

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 13 回 シミュレーション結果の分析 (7 月 19 日)

今日の目的:

シミュレーション結果を細かく分析したい時には、空間やグラフの出力だけでは不十分なこともあります。今回は、ログを取って試行を丸ごと記録する方法、必要な情報をファイルに出入力する方法を紹介します。

ログファイルの作成と操作

モデルを開いた状態で、上のメニューから、「ログ>記録」を押すと、シミュレーションメモという画面が出てきます。ここでログに名前を付けます。了解を押すと、試行が開始され、停止ボタンを押すと終わります。ログは MAS の置いてあるフォルダの直下の replaylog フォルダに保存されます。

ログを再生するには、普通に MAS を起動したあと「ログ>再生」を選択してください。シミュレーション選択画面で再生したいログを選択すると、ログが読み込まれ、再生が開始できます。ログは再生、逆再生、ステップ再生に加えて、スライダーで好きなステップを選択することもできます。ログ再生パネルの停止ボタンを押すと、ログモードを終了して、まっさらな状態に戻ります。

練習:

課題 10 - 1(過去に作った他モデルでも良い)を用いてログファイルを作ってみてください。

ファイル出力設定

シミュレーション結果を厳密に分析するためには、統計的な処理が必要になることもありますし、外部のグラフ用ソフトを用いて数百回の試行データをまとめてプロットしたい場合も出てくるでしょう。このような作業に便利なのが、ファイル出力です。

ファイル出力は、出力設定で追加します。まずはファイル名を指定すると、その名前でも、モデルファイルと同じディレクトリに出力されます。それぞれのデータを区切る文字は指定することができます。普通はタブかコンマを選択しておく、エクセルで読み込むときに便利でしょう。direction,X,Y を出力するように設定した場合、データは以下の

ような形式で書き出されます。

```
//05/07/17 12:21:13
ステップ,direction,X,Y
0,221,25.0,10.0
1,15,24.0,20.0
2,74,23.0,24.0
3,320,23.0,22.0
4,155,22.0,15.0
```

ステップごとに改行が入り、各ステップ終了時の指定した値が、指定した区切り文字(この場合はコンマを選択)で区切られて出力されます。

例題 1 :

課題 10 - 1 の解答例をダウンロードし、腐ったミカンの数をステップごとに出力してください。それをエクセルで読み込んでみましょう。

まずは universe に腐ったミカンの数を表す変数を作りましょう。そして…

```
Univ_Step_End{
dim set as agtset
dim obj as agt
```

```
MakeagtsetSpace(set, universe.TUKUE)
```

机の上のエージェント全てをリスト

```
universe.RNum = 0
```

変数の初期化 (毎回 0 から数える)

```
for each obj in set
```

```
  if obj.fuhai == true then
```

腐敗フラグが true ならば…

```
    universe.RNum = universe.RNum + 1
```

変数に + 1

```
  end if
```

```
next obj
```

```
}
```

ルールから直接書き出し、読み出し

ファイル入出力関数を用いると、ルールの中から直接 CSV ファイルを読み込んだり、書き出したりすることができます。ファイル入出力は以下のような流れで行います。

ファイルを開く	OpenFileCSV(名称, 識別番号, モード)
データの読み書き	ReadFileCSV(識別番号)、WriteFileCSV(識別番号,内容,改行)
ファイルを閉じる	CloseFileCSV(識別番号)

まずは、ファイルを指定し、開かなくてはなりません。空のテキストファイルを作成し、適当な名前を付けて、モデルファイルと同じフォルダに入れておきましょう。名前が data.txt であれば、一つ目の引数は、“data.txt” (必ず“ ”付き)になります。

二つ目の引数はファイル識別番号です。この番号を指定することで、いくつものファイルを同時に開いても、どのファイルを操作すべきか分かります。この番号には 1 以上の整数を使います。

三つ目の引数はオープンモードです。1 なら read(読み込み。元のデータはいじりません)、2 なら write(書き込み。元のデータに新しいデータを上書きします)、3 なら append(追加書き込み。元のデータの最後に新たなデータを追加します)です。目的にあわせて選んでください。

開いたファイルを読み書きします。読むときには ReadFileCSV()を使います。引数はファイル識別番号だけです。この関数は、ファイルを開いてから 1 回目なら 1 個目のデータ、10 回目なら 10 個目のデータを読めます。したがって、data.txt の中身が「10 改行 14 改行 22 改行 13 改行 24 改行 5 改行 16 改行 4 改行」ならば、

```

dim i as integer
OpenFileCSV("data.txt", 1, 1)  data を識別番号 1 とし read モードで開く
for i = 0 to my.id              ID 番目のデータを読み込むための記述
    my.x = ReadFileCSV(1)      つぎの 1 データを読み込んで X に代入
next i
CloseFileCSV(1)               識別番号 1 を閉じる

```

とすると、それらの値を ID 順に自分の X 値に読み込みます。(ID0 なら x は 10、1 なら 14)

書き込む時には WriteFileCSV()を使います。write モードならばそのまま置き換え、append モードならば後ろに追記します。引数は三つです。一つ目はファイル識別番号、二つ目は書き出す内容です。ブール型(true/false)、整数型、実数型、文字列型を文字列型として書き出すことができます。三つ目は改行するかどうかをブール型で書き込みます。たとえば、

```
WriteFileCSV(1, my.x, ture)
```

```
WriteFileCSV(1, my.y, ture)
```

とすれば、自分の X と Y を書き出し、その都度改行します。したがって、ファイルには

```
“ 3.4433333 ”
```

```
“ 42.321111 ”
```

などと書き出されます。

最後に CloseFileCSV() を使って開いたファイルを閉じます。引数は識別番号だけです。

例題 2 :

こんどはファイル入出力関数を使って、ステップ数と腐ったミカンの数を出力してみましょう。ミカンのうち、腐ったものを universe の step_end で選り分けて数えて出力してください。ステップ数は Getcountstep() で取得できます。

```
OpenFileCSV("data.txt", 1, 3)
```

```
WriteFileCSV(1, getcountstep(), false)
```

```
WriteFileCSV(1, universe.RNum, true)
```

```
CloseFileCSV(1)
```

ファイル入出力による初期値設定

変数の初期値などをファイルから読み込ませることもできます。簡単な方法としては、まず、設定したい項目の初期値設定から、空の CSV ファイルを出力します。それをエクセルなどで読み込んで設定すると良いでしょう。

例題

逃亡者モデルは、逃亡者と追跡者が山をさまようモデルです。山が高いと移動スピードが遅くなります。素材と地形データをダウンロードして、山の高さ入力をしてみましょう。地形データは CSV ファイルに入っています。入力できたら、山の高さも出力設定してみてください。

課題

分居モデルを用いて、各々のエージェントの満足度を 0 から 100 まで 10% 刻みで変化させて試行してみましょう。表計算ソフトを使って、100 ステップ後の平均分居状況(各自の周囲の仲間率の平均)のグラフを作ってみましょう。