Agent based modelling

11/7/2025

100 Points Possible

Attempt 1

11/7/2025

NEXT UP: Review Feedback

Attempt 1 Score: N/A

Add comment

Unlimited Attempts Allowed

11/21/2025

Details

Additional information:

- Please familiarise yourself with **regulations governing assessment offences including Plagiarism and Collusion (https://www.herts.ac.uk/__data/assets/pdf_file/0007/237625/Structure-and-Assessment-Regulations-Undergraduate-and-Taught-Postgraduate-Programmes-AS14-Apx-3-Academic.pdf)** (UPR AS14).
- Engage with guidance for avoiding plagiarism via the **University SkillUp module (https://ask.herts.ac.uk/library-skillup)** and **1-2-1 sessions with our specialist SASH team (https://herts.instructure.com/courses/100562/pages/dot-dot-dot-a-one-to-one-session-to-talk-about-my-studies-slash-assignments?module_item_id=2430174)**.
- For **postgraduate modules**:
  - a score of 50% or above represents a pass mark.
  - late submission of any item of coursework for each day or part thereof (or for hard copy submission only, working day or part thereof) for up to five days after the published deadline, coursework relating to modules at Level 7 submitted late (including deferred coursework, but with the exception of referred coursework), will have the numeric grade reduced by 10 grade points until or unless the numeric grade reaches or is 50. Where the numeric grade awarded for the assessment is less than 50, no lateness penalty will be applied.
- In this assessment**, the student is expressly prohibited** from (i) using GenAI tools for the creation of content and (ii) using GenAI tools or a proofreader or a proofreading service for proofreading.

This Assignment assesses the following module Learning Outcomes (from Definitive Module Document):

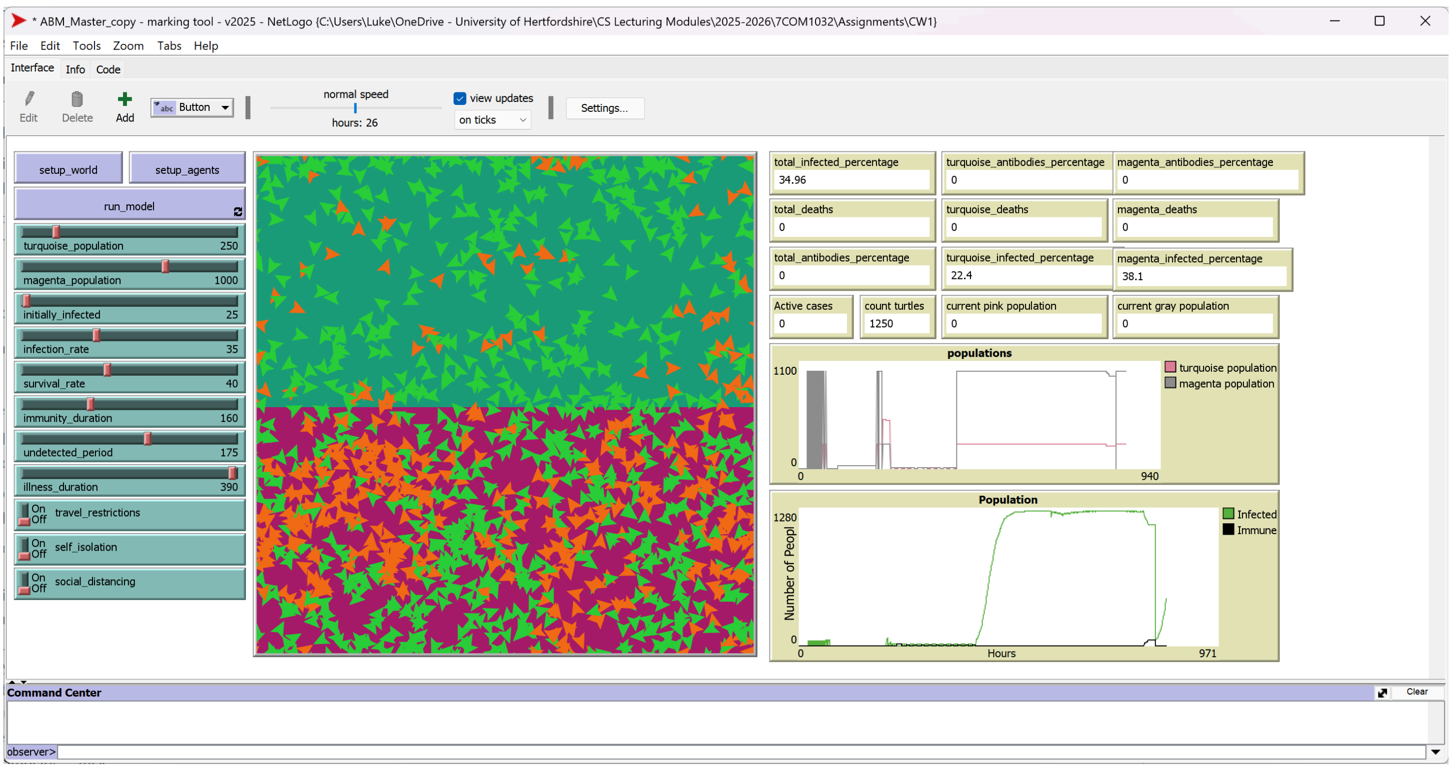
1. Demonstrate knowledge and understanding of a variety of Artificial Life techniques and methods applicable across domains including computer science and robotics.

2. Demonstrate the ability to critically evaluate Artificial Life systems.

3. Demonstrate the ability to implement an Artificial Life system according to a defined specification.

Agent Based Modelling

In this project you will be developing a model to ascertain important factors the impact the spread of an infectious disease among a population of agents within a particular timeframe. You will have 2 populations of agents that are infected with a virus which will spread among the populations. You will analyse factors that impact the spread and mortality rate such as population density, social distancing, self-isolation and the amount of time the virus can go undetected whist transmissible.



Assignment Brief:

In this project you must develop a model in order to ascertain the spread of an infectious disease among a population of agents within a particular timeframe. The purpose of this model is to analyse the following aspects of a disease:

- Assess the impact of agents staying local within a designated region vs free movement
- Assess the impact of social distancing between agents vs not social distancing
- Assess the impact of individuals who are infected self-isolating vs not self-isolating

Since this is meant to provide a relatively realistic representation of a real world situation, the criteria for this model is clearly outlined below.

You must design simulation in NetLogo that adheres to the following requirements:

1. **Overall Model Requirements (5 marks)**
    - Your model MUST run without error. If your model crashes you will get no marks.
    - The following code must be placed at the beginning of your model (copy and paste this in):

```
extensions [csv]

globals[
  ;Variables to be used for tutor marking
  student_id
  student_name
  student_score
  student_feedback

  ;Variables for your analysis
  most_effective_measure
  least_effective_measure
  population_with_highest_mortality_rate
  population_most_immune
  self_isolation_link
  population_density

  total_infected_percentage
  turquoise_infected_percentage
  magenta_infected_percentage

  total_deaths
  turquoise_deaths
  magenta_deaths

  total_antibodies_percentage
  turquoise_antibodies_percentage
  magenta_antibodies_percentage


  stored_settings
]

turtles-own[
  infected_time
  antibodies
```

```
  group
]
```

- Your submitted file must not use any of the following commands:
  - clear-all
  - clear-globals
  - clear-ticks
  - clear-drawing
  - clear-all-plots
  - clear-output
  - stop
  - show
  - print
  - write

If any of these commands are found in your model this will result in a 5 mark deduction per command found.

- The default turtles must be used to represent the agents in the model, no additional breeds can be created.
- You must not use the global variables student_score or student_feedback at all in your model, if this is found in your model this will result in a 5 mark deduction.

1. The world must be 101 patches wide by 101 patches high.

- The origin of the world must be at the centre (center).
- The world must wrap both horizontally and vertically.

2. The patch size must be set to 5.
3. The framerate must be set to 20 frames per second.

1) World setup (5 marks)

The model must have a function called setup_world that initiates the following actions in your model when called:

- The tick counter must be reset
- All turtles must be cleared from the world

1. The world must be split into 2 distinct halves which are both of an (almost) equal size that are one continuous body of color (see example image). One half must be identified with the patches being set to the color turquoise whilst the other half must be set to the color magenta.

- The global variables student_id and student_name must be set to your student ID number and full name respectively.
- The model must contain the following global variables (these can be defined in the code or using the interface) and be set to the following values when submitted:

1. turquoise_population = 250
2. magenta_population = 1000
3. initially_infected = 25
4. infection_rate = 35
5. survival_rate = 40
6. immunity_duration = 160
7. undetected_period = 175
8. illness_duration = 390
9. travel_restrictions = false
10. social_distancing = false
11. self_isolation = false

- The setup_world function must not call the initialise_agent function or the run_model function.

2) Agents setup (10 marks)

The model must have a function called setup_agents that initiates the following actions in your model when called:

turquoise population

- Create a population of turtles that are set to the color lime, size 4, and are randomly placed on the area of turquoise patches. The number of turtles created must be determined by the variable turquoise_population. All of these agents must be initialised with their own antibodies variable set to 0 and their group to “turquoise turtle”. NOTE: antibodies must always be an integer value.

magenta population

- Create a population of turtles that are set to the color lime, size 4, and are randomly placed on the area of magenta patches. The number of turtles created must be determined by the variable magenta_population. All of these agents must be initialised with their own antibodies variable set to 0 and their group to “magenta turtle”. NOTE: antibodies must always be an integer value.

Initially infected individuals

- Create a number of infected turtles in each population (turquoise and magenta) determined by the global variable initially_infected. For example if initially_infected is set to 15 there would be 15 individuals in the turquoise population infected and 15 individuals in the magenta population infected.

- The setup_agents function must not call the setup_world function or the run_model function.

3) Running requirements (50 marks)

The model must have a function called run_model that initiates the following actions in your model when called:

- The **tick** command must be called to add 1 to the tick counter.

Turtle Behaviours

The turtles in your model must behave according to the following parameters:

- The speed at which turtles move must be set to 0.3 per step taken and they must always move in a forwards direction (relative to their heading).
- The orientation (heading) of turtles must be set randomly when wandering but must be limited to a range of 40 degrees (20 to the left and 20 to the right) per step taken, the only exception to this is when avoiding other agents, in this instance a maximum turn of 90 degrees is permitted.
- Turtles must follow the rules of the global variables travel_restrictions, social_distancing and self_isolation as follows:
- **Travel Restrictions**

If the travel_restrictions variable is set to true the turtle must stay within its own region (turquoise for turquoise turtles, magenta for magenta turtles), if the turtle is not within its own region (i.e. the variable was changed mid-simulation) the turtle must take the shortest route back to its own region following any other conditions instated (i.e. social_distancing or self_isolation).

- **Social distancing**

If the social_distancing variable is set to true the turtle must keep a minimum distance of 1 patch between itself and another agent when moving (i.e. check to see if another agent is in front before moving forward, a strategy to avoid collisions).

- **Self-isolation**

If the self_isolation variable is set to true the agent must stop moving and turn blue to indicate it is in self-isolation when infected after the limit undetected_period (i.e. the illness_duration - undetected_period).

- Agents become infected when they are within a radius of 1 of another agent that is infected that is not in self-isolation and does not have antibodies.
- The probability of agents becoming infected by coming into contact with other infected agents must be determined by the global variable infection_rate which must range from 0 to 100 (0 = 0%

chance of becoming infected, 100 = 100% chance of becoming infected).

- Agents who are infected must turn orange to indicate this and also set their infected_time variable (stored in the agents own variable) to the value stated in the global variable illness_duration and reduce by 1 per tick to indicate the passing of time of the illness.
- The survival rate of an agent must be dictated by the variable survival_rate (i.e. if survival_rate is set to 60 this would mean that 60% of the time agents survive the illness)
- If an agent survives an infection at the end of the infected_time the agent must develop antibodies (stored in the agents own variable called antibodies) which must be set according to the global variable immunity_duration and reduce by 1 per tick to indicate the reduction of antibodies over time (again a count down timer). An agent that has antibodies must turn brown while they have antibodies as a visual representation.

- **Model outputs**

Whilst run_model is being called (whist your model is running) you must update the following global variables in real-time (after each tick):

- The global variable **total_infected_percentage** must show the percentage of the current total living population that are infected.
- The global variable **turquoise_infected_percentage** must show the percentage of the current turquoise living population that are infected.
- The global variable **magenta_infected_percentage** must show the percentage of the current magenta living population that are infected.
- The global variable **total_deaths** must be updated on every tick and show the number of turtles who have died in total.
- The global variable **turquoise_deaths** must be updated on every tick and show the number of turquoise turtles who have died in total.
- The global variable **magenta_deaths** must be updated on every tick and show the number of magenta turtles who have died in total.
- The global variable **total_antibodies_percentage** must show the percentage of the current total population that are have antibodies and are immune to infection.
- The global variable **turquoise_antibodies_percentage** must show the percentage of the current turquoise population that are have antibodies and are immune to infection.
- The global variable **magenta_antibodies_percentage** must show the percentage of the current magenta population that are have antibodies and are immune to infection.

4) Your analysis (30 marks)

By the end of your implementation you should have a working model that you can use to perform some analysis. You must have a function called my_analysis that populate a series of global variables with your answers. This must be a self-contained function and not call any other function in your model.

5.1. Most effective measure

Having run the model with the default population over 5,000 ticks in different conditions, what do you think the single most effective measure was for reducing the spread of the virus?

1 = social distancing

2 = self-isolation

3 = stay local

Set the global variable most_effective_measure to one of the 3 answers above (just set the variable to the number).

- **Least effective measure**

Having run the model with the default population over 5,000 ticks in different conditions, what do you think the single least effective measure was for reducing the spread of the virus?

1 = social distancing

2 = self-isolation

3 = stay local

Set the global variable least_effective_measure to one of the 3 answers above (just set the variable to the number).

- **Population with highest mortality rate**

Having run the model with the default populations which population had the highest mortality rate (total number of deaths)?

1 = turquoise

2 = magenta

3 = equally affected

Set the global variable population_with_highest_mortality_rate to one of the 3 answers above (just set the variable to the number).

- **Population with highest immunity**

Having run the model with the default populations which population gains greater immunity?

1 = turquoise

2 = magenta

3 = equally affected

Set the global variable population_most_immune to one of the 3 answers above (just set the variable to the number).

- **Links with self-isolation**

Which global variable has the most impact on the self-isolation measure?

1 = turquoise_population

2 = magenta_population

3 = initially_infected

4 = infection_rate

5 = survival_rate

6 = immunity_duration

7 = undetected_period

8 = illness_duration

Set the global variable self_isolation_link to one of the 8 answers above (just set the variable to the number).

- **Impact of population density**

Running the model with all of the default settings but having the travel_restrictions switched on what happens to the populations after 20,000 ticks?

1 = both populations reduce equally

2 = the turquoise population is completely eliminated

3 = the magenta population is completely eliminated

4 = the virus dies out

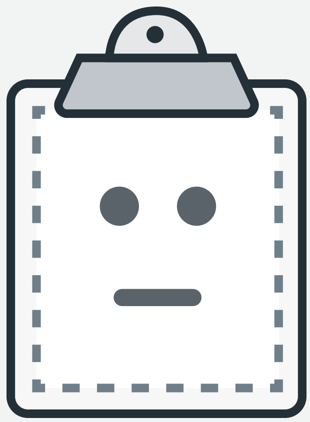
5 = both populations settle to a similar density and the virus dies out

6 = both populations settle to a similar density

Set the global variable population_density to one of the 6 answers above (just set the variable to the number).

What to Submit

This assignment requires the submission of 1 NetLogo file version 6.1.0 or above. The file should be named as follows: <your student id>_ABM.nlogo



Preview Unavailable

24135441_ABM.nlogox

Download (https://herts.instructure.com/files/12263416/download?download_frd=1&verifier=nd7IgfSBly2kbbEOnPS8IXUf9wLV2sPKDFbhJFWW)