

MEMORYMAZE

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



METROPOLIA

OHJELMISTOTUOTANTOPROJEKTI 1

Contents

1. What is the project about? 2
2. How to play the game & requirements..... 3

| | | |
|-------|--|---|
| 3. | Description of the application & the user interface..... | 3 |
| 3.1 | Functionality | 3 |
| 3.1.1 | Intro | 3 |
| 3.1.2 | Authentication | 4 |
| 3.1.3 | Scores | 4 |
| 3.1.4 | Chart | 4 |
| 3.1.5 | Difficulties | 5 |
| 3.1.6 | The game | 5 |
| 3.2 | Functionality that was planned, but never made | 6 |
| 3.2.1 | Expert mode | 6 |
| 3.2.2 | Timer visualization | 6 |
| 4. | Actor and use cases | 6 |
| 5. | Data Model | 6 |
| 6. | Structure | 7 |
| 7. | Activity | 7 |
| 8. | Development | 8 |
| 8.1 | Development process | 8 |
| 8.2 | Testing..... | 9 |
| 8.3 | Jenkins..... | 9 |
| 9. | Conclusion..... | 9 |

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.

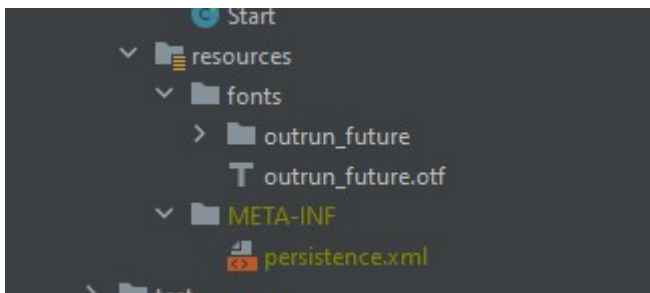
2. How to play the game & requirements

Requirements:

IDE to build the project. (We recommend IntelliJ Idea)

Java 17+

We recommend using IntelliJ to play the game, since everyone of us used it and it's known to work on it. In order to play the game, first clone the project. After cloning insert the META-INF directory inside the project's resources (image below).



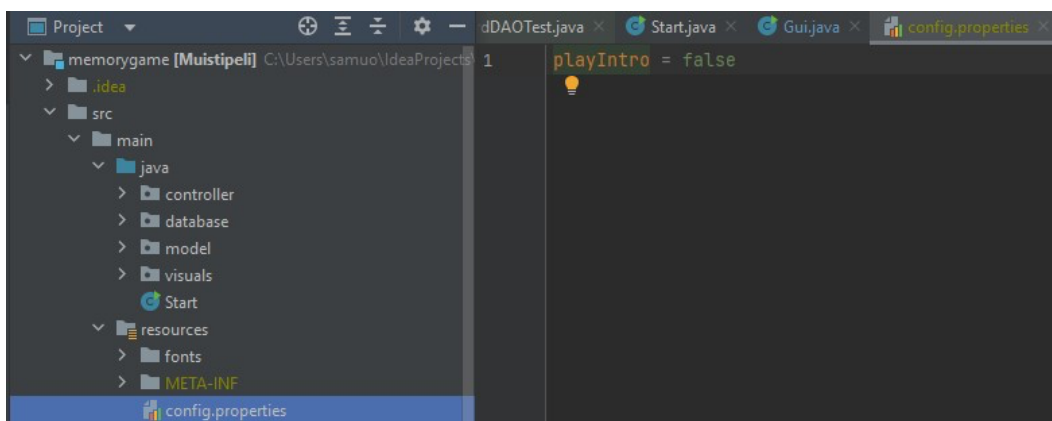
After that, build the project and the game should be fully functional.

3. Description of the application & the user interface

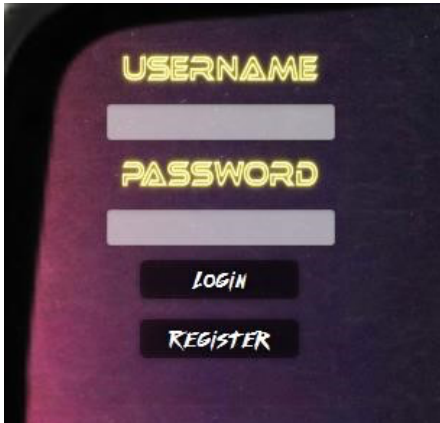
3.1 Functionality

3.1.1 Intro

There is an intro when you launch the game. Intro can be enable/disable 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".

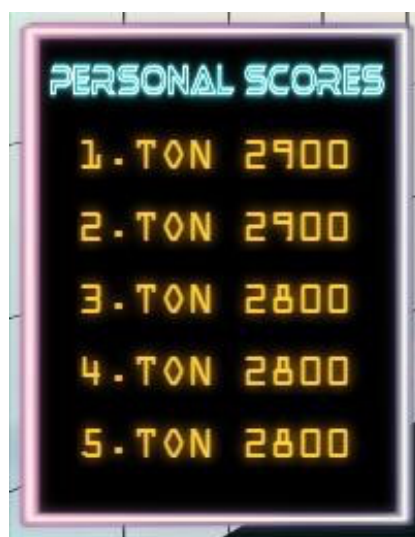
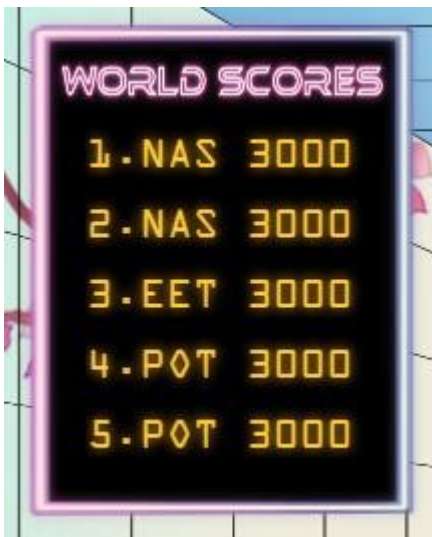


3.1.2 Authentication



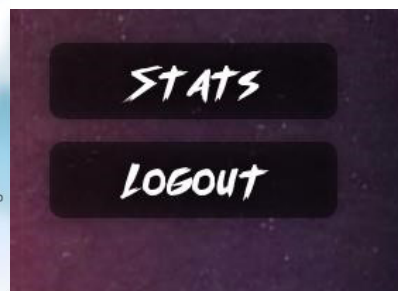
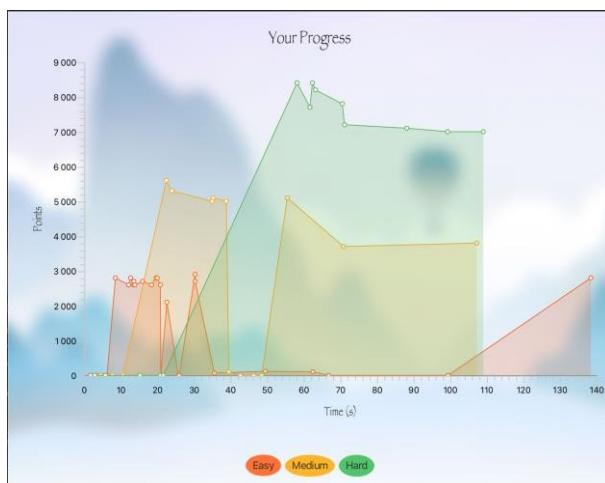
This is where the user can create a user and login.

3.1.3 Scores



Global & local scoreboards, so the user can keep track of scores and compete against each other. Scores are saved to a database.

3.1.4 Chart



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.

3.1.5 Difficulties



There are three different difficulties ranging from easy to hard. each difficulty is harder than the one before it.

3.1.6 The game



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.

3.2 Functionality that was planned, but never made

3.2.1 Expert mode

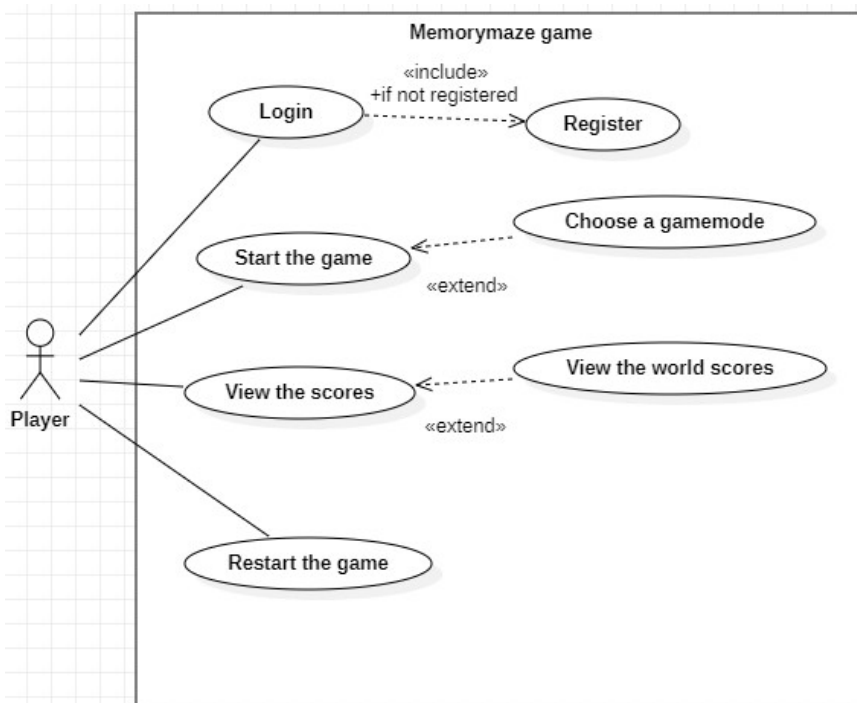
The idea was to create a difficulty that would be even harder than the “hard” difficulty. The plan was to have the cards changing places after a few wrong guesses in a row.

3.2.2 Timer visualization

We had an idea to create an ongoing timer while the game is running. The timer was made and is ready to be implemented to the visual UI, but we did not have time to implement it to the first version of the game.

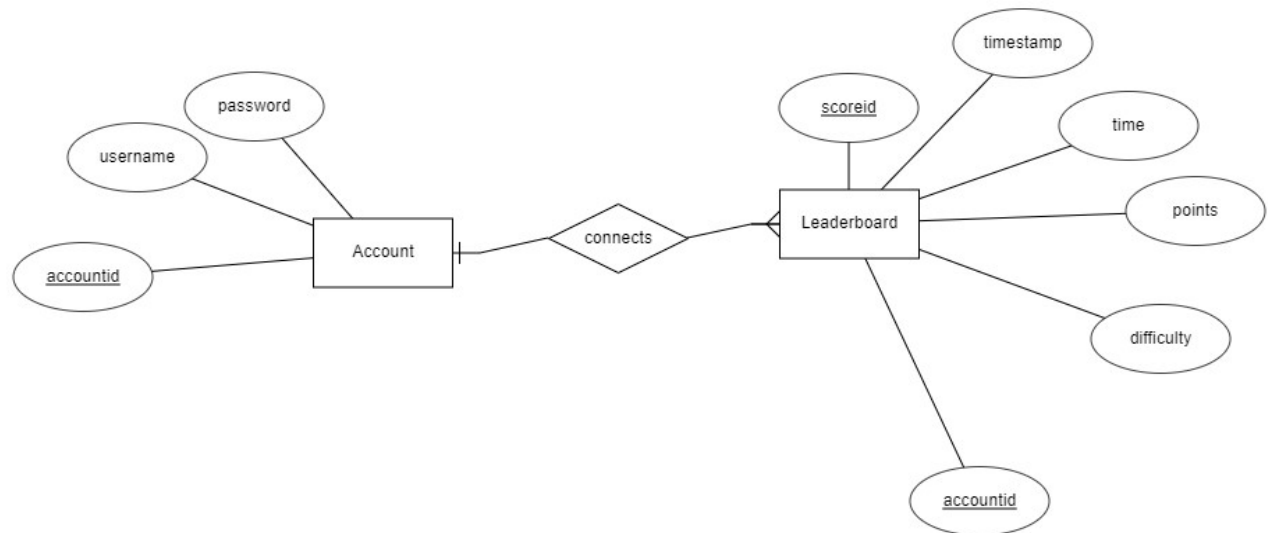
4. Actor and use cases

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.



5. Data Model

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



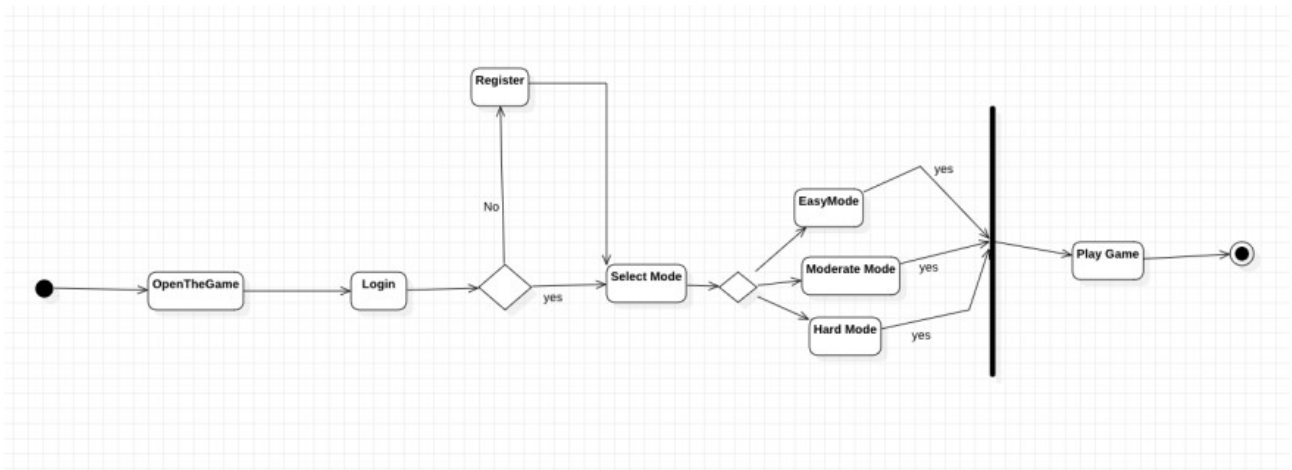
6. Structure

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:

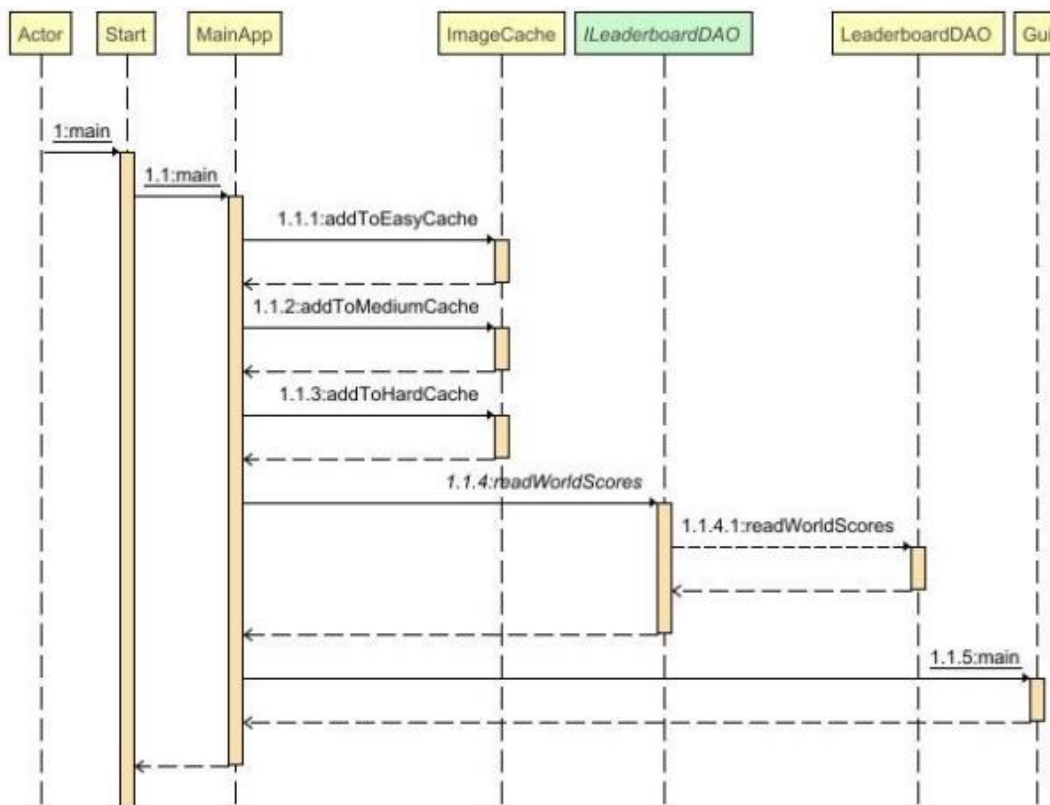
<https://prezi.com/view/c2Z1qEHTt1Eys2uqJZ8U/>. On the site click "Application", then "Diagrams" and finally "Application" to view the class diagram.

7. 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.



8. Development

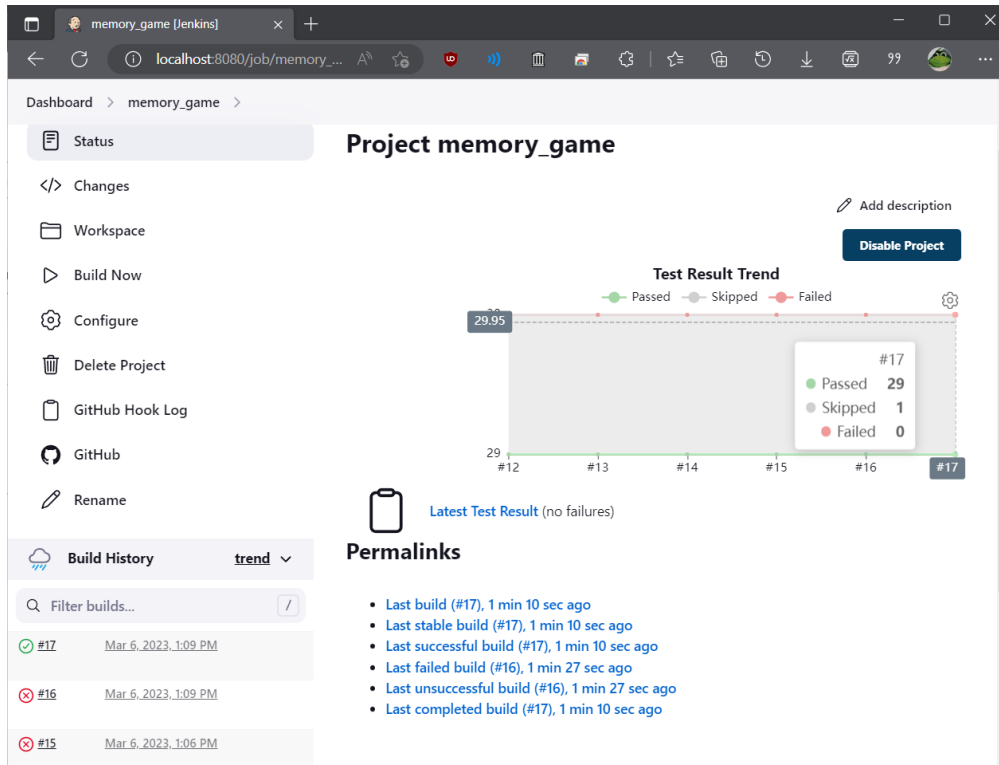
8.1 Development process

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.

8.2 Testing

The software was 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 “testresults” folder.

8.3 Jenkins



We run the Unit tests for controller and model in Jenkins.

9. Conclusion

The development went well 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 refactor the code a bit.