**CS5590BD – BIG DATA ANALYTICS**

**INCREMENT 3 REPORT (GROUP 8)**

**THE MAZE GAME**

**INTRODUCTION:**

‘The maze’ is an android mobile game that uses gesture recognition to guide a ball in a maze. Traditionally,   users guide the ball through touch, but in our app we will be able to do it by gesture recognition using a sensor tag. We will integrate a sensor tag to the existing game, and collect the data from its movements in all direction which will be in the form of 3d plane coordinates. This data is then sent to hadoop in the backend and is used to process coordinates. Depending on the coordinate points which are in reference to point of origin, we will guide the ball in maze.

**PROJECT GOAL & OBJECTIVES:**

We have selected a maze game in which the users need to direct the ball through touch and tap gestures in a maze. Users can direct the ball in four directions right, left, up and down. Our project’s main aim is to implement the same touch functionalities to guide the ball through gesture recognition. Using the sensor tag, we will be guiding the ball in the required direction to complete the game. In addition to these four gestures, we are also planning to implement another gesture either to start a game or to change the background color.

**SIGNIFICANCE:**

Our gesture controlled game is moving technology ahead to integrate gesture recognition to mobile gaming. Integrating big data and hadoop in the backend, we will be processing the data collected using sensor tag and depending on the type of gesture we can add functionalities and perform actions accordingly. This is just the beginning of implementing gestures but in the future a lot more can be done using them.

**ACTIVITY RECOGNITION SCENARIO AND DATA COLLECTION:**

In our maze game we move the ball to the target with the help of gestures. When the user uses the gestures the sensor will detect the gestures. The sensor will be trained beforehand about the gestures. We collect the data and analyze it and after analyzing we give some good suggestions to the users so that they can improve their way of playing.

Devices/Sensors:

1) Sensor Tag:

It is an excellent development kit when used along with a smart phone. It is an inspiring design for various other accessories of the smart phone. It is very useful to the smart phone app developers.It has the sensors for magnetometer, pressure, gyroscope, humidity, accelerometer, temperature. In total 6 sensors available.

2) Chronos Watch:

It is a profound integrated device with a wireless development system. It has a motion sensitive control accelerometer and a pressure sensor along with LCD display .It is a wireless system which is hub by surrounded wireless sensors. We can reprogram it as we want by disassembling it.

3) Android Device:

To connect the sensor tag we need an android device which has 4.3 or above OS.And it should have Bluetooth 4.0 compatible android device.

**Data Collection:**

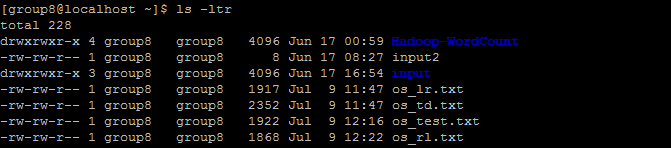
Data is collected for three types of gestures i.e. right to left, left to right and top to bottom gestures.

os\_lr.txt: left to right gesture data is stored in this file.

os\_rl.txt: right to left gesture is stored in this file.

os\_td.txt: top to bottom gesture is stored in this file.

os\_test.txt: It contains combination of left to right, right to left and top to bottom gestures.



Upon executing below URL’s three training sequences files are generated using above 3 text files.

<http://134.193.136.147:8080/HMMWS/jaxrs/generic/TrainFileOperation/-home-group8-os_lr.txt/-home-group8-os_lr_1.seq>

<http://134.193.136.147:8080/HMMWS/jaxrs/generic/TrainFileOperation/-home-group8-os_td.txt/-home-group8-os_td_1.seq>

<http://134.193.136.147:8080/HMMWS/jaxrs/generic/TrainFileOperation/-home-group8-os_rl.txt/-home-group8-os_rl_1.seq>

<http://134.193.136.147:8080/HMMWS/jaxrs/generic/TestFileOperation/-home-group8-os_test.txt/-home-group8-os_test_1.seq>

<http://134.193.136.147:8080/HMMWS/jaxrs/generic/HMMTrainingTestThree/-home-group8-os_lr_1.seq/-home-group8-os_td_1.seq/-home-group8-os_rl_1.seq/-home-group8-os_test_1.seq>

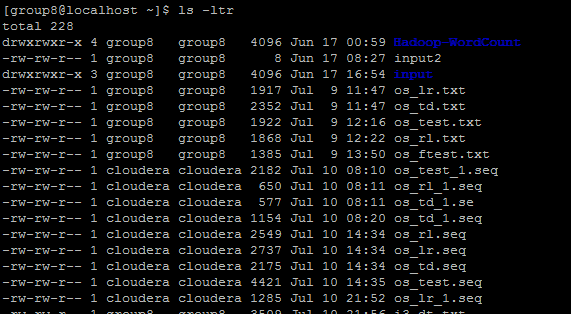
Below sequence files are generated upon running above URL’s.

os\_lr\_1.seq: left to right gesture sequence file.

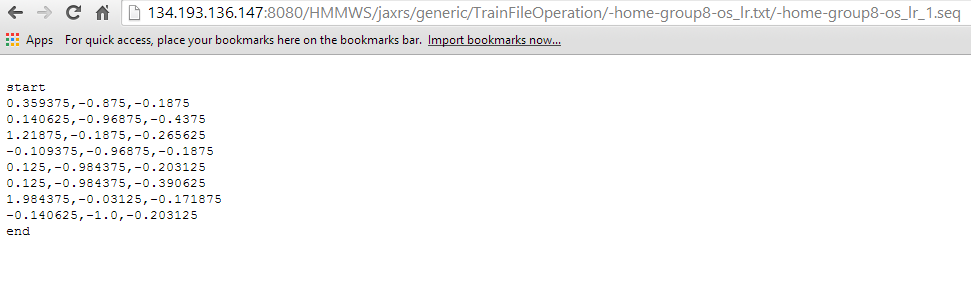
os\_rl\_1.seq: right to left gesture sequence file.

os\_td\_1.seq: top to bottom gesture sequence file.

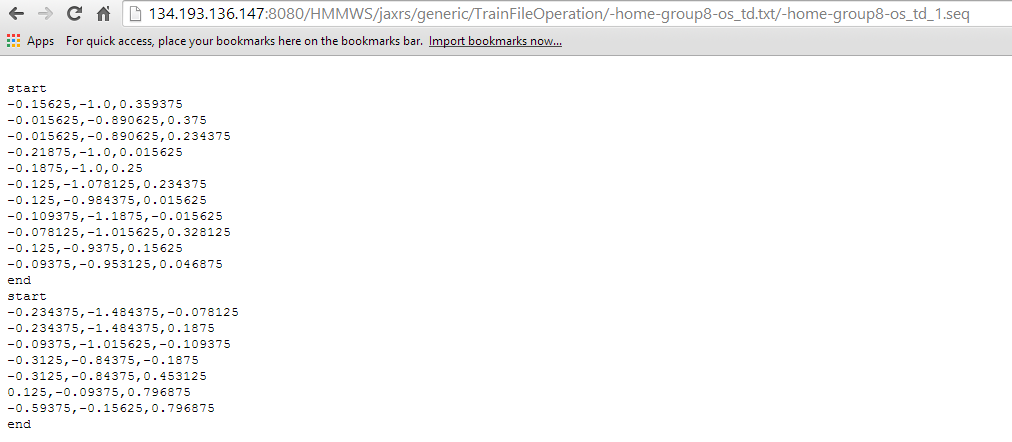
os\_test\_1.seq: combination of all gestures.



Left to Right Gesture Recognition: Train file



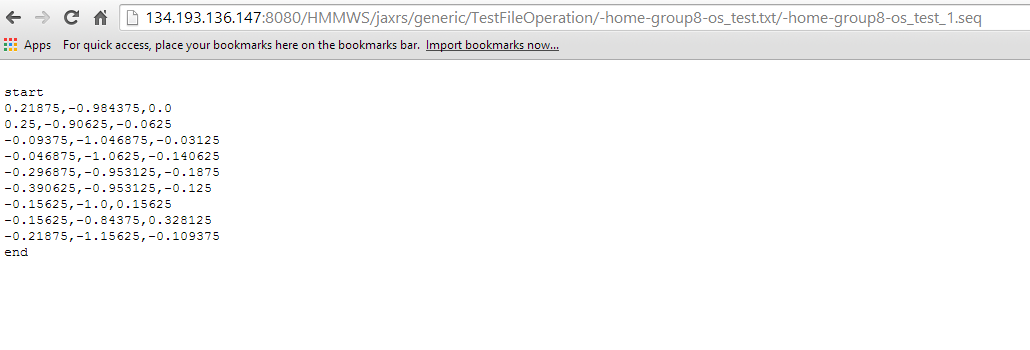
Top to bottom gesture Recognition: Train file



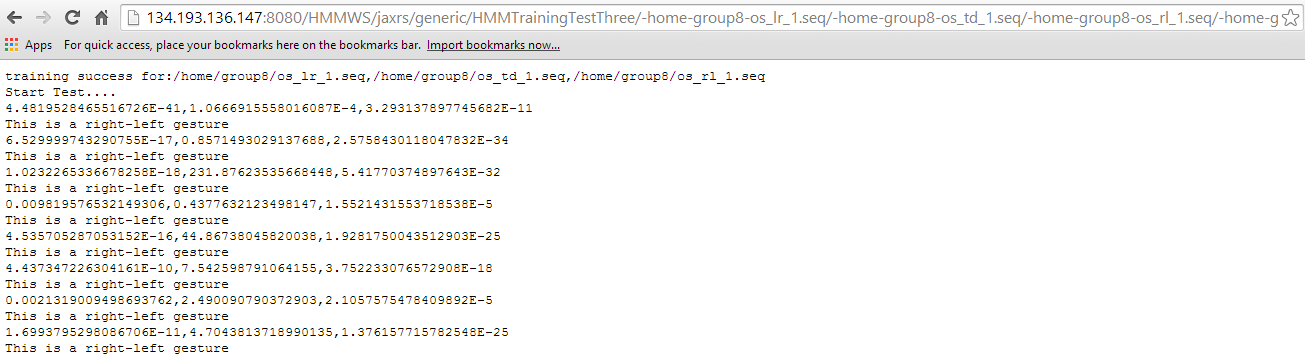
Right to Left gesture Recognition: Train file

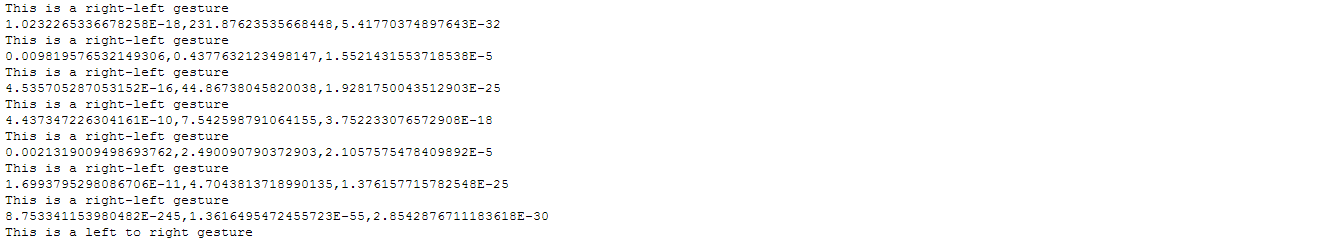


Combination Test file:



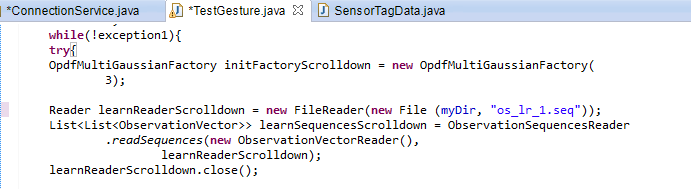
Final right to left and left to right gestures are recognized:





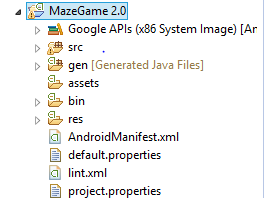
**Analytical Tasks:**

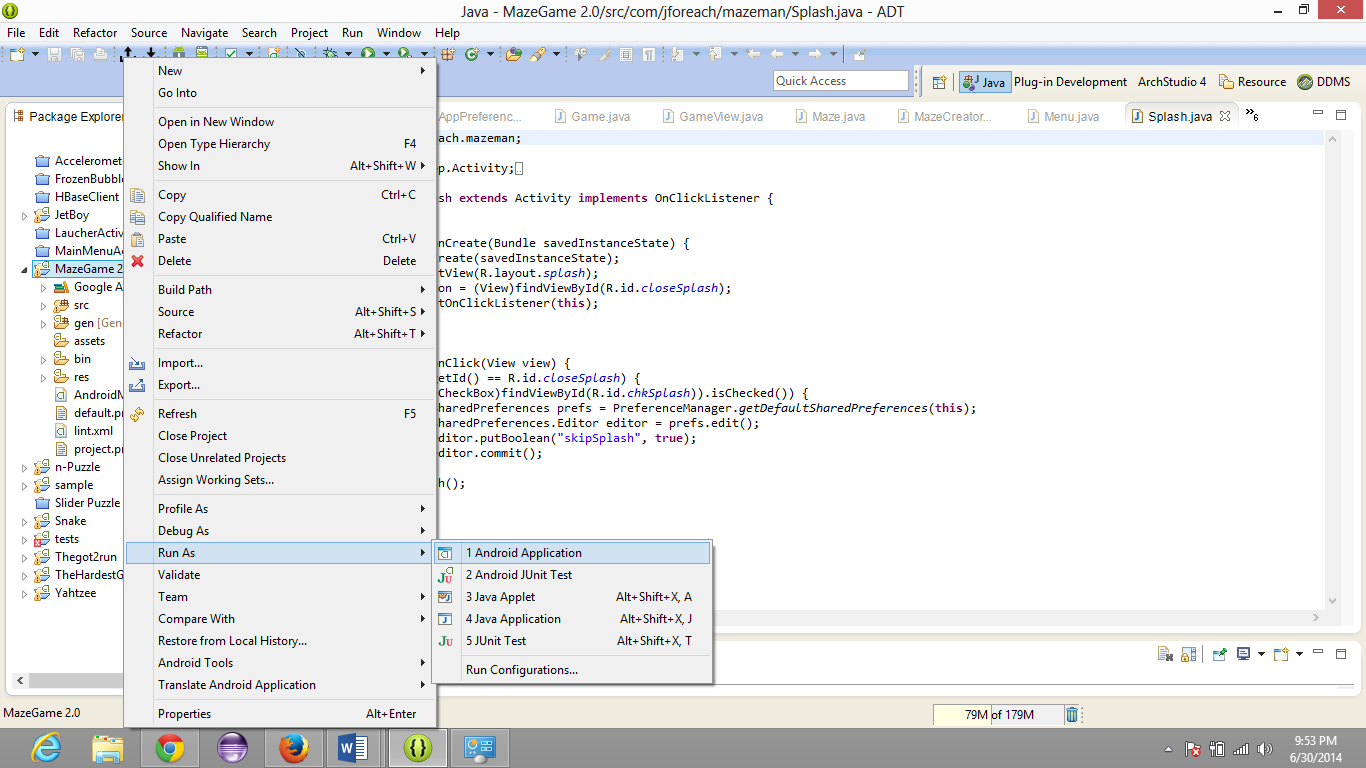
Our generated sequence files are then placed in connection service file of Maze game. These files are given as input to three different gestures model and signal sent to broadcast signal. Broadcast receiver will then receive these signals and the methods are called.



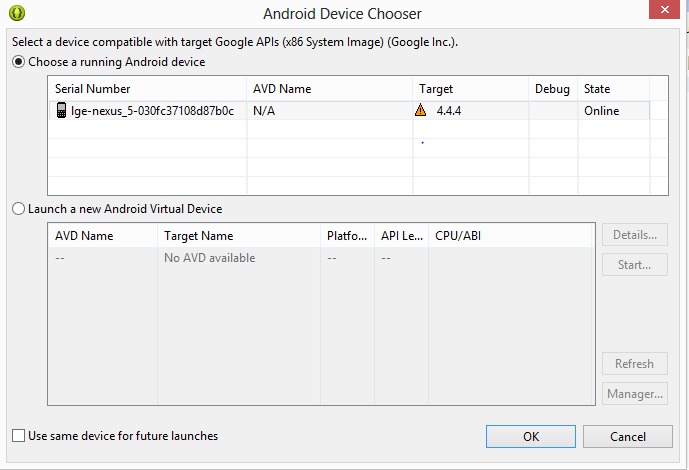
**Design of mobile client:**

Initial step is to import the chossen project into the eclipse workbench. Next is resolving issues if any ( setting projectc properties,installing google API, updating device driver’s software, etc.) Below is the screen shot of the project choosen “MazeGame 2.0” .

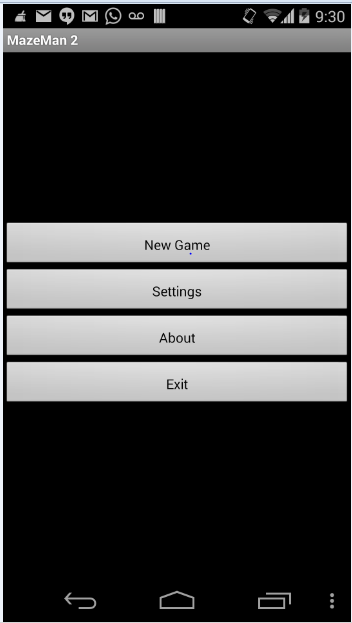




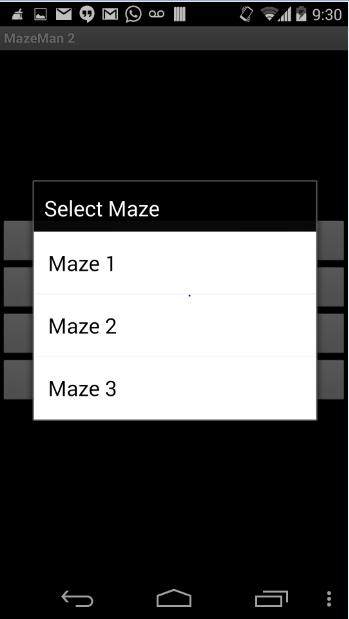
Running project as an android application on an android device(Nexsus 5 )



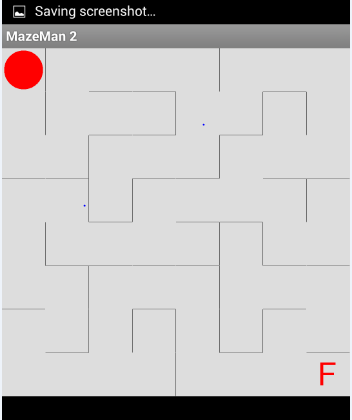
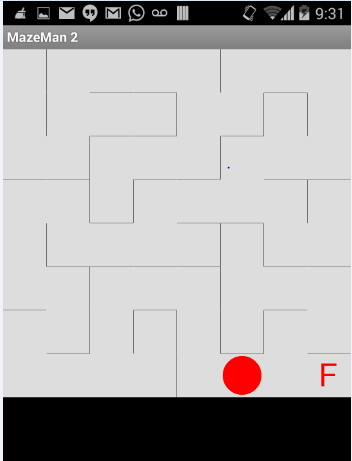
Below is a screen shot of the main screen of the game. “New Game, Setting, About, Exit.”



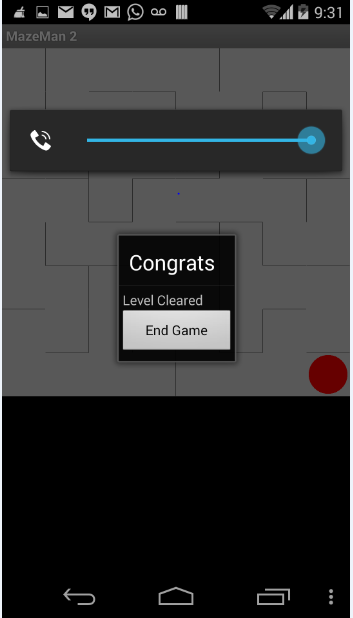
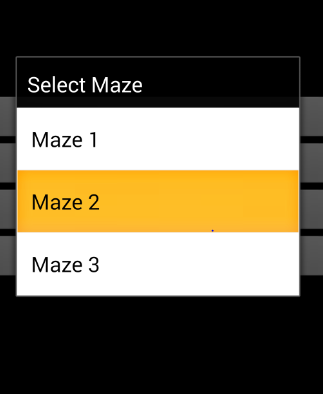
Choosing New Game Below pop up pops up. It looks as follows.



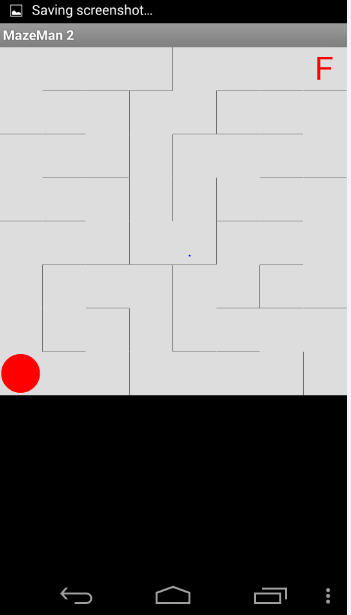
Real game screen is with a red ball which starts from a source and player has to take it to the destination “F”

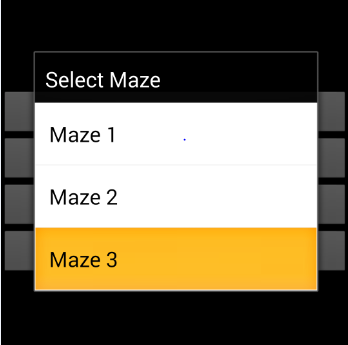
Clicking on End game ends the game level of maze 1.

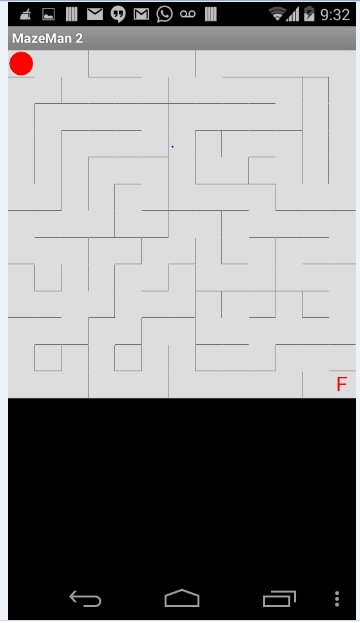
 

Above is the screen of three maze level available.

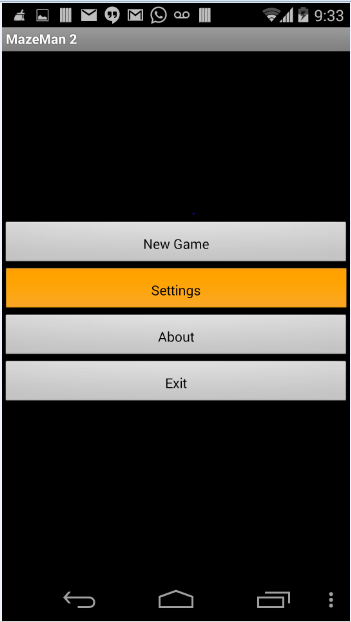


This is how Maze 2 looks like below.

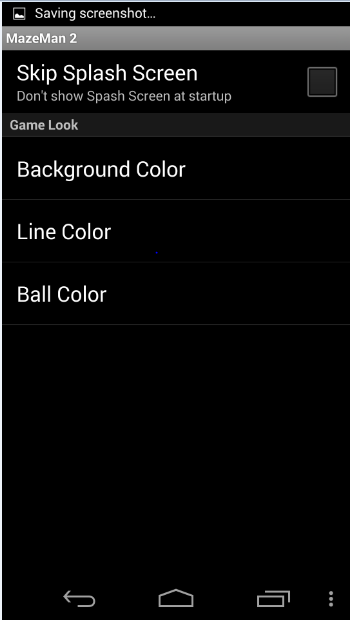
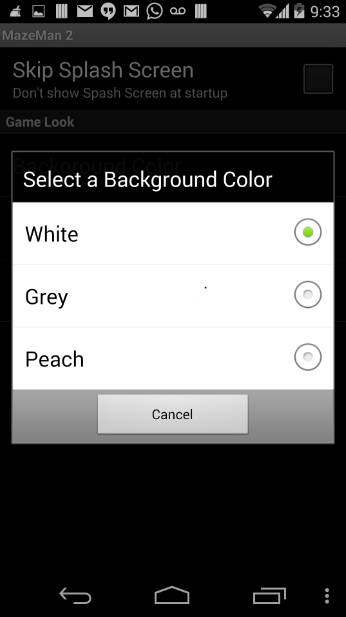


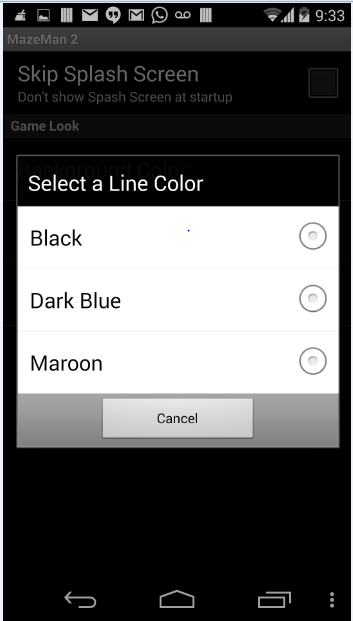
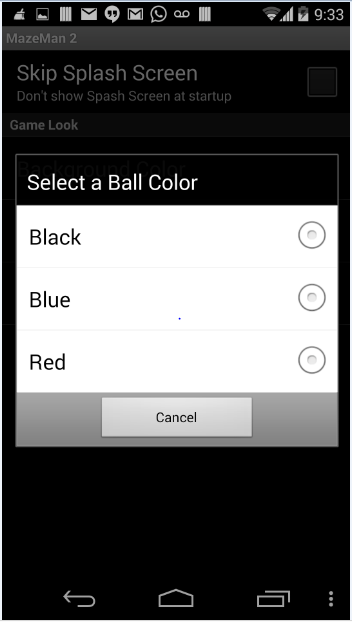


Every game has some options. And this game has the following options.



On clicking settings button, setting options are displayed and they can be modified as needed.

**Design and implementation of Big Data Analytics:**

Motion/Activity Models:

-Here we first filter the data.

-Then quantize the data as it is vector data.

-Then we send the data through Hidden markov model.

-And then use the Bayes theorem to classify it.

TRAINING GESTURES

FIND [X,Y,Z] FROM SEQ FILE SEGMENT🡪SEQUENCE🡪TEXT🡪(RIGHT,LEFT,DOWN,UP MODELS)

BROADCAST “SEND” TO GAME MODULE

“RECEIVER”

IF ACTION==RIGHT MOVE RIGHT

IF ACTION==LEFT MOVE LEFT

IF ACTION==DOWN MOVE DOWN

IF ACTION==UP MOVE UP

IF ACTION==RIGHT MOVE RIGHT

IF ACTION==RIGHT MOVE RIGHT

IF ACTION==RIGHT MOVE RIGHT

Data Filtering:

Here we first pass our application through idle state and directional equivalence filters. The first filter removes the deviating gestures where the second one removes the repeated sampling points.

Model Used:

We use Hidden markov model to recognize the gesture.

Evaluation Model:

We took the training sets of the five gestures we are using

a)top

b)bottom

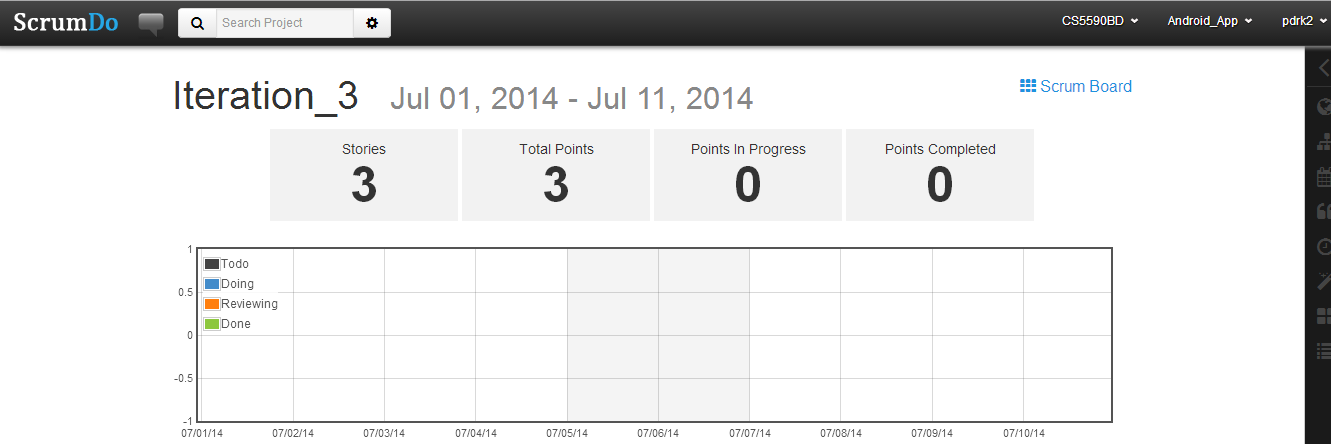
c)left

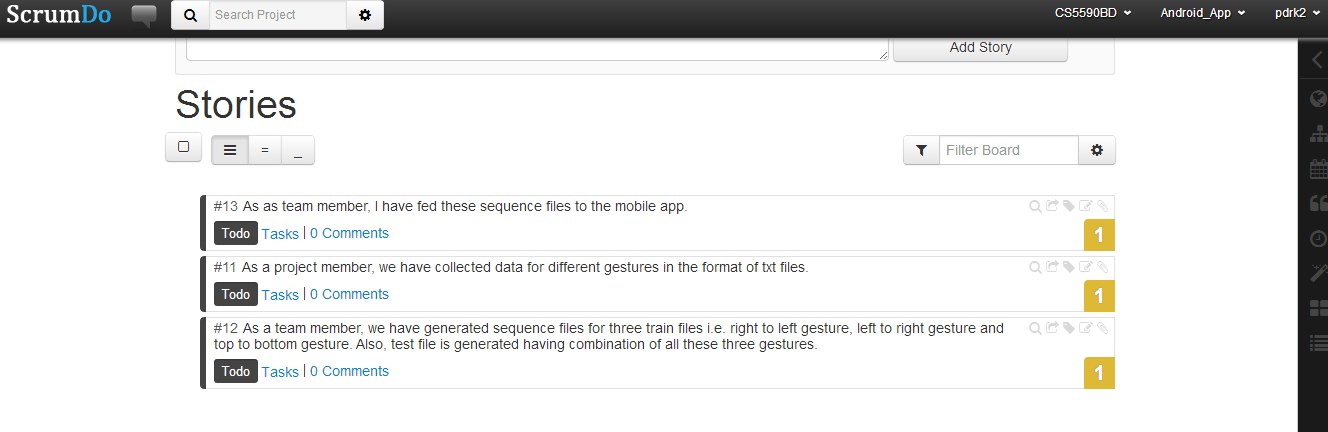
d)right

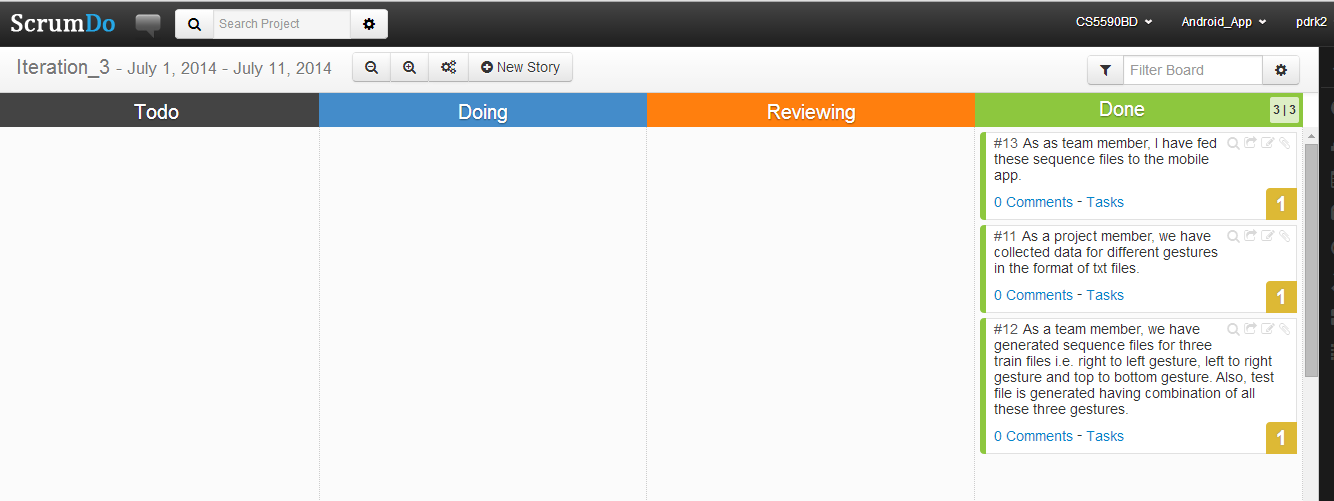
e)strat

Using sensor tag and the text files are generated and after that the gestures were recognized.

**Srumdo Work:**

****

****

****

**Related Work:**

Magic Stone:

It is an application that uses sensor tag. It takes accelerometer information as input and using that information a stone will be moving in the same way as the accelerometer moves. The input values of accelerometer are stored and analyzed.

**Bibliography**:

<http://www.androidmix.com/android-game-programming-tutorial-mazeman>

<http://developer.android.com/guide/topics/sensors/sensors_overview.html>

<http://www.codeproject.com/KB/android/>

<http://www.sourcecodester.com/android/6230/simple-game-android.html3>