PSI 3442 - Gazebo

Fernando Zolubas Preto

Na aula de hoje

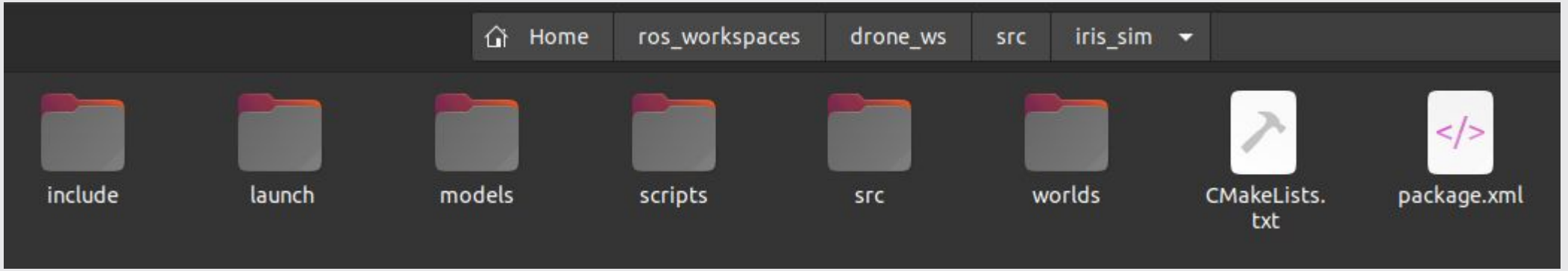
- Gazebo
  - Elementos principais
    - World, models, launch
  - Modificando o solo
  - Adicionando um céu
  - Adicionando objetos
  - Adicionando um drone Iris do firmware PX4
  - Adicionando uma câmera ao drone

Elementos básicos de Gazebo

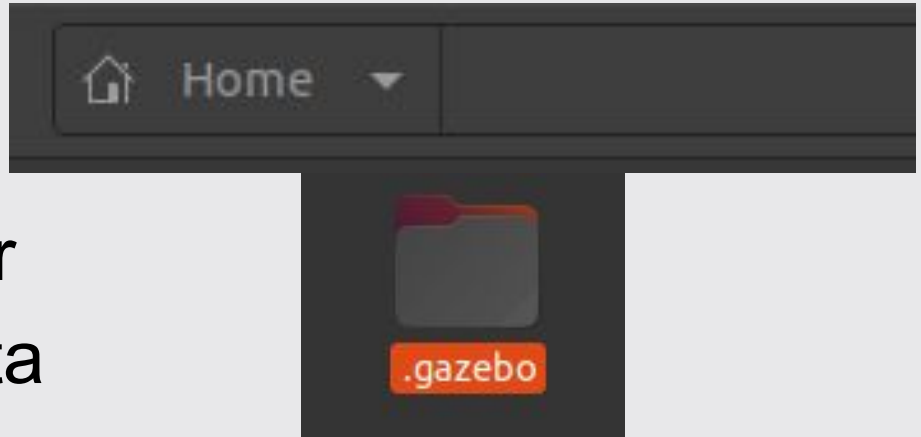
Elementos básicos de Gazebo

Esquema de pastas:

Rospackge: iris_sim



Elementos gazebo: ~/.gazebo



use ctrl+h para ver essa pasta

Gazebo: Worlds

Worlds:

Arquivo para definir o mundo do gazebo e seus elementos. Além disso, define-se a física do mundo.

```xml
<?xml version="1.0" ?>
<sdf version="1.5">
  <world name="default">
    <!-- A global light source -->

    <scene>
      <sky>
        <clouds>
          <speed>12</speed>
        </clouds>
      </sky>
      <ambient>0.95 0.95 0.95 1</ambient>
      <background>0.3 0.3 0.3 1</background>
      <shadows>true</shadows>
    </scene>

    <include>
      <uri>model://sun_customized</uri>
    </include>

    <!-- A ground plane -->

    <include>
      <uri>model://ground_plane</uri>
    </include>

    <include>
      <uri>model://grass_plane</uri>
      <pose>0 0 0 0 0 0</pose>
    </include>

    <!-- Parede -->
    <include>
        <uri>model://parede</uri>
        <pose>4 1 0 0 0 1.57</pose>
    </include>
```

Gazebo: Worlds

Worlds:

Arquivo para definir o mundo do gazebo e seus elementos. Além disso, define-se a física do mundo.

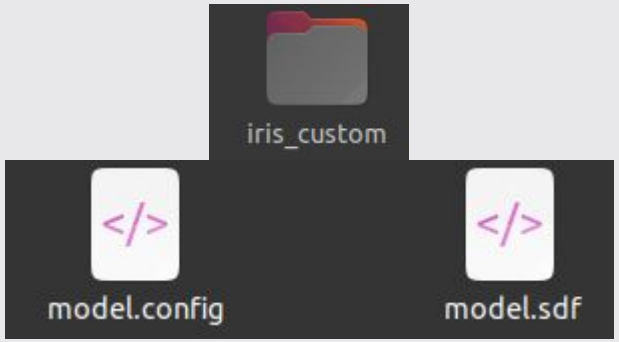
Home | ros_workspaces | drone_ws | src | iris_sim | worlds

```
57      <!-- Física -->
58      <physics name='default_physics' default='0' type='ode'>
59        <gravity>0 0 -9.8066</gravity>
60        <ode>
61          <solver>
62            <type>quick</type>
63            <iters>10</iters>
64            <sor>1.3</sor>
65            <use_dynamic_moi_rescaling>0</use_dynamic_moi_rescaling>
66          </solver>
67          <constraints>
68            <cfm>0</cfm>
69            <erp>0.2</erp>
70            <contact_max_correcting_vel>100</contact_max_correcting_vel>
71            <contact_surface_layer>0.001</contact_surface_layer>
72          </constraints>
73        </ode>
74        <max_step_size>0.004</max_step_size>
75        <real_time_factor>1</real_time_factor>
76        <real_time_update_rate>250</real_time_update_rate>
77        <magnetic_field>6.0e-6 2.3e-5 -4.2e-5</magnetic_field>
78      </physics>
79    </world>
80  </sdf>
```

Gazebo: Models tipo 1

Models:

Tipo 1: Ficam no pacote iris_sim e são usados para customizar o drone



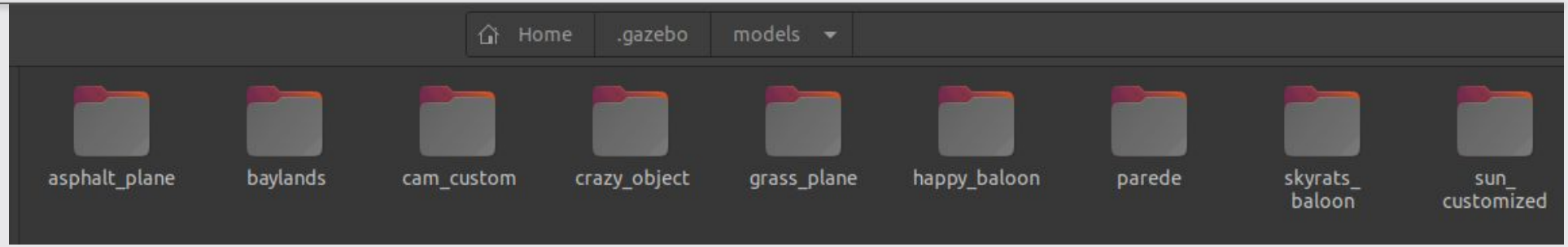
Home > ros_workspaces > drone_ws > src > iris_sim > models

home > user > ros_workspaces > drone_ws > src > iris_sim > models > iris_custom > model.sdf

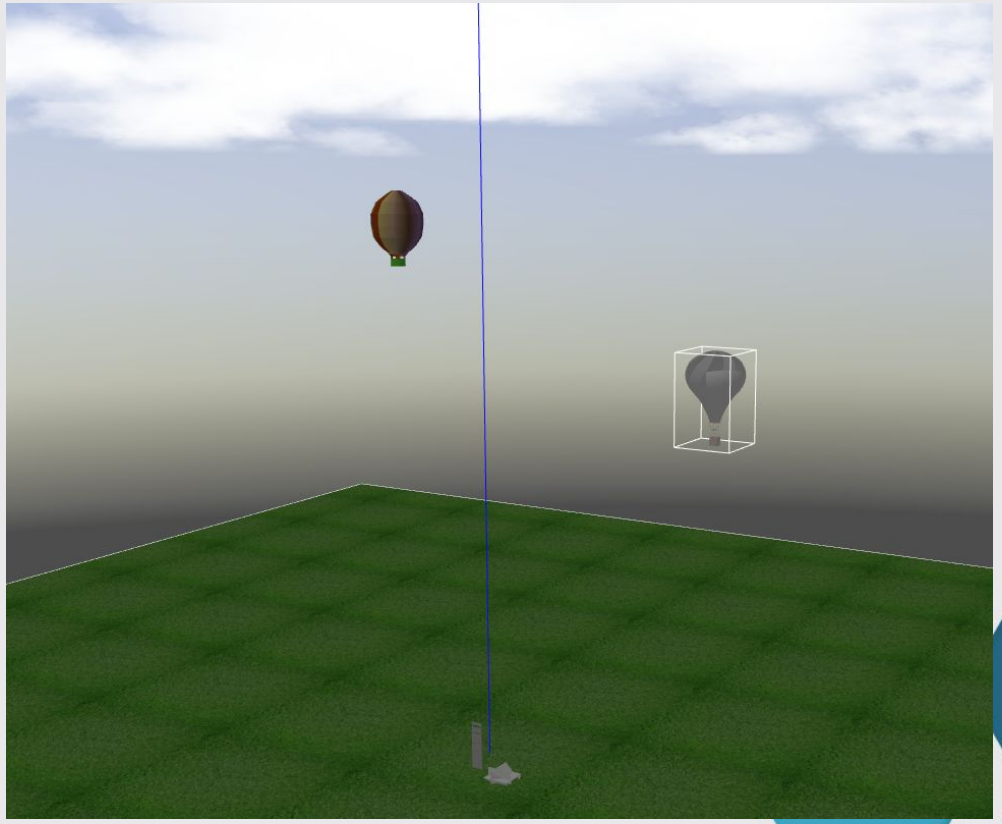
```
<?xml version="1.0" ?>
<sdf version='1.5'>
  <model name='iris_custom'>

    <include>
      <uri>model://iris</uri>
    </include>

    <!-- Add your sensors -->
    <include>
          <!--<uri>model://fpv_cam</uri>-->
          <uri>model://cam_custom</uri>
          <pose>0.097 0 0.02 0 0 0</pose>
    </include>
    <joint name="fpv_cam_joint" type="fixed">
      <!--<child>fpv_cam::link</child>-->
      <child>cam_custom::link</child>
      <parent>iris::base_link</parent>
      <axis>
        <xyz>0 0 1</xyz>
        <limit>
          <upper>0</upper>
          <lower>0</lower>
        </limit>
      </axis>
    </joint>

  </model>
</sdf>
```

Gazebo: Models tipo 2

Models:



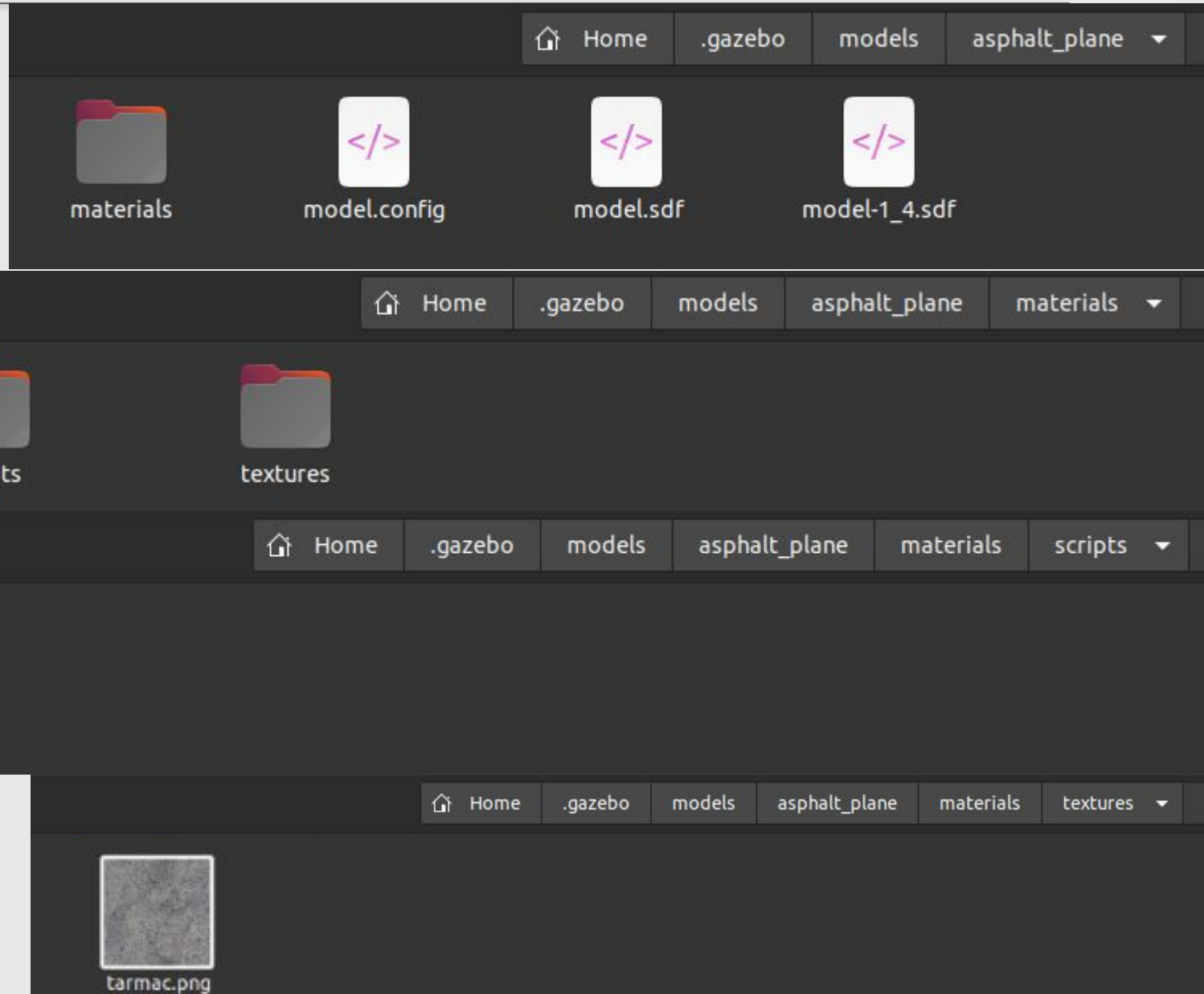
Tipo 2: Ficam no diretório .gazebo/models
e são usados para definir objetos do mundo gazebo: Chão, balões, paredes, etc. (exceto o drone que nesse caso vem do firmware)



Gazebo: Models tipo 2: aslphalt plane

Models:

asphalt plane:
Define o solo
do mundo do gazebo.



Gazebo: Models tipo 2: aslphalt plane

Models:

asphalt plane:
Este arquivo define
o material, características
visuais e interação com a luz
(ambient,diffuse,specular)*
e a textura do objeto “tarmac.png”

Open

asphalt.material
~/.gazebo/models/asphalt_plane/materials/scripts

```
material vrc/asphalt
{
        technique
        {
                pass
                {
                        ambient 0.5 0.5 0.5 1.0
                        diffuse 0.5 0.5 0.5 1.0
                        specular 0.2 0.2 0.2 1.0 12.5

                        texture_unit
                        {
                                texture tarmac.png
                                filtering anistropic
                                max_anisotropy 16
                                scale 0.1 0.1
                        }
                }
        }
}
```

*Obs:Confira a documentação sobre light na documentação sobre arquivos sdf (simulation description format)
http://sdformat.org/tutorials?tut=spec_materials&cat=specification&

Gazebo: Models tipo 2: aslphalt plane

Models:

asphalt plane:

Esse arquivo descreve o objeto,
seu tamanho, sua descrição (scripts)
e textura (textures)

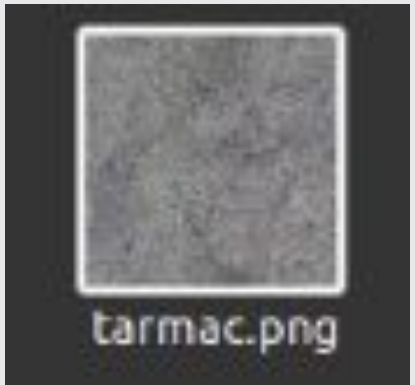
home > user > .gazebo > models > asphalt_plane > model.sdf

```
1  <?xml version="1.0" ?>
2  <sdf version="1.5">
3    <model name="asphalt_plane">
4      <static>true</static>
5      <link name="link">
6        <collision name="collision">
7          <geometry>
8            <box>
9              <size>20 20 .1</size>
10           </box>
11         </geometry>
12       </collision>
13       <visual name="visual">
14         <cast_shadows>false</cast_shadows>
15         <geometry>
16           <box>
17             <size>20 20 .1</size>
18           </box>
19         </geometry>
20         <material>
21           <script>
22             <uri>model://asphalt_plane/materials/scripts</uri>
23             <uri>model://asphalt_plane/materials/textures</uri>
24             <name>vrc/asphalt</name>
25           </script>
26         </material>
27       </visual>
28     </link>
29   </model>
30 </sdf>
```

Gazebo: Models: Elementos padrão

Models:

Nesse exemplo vemos um “model” que é “estático”, com propriedades de “colisão” definidas com um objeto quase plano (20,20,0.1). E ainda são definidas propriedades “visuais”, isto é, um objeto (20,20,0.1) de textura vrc/asphalt. O objeto asphalt plane é definido como um “link”.

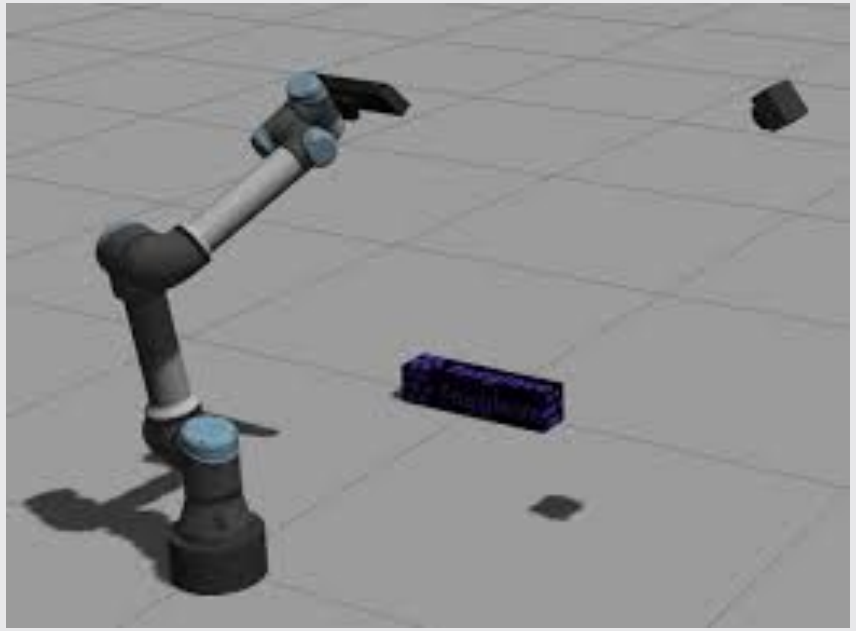


home > user > .gazebo > models > asphalt_plane > model.sdf

```
1   <?xml version="1.0" ?>
2   <sdf version="1.5">
3     <model name="asphalt_plane">
4       <static>true</static>
5       <link name="link">
6         <collision name="collision">
7           <geometry>
8             <box>
9               <size>20 20 .1</size>
10            </box>
11          </geometry>
12        </collision>
13        <visual name="visual">
14          <cast_shadows>false</cast_shadows>
15          <geometry>
16            <box>
17              <size>20 20 .1</size>
18            </box>
19          </geometry>
20          <material>
21            <script>
22              <uri>model://asphalt_plane/materials/scripts</uri>
23              <uri>model://asphalt_plane/materials/textures</uri>
24              <name>vrc/asphalt</name>
25            </script>
26          </material>
27        </visual>
28      </link>
29    </model>
30  </sdf>
```

Gazebo: Models: Links vs Joints

Models:
Intuitivamente os links são os objetos
e as joints são as juntas.



No gazebo a definição é parecida, mas todos os objetos são links (incluindo as juntas).
As joints são somente definições de como um objeto filho se movimenta em relação a um objeto pai ao qual o filho está associado fisicamente.
Exemplo, como uma haste do robô se move em relação a junta a qual ela se conecta.

Gazebo: Launch

Launch:
Prepara o mundo chamando o drone IRIS definido no firmware PX4, o modelo de customização do drone “iris_custom” e o mundo “custom_world”.
Este ultimo chama os demais elementos do mundo.

Em um terminal execute: **roslaunch iris_sim simulation.launch** oara rodar a simulação.

home > user > ros_workspaces > drone_ws > src > iris_sim > launch > simulation.launch

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <launch>
3      <include file="$(find px4)/launch/mavros_posix_sitl.launch">
4          <arg name="vehicle" value="iris"/>
5          <arg name="sdf" value="$(find iris_sim)/models/iris_custom/model.sdf"/>
6          <arg name="world" value="$(find iris_sim)/worlds/custom_world.world" />
7      </include>
8  </launch>
```