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

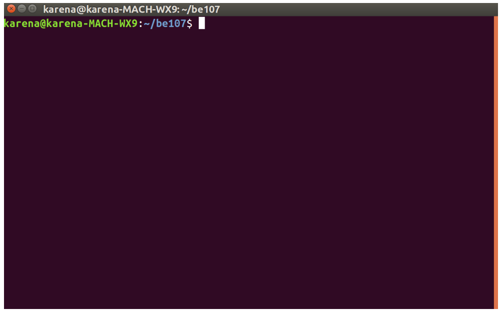
- The OS manages all software and hardware resources in your laptop.

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



https://www.linux.com/what-is-linux

Helpful Shell Commands

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

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

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

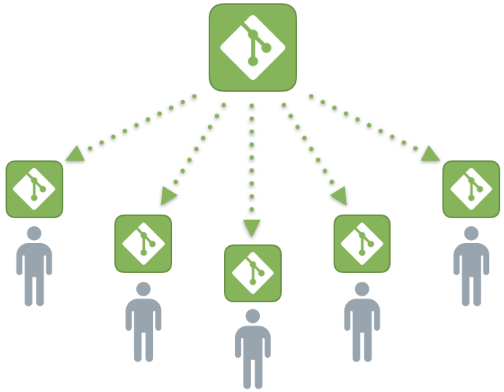
** shortcut: use tab to finish filling in line when typing a command

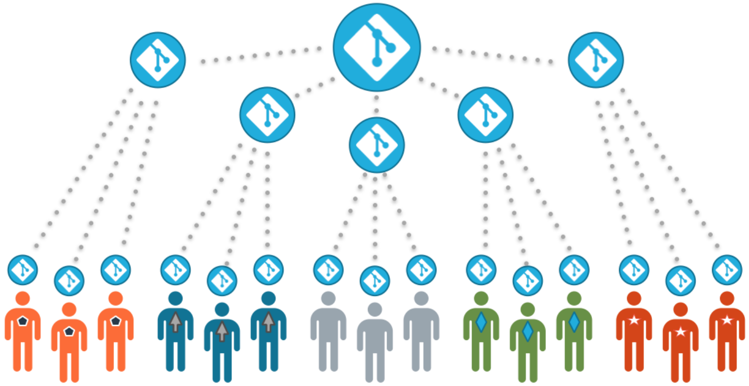
https://www.geeksforgeeks.org/basic-shell-commands-in-linux/

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

<u>git clone [insert link to git repo]:</u>

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

<u>git pull</u>: pull changes from remote

<u>git reset --hard:</u> 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)
<u>git init :</u> to initialize a git repository for a new or existing project
<u>git status:</u> used to check status of files you have changed
<u>git branch:</u> lists out all the branches in your repository
<u>git checkout:</u> 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

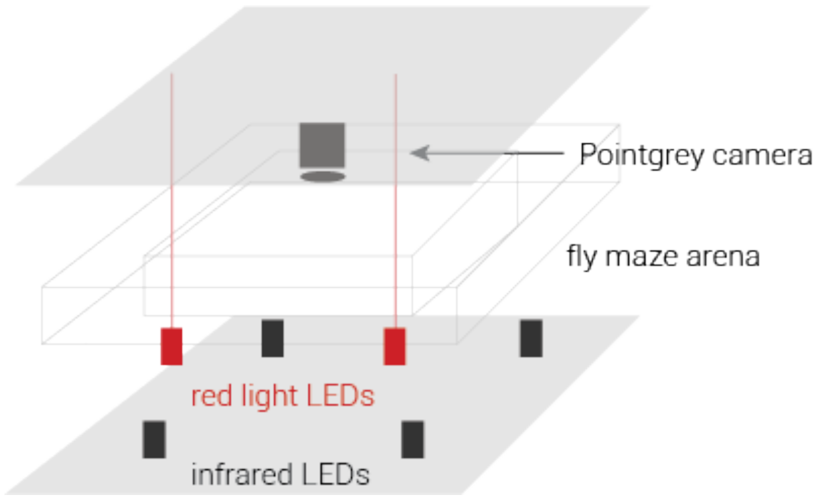
Step 3: Clone group repository onto robot

Step 2: copy files over

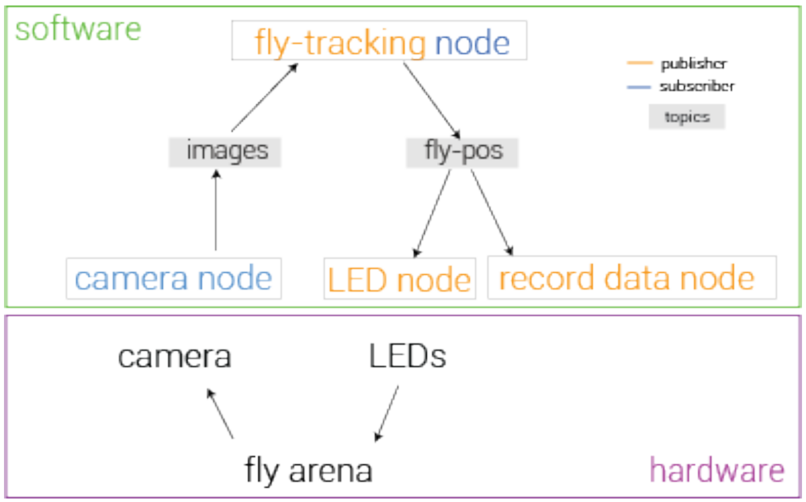
Step 1: make empty git repository BE107Group1/Lab1

Part 1: Fruit-fly Experiment Overview

Experimental set-up

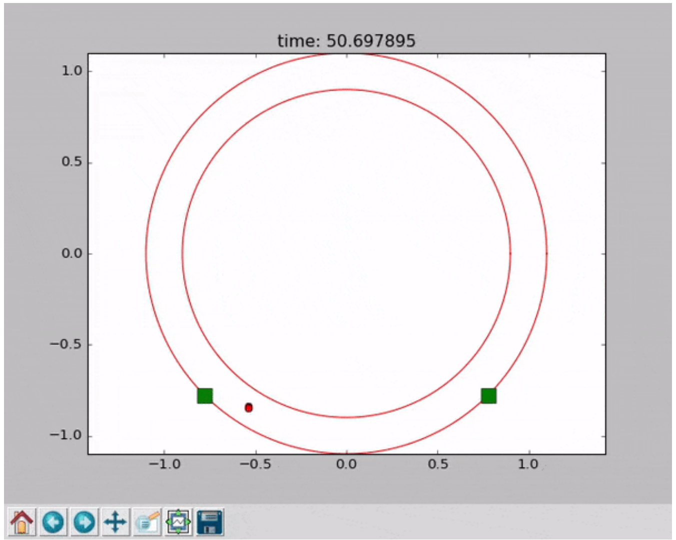


Software architecture in ROS



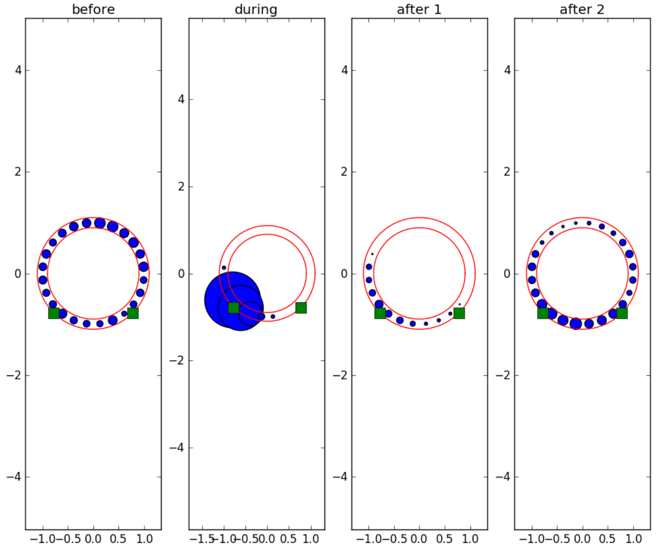
Experiment Animation and Sample Results

Experiment Animation



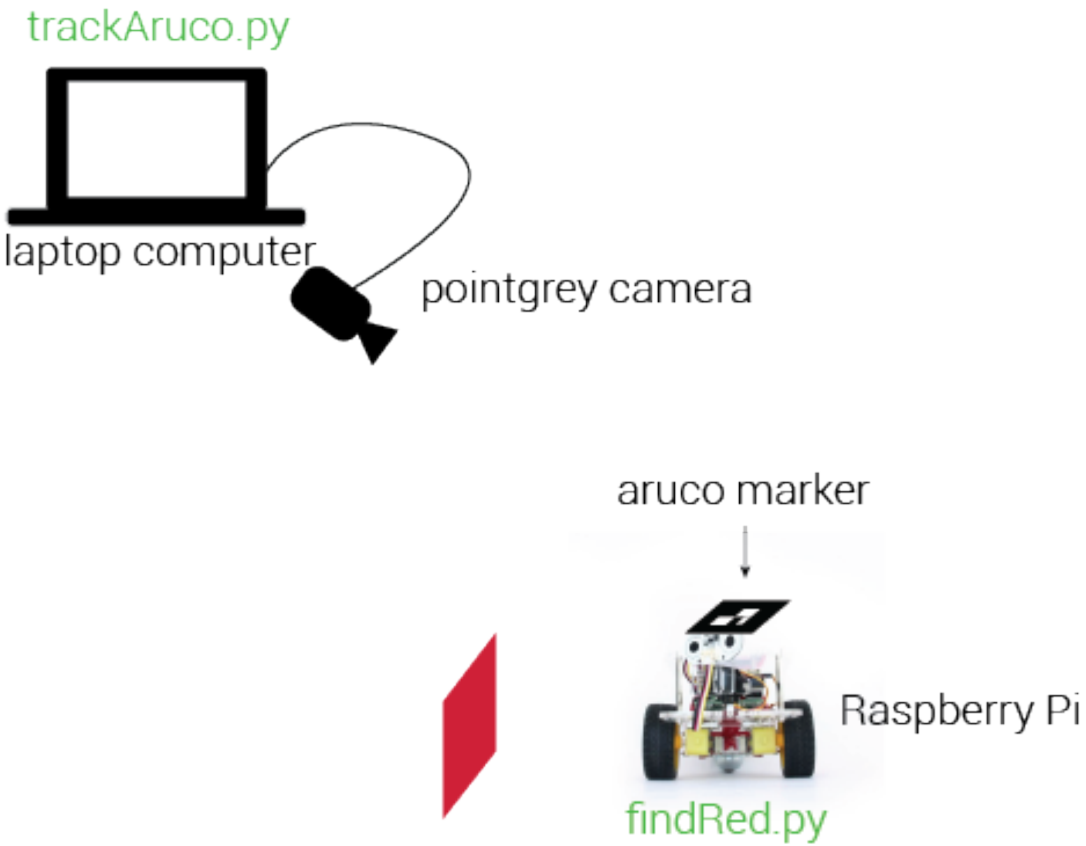
Red dot: fly
Green spots: LED activation region

Results



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

Part 2: Fly-Inspired Robot Experiment Overview



Tips for Data Analysis

analyzeData.py

```python
import h5py
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# importing data from data files
data_robot = h5py.File("data_robotTracks.h5", "r")
robot_time = data_robot['time'] # is time the title?
robot_translation = data_robot['t_vecs']
robot_rotation = data_robot['r_vecs']

data_fly = h5py.File("data_flyTracks.h5","r")
fly_time = data_fly['time']
fly_x = data_fly['pos_x']
fly_y = data_fly['pos_y']
fly_angle = data_fly['angle']

# extracting rotation and translation components
robot_x = robot_translation[:,0]
robot_y = robot_translation[:,1]
robot_z = robot_translation[:,2]

robot_rx = robot_rotation[:,0]
robot_ry = robot_rotation[:,1]
robot_rz = robot_rotation[:,2]

# example of plotting data
fig = plt.figure(1)
ax = fig.add_subplot(111, projection = '3d')
ax.scatter(robot_x, robot_y, robot_z)

fig2 = plt.figure(2)
# preliminary fly data analysis
ax2 = fig2.add_subplot(111)
ax2.scatter(fly_x, fly_y)

plt.show()
```

Filenames of data files

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