**CS590BD Big Data Analytics and Apps**

**Third Increment Report – Group2**

**Tetris**

**A Sensor Based Motion Game**

**Kommineni, Siva Krishna**

**Ponnam, Balakrishna**

**Boyanapalli, Swathi**

**Pathuri, Savya Sri**

**Summary:**

To brief out the implementation done in the prior increments we have completed the collection of the open source game, designing of the UI and sensor connectivity. In this increment we have made considerable progress in data collection and training the model. We have generated sample sequence files for actions based on the gestures and motion recognition. We used the web services for the generation of these files based on the training and testing data. This data is clustered using the K-means clustering and the formed clusters are trained in various sequence operations for the perfect motion recognition

**Framework Specifications:**

In this increment we made progress in the selection of the game and analyzing the direction movements present in it so that we can match it with the sensor tag direction movement. First we should collect the data, push it into HBase and host a restful web service using the Glassfish server. We are enhancing the connection establishment between the accelerometer data in the application file and the sensor tag accelerometer connections. Currently we are running the application on the emulator for searching the exact movements. The mobile client is entirely developed in Android and it is a native application for android users. The version supports from Froyo to Jellybean’s. We are using the K-means clustering to form clusters and analyze the motion of the data using the sequence file operations like training and testing.

**Application Specification:**

* **Software Specification**
  + Tools: K-Means, R, HMM, Android Development Kit, Eclipse Kepler
  + Operating System: Android
  + Development Operating System: Windows 8
  + Programming Language: Java 7.0
  + Databases: HBase, Hadoop

**Activity Recognition Scenario and Data Collection**

* **Devices/Sensors:**

## Sensors: TI Sensor Tag - CC2541DK-SENSOR

**Devices: Android devices**

* Bluetooth 4.0 compatible android device
* Android 4.3 or above OS
* GPS

**TI chronos Watch – EZ430-Chronos**

**Implementation:**

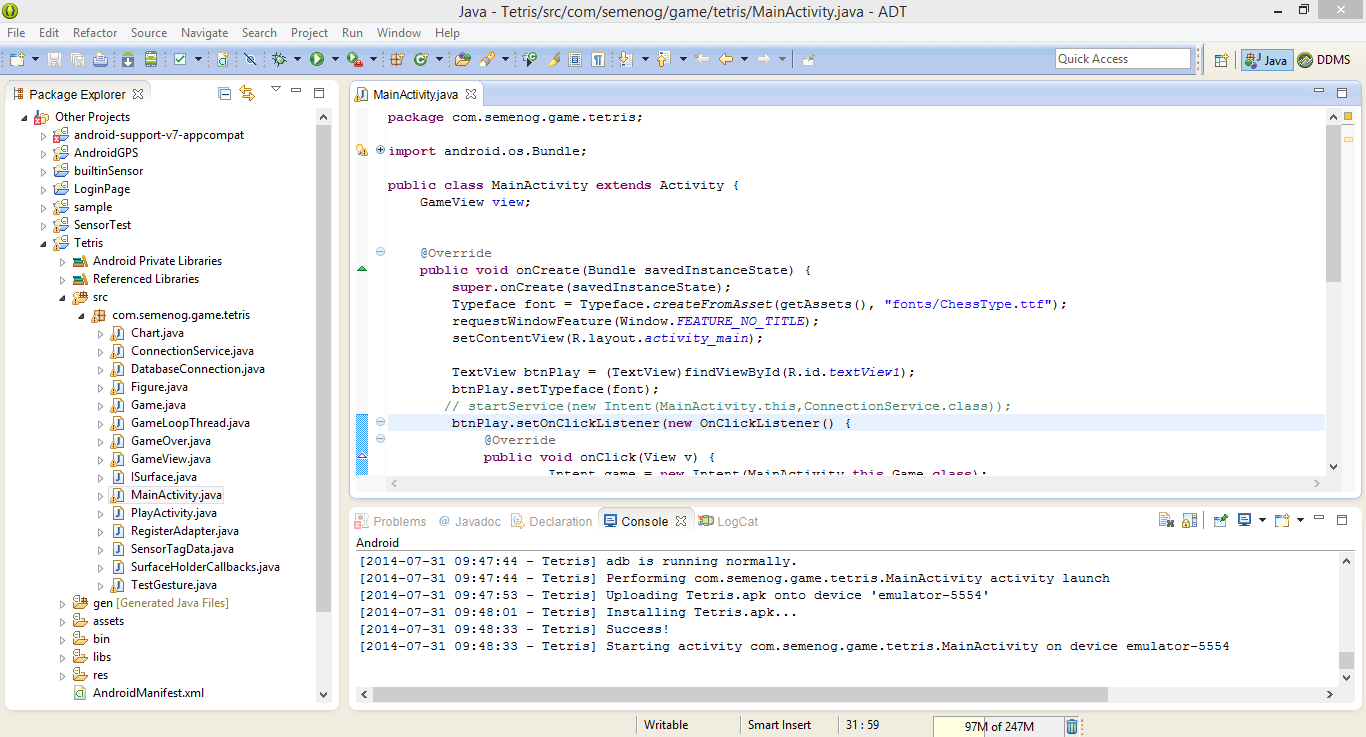
The implementation part typically consisted of following steps.

* Selection of an open source game.
* Enhance the application according to our requirements.
* Collection of data from game for required motions.
* Clustering and analyzing data and later train and test the data.
* Activity and motion recognition based on the available files generated.

First we need to deploy our open source game on an emulator to analyze the motions and actions based on the sensor tag movements.

Enhance the existing code for establishing the connection with the sensor tag. Later we collect the data from the game based on the actions.

The collected data is clustered using K-means clustering and later trained, tested using the sample web services for the generation of the sequence files.

****

**Enhancements to existing open source project:**

To those downloaded code we have added three classes for running with sensor one of the class for establishing the sensor connection. One of the classes is for identifying the patterns by comparing with the probability of the gesture.

In Android Manifest I have added some of the permissions and the Bluetooth connection and some of the services.

ConncetionService 🡪 This Class is added for establishing a connection with the sensor by using the UUID tags and collecting the data of particular service as present I have taken accelerometer, Humidity and Temperature.

SensorTagData 🡪 This class is added for collecting the data from the sensor tag which is connected and matching those data with the sequential file which is added by training the data.

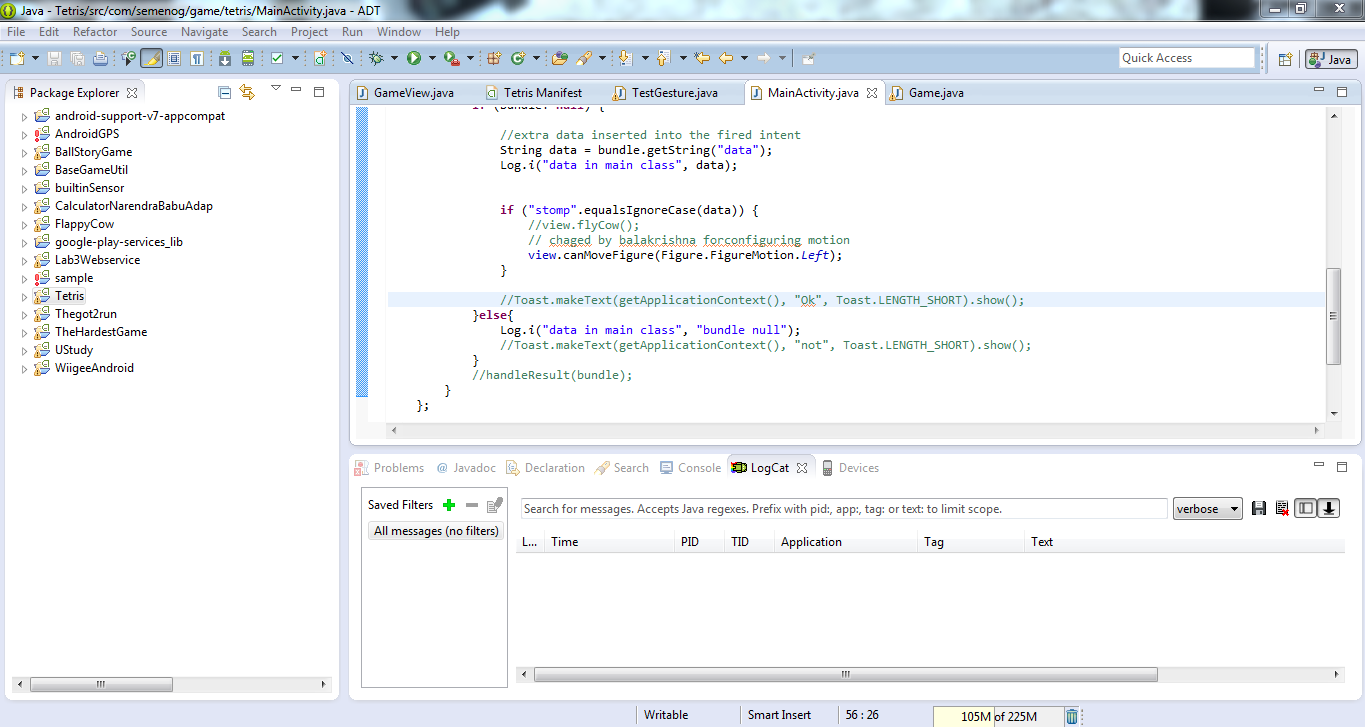
TestGesture 🡪 This class is added giving the sequence file and for choosing the gesture from the data to which side need move it may be Right, Left or to rotate which is done by comparing with the probability.

AndroidManifest 🡪 Have added the services, Established Bluetooth connection and the permissions.

And some of the supporting Jar files and the Google play service API.

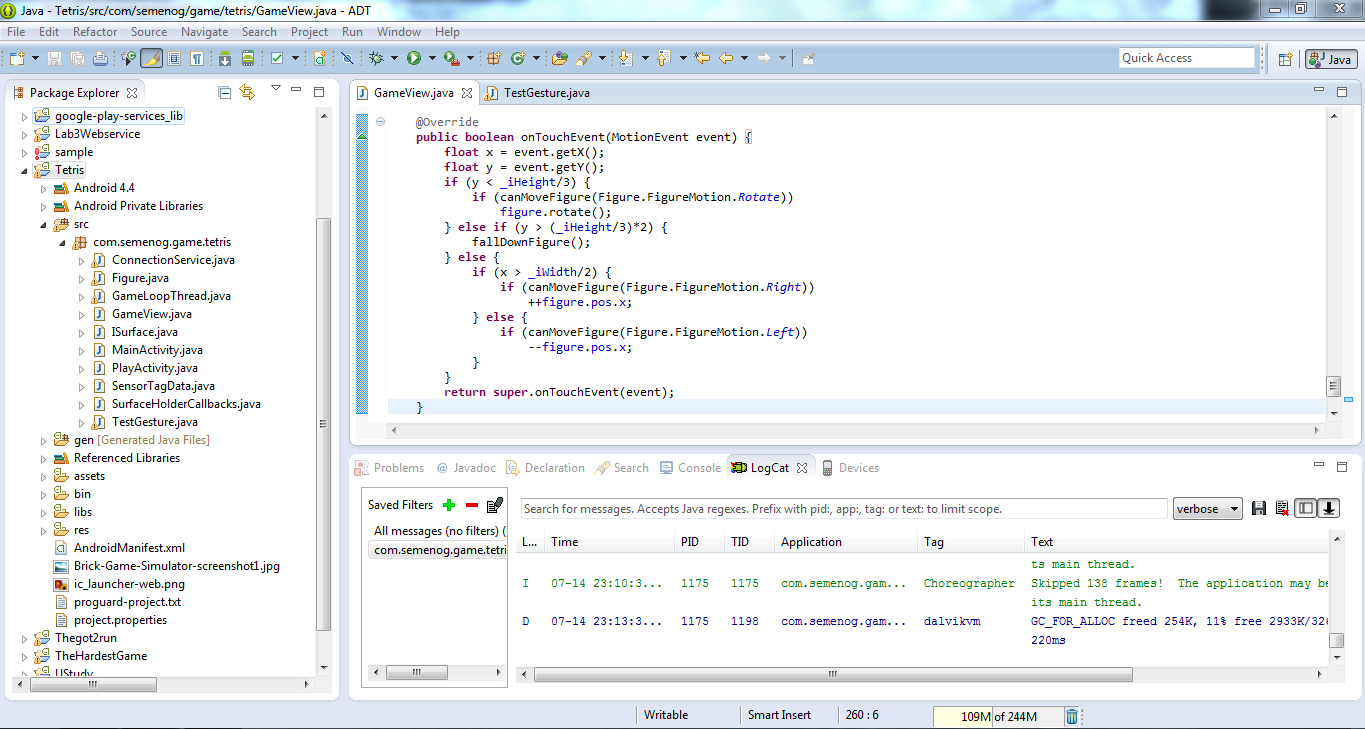
Those above mentioned are the some of the changes done to the source code available.

The below screen shows the function calling the motion method:



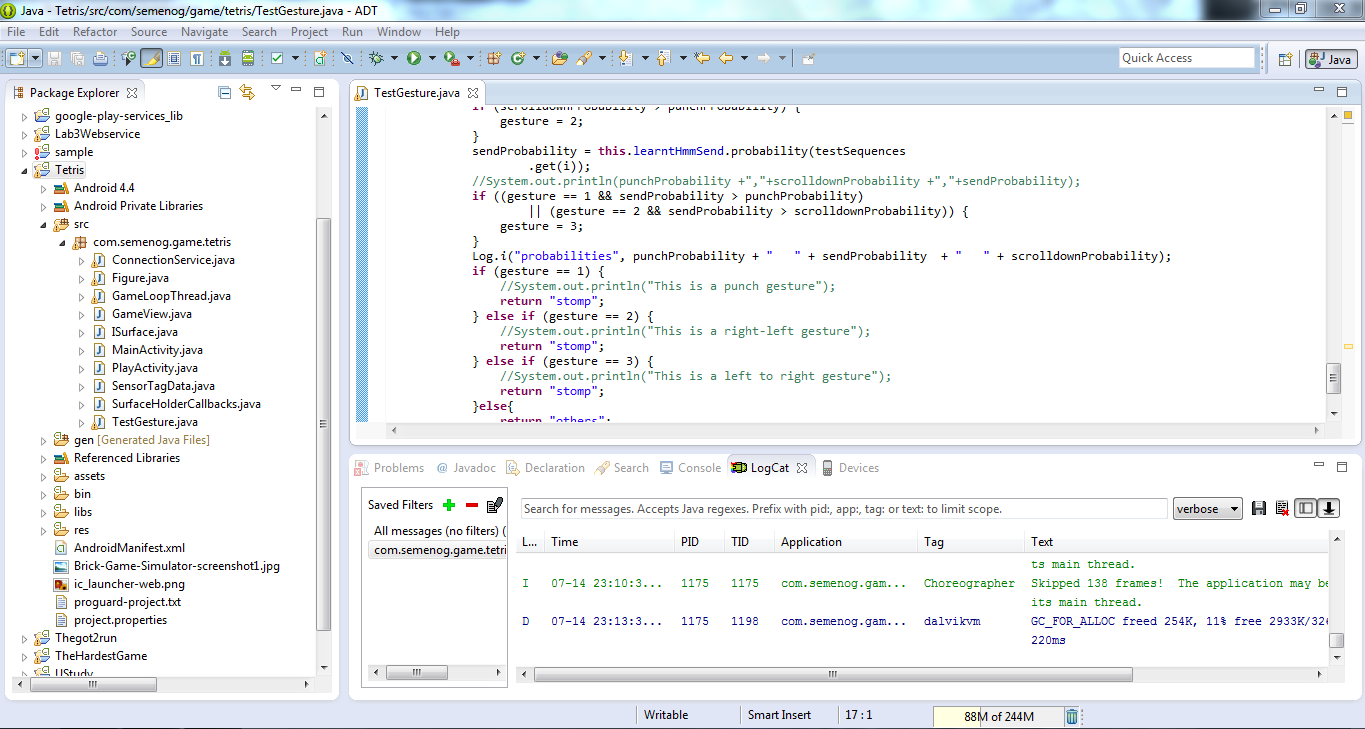
On calling the motion method it detects what motion it is and perform the particular operation of the motion weather to move Left, Right or to rotate.

The below screen shows the **onTouchEvent** method:

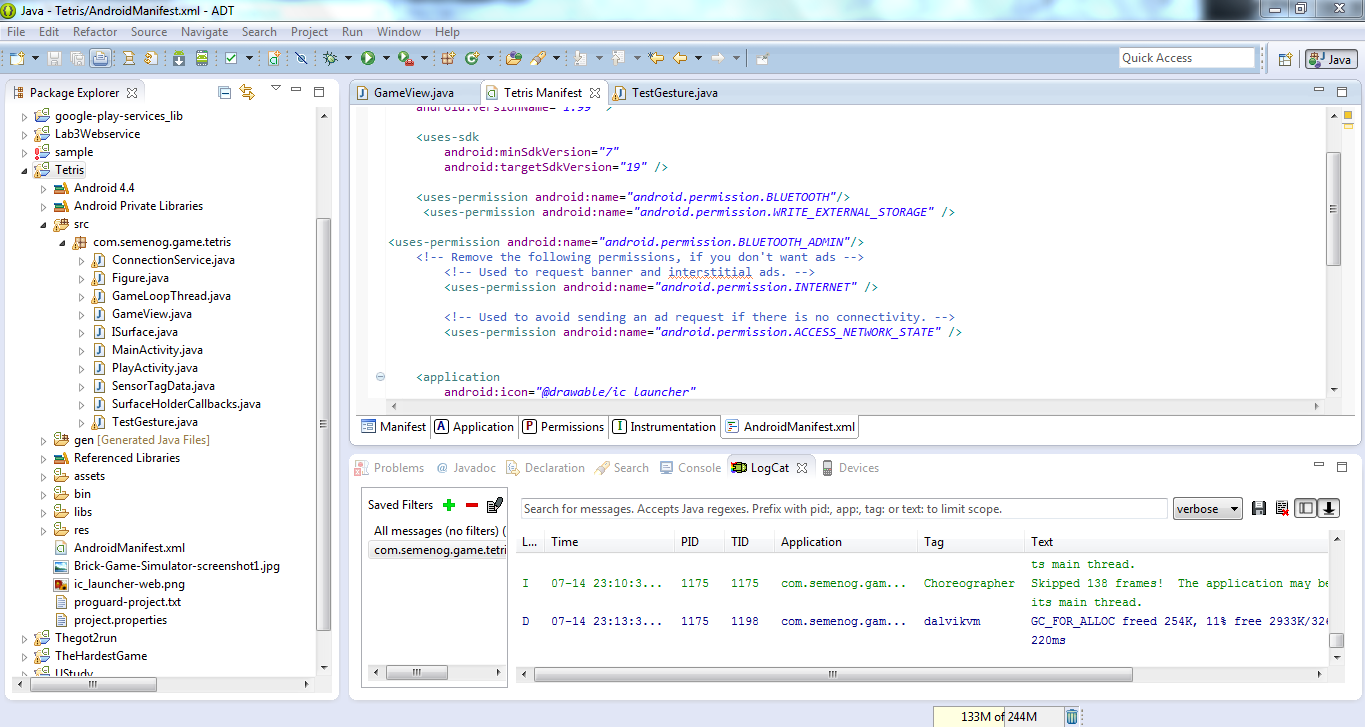


The **onTouchEvent** method checks the condition and position of the object and then based on the motion it moves the object towards the detected motion which may be Right, Left or Rotation.

The below screen shows the probability of the gesture:



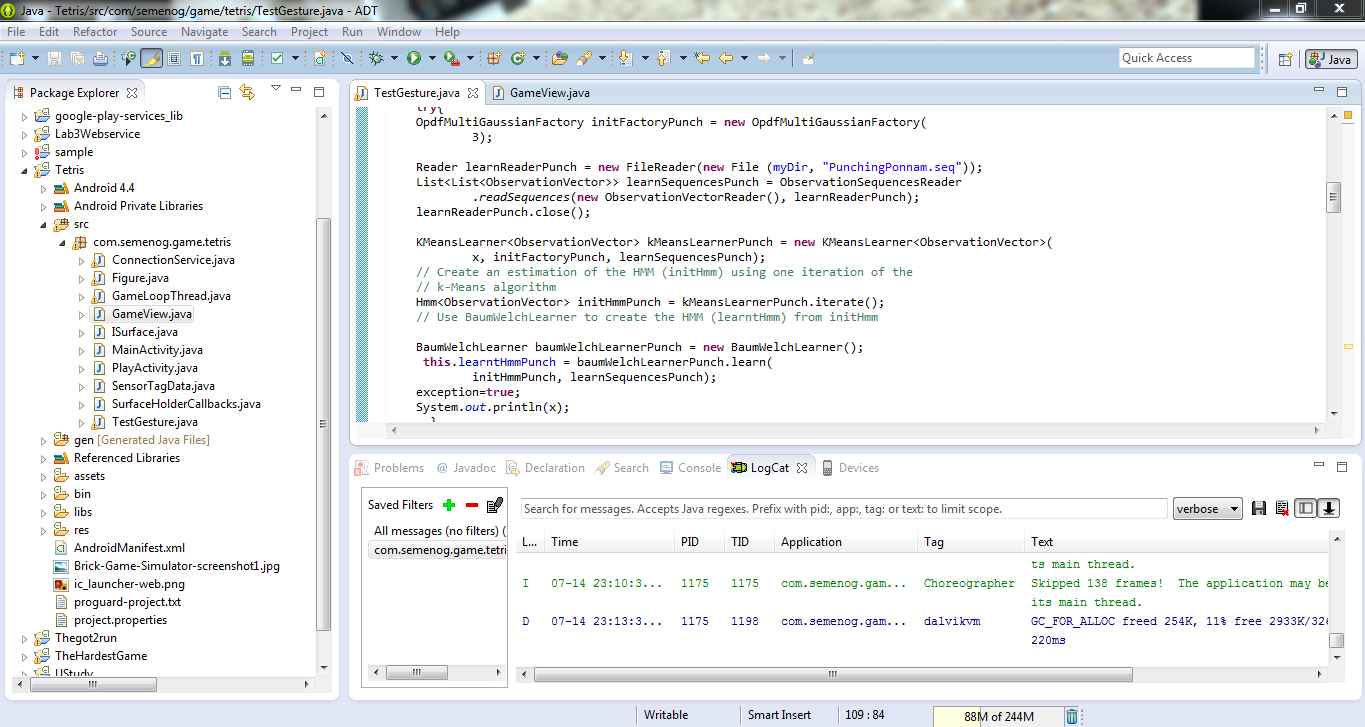
The above code in the screen checks the probability of the gesture and decides the motion to be generated and then calls the particular method for the motion.

The below screen shows the Android manifest file:

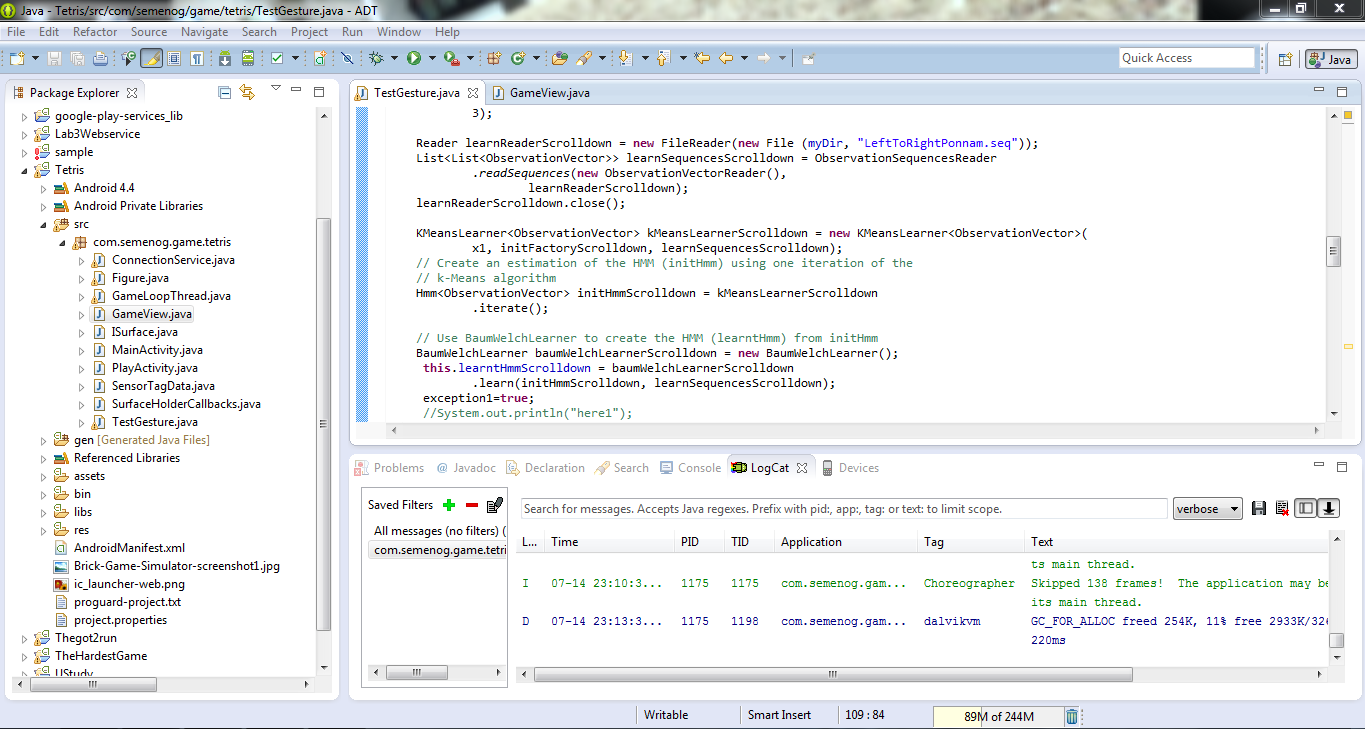
The above screen shows the android manifest file which has added some services and Bluetooth connections.

In the Textgesture class we add the sequential files which are generated. I need three files Left to Right, Right to Left and the rotation for this game. So I have collected those three sequential files and included.

The below screen shot shows the adding rotation generated sequential file to the test gesture class:



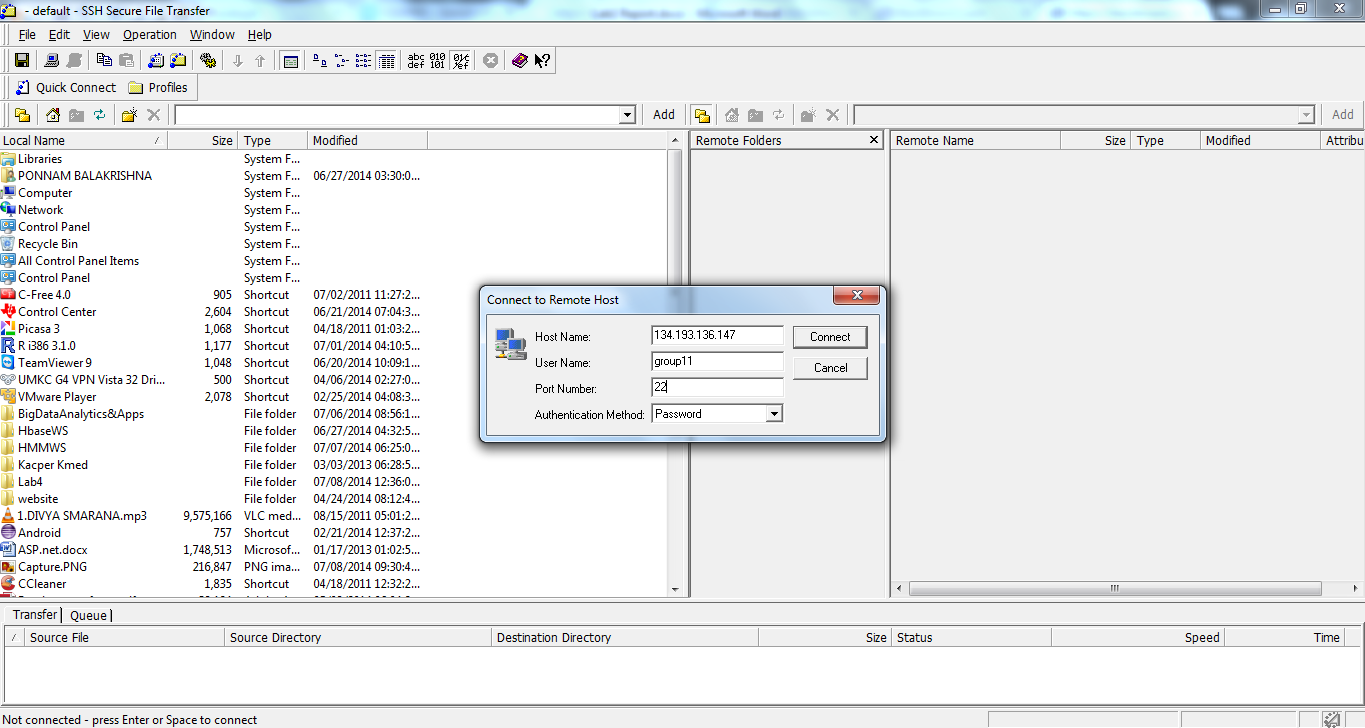
The below screenshot shows the adding Left to right generated sequential file to the class:



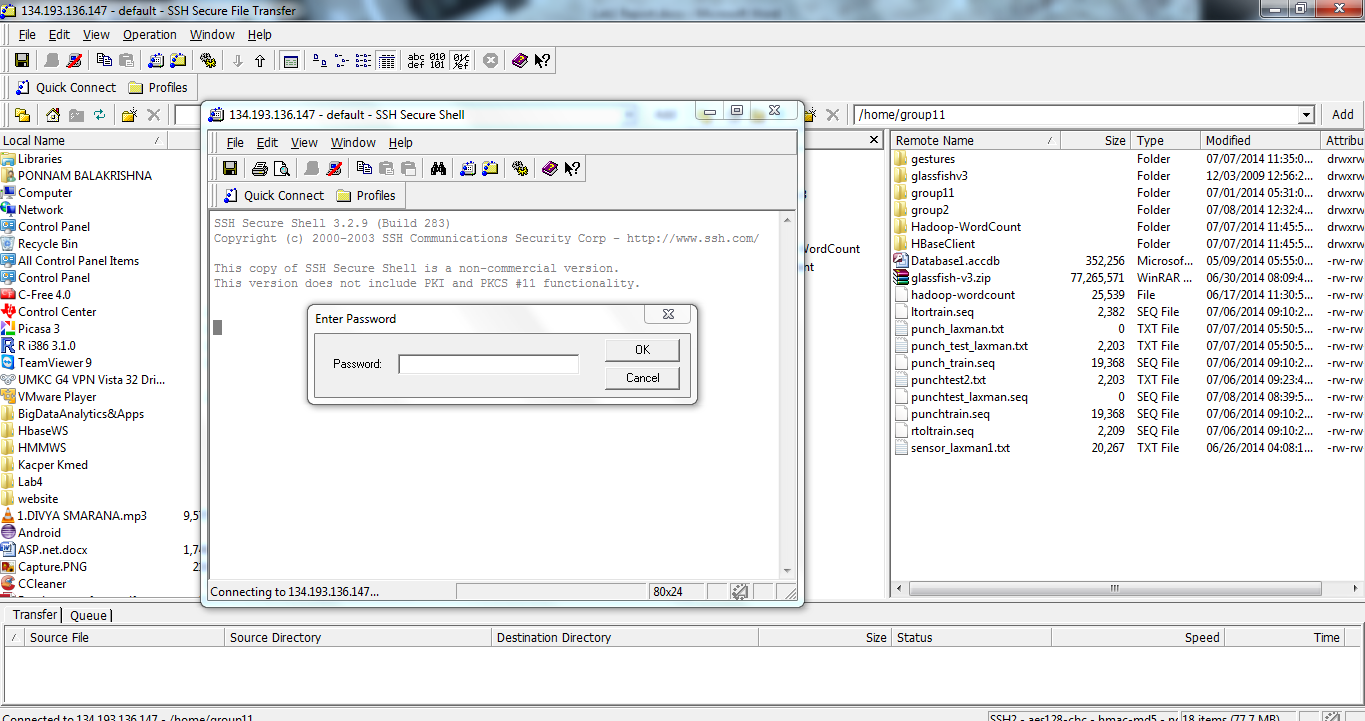
* We collect the data from these applications and then cluster, analyze it and compare them by using the machine learning algorithms to form grouping of the data based on the actions, direction of movements.
* The following screens show the file transfer and retrieval mechanism of the data collected from the application.
* It shows the file is transferred successfully.

For training and testing need to create two files and then need to transfer those files remotely by using the Putty or Secure Shell File Transfer

The below screen short for connecting remotely using secure shell file transfer:

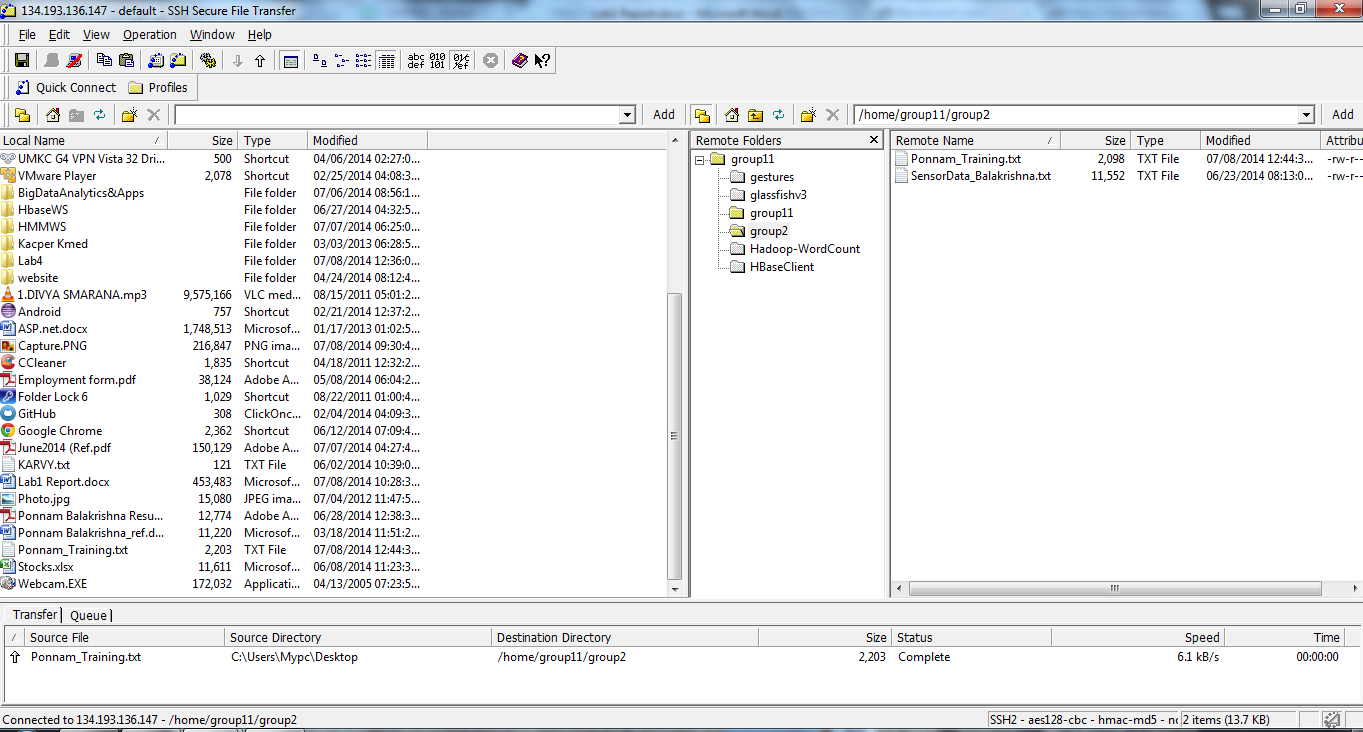


After giving the host name and the group to which we need to connect. Then press connects, and then it asks for the password. After connecting to the host it displays the our space of group remotely



Then drag and drop the files which need to be trained and tested

The below is the screen short after transferring the files:



Then for training and testing access these files and give this path to generate a sequential file

If the data has multiple services then we need perform the operation several times which indicates no of services we have those number of times we need to perform the operations

* Later we retrieve the data and save the file so that the saved data files are used for the training and testing purpose.
* The most important part later is the data preparation part where we prepare two files for testing and training the raw data which are tab separated.
* We transfer these files to the remote machine using the file transfer mechanism to perform further operations.
* **Performing Activity recognition:-**

To perform the Activity recognition patterns on the given data file we first prepare the training and testing data.

Later we use K-means clustering to perform the clusters of the data based on the approximations.

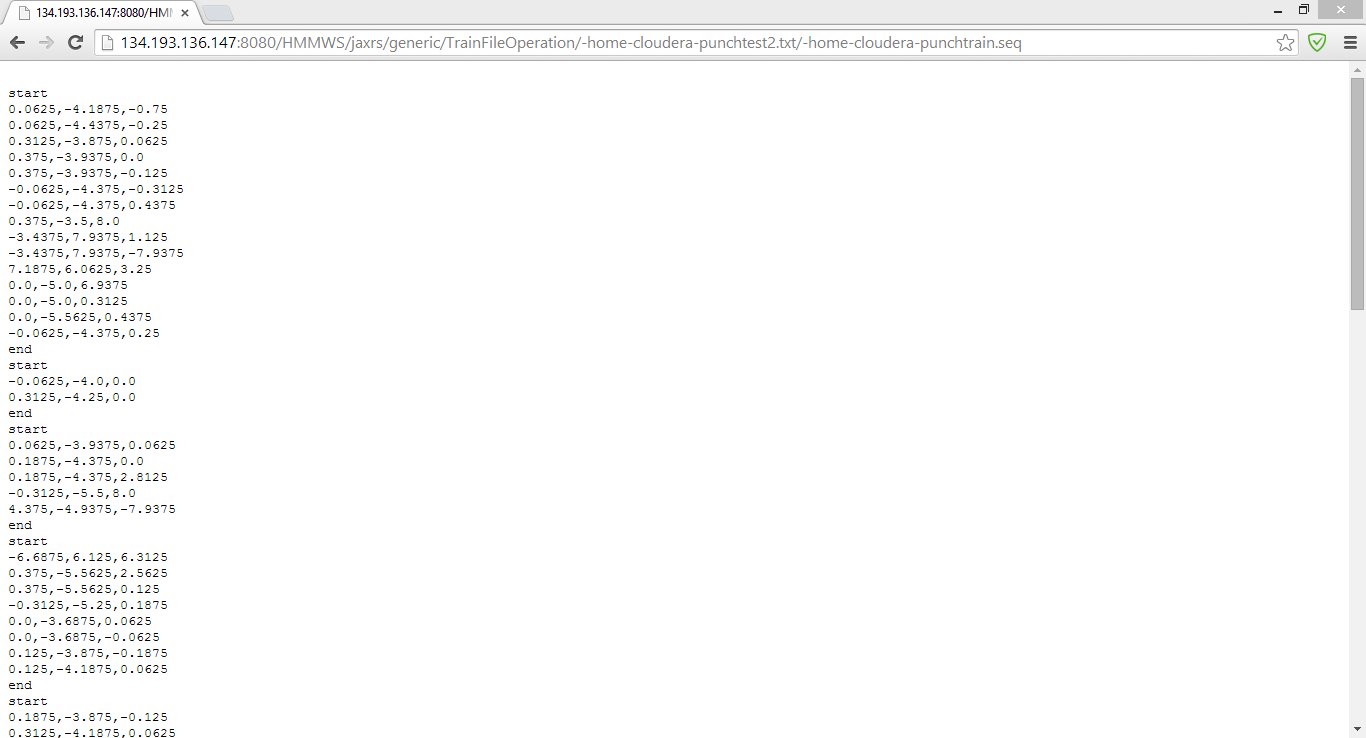
Basically we use three methods in the service they are

1. Training file operations
2. Testing file operations
3. Training and testing file operations

We provide the raw data as the input file for the generation of the output sequence file as follows.

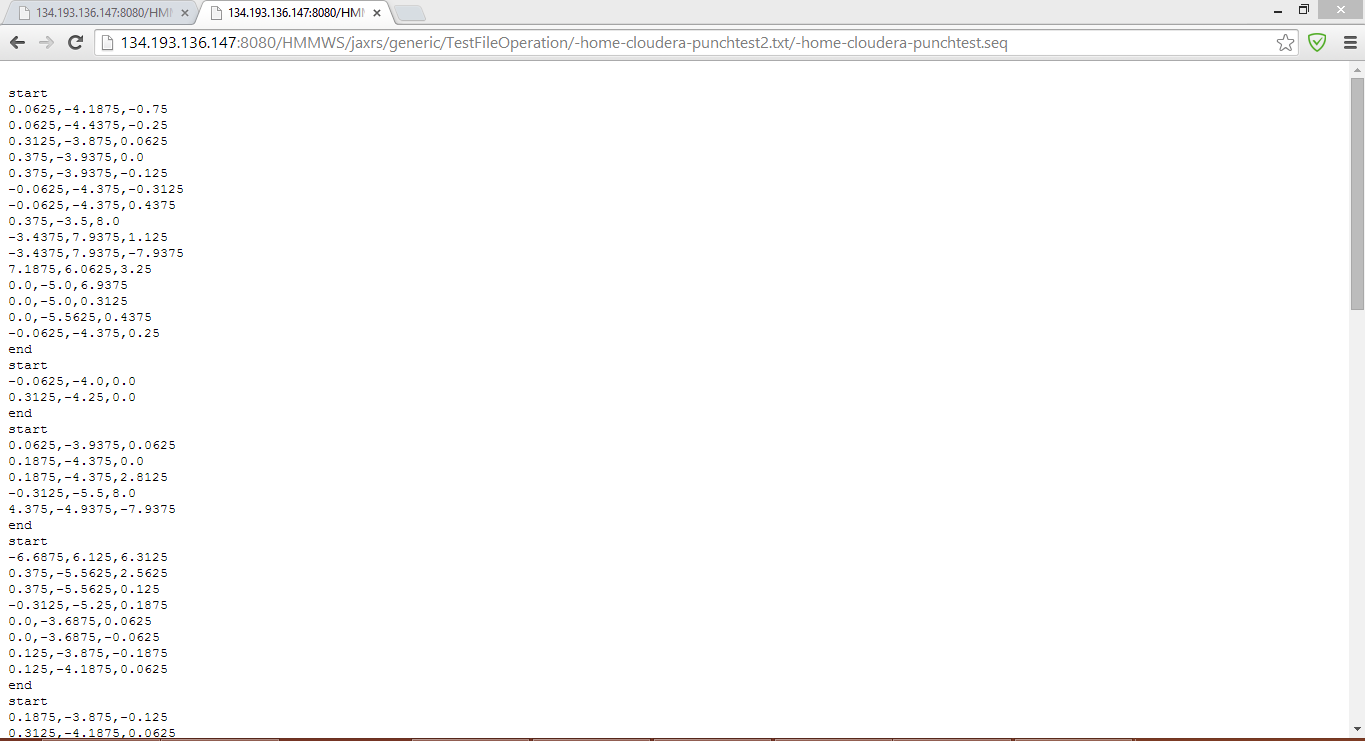
The below shows the performed training file operation for activity recognition.

We should train the system if there are more number of actions present.

****

Later second operation is the generation of the output sequence file from the raw data as input.

The below screen shows the generated file after performing the testing file operations.

****

Later we perform both the training and testing operations as input sequence files to get the required output file.

The data trained is the combination of all the actions for the generation of the sequence files.

**Project Management:**

All the task planning according to the iterations are maintained in scrumdo and task allocation is shared equally among the team members.

<https://www.scrumdo.com/organization/umkc94/dashboard>

* **Fourth Increment:**

The tasks that will be included for fourth increment are:

1. Modify the existing features of game according to the application development.
2. Enhancements for connection between the available open source game and sensor.
3. Implementation of graphical reports for analyzed data
4. Activity and motion recognition based on the available data.
5. Final testing and debugging
6. Video production and document production.

The above mentioned tasks will be uploaded the scrumdo tool with specified timelines.