MEMORYMAZE

By: Otto Oksanen, Eetu Soronen, Hasan Safdari & Samu Oksala

METROPOLIA

OHJELMISTOTUOTANTOPROJEKTI 2

Contents

1. What is the project about?

Our project is a memory game designed to help the players improve their mental capabilities. The game is made to be fun & challenging at different levels.

2. How to play the game & requirements

Currently we have two different methods of accessing the game.

2.1 Requirements

Java 17+

IDE to build the project. (We recommend IntelliJ Idea, only required if using the method found on chapter 2.3)

2.2 Installer

We currently have an installer available for our game, that can be found in GitHub. This way, to play the game all you must do is run the installer and follow the steps.

2.3 Compiling via GitHub

If you choose this way, we recommend using IntelliJ to play the game, since every one of us (developers) uses it and it's known to work on it. In order to play the game, first clone the project.

Note that the database server credentials are not stored inside the GitHub repository. You need to create a config.properties file in the resources folder with correct values yourself to access the database features. Precompiled Jar-file and MSI-installer come with the credentials included.

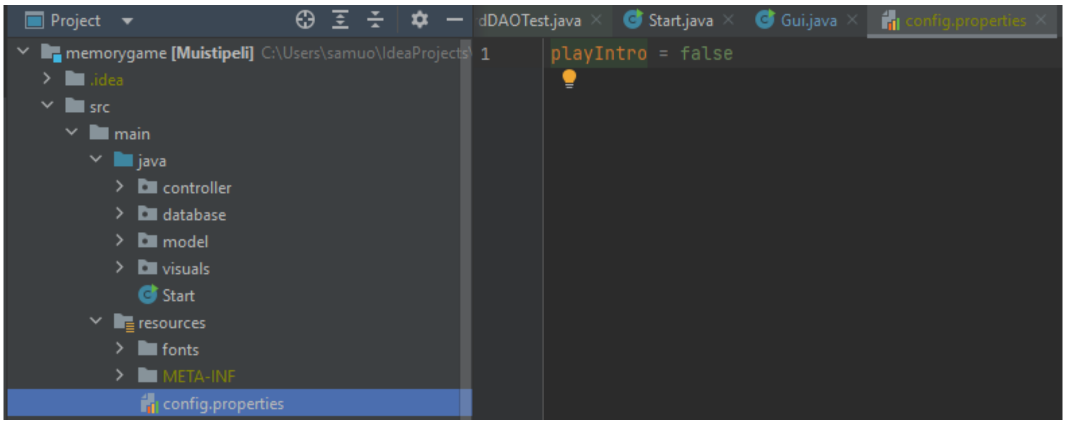
After that, build the project and the game should be fully functional.

3. Description of the application & the user interface

3.1 Functionality

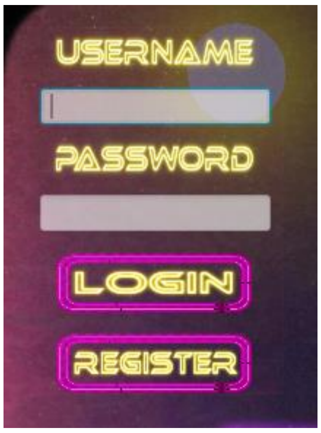
3.1.1 Intro

There is an intro when you launch the game. Intro can be enable/disabled by editing the "config.properties" file inside the projects resources directory. If intro is wanted, set it as "playIntro = true", if not set it as "playIntro = false".



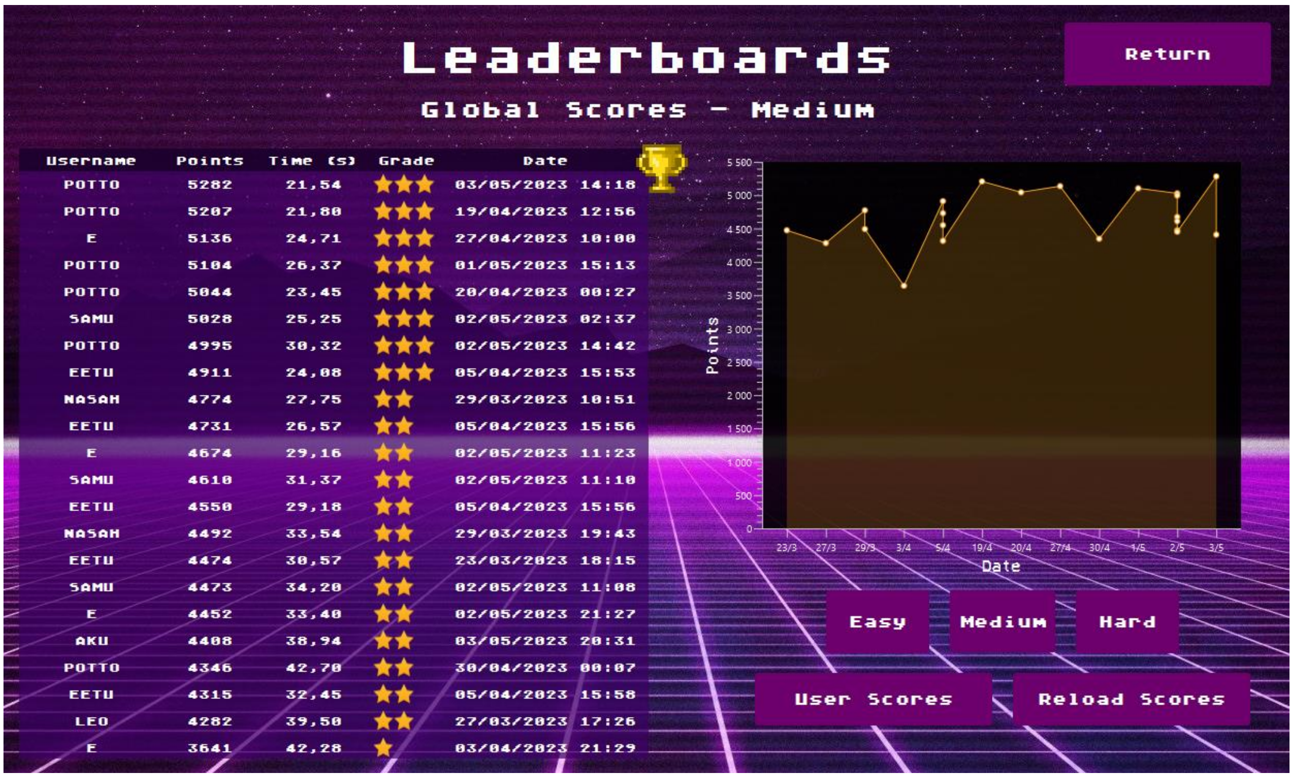
```
playIntro = false
```

3.1.2 Authentication



This is where the user can create a user and login.

3.1.3 Leaderboards & Charts

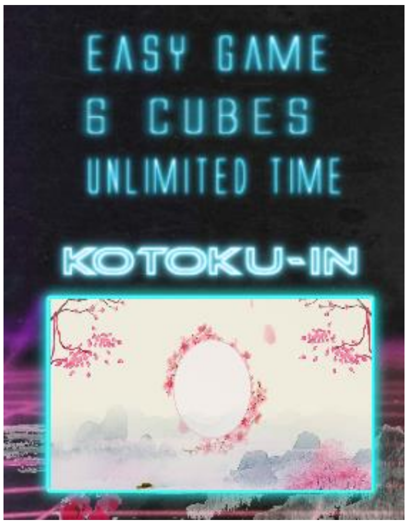


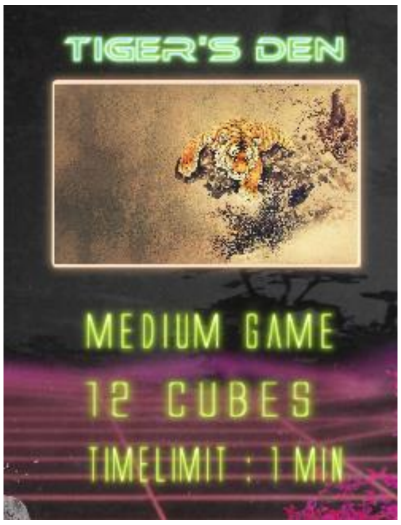
Username	Points	Time (s)	Grade	Date
POTTO	5282	21,54	★★★	03/05/2023 14:18
POTTO	5207	21,80	★★★	19/04/2023 12:56
E	5136	24,71	★★★	27/04/2023 10:00
POTTO	5104	26,37	★★★	01/05/2023 15:13
POTTO	5044	23,45	★★★	20/04/2023 00:27
SAMU	5028	25,25	★★★	02/05/2023 02:37
POTTO	4995	30,32	★★★	02/05/2023 14:42
EETU	4911	24,08	★★★	05/04/2023 15:53
NASAH	4774	27,75	★★	29/03/2023 10:51
EETU	4731	26,57	★★	05/04/2023 15:56
E	4674	29,16	★★	02/05/2023 11:23
SAMU	4610	31,37	★★	02/05/2023 11:10
EETU	4550	29,18	★★	05/04/2023 15:56
NASAH	4492	33,54	★★	29/03/2023 19:43
EETU	4474	30,57	★★	23/03/2023 18:15
SAMU	4473	34,20	★★	02/05/2023 11:08
E	4452	33,40	★★	02/05/2023 21:27
AKU	4408	38,94	★★	03/05/2023 20:31
POTTO	4346	42,70	★★	30/04/2023 00:07
EETU	4315	32,45	★★	05/04/2023 15:58
LEO	4282	39,50	★★	27/03/2023 17:26
E	3641	42,28	★	03/04/2023 21:29

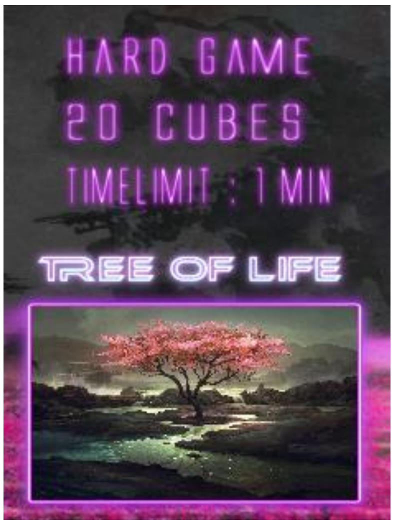
The scoreboards & charts are now combined into a single view. Global & local scoreboards are available, so the user can keep track of scores and compete against each other (in the image: medium difficulty – Global leaderboards, personal leaderboard would be viewed by clicking the “User Scores” button). Scores are saved to a database.

With the help of the chart/stats, users can easily track their progress and see how far they have come. Additionally, the easy, medium, and hard buttons provide users with the ability to filter the data they see on the chart. For example, if a user clicks on the easy button, the chart will update to remove the easy data.

3.1.4 Difficulties

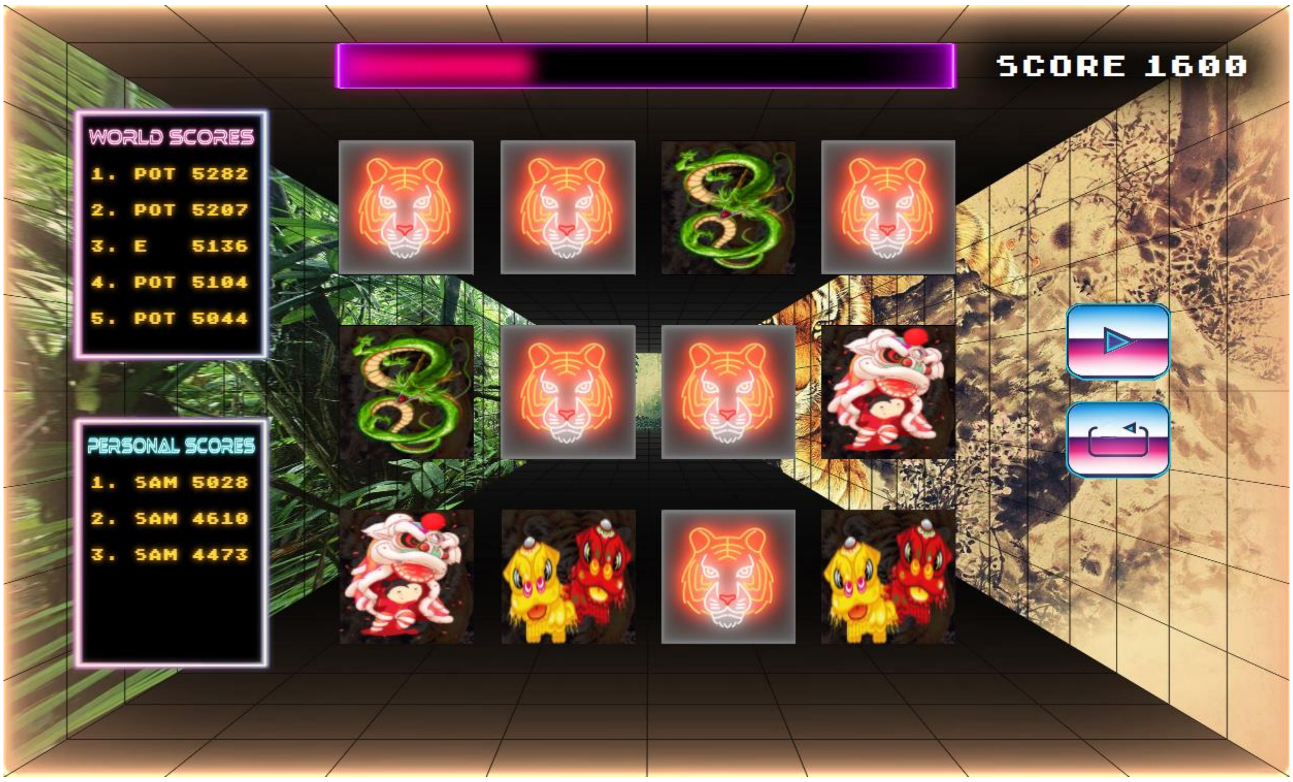






There are three different difficulties ranging from easy to hard. Each difficulty has more cubes & a time limit making them more difficult compared to the previous difficulty.

3.1.5 The game



SCORE 1600

WORLD SCORES

```
1. POT 5282
2. POT 5207
3. E   5136
4. POT 5104
5. POT 5044
```

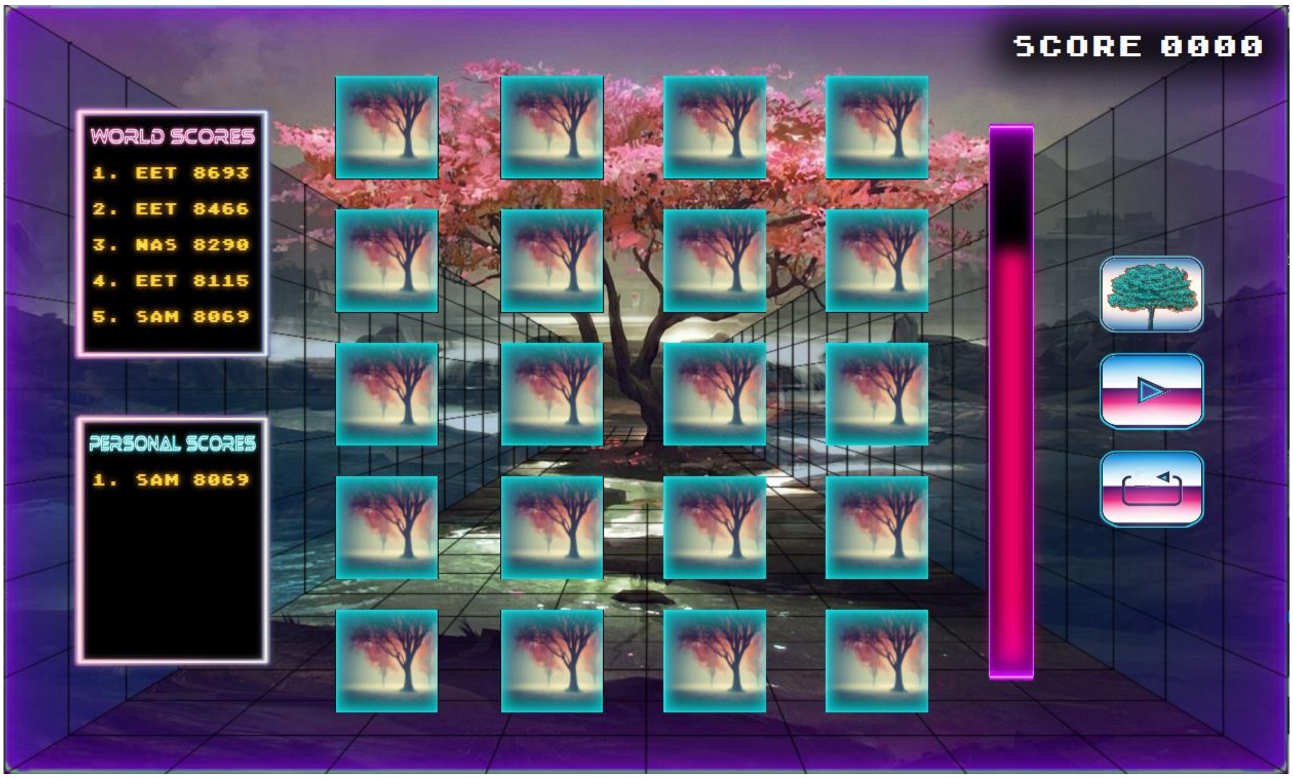
PERSONAL SCORES

```
1. SAM 5028
2. SAM 4610
3. SAM 4473
```

A memory game where the user clicks on the cards to reveal one. After it being revealed the user selects another one and if they are matching the user gets points. If the cards are not matching, the user must try again. Picture taken from the “Medium” difficulty.

3.1.6 Practice mode

The game has a built-in practice mode for the “hard” game mode. During the testing, we found that the “hard” mode is difficult, so we decided to build a practice mode.



In the image you can see the “Tree” icon above the play button. Pressing this enables/disables the practice mode. The practice mode itself disables the timer and gives you hints after a certain number of failed attempts. The scores are not submitted to the database from practice mode.

3.2 Functionality that was planned, but never made.

3.2.1 Expert mode

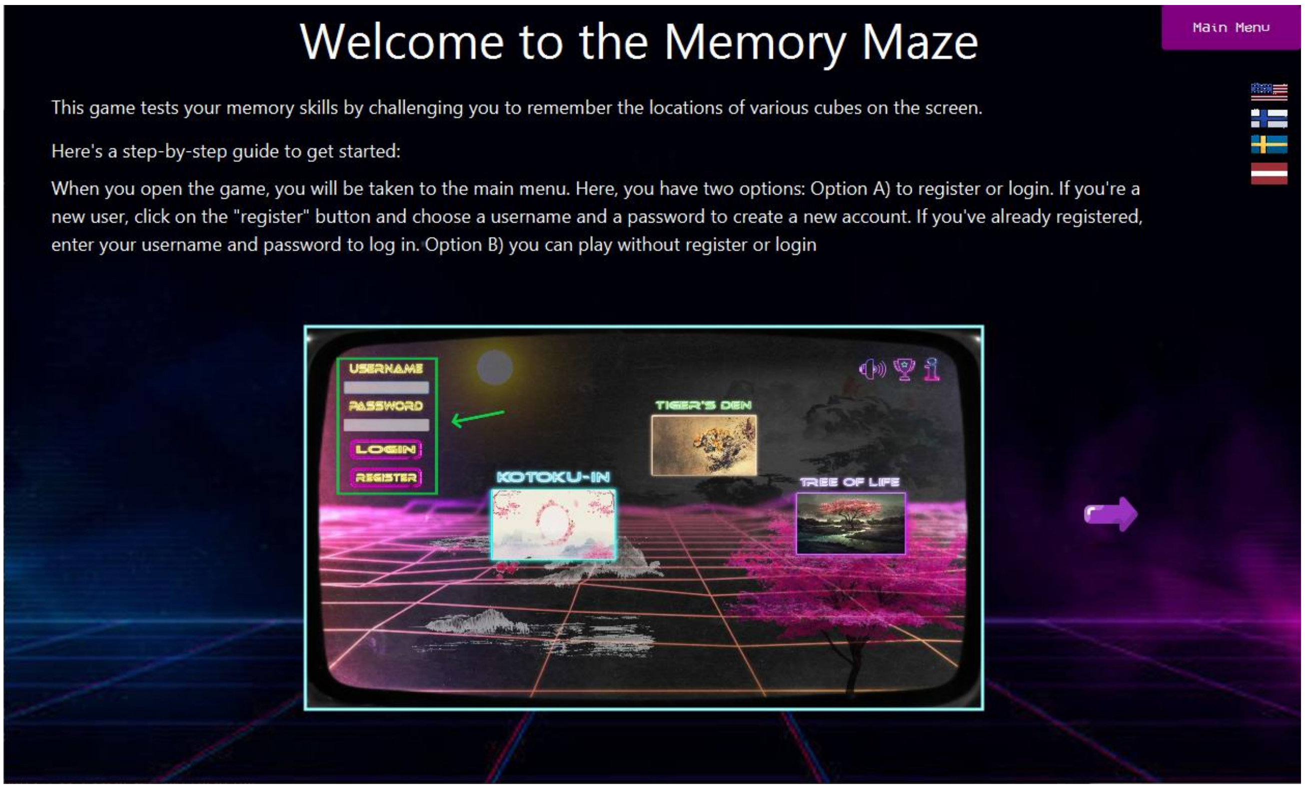
The idea was to create a difficulty that would be even harder than the “hard” difficulty. The plan was to have the cards changing places after a few wrong guesses in a row.

3.2.2 Changeable usernames & user icons

We had an idea to make the profiles more customizable but did not have the time for it. The idea was to make it so you could change your username & user icon, but currently it is not possible. Everyone has the default profile pictures and the usernames that were chosen upon creating their users.

4. Localization

During the OTP-2 course we localized the software. This means that the software is completely translated into multiple languages. The languages that we have available currently are English, Finnish, Swedish & Latvian. Our customer is a Latvian gaming company, so that is why we have Latvian. The language can be changed from the instructions menu which is shown in the image below.


Main Menu

Welcome to the Memory Maze

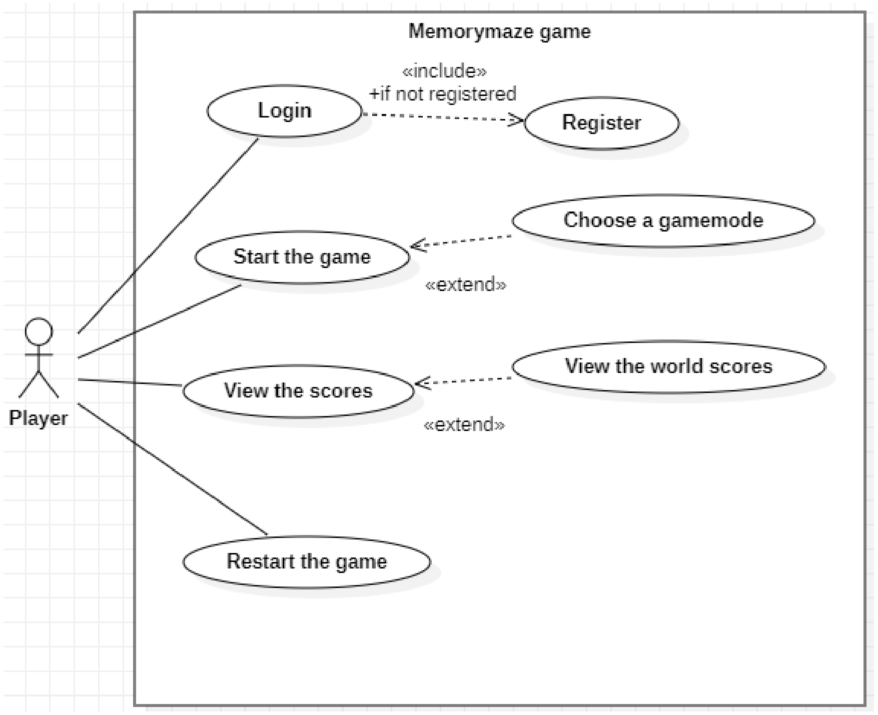
This game tests your memory skills by challenging you to remember the locations of various cubes on the screen.

Here's a step-by-step guide to get started:

When you open the game, you will be taken to the main menu. Here, you have two options: Option A) to register or login. If you're a new user, click on the "register" button and choose a username and a password to create a new account. If you've already registered, enter your username and password to log in. Option B) you can play without register or login

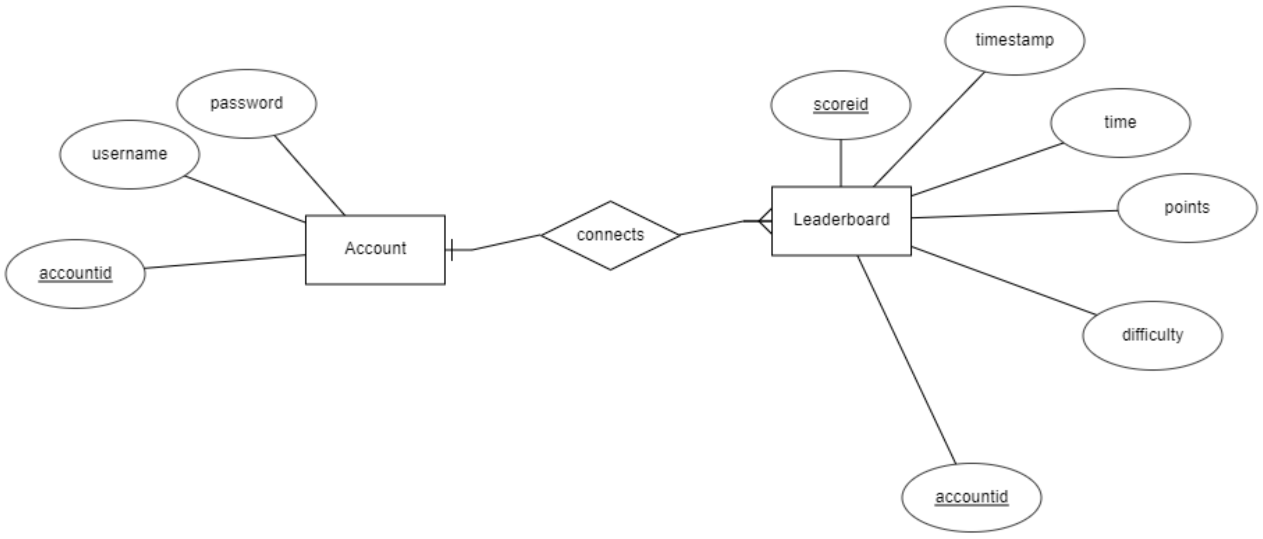
5. Actor and use cases

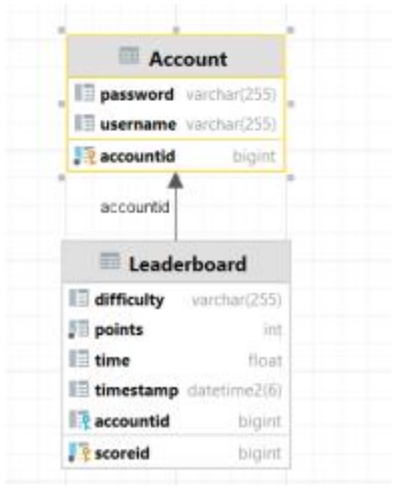
The actor is the player. Our use cases include Login, Playing the game, Viewing the scores, and restarting the game. The use case diagram can be found below.



6. Data Model

We are using the relational model to view and save our data. The images below represent how the data is saved.



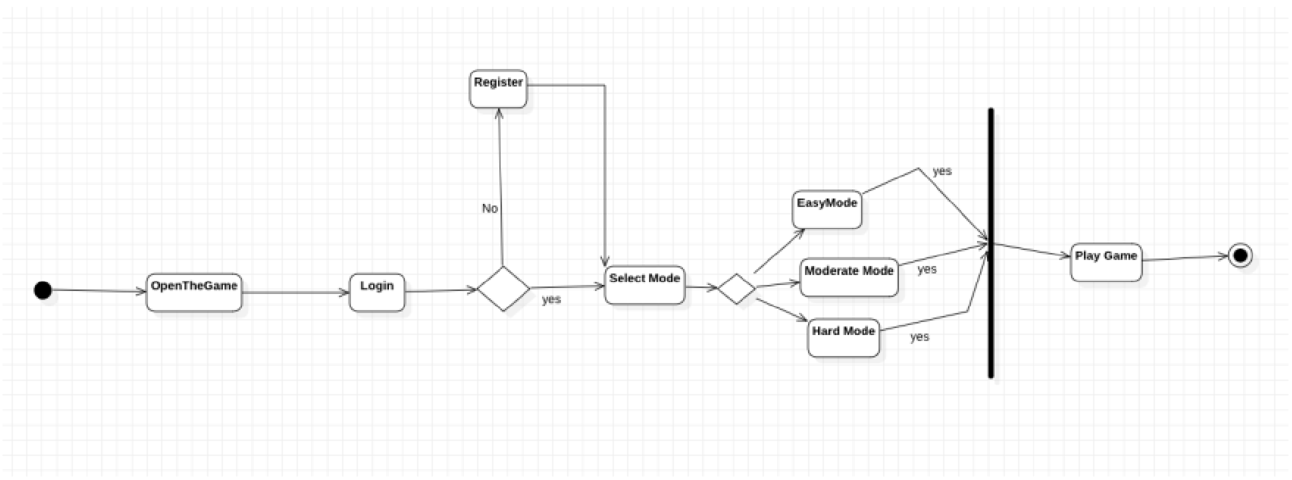


7. Structure

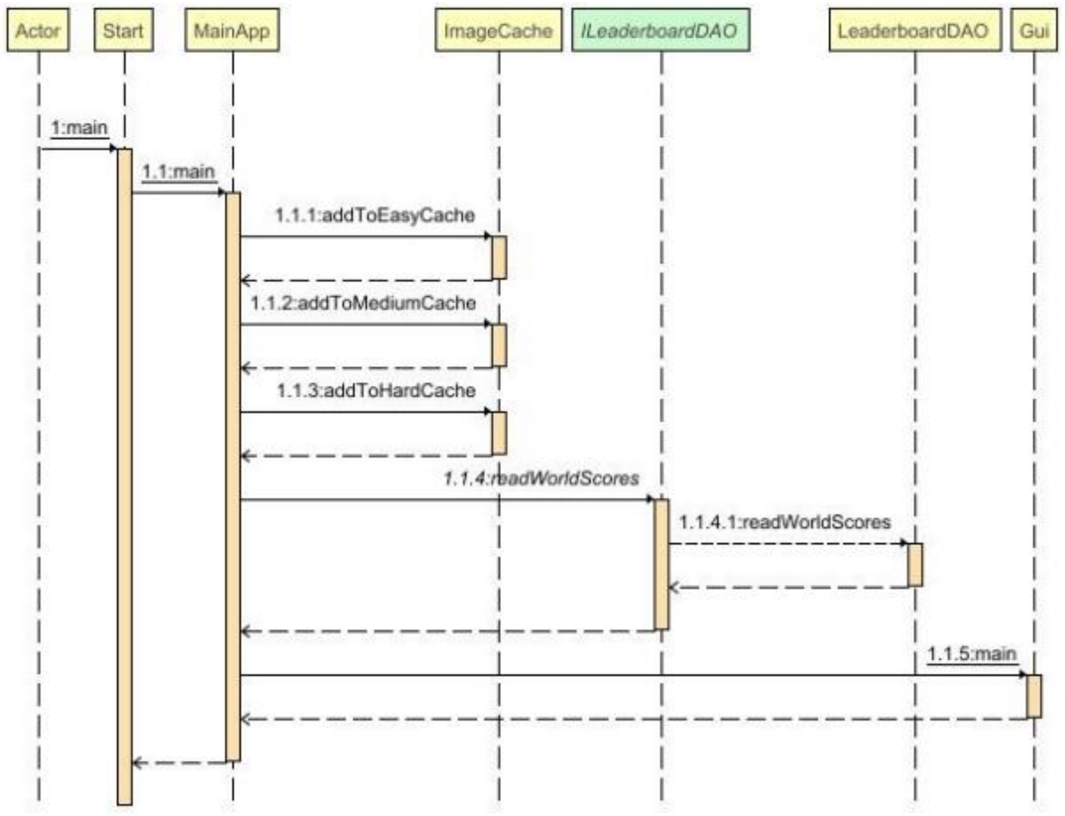
The software is made using the MVC-model. We have model, view (called visuals), controller & database packages, each containing classes related to the package's meaning. The class diagram is unfortunately too large to fit in this document properly, so it can be found here: https://prezi.com/view/c2Z1qEHTt1Eys2uqJZ8U/ . On the site click "Application", then "Diagrams" and finally "Application" to view the class diagram.

8. Activity

The way the software works is, first the player launches the game. After that the player has an option to register or login to an existing account. After that, regardless of if the player is signed in or not, the player has a choice of three difficulties: easy, moderate & hard. The player makes a choice, and the game begins on the chosen difficulty. The activity diagram describing this can be found below.



At the startup of the program, it calls a class called "Start" and loads all the images and other files to the memory cache. It also creates a connection to the database using the DAO-class. All this happens fast and after it, the GUI launches, and the game is fully functional. The sequence diagram describing this can be found below.



9. Development

9.1 Development process

The way we went about the development process was SCRUM. SCRUM is an agile development methodology that made the development process nice and quick for us. We all had our individual tasks and we had SCRUM-meetings twice a week. During these meetings we discussed what everyone had done, if there had been any issues and if the goals, we set for the period had been met. After that we set up new goals and made sure that everyone had something to work on for the period.

9.2 Testing

9.2.1 Junit

The software was first tested using Junit-tests. Tests were made early on for the classes that are important and require testing. The test results were generated to a HTML-report and can be found on GitHub under “test-results” folder.

9.2.2 Jenkins

Jenkins was then added to the software and tests were run on it. The Jenkins setup can be found on the GitHub front page, and anyone can set it up.

9.2.3 Git

The project has GitHub actions set up, so every time a pull request is made, the same tests which would run in Jenkins, run now automatically. These test reports can be downloaded from the “actions” tab inside the projects GitHub.

9.3 Javadoc

The project has a well-done Javadoc that contains all the relevant information about the methods & parameters of the software. This Javadoc can be found on GitHub inside the “Javadoc” directory.

10. Conclusion

The development went well during both OTP-1 and OTP-2 and all the goals we wanted were achieved. The software is working well, and the overall feeling of the project is good. We had fun creating the project and we all learned a lot. For future development, we want to create different difficulties and create more unique features.

Here’s a short recap of what was done during both of the courses:
OTP-1:
- Create a working game
- Test the game
- Create a database to store/retrieve scores from
- Charts to track progress

OTP-2:
- Make the game more visually appealing
- Test the game even more
- Localize the game
- Practice mode