Implementation of Math Model

On previous week, there was implemented Mathematical logic of accelerometer fall detection. The architecture of the implementation was easy and reliable for the first look. Program starts a thread, when the function of maths model is triggered by main package and it got new measurements from sensors. However, it was found that amount of measurements per second is too high. It is mean that there will be 25 threads each moments. Also, minimum fall time is 450 ms, consequently program can skip calculation for 10 measurements. Finally, the mathematical model will not care about calculations 500 ms, during that time Mathematical class will only add new measurements into the array of data. It is obviously that there will be only two treads of calculations in one moment. This brings less memory usage and increase of speed of the application.

Another one point was made, that thread should be started inside Mathematical class, instead of starting it outside. It brings more flexible for the program, because application will just trigger function of this class, and cares about nothing what happens inside this class.

Gyroscope

To use only accelerometer is not enough to detect the fall. The application should have protection of making wrong decisions. That is why it was figure out logic of additional part in Mathematical package.

First, application should store data from sensors for Gyroscope like for the accelerometer part in parallel mode. Secondly, main mathematical part (accelerometer) should detect a fall. If it happens, the main part should trigger method of Gyroscope class and gives time stamps of the beginning and ending of the fall. Then, Gyroscope part is going to the previous measurements, when falling is started and accelerometer has approximately **-1 g** on OZ axis. Between these two timestamps, application started to measure angle of the fall on two axis. If composition angle of two axis is equal to:

,

then it is mean that patient is laying and Gyroscope class proofs the fall.

The main problem of it is to understand from what point of time application should start calculations of the patient tilt.

Machine Learning

On the one of the previous team meeting, it was stated, that out team would implement Machine Learning only if we will have enough time at the end of the project. On the previous week, it was decided that application should be more reliable for decisions. That is why, it was figured out that application also should calculate angle of the patient during the fall.

To minimize risks, there was made decision to check how can be Machine Learning implemented in our app. If ML is faster to implement it makes sense to change the whole strategy of Mathematical part. However, after research, it was figure out that to implement ML, team faces with several problems:

* State a model for ML;(It is easy to do)
* Convert data from sensors to the data, which can be understandable by ML (it will be the biggest problem, because data should be pre-proceed. It is mean that from all data that application has it should be converted to the boolean data. Consequently, ML cannot decide if its fall or not depending on exact data from sensors)
* Implement learning techniques for ML.

Model for ML can contain next boolean fields:

* isImpact – absolute value of acceleration more than 2 g
* isLayng – absolute value of acceleration after 400 ms on OX and OY axis is equal to **1 g**
* isMoving – absolute value of acceleration still more than **1,5 g**
* isRotating – angular speed more than normal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **isImpact** | **isLayng** | **isMoving** | **isRotating** | **Fall** |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |

As the result, ML in fall detection can make only small part of the work that is why it is not profitably to implement it at the field of study.