# Kratos – Autonomous Subsystem QSTP-2021

## Week 1

This week is devoted to setting up the ROS development environment, understanding basic ROS concepts and implementation and getting familiar with the gazebo simulation environment. A prerequisite to this course is familiarity with python, so if you haven't learned it yet or need to brush up can quickly do so from the following resources. While it was expected to already be comfortable with python, any students comfortable with C should find it extremely easy to pick up quickly.

Python Resources:

- https://www.programiz.com/python-programming
- Sentdex python tutorials on youtube: sentdex

ROS (Robot Operating System) is an extremely useful framework that provides libraries and tools to build automation applications.

Some resources to learn all about it include:

- ROS wiki: (Refer to this to setup ROS environment on your machines, also a useful resource to pick up the basics of using ROS), for tutorials: complete the Beginner Level track.
- https://roboticsbackend.com/category/ros/
- Books: Morgan Quigley's Programming Robots with ROS (great resource)
- Wyatt Newman- A Systematic Approach to learning robot programmig

Next you need to learn about Simulating in Gazebo and using turtlebot3
I have attached a useful pdf (credits to ERC) for learning all about it.

Sample Task (Ungraded):
1. Create a calculator service that takes as input two numbers and a string named operator, performs the operation (addition, subtraction, multiplication or division) according to the operator sent as the string.

2. Write a publisher that makes the turtlebot rotate in a circle.

TASK:

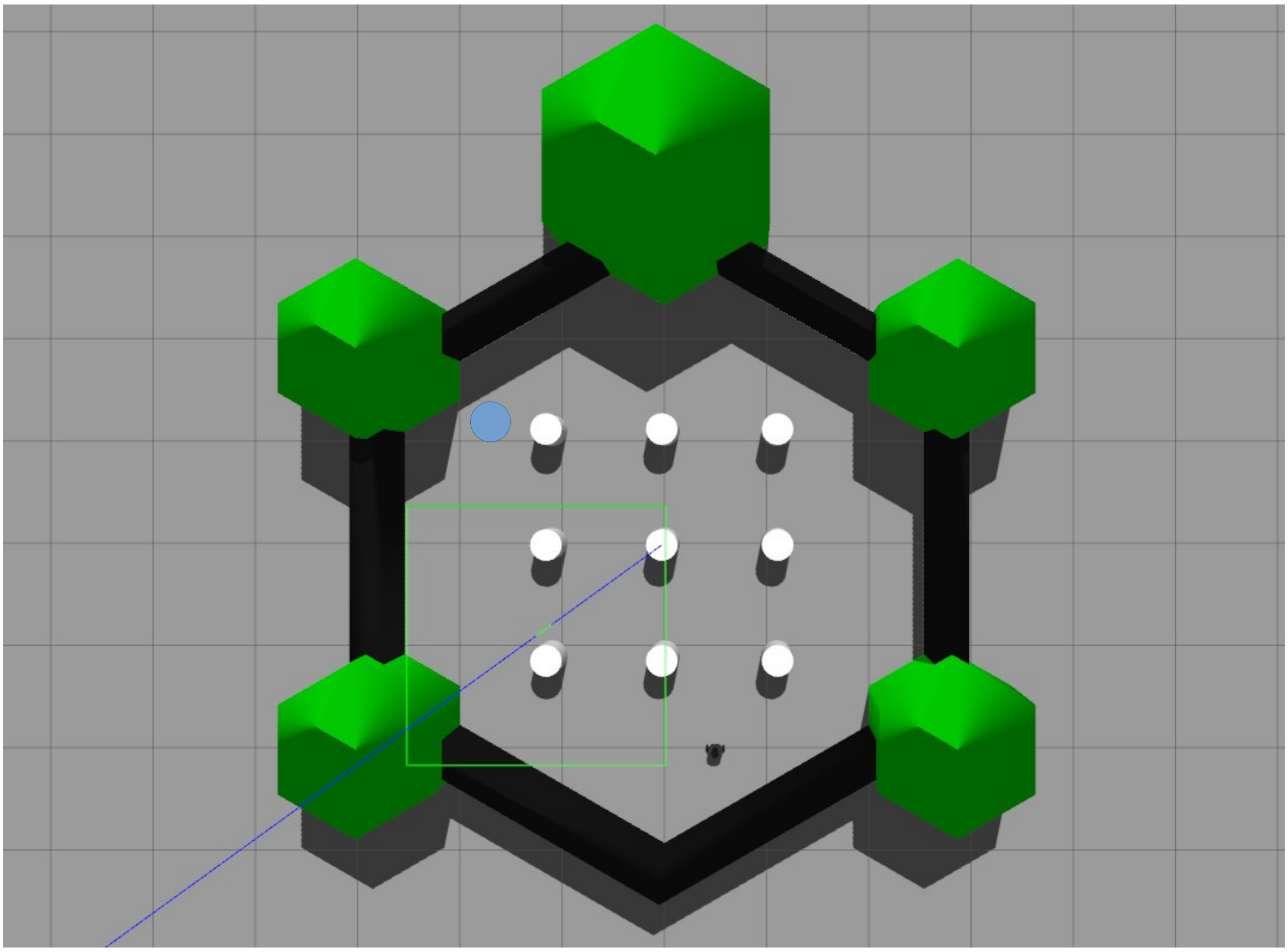
Your task is to create a ROS package that will implement a simple State Machine for a turtlebot and create a service that will be used to switch the bot from one state to another.
See the below site to learn how to implement Oops in ROS:
https://roboticsbackend.com/oop-with-ros-in-python/

Subtask 1: (Till week 1)
Write a python script that contains a class that implements the 3 states of a robot (Move forwards/backwards , Rotate Clockwise/Anticlockwise, Stop) using 3 functions for the 3 states and write a publisher into it that makes the turtlebot move by publishing the velocity commands to /cmd_vel. The class should contain a member variable that holds the state the bot is currently in. For now you can hardcode the state of the bot.

Subtask 2:
Create a services that you can use to switch the bot to the 3 different states in the following format

```
string state        // state to switch to
int32 dir           // dir=1 means bot moves forward, dir=-1 means bot
                            moves backwards
------------------
bool success       // Indicates if the state change occurred succesfully
```

Subtask 3:
Run the turtlbot3_world using
roslaunch turtlebot3_gazebo turtlebot3_world.launch



Try to use the services you have created earlier to move the bot from it’s start point to the the blue goal in the image above. Create a rosbag file and upload it along with the package.

BONUS TASK:
Create another state in which the bot moves in an eight/infinity symbol path. The direction variable will signify the two opposite eights it can form.

Final Submission:
Should include the ROS package you have created with all the scripts as well as the rosbag file of your simulation.