**< WEEKLY REPORT FOR WEEK 4 >**

Name: Oh SuJin

Project: Multimodal Sensor Interfacing, Acquisition and Visualization

**I) Project Work Summary**

**Finished:**

* Video recording using Kinect
* Data acquisition for research purpose: I learn how to export various CSV files from Eclipse Java to store hand information
* Installing Drools
* Understanding various hand tracking information from Leap motion

**Ongoing:**

* Learning about WPF, Drools rule engine

**II) Tasks Assigned**

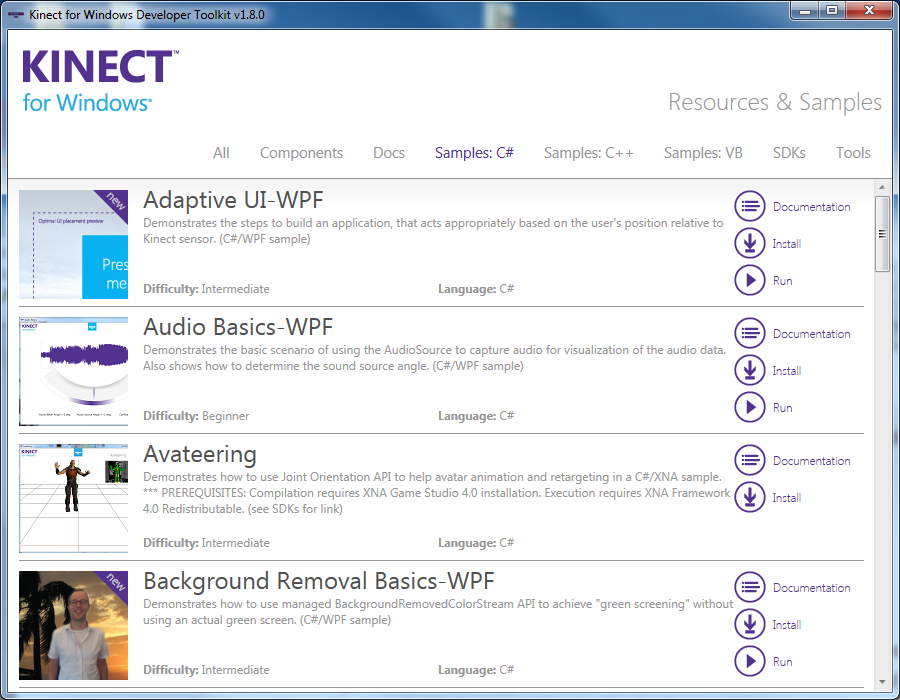
**III) Detailed Activities / Accomplishments**

**DAY 1:**

* Data acquisition for research purpose: I learn how to export various CSV files from Eclipse Java to store hand information. The information mainly contains
* Position /direction vector of palm, fingers and finger bones
* Gestures’ (circle, swipe, key tab, and screen tab) speed, position, and angle
* Grab strength and pinch strength

for each time frame.

* Before working on Kinect applications, learning about **WPF, Windows Presentation Foundation** is a prerequisite. This is a presentation system for building Windows client applications. Most of resources & samples related to speech recognition and face tracking which will come handy for iFarm application are in WPF format.



WPF offers additional programming enhancements for Windows client application development. One obvious enhancement is the ability to develop an application using both *markup* and *code-behind*, an experience that ASP.NET developers should be familiar with. You generally use Extensible Application Markup Language (XAML) markup to implement the appearance of an application while using managed programming languages (code-behind) to implement its behavior. This separation of appearance and behavior has the following benefits:

* Development and maintenance costs are reduced because appearance-specific markup is not tightly coupled with behavior-specific code.
* Development is more efficient because designers can implement an application's appearance simultaneously with developers who are implementing the application's behavior.
* Multiple design tools can be used to implement and share XAML markup, to target the requirements of the application development contributors; [Microsoft Expression Blend](http://go.microsoft.com/fwlink/?LinkID=88924) provides an experience that suits designers, while Visual Studio 2005 targets developers.
* Globalization and localization for WPF applications is greatly simplified (see [WPF Globalization and Localization Overview](http://msdn.microsoft.com/en-us/library/ms788718(v=vs.110).aspx)).

The following is a brief introduction to WPF markup and code-behind. For more information on this programming model, see [XAML Overview (WPF)](http://msdn.microsoft.com/en-us/library/ms752059(v=vs.110).aspx) and [Code-Behind and XAML in WPF](http://msdn.microsoft.com/en-us/library/aa970568(v=vs.110).aspx).

**DAY 2:**

* Discussion on iFarm; so far I have successfully obtained/stored essential hand information from Leap Motion in Java. However more in-depth understanding of various information from Leap Motion is required. For instance, I had difficulty explaining what timestamp, gesture ID and swept angle (for circle gesture) mean. Clear understanding of what each data means must be understood and documented for future studies. Furthermore more studies are required how to apply such data to determine what kinds of

1. Primitive events

2. Simple actions, and therefore

3. Hand exercises

can be used for iFarm application.

* XML, CS
* Install Jboss Drools, the Rules Engine chosen for iFarm. Drools provides an Eclipse-based IDE and hence simply installing the drools plugin in Eclipse will set the environment to develop rules. The process is as simple as shown below. (http://download.jboss.org/drools/release/6.0.0.Final/org.drools.updatesite/)

1. Start Eclipse (Kepler, in this project)

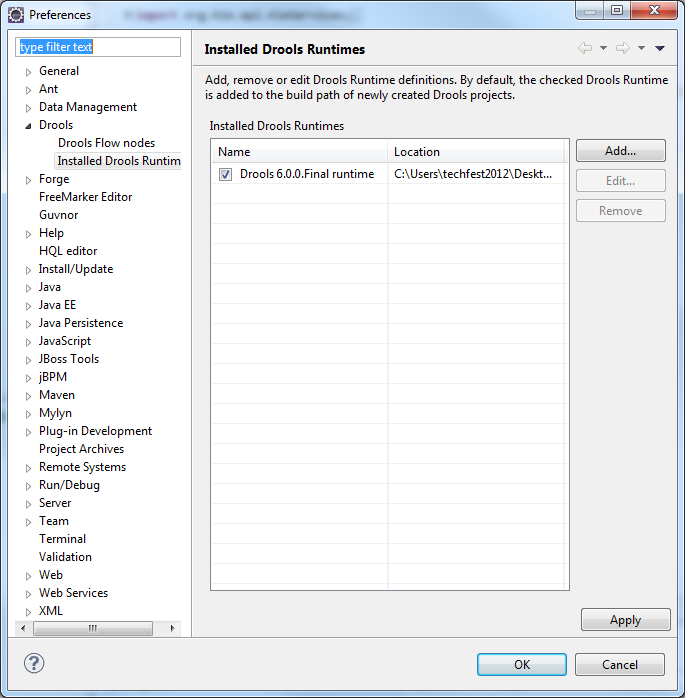
2. Help-Install New Software

3. In the Work With input field, enter "http://download.jboss.org/drools/release/6.0.0.Final/org.drools.updatesite/" and click the "Add" button

4. For the Name, you can just enter “Drools”,

5. Check the Drools and jBPM checkbox and follow the instructions to get it installed.

* Once it is successfully installed, you are required to define one or more Drools runtimes using the Eclipse preferences view. To open up your preferences, in the menu Window select the Preferences menu item. Under the Drools category, select "Installed Drools runtimes". The panel on the right should then show the currently defined Drools runtimes. In this project, I choose Drools 6.0.0. Final runtime, the latest version available at this moment.



**DAY 3&4:**

* Documentation of terms used in Leap Motion data acquisition, as highlighted in day 2.
* Walk through some examples of Jboss Drools. As Eclipse is now running Drools 6.0.0. Final runtime, drools-distrubution-6.0.0.Final.zip was downloaded. This contains not only some necessary libraries make up JBoss Drools but also some various examples and games which will help us learn the basics of Drools.

**DAY 5:**

* Video recording using Kinect: Using Processing-SimpleOpenNI library, export a video in OpenNI recording file format, oni.
* Include java system generated timestamp in hand gesture data.