Software Specification

Virtual Reality for Sensor Data Analysis

Project: Virtual Reality for Sensor Data Analysis
Author: Gero Birkhölzer, Johannes Blank, Alexej Gluschkow, Fabian Klopfer, Lisa-Maria Mayer
Last Change: May 15, 2017 , Version 2.0

Contents

1 Purpose

The software project in the summer term 2017 at the University of Constance focuses on the development of apps for mobile devices. In the course of the project an Android app is being developed which allows the user to explore sensor data in virtual reality.

Especially, this Software Requirements document intends to describe the functionality and requirements of the app being developed. Furthermore, the internal structure of the app as well as some test cases are specified.

1.1 Product Idea and Goal

The general idea of the product is to allow the user to record data about their environment and later explore the data in a three-dimensional scene via virtual reality. Therefore, the developed product will consist of two parts:

Firstly, the app itself. It's main goal is to connect to an external sensor device via Bluetooth and to process and save the data collected by the sensor (referred to as "app").
In order to view the saved data, the second part consists of a web application where a virtual reality scene is generated and the stored data are visualized (referred to as "web application").

These two parts will be connected in such a way that the user can open a browser with the according web application from within the app.

1.2 Definitions

App When the app itself is mentioned, we refer to the application running on the smartphone that, as stated above, handles the recording of data and invokes a web browser with the web application.

Web application This term refers to the web site that can be invoked by the app, runs in a web browser, and provides the virtual reality display of the data gathered by the app.

Sensor (device) When referring to the sensor, we're talking about the sensor device (more clearly specified in section 2) containing several sensors.

Data With data we generally mean information that has been gathered by the sensor.

Virtual Reality (scene) This term describes the three-dimensional world in which the data will be displayed.

1.2.1 Abbreviations

TI Texas Instruments

VR Virtual Reality

3D three-dimensional

DB Database

App Application

BLE Bluetooth Low Energy

1.2.2 Glossary

Stereoscopic 3D The impression of 3D is created by rendering different pictures for every eye of the viewer.

Virtual reality By using a headset in which the smart phone can be integrated, the user can view the three-dimensional world in stereoscopic 3D and thereby experiences the feeling of being fully immerged in the scene.

Augmented reality Displaying 3D objects in a real-world surrounding while providing an immersive experience like virtual reality.

Gyroscope sensor Sensor for measuring orientation in space.

Web application Web site that offers functionalities similar to those of "normal" desktop or mobile applications but runs in a web browser.

Service From AndroidDoc: "A Service is an application component that can perform long-running operations in the background, and it does not provide a user interface".

GUI From AndroidDoc: "They (Activities) serve as the entry point for a user's interaction with an app, and are also central to how a user navigates within an app (as with the Back button) or between apps (as with the Recents button)".

1.3 Mandatory Criteria

M1 The app shall use the Bluetooth adapter of the smartphone to connect to the sensor.

M2 The app shall track the position of the sensor with up to 10m tolerance.

M3 The app shall store the data retrieved from the sensor.

M4 The web application shall display a virtual reality scene using the WebVR framework.

M5 The web application shall display the stored data within the virtual reality scene depending of their respective location.

M6 The virtual reality scene in M4 shall be explorable for the user by using an external controller.

1.4 Desired Criteria

D1 The product could contain a visualization of the stored data in augmented reality.

D2 The virtual reality world could represent more than a single scene.

D3 The product could contain the functionality to view not only one set of data at a time but to generate a time lapse of the data that can be experienced like an interactive video where the user can move around and change the camera perpective.

D4 The product could provide functionalities to interact with more than one sensor.

D5 The app could provide an advanced way to determine the location where the data are recorded in order to improve the display of the data.

2 Product Environment

2.1 Software

- Android (5.0 Lollipop or higher)
- Google Chrome (Version 58.0.3029.110 or higher) (referred to as "browser")

2.2 Hardware

- Bluetooth-enabled Smartphone (referred to as "smart phone")
- TI SimpleLink SensorTag device (referred to as "sensor")
- Victorstar VRBox 2.0
- VR-Park Bluetooth Controller

3 Development Environment

3.1 Software

OS Windows 10, macOS Sierra, Linux Mint 18.1
IDEs Android Studio, Sensor Controller Studio 1.4.1, Atom, Chrome DevTools
VCS Git, GitHub
UML-Editor Enterprise Architekt, MS Visio, draw.io, UMLetino
Zeichensatz LaTeX

3.2 Hardware

Smartphone Motorola XT1572
Sensor TI CC2650STK
VR-Headset Victorstar VRBox 2.0
Bluetooth-Controller VR-Park distributed with VRBox 2.0

4 Product Functions

In the following, the required functionalities of the product are stated.

4.1 Features of the App

The app itself provides functionalities to interact with the sensor (such as to connect and retrieve data), to handle the data (process and store them in a way the web apliccation can acces them) and to invoke the web application.

4.1.1 General Features

F1.1 The app shall be able to connect to a sensor.

F1.2 The app shall be able to store information about the location where a set of data is recorded.

F1.3 The app shall provide a data view of the sensor feedback in human readable form.

F1.4 The app shall be able to save the data transferred from a connected sensor.

F1.5 The app shall be able to invoke the web application inside a browser.

F1.6 The app shall be able to link all recorded data to the location where they were recorded.

F1.7 The app shall shall save every set of data together with the respective location.

4.1.2 Settings

F2.1 The app shall be able to show information about the sensor (such as settings and state).

F2.2 The app shall list the connected devices (such as sensor, headset, controller).

F2.3 The app shall present an user interface for controlling the connection of the sensor.

4.2 Features of the Web Application

The web application handles the visualization of the stored data. By using the webVR framework, virtual reality scenes can be created as web sites and therefore be displayed using a browser like Google Chrome.

The web application can be invoked via the app and needs to access the data to generate the virtual reality scene.

4.2.1 General Features

F3.1 The web application shall be able to access the stored data.

F3.2 The web application shall be able to display the stored data at the respective position.

F3.3 The web application shall allow the user to switch between stereoscopic and normal 3D view.

F3.4 The web application shall allow the user to exit the 3D view and return to the app.

F3.5 The web application shall offer a settings menu where the user can choose which data shall be displayed.

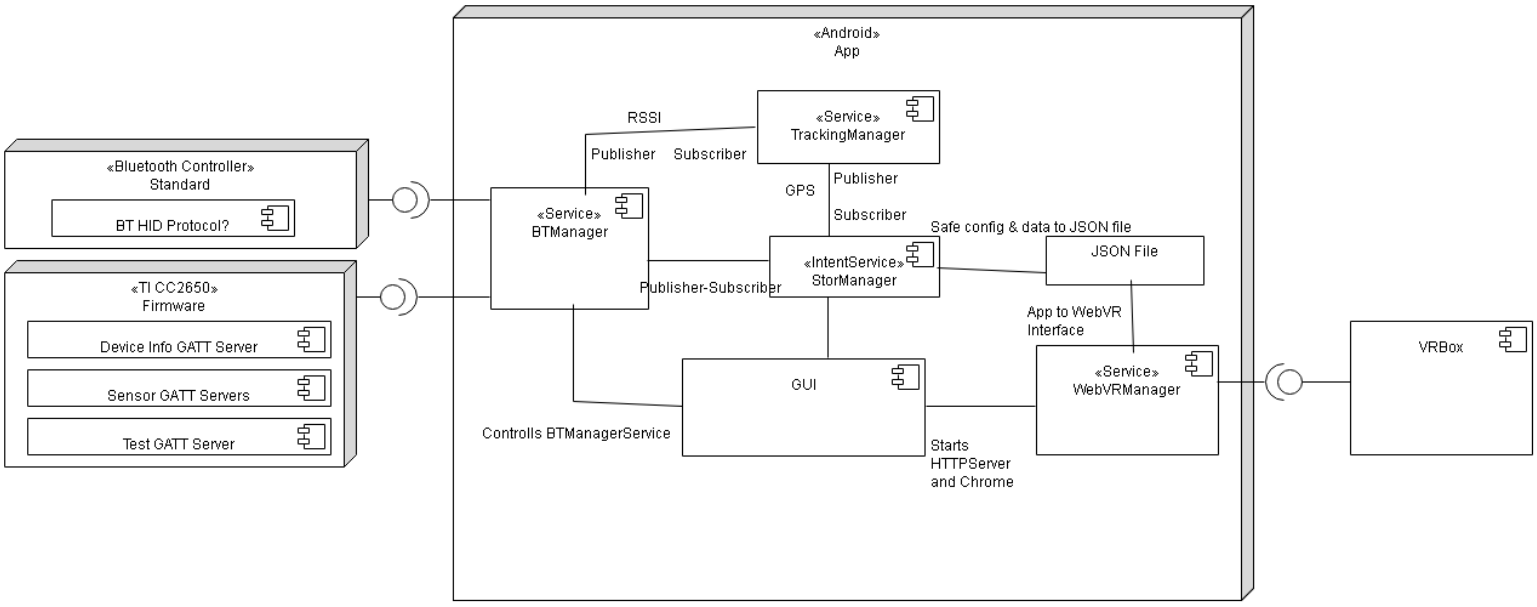
F3.6 The web application shall contain a visualization of the data which consists of a mesh over all recorded points from the sensor, while the height is the value of the given data.

F3.7 The user shall be able to move the camera around in the virtual reality scene.

5 Proposed Architecture

A better zoomable representation of these diagrams can be found in the github repository of this project in /doc/pflichtenheft/pics, where also the xml sources are.

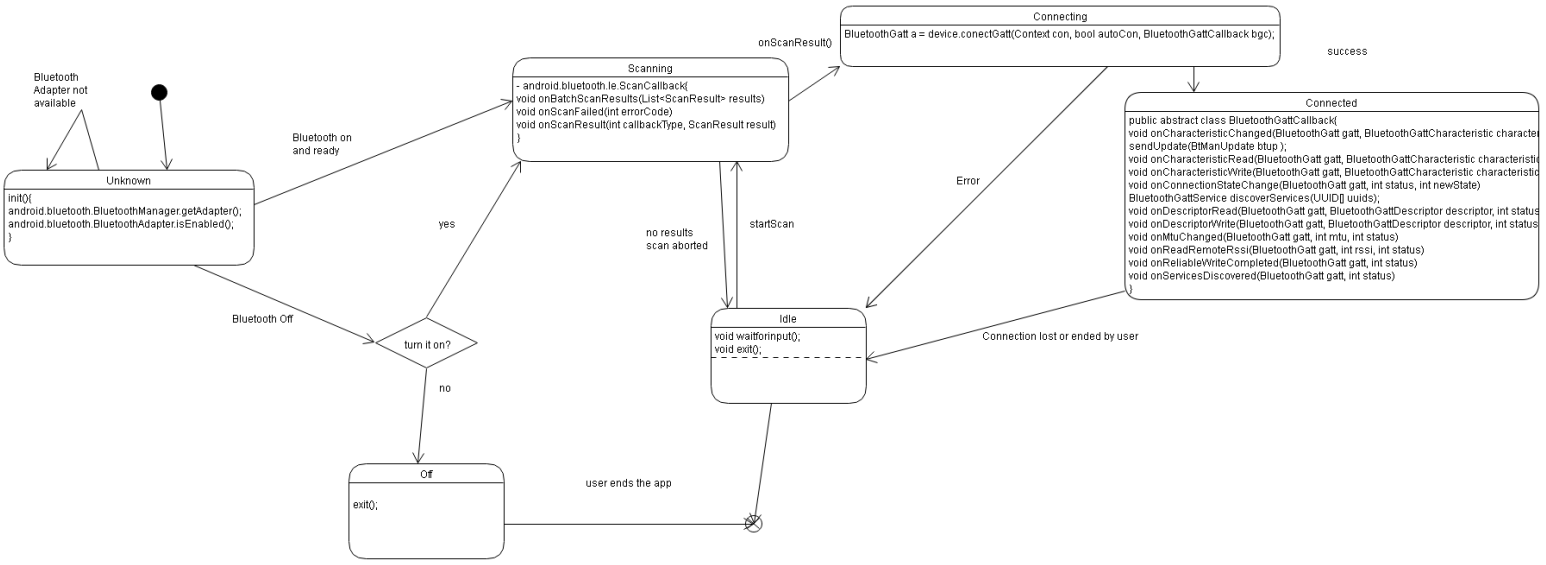
5.1 Overview



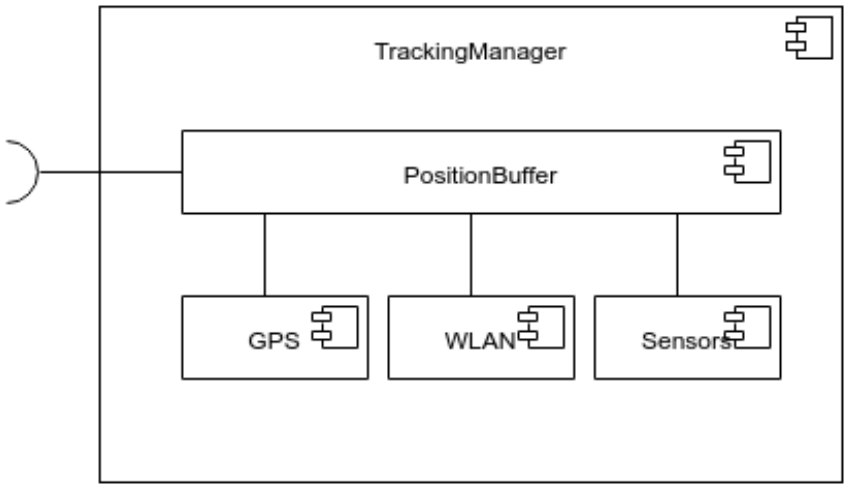
5.2 Component Decomposition

5.2.1 Services

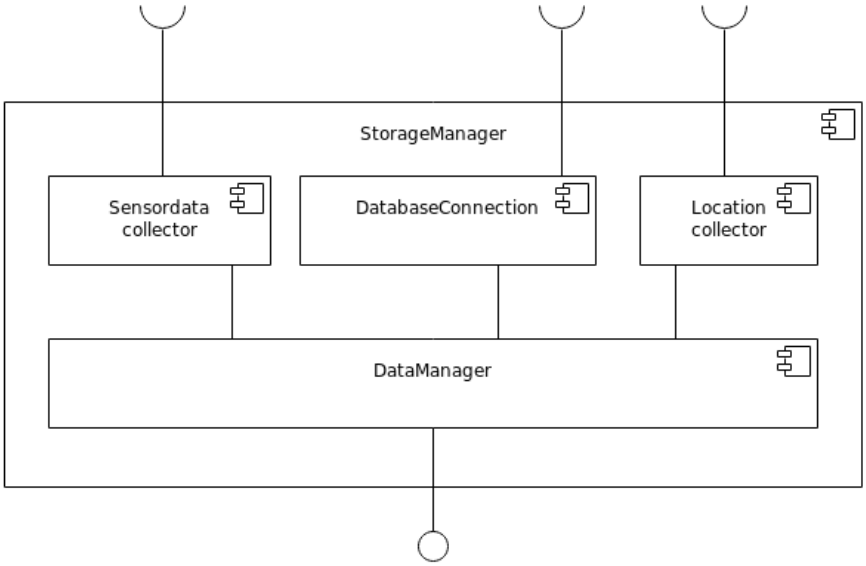
- **BluetoothManager:** Uses the android.bluetooth and especially the android.bluetooth.le libraries to fetch the sensor data from the sensor device.



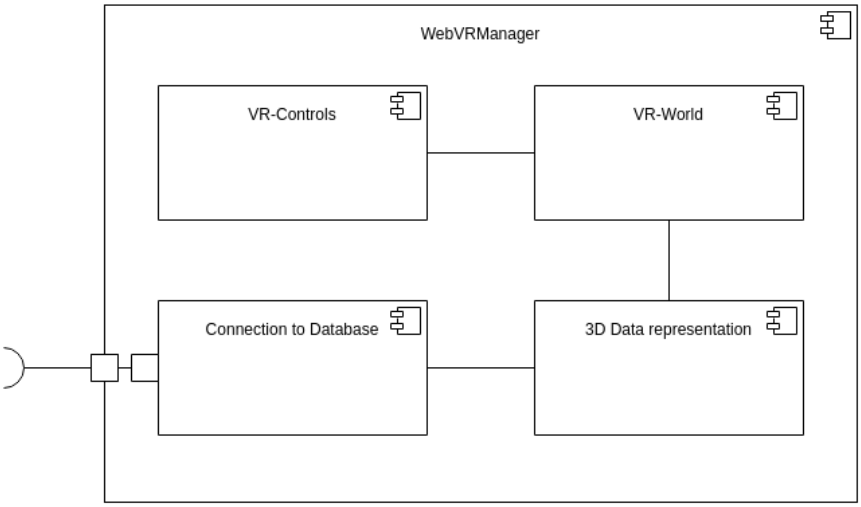
- **TrackingManager:** Handles the tracking of the (current) location where the data are recorded. The current position is determined by GPS and enhanced by the cellphone sensor and wifi data.



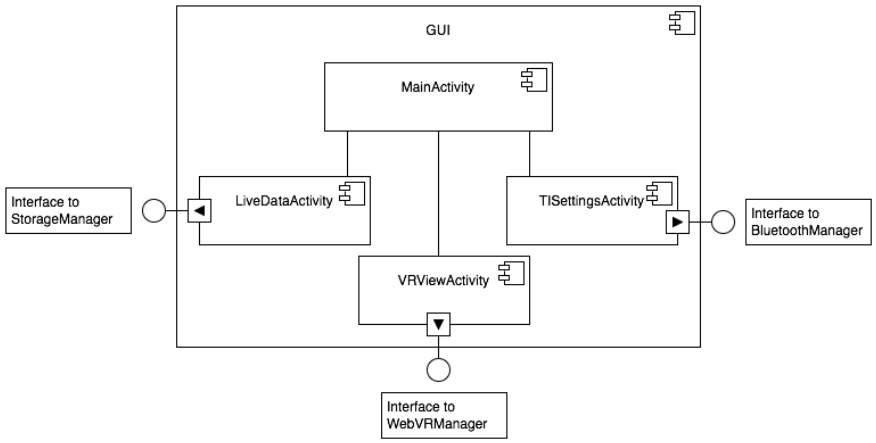
- **StorageManager:** Processes the data provided by the TrackingManager and the BluetoothManager. Uses a JSON file to store data.



- **WebVRManager:** Handles the display of the virtual reality scene and the given data from the sensor.



5.2.2 GUI



- **MainActivity** Provides the main startup screen as the main entry point.
- **VRViewActivity** Shall open a new browser window to display the WebVR webpage.
- **LiveDataActivity** Shall provide a view of the sensor data in human readable form.
- **TISettingsActivity:** Shall provide a settings screen containing scanning and connecting, connected devices and device settings fragments.
  - **ScanningConnectingFragment** shall show the scanning results, delivered by the SensorTagBluetoothReceiverService and provide a connect on/off control.
  - **ConnectedDevicesFragment** shall show the connected devices and a short info about the current setting and state of the sensor device.
  - **ConnectedDevicesSettingsFragment** shall implement the configuration of the app features of the sensor.

5.2.3 Additional Classes

- **GATT Profiles** (for each sensor one)
- **GATT Sensor Service UUIDs**
- **Parser Functions** because the BLE protocol implemented in the TI CC2650 delivers raw sensor output

6 Product Data

6.1 VR-World

D1.1 Models: The models representing the VR-World will be saved as .obj files using Blender.

D1.2 Textures: They will be saved as .png files.

6.2 Bluetooth Functionality

Service UUIDs Device Info Service 0000180a-0000-1000-8000-00805f9b34fb

Firmware Revision 00002A26-0000-1000-8000-00805f9b34fb

IR Temprature Service f000aa00-0451-4000-b000-000000000000

IR Temprature Data f000aa01-0451-4000-b000-000000000000

IR Temprature Configuration f000aa02-0451-4000-b000-000000000000

IR Temprature Time Period f000aa03-0451-4000-b000-000000000000

Accelerometer Service f000aa10-0451-4000-b000-000000000000

Accelerometer Data f000aa11-0451-4000-b000-000000000000

Accelerometer Configuration f000aa12-0451-4000-b000-000000000000

Accelerometer Time Period f000aa13-0451-4000-b000-000000000000

Humidity Service f000aa20-0451-4000-b000-000000000000

Humidity Data f000aa21-0451-4000-b000-000000000000

Humidity Configuration f000aa22-0451-4000-b000-000000000000

Humidity Time Period f000aa23-0451-4000-b000-000000000000

Magnetometer Service f000aa30-0451-4000-b000-000000000000

Magnetometer Data f000aa31-0451-4000-b000-000000000000

Magnetometer Configuration f000aa32-0451-4000-b000-000000000000

Magnetometer Time Period f000aa33-0451-4000-b000-000000000000

Optical Service f000aa70-0451-4000-b000-000000000000

Optical Data f000aa71-0451-4000-b000-000000000000

Optical Configuration f000aa72-0451-4000-b000-000000000000

Optical Time Period f000aa73-0451-4000-b000-000000000000

Barometer Service f000aa40-0451-4000-b000-000000000000

Barometer Data f000aa41-0451-4000-b000-000000000000

Barometer Configuration f000aa42-0451-4000-b000-000000000000

Barometer Calibraton f000aa43-0451-4000-b000-000000000000

Barometer Time Period f000aa44-0451-4000-b000-000000000000

Gyrometer Service f000aa50-0451-4000-b000-000000000000

Gyrometer Data f000aa51-0451-4000-b000-000000000000

Gyrometer Configuration f000aa52-0451-4000-b000-000000000000

Gyrometer Time Period f000aa53-0451-4000-b000-000000000000

Movement Service f000aa80-0451-4000-b000-000000000000

Movement Data f000aa81-0451-4000-b000-000000000000

Movement Configuration f000aa82-0451-4000-b000-000000000000

Movement Time Period f000aa83-0451-4000-b000-000000000000

Test Service f000aa64-0451-4000-b000-000000000000

Test Data f000aa65-0451-4000-b000-000000000000 shall equal the test result

Period in tens of milliseconds Configuration: 0: disable, 1: enable; in case of 3D value: 0: disable, bit 0: enable x, bit 1: enable y, bit 2: enable z

7 Quality Requirements

	very important	important	less important	lesser important
Functionality				
Adequacy		**X**		
Correctness		**X**		
Interoperability				**X**
Security				**X**
Reliability		**X**		
Usability				
Comprehensibleness			**X**	
Usability			**X**	
Efficiency				
Time response			**X**	
Resource costs			**X**	
Portability				**X**

Functionality All functions should work as intended, but neither the interaction with other software nor the security of the system is taken into account.

Reliability Errors should be reduced to a reasonable amount.

Usability The App should be usable, but user-friendliness is not stressed during the development.

Efficiency The App should respond in reasonable time to inputs. It also should use reasonable amounts of processor time and storage.

Portability The App will be developed for Android without consideration for other operating system.

8 Test Cases

/T0300/ *Look around:* While in normal 3D mode the tester shall click the screen and drag first up to move the camera up. Then move down to move the camera down, then at last left and then right, all the time the camera must follow the movement of the finger. After this the tester shall tilt the phone up to move the camera up, then tilt it down, left and right. The camera shall follow the tilt direction of the phone all the time with no delay.

This test shall be repeated in stereoscopic 3D view. While the clicking and dragging shall not work, the tilting of the phone shall be the only way to pan the camera.

/T0310/ *Move inside the virtual reality scene:* While in normal 3D mode the tester shall tilt the joystick on the controller forward and the camera shall move forward. By tilting the joystick backward the camera shall move back, by tilting left the camera shall move left and by tilting right it shall move right. The camera shall always follow the view point, so forward is always in the center of the camera.

This test shall be again repeated in stereoscopic 3D view and all functions shall work the same.

/T0320/ *Searching, connecting and disconnecting devices:* The tester shall search a sensor device by pressing the "scan" button in the settings menu. All devices nearby shall be shown in a list with distinguishable entries. By tapping on a list entry a connection to the device shall be established. By tapping again on the list entry the connection shall be terminated.

/T0330/ *Displaying Data:* While in normal 3d mode the tester shall be able to see the data collected from the sensor. The tester shall be able to see the date relative to its stored position.
The test shall be again repeated in stereoscopic 3D view and shall work the same.

/T0340/ *Transferring Bluetooth data:* When the connection to a sensor device is established, the tester shall enter the data view wihin the app and see some representation of the transmitted data which allows him to check the connectivity and functionality to/ of the sensor device.

/T0350 *Storing Sensor data:* The tester shall be able to connect to a sensor and while connected the data received from the sensor shall be stored together with its current position, on the local file system. The tester shall open this file and check if the new data is in it.

/T0360 *Transferring sensor data to the Web Application:* The tester shall check if the data is transfered to the web application and shall check if the data is corresponds to the collected sensor data.

9 Project Time Line

Week / Final Date	**Event / Tasks**
25.5.- 1.5.	first research, write Software Specification
2.5.	release SW Specification, project plan, subjects of milestones
2.5.- 8.5.	distribute tasks, decide on design
9.5.- 15.5.	start building, finalize SW Spezification
16.5.- 22.5.	
22.5.	*Milestone 1:* Bluetooth and sensor location data can be gathered, a VR-Room is built, a GUI is implemented
23.5.- 29.5.	
30.5.- 5.6.	
6.6.- 12.6.	
12.6.	*Milestone 2:* Gathered data can be displayed in 3D, *intermediate assessment*
13.6.- 19.6.	
20.6.- 26.6.	
27.6.- 3.7.	
4.7.- 10.7.	
11.7.- 17.7.	
17.7.	*Milestone 3:* All functionalities are implemented and successfully tested.
18.7.- 24.7.	prepare presentation and usage examples
25.7.	*final presentation*