Bilkent University

Department of Computer Engineering

CS319 Term Project

Katamino

Final Report

Group 1H

Ege Özcan, Mustafa Bayraktar, Simge Tabak, Yağız Mertol, Zeynep Gözel
Supervisor: Eray Tüzün

Progress Report
24.12.2018
This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfilment of the requirements of the Term Project course CS319.

Contents

1. Introduction

Current State of Our Implementation

We are currently in the final stage of our project. We have finalized our analysis and design reports, prepared our demo video and completed our game obeying the requirements stated in our requirements report. We have implemented the play game screen, create board screen and leaderboard which construct the main parts of our game. We are glad and proud with what we have right now as a project.

2. Design Changes

Although we tried to stick to our requirements and design reports, there occurred some changes in the implementation phase. We did not implement some features we included in our requirements report. However, these were not major features, therefore they did not affect the overall playing of the game. In the requirements report, we said there will be a hint button but we did not implement it. The hint button we aimed to implement needed to calculate the next move after each specific move of the user, meaning that the hint would change depending on the moves. Therefore, it became a complex feature to implement in the limited time we had. Instead of making the hint button, we added a more desirable additional feature which is importing and exporting boards. The player can import his/her friends' boards now and this made our game much more fun. In addition, in the requirements phase, we thought that rotating blocks will be a useful feature since we can rotate the blocks in real-life Katamino as well. After we implemented the game and played it ourselves, we realized that rotation is not essential to play the game because the game is more challenging without it and this makes the game more enjoyable. Other than hint and rotation, we implemented everything stated in the requirements report.

3. Lessons Learnt

The most important thing that we learnt in this course is how a project is designed before starting the implementation. We learnt that a requirements and analysis report is essential to make a design report since we should know exactly what to implement and understand the domain comprehensively before we start to make high and low level designs. If we did not understand what Katamino is in real life, we would not be able to add new features or implement existing features as a desktop game.

We learnt how to make proper use-case, sequence, object and activity diagrams which will be necessary for real-life projects as well.

We also learnt how important a system design is to work iteratively on a project. Firstly, we prepared our object design. Since our object design was a large and complex structure, it was hard to both make and see changes in our designs by only using object design. To reduce our system's complexity and see the changes we made individually easily, we decomposed our system into subsystems. We first used MVC architectural pattern to work faster and individual. After this MVC architectural pattern formed the fundamental structure of our system, we added Façade design pattern for smaller parts. Using Façade design, we formed larger structures from our objects which added flexibility for making changes. We realized that owing to this MVC design, our implementation gained speed as we expected since we worked individually on model, view and controller parts. We started our implementation based on our design but the process was not linear as we expected. During implementation, we had to go back and change object design. This experiment taught us that a project cannot proceed linearly.

Furthermore, we had not used Github extensively before this project. It was a great

experience to learn implementing a project collaboratively using Github.

4. User's Guide

4.1. System Requirements and Installation

1. Pull the project to your favourite IDE.
2. Specify the Project SDK to a Java SDK to any version above version Java 8.
3. Compile and have fun.
4. We suggest to play the game on a Windows OS and 1080p resolution.

4.2. How to Use

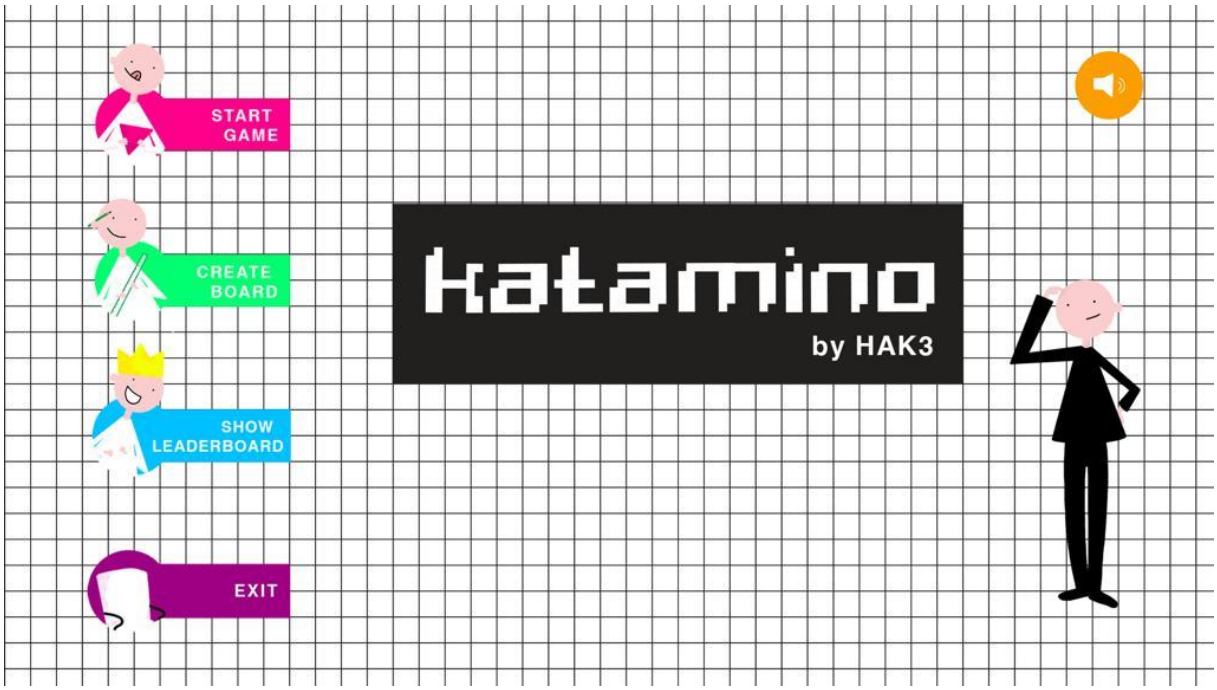

Figure 1: Menu Screen of Katamino

Figure 1 shows the first menu screen of Katamino. There are four buttons for the user to go on. First: "Start Game", second: "Create Board", third: "Show Leaderboard", four: "Exit". The user can mute or unmute the sound by clicking on the sound set button shown in top right. To quit the game, the user clicks on "Exit".

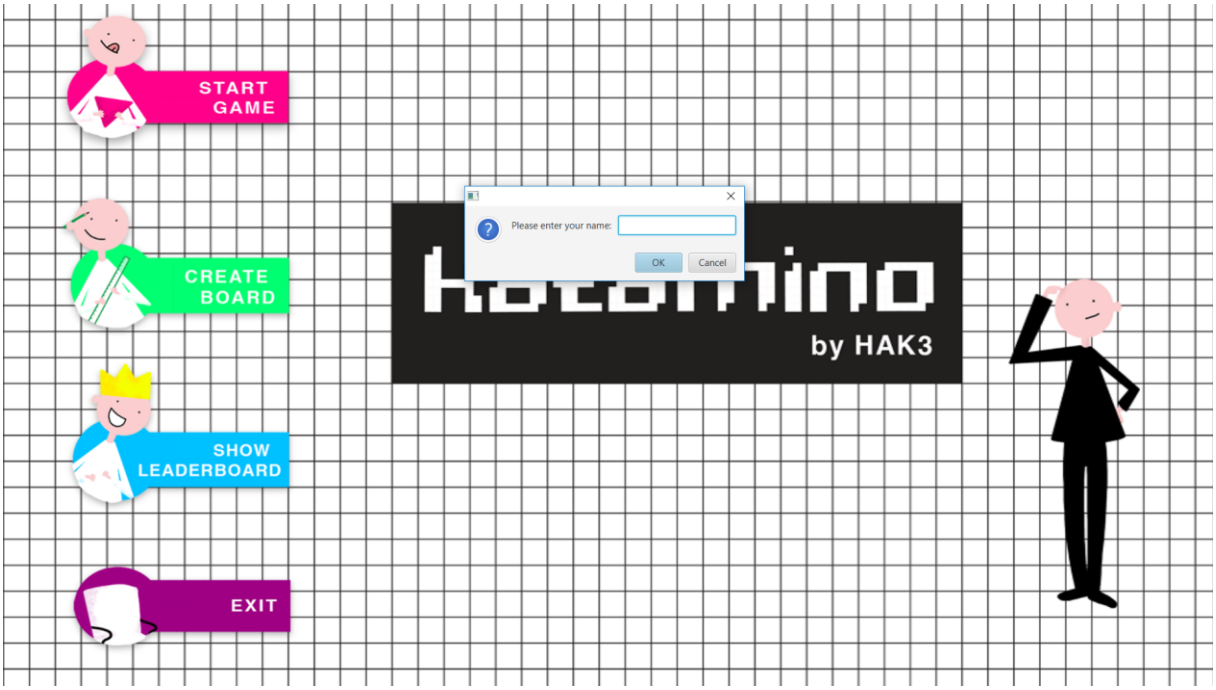

Figure 2: User name input

When the user presses "Start Game" button in figure 1, a text area pops up for the user to enter his/her name. Enter your name and press "Ok" to continue.

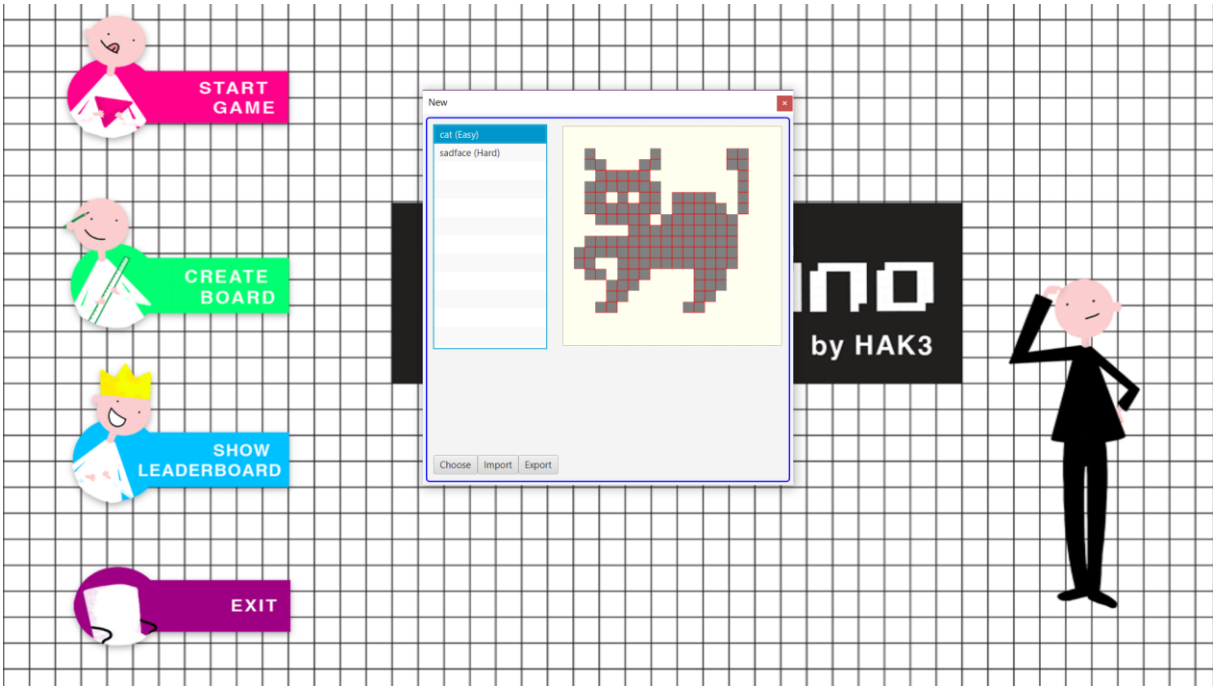

Figure 3: Board lists

When the user selects "Start Game", boards and their names with their levels appears. The user can choose a board from the list and see the board on the right before choosing it. To choose the board, press the "choose" button shown in bottom left.

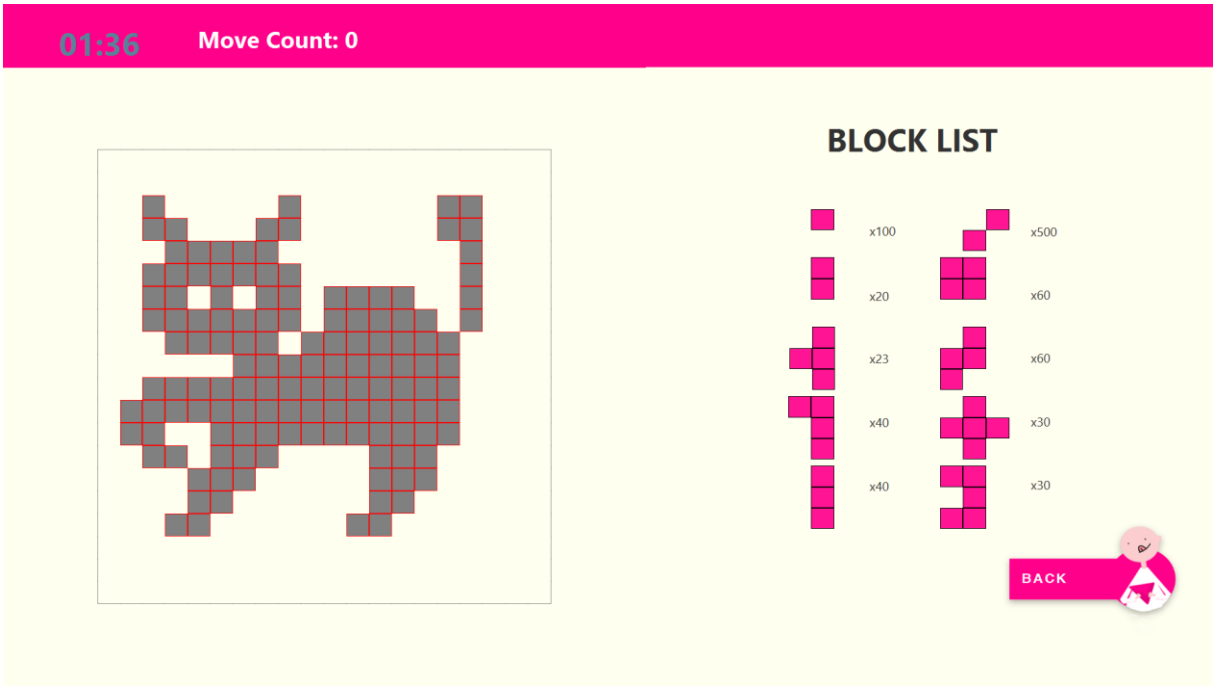

Figure 4: Play game screen

When the user chooses a board, play game screen for that board appears and the timer shown in top left begins to count down. The user loses the game if he/she cannot fill the board in the given time. The user clicks on a block under the block list and drops on the board to fill it.

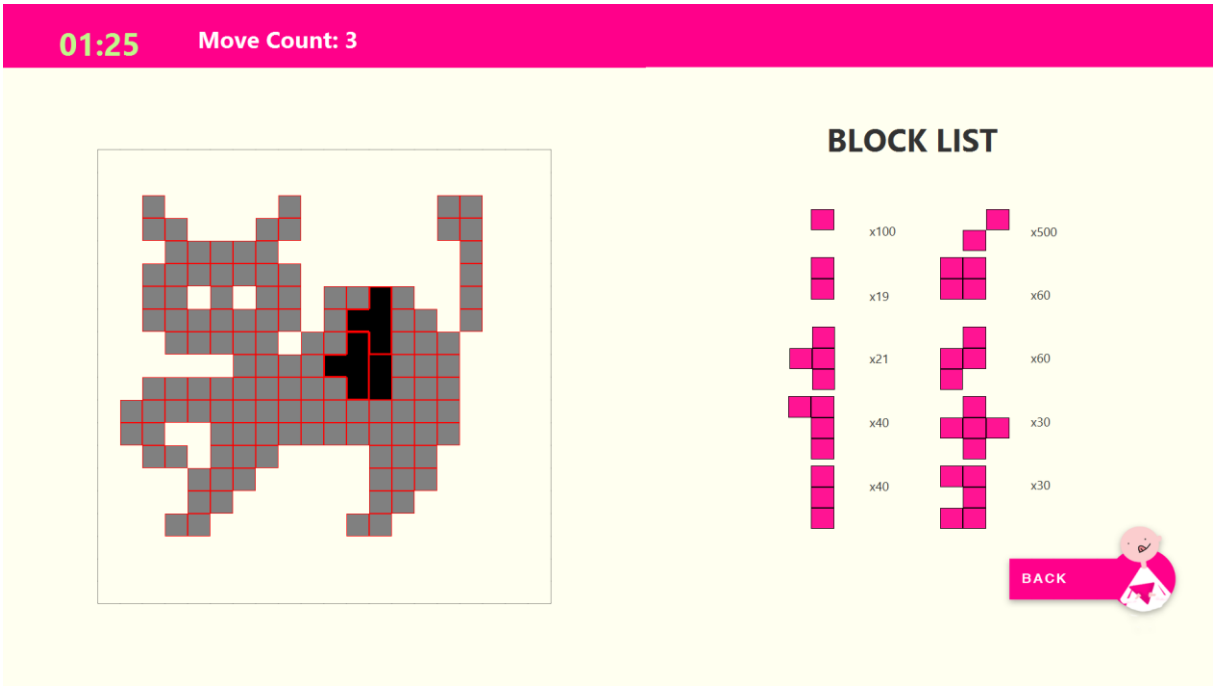

Figure 5: Playing the game

After each drop, move count increases. The real colour of the board appears after dropping the block. The block number next to the block decreases after it is placed. To take the placed block back, the user just clicks on the placed block.

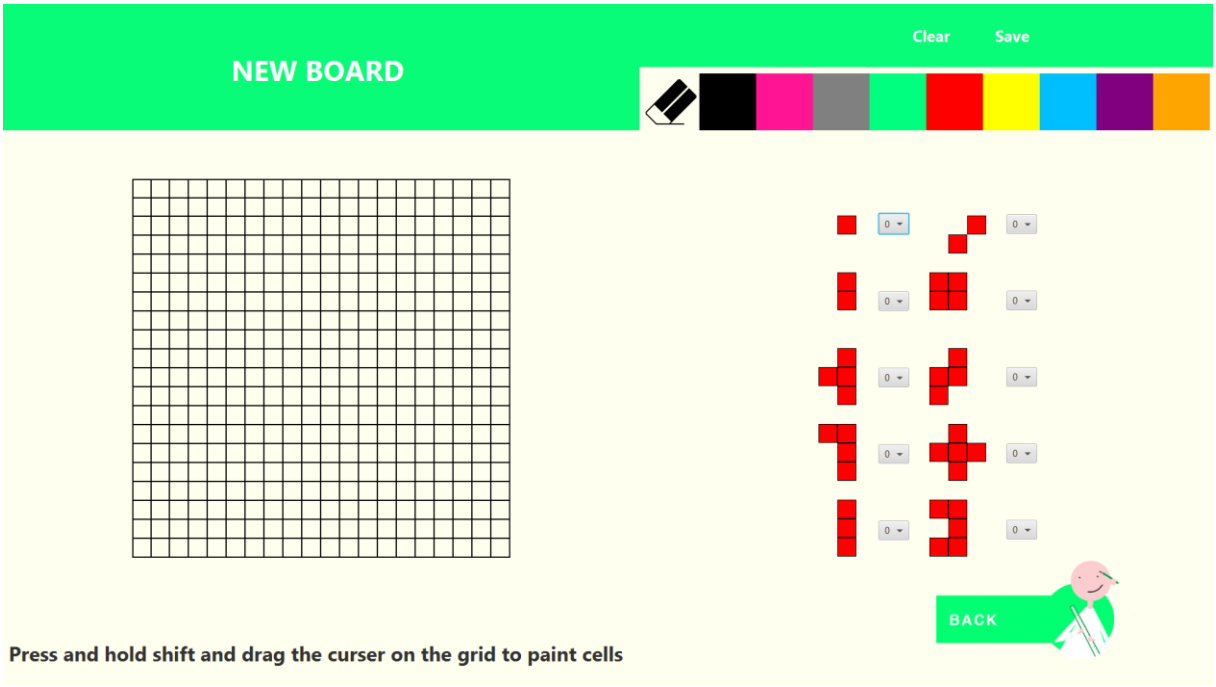

Figure 6: Create board screen

When the user presses "Create Board" button in figure 1, create board screen with an empty grid appears. The user clicks on one of the colours shown in top right and either clicks on the cells one by one, or presses and holds shifts and drags the cursor on the grid to paint cells.

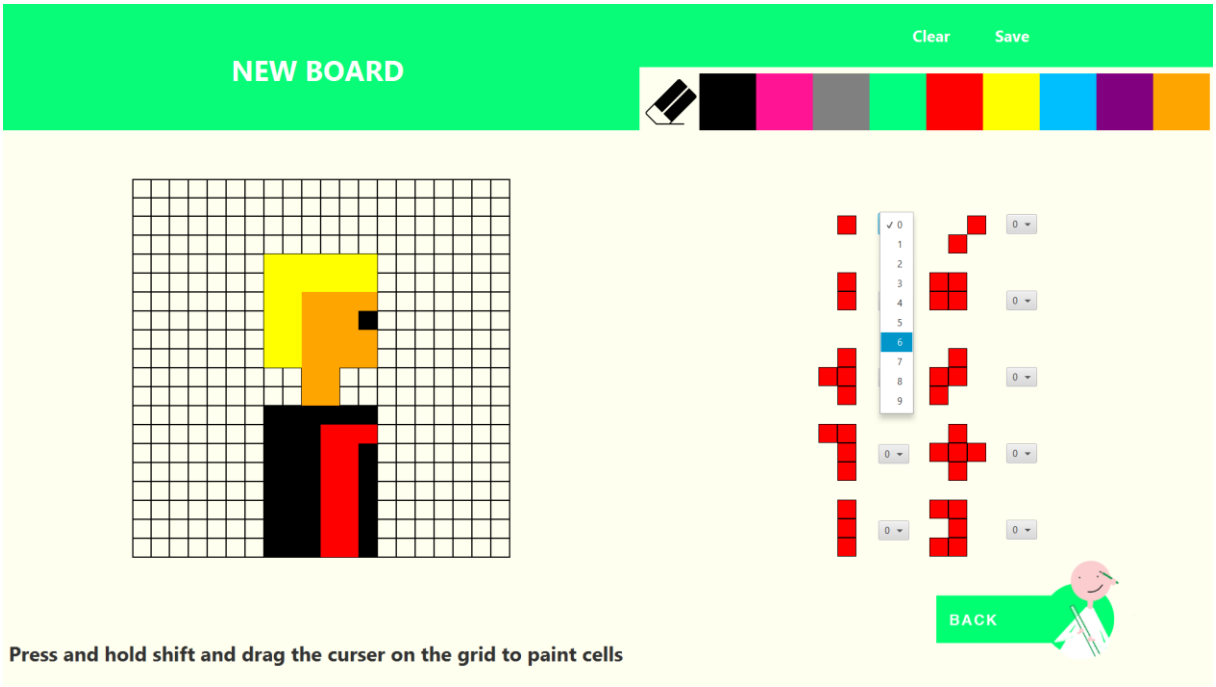

Figure 7: Create board screen with completed board

User can choose the amount of blocks to be used to fill his/her own board by using the choice boxes next to the blocks. Clear button on top right clears the whole board. Save button saves the board to the board list.

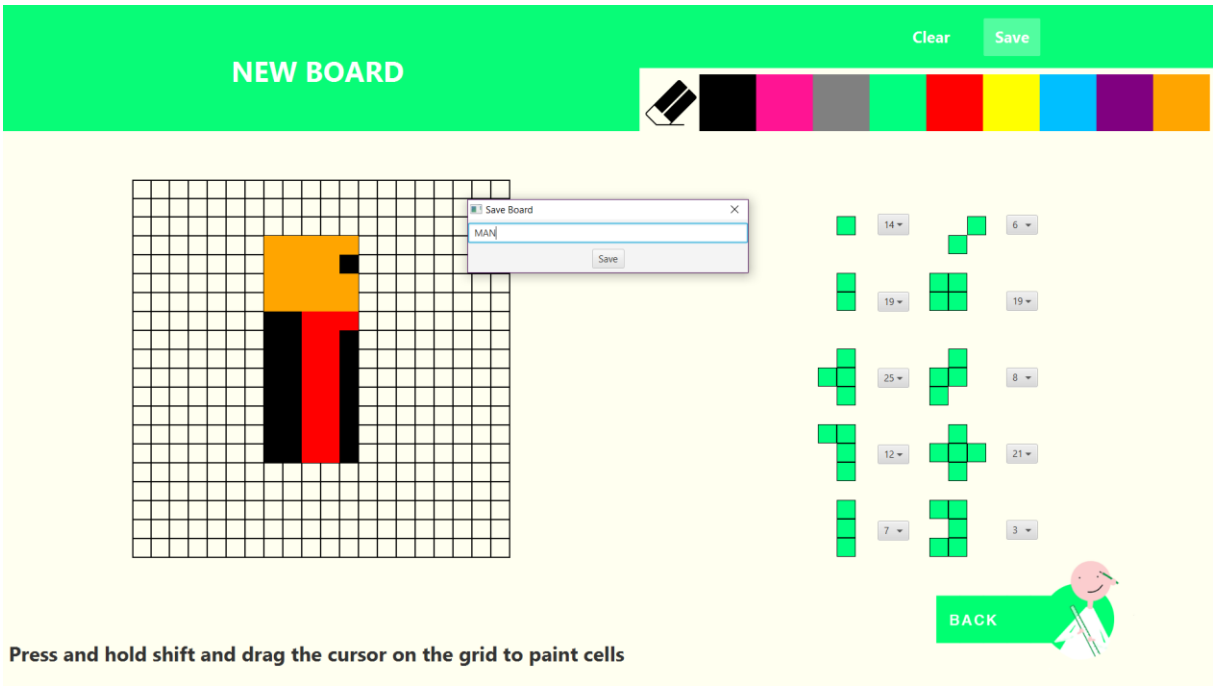

Figure 8: Save created board

After clicking the save button, a text box appears to enter the board name.

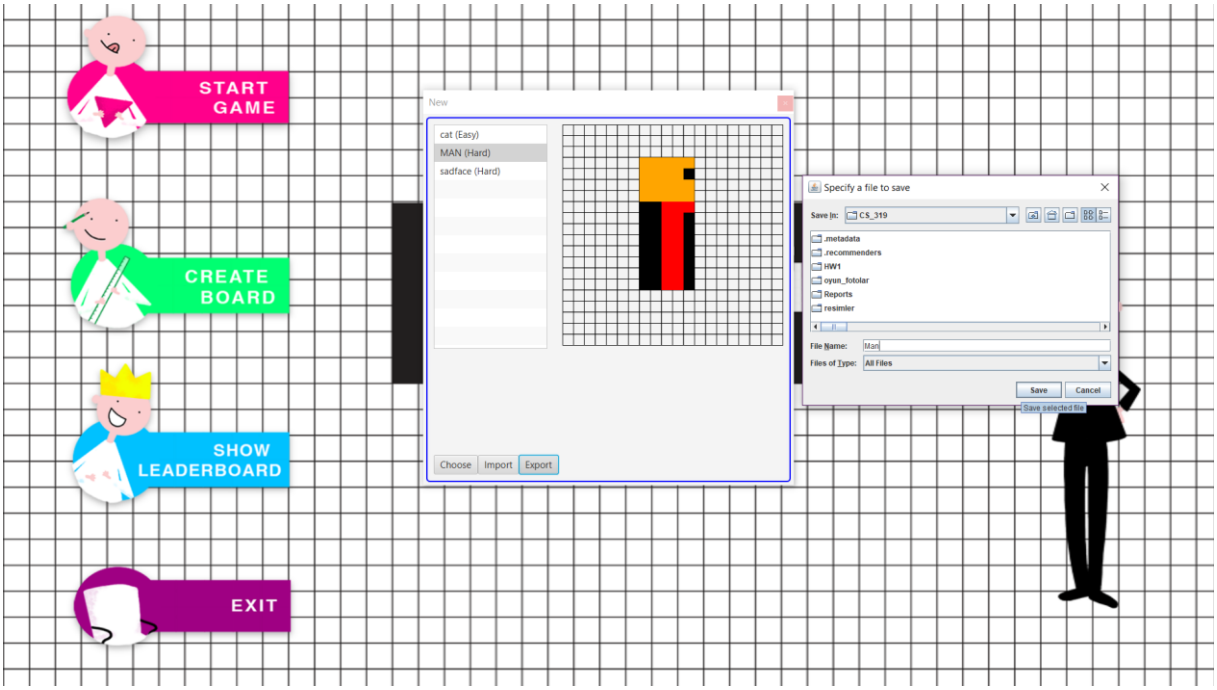

Figure 9: Export board

If the user wants to export a board for a friend, he/she chooses the board to be exported and clicks on “export” button. Then saves the board to a zip file in a specified directory. Likewise, he/she can import a board by pressing the “import” button.

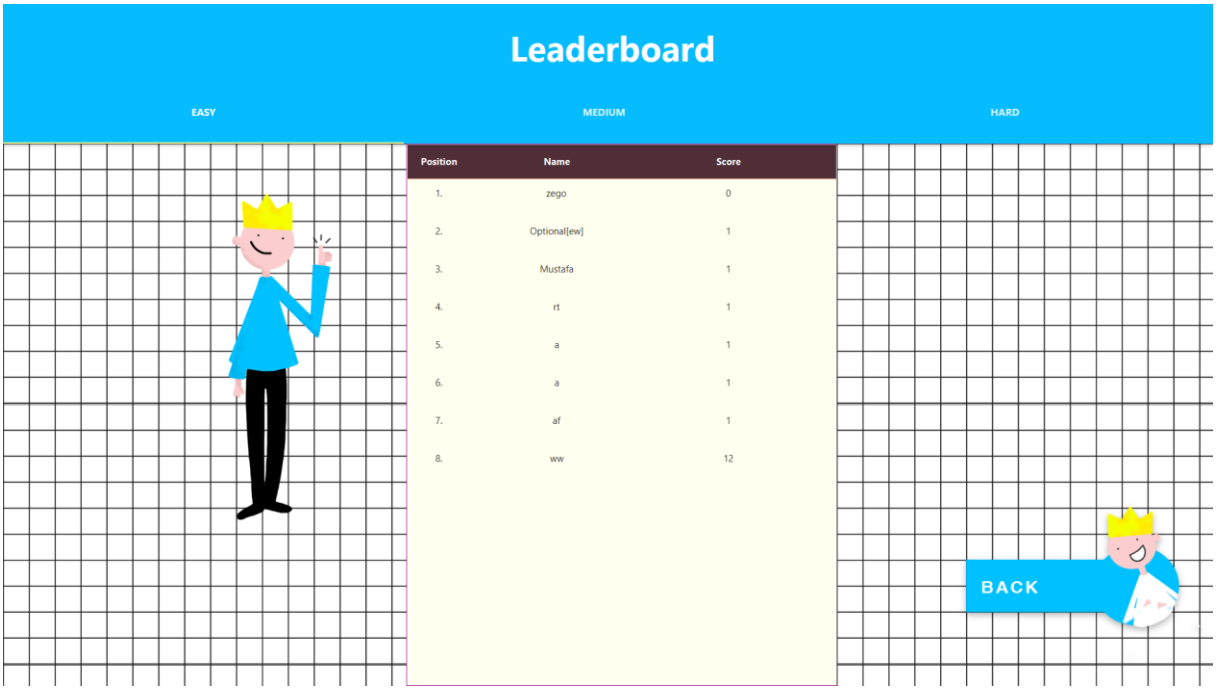

Figure 10: Leaderboard

If the user chooses "Show Leaderboard" from the main menu, leaderboard screen opens. There are three different leaderboards for easy, medium and hard levels. By clicking on "easy", "medium", and "hard" shown in top, these leaderboards can be seen.

5. Work Allocation

- Yağız worked on Controller and GameController classes and their user interface counterparts. He added the additional feature import/export board.
- Simge worked on the model classes. Block, Board, Cell, Time, User. She also helped with the design of the play game screen.
- Ege worked on createBoardController and the create board screen. He also worked on the Time class.
- Mustafa worked on the model classes. He also worked on improving the aesthetics of the game. Ege and Yağız wrote the FXML codes for the screens and Mustafa made these screens much more pleasing.
- Zeynep worked on the model classes. She also worked on many other parts of the game. When somebody needed help Zeynep helped them, so Zeynep contributed to the GameController, CreateBoardController and to the visuals of the game.
- For the requirements and analysis report:

  Simge drew the use case model. Ege drew sequence diagrams. Mustafa drew the activity diagram. Zeynep, Simge and Mustafa drew object and class. Ege and Simge drew the mock-ups and user interface part. Yağız wrote the introduction, overview, functional and non-functional requirements.
- For the design report: Ege and Yağız wrote Introduction, High Level Software Architecture and Object Design Trade-Offs. Zeynep, Simge and Mustafa wrote final object design, packages and class interfaces. Improvement summary was written with the contribution of everybody.
- For the final report: Ege and Yağız wrote design changes. Zeynep and Mustafa wrote lessons learnt. Simge wrote user's guide.