



# artisoc Cloudレシピブック

## 02. エージェント同士を相互作用させよう

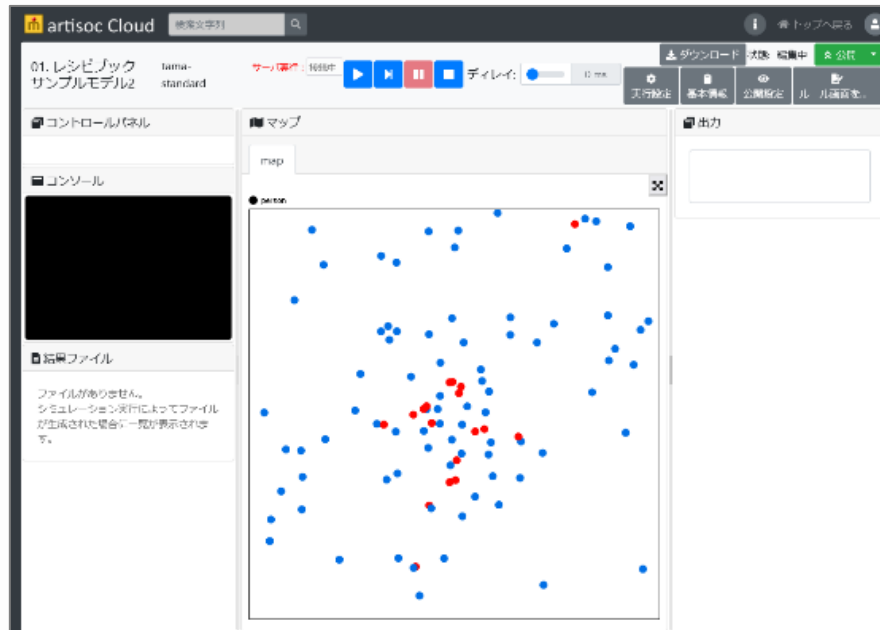
(株) 構造計画研究所

創造工学部

<https://mas.kke.co.jp>

01. レシピブックのサンプルモデル1を改造して、相互作用するモデルを作成します。下記のモデルを開き、継承して新規作成してください。

- a. ぶつかると赤くなるエージェント
- 周りを見回して隣人を探します。
  - ぶつかったら赤くします。



## 02. レシピブック サンプルモデルa

隣人（neighbor）を定義して、エージェント同士を相互作用させます

- モデルツリーの「person +」をクリックします。  
変数名： neighbor
- モデルツリーの「person」をクリックしてルールエディタを表示します。

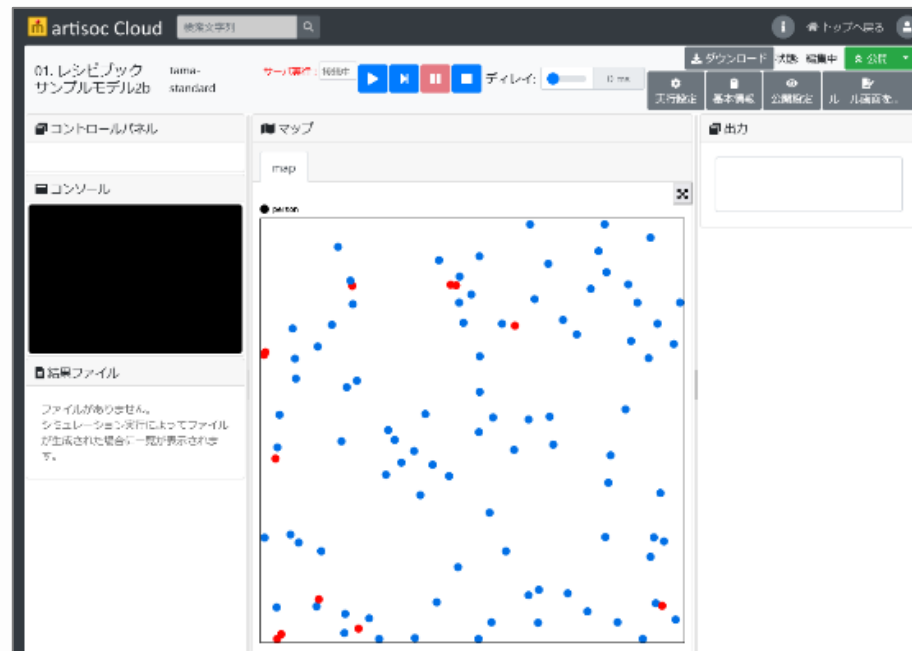
```
8 def agt_step(self):  
9     self.forward(self.speed)    # 前に進む  
10  
11     self.neighbor = self.make_agtset_around_own(1, False)    # 隣人を取得する  
12     if count_agtset(self.neighbor) > 0:    # 隣人の数を数えてぶつかったら赤にする  
13         self.turn(rand() * 360)  
14         self.color = COLOR_RED  
15     else:  
16         self.color = COLOR_BLUE
```

- シミュレーションを実行します。

下記のモデルを開き、継承して新規作成してください。

## b. ループしない空間とランダムな配置

- 空間がループしない場合は端点の処理が必要になります。
- エージェントをランダムに配置します。

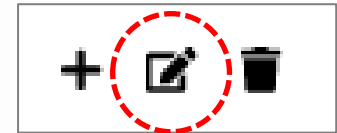


## 02. レシピブック サンプルモデルb

## b. ループしない空間とランダムな配置 (1)

### ループしない空間を定義します

- モデルツリーの「map」  をクリックします。  
ループする：      チェックをはずす



Add Edit Delete

- モデルツリーの「Universe」をクリックしてルールエディタを表示します。

```
1 def univ_init(self):  
2     create_agt(Universe.map.person, num=100) # personを生成  
3  
4     people = make_agtset(Universe.map.person) # personの集合を取得  
5     random_put_agtset(people) # personをランダムに配置
```

- モデルツリーの「person」をクリックしてルールエディタを表示します。

```
1 def agt_init(self):  
2     self.color = COLOR_BLUE # 表示色は青色  
3     self.speed = 1 # 歩行速度は1  
4     self.turn(rand() * 360) # 指定した角度だけ回転
```

壁にぶつかったときの処理を加えます

- モデルツリーの「person」をクリックしてルールエディタを表示します。

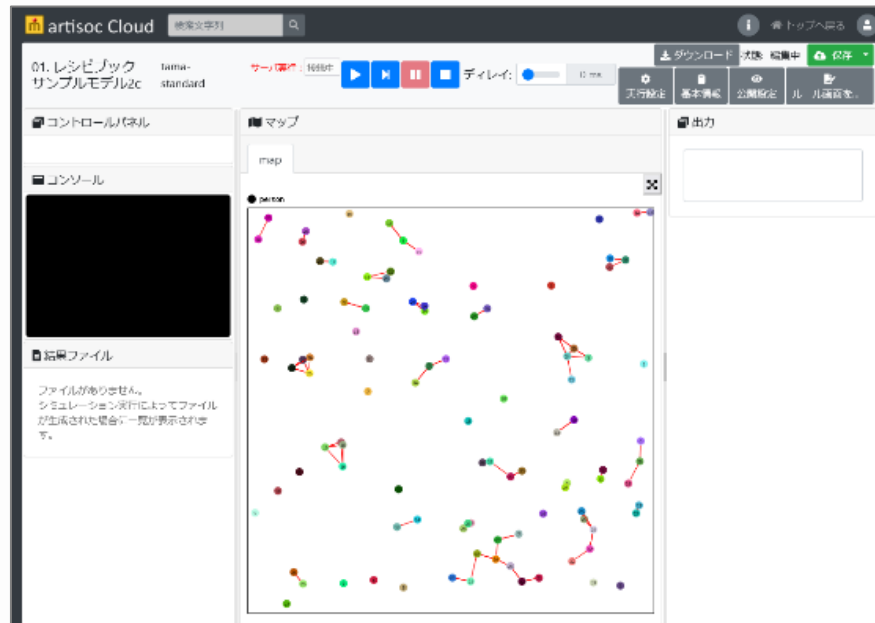
```
6 ▾ def agt_step(self):  
7 ▾     if self.forward(self.speed) != -1:      # 壁に当たったら向きを変える  
8 ▾         self.turn(rand() * 360)  
9 ▾  
10 ▾     self.neighbor = self.make_agtset_around_own(1, False)  # 隣人を取得する  
11 ▾     if count_agtset(self.neighbor) > 0:      # 隣人の数を数えてぶつかったら赤にする  
12 ▾         self.turn(rand() * 360)  
13 ▾         self.color = COLOR_RED  
14 ▾     else:  
15 ▾         self.color = COLOR_BLUE  
16 ▾
```

- シミュレーションを実行します。

下記のモデルを開き、継承して新規作成してください。

## c. エージェントに個性を持たせる

- 違う色、違う歩行速度を持つエージェントを定義します。
- 近くにいるエージェント同士を線で結びます。



## 02. レシピブック サンプルモデルc

## c. エージェントに個性を持たせる（1）

エージェントは異なった色「color」と歩行速度「speed」を持ち、接触回数「touch」をルールに組み入れます



- モデルツリーの「person +」をクリックします。  
変数名： touch
- モデルツリーの「person」をクリックしてルールエディタを表示します。

```
1 def agt_init(self):
2     self.color = rgb(randint(0,255), randint(0,255), randint(0,255)) # ランダムに色を指定
3     self.speed = rand() # 歩行速度をランダムに定義
4     self.turn(rand() * 360) # 指定した角度だけ回転
5
6 def agt_step(self):
7     if self.forward(self.speed) != -1: # 壁に当たったら向きを変える
8         self.turn(rand() * 360)
9
10    self.neighbor = self.make_agtset_around_own(3, False) # 視野を3にする
11    if count_agtset(self.neighbor) > 0:
12        self.touch += 1 # 接触回数を加算
13        self.turn(rand() * 360)
```



## c. エージェントに個性を持たせる（2）

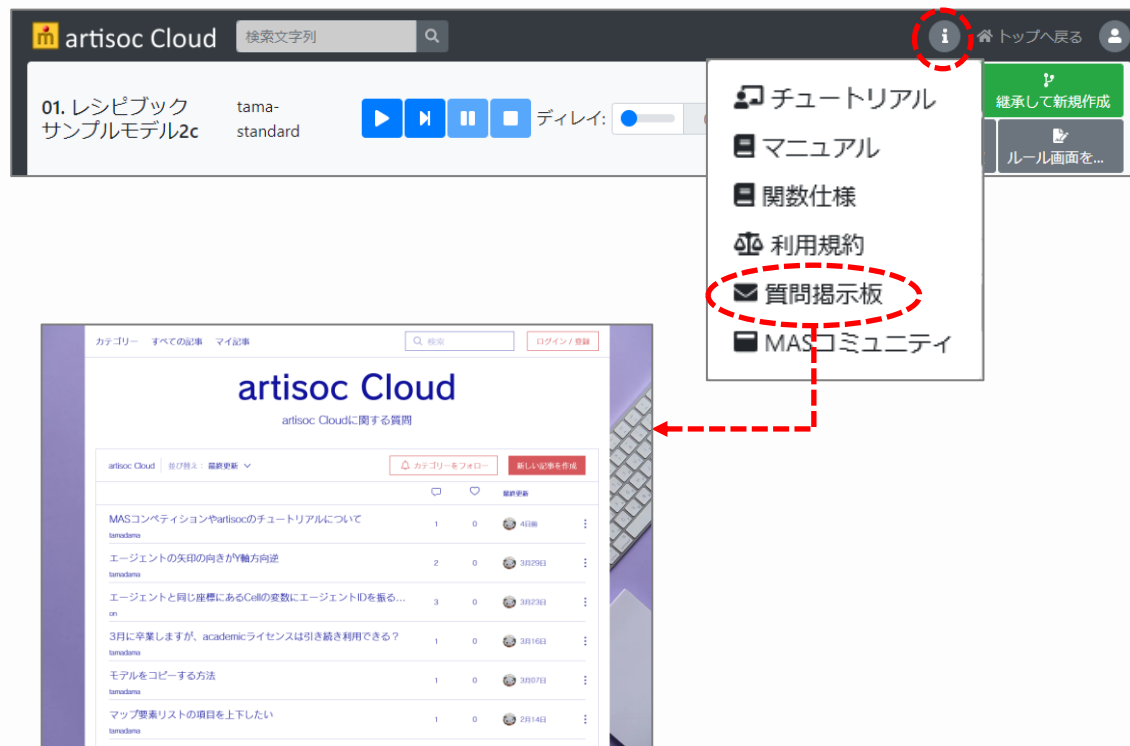
### マップ出力を設定します

- マップ出力の「map」 をクリックします。
- マップ要素リストの「person」 をクリックします。  
[エージェント情報の表示]  
表示する変数： touch  
[エージェント間に線を引く]  
対象の変数： neighbor  
線の色： 255, 0, 0
- シミュレーションを実行します。



# [Tips] レシピブックに質問があるとき

- 右上の i アイコンを選択し、「質問掲示板」をクリックすると質問ができます。



## [Tips] 線を引く相手を選ぶとき

- p.8で周りにいるエージェントを self.make\_agtset\_around\_own で取得して self.neighbor に代入しました。

```
6 def agt_step(self):
7     if self.forward(self.speed) != -1:      # 壁に当たったら向きを変える
8         self.turn(rand() * 360)
9
10    self.neighbor = self.make_agtset_around_own(3, False)  # 視野を3にする
11    if count_agtset(self.neighbor) > 0:
12    self.touch += 1      # 接触回数を加算
```

- 特定の条件のエージェントとのみ線を引きたい場合は、下記の通り、set()で初期化して、addでエージェントを代入すると、相手を選んで線を引くことができます。

```
6 def agt_step(self):
7     if self.forward(self.speed) != -1:      # 壁に当たったら向きを変える
8         self.turn(rand() * 360)
9
10    persons = self.make_agtset_around_own(3, False)  # 視野を3にする
11    if count_agtset(persons) > 0:
12        one_person = randchoice(persons)      # 隣人をランダムで選択する
13        self.neighbor = set()
14        self.neighbor.add(one_person)
15        self.touch += 1      # 接触回数を加算
16    self.turn(rand() * 360)
```

## 02. レシピブック サンプルモデルd