Workshop on World Modeling・Workshop on Methods of Human Security Studies
2005 Summer Semester

Presiding Professor: Susumu Yamakage
TA: Takuto Sakamoto, Kazutoshi Suzuki
Hiroyuki Hoshiro, Katsuma Mitsutsuji
Kazuya Yamamoto

Lecture Six:　God Appears!（May 31st）

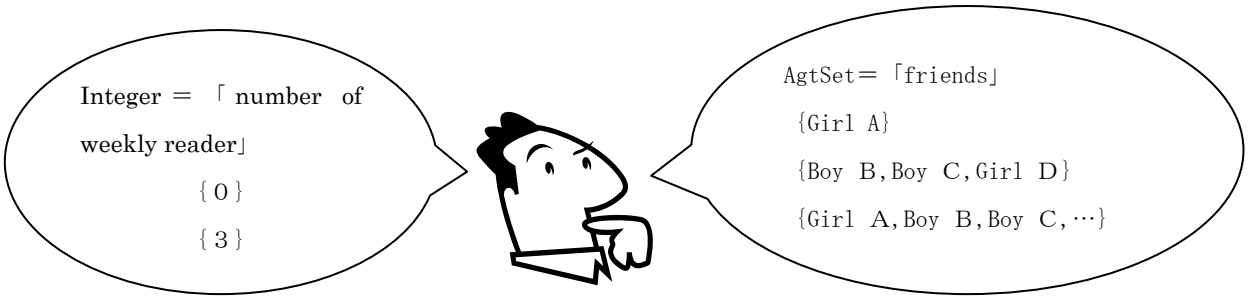
●**Outline**

○Cell Type Model
○Establishment of Control Panel
○For Sentence
○CreateAgt
○RandomPutAgtSetCell
○MakeAgtSet
○MergeAgtSet
○Assignment

●What we have learnt so far.

☆AgtSet variable

In this variable a set of agents can be stored. Integer type variables take the form of 1, 2 and so on and AgtSet variables take the value of a set of agents.



☆AgtSet Function

In KK-MAS there are numerous rules（functions）that handle Agtset variables. Let's learn the following two:

－**MakeOneAgtSetAroundOwn**(AgtSet variable, Scope, Agttype, Include oneself？)
Store the list of agents surrounding oneself inside AgtSet variable.

－**CountAgtSet**(AgtSet Variable)
Count the number of agents stored in AgtSet.

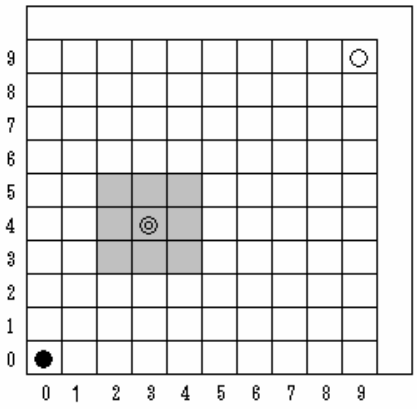
●Present Target

The gist of Multi-Agent Simulation is the interactive communication between agents made possible by the provision of rules of behavior to agents. In order to support this, what is prepared is the Universe. (God, the agent of the model creator) By establishing variables and rules, the Universe is able to process anything and compute the conditions of the whole model. With the present and following lectures, we will learn the methods to process variables and rules while creating segregation models.

●Cell Type Model

In my previous three lectures, the agents freely moved in space. This time we will learn about models where agents exist (move)in boxes created by grids made in space. Imagine the game of Othello or *shogi* or chess. This type of space formation is called a Cell Type.

（1）Coordinates are integers only
（2）Coordinates start from 0
（3）The vicinity shape is different
（4）Modify output scope



☆Instead of Forward(move toward own set Direction) used in former situations, in cell type space, we apply ForwardXCell and ForwardYCell.
ForwardXCell(integer) = proceed ○ boxes towards X axis
ForwardYCell(integer) = proceed ○ boxes towards Y axis

★Let's make a city. This will be the stage for the segregation model. The basics in making it is completely the same as before. The size is 10*10.
★Also prepare agents（red turtles, blue turtles）

●Universe：Use of Rule Editor

By writing instructions in the Rule Editor of the Universe, many things will become possible. The order by which it is done is as follows: Please open the Editor. The Rule Editor of theUniverse is categorized into four sections; Univ_Init{},Univ_Step_Begin{},Univ_Step_End{},Univ_Finish{}.

Univ_Init{}	executed　only once at the beginning of simulation
Univ_Step_Begin{}	executed at start of each step
Univ_Step_End{}	executed at end of each step
Univ_Finish{}	executed only at end of simulation

●Make a Control Panel

This will enable you to alter from the outside the model you are controlling. This Control Panel is equipped with Universe level variables alone.

★Variables that show「number of red turtles」and「number of blue turtles」need to be established in the Universe. And have those set included the Control Panel as well.
（We haven't written any rules yet but just set these）Do the following:
［Set＞Control Panel Set＞Add］

●Grammar Lessons for Today

☆**For sentence**（sentences of repetition）

This is a basic command similar to if sentence（conditional sentence）It is used to repeat a command by only an identified number of times. This is the way to write to repeat rule A ten times.

```
dim i as integer
For i = 0 to 9
        ＜Rule A＞
Next i
```

☆Agent ID（requires a bit of sophistication）

This will be a bit difficult. But since it can be used together with **For sentence**,I will explain about Agent ID. Either it will be a red or blue turtle, inside the tree they come as two categories. But in fact there are numerous（plural number of agents） turtles in existence. In order to distinguish each agent, Agent ID is used.

For example, if there are ten red turtles, it will look like this: RedTurtle(0), RedTurtle(1), RedTurtle(2)…, RedTurtle(9) If you want to operate the X axis of one turtle of the ten red turtles（e.g.no.5）,it can be done in the following way.

Ex. Universe.Streets.RedTurtle.X(5) = 9

By combining this ID and `for` sentence,we can express how the Universe can allot agents （a certain amount）regularly or control them.

```
dim i as integer
For i = 0 to 9
        Universe.Streets.RedTurtle.X(i) = i
        Universe.Streets.RedTurtle.Y(i) = i
Next i
```

☆**CreateAgt**　　Let's create an agent.

We will create one agent of a designated kind. One condition is established.（＝argument）

```
CreateAgt( kind of agent to be created )
```

Ex. CreateAgt(Universe.Streets.RedTurtle)

★With the rules stipulated so far, let's write up the number of red turtles and blue turtles in the control panel and the first rule of theUniverse.

★Let's learn more about AgtSet variables.

☆**RandomPutAgtSetCell**　　agents are allotted at random in a set

This is a function to allot inside space agents which are included in a designated set. The chosen set is a cell, so a Cell is indicated.

RandomPutAgtSetCell(AgtSet variable,　allow duplication （True/False）)

Ex. RandomPutAgtSetCell(Universe.Turtles, False)　　＞set in tree

Ex. dim Turtles as AgtSet

RandomPutAgtSetCell(Turtles, False)　　＞set in rule

（There is the need to declare this in the rule beforehand （at the onset））

☆**MakeAgtSet**

We will store a list of all agents of a designated description in AgtSet variable. Last week we learnt about AgtSet function; MakeOneAgtSetAroundOwn This was a variable that stored a list of agents surrounding oneself in AgtSet variable. This function that can store inside a variable, all agents of a designated class description without any conditioning ; MakeAgtSet.

MakeAgtSet(AgtSet variable in which collected agents are inserted, agent kind)

Ex. MakeAgtSet(Universe.Turtles, Universe.Streets.RedTurtle)　　＞set in tree

Ex. dim Turtles As AgtSet

MakeAgtSet(Turtles, Universe.Streets.RedTurtle)　　＞set in rule

☆**MergeAgtSet**　　Combine AgtSet

This function will let you add to a certain AgtSet variable, a list of agents who belong to other AgtSet. Two conditions will be established The list of agents in .variable 2 can be stored in variable 1

MergeAgtSet(AgtSet variable 1, AgtSet 2)

Ex. MergeAgtSet(Universe.Turtles1, Universe.Turtles2)　　＞set in tree

Ex. dim Turtles1 As AgtSet

dim Turtles2 As AgtSet

MergeAgtSet(Turtles1, Turtles2)　　＞set in rule

★Using the rule set so far, let's write a rule to randomly scatter, inside space, the red and blue turtles that have been created.

★This completes the model that will be the foundation of the segregation model.

●Assignment

＜Agenda Initial Model＞ is prepared with 10×10 cell type space and ten turtles.

（1）Try to have ten lined up on the left hand side.

（2）Try to have the ten lined up diagonally

（3）Have the ten march.

（4）☆Let's write a rule that, if a turtle comes to either the right or the left edge, that he will make a turn

（5）At first, make it that the turtles are randomly allotted.

（6）Establish that the turtles will go right or left as well as move vertically or horizontally as they wish.

（7）☆Let's write a rule that if there are more than three turtles in the vicinity ,he must stay in that place. ［ MakeOneAgtSetAroundOwnCell 〈 AgtSet variable,Range,Agt kind,include myself or not〉］