

# Simulator

## Gazebo Install

To install software for the gazebo simulator please follow these steps:

```
sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu trusty main" > /etc/apt/sources.list.d/gazebo-latest.list'  
wget http://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -  
sudo aptitude update  
sudo aptitude install robosub-simulator
```

Then clone the repo: [https://github.com/PalouseRobosub/simulator\\_gazebo](https://github.com/PalouseRobosub/simulator_gazebo) to your catkin workspace.

## Running Gazebo Simulator

Next build everything by running `catkin_make`. Then run the following command:

```
roslaunch robosub_simulator gazebo.launch
```

## (Legacy) Unity Simulator Setup

To install the Unity Editor, run:

```
sudo aptitude install unity-editor
```

This is a large package (~3.7G), it will take a while!

To get the code for the simulator, `cd` into the robosub repo, then run

```
git submodule init  
git submodule update
```

This is also a lot of data, it will take a while. When it is done, all the code will be in a folder called *simulator* within the repo.

Once installed and you have an account setup, you need to open the folder containing `/Assets` and `/ProjectSettings` as a project. The project will generate all of the meta files needed to run the simulator. Once generated, you will have a list of scenes that you can open in the directory view. Opening Competition will pull up the competition scene.

Once the scene is loaded you can dynamically move the submarine around and edit scripts within the project. The software will run normally with Sensor, Thruster, and Camera being run within the engine. Make sure to have the correct IP address set in the broker settings file. Localhost (127.0.0.1)

should be used for running isolated personal machines. To verify that the software is talking normally you can run helm and view sensor data or view the broker log to make sure data is being sent.

The control system is currently loaded from the control file within the parent repo. Thrusters currently need to be in order (front, back, left, right, top, bot) for the control system to output correctly. It's best to checkout the simulator branch in robosub to grab a test control system layout where the sub is balanced. Once you run control, the broker, and the simulator you can use messenger to test movement and thruster outputs.

From:

<http://robosub-vm.eecs.wsu.edu/wiki/> - **Palouse RoboSub Technical Documentation**

Permanent link:

<http://robosub-vm.eecs.wsu.edu/wiki/cs/simulator/start?rev=1481146187>



Last update: **2016/12/07 13:29**