

artisoc Tutorial

The inquiry : Innovative Information Technology dept.

TEL : 03-5342-1125

E-mail : artisoc@kke.co.jp

Let's simulate the social phenomenon

■ User's friendly multi-agent simulator "artisoc"

- Three steps for multi-agent simulation
- Be mutually interactive the agent.
- Invitation to real model



Three steps for multi-agent simulation

■ The procedure to simulate it with artisoc is as follows.

Operating procedure

- ① “Agent” is defined with the tree.
- ② Agent’s rule is described.
- ③ The output setting is done.

The simulation is executed.

The screenshot displays the artisoc software interface with the following components labeled:

- ① Tree:** A hierarchical tree view on the left showing the agent definition structure, including 'Universe', 'TwoDimensionalSpace', and 'Walker'.
- ② Rule Editor:** A panel below the tree showing the rule definition for the 'Walker' agent, including variables like 'MyWalkerColor' and 'Turn(360*Rnd())'.
- ③ Output Panel:** Two panels on the right: 'Space Map' showing a 2D grid with colored dots representing walkers, and 'The Amount of Red Walkers' showing a line graph of the number of red walkers over time.

① “Agent” is defined with the “Tree”

■ “Space”, “Agent”, and “Variable” are defined in the tree

- Select and right-click “Universe” in the tree, and select “Add Space”

- Space name: ground

- Select and right-click “ground” in the tree, and select “Add Agent”

- Agent name: farmer

- Agent count: 3

- Select and right-click “farmer” in the tree, and select “Add Variable”

- Variable name: color

- Variable type: Integer

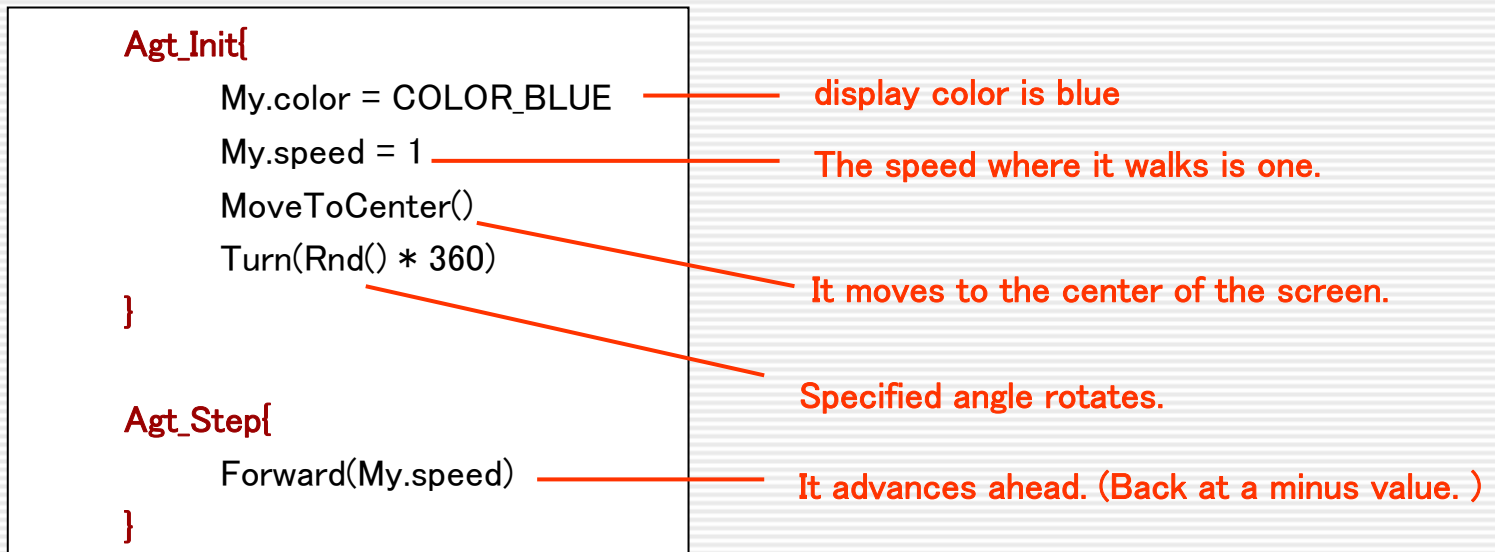
- Variable name: speed

- Variable type: Double



② Agent's rule is described.

- Agent's color and the speed where it walks are specified, it moves to the center of the screen, and it changes my direction.
 - Select and right-click "farmer" in the tree, and select "Rule Editor"



There is no distinction between the capital letter and the small letter.

There may not be space at the head of line.

The candidate of the variable is displayed when describing "My."

When the Ctrl + SPACE key is pushed while several characters of the start of the function name are input, the candidate of the function is displayed.

About details of the functions, please refer [artisoc]-[documents]-[help english] (help-en.pdf) on the Start menu.

③ The output setting is done.

■ The output item is set.

- Select “Settings” > Outputs to open Output Item List.
- Select Map from pulldown menu, and then click the Add button.

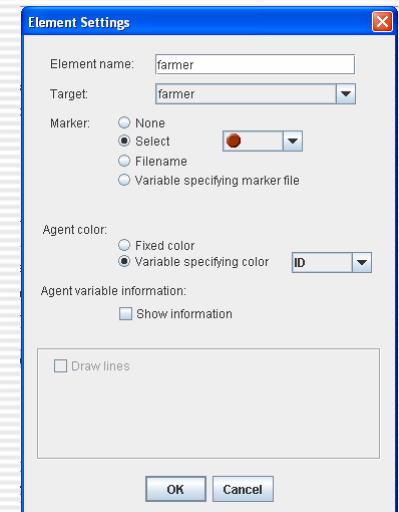
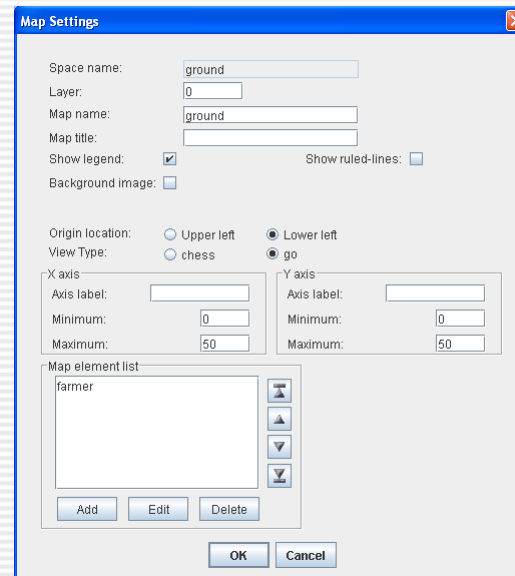
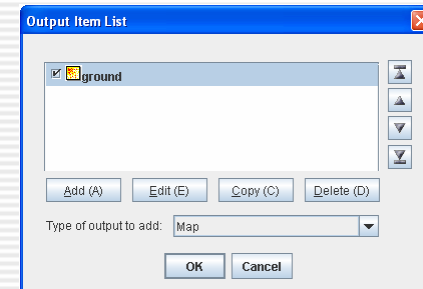
□ Map name: ground

- Select Map element List, and then click the Add button.

□ Element name: farmer

□ Agent color:

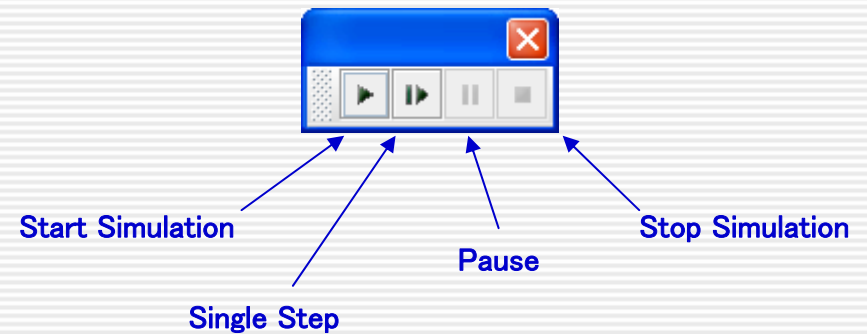
□ Variable specifying color: color



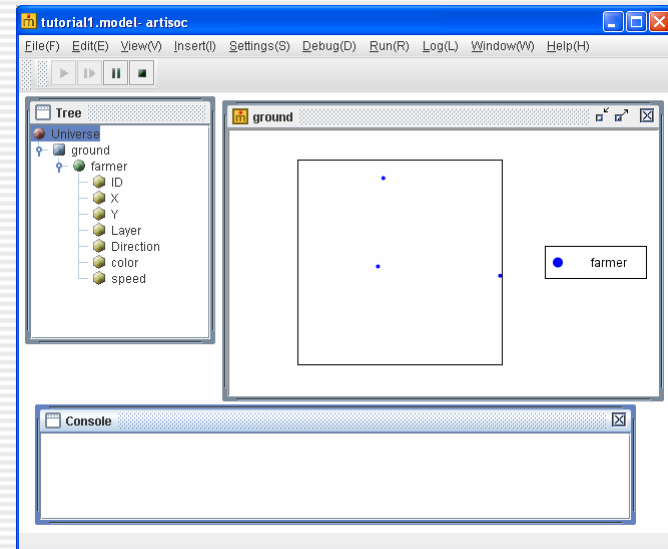
The simulation is executed.

■ Execution and the stop of the simulation

- Click “Single Step” once or twice, and check simulations.
- Click “Start Simulation”
- Click “Stop Simulation”



Let's adjust the number of agents to 100.



tutorial1.model

Be mutually interactive the agent.

■ The agent who reddens when knocking against

- When the agent is collectively treated, the variable of “AgtSet” type can be used.
- It looks for neighbors.

■ Space that doesn't loop and random arrangement

- When the space doesn't loop, the processing of the edge point is needed.
- The agent is arranged on the space at random.

■ The agent has individuality

- The agent who has a different color and a different speed where it walks is defined.
- The agent who is talking is connected in the line.

The agent who reddens when knocking against

■ Neighbors are defined, and the agent is mutually interactive.

● Select and right-click “farmer” in the tree, and select “Add Variable”

□ Variable name: neighbor

□ Variable type: AgtSet

● Select and right-click “farmer” in the tree, and select “Rule Editor”

Agt_Step{

Forward(My.speed)

MakeAllAgtSetAroundOwn(My.neighbor, 1, False)

If CountAgtSet(My.neighbor) > 0 **Then**

Turn(Rnd() * 360)

My.color = COLOR_RED

Else

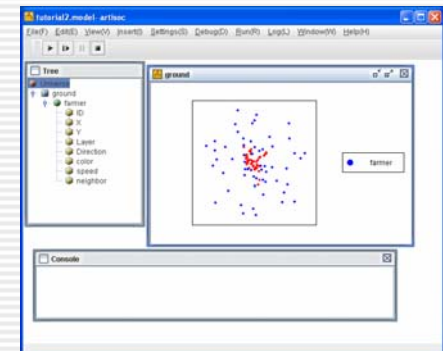
My.color = COLOR_BLUE

End If

}

Neighbors are acquired
looking about surroundings.

The number of neighbors is counted.
It reddens when knocking against.



tutorial2.model

Space that doesn't loop and random arrangement (1)

■ The space that doesn't loop is defined.

● Select and right-click “ground” in the tree, and select “Properties”

□ End of Space: Don't loop (dead end)

● Select and right-click “Universe” in the tree, and select “Rule Editor”

```
Univ_Init{  
    Dim myAgtSet As AgtSet  
  
    MakeAgtSet(myAgtSet, Universe.ground.farmer)  
    RandomPutAgtSet(myAgtSet)  
}
```

Definition of variable

Sets of farmer are acquired.

Farmer is arranged at random.

● Select and right-click “farmer” in the tree, and select “Rule Editor”

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

Space that doesn't loop and random arrangement (2)

- The processing when running up against a stone wall is added.

- Select and right-click “farmer” in the tree, and select “Rule Editor”

```
Agt_Step{
```

```
  If Forward(1) <> -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
```

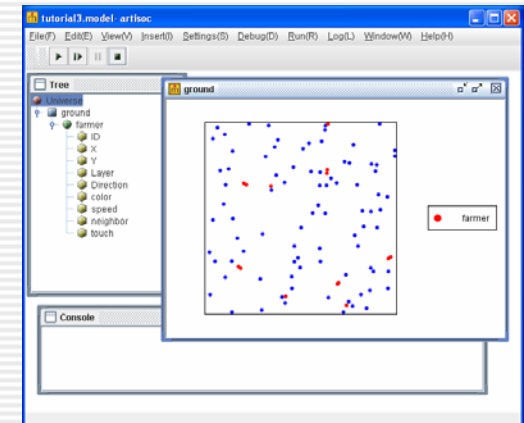
```
  Else
```

```
    My.color = COLOR_BLUE
```

```
  End If
```

```
}
```

When it hits the wall,
it changes front.



tutorial3.model

The agent has individuality (1)

■ It has “color” different in each agent and different “speed” where it walks.

- Select and right-click “farmer” in the tree, and select “Add Variable”
 - Variable name: touch
 - Variable type: Integer
- Select and right-click “farmer” in the tree, and select “Rule Editor”

```
Agt_Init{
```

```
    My.color = RGB(CInt(Rnd()*256), CInt(Rnd()*256), CInt(Rnd()*256))
```

```
    My.speed = Rnd()
```

```
    Turn(Rnd() * 360)
```

```
}
```

It specifies it at random the color.

The speed where it walks is defined at random.

The agent has individuality (2)

- “color” and “speed” and contact frequency “touch” where it walks are put in to the rule.

- Select and right-click “farmer” in the tree, and select “Rule Editor”

```
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
```

```
}
```

View is adjusted to three.

The contact frequency is added.

The agent has individuality (3)

■ The output item is set.

- Select “Settings” > Outputs to open Output Item List.
- Select “ground”, and then click Edit button.
- Select Map element List, and then select “farmer” and click the Edit button.

□ Agent color:

□ Variable specifying color: color

□ Agent variable Information:

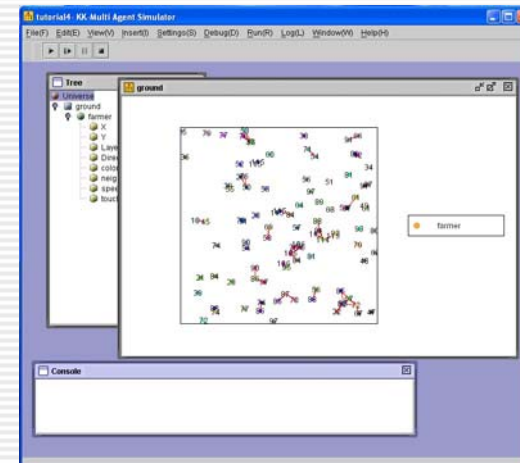
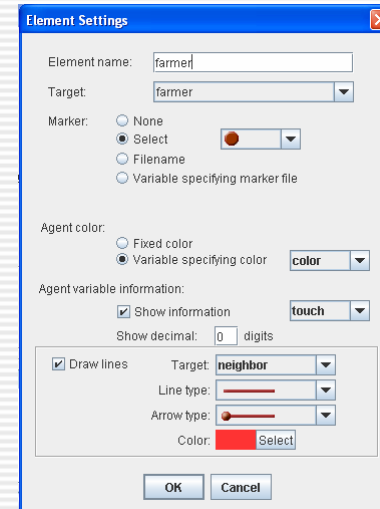
□ Show Information: touch

□ Draw Lines

□ Target: neighbor

□ Arrow type: 

□ Color: Red



tutorial4.model

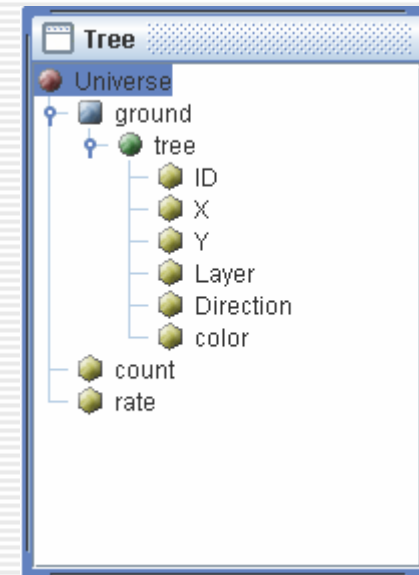
Invitation to real model

■ The forest fire model

- The control panel is used.
- Only the number acquired from the control panel generates the agent.
- The memory is given to the agent, and burning out of the fire is expressed.
- The end condition of the simulation is defined.

The forest fire model (1)

- “Space”, “Agent”, and “Variable” are defined in the tree
 - Select and right-click “Universe” in the tree, and select “Add Space”
 - Space name: ground
 - Select and right-click “ground” in the tree, and select “Add Agent”
 - Agent name: tree
 - Select and right-click “tree” in the tree, and select “Add Variable”
 - Variable name: color
 - Variable type: Integer
 - History size: 8
 - Select and right-click “Universe” in the tree, and select “Add Variable”
 - Variable name: count
 - Variable type: Integer
 - Variable name: rate
 - Variable type: Double



The forest fire model (2)

■ “tree” is generated to “Univ_Init” at the start of the simulation.

● Select and right-click “Universe” in the tree, and select “Rule Editor”.

Univ_Init{

Dim i As Integer

Dim myAgt As Agt

Dim myAgtSet As AgtSet

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

Next i

MakeAgtSet(myAgtSet, Universe.ground.tree)

RandomPutAgtSetCell(myAgtSet, False) ————— “tree” is arranged at random

myAgt = GetAgt(myAgtSet, CInt(Rnd() * CountAgtSet(myAgtSet)))

myAgt.color = COLOR_RED ————— It ignites one tree

}

“tree” of the number in which density (rate) of the forest is put on the size of the space are generated.

The forest fire model (3)

■ The end condition is checked (Univ_Step_End) when each step is ended.

● Select and right-click “Universe” in the tree, and select “Rule Editor”.

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
```

Sets of “tree” are acquired.

The agent is acquired from the agent set
in the For Each sentence one by one.

Number of burning tree

Number of tree that has
remained unburnt

(It continues to next page)

The forest fire model (4)

■ The end condition is checked (Univ_Step_End) when each step is ended.

- Select and right-click “Universe” in the tree, and select “Rule Editor”.

(It continues from last page)

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

When the number of the burning tree is 0

The number and the total of the tree that has remained unburnt are displayed.

The simulation is ended.

The forest fire model (5)

- Select and right-click “tree” in the tree, and select “Rule Editor”.

Agt_Step1

```
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
```

```
}
```

When the tree burns

A surrounding tree is acquired

When there is one tree
or more in surroundings

One is selected
at random.

It burns when
not burning.

The state in front of eight steps
is seen and it burns out.

The forest fire model (6)

■ The output item is set.

- Select “Settings” > Outputs to open Output Item List.
- Select Map from pulldown menu, and then click the Add button.

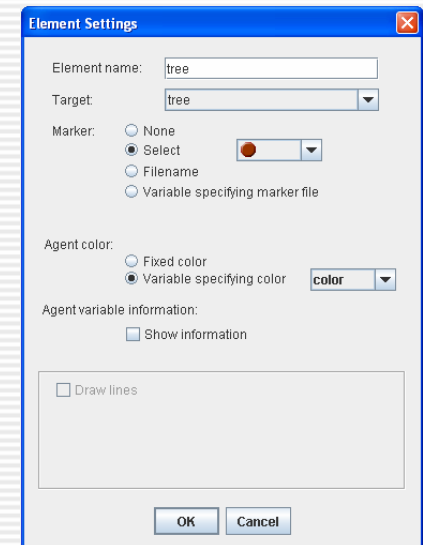
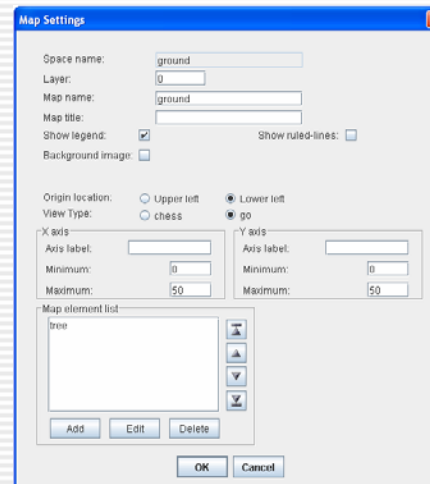
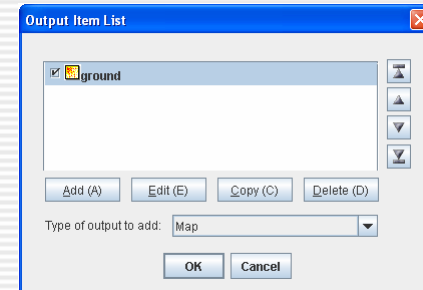
□ Map name: ground

- Select Map element List, and then click the Add button.

□ Element name: tree

□ Agent color:

□ Variable specifying color: color



The forest fire model (7)

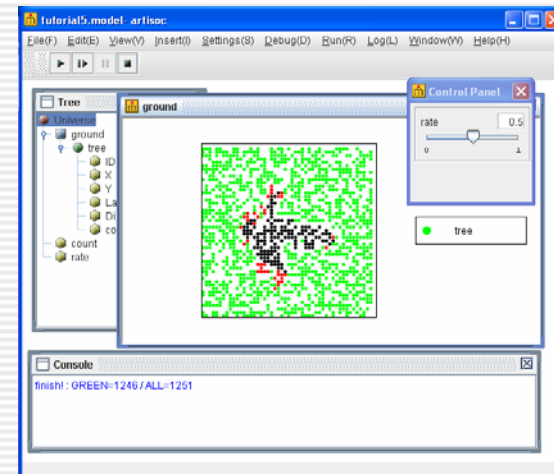
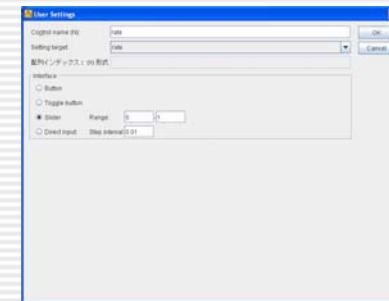
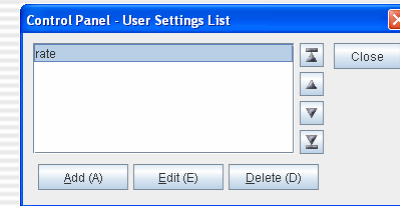
■ The control panel is set.

- Select “Settings” > Control Panel to open User Setting Item List.

- Click the Add button.

- Setting target: rate
- Interface: Slider
 - Range: 0 – 1
 - Step Interval: 0.01

I will look for the condition when burning down by operating the control panel.



tutorial5.model

Replay log

■ To analyze the simulation result, the log is recorded.

- Select “Run”> Run and Save Simulation to open Simulation Memo Dialog.
- Enter simulation memo.
 - “ test log ”
- Click “Stop Simulation”
- Select “Log”> Start Replay to open Replay Simulation Dialog, and Click “OK”.
- Click “Start Replay” on the Log Panel.

