**< WEEKLY REPORT FOR WEEK 1 >**

Name: Oh SuJin

Project: Multimodal Sensor Interfacing, Acquisition and Visualization

Duration: 19/5/2014 – 23/5/2014 (Week 1)

**I) Project Work Summary**

**Finished:**

* Learn how to develop Kinect applications in the Processing environment with Simple-OpenNI.
* How to export data from the Processing in txt, xml and csv.
* Study various Kinect and Leap motion based software.
* Blueprint for hand rehab application, iFarm (working title). Understand my tasks and what detection capabilities and hand/finger data from Leap Motion (Leap Motion SDK V2) would be required for such project.
* Study hardware components for iFarm project: LilyPad and Femtoduino.

**Ongoing:**

* Study different hardware from Kinect to AcceleGlove and SDKs provided. Build more programs in Eclipse to test various samples provided in SDKs. Get myself familiar with various APIs to obtain the data desired.
* Study on various Kinect library and tools to come up with framework suitable for Kinect-based rehab application.
* Review on past IO reports of Ericko and Bredan, where relevant.

**II) Tasks Assigned**

* Detailed documentation on Kinect : Learn and evaluate Kinect-based rehab applications. Study relevant library, IDE and data collection (structured file format) to come up with development framework.
* Study various sensors and make a comparison to choose which one is the most suitable for the project.
* Kinect
* Asus Xtion Pro Live
* Leap Motion
* AcceleGlove
* MYND Play
* Review on other projects including past IO reports.
* Study relevant products and prototypes available on rehab application.
* Overview of hand rehab application, iFarm.

**III) Detailed Activities / Accomplishments**

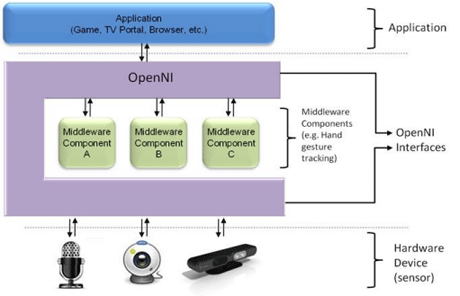
**DAY 1:**

* Research on various Kinect Development Software. Among four mostly used Kinect development libraries (Open Kinect’s libfreenect, CL NUI, OpenNI and MS’s Kinect for Windows SDK) I chose OpenNI and MS SDK and studied the pros and cons of each.
* **MS SDK**
* Pros: Good Kinect audio support. Java sound API can be connected to it and capture audio input.
* Cons: Relatively new in market. Works only on Windows 7, not XP or any non-Window platform.

The main reason I did not choose MS SDK is that I am most familiar in Java. Applications must be written in C++, C# or Visual Basic when using MS SDK.

* **OpenNI**
* Pros: Multiplatform. Compatible with other sensor available such as Asus Xtion Pro live.
* Cons: Not many audio related APIs are implemented at the moment.

I chose OpenNI as it has Java and Processing bindings. **Processing** is a sketching language that is translated into Java at compile time. OpenNI has a Processing binding called **Simple-OpenNI** which is an excellent OpenNI and NITE (middleware) wrapper for processing language.



* Therefore installing of latest SW and library required for this project & evaluation purpose had been carried out.
* Kinect SDK 1.8/Developertoolkits 1.8/Runtime 1.8
* OpenNI 2.2
* Nite 2.2
* Processing 2.2
* Processing library : Simple OpenNI

Besides Kinect, other sensor devices had been tested and studied as follows:

* **Leap Motion**
* **AcceleGlove**
* **MyndPlay**
* **Shimmer’s IMU sensor**
* Case study : Martin O’Reilly’s biofeedback application

Caparison between numerous sensors had been carried out to determine which one fits in for which purpose. For instance Leap Motion could be used for hand-related exercise application. On the other hand, Kinect could be more for upper limb & full body involved rehab programme.

**DAY 2:**

* Testing and exploring various functions available on Simple-OpenNI in the Processing environment.
* There are numerous examples which support multiple cameras, 3d depth map, Etc.
* I initially focused on how to visualize skeleton model of user as well as detecting angles between joints. As there are many previous applications / examples on such Kinect-involved exercise programme, I was advised to study how to stream out such data and store them in a structured file format. This provides some raw data for younger students to analyse and work on.
* Therefore studies on different file format from txt, xml to csv had been done.
* Simple examples to understand how to export data from the Processing were written and tested, mostly using *saveTable()* and *readTable()* statements. This aids my understanding of exporting and storing data in structured format.

**DAY 3:**

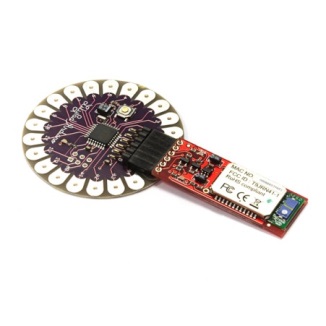
* Discussion on **iFarm** (working title): With two other interns from NUS, we were advised to work on an Arduino-controlled hand rehab application. This incorporates
* **Leap motion** to capture relevant hand gesture data
* Unity to visualize such data (by inters from NUS)
* **Arduino-controlled actuators** such as LEDs, vibration motor and buzzer to inform users about feedback.

As I have past experience working with Leap motion and Arduino-controlled actuators, I was to take charge of capturing and exporting raw data as well as realizing biofeedback system for users.

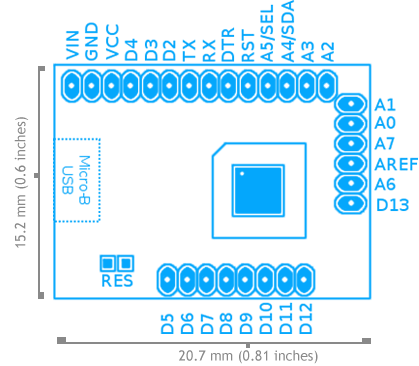
* Research on various hand rehab exercises and determine which could be realized by iFarm.
* Studied difference between virtual reality and augmented reality. While Virtual reality is all about the creation of a virtual world that users would find it difficult to tell the difference from what is real and what is not, **Augmented reality** on the other hand, is the blending of virtual reality and real life. Developers can create images within applications that blend in with contents in the real world. This explains our iFarm project where we aim to visualized 3d model of user’s hand in order to aid and improve user experience and understanding of the exercise. Users will still be in contact with the real world while interacting with virtual objects around them.
* Studied various examples and reading materials (programing guide and API reference) on Leap motion developer portal to understand newly released **Leap Motion SDK V2 Beta**.
* Compared to V1 tracking which I am more familiar with, this new V2 skeletal tracking beta provides additional information about hands and fingers and also improves overall tracking data.
* For instance, the additions which might be useful for iFarm app are
* Reporting of a confidence rating : *Hand.confidence()*
* Reporting of the position and orientation of each finger bone: *Bone.basis(), Bone.direction()*, etc.
* Reporting of grips factors indicating whether a user is pinching or grasping : *grabStrength, pinchStrength*
* Reporting of five fingers for each hand
* Leap Motion SDK V2 Beta supports Java version 6/7. Hence data collection/analysis process can be developed in Eclipse IDE which I am comfortable with.

**DAY 4:**

* Research on hardware components (Arduino-controlled actuators) for iFarm. Study pros and cons of each LilyPad Arduino board as well as Femtoduino.
* **LilyPad Arduino** : It is a small sewable Arduino board. This can be stitched to fabric and connected to other electronic pieces with conductive thread. Hence this helps us build wristband (for biofeedback) without going through hassles of connecting wires & soldering if required. Necessary actuators like LilyPad TriColor LED, LilyPad Vibe Board and LilyPadBuzzer already had been purchased by the lab.
* However the biggest problem is its size. It is too big for wristband and becomes huge to handle when connected with Bluetooth mate gold (below) for Bluetooth communication.

* **Femtoduino** : On the other hand, Femtoduino is the smallest Arduino board which weighs only 2g, 20.7mm X 15.2 mm in size. Despite its tiny size, it comes with the same power and pin count of an Arduino UNO, voltage regulator and micro-B USB port to communicate with PC. This is also compatible with Bluetooth Mate Gold for Bluetooth communication.

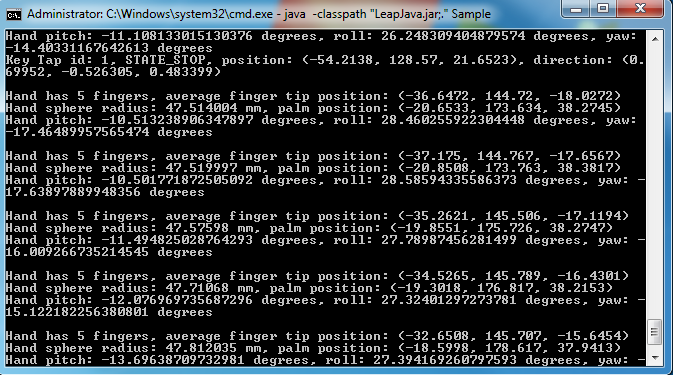


After much research, Femtoduino seems more suitable for our iFarm project due to its size and performance as powerful as Arduino Uno.

* Studied Enricko’s (NTU-IO) past reports. Although he did not touch Leap Motion, he did extensive research on how Kinect application can be programmed in the Processing environment. I explored possible Kinect-Processing (+Simple-OpenNI) application capabilities with the aid of his previous findings.

**DAY 5:**

* Continued my research on Leap Motion SDK. I wrote a simple code in Java using Eclipse IDE, instead of the Processing. This is to make sure that the future application is not limited by the Processing environment only. Furthermore most of leap motion libraries for Processing are developed before the release of leap motion SDK V2, only incorporating old APIs in V1.
* I tested new tracking API in Leap motion SDK V2. Below is the successful print out of hand/finger data desired (position, angle and gesture detected) for every frame.



* Furthermore, I was advised to continue my research on AcceleGlove, another hand motion recognition device. I was to focus on its hand/finger data acquisition capabilities; what types of data can be obtained, how accurate they are, etc.
* This will give us convincing reasons why we chose to work with leap motion over AcceleGlove and its superior detection capabilities over other similar hand exercise applications using AcceleGlove.

**IV) Future Works**

* Build more simple applications in Eclipse IDE to export/store data acquired from various sensors.
* Interfacing Leap Motion with Arduino with Node.js (research on Node.js)