creates his/her own board and clicks on the "save board" button. A text field pops up for the user to enter the board's name.

| | | |
|-----|-------|----|
| 1. | zego | 52 |
| 2. | ege | 47 |
| 3. | mmm | 40 |
| 4. | david | 34 |
| 5. | a123 | 30 |
| 6. | asd | 27 |
| 7. | sss | 23 |
| 8. | hhh | 20 |
| 9. | abc1 | 17 |
| 10. | kkk | 13 |

Figure 11: Leaderboard

Leaderboard screen shown in figure 11 appears when the user presses the “Show Leaderboard” button from the first menu screen in figure 1, or appears automatically after the game ends.

6. Glossary & References

[1] "Katamino." Game Rules, www.thegamerules.com/en/board-games/family/katamino-907-detail.

[2] Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.