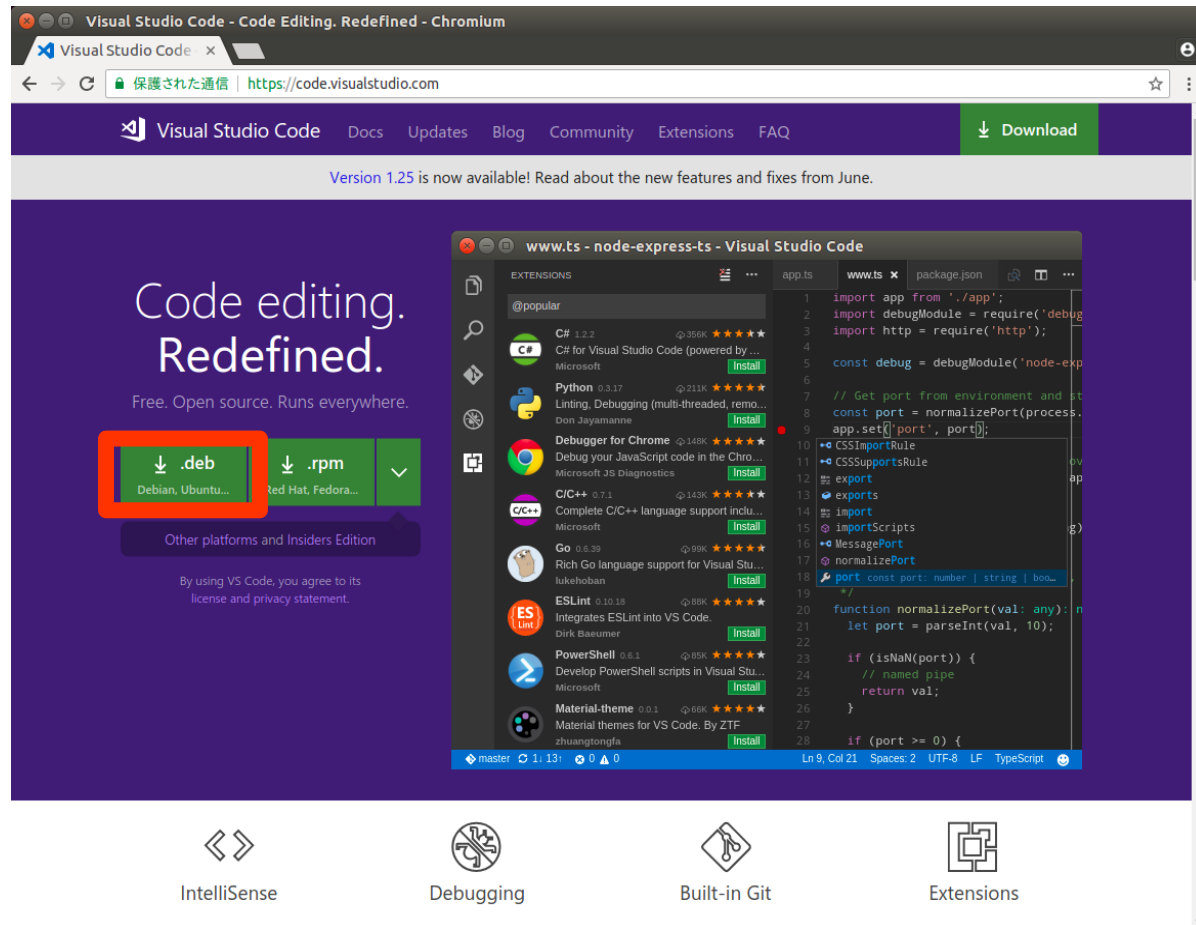


# VSCodeでROSデバッグ v0.6

チームリラックマ  
gogo5nta

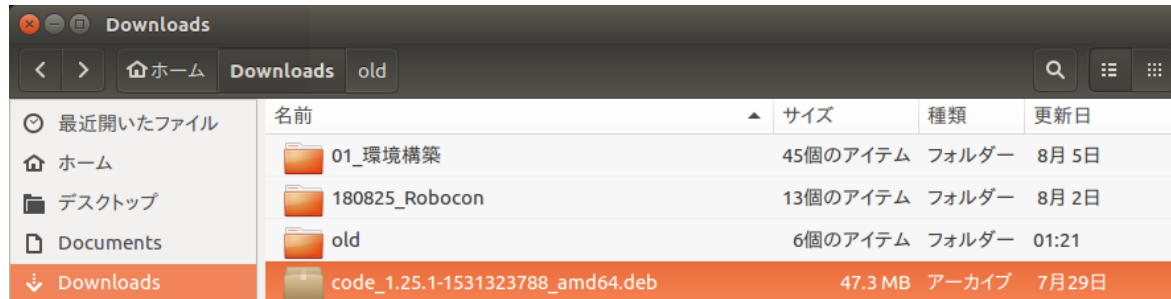
2018.08.07

# 1. VSCodeの入手

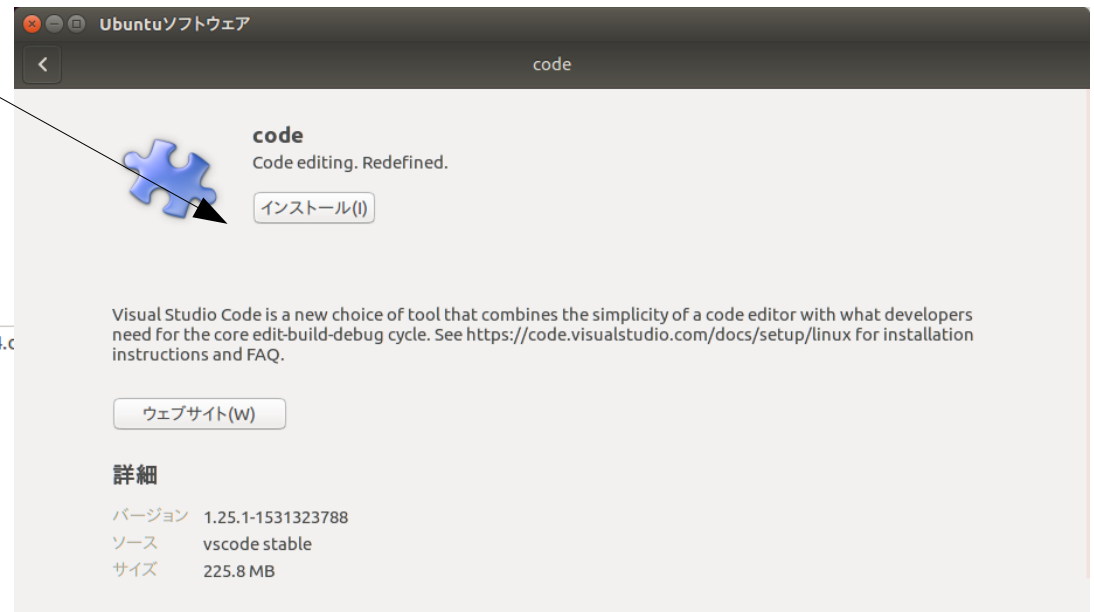


<https://code.visualstudio.com/>  
上記URLから.debをダウンロード

## 2. VSCodeのインストール



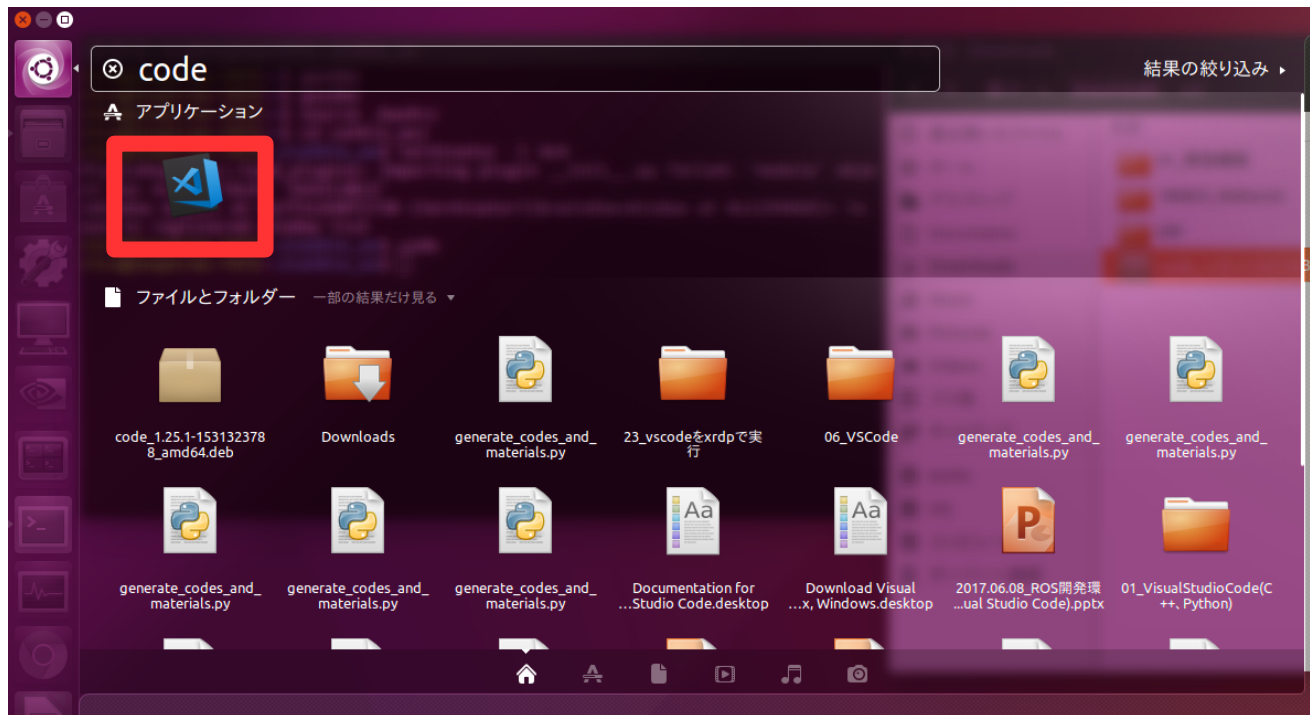
マウス右 > ソフトウェアのインストールで開く



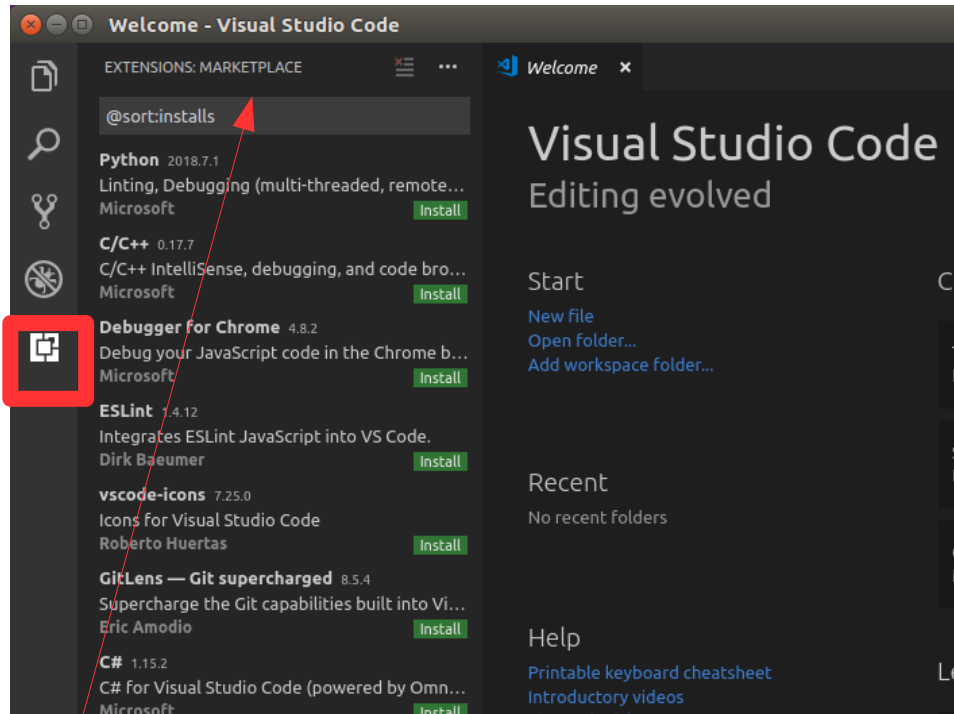
# 3. VSCode(code)起動

ターミナル上で  
\$ code

またはcodeで検索



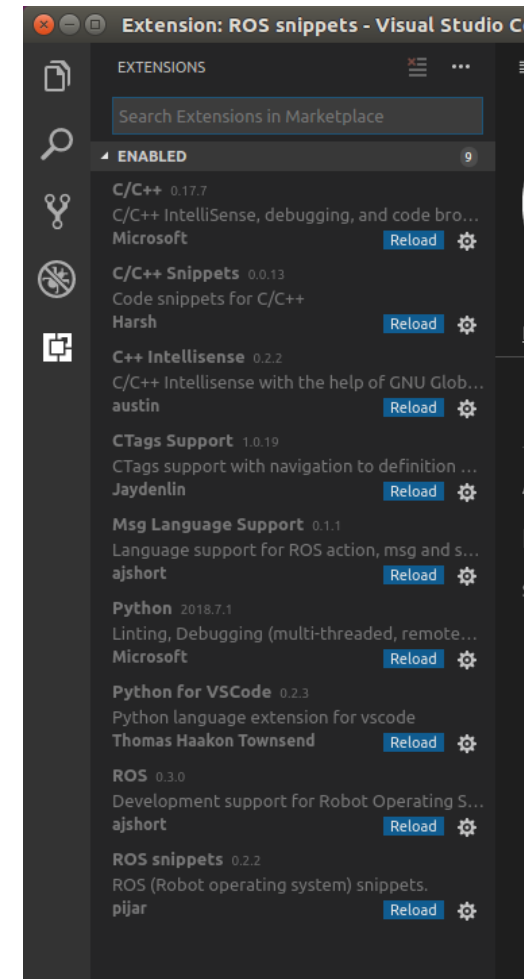
# 4. VSCodeの設定(初回)



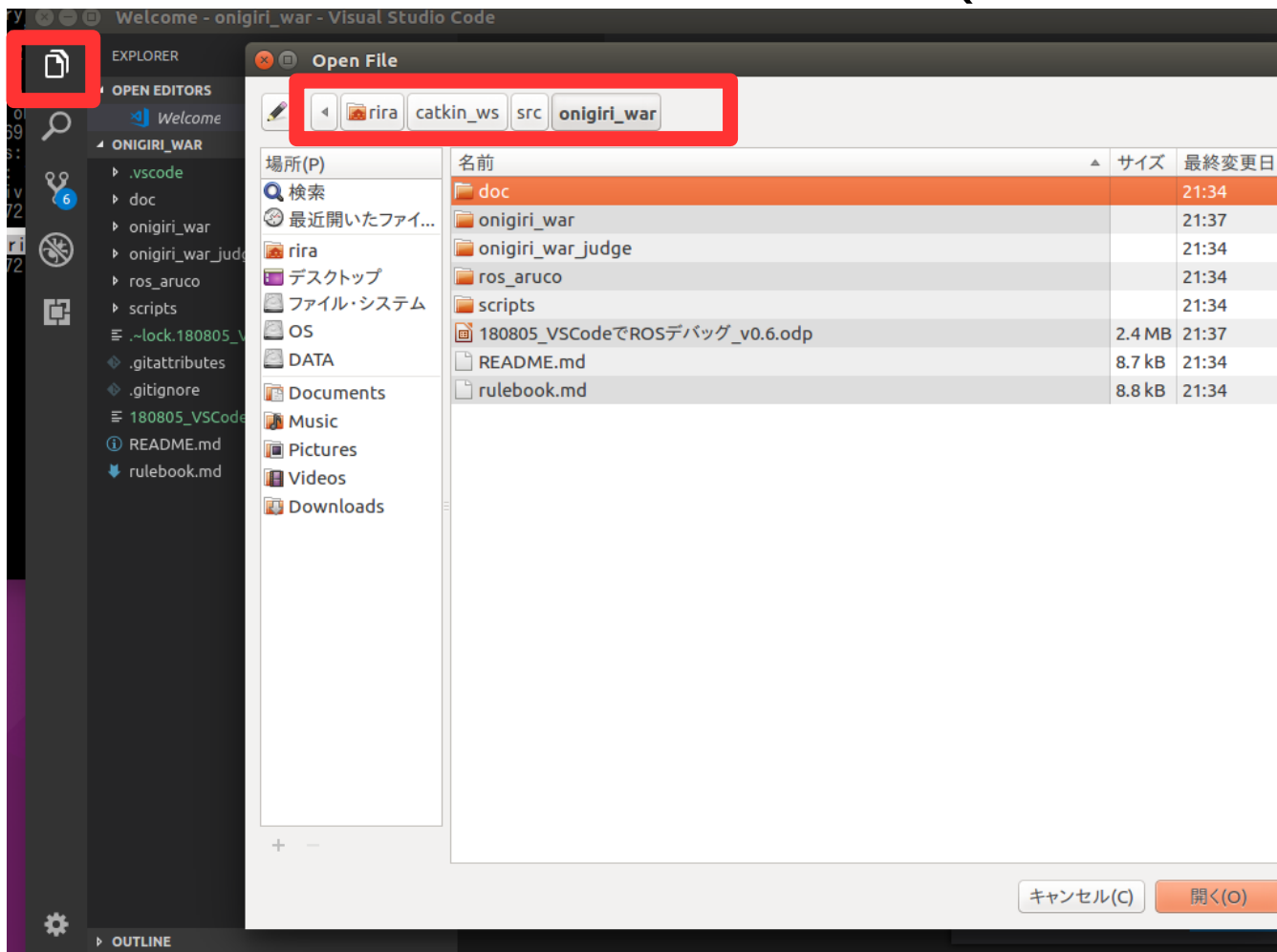
C++, python、ROSで検索

拡張機能のおすすめ

- C/C++
- C/C++ Snippets
- C++ intellisense
- Ctags
- Python
- Python for VSCode
- ROS
- ROS snippets



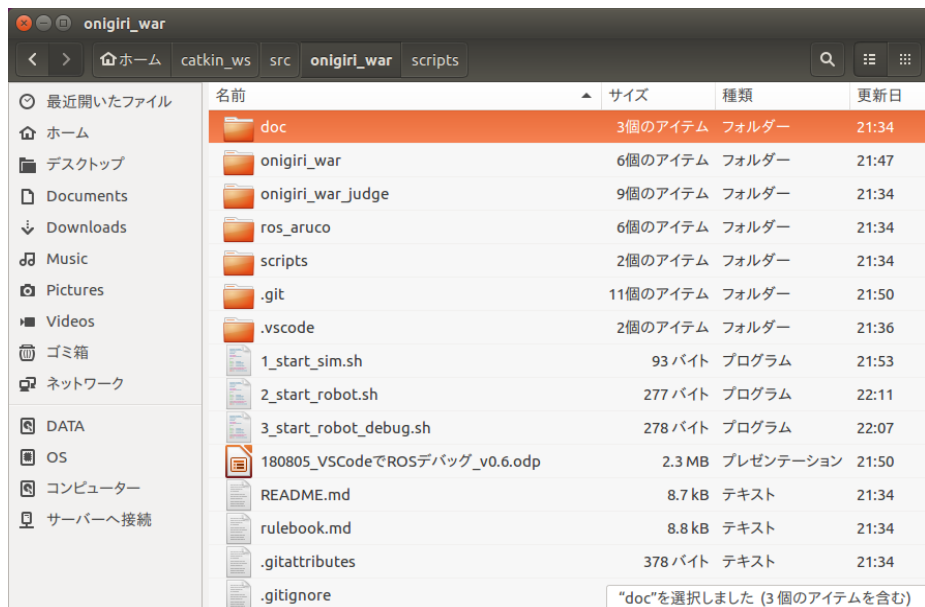
## 5. フォルダの設定(プロジェクト毎初回のみ)



Open Folderで、以下を選択  
~/catkin\_ws/src/onigiri\_war

# 6. Debug用に修正

debug用に以下のファイルを作成



onigiri\_war/onigiri\_war/launch/sim\_robot\_debug\_run.launch (debug用robot起動)

```
<?xml version="1.0"?>
<launch>

<!-- blue side robot -->
<group ns="blue_bot">
  <!-- enemy bot run -->
  <node pkg="onigiri_war" type="enemy.py" name="enemyRun"
output="screen"/>
</group>

<!-- red side robot -->
<group ns="red_bot">
  <!-- Your robot control node run -->
  <!-- include file="$(find onigiri_war)/launch/your_onigiri.launch" /-->
</group>

</launch>
```

1\_start\_sim.sh (sim起動)

```
#!/bin/bash

echo "start onigiri_setup_sim "
roslaunch onigiri_war onigiri_setup_sim.launch
```

2\_start\_robot.sh (robot起動)

```
#!/bin/bash

echo "start robot script(debug)"

# set judge server state "running"
bash onigiri_war_judge/test_scripts/set_running.sh localhost:5000

# launch robot control node
roslaunch onigiri_war sim_robot_run.launch
#roslaunch onigiri_war/launch/sim_robot_debug_run.launch
```

3\_start\_robot\_debug.sh (debug用robot起動)

```
#!/bin/bash

echo "start robot script(debug)"

# set judge server state "running"
bash onigiri_war_judge/test_scripts/set_running.sh
localhost:5000

# launch robot control node
# roslaunch onigiri_war sim_robot_run.launch
roslaunch onigiri_war/launch/sim_robot_debug_run.launch
```

# 7. ROS Pythonでデバッグ

The screenshot shows a multi-terminal environment for ROS. The top-left terminal displays a series of 'Requests Error Please Check URL' messages. The top-right terminal shows the launch configuration for 'onigiri war' with ROS\_MASTER\_URI=http://localhost:11311 and process[blue\_bot/enemyRun-1] started with pid [14409]. The bottom-left terminal shows the command 'rira@Inspiron-7472: ~/catkin\_ws/src/onigiri\_war' and 'rira@Inspiron-7472: ~/catkin\_ws/src/onigiri\_war\$'. The bottom-right terminal shows the command 'rira@Inspiron-7472: ~/catkin\_ws/src/onigiri\_war\$'. The VS Code interface at the bottom shows the 'randomRun.py' file with a 'debug python: current File' button highlighted in a red box. The 'Call Stack' panel shows the execution flow from 'MainThread' to 'calcTwist' to 'strategy' to 'randomRun.py'. The 'Debug Console' shows a 'SyntaxError: invalid syntax' error in 'randomRun.py' at line 22.

複数ターミナル起動は、terminatorがおすすめ。

#インストール

\$ sudo apt-get install terminator

\$ cd ~/catkin\_ws/src/onigiri\_war

ターミナル1: bash 1\_start\_sim.sh

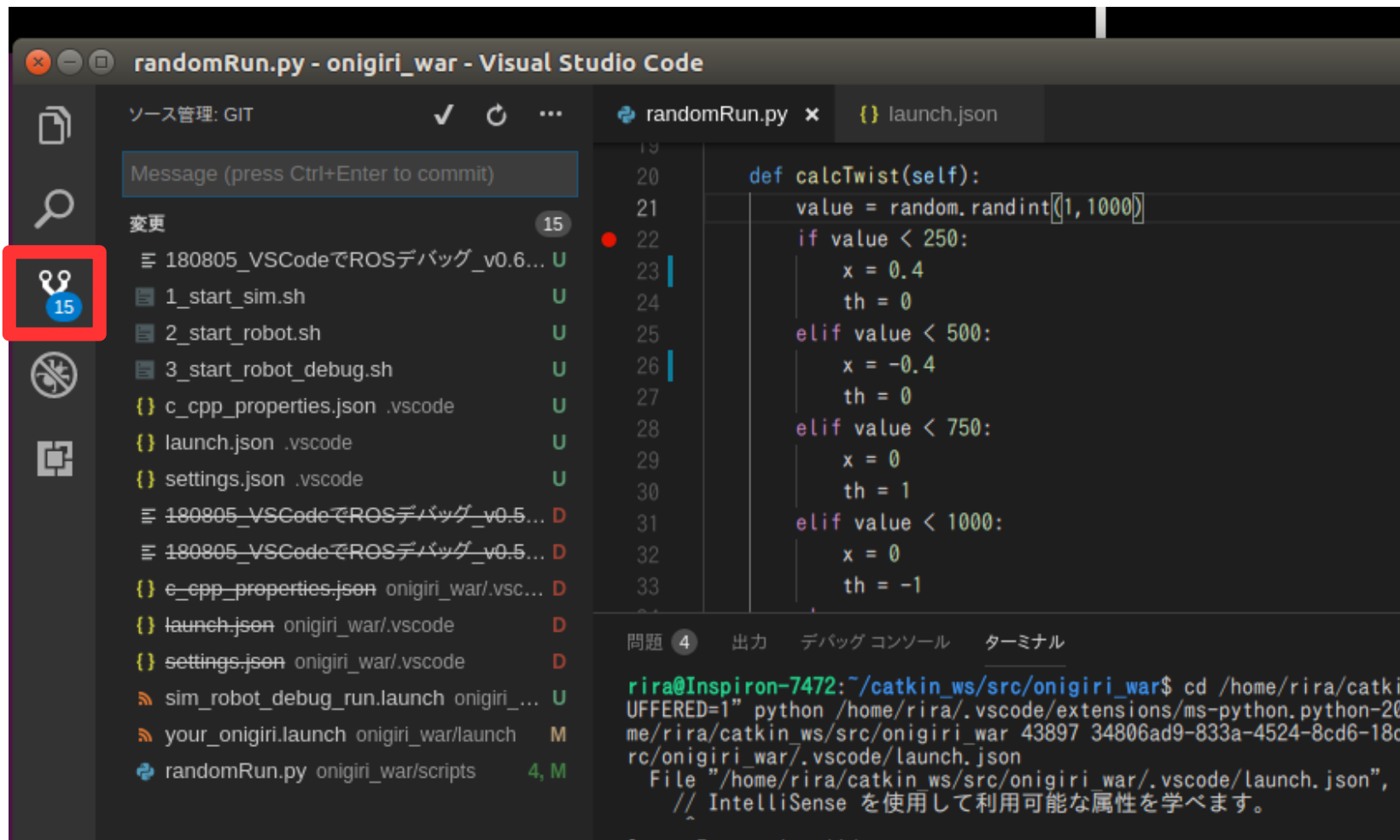
ターミナル2: bash 2\_start\_robot.sh (通常)

bash 3\_start\_robot\_debug.sh (debug)

以下をファイルを開き、debug python: current Files (F5)を実行  
onigiri\_war/onigiri\_war/scripts/randomRun.py



## 8. gitによるソース管理



VSCoide上から簡単にgit管理が可能  
※ 事前にgit cloneしたソースの場合