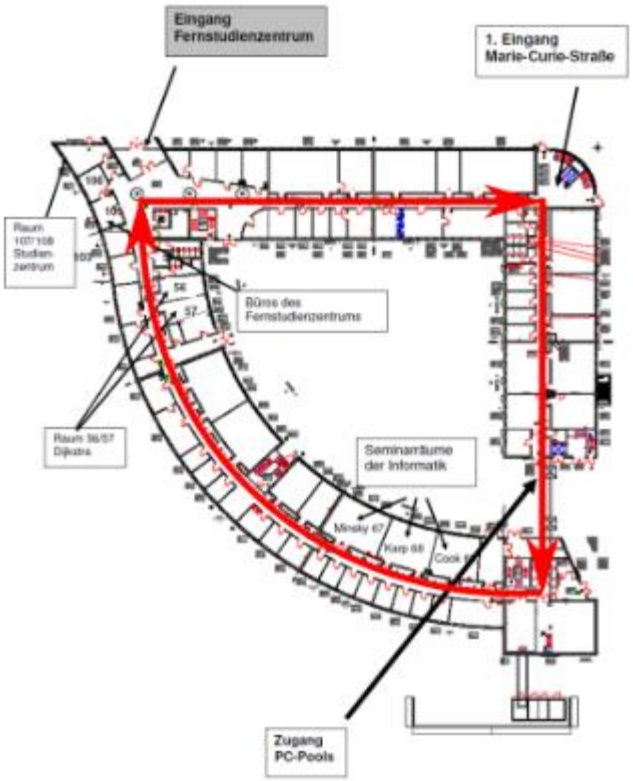
# Indoor Localisation




Institute of Medical Informatics

University of Lübeck

# Contents

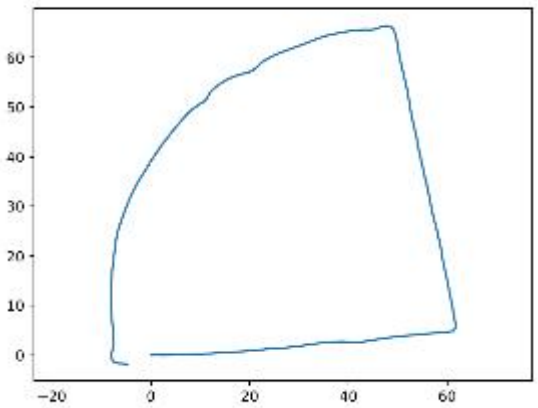
# 1. Project presentation

# About Indoor Localisation

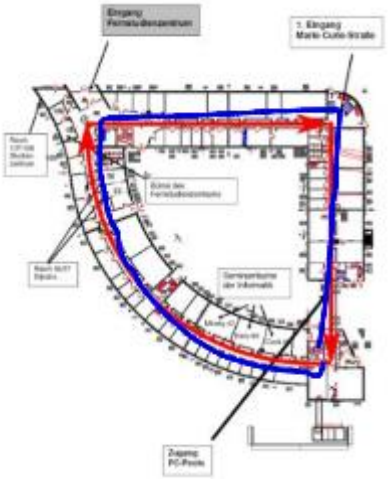
- **Goal**: localise moving objects/people in an indoor environment using wearable devices
- Tracking people in an indoor environment is a difficult task:
  - GPS not working well indoors
  - Few sensor devices available, mostly smartphone-based
  - Factors can affect the quality of the recorded data (e.g. movements of the sensors in addition to the movement of the person)
- Main solutions = probabilistic approaches:
  - Based on estimating a state, and correcting the estimation based on observations
  - E.g. Kalman filter, particle filter, …

# Project general description

- Obtaining a good indoor tracker is a complex and time consuming task (e.g. Kalman filtering, particle filtering, …)
- Scenario 2 = Python programming of a "simple" **step estimator** using smartphone accelerometer and gyroscope data
- In this scenario, no use of external information (e.g. floor plan, position of obstacles, … etc)




**Estimated track**



**Real track**

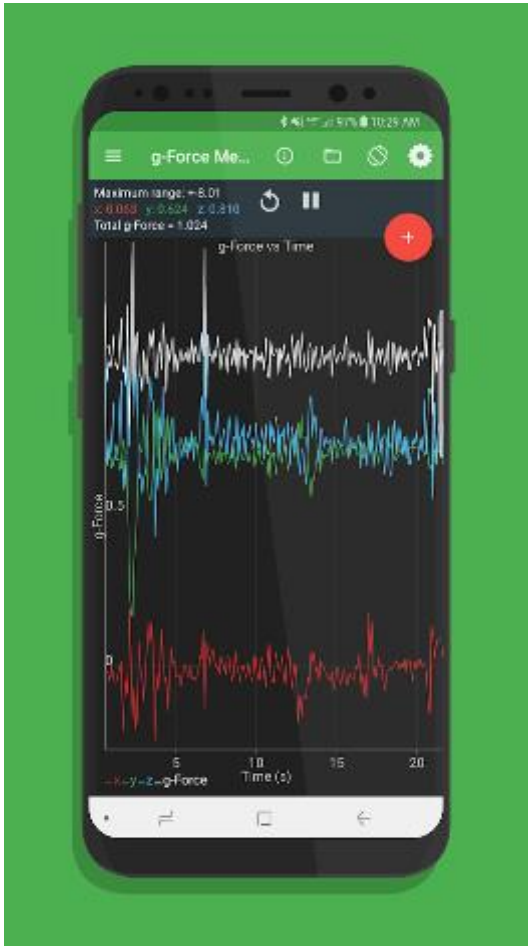
# 2. Basic step estimator

# Principle of a basic step estimator

- Process in **three main steps**:
  - Accelerometer and gyroscope data acquisition
  - Step detection in accelerometer data
  - Angle estimation in gyroscopic data
- Once estimated, steps and angles can be plotted and (theoretically) compared to some ground truth to check
- **Note**: acquiring accurate indoors ground truth is complex and time-consuming → it will not be done in this project

# Data acquisition (1/2)

- Data acquired using a smartphone with accelerometer and gyroscope sensors
- To acquire data, possibility to use the following apps:
  - **Android**:
    - Physics Toolbox Sensor Suite
    - Phyphox
    - Other?
  - **iOS:** ? (Good apps available, but most are paying only)




**Physics Toolbox Sensor Suite (Android)**

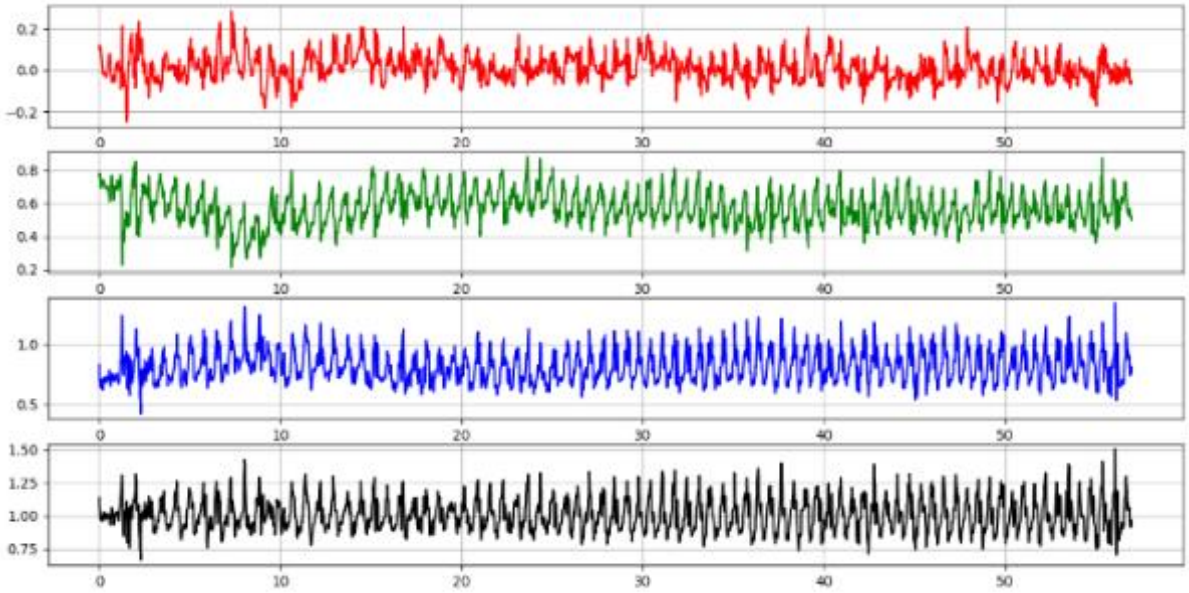
# Data acquisition (2/2)

About the Physics Toolbox Sensor Suite (Android):

- How to record data from several sensors simultaneously:
  - Open the menu by tapping the top left ≡ icon
  - Select “Multi record”
  - Select the sensors you want to record
  - Tap the ⊕ sign to start the recording
  - Tap the ■ sign to stop the recording
  - Save the record under the CSV format
- **Notes**:
  - the accelerometer sensor is called “G-Force meter”
  - data are saved on the smartphone SD card. The app gives the option to export them (e.g. via e-mail)
- The data acquisition protocol will be specified in the next slides

# Step detection using accelerometer data (1/3)

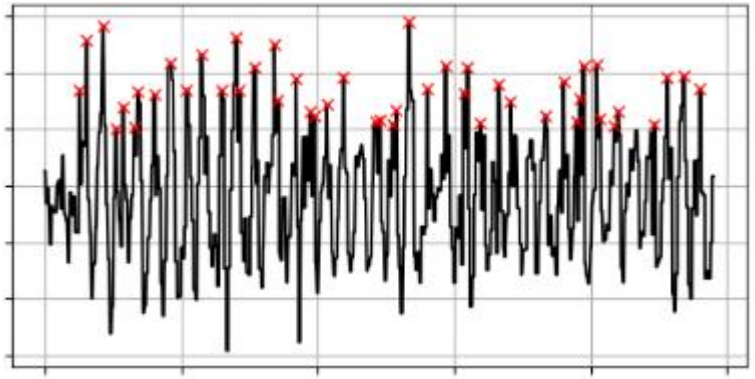
- Analysis of accelerometer data to detect steps
- One step = one peak in the signal
- Only using the norm of the acceleration is enough



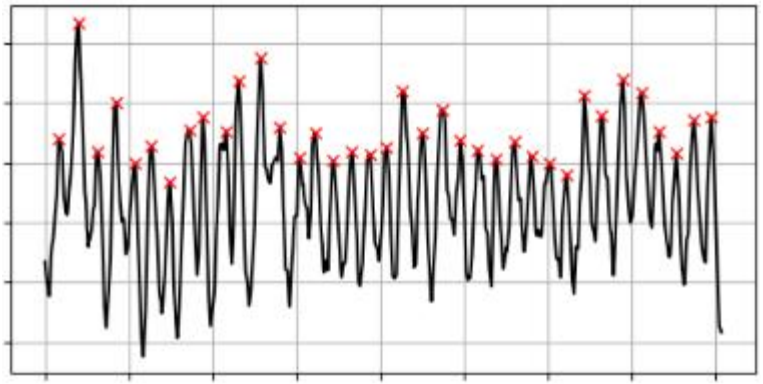

**Acceleration data acquired while walking**

# Step detection using accelerometer data (2/3)

- **Goal**: detect peaks in the acceleration signal and their beginning/end
- Sounds simple, but actually difficult because many **challenges**:
  - How to define a peak? What is the threshold? when it begins or ends?
  - What methods to use to detect a peak?
  - How to reduce the effect of noise?



**Peak detection on raw signal**



**Peak detection on smoothed signal**

# Step detection using accelerometer data (3/3)

- Two steps to consider:
  - Noise removal (pre-processing)
  - Peak detection
- The choice of methods for those two steps is up to you
- Some possibilities … :
  - … for noise removal:
    - Low pass filter
    - Moving average
    - Others?
  - … for peak detection:
    - scipy.signal.find_peaks(…)
    - Smoothed z-score peak detection
    - Others?

# Angular integration using gyroscope data (1/6)

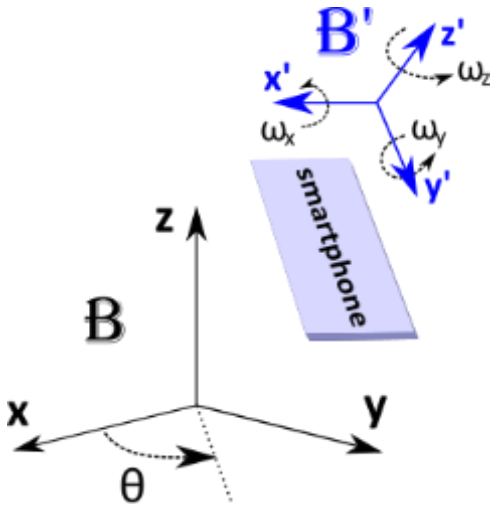
- **Goal**: detect the turning angle Θ of the subject at each step using angular velocity data provided by the smartphone

- **Theory**: if you have the angular velocity $\omega = \frac{d\theta}{dt}$, obtaining Θ is only a matter of temporal integration, i.e.

$$\theta_t = \int_0^t \omega(k)dk$$

or in the discrete case:

$$\theta(k) \approx \sum_{n=1}^{k} \omega(n-1).\Delta t$$




- In practice, not that easy because Θ and $\omega$ are not measured in the same basis!

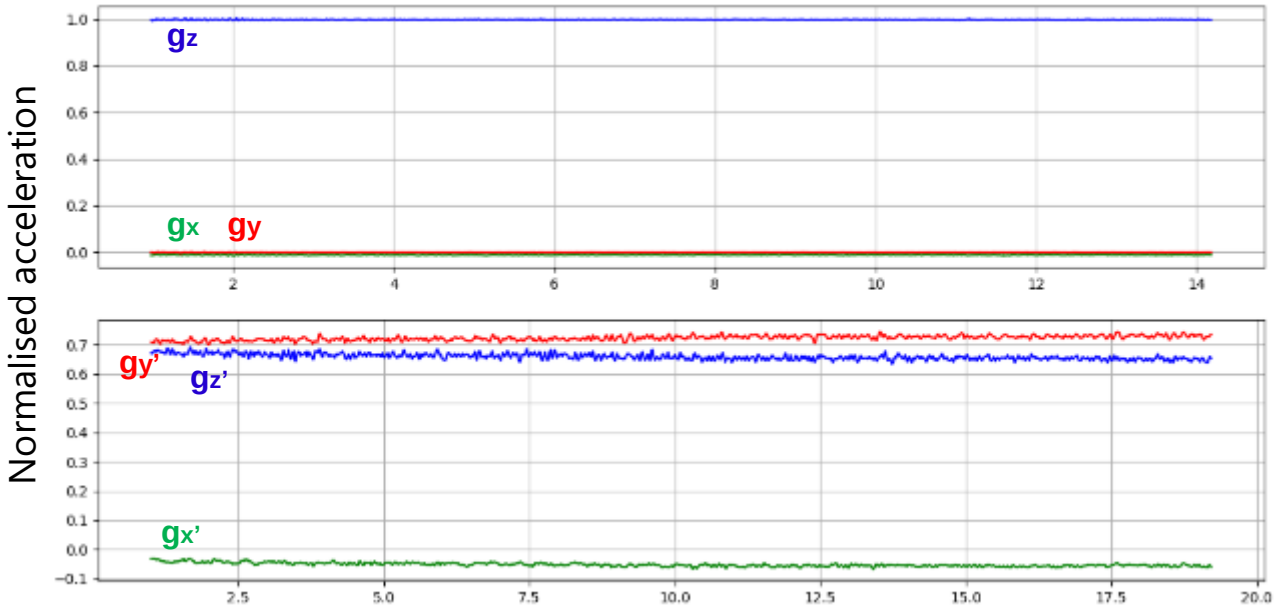
# Angular integration using gyroscope data (2/6)

- To solve this issue:
  - Find the matrix $\mathbb{R}$ of rotation between basis $\mathbb{B}$ and $\mathbb{B}'$
  - Multiply the gyroscope readings by the matrix $\mathbb{R}^{-1}$
  - Integrate the readings over time to get the angle
- How to determine $\mathbb{R}$?
  - By using the accelerometer to determine the relation between the gravity vector $\overrightarrow{g_B} = (0, 0, 9.81)^T$ and $\overrightarrow{g_{B'}}$
  - If $\overrightarrow{g_{B'}}$ is available, it is possible to define $\mathbb{R}$ as follows:
    - $\overrightarrow{u_z} \overset{\text{def}}{=} \overrightarrow{g_{B'}}$
    - $\overrightarrow{u_x} \overset{\text{def}}{=} (0,1,0)^T \wedge \overrightarrow{u_z}$
    - $\overrightarrow{u_y} \overset{\text{def}}{=} \overrightarrow{u_z} \wedge \overrightarrow{u_x}$
    - $R \overset{\text{def}}{=} \left[\frac{\overrightarrow{u_x}}{\|\overrightarrow{u_x}\|}, \frac{\overrightarrow{u_y}}{\|\overrightarrow{u_y}\|}, \frac{\overrightarrow{u_z}}{\|\overrightarrow{u_z}\|}\right] \in \mathcal{M}_{3\times3}(\mathbb{R})$

with the ^ operator denoting the cross product

# Angular integration using gyroscope data (3/6)

- Both $\overrightarrow{g_B}$ and $\overrightarrow{g_{B'}}$ can be obtained by **keeping the smartphone still** in their respective basis $\mathbb{B}$ and $\mathbb{B}'$




**Phone on a flat surface ($\mathbb{B}$)**

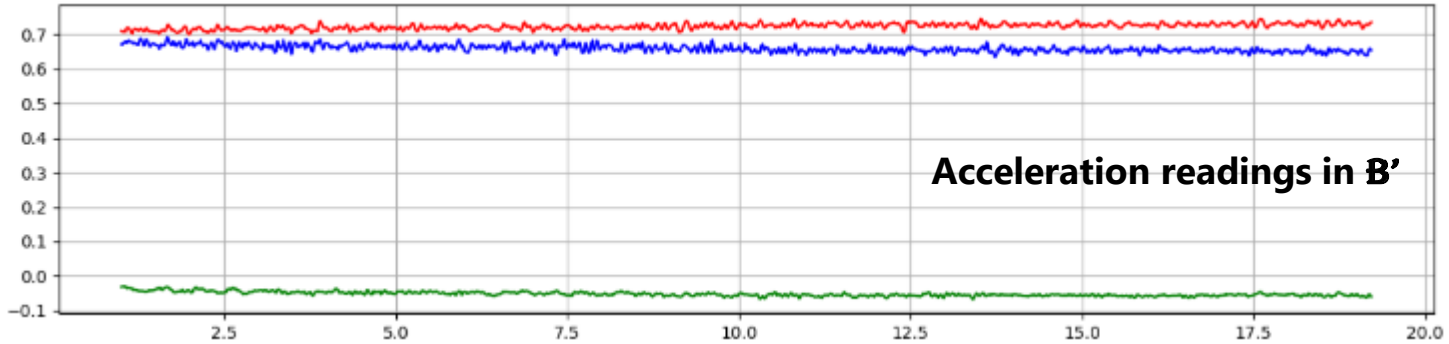
**Inclined phone ($\mathbb{B}'$)**

- **Note**: assumption = $\mathbb{B}'$ remains unchanged during the data acquisition, i.e. the phone orientation is kept the same.

# Angular integration using gyroscope data (4/6)

To get an estimation of the components of $\overrightarrow{g_{B'}}$:

- Record accelerometer data while keeping the phone in the same orientation that will be used for the data acquisition
- Estimate each component by averaging the data values acquired
- Possibility to smooth the acquired data before averaging (e.g. with a low-pass filter) for a better estimation




$$\overrightarrow{g_{B'}} = \begin{pmatrix} g_{x'} \\ g_{y'} \\ g_{z'} \end{pmatrix} \simeq \begin{pmatrix} \text{average}(\text{green}) \\ \text{average}(\text{red}) \\ \text{average}(\text{blue}) \end{pmatrix}$$

# Angular integration using gyroscope data (5/6)

- In practice, the orientation of the phone doesn't always remain the same, so $\mathbb{R}$ must be recomputed at all times to improve the estimation accuracy.
- Updating $\overrightarrow{g_{B'}}$ can be easily done by using a moving average with a low-pass filter:

$$\forall t \geq 0, \overrightarrow{g_{B'}}(t) = \begin{cases} \overrightarrow{g_0} \ \ if \ t = 0 \\ \mu \times \overrightarrow{g_{B'}}(t-1) + (1-\mu) \times \vec{a}(t) \quad else \end{cases}$$

with

- $\overrightarrow{g_0} \stackrel{\text{def}}{=} \overrightarrow{g_{B'}} \in \mathbb{R}^3$ (c.f. previous slide)
- $\vec{a}(t) \in \mathbb{R}^3$ acceleration returned by the smartphone at time $t$
- $\mu \in [0.5, 1[$ constant (e.g. $\mu = 0.9$)

# Angular integration using gyroscope data (6/6)

- The projected turn rate is given by $\frac{\partial \theta}{\partial t}(t) = \left[R^{-1}(t)\omega(t)\right]_z \stackrel{\text{def}}{=} \dot{\Theta}_z(t)$
  - **Note**: $\mathbb{R}$ orthogonal matrix $\Rightarrow R^{-1} = R^T$
- Integrating the projected turn rate yields the estimated turning angle $\Theta(t)$ at time $t$:

$$\Theta(t) \simeq \sum_{k=1}^{t} \dot{\Theta}_z(k) \times \Delta t$$

with $\Delta t$ interval of time between two consecutive sensor readings.

- **Note**: the accuracy on the value of $\Delta t$ is highly impactful on the precision of the final results

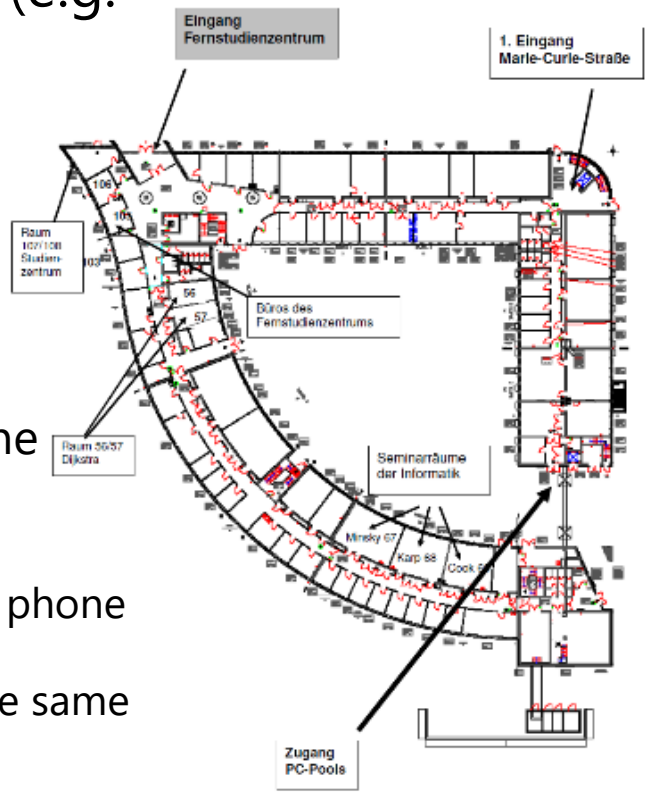
# Plotting estimated tracks

- Plotting of the estimated track possible after getting the results of the step estimation + estimated turning angles
- Let $(x_t, y_t)$ = position of the subject in the plane $(\widehat{xOy})$, $\lambda_t$ = step length, and $\theta_t$ = estimated turning angle at time $t$. Then:
  - $x_{t+1} = x_t + \lambda_t \times \cos\theta_t$
  - $y_{t+1} = y_t + \lambda_t \times \sin\theta_t$
- **Note**: estimating the length of each step is a (very) difficult problem. It is easier to take $\forall t, \lambda_t = \lambda$, with $\lambda$ constant. Usually, $\lambda \in [70, 80]\ cm$.

# 3. Data acquisition protocol

# Data acquisition experimental protocol (1/2)

- Data acquisition: used to be in building 64
- Under the new regulations: any closed track with characteristic shape should be fine (e.g. outline of a flat, square, …)
- Device: hand-held smartphone with accelerometer and gyroscope
- **Experimental protocol**:
  - Choose starting position and orientation
  - Choose starting orientation of the smartphone
  - Start recording data
    - Stay idle for 5-10 seconds
    - Move along the track while keeping the same phone orientation
    - End the walk at the same position and with the same orientation than the start
  - Stop recording data



# Data acquisition experimental protocol (2/2)

- **Note**: no acquisition of ground truth data in the experimental protocol, so no rigorous evaluation of the results.
- **Optional**:
  - If you are acquiring data outside: GPS
  - Number of steps count
- **Pieces of advice**:
  - Walk normally, no need to exaggerate steps
  - Try to keep the phone orientation the same
  - Try to acquire 2-3 tracks

# 4. Organisation of scenario 2

# Objectives of the project

- **Objectives**:
  - Acquire data following the experimental protocol
  - Program a step estimator using accelerometer and gyroscope data in Python
  - Plot the estimated tracks
- **Constraint:**
  - Data acquisition must follow the specified protocol
- **Additional specifications:**
  - None

# Project organisation

- Same group organisation as the first scenario
- Advised milestones:
  - **1st week**: data acquisition and pre-processing
  - **2nd week:** step detection in accelerometer data
  - **3rd week:** turning angle estimation with accelerometer and gyroscope data
  - **4th week**: finalization of the previous steps, step estimation plotting and report writing

# Project evaluation

For the second scenario about indoor localisation:

- **Solution** = Python code which implements the step estimator, i.e. step detection, angle estimation and plotting
- What is expected to be in the report at minima:
  - Description of the methods used for the different steps of the project
  - Overall description of your Python implementation
  - Plot(s) of the estimated track(s)
  - Small analysis of the estimated tracks you obtained (i.e. were the results satisfying? What could have been improved?)
- The structure and length of the report are up to you.

# Advised checklist for week 1

- Discuss group organisation
- Check how to use the data acquisition app
- Acquire data.
  - Performing several data acquisition rounds can ensure you get data of quality good enough to be worked on.
- Check how to convert CSV to a data type suitable for Python analysis (e.g. numpy array)
- Check if data pre-processing is needed
- Write Python data (pre-)processing scripts
- Write data plotting scripts

# Advised checklist for week 2

- Check which pre-processing and peak detection Python methods to test
- Plot the results of the peak detection to make sure that the results are plausible
- Try to find the optimal hyper-parameters of your peak detection method
- **Note**: the best approaches for peak detection and pre-processing might depend on the data acquired by your smartphone (sampling rate, quality, …). Testing different approaches is advised.

# Advised checklist for week 3

- Determine the matrix and change of basis $\mathbb{R}(0)$ by using the accelerometer data acquired at the beginning of the protocol
- Write a script to obtain the matrix of change of basis $\mathbb{R}$ at any time $t$
- Write a Python script for angle estimation using gyroscope data
- Write a Python script plotting the estimated track given estimated angles
- **Note**: the gyroscope provides readings using radians

$$1\,rad = \frac{\pi}{180}\,degree$$