**TETRIS**

**A Sensor Based Motion Game**

**Ponnam Balakrishna**

**Kommineni Siva Krishna**

**Boyanapalli Swathi**

**Pathuri Savya Sri**

**Summary:**

Our application Tetris is a sensor based motion game. It is an application developed for android users. Here the game motion is controlled by the sensor tag movements. We selected an open source game called Tetris and established a connection with the sensor tag using the connection service. Data is first collected from the sensor tag for various motions. The data collected is useful for analyzing the direction of the movements and train the system and generates the sequential files. Thus the game runs in the direction of movement of sensor tag. We used activity recognition and motion sensing devices and sensors to perform all the operations collectively. All the data by playing the game is collected and represented in the form of charts.

**Framework Specification:**

The frame work mainly consists of the following stages Android GUI, Sensor tag and testing, training the data. Android GUI is basically an open source android game. We established the connections between the open source project and the sensor tag.

**System Architecture:**

Our architecture consists of sensor tag for generating the motions. First we collect some data from the sensor tag. Those data is collected for different motions. On collecting the data it is analyzed using some of the machine learning algorithms and generates the sequence files for the particular motion. Those generated sequence files are trained and tested for generating the motion to occur when we perform the particular motion. A connection is developed between the sensor tag and the android UI by enabling the Bluetooth and pairing those two devices. While playing the game sensor is connected to the device and by the Bluetooth and connection service, the connected sensor tag generates the motion and the motion compared with the motion recognized from sequence files and moves the object towards based on the motion. Based on that motion the object gets moved in the UI and the game can be played using those sensor.

The below diagram represents the architecture diagram of the developed game which is played using sensor device:



* **Implementation:**

We have chosen a game called **Tetris** played by most of them at their childhood in videogames.

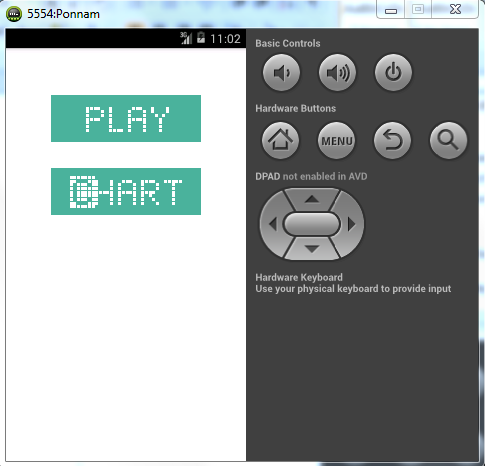
This game sent the different shapes coming down. Those objects can be **Rotate** to change their shape and can be moved towards **Left** or towards **Right.** On making a Row without any space the row will be deleted. Thus if the rows are made with spaces left then the objects get increasing and will touch the high where the game will be finished.

Now we have done the game running by using the sensor tag. We have made a total of five gestures **Right, Left, Rotate, Punch and Down.** The Right is to move the object towards Right, Left is to move the object towards Left, Rotate is to Change the shape of the object, down is to bring the object down in fast and punch as to play the game.

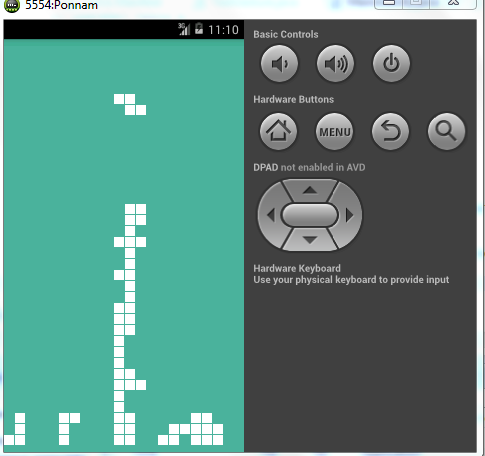
On running the game with sensor devices we get the Play and chart button.

On Punch the play button gets activated and the game will be running by generating the objects. Those objects can be moved towards Right, Left and can also rotate the object using the sensor.

The below screen shot shows the beginning of the game with the play and Chart button:



The below screen shows the objects falling down and which are moved towards left, right and rotated by using the Sensor device:



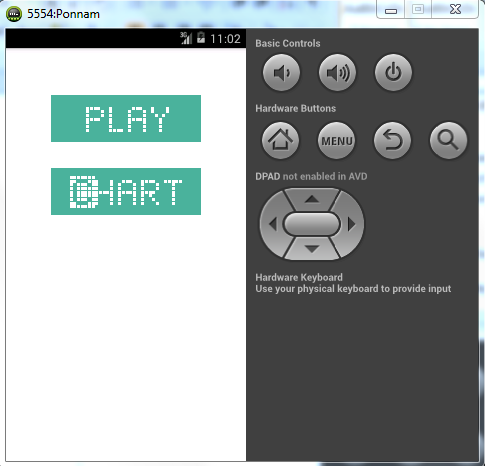
**Generating Charts for the Gestures:**

The game is played by using the sensor tag where generating five different Gestures.

We have collected a report of how many times the particular gesture is being performed. We collected five gestures Rotate, Right, Left, down and Punch. We took the count of each gesture performed while playing the game. The individual Gesture count is sent to the database with the date.

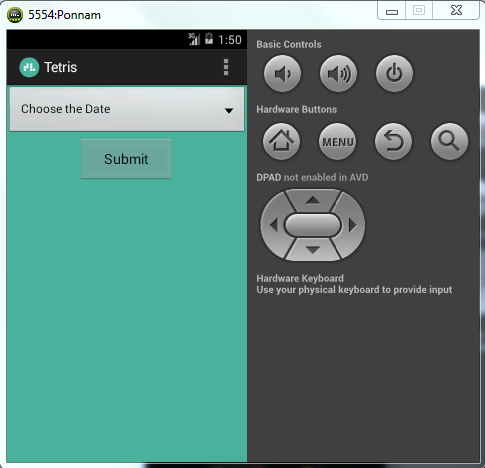
Thus we generate a pie chart for all gestures based on the date by retrieving those data from the database.

The below screen shows the main screen with play and chart button:

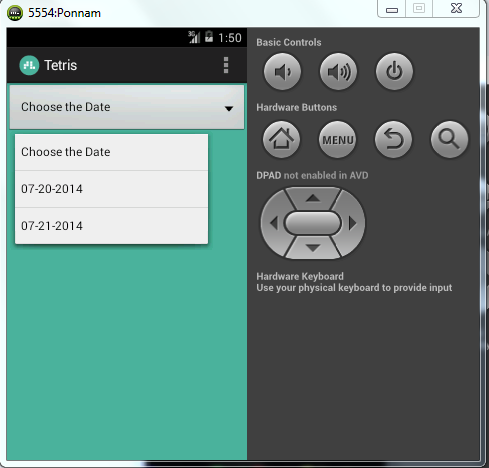


We have a chart button, on pressing the chart button we get the drop down which has all the dates in which we played the game. On selecting the date and pressing the submit button we get the pie chart with gestures we have used.

The below screen shows the selection of Date with drop down and submit button:



The below screen shot shows the selection of date from the dropdown:

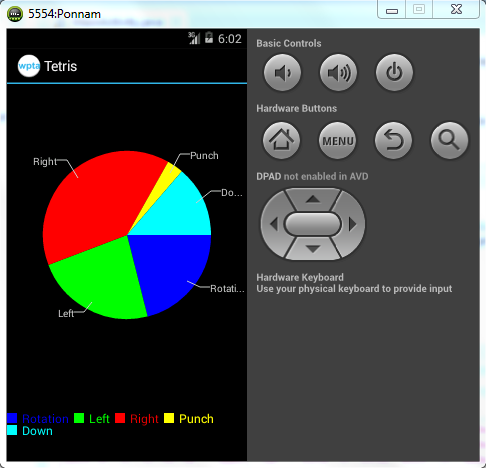


On selecting the date and pressing the Submit button we get the Pie Chart with the gestures we have used.

Here the Date is from the database which selects all the dates with the gesture. It removes the Duplicates and shows the available dates in the drop down box.

On selecting the particular day it fetches all the gestures used and its count and draws the pie chart based on the count of the particular gesture.

The below screen shows the Pie Chart of the Gestures we have used:



* **System Features:**

Tools: R, Hadoop, Android Development Kit, Eclipse kepler.

Operating System: Android

Development Operating System: Windows 7

Programming Language: Java 7.0

Database: HBase

* **Activity Recognition Scenario and Data Collection**
* **Devices/Sensors:**

## Sensors: TI Sensor Tag - CC2541DK-SENSOR

## It is the first development kit designed for the smart phone application developers. It has total of 6 sensors which can sense Temperature, Humidity, Pressure, Accelerometer, Gyroscope, and Magnetometer. With the available sensors we can collect any data and build any applications related to health, education, fitness etc.. Based on the information available from the sensors. Due to the wide variety of the data collection facility from the sensors it can be implemented in various applications in our daily life.

**Devices: Android devices**

* Bluetooth 4.0 compatible android device
* Android 4.3 or above OS
* GPS
* **Data Collection:**

Data is collected using the sensor device for certain actions and gestures that need to be trained to the system for the generation of the sequence files. We collected data for the five varieties of motions. They are left, right, down, rotate and punch. By sensor movements we collect the sample text files for all these actions and later test, train and generate the sequence files for performing the required action. These data is useful for the motion recognition and detection during the game. We used Sensor tag data for the data collection.

* **Motion/Activity Model:**

We used a connection service class to establish the connection between the sensor tag and the android UI. So that the game can be controlled easily with the help of the Sensor tag.

There is a test gesture class for testing the gestures based on the generated sequence files and then acts accordingly to the sensor tag movement. The gestures are analyzed here using some

**Activity Recognition**

With the performed gestures, we are having some actual and predicted values. With those values we have calculated precision, Recall and F-measure for all gestures performed in the game.

Precision is the % of selected items that are correct.

Recall is the % of items that are selected.

We performed 50 gestures for each motion; from these we separated true negative, true positive, false positive and false negative values for all the gestures. By using those values precision and Recall are calculated for each gesture. F-measure is calculated with both precision and recall values. F-measure is used as accuracy.

The below table shows the predicted values for different gestures

**Predicted Values:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activity** | **Left** | **Right** | **Punch** | **Down** | **Rotate** |
| Left | 41 | 2 | 0 | 0 | 7 |
| Right | 5 | 45 | 0 | 0 | 0 |
| Punch | 0 | 0 | 50 | 0 | 0 |
| Down | 0 | 0 | 2 | 40 | 8 |
| Rotate | 0 | 0 | 0 | 2 | 48 |

The below table shows the precision, Recall and F-measure by using the predicted values for different motions which are generated

**Calculations:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Precision=TP/TP+FP** | **Recall=TP/TP+FN** | **F=2\*P\*R/P+R** |
| Left | 41/41+5  =0.8913 | 41/41+2+7  =0.82 | 1.4617/1.7113  =0.8541 |
| Right | 45/45+2  =0.9574 | 45+45+5  =0.9 | 1.7233/1.6574  =0.9278 |
| Punch | 50/52  =0.9615 | 50/50  =1 | 1.9230/1.9615  =0.9804 |
| Down | 40/42  =0.9523 | 40/50  =0.80 | 1.5236/1.7523  =0.8694 |
| Rotate | 48/48+8+7  =0.7619 | 48/50  =0.96 | 1.4628/1.7219  =0.8495 |

**Project Management:**

All the tasks and their day today increments are being updated in the scrum do.

* <https://www.scrumdo.com/organization/umkc94/dashboard>

The work has been equally distributed between the four members of the data with two members dealing with back end and two members dealing with GUI design in each successive increment.

All the stories with time allocations are updated in the scrum do for all the increments.

**Deployment:**

The code and the respective data files are updated to the GitHub in the following link:

* <https://github.com/BP8G6/CS590BD-4th-Increment>

And the YouTube link for video presentation is given below:

* <https://www.youtube.com/watch?v=ZPPb_JBLfB0>
* **Bibliography:**

1. Developing Android Applications –

<http://developer.android.com/training/basics/firstapp/index.html>

1. Sensors: TI Sensor Tag –

<http://www.ti.com/tool/cc2541dk-sensor>

1. Application code for reference

<https://github.com/semenoh/Tetris>