Assignment 3

Android, Sensors and UDP

Deadline: Monday, 23rd November 2015, 9:00am

Task 3.1: Accessing the Sensors (3 Points)

You should implement an Activity that displays the data from the following sensors:

1. Accelerometer
2. Gyroscope
3. Magnetometer
4. Barometer (if present)

Use the `SensorManager` to register your `SensorListener`. Visualize the sensor data by drawing on a view, you can for example draw graphs or you may think of another nice way to visualize the sensor's data. Use the sensors to build a `shake sensor` that detects if the device was shaken, show this accordingly on the display. Do not use any kind of system provided shake detection.

Task 3.2: Distribution of Sensor Data, UDP (5 Points + *1 Point)

You already implemented a TCP server in Assignment 1. This time we want to use `UDP` as the transport protocol. Therefore, modify your old server to use `UDP` or write a new one. You can choose the programming language, however note *.

Your server is used as a distribution point for sensor data. Clients should register with the server (using the `REGISTER` datagram) and thereby tell the server their name and address. Every time a client detects a shake, the client sends this event to the server (using the `EVENT` datagram). The server then broadcasts this message to all connected clients (except the source) (using the `SHAKE` datagram). Upon receiving a `SHAKE` datagram, the client should display a notification including the name of the device that was shaken and the time it was shaken. The clients should also be able to `UNREGISTER` from receiving events. Furthermore, a client should send keep-alive datagrams to the server every 15 seconds, otherwise, if the server does not receive a `KEEPALIVE` datagram within 20 seconds, it acts like the client sent an `UNREGISTER` datagram and stops sending shake events to this client.

Extend your Activity from Task 1 to act as the client. Be sure to include text fields in your App to dynamically change the server IP and port.

***Write your server in C to get bonus points.**

Teaching assistants:
Jan Rüth: `rueth@comsys.rwth-aachen.de`
Jó Ágila Bitsch: `jo.bitsch@cs.rwth-aachen.de`
Martin Serror: `martin.serror@comsys.rwth-aachen.de`

For interoperability between all groups we define the different datagrams:

```
#define TYPE_REGISTER 1
#define TYPE_UNREGISTER 2
#define TYPE_KEEPALIVE 3
#define TYPE_EVENT 4
#define TYPE_SHAKE 5
```

REGISTER:

Type	Length	Description
Unsigned Byte (uint8_t)	1 byte	Type
Unsigned Byte (uint8_t)	1 byte	Length of name
String (char[], UTF-8)	variable	Name

UNREGISTER and KEEPALIVE:

Type	Length	Description
Unsigned Byte (uint8_t)	1 byte	Type

EVENT:

Type	Length	Description
Unsigned Byte (uint8_t)	1 byte	Type
Unsigned Long (uint64_t, Big Endian)	8 byte	UNIX Timestamp

SHAKE:

Type	Length	Description
Unsigned Byte (uint8_t)	1 byte	Type
Unsigned Long (uint64_t, Big Endian)	8 byte	UNIX Timestamp
Unsigned Byte (uint8_t)	1 byte	Length of name
String (char[], UTF-8)	variable	Name

Be sure to be able to answer the following questions:

1. UDP is an unreliable protocol, what could happen to the datagrams? What could you do to fight the arising issues?
2. We all know sensor data can potentially contain privacy concerning data. What could you do to protect against someone eavesdropping on the communication channel? What could you do to hide the identity of the device that was shaken?

Task 3.3: Performance Evaluation (2 Points)

Evaluate the performance of your Android app according to the following criteria:

1. Required size for app (.apk file);
2. Average memory consumption / CPU usage;
3. Average frames per second (you should be able to achieve at least 60 FPS).

Your results should be presented in one or two slides. Upload the slide(s) as a solution document to this assignment in L2P.

Teaching assistants:
Jan Rüth: rueth@comsys.rwth-aachen.de
Jó Ágila Bitsch: jo.bitsch@cs.rwth-aachen.de
Martin Serror: martin.serror@comsys.rwth-aachen.de