Assignment 4 - Research Papers

Our research paper is the "Expressy Smartwatch". In this paper, the authors proposed one technique in order to improve the user's experience with touch applications. The main idea proposed is to divide the touch operation into 3 stages:

1. Intention: the period before the touch;
2. Enrichment: the period during the touch;
3. Follow-up/Recovery: the period immediately after the touch.

With this division, they used the IMU (Inertial Measurement Units) sensors, such as accelerometer and gyroscope, to enrich the user's touch experience. After explaining this approach, the authors proposed possible applications that would benefit from this procedure. Our group chose two of these apps to implement. The apps are *Maps* and *Jigsaw Puzzle Piece Rotation*. To achieve this, we developed two different softwares on the Android platform. On both applications, we used a smartphone and a smartwatch. The smartwatch only serves as the data provider and the smartphone which has better processing capacity will process and use data.

The wearable device with Android Wear 1.0 cannot communicate with a network directly. To achieve this, the wearable must have access to an Android or iOS phone via Bluetooth. Then the smartphone can connect with Google Servers through mobile data or WiFi to synchronize data between devices. Targeting to collect data from the smartwatch's sensors, we used the android development concept of *bind service*. Basically, after opening the application on the smartphone side, the service which is responsible to communicate and collect data from the smartwatch is initialized. The service is in charge of handling the data between the smartwatch and the mobile phone. This concept was used on both softwares, *Maps* and *Jigsaw Puzzle Piece Rotation*.

Maps

The Maps application provides the street view of one specific location. The smartwatch can sense the movement of the user’s arm. Based on the result of movement detection, the street view direction will be changed consistently, which can reduce the operation steps and improve the efficiency to make it more user-friendly.

Essentially, the main difference between the paper's proposed application and our solution is the touch interaction. In order to move the camera of the street view, we use the accelerometer data collected from the smartwatch to infer the direction and then move the camera. The user does not need to touch the screen. It is only necessary to move the arm on which the smartwatch is worn. On the other hand, in the solution proposed by the authors, the users need to touch the screen and hold then roll their wrist to rotate the camera.

We developed an Android application that uses the Google Maps API, which provides a Street View service. So as to move the camera, we used the accelerometer data collect from the smartwatch. With this data, we infer to which side the user is moving and then call a function responsible to rotate the camera's view. To reduce noise, we applied a threshold to the accelerometer data aiming to improve the rotation and application response.

Code link (GitHub): https://github.com/erichflock/Maps-MiS.git

Jigsaw Puzzle Piece Rotation

We used an open source jigsaw puzzle application in our implementation (https://github.com/worldsproject/android-jigsaw-puzzle-library). We did modifications not only in the way of the communication between devices, but also in the way the puzzle pieces rotate.

We developed two versions of the Jigsaw Puzzle. In the first version, the applications from the paper and ours differed in the way they rotate the pieces. On the paper application suggestion, the piece rotates following these steps:

1. Touch the piece;
2. Roll your wrist to rotate the piece.

In our application, the pieces were rotated using this sequence of steps:

1. Touch the piece you want to rotate;
2. Remove your finger from the piece;
3. Flick your wrist to rotate the piece by 90 degrees.

In the second version, we followed the paper's suggestion and the rotation of the pieces follow the same steps of the paper's application. One issue we encountered with this approach was that it was quite challenging to find the right threshold values to determine the correct rotation angles.

Code link (GitHub): https://github.com/mastersang/AndroidWearJigsaw