|  |  |
| --- | --- |
|  |  |
| Mobile Information Systems |  |
|  |  |
| Truth Estimation from Mobile Interaction |  |
|  |  |
|  | Written by  Jyotsna Gorle | Matr.Nr.: 120290  Christopher Heiden | Matr.Nr.: 120249 |
|  |  |
|  | Submitted to Prof. Echtler |
|  |  |
|  | Submission date: |

Contents

[1 Theoretical Part 1](#_Toc11231469)

[1.1 A Summary of the paper ” Veritaps: Truth Estimation from Mobile Interaction” 1](#_Toc11231470)

[1.2 Focused part of the paper 1](#_Toc11231471)

[2 Practical Part 1](#_Toc11231472)

[2.1 Differences between paper code 1](#_Toc11231473)

[2.2 Important code snippets 1](#_Toc11231474)

[2.2.1 Transitions 1](#_Toc11231475)

[2.2.2 AI programming 2](#_Toc11231476)

[2.3 Code Structure 2](#_Toc11231477)

[3 Literature 4](#_Toc11231478)

[4 Annex 5](#_Toc11231479)

# Theoretical Part

## A Summary of the paper ”Veritaps: Truth Estimation from Mobile Interaction”

## Focused part of the paper

# Practical Part

## Differences between paper code

In the project, we are just working on an implementation of the third task (see “Focused part of the paper”). The group decided to change the design to make it more appealing (see Figure 2, 3, 4, 5, 6). Furthermore, they added some screens to the original game app, so the user can understand the app without any explanation. Therefore, an information screen has been added that explains the game itself and how it works. Sure, it does not tell the user that the backend stores the answering time.

## Important code snippets

### Transitions

The most difficult part in programming the interface was to find out how to animate the transitions also and change the screens. Therefore, the group had to create an” anim folder”. There, xml files had to be created with a set-tag and a translate-tag. In these tags programmers can implement transition aspects like the duration of the transition and the x- and y-coordinates where the transition starts and where it ends.

To use the transition, programmers have to define an FragmentTransaction. With that, you can replace the older screen and also set up a costume animation (see *Code 1*).

|  |  |  |
| --- | --- | --- |
|  | SecondScreen fragment = SecondScreen.newInstance(); FragmentManager fragmentManager = getSupportFragmentManager(); FragmentTransaction transaction = fragmentManager.beginTransaction(); transaction.setCustomAnimations(R.anim.enter\_from\_bottom, R.anim.exit\_from\_bottom); transaction.addToBackStack(null); transaction.add(R.id.fragment\_container, fragment, "SECOND\_SCREEN").commit(); |  |

Code 1 shows how to change screens in an app and also how to call the self-generated transitions.

### AI programming

## Code Structure

The code structure for the whole application can be found in Figure 1.

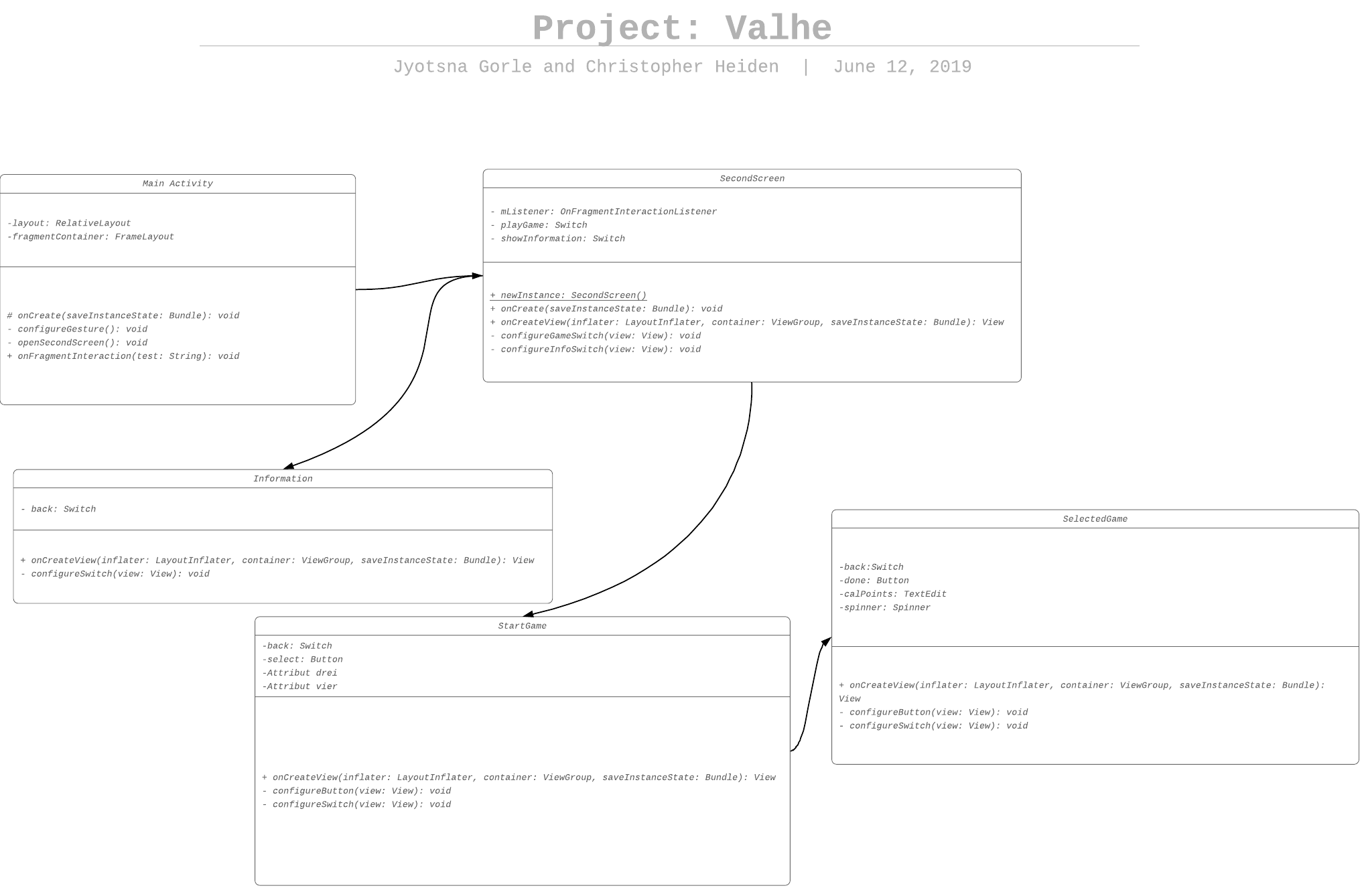


Figure 1 shows the code structure of the project.

# Literature

# Annex

|  |  |
| --- | --- |
|  |  |
| Figure 2 shows the first screen of the app which acts as an opening screen. | Figure 3 shows the second screen in which the user can either start the game or find out how the game works. |
|  |  |
| Figure 4 shows the information screen that the user can open to learning how the game works. | Figure 5 shows the rolling dice screen and the selecting dices part of the game. |
|  |  |
| Figure 6 shows the last screen of the app in which the user can choose the combination and the points he/she gets. |  |