A TypeScript Based Agent-based Disease Model

Heyan Zhu (Anson)

Mathmatics & Computer Science Class

heyan.zhu@ulink.cn

July 5, 2024

Abstract

This paper presents an agent-based model implemented in Type-Script, which simulates the spread of diseases. It compares this model with the SIR model.

1 Introduction

An agent-based model is a computer simulation of the interaction of agents (people) that is used to predict and demonstrate the spread of diseases [2]. This paper presents a TypeScript-based agent-based model that can partially simulate the spread of disease.

The code of the model is available on GitHub [1].

1.1 Mathematics theory of spread of disease

In the field of mathematics, there also exists a model known as the SIR model, which is a set of differential equations that describe the number of people who are susceptible (S, not infected), infected (I), and recovered (R, have recovered from the illness and are no longer susceptible to the same disease). The equations of the SIR model are as follows:

---

[1]https://github.com/Anson2251/camp-agent-base-model

$$\frac{\mathrm{d}S}{\mathrm{d}t} = -\beta SI \tag{1}$$

$$\frac{\mathrm{d}I}{\mathrm{d}t} = \beta SI - \gamma I \tag{2}$$

$$\frac{\mathrm{d}R}{\mathrm{d}t} = \gamma I \tag{3}$$
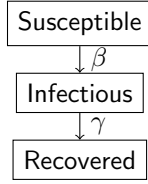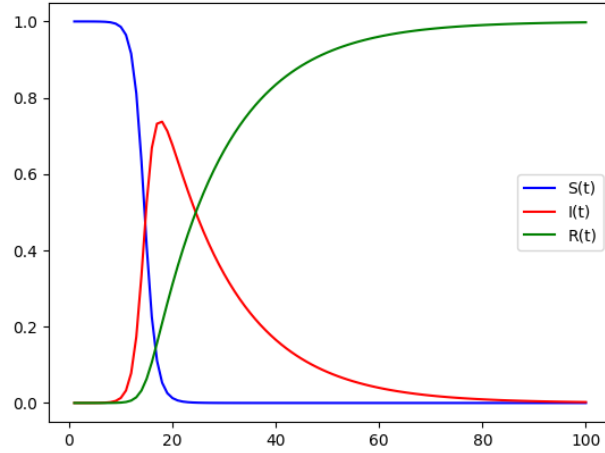

Figure 1: The flow chart[3] of SIR model.

Figure 2: The diagram[1] of SIR model. ($\beta = 1$ and $\gamma = 0.1$)

2 Methodology

2.1 Agent-based Model

In order to construct an agent-based model, it is necessary to define the agents, their behaviours, the interactions between them, and the environment in which the agents operate. The agent-based model was implemented in TypeScript due to the author's familiarity with this programming language. In general, agents are collections of properties, such as name, age, gender, health, and so on. They are used to represent the individuals, playing an

important role in the simulation. The behaviours of the agents in this model represent their daily routines and the locations of the agents at a specific time.

2.2 Environment

The term "environment" is defined as the space in which the agents reside. The dynamics of the agents can be understood as the movement of the agents within this space, which is influenced by their routines.

2.3 Dynamics

2.3.1 Infection Possibility

In order to simulate the spread of the disease, it is necessary to consider how healthy agents interact with infected agents and become infected. If infected agents move in close proximity to healthy agents, the risk of infection will be increased. A simple formula that can be used to roughly represent this is as follows.

$$\mathbb{P}(\text{Infection}) = \begin{cases} \dfrac{1}{d^2} & (\text{for } d > 1) \\ 1 & (\text{for } d \leq 1) \end{cases} \quad (d \text{ is the between the agents})$$

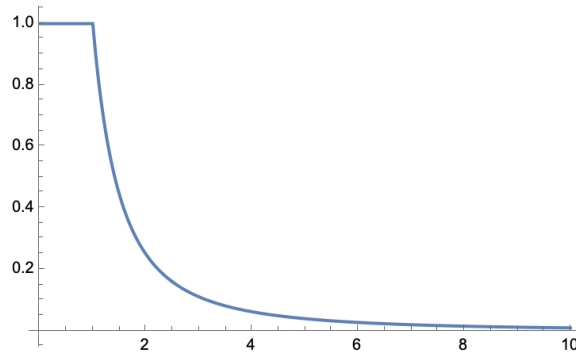The graph of the above formula is shown in the figure below.

Figure 3: The graph of possibility of infection.

3

Where the x is the distance between the agents, calculated by the formula.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Every agents will randomly get infected with a certain probability.

2.3.2 Restoration

Upon infection of the agents, a timer will commence a countdown. Upon reaching zero, the timer will initiate the restoration of the agents. It can be assumed that the restored agents will not become infected by the same disease.

2.4 Implementation

This model is implemented in TypeScript for the purpose of type safety, which enables the model to be well structured in a logical and comprehensible manner. However, the graphing of the model is implemented in R due to the availability of the powerful plotting package `ggplot2`.

The model's graphical representation (Fig. .2), which illustrates the design and the manner in which the modules interact with one another, can be found in the appendix. The flow chart of the model (Fig. .3) is also included in the appendix.

3 Results

The report generated by the model is appended in the Appendix. The graphic part of the report and the parameters are shown below.

- Simulation Time: 60

- Agent Number: 40

- Room Size: 40

The disease which in this simulation model:

- Disease Name: Cold

- Disease Severity: 100

- Disease Time to Restore: 10
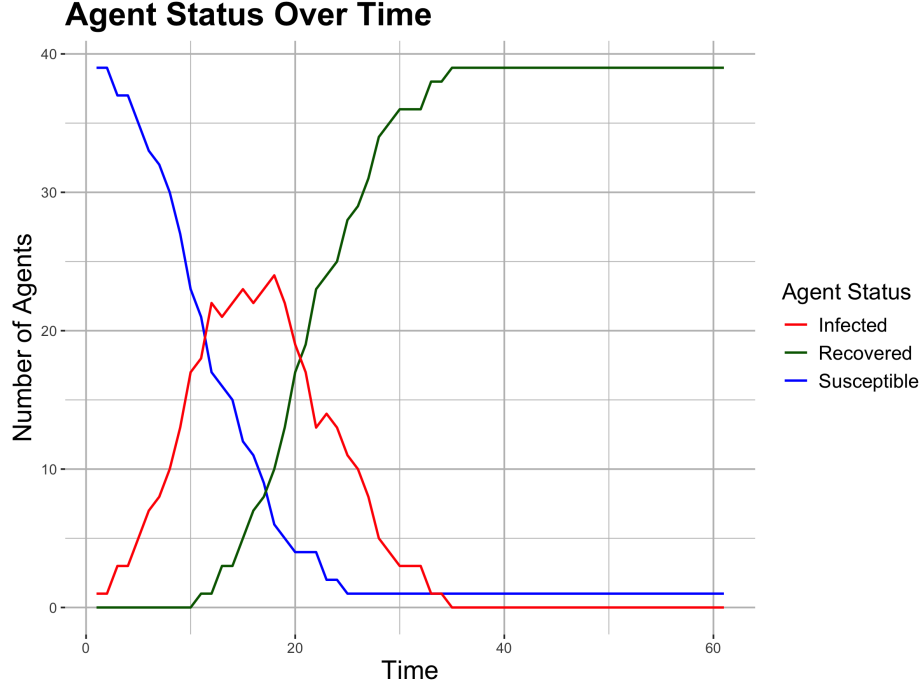
Agent Status Over Time

Figure 4: The diagram of the created agent-based model

It is evident that this model is capable of partially simulating the spread of the disease with given parameters in relation to the agents and diseases, due to the similarity between the previous figures (Fig.1.1) and this figure (Fig.3), which depict outbreaks and extinctions.

4 Discussion

This model is an effective tool for simulating the simple spread of the disease. It has been developed in TypeScript, a language chosen for its maintainability, but it lacks the ability to simulate the multiple diseases with complex dynamics. Some features that could facilitate the migration and control of parameters such as β and γ are missing and should be added in the future. And this model use recover time to simulate, not the possibility to recover for the agents. The author has chosen to write in TypeScript, a language

5

that is not commonly used for data analysis, due to its familiarity. It would be preferable if the model could be implemented in Python or R, which is more commonly used for data analysis.

5 Conclusion

The model described above is capable of accurately simulating the spread of the disease, including the occurrence of outbreaks and extinctions.

6 References

[1] Joaquin de Castro. *Graphing The SIR Model With Python.* `https://pub.towardsai.net/graphing-the-sir-model-with-python-e3cd6edb20de`. Accessed: 2024-07-01. 2020.

[2] Columbia University Irving Medical Center. *Agent-Based Modeling.* `https://www.publichealth.columbia.edu/research/population-health-methods/agent-based-modeling`. Accessed: 2024-07-01. 2024.

[3] Le Nguyen. *Simulation 102: Agent-Based Disease Modeling.* `https://medium.com/@ln8378/simulation-102-agent-based-disease-modeling-17307f2de2f3`. Accessed: 2024-07-01. 2023.

7   Appendix

.1   The test run report for the model

The content of the report can be found in the following URL (`JSON` format)

`https://pastecode.io/s/twvh8go9`

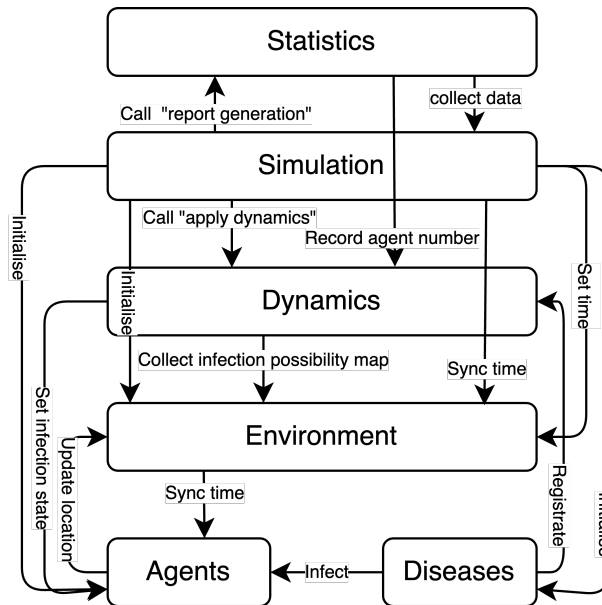## .2   The structrue of the agent-based model

Figure 5: The structure of the created agent-based model
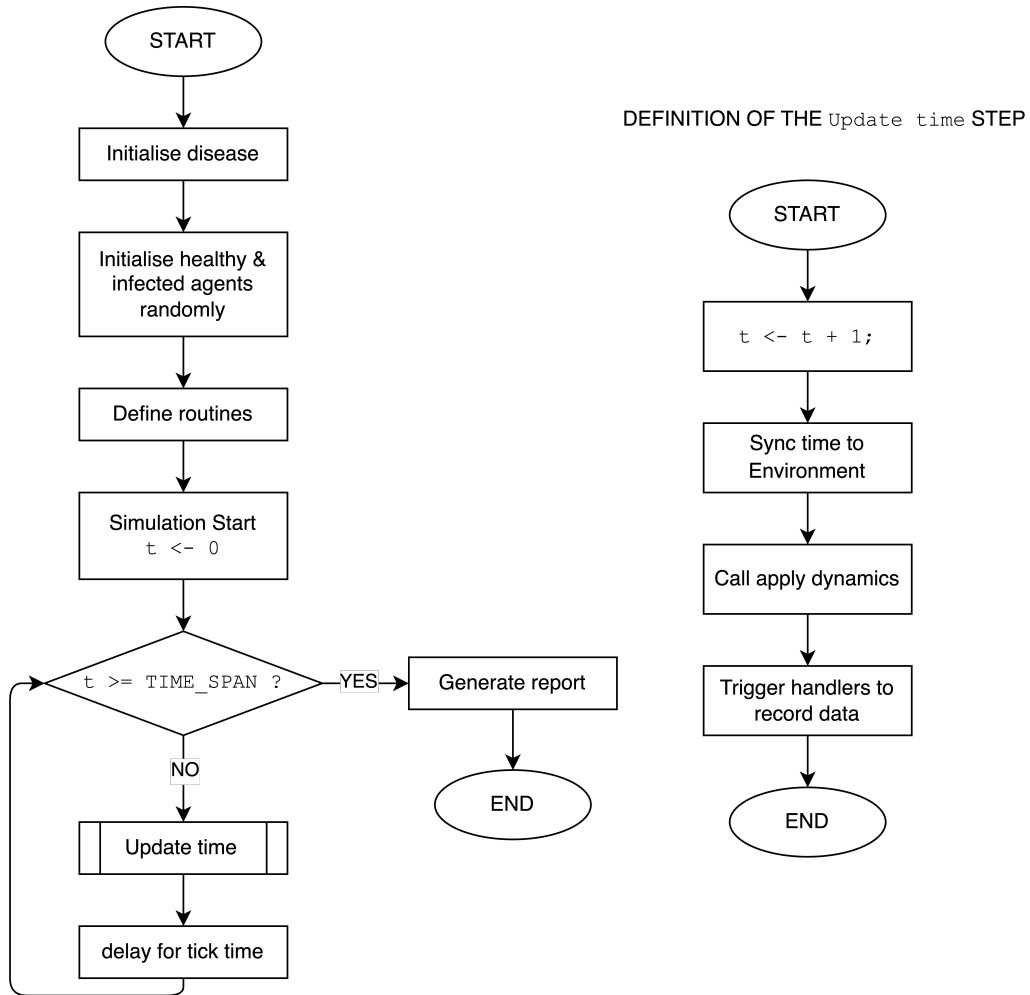
.3 The flow chart of the agent-based model

Figure 6: The flow chart of the created agent-based model