

担当 山影進

TA 阪本拓人、鈴木一敏、保城広至、
光辻克馬、山本和也

第九回 状態・属性をもたせる; 他者の選び出しと働きかけ(6月21日)

今日の目的:

今回は、エージェントの性質・状態などによって、自分の色を動的に変化させることを学びます。さらに、性質や状態によって、相手を個別に選別して作用する方法を学びます。(前回選別の仕方を学びましたが、距離や種類でしか選択できませんでした。今回は相手のスピードや状態などもっと細かな要因を一匹ずつ判断して選別します)

変数で色を変える

変数は様々なことを表すことができます。スピード、所持金、生死、種類(ヒトエージェントの雄と雌)、バックマンが無敵状態かどうか etc... そうなると、

- ・ 値が多いほど色を濃くしたい
- ・ 種類によって色を変えたい

といった場合も出てくることでしょう。こんなときには、エージェント直下に整数型の変数を作りましょう。その変数の中に色を指定する値を入れるわけです。たとえば、iro という変数を作って、my.iro = Color_Black などとすれば、勝手に黒になるような数値を入れてくれます。そのうえで、出力設定のマップ要素設定の「エージェント表示色」のところで、「変数指定」を選択し、my.iro を指定してあげれば、その変数の中に入った数値に基づいた色で表示できます。このように指定できる色には、

Color_Red 赤
Color_Green 緑
Color_Blue 青
Color_Yellow 黄
Color_Cyan 水色
Color_Mazenta 紫
Color_Black 黒
Color_White 白

があります。もし、もっと細かな色を指定したければ、RGB()という関数を使います。引数は、2つ目が赤、2つ目が緑、3つ目が青の成分を表しており、それぞれ0から255までの

整数で指定します。たとえば、

```
my.iro = RGB(0, 0, 255)
```

と指定してあげれば青色、RGB (255, 255, 255)であれば白、RGB (0, 0, 0)であれば黒が出ます。

例題 :

ループした空間上の中央にエージェントを 1000 ほど配置し、ランダムな方向に 1 ずつ進む様に見てみましょう。画面の右端では青っぽく、左端では黒っぽくなるように色を付けてみましょう。

- ・ まず、実効環境設定で、ガベージコレクションを 1 に設定してください。
- ・ 空間とエージェントを作成しましょう。もう細かな説明が無くても出来ますね。
- ・ エージェントルールには以下のように書きます。別の書き方をしても当然 OK です。

```
Agt_Init{
```

```
  movetocenter()
```

空間の真中に移動します。

```
  my.Direction = 360*rnd()
```

ランダムな方向を向かせます。

```
}
```

```
Agt_Step{
```

```
  forward(1)
```

1 だけ進みます。

```
  my.iro = RGB(0,0,cint(my.X*5))
```

0,0,0 が黒です。0,0,255 は青です。空間は 50×50 ですから X の取りうる値は 0 以上 49 未満、したがって、my.X*5 は 0 以上 245 未満の実数です。cint()というの
は変数を整数型に変えるための関数です。5.5 なら 5 と
いうふうに切り捨てられます。この場合は無くても動
きますが、こう書いた方が丁寧です。

```
}
```

もし、「右半分にいたら青、左半分にいたら黒」とするのであれば、最後の一行の代わりに

```
if my.X < 25 then
```

```
  my.iro = color_black
```

```
else
```

```
  my.iro = color_blue
```

```
end if
```

と書きましょう。動かない場合は、出力設定がきちんと出来ているか確認してください。

一つ一つの相手に作用する

前は集合同士を足し引きして操作しました。今回は一つ一つのエージェントをよく調べて選別・作用します。そのためには個別エージェントを一つ入れる「エージェント型」という変数の型を使います。ツリーで作るときにはプロパティで「エージェント型」を選びます。一時的な変数として宣言するときには、ルールの先頭で `dim aite as agt` といった風に入ります。すると、たとえば「`aite.x`」と書くことで、変数の中に入ったエージェントの `x` 値を表すことができます。例題を見てみましょう。

例題

ループした空間にヒトエージェント(赤)を 100 個ほどランダムに配置し、ランダムな方向に 1 ずつ歩く様にしてください。そのうち一匹だけをキョンシー(黒)にしましょう。キョンシーは 2 歩ずつ飛んで進み、ヒトに接触すると噛み付きます。噛み付かれたヒトはキョンシーになってしまいます。

- ・ 空間とヒトエージェントを作ってください。ヒトエージェントのルールには、以下のように入ります。(あくまで一例です)

```
Agt_Init{
```

```
my.X = rnd()*50           X 値を指定します。0 以上 50 未満の実数です。
```

```
my.Y = rnd()*50           同様に Y 値を指定します。
```

```
my.Direction = 360*rnd()  Direction を指定します。
```

```
if my.id == 0 then        id は 0 から振られるので、一人だけキョンシーになります。
```

```
    my.iro = color_black  黒はキョンシーを表します。
```

```
else
```

```
    my.iro = color_red    残りは普通のヒトです。
```

```
end if
```

```
}
```

```
Agt_Step{
```

```
dim aite as agt           これがエージェント型変数です。
```

```
dim mawari as agtset      周囲を観察するのでエージェント集合型も一つ要ります。
```

```
if my.iro == color_black then  もし自分がキョンシーだったら...
```

```
    forward(2)            二歩進みます。
```

```
    makeallagtsetaroundown(mawari, 1, false)  周囲のヒトを見回します。
```

```
    for each aite in mawari  mawari 中にある aite 一人一人に対して
```

```

        if aite.iro == color_red then      もし aite が普通の人ならば
            aite.iro = color_black        (噛み付いて)キョンシーにします
        end if
    next aite                             次の aite に対して上記ルールを繰り返します
else                                       もし自分がキョンシーでなければ・・・
    forward(1)                            一歩だけ進みます。
end if
}

```

* for each 文

エージェント集合の中のエージェント一つ一つに作用したい、選別したい、と言うときには for each 文を使うと便利です。

```

    for each 「エージェント型変数」 in 「エージェント集合型変数」
        命令したい内容
    next 「エージェント型変数」

```

と書くと、一つ一つのエージェントに対してルールを実行できます。

* 上記の「命令したい内容」で、相手に作用する方法はいくつかあります。たとえば、

- ・ 「aite.x = 10」のようにエージェント型変数に入っている相手の変数をいじる
- ・ killagt(aite)として相手を抹殺する
- ・ addagt(nakama, aite) として別のエージェント集合型変数に追加する

などです。一番下のものと if 文を使えば、キョンシーだけ、お金持ちだけ、スピードが速い車だけなど、距離やエージェント種よりもっと細かな条件に基づいてエージェント集合を作ることも出来るでしょう。

* (上級)(おまけ)ランダムに相手を選びたいときには・・・

```

    dim hito as agtset
    dim ninzuu as integer
    dim aite as agt

    makeagtset(hito, universe.syakai.ningen)
    ninzuu = countagtset(hito)
    aite = getagt(hito, cint(ninzuu*rnd()))

```

とします。この場合、ningen エージェントを hito という集合にリストアップし、getagt()

によってその `cint(ninzuu*rnd())`番目の `ningen` を `aite` の中に格納します。対象の中の一人だけに作用したい場合に便利です。 `getagt()`の二つ目の引数に、リストよりも大きい数を指定するとエラーになります。たとえば、3 匹しかいないときには、リストは 0,1,2 までです。3 や 4 が入ってしまわないよう気を付けましょう。

課題:

空間上にエージェントを敷き詰めてください。そのエージェントの色を、`RGB()`で指定しましょう。そして、RGB それぞれの成分をコントロールパネルで操作できるようにしてみましょう。思った色を出すのは意外に難しいと思います。出来たファイルは名前を付けて保存しておきましょう。

キョンシーモデルに導師を 3 人追加しましょう。導師はヒトと同じように動き、キョンシーを見つけると御札を張ります。すると、キョンシーは黄色くなって動かなくなります。

以下上級

キョンシーが一度に一人しか噛めないようにしてみましょう。作用する相手はランダムに選びます。

各自仕様を考えて、風邪が流行するモデルを作ってみましょう。感染者が別の人に接触するとうつり、しばらくすると自然に治ります。治癒にかかる期間、接触した際の感染の確率なども設定できるといいかもしれません。