

This project uses Python 3. It can be installed by:

```
sudo apt-get install python3.6
```

You may also need to install some modules. If it says 'no module named cv2', then type:

```
pip3 install python-opencv
```

The project contains different files that contain python classes for path planning, filtering, computer vision, and communication. Most files are solely for the purpose of importing, but some also contain runnable demos (python3 file.py).

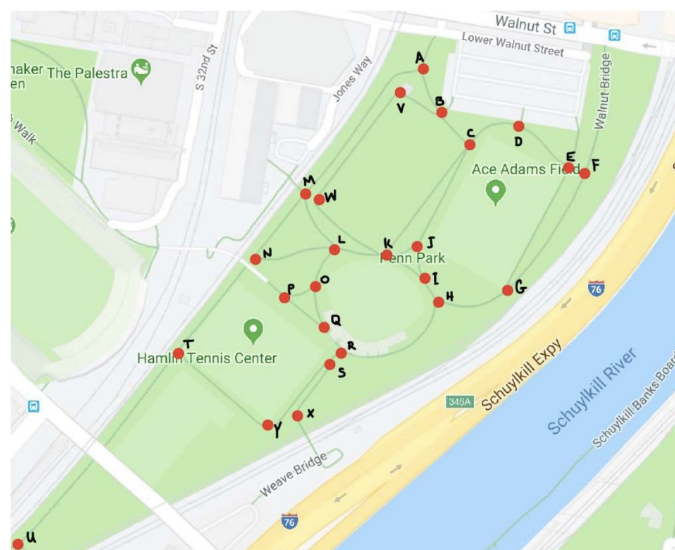
File Descriptions:

Mainloop.py:

- The main function that imports other files and runs several threads simultaneously.
- self.threads contains booleans that determine which threads will be run. It might be easiest to turn on only one at a time to test them in isolation.
- Run the file (python3 Mainloop.py) to run the loop. This will spin up a bunch of threads and return control to the user. The user can adjust parameters, print values, stop/start threads, and view the image processing while the program is running.
- self.start(thread) and self.stop(thread) will start and stop a thread during runtime.
- self.print(var) will continuously print a class variable every self.printfreq seconds.

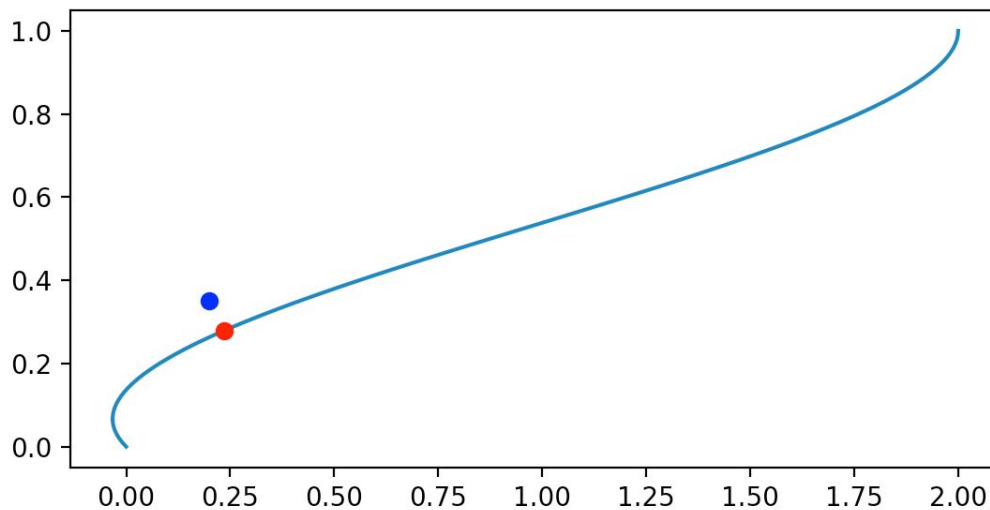
Map.py:

- Predicts the map position by snapping the gps to the nearest road.
- Implements A* search path planning to the destination node
- Running Map.py will run a demo of the A* search. There is also a demo for gps position snapping, which can be selected by changing the last line of the file from testPathPlan() to testPredict().



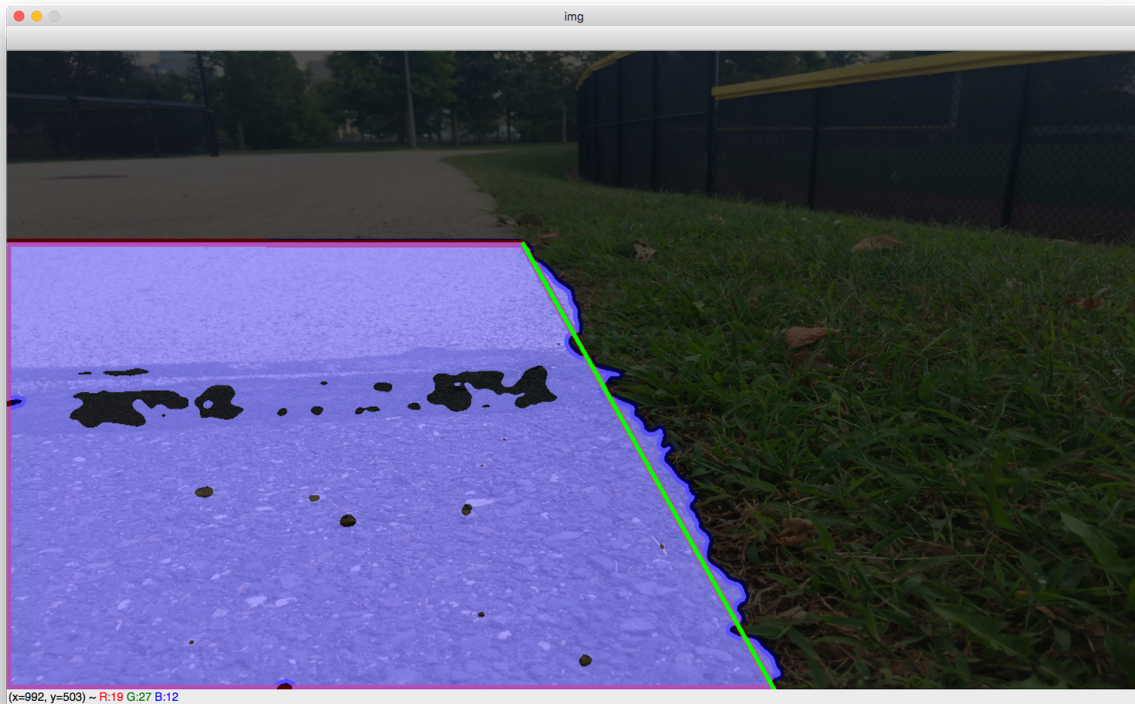
Planning.py

- Uses a hermite spline to generate a path from the current position and orientation to the next waypoint (facing directly down the road, 1m from the edge).
- When updated, the inverse hermite function calculates the most likely position along the spline. The steering angle is calculated by the derivative of the hermite spline at a small step in the future. When the car reaches the end of the spline or strays too far away, a new spline is calculated.
- Running Planning.py will run a demo that creates a spline from a start position and orientation to an end position and orientation, then predicts the closest point on the spline to a given point in space.



ComputerVision.py

- First this function finds the edge of the road by color thresholding the ground, simplifying the bounding box, and choosing the side that is most likely to be the edge of the road. Next, it transforms the position of the edge on the screen to its position in 3D space.
- Running ComputerVision.py will run a demo on snapshots from the Pi Camera in the Files/CarPictures directory.



Filter.py

- Implements the complementary filter and integrator

ManualControl.py

- Gets joystick input

Message.py

- Sends and receives data between the Pi and Arduino over I2C