**CS5590BD BIG DATA ANALYTICS**

**FINAL PROJECT SUBMISSION**

**MAZE GAME (GROUP 8)**

**GROUP MEMBERS:**

ANUDEEP VATTIPALLI

PRATHYUSHA DINNE

RUPESH UTUKURI

SINDHU KONERU

SUJITHA ONTERU

**INCREMENT -1 PROJECT REPORT**

**PROJECT GOAL AND OBJECTIVES**

***INTRODUCTION:***

Best bud is a mobile app that provides its user quick feed on their workout statistics through interactive graphs, calorie burn meter, work out pro levels and many more. It provides them the essential joy in working out by earning themselves awards and to also challenge others. By allowing other user to join in their workouts sessions, users will be able to compare with each other thus creating a competitive environment and give them the essential boost to do more.

Other than the above mentioned features, we are including some more additional functionality into our app. User will be able to provide their GPS location and using it, app will prompt happening events in and around user’s current location.

As we are still in the process of requirement gathering and playing around with sensor, if we can find any other useful feature, we will try to implement it into it.

***OBJECTIVES:***

Best bud’s main objective is to provide its user’s a continuous update on their fitness levels. It is a natural habit for people to easily give up on one’s fitness but with our app it makes its users connected and become fitter and healthier day by day. It also helps its user to constantly increase their fitness levels interactively and challenge other group users and grow collectively.

***SIGNIFICANCE:***

Our app keeps collecting data from user’s workout (jogging, running and walking) activities and prompts details of it to encourage the user to do more. It keeps people close to him always connected and help them shape themselves to become more healthy and fit.

**ACTIVITY RECOGNITION SCENARIO AND DATA COLLECTION**

Devices/Sensors being used:

**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:**

We use Apache Hbase to access our data. We first collect the data .We record it into a file. We will read the file and push data to Apache HBase. We use a java program to read local file and push data to hbase.

**DESGIN OF MOBILE CLIENT**

Below is the screenshot of home page in our app. User needs to login with his credentials for logging in and to check his details.

***FEATURES:***

**Event Finder:** We have included this functionality to track users GPS location through the sensor tag. Using this, we will be looking for happening events in and around his location and prompt the user. Additionally, we will include user to mark his event so as to store the event in the DB and remind him before the event and notify him just like an alarm.

**Work Stats:** Using the sensor tag, while workout (running, jogging and walking) we will keep tracking user location continuously by tracking through GPS and store the data. In the backend we will process all this data and use them to calculate calories burnt in a session. We will be keeping all this data for a week or month and plot graphs using amount of calories burnt per each day over a week or month. And also, user can compare with other users he/she is connected to and look into performance levels of each user. Apart from this, all the group user details are collected including the phone numbers. So when a user needs to talk to one of the group member while looking at quick work stats, he just need to click on call button right from the app instead of going out of the app and searching for them in the contact list

**Awards:** As previously said, users will be able to join groups with few members and constantly keep an eye over their calories burnt in regular session. As all the group users data is collected, every week/month, an award or a medal of honor will be presented to the best performer. This keeps the users much more excited about their fitness and makes them to strive more in building themselves stronger, fitter and healthier.

**Videos:**  We have a section in best bud where we include motivational and training videos pertaining to their workout. Users can view apps on the go.

**Game:** Apart from the above mentioned features, we have decided to make our application more interesting by including a game which captures user’s movement to control the game using a sensor tag. This is one additional feature which makes our app more reachable to users.

**Weather:** By gathering users GPS coordinates, we will be prompting the user of weather conditions in his location. This way when going out for a workout, if the weather does not suite for work out scenarios, user can be aware of it.

**Friends:** This feature is provided to add, delete users into existing groups.

***TECHNOLOGIES USED:***

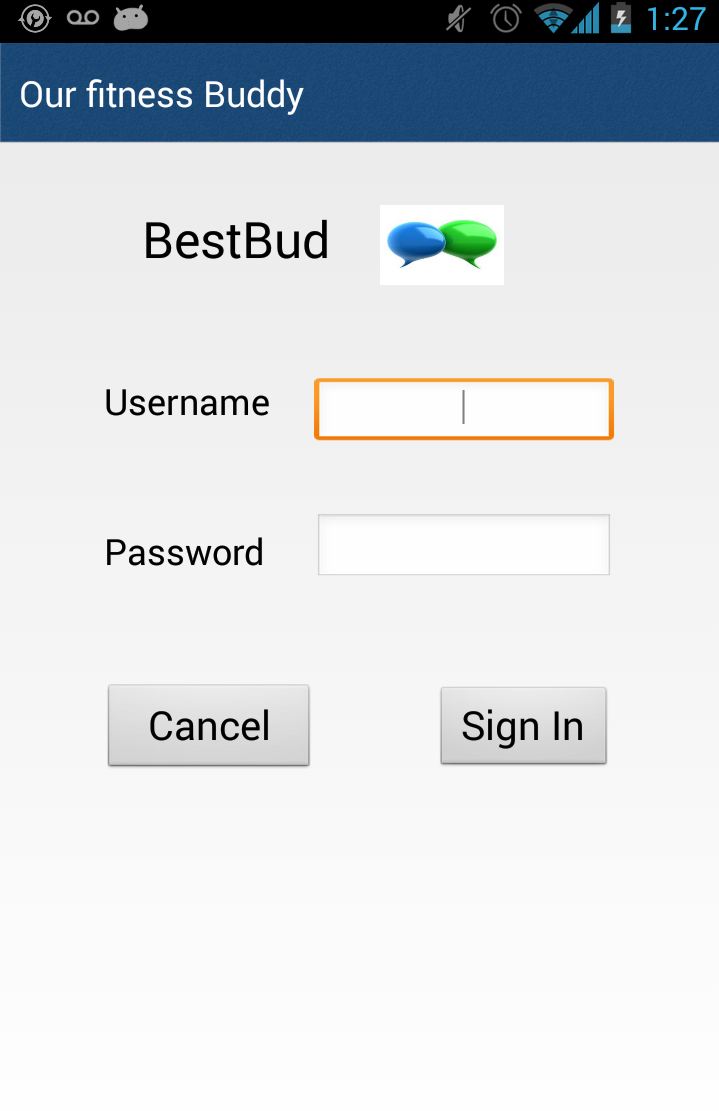
* HBase Database
* Java
* Android Platform
* Hadoop & Cloudera

**SCREEN SHOTS:**

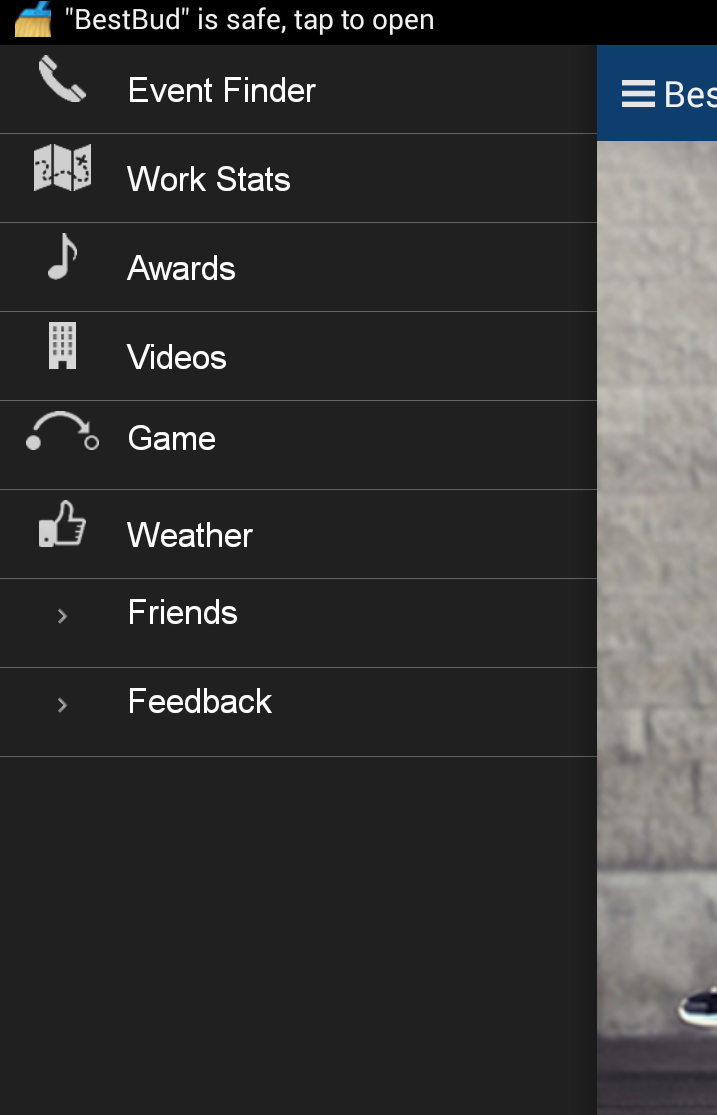
***HOME PAGE:***



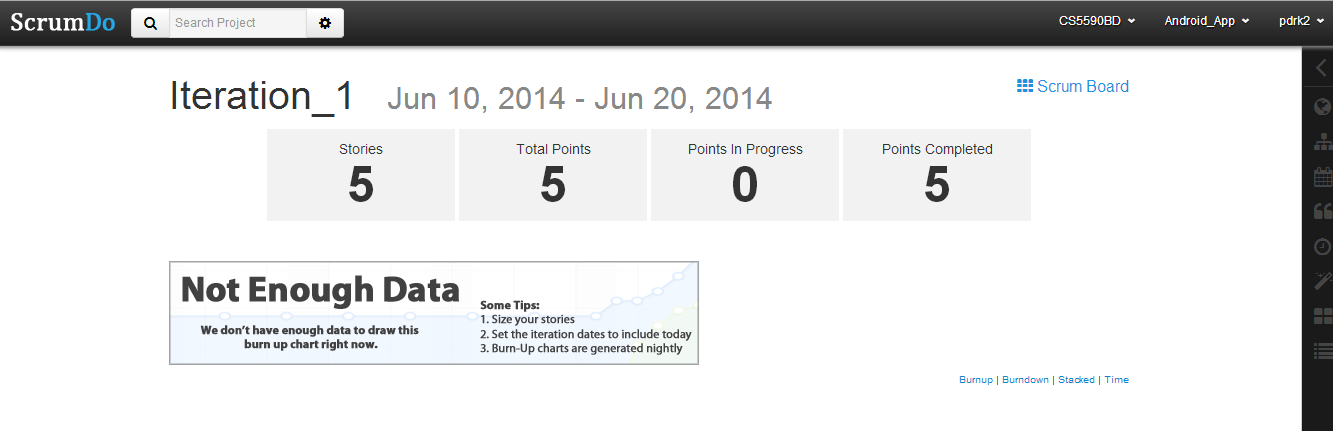
***LOGIN PAGE:***

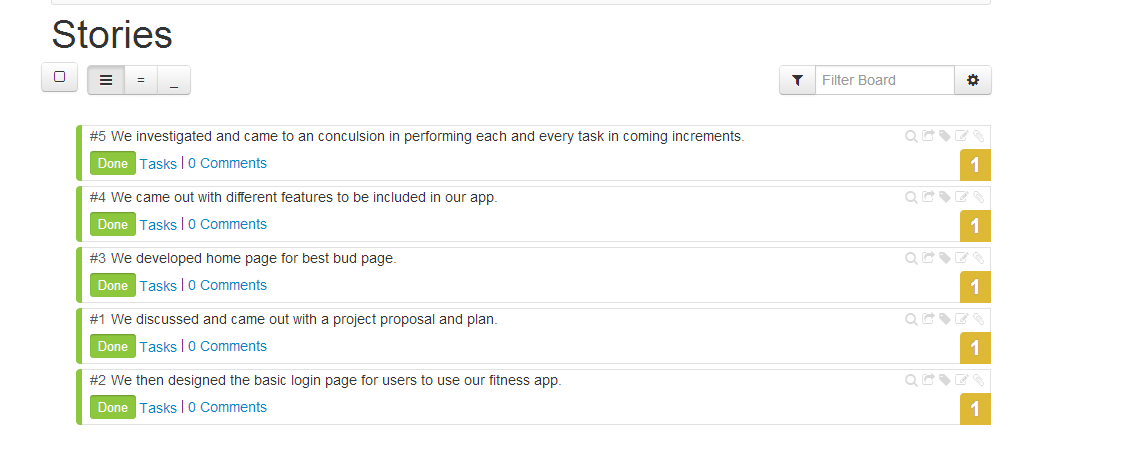


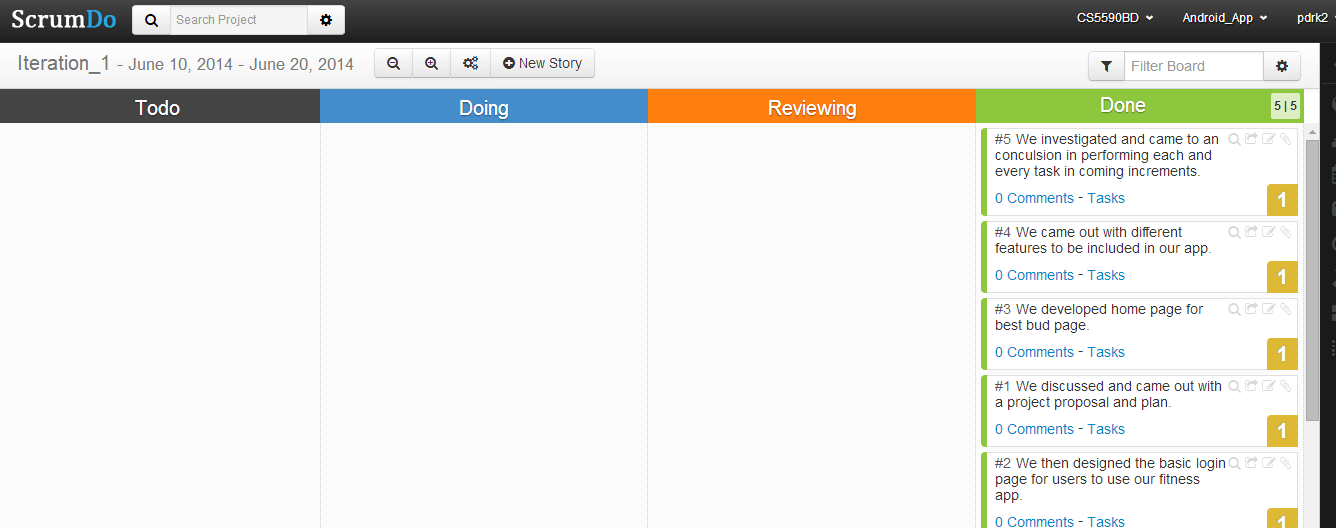
**FEATURES INCLUDED IN OUR APP:**



**PROJECT PLANNING WITH SCRUMDO:**

****

****



**INCREMENT 2**

**Thru the maze**

**INTRODUCTION:**

‘Thru 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:

We first collect some training data for each gesture.

By using HBase we store the data.

A model will be developed for the recognition of the gestures.

When the user applies the gesture using the sensor the values will be collected from the sensor tag then this data is also pushed into the HBase.

Now we analyze the data using some mapreduce technique and get the results.

Motion/Activity Model:

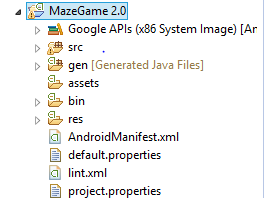
The application is executed in a device which has android OS 4.3 or above and a Bluetooth of 4.0 .The sensor tag will be in the user’s hand. When the game is started the user performs required gestures using the sensor tag to play the game. The gestures are recognized and the game is progressed. The whole data is analyzed and after completion of the game the user is given suggestions for improvement of his playing skills.

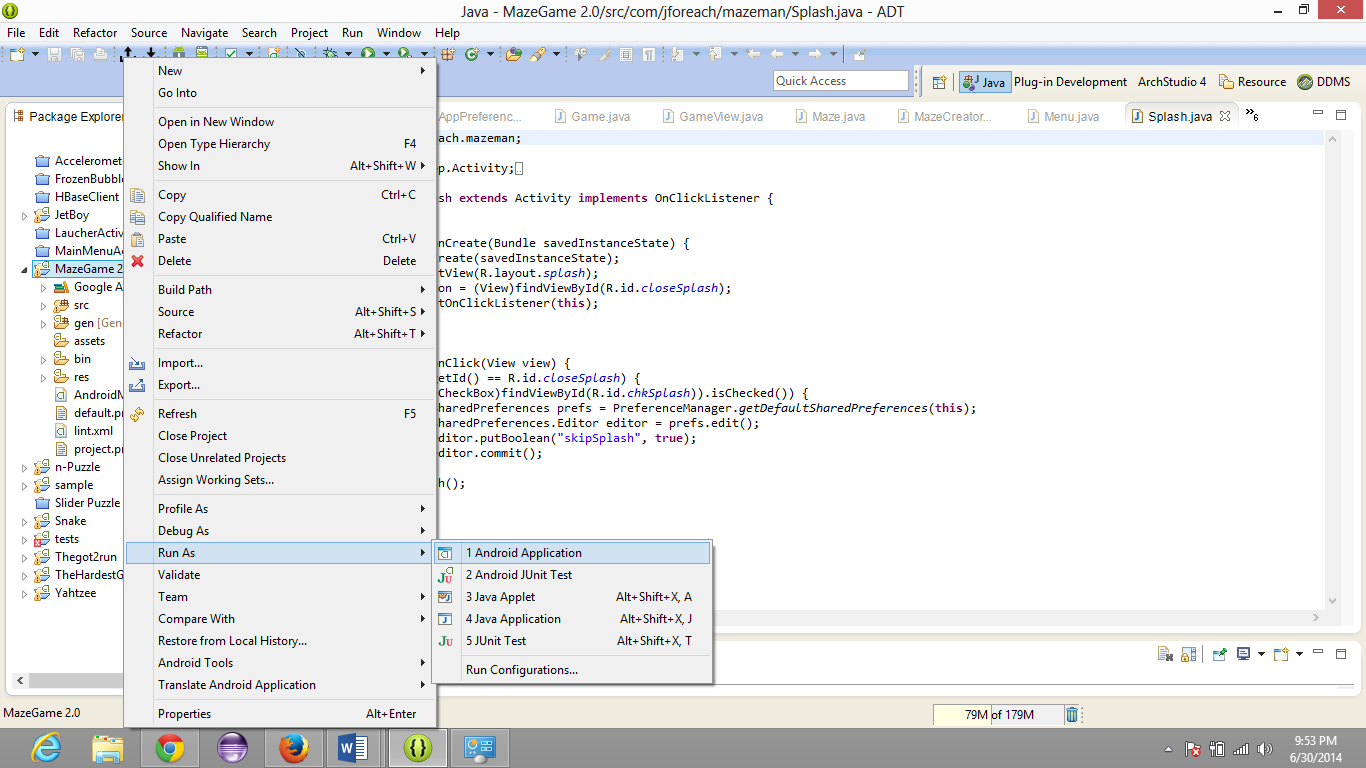
Analytical Tasks:

We need to first detect the gesture pattern by building a model .Now recognize the input data and know the gesture and then we need to produce output results.

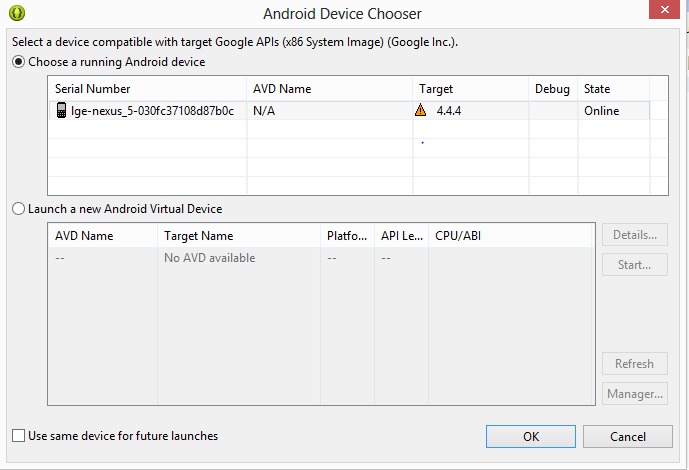
**Design of mobile client:**

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” .

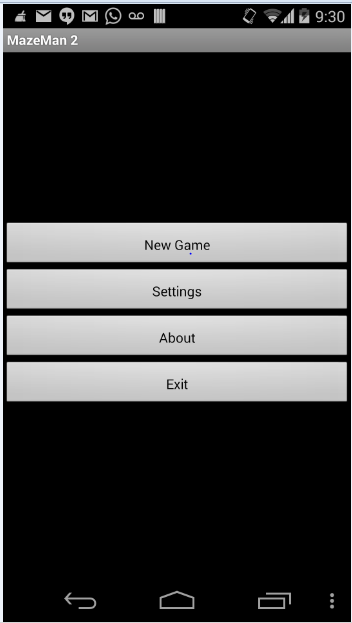




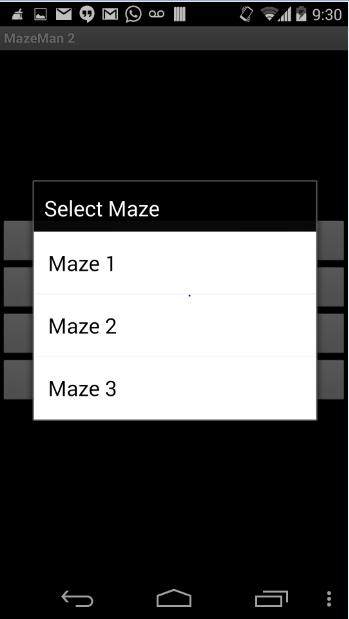
Running project as an android application on an android device(Nexsus 5 )



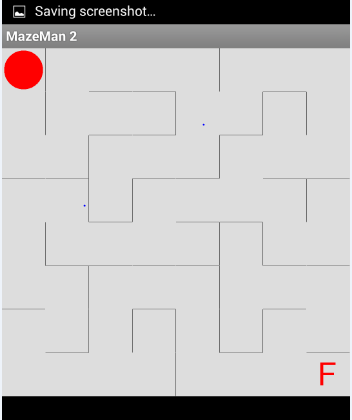
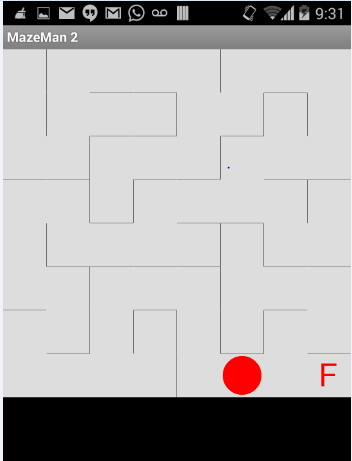
Below is a screen shot of the main screen of the game. “New Game, Setting, About, Exit.”



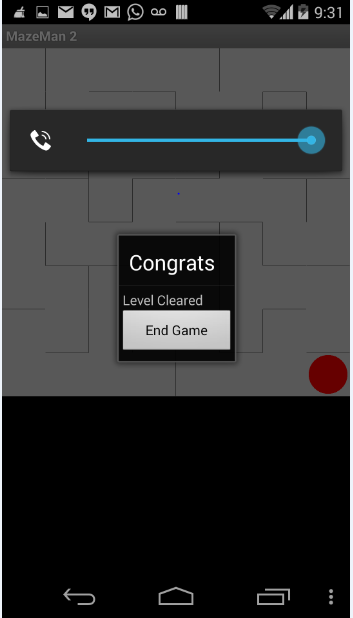
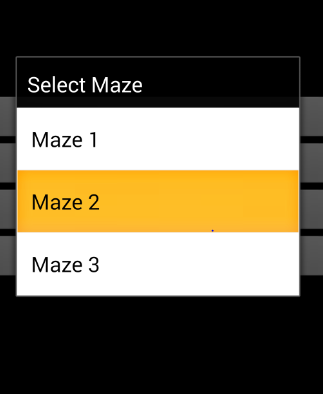
Choosing New Game Below pop up pops up. It looks as follows.



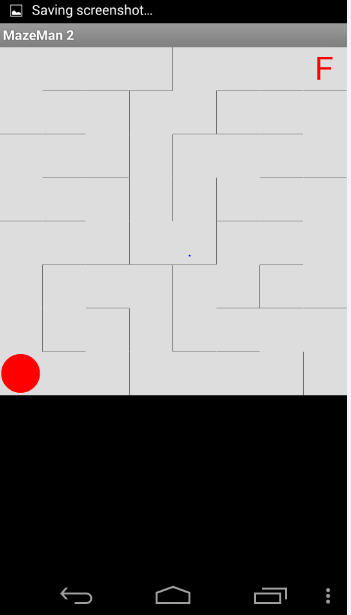
Real game screen is with a red ball which starts from a source and player has to take it to the destination “F”

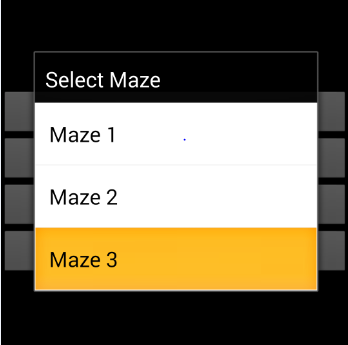
Clicking on End game ends the game level of maze 1.

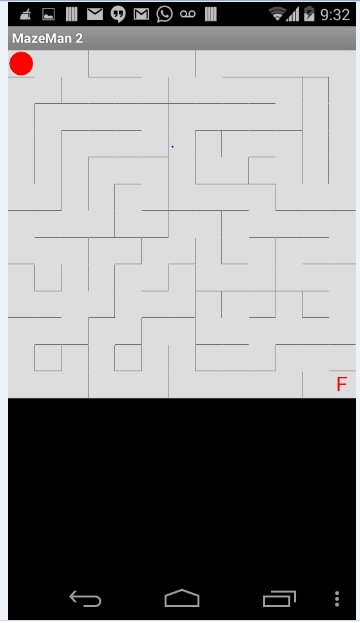
 

Above is the screen of three maze level available.

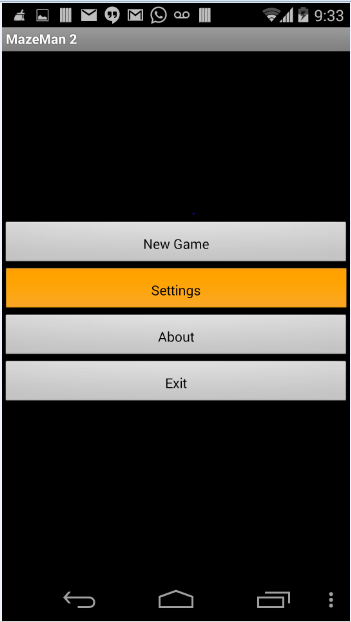


This is how Maze 2 looks like below.

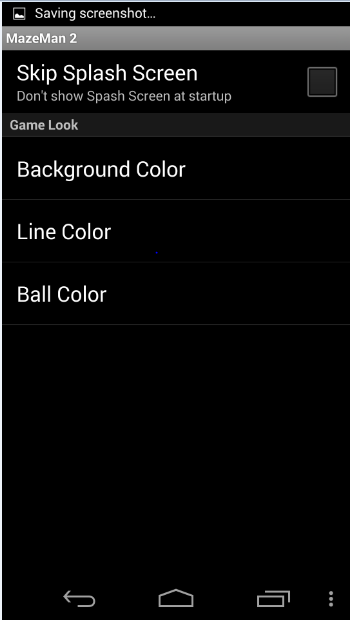
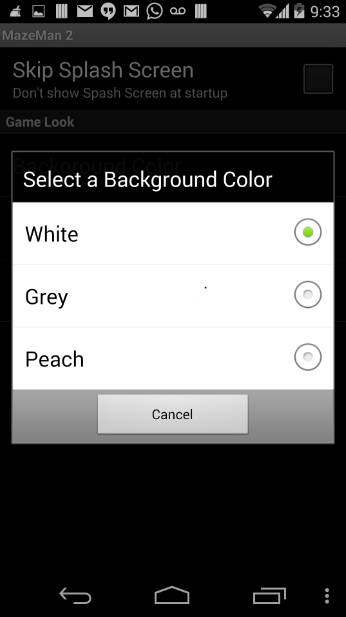


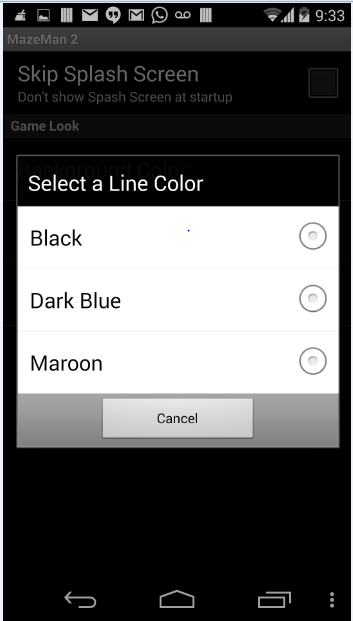
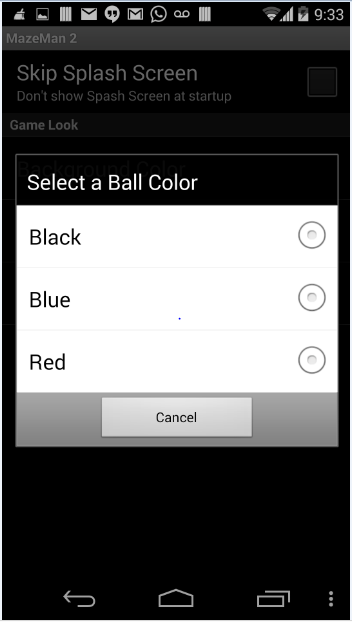


Every game has some options. And this game has the following options.



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.

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.

**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>

**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:**

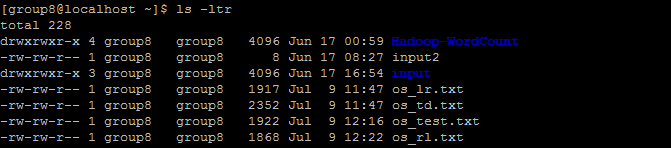
Data is collected for three types of gestures i.e. right to left, left to right and top to bottom gestures.

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>

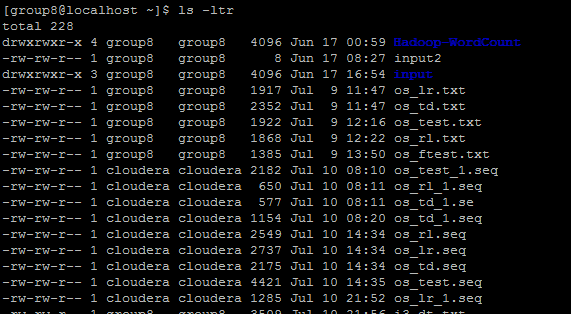
Below sequence files are generated upon running above URL’s.

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

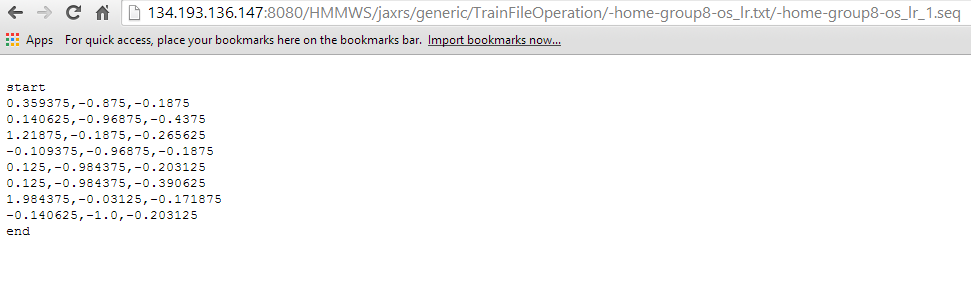
os\_rl\_1.seq: right to left gesture sequence file.

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

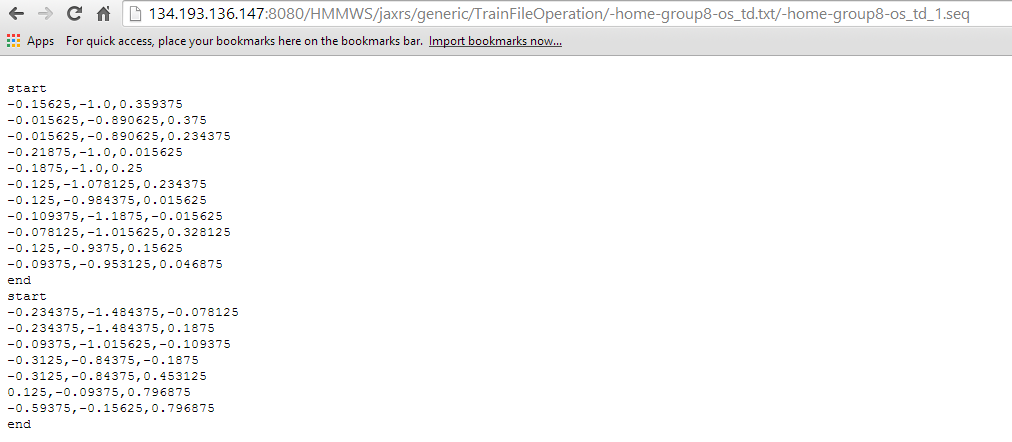
os\_test\_1.seq: combination of all gestures.



Left to Right Gesture Recognition: Train file



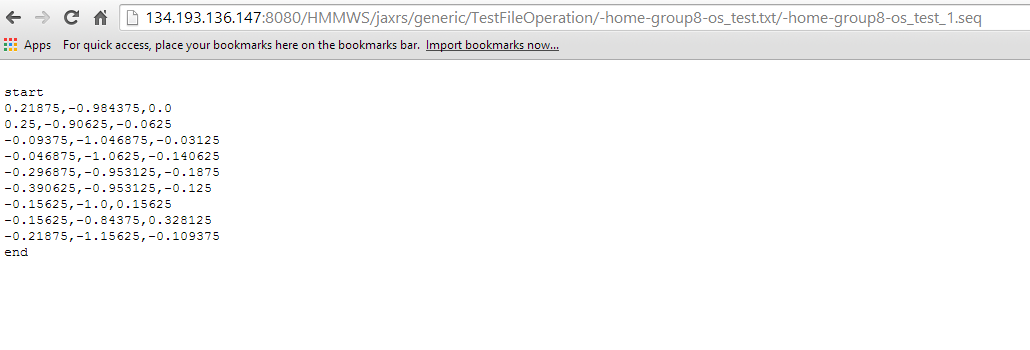
Top to bottom gesture Recognition: Train file



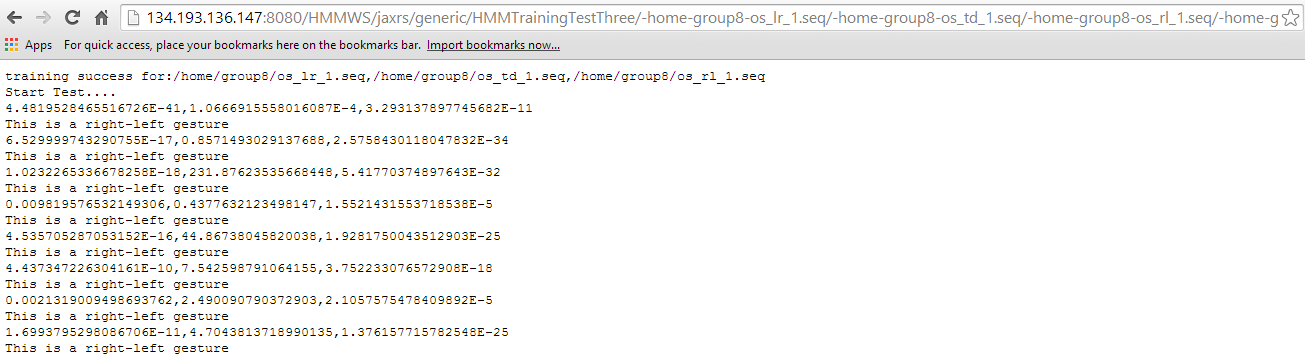
Right to Left gesture Recognition: Train file

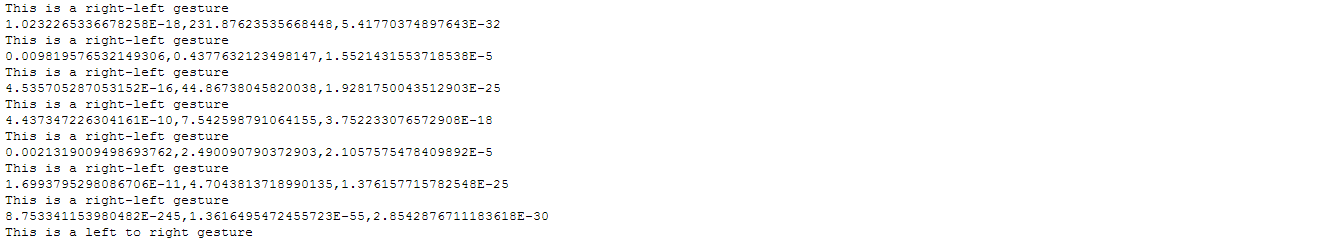


Combination Test file:



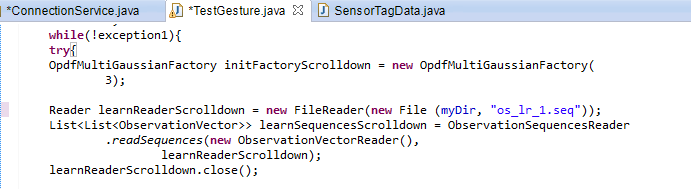
Final right to left and left to right gestures are recognized:





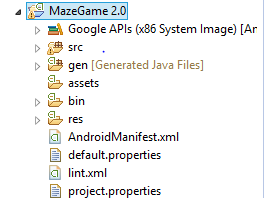
**Analytical Tasks:**

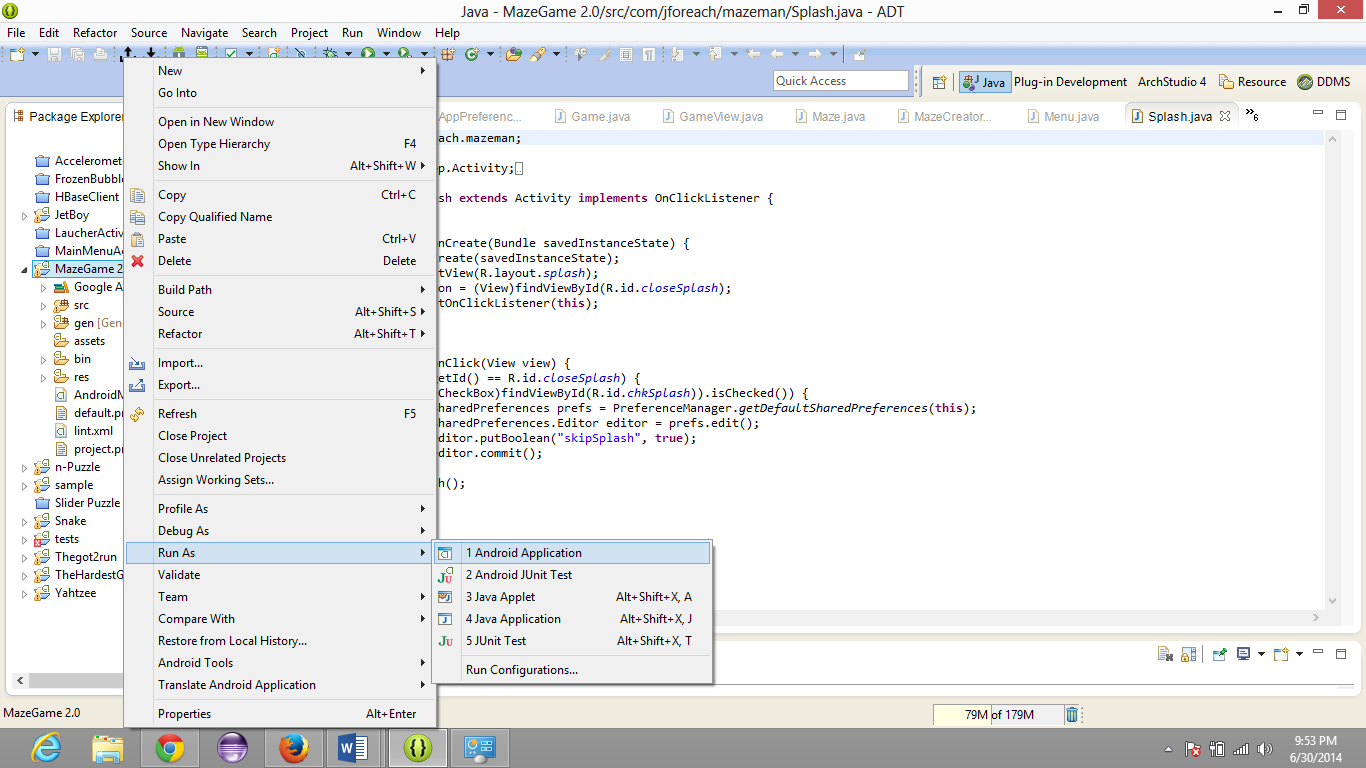
Our generated sequence files are then placed in connection service file of Maze game. These files are given as input to three different gestures model and signal sent to broadcast signal. Broadcast receiver will then receive these signals and the methods are called.



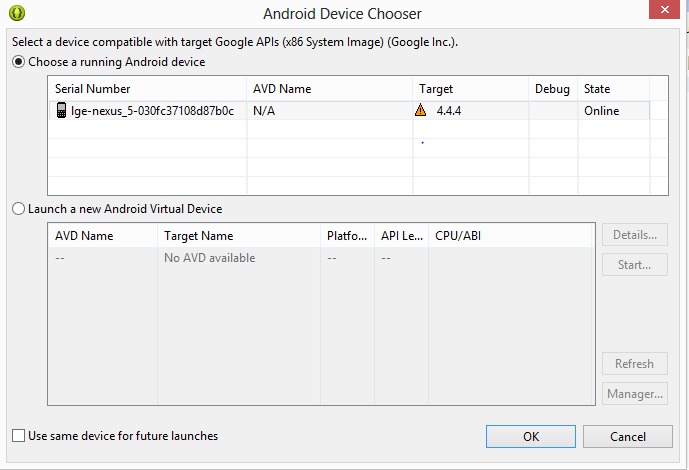
**Design of mobile client:**

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” .

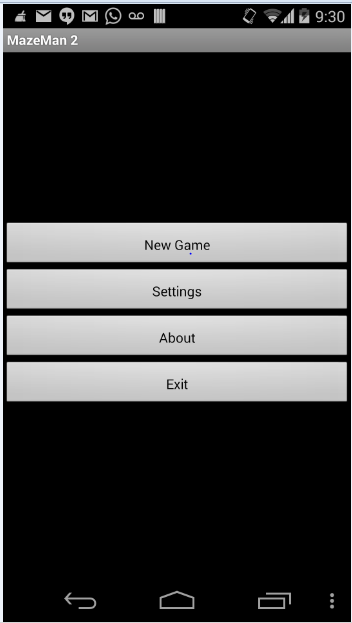




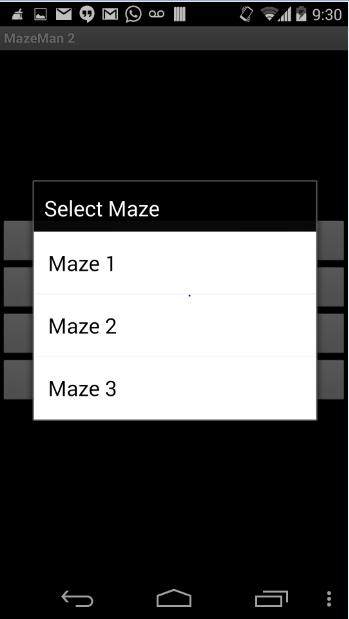
Running project as an android application on an android device(Nexsus 5 )



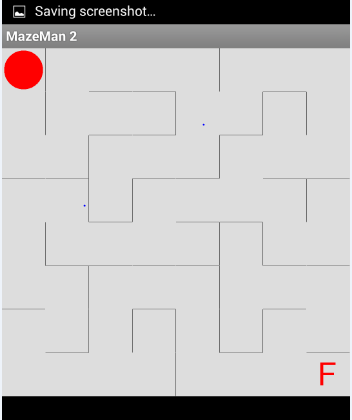
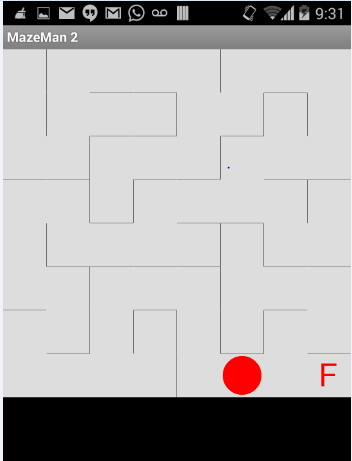
Below is a screen shot of the main screen of the game. “New Game, Setting, About, Exit.”



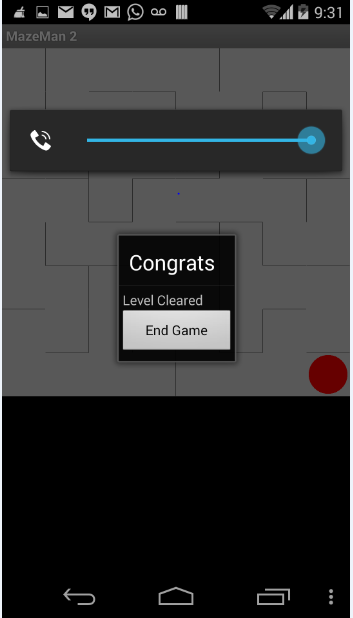
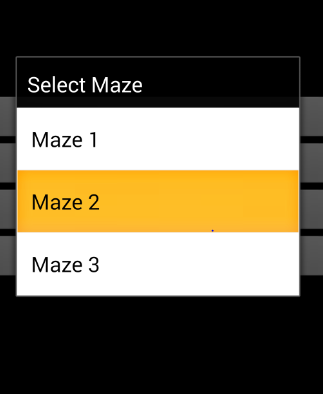
Choosing New Game Below pop up pops up. It looks as follows.



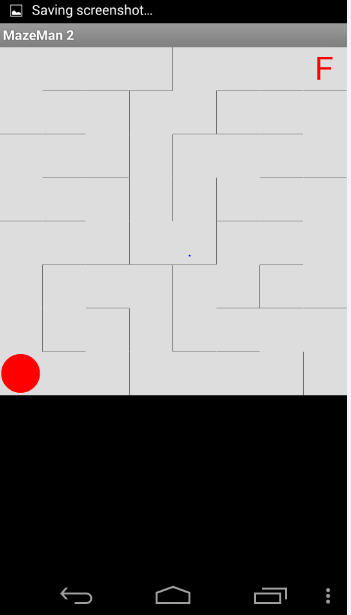
Real game screen is with a red ball which starts from a source and player has to take it to the destination “F”

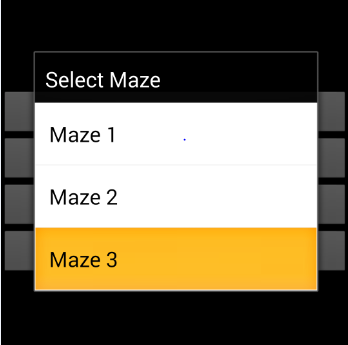
Clicking on End game ends the game level of maze 1.

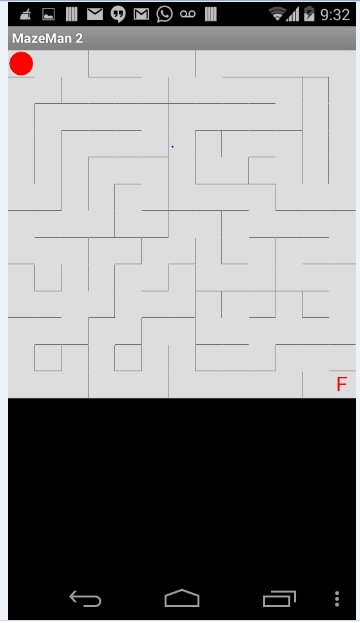
 

Above is the screen of three maze level available.

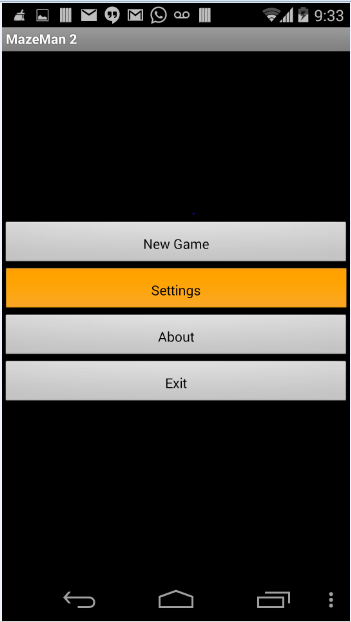


This is how Maze 2 looks like below.

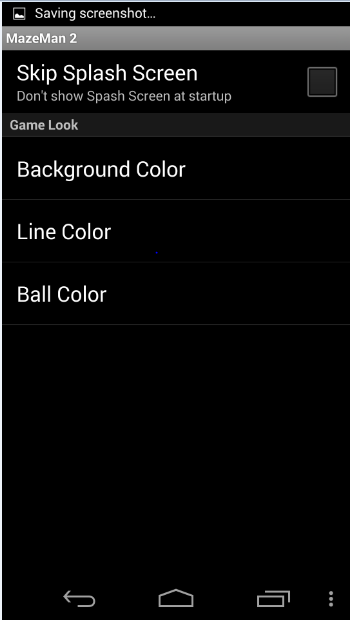
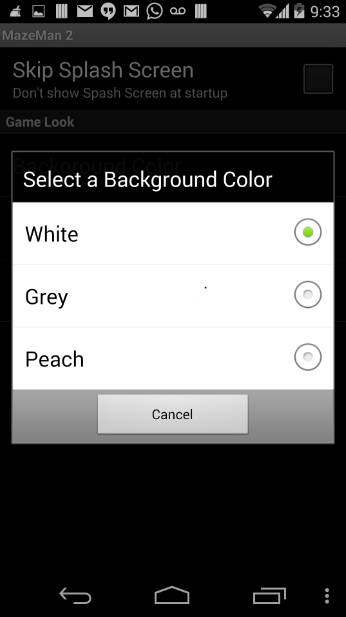


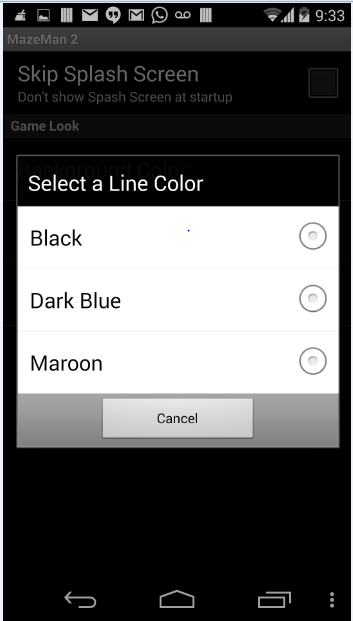
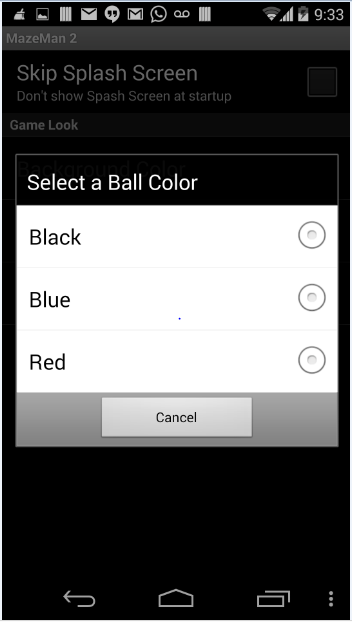


Every game has some options. And this game has the following options.



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

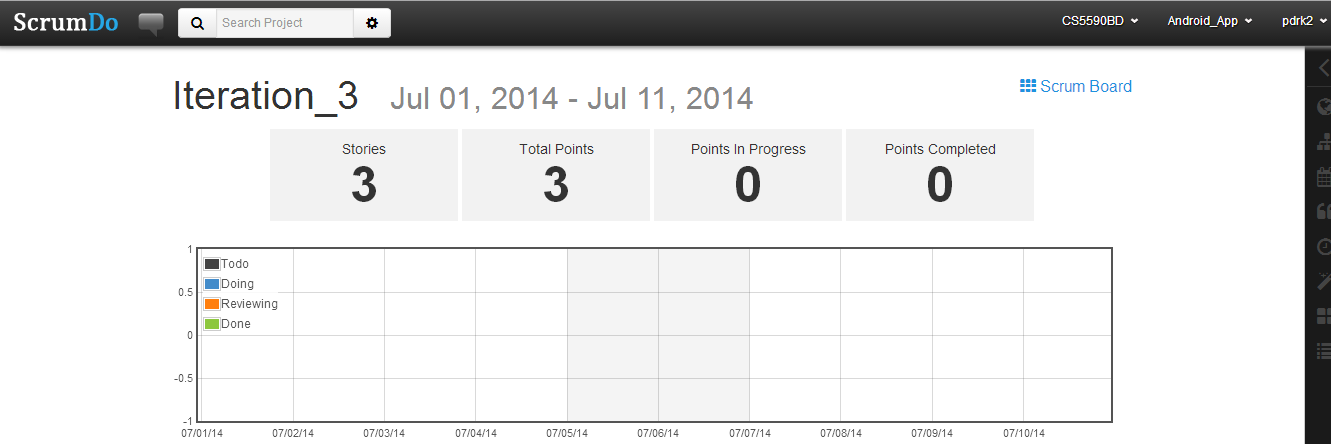
c)left

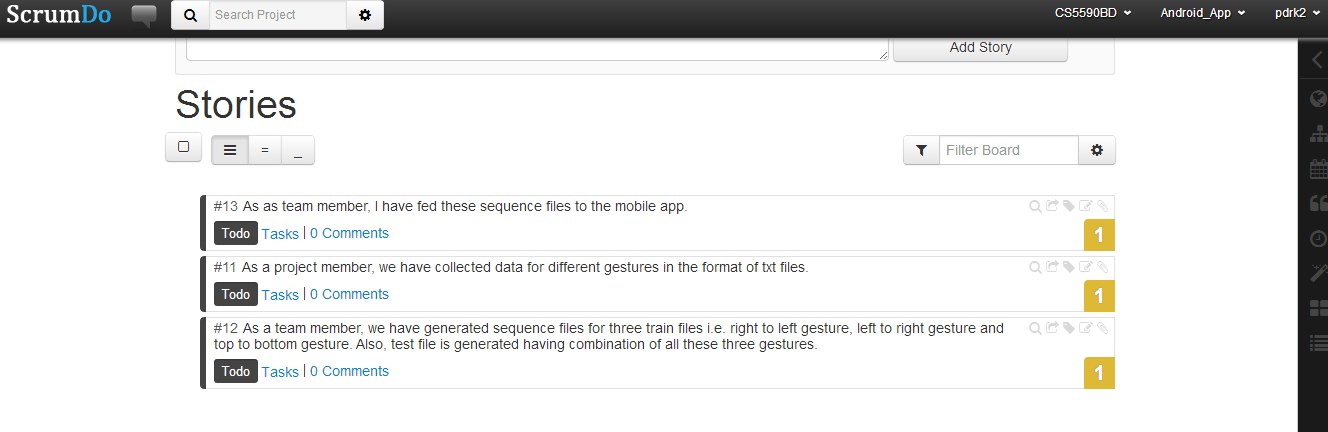
d)right

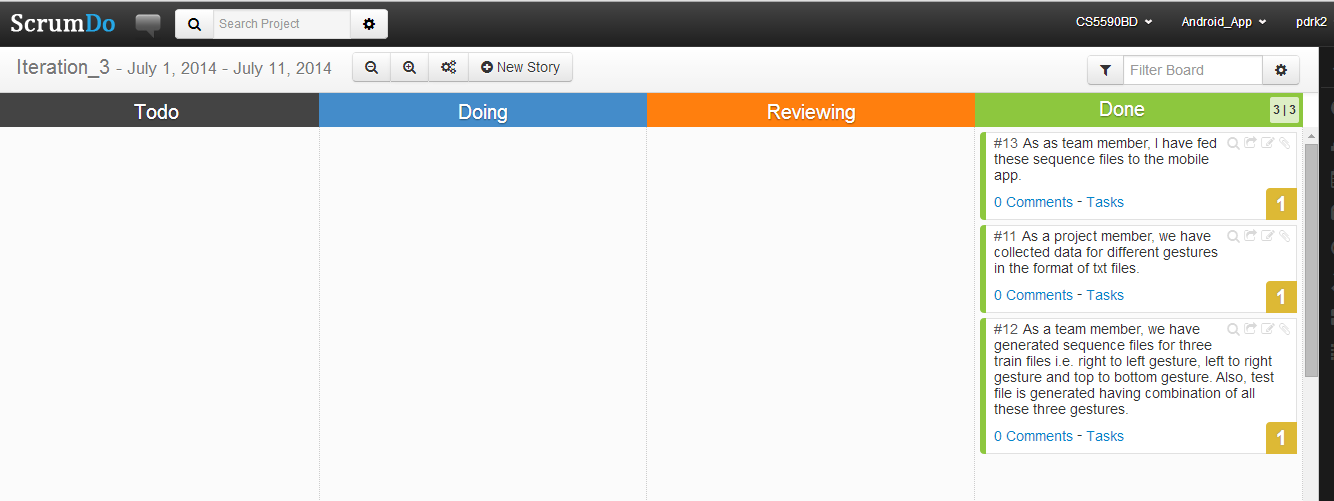
e)strat

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>

Fourth Increment

Group 8

Thru the Maze

**Project Goals and Objectives:**

**Motivation:**

‘Thru the Maze’ is a mobile based android game that uses gesture recognition to guide a ball in a maze. Traditionally,   users guide the ball through mobile phone buttons even more lately using touch motions, but in our app we will be doing it by gesture recognition using a sensor tag. We chose this game as we can embed online motion based sensing into this game and then generate an activity report by invoking web services of Big Data Analytics platform like HBase Retrieval and Activity Recognition. After all these we can evaluate these activities.

**Significance:**

We are integrating gesture recognition to mobile gaming. This gesture controlled gaming is moving technology ahead. We will be processing the data collected using sensor tag with big data and hadoop in the backend and depending on the type of gesture we can add functionalities and perform actions accordingly. Each and every technology used in this project are very much developing in the practical world. This is just the beginning of implementing gestures but in the future a lot more can be done using them.

**Objectives:**

We have selected a maze game in which the users need to direct the ball through start to end positions by 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. The gestures using the sensor tag are collected in the form of data mainly in 3d plane coordinated .This data is sent to hadoop in the backend and is used to process coordinates. Depending on these points as reference, the ball in the maze is guided.

**System Features:**

1) On-line Motion-based Gaming: (Left, Right, Top, Bottom)

The ball in the maze game is directed to the destination with the help of the gestures given by the sensor tag. Here four gestures are given to four operations. By performing these gestures the game is played. We should move the sensor tag from left to right for right gesture, right to left for left gesture, bottom to top for top gesture and punch for top to bottom gesture. The game perfectly detects all these gestures.

2) Activity Report:

The gesture values are inserted into HBase in a table format. Then values are retrieved and the respective gesture value are counted and the values are sent to graphical view for statistics.Using these values we plotted various graphs:

1. Timeline Report
2. Line Graph
3. Pie Chart Graph

3) Performance Meter :

The statistics are collected from the user play are used. And when he presses the performance Meter button he could see how he had played the game. Like whether he is a good player , avarage player or worst player. Thereby he can get measure for his performance.

4) SMS :

From the collected statistics we calculate in how many moves the user has completed the game .And he can share his activity with his dearest one through SMS by just clicking a button “send” displayed on app.

5) Map : ( Exercise gesutre )

This is an alluring feature of our app. Often doctors need to track patients excersice while diagnoising . So this feature allows him to do so. The patient performs an activity with sensor while he is walking and readings attained from instrument are used by the doctor to view how much distance he has walked along with the path he has walked on the map.

6) Track the Geolocation :

The user can track the current geolocation uisng this application in his vicinity.

**Online Application : Game by Motion**

In our maze game we move the ball to the target with the help of gestures. When the user uses gestures the sensor will detect gestures. Sensor will be trained beforehand about the gestures through data files which are then converted to sequence files. 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:**

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 various sensing capabilities like magnetometer, pressure, gyroscope, humidity, accelerometer, temperature. In total 6 sensors are available. As an initiative step to Online application sensing device is connected to mobile phone through Bluetooth the data then is taken to Data processing unit there after it goes motion recognition and is broadcasted to game app.

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. For further functional requirements we use BLE sensor tag app and Astro File Manager which can be downloaded from play store.

**Motion Models:**

Distance d is calculated using x ,y and z co-ordinates.

Where x(n) -current acceleration in x direction

x(n-1) -previous values of acceleration in x direction.

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

**Game Logic or features:**

Mere aim is to make a human play the game with the help of big data analytics applications. Accordingly user waves the sensor in the pre captured and trained gestures. Basing on that a chronologically order time line can be seen, his records in the game using performance meter, history of his different gestures and map which can be availed for tracing his physical exercise readings for proper evaluation.

**Recognition/Analysis Approach:**

Workflow of system is as follows:

Training Data from Sensor Tag

Testing Data from Sensor Tag

**Working Unit**

Put files into the intended folder on device

Run application on target device

Data from game is send to the testing unit where data is compared for results

Results will be used for accurate motion in the game

Algorithms, Input and Output:

**Input🡪K-Means🡪Hidden markov Models🡪output**

Input- accelerometer

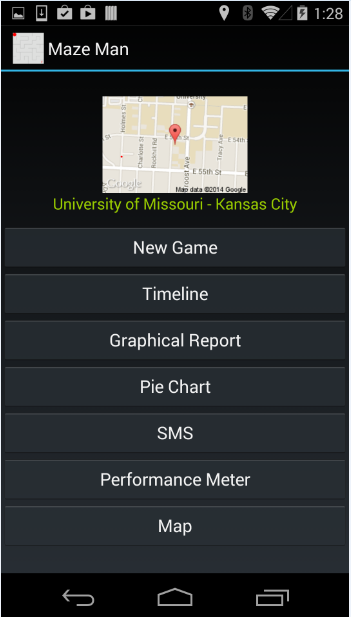
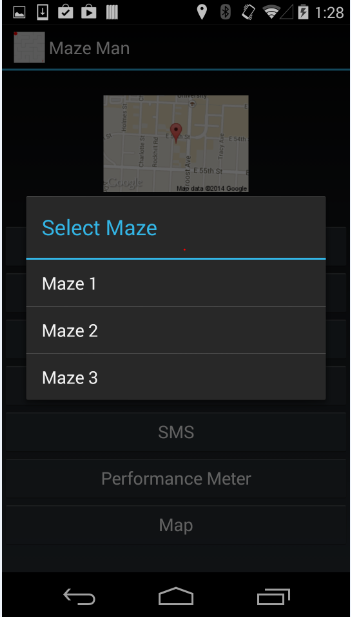
K-means-Unclustered is transformed as clustered data.(start and end points)

HMM- algorithm for generating seq files so as to recognize motion

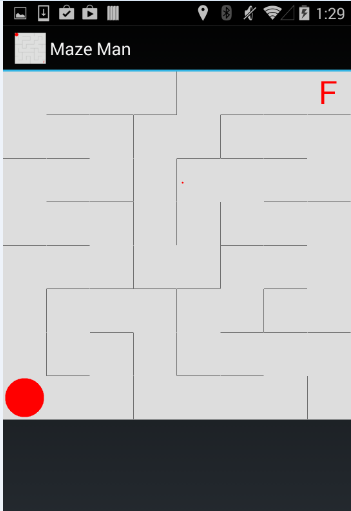
Output- gestures are turned as output.

**System Features and Android app User Interface:**

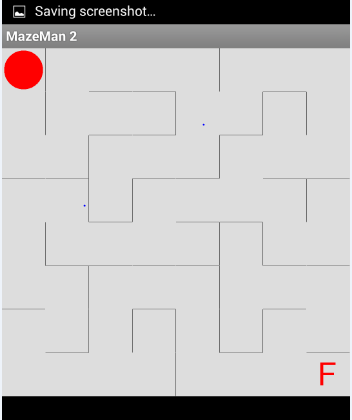
Below is basic view of the UI of the Maze Man app. On selecting new game option we can see the available maze levels 1, 2 and3.

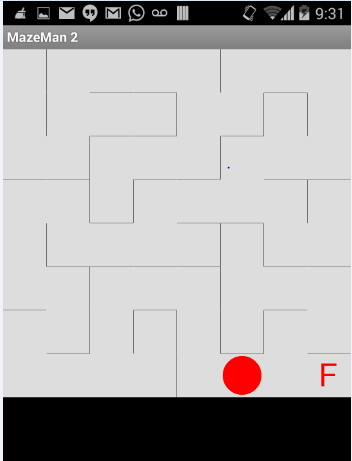
Game view of the maze 2 level is follows.



If the user wants to play he can move the ball by giving the appropriate gesture.



After giving the gestures



**Offline Application: Activity Report**

**Devices/Sensors:**

Sensor Tag: Sensor Tag is moved in directions looking into the maze on the android device screen. Now the ball is moved from current start location to the end location at maze “F”. There will be path for user to move but if user moves with less interest then user may take much time and much more moves for moving the ball.

Android Device: Device gets values from sensor and moves accordingly. Device is made user friendly by allowing such kind of access. We use devices like Nexsus phone and Tabs. Data from sensor tag with necessary values along with time values is noted. This is done to facilitate more accurate activity reports.

**Activity Models:**

Sensor uses the usual coordinate axes system virtually for getting the coordinate details. As Sensor is moved onto right of the device x coordinate will be positive, for left the x-coordinate is negative, for top y is positive and down y is negative. Apart from this z axes are also used when sensor tag is move towards device z is considered positive else negative. On investigation we noticed that pie chart, timeline graph and timeline report could as best source for viewing the over play of a player. Exercise of a user is plotted on the map with start and end points.

**Features:**

Summary of the game movements can be visualized using Hbase data values. For the sake of visualization we use timeline graph distinct color for each gesture and curves are plotted dynamically. Pie char facility is also provided to see the ratio of movements of gestures on a 360 degreed circle, distinction is featured through colors for gestures. Timeline report will have various values arranged chronologically giving an idea of how gestures are recognized and performed. Exercise of a person are allowed to remain transparent to the physician through this app. User can play the game even during their exercise session and movements of the person are collected for further regulation of user exercises.

a)Timeline Chart :

Here we get the chart with the date and time the particular gesture is performed.

b)Pie Chart :

We will have pie chart for the four gestures performed. The count of each gesture is taken is saved and retrieved from there we get the pie chart. We can also see the values for the dates selected from calendar.

c)Line Chart :

A graph is plotted for the time and the occurrence of gesture. Not only current day’s graph the other days graph also can be seen.

**Workflow:**

Use data sequence files on SD card

Push data onto Hbase

Run android application

Connect Sensor tag

Analyze the given gesture

Get the right gesture

Move the ball accordingly

Get results through report on device’s screen

Inputs 🡪 User movements

Output🡪 Activity report of the user for further analysis.

-Here we first filter the data.

-Then quantize the data as it is vector data.

-Then we send the data through Hidden markov model.

**System Features:**

1. Time line view: all the gesture which are sensed after checking its probability are shown list wise for recently played game.
2. Graphical view: graphical information is definitely a very good source of info. It depicts on how long a gesture is continued or stable or not used at all. Using such valuable info some valuable results can be brought out.
3. Pie chart: Over a full circle gesture color indicate their independent ratio. All those juxtaposed all along area of the circle. They also act as a good source of info. These are specially used in cases where some comparison kind of results are expected.
4. Performance meter: User can check his gaming results decide himself if he is going well or not. He can challenge himself for every single play as results are put out using each play’s values.
5. Exercise: Physician can get exact action of user if he is continuing with his exercise or not without wasting any time for checking him on hourly or daily basis. This facilitates user as he need not go to doctor everyday rather he could do manage himself and decide when to meet physician in person for more progress in treatment.

**Software Architecture:**



**HBASE**

CREATE TABLE

SAVE

DATA

GET SEQ

FILES

FILES

**DATA PROCESSING**



CREATE HMM

DECIDE K-MEANS K VALUE

OBSERVATION VECTOR

TRAINING DATA

**HMM ALGORITHM LAYER**

TESTING DATA

CHECK PROBABILITY

LOAD TEST DATA

USE HMM VECTOR

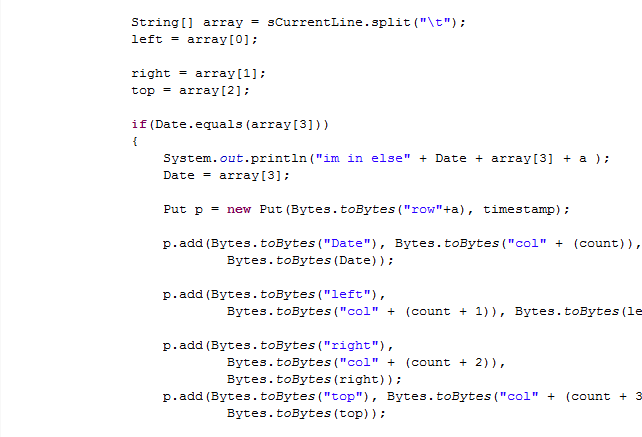
**ACTIVITY REPORT**

**GET GESTURE**

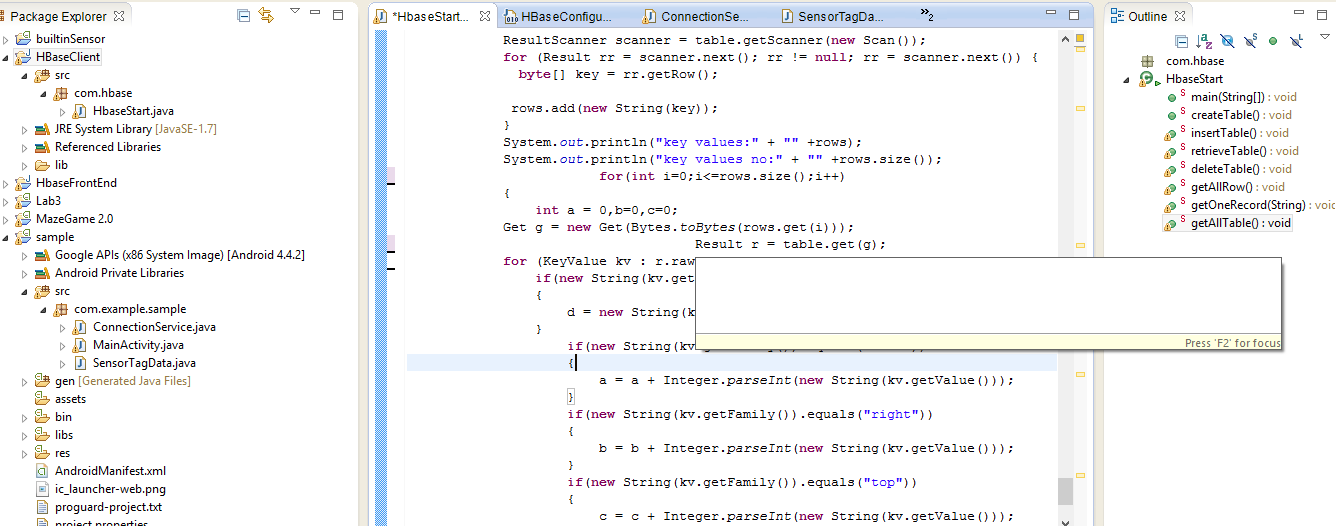
**MAZE MAN**

**Hbase Design:**

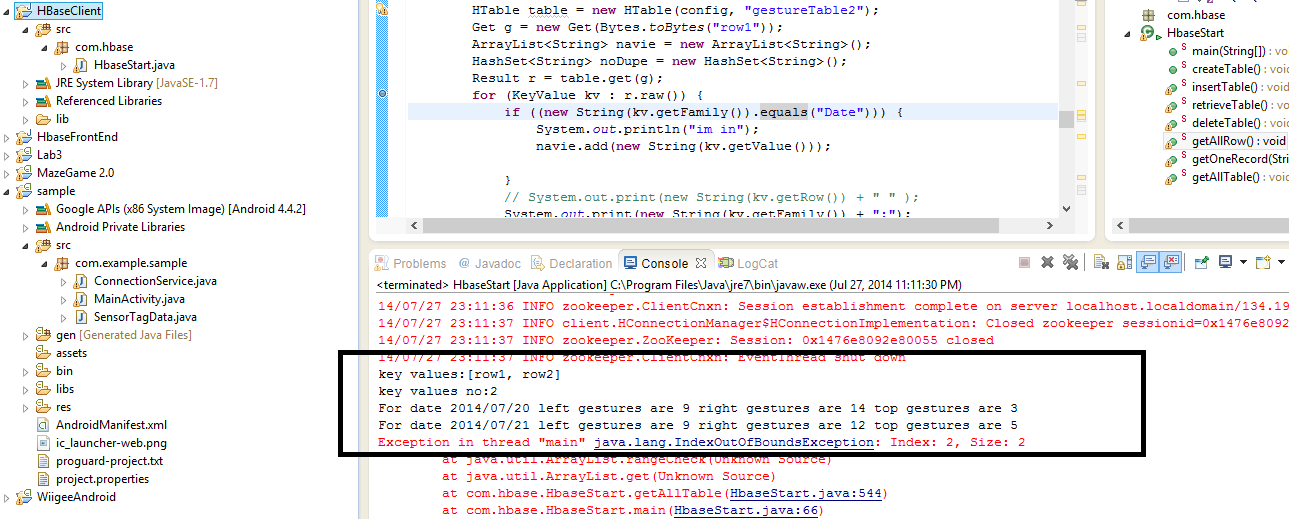
The values are inserted in Hbase by taking row key as a primary key. Here row key was defined uniquely for each different date.



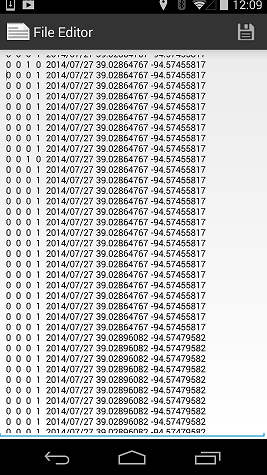
The values were retrieved using Hbase using unique date which is taken as row key.



In this way, gesture count is retrieved for each row key i.e. date.

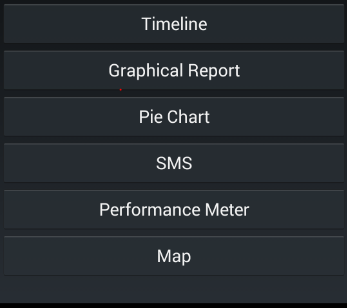


The values are stored in textfile before uploading to Hbase.

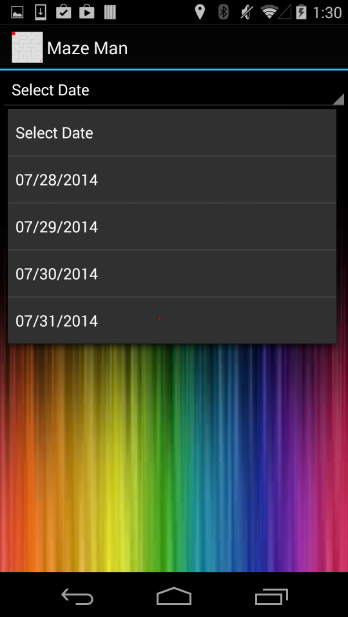
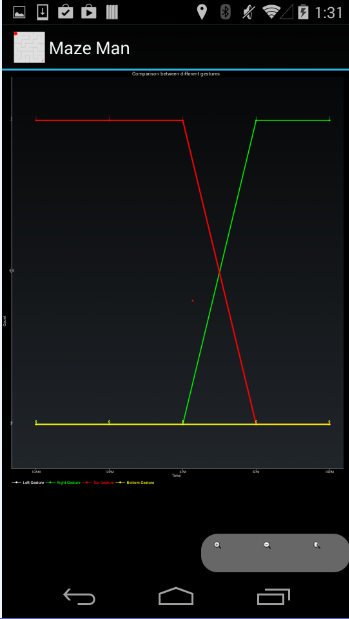


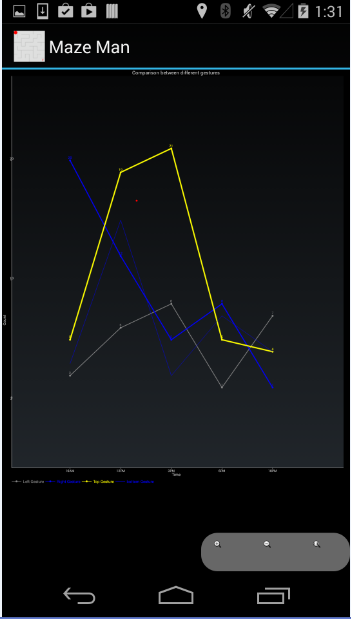
**Android app GUI:**

Below shows the options for the activity report. Clicking on timeline looks as later shot

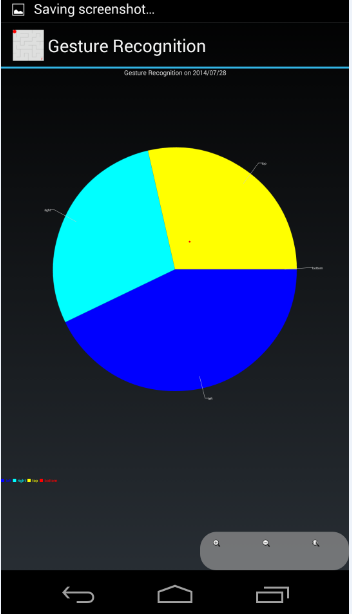
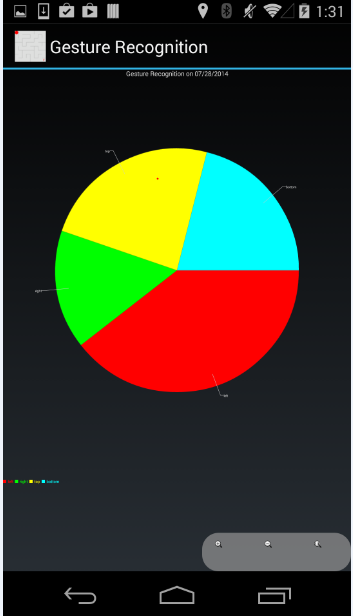




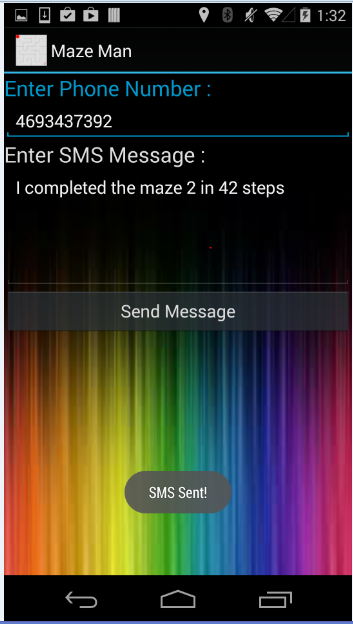
Below is the Graphical report of the game. It can be chosen based on the date selection. 

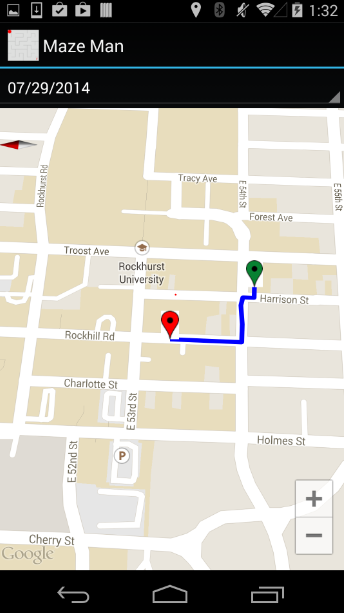
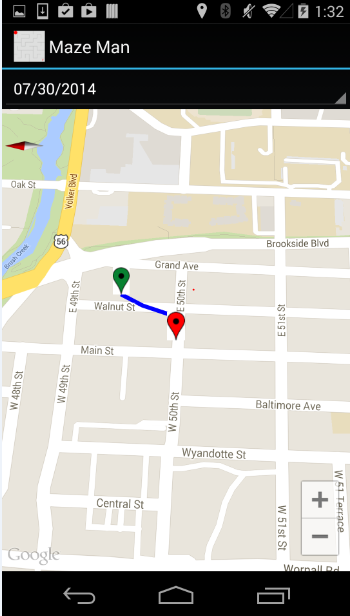
Next Option is the pie chart representation of current play’s gesture. Each color speaks about each of the gestures.

Next is send sms feature. Person can send message by entering the contact number. Message can be directly messaged from the app itself. Message can be anything like the number steps he used for crossing the level. It also shows the status of the message if sent then a sent message is been displayed.

Left picture is a sample of the performace meter. He did play well and hence is the appreciation message displayed.

This is how map option works. When clicked his location along with the path of his movement during exercise is displayed on google maps. It also shows the markers for path boundary. Green as start Red as end.

**EVALUATION: MOTION/ACTIVITY RECOGNITION**

Number of Users: Five users

Types of Motions/activities:

a) Up

b) Down

c) Left

d) Right

e) Exercise

Cross Validation: Independent datasets are used to take a good value of K. By evaluating all these we have taken K value as 6.

Confusion Matrix for Activity Recognition:

PREDICTED

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ACTUAL |  | UP | DOWN | LEFT | RIGHT | EXERCISE |
| UP | 19 | 1 | 0 | 0 | 0 |
| DOWN | 3 | 17 | 0 | 0 | 0 |
| LEFT | 0 | 0 | 20 | 0 | 0 |
| RIGHT | 0 | 0 | 0 | 20 | 0 |
| EXERCISE | 0 | 0 | 0 | 0 | 20 |

Precision: Exactness –what % of tuples that the classifier labeled as positive are actually positive.

Precision = TP/ (TP + FP)

Recall: Completeness – what % of positive tuples did the classifier label as positive ?

Recall = TP/ (TP+FN)

F-measure (F1 or F score) : Harmonic mean of precision and recall

|  |  |  |  |
| --- | --- | --- | --- |
| A/P | C | ~C |  |
| C | TP | FN | P |
| ~C | FP | TN | N |
|  | P’ | N’ | ALL |

F-measure = 2\*Precision\*Recall/ (Precison+Recall)

UP

Precision UP) = TP/ (TP + FP) =0.8636

Recall (UP) =TP/ (TP + FN) =0.95

F-Measure (UP) = 1.64084/1.8136 = 0.9047

DOWN

Precision (DOWN) = TP/ (TP + FP) =0.9444

Recall (DOWN) =TP/ (TP + FN) =0.85

F-Measure (DOWN) =1.60548/1.7944 = 0.894

LEFT

Precision (LEFT) = TP/ (TP + FP) =1

Recall (LEFT) =TP/ (TP + FN) =1

F-Measure (LEFT) = 1

RIGHT

Precision (RIGHT) = TP/ (TP + FP) =1

Recall (RIGHT) =TP/ (TP + FN) =1

F-Measure (RIGHT) = 1

EXERCISE

Precision (EXERCISE) = TP/ (TP + FP) = 1

Recall (EXERCISE) =TP/ (TP + FN) = 1

F-Measure (EXERCISE) = 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | UP | DOWN | LEFT | RIGHT | EXERCISE | OVERALL ACCURACY % |
| KNN | 0.9047 | 0.894 | 1 | 1 | 1 | 95.9 % |

Limitations:

* When there are many sensor tags being enabled. The application gets confused and does not connect to any one of those. In those circumstances, we should first make sure our sensor tag is connected with the help of BLE sensor tag app and then run our application.
* The user may get strained by giving the gestures for long time. If he plays for much time.

**You Tube Link:**

<http://youtu.be/iPAMduIBXMM>

**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>

<http://wptrafficanalyzer.in/blog/android-drawing-pie-chart-using-achartengine/>

<http://autofei.wordpress.com/2012/04/02/java-example-code-using-hbase-data-model-operations/>

<https://www.altamiracorp.com/blog/employee-posts/handling-big-data-with-hbase-part-4-the-java-api>

<https://www.youtube.com/watch?v=zPhjnN0HD2E>

<https://www.youtube.com/watch?v=wDmosRnEfiw>