

# Compilation steps

```
cd ardupilot
sudo git submodule update --init --recursive
sudo ./waf configure --board sitl
sudo ./waf copter
```

if this fails, head over to ~/DroneSim-Installer and ``sudo rm -rf ardupilot`` and run

- `sudo apt-get install git -y`
- `git clone https://github.com/ArduPilot/ardupilot.git`

*(repeat the initial steps)*

## Launching flightgear -

- head over to ~/DroneSim-Installer/ardupilot/Arducopter
- run `../Tools/autotest/fg_quad_view.sh``

## Setting up the environment -

- head over to ~/DroneSim-Installer/ardupilot/Arducopter
- run ``sudo ../Tools/autotest/sim_vehicle.py -L KSFO``

(if this fails, open a fresh terminal window (not tab) and run it again, sometimes flightgear leaves cache files that may cause some problems, launching a new terminal window essentially reloads the .bashrc file which clears the cache)

## Launching mission planner -

- head over to ~/DroneSim-Installer/abdaz
- run ``mono MissionPlanner.exe``

## Setting up the environment for 2D map with console -

Although this is not required if you already launch the 3D simulator, if all fails, then the 2D simulation would work without any compilation errors of sorts. It will still automatically connect to mavlink and show up in missionplanner.

To launch this, head over to ardupilot/ArduCopter and run ``sudo ../Tools/autotest/sim_vehicle.py - --console --map``