

**By:**

Praneeth Paruchuri

Divya Battu

Mounika Karampudi

GireshBabu Yemparala

**CS590BD Big Data Analytics and Apps**

School of Computing and Engineering, University of Missouri –Kansas City

**PAGE OF CONTENTS**

1. PROJECT DEPLOYMENT
2. Introduction
3. How to use
4. Error recognition and handling
5. Sample interaction accompanied with screen dumps
6. Bugs and deficiencies
7. Future work
8. Links & References
9. PROJECT MANAGEMENT
10. First Increment
11. Second Increment
12. Third Increment
13. Fourth Increment
14. FINAL PROJECT EVALUATION

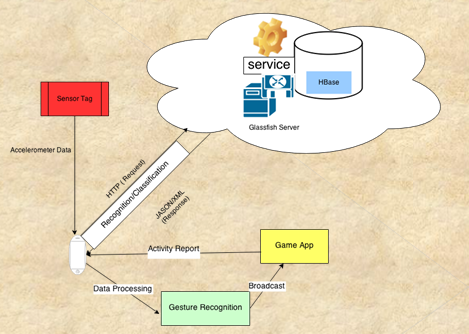
**PROJECT DEPLOYMENT**

# INTRODUCTION

Touch based games are drastically being replaced by motion sensing games worldwide. Due to such technological advancements we have decided to take the age old snake game which is popular among all the age groups to a whole new level. In order to achieve this we replaced the touch in the game with motion sensing. The aim of this game is to increment the score by making the snake grow longer by collecting the apples. The game comes to an end when the snake touches either the wall of the screen or itself. The Existing game does not have any user activity monitoring. Thus, we have implemented gesture recognition into our project MSNAKE.

We are using the Texas instruments sensor tag to implement the various gestures. There are many inbuilt sensors like Gyroscope, Accelerometer, barometer, pressure sensor and Humidity sensor. We are using the Accelerometer sensor to obtain the data. The various gestures we used are Up, Down, Right, Left and Pause. We’ve taken fivegestures on the whole, one for every movement of the snake and the fifth is the back button which is used to pause the game. Fly indicates upward movement, stomp for downward movement, left for left movement, right for right movement and back to pause the game.

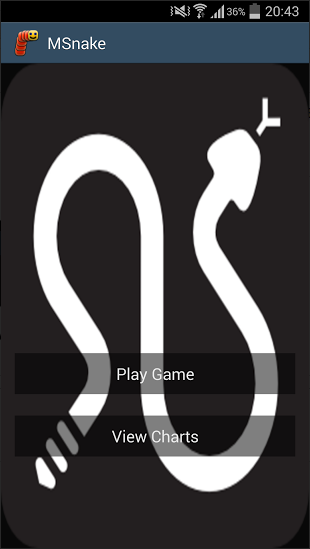
The architecture of our game is as follows.



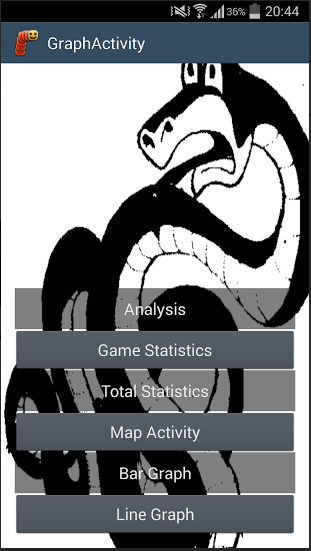
 It includes the components android device, hbase, service, server, The data is collected from sensor by android device and it is then processed. This processed data is then segemeted to sequence files. The sequence files generated are then analysed using the JaHMM library in Gesture Recognition process, by the KMeans procedure it recognizes the trained gesture and returns a value, this values is then broadcasted into the Game App. The Broadcasted value then triggers the action in the Game. The activity of the user is recorded and is then sent to Hbase for recognition and classification. On data classification the app generates reports for user activity.

The Hidden markov model was used to detect the different gestures. K- means clustering is used to detect the activity. We started with k as 10 and if there was no error, k was 10, but if the value of k was decreased whenever an error was encountered and so on. We collected the data for five gestures using hand movements, generated sequence files for each gesture and trained each gesture for atleast 16 times. We deployed the HMMWS war file into our local glassfish and converted the txt files into sequence files with the recognition of the start and end of every gesture. These sequence files are trained at the start of the game, and once they are trained the gestures are recognized and the corresponding function is called, providing the movement of the snake. The value of k for the five gestures, fly, stomp, left, right and back are 7, 9, 10, 8 and 3 respectively.

# HOW TO USE?

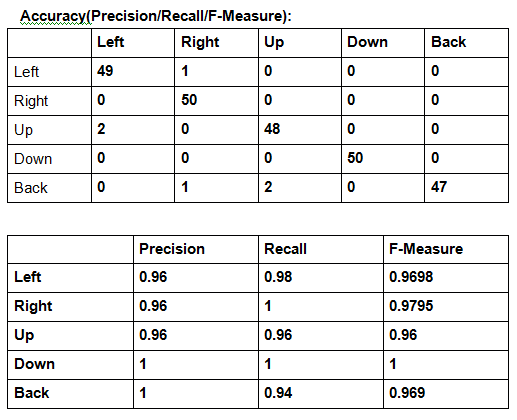


Our application is compatible with any android device like the nexus & tablet/ nexus 5 smartphone or the Samsung S4, etc which has a Bluetooth version of 4.2 or higher and a GPS. User needs to go to the MSnake screen as shown in the figure and click on the Play Game button in order to start the game. The user has to first let the training to be completed in order to start playing the game using the gestures. Once the game starts the user needs to make the four gestures, fly, stomp, left, right using the sensor tag in order to make the snake move upwards, downwards, left and right respectively to collect the apples and grow longer. The user can use the fifth gesture i.e. back gesture, which is used to pause the game. Once the user finishes playing, he can quit playing and



come back to the home screen and can click the view Charts button in order to view the different types of graphs. He can do it by clicking on the respective buttons on the GraphActivity screen. The analysis button displays the total number of gestures ever used and also the total number of times each gesture had been used. The game statistics displays a pie chart with each color representing the total number of times each gesture was used in the most recently played

played game. The map activity displays markers which give the exact location of the player along with the date and time. In the bar graph, each bar represents the total amount of time in milliseconds the player has spent on playing the game per day.The save data button is used to save all the data whenever the game comes to an end or whenever the user exits the game or whenever the snake crashes by hitting the wall. We have achieved an accuracy which is more than 97 percent.



# ERROR RECOGNITION AND HANDLING

The pie chart showing the statistics for the last game played may not be visible sometimes, this happens only when the user hasn’t played the game with gestures or started the game but hasn’t made a move and exited. The pie values in the pie chart are the number of times a gesture is performed(recognized). So, inorder to handle this error, play the game with gestures, it’s more fun this way.

The game might not start sometimes, this problem occurs either when the GPS is not enabled or there is no GPS lock on your location. Whenever you start a game, the GPS location(the coordinates) and the system time are stored and this information is used to plot the marker on the map, to view your activity or to track your movements. To handle this issue, switch on the GPS before you start the game and make sure the GPS gets a lock on your position.

The gesture recognition doesn’t start as soon as the game starts, don’t panic, the game works fine, in fact unbeatable, it takes 1-2 minutes to train the gestures with the use of sequence files.

# Sample interaction accompanied with screen dumps

It is very difficult to show our gestures using images. We have shown the various gestures to be used in our Youtube video

# BUGS AND DEFICIENCIES

Whenever the game ends with zero gestures/moves, the pie chart showing the last game statistics is blank and can only be seen as a dot on the left-top corner of the screen.

GPS is one of the main features for the game, so the game doesn’t start if there is no lock on the GPS.

# FUTURE WORK

We already have almost 100% accuracy for gesture recognition, but after the gesture is recognized the movement of the snake on the screen takes a second to take effect. We aim to reduce the reaction time as much as we can.

Add a video on the home screen, to show the different gestures for the different movements.

# LINKS & REFERENCES

**https://gitorious.org/f-droid-mirrors/msnake/source/f030e855c2c32af11f2a685c9e8f1232b8dc831c:**

[**https://code.google.com/p/achartengine/**](https://code.google.com/p/achartengine/)

[**http://processors.wiki.ti.com/index.php/SensorTag\_User\_Guide**](http://processors.wiki.ti.com/index.php/SensorTag_User_Guide)

[**https://developers.google.com/maps/documentation/android/**](https://developers.google.com/maps/documentation/android/)

**GitHub**

**https://github.com/divyabattu/Project\_Finalsubmission**

**YouTube:**

**https://www.youtube.com/watch?v=GYG2jwU8mmM**

# 

**PROJECT management**

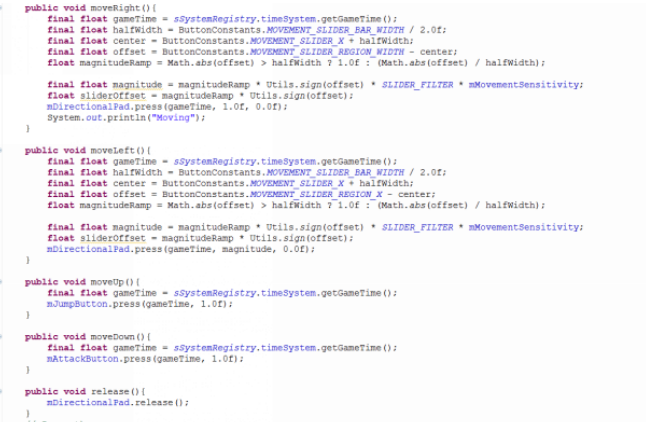
**(Project management Report)**

# FIRST INCREMENT

We have chosen an open source android game, “Replica Island” to replace the touch with motion sensing. The sensors being used were the accelerometer sensor and the gyroscope. The right gesture is used to enable the person move forward continuously whereas the left gesture is used to move backwards continuously. Upward movement is used to enable the person jump continuously and the Downward movement is used to enable the person move downwards continuously. The data generated using the sensor tag from the above movements is then stored in HBase. The relevant data obtained from Solr is displayed by using webservices.

# SECOND INCREMENT

We made changes in the source code of the replica island. There are four motions in the game: Flying, move left, move right, attack(stomp). We took four motions as training data, 8 times each. Now, it can detect if it’s fly or left or right or attack gesture. We made necessary changes to the source code. We used Hidden Markov model and K-means clustering for activity recognition.



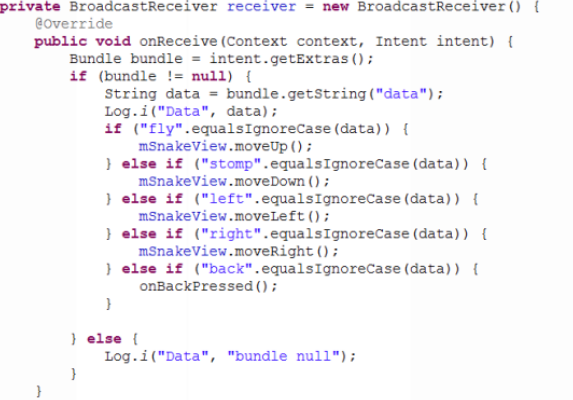
# THIRD INCREMENT

Due to a few limitations in the previously chosen game, we decided to move to another open source android game, “MSnake”, to replace touch with motion sensing. We tested each gesture 50 times to verify the accuracy. We took five motions as training data, 50 times each. Now, it can detect if it’s up, down, left, right or pause gestures. We used the HMM to identify the gestures and K-means to detect the activity. We started with a k value of 10 but decreased it whenever an exception was encountered. We deployed HMMWS war file into local glassfish and converted the text files to sequence files with the recognition of start and end of every gesture.



# FOURTH INCREMENT

The sequence files are trained at the start of every game and once they are trained the gestures are recognized and the corresponding function is called which helps with the movement of the snake. The K value of the five gestures was 7, 9, 10, 8 and 3 respectively. Play game button directs you to the actual snake game whereas the view charts button directs you to another activity where you can view pie charts, maps and other graphs. The data is stored in HBase after the game and is retrieved to use it in the various activity models. We tested each gesture 50 times and plotted the confusion matrix and calculated the precision, recall and f-measure values based on this confusion matrix. We achieved an overall accuracy more than 97 per cent.





# FINAL PROJECT EVALUATION

The main aim of our project was to use the different functions of a game with the use of gestures and show the activity report on the system and user performance. We’ve started with a different game at the start, called replica island. We’ve worked a lot on that game, but due to time constraint we had to switch to the msnake game.By the end of the project we have achieved almost all of the expected features. We are extremely happy with the final outcome of our project.

Agile process was very helpful as it helped us in staying on track and helped all the teammates to come upto the same point in case they got off track. Due to the agile process we were able handle exceptions in the very beginning rather than finishing the whole project and starting it all over again in case an exception was encountered. It helped us to make increments to our project from the point where we had previously stopped. We can say that we have stuck to the project plan schedule right from the beginning.

There was no real management structure within the group. Instead, all of us picked a feature that interested us the most and worked on it. “Too many cooks spoil the dish” was what made us do the things the way we have done. We had planned on this structure from the very beginning and implemented the same. This helped us in working on a particular part and resolving any issues that we encountered while working on that part. This led every team member to work hard on their part and contribute to the project equally. There was no difficulty faced in working together as the whole project is the outcome of each person’s dedication.

We’ve worked hard on the motion recognition part and improved the accuracy from 85 per cent to more than 97 per cent. For this we’ve trained each gesture for 16-18 times and also made use of the gyroscope on the sensor tag for the orientation during the gesture. These factors improved the accuracy drastically. We started with a basic plan, but achieved more than what we expected with the combined group effort. The design of the game may not be complex to look at, but the complexity is all in the code. For the activity report we dynamically generated the pie charts, bar graph, line chart and markers on the map, by retrieving the data from hbase and SQLite. This wasn’t an easy task to accomplish, but we’ve done it.

If this were a real world project, the whole structure would have been different. We would have had a team lead who monitors us and who might help us in resolving an issue whereas here, we had to resolve our own issues. We would have been given a game to work on and would have been provided with the design specifications beforehand rather than letting us choose our own game and implement whatever features we would like.

The project was very informative and took us one step further into today’s technology. It helped us in working with android and in implementing additional features to an existing game. The only suggestion would be about the time constraint. We were unable to add other features due to the time constraint. The work load was very heavy for a two month period.