Project Plan | Appropriate Gesture

Team Better Recognize
Version 1.1
October 31, 2009

Revision Sign-off

By signing the following, the team member asserts that he/she has read all changes since the last revision and has, to the best of his/her knowledge; found the information contained herein to be accurate, relevant, and free of typographical error.

Name	Title	Signature	Date
Josh Alvord			
Alex Grosso			
Jose Marquez			
Sneha Popley			
Phillip Stromberg			
Ford Wesner			

Revision History

Project Version	Date edited	Reason Edited
Version 1.0	10/31/09	Standardizing content layout and minor revisions

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to introduce the project named Appropriate Gesture. Defining things that the project will contain and how it is planned to go about in the various stages of development. The staff working on it are introduced as well as their roles in development. Also outlined is the resources and constraints faced when developing this software.

1.2 Scope and Objectives

The Appropriate Gesture project will provide a framework for mobile devices to connect into for gesture training and recognition. We will provide one plug-in (for Sun SPOTS) for this framework. Appropriate Gesture will support interaction from up to four users simultaneously, as well as provide consistent and correct recognition based upon the previously trained gestures from these users.

1.3 Background

Appropriate Gesture is closely based on a multitude of other projects such as the Wiigee and Live Motion. We draw from these and other such projects their use of a Hidden Markov Model (HMM), which provides an effective mathematical method for gesture training and recognition. While these projects are useful as references and models for implementation, Appropriate Gesture will be a completely independent framework.

1.4 Document Organization

This document is divided into multiple sections, with each section containing relevant information related to the Appropriate Gesture project.

Section 2 – Definition of Terms: Defines any technical terms or abbreviations used in this document.

Section 3 – Project Overview: Describes how the system handles data flow and how the user interacts with the system.

Section 4 – Resource Requirements: Describes any software, hardware, or personnel requirements needed in development and execution of the project.

Section 5 - Project Management Approach: Defines the staff of the project, their responsibilities, how the project was setup, and how scheduling and tracking was handled

2. Definition of Terms

JSR-82

Java API for Bluetooth communication. There are several implementations of JSR-82. The BlueCove project is an open source and well-developed option for those wanting to use a variety of different Bluetooth stacks and operating systems.

Hidden Markov Model

A doubly stochastic (as opposed to deterministic) math model being used to represent a gesture. Constructed HMMs are then used in recognition. “A statistical model in which the system being modeled is assumed to be a Markov process with unobserved state.”

K-means Clustering

Method of cluster analysis which aims to partition n observations into k clusters in which each observation is clustered with the nearest mean.

Sun SPOT

Sun Small Programmable Object Technology: The wireless motes developed by Sun Microsystems we are using in Appropriate Gesture. Contains a 180MHz 32-bit processor, 512K RAM, and a variety of sensors including a three-axis accelerometer.

Gesture

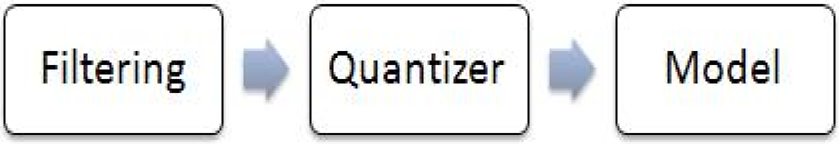
A motion someone makes with their hands such as a circle or a square

3. Project Overview

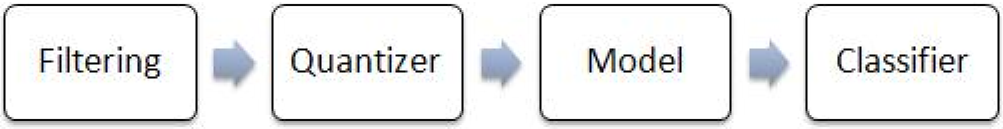
3.1 System

The system will be based on the Wiigee interface and takes inspiration from its system pipeline. It will utilize two pipelines: one for training and one for recognition.

Training:



Recognition:



The different components of the pipeline will be:

- **Filtering**: The accelerometer data obtained from mobile devices may contain noise in the form of vectors that do not contribute significantly to the gesture or multiple adjacent similar vectors. Such vectors will be removed with the application of two filters: the idle state filter and directorial equivalence filter. Idle state is a simple threshold filter while directional equivalence compares adjacent vectors to test for similarity.

- **Quantizer**: The k-means algorithm (with k =14) will be used to codify (or quantize) the filtered vectors during training so as to reduce the amount of effective information coming in from accelerometers. The 3-D vectors will be clustered with k suitable centers to form a “codebook.” This is done to ensure a workable HMM. In recognition, the incoming gesture will be translated into a series of codebook vectors so as to compare against any given trained HMM (a trained gesture).

- **Model**: During training, codified observations (where observations would be training gestures) will be used to form a left-to-right Hidden Markov Model (HMM) that represents the gesture. The HMM must be initialized by default, and will then be iteratively optimized according to each input training gesture. This optimization will be done utilizing a modified Baum-Welch algorithm. In recognition, the constructed model will be used to recognize a gesture in combination with the classifier.

- **Classifier**: A Bayes-Classifier will be used in recognition to compare a test gesture to the trained data. In this case, the most probable trained gesture (above a certain threshold to detect non-matches) will be identified as the performed gesture.

Additionally, the users will have the option of saving the trained data in the form of HMMs in a text file. The format of this text file will be one familiar to the program. That is to say, if in a future instance the user wishes to retrieve the data, opening the file in the created framework will be possible.

3.2 Modes

The user interface of the project will represent four stages. These stages will allow the user to experiment with, implement, and test changes in parameters. The stages are:

- **Training**: This is the gesture training interface. In this interface, the user will be able to select which filters will be applied (idle state, directorial equivalence, etc). The user will also be able to see the 3-D accelerometer data graphed in the interface. Additionally, the user will be able to modify the parameters for the k-means algorithm. Finally, the user will be able to select the sampling rate from the mobile devices in question (if they support such functions). The interface will also give the user the ability to load/save training sets into the system.

- **Recognition**: The recognition interface will allow the user to validate the accuracy of their training. The interface will show the user the computed probabilities for select steps in computation as well as return the most probable gesture from the training data.

- **Evaluate**: The evaluation mode will have the role of enabling the user to monitor the state of the hardware and software system. The hardware system monitoring will include the ability to view the connectivity of mobile devices, whether devices are on or off, as well as if the devices are communicating data at the expected rate, etc. Similarly, the software system monitoring will include the ability to view the status of the framework, the training set being utilized, previous test data, etc. This interface will help the user ensure that the system is functioning normally as well as allow monitoring of the performance and efficiency of the system pipelines (algorithms, classifying, etc…).

- **Demo**: Gesture recognition has a multitude of applications, and the demo stage will let the user apply the trained gestures to an interactive interface. The purpose of the demo will be to provide a useful but entertaining way for the user to test the system.

4. Resource Requirements

4.1 System

Team Better Recognize will need a set of essential software tools in order to create a product that is not only well-designed, but delivered on-time. The following list is a non-exhaustive list of necessary programs.

- Eclipse Java IDE
- Subversion Server and Client
- Netbeans IDE (for initial Sun SPOT development)
- Microsoft Office including Project and Visio
- Adobe Creative Suite (for web design and graphics)
- Windows XP or newer
- WIDCOMM Bluetooth stack (for Wiimote and other Bluetooth device development)
- JRE
- Sun SPOT Manager
- Sun SPOT SDK 5.0

4.2 Hardware

Team Better Recognize will require the following hardware components to ensure proper development and testing.

- PCs and/or Mac computers
- Bluetooth 2.0+EDR Dongles
- Sun SPOTs including base stations

4.3 Constraints

- We are limited to the academic year ending on May 11, 2010
- The computers being used to run and test our software are old, we may need to upgrade to newer computers for better performance.

5. Project Management Approach

5.1 Start Up

5.1.1 Resource Acquisition & Setup

We created a support environment for ourselves in the lab by installing Eclipse, a Subversion and web server, Microsoft Project, Adobe Creative Suite 4, and Camtasia for documenting our process for future software engineering groups looking to get started quickly.

The team started out with Sun SPOTs that had been used by the previous software engineering class. The previous team left documentation on how to get Sun SPOTs installed. We found, however, that the Sun SPOTS presented difficulty in set-up within an environment where students do not have administrator privileges.

5.1.2 Staff

The following are senior students at Texas Christian University.

Phillip Stromberg is a computer science major.
Josh Alvord is a computer science and mathematics major.
Alex Grosso is a computer information technology major.
Jose Marquez is a computer science major.
Sneha Popley is a computer science and mathematics major.
Ford Wesner is a computer information technology major.

In addition to the students above, the team sponsor is TCU professor **Dr. Donnell Payne**.

5.1.3 Staff Training

The team set out immediately upon the project’s inception to understand as much about the mathematics and code behind gesture recognition as possible. We looked to similar projects for guidance and found the Wiigee project to be very similar to what we were trying to accomplish.

In addition to training relating to gesture recognition, the team became familiar with components of our project support environment, specifically Microsoft Project, Adobe Creative Suite, and Camtasia.

5.2 Work Planning

5.2.1 Milestones & Deliverables

Project Support Environment #1
Date: September 21, 2009
The first Project Support Environment will be comprised of installation of required software and study to gain practical knowledge of working environments. Tutorials will be created for relevant software.

Mathematics Research
Date: November 3, 2009
The team members responsible for mathematics research will study the filtering, quantizer, modeling, and classifier algorithms, as well as present them to the rest of the team.

Project Plan v1.0
Date: October 27, 2009
The first version of the Project Plan will be completed by this date. The document will be edited throughout the project.

Project Proposal
Date: October 27, 2009
The Project Proposal will be formally documented by this date and presented to the instructor.

Software Requirements Specification
Date: November 3, 2009
The Project Requirements will be solidly defined, allowing us to move forward with confidence.

Iteration 1
Date: November 24, 2009
The first iteration will consist of a semi-functional GUI, a filtering and k-means system, developed plug-ins and classes, and a developed Sun SPOT test plan.

Iteration 2
Date: December 8, 2009
The second iteration will consist of refined components from iteration one, as well as an implemented but rudimentary training and recognition system.

Iteration 3
Date: February 16, 2010
The second iteration will consist of refined components from iteration one, as well as an implemented but rudimentary training and recognition system.

Iteration 4
Date: March 11, 2010
Iteration four will add the proof-of-concept demo game, and support for the Android platform.

Code Complete
Date: March 11, 2010
By this date we will be code complete

5.2.2 Project Schedule

See attached Gantt chart

5.2.3 Rules & Responsibilities

These are the general areas of responsibility for each group member. While each member is finally responsible for his or her area, they are expected to assist in areas outside their own.

Phillip Stromberg: Project Manager

- Overall schedule planning
- Primary customer communication
- Assistance with general issues

Josh Alvord: Backend

- Study and presentation of algorithms
- Backend coding
- Document editing

Alex Grosso: Quality Assurance

- Hardware and Software safety
- Testing and quality assurance
- Maintenance of website

Jose Marquez: Graphics and User Interface

- Graphical User Interface, design and implementation
- Team graphics and artwork
- Maintenance of document templates

Sneha Popley: Backend

- Study and presentation of algorithms
- Backend coding
- Data Structures

Ford Wesner: Frontend

- Work with Gantt chart schedule
- Integration of backend and interface
- General Quality Control

5.3 Project Control

5.3.1 Project Meetings, Reporting and Communication

Project meetings occur twice a week. At the beginning of the meetings, members go around the table and report what they have accomplished since the last meeting. We go over what needs to be done next, make sure that the whole group understands where we are now and how close the deadline for that item is.

These weekly activity reports collected at the beginning of our meetings are posted to our website for everyone to see what has been done that week. Most importantly, it allows the sponsor, Dr. Payne, to keep tabs on who is not pulling their weight or who is going the extra mile.

Each team member is responsible for communicating with the rest of the team even outside of meetings. If a member is going to change something about the project (perhaps the code or documentation) they should let the others know it by e-mail if at all possible. The Subversion server does keep us from stepping on each other's toes but is certainly not a replacement for team communication.

5.3.2 Schedule Tracking and Review

The project schedule will be maintained through weekly activity reports and plotted on the Gantt chart. If a task is not being completed, the rest of the team will take steps conducive to productivity, either by motivating the responsible person or moving the task responsibility to a different member.

5.3.3 Requirements Control

Ensuring requirements are to specification will be accomplished through the use of class diagrams and demos in class and meetings. In addition, the team will do everything possible to create requirements in a group environment to minimize discrepancies in understanding of said requirements.

5.3.4 Quality Control

Each stage of our system will be rigorously tested and re-tested to validate requirement functionality and safety. If necessary, a team member will be appointed to oversee testing to ensure what is produced is deliverable.

5.4 Risk Assessment

Risk	Probability / Severity	Mitigation Strategy
Performance issues on SPOT or computer	Medium/High	Re-factor code for higher performance
Tasks more complicated than initially thought, deadline in risk of being missed	Medium/High	Re-organize group, double efforts
Wiigee Project does not work as expected	Medium/High	Devote extra time to understanding existing project
Team Member sick or excessively absent	Low/High	Early documentation will allow others to pick up the slack
Server failure causes data loss	Low/High	Make frequent backups of server data
Requirements must be changed	Medium/Medium	Early documentation will allow more flexibility later