|  |  |
| --- | --- |
| **ROS Workspace Setup** | https://courses.dojofordrones.com/courses/1119413/lectures/24006293 |
| When previously launching Gazebo, error was thrown up indicating a ROS node for Gazebo had not been initialized.  This is because the world file contains a virtual camera feed that expects a ROS network to connect to |  |
| Gazebo-ROS should already be installed. ‘Launch’ files both start Gazebo with a world file and initialize the ROS network  Use command ‘roscd gazebo\_ros’ to navigate to the gazebo\_ros’ node and ‘cd launch’ to navigate to the launch file directory |  |
| Create a new launch file named ‘iris\_world.launch’ by duplicating the ‘empty\_world.launch’ file  Sudo cp empty\_world iris\_world.launch |  |
| Edit ‘iris\_world.launch’ file  Sudo gedit iris\_world.launch  File contains instructions for launching gazebo and ros together, contains many arguments that can be passed in or set to specific values or pathways for the launch process |  |
| Arg named “world\_name” corresponds to the default world file called when Gazebo-ROS is launched  Default world was changed to be ‘aruco\_landing.world’, which is a world file located within the ‘ardupilot\_gazebo’ repo |  |
| Launch gazebo-ros with the command:  Roslaunch gazebo\_ros iris\_world.launch  This will launch gazebo and initialize the ROS network, loading the world file specified as the default in the launch file |  |

|  |  |
| --- | --- |
| **Program ROS Publishers and Subscribers** |  |
| Create new directory ‘catkin\_ws’ while navigated to the ‘courseRoot’ directory  Cd catkin\_ws  Cd src |  |
| Navigate to ‘src’ and run catkin\_make  Cd src  Catkin\_make  This creates ‘build’, ‘devel’, and ‘src’ directory within ‘catkin\_ws’ |  |
| Add ‘local\_setup.bash’ file in the ‘devel’ directory to .bashrc file:  Sudo echo “source /home/user/courseRoot/catkin\_ws/devel/setup.bash” >> ~/.bashrc |  |
| Navigate to ‘src’ directory and create ROS package:  Cd courseRoot/catkin\_ws/src  Catkin\_create\_pkg example\_pkg rospy roscpp  ‘Example\_pkg’ is the package name and ‘rospy’ and ‘roscpp’ are the required dependencies |  |