Chapter 1

Installation

1.1 GitHub

1.1.1 Setup

`sudo apt-get install git`	Install git
`git init`	Setup as git capable repository
`git config --global user.name "name"`	Configure machine to have an author
`git config --global user.email "email"`	Configure machine to have an author`git remote -v`

1.2 Emacs Shortcuts

1.2.1 Motion

line:	`C-a`	`C-e`	(beginning/end)
character:	`C-f`	`C-b`	(forward/backward)
word:	`ESC-f`	`ESC-b`	(forward/backward)

Chapter 2

Current System

2.1 Software

2.1.1 Techniques

These are some of the general techniques that are being used in the structure/implementation of the `WormTracker`

1. Logging
2. Passing execution parameters through the command line
3. Qt (User Interface)
4. GitHub for versioning and code distribution. Current branch being used is iss20

2.1.2 Classes/Files

- `install-opencv-2_4_8.sh`: shell script to have identical OpenCV installations across machines
- `tracker.py`: the master class executed for functionality of tracker. Defines `Tracker` class. Initializes Qt Windows, passes arguments to the `finder`.
- `easyEBB.py`: management class for the EiBotBoard (motor control board)
- `eggbot_scanlinux.py`: class used to search USB devices connected for the EiBotBoard. Modified from Inkscape Python extension for EiBotBoard.
- `finder.py`: class containing all of the methods used to locate the worm in an image and make decisions about whether or not it needs to move.
- `managers.py`: management class for the Qt display and recording
- `imgProc.py`: depreciated

2.1.3 Command line arguments

NOTE: some of the execution methods and logging levels have certain UI behaviors attached to them.
Sample execution of the tracker program is as follows: `sudo python tracker.py -m lazyc -l warning -s 0`

NOTE: `sudo` needed to be able to interact with the stepper motors

Source (-s)

These are numbered as follows:

0. Camera source 0 (webcam on laptop or USB camera on Intel NUC)
1. Camera source 1 (USB camera on laptop)
2. `led_move1.avi`
3. `screencast.avi`
4. `shortNoBox.avi`
5. `longNoBox.avi`
6. `H299.avi`

A sources 2-7 are useful to test execution of tracker without using live worms. An improvement in the tracker from a previous version can be obvious when running it on

Execution Methods (-m)

These are 4-5 letter specifications (command line `-m` flag) specifying certain elements of operation of the

1. `surf`
2. `sift`
3. `lazy`
4. `lazyc`
5. `lazyd`
6. `full`
7. `test`

The default method is specified line 208 in `tracker.py` as is currently `lazy`

Logging Levels (-l)

These are standard and will impact what information is seen in the console. There are also UI considerations and dependencies on the execution method.

- `critical` associated with value 50.
- `error`
- `warning`
- `info`
- `debug`

Recommended combinations of -m -l and -s depending on goals

Chapter 3

Improvements

3.1 Hardware

3.2 Software

Overall important things to change:

1. Make the system multithreaded. The findworm

3.2.1 findworm

3.2.2 Interactions with hardware

It is important to remember that any time the stepper motors are engaged there is about a 1s recovery time to return to the

1. Camera: due to OpenCV issues with the camera delivering video in `GRAY-8` part of the OpenCV C class

Specification of the camera solution

The file to change is located at the following path (on LT-Valerie)
`home\install\opencv-2.4.8\modules\highgui\bin\cap_libv4l.cpp`

The lines to change and how to change them are outlined in the table below:

724	`capture->form.fmt.pix.pixelformat = V4L2_PIX_FMT_BGR24;`
724	`capture->form.fmt.pix.pixelformat = V4L2_PIX_FMT_GREY;`
734	`if (V4L2_PIX_FMT_BGR24 != capture->form.fmt.pix.pixelformat) {`
734	`if (V4L2_PIX_FMT_GREY != capture->form.fmt.pix.pixelformat) {`
845	`IPL_DEPTH_8U, 3, IPL_ORIGIN_TL, 4 );`
845	`IPL_DEPTH_8U, 1, IPL_ORIGIN_TL, 4 );`
931	`capture->imageProperties.palette = VIDEO_PALETTE_RGB24;`
931	`capture->imageProperties.palette = VIDEO_PALETTE_GREY;`
943	`if (capture->imageProperties.palette != VIDEO_PALETTE_RGB24) {`
943	`if (capture->imageProperties.palette != VIDEO_PALETTE_GREY) {`
982	`IPL_DEPTH_8U, 3, IPL_ORIGIN_TL, 4 );`
982	`IPL_DEPTH_8U, 1, IPL_ORIGIN_TL, 4 );`
1258	`IPL_DEPTH_8U, 3, IPL_ORIGIN_TL, 4 );`
1258	`IPL_DEPTH_8U, 1, IPL_ORIGIN_TL, 4 );`
1271	`IPL_DEPTH_8U, 3, IPL_ORIGIN_TL, 4 );`
1271	`IPL_DEPTH_8U, 1, IPL_ORIGIN_TL, 4 );`

3.2.3 Ideal execution