担当 田中明彦 tanaka@ioc.u-tokyo.ac.jp

TA 版本拓人 takutos@nifty.com 保城広至 hoshiro@ioc.u-tokyo.ac.jp

第2回 エージェントをうごかす!(10月14日)

概略

登録・動作状況の確認 新規モデルの作成 ッリー構造を設定(空間の設定、エージェントの設定) 出力設定 ルール:ルールエディタと実行順序 文法(「前進」「方向を変える」「代入」「My.」「乱数」)

登録・動作状況の確認

皆様、構造計画研究所の HP に登録し、ダウンロードした KK-MAS は無事に動いたでしょうか?ご報告をお願いします。

新規モデルの作成・ツリー構造の設定

今回からは、皆さん自身でモデルを作ってみます。

ダウンロードした KK-MAS.zip を解凍し、フォルダを開きます。 "kk-mas.bat"ファイルをダブルクッリクすると、KK-MAS が立ち上がります。





KK-MAS を立ち上げた時点で、自動的に新規モデルの作成画面になります。

空間を作ってみましょう。

[Universe を右クリック] > 空間の追加]

([挿入][空間の追加]でもできます。)



- ・空間の命名 (予約(禁止)語、"Space")
 「二次元空間」とします。
 ・空間のタイプ (格子モデルか、六角モデル)
 「格子モデル」とします。
 ・空間の大きさ (X軸、Y軸、レイヤ)
 それぞれ「50」「50」「1」とします。
- ・端点の処理 (ループのある、なし) 「ループする」にチェックします。

| 🔚 空間プロパティ | | | | |
|-----------|---------|--|--|--|
| 空間名: | 二次元空間 | | | |
| 空間種別: | 格子モデル ▼ | | | |
| 空間の次元数: | 2 | | | |
| 空間の大きさ X: | 50 | | | |
| Υ: | 50 | | | |
| レイヤ: | 1 | | | |
| 「端点の処理 | | | | |
| ◎ ループする | | | | |
| ○ ループしない | | | | |
| 了解取消し | | | | |

エージェントを作ってみましょう。

[二次元空間を右クリック]>エージェントの追加]

(二次元空間を選択したまま、[挿入][エージェントの追加]でもできます)

| <mark>情</mark> 新規ファイル - M | (K-Multi Agent | Simulator | | | | | | |
|---|--------------------|-----------|-------|---------|-------|-----|--------|------|
| ファイル(E) 編集 | 集(E) 表示(V) | 挿入() | 設定(S) | デバッグ(D) | 実行(R) | ログ山 | ウィンドウ₩ | ヘルプ田 |
| | 1 | | | | | | | |
| □ ッリー ● Universe □ 二次元空 | ロ ルールエデ・ | ۲.۶(R) | | | | | | |
| | 空間の追加(| 5) | | | | | | |
| | エージェン | ~の追加(| A) | | | | | |
| 1 | 変数の追加(| ۷ ا | | | | | | |
| 1 | 削除(D) | | | | | | | |
| | 初期値設定(|) | | | | | | |
| | ブロバティ(| Ð | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

> エージェントを命名します

「walker」と命名してみます

> エージェントの数

「1」と入力します

| 🔲 エージェントプロパティ |
|-----------------|
| エージェント名: Walker |
| エージェント数: 1 |
| 了解取消し |

> エージェントのプロパティをみるには

[Walkerを右クリック]>プロパティ]

| 📩 新規ファイル - KK | -Multi Agent | Simulator | | | | | | |
|--|--------------|-----------------|-------|---------|-------|-----|----------|--------|
| ファイル(F) 編集(| (E) 表示(\) | 挿入() | 設定(S) | デバッグ(D) | 実行(R) | ログ山 | ウィンドウ(W) | ヘルゴ(出) |
| | | | | | | | | |
| ご ダリー ● Universe ● ■ 二次元空間 ● ● Walker | 1 | | | | | | | |
| | ルールエデ | ィタ(R) | | | | | | |
| | 空間の追加 | (S) Noticita | (A) | | | | | |
| | 変数の追加 | (V) | | | | | | |
| | 削除(D) | | | | | | | |
| | 初期値設定 | 0 | | | | | | |
| | プロパティ | ® | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

変数を見てみましょう。

・"Walker"横の虫メガネのようなボタンをクリックしてください。

・エージェントには作成した時点で、自動的に幾つかの変数が備えられます。

| >X、Y、Layer | 各々、X 座標、Y 座標、レイヤを表します。 |
|-------------|-------------------------------|
| > Direction | エージェントの向きを表します。X 座標正方向が 0 で、 |
| | 左周りに一周で 360 となっています。度数表示。 |
| > 変数の初期値 | 全て 0 です。[右クリック] > 初期値設定] で設定で |
| | きます。 |
| > 変数のプロパティ | 変数の特徴や性格です。[右クリック] >プロパティ] |
| | で設定できます。詳しくは次回以降。 |

| <mark>市</mark> 新規ファイル - KK-Mult | i Agent S | Simulator | | | | | | |
|---------------------------------|----------------|-----------------|-------|---------|-------|-------|--------|------|
| ファイル(E) 編集(E) | 表示⊘ | 挿入() | 設定(S) | デバッグ(D) | 実行(R) | ログ(L) | ウィンドウ₩ | ヘルプ田 |
| | | | | | | | | |
| שיש 🗖 🔄 | | | | | | | | |
| ● Universe ● ■ 二次元空間 | | | | | | | | |
| ∳ | | | | | | | | |
| Ŷ | | | | | | | | |
| Layer | | | | _ | | | | |
| | ルール | エディタ | R (R) | | | | | |
| | 空間の | | | | | | | |
| | - エーシ - 密覇の | 'エントの istmAA | | | | | | |
| | 買用金(C |)) | | - | | | | |
| | 初期値 | 。 設定(1) | | - | | | | |
| | プロバ | דר® | | | | | | |
| <u> </u> | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

新しい変数を一つ追加してみましょう。 [Walker を右クリック]>変数の追加] ("Walker"を選択したまま、[挿入][変数の追加]でもできます)



 ・変数を自由に追加することができます。このエージェント(Walker)には、あとで 動いてもらうので、その速さを表す変数("Speed")を加えてみましょう。

> 変数の型 変数のとる値の型です。全ての変数に対して型を設定 しておく必要があります。実数型にしておきましょう。

その他は変更する必要はありません。

「了解」をクリックします。

| 📃 変数プロパテ | 1 |
|----------|--------|
| 変数名: | Speed |
| 変数の型: | 実数型 ▼ |
| 記憶数: | 0 |
| 次元数: | 0 |
| 0 🕂 次元目(| の配列数: |
| | 了解取消し |

ッリーの Walker 下に"Speed"が追加されました。



これで一応、ツリー構造の設定が終わりました。ツリー構造は好きなときに好きなよう に書き換えることが出来ます。

出力の設定

空間やエージェントができました。しかし、出力設定をしないと、それを見る(見せる) ことは出来ません。[設定>出力設定>追加(追加する出力種類は「マップ出力」)]で、 作った空間を画面に登場させましょう。

| ファイルの 標準(の) 表示の 挿入の 時定(の) デバッグ(の) 事件(の) ログ(1) ウィンドウ(の) | | | | | | |
|--|------|--|--|--|--|--|
| STING WE WIG WIG WIG TO STILLE | ヘルブ田 | | | | | |
| | | | | | | |
| 出力設定① | _ | | | | | |
| 第行環境設定(R) | | | | | | |
| Oniverse コントロールパネル設定(C) コントロールパネル設定(C) | | | | | | |
| ♥ ● Walker リモート設定(③) | | | | | | |
| | | | | | | |
| - 🥥 Layer | | | | | | |
| - Pirection | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| J | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

| 🔲 出力項目リスト |
|--------------------------|
| |
| |
| 道加(A) 編集(E) 北°~(C) 削除(D) |
| 追加する出力種類: マップ出力 ▼ |
| 了解 取消し |

マップ名、マップタイトルを「二次元空間」として、

[マップ要素リスト>追加]で、エージェントも画面に表示しましょう。

| 空間名: 志示階層: マッブ名: マッブタイトル: 凡例表示: 皆景画像: |
|--|
| 原点位置: 左上 ● 左下 表示型: ● チェス型 ● 囲碁型 ×軸設定 ● カール: ● カール: 最小値: 0 ● 泉大値: 0 最大値: 50 ● 泉大値: 50 |
| |

要素名を"Walker"として、了解を押します。

| 🔲 要素設定 🔅 | |
|----------|---------------|
| 要素名: | Walker |
| 出力対象: | Walker 🗸 |
| マーカー: | ○ なし ● 選択 |
| エージェント語 | 長示色: ● 固定色 |
| エージェントす | 愛数情報: 🗌 情報表示 |
| □ 線を引く | |
| | 了解取消し |

マップ出力設定、出力項目リストも[了解]をクリックして閉じます。

> とりあえずモデルの登場人物はそろいました。実行ボタン(三角が横になっているボタン)を押せば「歩行者」が現れます。ただしまだ動きません。



> 出力設定を使って、マップ出力の他に、時系列グラフ、棒グラフ、値画面出力、ファイ ル出力などができます。これらの設定も詳しくは次回以降。

ファイルを保存しておきましょう。

[ファイル>名前を付けて保存]で、好きな名前を付けて保存します。 >開くときは、[ファイル>開く]で、該当するファイルをクリックして開きます。 ルール:ルールエディタと実行順序(簡易版)

>今のところ、エージェントは、何も指示されていないので、何も行動しません。私たち がルールを書き込んでやることで、これらの登場人物たちを動かします。ルールを書き 込む場所(ルールエディタ)は、Universe かエージェントを[右クリック>ルールエデ ィタ]で開くことが出来ます。ダブルクリックでも開きます。

| 📩 新規ファイル - KK-Multi Agent Simulator | |
|--|------------------------------------|
| | デバッグ(D)実行(R) ログ(L) ウィンドウ(M) ヘルプ(H) |
| | |
| □ ッリー ● Universe ● ■ 二次元空間 ● ■ Walker | |
| ・ ● X 少 → Y 空間の追加(S) ● L ● L ● DI エージェントの追加(A) | |
| - | |
| 削除(D) | |
| 初期值設定() | |
| プロパティ(P) | |
| | |
| | |
| | |
| | |
| | |

| Agt_Init{} | 最初に(登場したとき)一度だけ行うルールを書きます。 |
|------------|-----------------------------|
| | 初期値設定をツリーではなく、ここで行うことが出来ます。 |
| Agt_Step{} | 各ステップに行うルールをここに書きます。 |
| | ここに書かれたルールは、ステップごとに実行されます。 |
| | |

実行順序については、あとでもう一度詳しく取り上げます。(ちなみにとても重要です) これでルールを書き込む用意ができました。 見た目を美しくするために、[設定>実行環境設定>ガーベージコレクション間隔]を 10から1に変更しておきましょう。(モデルを早く動かしたいときは、元に戻してくだ さい)



| 🔲 実行環境設定 |
|-------------------------------|
| (シミュレーション) 実行順度 |
| |
| シミュレーション終了条件 |
| 最大ステップ数: ステップ |
| 最大実行時間: 分 |
| 終了条件式: |
| 実行ウェイト: 0 ミリ秒 |
| 乱数シード値: |
| 出力タイミング: ④ ステップ毎 1 |
| ○ エージェント毎 ガーベージコレクション間隔: 1 |
| 了解 取消し |

今日の文法事項

「前進」「方向転換」ためのルールの書き方や数値や変数を扱うための基本的なルール。

Forward (

の距離だけ前に(=自分の Direction の方向に)進みます。 例、Forward(1)

Turn (_____)

の角度(度数)だけ左に曲がる。Directionを変化させます。 例、Turn(1), Turn(-10)

My.

エージェントが自分自身の変数を指定するときに用いる。

例、My.X, My.Y, My.Direction

=

代入を表すための命令文です。左辺の変数に右辺の値を代入します。 例、My.X = 25(自分の X 座標を 25 とします)

rnd ()

0 以上、 1 未満の一様乱数を発生させます。 例、My.Speed = rnd()*10 (

それでは、いくつかの例をつくってみましょう。

[1]X軸方向に1ずつ前進する歩行者

> 下記のように記述します。簡単ですね。

Agt_Init{ } Agt_Step{ forward(1) }

)

たった一行記述するだけで、Walker が動き出しました!

| 🚹 二次元空間 | | r ⊠_ ⊠ |
|---------|-------|----------------------------|
| | 二次元空間 | _ |
| | | |
| | | |
| | | |
| | | Walker |
| | | |
| | | |
| | |] |
| | | |

[2]右上45°方向に1ずつ前進する歩行者

> 下記のように記述します。最初に Direction を指定すれば良いだけです。



- [3]1ずつ前進しながら、5°ずつ左に曲がっていく歩行者
- > 下記のように記述します。ステップ毎に Direction が変わるようにします。

| Agt_Init{ | | |
|------------|--|--|
| } | | |
| Agt_Step{ | | |
| forward(1) | | |
| Turn(5) | | |
| } | | |

[4]空間の真ん中からいろいろな方向にまっすぐ1ずつ歩き出す10人の歩行者。 >まずは Walker のプロパティでエージェント数を10とします。

| ジェントプロパティ |
|-----------------|
| エージェント名: Walker |
| エージェント数: 10 |
| 了解 取消し |

> 下記のように記述します。空間は 50 × 50 なので、最初に現れる位置を X、Y ともに真 ん中の数である 25 を代入し、Direction をランダムに決めます。方向は 360 度なので、 360 を掛けてください。

| Agt_Init{ |
|----------------------------|
| my.X = 25 |
| my.Y = 25 |
| my.Direction = rnd() * 360 |
| |
| } |
| |
| Agt_Step{ |
| forward(1) |
| |
| } |

- [5]空間の真ん中からいろいろな方向にそれぞれの速さで歩き出す10人の歩行者。
- > 下記のように記述します。最初に自分の歩く早さをランダムに決めて(必ずしも 10 を 掛ける必要はありません)、それを forward に放り込みます(この数を引数 - ひきすう ーと呼びます)。

```
Agt_Init{

my.X = 25

my.Y = 25

my.Direction = rnd() * 360

my.Speed = rnd()*10

}

Agt_Step{

forward(my.Speed)

}
```



[6]エージェントの数を1に戻して、どんどんと歩く早さを加速していく孤独なランナー >下記のように記述します。

ステップ毎に、スピードを増加させるには、今回のステップの速さが、前回のステップ のそれより1大きければ良いと考えます。"="は等式ではなく、代入であることに注意。

```
Agt_Init{

my.X = 25

my.Y = 25

my.Direction = rnd() * 360

}

Agt_Step{

my.Speed = my.Speed + 1

forward(my.Speed)

}
```

課題

>では皆さんに宿題です。

[7]千鳥足でふらふらと歩く酔っぱらいたち(歩く速度、方向がステップ毎に変わる歩 行者)

[8]100×100の空間上で、各々別々なところからスタートして、いろいろな方向にそれであるで歩き出す100人の歩行者。

[9]forward()以外にも、今回の文法事項だけでエージェントを移動させる方法があります。そのやり方を考えてみてください。

< ヒント! >

現在自分のいる位置がX10、Y10だとします。次にX11、Y9の座標に進みたかった らどうするか?? > 下記のように記述します。

スピードはゆっくりの方が千鳥足っぽい雰囲気がでるので0~1の範囲に抑えています。 方向は、+30~-30の範囲としています。もちろんこの範囲以外でも構いません。

```
Agt_Init{

my.X = 25

my.Y = 25

}

Agt_Step{

my.Speed = rnd()

my.Direction = rnd()*60

forward(my.Speed)

}
```