Global Meeting Report 12.11.2013

Team ITU

Date	Time	Duration	Attendees	Communication medium
Tuesday, 12.11.2013	17:00pm (CET) 19:00pm (EAT)	~ 2:00 h	ITU Team (4 total): • Tomas • Christoffer • Jacob • Theresa Kenya Team (2 total): • Cecil • Ann	Skype

Agenda: Status-Update

Agenda (planned):

1. Status update from Team ITU (Image Processing, Prediction Models, Server)
2. Status update from Team Kenya (Mockups, Android App, Equipment)
3. Data and format (JSON) for the Android App
4. SCRUM
5. Other

Meeting progress

- Started at around 17:06 CEST (19:06 EAT)
- Kenya student (Ann) attended at 17:36 CEST (19:36 EAT)
- Kenya student (Wayua) was missing

Status update from Team ITU

Image Processing

- We can detect multiple people moving around the building and to some degree differentiate between them
    - Simply record last x,y coordinate of every detected object
    - Next time we detect movement, we simply go through previously detected objects (if there are any) and if they are within +/- 20 pixels of the last objects, we interpret it as the same objects. Otherwise it's a new object.
    - If several people are close to each other they will be detected as one entity, but otherwise this method works fairly well.
- When we detect all objects, we simply generate a JSON formatted string and pass it to the server for processing and storage.

- The format of the JSON string, which is passed between the RaspberryPIs and the Server is on Github: https://github.com/meshake/occupancy_analyzer/commit/a706b0f7b0c9ee9544af39291e077a1d0897ea44 (you have to be logged in to see this)
- Except all the integers on rip_id, room_id and object_id will be in UUID format

Prediction model

- With regards to the prediction models we started out by making our own interpretation of a Hidden Markov Model
- We can predict fairly well where a person's next position will be given a current position and a previous position
- Predicting which exit the person will take given any position is still a work in progress. As it is now, the plan is that it should be possible to ask for prediction information of a given occupant and one should receive probabilities of the occupant going to the various exits. The probabilities are calculated given stored occupancy data of course.

Server

- The server model can receive multiple connections from both Raspberry PIs and Androids
- The data base layer is close to be done
- What is missing is linking the JSON input to the database, but this is soon done with the Raspberry PIs part
- What is missing is that we (ITU and Kenya) have to agree on which data should be available for the Android and which format the JSON should be in

Setting up RaspberryPIs

- Installation and configuration guide for RaspberryPIs is on Google Drive (It’s a draft; record issues to Theresa if a step is unclear or if something is missing)

Status update from Team Kenya

- Team Kenya drew some Mockups for the Android App
- Will meet with university contact person tomorrow to ask of the status of receiving Raspberry PIs
- Team Kenya doing exams this week and another term project

Data Format for Android App

- Team Kenya:
  - Same JSON, which is used between RaspberryPIs and Server for Android App
  - Variables: Coodinates, room, floor(if applicable), date/time, total number of occupancies
- Team ITU:

- Team Kenya can get a sequence of objects in a room (occupants) in a given timespan. These objects will have a start position and the path they follow around. From this info the amount of people in a room can be extracted easily
- i.e. int getOccupants(startTime, endTime) will return the number of occupants detected in a room in the given timespan
- The Android App calls the server with JSON request, and the server will give some JSON formatted response

Conflict in Requirements

- Team Kenya: Monitor room occupation and resources to be used i.e if the room had fewer people reduce resource to a certain degree for example if a room had a certain number of people you will be required to regulate air circulation to a certain degree etc.
- Team ITU: Prediction where the occupancies go (which room for example). So the resources can be started in advance (heat up the room in advance or something like that)

SCRUM

- Team ITU was interested in if Team Kenya knows the agile software development framework “SCRUM”, because Team ITU was discussing to use it in the project
- Because Team Kenya don’t know much about “SCRUM” and the knowledge about “SCRUM” in Team ITU is also rare, we decided to not use it

Assignments Team ITU

- Writing meeting report
- Continue with the image processing, RaspberryPIs, server and prediction model
- Will give feedback on the required methods from Team Kenya (see assignments for Team Kenya) in the next meeting (Tuesday, 19.11.2013)
- Write a status update on next Tuesday and send it to Team Kenya at least 15 Minutes before the meeting (19.11.2013 16:45CEST)
- Discuss and try to include the requirements of Team Kenya (see also Conflict)

Assignments Team Kenya

- Ann will send the Mockups, which Team Kenya drew, on Wednesday (13.11.2013)
- Clear requirements for the API-methods: Write down methods, which the Android App will call and send it to Team ITU till Sunday (17.11.2013)
  - What input-parameters and what output is needed
  - Format of the JSON response
- Start on programming on the Android app (UI, back-end) on Monday (18.11.2013)
- Write a status update on next Tuesday and send it to Team ITU at least 15 Minutes before the meeting (19.11.2013 18:45EAT)
- Ann: Check on the updates in the Skill-/Preference-Sheet and fill out missing information

- Send GitHub Usernames to Tomas, so he can add everyone to the repository Share Mockup results with Team ITU
- Optional: Have a look into the document “Client-Server-Model vs. Peer-to-peer” in the folder “Team ITU” and add information/comments if you want to

Reminder

1.12.2013 Finishing a first working prototype of the occupancy analyzer. After this Team ITU will concentrate on writing the mandatory report. Further improvements and testing will be made in parallel.

16.12.2013 Hand-in of the report (End of project)

➔ **Just 2 weeks for finishing the prototype!**