世界モデル実習（SR）・人間の安全保障実験実習Ⅲ（GR）　　2005 年度夏学期

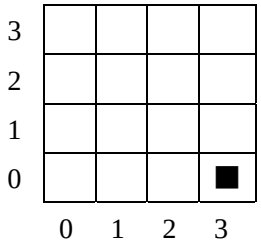
担当　山影進
TA　阪本拓人、鈴木一敏、保城広至、
光辻克馬、山本和也

第 10 回　空間に痕跡を残す; 実行順序(6 月 28 日)

今日の目的:
今回は空間に痕跡を残す方法を学びます。また、正確なシミュレーションのため、実行順序について少しだけ厳密に考えてみます。

● 空間に痕跡を残す

地面にフェロモンを残したり、砂をつんで山を作ったり、海面に航跡を残したり。「場」に属するような変数を作りたい場合はどうしたらよいでしょうか？ 空間に痕跡を残すには、ツリーで空間の直下に作る変数、「空間変数」を使います。この種の変数では、値は x、y、layer の整数値ごとに保存されます。たとえば、以下のような空間であれば、



各階層(レイヤー)について 16 のマス目ごとに、値を保存することができます。ルールから個々の値を指定するには Universe.Kuukan.hensuu(x, y, layer) のように書きます。上記の空間の■の場所の値だけを 5 にしたいのであれば、

Universe.Kuukan.hensuu(3, 0, 0) = 5

と書きます。

例題①

納屋の机の上にたくさんのみかんが置いてあります。幾つかは初めから腐っています。腐ったみかんからは周囲 1 近傍全部に、毎回 1 のカビ胞子が飛びます。自分の所に胞子が 5 個たまると、腐っていないみかんも腐ってしまいます。

回答例:

・51×51 の机を作ってください。机はループしません。みかんも作りましょう。みかんの数はコントロールパネルで決定します。

Universe に MikanSousuu(整数型)を作ったうえで、

```
Univ_Init{
dim i as integer
for i = 0 to universe.MikanSousuu – 1
        createagt(universe.TUKUE.Mikan)
next i
}
```

などとすると良いでしょう。

・次に、みかんを机の中央からバラバラに置いておきましょう。ただし、落ちるとイヤなので、端から 5 マスには置きません。

・一つ一つのみかんは、一定の確率で初めから腐っています。この確率はコントロールパネルで設定できるようにしましょう。

```
Agt_Init{
        Movetocenter()                  ←机の中心に移動します
        Movetospaceowncell(20)          ←そこから上下左右 20 歩以内の
                                        空き地にランダムに転がります
        if rnd() < universe.RRate then  ← RRate が腐敗率(0－1)です
                my.fuhai = true         ← 腐敗してます
                my.iro = RGB(150, 70, 0) ← 腐った色はこんな感じです
        else
                my.fuhai = false        ← 腐敗してません
                my.iro = RGB(255, 170, 0)←みかんの色はこんなもんでしょう
        end if
}
```

・腐ったみかんは、毎回周囲 1 の範囲に胞子を 1 とばします。腐っていないみかんは、自分の所に胞子が 5 以上あると、腐ります。

```
Agt_Step{
        dim i as integer
        dim j as integer

        if my.fuhai = = true then          ← 腐っていたら
                for i = 0 to 2             ← i が 0,1,2 まで
                        for j = 0 to 2     ← j が 0,1,2 まで
                                universe.TUKUE.Houshi(my.X+i-1, my.Y+j-1, 0) =
universe.TUKUE.Houshi(my.X+i-1, my.Y+j-1, 0) + 1
                        next j             ↑ i±1 は、-1,0,1 です。9 カ所で+1 します
                next i
        else                               ← 腐っていなかったら
                if universe.TUKUE.Houshi(my.X, my.Y, 0) >= 5 then    ←5 以上あると
                        my.fuhai = true                ←腐ります
                        my.iro = RGB(150, 70, 0)       ←腐るとこんな色
                end if
        end if
}
```

・出力設定も忘れずに。胞子も出力できます。胞子の色が 0 から 5 で色が変化するようにしてみましょう。

＊ 入れ子の For 文

上記のルールでは、For 文が二段階になっており、3×3 で計 9 回胞子をばらまく作業が行われます。その時の i と j の値を用いて、自分の x の±1、y の±1 に、胞子を 1 プラスしているわけです。このように、for 文は多くのマスの空間変数を一斉に変えたい時に便利です。たとえば、雪の降るモデルで、すべてのマスの雪を 1 にしたい場合には、

```
dim i as integer
dim j as integer

for i = 0 to getwidthspace(universe.jimen) - 1
        for j = 0 to getheightspace(universe.jimen) - 1
                Universe.jimen.yuki(i, j, 0) = 1
        next j
next i
```

と書けば良いわけです。Universe の init に書けば、一番はじめに一回だけ、Step_begin に書けば、毎ステップ開始時に 1 になります。

● 実行順序

ところで、実は今作った腐ったみかんモデルには問題があります。ここでは単純化のため、一つの胞子でみかんが腐ると考えてみてください。a,b,c,d4 つのみかんがくっついて並んでおり、以下のような順序で実行されたとしましょう。

① ③	a c
② ④	b d

すると、

1: みかん a	胞子は一つもない
2: みかん b	胞子は一つもない
3: みかん c(腐)	胞子を飛ばす
4: みかん d	胞子があるので腐る

という動きをしてしまいます。本来ならば、c の出す胞子は a や b にも影響しなければ変です。今ステップに出した胞子が次のステップで影響するようにすれば、以下のようにしてこの問題を回避できるでしょう。

ステップ 1　腐ったみかん c が胞子を飛ばす
↓
ステップ 2　全てのみかんが前回の最後を見て、胞子があったら腐る(a,b,d が腐る)
腐ったミカン a,b,c,d が胞子を飛ばす
↓
ステップ 3　全てのみかんが前回の最後を見て･･･

こんなときには、Gethistory()を用いて、エージェントが前のステップの最後の状況を見て判断し行動するようにします。過去の変数の値を参照したいときには、まず、変数のプロパティで値を記憶するように設定しなければなりません。プロパティを開け「記憶数」で何ステップ前まで記憶するか指定してください。それを呼び出すためには Gethistory(変数名, 履歴番号)を使います。たとえば、1 ステップ前の最後の aite の x 座標を参照したければ、Gethistory(aite.x, 1)と書きます。今回の場合は、自分が腐るかどうかを判断する部分

を、

```
if gethistory(universe.TUKUE.Houshi(my.X, my.Y, 0), 1)>= 5    then
```

と書き直してあげれば良いわけです。

課題①

腐ったみかんモデルを改造して、周囲 1 近傍には 2、周囲 2 近傍には 1 の胞子をばらまくようにしてみましょう。ヒント: 1 + 1 = 2

課題②

雪に跡を付けながらソリが進むモデル、もしくは、船が航跡を残しながら進むモデルを作ってみましょう。走った跡はしばらくしたら消えます。

課題③

自分で森林火災モデルを作ってみましょう。

・ ランダムに木エージェントが生えています。木の数はコントロールパネルで設定できます。

・ 周囲一近傍の木が一つでも燃えていたら燃え移ります。

・ 実行順序に気を付けてください。