世界モデル実習（SR）・人間の安全保障実験実習Ⅲ（GR）　　2006 年度夏学期

第五回 神の立場からシミュレーションの世界を創る

前回の課題のポイント

① 1 以内に同じ方向を向いた人がいたら、0.5 ずつ進む

```
MakeOneAgtsetAroundOwn(syuui, 1, universe.concourse.eastward, false)
a = countagtset(syuui)
if a > 0 then
        forward(0.5)
else
        forward(1)
end if
```

② 前方 1 から半径 1 の範囲を見る

```
forward(1)
MakeOneAgtsetAroundOwn(syuui, 1, universe.concourse.eastward, false)
a = countagtset(syuui)
forward(-1)
```

今回は、分居モデルを作りましょう。シェリングの分居モデルでは、チェッカー盤にダイムとペニーをランダムに配置し、それぞれが周りの 8 マスの周囲を調べます。同種のコインが 1/3 いない場合、コインは空き地に引っ越してしまいます。

Universe のルールと実行順序

Universe のルールを使って、シミュレーションの世界の初期設定を行う方法を学びましょう。Universe のルールエディタを開いてみてください。

Univ_Init{}	シミュレーションの最初に一度だけ実行される
Univ_Step_Begin{}	各ステップの初めに実行される
Univ_Step_End{}	各ステップの終わりに実行される
Univ_Finish{}	シミュレーションが終了するときに一度だけ実行される

という四つの括弧が見えるはずです。今回は初期設定をしたいので、Univ_Init を使います。

エージェントの創造

エージェント種をつくる時に、プロパティでエージェントの数を設定する事ができました。これは最初からいるエージェントの数です。artisoc では、ルールの中でエージェントを 1 体 2 体と生み出す事もできます。(エージェント種でなく、エージェントをつくり出すこと

に注意!!) そのためには、createagt()という関数を使います。たとえば tori モデルの tori を創りたければ、

```
CreateAgt(universe.oozora.tori)
```

と書けば良いだけです。これで tori が 1 羽追加されます。Univ_Init で 1 回このように書けば、初期設定+1 の tori をつくる事ができます。

それでは、bunkyo という名前でモデルを作成してください。city という８×８のループした空間を作り、その下に red, blue のふたつのエージェント種を作成しましょう。エージェント数はともに 0 にしておいて、Univ_Init で赤青一匹ずつ作ってみましょう。

ルールを繰り返す

一匹ずつ作るだけなら、createagt()を 2 回書くだけで済みますが、何十、何百匹と作るときにはちょっと困ります。そこで、書いたルールを繰り返す方法を学びましょう。それには for next 文と呼ばれるものを使います。

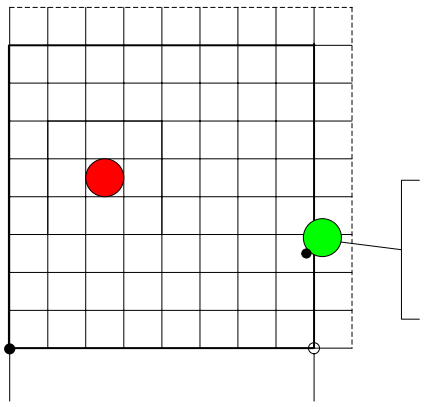
```
dim i as integer

for i = 0 to 19
        CreateAgt(universe.city.blue)
        CreateAgt(universe.city.red)
next i
```

こんな風に書くと、i に 0 から整数を順番に入れていって、19 になるまでその下のルールを繰り返してくれます。つまり、20 体づつ、合計 40 体のエージェントが出来るわけです。 i に当たるところは整数型の変数であればなんでも構いません。

セル型の空間の特徴

では、出力設定をあけて、罫線を表示、チェス型を選んでから、実行を押してみてください。このように空間にマス状の格子があってそのマスに入る形でエージェントが存在する（移動する）モデルの空間表現をセル型と呼びます。オセロやチェスを思い浮かべてください。

（１）座標は整数のみ。
（２）さらに、座標は０から数える。
（３）近傍の形が違う。
（４）出力範囲に注意！！



セル型空間では、整数ずつ動かなければなりません。それには Cell が付いた関数群を用

いて、あたかもマスがあるように動かします。たとえば、Forward()を使ってしまうと、たとえ 90 度や 0 度を向いていても、わずかな誤差が生じてしまいます(実数を使うので誤差が出ます)。そこで ForwardXCell と ForwardYCell というルールを用います。

ForwardXCell(整数) : X 軸方向に○マス進む

ForwardYCell(整数) : Y 軸方向に○マス進む

MoveToSpaceOwnCell(整数) : 自分の空間の空きマスに移動。数字は視野の広さ

試しに Ags_Step にかきこんで、X 方向、Y 方向に動かしてみましょう。

ばらまく

折角沢山のエージェントを作ったのに、重なっていては何だか分かりません。Universe のルールを使って空間上にばらまいてみましょう。ばらまくには、RandomputAgtset()や RandomputAgtsetCell()という関数を使います。とりあえず、上のルールでエージェントを作ったあとに、

MakeAgtsetSpace(set, universe.city)

RandomputAgtset(set)

と書いてください。ルールの頭に dim set as Agtset と宣言文を入れるのを忘れずに。エージェント集合型変数にリストされているエージェント(つまり空間上の全てのエージェント)を、その空間上にばらまきます。実行してみてください。

ところで、これではチェッカー盤になりません。きちんとマス目に、重なり無く納めなければなりません。Randomput のところを、

RandomputAgtsetCell(set, false)

と書き直す必要があります。最後 false は重なりを許さない事を意味しています。重なってもよい場合は true と書きます。では、ここまで作って一度実行してみてください。出力設定を修正しましょう。

コントロールパネルで作る数を指定する

エージェントを沢山つくるためには、何体つくるかを指示する必要があります。上の例では 0 to 19 で 20 体と指定してあります。数を変えるたびに毎回ルールエディタを開いて書き換えるのは面倒です。そこで、コントロールパネルで設定できるようにしましょう。

① Universe レベルにエージェントの数を指定するための変数を作ります。たとえば N としましょう。

② Universe のルールの For 文の 19 と書かれた部分をその変数に書換えます。ただし、universe.N - １と入れる必要があります。

③ 「設定＞コントロールパネル設定」から今作った変数を指定して、スライドバーを追加します。赤青それぞれ 32 で一杯になってしまいますから、値の範囲

は 0 から 32 にしておきましょう。

エージェントのルールを書く

・ 同種の数を数える
・ 異種の数を数える
・ 周囲 8 マスの隣人の中の同種の割合を計算
・ 同種の割合が 1/3 未満なら移動

```
Agt_Step{
dim R as integer
dim B as integer
dim set as agtset
dim rate as double

MakeOneAgtsetAroundOwnCell(set, 1, universe.city.blue, false)
B = CountAgtset(set)

MakeOneAgtsetAroundOwnCell(set, 1, universe.city.red, false)
R = CountAgtset(set)

if R + B = = 0 then
        rate = 0
else
        rate = B / (R+B)
end if

if rate < 1/3 then
        MovetoSpaceOwnCell(2)
end if
}
```

課題

・ もっと広い空間にしてみましょう。それに併せてエージェント数の上限も変えます。
・ red、blue の数をコントロールパネルで別々に設定できるようにしてみましょう。
・ 今まで不満を感じる閾値が 1/3 (0.333...)で固定でした。これもコントロールパネルで 0〜1 まで 0.1 単位で設定できるようにしましょう。