ROB 456: 7.2 Final Project ROS Setup

Outcomes:

1. Clone git repositories into your catkin workspace
2. Create ROS packages and nodes
3. Write ROS launch files
4. Use rviz to visualize ROS data
5. Publish and echo rostopic data from the command line

Instructions:

1. In a terminal, navigate to ~/catkin_ws/src
2. Use the following commands to clone the stagebot_2dnav and simple_navigation_goals repositories to the local machine

```
git clone https://github.com/JenJenChung/stagebot_2dnav.git

git clone https://github.com/JenJenChung/simple_navigation_goals.git
```

3. Create a package in ~/catkin_ws/src called nav_bundle

```
catkin_create_pkg nav_bundle move_base gmapping
```

4. Navigate to the nav_bundle folder and create a new folder called launch
5. Navigate to the launch folder and create a new launch file called nav_bundle.launch

```
gedit nav_bundle.launch &
```

6. Write a launch file to run slam_gmapping, move_base and the waypoint navigation clients
7. Create a package in ~/catkin_ws/src called rob456_project

```
catkin_create_pkg rob456_project roscpp rospy move_base gmapping
```

8. Navigate to the rob456_project folder and create a new folder called launch
9. Navigate to the launch folder and create a new launch file called rob456_project.launch

```
gedit rob456_project.launch &
```

10. Write a launch file to run stageros and rviz
11. Navigate to the rob456_project/src folder and create a new node called move_in_square

```
gedit move_in_square.py &
```

12. Create a program that sends waypoints to make the robot move in a square
13. Include the move_in_square node in the project launch file

Notes:

- Make sure your .py nodes are executable, you can change their permissions by running

```
chmod +x node_name.py
```

- When you launch rob456_project.launch with the move_in_square.py node, the robot will not initially begin moving. This is because nothing is publishing to /cmd_vel. To start the move_in_square algorithm, either publish a zero velocity message to /cmd_vel

```
rostopic pub /cmd_vel geometry_msgs/Twist '[0.0, 0.0, 0.0]'
'[0.0, 0.0, 0.0]' -1
```

  Or send it a single waypoint (through clicking in rviz or via the command line), the robot will begin moving in a square once it has reached the first waypoint

- move_in_square.py is a demonstration of a waypoint command function and is in no way a perfect solution for sending sequential waypoints. You should aim to improve this functionality in your own project.