User Manual

FROG Recognizer of Gestures

Team Better Recognize

Version 2.0

April 28, 2010

Revision Sign-off

By signing the following, the team member asserts that he/she has read the entire document and has, to the best of his or her knowledge found the information contained herein to be accurate, relevant, and free of typographical error.

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

Revision History

The following is a history of revisions of this document.

Document Version	Date Edited	Changes
Version 1.0	03/09/10	Initial draft
Version 1.1	04/15/10	Section content complete
Version 2.0	04/28/10	Finalized screenshots, updated log description

Table of Contents

1. Introduction

1.1 Purpose

This document is intended to provide assistance to any user of the FROG system. It provides a general introduction to FROG, the installation procedure, and a walkthrough of the four modes and non-mode specific utilities. Additionally, this manual provides a troubleshooting guide and describes options for advanced users.

1.2 Scope

This document will be useful as a reference to first time users of FROG as well as users already familiar with motion-based gesture recognition. With the appropriate plug-in installed, the FROG framework supports input from multiple heterogeneous mobile devices. Explanations of plug-in specific features are described in their respective manuals.

1.3 Definition of Terms

Gesture

A continuous combination of motions made by an individual (usually with the hands) that are related and meaningful as a whole. Gestures are the entities which shall be modeled as well as recognized by the FROG project.

Hidden Markov Model (HMM)

Double stochastic mathematical model that forms the foundation for the training and recognition algorithms used in FROG. For a more in-depth explanation, see the FROG Software Requirements Specification.

Training Instance

A training instance is a single motion of the mobile device by a user representing a gesture to be trained.

Training Session

A training session is a collection of training sets created by a user. A session is saved in a file format and reloaded to perform recognition. A training session can be thought of as a project file containing representations of multiple gestures.

Training Set

A training set is a sequence of training instances created by the user in order to train a gesture. A training set can be thought of as the complete set of data used to create an HMM representation of a gesture.

2. Describing the System

2.1 Key Features

FROG is an acceleration-based gesture recognition system. With the appropriate plug-in installed, the user can connect to any FROG-enabled mobile device containing an accessible three-axis accelerometer and perform gesture training and/or recognition. FROG is divided into four modes, each briefly described below.

2.1.1 Training

This mode allows one user to connect a 3D acceleration sensing mobile device and train the system with multiple gestures. The training data is captured, analyzed, and saved for later use in the recognition, evaluation, and demo modes.

2.1.2 Recognition

This mode allows up to four users to simultaneously connect to FROG for gesture recognition. Each user must load a library of previously trained gestures in order to perform recognition. One library can be used for multiple users but this is not recommended for best results. When a gesture is received in this mode the system will either display the gesture that best matched or notify that it was unrecognizable.

2.1.3 Evaluation

This mode allows one user to connect and evaluate the recognition accuracy. In this mode the system prompts the user to perform a specific gesture and tracks whether it matched the requested gesture, another gesture, or was not recognized. The user can select what gestures to evaluate, how many times each gesture should be prompted, and if the gestures should be presented in a random order. Evaluation results are presented on-screen.

2.1.4 Demonstration

In this mode up to four users can connect to play a competitive game using pre-trained gesture libraries. Each library should be user specific for the best recognition accuracy. The object of the game is to demonstrate the practical applications of the gesture recognition technology employed by FROG.

2.1.5 Non-Mode Specific Utilities

FROG Console: This console can be set to display and log any input/output of the system. Log and display settings can be configured from the Terminal Controls section of the interface window.

Device Panel: This pop-up panel is used to locate and connect available devices to the system as well as to enable filters. The discover button will fill a list of available devices, displaying their device type (i.e., SPOT, Android, Windows Mobile HTC Fuze). To connect a device simply select it from the list and hit the connect button.

2.2 Environment

FROG requires the Java Runtime Environment version 6.0. FROG was tested extensively on a 3.2GHz machine with 1GB RAM running Windows XP. It was also observed to run equally well on a 1.6GHz Intel Atom-based netbook running Windows XP and Windows 7.

3. Installing, Starting, and Stopping the System

3.1 Installing the System

Prerequisites for FROG include the following:

- Java Runtime Environment (JRE)
    - To download and install the latest version of JRE please visit http://www.java.com/en/download/manual.jsp
- Project self-extracting executable (includes SPOT plug-in)
    - Double-click to extract all files to target destination.
- Device-specific Plug-ins
    - FROG alone cannot interface with any devices. Plug-ins for specific devices (other than Sun SPOTs) must be added.
    - Refer to the device specific plug-in manuals
- FROG enabled mobile device (see appropriate plug-in manual)
    - Appropriate software on mobile device

3.2 Starting the System

To start up FROG simply double click the .bat file obtained from the project self-extracting executable. Once loaded, the main FROG screen should look like Figure 3-1.

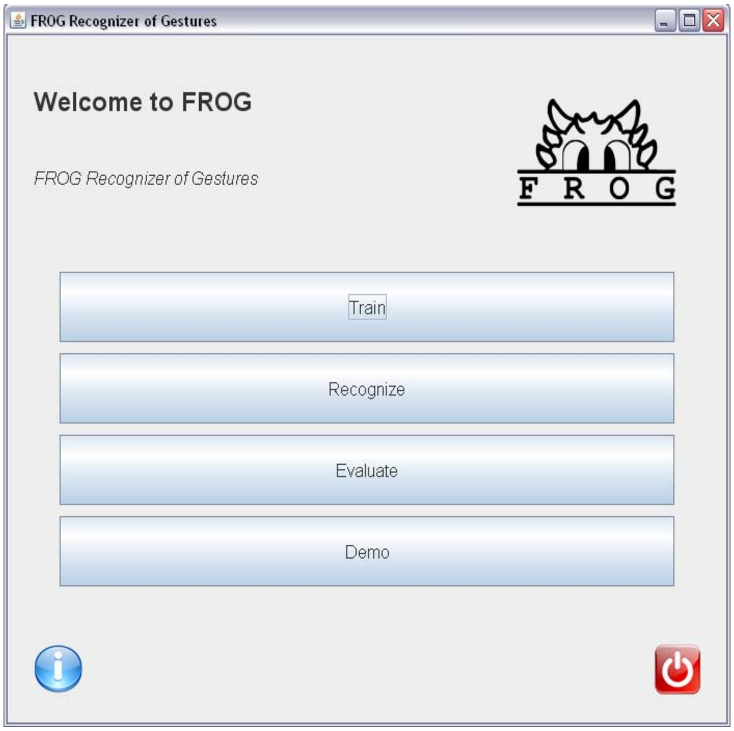


Figure 3-1: The Main Panel of FROG

From this screen a mode may be selected, team information can be displayed by selecting the bottom left information icon, or the program can be exited through the bottom right power button icon.

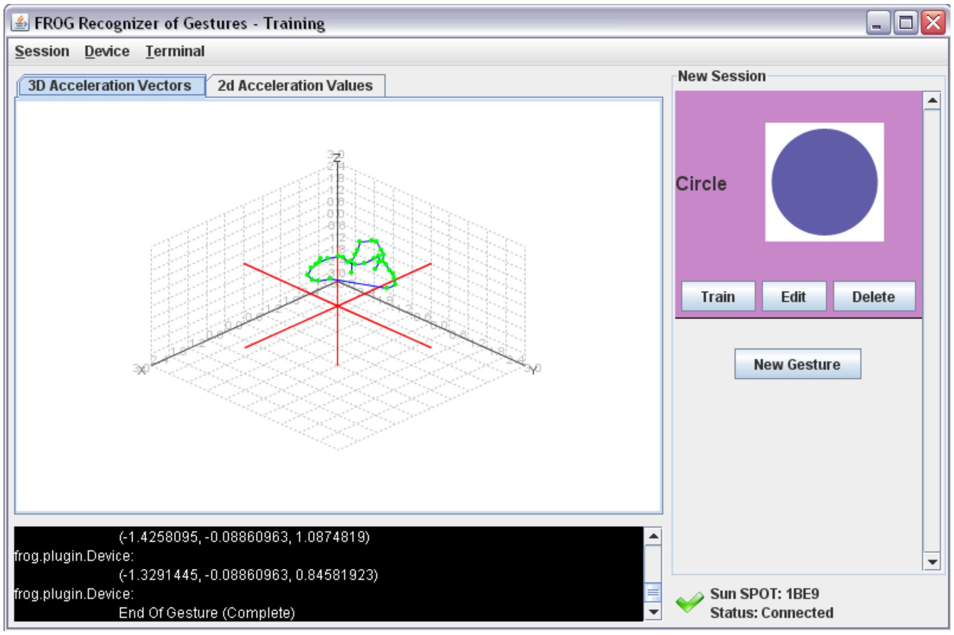
3.3 Stopping the System

The system may be stopped at any time by closing the main window. Stopping the system with unsaved gesture data will result in data loss.

4. System Walkthrough

4.1 Training Mode

Before any recognition or evaluation can be done, a session of gestures must be trained. This is accomplished in the Training mode. Training may be entered by clicking the corresponding button on the main screen of FROG (Figure 3-1). Figure 4-1, shown below, demonstrates a training session in progress.



```
                (-1.4258095, -0.08860963, 1.0874819)
frog.plugin.Device:
                (-1.3291445, -0.08860963, 0.84581923)
frog.plugin.Device:
                End Of Gesture (Complete)
```

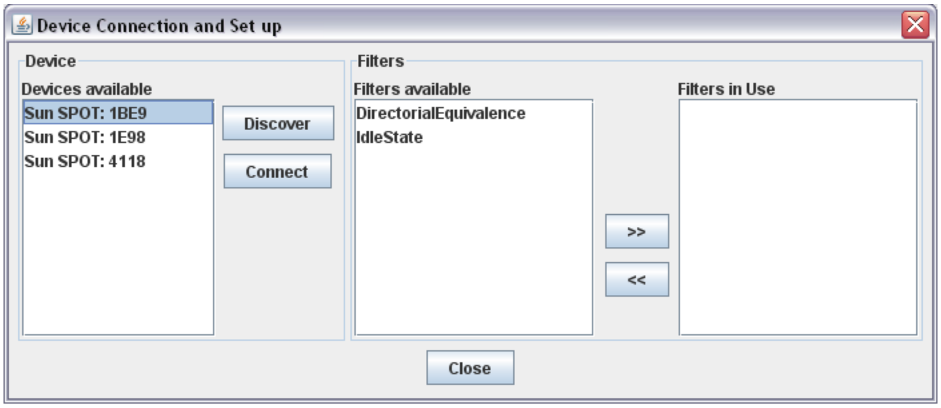
Sun SPOT: 1BE9
Status: Connected

Figure 4-1: The Training Panel

4.1.1 Connecting a Device

In order to obtain acceleration data for training, FROG must first connect to a device via the Device Connection dialog. This is performed from the Device menu on the toolbar by selecting "Configure Device". The Device Connection and Setup window should appear.

If plug-ins have been loaded correctly and the device to be used has been correctly configured (please see a device's specific plug-in manual for more information), pressing the "Discover" button should result in the "Devices available" list being populated with device names as they are found (Figure 4-2).

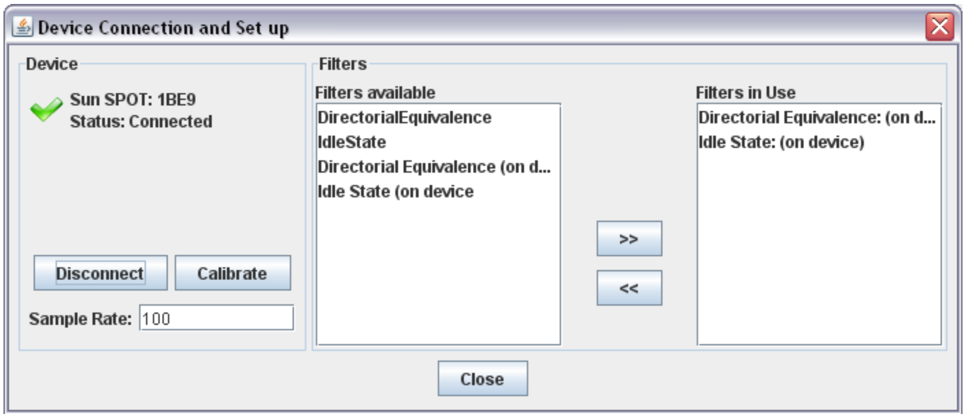


Sun SPOT: 1BE9
Sun SPOT: 1E98
Sun SPOT: 4118

DirectorialEquivalence
IdleState

Figure 4-2: The Device Connection and Setup Dialog

Selecting a device followed by pressing Connect will cause FROG to connect to that device and begin listening for communication (Figure 4-3). Any device-specific filters will also appear in the "Filters available" list. Sampling rate is given in samples per second and may be subject to limitations imposed by the particular device. Calibrate will perform device-specific calibration. (Refer to the device’s plug-in manual for more information.)



Sun SPOT: 1BE9
Status: Connected

Sample Rate: 100

DirectorialEquivalence
IdleState
Directorial Equivalence (on d...
Idle State (on device

Directorial Equivalence: (on d...
Idle State: (on device)

Figure 4-3: Device connected and on-device filters added

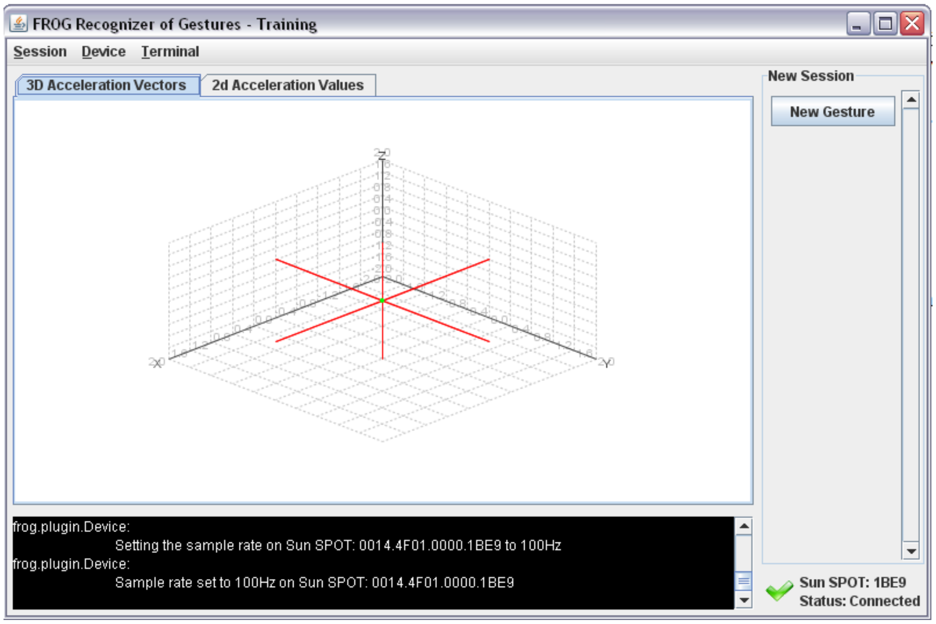
4.1.2 Enabling Filters

Enabling filters is accomplished by clicking the desired filter in the "Filters available" list and clicking the ">>" arrow button to add them to the "Filters in Use" list. Upon clicking the arrow button, a prompt will appear for parameters for that particular filter. Two filters come with FROG: Directorial Equivalence and Idle State. Filters are applied in the order they are added. After filters, calibration, and sampling rate are entered, the connection pane may be closed.

Additional information about filters can be found in section 4.5.1 - Filters.

4.1.3 Training a Gesture

To train a gesture, it is important to first know how FROG operates. FROG requires multiple "instances" of a gesture for good recognition results. For example, to have FROG recognize a circle, begin by clicking “New Gesture” on the right side of the panel (Figure 4-4).

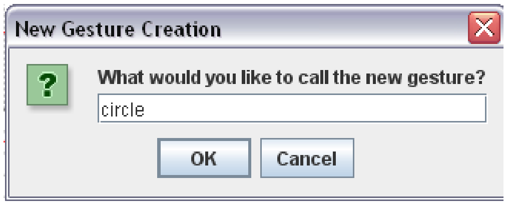


```
frog.plugin.Device:
        Setting the sample rate on Sun SPOT: 0014.4F01.0000.1BE9 to 100Hz
frog.plugin.Device:
        Sample rate set to 100Hz on Sun SPOT: 0014.4F01.0000.1BE9
```

Sun SPOT: 1BE9
Status: Connected

Figure 4-4: Creating a new gesture

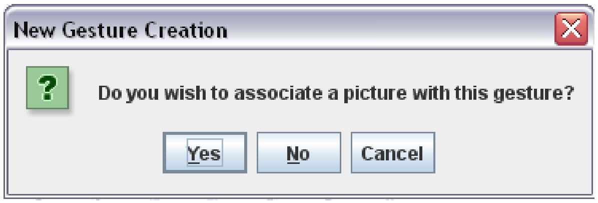
An input dialogue will pop up asking for a gesture name (Figure 4-5), and then a pop-up prompt for an image (.gif format) to associate with the gesture will appear (Figure 4-6). This will bring up a file dialogue to select an image for association. While an image is not strictly necessary, it helps identify gestures at a glance.



What would you like to call the new gesture?

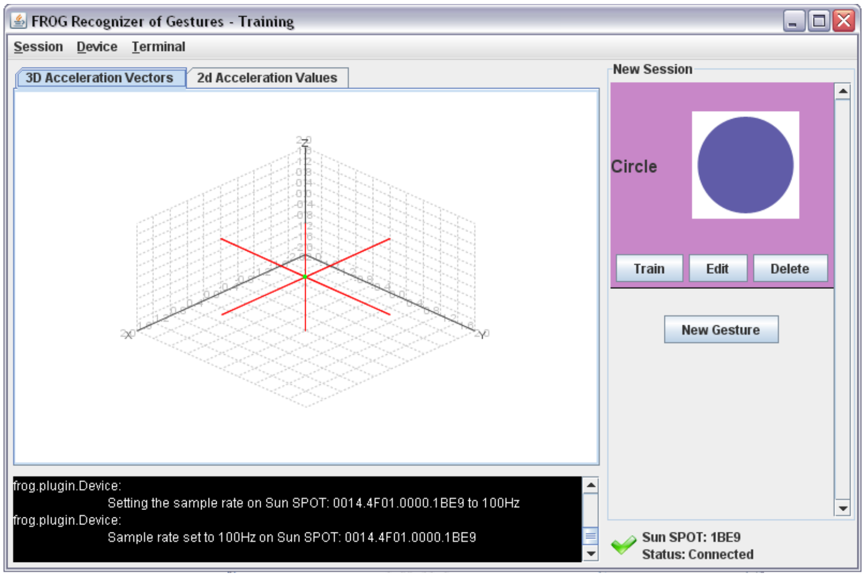
circle

Figure 4-5: New Gesture Naming



Do you wish to associate a picture with this gesture?

Figure 4-6: New Gesture Picture Association

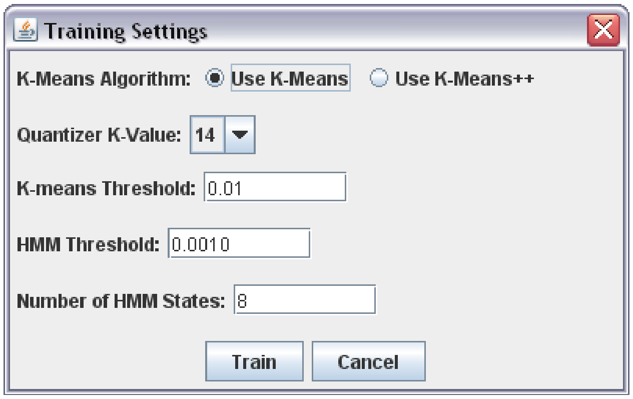


```
frog.plugin.Device:
        Setting the sample rate on Sun SPOT: 0014.4F01.0000.1BE9 to 100Hz
frog.plugin.Device:
        Sample rate set to 100Hz on Sun SPOT: 0014.4F01.0000.1BE9
```

Sun SPOT: 1BE9
Status: Connected

Figure 4-7: New gesture (circle) created and picture loaded

At this point, begin performing the gesture. Hold down the device's gesture start button (consult the device's plug-in manual for more information) and do not release the button until the gesture has been completed. It is recommended the gesture length not exceed three seconds. Do this repeatedly, ideally at least ten times. When all "instances" of this gesture are created, you may click "Train" on the particular gesture in the “Gestures” side of the training panel. When “Train” is clicked, a “Training Settings” menu will pop up (Figure 4-8). From here various parameters can be modified. The default settings are suitable for the purposes of this walkthrough. To start working on another gesture, follow the steps starting with clicking the “New Gesture” button.



K-Means Algorithm: ● Use K-Means ○ Use K-Means++

Quantizer K-Value: 14

K-means Threshold: 0.01

HMM Threshold: 0.0010

Number of HMM States: 8

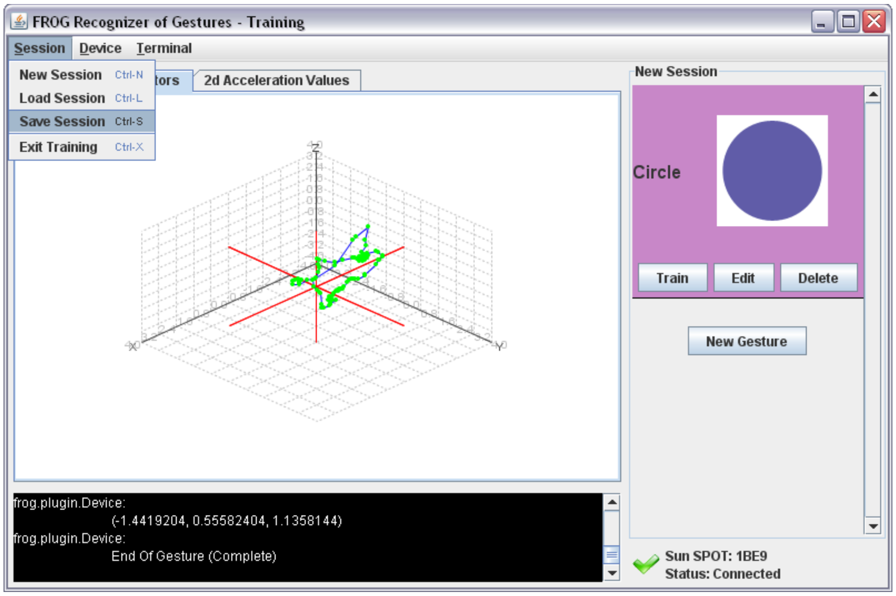
Figure 4-8: Training Settings

Although this example suggests to “Train” after completing all instances of one gesture set, this is not required. The user may “Train” after entering a few gesture instances, add more, and train again. The user may also wait to train after multiple gesture sets have been entered. Additionally the user can return to a particular gesture set, after working with other gestures, and train. All should work equally well.

It is important to note that performed gesture instances will apply to the selected gesture (highlighted). Click the gesture to select it.

4.1.4 Saving a Session

After gesture instances are created and trained, they can either be saved for later use or used in other modes without saving. Save a session for later retrieval by simply going to the "Session" menu (Figure 4-9) and selecting "Save Session." This will cause a save dialog to open in which a location and name for a .frog file can be chosen that will contain all gesture-related data.



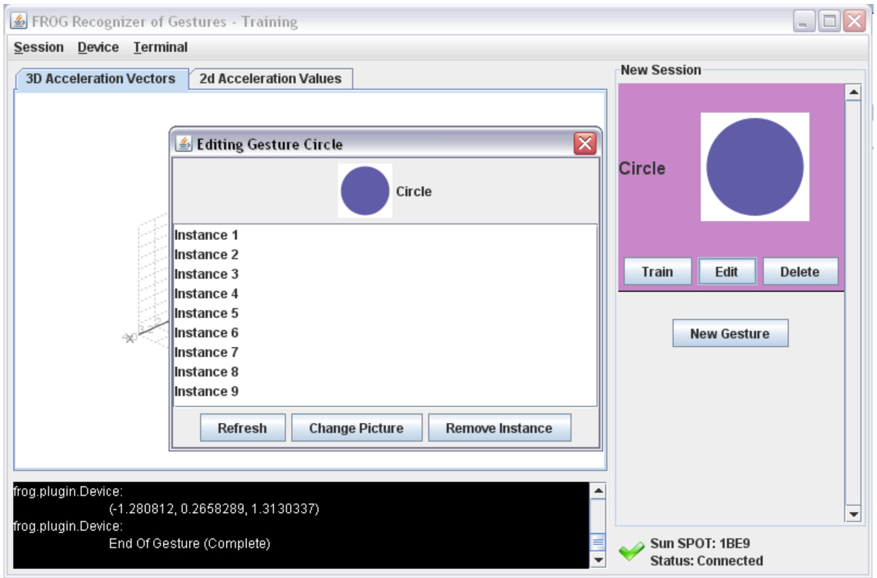
```
frog.plugin.Device:
                (-1.4419204, 0.55582404, 1.1358144)
frog.plugin.Device:
                End Of Gesture (Complete)
```

Sun SPOT: 1BE9
Status: Connected

Figure 4-9: The Training Panel with Session Dialogue open

4.1.5 Editing a Gesture

From the individual gesture under the “Gestures” side of the training panel the “Edit” and “Delete” options can be selected. “Delete” will delete the entire gesture set and remove all training instances. From the “Edit” screen (Figure 4-10) the picture associated with the gesture can be changed, or individual gesture instances can be removed. The list can be refreshed if it does not reflect the actual number of gesture instances trained.



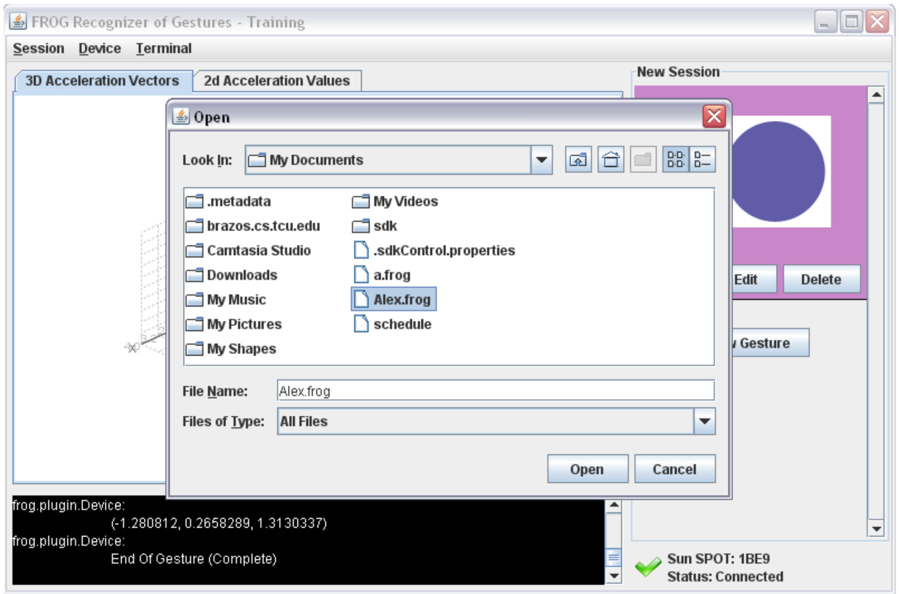
```
frog.plugin.Device:
                    (-1.280812, 0.2658289, 1.3130337)
frog.plugin.Device:
                    End Of Gesture (Complete)
```

Sun SPOT: 1BE9
Status: Connected

Figure 4-10: The Training Panel with Gesture Edit panel open

4.1.6 Loading a Session

Loading a session is similar to saving it. Simply go to the "Session" menu and choose "Load Session." This will open a dialog like the one depicted in Figure 4-11.



Look In: My Documents

File Name: Alex.frog

Files of Type: All Files

```
frog.plugin.Device:
                    (-1.280812, 0.2658289, 1.3130337)
frog.plugin.Device:
                    End Of Gesture (Complete)
```

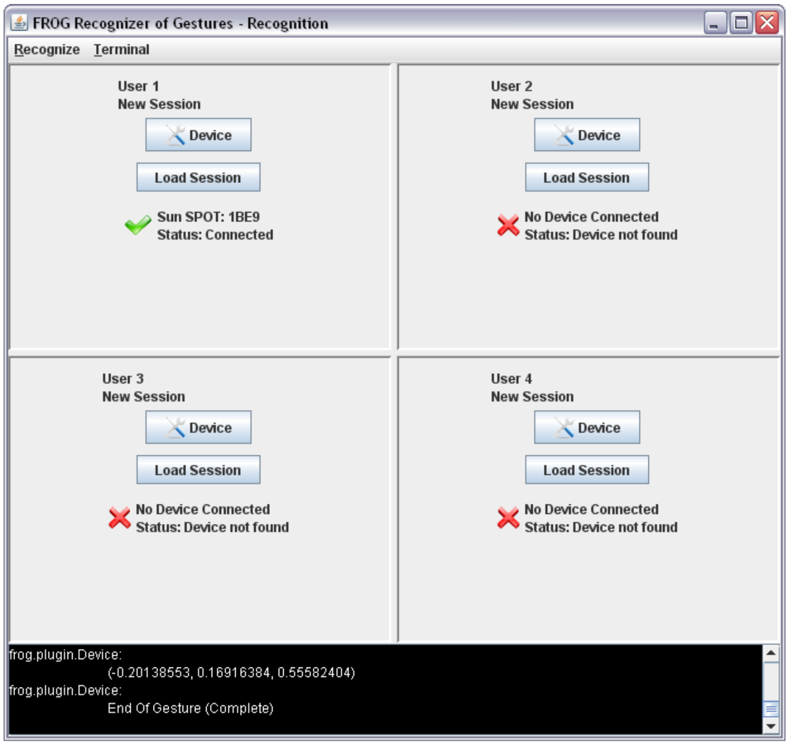
Sun SPOT: 1BE9
Status: Connected

Figure 4-11: The Training Panel Load Session open and Gesture Session Alex.frog selected

Upon loading a session no further action is required unless a device is not yet connected. Loading a session prevents a user from having to train new gestures every time FROG is used. Gesture sets, as well as individual instances, can be added or deleted from a loaded session. When a session is loaded, it can be re-modeled with different parameters by selecting “Train” once more for the particular gesture to modify.

4.2 Recognition Mode

Recognition mode allows up to four users to be simultaneously connected and perform gestures. Each user must have a session loaded to have a gesture reference library. Clicking "Recognition" from the FROG main menu will bring the user to the screen depicted in Figure 4-12.



Sun SPOT: 1BE9
Status: Connected

No Device Connected
Status: Device not found

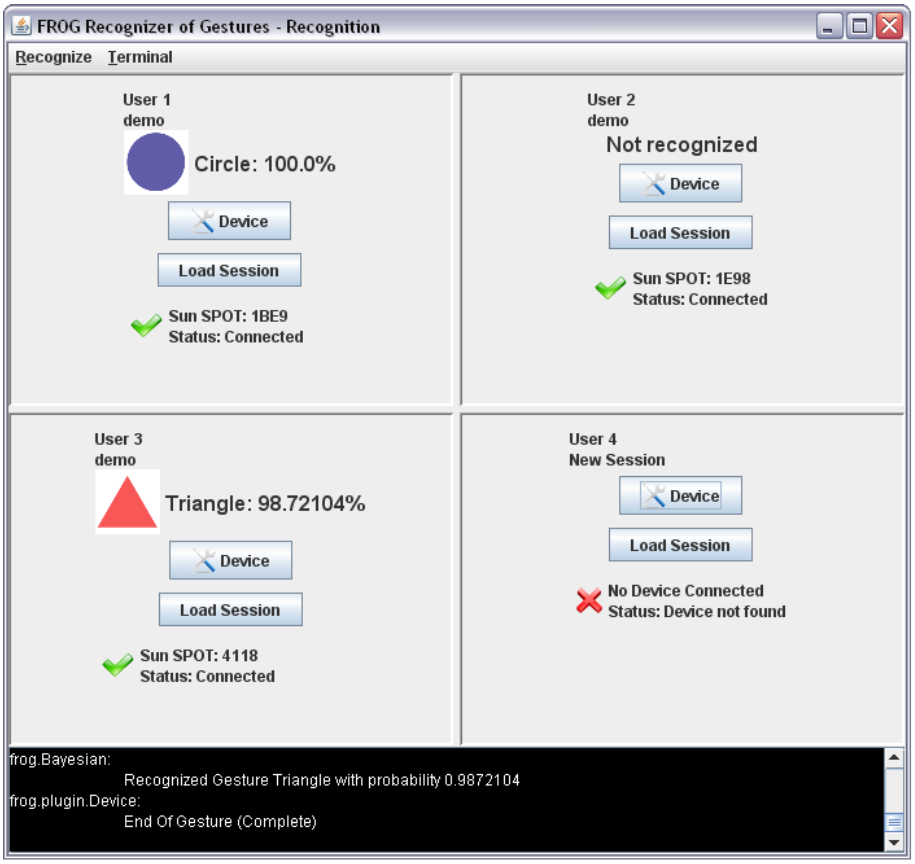
No Device Connected
Status: Device not found

No Device Connected
Status: Device not found

```
frog.plugin.Device:
                (-0.20138553, 0.16916384, 0.55582404)
frog.plugin.Device:
                End Of Gesture (Complete)
```

Figure 4-12: Recognition Panel

Each user must connect their device (Device button) and load a library (Load Session button) to begin recognition (Note: If a user has used training or evaluation mode previous to this session, that user’s device will be automatically loaded into the first available slot and the previously created library will be used). Once the libraries are loaded, users can perform gestures. As they are recognized, the name of the gesture (or "Not recognized" if FROG could not determine the gesture with enough certainty) will be displayed under the user's name (Figure 4-13).

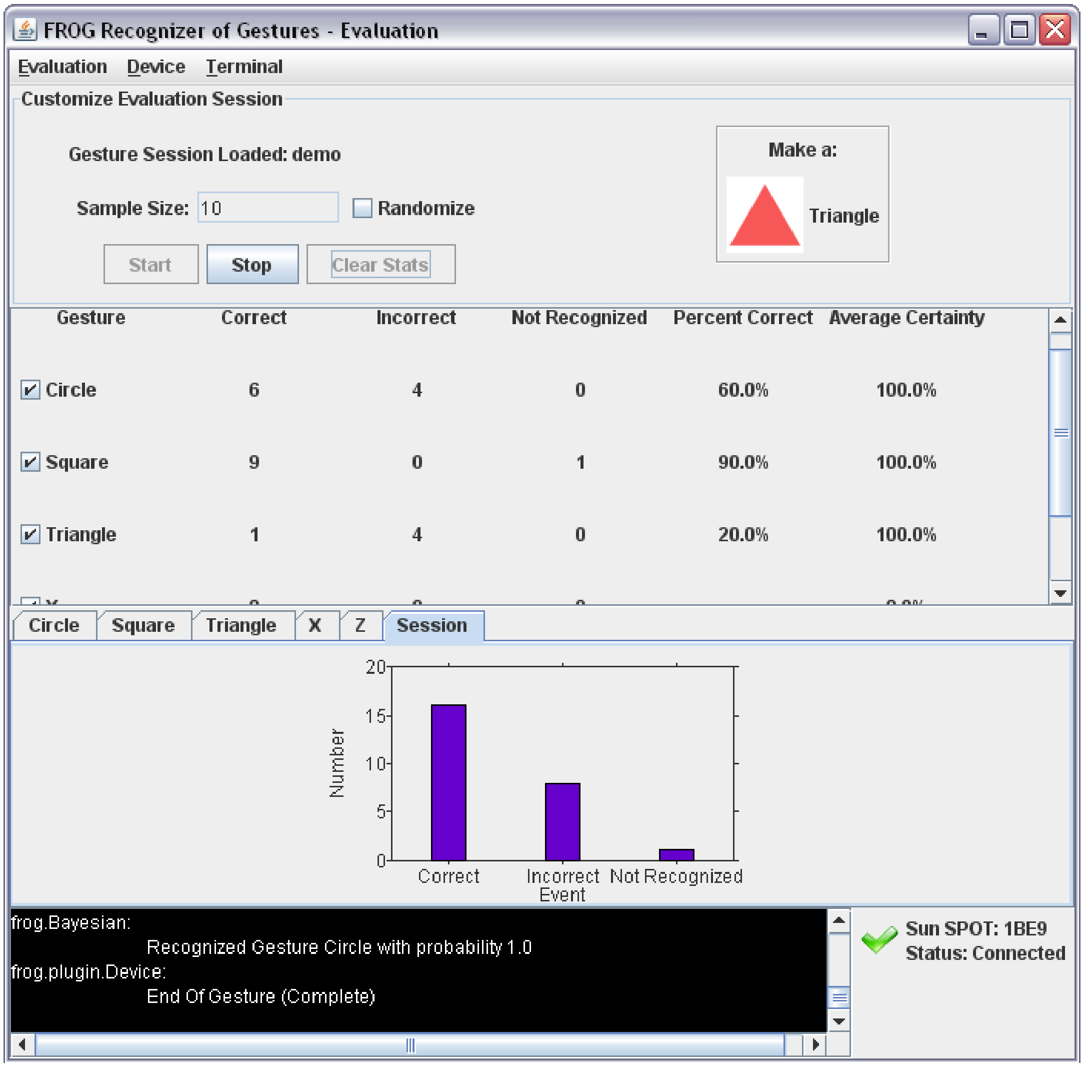


```
frog.Bayesian:
              Recognized Gesture Triangle with probability 0.9872104
frog.plugin.Device:
              End Of Gesture (Complete)
```

Figure 4-13: Recognition Panel with connected users

4.3 Evaluation Mode

Evaluation mode is similar to recognition mode in that it recognizes gestures, but it provides more detailed feedback to a single user only. The system prompts the user to perform a gesture and tracks statistics based on what the user performed. Evaluation mode includes a device settings panel, recognition results panel, and the standard log panel (Figure 4-14).



Customize Evaluation Session

Gesture Session Loaded: demo

Sample Size: 10 ☐ Randomize

Make a: Triangle

Gesture	Correct	Incorrect	Not Recognized	Percent Correct	Average Certainty
☑ Circle	6	4	0	60.0%	100.0%
☑ Square	9	0	1	90.0%	100.0%
☑ Triangle	1	4	0	20.0%	100.0%
☑ X	0	0	0		0.0%

```
frog.Bayesian:
                Recognized Gesture Circle with probability 1.0
frog.plugin.Device:
                End Of Gesture (Complete)
```

Sun SPOT: 1BE9
Status: Connected

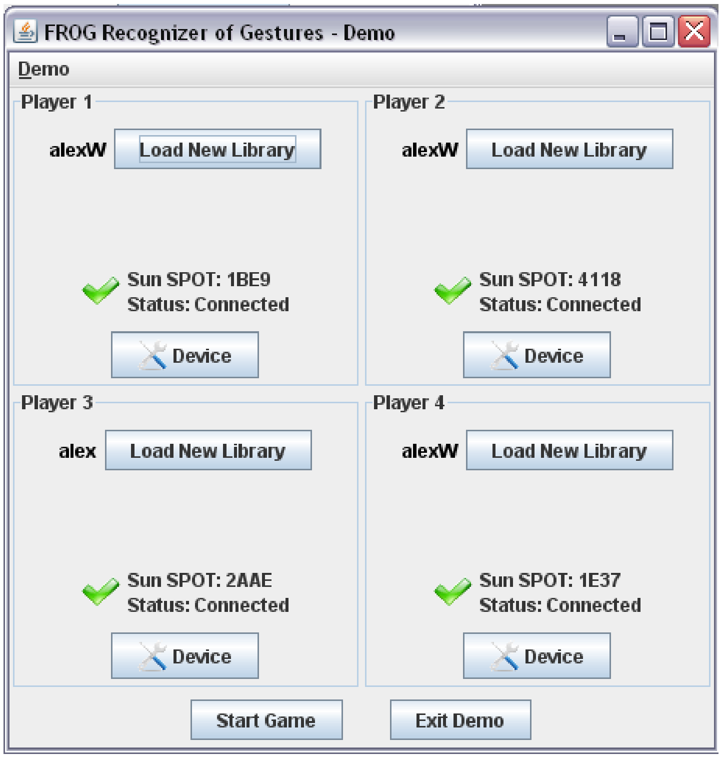
Figure 4-14: Evaluation panel

Using a gesture library, the user will select which gestures from that library will be evaluated. The system provides the ability to present a user-specified number of test gestures in sequential or random order. When sample size and gestures to be evaluated are chosen, evaluation can be started with the "Start" button. "Clear Stats" will remove recognition results from the recognition results panel, but only after the all samples have been generated. The results panel displays the number of correct, incorrect, or not recognized gestures as well as average certainty of correct gestures. Additionally, graphical display of these values is also available.

Saving a session after evaluation causes the evaluation results to be added to the .frog file. Reloading a session with saved evaluation results will display previous recognition results.

4.4 Demo Mode

In this mode, up to four users can connect to play a competitive game using pre-trained gesture libraries. Each library should be user specific for the best recognition accuracy. Initially, this mode looks and acts similar to recognition mode, with a device connection and library buttons specific to each user (Figure 4-15). When players are connected and gesture libraries loaded, the game can be started using the “Start Game” button. The game does require a specific set of gestures to play. A set of five gestures are used and should be labeled as such: Circle, Square, Triangle, X, and Z. Each of these are required but are not case-sensitive.



Player 1
alexW
Sun SPOT: 1BE9
Status: Connected

Player 2
alexW
Sun SPOT: 4118
Status: Connected

Player 3
alex
Sun SPOT: 2AAE
Status: Connected

Player 4
alexW
Sun SPOT: 1E37
Status: Connected

Figure 4-15: Demo setup panel

The objective of the game is to create gestures displayed on UFOs on screen in order to prevent them from abducting cows wandering on the ground. Each correctly recognized gesture with the correct UFO on screen will cause the UFO to explode and that user to gain points for the kill; incorrect gestures do not penalize the user. Users have until a UFO has abducted a cow and moved off screen to save a cow. Scores are tracked at the left of the screen for each user. Once all cows have been successfully abducted the game will end. To exit press the escape key to return to the demo setup panel.

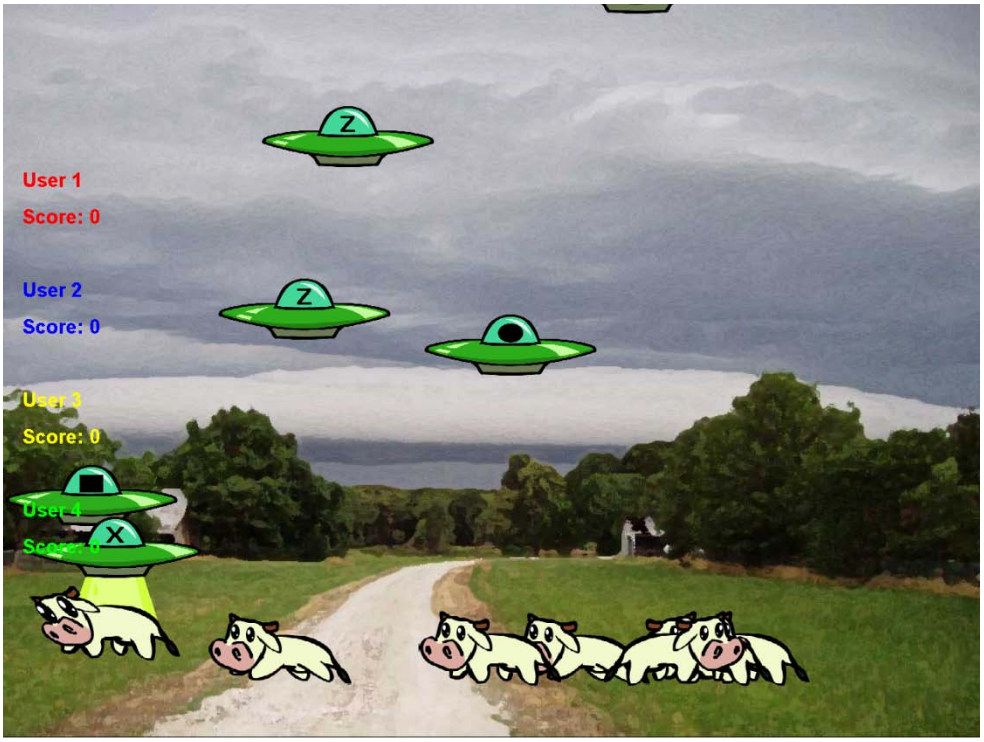


Figure 4-16: Demo game panel

4.5 Non-Mode Specific Utilities

4.5.1 Filters

FROG includes two built-in filters: Directorial Equivalence and Idle State. Some devices support "onboard filtering" where the device itself filters vectors. The Device Connection and Setup window handles both these types of filters.

Directorial Equivalence is a filter that discards acceleration vectors based on their similarity to previous vectors. Directorial Equivalence takes a single parameter for its argument, a threshold level. The default threshold (if nothing is entered) is 0.2. The formula for discarding vectors is as follows:

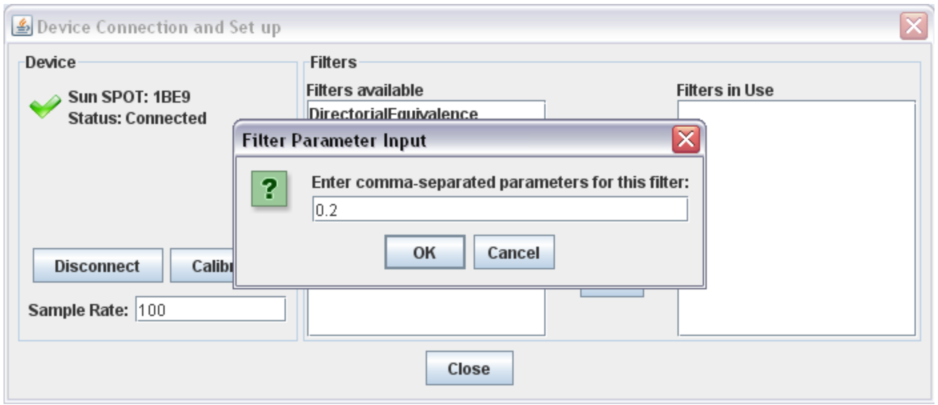
$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + \ (z_1 - z_2)^2} < Threshold$$

If the above inequality is *true*, the second vector is discarded. The reference vector (x_1, y_1, z_1 above) is the last vector not eliminated by this filter.

The *Idle State* filter removes vectors whose magnitude is less than a specified threshold. This is to remove accelerations that come in while the device is mostly immobile and therefore do not contribute much to the overall gesture. Idle State takes a single parameter for its argument, a threshold level. The default threshold (if nothing is entered) is 1.2. The formula for discarding vectors is as follows:

$$\sqrt{x^2 + y^2 + z^2} < Threshold$$

If the above inequality is *true*, the vector is discarded.



Device Connection and Set up

Sun SPOT: 1BE9
Status: Connected

Filters available: DirectorialEquivalence

Filter Parameter Input

Enter comma-separated parameters for this filter: 0.2

Sample Rate: 100

Figure 4-20: Adding a Directorial Equivalence filter with a threshold of 0.2.

If a device supports filtering "onboard," the filters it supports will appear in the filter list when it has been connected. These can be added and removed just like FROG's built-in filters. The difference is that all processing for these filters is done by the device itself before being transmitted to FROG.

4.5.2 Terminal and Log Settings

FROG features an output terminal in Training, Recognition, and Evaluation modes. Located on the toolbar, the terminal settings allow console output and logging of various data including acceleration vectors, filtering, training and recognition results, and system messages (Figure 4-21). If a Log File is chosen through the appropriate dialogue, the system will save all log data to a user specified text file, otherwise it is saved to a system-created file named with the date and time. The terminal can also be displayed in a separate window using the New Terminal Window option (Figure 4-22). This leaves the default terminal window running as well, but allows a window to be moved around the screen for easy viewing.

The log has the following options for output:

All Acceleration Data: Writes acceleration vectors as they are received by FROG.

Gesture Model: Writes HMM data when the HMM is created after clicking “Train.”

Classifier: Writes gesture recognized and probability when the model is classified.

Quantizer: Writes quantizer creation parameters and the actual mean values.

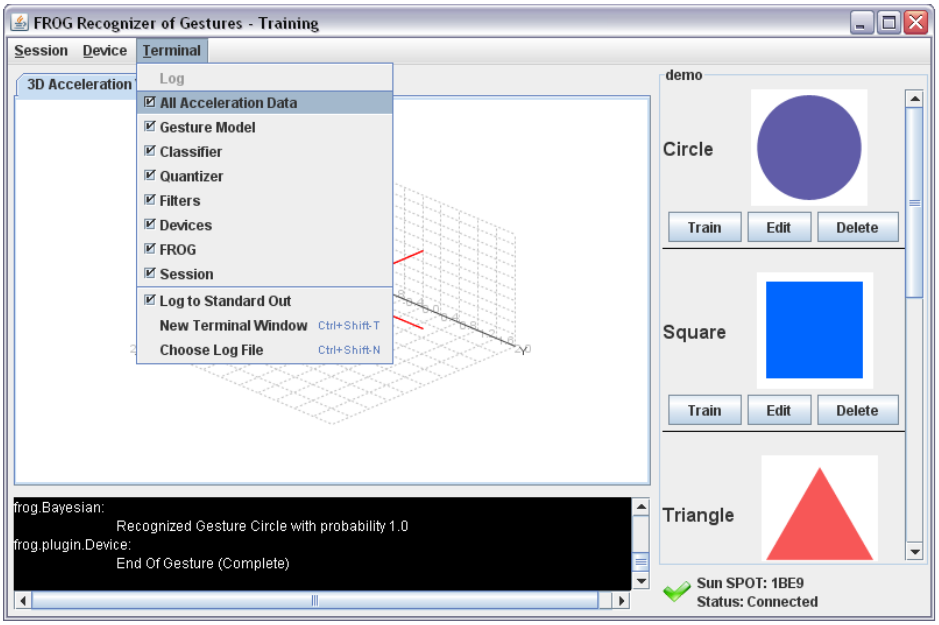
Filters: If a vector is modified by a filter, the filter used and the vector data are written.

Devices: Writes device connects and disconnects, when a gesture is completed, and sampling rate for the device when first connected.

FROG: Writes top level messages such as the path FROG is running from.

Session: Writes session commands such as “Training all gestures” when the session is saved, loaded or re-created.

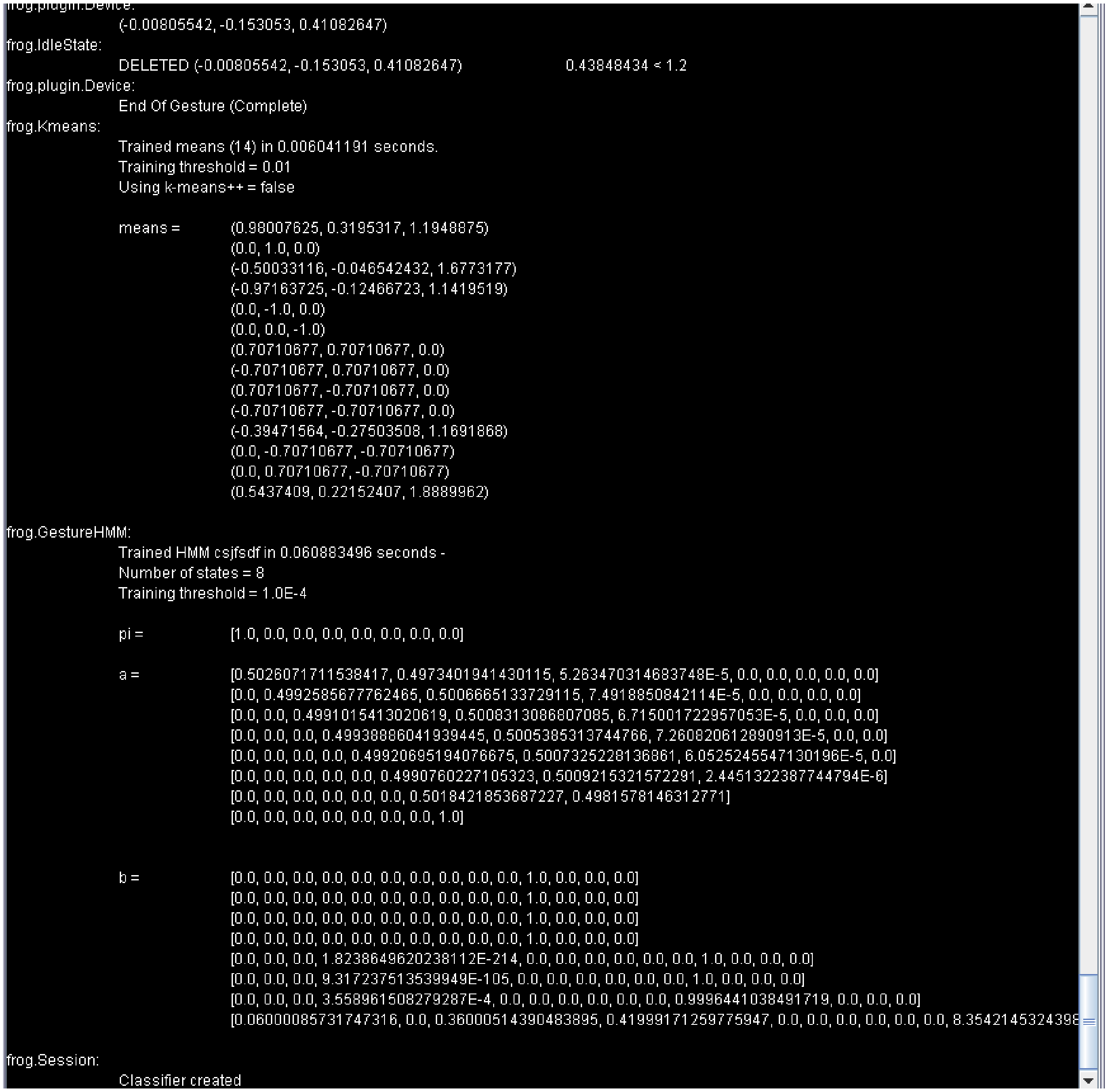
Choose Log File: Allows output to be saved to a specified file, rather than an automatic one with a timestamp as the file name.



```
frog.Bayesian:
            Recognized Gesture Circle with probability 1.0
frog.plugin.Device:
            End Of Gesture (Complete)
```

Sun SPOT: 1BE9
Status: Connected

Figure 4-21: The Training Window with Terminal toolbar open



```
frog.plugin.Device:
                (-0.00805542, -0.153053, 0.41082647)
frog.IdleState:
                DELETED (-0.00805542, -0.153053, 0.41082647)                0.43848434 < 1.2
frog.plugin.Device:
                End Of Gesture (Complete)
frog.Kmeans:
                Trained means (14) in 0.006041191 seconds.
                Training threshold = 0.01
                Using k-means++ = false

                means =         (0.98007625, 0.3195317, 1.1948875)
                                (0.0, 1.0, 0.0)
                                (-0.50033116, -0.046542432, 1.6773177)
                                (-0.97163725, -0.12466723, 1.1419519)
                                (0.0, -1.0, 0.0)
                                (0.0, 0.0, -1.0)
                                (0.70710677, 0.70710677, 0.0)
                                (-0.70710677, 0.70710677, 0.0)
                                (0.70710677, -0.70710677, 0.0)
                                (-0.70710677, -0.70710677, 0.0)
                                (-0.39471564, -0.27503508, 1.1691868)
                                (0.0, -0.70710677, -0.70710677)
                                (0.0, 0.70710677, -0.70710677)
                                (0.5437409, 0.22152407, 1.8889962)

frog.GestureHMM:
                Trained HMM csjfsdf in 0.060883496 seconds -
                Number of states = 8
                Training threshold = 1.0E-4

                pi =            [1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

                a =             [0.5026071711538417, 0.4973401941430115, 5.263470314683748E-5, 0.0, 0.0, 0.0, 0.0, 0.0]
                                [0.0, 0.4992585677762465, 0.5006665133729115, 7.4918850842114E-5, 0.0, 0.0, 0.0, 0.0]
                                [0.0, 0.0, 0.4991015413020619, 0.5008313086807085, 6.715001722957053E-5, 0.0, 0.0, 0.0]
                                [0.0, 0.0, 0.0, 0.49938886041939445, 0.5005385313744766, 7.260820612890913E-5, 0.0, 0.0]
                                [0.0, 0.0, 0.0, 0.0, 0.49920695194076675, 0.5007325228136861, 6.0525245547130196E-5, 0.0]
                                [0.0, 0.0, 0.0, 0.0, 0.0, 0.4990760227105323, 0.5009215321572291, 2.4451322387744794E-6]
                                [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5018421853687227, 0.4981578146312771]
                                [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0]


                b =             [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0]
                                [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0]
                                [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0]
                                [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0]
                                [0.0, 0.0, 0.0, 1.8238649620238112E-214, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0]
                                [0.0, 0.0, 0.0, 9.317237513539949E-105, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0]
                                [0.0, 0.0, 0.0, 3.558961508279287E-4, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.9996441038491719, 0.0, 0.0, 0.0]
                                [0.06000085731747316, 0.0, 0.36000514390483895, 0.41999171259775947, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 8.3542145324398

frog.Session:
                Classifier created
```

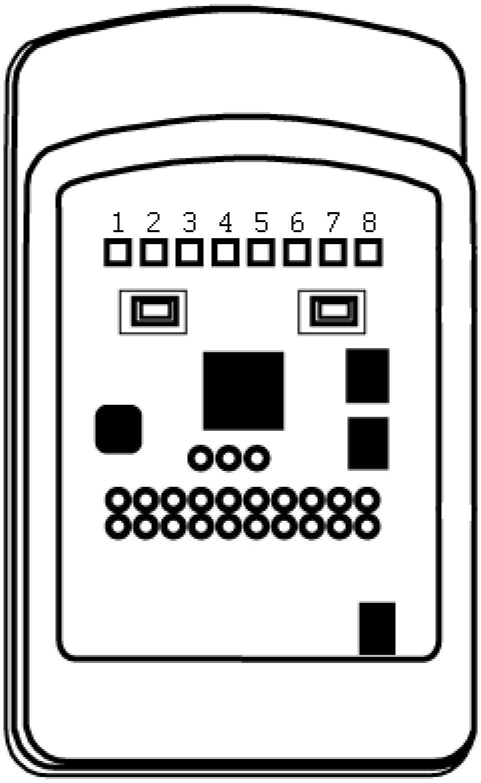
Figure 4-22: Detached Terminal window

5. Troubleshooting

5.1 Errors, Malfunctions, and Emergencies

The most conspicuous error message received will be a “gesture not recognized” message when in evaluation mode. This message simply indicates either the training or evaluating gesture were different than what was intended. Let the system try to recognize the gesture a few more times. If results don’t improve, re-train the gesture from the main training mode. In normal use there should not be any errors during runtime of the code itself, but if that is not the case errors will be outputted to the console and written to the log file if that option is enabled.

The onboard LEDs serve to display error and status messages regarding normal operation. The following table displays the possible LED codes encountered, with the position marked on the diagram below:



Error Code:	**LED position:**
SPOT is searching for a basestation host	1
SPOT is connected successfully	2
Basestation discovery failure	3
Accelerometer reading failure	4
Filter failure	5
Data transmission failure	6
Data receive failure	7
Watcher Thread failure	8

5.2 System Output

FROG allows for output of training and recognition results, through console and system messages. Refer to section 4.5.2 for terminal and log output.