Agent-Based Modelling
NetLogo Self Assessment

The following tasks cover a minimum of knowledge required to successfully complete the final assessment for the ABM module. Work through them to assess which you can complete and report your performance via the Moodle survey by 17:00 on Friday 01 March.

In most cases, you should be able to skip a task that you cannot complete to attempt those that follow. Try to do this where possible.

To complete these tasks, you should use the models that you have seen and built so far, along with any online help and resources you need. The goal is not to complete everything without looking anything up.

(Note that the final model will be somewhat similar to the Sugarscape models, though considerably simpler. It is not intended to represent anything in particular, just to test key knowledge.)

1. Create the basic Netlogo program structure *(i.e. the basic procedures and commands that appear in every Netlogo program we have looked at)*.

2. Set the grid size to -10 to +10 on both axes and turn off horizontal and vertical wrapping.

3. Make the simulation stop after 100 ticks.

4. Create 100 turtles.

5. Place the turtles at random locations.

6. Set the colour of the turtles to red.

7. Create global variables called turtle-maximum and patch-maximum.

8. Set turtle-maximum to 2 and patch-maximum to 10.

9. Give the turtles variables called t-var-one and t-var-two.

10. Give the patches a variable called p-var.

11. For each turtle, set t-var-one to a random value between 1 & turtle-maximum.

12. For each turtle, set t-var-two to a random string, either "yes" or "no".

13. For all patches, set p-var to the value patch-maximum.

14. Write a procedure called "move", which makes the turtles step from patch to patch (neighbouring) at random.

15. Write a procedure called "eat", to activate after turtles move. If the p-var of the patch a turtle is on is at least one, the procedure should reduce the p-var of the patch a turtle is on by one and increase the value of the turtle's t-var-one by one.

16. Write a procedure called "steal", to activate after turtles eat. The procedure should check whether there are any other turtles on the same patch as the current turtle. If there are, choose one of those turtles and reduce its t-var-one by two. Also increase the t-var-one of the current turtle by two.

17. Write a procedure called "death", to activate after turtles steal. All turtles whose t-var-one is 0 or less should die (be removed from the simulation).

18. Write a reporter called yes-average that returns the mean of t-var-one for all turtles whose t-var-two is equal to "yes".

19. Add a plot that tracks the progress of yes-average over time.

20. At the end of the simulation, print the total population of turtles.

21. At the end of the simulation, colour the patches in shades of green, based on the value of p-var.

22. Rather than setting turtle-maximum and patch-maximum to specific values, create sliders for each of them in the Interface. Allow them to range from 1 to 10 and 1 to 20 respectively.

23. Use BehaviorSpace to conduct an experiment, allowing each of turtle-maximum and patch-maximum to range over 5 different values. Conduct 10 replicates for each combination of parameters. Record yes-average at the end of each simulation.