**2048**

CS5590BD Jagadish Tirumalasetty Murali Krishna Kalvakuri

Fourth Increment Report Prudhvi Nalluri Sowmya Kamaraju

**Project Goal and Objectives**

**Motivation:**

2048 is played on a simple gray 4×4 grid, with numbered tiles that slide smoothly when a player moves them using the four gestures (left, right, top and bottom) Every turn, a new tile will randomly appear in an empty spot on the board with a value of either 2 or 4.Tiles slide as far as possible in the chosen direction until they are stopped by either another tile or the edge of the grid. If two tiles of the same number collide while moving, they will merge into a tile with the total value of the two tiles that collided. The resulting tile cannot merge with another tile again the same move. Higher-scoring tiles emit a soft glow. In order to make a good score in this game, player should follow some kind of pattern like focusing on only three motions i.e., always avoid making one of the four motions.

**Significance**:

2048 is a puzzle game so by using this game we would be able to make some predections of player’s mentality. In 2048 we have used five movements left, right, up, down and restart and used sensor tag to decide which movements are made based on the gestures made by the player.

**Objective:**

The main objective is drawing a sketch of the user mentality on how does he/she treats a problem. How safe sided is the approach taken or how conventional is the problem solving technique. In order to do this we use the motion sensor and build a 2048 android application using the large amount of data generated by the motion sensor i.e., sensor tag and use the data provided by the motion sensor to decide the mood of the player who is playing i.e., he is upset or calm. The game would give out a report with all these calculations of the predictability of user’s move on the basis of previous move.

For this to achieve we first have to train our system by taking different sample players and ask them to play the game and try to solve it completely. This would be primary data for our machine learning algorithm. We plan on training the system with at least 10-20 players playing the game at least 5-10 times to capture as many possible cases in between the game that different people could take different ways around it.

On top of this data we would define a set of rules and trends to define a person’s mentality depending on the stage at which the player is making the move. We are considering the level as well in which the move is being made cause in the earlier stages of the game it is quite simple to add up the numbers and the plan of any player would show his own mark when compare to others from a particular stage of the game.

**System Features:**

● Android Developers Tools (Android 4.4)

● Cloudera Virtual Machine and Hadoop tools

● HBase

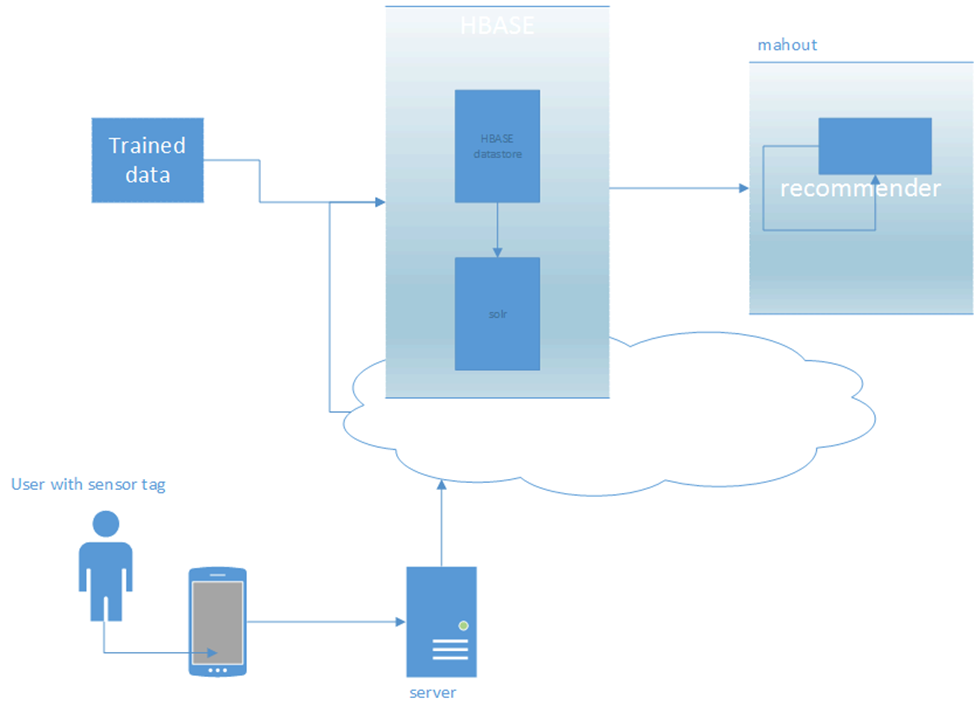
● Mahout

● RESTful services

**Online Application: Game by Motion**

**Devices/Sensors**

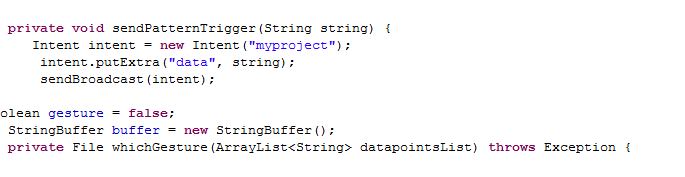
Sensor tag is used as a motion sensor to play the 2048 game in android device (Nexus).



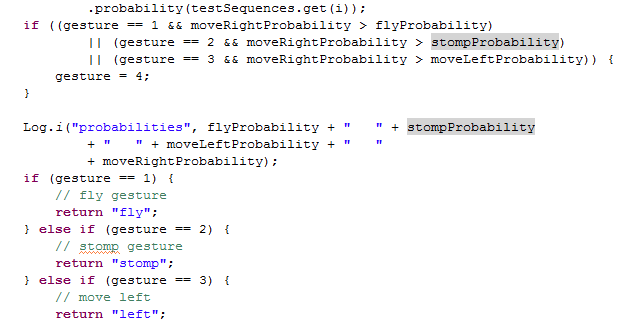
* Our Project mainly focusses on usage of 2048 android application.
* The data obtained from training will be stored in the Hbase (solr).
* Sensor tag will generate the data about the moves made by user.
* Mahout will be used to provide the Players mood by analyzing the recommendations from the trained data.
* Web services will be used to display the relevant information obtained from solr.

**Recognition/Analysis Approaches**

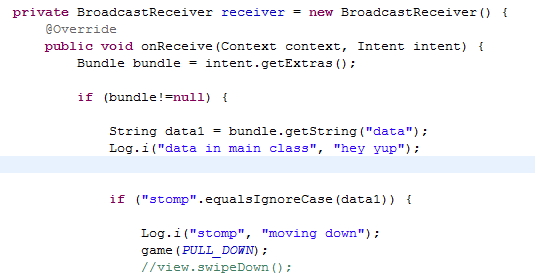
In the 2048 project we have created main\_activity.java which will invoke the connection service for the sensor tag by calling connection service.java where different sensors like accelerometer, humidity of the sensor tag can be enabled and functions were written to collect data and the collected data is modified to standard accelerometer or humidity data by using sensordata.java. An object of Testgesture is created and Train method is called using this object after getting connected to sensor tag. Here we have used **HMM training algorithm** to train the data. After training is done we will call Test method using the Testgesture object and will find what kind of motion our sensor tag represent by comparing with the probabilities of all motions and broadcast the motion with maximum probability. The below given code is for broadcasting motion.



Below given code is of Test method in Testgesture.java which will find what kind of motion our sensor tag represent by comparing with the probabilities of all motions

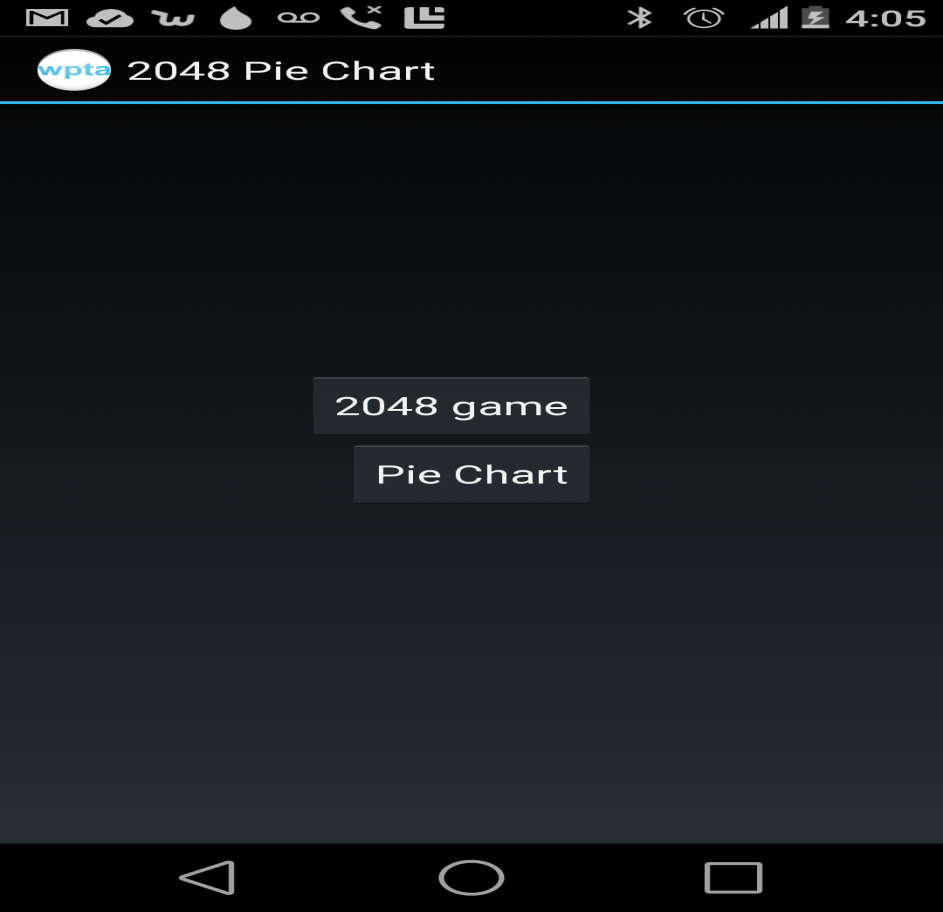


In main\_activity.java we will receive the broadcasted motion from Connectionservice.java and make the necessary movement in the 2048 game. Below code for receiving broadcasted data and making necessary changes in 2048.

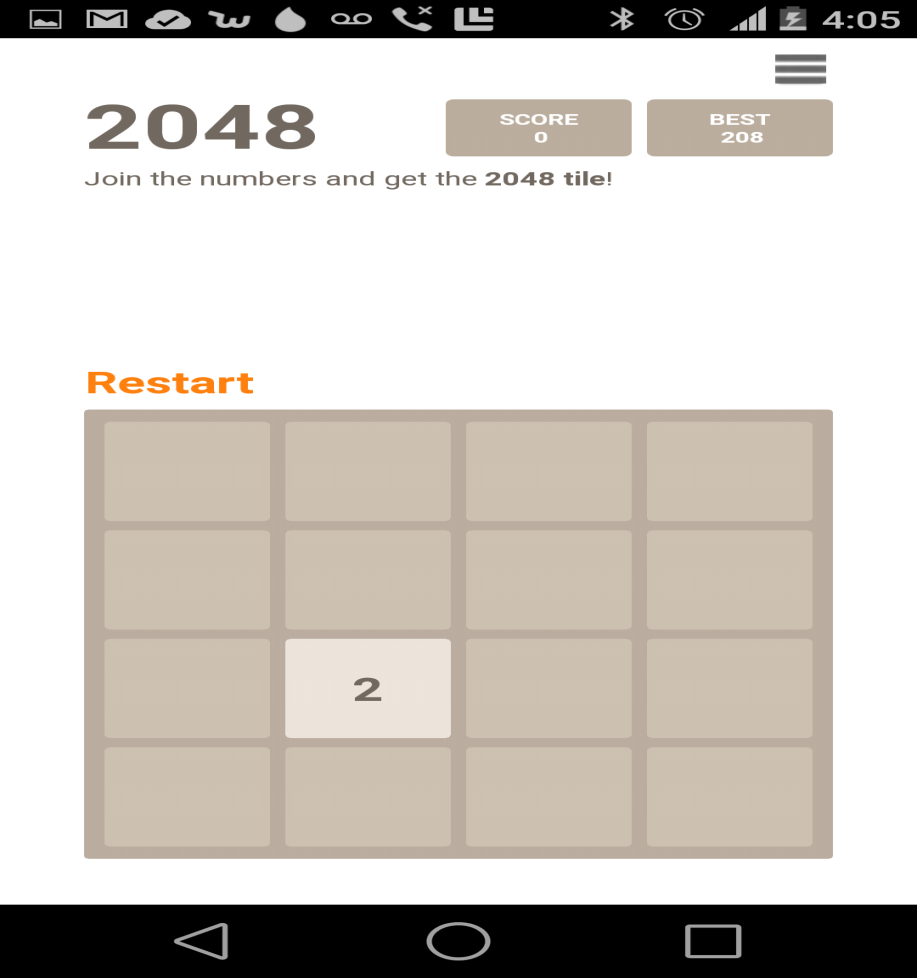


**System Features and Android App GUI**

Below figure is the main screen of 2048 app where we have 2 buttons. 2048-game button will take us to 2048 game and the pie chart button will take us to the piechart of number of left, right, top, bottom, restart moves.



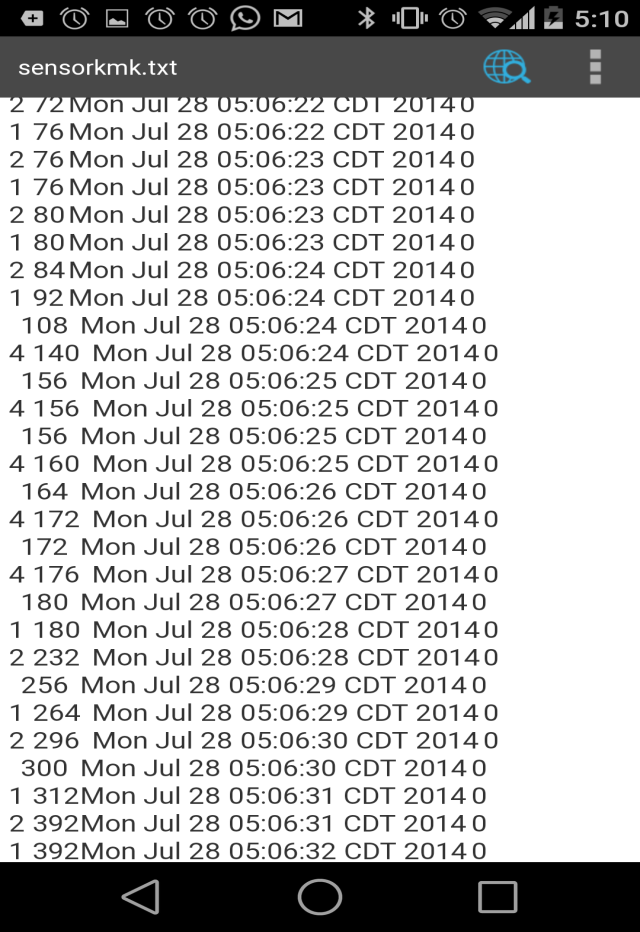
2048 game



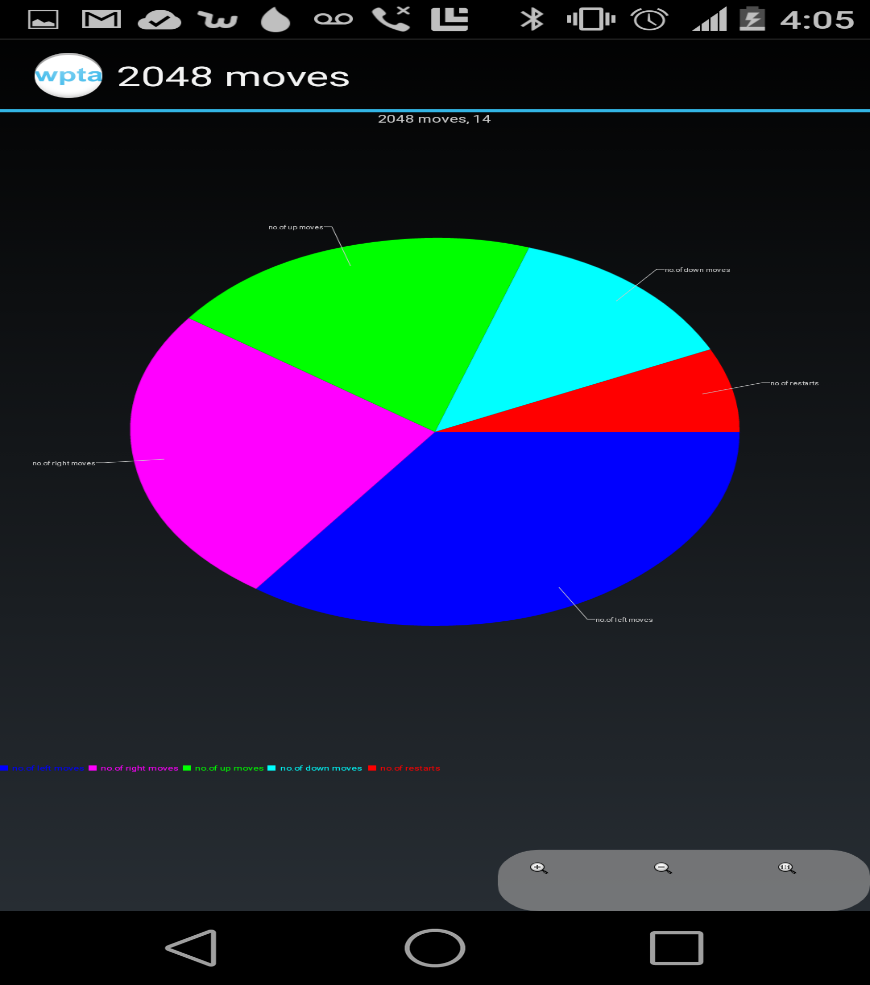
**Offline Application: Activity Report**

When 2048 game is running in background we are capturing moves which we are making along with the move and the score which we obtained by making that move. After the game we are making some evaluations and some graphs based on the data obtained.

Below figure is the data which we obtained after playing the game.

****

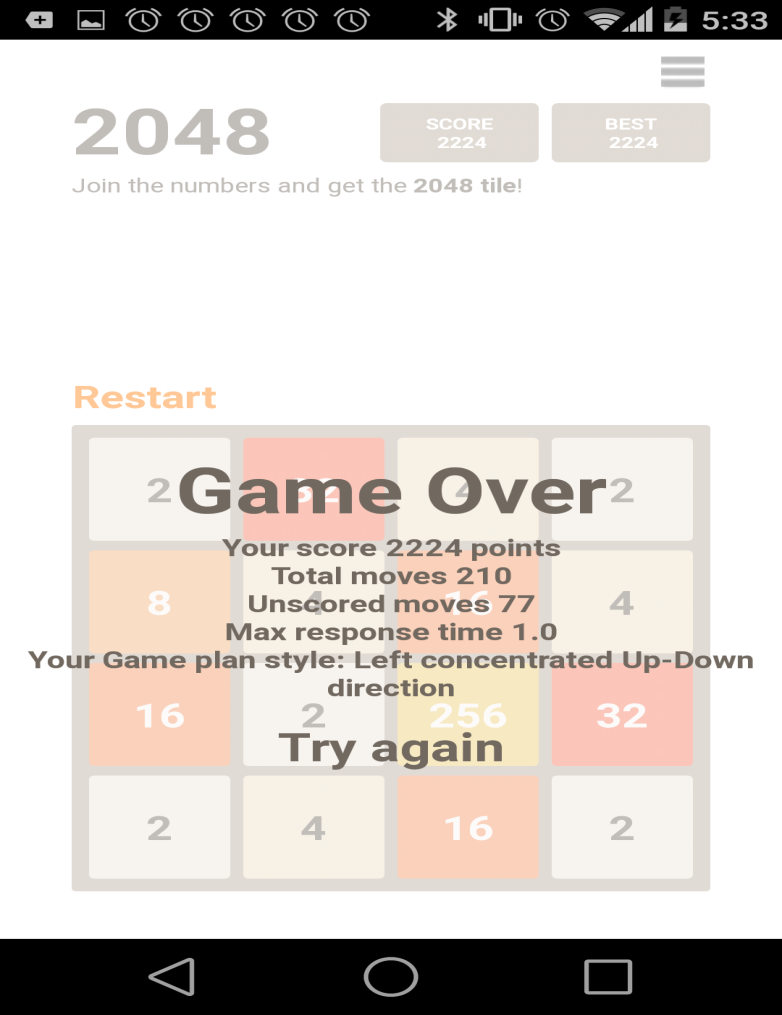
From the data which we obtained from the game were able to draw a pie diagram by counting the number of left, right, up, down, restart moves in order to get the visual representation of analysis which we make.



**Evaluation: Motion/Activity Recognition**

Our 2048 motion sensor app is a single player game and it can recognize five gestures like left, right, up, down, restart.

As in the above data which we have collected, we were capturing each and every move which we are making by using this we were able to find the total number of moves made by the player. We are also showing the number of unscored moves made by the player because by this data player might understand how many moves were wasted without scoring. Max response time is the maximum time taken by the player in order to make a move. We were also able to estimate the playing style of the user by using the data because in 2048 for making good score we must be able to follow a certain pattern like using only three movements in the game.

****

After cross validation we were able to notice that out of 100 gestures made and we were able to retrieve 95 gestures among them 86 gestures are relevant. Total relevant gestures were 90. So the precision was 90.5%, recall was 95.5% and f-measure was 92.9%.

**Limitations**

* Unable to obtain 100% precision and recall.

**Related Work**

* 2048 (open source android game)
* Sensor tag (Open source Android application)

**Youtube link**

<https://www.youtube.com/watch?edit=vd&v=XZsKiUg2wFU>

**Bibliography**

<http://processors.wiki.ti.com/index.php/SensorTag_User_Guide>

<http://en.wikipedia.org/wiki/2048(video_game)>

<https://github.com/uberspot/2048-android>

<https://play.google.com/store/apps/details?id=com.digiplex.game>