

第六回 シミュレーションの状況を集計する

前回の課題

- ① 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。

→ 空間のプロパティから大きさを変えます。表示は自動的に変わります。エージェント数の上限はコントロールパネルから変更します。空間を 20×20 に、エージェントの上限を 200 くらいにしてください。

- ② 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。

→ 今までは N で数を指定していましたが、これを消して、NR と NB の二つを作ります。それをコントロールパネルで制御できるようにしてください。ルールは

```
for i = 0 to universe.NB - 1
    CreateAgt(universe.city.blue)
next i
for i = 0 to universe.NR - 1
    CreateAgt(universe.city.red)
next i
```

とすればいいですね。エージェントは 150 ずつ位が良いかも。

- ③ 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。

→ 実数型の変数を Universe に作り、コントロールパネルを設定します。そして、

```
if rate < Universe.genkai then
    MovetoSpaceOwnCell(2)
end if
```

としましょう。

これで、エージェントの数、我慢の限界をコンパネで指定できるようになりました。

しかし、このままだと、分居しているのは分かるのですが、どれくらい分居しているのかは分かりません。

数値の出力

エージェントルールを思い出してください。各々が周囲を見回したときに同種の割合を計算していました。その割合(rate)がどういう値になっているのか覗いてみましょう。エージェントルールで、rate を計算したあとに

```
println(rate)
```

と書いて、ステップ実行してみてください。rate の中の数値がコンソール画面に出力され

ます。`print(rate)`とすることもできますが、改行なしで続けて出力されてしまいます。ちなみに、`print()`系の関数は、括弧の中に””で囲んで、好きな文字を出力する事ができます。””は文字列をそのまま扱うというしるしです。たとえば、

```
print(“同族の割合”)
println(rate)
```

と続けて書けば、

```
同族の割合 0.75
```

などとラベル付きで出力することもできます。試してみてください。コンソール画面に数値出力を続けていると、シミュレーションが重くなってきます。コンソール画面の文字を消去するには、`ClearConsoleScreen()`と書きます。試行ごとにクリアしたければ `Univ_Init` に、毎ステップクリアするには、`Univ_Step_Begin` に書くと良いでしょう。

コメント

とりあえず数字の出力を止めたいときには、コメントアウトといわれる方法もあります。

```
// や ‘ と書くと、それ以降の行全体が緑になって、シミュレータに無視されます
/* */ の間に挟まれた部分も全て無視されます。
```

この機能を使って、ルール中に説明を入れたり、一時的にルールを無効化できます。上の `print()` を無効化しておいてください。

Universe のルールによる集計

このままだと、全体的な傾向をつかむには向いていません。そこで平均的な分居率を計算してみましょう。まずは、集計しなければなりません。Universe に入れ物を作り、それぞれのエージェントに `rate` の値を報告させましょう。まずは Universe に変数を作ります。Univ_Step_Begin でその値を 0 にして、Univ_Step_End でその値をエージェント数で割ってあげると、そのステップにエージェントたちが直面した状況の平均を求めることができます。

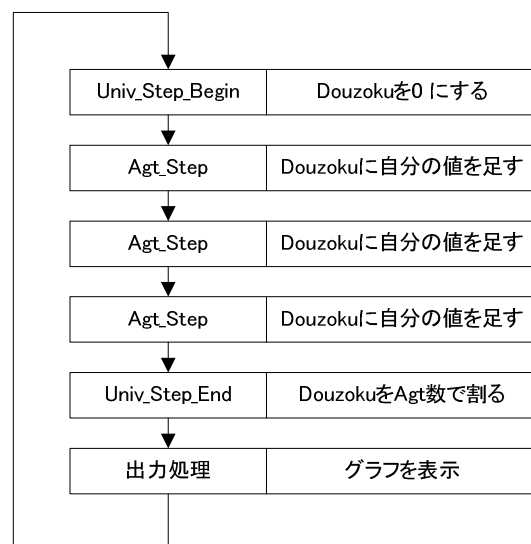
```
Univ_Step_Begin{
```

```
Universe.Douzoku = 0
```

```
}
```

```
Univ_Step_End{
```

```
Universe.Douzoku = Universe.Douzoku / (Universe.NR + Universe.NB)
```



```
print(getcountstep())
print(" ")
println(Universe.Douzoku)
}
```

エージェントルールには場所を考えて次のルールを追加しておいてください。

```
Universe.Douzoku = Universe.Douzoku + rate
```

グラフ出力の設定

では、この値をグラフに出力してみましょう。二次元マップ出力と同様、出力設定から時系列グラフを追加してください。二つの要素を追加します。一つ目は、平均的な同族の割合です。出力値は `Universe.Douzoku` と書いてください。もう一つ、我慢の限界も同時に示しましょう。出力値は変数の名前に応じて `Universe.genkai` などと書いてください。

→ 各々のエージェントの希望の同族率と、現実の同族率の違いを観察してください。

シミュレーションの自動終了

`artisoc` では、いくつかの方法でシミュレーションを終了させる事ができます。

① ■ ボタンを押す

② 実行環境設定で終了条件を設定する

最大ステップ数、最大実行時間、終了条件式で指定します。終了条件式に `universe.Douzoku > 0.7` と書けば、平均的な同族の割合が 0.7 以上になると自動的に終了します。

③ ルールのなかに試行を終了させる関数を書く

ルール中で、`ExitSimulation()` `ExitSimulationMsg()` `ExitSimulationMsgln()` のいずれかを書く事で終了します。`ExitSimulationMsg()` `ExitSimulationMsgln()` は、`print()` の機能を兼ねており、数値や文字列を出力する事ができます。`ln` が付く方は改行付きで出力します。たとえば、こんな風に使います。

```
if universe.Douzoku > 0.7 then
    Exitsimulationln("シミュレーションを終了します。")
end if
```

課題

- ・ 満足したエージェントの割合をグラフに出力してみましょう。
- ・ 全てのエージェントが満足したら終了するようにしてみましょう。