



#### Unit 5: Rover センサ

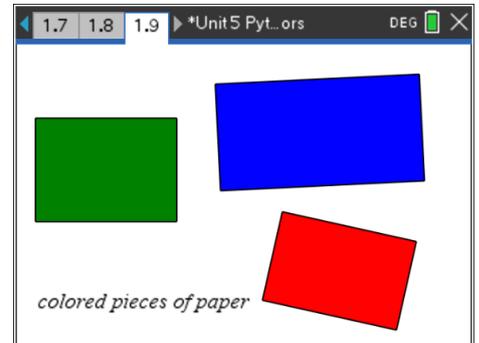
#### Skill Builder 3: 色を見つける

このレッスンでは、Roverのカラーセンサを使って、色が検出されたときに方向を変える方法を学びます。  
このレッスンでは、色紙または大きな色の形状物が必要です。

#### 目標

- カラーセンサで色を検出して反応

Roverの下部にはカラーセンサがあります。カラーセンサの下の床に光が輝いているのが見えます。光はRoverがその下の色を見るのを助けます。最初にtestプログラムを作成して、カラーセンサが生成する値の種類を確認します。つぎに、さまざまな色に反応するプログラムを作成します。画用紙のような色紙を用意するか、右図の長方形の色の形を印刷するだけです。それらは、Roverが「見る」のに十分な大きさが必要です。



menu > TI-Rover > Inputs (メニュー>TI-Rover>入力)では、5つの異なる色測定が利用可能です(右図参照)。関数color\_measurement()は、1から9までの値を返します。その値は、次のとおりです。

1=red(赤), 2=green(緑), 3=blue(青), 4=cyan(シアン), 5=magenta(マゼンタ),  
6=yellow(黄色), 7=black(黒), 8=white(白), 9 = gray(灰色)

他の4つの測定値は、メニューに示されているように0~255の範囲で、示された色の量を返します。

2	color_measurement()	1-9
3	red_measurement()	0-255
4	green_measurement()	0-255
5	blue_measurement()	0-255
6	gray_measurement()	0-255

1. これは、Rover Codingテンプレートを使って、色測定関数が生成する値を決定する短いtestプログラムです。

```
while get_key() != "esc":
    c = rv.color_measurement()
    plt.text_at(7, str(c), "left")
```

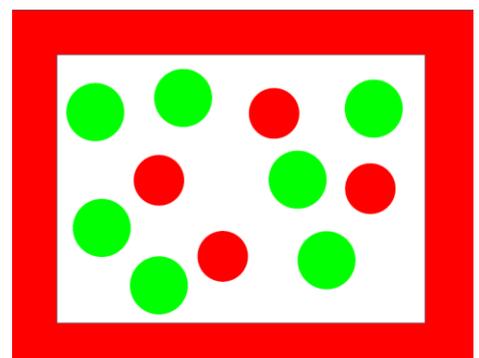
さまざまな色の表面で5つの色測定(color\_, red\_, green\_, blue\_, gray\_)をすべて試し、表示される値を観察します。

2. 床の色に応じてRoverが床からの色の変化を「見る」ことができるように、Roverの前にいくつかの色付きの紙(付箋やカラフルな画用紙など)を置きます。最初にこれらのパッチ(一部分)をテストして、Roverが「見る」ものを確認します。

Roverが色の違いに反応するプログラムを作成します。右図のサンプルでは、赤い境界線がRoverをページ上にあるようにします。

Roverが赤を「見る」とき、振り返ります。

Roverが緑色を「見る」ときは、右または左に曲がります。



プログラムにステートメントを追加して、ライトまたはカラーLEDを制御することもできます。



3. 開始したカラーtestプログラムを編集します。ループの先頭に次のステートメントを追加して、Roverの前進を開始します。

**rv.forward(10)**

つぎに、Roverの下の色を監視します。変数cをcolorにすると、プログラムが読みやすくなります。

**color = rv.color\_measurement()**

色が赤の場合はRoverを180°回転させ、色が緑の場合は90°回転させます。これには2つのifステートメントが必要です。

plt.text\_at(7, str(c), "left")を削除します。

```

1.5 1.6 1.7 *Unit5 Pyt...ors RAD 13/34
*u5sb3.py
from time import *
#-----
while get_key() != "esc":
    rv.forward(10) # one meter
    color = rv.color_measurement()
    |
    |
    |
  
```

4. 2つのifステートメントを追加し、?の代わりにcolorサンプルのテストから取得した値を使います。

**if color == ?:**

**block**

**if color == ?:**

**block**

(ifステートメントのプレースホルダー(仮に確保した場所)blockは表示されません。右図のredとgreenという単語は単なるプレースホルダーです。)

```

1.5 1.6 1.7 *Unit5 Pyt...ors RAD 17/36
*u5sb3.py
from time import *
#-----
while get_key() != "esc":
    rv.forward(10) # one meter
    color = rv.color_measurement()
    if color == red:
        block
    if color == green:
        block
  
```

5. 2つのifブロックはよく似ています。赤-右(red-right)と緑-左(green-left)を覚えておいてください。

各blockは次のようになります。

- Roverを停止
- 180°回転
- 少し前に進み、色付きのスポットから離れる
- Roverがこれらの3つのタスクを完了するまで待つようにTI-Nspire CX IIに指示

先に進む前に自分で試してみましょう。

```

1.5 1.6 1.7 *Unit5 Pyt...ors RAD 14/29
*u5sb3.py
if color==1:
    rv
    rv
    rv
    rv
    rv #get off the red spot
if color==2:
    rv
    rv
    rv
    rv
    rv #get off the green spot
  
```



6. ifブロックは、以下のものに似ていますか。

**if color == 1:** (赤。色が赤でないとき、別の数字を使ってください。)

**rv.stop()**

**rv.right(180)**

**rv.forward(1)**

**rv.wait\_until\_done()**

**Tip:** プログラム先頭で変数red = 1を割り当てると、ifステートメントにif color == red:と書くことができます。これにより、意図が明確になります。

```
*u5sb3.py 23/29
if color == 1:
    rv.stop()
    rv.right(180)
    rv.forward(1)
    rv.wait_until_done() #get off the red spot
if color == 2:
    rv.stop()
    rv.left(180)
    rv.forward(1)
    rv.wait_until_done() #get off the green spot
```