# Robotics Nanodegree

## Search & Sample Return Project

This project is modeled after the [NASA sample return challenge](https://www.nasa.gov/directorates/spacetech/centennial_challenges/sample_return_robot/index.html) and it will give you first hand experience with the three essential elements of robotics, which are perception, decision making and actuation. You will carry out this project in a simulator environment built with the Unity game engine.

If you need further information, assistance or referral about a project issue, please contact kiang.ng@hotmail.com.





Ng Fang Kiang

Kiang.ng@hotmail.com

Instagram.com/jorcus96

linkedin.com/in/jorcus

fb.com/ngfangkiang

github.com/jorcus

*Ng Fang Kiang*

Project 1: Follow Me

**Prepared/Updated :** 11 Nov 2017

**Project: Search and Sample Return**

This project is modeled after the [NASA sample return challenge](https://www.nasa.gov/directorates/spacetech/centennial_challenges/sample_return_robot/index.html) and it will give you first hand experience with the three essential elements of robotics, which are perception, decision making and actuation. You will carry out this project in a simulator environment built with the Unity game engine.



**The goals / steps of this project are the following:**

**Training / Calibration**

* Download the simulator and take data in "Training Mode"
* Test out the functions in the Jupyter Notebook provided
* Add functions to detect obstacles and samples of interest (golden rocks)
* Fill in the process\_image() function with the appropriate image processing steps (perspective transform, color threshold etc.) to get from raw images to a map. The output\_image you create in this step should demonstrate that your mapping pipeline works.
* Use moviepy to process the images in your saved dataset with the process\_image() function. Include the video you produce as part of your submission.

**Autonomous Navigation / Mapping**

* Fill in the perception\_step() function within the perception.py script with the appropriate image processing functions to create a map and update Rover() data (similar to what you did with process\_image() in the notebook).
* Fill in the decision\_step() function within the decision.py script with conditional statements that take into consideration the outputs of the perception\_step() in deciding how to issue throttle, brake and steering commands.
* Iterate on your perception and decision function until your rover does a reasonable (need to define metric) job of navigating and mapping.

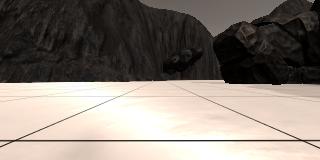
**Notebook Analysis**

1. Run the functions provided in the notebook on test images (first with the test data provided, next on data you have recorded). Add/modify functions to allow for color selection of obstacles and rock samples.

Here is an example of how to include an image in your writeup.[](https://github.com/jorcus/RoboND-Rover/blob/master/misc/rover_image.jpg)

1. Populate the process\_image() function with the appropriate analysis steps to map pixels identifying navigable terrain, obstacles and rock samples into a worldmap. Run process\_image() on your test data using the moviepyfunctions provided to create video output of your result.

And another!

[](https://github.com/jorcus/RoboND-Rover/blob/master/calibration_images/example_grid1.jpg)

Autonomous Navigation and Mapping

1. Fill in the perception\_step() (at the bottom of the perception.py script) and decision\_step() (in decision.py) functions in the autonomous mapping scripts and an explanation is provided in the writeup of how and why these functions were modified as they were.

2. Launching in autonomous mode your rover can navigate and map autonomously. Explain your results and how you might improve them in your writeup.

Note: running the simulator with different choices of resolution and graphics quality may produce different results, particularly on different machines! Make a note of your simulator settings (resolution and graphics quality set on launch) and frames per second (FPS output to terminal by drive\_rover.py) in your writeup when you submit the project so your reviewer can reproduce your results.

Here I'll talk about the approach I took, what techniques I used, what worked and why, where the pipeline might fail and how I might improve it if I were going to pursue this project further.

[](https://github.com/jorcus/RoboND-Rover/blob/master/calibration_images/example_rock1.jpg)

**Future Enhancement**

Nothing is perfect, there’s always lots of fun works to improve.

1. -