

Launch File Code and Explanation

Code:

```
<launch>

  <!-- Gazebo -->

  <include file="$(find gazebo_ros)/launch/empty_world.launch">

    <arg name="world_name" value="$(find
follower_robot_pkg)/world/wall_line.world"/>

    <arg name="paused" value="false"/>

    <arg name="use_sim_time" value="true"/>

    <arg name="gui" value="true"/>

    <arg name="headless" value="false"/>

    <arg name="debug" value="false"/>

  </include>

  <!-- Spawning Robot -->

  <param name="robot_description" command="$(find xacro)/xacro $(find
turtlebot3_description)/urdf/turtlebot3_burger_for_autorace_2020.urdf.xacro " />

  <node pkg="gazebo_ros" type="spawn_model" name="spawn_urdf" args="-urdf -model
turtlebot3_burger -x 0.0 -y 0.0 -z 0.0 -Y 1.557 -param robot_description" />

  <!-- Obstacle - Wall Gazebo Simulation - Node -->

  <node pkg="follower_robot_pkg" type="world_control.py" name="World_control_node"
output="screen" required="true"/>

  <!-- Wall Following Robot - Node -->

  <node pkg="follower_robot_pkg" type="wall_line_following.py"
```

```
name="wall_line_following_node" output="screen" required="true"/>

<!-- Camera view - Node -->

<node pkg="image_view" type="image_view" name="image_view" output="screen">

    <remap from="image" to="/camera/image"/>

</node>

</launch>
```

Explanation:

Launch File Code Explanation

1. Gazebo Setup:

- This section launches an empty Gazebo world.
- The 'world_name' argument specifies the custom world file 'wall_line.world' located in the 'follower_robot_pkg' package.
- Parameters control Gazebo's state, like paused state, GUI mode, and more.

2. Robot Spawning:

- Spawns the 'turtlebot3_burger' robot in the Gazebo world.
- URDF model details are set using 'xacro'.
- Position (-x 0.0 -y 0.0 -z 0.0) and orientation (-Y 1.557) are defined, with the model held by the 'robot_description' parameter.

3. Wall Obstacle Node:

- Runs the 'world_control.py' script in 'follower_robot_pkg', controlling wall obstacles for simulation interaction.

4. Wall Following Node:

- Runs 'wall_line_following.py' in 'follower_robot_pkg', handling the robot's wall-following behavior.

5. Camera View Node:

- Uses 'image_view' to display the robot's camera feed.
- Remaps the camera feed to '/camera/image', providing visuals from the robot's camera perspective.