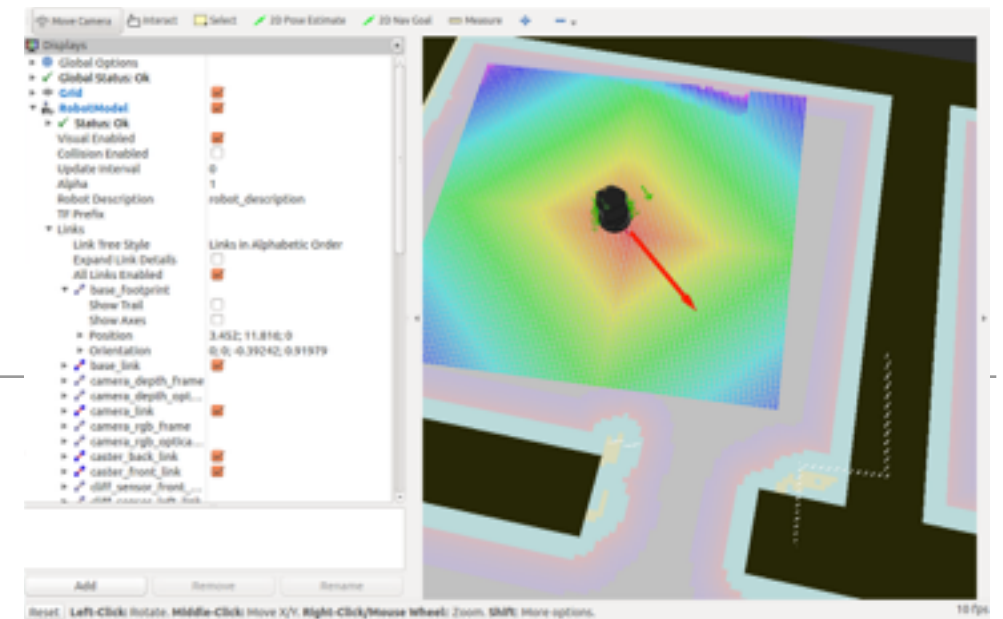


入門ハンズオンセミナー

ROS NavigationとSTDR Simulatorで学ぶROS

akio



本日の内容

- ・ 概要説明 (10分)
- ・ 開発環境を作ろう (20分)
- ・ rospy入門 (30分)
- ・ 開発してみよう (30分)

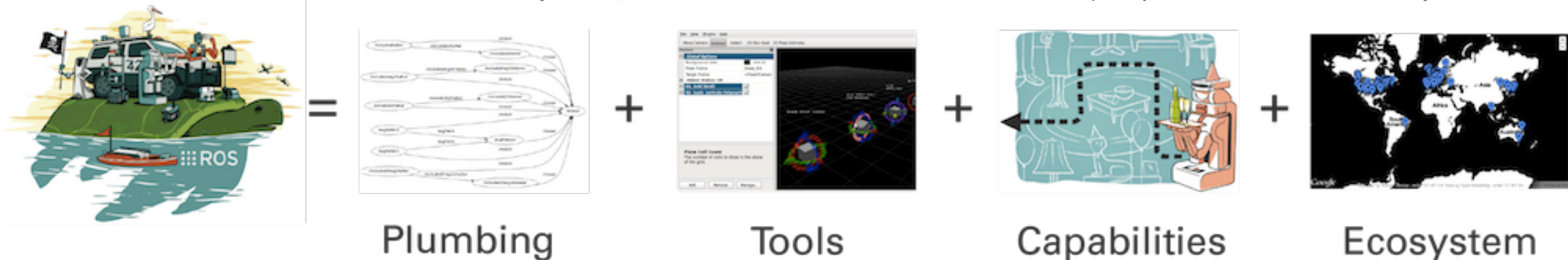
概要説明

本日の内容

- STDR(Simple Two Dimensional Robot) SimulatorとROS Navigationによる環境で、自律移動ロボットのプログラミングを体験します。
- ROSのツールやエコシステムの説明はなるべく省略します。詳しく知りたい方は公式サイトや各種ブログなどをお読みください。

ROSとは？

- ・ ロボットのソフトウェアを開発するための一つの大きなフレームワーク（枠組み）です。
- ・ 色々な構成要素で成り立っています。
 - ・ 通信ミドルウェア（クライアントライブラリ、ネームサーバ、etc...）
 - ・ 開発ツール（ビルド、可視化、ロギング、etc...）
 - ・ 高度な機能ライブラリ（自律移動、マニピュレーション、etc...）
 - ・ コミュニティインフラ（Wiki、Q&Aシステム、パッケージ配布システム、etc...）



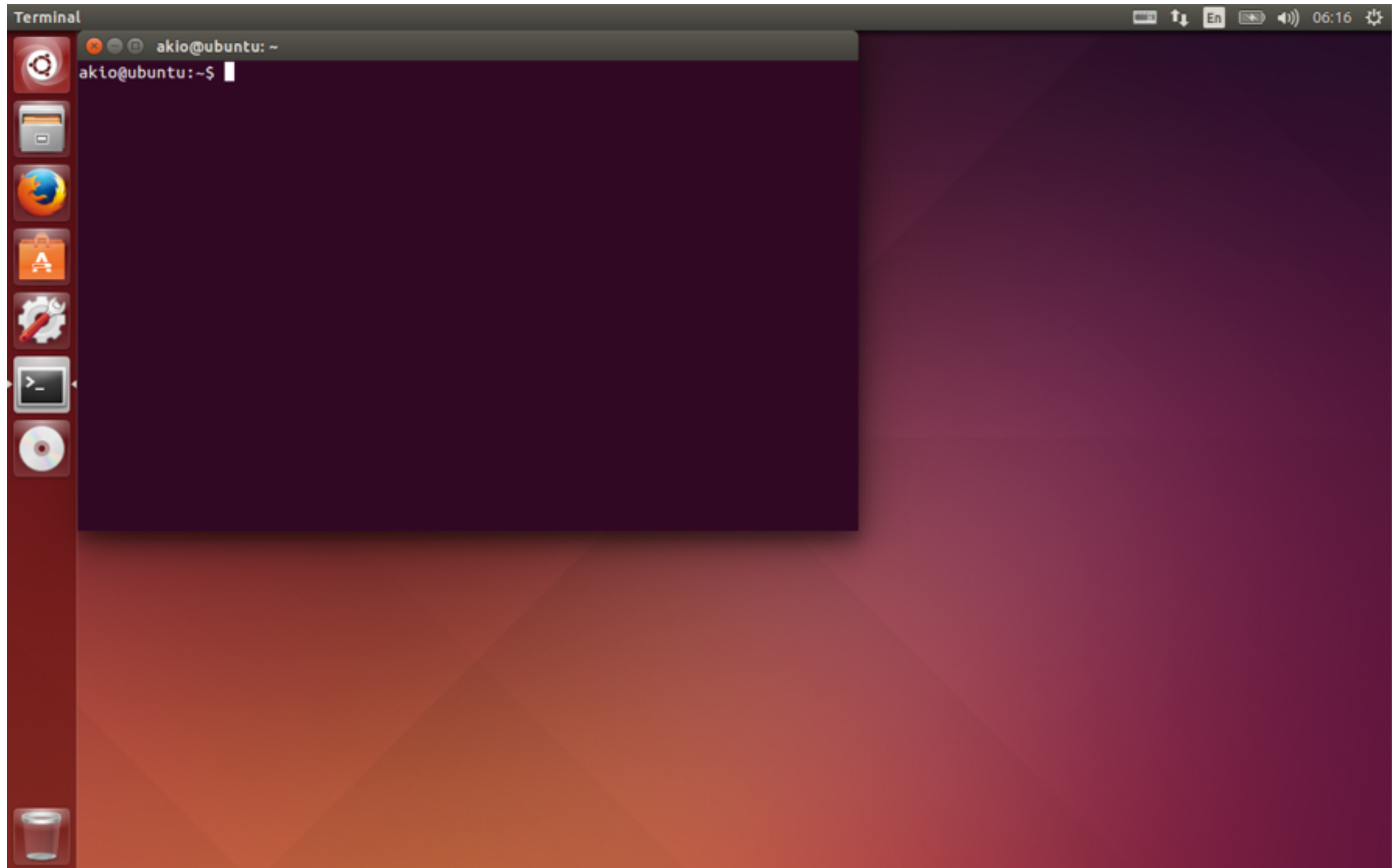
引用元：www.ros.org/about-ros

開発環境を作るう

仮想端末を開きます



基本的にこのウィンドウで作業します



ROSのインストール

- Ubuntu 14.04 (Trusty)

- 次のコマンドで、ROSのパッケージリポジトリを更新リストに追加します

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu trusty main" > /etc/apt/sources.list.d/ros-latest.list'
```

- パッケージ作成者確認用の設定を行います

```
wget https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -O - | sudo apt-key add -
```

- 依存関係DBの初期化

```
sudo rosdep init  
rosdep update
```

- （お好みで）ROS環境変数の自動読み込み設定

```
echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc  
source ~/.bashrc
```

- パッケージデータベースを更新します

```
sudo apt-get update
```

Ubuntuのパッケージをインストール

- 基本的に次のコマンドでインストールします

```
sudo apt-get install パッケージ名
```

- 今回のチュートリアルで使うパッケージ
 - `ros-indigo-stdr-simulator`
 - `ros-indigo-turtlebot-stdr`
 - `ros-indigo-turtlebot-navigation`
 - `git`

ワークスペースを作る

- Catkin (ROSのビルドツール) のワークスペースを作ります

```
source /opt/ros/indigo/setup.bash
mkdir -p seminar_ws/src
cd seminar_ws
catkin_init_workspace src
```

- 練習問題のコードを取ってきます

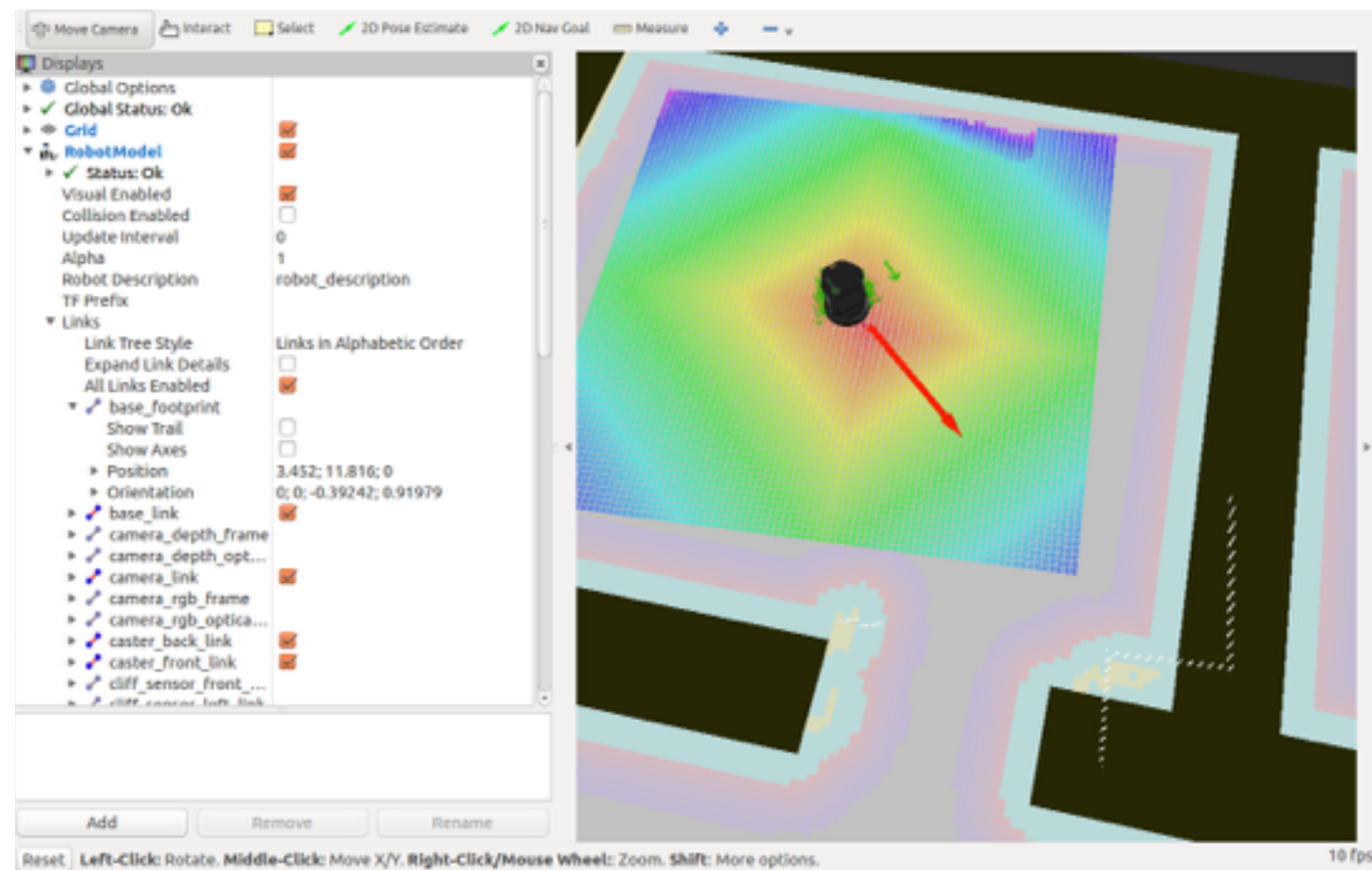
```
git clone https://github.com/akio/rosjug_seminar5.git
catkin_make -C ..
source ../devel/setup.bash
cd rosjug_seminar5
```

シミュレーション環境を起動してみよう

- rosjug_seminar5ディレクトリ直下で以下のコマンドを実行して下さい

```
roslaunch start.launch
```

- 右下のようなウィンドウが出現します。
- ゴールを設定して遊んでみましょう



rospy入門

rospy入門

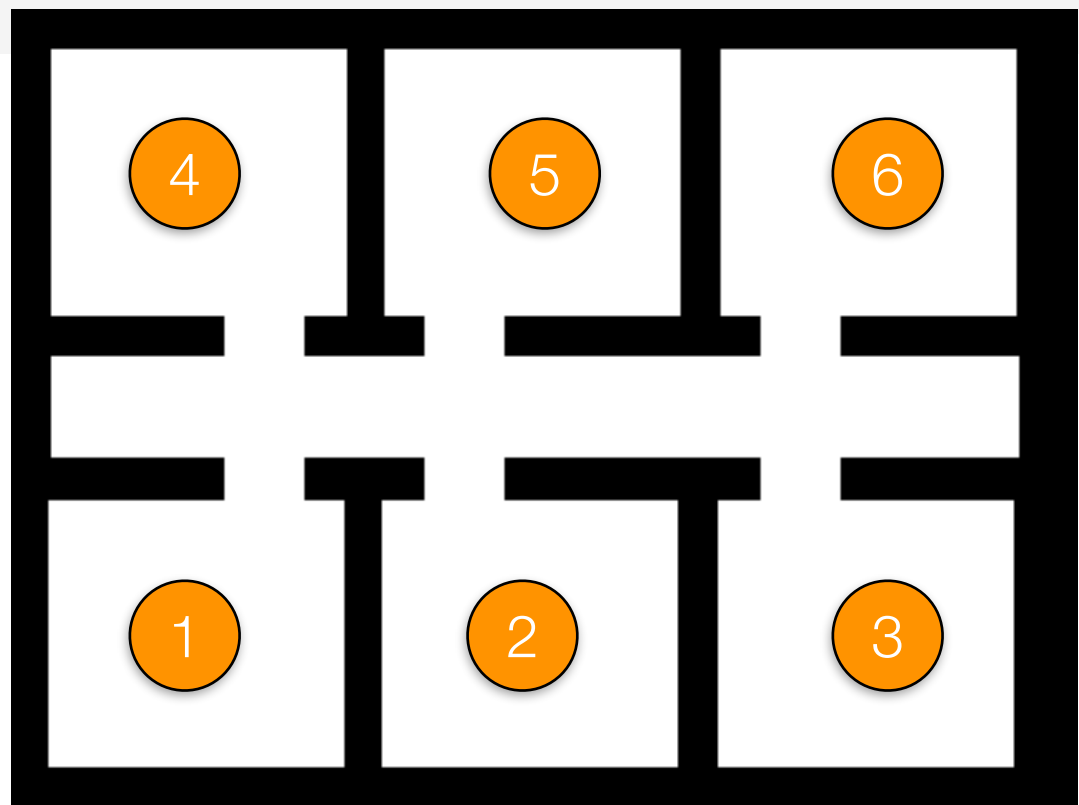
- Python
 - Python基礎文法最速マスター
- rospy
 - シンプルな配信者 (Publisher)と購読者 (Subscriber) を書く。

開発してみよう

課題説明

- exercise.pyを編集して、部屋を1から6まで順番に自律移動するプログラムを開発しましょう。
- 編集は好きなエディタなど。よくわからない人はgeditというコマンドでGUIエディタが起動します。
- start.launchが起動している状態で下記コマンドで実行できます。

```
python exercise.py
```



各部屋の大体中心の座標

- /amcl_poseトピック (geometry_msgs/PoseWithCovarianceStamped型) を読んで比較すると,ゴール判定できます。
- 部屋 1 : $x = 3.5, y = 3.5$
- 部屋 2 : $x = 9.5, y = 3.5$
- 部屋 3 : $x = 16.0, y = 3.5$
- 部屋 4 : $x = 3.5, y = 11.5$
- 部屋 5 : $x = 9.5, y = 11.5$
- 部屋 6 : $x = 16.0, y = 11.5$