
Triton Town: A Platform for Education and Research

Jarrett Ratelle & Ivan Ferrier

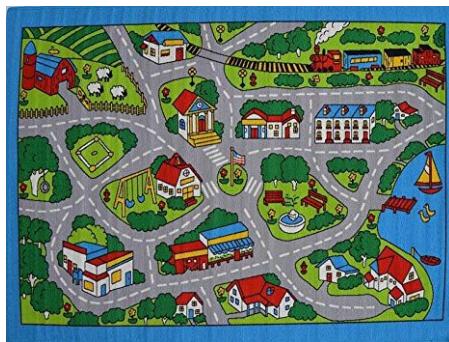
Mentor: Jack Silberman

What is Triton Town?

- Self driving vehicles are on the rise
- People want to get experience and test new ideas
- Few outlets to implement code exist
- Triton Town will attempt to give people a platform to upload and test self driving algorithms

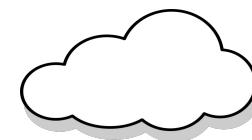
Building a Good Foundation

1) Vehicle



2) Map

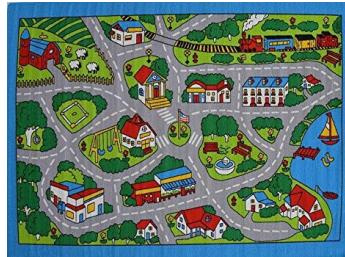
3) Navigation



4) Remote Updates

Building a Good Foundation

1) Vehicle



2) Map

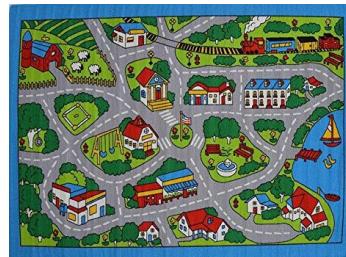
3) Navigation



4) Remote Updates

Building a Good Foundation

1) Vehicle



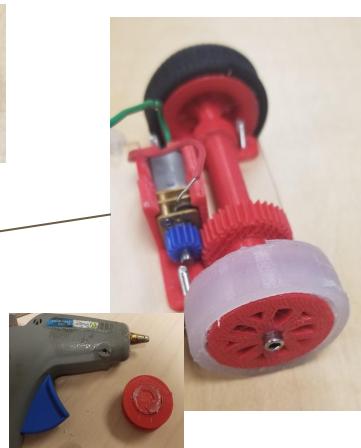
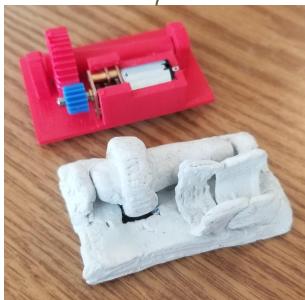
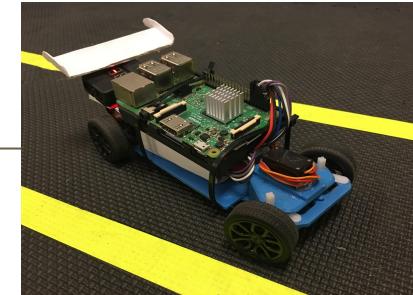
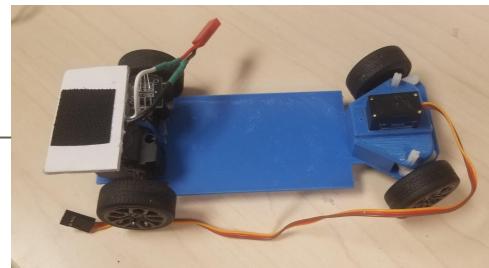
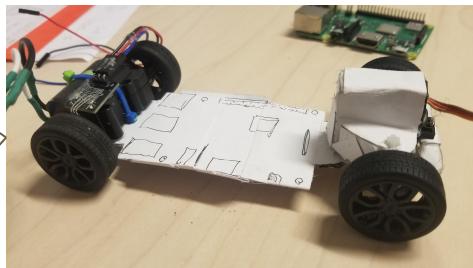
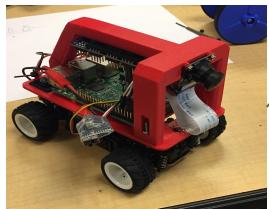
2) Map

3) Navigation

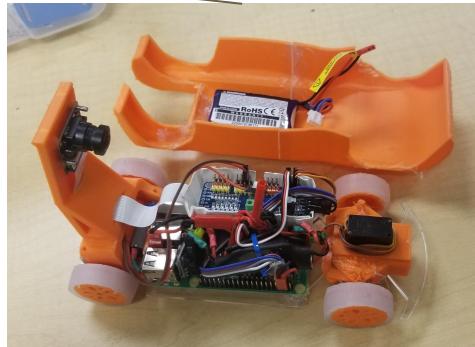


4) Remote Updates

Designing the Car



Designing the Car



Building a Good Foundation

1) Vehicle



2) Track

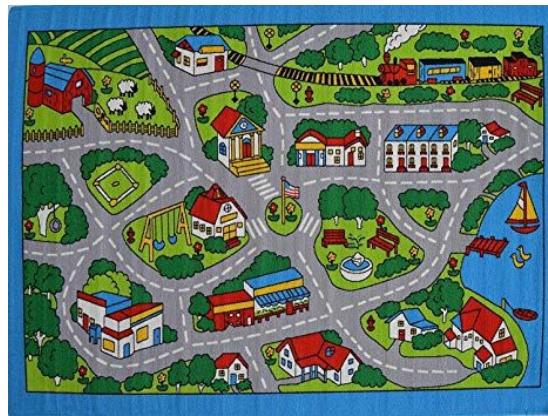
3) Navigation



4) Remote Updates

Building a Good Foundation

1) Vehicle



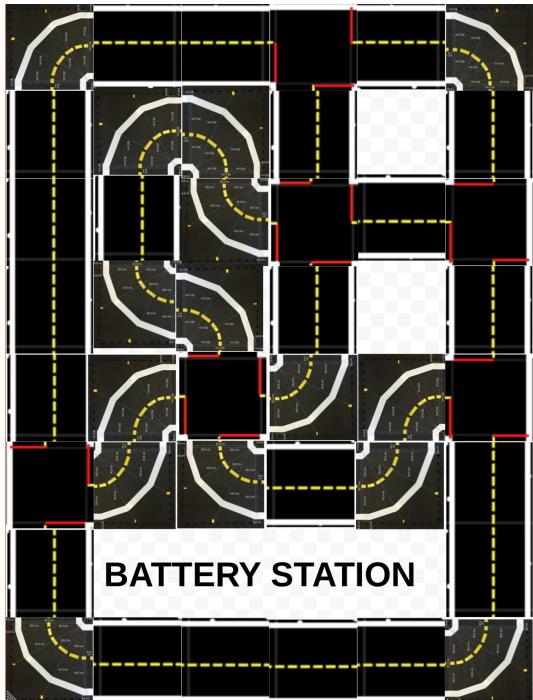
2) Track

3) Navigation



4) Remote Updates

Designing the Track



Building a Good Foundation

1) Vehicle



2) Track

3) Navigation



4) Remote Updates

Building a Good Foundation

1) Vehicle



2) Track

3) Navigation



4) Remote Updates

Building a Good Foundation

Navigation



1) Getting Location

- Defining the region
- Locating the car
- Deciding the destination

2) Adding Constraints

- Minimum Turning Radius
- Keeping the car in its lane
- Taking the shortest path

Building a Good Foundation

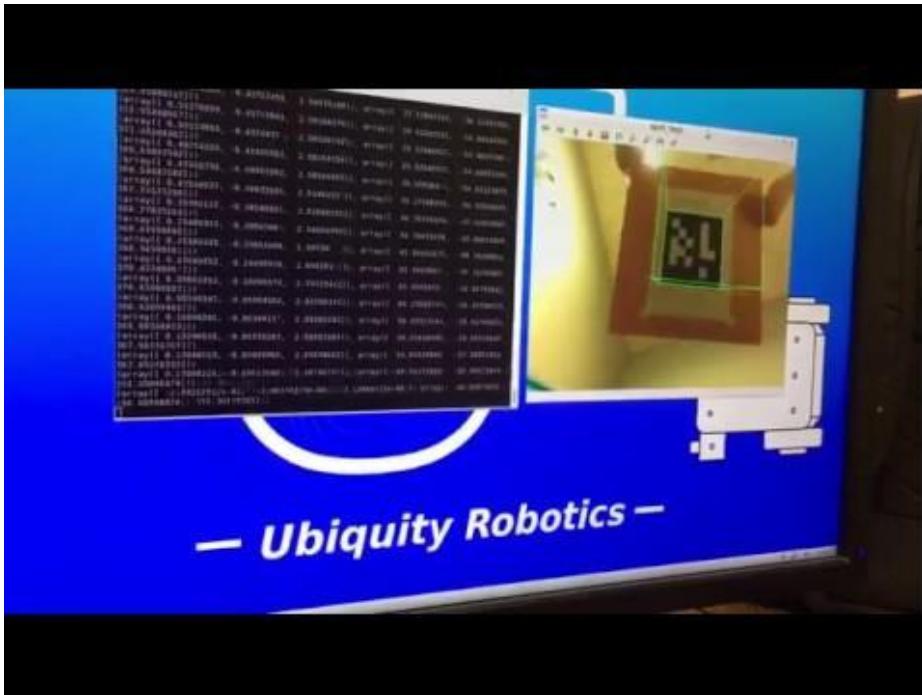
Navigation



- 1) Getting Location
 - Defining the region
 - Locating the car
 - Deciding the destination

- 2) Adding Constraints
 - Minimum Turning Radius
 - Keeping the car in its lane
 - Taking the shortest path

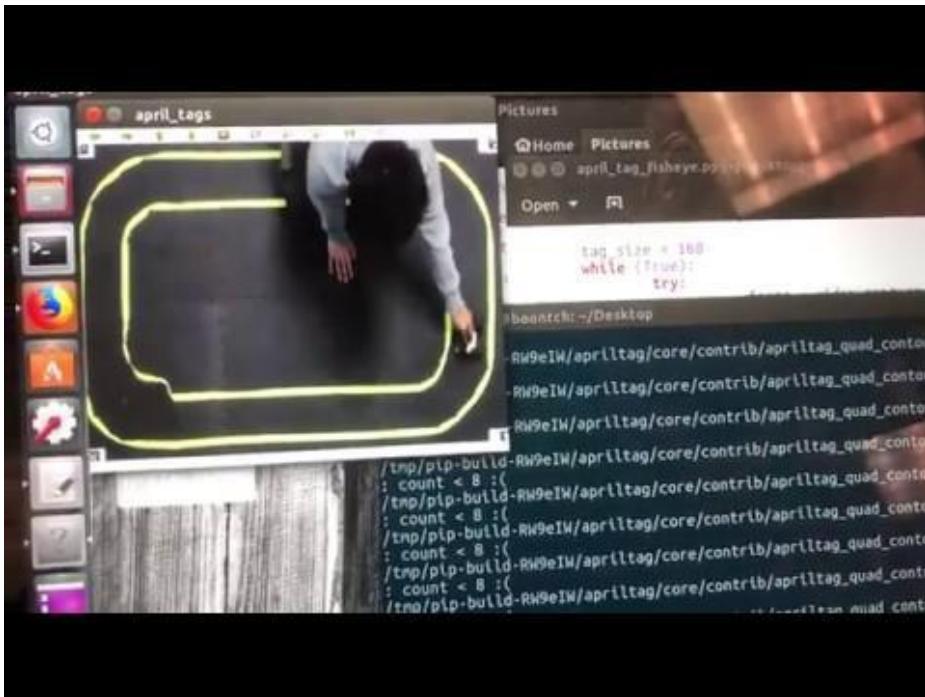
Defining the Region



Detecting Apriltags with OpenCV

- Calibrate camera
- Define the physical tag size
- Get 3D pose of tag relative to the camera lens

Defining the Region



Using Apriltags to Crop Map

- Apriltags are placed at each corner
- Distortions removed with opencv
- Unwanted area cropped out

Building a Good Foundation

Navigation



1) Getting Location

- Defining the region
- Locating the car
- Deciding the destination

2) Adding Constraints

- Minimum Turning Radius
- Keeping the car in it's lane
- Taking the shortest path

Building a Good Foundation

Navigation



1) Getting Location

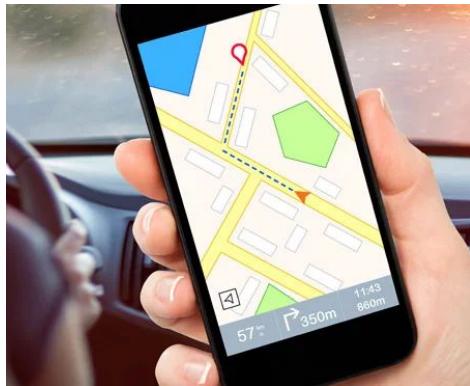
- Defining the region
- Locating the car
- Deciding the destination

2) Adding Constraints

- Minimum Turning Radius
- Keeping the car in its lane
- Taking the shortest path

Building a Good Foundation

Navigation



- 1) Getting Location
 - Defining the region
 - Locating the car
 - Deciding the destination

- 2) Adding Constraints
 - Minimum Turning Radius
 - Keeping the car in its lane
 - Taking the shortest path

Building a Good Foundation

Navigation



1) Getting Location

- Defining the region
- Locating the car
- Deciding the destination

2) Adding Constraints

- Minimum Turning Radius
- Keeping the car in it's lane
- Taking the shortest path

Building a Good Foundation

Navigation



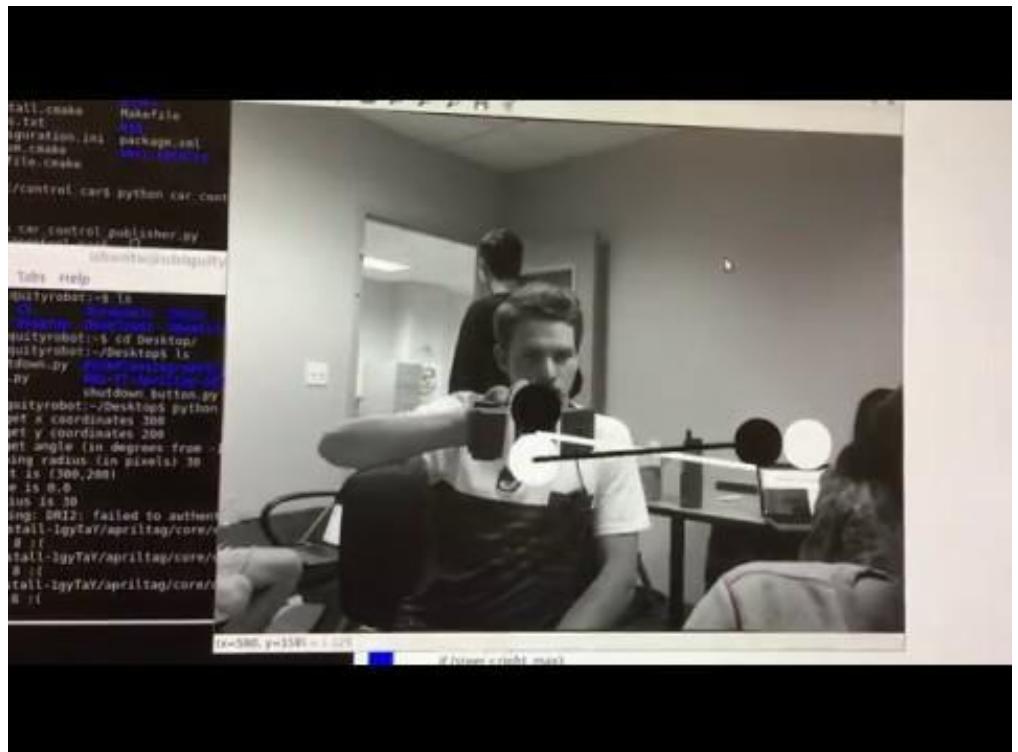
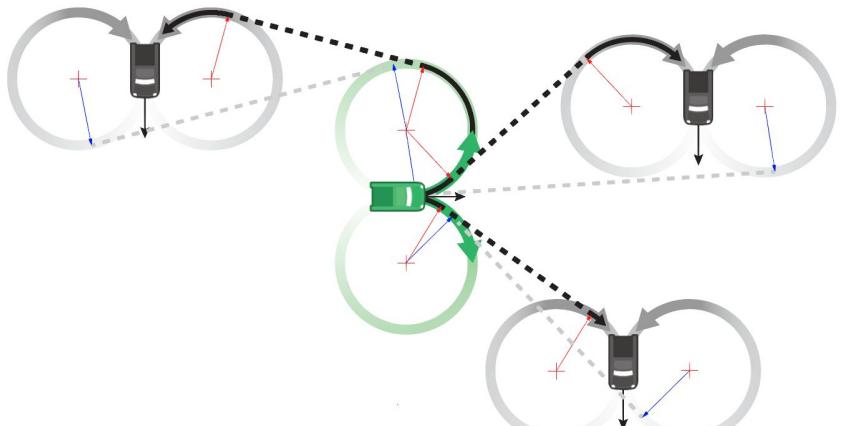
1) Getting Location

- Defining the region
- Locating the car
- Deciding the destination

2) Adding Constraints

- **Minimum Turning Radius**
- Keeping the car in it's lane
- Taking the shortest path

Minimum Turning Radius



Building a Good Foundation

Navigation



1) Getting Location

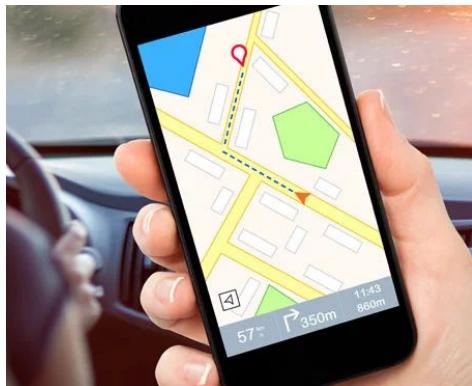
- Defining the region
- Locating the car
- Deciding the destination

2) Adding Constraints

- Minimum Turning Radius
- Keeping the car in it's lane
- Taking the shortest path

Building a Good Foundation

Navigation



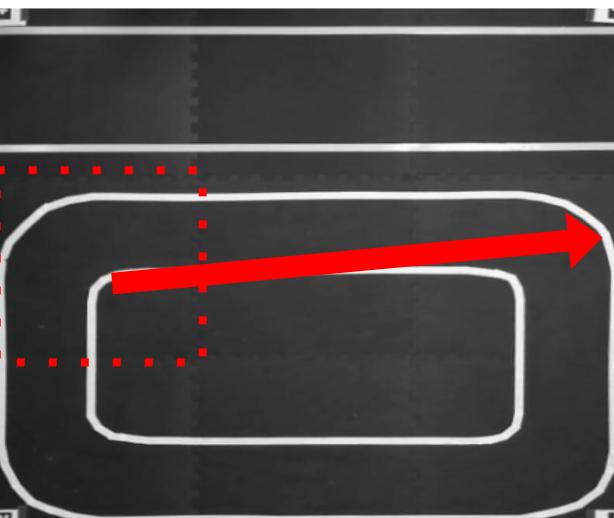
1) Getting Location

- Defining the region
- Locating the car
- Deciding the destination

2) Adding Constraints

- Minimum Turning Radius
- **Keeping the car in its lane**
- Taking the shortest path

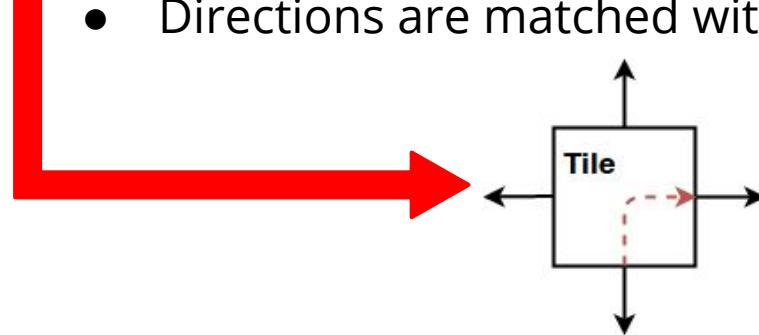
Keeping the Car in it's Lane



```
Tile (0, 0):
Can go up: False
Can go down: False
Can go right: True
Can go left: False
Tile (0, 1):
Can go up: False
Can go down: False
Can go right: True
Can go left: True
Tile (0, 2):
Can go up: False
Can go down: False
Can go right: False
Can go left: True
Tile (1, 0):
Can go up: False
Can go down: True
Can go right: True
Can go left: False
Tile (1, 1):
Can go up: False
Can go down: False
Can go right: True
Can go left: True
Tile (1, 2):
Can go up: False
Can go down: True
Can go right: False
Can go left: True
Tile (2, 0):
Can go up: True
Can go down: False
Can go right: True
Can go left: False
```

Creating Tile Objects

- Image is cropped with Apriltags
- Each region is analysed by contours
- Directions are matched with a tile



Building a Good Foundation

Navigation



1) Getting Location

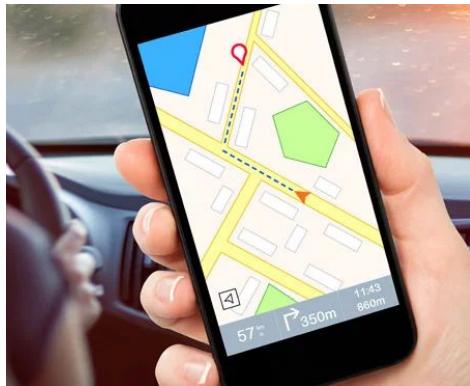
- Defining the region
- Locating the car
- Deciding the destination

2) Adding Constraints

- Minimum Turning Radius
- Keeping the car in it's lane
- Taking the shortest path

Building a Good Foundation

Navigation



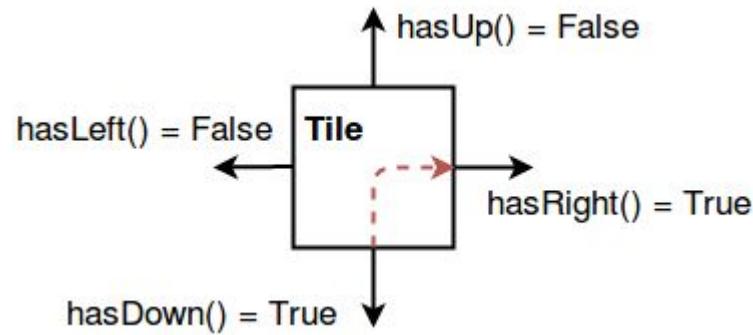
1) Getting Location

- Defining the region
- Locating the car
- Deciding the destination

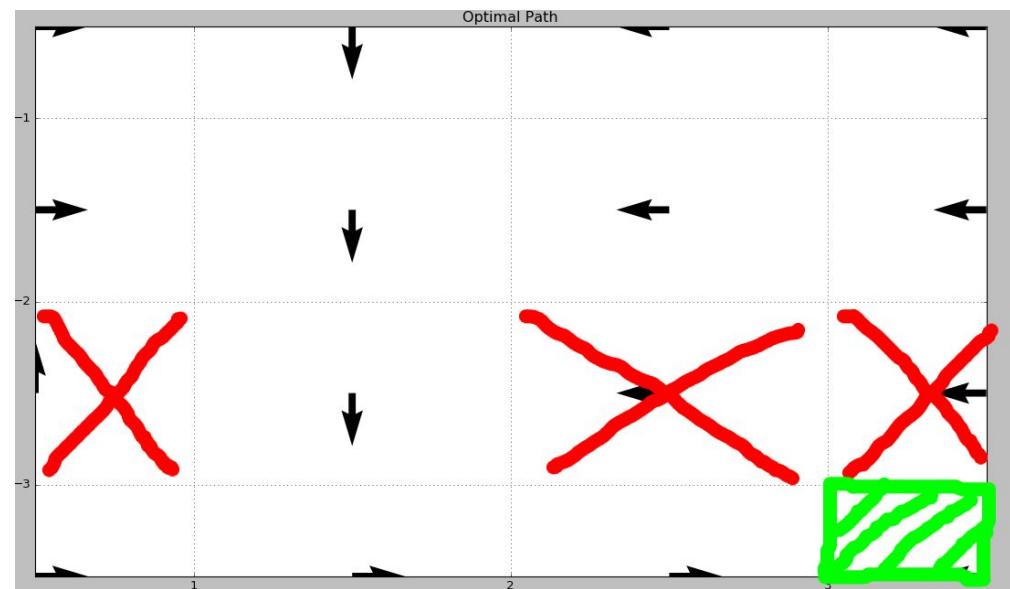
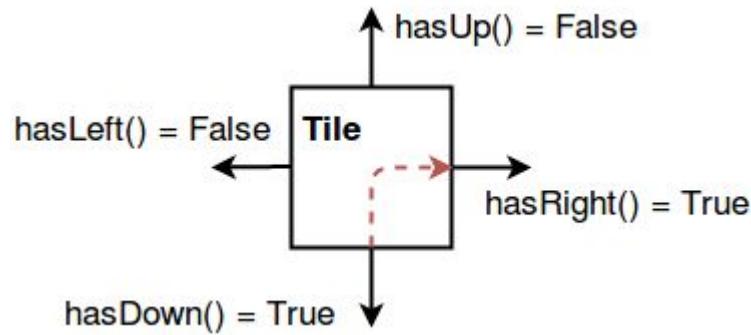
2) Adding Constraints

- Minimum Turning Radius
- Keeping the car in its lane
- **Taking the shortest path**

Dijkstra's Algorithm



Dijkstra's Algorithm



Building a Good Foundation

1) Vehicle



2) Map

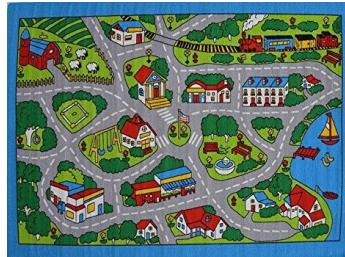
3) Navigation



4) Remote Updates

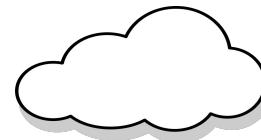
Building a Good Foundation

1) Vehicle



2) Map

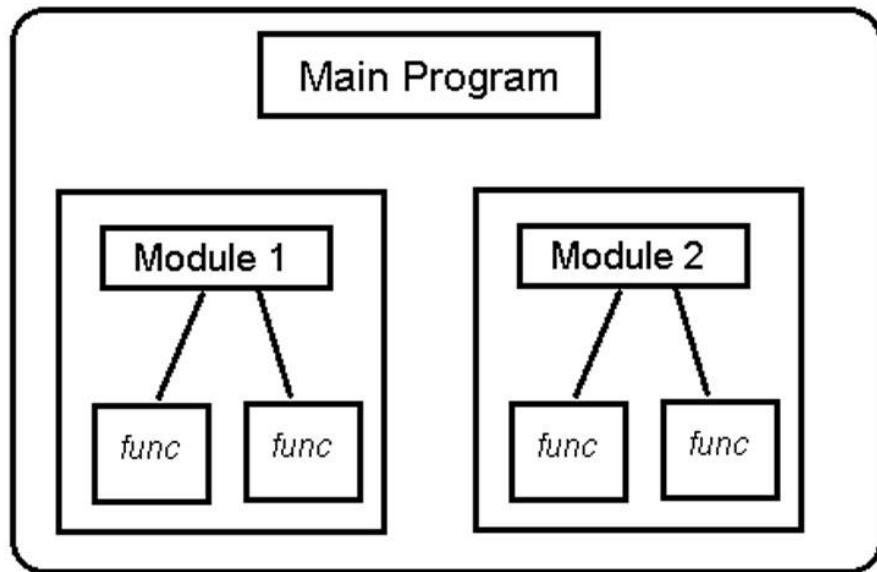
3) Navigation



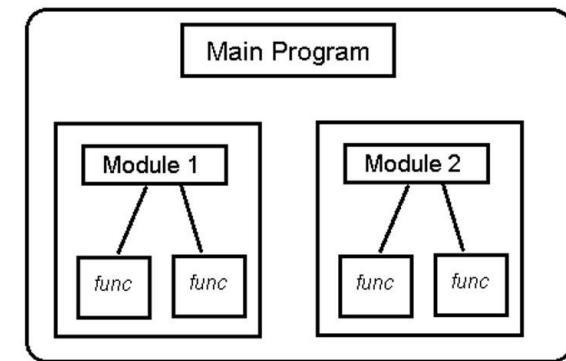
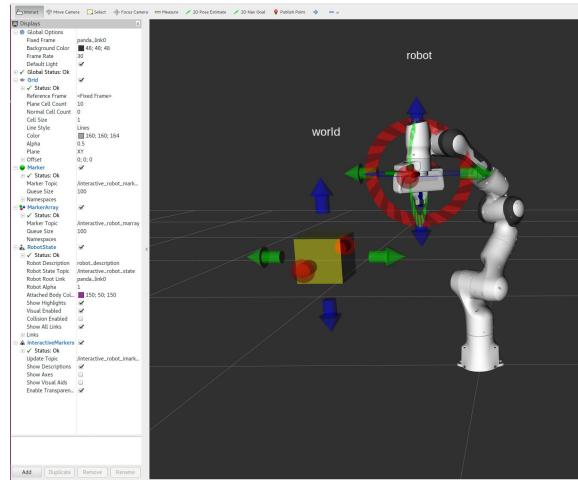
4) Remote Updates

**Looking to
the Future**

Remote updates

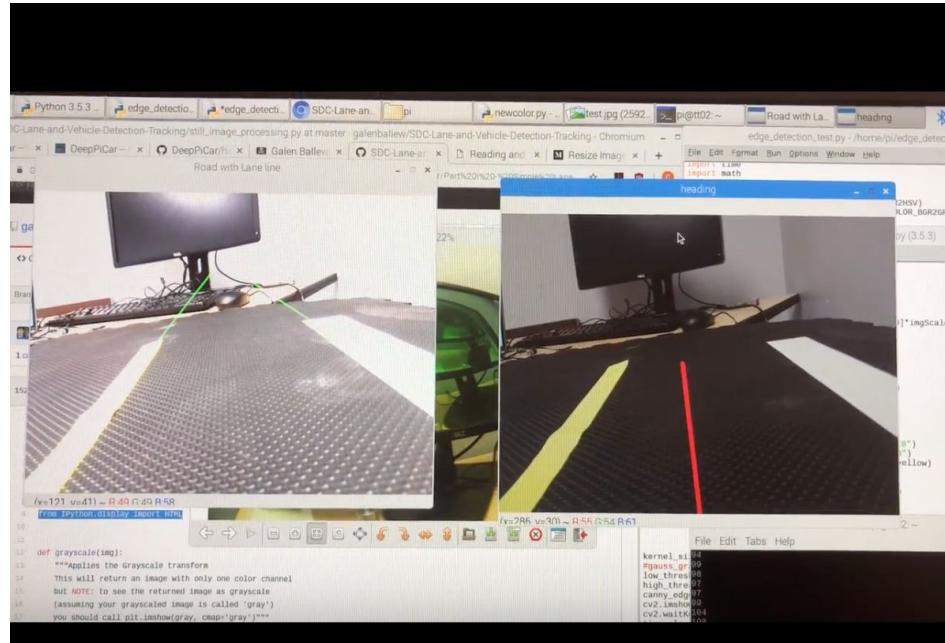


Remote updates



Line Following

- Lane following using a line derived from two lane lines
 - Hope to use this as main driving mechanism



Thank you!

Q&A ?

