

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

第八回 他者の選び出し(6月14日)

今日の目的:

自分が選び出した他者に線を引く方法を学びます。さらに、相手の種類や距離などの情報を元に、特定の条件に当てはまる他者を選び出す方法を少し詳しく学習します。たとえば、道路にトラック、乗用車、自転車、歩行者のうち、トラックと乗用車を選び出して駐車違反を取り締まったり、近くや遠くの仲間(敵)だけを選び出したりする方法です。

線を引く

エージェント直下にエージェント集合型変数を作ると、その中に入っている個々のエージェントに対して、二次元マップ上で線(矢印)を引くことができます。何種類も線を引きたいときには、エージェント集合型変数をその分だけ作り、その分だけ表示リストに追加します。複数の線やエージェントを表示するとき、どちらを上を描くかは、マップ要素リストの右にある矢印で入れ替えられます。

例題

では、「空間上にちらばる複数のエージェントの間が線で繋がっている」状態を作る手順を見てみましょう。まずはモデルを新規作成して下さい。

- ・ 50×50 の空間(ループなし)と、一種類のエージェント(たとえば nattou を 5 つ)を作成しましょう。

- ・ Universe の Univ_init でエージェントをランダムに配置します。

```
Univ_Init{
  dim hachi as agtset           エージェント集合型変数が必要です
  MakeAgtSet(hachi, Universe.jutan.nattou)  hachi に nattou 全部格納します
  randomputagtset(hachi)       hachi に入っているモノをばらまきます
}
```

- ・ ツリーで、エージェント直下に、エージェント集合型変数を一つ作ります。

- ・ エージェントルールで、そのエージェント集合型変数に同種エージェントを格納します。

```
Agt_Step{
  MakeAgtSet(my.HokaNoTubu, Universe.Jutan.nattou)
}
```

・「設定>出力設定」でマップ出力を追加してください。マップ要素リストにエージェントを追加します。「線を引く」にチェックを入れると、線引対象、線種、矢印種別、色が設定できます。先ほど作ったエージェント集合型変数を線引対象に指定しましょう。「了解」を押して行けば設定は終了です。余裕です。

```
# 選択したエージェント直下にエージェント集合型変数が無い場合には、
# 「線を引く」は選択できません。
```

他者の選り出し

これまでに、MakeAgtSet()や MakeAllAgtSetAroundOwn()などの関数を学習しました。ここでは、こうした関数で作成したエージェント集合型変数を足し合わせたり、差分をとったりして、特定の条件に当てはまるものだけを選り出す方法を紹介します。現在、こうした作業に利用できる関数には、以下のようなものがあります。

- | | |
|-----------------------------|-------------------------------|
| ・ ClearAgtset(a) | 集合 a を空にする。 |
| ・ CopyAgtset(a, b) | 集合 a を集合 b にコピーする |
| ・ DelAgtset(a, b) | 集合 a から集合 b に含まれる要素を削除する |
| ・ JoinAgtset(a, b) | 集合 a に集合 b を足し合わせる(重複あり) |
| ・ MergeAgtset(a, b) | 集合 a に集合 b を足し合わせる(重複なし) |
| ・ PurifyAgtset(a, b) | 集合 b の重複を取り除いたものを集合 a に入れる |
| ・ MakeDiffAgtset(a, b, c) | 集合 b、c の両方に含まれるものを集合 a に入れる |
| ・ MakeCommonAgtset(a, b, c) | 集合 b、c の片方だけに含まれるものを集合 a に入れる |

全てを覚える必要はまるっきりありません。ただ、「こんなものもある」ということだけ知っておいてください。そうすれば、必要になったときにヘルプを見て利用できます。では、具体的な使い方の例を見てみましょう。

例題

さきほど例題 で作成したモデルを利用し、「20 ほどのエージェントがあり、線で繋がっているが、距離が 20 以内だと焦げ茶色の線、それ以上だと薄茶色の線になる」というモデルを作ってみましょう。

- ・ nattou 直下に cyou、tan の二つのエージェント集合型変数を作ります。

```
Agt_Step{
  MakeAgtset(my.cyou, Universe.Jutan.nattou)  全ての nattou を格納します
```

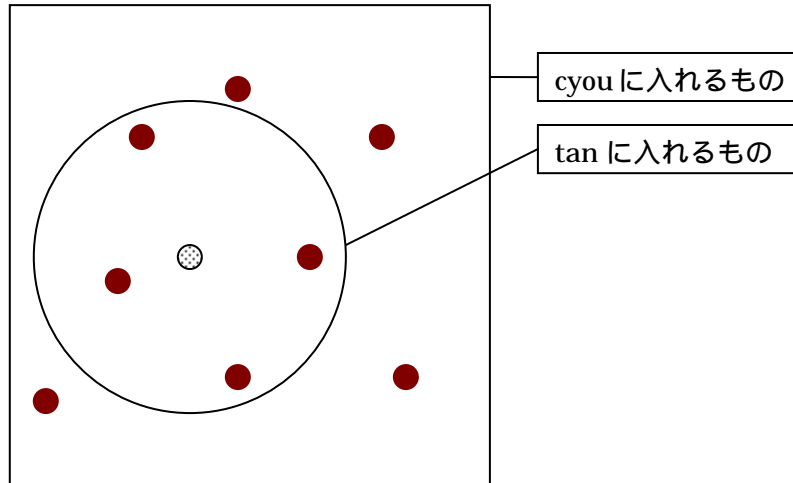
```
MakeOneAgtsetAroundOwn(my.tan, 20, Universe.Jutan.nattou, true)
```

距離 20 以内の nattou だけを格納します。

```
DelAgtSet(my.cyou, my.tan)
```

全体の集合から、距離 20 以内の nattou を削除します。

```
}
```



- ・ 線を引くように出力設定します。二種類の線なので、要素リストを二つ設定しなければなりません。うまくいったら、こんどはリストの順番を変えてみましょう。
- ・ はじめにランダムな方向を向き、ステップごとにその方向に 1 進むようにしてみましょう。距離が変わるに従って、線の色が変わっていますか？

課題

- ・ 新規にモデルを作成し、50×50 の空間の中に、1000 のエージェントをランダムにばらまいてください。それとは別のエージェントを 1 つ作り、空間の中央に配置してください。

中央のエージェントから視野 8 以内にいるエージェントに線を引いてみましょう。

MakeOneAgtsetAroundOwn MakeOneAgtsetAroundOwnCell それぞれを用いて試してみましょう。

視野が 0 から 24 まで徐々に広がり、また 0 に戻るように設定してみましょう。

モデルを別名で保存し、視野をコントロールパネルで自由に操作できるようにしてみましょう。

上記のモデルで、視野の半分までは別の色になるようにしてみましょう。

(こんなの余裕だという人用)距離が視野以内、その 2/3 以内、1/3 以内と近くなるにつれて、3 段階に線の色が濃くなるようにしてみましょう。