artisoc チュートリアル

お問合せは 創造工学部 まで

TEL　：　03-5342-1125
E-mail：　artisoc@kke.co.jp

株式会社　構造計画研究所
〒164-0012 東京都中野区本町 4-38-13　創造工学部　TEL：03-5342-1125　FAX：03-5342-1225

http://www.kke.co.jp

社会現象をシミュレーションしよう

■ユーザフレンドリーなマルチエージェント・シミュレータ『artisoc』を体験します。

- ●3つのステップではじめるマルチエージェント・シミュレーション
- ●エージェント同士を相互作用させよう！
- ●本格的なモデルへのいざない

3つのステップではじめるマルチエージェント・シミュレーション

■artisocでシミュレーションを行うための手順は次の通りです。

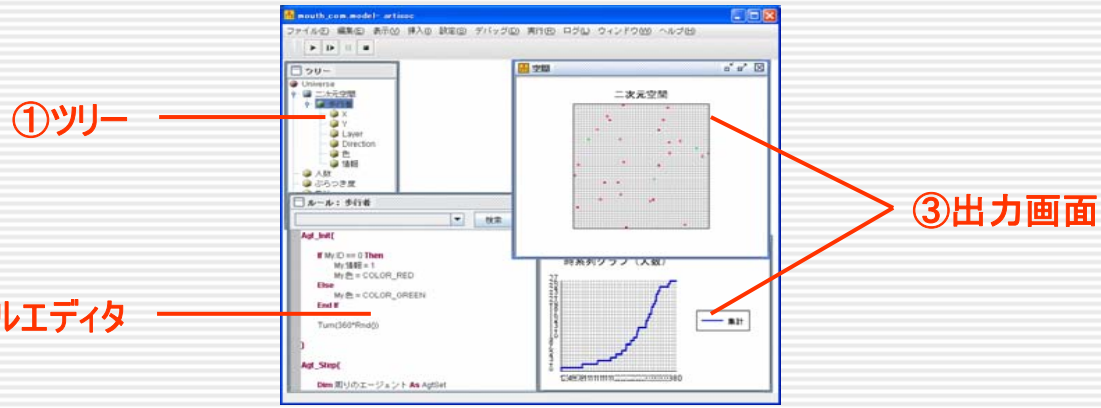
操作手順

① ツリーで「エージェント」を定義します。

② エージェントのルールを記述します。

③ 出力設定を行います。

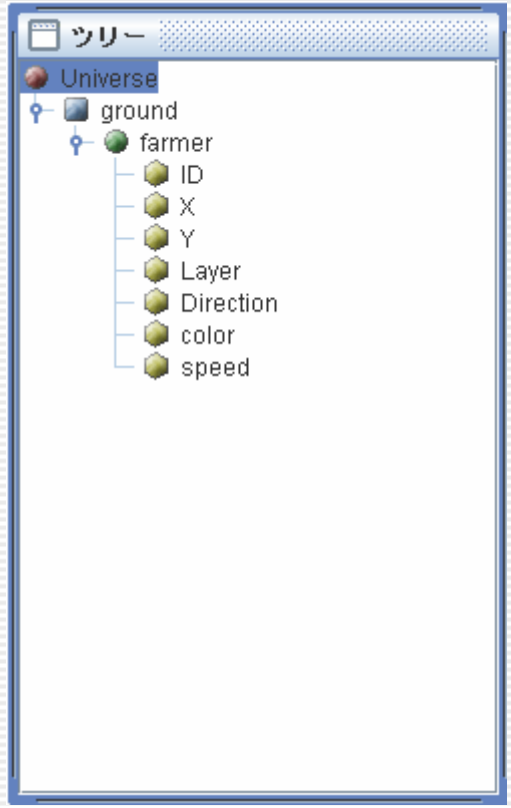
シミュレーションを実行します。



① エージェントを定義する

■ツリーで「空間」と「エージェント」と「変数」を定義します。

- ツリーの「Universe」で右クリックして、「空間の追加」を選択します。
  - 空間名： ground
- ツリーの「ground」で右クリックして、「エージェントの追加」を選択します。
  - エージェント名： farmer
  - エージェント数： 3
- ツリーの「farmer」で右クリックして、「変数の追加」を選択します。
  - 変数名： color
  - 変数の型： 整数型
  - 変数名： speed
  - 変数の型： 実数型



② エージェントのルールを記述する

■エージェントの色と歩く速度を指定し、画面中央に移動、向きを変える。

●ツリーの「farmer」で右クリックして、「ルールエディタ」を選択します。

```
Agt_Init{
        My.color = COLOR_BLUE
        My.speed = 1
        MoveToCenter()
        Turn(Rnd() * 360)
}

Agt_Step{
        Forward(My.speed)
}
```

表示色は青色

歩く速度は1

画面中央に移動

指定した角度(°)だけ回転する

前に進む(マイナス値のときはバック)

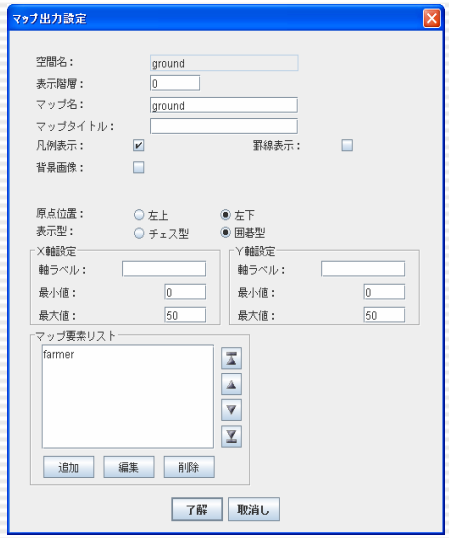
大文字、小文字の区別はありません。
行頭のスペースはなくてもよいです。
「My.」と記述すると、属性の候補が表示されます。
関数名のはじめの数文字を入力した途中で Ctrl + SPACE キーを押すと関数の候補が表示されます。

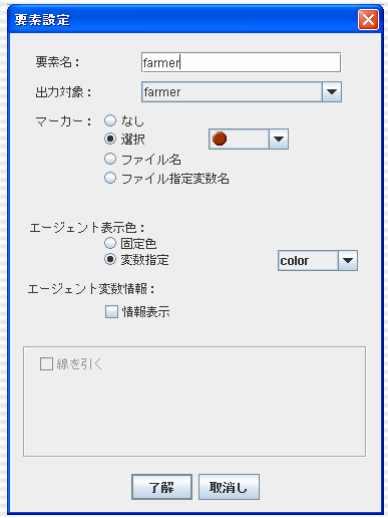
関数の詳細については、スタートメニューの[artisoc]-[documents]-[help Japanese] (help-ja.pdf)をご確認下さい。

③ 出力設定を行う

■出力項目の設定を行います。

- ツールバーの[設定]-[出力設定]を選択します。
- 出力項目リストが表示されますので、追加する出力種類を「マップ出力」にして「追加」ボタンをクリックします。
  - マップ名： ground
- マップ要素リストで「追加」ボタンをクリックします。
  - 要素名： farmer
  - エージェント表示色：
    - 変数指定： color



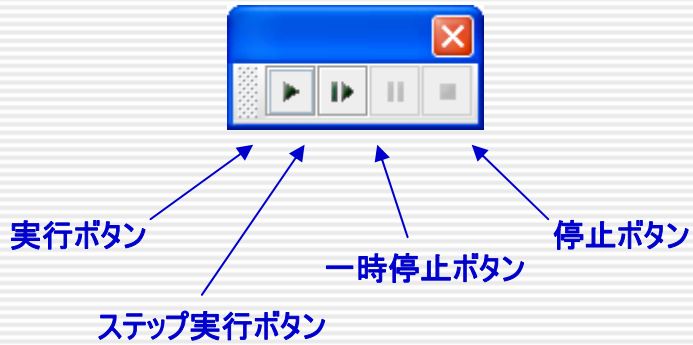


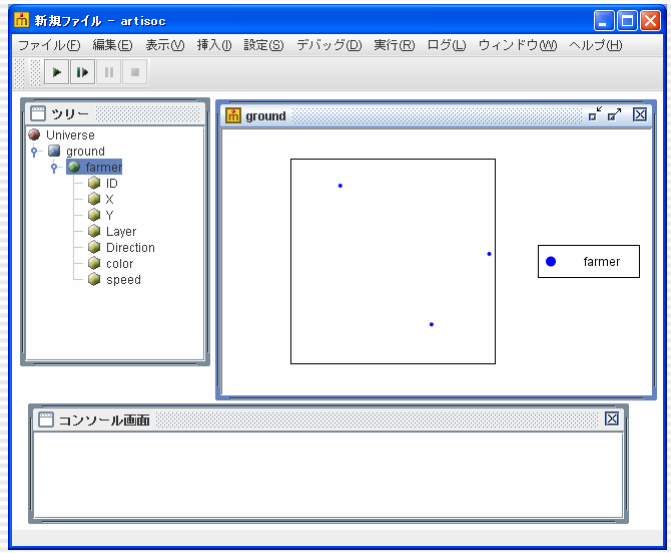
シミュレーションを実行します

■シミュレーションの実行と停止を操作します。

- 「ステップ実行ボタン」を何回かクリックして、ステップ毎の動きの違いを確認します。
- 「実行ボタン」をクリックしてシミュレーションを実行します。
- 「停止ボタン」をクリックしてシミュレーションを停止します。

※ エージェントの数を100にしてみましょう。





tutorial1.model

エージェント同士を相互作用させよう！

ぶつかると赤くなるエージェント

- エージェントをまとめて扱うときは、エージェント集合型という変数を使うことができます。
- 周りを見回して隣人を探します。

ループしない空間とランダムな配置

- 空間がループしない場合は端点の処理が必要になります。
- エージェントを空間上にランダムに配置します。

エージェントに個性を持たせる

- 違う色、違う歩く速度を持つエージェントを定義します。
- 会話をしているエージェントを線で結びます。

ぶつかると赤くなるエージェント

■隣人(neighbor)を定義して、エージェント同士を相互作用させます。

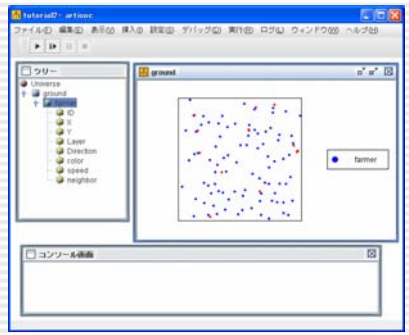
- ツリーの「farmer」で右クリックして、「変数の追加」を選択します。
  - 変数名:　neighbor
  - 変数の型:　エージェント集合型

- ツリーの「farmer」で右クリックして、「ルールエディタ」を選択します。

```
Agt_Step{
        Forward(My.speed)

        MakeAllAgtSetAroundOwn(My.neighbor, 1, False)
        If CountAgtSet(My.neighbor) > 0 Then
                Turn(Rnd() * 360)
                My.color = COLOR_RED
        Else
                My.color = COLOR_BLUE
        End If
}
```

周りを見回して隣人を取得

隣人の数をカウントして
ぶつかっていたら赤くなる



tutorial2.model

ループしない空間とランダムな配置（1）

■ループしない空間を定義します。

- ツリーの「ground」で右クリックして、「プロパティ」を選択します。
  - 端点の処理： ループしない

- ツリーの「Universe」で右クリックして、「ルールエディタ」を選択します。

```
Univ_Init{
        Dim myAgtSet As AgtSet

        MakeAgtSet(myAgtSet, Universe.ground.farmer)
        RandomPutAgtSet(myAgtSet)
}
```

変数の定義

farmerの集合を取得

farmerをランダムに配置

- ツリーの「farmer」で右クリックして、「ルールエディタ」を選択します。

```
Agt_Init{
        My.color =COLOR_BLUE
        My.speed = 1
        Turn(Rnd() * 360)
}
```

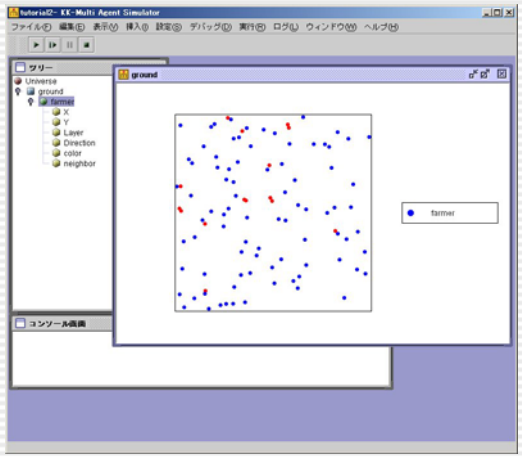
ループしない空間とランダムな配置（2）

■壁にぶつかったときの処理を加えます。

●ツリーの「farmer」で右クリックして、「ルールエディタ」を選択します。

```
Agt_Step{
        If Forward(My.speed) <> -1 Then
                Turn(Rnd() * 360)
        End If

        MakeAllAgtSetAroundOwn(My.neighbor, 1, False)
        If CountAgtSet(My.neighbor) > 0 Then
                Turn(Rnd() * 360)
                My.color = COLOR_RED
        Else
                My.color = COLOR_BLUE
        End If
}
```

壁に当たったら向きを変える



tutorial3.model

エージェントに個性を持たせる（1）

■エージェント毎に異なった色「color」、異なった歩く速度「speed」を持ちます。

- ツリーの「farmer」で右クリックして、「変数の追加」を選択します。
  - 変数名：　touch
  - 変数の型：　整数

- ツリーの「farmer」で右クリックして、「ルールエディタ」を選択します。

```
Agt_Init{
        My.color = RGB(CInt(Rnd()*256), CInt(Rnd()*256), CInt(Rnd()*256))
        My.speed = Rnd()
        Turn(Rnd() * 360)
}
```

ランダムに色指定

歩く速度をランダムに定義

エージェントに個性を持たせる（2）

■色「color」と歩く速度「speed」と接触回数「touch」をルールに組み入れます。

●ツリーの「farmer」で右クリックして、「ルールエディタ」を選択します。

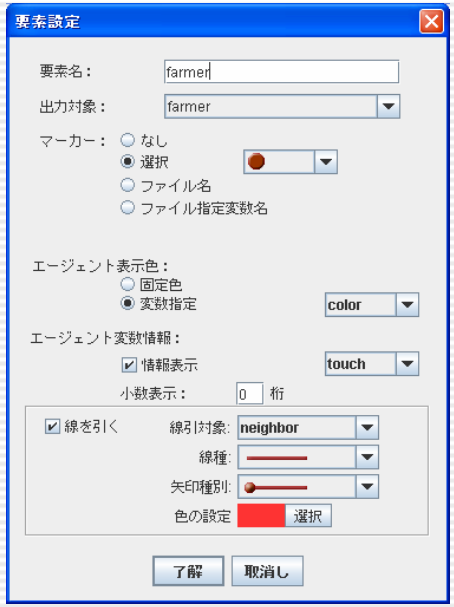
```
Agt_Step{
        If Forward(My.speed) <> -1 Then
                Turn(Rnd() * 360)
        End If

        MakeAllAgtSetAroundOwn(My.neighbor, 3, False)
        If CountAgtSet(My.neighbor) > 0 Then
                My.touch = My.touch + 1
                Turn(Rnd() * 360)
        End If
}
```

視野を3にする

接触回数を加算

エージェントに個性を持たせる（3）

■出力項目の設定を行います。

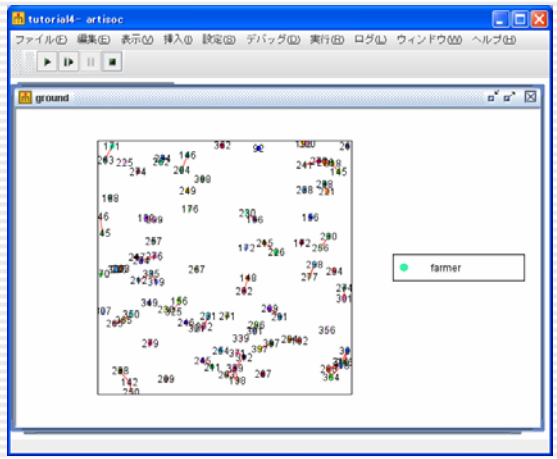


● ツールバーの[設定]-[出力設定]を選択します。

● 出力項目リストが表示されますので、「ground」を選択して「編集」ボタンをクリックします。

● マップ要素リストが表示されますので、「farmer」を選択して「編集」ボタンをクリックします。

- エージェント変数情報：
  - 情報表示： touch
- 線を引く
  - 線引き対象： neighbor
  - 矢印種別：
  - 色の選択： 赤



tutorial4.model

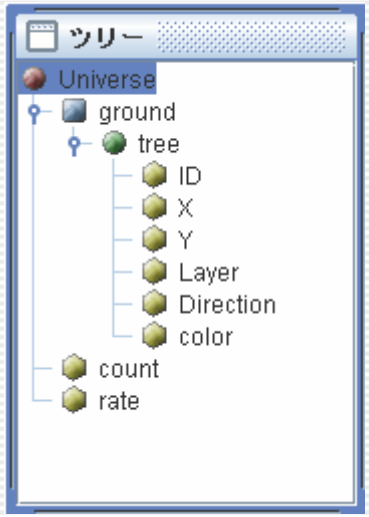
本格的なモデルへのいざない

■森林火災モデルを作る

- コントロールパネルを利用します。
- コントロールパネルから取得した数だけエージェントを生成します。
- エージェントに記憶を持たせて、火の燃え尽きを表現します。
- シミュレーションの終了条件を定義します。

森林火災モデル（1）

■ツリーで「空間」と「エージェント」と「変数」を定義します。



- ツリーの「Universe」で右クリックして、「空間の追加」を選択します。
  - 空間名： ground
  - 端点の処理： ループしない
- ツリーの「ground」で右クリックして、「エージェントの追加」を選択します。
  - エージェント名： tree
- ツリーの「tree」で右クリックして、「変数の追加」を選択します。
  - 変数名： color
  - 変数の型： 整数型
  - 記憶数： 8
- ツリーの「Universe」で右クリックして、「変数の追加」を選択します。
  - 変数名： count
  - 変数の型： 整数型
  - 変数名： rate
  - 変数の型： 実数型

森林火災モデル(2)

■シミュレーション開始時(Univ_Init)に木(tree)を生成し、火をつけます。

●ツリーの「Universe」で右クリックして、「ルールエディタ」を選択します。

```
Univ_Init{
        Dim i As Integer
        Dim myAgt As Agt
        Dim myAgtSet As AgtSet

        For i = 0 To CInt(GetWidthSpace(Universe.ground) * GetHeightSpace(Universe.ground) * Universe.rate)
                myAgt = CreateAgt(Universe.ground.tree)
                myAgt.color = COLOR_GREEN
        Next i

        MakeAgtSet(myAgtSet, Universe.ground.tree)
        RandomPutAgtSetCell(myAgtSet, False)

        myAgt = GetAgt(myAgtSet, CInt(Rnd() * CountAgtSet(myAgtSet)))
        myAgt.color = COLOR_RED
}
```

空間の大きさに森林密度(rate)をかけた数の木(tree)を生成

treeをランダムに配置

1本の木に火をつける

森林火災モデル(3)

■各ステップの終了時(Univ_Step_End)に終了条件のチェックを行います。

●ツリーの「Universe」で右クリックして、「ルールエディタ」を選択します。

```
Univ_Step_End{
        Dim myAgtSet As AgtSet
        Dim myAgt As Agt
        Dim ctRed As Integer
        Dim ctGreen As Integer

        ctRed = 0
        ctGreen = 0
        MakeAgtSet(myAgtSet, Universe.ground.tree)
        For Each myAgt In myAgtSet
                If myAgt.color == COLOR_RED Then
                        ctRed = ctRed + 1
                ElseIf myAgt.color == COLOR_GREEN Then
                        ctGreen = ctGreen + 1
                End If
        Next myAgt
```

木(tree)の集合を取得

For Each文でエージェント集合から1つずつエージェントを取得する

燃えている木の本数

焼け残っている木の本数

～次ページに続く～

森林火災モデル(4)

■各ステップの終了時(Univ_Step_End)に終了条件のチェックを行います。

●ツリーの「Universe」で右クリックして、「ルールエディタ」を選択します。

～前ページから続く～

```
        // finish condition
        If ctRed == 0 Then
                PrintLn("finish! : GREEN=" & ctGreen & " / ALL=" & CountAgtSet(myAgtSet))
                ExitSimulation()
        End If
}
```

燃えている木の本数が0のとき

焼け残っている木の本数と総数を表示

シミュレーションを終了

森林火災モデル(5)

- ツリーの「tree」で右クリックして、「ルールエディタ」を選択します。

```
Agt_Step{
        Dim myAgtSet As AgtSet
        Dim myAgt As Agt
        Dim ct As Integer

        If My.color == COLOR_RED Then
                MakeAllAgtSetAroundOwnCell(myAgtSet, 1, False)
                ct = CountAgtSet(myAgtSet)
                If ct > 0 Then
                        myAgt = GetAgt(myAgtSet, CInt(Rnd() * ct))
                        If myAgt.color == COLOR_GREEN Then
                                myAgt.color = COLOR_RED
                        End If
                End If
        End If
        If GetHistory(My.color,8) == COLOR_RED Then
                My.color = COLOR_BLACK
        End If
}
```

木が燃えているとき

周りの木を取得する

周りに1本以上の木があるとき

ランダムに1本を選択

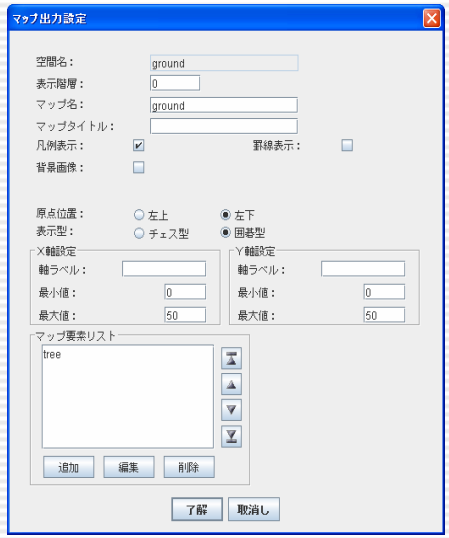
燃えてなかったら燃やす

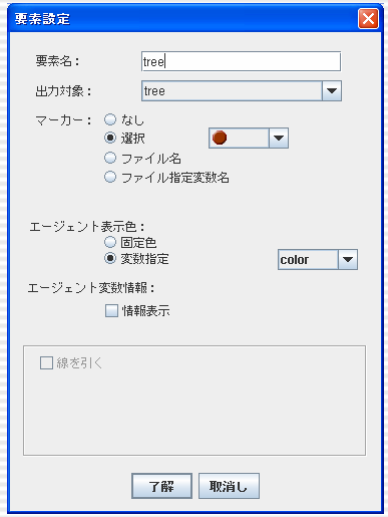
8ステップ前の状態を見て
燃え尽きる

森林火災モデル(6)

■出力項目の設定を行います。

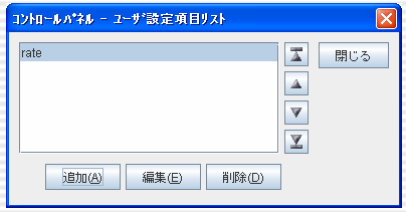
- ツールバーの[設定]-[出力設定]を選択します。
- 出力項目リストが表示されますので、追加する出力種類を「マップ出力」にして「追加」ボタンをクリックします。
  - マップ名： ground
- マップ要素リストで「追加」ボタンをクリックします。
  - 要素名： tree
  - エージェント表示色：
    - 変数指定： color



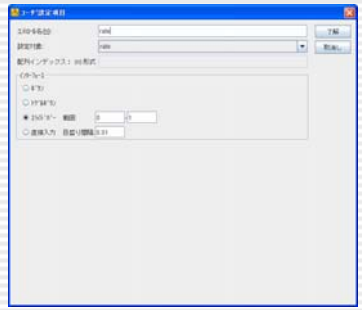


森林火災モデル(7)

■コントロールパネルの設定を行います。



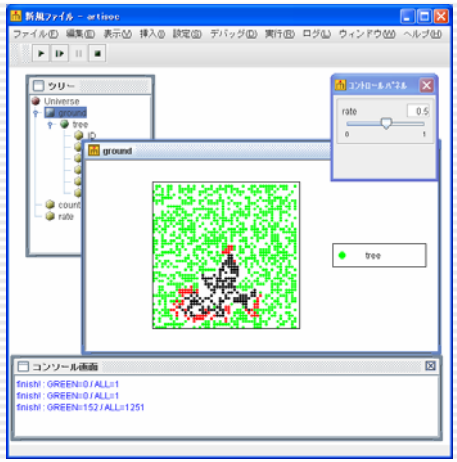
● ツールバーの[設定]-[コントロールパネル設定]を選択します。



● ユーザ設定項目リストが表示されますので、「追加」ボタンをクリックします。

- 設定対象: rate
- インターフェース: スライドバー
  - 範囲: 0 ～ 1
  - 目盛り間隔: 0.01

※ コントロールパネルを操作して、全焼するときの条件を探してみましょう。



forest.model

ログの記録と再生

- シミュレーション結果を分析するために
  ログ保存を行います。

  - 実行メニュー の[記録して実行]を選択します。

  - シミュレーションメモ入力ダイアログが表示
    されるので値を入力します。
    - シミュレーションメモ：　テストログ

  - 「停止ボタン」をクリックしてシミュレー
    ションを停止します。

  - ログメニューの [再生]を選択します。

  - シミュレーション選択ダイアログが表示
    されるので「テストログ」を選択します。

  - ログ再生パネルを利用して再生します。