



CS 319 - Object-Oriented Software Final Report

Katamino

Group 1-G

Mert Epsileli

Yusuf Samsum

Fırat Yıldız

Burak Korkmaz

Faruk Ege Hatırnaz

Table of Content

1. Implementation Process	3
1.1 System Requirements	4
1.2 How to Use Katamino	5
2. Design Changes and Improvements	6
2.1 Flexible Game Improvement	7
2.2 Dragging and Rotation Improvement	7
2.3 Immobilizing the Block Improvement	9
2.4 Music Usage Improvement	10
2.5 Leaderboard Improvement	11
3. User's Manual	12
3.1 Start Frame	12
3.2 Settings Frame	14
3.3 Nickname Screen	14
3.4 Choose Game Mode Screen	15
3.5 Game Screen	16
3.6 Leaderboard Screen	18
4.0 Glossary & References	20

1. Implementation Process

In the implementation process, IntelliJ IDEA CE was decided to use.

Members of the group installed the IntelliJ IDE inside their computers. By the help of IntelliJ IDE, each computer has been connected to each other by the help of GitHub. Each computer connected to game repository in the GitHub and attain the power of commit changes on project. The changes made directly should not be sent another one by pulling the commits from GitHub, the changes made on project could easily be sent to members computers suddenly.

There was quite hard to distribute works at the beginning part of the project because the system has seen as inseparable. We tried to separate missions inside the group by looking at the functions decided in the design process. The team meeting were preferred instead of social media applications like Skype. We did not meet in each time with whole group, small separations inside group like 2 or 3 people create more efficient works. Also the usage of such social medias may cause disorder.

What's more, we had changed some features we decided in the design process because unpredictable problems such as the problems in the movement of the blocks forced us these changes. However, Object Oriented Programming which is the main aim of this project was tried to be sustained

during the project.

When compared to the analysis and design processes, we have changed Vertex classes according to the solutions obtained from the database. We have added time and leaderboard which increases the competition when player data is uploaded to leaderboards. In main menu screens, we have made some tweaks and added sections such as “How To Play” and language selection options.

Aside from those changes, we have given up on some features that we have decided in the design and analysis steps. Such as having a shuffle button and offering the player some hints. We did not implement those features in our final product due to solutions of the puzzles are greater than expected. It is hard to foresee many solutions that could be implemented by the user. Since some puzzles have large number of solutions and we would have to give a hint on what the user had in mind as a solution. As we would not know that particular solution beforehand, we did not implement the hint feature.

Even though we did not implement some of the features discussed on earlier product stages, the Katamino Game that we developed meets the main requirements as created in the end.

During the semester, members of the group work in different areas inside the project.

- Mert and Yusuf was responsible from class-diagrams and their

explanations in the analysis and design reports.

- Fırat was responsible from sequence diagram and its explanation.
- Burak and Fırat were responsible activity diagram and its components.
- Ege was responsible from the mock-up design, requirements, high level system architecture and version control.
- After the low level design process, Fırat and Ege were responsible from the UI design and it's implementation.
- Mert was responsible from implementation of databases in the game.
- Yusuf was responsible from game play, game bugs, game design (such as drawing blocks, board etc.)
- Burak was responsible from block movements and placing blocks true positions inside game board.
- Mert, Yusuf and Burak were responsible from the different game modes.
- Fırat was responsible from the time management.
- Mert, Fırat and Ege were responsible from leaderboard screen.

Finally, all the smaller assignments such as report explanations, presentations, small bugs are solved by distribution between the group members.

1.1 System Requirements

Katamino Game works in JAVA Virtual Machine and requires Java SDK8. Game may work any type of operating system.

PostgreSQL 10.6 must be installed in order to use database.

As library dependencies, the game needs postgresql-42.2.5.jre6.jar installed.

1.2 How to Use Katamino

- Enter the link in GitHub account:
<https://github.com/egehatirnaz/1g-katamino-CS319>
- Pull the last version of the game into the computer
- Download the appropriate PostgreSQL 10.6 for your operating system from the link
<https://www.postgresql.org/download/>
- While installing the PostgreSQL 10.6, it will require a password, enter it and do not forget (After installation, there will be a suggestion to install Stack Builder, it is not necessary.)
- After determining the password, you will need to assign the determined password into GamePlay class' *private final static password* variable
- Add postgresql-42.2.5.jre6.jar as a path into the project. It can be found in "libs" folder
- Read the expressions in the Readme file to see the features of the game
- Run the application by executing UI_Object class inside "UIManagement" file
- Push the "Start" button
- Enter a nickname (or sign in if the username exists)
- Select the mode
- Wait a short time for setting up of databases

- Start to play and enjoy your game!

2. Design Changes and Improvements

At start of the design process, all blocks which were used in the game were drawn in Block Maker class and it was saved by hand for all solutions. However it is fixed in the final iteration. The game started to draw and save blocks inside Vertex class. Also solutions were being kept inside the GameAdmin class however they started to keep inside SolutionDataBase which means databases are specified for different purposes. What's more, movement of blocks has been developed. All the blocks may turn 90 degree in each click and make mirror movement with the right click. Blocks can also specified in a line inside the game frame and they are qualified to find their true places when they approach the right places. In the start, there were a hint system to make the game for unqualified solvers more fun but it is removed because of many solutions which may not predicted by the developers. So the classes related with this has also been removed. Leaderboard is add to increase the competition. The main and sub frames turn into better shaped and easier to use. In order to make the game more user-friendly, "How To Play" section has been add. Dynamic and Challenge which are talk about in other sections of the report has been add to game in order to user experience different features of game and make more fun. Time property added to game to rank players and for them to compete each other. Music has been add to the game and buttons are also soundy. Finally, nickname properties are add to the game which means players should play the game with valid nickname because their time performance has been saved in each play. By the help of this players may compete to each other.

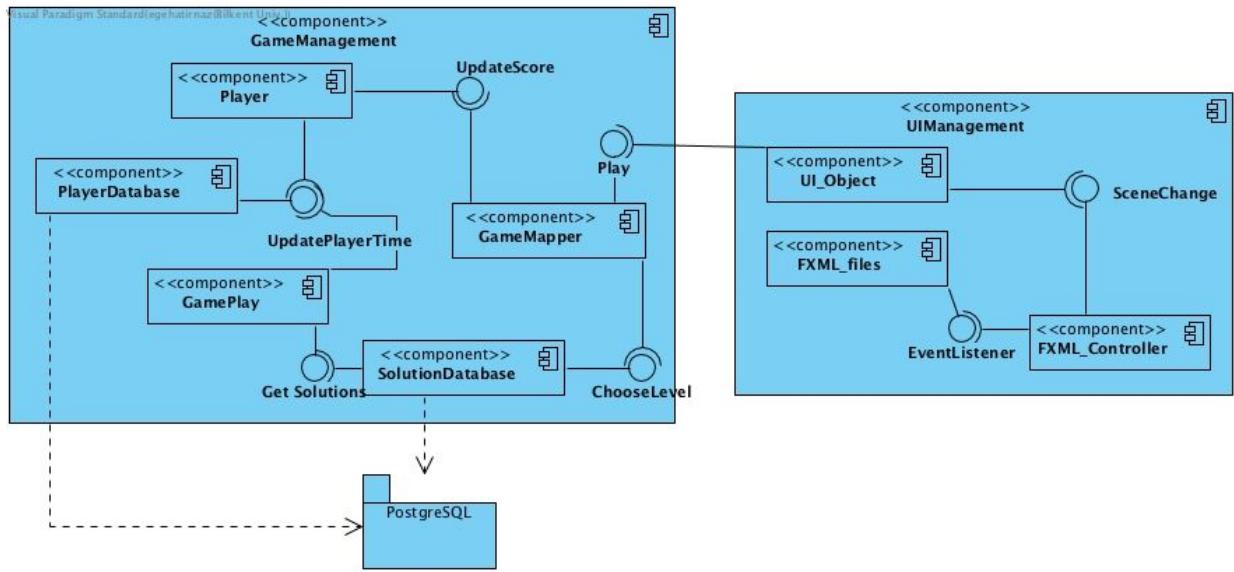


Figure 1) The System Diagram

2.1 Flexible Game Improvement

At the beginning, we promised to implement the game which shapes according to user's playing style for the very first (design or analysis) report. However, we could not managed to do it. Instead of it, we added an increasing board size profile in the game. That offers to players a game which can be played in the changing sizes. The game frame is also flexible for the changing decisions of players. Players have chance to change the position of the block after it puts as it is seen in below. The brown stick shown in Figure 2 which indicates the level, moves when a user passes on another level such as in Figure 4.

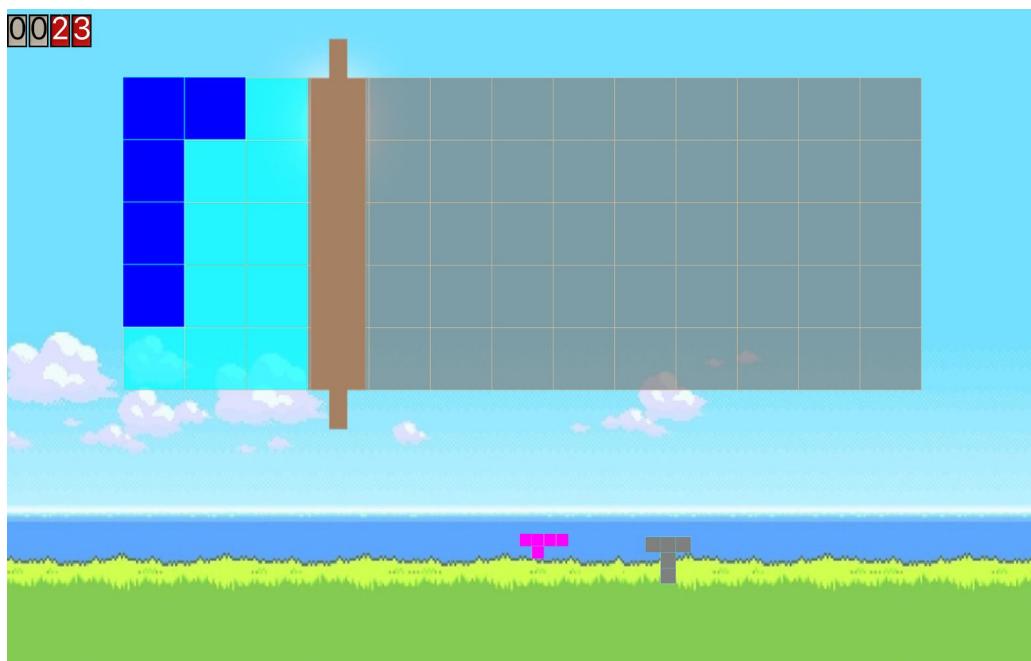


Figure 2) Flexibility in Main Screen

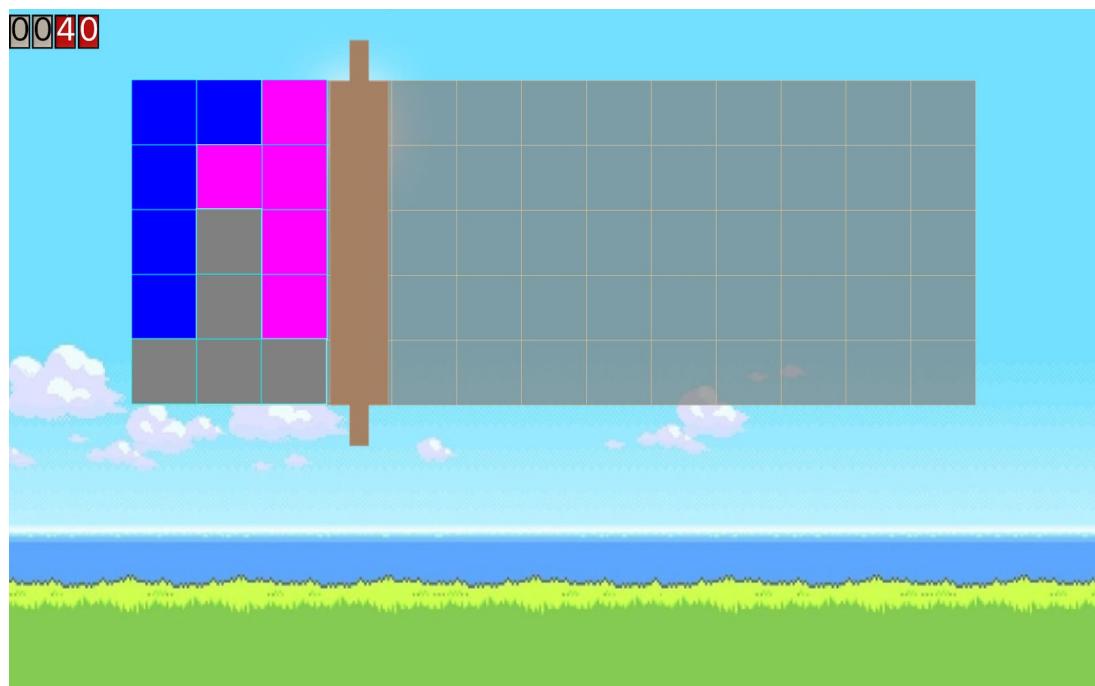


Figure 3) End of a level

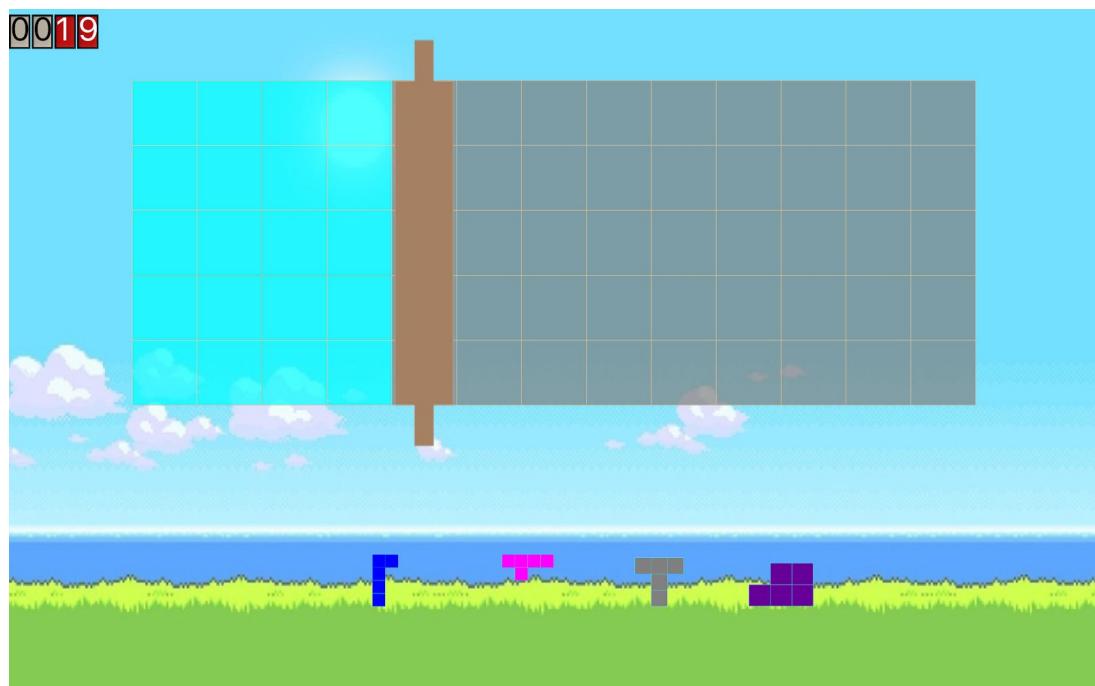


Figure 4) Movement of the Level Stick

2.2 Dragging and Rotation Improvement

In the design stage, the drag and rotation with the blocks has not been promised. However, we managed to implement them by the help of the mouse events in Java. In Java, it is possible to change the track of the position of a block or an image. So we use the positions of the blocks and the position of the board to implement the logic of the drag, drop and rotations of the blocks on the board accordingly. Normally, the refresh blocks feature have been promised but we did not managed to implement it because the blocks become inactive after they add into the block list. The blocks were not able to immobilize the blocks in a game frame but we managed to do it in the final version. The actions are shown in Figure 5 and 6 are about the rotation of a block and reflection of a block in the x axis. In the game, players have ability to rotate the block 90 degree with double click, to change the position of the block in x direction like a mirror effect with right click and drag with mouse.

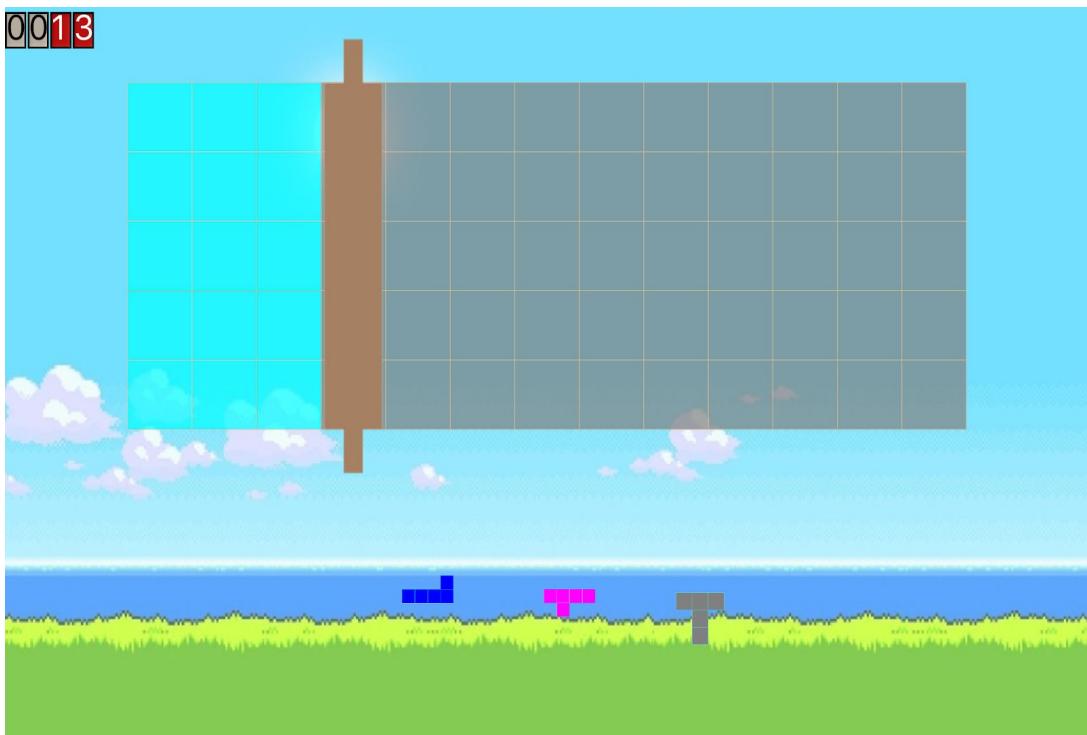


Figure 5) Rotation of the blue block according to Figure 2

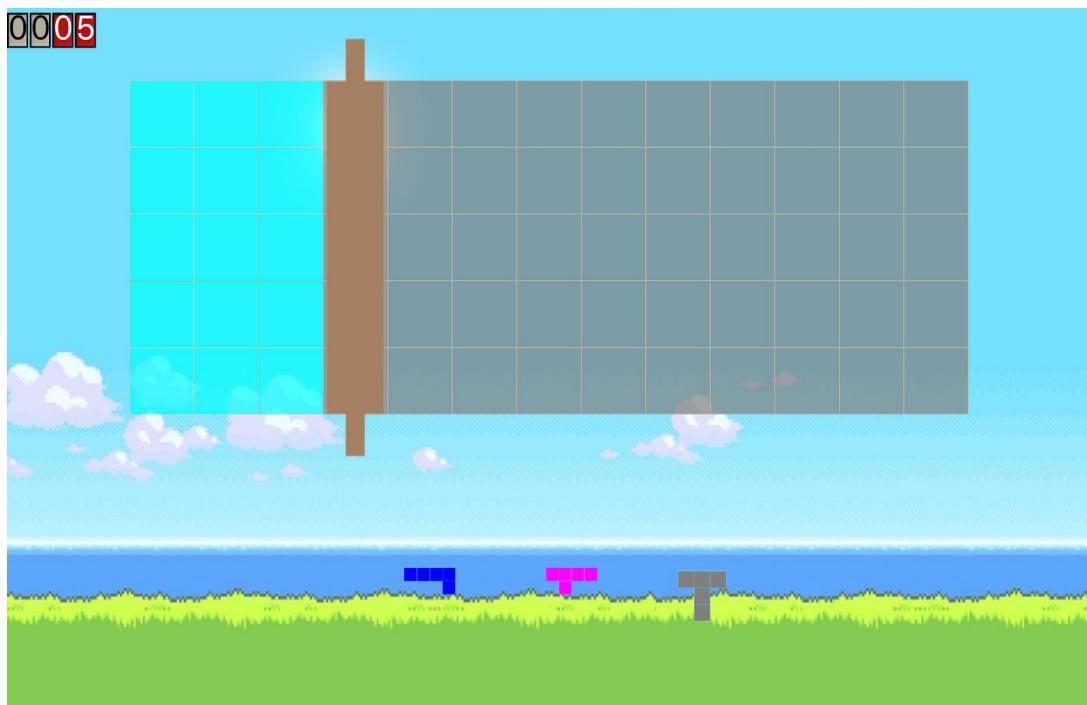


Figure 6) Reflection of the blue block according to Figure 5

2.3 Immobilizing the Block Improvement

The block design has been implemented to immobilize all pieces in the final design. Blocks are drawn in the Block Mapper class, saved inside a document and used in the game with immobilizing feature. Blocks can find their appropriate place and stick there. By the help of the data which says all the pieces of the board is filled, the game can be finished. As it is shown in Figure 5, pieces can return theris first places if they are placed to inactive part of the game board or above of a filled square.

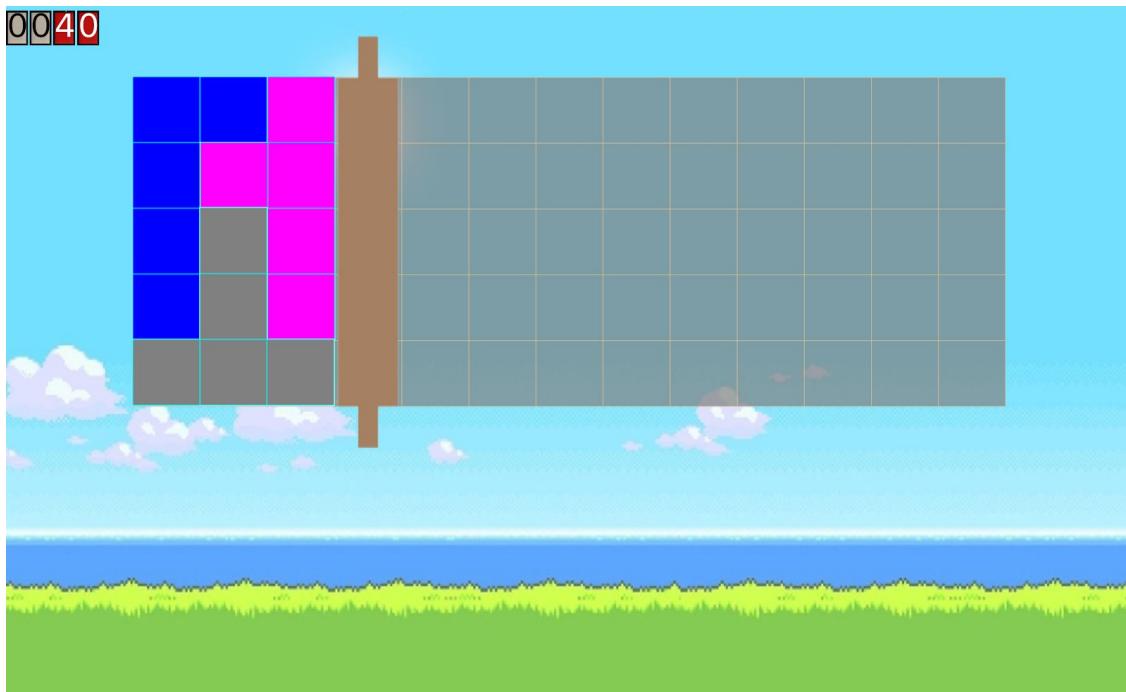


Figure 7) Immobilizing in Main Screen

2.4 Music Usage Improvement

At the beginning of the design report, we decided to manage the sound preferences for adjusting the music level or changing the genre of the music on the settings part of our game. Later we decided to put it on the start screen to make it simpler for the user. By this decision user will not spend the time on the settings screen. Just a click on the music turn on/off button will handle the music preferences. We decided to use our official game music on the whole game to give the atmosphere of the game to user. So we settled on to cancel the changing musics in the game. Because we found it useless for the users. A user can easily select its music for the game in its favourable music application such as Spotify, Apple Music, Soundcloud... with turn off the music of the game.

In *Figure 8* the music preference can be seen under the red arrow.

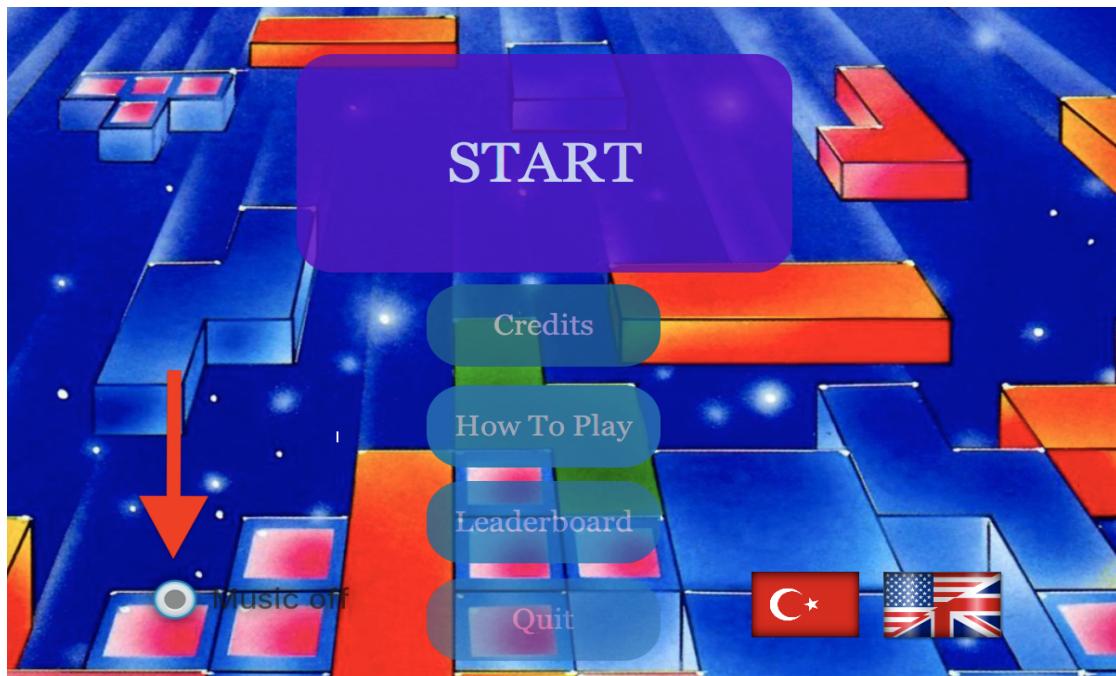


Figure 8) Turning music on-off

2.5 Leaderboard Improvement

As it is promised, we added a leaderboard screen when the game finished for the last demo. It was not available in the first demo because the name control system was finalized after the first demo. We hold the Leaderboard on the database that we created, and it updates automatically with the information came from the Gameplay. The scores and the valid names that being kept in the system, will pop up a leaderboard of the players at the end of the game. The Leaderboard Screen can be seen as below as *Figure 9*. It shows the top five scores in the game . Moreover it can be visible at the main menu such as “Leaderboard” option in *Figure 10* and “Sıralamalar” option in *Figure 11*.

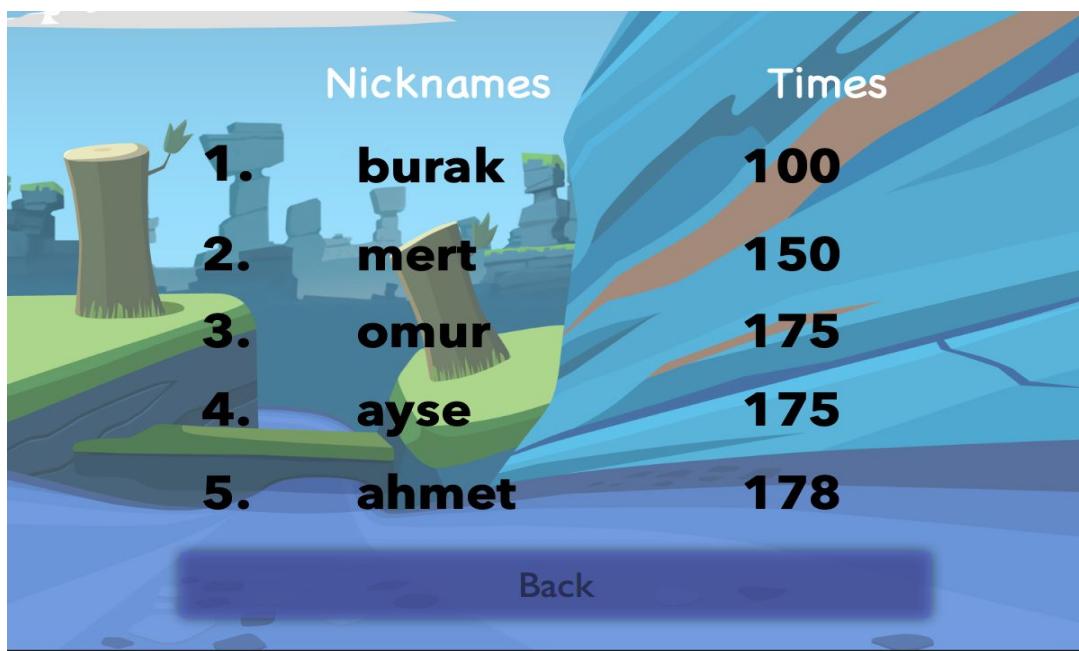


Figure 9) Leaderboard Screen

3. User’s Manual

3.1 Start Frame

The first screen which is seen by users firstly is the start frame. The start frame includes the options “Start”, “Credits”, “How to Play”, “Leaderboard”, “Quit”, Music on/off option, additionally it includes a language option such as Turkish, and English . User may choose either of them. Game starts with the “Start” button.

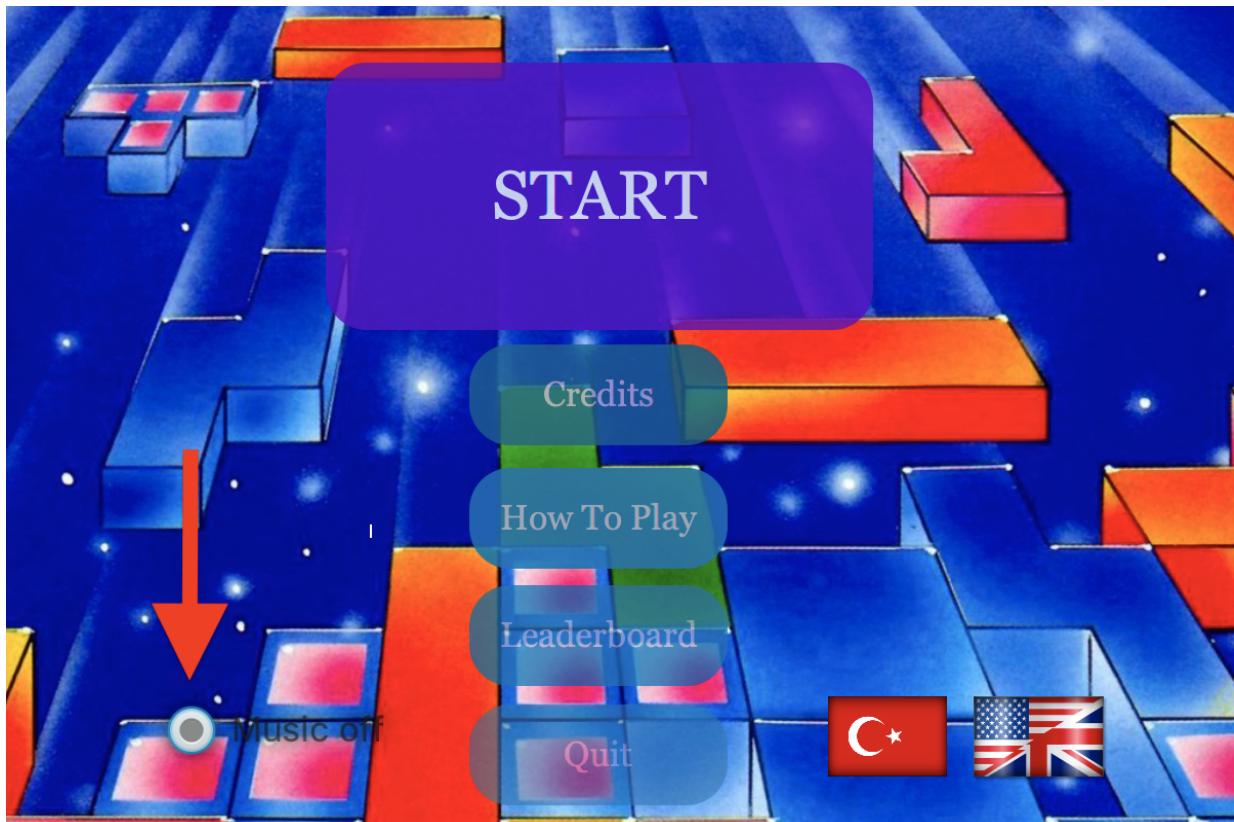


Figure 10) Start Screen



Figure 11) Start Screen in Turkish

3.2 Nickname Screen

The Nickname Screen will get a name from the user and will check whether it is taken. If it is taken, user needs to proceed with the “Sign In” option. Else, game starts with the new nickname.

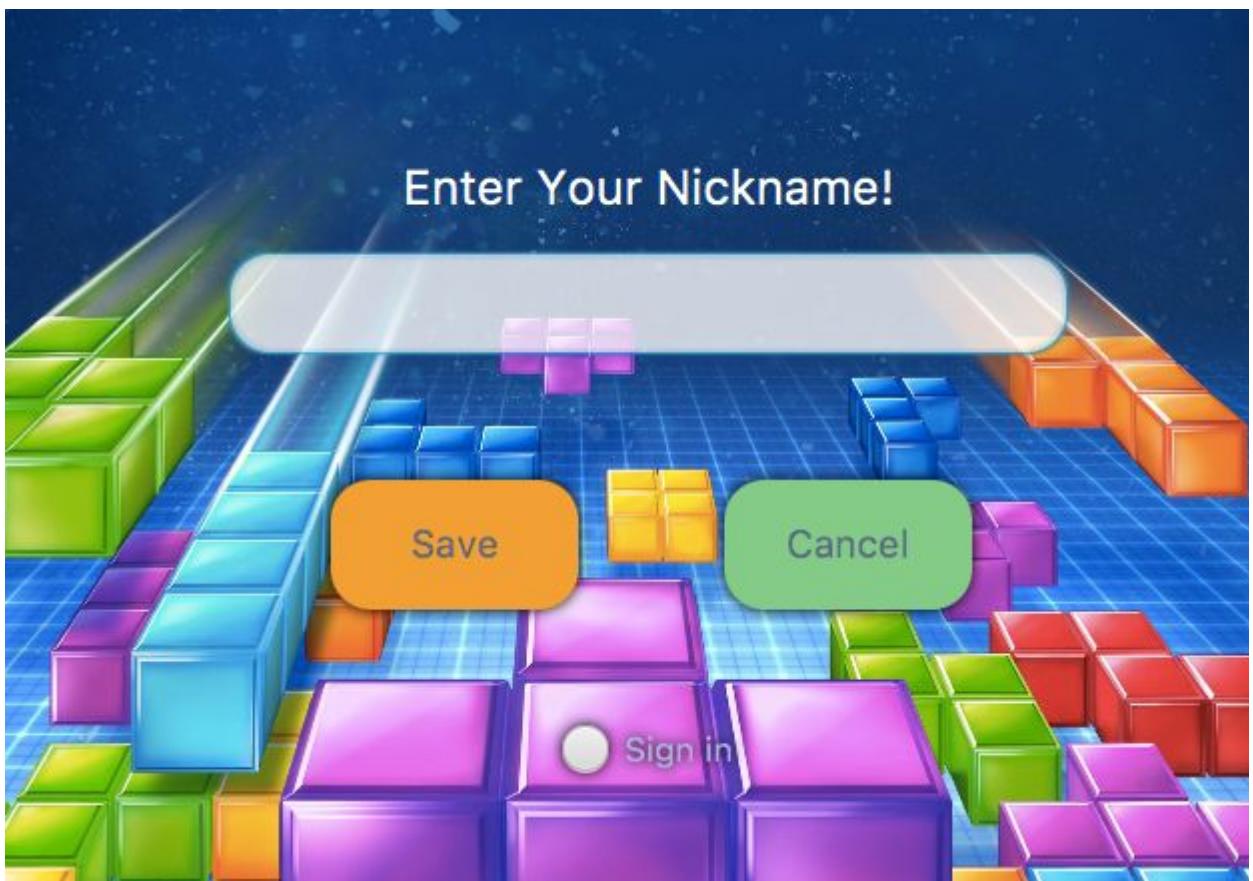


Figure 12) Nickname Screen

3.3 Choose Game Mode Screen

This screen is for user to choose the game mode he wants to play

and a link between the game frame and nickname choose class.



Figure 13) Game Mode Screen

3.4 Game Screen

For the Game Screen, a new window opens: A small game frame and the area where blocks stay are inside the window. The hint button is used when the user can not find the solution. Blocks in the frame are available for dragging and game will be played dragging these blocks to the board. Game frame will be updated if the block is put in it's right place. The user also has the chance to quit game by closing the screen. As shown in the Figure 10, the Game Screen is quite simple and easy to play.

Player will just drag these components into the blue area. Shaded area is forbidden for players. It is open in increasing level

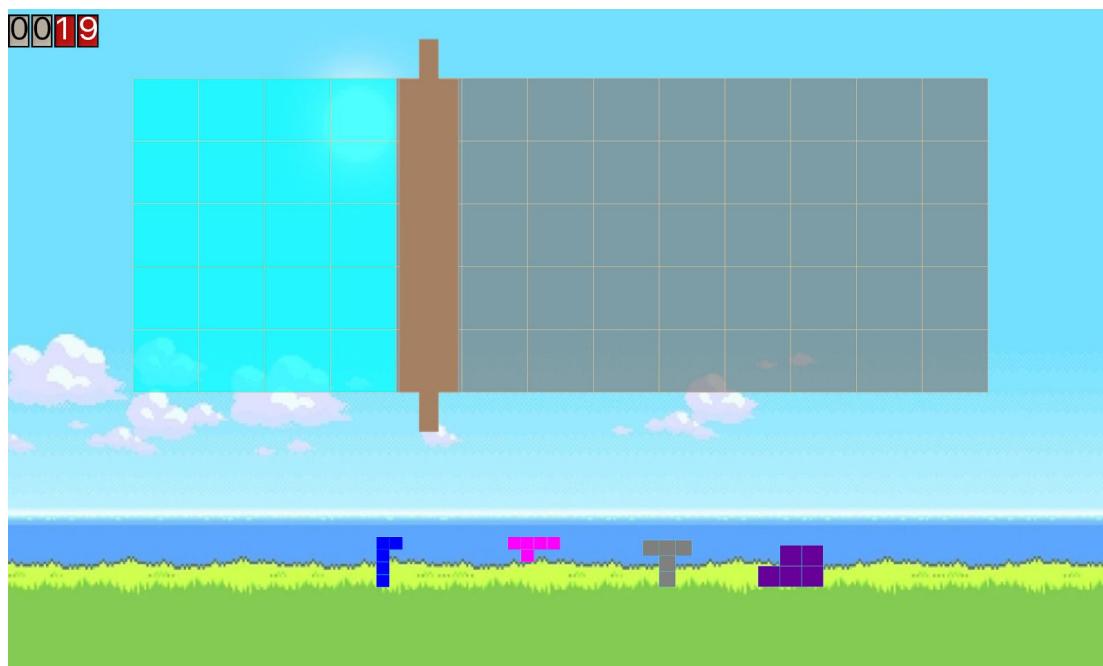


Figure 14) Game Screen

3.5 Leaderboard Screen

The leaderboard screen is visible when the game finished. It is displayed automatically at the end of the game. It gets the user data from the Player Database and shows the top 5 players.



Figure 15) Leaderboard Screen

4.0 Glossary & References

1. “JavaFX Rotate ImageView,” *Stack Overflow*. [Online]. Available: <https://stackoverflow.com/questions/34166627/javafx-rotate-imageview>. [Accessed: Nov. 05, 2018].
2. J. P. Llosa, “Java JDBC PostgreSQL Connection Example,” *Examples Java Code Geeks*, Jun. 20, 2018. [Online]. Available: <https://examples.javacodegeeks.com/core-java/sql/java-jdbc-postgresql-connection-example/>. [Accessed: Dec. 05, 2018].
3. “Java Development Kit,” *Wikipedia*, Oct. 16, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Java_Development_Kit. [Accessed: Nov. 20, 2018].