



Department of
Computer Engineering

به نام خدا



Amirkabir University of Technology
(Tehran Polytechnic)

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی کامپیوتر
اصول علم ربات

تمرین سری اول

نام و نام خانوادگی	حمیدرضا همتی
شماره دانشجویی	۹۶۳۱۰۷۹
تاریخ ارسال گزارش	

فهرست گزارش سوالات

سوال ۱ – محاسبه درجه آزادی ۳

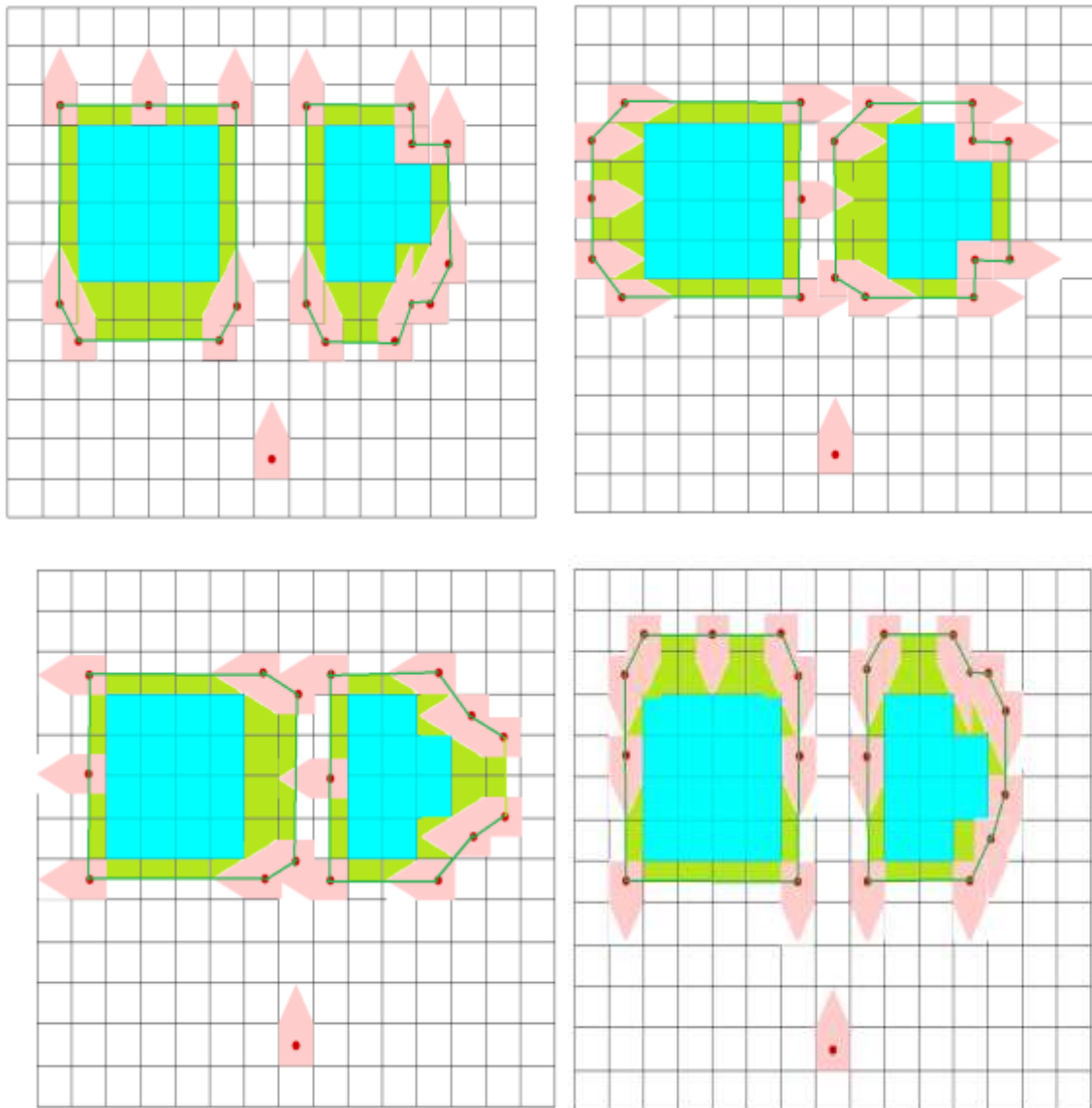
- سوال ۲ - محاسبه درجه آزادی ۴
- سوال ۳ - تشخیص C-Space ۵
- سوال ۴ - سناریو اول ۶
- سوال ۵ - سناریو دوم ۷

سوال ۱ - محاسبه درجه آزادی

سوال ۲ - محاسبه درجه آزادی

سوال ۳ - تشخیص C-Space

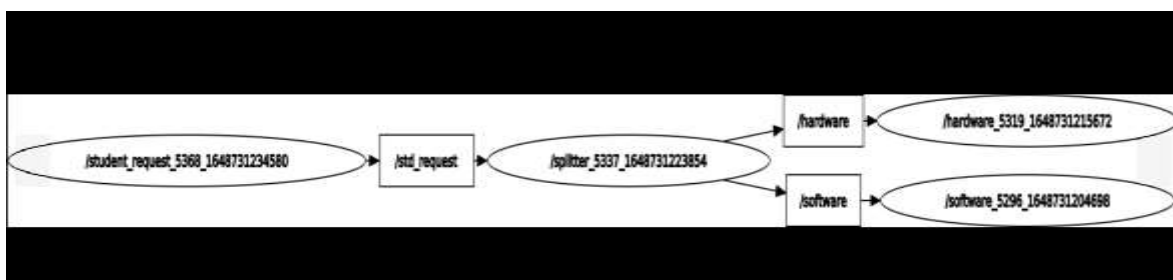
در این سوال با توجه به این که ربات میتواند دوران های 90° داشته باشد پس ربات کلا برای حرکت های خود چهار جهت 0° ، 90° ، 180° و 270° درجه میتواند داشته باشد. با توجه به این نکته باید c-space را برای این ۴ زاویه به دست آوریم. در شکل های زیر c-space به ترتیب از ربات در حالت 0° درجه تا 270° رسم شده است.



سوال ۴ – سناریو اول

پکیج ایجاد شده برای این سوال با نام rand_Stu درون مسیر 1 codes/scenario قرار داده شده است. درون این پکیج یک فولدر با نام msg که فرمت custom message درون آن مشخص شده است قرار دارد. همچنین فایل های کد درون پوشه src قرار گرفته اند. درون این پوشه ۴ فایل کد با نام های random_Student, splitter, hardware و software قرار دارند که با توجه به صورت سوال هر کدام تسک مربوط به خود را انجام میدهند.

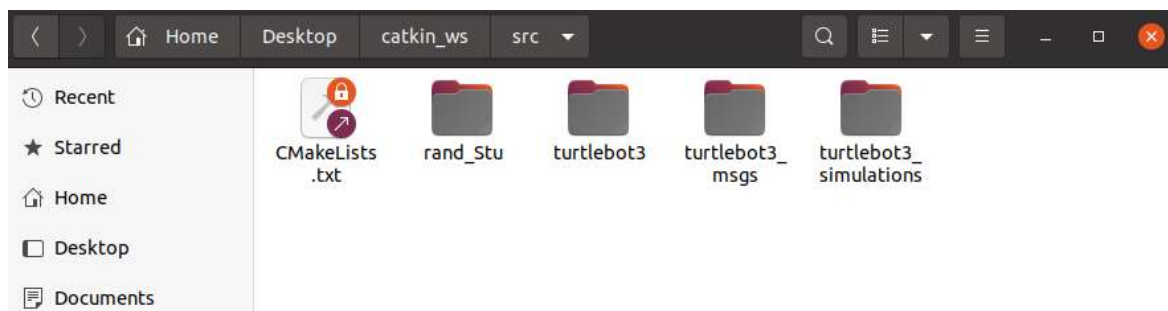
در نهایت پس از run کردن تمام node ها گراف مربوط به شکل زیر میباشد.



شکل زیر بخش از خروجی بعد از run کردن هر چهار node است. Node سمت چپ random_student میباشد. Node وسط splitter است و node های سمت راست به ترتیب از بالا hardware و software میباشد.

سوال ۵ – سناریو دوم

برای این بخش ابتدا کامند های گفته شده در فایل پروژه برای clone کردن turtlebot3 زده شد. پس از زدن این کامند ها پوشه catkin_ws/src به شکل زیر درآمد.



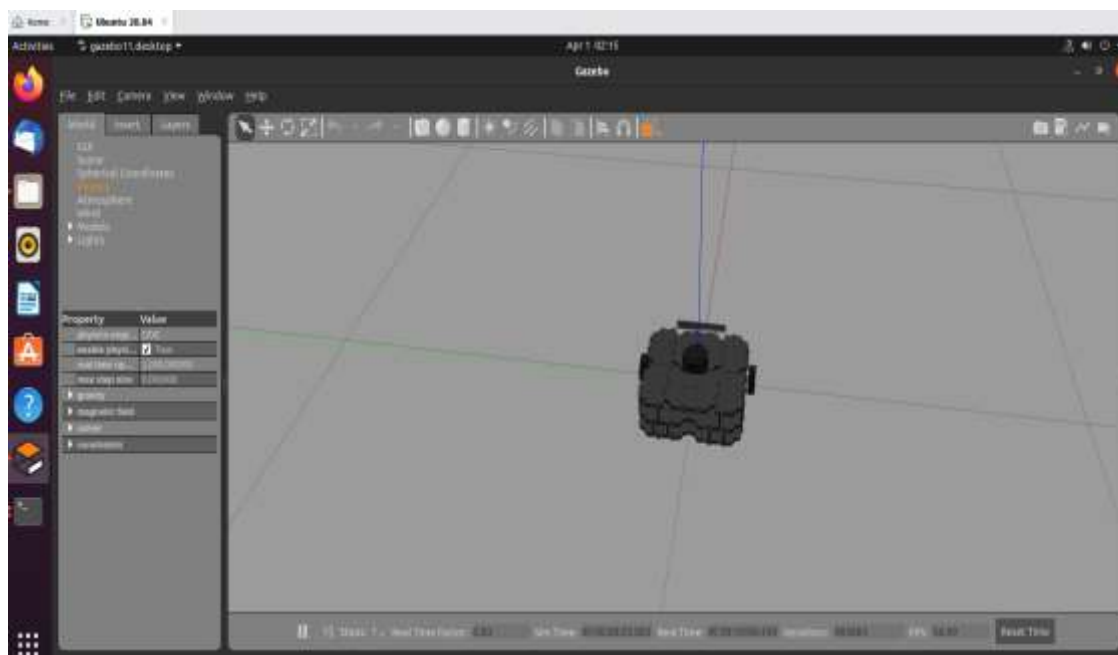
سپس به انجام موارد گفته شده به ترتیب پرداخته شد. در زیر این مراحل با ترتیبی که در فایل پروژه گفته شده است توضیح داده شده است.

۱. در این مرحله محیط شبیه سازی gazebo را با ربات waffle در محیط empty_world اجرا کردم. در این بخش مختصات ابتدایی ربات را تغییر نداده ام. در شکل های زیر کامند اجرا شده در ترمینال همچنین محیط gazebo را میتوانید مشاهده کنید.

```
/home/hamidreza/Desktop/catkin_ws/src/turtlebot3_simul...
hamidreza@ubuntu:~/Desktop/catkin_ws/src$ export TURTLEBOT3_MODEL=waffle
hamidreza@ubuntu:~/Desktop/catkin_ws/src$ roslaunch turtlebot3_gazebo turtlebot3_empty_world.launch
WARNING: Package name "rand_Stu" does not follow the naming conventions. It should start with a lower case letter and only contain lower case letters, digits, underscores, and dashes.
... logging to /home/hamidreza/.ros/log/53bc0fc0-b199-11ec-97cb-a9a7a17c9601/roslaunch-ubuntu-2631.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

WARNING: Package name "rand_Stu" does not follow the naming conventions. It should start with a lower case letter and only contain lower case letters, digits, underscores, and dashes.
xacro: in-order processing became default in ROS Melodic. You can drop the option.
started roslaunch server http://ubuntu:43955/

SUMMARY
=====
PARAMETERS
```



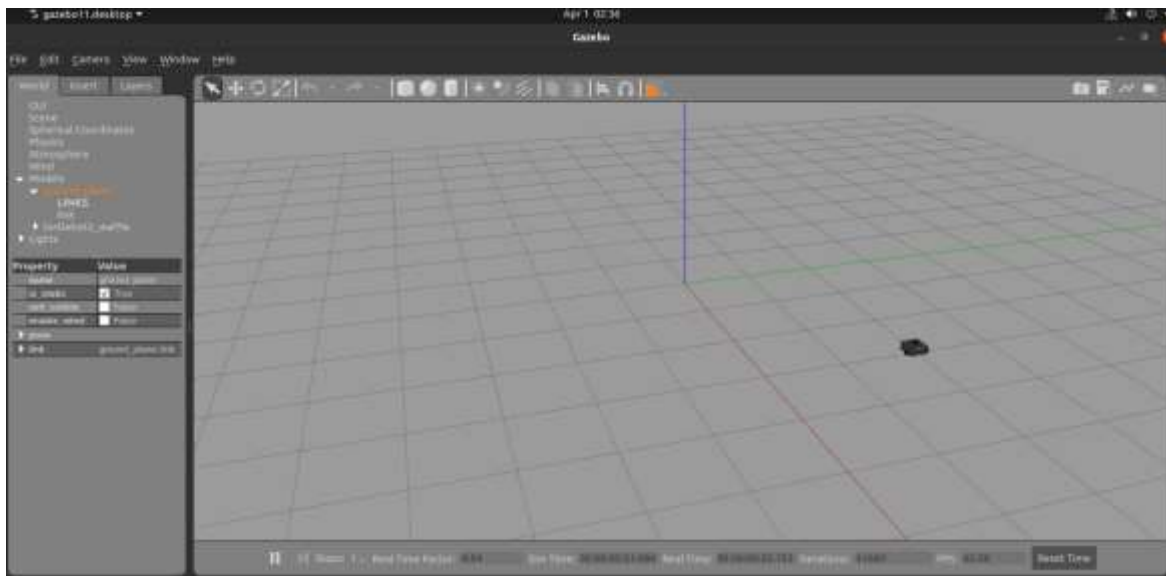
۲. در این مرحله در محیط empty_world ربات waffle را اجرا کردم با این تفاوت که مختصات اولیه ربات درون محیط $x = 3$ و $y = 2$ میباشد. برای تغییر مختصات فایل turtlebot3_empty_world.launch را درون پوشه catkin_ws/src/turtlebot3_simulations/turtlebot3_gazebo/launch به شکل زیر تغییر دادم. داخل این فایل مختصات اولیه ربات درون محیط empty_world مشخص شده است که با تغییر آن ها میتوان مختصات شروع ربات درون محیط را تغییر داد.

```

1 <launch>
2   <arg name="model" default="$(env TURTLEBOT3_MODEL)" doc="model type [burger, waffle,
  waffle pi]"/>
3   <arg name="x_pos" default="3.0"/>
4   <arg name="y_pos" default="2.0"/>
5   <arg name="z_pos" default="0.0"/>
6
7   <include file="$(find gazebo_ros)/launch/empty_world.launch">
8     <arg name="world_name" value="$(find turtlebot3_gazebo)/worlds/empty.world"/>
9     <arg name="paused" value="false"/>
10    <arg name="use_sim_time" value="true"/>
11    <arg name="gui" value="true"/>
12    <arg name="headless" value="false"/>
13    <arg name="debug" value="false"/>
14  </include>
15
16  <param name="robot_description" command="$(find xacro)/xacro --inorder $(find
  turtlebot3_description)/urdf/turtlebot3_$(arg model).urdf.xacro" />
17
18  <node pkg="gazebo_ros" type="spawn_model" name="spawn_urdf" args="-urdf -model turtlebot3_$(
  arg model) -x $(arg x_pos) -y $(arg y_pos) -z $(arg z_pos) -param robot_description" />
19 </launch>

```

شکل زیر محیط gazebo را بعد از این تغییر نشان میدهد.



۳. در این مرحله ربات waffle را در محیط empty_world با استفاده از کیبورد کنترل میکنیم. برای انجام این کار یک ترمینال جدید باز کرده و کامند های زیر را میزنیم.

```

/home/hamidreza/Desktop/catkin_ws/src/turtlebot3/turtle...
/home/hamidreza/Desktop/catkin_ws... /home/hamidreza/Desktop/catkin_ws...
hamidreza@ubuntu:~/Desktop/catkin_ws/src$ export TURTLEBOT3_MODEL=waffle
hamidreza@ubuntu:~/Desktop/catkin_ws/src$ roslaunch turtlebot3_teleop turtlebot3
_teleop_key.launch
WARNING: Package name "rand_Stu" does not follow the naming conventions. It shou
ld start with a lower case letter and only contain lower case letters, digits, u
nderscores, and dashes.
... logging to /home/hamidreza/.ros/log/c0aa73b8-b1a0-11ec-97cb-a9a7a17c9601/ros
launch-ubuntu-5503.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ubuntu:35479/

SUMMARY
=====

PARAMETERS
* /model: waffle
* /rostdistro: noetic
* /rosversion: 1.15.14

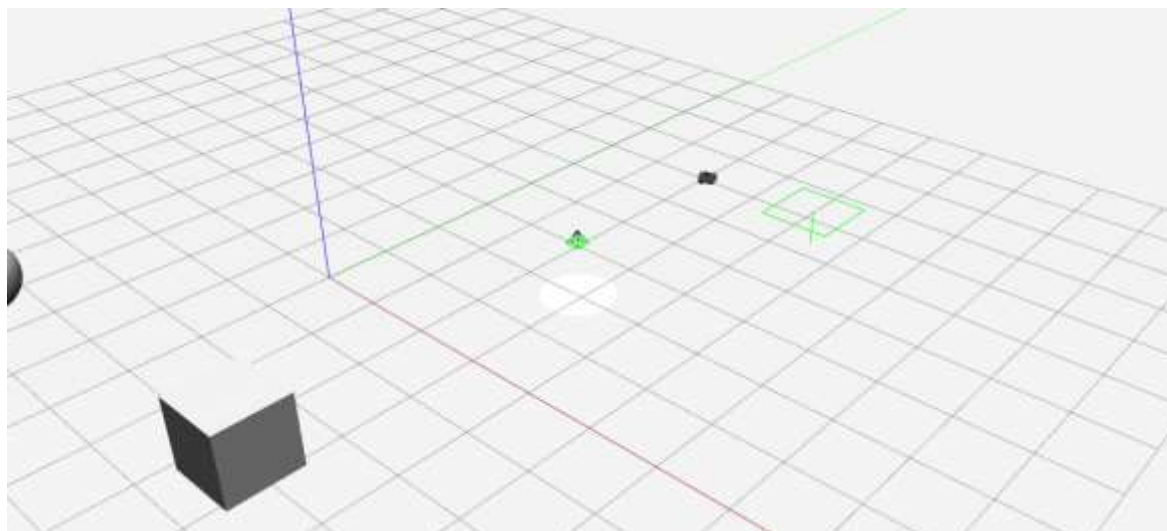
NODES
/

```

سپس برای حرکت دادن ربات در gazebo باید select روی همین ترمینال باشد که دستور بالا را درون آن زدیم. هنگامی که با استفاده از کیبورد ربات را حرکت میدهم log های روی ترمینال به شکل زیر هستند.

[illegible]

محیط شبیه سازی gazebo بعد از حرکت دادن ربات در شکل زیر قابل مشاهده است.



۴. در این مرحله ابتدا کارهای زیر برای ایجاد محیط funky maze انجام شد.

a. فایل funky-maze.world درون پوشه worlds قرار داده شد.

b. در پوشه launch یک فایل با نام turtlebot3_funky_maze_world.launch ایجاد شد.

در شکل داده های درون این فایل مشخص است.

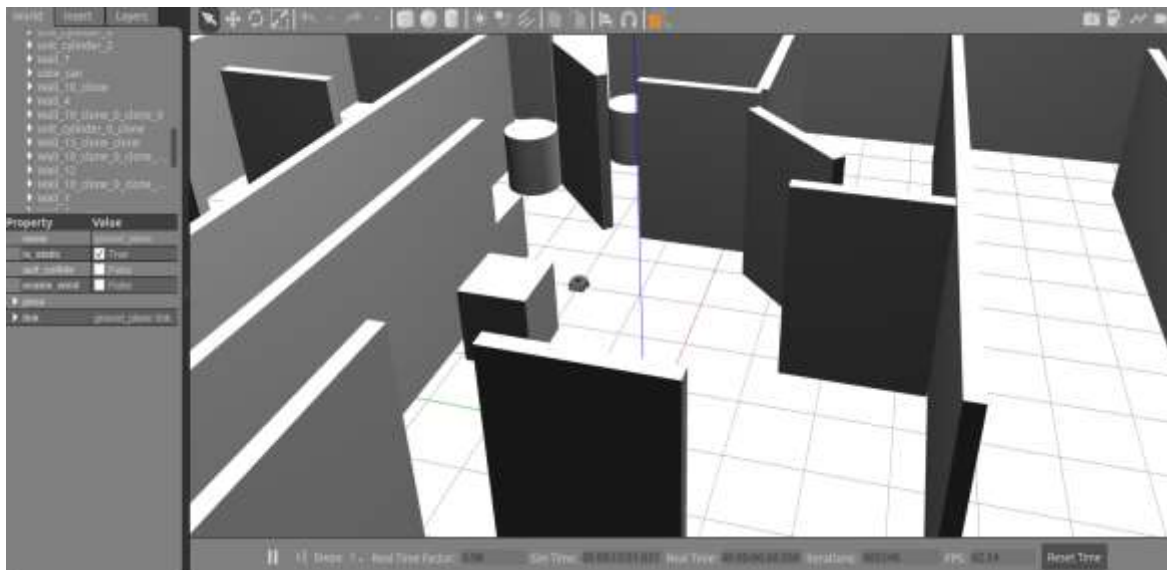
```

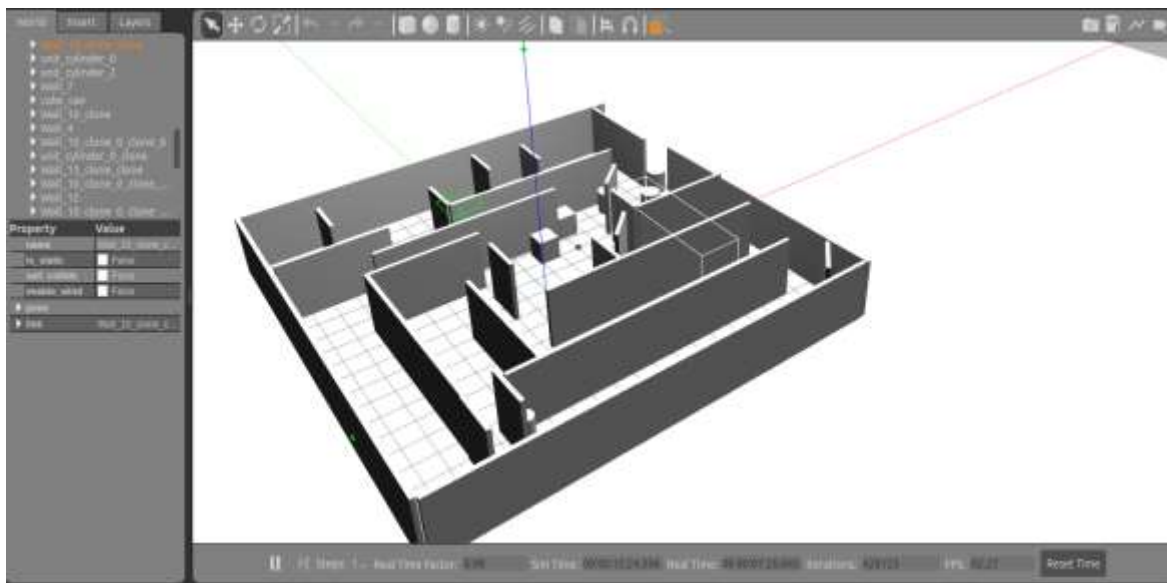
1 <launch>
2   <arg name="model" default="$(env TURTLEBOT3_MODEL)" doc="model type
   [burger, waffle, waffle_pi]"/>
3   <arg name="x_pos" default="3.0"/>
4   <arg name="y_pos" default="2.0"/>
5   <arg name="z_pos" default="0.0"/>
6
7   <include file="$(find gazebo_ros)/launch/empty_world.launch">
8     <arg name="world_name" value="$(find turtlebot3_gazebo)/worlds/-
   funky-maze.world"/>
9     <arg name="paused" value="false"/>
10    <arg name="use_sim_time" value="true"/>
11    <arg name="gui" value="true"/>
12    <arg name="headless" value="false"/>
13    <arg name="debug" value="false"/>
14  </include>
15
16  <param name="robot_description" command="$(find xacro)/xacro --
   inorder $(find turtlebot3_description)/urdf/turtlebot3_$(arg
   model).urdf.xacro" />
17
18  <node pkg="gazebo_ros" type="spawn_model" name="spawn_urdf" args="-
   urdf -model turtlebot3_$(arg model) -x $(arg x_pos) -y $(arg y_pos) -
   z $(arg z_pos) -param robot_description" />
19 </launch>
20

```

در شکل های زیر محیط funky-maze بعد از ران شدن کامند زیر قابل مشاهده است.

roslaunch turtlebot3_gazebo turtlebot3_funky_maze_world.launch





در شکل زیر تغییر موقعیت ربات با استفاده از کیبورد مشخص است.

