国際政治と情報（2005 年後期）

担当　田中明彦 tanaka@ioc.u-tokyo.ac.jp
TA　阪本拓人 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
○ツリー構造を設定（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。

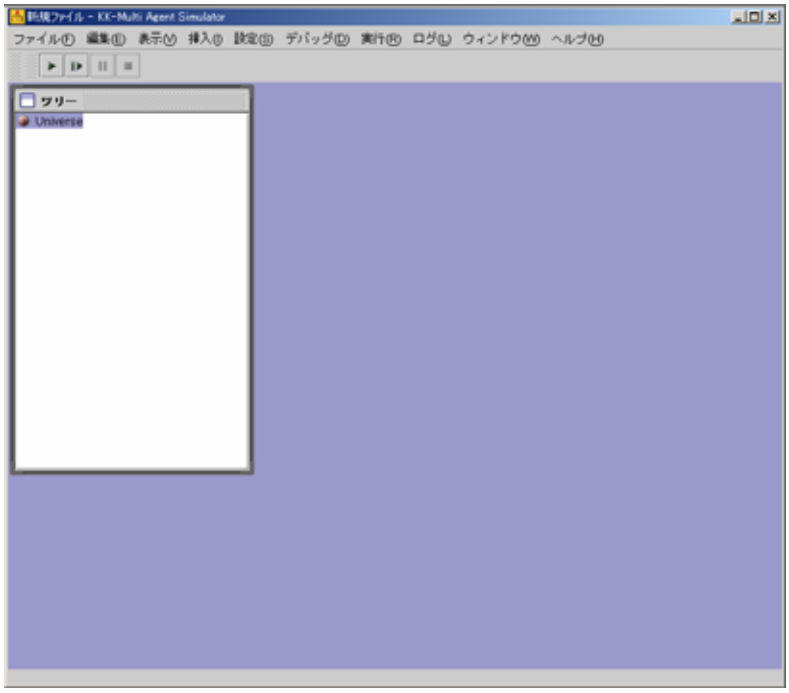
★★★新規モデルの作成・ツリー構造の設定★★★

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。

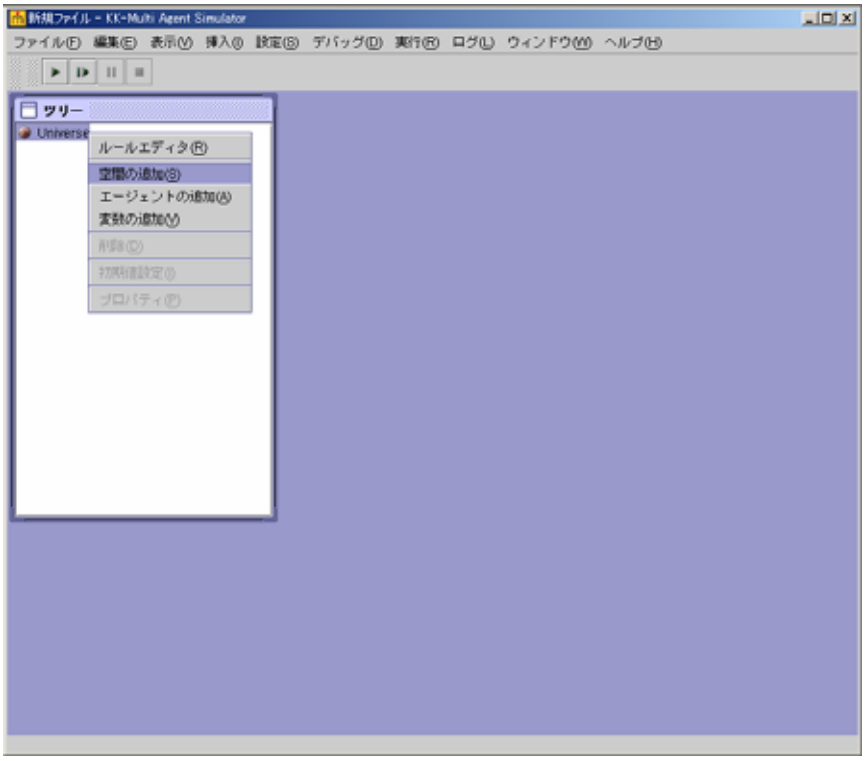
kk-mas.bat

●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）

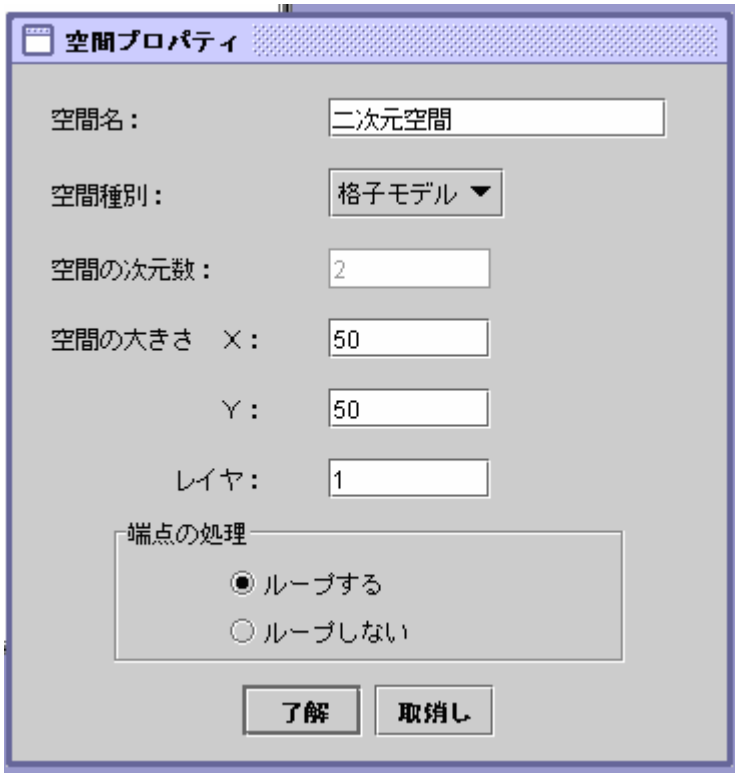
⇒「格子モデル」とします。

・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）

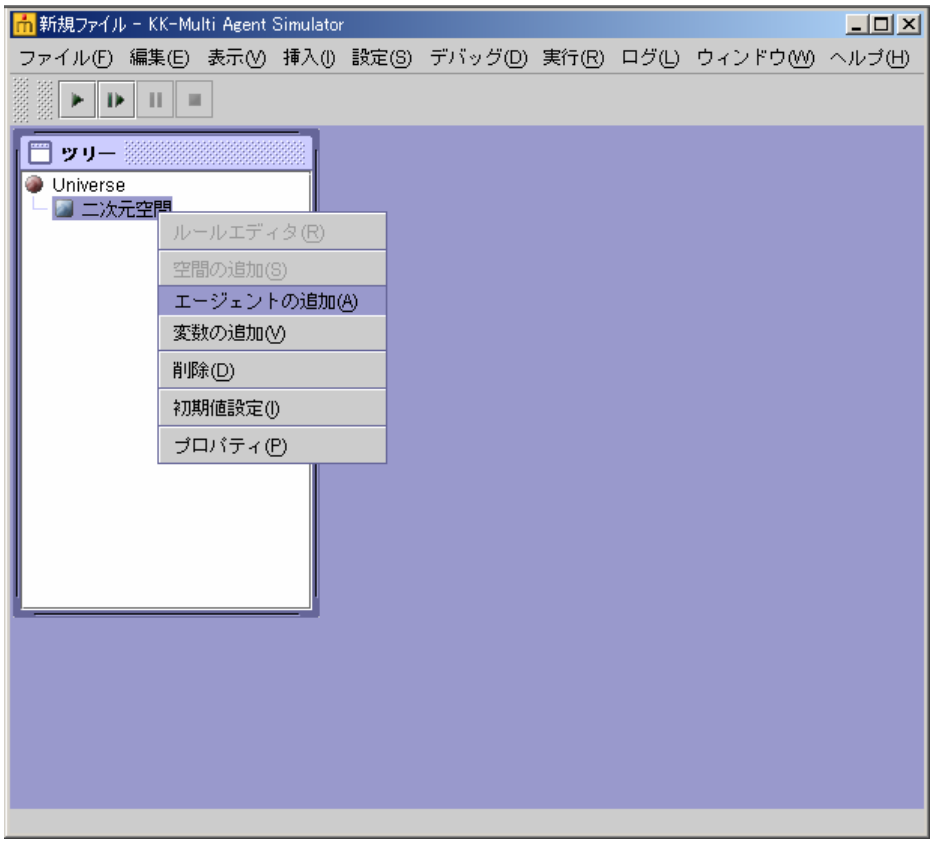
⇒「ループする」にチェックします。



●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）

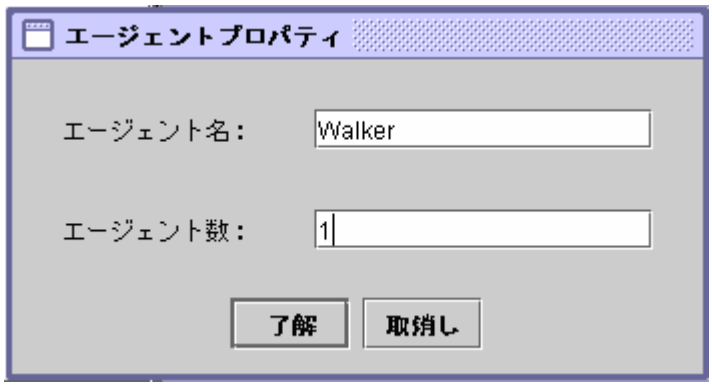


＞エージェントを命名します

⇒「walker」と命名してみます

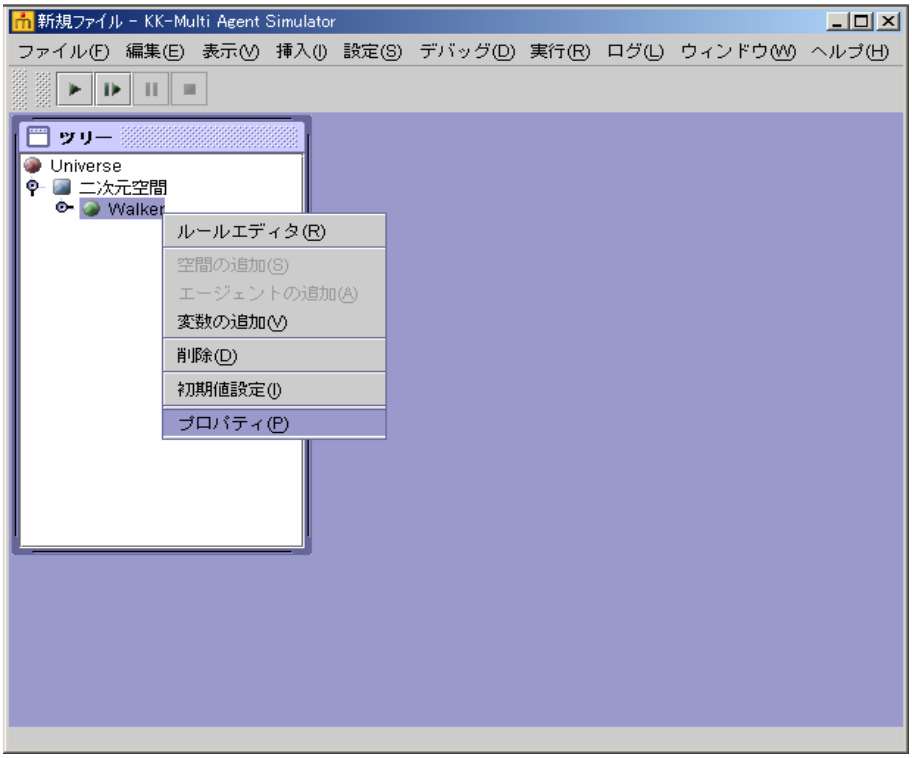
＞エージェントの数

⇒「1」と入力します



＞エージェントのプロパティをみるには

［Walker を右クリック］＞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。

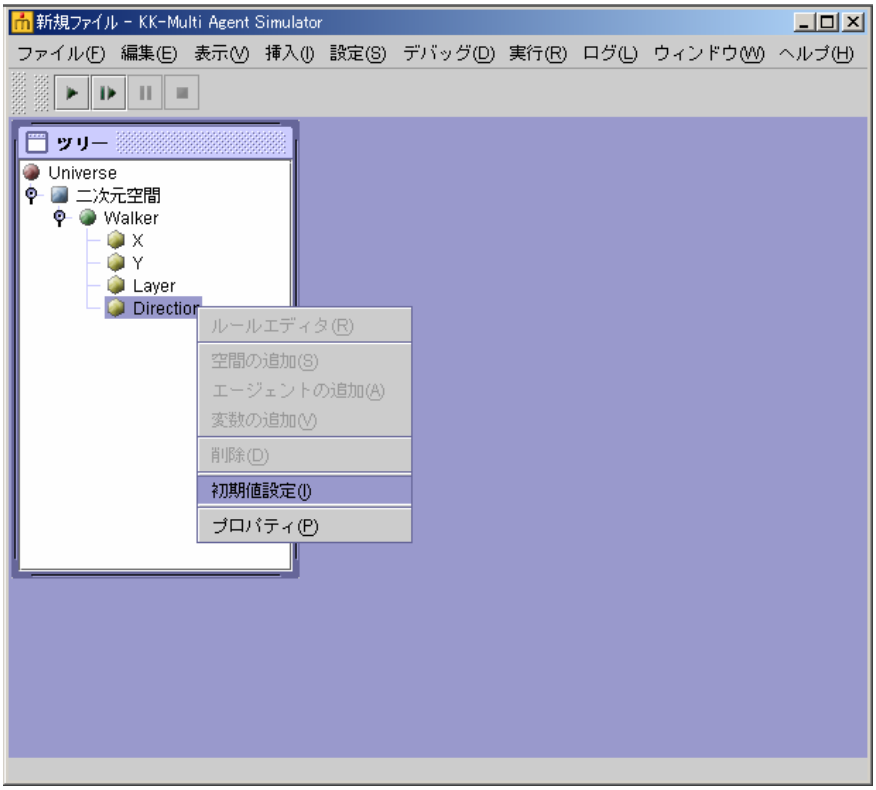
・エージェントには作成した時点で、自動的に幾つかの変数が備えられます。

＞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。

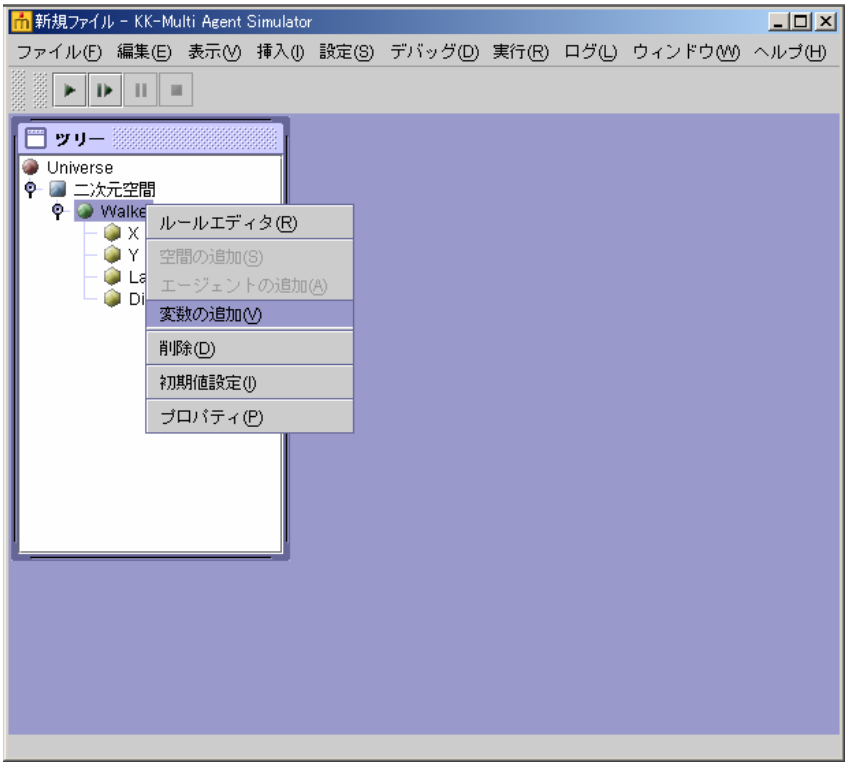
＞変数の初期値　　全て 0 です。［右クリック］＞初期値設定］で設定できます。

＞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。



●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）



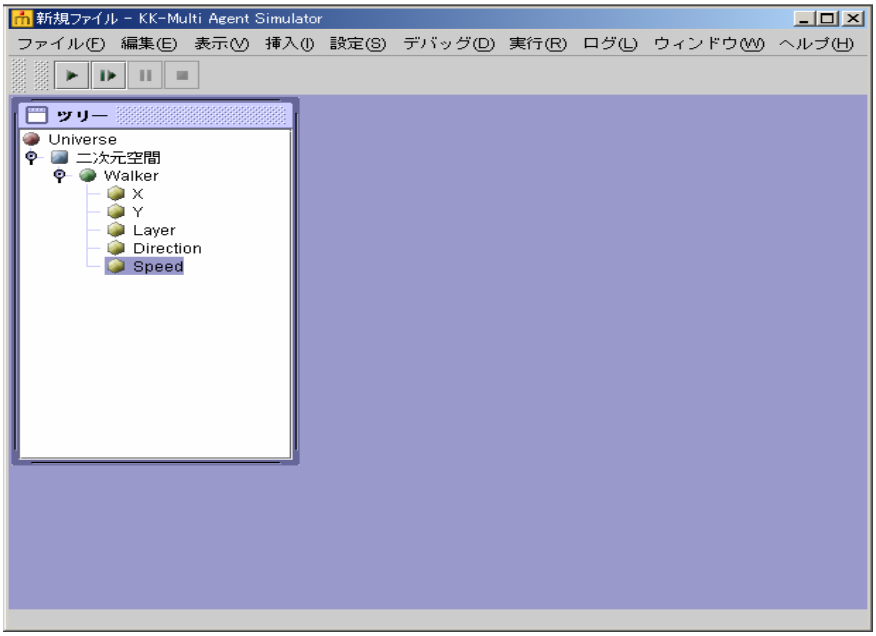
・変数を自由に追加することができます。このエージェント（Walker）には、あとで動いてもらうので、その速さを表す変数（”Speed”）を加えてみましょう。

＞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。

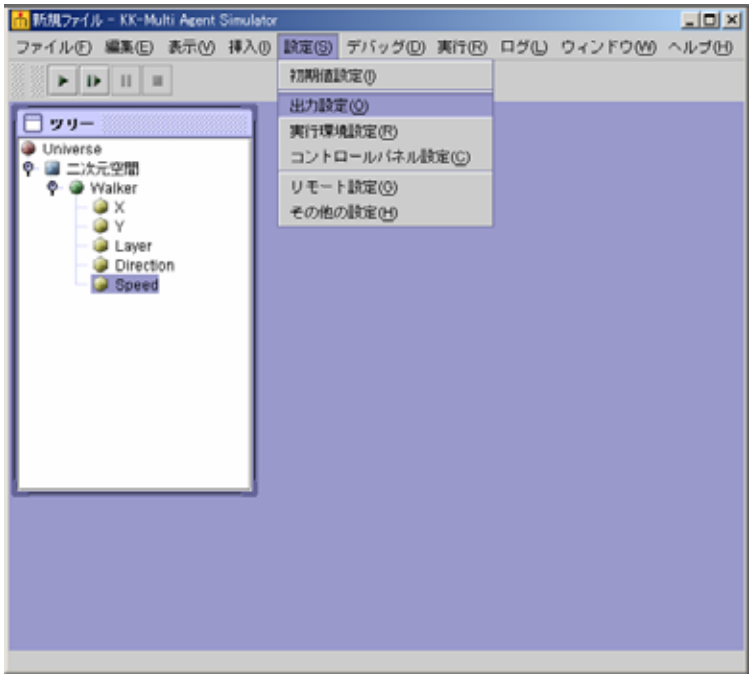
ツリーの Walker 下に”Speed”が追加されました。



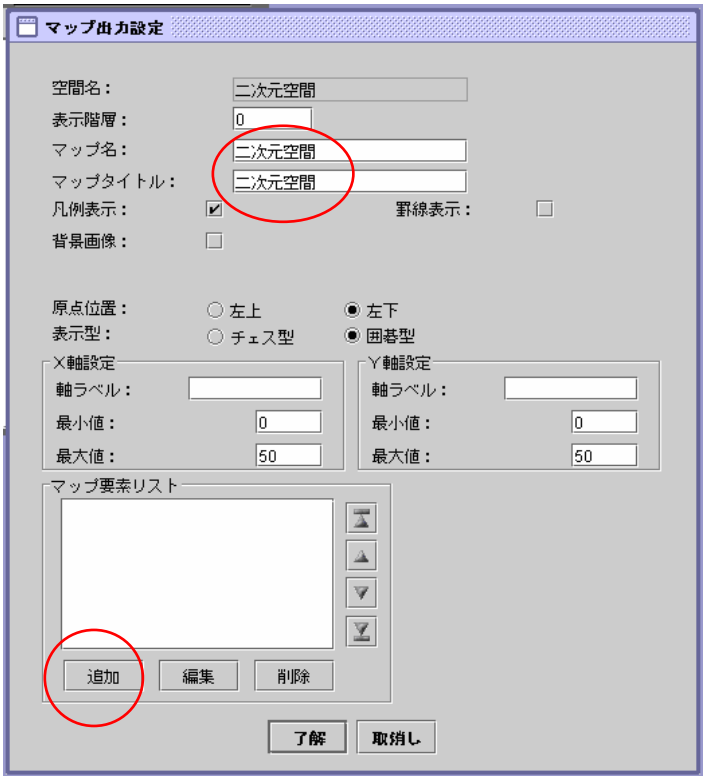
○これで一応、ツリー構造の設定が終わりました。ツリー構造は好きなときに好きなように書き換えることが出来ます。

●出力の設定

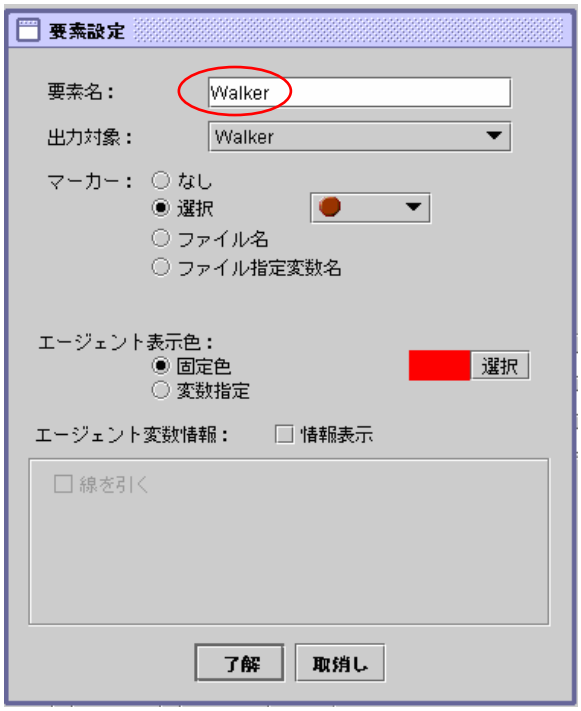
空間やエージェントができました。しかし、出力設定をしないと、それを見る（見せる）ことは出来ません。［設定＞出力設定＞追加（追加する出力種類は「マップ出力」）］で、作った空間を画面に登場させましょう。



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。

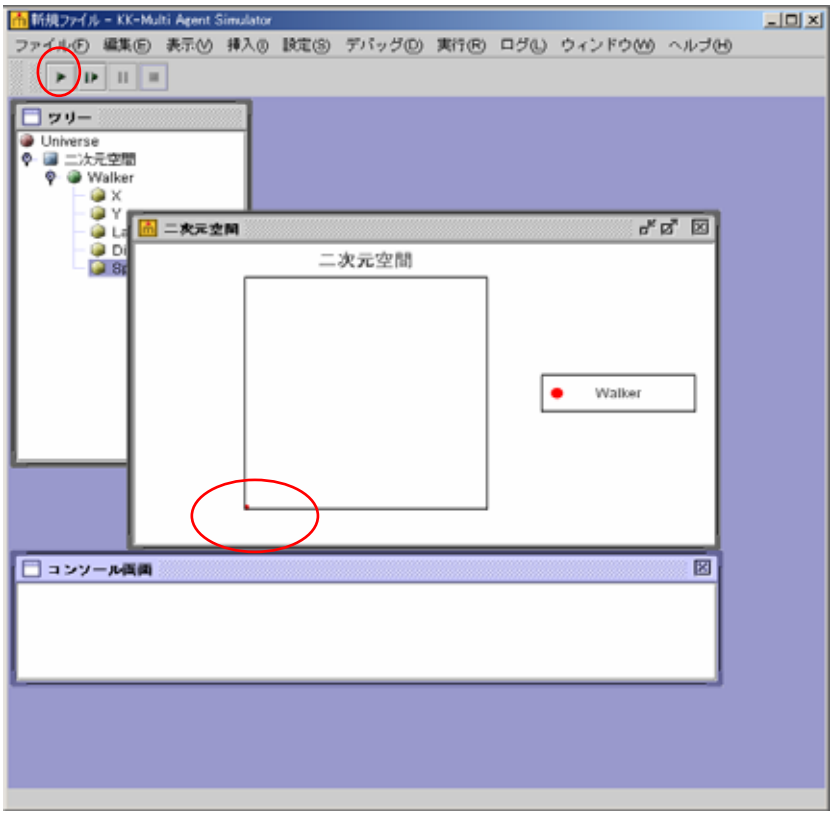


要素名を“ Walker ”として、了解を押します。



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。

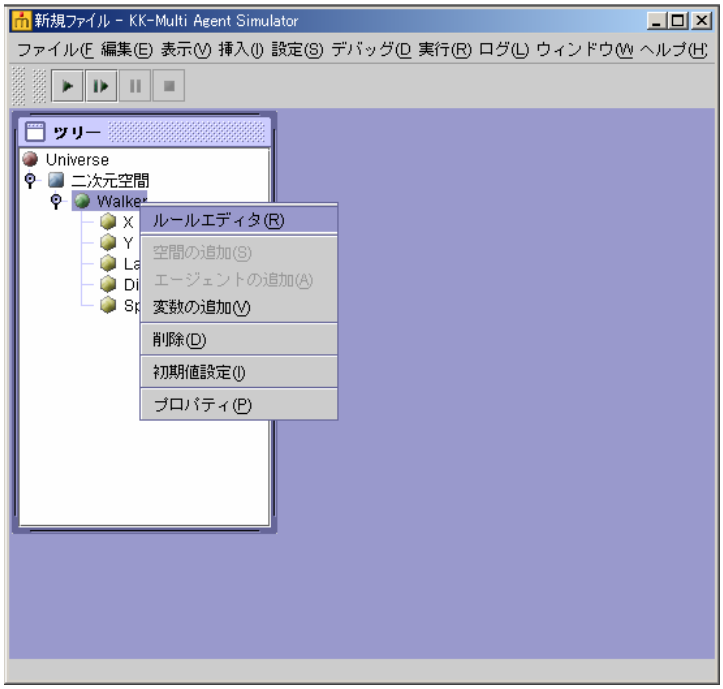
●ファイルを保存しておきましょう。

［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。

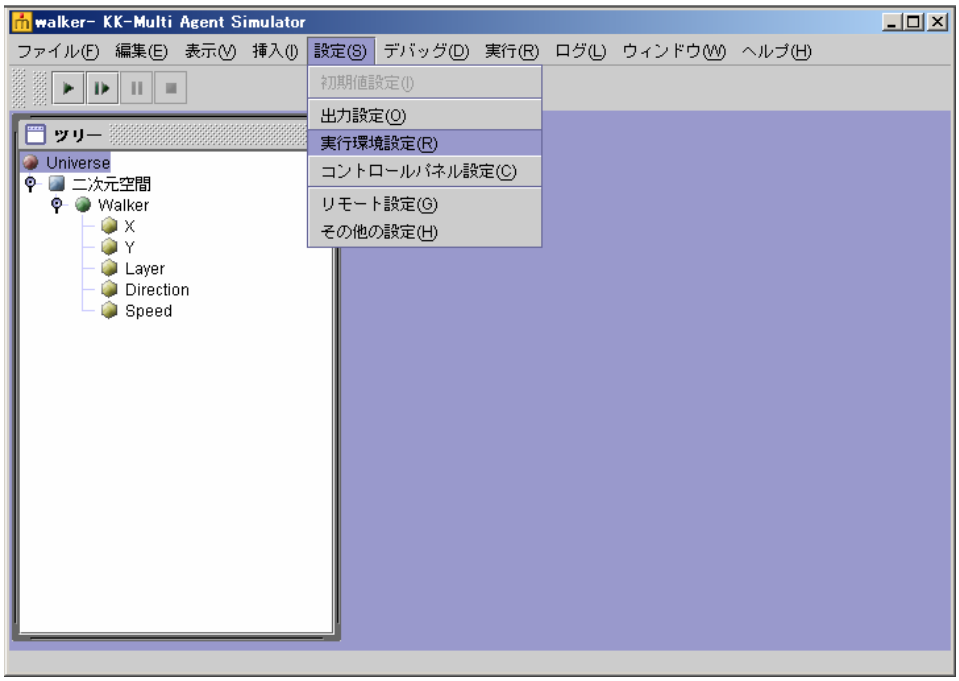


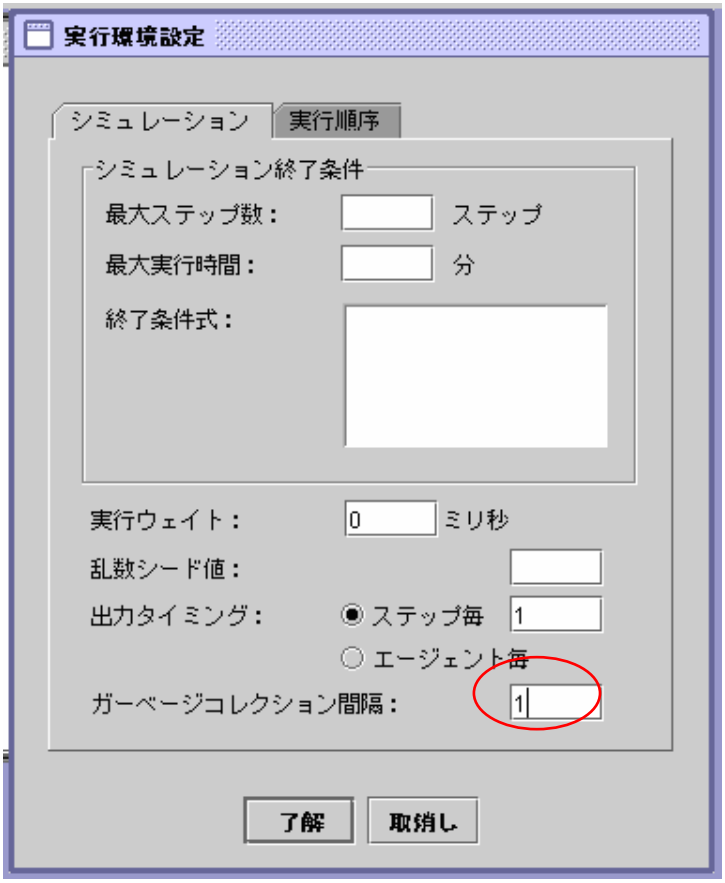
Agt_Init{}	**最初に（登場したとき）一度だけ行うルールを書きます。** **初期値設定をツリーではなく、ここで行うことが出来ます。**
Agt_Step{}	**各ステップに行うルールをここに書きます。** **ここに書かれたルールは、ステップごとに実行されます。**

○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）





●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。

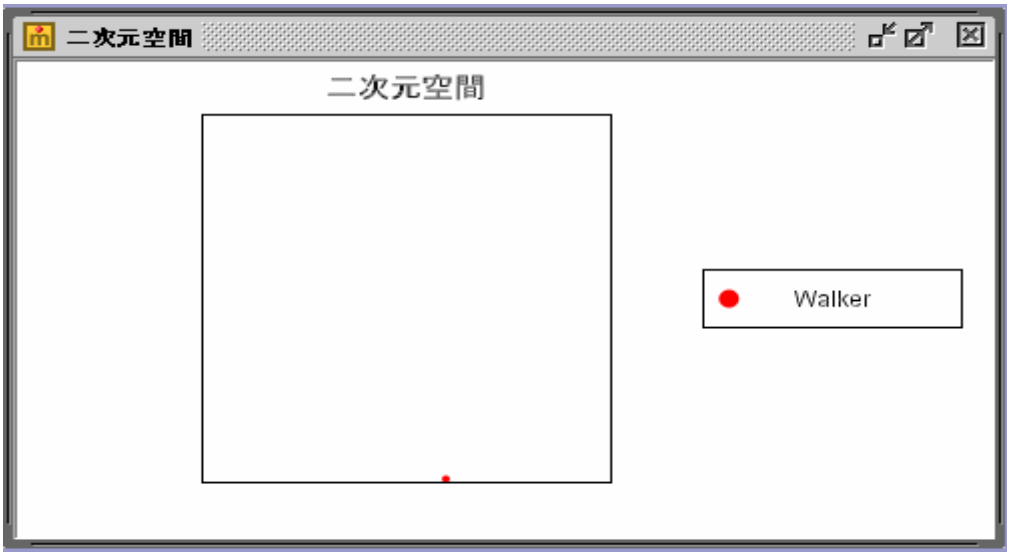
［1］X 軸方向に１ずつ前進する歩行者

＞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！



［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。

```
Agt_Init{
   Turn(45)
}

Agt_Step{
 forward(1)
}
```

［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。

＞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。

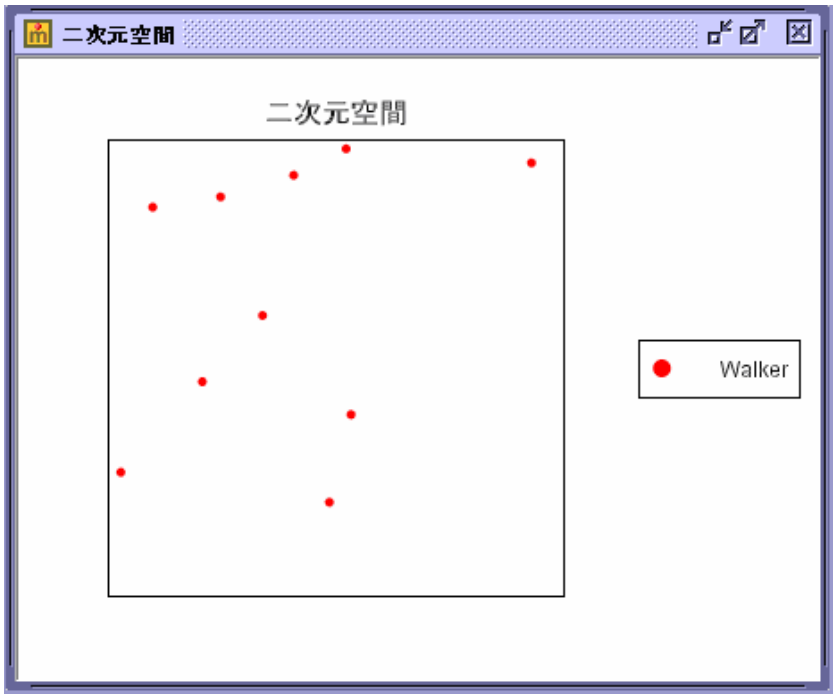
＞下記のように記述します。最初に自分の歩く早さをランダムに決めて（必ずしも 10 を掛ける必要はありません）、それを forward に放り込みます（この数を引数－ひきすうーと呼びます）。

```
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ぞれの速さで歩き出す 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〜ー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)

}
```