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

第七回 他者に働きかける、選び出す

●前回までの復習と今回の趣旨

○前回までの５回で、マルチエージェントシミュレーション（MAS）のモデルの基本的な大枠はマスターしました。エージェントが周囲の情報に基づいて判断し、他者と相互作用して、そこからエージェント間に相互関係（社会関係）が形成されていく、というのが MAS の基本的なながれです。

○今回からは、より複雑なエージェントの判断、相互作用、相互関係を表現するために、ルールの幅を広げていきます。

○これまで学んだ相互作用の主たるものは「周りにいる他者を数える」というものでした。今回は、エージェントに、もう少し「深い」相互作用もさせてみましょう。

○今回は、『交差点の出会い』というモデルを作ります。交差点で出会う男女（と犬）のモデルです。（モデルを作りながら必要に応じて文法事項を説明していきます）

★エージェント集合やエージェントの操作

★集合をチェックする（ForEach 文）

●［第一工程］モデルの枠組みを作る

［01］空間を作ります（名称は Downtown、設定はデフォルトのままで良いです）

［02］エージェントを２種類作ります（Male を 20 人、Female を 20 人）

［03］エージェントに初期設定を与えます（男女ともに）。座標と方角をランダムに与えましょう。

```
My.X = rnd()*50          ……確認ですが、Agt_Init のルールに書きます。
My.Y = rnd()*50
My.Direction = rnd()*360
```

［04］出力設定もしておきましょう。デフォルトのままで大丈夫です。お好みがあれば、色など調節してみましょう。

●［第二工程］男性と女性に行動ルールを与えます

［05］男性は、周囲（視界は 3）を見回し、女性がいれば立ち止まり、居なければぶらぶら歩くというルールを書いてみましょう。

［06］男性エージェントに、その男性のターゲットをリストアップしておくための変数をつくります。（名称 Target、エージェント集合型）

［07］ 周囲を見回す＝MakeOneAgtsetAroundOwn(Agt 集合, 視界, エージェント種, 含不含)
周りを数える＝CountAgtset(Agt 集合型変数)
ぶらぶら歩く＝ランダムに左右に各 10 度の範囲で方向を変えながら、1 ずつ進む。

［08］毎回、冒頭に、ターゲットをからっぽ（初期化する）ルールを加えておいて下さい。

```
ClearAgtset(My.Targets)                          ……まず Targets のなかみを空にする
MakeOneAgtsetAroundOwn(My.Targets, 3, Universe.Downtown.Female, False)
                                                 ……周りの女性をリストアップ
If CountAgtset(My.Targets) == 0    then          ……Target が空なら
        Turn(rnd()*20-10)                        ……ぶらぶら歩く
        Forward(1)
End if
```

［09］女性にもぶらぶら歩くルールを与えましょう。

```
Turn(rnd()*20-10)
Forward(1)
```

●［第三工程］男性の視線を表示してみましょう

［10］設定＞出力設定＞出力項目リスト＞編集＞マップ要素リスト（男性を選択）＞編集＞「線をひく」を選択する。線引対象はターゲットをあらわす変数を選択します。（線種や種別は自由に選んでください）

［11］（保存してから実行してみてください）女性に気をとられて立ち止まる男性（女性がいなくなると立ち直りますが。。）が表現できているでしょうか。（ベタですいません。ジェンダー的に気になる人は、有名人と一般人と解釈してください。）

●［第四工程］犬を追加して、犬に気をとられるルールを女性に与えましょう

［12］新たに、犬エージェントを加えましょう。（名称 Dog、数は 3 匹にしましょう）座標と方角はランダムに与えておきます。（男性や女性と同じです。分からない人は［3］）

［13］犬にもルールを与えます。犬は左右に各 30 度の範囲で方向を変えながら、1 ずつ進

みます。犬は周囲をまったく気にしません。

```
Turn(rnd()*60-30)                          ……犬のほうがふらふら歩きます。
Forward(1)
```

［14］女性は（男性を気にしませんが)、犬がいると立ち止まるルールにしましょう。（ほとんど男性と同じです）

```
ClearAgtset(My.Targets)                    ……まず Targets のなかみを空にする
MakeOneAgtsetAroundOwn(My.Targets, 3, Universe.Downtown.Dog, False)
If CountAgtset(My.Targets) == 0    then    ……まわりに誰もいなかったら
        Turn(rnd()*20-10)
        Forward(1)
End if
```

［15］（保存して実行してみてください）女性も犬に気をとられて立ち止まります。男性もつられて立ち止まります。

●［第五工程］犬もいろいろ考える

［16］犬エージェントは、人間は気にしませんが、他の犬エージェントを気にしているものとしましょう。他の犬を見つけると、（喜んで）その場でぐるぐる跳びはね始めるというルールにしましょう。（ほとんど、男性や女性と同じです）

```
ClearAgtset(My.Targets)                    ……まず Targets のなかみを空にする
MakeOneAgtsetAroundOwn(My.Targets, 3, Universe.Downtown.Dog, False)
If CountAgtset(My.Targets) == 0    then    ……まわりに誰もいなかったら
        Turn(rnd()*60-30)
        Forward(1)
Else                                       ……誰かいたら
        Turn(30)                           ……回り続ける
        Forward(0.2)                       ……飛び跳ねるので、ゆっくりめに
End if
```

［17］（保存して実行してみてください）犬たちがじゃれ合うようになったでしょうか。そしてそのまわりに人だかりができるようになったでしょうか。

●［第六工程］犬もいろいろ考える II

［18］犬が人間を気にしないのもおかしなものです。犬は、周囲にたくさんの人間（20 人よりたくさんとしましょう）が集まると（怒って吠えて）、人間たちを後ろに飛び退かせるというルールを与えましょう。

［19］MergeAgtset(Agt 集合**あ**,Agt 集合**い**)という関数をもちいて、（集合**あ**）に（集合**い**）を追加するという操作をすることができます。例えば、犬は周りを見て、男性集合を見て人間集合に追加し、女性集合を見て人間集合に追加する、という作業をさせれば、犬は周りにいる人間を認識できます。

［20］For Each 文というルールを用いることで、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、選別することができます。エージェント集合の中のエージェントを、順番に＜エージェント型変数＞に入れて、作用していきます。

```
For Each　＜エージェント型変数＞ in ＜エージェント集合型変数＞
                作用する内容
Next　　＜エージェント型変数＞
```

例えば、以下のようなルールを書くことで、隣人を（25,25）に集めることができます。

```
For Each　＜隣人＞ in ＜隣人達＞
                隣人.X = 25
                隣人.Y = 25
Next　　＜隣人＞
```

［21］［19］と［20］のルールを用いて、犬が人間を飛び退かせルールを書いてみましょう。自信のある人は自力で、ない人は下のルールを参考にしてください。

```
dim NeighborMales as Agtset　　……周囲の男性認識用の Agt 集合型変数
dim NeighborFemales as Agtset　……周囲の女性認識用の Agt 集合型変数
dim NeighborHumans as Agtset　……周囲の人間認識用の Agt 集合型変数
dim Neighbor as Agt　　……集合内の人間を扱うための Agt 型変数

MakeOneAgtsetAroundOwn(NeighborMales, 10, Universe.Downtown.Male, False)
                ……周囲の男性を認識する
MakeOneAgtsetAroundOwn(NeighborFemales,　10,　Universe.Downtown.Female,
```

```
False)                                  ……周囲の女性を認識する
MergeAgtset(NeighborHumans, NeighborMales)      ……人間集合に男性を追加
MergeAgtset(NeighborHumans, NeighborFemales)    ……人間集合に女性を追加


If CountAgtset(NeighborHumans) > 20 then ……もし人間が 20 人より多くなったら
    For Each Neighbor in NeighborHumans ……集合内のそれぞれの人間に対して
        Neighbor.X = Neighbor.X+(Neighbor.X-My.X)*0.1  ……飛び退かせる
        Neighbor.Y = Neighbor.Y+(Neighbor.Y-My.Y)*0.1  ……飛び退かせる
    Next Neighbor
Else

End if
```

●（第七工程）すこし見栄えを良くしましょう。

［21］これでモデルは完成ですが、吠えている犬の色を変えてみましょう。まずに犬の色を表現するための変数（名称 Color、整数型）を設定します。

［22］その変数に基づいて色を表示するように設定しておきましょう。設定＞出力設定＞マップ出力の編集＞マップ要素リストの編集＞エージェント表示色で設定します。

［23］色を設定、変化させるルールを書きましょう。（１）まず Agt Init のルールで色の初期値を与え（２）吠えているときは色を（例えば赤に）変えて（３）吠え終わるともとの色に戻します。（どこに入れるかは、考えてみてください）

```
My.Color = Color_Yellow                 ……色を黄にする
My.Color = Color_Red                    ……色を赤にする
```

●今回の復習

☆エージェント集合を操作するための関数いろいろ

今回（と前回まで）、ClearAgtset、MergeAgtset、CountAgtset というエージェント集合を操作する関数を勉強しました。artisoc には他にも、エージェント集合を操作するいろいろなルールを書くことができます。以下にまとめておきます。

ClearAgtset(集合 A)	集合 A を空にする
CopyAgtset(集合 A,集合 B)	集合 A を集合 B にコピーする
DelAgtset(集合 A,集合 B)	集合 A から集合 B に含まれる要素を削除する
JoinAgtset(集合 A,集合 B)	集合 A に集合 B を足し合わせる（重複あり）
MergeAgtset(集合 A,集合 B)	集合 A に集合 B を足し合わせる（重複なし）
PurifyAgtset(集合 A,集合 B)	集合 B の重複を取り除いて集合 A とする

MakeDiffAgtset(集合 A,集合 B,集合 C)

集合 B、C の片方だけに含まれているものを集合 A とする

MakeCommonAgtset(集合 A,集合 B,集合 C)

集合 B、C の両方に含まれているものを集合 A とする

☆線をひくやり方

エージェントが自分のもつ（エージェント集合型の）変数の中に他のエージェントをリストアップすると、それらのエージェントに対して線を引っ張ることができます。エージェント間にどんな社会関係が形成されているのかの図示に利用しましょう。ただし、ツリーの中で定義した変数でしか、この機能は使えません。（ルールのなかで定義した一時的な変数では使えません。）

☆ForEach 文で、エージェント集合に含まれるエージェントそれぞれに影響する

For Each 文を用いることで、エージェント集合のなかのエージェントそれぞれに、順番に作用していくことができます。

```
For Each　＜エージェント型変数＞ in ＜エージェント集合型変数＞
  作用する内容（エージェントを指定するために、エージェント型の変数を用います）
Next　　＜エージェント型変数＞
```

☆色を設定するやり方（簡易法）

整数型の変数を作っておいて、その変数に Color_Red、Color_Blue などの値を入れておきます。そして、その変数を「エージェント表示色」を指定してやると、エージェントをその色にすることができます。この機能を使えば、ルールのなかで、エージェントにその状態によって色を変えさせるということもできます。

色の一覧

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