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

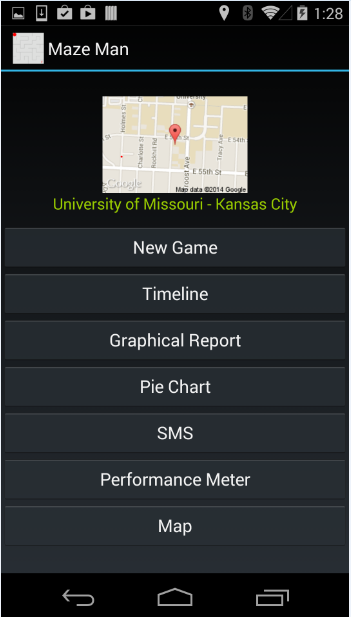
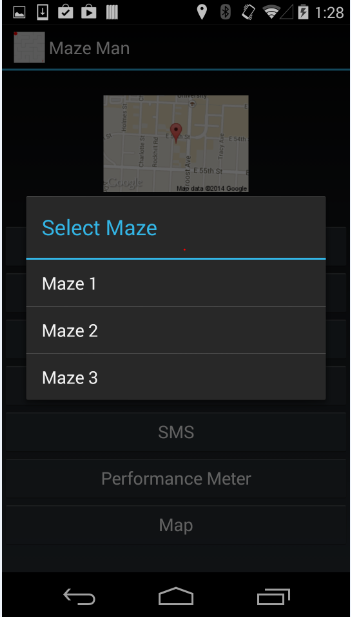
K-means-Unclustered is transformed as clustered data.(start and end points)

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

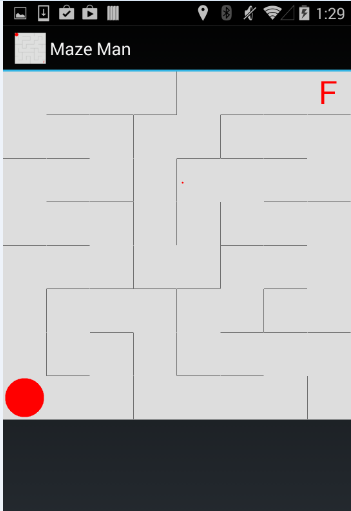
Output- gestures are turned as output.

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

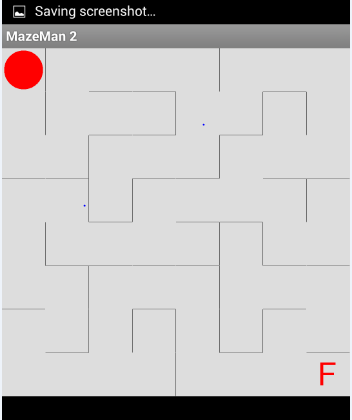
Below is basic view of the UI of the Maze Man app. On selecting new game option we can see the available maze levels 1, 2 and3.

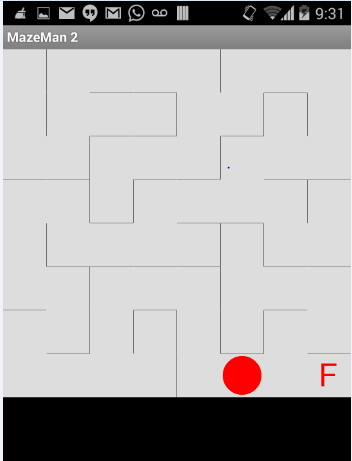
Game view of the maze 2 level is follows.



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

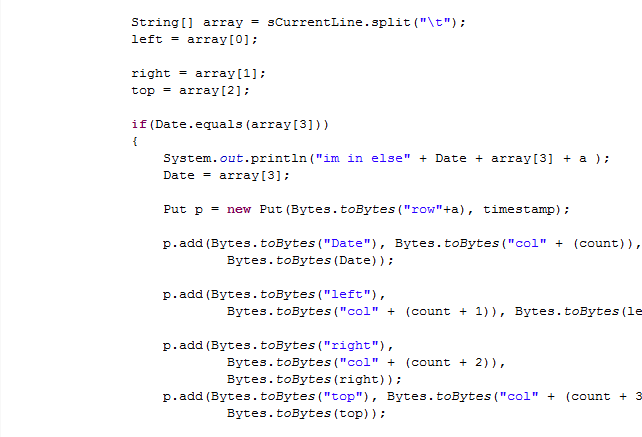
**ACTIVITY REPORT**

**GET GESTURE**

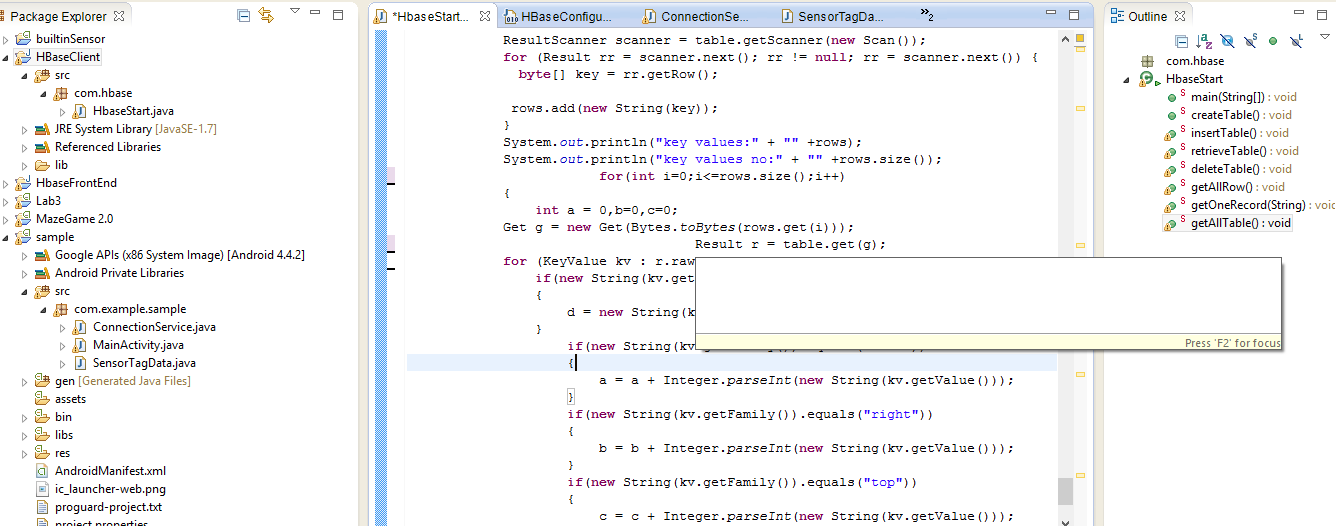
**MAZE MAN**

**Hbase Design:**

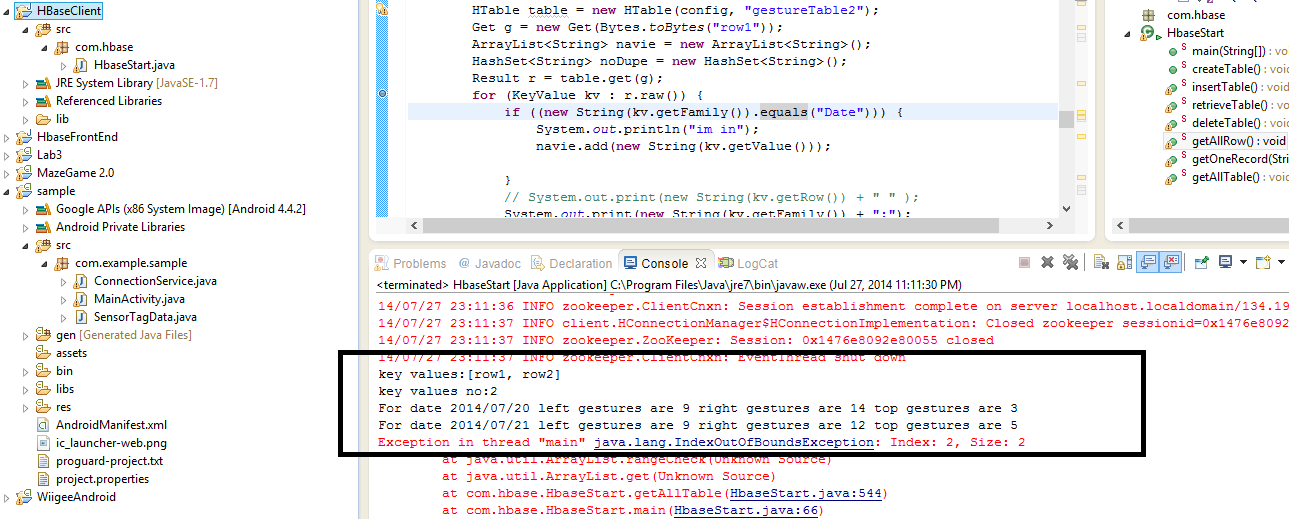
The values are inserted in Hbase by taking row key as a primary key. Here row key was defined uniquely for each different date.



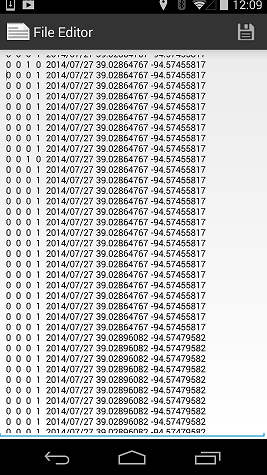
The values were retrieved using Hbase using unique date which is taken as row key.



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

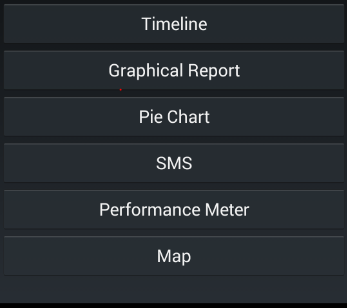


The values are stored in textfile before uploading to Hbase.

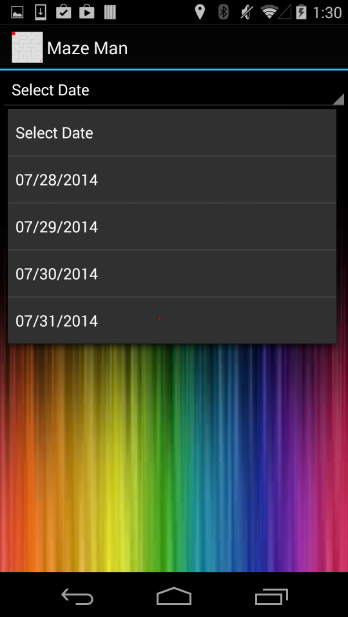
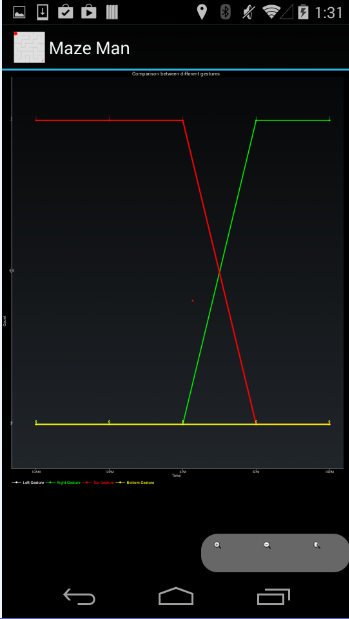


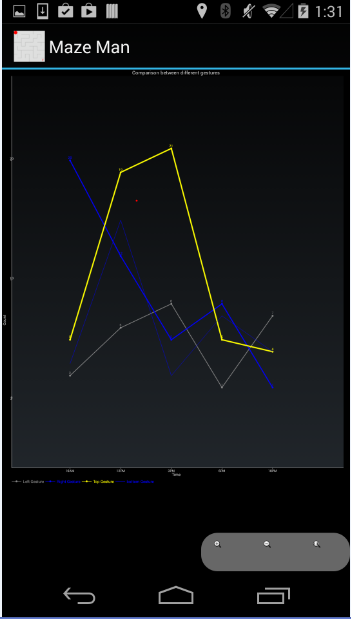
**Android app GUI:**

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

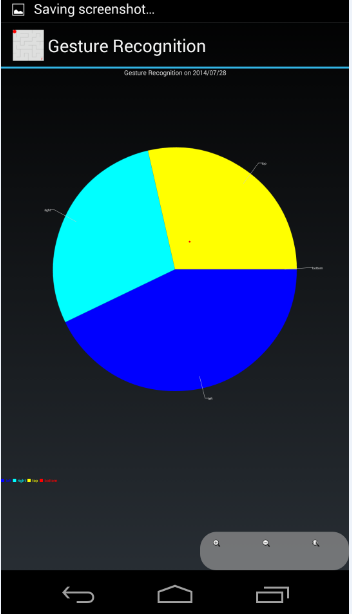
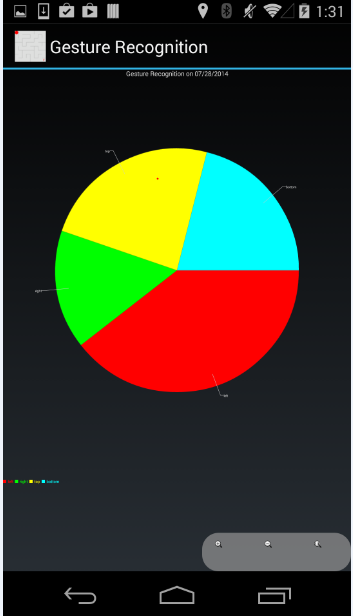




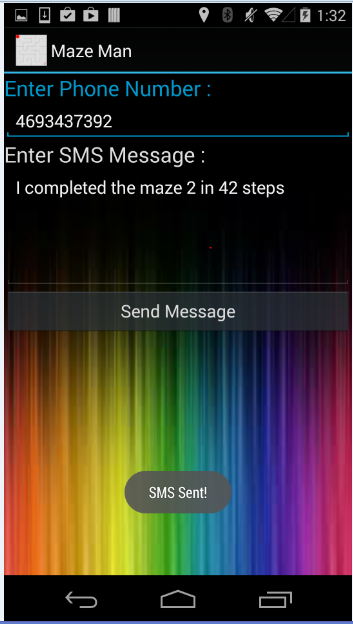
Below is the Graphical report of the game. It can be chosen based on the date selection. 

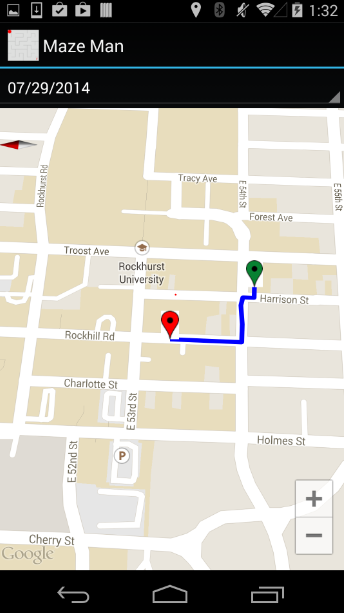
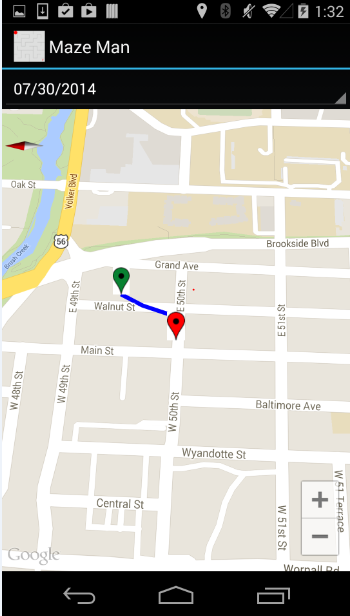
Next Option is the pie chart representation of current play’s gesture. Each color speaks about each of the gestures.

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.

**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>