**COMBAT Commander**C:\Users\AyyappaKumar\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\OW7O1GA0\MC900018512[1].wmf Mobile Game

**Supervised By**: Dr Yugyung Lee **Team:**

Dr Shen Vishnu Gude Shiva Pothala Ayyappa Neerukonda

Akhilesh Vemparala

# Third Increment

## Introduction

Advancements in the today's mobile technology created a huge scope to develop all kinds of user interactive applications. Multi player mobile games are more popular and fun to design by using motion sensor devices. This game allows multiple users to play and shoot each other and also it tracks user movements and acts according user simulation in the application.

## Project Goal and Objectives

* **Motivation**

The main motivation of this project comes from popularity of mobile games everywhere. From our childhood playing a game with friends always fun and we dreamt to have our own 3D game that everybody can play with our creatures and other 3D effects. After knowing about Sensor Tags and other user motion detective devices, we found a scope to develop this amazing application.

* **Objectives**

This collects data from different users through sensor tags and acts according to user motion in the shooting game. This would also rates a user in the game by giving performance based score. This kind of applications are designed to provide entertainment to users and show them how sensor tags can track their motion when they play around.

* **Significance**

There are plenty of mobile games available and this just one among them. But we are trying add few more features to it. This can be played on android or windows supporting devices like mobile phones or Tablets. With the animated 3D features user can have live play feeling when operating sensor tags.

## System Features:

* **Activity Models:**

Four hand gestures Left, Up, Stomp, and Punch are used for determining Left, Right, walk and Punch activities. Our offline part is to generate sequence files for the trained data. Here each hand gesture is trained with ten samples. Now these data has been trained and the respective sequence files have been generated for each and every hand gesture.

* **Motion Models:**

We basically have four hand gestures to detect. Since we already have sequence files of the trained data. Now the new gesture data will be collected using the sensor tag and then a sequence file will be generated for these data. Now these sequence file will be compared with the trained sequence files and the appropriate gesture is detected.

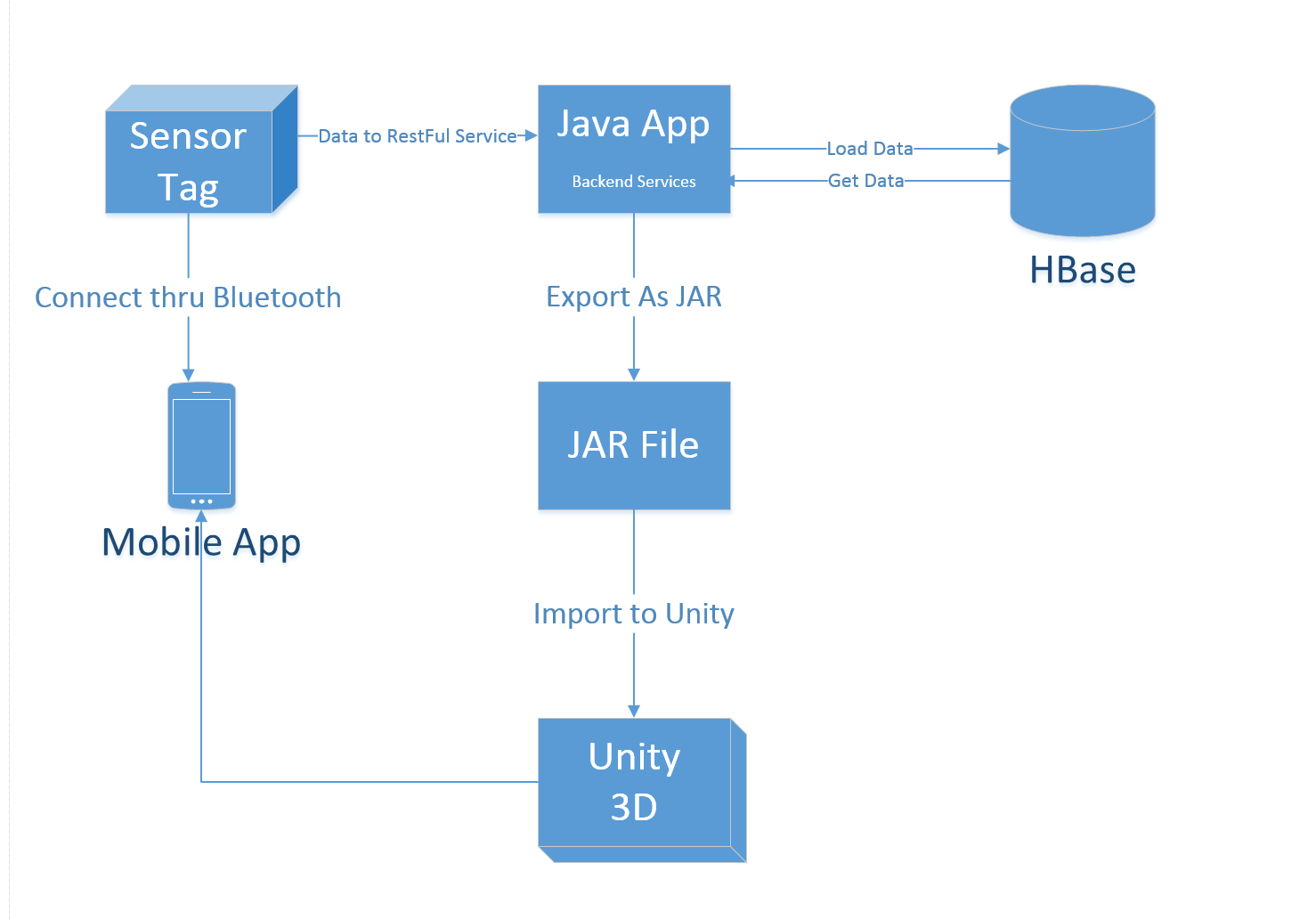
## System Requirements:

* CC2541 TI Sensor Tag
* Android 4.3 and above OS
* Bluetooth 4.0 and above

## Technological and Architectural Requirements:

* Hadoop, H Base
* Eclipse, Java, Android
* Data from sensor tag
* Unity 3D

## Activity Recognition Scenario and Data Collection:



**Offline Application: Activity Report**

**Devices/Sensors:**

Devices which we have used for our emotion detection application are:

* Android mobile device
* Sensor Tag

The sensor tag here we used is a product of Texas Instruments.

**Motion Models:**

We used four hand gestures to detect. Since we already have sequence files of the trained data. Now the new gesture data will be collected using the sensor tag and then a sequence file will be generated for these data. Now these sequence file will be compared with the trained sequence files and the appropriate gesture is detected.

1. Left Hand gesture is represented as Left
2. Right Hand gesture is represented as Right
3. Stomp hand gesture is represented as Walk
4. Punch hand gesture is represented as Shoot

**Application Features:**

Multiple players can join the application server and play shooting each other. Each Player has to shoot thrice (three times) the other player to kill him/her. These application is basically designed for the Android users and the support will be provided starting from SDK version 18 till SDK version 19. Users have to install Unity 3D plugin to load 3D gaming environment. Here overall accuracy of sensor motion is recorded as 95%.

**Recognition/ Analysis Approaches:**

As mentioned earlier in our application we have used the K-NN logic and also HMM model for the detection of hand gestures. Using already prepared sequence files our application works efficiently with minimum time delay of less than one sec. Now in our application on the foreground we will first detect the hand gesture. This will be done by using two buttons namely “Start recognition” and “End recognition” which indicates the start and end of hand gesture respectively. The detect motion has two important methods

* Train()
* Test()

In the train function first we shall take the trained sequence files and train the application. Now once we have the trained results, which will be stored internally in the cache by the K-NN algorithm, we shall take the currently tapped sequence file data and map them to the trained results so as to detect the emotion.

**Evaluation: Motion/Activity Recognition**

**Number of Users:**

The data has been collected from five users and each user will be making five samples of actions for each gesture.

**Types of Motion/Activities:**

We have typically used four gestures in our application to detect different motions. The four gestures which we use for detection are:

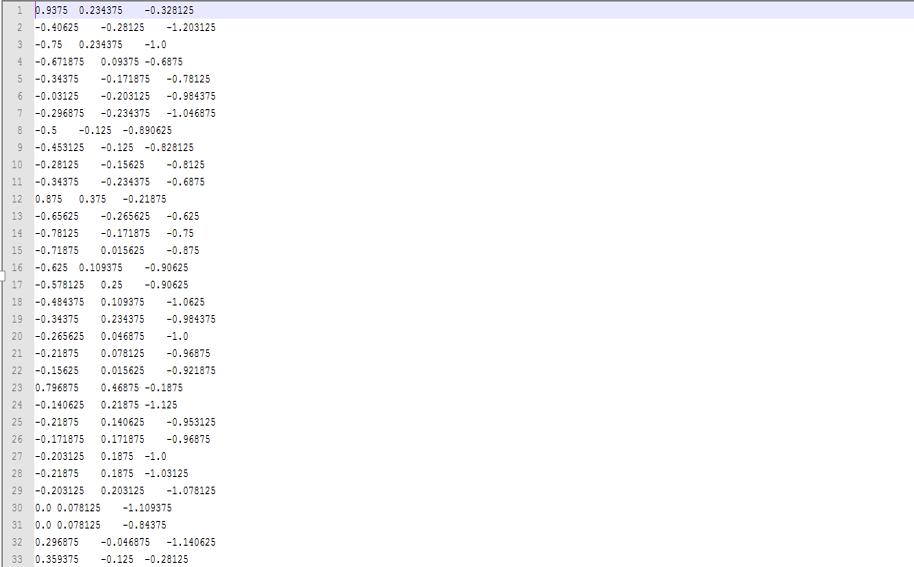
1. Left Hand gesture is represented as Left move
2. Up Hand gesture is represented as Right move
3. Stomp hand gesture is represented as Walk
4. Punch hand gesture is represented as Shoot

**Size of Data/Number of Gestures:**

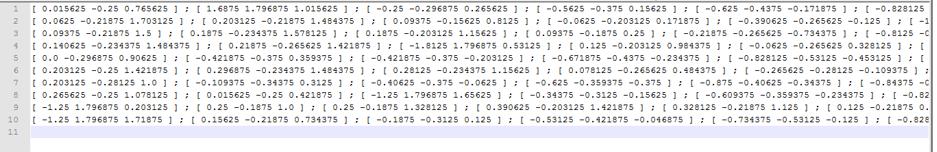
The size of data for each trained gesture is found to be around 25 KB based on the gesture. But each training file has a huge number of values which are found be good enough for detecting the activities.

**Data Preparation:**

The data has been collected using the sensor tags. And after recording of each gesture we will be using these files for machine learning to generate the sequence files. The following are the snapshots of the respective data collected for one kind of gestures.



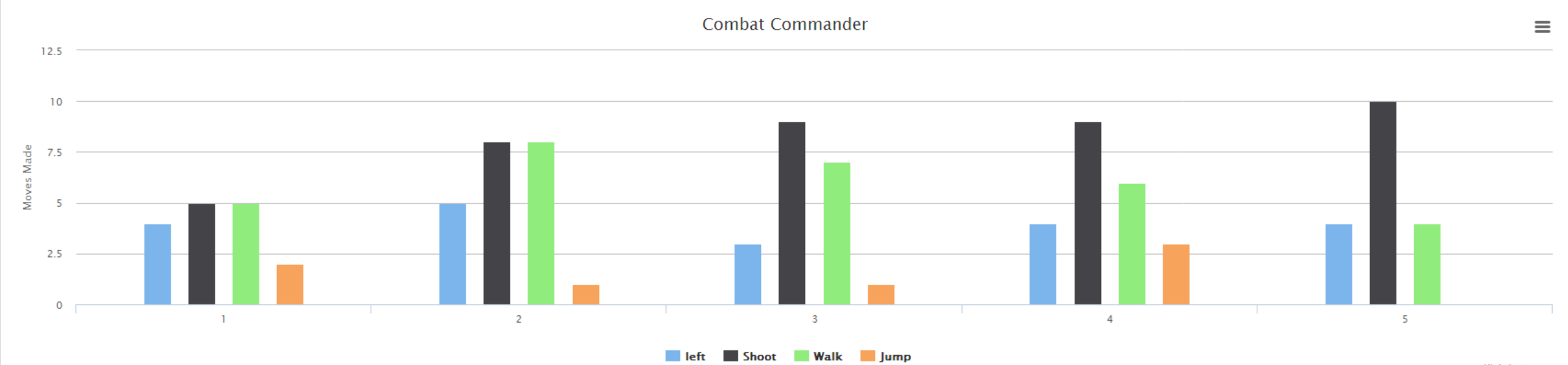
Sequence files:



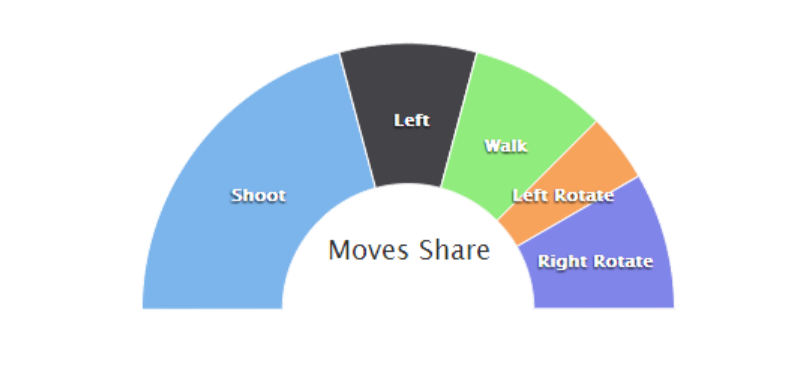
**Reports generated:**

**Steps involved in preparing Offline charts:**

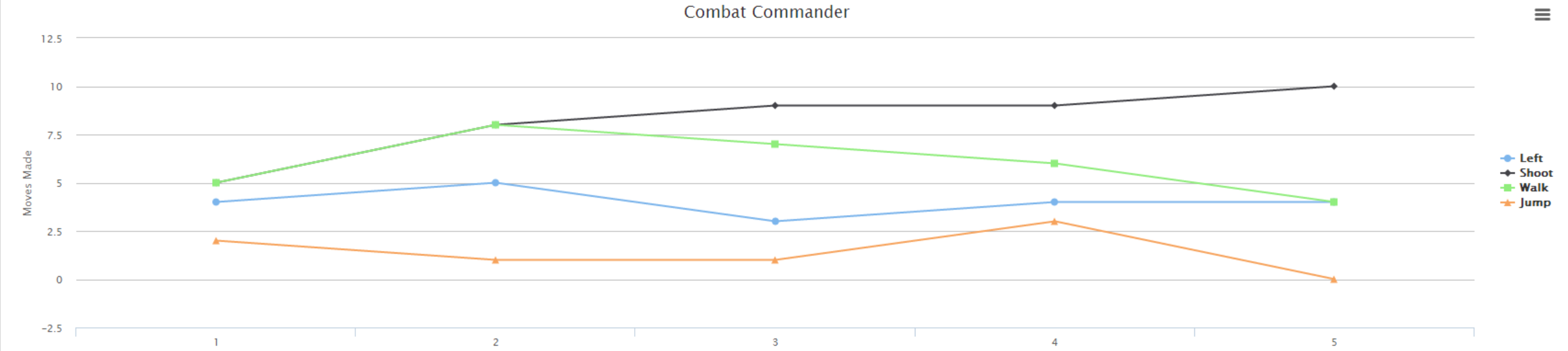
1. Train sequence files through HMM
2. Push data to HBase
3. Pull Data from HBase and send it to High Charts
4. Generate Reports
5. **Number of Moves made**



1. **Moves Share**



1. **Line Chart**



**Confusion Matrix:**

Accuracy has been calculated using trained data collected from sequence files for all sensor motions left, right, stomp, and punch. We have seen 100% accuracy for Left and Right motions whereas for walk it is 91% and for shoot is 89%.

**Observations:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Observations | Left | Right | Walk | Shoot |
| Left | 10 | 0 | 0 | 0 |
| Right | 0 | 10 | 0 | 0 |
| Walk | 0 | 0 | 10 | 0 |
| Shoot | 0 | 0 | 2 | 8 |

**Accuracy:**

|  |  |  |  |
| --- | --- | --- | --- |
| Accuracy | Precision | Recall | F-Measure |
| Left | 1 | 1 | 1 |
| Right | 1 | 1 | 1 |
| Walk | 0.833 | 1 | 0.908 |
| Shoot | 1 | 0.8 | 0.888 |

**Limitations:**

Amount rotations made by each player may not be accurate if the sensor tag is not moved properly.

**Project Video:**

The following is the URL for the video presentation on the application usage and its brief insight:

## Related Work:

**Combat Trigger: Modern Dead 3D**

It is a war game where user can select different weapons and fight in different battle fields.

**Features included:**

Exciting campaign with plenty of battles, futuristic weapon, big bang grenades, armor and health customization, millions of enemies to kill, exotic planets and fallen human cities to fight in, battlefields on exotic planets, combat in fallen human cities

## Bibliography:

<https://play.google.com/store/apps/details?id=com.thunderbull_entertainment.callofdead&hl=en><https://play.google.com/store/apps/details?id=com.csshooter.roadshoot&hl=en>

<https://play.google.com/store/apps/details?id=com.fungame.wars&hl=en>

<https://play.google.com/store/apps/details?id=com.dseffects.StreetCombat&hl=en>

<https://play.google.com/store/apps/details?id=com.rubicon.dev.combatmonsters&hl=en>