

## Gazebo para ROS Kinetic

Primero es necesario contar con ROS, en este caso ROS Kinetic.

### Gazebo

Para iniciar es necesario instalar los paquetes de Gazebo. Esto se puede lograr insertando la siguiente línea de comando en una terminal nueva para una instalación binaria:

```
$ sudo apt-get install ros-kinetic-gazebo-ros-pkgs ros-kinetic-gazebo-ros-contro  
l
```

### ardrone\_autonomy

Cuando se termine la instalación se tendrá instalado Gazebo, ahora es necesario la instalación de ardrone\_autonomy. Para esto se escribe la siguiente línea de comando en la misma terminal:

```
$ apt-get install ros-kinetic-ardrone-autonomy
```

### tum\_simulator

Una vez hecho esto debemos crear un Workspace para el tum\_simulator.

Los siguientes pasos nos generaran un workspace en nuestro directorio home. En la misma terminal o una nueva se escriben las siguientes líneas de comando:

```
$ mkdir -p ~/tum_simulator_ws/src  
$ cd ~/tum_simulator_ws/src  
$ catkin_init_workspace
```

Descargar paquete KineticGazebo desde:

<https://github.com/ernestocruzesq/KineticGazebo>

Descomprimir paquete KineticGazebo.

Descomprimir el paquete encontrados dentro (tar.gz):

- tum\_simulator

Entrar a: home/tum\_simulator\_ws/src por medio del explorador.

Dentro de este directorio copiar las carpetas que se tienen en el paquete KineticGazebo:

- tum\_simulator

Una vez hecho esto se abre una terminal nueva y se insertan los comandos:

```
$ -cd ~/tum_simulator_ws/  
$ catkin_make
```

```
$ source devel/setup.bash
$ roslaunch cvg_sim_gazebo ardrone_testworld.launch
```

### Forma alternativa

Otra forma de instalación:

```
$ git clone https://github.com/AutonomyLab/ardrone_autonomy.git
$ git clone https://github.com/angelsantamaria/tum_simulator.git
$ cd ..
$ rosdep install --from-paths src --ignore-src --rosdistro indigo -y
```

Estas líneas descargan el contenido de ardrone\_autonomy y del simulador en el src. Seguido a esto regresamos a la carpeta principal del workspace e instalamos las dependencias necesarias. Una vez terminado este procedimiento se escribe la línea:

```
$ catkin_make
```

Con esta línea está terminada la instalación, ahora para correrlo es necesario escribir la siguiente línea de código para tener de fuente el workspace que estamos usando:

```
$ source devel/setup.bash
```

Para correr la simulación de Gazebo se utiliza:

```
$ roslaunch cvg_sim_gazebo ardrone_testworld.launch
```

Esto hará correr la simulación de Gazebo, en ocasiones la primera ocasión que se abre no funciona correctamente. La primera vez se abre con la pantalla principal en negro y el menú vacío, por lo que se tiene que volver abrir y dejar esperar un tiempo hasta que funcione correctamente.

### Mundos

En el paquete KineticGazebo es posible encontrar un archivo con el nombre ardrone\_square.

Este paquete se mueve al directorio en el explorador:

```
home/tum_simulator_ws/src/tum_simulator/cvg_sim_gazebo/worlds
```

Seguido a esto se entra en el directorio:

```
home/tum_simulator_ws/src/tum_simulator/cvg_sim_gazebo/launch
```

Aquí se abre el archivo ardrone\_testworld.launch con algún software de edición y se cambia la línea de código:

```
<arg name="world_name" value ="$(find cvg_sim_gazebo)/worlds/ardrone_testworld.world">
```

La ultima parte "ardrone\_testworld.world" se cambia por el nombre del archive que se movio a worlds.