TDA367: Object-oriented programming project
Chalmers University of Technology

Grey Matter
RAD-documentation

and I OOPP:
ALINE EIKELAND
HUGO GANELIUS
VIKTOR JOHANNESSON
EMIL LUNDQVIST
FELIX OLIV

1 Introduction

1.1 Purpose

The project aims to construct an Android application for mental training. The application is to be competitive and rank users in relation to others as well as own past performances.

With an exploding population getting smartphone access the mobile app market is, if not brand new and unproblematic, still growing and developing. Strong influences seeking more meaningful usage of their devices can be attracted to a scientific, low weight and healthy game app.

1.2 Application specifications

The application is to be an android application compatible with android 8.0 and upwards. It will be adaptable for various screen sizes.

The application will work as a hub for multiple mental training mini games. These games can be reached from a common menu. Common for the games are also a leaderboard which ranks players internationally and amongst their friends as well as a personal progress analyzation tool. Both the leaderboards and the analyzer tool will be scoring via the Neurån scoring system. This competetive point system will have a scientific aspect as it is based on the normal distribution of the player base.

The user will be able to add other users as their friends. They can show their friends in a list and is able to remove or block existing as well as search for new ones.

Additionally the app will be able to notify the user when they are passed on the friend leaderboard, when they have not used the application for a specific amount of time and The notifications can be modified by the user. The user will also be able to modify their username and picture to be shown at the international leaderboards.

1.3 Scope of application

The application will include at least four mental training mini games. It will host an international leader board and a friend leaderboard for each of the games. *Grey Matter* will store the users player history in a personal profile which can add other profiles as their friends.

1.4 Definitions, acronyms and abbrevations

Gamification - applying gameplay principles on non-gaming activity to increase attractiveness
JSON - JavaScript Object Notification, easily readible file format for data storage

JSON-server - JSON module to easily mockup servers without having to program server part.
GSON - Google´s JSON server service.
MVC - Model View Controller pattern.
MVVC - Model View ViewModel pattern.
Normal distribution - Natural probability calculation, looks like a bell curve.
Standalone application - A application that runs locally on the device and does not require anything else to be functional.
Game/test - a short activity that challenges a player in some way, rewarding a score based on performance. Neurån - a scoring system as well as a score unit. Based on normal distribution.

2 Requirements

2.1 User Stories

> **US01 Score History**
> As an esports professional, I want to be able to see my score history, to know how well I've done previously.

Confirmed by following functional requirements:

- Can I view my history for different games separately?
- Can I see my highest score in each game?

And by nonfunctional below:

- Availability
  - Can I view and update my score history without an internet connection?
- Security
  - Are unauthorised people prevented from viewing my score history?

> **US02 Score Graph**
> As a e-sports professional, I want to be able to see a graph of my score over time, to see if I'm improving.

Confirmed by following functional requirements:

- Can I select time period to be drawn on the graph?
- Does the graph show one value for each day played?

And by nonfunctional below:

- Availability
  - Can I select if the graph shows daily averages or daily top scores?
- Security
  - Are unauthorised people prevented from viewing my graphs?

US03 Leaderboard

As a competetive person, I'd like to see my score in a leaderboard, so I can compare myself to others.

Confirmed by following functional requirements:

- Can I select to view a leaderboard for friends and worldwide separately?
- Can see a leaderboard for each game separately?

And by nonfunctional below:

- Availability
  - Can I view the leaderboard without an internet connection?
- Security
  - Can I choose who can view my scores from worldwide/friends?

US04 Score Rank

As a competetive person, I want to rank my score against everyone in the world.

Confirmed by following functional requirements:

- Can I see which percentile of all player's scores my own performance reaches?

US05 Daily Reminder

As a senior citizen, I'd like a daily reminder to test myself, so I can keep a regular routine.

Confirmed by following functional requirements:

- Can I choose to receive a daily reminder?

> **US06 Notification options**
> As a busy person, I'd like to customize how and when the application notifies me

Confirmed by following functional requirements:

- Can I select which weekdays I receive a notification?
- Can I select which time of the day I get notified?

And by nonfunctional below:

- Security
  - Are unauthorised people prevented from changing my settings?

> **US07 Notification options**
> As a user I want to play memory games to improve my short-term memory

Confirmed by following functional requirements:

- Can I play games that work on my short-term memory?
- Can I play memory games on the public transport?

> **US08 Notification options**
> As an employee I want to improve my problem solving abilities to advance my performance as an employee

Confirmed by following functional requirements:

- Can I play puzzle games to improve my problem solving?
- Can I get a score to compare previous results?

2.2 Definition of Done

The following criteria must be met for all user story implementation before they can be considered complete.

- All feature should be completely tested with jUnit tests.
- All features should work as expected by the user, with no major bugs.
- All coded features must be added to version control, in our case git.

2.3 User interface

The first iterations of the GUI:

Figure 1: The scrollable list of the games, where the user can press on one of the cards and the corresponding games starting screen comes up as seen in figure 2

Figure 2: The starting screen of the game “Chimp Test”. It contains some information about the game and a button to start the game with

Figure 3: The profile page where we want the user to be able to change some settings and possibly check on their friends and statistics

Figure 4: A dialog window that contains a list of the users friends

Figure 1, 3 and 5 are all accessible by the bottom navigation bar. Figure 6 pops up

Figure 5: A scrollable list of statistics on different games. It shows the users score in each game and how it compares to the rest of the world

when you press on a game in figure 1. Figure 4 is accessed by the “Friend” button in figure 3.

3 Domain model

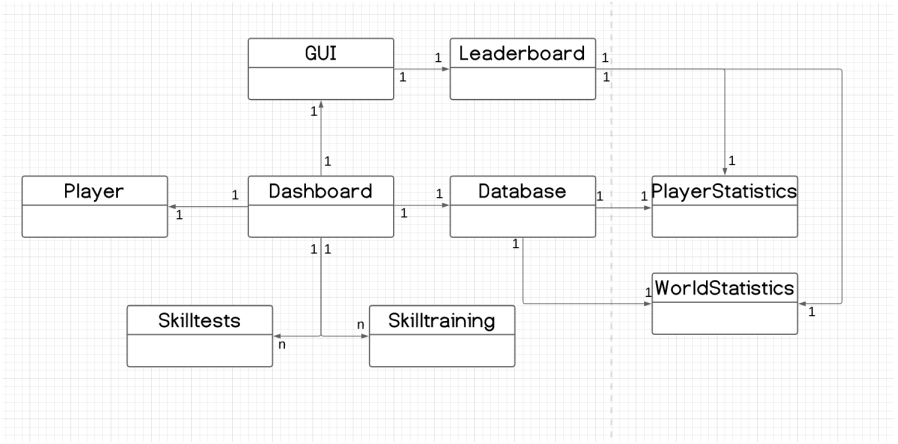

Figure 6

Multiplicity - How many instances exist of the object
Relationships - How the classes relate to eachother
Mutability - If the object is able to change/mutate
Persistence - If the object will save the data after execution

3.1 Class Responsibilities

Dashboard - Represents the first screen the user sees when entering the program, from which they can select games to play. It is not mutable.

Player - The signed in user currently using the app.

Skilltests/skilltraining - Individual tests the player can perform.

PlayerStatistics - Data for the player’s previous performances, mutable through the player’s actions.

WorldStatistics - Data for other player's statistics. Not mutable for the user, but through

Leaderboard - Shows the user's high score among other user's high scores. It is mutable as users will improve their score.

4 References