

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

第六回 神様、観察する！（6月7日）

概略

前回の課題と前回の復習

エージェント集合関数 (MakeAgtSet 系の関数群)

MoveToSpace による移動ルール

時系列グラフへの出力

デバッガの使い方

課題

前回の課題と復習

6 名の方に提出していただきました。

前回学んだこと

セル型の空間。(出力範囲の修正を忘れずに)

Universe がルールや変数をもつことができる。

コントロールパネルの作成。Universe に設定した変数のみ。

For 文

CreateAgt (創り出すエージェント種別)

RandomPutAgtSetCell (エージェント集合型変数, 重なりを許すか (True/False))

MakeAgtSet (生成されるエージェント集合型変数, エージェント種別)

MergeAgtSet (エージェント集合型変数 1, エージェント集合型変数 2)

今回のねらい

前回は Universe にルールを書くことを学びながら、T.Schelling の分居モデルを作成しました。今回はモデルを完成させつつ、Universe のもう一つの役割 (世界を観察する) について学びます。

今週の文法

「Make=AgtSet 系列」の関数のまとめ

これまで、エージェント集合をとりあつかうための関数（エージェント集合関数）をいくつか学んできました。とくに、最初にエージェント集合を作成する関数（「Make=AgtSet 系」と仮に呼びます）は、エージェントが周囲を認識するルールを書くための重要なものです。たくさんの種類があるので、その区別をまとめておきます。（覚える必要はありません。ヘルプを参照しながら使用しましょう。）

> MakeAgtSet

この系列の中では少し特別な関数です。ある一つの種別のエージェントを全部、一つのエージェント集合型関数に格納します。

> Make **A** [One, All, なし] **AgtSet** **B** [AroundOwn, AroundPosition] **C** [Cell, なし]

Ex **MakeOneAgtSetAroundOwnCell** [Agt 集合型変数, 範囲, Agt 種別, 自分自身を含む?]

名称が複雑に見えますが、規則性を押さえると理解しやすいです。

A : どんな種類のエージェントでエージェント集合を作るのかを指定します。

{ 一種別のエージェントか、全種のエージェントか、指定したエージェント集合か }

B : どの視点からエージェント集合を作るのかを指定します。

{ 自分自身からか、指定した座標からか }

C : どのような空間認識によってエージェント集合を作るのかを指定します。

{ セル型か、実数空間か }

必要とされる引数がそれぞれ異なるので、ヘルプを参照しながら用いましょう。

前回作成したモデルをもとに、亀に周囲を認識させるルールを書きましょう。周囲の赤亀の数、青亀の数、亀の数を数えるルールを書いてみましょう。

コメント文を活用してみましょう

赤亀、青亀、両方にルールを書くのを忘れずにね。

ツリーに変数を作成した人は出力設定で情報表示をさせて、正しく認識しているかどうかを確かめましょう。

MoveToSpaceOwnCell

これまで、エージェントに移動させるのに、Turn, Forward という関数を持ちてきました。進む方向と距離を指定して移動させたのですが、セル型では、もう少しいい加減な(高度な?)移動方法もあります。「周囲に空いている空間があれば、そこに移動する」というルールです。

```
MoveToSpaceOwnCell( 視野 )
```

Ex. MoveToSpace(2)

不満があれば移動するルールを亀に書いてみます。

それぞれの亀に自分の分居率を計算させます。そして一般に亀が満足する分居率の基準(満足水準、閾値)を設定してやり、それと比較して不満か、どうかを決めます。この満足水準を Universe の変数として設定してやると、コントロールパネルで操作することが出来て簡単です。<ただし、**割り算の掟**に注意>

割り算の掟

一、ゼロで割ってはいけません。

周りに誰もいない亀は、分母が0になってしまいます。あらかじめ、周りいる亀が0の亀は、自動的に分居率が0になるように、If文で設定してあげましょう。

一、整数 [割る] 整数は自動的に、整数になってしまいます。

コンピュータプログラムの世界では、整数型変数 / 整数型変数は、自動的に整数として計算することになっています。なので、そのままだと分居率はいつも0になります。亀の数(赤亀の数、青亀の数)を実数型に設定してあげましょう。

Universe を使って、系を観察

Universe のルールを使って、系を観察することができます。不満をもつ亀の数を観察してみましょう。(課題で平均分居率についても計算してもらいます)系を観察するルールは、Universe の Step_End{}に書くのが適しているでしょう。まず、集計するための変数を用意します。今回は、Universe で全部処理してしまう方法と、エージェントにも協力してもらう方法を説明します。

(1) やや野暮ったいが、素直な方法

```
Univ_Step_End{
dim i as integer
//変数のもつ値を初期化します
Universe.Num_Unsatisfied = 0
// 不満をもつ亀の数を数えます
    // 不満をもつ赤亀を加えていきます
    For i = 0 to Universe.Num Reds-1
        If Universe.Streets.RedTurtle.Seg_Rate(i)< Universe.Seg_Criteria then
            Universe.Num_Unsatisfied = Universe.Num_Unsatisfied + 1
        end if
    Next i
    // 不満をもつ青亀を加えていきます
    For i = 0 to Universe.Num Blues-1
        If Universe.Streets.BlueTurtle.Seg_Rate(i)< Universe.Seg_Criteria then
            Universe.Num_Unsatisfied = Universe.Num_Unsatisfied + 1
        end if
    Next i
}
```

(2) お洒落でスムーズな方法

```
Univ_Step_Begin{
//変数のもつ値を初期化します
Universe.Num_Unsatisfied = 0
}

( 赤亀、青亀のルールに )
Agt_Step{
// 不満をもつ赤亀 ( 青亀 ) を加えていきます
If My.Seg_Rate < Universe.Seg_Criteria then
    Universe.Num_Unsatisfied = Universe.Num_Unsatisfied + 1
end if
}
```

観察結果の出力

観察した結果は、出力設定をしておく必要があります。今回は、時系列グラフ出力と値画面出力をやってみましょう。どちらも、[設定>出力設定]で、時系列グラフ出力/値画面出力を選択肢、[追加]を選択します。必要な項目を設定すれば、完成です。

デバッガの使用（ちょっと上級）

KK-MAS には、作成したモデルが思ったように正確に動いているかどうかをチェックするためのデバッガという機能がついています。デバッガを利用して、超コマ送りでモデルを実行し、好きなタイミングで止めて、その時点に変数などがどういう値をとっているかを調べることができます。

ステップイン（ステップオーバー）実行

ブレークポイント

変数の値を調べる

課題（分居モデルの改造）

（改造1）平均分居率の計算

今回の講義では Universe のルールを用いて、不満足な亀の数を数えてみました。次に、平均的な分居率を計算して、出力してみましょう。

（改造2）満足水準の個別化

これまで作った分居モデルはどれくらいの分居率で満足するかという値は、Universe によって一律に与えられていました。赤亀と青亀の持つ満足水準を別々に設定できるようにしてみましょう。

（改造3）亀のルールの変更

分居モデルは、一定の比率の同種別の亀がいないと引っ越しするというルールでした。では、ある一定の異種別の亀がいると、嫌がって引っ越しするというルールでは、系はどうふるまうでしょうか。