Lab 1 Introduction and Helpful Tips

BE107//April 2, 2019

What is Linux?

Linux is an operating system like Windows 7, Windows 8, Mac OS X
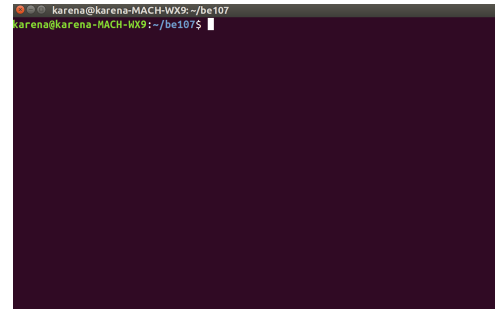- The OS manages all software and hardware resources in your laptop.

Why Linux?

Linux is:
- Reliable and free
- Open-source
- Many robotics software only run in Linux (i.e. ROS)

The Terminal: a command process that allows you to control the computer via commands typed into a text interface

Helpful Shell Commands

<u>sudo:</u> execute following command as super user (needed for installation commands, changing computer root files, etc.)
<u>sudo apt-get install [package]:</u> installs Linux packages to your computer

<u>cd</u>: change directory
<u>ls:</u> list all files and folders in current directory
<u>pwd</u>: shows the full filename of the current working directory
<u>mkdir:</u> make a new directory
<u>cp:</u> copies files from source path to destination path
<u>mv:</u> used to move files or directories
<u>rm:</u> used to remove files or directories

<u>touch:</u> used to create or update a file
<u>ssh:</u> secure shell, allows you to access another computer over the network (i.e. ssh pi@192.168.0.25)

** shortcut: use tab to finish filling in line when typing a command

https://www.geeksforgeeks.org/basic-shell-commands-in-linux/

What is Git?

Git is version control software

- It records and stores the history of every change you have made
- Stores your code in databases called repositories
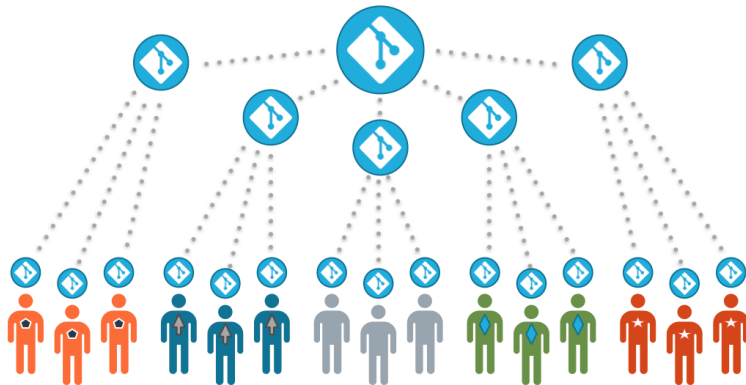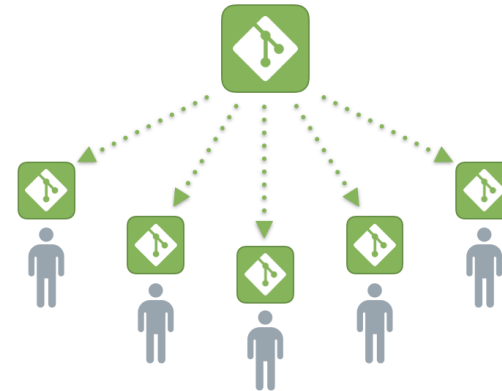
Why use Git?

- Allows you to revert to old versions of code.
- Allows you to collaborate with others on the same code.
- Allows for easy sharing of code on different computers.

Helpful Git Commands

git clone [insert link to git repo]:

git add . : adds changes to stage/index in your working directory
git commit –m "insert message here": commits your changes and sets it to new commit object for your remote
git push: push changes to your remote

git pull: pull changes from remote

git reset --hard: make your current repository point to older committed version (if you run this command before git pull, you delete all the local changes you made to the code)
git init : to initialize a git repository for a new or existing project
git status: used to check status of files you have changed
git branch: lists out all the branches in your repository
git checkout: used to switch to different branches

https://dev.to/dhruv/essential-git-commands-every-developer-should-know-2fl

be107 Git Repository

https://github.com/karenacai/be107.git

be107/Lab 1
- cam_capture.py
- cam_calibrate.py
- cam_detectAruco.py
- /calibration_images

- robot_findRed

- data_flyTracks.h5
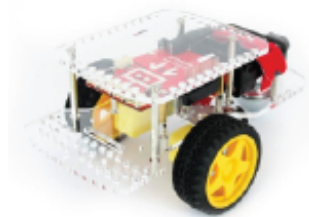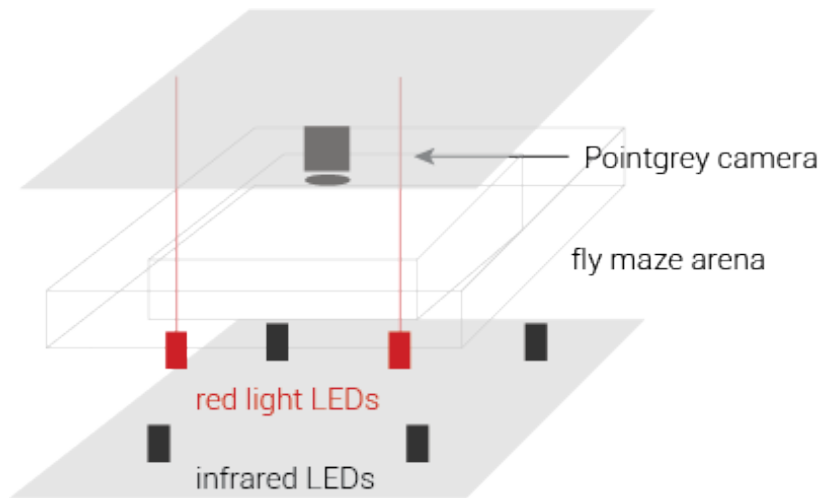- data_robotTracks.h5

- analyzeData.py

Step 3: Clone group repository onto robot

Step 2: copy files over

Step 1: make empty git repository BE107Group1/Lab1

Part 1: Fruit-fly Experiment Overview

Experimental set-up

Pointgrey camera

fly maze arena

red light LEDs

infrared LEDs

Software architecture in ROS

software

fly-tracking node

— publisher
— subscriber

topics

images

fly-pos

camera node

LED node

record data node

camera

LEDs

fly arena

hardware
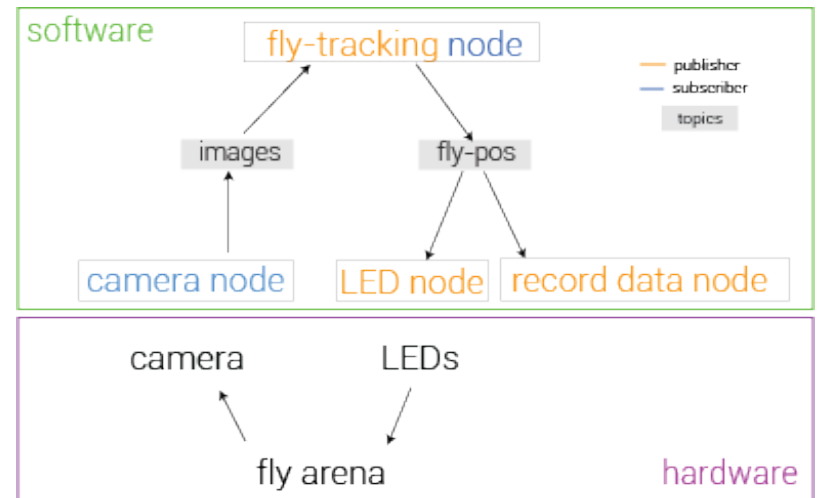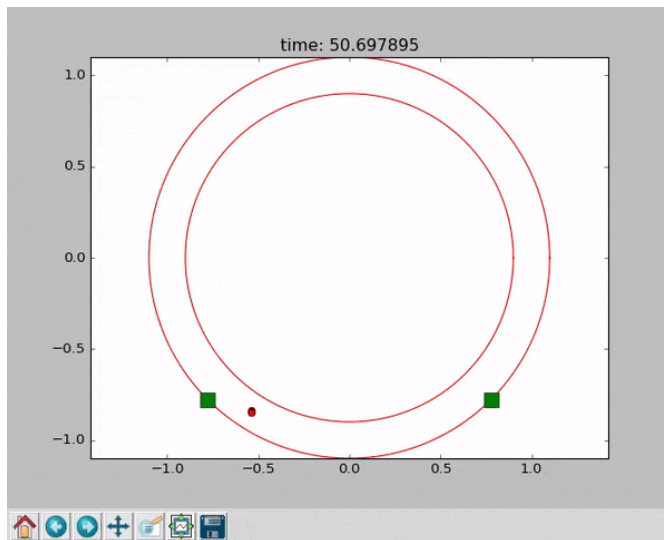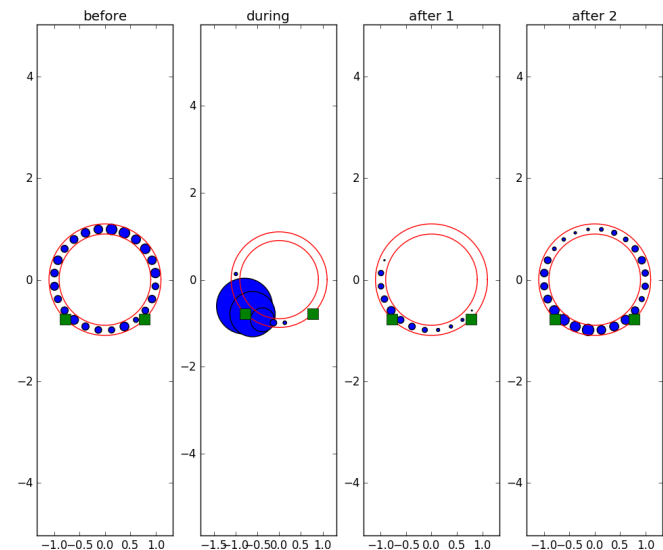
Experiment Animation and Sample Results
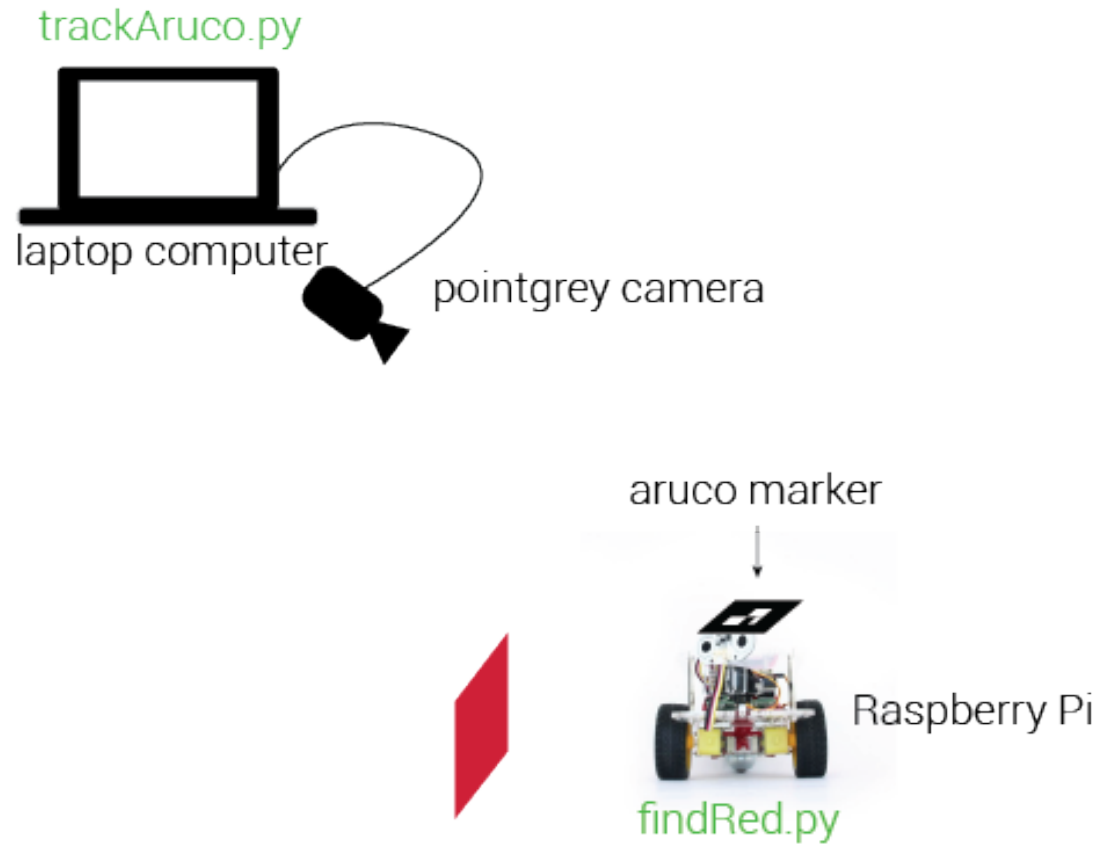
Experiment Animation

Red dot: fly
Green spots: LED activation region

Results

Blue dots denote frequency of visiting that region of the maze arena.

Part 2: Fly-Inspired Robot Experiment Overview

Tips for Data Analysis

```python
analyzeData.py                    ✕

1    import h5py
2    import numpy as np
3    import matplotlib.pyplot as plt
4    from mpl_toolkits.mplot3d import Axes3D
5
6    # importing data from data files
7    data_robot = h5py.File("data_robotTracks.h5", "r")
8    robot_time = data_robot['time'] # is time the title?
9    robot_translation = data_robot['t_vecs']
10   robot_rotation = data_robot['r_vecs']
11
12   data_fly = h5py.File("data_flyTracks.h5","r")
13   fly_time = data_fly['time']
14   fly_x = data_fly['pos_x']
15   fly_y = data_fly['pos_y']
16   fly_angle = data_fly['angle']
17
18   # extracting rotation and translation components
19   robot_x = robot_translation[:,0]
20   robot_y = robot_translation[:,1]
21   robot_z = robot_translation[:,2]
22
23   robot_rx = robot_rotation[:,0]
24   robot_ry = robot_rotation[:,1]
25   robot_rz = robot_rotation[:,2]
26
27   # example of plotting data
28   fig = plt.figure(1)
29   ax = fig.add_subplot(111, projection = '3d')
30   ax.scatter(robot_x, robot_y, robot_z)
31
32   fig2 = plt.figure(2)
33   # preliminary fly data analysis
34   ax2 = fig2.add_subplot(111)
35   ax2.scatter(fly_x, fly_y)
36
37   plt.show()
```

Filenames of data files

More plotting commands can be found at: https://matplotlib.org/