**ABM – Week 3 – Seminar**

**Simplified Model – Full Guidance**

**Model**

In this task, you will create a much simplified version of the Marriage Model, from the ODD description provided by Railsback & Grimm (2011), which is available on Moodle.

**Task**

Create a model where…

* Turtles have variables representing their age, gender, marital status (married or not) and their marital partner (either another turtle or ‘nobody’). Turtles can only be married if they are over some minimum age.
* Turtles have a random y-coordinate, while their x-coordinate represents their age.
* At each tick…  
    
  a) … a certain number of new turtles are born;  
  b) … all turtles aged over 61 die;  
  c) … *[HARDER]* one randomly selected unmarried turtle gets married to a randomly selected partner.***\****
* Turtles’ gender and marital status is communicated by their colour.
* Create a histogram to monitor how many turtles are married in ten-year age brackets.

***\**** *The original model from the literature required marriages to be between turtles of opposite genders, but feel free to relax this criterion (which makes the model easier) or to require same-gender marriages instead. If you wish, you could give each turtle an individual preference for male or female partners, but that would increase the difficulty of the task.*

**Step-By-Step Guide**

*After each change you make, check that the code runs before you move on to the next step…*

Listing the Variables

1. Add variables to globals to represent the minimum age of marriage, the number of new turtles born at each tick and the initial population.
2. Add variables to turtles-own to represent a turtle’s age, gender, marital status (married or not) and their marital partner.

Setting up the Simulation

1. In “Settings…” set the origin to the bottom left corner with the max-pxcor = 60 and the max-pycor = 59.
2. In the setup procedure, set the values of the minimum age of marriage (e.g. 16), the number of new turtles born at each tick (e.g. 3) and the initial population (e.g. 180) *[Useful primitives: set]*

Creating the agents

1. In the setup procedure, create the appropriate number of turtles.  
   *[Useful primitives: crt]*
2. In the create turtles block:
   1. Set the age of each turtle to a random number between 0 and 59.  
      *[Useful primitives: set, random]*
   2. Set the gender of each turtle to be either “m” or “f”.

*[Useful primitives: set, one-of]  
HINT: You will need to include the set* ["m" "f"]*…*

* 1. Set the y-coordinate of each turtle to that of a random patch. and the x-coordinate of each turtle equal to their age

*[Useful primitives: setxy, random-pycor]*

* 1. Set the marital status of each turtle to false.

*NOTE: The marital partner variable will automatically be set to nobody.*

Now, when you click the setup button, the world should fill up with turtles. If you inspect a turtle, you should see its age, gender and marital status, and the age should correspond to its x-coordinate.

Adding a behaviour: Ageing

1. Create a new procedure to model turtle ageing *[e.g. to ageing … end]*. The procedure should be empty for the moment. Add the name of this procedure to the go procedure, so that it runs every tick.
2. In the new procedure, create an “ask turtles” block to move into turtle context.

Inside the “ask turtles” block…

1. … make the age of each turtle increase by 1. *[Useful primitives: set]*
2. … make turtles die if their age exceeds 60  
   *[Useful primitives: if, die]*

1. … reposition each turtle according to its new age.  
   *[Useful primitives: set, xcor.]*

Now, when you run your simulation, turtles should move across the world from left to right, disappearing when they reach the right hand side of the display. Soon there will be no turtles remaining and the simulation may end with an error message.

Adding a behaviour: Marriage

This part of the program is more tricky. The main considerations are:

* Making sure that the chosen turtle is unmarried and of an appropriate age. *(See Step 13)*
* Making sure that they choose a similarly appropriate partner, of the appropriate gender (if you specifically require mixed gender or same gender marriages). *(See Step 14)*
* Ensuring that the chosen turtle and their partner (the more difficult part) update their variables on their marital status and their marital partner.  
  *(See Steps 14-15)*
* Ensuring that these variables are updated when a turtle’s marital partner dies.

1. Create a new procedure to model marriage *[e.g. to marriage … end]*. The procedure should be empty for the moment. Add the name of this procedure to the go procedure, so that it runs every tick.
2. In the new procedure, create an ask block for a randomly chosen unmarried turtle of marriageable age. The block should be empty for the moment  
   *[Useful primitives: ask, one-of, turtles, with, and]*
3. Inside the ask block, set the turtle’s marital status to true and set their marital partner to be a randomly chosen unmarried partner of marriageable age and of an appropriate gender.  
   *[Useful primitives: set, one-of, turtles, with, and, of, myself]  
   HINT: If you are specifying the gender of the partner, you may need one of the following code extracts:*(gender != ([gender] of myself)) *or*(gender = ([gender] of myself))
4. Still inside the ask block, ask the turtle’s partner to set their marital status to true and set their marital partner to be the originally chosen turtle.  
   *[Useful primitives: ask, set, myself]  
   HINT: ‘myself’ refers back to the original turtle (the agent that executed the most recent ‘ask’ command.)*
5. In the ageing procedure, immediately before the ‘die’ command, ask a dying turtle’s partner to reset their marital status to false and their partner to nobody.  
   *[Useful primitives: ask, set, nobody]*

Now, when you run your simulation, turtles should get married and remember their partners, though you will so far only be able to check this by inspecting them.

Colouring turtles by marital status (and gender, if desired)

1. This is left as an exercise. *[Useful primitives: ifelse, set, color]*

Make sure that turtles set their colour in all the following circumstances:

* When the simulation is initialised with the setup command.
* When new turtles are born.
* When they get marries.
* When their marital partner dies.

Once you have added the code for turtle colours, you should be able to observe marriages in your model more easily.

Creating a histogram of marriage numbers by age

1. Create the following two procedures:

to setup-plotting

set-current-plot "Histogram of Married Individuals by Age"

set-plot-x-range 0 60

set-plot-y-range 0 20

set-histogram-num-bars 6

end

to plotting

set-current-plot "Histogram of Married Individuals by Age"

histogram [age] of (turtles with [married? = true])

end

Add setup-plotting to the setup procedure and plotting to the go procedure. Add a plot object to the interface, with the title “Histogram of Married Individuals by Age”.