Modelling COVID-19 Transmission in a Restaurant Setting

CHRISTOPHER HODGSON

2615649

Supervised by Dr Savi Maharaj

April 2021

Dissertation submitted in partial fulfilment for the degree of Bachelor of Science with Honours *Computing Science*

Computing Science and Mathematics

University of Stirling

Abstract

Problem: Studies show that the large majority of COVID-19 cases originate from indoor spaces, primarily due to poor ventilation, overcrowding and prolonged exposure. This project focuses on testing the effectiveness of different safety measures in their ability to reduce the risk of virus transmission through aerosols, droplets, and fomites. The SARS-Cov-2 virus thrives in closed spaces like restaurants, classrooms, and offices, where virus-laden aerosol particles can remain in the air for several hours, and close contact is frequent.

By looking at the data on COVID-19 cases, a large spike known as 'the second wave' appears to occur shortly after restaurants reopened following a lengthy lockdown. It has been argued that the 'Eat Out to Help Out' scheme encouraging people to dine at restaurants might have contributed to this. Before reopening restaurants, more research must be done to reduce indoor transmission risk further and prevent another spike in cases.

Objectives: This project aims to create a computer simulation to capture the complexity of virus transmission in a restaurant environment. By modelling the likes of ventilation, human behaviour, and various other safety measures, the simulation is intended to be used as a tool to aid research into their effectiveness at reducing COVID-19 transmission.

Using the simulation to research each safety measure's effectiveness, we can estimate how large an impact it will have. It will help answer questions such as what effect does employing the proper air change rate in a restaurant have on the number of cases produced? Or how large an impact does limiting restaurant capacity have on reduction in infections?

Methodology: The simulation was created in the agent-based modelling software Netlogo. The model is composed of four main sub-models: the spatial sub-model: capturing the restaurant environment such as seating configurations and the network of nodes forming the pathways through the restaurant, the movement sub-model: handling action scheduling and movements of waiters and customers in the restaurant, the infection sub-model: simulating the virus emission of an infected person, virus intake/infection risk of a healthy person and the different precautions a person can take such as mask-wearing, and finally the ventilation sub-model: which represents the removal of aerosol particles from the air through mechanical and natural ventilation. The simulation parameters are controlled through a simple user interface allowing the investigation of a single or combination of multiple safety measures.

The data output from the model was analysed in excel to assess each safety measure effectiveness.

Achievements: This project's objectives were achieved successfully by creating a simulation that captures how COVID-19 is transmitted from person to person in a restaurant setting. This includes transmission through viral aerosol particles, respiratory droplets, and fomites. The simulation also allows for COVID-19 safety measures such as physically distanced tables and face masks to be implemented. The output data can be analysed to gauge how effective a safety measure was, based on how significantly it reduced probable infections and infection risk. A combination of safety measure can also be tested to create the greatest reduction in infection risk.

Attestation

I understand the nature of plagiarism, and I am aware of the University's academic misconduct policy.

I certify that this dissertation reports original work by me during my University project.

Signature: *Chris Hodgson* **Date:** *14/04/2021*

Acknowledgements

Firstly, I would like to thank my supervisor, Dr Savi Maharaj, for the incredible guidance and support she has provided me through the project, and Dr Paul McMenemy for the valuable feedback he gave on the initial prototype of the simulation. Thanks to all the staff at the university for supporting me through my undergraduate degree.

I'd like to thank my friends and family for motivating me and keeping my spirits high throughout the year.

Thanks to the people who participated in user testing for the fantastic feedback they provided.

Thanks to the management at The Kailyard for granting me access to the building to gather data on restaurant ventilation.

Table of Contents

Abstract i

Attestation ii

Acknowledgements iii

Table of Contents iv

List of Figures viii

1 Introduction 1

1.1 Background and Context 1

1.2 Scope and Objectives 2

1.3 Achievements 3

1.4 Overview of Dissertation 4

2 State-of-The-Art 5

2.1 Modelling techniques 5

2.2 Other simulations of COVID-19 transmission 5

2.2.1 Aerosol transmission of SARS-CoV-2 5

2.2.2 A room, a bar and a class. How the coronavirus spreads through the air 6

2.2.3 COVID Airborne Transmission Estimator 7

2.2.4 Simulated social distancing. 9

2.3 Pathfinding algorithms. 10

2.3.1 Dijkstra’s algorithm 10

2.3.2 A* algorithm 10

2.3.3 Best-first search 11

2.3.4 Pathfinding algorithm visualisation tool 11

2.3.5 Conclusion 11

2.4 Agent-based Modelling Systems 12

2.4.1 NetLogo 12

2.4.2 Repast 13

2.4.3 Swarm 13

2.4.4 Conclusion 13

2.5 Statistical modelling software 14

2.5.1 R Studio 14

2.5.2 Excel 14

3 Requirements 15

3.1 Requirements 15

3.2 Use Case Diagram 17

4 Design and Implementation 18
4.1 Mind map 18
4.2 Spatial sub-model 19
4.2.1 Room dimensions 19
4.2.2 Seating configurations 19
4.2.3 Node network 21
4.3 Movement sub-model 22
4.3.1 Node network navigation 22
4.3.2 Scheduling 23
4.3.2.1. Table states 23
4.3.2.2. Action priority 24
4.3.2.3. Table contact time 24
4.4 Infection sub-model 24
4.4.1 Virus emissions 25
4.4.1.1. Chance of being infectious 25
4.4.1.2. Aerosols 25
4.4.1.3. Droplets 28
4.4.1.4. Fomites 28
4.4.2 Infection risk 29
4.4.2.1. Inhalation of virus 29
4.4.2.2. Face touches 30
4.4.3 Protective measures 30
4.4.3.1. Masks 30
4.4.3.2. Hand washing 32
4.4.3.3. Exposure time 32
4.4.3.4. Reduced capacity 33
4.4.3.5. Surface sanitisation 33
4.5 Ventilation sub-model 34
4.5.1 Mechanical ventilation 34
4.5.1.1. Air conditioning 34
4.5.1.2. Portable HEPA filters 35
4.5.2 Natural ventilation 36
4.5.3.1. Windows 36
4.6 User interface 41
4.6.1 View 42
4.6.2 Controls 42

4.6.3 Outputs ... 42
5 Testing 43
5.1 Validation testing ... 43
5.1.1 General safety parameters ... 43
5.1.1.1. Capacity ... 43
5.1.1.2. Length of bookings ... 43
5.1.1.3. Length of service ... 43
5.1.2 Movement ... 43
5.1.2.1. Pathfinding ... 43
5.1.2.2. Scheduling ... 44
5.1.3 Virus emissions ... 44
5.1.3.1. Manual and automatic infection ... 44
5.1.3.2. Aerosol emission ... 45
5.1.4 Infection risk ... 47
5.1.4.1. Aerosols and droplets ... 47
5.1.4.2. Fomites ... 47
5.1.4.3. Surface sanitisation ... 47
5.1.5 Individual safety measures ... 47
5.1.5.1. Face masks ... 47
5.1.5.2. Hand washing ... 48
5.1.6 Ventilation ... 48
5.1.6.1. Mechanical ventilation ... 48
5.1.6.2. Natural Ventilation ... 49
5.2 Safety measure evaluation ... 50
5.2.1 Physically distanced seating ... 50
5.2.2 Reduced capacity ... 51
5.2.3 Reduced exposure time ... 52
5.2.4 Volume ... 53
5.2.5 Face Masks ... 53
5.2.6 Hand washing ... 54
5.2.7 Increased mechanical ventilation ... 54
5.2.8 Increased natural ventilation ... 55
5.3 Conclusion ... 55
6 User Testing 57
7 Example study 58
7.1 Parameters ... 58

7.2 Results 59
8 Conclusion 61
8.1 Summary 61
8.2 Evaluation 62
8.3 Future Work 63
References 64
Appendix 1 69
Appendix 2 – User guide / Installation Guide 71

List of Figures

Figure 1. Image of CFD simulation of aerosol particles exhaled from an infectious customer. [13] 6
Figure 2. Graphical representation of data showing the importance of COVID-19 safety measures. [16] 7
Figure 3. Image of excel spreadsheet used to implement the mathematical model. [15] .. 8
Figure 4. Simplified model input from National Geographic [17] 8
Figure 5. Simplified model output from National Geographic [17] 9
Figure 6. Washington Post's social distancing simulation. 9
Figure 7. Visualisation of Dijkstras path finding algorithm [20] 10
Figure 8. Visualisation of A* pathfinding algorithm [18] 10
Figure 9. Comparison of paths taken by each algorithm in the restaurant. [22] 11
Figure 10. Agent-based modelling phases [23] 12
Figure 11. Simulation use case diagram. 17
Figure 12. Simulation requirements mind map. 18
Figure 13. NetLogo switch 20
Figure 14. Comparison of physically distanced and non-physically distanced seating configurations 20
Figure 15. Image of restaurants node network with node types labelled. 21
Figure 16. Pathfinding process of selecting next node. 22
Figure 17. Table showing the triggered action of each table state. 23
Figure 18. Code snippet showing the table state progression calculation. 23
Figure 19. Image showing possible respiratory transmission routes [35] 24
Figure 20. Yellow coloured patches showing patches included in aerosol concentration calculations. 27
Figure 21. Area covered by aerosol particles output from breathing and talking. 27
Figure 22. Distribution of aerosol particles for a cough (left) and sneeze (right) 27
Figure 23. Fomite infected table 29
Figure 24. Infection risk equation 29
Figure 25. Example calculation of infection risk 30
Figure 26. Table of inhalation and emission efficiency for different mask types. 31
Figure 27. Images showing The Kailyard's air conditioning configuration. Intake (left) and exhaust (right). Red arrows showing air direction. 35
Figure 28. Air change concentration calculation 35
Figure 29. Image showing geographic elevation of Dunblane Hydro, the hotel housing The Kailyard restaurant [60]. The Kailyard circled in red. 36

Figure 30. Graph showing the air change rate provided by opening only the front windows. 37
Figure 31. Graph showing the air change rate provided by opening only the side windows. 37
Figure 32. Graph showing the air change rate provided by opening both front and side windows. 37
Figure 33. Image showing how air direction measurements were gathered in The Kailyard. 38
Figure 34. Image showing the air paths throughout the restaurant with front windows open only. 38
Figure 35. Diagram of air paths through the restaurant with side windows open only .. 39
Figure 36. Diagram of air paths through the restaurant with both front and side windows open 39
Figure 37. Graphic showing the movement of air in rooms without cross ventilation [61] 40
Figure 38. Diagram showing how air direction is calculated on a grid of patches in NetLogo using the angle of direction and distance. 41
Figure 39. User interface wireframe 41
Figure 40. Image showing an improvement made to the node network after testing. 44
Figure 41. Graph showing an agent's chance of being infectious vs the actual percentage of infectious agents produced in the simulation. 44
Figure 42. Patch colouring of different viral aerosol concentrations. 45
Figure 43. Bar graph showing the impact of coughing and sneezing on average viral aerosols per patch. 46
Figure 44. Bar graph showing the impact of reducing coughing frequency on the average viral aerosols per patch. 46
Figure 45. Graph showing the impact of average viral aerosols per patch on the average risk of infection over the first 250 ticks of simulation with one infected agent. 47
Figure 46. Graph showing the impact of mask emission efficiency on the average viral aerosol concentration per patch. 48
Figure 47. Example of reduced concentration calculation from mechanical ventilation 48
Figure 48. Graph showing the impact of air change rate on the average number of viral aerosols per patch. 49
Figure 49. Graph showing the impact of increasing the number of portable HEPA filters in the restaurant on the average viral aerosols per patch. 49
Figure 50. Table comparing the average viral aerosol concentration resulting from having all windows open vs AC on 5 ACH. 49
Figure 51. Table of base test parameters 50
Figure 52. Bar graph comparing the percentage of probable infections for physically and non-physically distanced seating configurations. 51

Figure 53. Graph showing the impact of restaurant capacity on the percentage of probable infections 51

Figure 54. Graph showing the percentage of probable infections for different booking lengths (mins). 52

Figure 55. Graph showing the number of probable infections for different service lengths (hours). 52

Figure 56. Line graph showing the impact of restaurant volume on the percentage of probable infections 53

Figure 57. Graph showing the impact of different mask efficiencies on the percentage of probable infections 53

Figure 58. Graph showing the impact of handwashing thoroughness on customers fomite infection risk. 54

Figure 59. Graph showing the impact of air change rate on the percentage of probable infections produced 54

Figure 60. Table comparing the average viral aerosol concentration resulting from having all windows open vs AC on 5 ACH. 55

Figure 61. Table ranking the how much each safety measure reduced the percentage of probable infections 55

Figure 62. Table of safe vs unsafe restaurant test parameters 59

Figure 63. Table of test results from safe vs unsafe restaurant. 59

1 Introduction

As of April 2021, the cumulative number of confirmed COVID-19 cases has reached over 130 million, resulting in more than 2.8 million deaths worldwide [1]. The novel coronavirus disease originated in Wuhan, China, before spreading rapidly to the rest of the world, catching many countries off guard. Cases quickly began to skyrocket, resulting in nationwide 'lockdowns' in attempts to get the virus under control. These restrictions have forced many businesses to stay closed for most of the past year resulting in the largest economic recession since the 2008 financial crisis [2], with the hospitality sector being among the most impacted industries by overall GDP [3].

In July 2020, the first UK lockdown was lifted allowing the hospitality sector to reopen. Restaurants were made to operate under a set of mandatory restrictions such as face coverings, physical distancing, household and group size limits [4]. To provide a kickstart to the hospitality sector, the UK Government launched the controversial 'Eat Out to Help Out' scheme that ran during August, giving customers 50% off their meal. This sparked a large debate about whether this was a contributing factor towards the second wave of cases we faced in the following weeks. This could be in part the reason that restaurants doors have remained closed since November 2020.

Using a computer simulation of a restaurant, this project investigates the impact of a range of safety measures and environmental parameters on the transmission of the SARS-CoV-2 virus (the virus causing COVID-19 disease) through aerosols, respiratory droplets, and fomites.

1.1 Background and Context

After a year in and out of lockdowns and tier systems, the UK Government has recently announced their Coronavirus roadmap to guide the country back to normality. As businesses begin reopening over the coming months, their owners and customers must be provided with the necessary information to resume operation safely.

Some café and restaurant owners claim that their businesses will not survive much longer under the current restrictions [5]. Therefore economically, these restrictions are unsustainable in the long term, meaning we must devise a more effective strategy for coming out and staying out of lockdown by adapting high-risk environments wherever possible.

Working in a restaurant myself, I have witnessed first-hand the pandemic's effect on the industry with severe loss of business and jobs. I can also see that more guidance is required to make restaurants a safer environment. The World Health Organisation (WHO) has encouraged more studies to be conducted on the transmission of COVID-19, it being a relatively new virus. For this reason, I believe the simulation would be useful towards improving COVID-19 safety within restaurants and other indoor spaces. As both a waiter and Computing Science student, I believe I can offer an insight into the industry that will help create a tool to aid the study of virus transmission.

There are many reasons why restaurants are considered a high-risk environment such as, difficulty social distancing, overcrowding, removal of masks while eating, poor ventilation, prolonged exposure, high noise levels causing the dispersal of more respiratory droplets, and high use and poor cleaning of tables, door handles etc. Even with restaurants open, '*a large portion of individuals (over 50%) are not willing to dine in at a restaurant*' [6], likely due to many safety concerns that must be addressed.

WHO states that there are several main ways that COVID-19 transmission can occur, the most likely vessels being respiratory droplets and aerosols. These are droplets of saliva and other respiratory secretions released from the mouth or nose when a person coughs, sneezes, speaks or breathes. An aerosol is any particle below the 5-10μm mark [7], and droplets are particles bigger than this. Research has shown that aerosol particles can remain in the air for up to three hours under certain circumstances. This allows them to travel further distances [8] and reach high concentration levels in poorly ventilated spaces resulting in a heightened risk of exposure for the surrounding individuals. On the other hand, Droplets tend to quickly fall to the ground or surrounding surfaces within 2 metres [9]. Contact is a slightly less common transmission method. This can be through direct contact such as shaking hands or indirectly from fomites. The term fomite refers to a surface or object contaminated with virus-containing respiratory secretions that could be transferred to the eyes, nose, or mouth by an unwashed hand. At the time of writing, on the 5th of April 2021, the CDC has reported that '*contact with a contaminated surface has less than a 1 in 10,000 chance of causing an infection*' [10], rendering the fomite transmission feature somewhat obsolete.

A computer-simulated restaurant will allow the user to study the impact of specific parameters on the overall COVID-19 transmission risk. Simulating such a complex environment is far more efficient as it cuts out the need for costly, impractical 'real-life' experiments while providing more controlled environment conditions for experiments. The simulation could also be used to help reassure wary members of the public that restaurants may be safer than they think if the correct precautions are taken.

1.2 Scope and Objectives

This project aims to produce a computer simulation that captures how the SARS-CoV-2 virus is spread in a restaurant setting. The model intends to allow the study of how well certain safety measures reduce virus transfer. It should be controlled through a simple user interface and output data such as a person's infection risk.

The simulation is based on the restaurant I have worked at for the past four years, The Kailyard in the Hilton Dunblane Hydro hotel. This restaurant was chosen for the simulation as being an employee, I am familiar with the layout of tables and the ventilation configuration. I was also granted access to the building to conduct some further analysis of the restaurant environment.

This project aims to allow experimentation with different parameters such as the above-mentioned ventilation configurations, table layouts, and human behaviour like handwashing and mask-wearing. With the data produced by the model, we can gauge roughly which safety measures are most/least effective at reducing the risk of transmission.

The long-term goal of work like this could be to aid the production of a set of guidelines for restaurants to follow, which would result in a safer environment for both staff and customers, but this is far beyond the scope of this project and would require extensive analysis of many different restaurants. It is also worth mentioning that the guidelines produced from further research could also apply to other indoor spaces such as offices, schools, and universities.

The objectives of this project are to:

1. Expand my knowledge of the problem and Netlogo.
2. Design a well-structured and expandable program.
3. Design a simple user interface.
4. Create a 2D multi-agent simulation of a restaurant capturing:
    - Customer and staff movements and actions
    - Virus emission and contraction through respiratory droplets, aerosols, and fomites

- Ventilation:
  - Natural ventilation (windows).
  - Mechanical ventilation (air conditioning).
  - Air paths.
- A range of safety measures e.g., face masks.

5. An evaluation of COVID-19 safety measures based upon data produced.
6. An optimised restaurant configuration to reduce COVID-19 transmission among customers and staff.

1.3 Achievements

This project's objectives were achieved successfully through the creation of a simulation that captures how COVID-19 is transmitted from person to person in a restaurant setting. This includes transmission through viral aerosol particles, respiratory droplets, and fomites. The simulation also allows for the implementation of COVID-19 safety measures such as physically distanced tables and face masks. The simulation can output data that can be analysed to gauge how effective a safety measure was based on how significantly it reduced probable infections and infection risk. A combination of safety measure can also be tested to create the greatest reduction in infection risk.

The specific objectives for this project were:

1. To expand my knowledge of the problem and NetLogo. This was achieved through extensive research into the transmission of COVID-19 and the effectiveness of safety measures. The creation of many simulation prototypes grew my knowledge of NetLogo to a large degree.

2. Design a well-structured and expandable program. This was achieved as the code is structured in a way that makes it easily expandable, which has been proven throughout development as features were added incrementally.

3. Design a simple user interface. This was achieved through careful design and user testing. The feedback gathered from user testing was implemented into the user interface.

4. Create a 2D multi-agent simulation of a restaurant capturing:

- Customer and staff movements and actions. Achieved using a best-first search path finding algorithms to navigate a network of nodes to specific locations controlled by an action schedule.
- Virus emission and contraction through respiratory droplets, aerosols, and fomites. Achieved by using data gathered from relevant literature to provide an estimate of virus emission and intake of viral aerosols, larger droplets, and fomites.
- Ventilation:
  - Natural ventilation (windows). Achieved by modelling the air change rate provided and the air paths through the restaurant when different combinations of windows are opened or closed, simulating how viral aerosols can be carried in air currents.
  - Mechanical ventilation (air conditioning): Achieved by implementing the ability to set the air change rate provided by the air conditioning, which will impact the number of viral aerosols in the air. Also, the

option to place portable HEPA filters around the restaurant to further reduce the number of aerosols.

    - Air paths. Mostly achieved. This feature was included for natural ventilation only, for reasons discussed later in this report, but it was successfully implemented for all natural ventilation configurations.

  - A range of safety measures. Achieved through the implementation of face masks, hand washing, physically distanced seating, capacity, booking length and service length limits, reduced restaurant volume, and increased ventilation.

5. An evaluation of safety measures based upon data produced. Achieved by gathering data using the NetLogo behaviour space to run the simulation under specific parameters. This data was analysed in excel to investigate its effectiveness at reducing probable infections.

6. An optimised restaurant configuration to reduce COVID-19 transmission among customers and staff. Achieved by setting up the restaurant simulation with a combination of safety measures such as the ones listed above and estimating the infection risk if these measures are implemented.

1.4 Overview of Dissertation

The 'State-of-The-Art' chapter will discuss the several different approaches that were considered for the project, including an analysis of potential simulation methods such as agent-based, equation-based or CFD modelling. A discussion of similar work will be shown through examples of each approach, weighing up its advantages and disadvantages. Based on this research, we will conclude why agent-based modelling is the most appropriate approach. Following this will be a comparison of several agent-based modelling and statistical analysis software products and present a conclusion on why NetLogo and Excel were selected for the project. Also covered will be an analysis of several potential pathfinding algorithms to handle the movement of agents within the simulation and why the best-first algorithm was selected.

The 'Requirements' chapter will present the requirements for the project as well as a use case diagram for the simulation. For each requirement, there will be a discussion of why the feature is required and how it will aid the simulation's functionality.

The 'Design and implementation' chapter discusses the sub-model breakdown of the simulation and the design/parameterization and implementation of each feature. Design/parameterization discusses the relevant literature that the feature is based upon. Implementation states how the feature was incorporated into the simulation.

The 'Testing' chapter covers both the validation testing of the simulation to ensure it is working correctly, and the testing/evaluation of different safety measures that can be put in place to reduce COVID-19 transmission.

The 'User Testing' chapter covers the portion of the project where fellow students evaluated the simulation and provided feedback that could improve it, highlighting any weaknesses of the simulation's user interface.

The 'Example Study' chapter compares a restaurant with and without safety measures implemented, showing the difference between the two and how the research carried out in this project helped improve COVID-19 safety in restaurants.

The final 'Conclusion' chapter summarises the project, including a recap of its achievements, an evaluation of its overall success, and ways in which the simulation could be improved by future work.

2 State-of-The-Art

The following section contains a comparison of different types of modelling, a critical analysis of several models assessing the risk of SARS-CoV-2 exposure in restaurants, bars, or similar indoor environments. Next is a comparison of several different pathfinding algorithms that could be used to capture human movement within the model. Also included is an analysis of the three most popular agent-based modelling tools: NetLogo, Repast and Swarm. Included in the discussion will be the key features of each software and its strengths and weaknesses. Based on this analysis, a conclusion has been reached on why NetLogo best fits the project's requirements. The section finishes with a brief evaluation of several statistical analysis software products that could have been used to analyse the data produced by the model.

2.1 Modelling techniques

There are many different modelling techniques, several examples of which will be analysed in the following sections. The first type is a mathematical model based on equations and relationships between x and y variables. An example of a purely mathematical model can be seen in section 2.2.3 implemented using an excel spreadsheet. Still, it should be mentioned that all modelling techniques will have some mathematical aspect. This type of model is quick to produce and alter but can be complicated to use for non-expert users.

A more user-friendly modelling technique is agent-based modelling. This is when '*a system is modelled as a collection of autonomous decision-making entities called agents*' [11]. There is often a highly graphical approach taken with this type of modelling, making it easier to understand when compared with mathematical modelling because it is possible to watch events and interactions as they happen. It is also highly flexible and said that '*One may want to use ABM when the appropriate level of description or complexity is not known ahead of time and finding it requires some tinkering*.' [11]. This is perfect for the project as at the beginning, although there was a plan, it was not certain what level/how in-depth the model would be. Agent-based modelling allows the model to be easily built up over time.

The final modelling technique that could be used for this problem would be computational fluid dynamics (CFD). This allows for the modelling of fluid flow problems and, in this case, can be used to capture the behaviour of aerosol particles in the air. An example of this can be seen in section 2.2.1. Many CFD models rely on high-performance computing, such as supercomputers and therefore, beyond the scope of this project [12].

2.2 Other simulations of COVID-19 transmission

This section contains an analysis of similar COVID simulation/modelling tools, including a brief description and the strengths and weaknesses of each.

2.2.1 Aerosol transmission of SARS-CoV-2

The first example presents '*Evidence for probable aerosol transmission of SARS-CoV-2 in a poorly ventilated restaurant'* [13]. In a crowded, poorly ventilated restaurant in China, there was a cluster of cases all seeming to originate from one lunchtime. There were three families sat at separate tables, all in the air path of a nearby recirculating air conditioning unit. One infected customer (purple, Figure 1) sat at the back-middle table, exhaling tiny virus-laden respiratory droplets which were carried through the restaurant in the air currents caused by the AC unit. As shown by the turquoise and orange gas, these aerosol particles appear to circulate the lower end of the restaurant. During this event, the ventilation rate was 0.75 – 1.04 L/s per person, but the recommended ventilation rate for commercial spaces like this would be 10 L/s per person [14]. As a result of this poor ventilation and these specific air currents, this has thought to have resulted in the infection of 10 customers, the furthest of which being 14ft away.

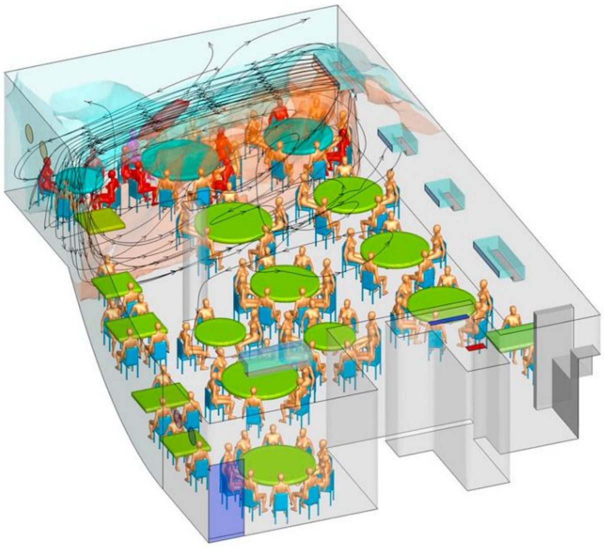

Figure 1. Image of CFD simulation of aerosol particles exhaled from an infectious customer. [13]

Overall, this study seems to provide evidence for '*probable aerosol transmission of SARS-CoV-2*'. What this study does well is recreating the specific environment to the finest details, such as considering a person's body heat as well as the heat of their food and how this would impact the airflow. Also, the use of a tracer gas mimicking the fine aerosol droplets produced from the infectious customer's mouth allowed the precise direction and speed of air currents to be captured. The data from these experiments was input into a CFD software to produce a 3-dimensional simulation of aerosols in the air, which shows the envelope of air being recirculated before eventually dispersing through the rest of the restaurant and being extracted through a small exhaust fan in the restroom.

As with any study of a past event such as this one, it is impossible to capture it with 100% accuracy, which is expected. But it must be mentioned that even though there were only ten confirmed cases in the city at the time, the possibility that these people caught the virus before or after the lunch can never be ruled out.

2.2.2 A room, a bar and a class. How the coronavirus spreads through the air

The following simulation used the 'COVID Airborne Transmission Estimator' model [15] from section 2.2.3 to calculate the probability of COVID-19 transmission, taking into consideration parameters such as exposure time, ventilation, capacity, and mask-wearing [16]. '*using real outbreaks that have been analysed in great detail*', this article presents a high-level visualisation showing '*the importance of measures that hinder aerosol transmission*'. This study does a great job of demonstrating the importance of different protective and prevention measures using professional graphics presented in a well thought out way.

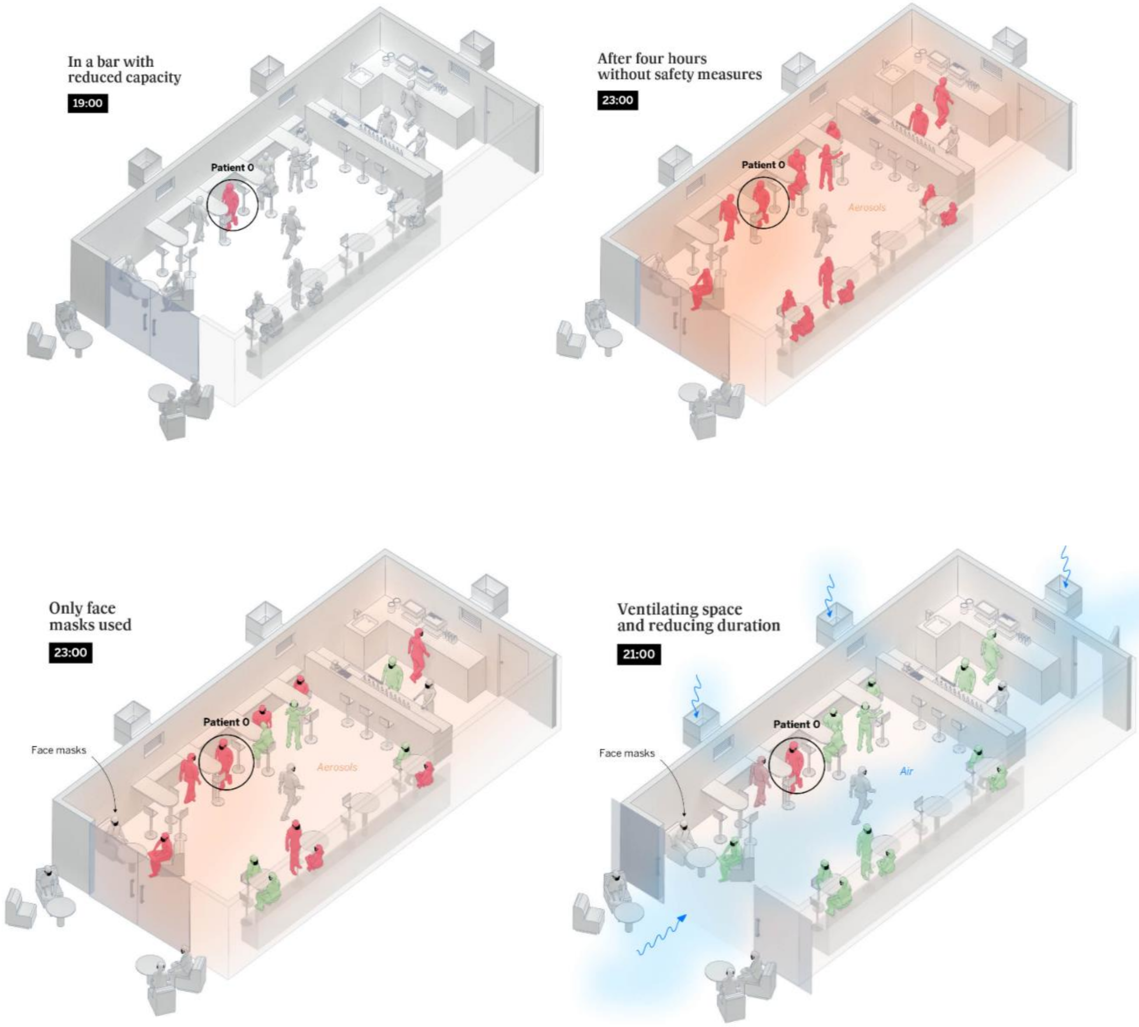

Figure 2. Graphical representation of data showing the importance of COVID-19 safety measures. [16]

The creators stated that '*The calculation is not exhaustive, nor does it cover all the innumerable variables that can affect transmission, but it serves to illustrate how the risk of contagion can be lowered by changing conditions we do have control over*' [16]. In other words, there are so many factors that contribute to transmission, its near impossible to capture them all into a simulation. So, this study in particular highlights aerosol transmission specifically, disregarding the possibilities of fomite and droplet transfer.

2.2.3 COVID Airborne Transmission Estimator

The third example is a publicly available model [15] that can be used to assess the risk of aerosol transmission under a variety of conditions in several different settings such as classrooms, university campuses and supermarkets. In the words of Prof. Jimenez (the lead developer): '*Different actions have very different costs, so the hope is that the tool can help allocate limited resources to reduce the risk of infection most effectively.*' [15]. To summarise, their goal is to provide a tool allowing the user to find the most effective and realistically achievable configuration of safety measures for a given situation.

As mentioned above, this model focuses on aerosol transmission alone by assuming adequate social distancing rules are being followed. The model is '*based on a standard model of aerosol* *disease transmission, the Wells-Riley model. It is calibrated to COVID-19 per recent literature'* [15] with all references provided. It is often updated as we learn more about the virus with the most recent update being just several weeks ago. This is reassuring to know that the model is up to date and consistent with the current literature.

The model works using an excel spreadsheet where the user enters details specific to the environment they are modelling. The model is split into environmental, infection and person/activity sub-models. The inputs provided are used in a series of mathematical equations to estimate how many infections will be produced due to the event. Several base models are provided, but it is possible to tune a particular model to represent a different setting, allowing for more flexibility.

Parameters related to people and activity in the room		
Total N people present	30	
Infective people	1	person
Fraction of population inmune	0%	
Susceptible people	29	people

Figure 3. Image of excel spreadsheet used to implement the mathematical model. [15]

A drawback of this model, touched on in section 2.3.2, is that there is no consideration for fomites or droplets. This could also be seen as an advantage by placing more in-depth focus on one particular aspect instead of a more general implementation covering all transmission methods.

To a person who is familiar with virus aerosol transmission, this model would be a very useful resource. Still, for the average person, some things could be confusing and overwhelming. Before using the model, they suggest reading the 'readme'. This 'readme' is roughly ten pages long and contains many technical terms that could be off-putting to users who lack in-depth knowledge on the matter. However, the technical terms are necessary to provide the background and instructions needed to use the model correctly. Also provided is a webinar and Q&A sessions on the tool and further background reading to aid understanding.

National Geographic [17] created a simplification of this model, which is more a more appropriate tool for use by the public. It cuts out a lot of the more specific parameters and instead provides a generalization of conditions for each setting. Using slider restricted input to guide the user into entering appropriate data and simple graphical output for the different settings.

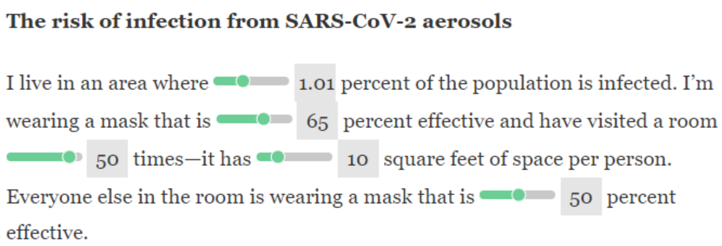

Figure 4. Simplified model input from National Geographic [17]

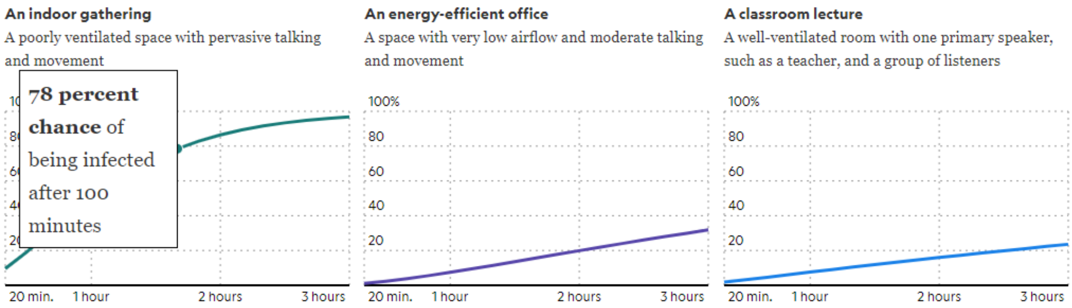

Figure 5. Simplified model output from National Geographic [17]

2.2.4 Simulated social distancing.

In March, when the COVID-19 outbreak was declared a pandemic, the Washington Post published an article showing a series of simple simulations to demonstrate the effectiveness of different degrees of social distancing at a community level [18]. What is being simulated is 200 mobile agents within a box to replicate the spread of their made-up virus 'simulitus', through a small town. Different levels of social distancing are simulated using a group of stationary agents who are self-isolating and another group of mobile agents who are not. Throughout each simulation, a live graph displays the number of healthy, recovered, and sick agents.

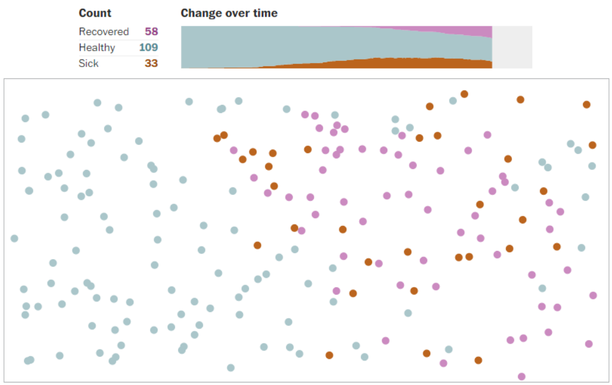

Figure 6. Washington Post's social distancing simulation [18].

While this simulation is somewhat oversimplified, it demonstrates the basics of social distancing and a good use of a simulation. Not every simulation has to go entirely into depth on a topic as some are more high-level visual models and others are highly detailed models used for scientific research.

Another research article [19] discusses the results of a NetLogo SIR (susceptible, infected recovered) spatial network model. The model investigates the effect of social distancing on controlling epidemic spread, factoring in economic cost. Many simulations (over 2 million) were carried out to produce reliable results. This project used the statistical package R for analysis which concluded that social distancing is the most cost-effective method of epidemic control. The article provides evidence that using an agent-based NetLogo model in conjunction with statistical analysis can be effective when studying methods of controlling virus spread.

2.3 Pathfinding algorithms.

A pathfinding algorithm is used to find a path from point A to point B, navigating around any obstacles on the way. These algorithms can have a high degree of variability in execution speed, and the quality of path found. Some algorithms will focus mainly on producing a reasonable path very quickly, whereas others focus on finding the absolute shortest path, which can take slightly longer. The ideal algorithm for this project would have a good balance between these two factors.

2.3.1 Dijkstra's algorithm

Named after its creator, Dijkstra's algorithm is a search algorithm used to find the absolute shortest path between two locations through an environment with obstacles preventing an agent from walking in a straight line from point A to B [20]. So, imagining a restaurant, the goal of the waiter is to travel from one side of the restaurant (A) to the other (B), navigating around tables and chairs (obstacles). It works by expanding all possible paths from the starting node until the destination is reached.

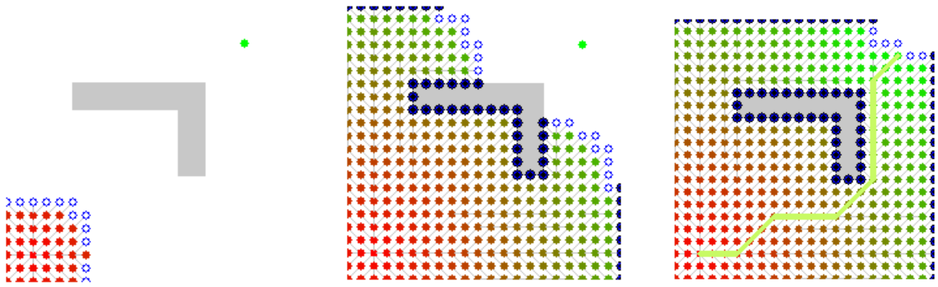

Figure 7. Visualisation of Dijkstras path finding algorithm [20]

This algorithm is time-consuming for larger networks [21] and can be inefficient as it calculates costs for paths that will never be travelled. So, as this algorithm focuses less on execution speed and more on finding the absolute shortest path between locations, it would not be wise to use such an algorithm for this project.

2.3.2 A* algorithm

A* (A star) is a search algorithm commonly used for pathfinding and graph traversal. It is an adaptation of Dijkstra's algorithm [20] and is argued to be the best pathfinding algorithm. Rather than expanding every node, it expands the node with the best chance of reaching the goal. Its efficiency depends on the heuristic used, but it expands fewer nodes than the exhaustive Dijkstra's algorithm.

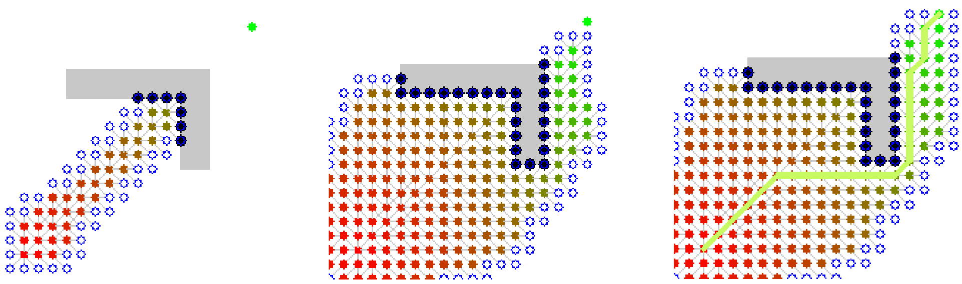

Figure 8. Visualisation of A* pathfinding algorithm [18]

This algorithm could be a good choice for this project, but it is unlikely to be used due to its coding complexity.

2.3.3 Best-first search

The best-first algorithm uses a greedy heuristic, choosing the best option available to it (closest to destination) at that time. It will not consistently achieve the shortest path, but it will choose a good enough path, and it will do so rapidly. This is essential for this project to ensure the simulation runs quickly.

2.3.4 Pathfinding algorithm visualisation tool

Using a tool hosted on Github [22], the floorplan of the restaurant this project is based upon was recreated to assess and visualise how each pathfinding algorithm would function for a typical action like a waiter walking from the service window (green) to a table (orange). As expected, the quickest to find a path was the best-first search taking roughly 0.17ms to find a viable path. Again, as expected, it was not the shortest path, but that is not a major issue for this project. The next quickest algorithm, taking roughly 0.2ms to find a path, was the A* algorithm, producing almost the shortest path possible. The algorithm with the poorest performance was Dijkstra's, taking roughly 1ms, around five times slower than the other two, but producing the absolute shortest path. The expanded nodes can be seen in blue.

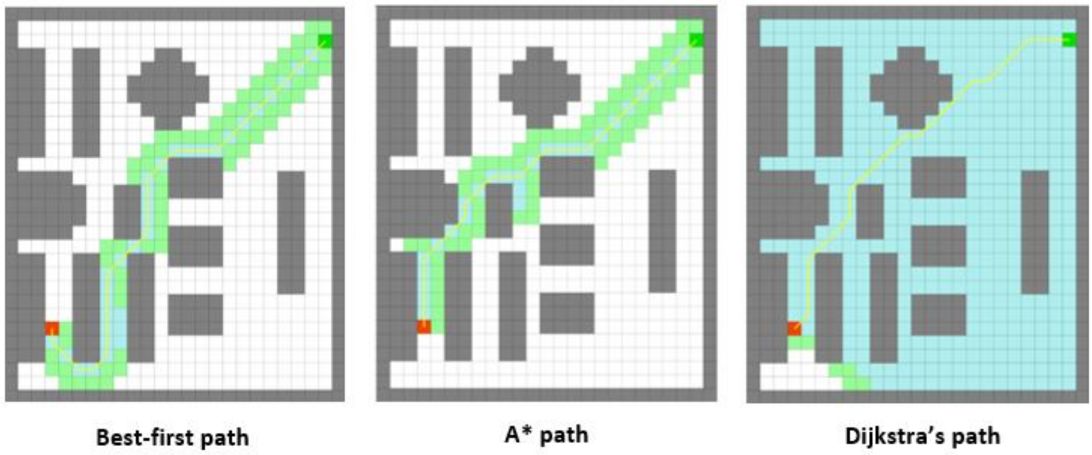

Figure 9. Comparison of paths taken by each algorithm in the restaurant. [22]

2.3.5 Conclusion

For the reasons outlined above, the decision has been made to use the best-first search algorithm with a greedy heuristic. There are two ways this algorithm could be implemented, the first using a grid of patches, as can be seen in Figure 9. For this, agents would inspect all surrounding patches and move to the one closest to the destination, repeating this until the final destination has been reached.

An alternate implementation of this algorithm would be to set up a network of nodes across the highways/popular paths throughout the restaurant. Instead of moving from patch to patch, agents would find the node closest to the destination and step towards it along the link between the two nodes. The benefit of this is fewer calculations would be required to move between locations as there are fewer options to compare, leaving computational power for the rest of the simulation. This would also make agents follow a more realistic path through the middle of tables.

These details are important as if the model runs slow, this can cause delay to research, and if the agents follow paths that do not replicate how humans move, it could negatively impact the model's results.

2.4 Agent-based Modelling Systems

A general agent-based model is a computer simulation that represents a world using agents in an environment with a set of rules of how they interact with other agents and the environment in which they have been placed. While most ABMS follow this same structure, they all vary to some degree, which will be explored in this section. The general structure of ABMS can be seen in Figure 10 below, beginning with the setup of the world, spawning in agents and setting global variables before runtime. During runtime, there is a continuous loop of updating agents and the environment according to the coded specification, then displaying the updates to the screen. This is repeated until the user terminates the simulation or some stop clause is reached.

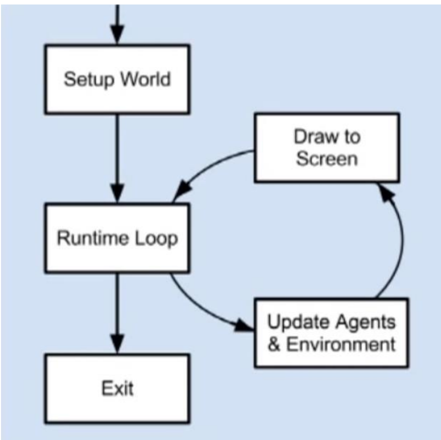

Figure 10. Agent-based modelling phases [23]

2.4.1 NetLogo

NetLogo is a free, open-source, '*programmable modelling environment for simulating natural and social phenomena'* [24]. A NetLogo model consists of three main components. First is the 'observer' or the user who instantiates the world. The next being 'turtles', which are the mobile agents within the model. These 'turtles' can represent many things depending on the model's context. In this case, some example breeds are waiters, customers, tables, chairs, and more. Finally, the environment is made up of a grid of 'patches' which are simply stationary agents. The agents are programmed defining how they interact with each other and the environment, for example, a waiter agent navigating a restaurant and interacting with customers agents. These interactions are displayed graphically through the NetLogo 'view' [24].

NetLogo has many advantages. The first being that it is free, which is perfect for keeping the project's cost at a minimum. It has a straightforward user interface and uses the language 'Logo', which has a very welcoming syntax for new users. Although it may seem simple, it does not lack the performance required for complex simulations as it can run thousands of independent agents. This is because the software was designed to have a '*low threshold and no ceiling'*, being easy for novice users, but experts do not feel limited by its capabilities [25]. A significant benefit of NetLogo is the extensive documentation it has online, as this seems to be a pitfall of many other ABMS. With a large model library, it is possible to find pre-existing models and adapt parts of them to fit the required solution [26]. The software is also well maintained, reducing the chances of encountering any bugs during the project [26].

Where NetLogo seems to fall short against its competitors is execution speed when running the simulations. This could have more of an impact on larger models [26]. Other than that, it seems to be an excellent fit for this project.

2.4.2 Repast

Standing for Recursive Porous Agent Simulations Toolkit, Repast is an ABM framework that is free and open source. It supports several different languages, such as the C++ version catering for larger projects using supercomputers or the most recent Java version 'Repast Simphony' catering to the typical computer in a similar fashion to NetLogo. The execution speed is fast, and the software is moderately easy to install. Repast is maintained well with a high amount of support from the academic community and also includes a large model library [26].

The disadvantages of Repast range from its average documentation to the requirement of prior knowledge of another complex language like C++, Python or Java, whereas NetLogo uses a simple language of its own. Its difficulty is in the middle ground between NetLogo and Swarm and will likely require several days of use before getting to grips with the user interface [26] [27].

2.4.3 Swarm

Swarm is another framework/library that was a precursor to Repast and is considered '*the grandfather*' of ABMS [26]. A swarm model contains swarms/collections of agents with a schedule of events for those agents. It is considered a moderately powerful simulation platform [27]. Its Objective-C version is the most mature library platform, and with its age, it brings stability and good organisation [28]. This seems to be the end of Swarms advantages.

There are plenty of drawbacks to Swarm, starting with the intense learning curve estimated to be several weeks, probably partly because of the poor documentation. Compared to the large model libraries of the other ABMS, Swarm only includes a few example models. It is also said to be very complicated to install as it is not well supported by different operating systems. The usage in the academic community is low due to the number of shortcomings it seems to possess when compared to other similar tools. These days Swarm is considered somewhat outdated as its userbase is diminishing [27].

2.4.4 Conclusion

In conclusion, the project requirements best align with the features and advantages that NetLogo provides. Having never used an ABMS prior to this project, at this stage I would consider myself a relatively new user which is best catered to by NetLogo due to its simple Logo language and graphical user interface. The extensive documentation available online compared with the other ABMS will be extremely helpful as a new user. NetLogo should have enough power and speed needed to run the planned simulation, meaning the other more powerful software should not be required. NetLogo's model library includes several existing models that could be incorporated into the planned model. Having plenty of experience using Java, I imagine it would be possible to pick up Repast Simphony, but after carrying out this research, for the reasons above, I have chosen to stick with NetLogo.

2.5 Statistical modelling software

The project requires a way of analysing and visualising the data produced by the simulation. This will allow us to study things such as: What impact does increasing the restaurant's air refresh rate have on the concentration of viral aerosol particles in the air?

To give some examples of the data we will need to plot, here are some of the experiments that will be run on the simulation:

For numerical parameters, data from the reporter variables is gathered using a technique called parameter sweeping, which performs multiple simulation runs at each value of the input variable. For example, the air refresh rate can range from 0 to 14 changes per hour. Many repetitions of the simulation will be run with 0 air changes per hour, measuring the percentage of customers and staff that became infected. An average of these values is taken and plotted on a graph. This process will be repeated for air refresh values from 2 to 14 in increments of 2. Such an experiment would hypothesise that viral aerosol concentration would decrease as the air refresh rate increases, which would be shown by a line graph.

For Boolean variables, multiple runs of the simulation are performed with the input variable set to true, and an average is taken. The process is repeated for the variable set to false. The results from the runs could be compared by plotting both results on a bar graph.

These experiments will typically produce a large quantity of data, with many rows and columns, so we will need to select a software that can handle a large amount of data and perform basic calculations like averages and produce basic plots and graphs.

2.5.1 R Studio

The first statistical modelling software we can look at is R. '*R is a language and environment for statistical computing and graphics.'* [29]. It is highly popular, but it is said to have a very steep learning curve. Due to the steep learning curve, the time required to learn the language would not fit the project's time frame. Also, R is a powerful language and is often required for complex statistical analysis, which will not be needed as this project will only undertake relatively simple plotting and graphics [30].

2.5.2 Excel

A better option will be Microsoft's Excel. While being less powerful than a language like R, it is a lot more user friendly and still has adequate tools for simple data visualisation and analysis [30], which should be perfect as all this project needs is a way of visualising the data output from our simulations, calculating averages and plotting standard error.

3 Requirements

This section will layout out the requirements of the simulation. The requirements were gained through a combination of online research and from my own experiences as a waiter.

3.1 Requirements

- The model should allow control over general safety parameters such as:
    - The ability to limit the capacity of the restaurant. This will allow investigation into whether limiting the rooms capacity results in a reduction of probable cases.
    - The ability to limit the booking length to investigate the impact of reducing length of exposure on the probable infections produced.
    - Like above, the ability to limit the total length of service. To investigate if a longer service results in a higher number of cases.
- The simulation should allow control over parameters relating to virus emission such as:
    - The ability to use the mouse to manually infect agents or have infections handled automatically based on the probability of an agent being infectious.
    - The severity of infectious agents' symptoms as some cases are asymptomatic and others may be super emitters. This can be done by altering the frequency of coughing and sneezing as well as the ability to turn these off completely.
    - The restaurant's volume as it has been shown that the louder a voice is, the more aerosol particles are released into the air. For example, if there is music playing, people will be forced to speak louder to compensate.
    - A visualisation of virus concentration should be shown by the intensity of red of a patch.
- The simulation should estimate the infection risk of a susceptible person based on tidal breathing rate, the air's viral aerosol concentration, exposure to viral droplets, and infectious dose.
- The simulation should include a representation of aerosol and droplet safety measures such as:
    - Face masks. Mask wearing should be able to be turned on and off. Also, many different types of face masks can have different inhalation and exhalation efficiencies. So, there should be a feature that allows for independent control over these factors. Face masks should be removed while a customer is eating.
    - Physically distanced seating. There should be the option to set up the restaurant with non-physically distances seating or physically distanced seating with 2m between chairs of different tables. This will allow investigation into whether physical distancing is an effective measure at reducing virus transmission in a restaurant.
- The simulation should include a method of tracking how the virus can be transferred around the restaurant through contact, including:
    - How infectious agents can deposit virus particles into surfaces like tables and doors through coughs, sneezes, and touch.

    - How susceptible agents can pick up virus with their hands from contaminated surfaces, capturing how hands would pick up more virus as more contact is made with the surface.
    - Face touches. Fomites are transferred into the body via unclean hands; therefore, the number of face touches with dirty hands should be tracked. The number of face touches can depend on if a person is wearing a mask or not.
- The simulation should include hand washing and the ability to prop open the front doors to protect against indirect fomite virus transfer.
- The simulation should capture the movement of people through:
    - A network of nodes/pathways
    - A best-first search path finding algorithm to find a reasonable path that a person might take from point A to point B.
    - Time spent at various locations like tables or the kitchen service window.
    - Seating of customers, either manually choosing which tables are filled using the mouse or have this handled automatically by the program.
- The simulation should represent different types of ventilation:
    - Mechanical ventilation through air conditioning units or portable HEPA filters. This should be represented as a general percentage-based reduction of virus concentration throughout the entire room controlled by altering the air refresh rate. A recirculation setting should also be included for the air conditioning system, demonstrating its effect on overall room virus concentration and concentration near recirculation vents.
    - Natural ventilation through windows. This will carry virus particles from the intake window to the outtake window following a set air path.
- The simulation must have outputs showing the current state of the simulation at any point. Some outputs should be displayed graphically, such as the average concentration per patch. Outputs like the total number of probable infections produced should be shown as a numerical value.
- Some non-functional requirements would be the ability to place focus on specific aspects of the model by hiding tables, chairs, and nodes. Also, the ability to enhance the colouring of patches to better show the concentration of aerosols for situation where a person may be asymptomatic and outputting a small number of aerosol particles which may be difficult to see.

3.2 Use Case Diagram

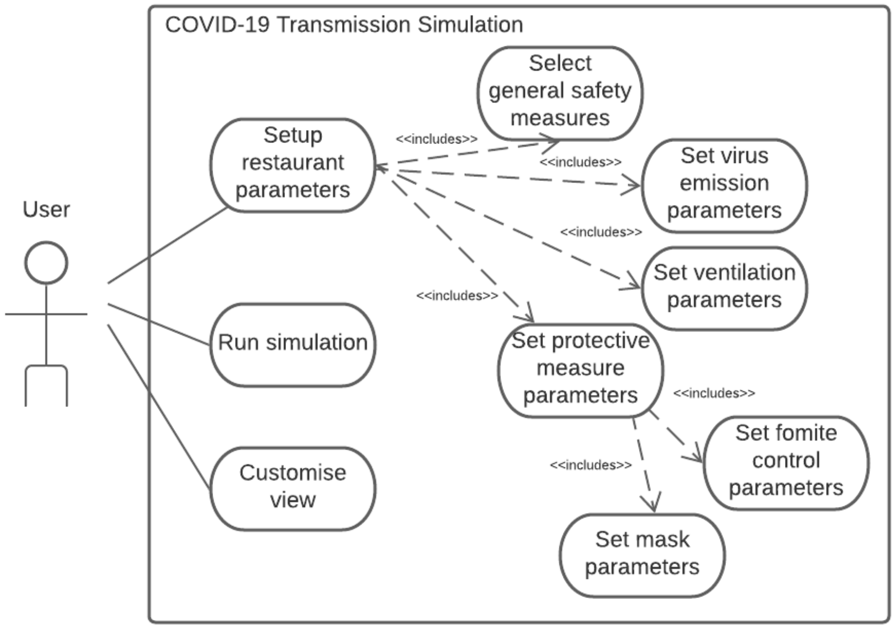

Figure 11. Simulation use case diagram.

In this use case diagram, we can see the user's actions to set up and run the simulation.

There is no specific parameter setup order required, but the first action could be to set the general safety measures, for instance, the capacity limit, dining time limit, service time limit and seating configurations. They could then set virus emission parameters such as the chance of a person being infected, the restaurant's volume, and severity of symptoms. After this, they could set up the ventilation parameters, e.g., open the windows, set the AC's air change rate and include any portable HEPA air filters. To finish setup, they could set individual protective measures, e.g., are people wearing masks and how effective are they and are people are washing their hands and how thoroughly.

At this point, the user can run the simulation and observe the experiment's output. During this, they can customize the view to highlight specific parts of the simulation for observation purposes by hiding tables and chairs to focus on the agent's movements and the aerosol concentration or hide the network of pathways to focus on fomite transfer.

4 Design and Implementation

The simulation is made up of 4 sub-models: the spatial, movement, infection, and ventilation sub-models. Each sub-model has its own role in the simulation with many components/parameters. This section will cover the parameterization and design of each component and what research each parameter is based on, as well as how the component was implemented into the simulation.

The spatial sub-model defines the location of stationary agents like tables, chairs, walls, and nodes, providing the environment for mobile agents to navigate through and interact with.

The movement sub-model handles the movement of mobile agents such as waiters, customers, and bartenders. This controls the agent's movement between locations, the pathways they travel, and their scheduling of actions like when to take an order.

Next is the infection sub-model, which can be split down into further sections of their own. These will be discussed further in later sections, but generally, it represents agent's virus emissions, risk of infection/virus intake and protective measures that can be implemented to reduce risk.

The ventilation sub-model captures the mechanical and natural ventilation configurations with variables like air direction and air refresh rate that affect the concentration of aerosol particles in the air.

Some components may fall between several sub-models; for example, the pathways agents use to travel around the restaurant fall partially under both the movement and spatial sub-models.

Lastly, will be a discussion of the design and implementation of the user interface's view, controls, and outputs.

4.1 Mind map

A mind map can be used to visualize the partitioning of requirements into different sub-models.

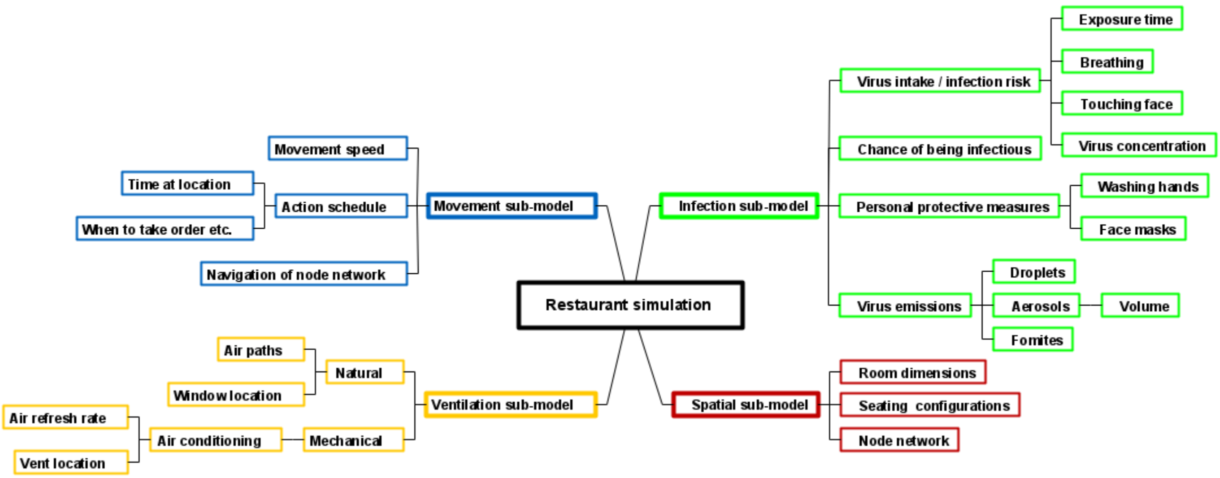

Figure 12. Simulation requirements mind map.

4.2 Spatial sub-model

The spatial sub-model refers to the grid of patches and stationary agents in NetLogo, making up the restaurant environment for mobile agents, like waiters, to interact with. Patches can represent different things such as walls, windows, doors or open space, and stationary agents include tables, chairs, sanitisation stations and air conditioning vents.

4.2.1 Room dimensions

Parameterization

First, the size of a patch should be equivalated to something in real life, so it is possible to define the restaurant's dimensions. Each patch is 0.5m in width and 0.5m in length. This value was chosen as it is roughly the amount of space a person occupies when sitting or standing, and due to this, chairs and tables can be sized in multiples of $0.5m^2$. I would estimate that 0.5m/s or 1 patch per tick is a comfortable pace for people to travel through the restaurant and lets agents travel smoothly in the simulation rather than jumping large distances which could make the simulation harder to follow. Perhaps the most important reason for choosing this size is that the level of detail it allows for is just right, and the simulation can be precise with the viral aerosol concentration and air direction of specific areas, and do not have to generalize as much is what would have to happen if the patches were larger, say $1m^2$. Also, if the patches were made any smaller, say to $0.25m^2$, the simulation would begin to struggle as this means four times more patches to compute aerosol concentration for, resulting in a delay to research and experiments.

As mentioned previously, the spatial sub-model is based on The Kailyard restaurant. The length and width of the room was measured and represented in the simulation. The length being 14 metres and the width being 12 metres translating to 28 patches by 25 patches.

The Kailyard has abnormally high ceilings of roughly 6 metres, so as this would not represent the average restaurant, the decision was made to use today's industry-standard value of 2.7 metres for the ceiling height [31].

It was considered to allow the ceiling height to be varied using a slider as if the volume of the room is greater, the concentration of aerosol would likely be reduced, but the decision was made to have a fixed value because making the ceiling higher is not a measure that can be implemented in real life.

The toilet dimensions are slightly larger than the recommended dimensions of an accessible toilet [32] room at 3 metres by 3 metres or 6 by 6 patches.

Implementation

The room dimensions were implemented by assigning each patch a patch type, e.g., 'wall' or 'door'. Each patch type is given its own colour to differentiate them from each other. The wall patches are coloured black, and these will serve as the boundaries of the restaurant. Only patches within the boundaries of the walls will be included in the viral aerosol concentration calculations.

4.2.2 Seating configurations

Parameterization

Before the COVID-19 pandemic, it was not uncommon to see tightly packed tables in restaurants, often with less than 1 metre between tables and backs of chairs being pressed together.

One of the focuses of the UK government was to highlight the importance of the 2m rule. This was suggested after SAGE determined with high confidence that physical distancing is '*an important mitigation measure*' [33].

A distance of 2 metres was chosen because most larger droplets quickly fall to the ground within roughly 1 metre due to gravity [34]. Therefore, having a rule of 2 metres reduces a person vulnerability to virus harbouring droplets. SAGE also states that '*current evidence suggests that 1m carries between 2 and 10 times the risk*' [33], which is yet another reason for the 2-meter rule being chosen.

Implementation

This feature was implemented using a switch to toggle between the physically distanced and non-physically distanced configurations. The user will select which layout to use for the experiment, and in the setup method, an if statement checks whether the switch is true or false. When false, then a set of methods are called to spawn and set the tables and chairs' locations as they would typically be. When true, a different set of methods spawn and set the locations of the tables and chairs with 4 patches between seats of neighbouring tables.

Figure 13. NetLogo switch

Each breed of turtles has their own method to spawn and set their locations. In a physically distanced seating arrangement, the position and number of tables and chairs must change as well as the position of nodes and links that form the pathways through the restaurant.

All spawn methods follow a similar structure, each having a list of the turtle's coordinates and how many turtles of that breed are required. The program runs a while loop for the number of turtles required and sets their x and y coordinates to the values in the corresponding position in the coordinates list. At this point, the program also initialises the turtle's colour, size and shape and sets its table or chair number if required and other turtles specific variables.

In Figure 14, we can see a replication of The Kailyards typical seating configuration and the proposed physically distanced layout with 2 metres between chairs of separate tables. With more efficient use of the floor space, it is possible to spread tables 2m apart. The side effect of physically distanced seating is a reduced seating capacity of 10 fewer people, indirectly lowering cases by reducing how many people are exposed to the virus.

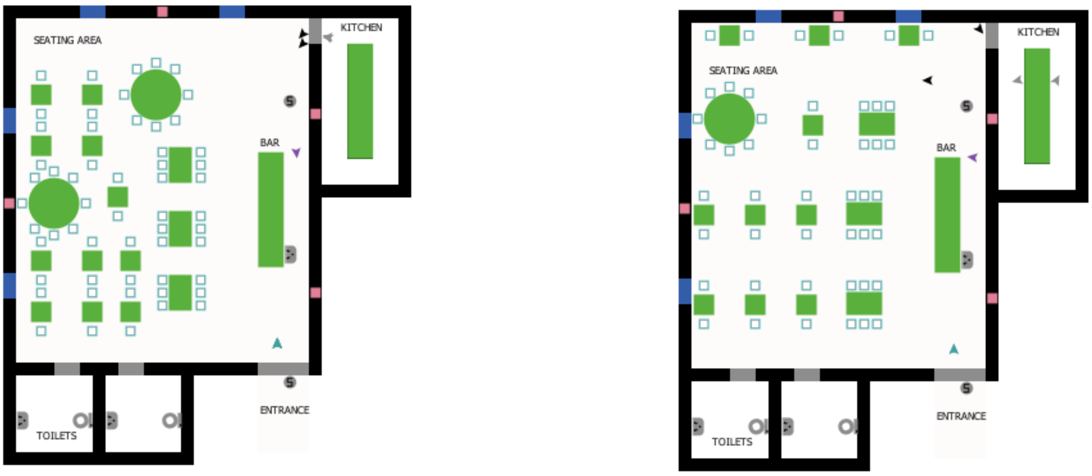

Figure 14. Comparison of physically distanced and non-physically distanced seating configurations

4.2.3 Node network

Design

A network of nodes must be designed that allows the mobile agents to navigate the restaurant. Nodes will store important information relating to the current location, such as node type. The nodes then must be connected to form the pathways they can follow using a pathfinding algorithm.

Implementation

A network of interconnected 'node' turtles are places to form the pathways through the restaurant. A node is placed at every point that a mobile agent might want to change direction on its journey between locations.

These nodes are connected using undirected links meaning they can be travelled both ways. The links are formed to create paths that allow agents to navigate obstacles like tables and capture how people move in real life.

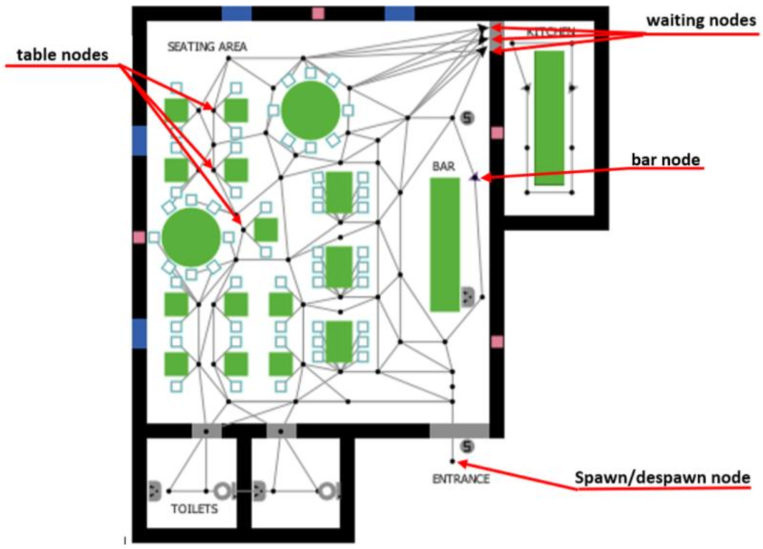

Figure 15. Image of restaurants node network with node types labelled.

Nodes are assigned a node type that marks if the node is a table node, waiting node, bar node or spawn/de-spawn node. A table node can be found next to each table and is assigned a matching table number. This table node will be used as the waiters' service point where they will stand when serving the table. The table nodes also store further details about the table's status, e.g., do they currently have food, do they need an order taken, table size, arrival tick, and other details specific to table nodes used for seating and scheduling. The decision to use table nodes to store these details rather than the table itself was to give waiters a place to navigate directly as they cannot navigate directly on top of a table. Also, due to how NetLogo works, the waiter can easily ask, 'is the node here my final destination?' rather than finding a way of checking the table that is not connected to the network or on the same patch as the waiter.

Waiting nodes are used as a place for waiters to wait while they are not serving the customers. They are placed next to the service window, so it would also be where the waiters give orders to the kitchen and pick up food. The bar node simply a waiting node for the bartender.

The spawn/de-spawn node is where customers will spawn before their meal and de-spawn after. It is placed just outside the restaurant's front door, so customers must pass through the door, which is crucial as the door will have a high amount of hand traffic, making this a significant contributor to fomite transfer.

Nodes and links can be hidden using a switch if the user wants to emphasize aspects of the simulation other than the movement.

4.3 Movement sub-model

The movement sub-model is the part of the simulation that controls the agent's movement between nodes, including the path they follow, when they go somewhere, and what they do when they get there.

4.3.1 Node network navigation

Design

We have a network of nodes; we now need to program the mobile agents to use it. As mentioned in the 'State-of-The-Art' section, the pathfinding algorithm used in the simulation is the best-first search algorithm. It uses a greedy heuristic to choose the best possible option possible for any given move, regardless of if it will result in the absolute shortest path.

Implementation

Waiters, customers, and bartenders all have their own method to handle their movement. All three of these movement methods are similar, but some functionality differences prevent using a shared movement method.

For example, the process of a waiter going from the starting location to the final location begins with the waiter checking the status of table nodes to find the tables that need attention. One of these nodes will be set as the final location for the waiter, and the table node will set its 'beingServed' Boolean to true to prevent other waiters from attending the table. At this point, the waiter will have the 'busy' Boolean set to true to stop it from checking for other tables that need to be served.

Starting from one of the waiting nodes (blue, Figure 16), the program will look at all neighbouring nodes (red) and calculate their distance from the table node (black). The closest node (green) will be set as the next location, and the waiter will step towards the next location 1 patch per tick to mimic how a person would walk. When this next location node is reached, it is set to the current location and a new next node is set in the same way. This is repeated until the final location is reached. The process is repeated when returning to the waiting nodes. Bartenders find their path in a similar way.

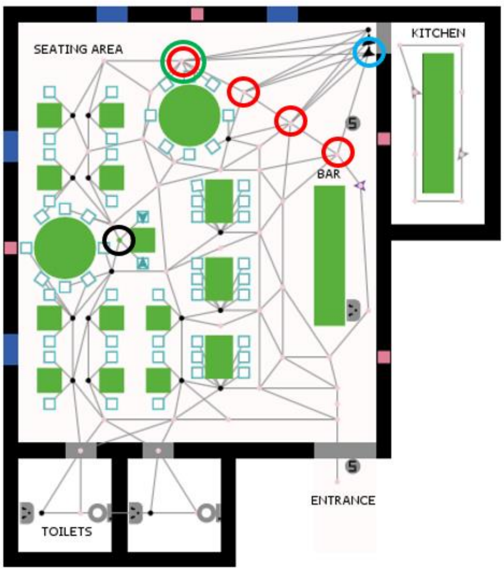

Figure 16. Pathfinding process of selecting next node.

Customer movement also works similarly. The customers are spawned in a group that matches the size of an unoccupied table. They receive an ‘assignedTable’ number and will navigate to the table node with the matching table number. When they reach this node, they are assigned a chair number and move to that chair, setting the customers ‘atTable’ and ‘sat’ Booleans to true to mark they are sat at the table.

4.3.2 Scheduling

In the simulation, there is a schedule that each table follows. The scheduling method controls the stage each table is at.

4.3.2.1. Table states

A table can be in the following states:

State	Triggered action
needsDrinksOrder	Bartender must take drinks order
needsDrink	Bartender must deliver drinks
needsOrder	Waiter must take order
needsFood	Waiter must deliver food to table
needsCheckback	Waiter must check up on table
needsCleared	Waiter must clear the table
needsBill	Waiter must deliver bill
needsSanisitised	Waiter must sanitise the table after customers have left

Figure 17. Table showing the triggered action of each table state.

Implementation

As mentioned earlier in section 4.1.3, a table's state is stored in the corresponding table node. The table node also stores if the table is occupied or not and the entry tick of the table (tick table became occupied). Using this information, the schedule method calculates the length of time the table has been occupied and divides the dining time limit by this value resulting in a ratio value representing how far along in the meal the table is. To make this clearer, here is an example:

Two customers enter the restaurant and are assigned a table. At this moment, the program takes note of their entry tick: e.g., 100. The booking length/dining time limit is set to 60 mins.

```
ask nodes with [occupied = true and nodeType = "table"] [

    let timeOccupied ticks - entryTick + 1

    if (diningTime * 60) / timeOccupied < 20 and visits = 0[
      set needsDrinksOrder true
 …
```

Figure 18. Code snippet showing the table state progression calculation.

The program tracks how long the customers have been at the table (timeOccupied). It then converts the diningTime from minutes to ticks and divides this by the timeOccupied. If the resulting value is less than a 20th of the dining time (in this case 3 mins) and the bartender has not visited the table, needsDrinksOrder is set to true. Only when the bartender visits the table and the timeOccupied is greater than 3 minutes will needsDrinksOrder be set to false. The process repeats for the next state, but rather than a 20th of the dining time, it will be another value, e.g., the table will receive their drinks after 5 mins or a 12th of the way through their meal, assuming the table is on schedule.

If the restaurant was understaffed with too few waiters, the tables will remain in the state they are in until a waiter visits the table, when they will then progress to the next state. This will result in longer exposure time.

4.3.2.2. Action priority

There is an action priority system in place that was implemented by placing if statements in a particular order which can be seen in the table above. If there is one table waiting for an order and another waiting to be cleared, the table waiting for an order will be given priority as this is typically how restaurants operate.

4.3.2.3. Table contact time

The table contact time highly depends on the type of restaurant. At a three-course, full-service restaurant, a waiter's table contact will be far higher than that at a fast-food restaurant, for example. In the case of this restaurant, there is a medium number of table contacts rather than modelling either extreme so that findings might be more representative and general.

Due to a lack of data on how long the average table interaction is, this was estimated to be 30 seconds based on my experience as a waiter. Some interactions like order taking could take slightly longer, whereas checking back on a table's food or delivering food would typically take less time.

4.4 Infection sub-model

The infection sub-model captures everything to do with an agent's virus intake and output. As mentioned previously, it can be broken further. The first part is an infected person's virus emission through aerosols, droplets, and fomites. The second part is a susceptible person's infection risk, and the last part is any protective measures an individual can take, such as wearing a face mask and washing their hands.

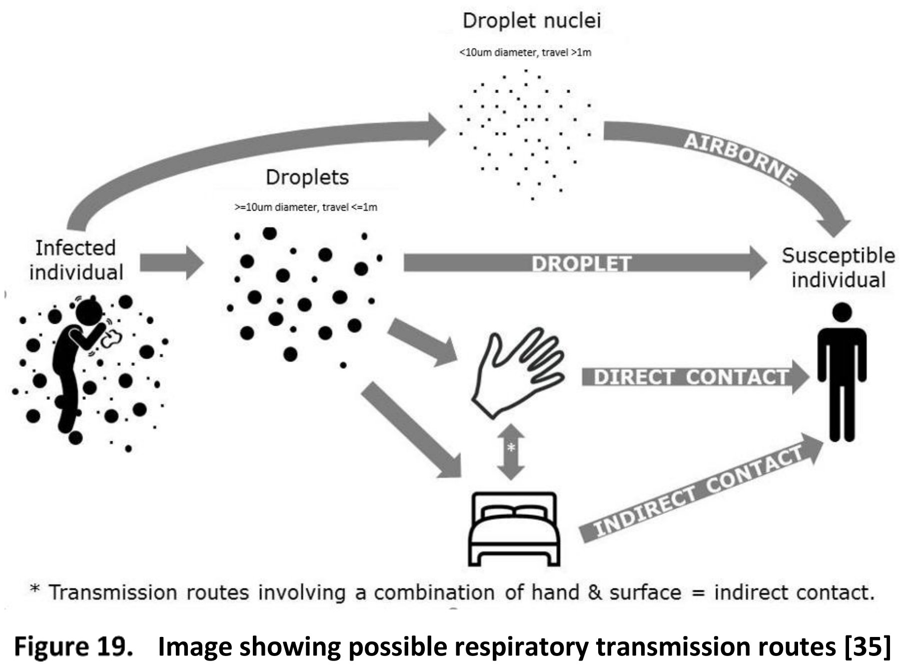

Figure 19. Image showing possible respiratory transmission routes [35]

4.4.1 Virus emissions

Virus emission can be affected by many factors such as how far along the person is into their infection, how severe their symptoms are, and their voice amplitude. Virus particles are transported out of the body through respiratory droplets, which come in many different sizes. The person also obviously must have been infected themselves.

4.4.1.1. Chance of being infectious

Parameterization

This will represent the region-specific prevalence of the virus. At the time of writing, in the Stirling area, the 7-day positive rate per 100,000 population is 53.1 or 0.053% [36], but this should be able to be altered to represent a variety of virus prevalence.

Implementation

The 'chance of being infectious' parameter is implemented using a slider in the user interface. When agents are spawned, they are set to be infectious or not based on the probability set by the slider. For example, if the chance of being infectious was 50% when an agent is spawned, the program generates a value from 0 to 10,000. If this value is less than the slider value multiplied by 100 (5000), then the agent is set to be infectious using a Boolean. The reason for multiplying by 100 is to allow for percentage increments as small as 0.001%. This is required as the region-specific virus prevalence must be represented accurately.

4.4.1.2. Aerosols

An infectious person will output viral aerosol particles when they breathe, talk, cough, and sneeze. Each respiratory event produces a different number of aerosol particles. An aerosol is defined as a particle that is less than 5-10μm. For reference, these particles are roughly 1/20th the size of a grain of salt and are invisible to the naked eye [37]. These particles are so light they can remain in the air for over 3 hours in poorly ventilated spaces [34] as they are not heavily affected by gravity like larger particles.

The defined size of an aerosol droplet is a heavily debated topic within the virology world, some saying an aerosol is any particle less than 5μm and some saying anything less than 10μm. In this simulation, it is assumed that an aerosol is any particle smaller than 10μm as the data found on virus emissions presumes the same.

Parameterization

Vocalisation

The UK Government has advised that restaurants should take steps to '*mitigate the increased risk of virus transmission associated with aerosol production from raised voices, such as when speaking loudly*' [38]. Suggested measures include lowering the volume of music and televisions as people will be forced to raise their voice to compensate for the background noise, which can highly affect their aerosol emissions.

A study was conducted to measure the exhaled respiratory particles through different speech events. The results were gathered by having 12 singers of various voice depths perform a series of talking and singing exercises in an airtight chamber with an aerodynamic particle sizer fitted to their face [39].

This experiment showed that breathing produced a median of 135 aerosol particles per second, normal talking at around 60db produced a median of 270 particles per second, and loud talking at around 70db produced 570 particles per second, showing a massive increase in aerosol particle output as volume increases [39]. It has been demonstrated that '*normal* *breathing over time can generate more viable virus aerosol than coughing, since the latter is a less frequent activity'* [40].

A noisy restaurant with loud talking is roughly 70db [41]. A restaurant at normal volume is around 60db, and a quiet restaurant is estimated to be about 50db with the background noise of people walking, chairs moving and glasses clinking etc.

We can assume that a person will not be talking for 100% of their meal, so it has been estimated that a person will spend maybe 60% silent (just breathing) while listening to conversation and eating and the other 40% talking, therefore an average of particles produced by breathing and talking will be used for talking in the simulation. This results in 189 particles output at 60db and 309 particles output at 70db.

Coughing and sneezing

Coughing is estimated to produce roughly 3000 aerosol particles and sneezing over 13 times that at approximately 40,000 aerosols [34].

The plume of aerosols produced from a cough can reach up to 3m, and a sneezes plume can reach 6m [42]. Over time they can be carried over large distances.

A continuous cough is one of the main symptoms of the COVID-19 disease. The NHS defines this as '*coughing a lot for more than an hour, or 3 or more coughing episodes in 24 hours*' [43].

Sneezing is not listed as a primary symptom of COVID-19 but must be included in the simulation as people can sneeze regardless of if they are ill or not. If an infected person does sneeze, an extremely high number of viral aerosols and droplets will be produced, putting surrounding people at a heightened risk of infection.

Viable virus

Not all respiratory particles output will contain viable virus particles that could lead to an infection. It has shown that breath samples report 26.9% of droplets produced will contain the virus [44]. This is measured by taking a breath sample for an infected person and testing for RNA (ribonucleic acid) to detect the past presence of SARS-CoV-2.

Being a relatively new virus, there are some gaps in the research for SARS-CoV-2. Unable to find data on the viable virus produced by a COVID-19 patients cough, we can use data from an influenza patient's cough to estimate the amount of viable virus that a COVID-19 patient's cough would produce. An analysis of the aerosol particles produced from the influenza patients cough found that 35% of aerosols produced contained viable virus [45]. Therefore, as influenza and COVID-19 are similar respiratory viruses, we will estimate that COVID-19 could be roughly the same.

<u>Implementation</u>

Concentration

Each patch stores the number of viral aerosols in that area. Rather than storing the number of viral aerosols per patch from floor to ceiling, the program only tracks the number of aerosols at a person's head height as these are the only particles they will be exposed to / inhale. This is done by dividing the total number of aerosols per patch by 5.4 (ceiling height * 2) to divide the floor to ceiling height into 0.5m layers.

The output graph that tracks the number of viral aerosols displays the average concentration across all restaurant patches. This is calculated by adding all indoor patches concentration and dividing by the number of indoor patches. All patches with a patch type: 'wall', 'window', 'door' and 'outside' are excluded from these calculations. The only patches included are the ones inside the walls with patch type 0 / indoor patches.

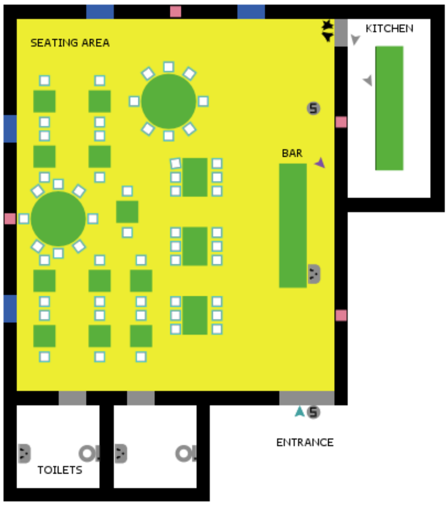

Figure 20. Yellow coloured patches showing patches included in aerosol concentration calculations.

Vocalisation

Different voice amplitudes are represented in the simulation using a slider where the user can vary the restaurant volume between 50, 60 and 70db. Using an if statement, the corresponding particle emission is set and will be output by each infectious agent per second.

In a room without strong air currents caused by having windows open, the air will remain mostly still apart from slight draughts caused by people walking around, doors opening periodically and a slight pull from the air conditioning. Thus, the highest concentration of aerosols will remain in the surrounding 1m of the infectious agent, which will fade into the surrounding area with a percentage of aerosols being immediately distributed around the restaurant.

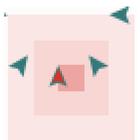

Figure 21. Area covered by aerosol particles output from breathing and talking.

Coughing and sneezing

As the aerosol plume from a cough will reach roughly 3 metres and spread out in a cone-like shape, the in-cone function in NetLogo has been used to disperse the particles evenly between all patches in a 45-degree cone on front of the agent, with a distance of 6 patches (3m).

A sneeze works using the method with a 30-degree angle and 12 patches (6m). The angle is tighter as a sneeze is more forceful, so particles will mostly be carried further and will have less spread closer to the agent.

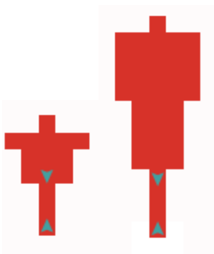

Figure 22. Distribution of aerosol particles for a cough (left) and sneeze (right)

To control symptom severity, the user can adjust a slider to control coughing and sneezing frequency. They can both be independently turned off and on, which is required to represent asymptomatic people or focus specifically on breathing and talking aerosol output.

Viable virus

The simulation shows the concentration of viral aerosols as not all particles contain viable virus. Of all aerosols produced by breathing, only 26.9% have viable virus, so the aerosol output will be multiplied by 0.269. A similar method will be applied to coughs and sneezes as roughly 35% of aerosols produced by these forceful respiratory events contain viable virus, so aerosol output will be multiplied by 0.35 to represent this.

4.4.1.3. Droplets

As mentioned previously, in this simulation, we are classifying an aerosol as any respiratory droplet less than 10μm; therefore, a regular respiratory droplet is any particle sized 10μm or greater.

Parameterization

The UK government encourages the 2-metre rule as research suggests that the majority of droplets produced will fall to the ground within 6 feet (2 metres) [9] due to gravity.

When a person coughs or sneezes, they will project a spray of respiratory droplets. It is difficult to represent such a wide variety of droplet sizes in NetLogo as they cannot be implemented as a concentration due to their high variability. The droplets will also have a wide variety of projection paths. Most droplets will fall within two metres, but without tracking individual droplets, there is no way of knowing exactly where they will land, e.g., past a person's head, onto their face or their clothes and transferred to the mouth indirectly. This is perhaps a limitation of carrying out the NetLogo project as it cannot model such specific scenarios.

Implementation

Respiratory droplets will be projected from an infectious person when they cough or sneeze in a similar way to how aerosols are projected using the 'in-cone' feature with an angle of 30 degrees and a distance of 2 metres. Rather than being represented as a concentration, any customers, waiters, or bartenders in the projection cone at the time of a cough or sneeze will have their infection risk increased by an estimated value. A cough will add 4 to a person's risk rating, and a sneeze will add 48, corresponding to 4 and 48 virus particles intaken, respectively.

4.4.1.4. Fomites

As mentioned in section 1, at the time of writing on the 5th of April 2021, the CDC has reported that '*contact with a contaminated surface has less than a 1 in 10,000 chance of causing an infection*' [10], rendering the fomite transmission feature somewhat obsolete.

Parameterization

In a restaurant, the objects with the most hand traffic are tables, and the handles on any doors, specifically the front door.

After carrying out some research, there does not seem to be any data quantifying how many virus particles someone is carrying on their hand. Also, there is no way of predicting how much contact a person's hand has with a surface, e.g., open hand firm contact or using one finger. Because of this lack of data, the simulation will not track fomite transfer accurately enough to have it contribute to the infection risk. This is not a huge issue as the chance of picking up the virus through fomite transfer is slim.

The simulation can track roughly how dirty a person's hand is by estimating how many times they are have contacted a contaminated surface and how many times a person touches their face. The risk can be quantified based on how dirty their hand is estimated to be and how many times the person touches their face and thus creates an opportunity for infection.

Infectious agents are assumed to have virus particles on their hands when they enter the restaurant that will be transferred to the surface they touch.

Implementation

Starting with the door, when an infectious agent travels through a door patch, the door is set to be fomite infected using a Boolean variable. When a susceptible agent travels over a fomite infected door, their hand contamination value increases by 1, representing one touch of a fomite infected surface.

As it is likely that an infectious person stays in their seat, only touching their own section of the table, the only way for a table to be contaminated is if it is coughed or sneezed on, not through contact. As NetLogo cannot divide agents into sections, once an infectious person coughs or sneezes on a table, the entire table is considered contaminated, as is coloured red to indicate that. Susceptible agents sat at a contaminated table are programmed to simulate periodic contact with the table, which will result in their hand contamination value increasing or, in other words, their hands picking up more virus particles as time goes on.

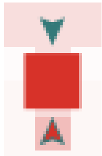

Figure 23. Fomite infected table

4.4.2 Infection risk

The infection risk represents the number of virus particles an agent has been exposed to through droplets or the inhalation of aerosols. To be infected by any virus, they must intake an infectious dose of virus particles. Currently, the infectious dose of the SARS-CoV-2 is unknown but is estimated to be roughly 300 particles [46].

4.4.2.1. Inhalation of virus

Parameterisation

To estimate how many virus particles a person is inhaling, we must know the concentration of viral aerosols in the surrounding air, the persons tidal breathing volume and their respiratory rate. Tidal volume is the volume of air being moved in and out of a person's lungs during a normal breath, which is 500ml of air for the average person with healthy lungs. The average person's respiratory rate is 12 full breaths per min or 1 inhalation every 5 seconds.

Implementation

The inhalation of viral aerosols is implemented by having agents inhale 500ml split into 100ml per tick to remove some variability from the results. A percentage of the viral aerosol concentration is added to the agent's risk variable. This is calculated for every agent on each tick using the following equation:

```
risk = risk + (viral aerosol concentration of patch-here / volume of patch-here) *
(tidal volume/inhalation rate)
```

Figure 24. Infection risk equation

Example:

```
Viral aerosol concentration of patch here: 50 aerosols
Volume of patch: 125,000ml
Tidal volume: 500ml
Inhalation rate: 5 seconds

risk = risk + (50 / 125000) * (500/5)
     = risk + 0.04
```

Figure 25. Example calculation of infection risk

4.4.2.2. Face touches

Parameterization

Although recently proven unlikely, the virus can be transferred into a person's body through touch if they have contacted a contaminated surface. It has been shown that, on average, a person will touch their face 20 times per hour/3 times per minute [47]. Each touch creates an opportunity for the virus to enter their system through the eyes, nose, or mouth.

As there is no accurate way of knowing how many virus particles are on a person's hand or where they are on the hand, this will be estimated by increasing the agent's hand contamination value every time they contact a contaminated surface. The person's fomite risk will increase every time they touch their face.

Implementation

An agent's hand contamination risk will increase by 1 each time they contact a contaminated surface like a door handle or table. A person has been estimated to touch the table once every three mins/180 ticks. This could be to pick up cutlery or a drink, touch the menu and similar actions. They will also contact the door on their way into the restaurant, which could be contaminated, increasing their hand contamination value.

Their fomite risk value is calculated by programming them to 'touch their face' every 3 mins/180 ticks. With each face touch, their hand contamination value will be added onto their fomite risk value.

4.4.3 Protective measures

Protective measures include the individual measures that will directly impact a person's virus emission and intake, such as mask-wearing or hand washing. This also includes protective measures that the restaurant can put in place, such as sanitising surfaces and reducing the restaurant's capacity.

4.4.3.1. Masks

Parameterization

Inhalation and exhalation efficiency

The efficiency of a mask can be measured by the percentage reduction of particles inhaled or exhaled. Inhalation and exhalation efficiencies are two separate values as certain masks only filter the air being inhaled and have an unfiltered exhalation value.

The efficiency is dependant not only on the mask's material but also on how well the mask is fitted/the seal around the person's face.

Mask type	Inhalation Efficiency (approx.)	Emission Efficiency (approx.)
N95 [48]	90%	90%
Surgical [48]	65%	65%
Cloth [48]	30%	30%
Face shield [48]	23%	23%
Exhaust value [48]	Dependant on mask type	0%
No mask	0%	0%

Figure 26. Table of inhalation and emission efficiency for different mask types.

As it is highly unlikely that every person in the restaurant is wearing the same type of mask, on average, the inhalation and exhalation values are estimated to be roughly 50% as most people will be wearing surgical or cloth masks.

These efficiencies also apply to larger respiratory droplets like the ones produced by coughing and sneezing. Some of these larger particles are forced through the mask's filter, but the distanced covered is reduced [49].

Wearing time

The amount of time the person is wearing a mask is also vital in reducing virus intake and emission over an extended period. In a restaurant, the customer would be unable to wear a mask for the entire meal as they need to eat and drink. Most restaurants only require customers to wear a mask until they reach their table, but as shown in poorly ventilated restaurants, the virus can be spread through aerosols to faraway tables [13]. This would suggest that to reduce virus emission even further, customers should wear masks until their food arrives. While this may be inconvenient to the customer, it may be worth it due to the considerable reduction in virus intake and emission.

The staff will be able to wear masks throughout the entire service.

<u>Implementation</u>

Inhalation and exhalation effectiveness

Masks can be toggled on or off using a switch in the user interface. The user can also set inhalation and emission efficiency percentages using two independent sliders. When the virus emissions are calculated, the number of particles emitted will be reduced by the efficiency percentage set with the emission efficiency slider. A similar calculation is done where the virus intake is calculated, which will reduce the virus inhaled by the inhalation efficiency percentage.

Wearing time

When a table receives their food, everyone at that table will have their mask removed. This was done by only calculating mask efficiency for customers whose table state is not 'hasFood'. The customers with food have their mask efficiencies set to 0%, and their virus emissions/intake will rise.

4.4.3.2. Hand washing

<u>Parameterization</u>

Hand washing, if done correctly, should reduce the number of virus particles on a person's hand. The handwashing thoroughness can be defined as '*the percentage of 6 surfaces across both hands (i.e., wrists, thumbs, and in between all fingers)*' [50]. The median thoroughness of a group of nurses was 83%. For the general population of non-healthcare workers, the thoroughness will likely be lower than this, perhaps around 70%.

<u>Implementation</u>

Hand washing can be toggled on or off using a switch in the user interface. The handwashing thoroughness will be defined using the slider to set a percentage value of contaminants removed. Handwashing facilities are provided through a sanitisation station outside the restaurant's front door and hand sanitiser on all tables. Hand washing is only implemented for customers, and we can assume that waiting staff are taking the required handwashing precautions. Possible fomite virus transfer between customers and waiting staff would be through passing plates or the bill. Ideally, this would have been implemented but could be a feature for future work.

As the customer's hand contamination value will be reduced by the percentage that has been set using the handwashing thoroughness slider, every time they touch their face, there is a lower chance of virus particles being on their hands which should result in a reduced fomite risk.

4.4.3.3. Exposure time

<u>Parameterization</u>

The longer a person is in a room with an infected person, the more virus particles they are likely to be exposed to. This will increase the likelihood that they will intake an infectious dose of the virus.

A potential way a restaurant could decrease people's exposure time would be to shorten the length of bookings, only leaving enough time necessary by making a booking slot 45 mins rather than an hour. This would prevent customers from sitting around, which would reduce the time an infected person can emit the virus and reduce the time a susceptible person is exposed to the virus.

To facilitate this change, it is possible that the restaurant would have to make some adjustments to service, such as leaving less time for orders to be taken, combining drinks and food orders or having a limited menu to allow food to be made faster.

Another way a restaurant could reduce exposure time would be to make the full length of service shorter. This could be achieved by closing an hour early.

For example, a 'pre-COVID' restaurant might have booking slots that are 1 hour long, and service may run for 4 hours, allowing for a table to be reused four times. With the safety measures in place, the restaurant would reduce booking slots to 45 mins by taking order more efficiently. This would then let the restaurant serve the same number of customers in a shorter length of time.

<u>Implementation</u>

Both the booking length and service length can be set using sliders. The length of a booking can range from 30 mins to 90 mins in 15-minute increments. As discussed previously in section 4.3.2, the scheduling is calculated based on how many ticks a person is through their booking in relation to the booking time set. If a booking slot of length 60 mins allowed for 20 minutes of eating, the 45-minute slot would allow 15 minutes. When the tables booking time limit is reached, and the waiter has given the table the bill, they will leave the restaurant and de-spawn.

Once the total ticks pass the service limit set by the user, customers will stop spawning, and the simulation will wait for all customers to finish their booking slots and leave. When all customers have left, the simulation will end.

4.4.3.4. Reduced capacity

Parameterization

Reducing the restaurants capacity will prevent over crowing. As a result, this will lower the probability of an infectious person being in the room. If there is an infectious person in the room, fewer people will be exposed to their virus emissions which will likely reduce the number of cases produced. Also, if there are fewer people in the room, then this allows for better physical distancing.

The maximum seating occupancy for the restaurant before COVID-19 would be 117, determined by '*applying an occupancy load factor of 1 persons per square metre, as prescribed in the Fire Safety Building Regulations'* [51]. This is 1 person per metre of useable floor space, not including the area behind the bar or blocking doorways. By applying the guidance of 2 metres between people on different tables, the restaurant's reduced maximum capacity is 50 people [51].

It would be advisable to reduce the capacity further if possible. This would not always have to result in a reduced number of customers. It could be possible to reorganise bookings to keep the occupancy at a lower steady rate rather than fluctuating between low and high occupancies.

Implementation

The reduced capacity feature was implemented using a slider where the user can set a capacity of 30, 40, 50, or 60 for a non-physically distanced seating configuration, or 30, 40, 50 for a physically distanced seating configuration.

When customers are being spawned, a check is made to ensure that the new customers will not exceed the capacity. If it will, then customers will not be spawned.

4.4.3.5. Surface sanitisation

Parameterization

Tables

If an infectious person is sitting at a table, it will likely have become contaminated with the virus due to the spray larger respiratory droplets settling on its surface. When a table of customers finish their meal and leave the restaurant, the waiter will sanitise the table for the next guest, which will kill the majority of the virus particles.

Door

To reduce the spread of the virus originating from customers using the front door, it should be periodically disinfected by a waiter every 30 mins or so.

Implementation

Tables

When the customers leave the table, the table state is set to 'needsSanitised'. This is the flag that lets a waiter know that the table is empty and needs to be wiped. A waiter will go over to the table and ‘clean’ it for 30 seconds, and the Boolean will be set to false, alerting customers that the table is clean and now available to be sat at.

Door

Sanitisation of the door is simulated by setting the ‘fomiteInfectedDoor’ Boolean of the door patches to false every 30 mins/1800 ticks, which will remain until an infectious person touches it again.

4.5 Ventilation sub-model

The ventilation sub-model handles the mechanical and natural ventilation configurations. This captures parameters such as which windows in the restaurant are open, the air refresh rate of the air conditioning and the air filtration of any portable air filters placed around the restaurant.

The ventilation of a room has been highlighted as a critical measure in reducing the virus's transfer [52] as poor ventilation allows for the build-up of viral aerosol particles. This will likely increase the virus intake of susceptible people, also increasing their infection risk.

‘*Building ventilation has three basic elements: Ventilation rate … airflow direction … air distribution or airflow pattern’* [53]. Ventilation rate refers to the quantity of outdoor air brought into the room, also known as the air change rate. The airflow direction is the general direction of the airflow through the building, ideally from clean air spaces to dirty. The air distribution or airflow pattern is how the clean air is distributed around the room to result in the most efficient removal of airborne pollutants [53].

4.5.1 Mechanical ventilation

Mechanical ventilation, sometimes referred to as forced ventilation, is any ventilation driven by fans with the goal of ‘*supplying air into, or exhausting air from, a room*’ [53].

Portable air filters could be considered air filtration, as opposed to mechanical ventilation. However, as these filters are driven by fans, they will be deemed a mechanical measure to remove aerosol particles from the air.

4.5.1.1. Air conditioning

<u>Parameterization</u>

The air change rate of a restaurant is the rate at which the air in the restaurant is replaced by ‘fresh’ outdoor air. An air change rate of 10 would mean that the ventilation system provides 10 times the room’s volume in air per hour. The recommended air change rate for a restaurant is 12 changes per hour [54]. The air change rate is controlled through the RPM/speed of the intake and exhaust fans. Higher fan speeds should result in a higher air change rate.

The Kailyards ventilation system consists of a large exhaust/extraction fan placed in the ceiling, with 4 smaller intake fans that pull in fresh air from outdoors, spread evenly around the room's perimeter. The intake fans are receiving fresh air from a centralised cooling unit on the roof of the building. This ventilation setup should reduce aerosol concentration as there is a constant intake of fresh air and extraction of potentially contaminated air from the room that should prevent a build-up of viral aerosols.

As seen in the restaurant in Figure 1, some air conditioning systems will not intake fresh air from outdoors but will recirculate the air around the room. These systems are more for climate control than ventilation as no air is extracted from the room, therefore no aerosols will be removed unless the unit is filtered. In the restaurant in China, a recirculating AC unit caused air currents that carried aerosols over 4 metres from the infectious person.

Although The Kailyard does not have a recirculating air condition configuration, the option to turn on recirculation will be provided to demonstrate its dangers.

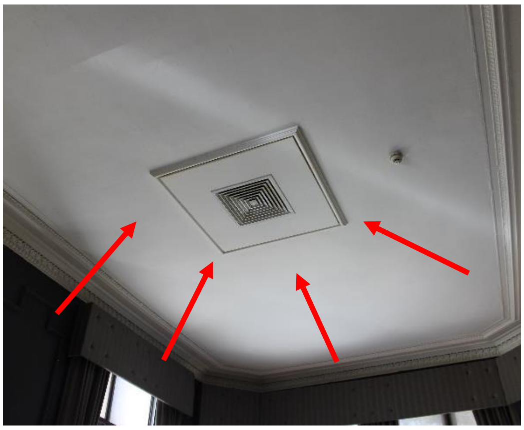

Figure 27. Images showing The Kailyard's air conditioning configuration. Intake (left) and exhaust (right). Red arrows showing air direction.

<u>Implementation</u>

The air conditioning was implemented using a slider that sets the air change rate of the air conditioning system. The program then calculates the percentage of air volume to be removed per tick based on the air change rate. For example, if the air exchange rate is 12 changes per hour, this corresponds to 1 change per 5 minutes, or 0.003% of air volume changed per tick. For each patch, the concentration/number of aerosol particles will be reduced by 0.003%. The equation for this can be seen below:

```
Reduced concentration = current concentration - (current concentration *
((100 / ((3600 / air change rate))) / 100)
```

Figure 28. Air change concentration calculation

4.5.1.2. Portable HEPA filters

A HEPA (High-efficiency particulate air) filter is a type of high-grade air filter. These are the filters that are often found on planes and in medical settings.

<u>Parameterization</u>

A HEPA filter can remove at least 99.97% of aerosol particles as small as 0.15μm [55]. This should filter out all virus-containing aerosols from the air that passes through the filter. An example of a portable HEPA filter can clean up to 500m^3/hr [56].

This is only suggested as a final option as these portable filters can reach £1000+, so it will likely exceed many restaurants' budget. A portable HEPA filter could be used if the ventilation is very poor and there are no other means of increasing it. Another use case would be in colder or rainy climates, which prevents the restaurant from opening the windows for natural ventilation.

<u>Implementation</u>

This feature is implemented using a slider. This will place 0 to 4 portable air filters around the room. Each unit will provide 1 added air change per second as even though it does not replace the air with outdoor air, it will essentially have the same effect at reducing viral aerosol concentration. The number of air filters, set with the slider, is added to air change rate as 1 filter provides 1 ACH worth of clean air.

Ideally, the airflow would be simulated flowing in and out of the air filter, but the priority was given to the natural ventilation.

4.5.2 Natural ventilation

Natural ventilation is any ventilation coming from natural forces that '*drive outdoor air through purpose-built, building envelope openings.*' These openings refer to the like of windows, doors, and trickle vents.

4.5.3.1. Windows

<u>Parameterization</u>

Air change rate

This simulation will only cover natural ventilation through windows as The Kailyard does not have any doors that lead outdoors as it is accessed through another building and does not have any trickle vents.

The effectiveness of natural ventilation depends on several factors. The first being human behaviour, as the windows will likely have to be opened by a human. The person has the option to partially open or fully open the window. Fully opening the window will be more effective than partially opening it as the larger opening area will allow for greater air change.

The second factor which will impact the effectiveness of natural ventilation is building location. If the building is in a city surrounded by many buildings, the wind will not flow as efficiently as if the restaurant was in the open countryside [57]. The Kailyard is in a relatively optimal geographic position to get good wind flow into the building. The building is on top of a hill which puts it higher than the surrounding area. This means that the wind will be able to take a relatively unobstructed path to the building's windows.

This takes us to the next two factors that will impact the natural ventilation, the wind speed and direction. At the location of The Kailyard, Dunblane, the yearly average wind speed is 5.5 m/s with a predominantly westerly wind direction [58]. On the day that the airflow measurements were taken, the wind was blowing in a west/south-westerly direction [59]. The building is west facing, meaning that the breeze on this day was directly into the windows on the façade of The Kailyard.

Figure 29. Image showing geographic elevation of Dunblane Hydro, the hotel housing The Kailyard restaurant [60]. The Kailyard circled in red.

Using the Air Change Rate Calculator [57] from Window Master, it is possible to enter details from The Kailyard such as building location, window height from ground to opening, room volume, window area, orientation, and location. The calculator outputs a graph that estimates the air change rate provided by the given window configuration. The calculator was used three times, the first with just the front windows open, the second with just the side windows open, and lastly, with both groups of windows open.

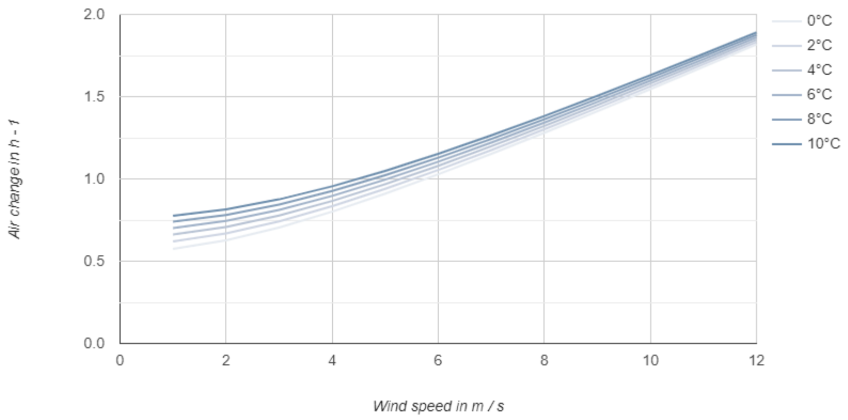

Figure 30. Graph showing the air change rate provided by opening only the front windows.

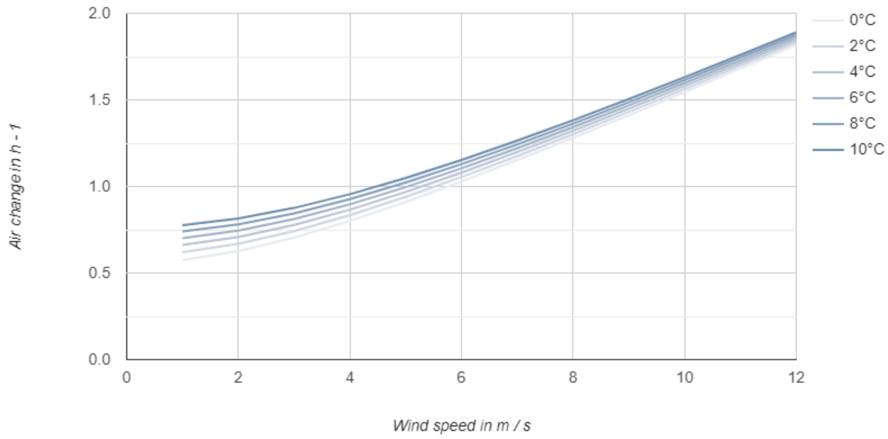

Figure 31. Graph showing the air change rate provided by opening only the side windows.

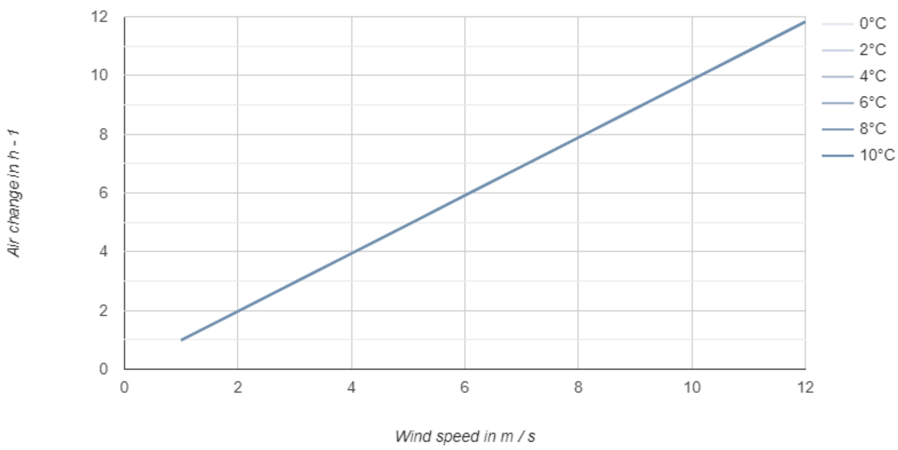

Figure 32. Graph showing the air change rate provided by opening both front and side windows.

By reading the graphs, we can see that for the average Dunblane wind speed of 5.5m/s, the air change provided is 1 air change per hour for only front or side windows open. With both sets of windows open, the air change rate provided is 5 air changes per hour.

Air paths

The information needed to simulate the restaurant's air paths was gathered using an in-person experiment in The Kailyard. This involved dividing the restaurant's floor into a 2-metre grid and taking an air direction measurement at each intersection. Air direction measurements were taken using an incense stick that produces smoke, taking note of which way the smoke would flow, as it will follow the airflow.

Figure 33. Image showing how air direction measurements were gathered in The Kailyard.

The first measurements were taken with all windows closed as a control. In this experiment, the smoke produce rose in a straight line, suggesting there were no initial air currents in the restaurant with the windows closed.

With the front windows open only, the following air paths were measured:

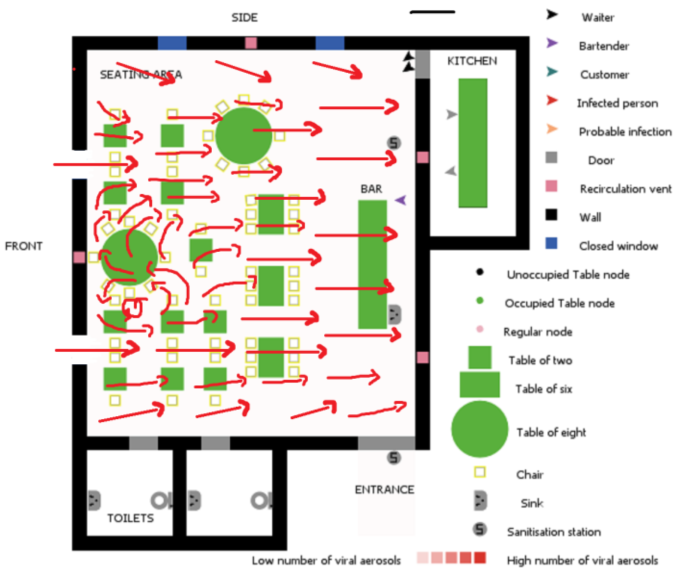

Figure 34. Image showing the air paths throughout the restaurant with front windows open only.

With the side windows open only:

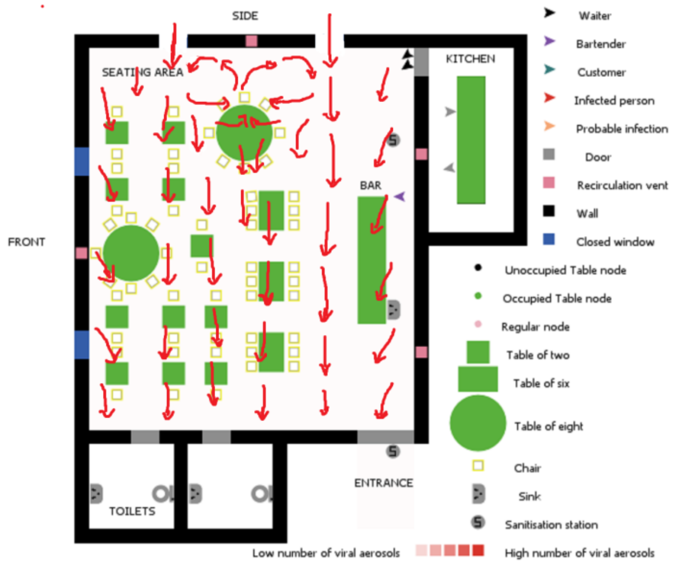

Figure 35. Diagram of air paths through the restaurant with side windows open only

With both front and side windows open:

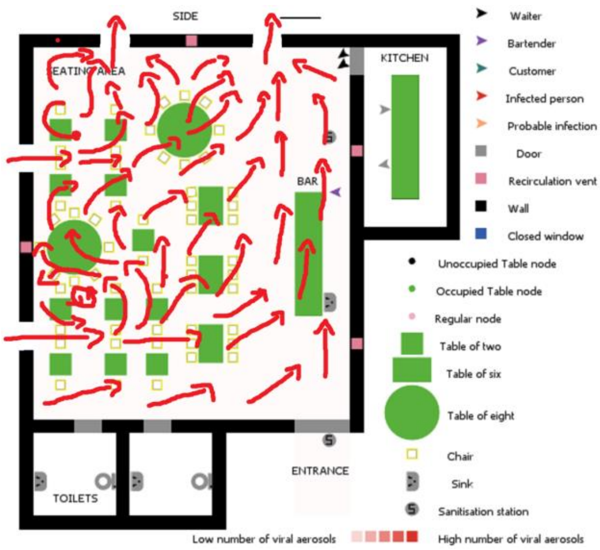

Figure 36. Diagram of air paths through the restaurant with both front and side windows open

When only one set of windows are open, the cooler outdoor air will come in the window and fall to the lower half of the room. As it moves through the room, it will heat up, and when it reaches the other side of the room, it will rise and be pushed along the ceiling back towards the window, where some of the air will flow out the top of the window. This image demonstrates the effect in a similar fashion:

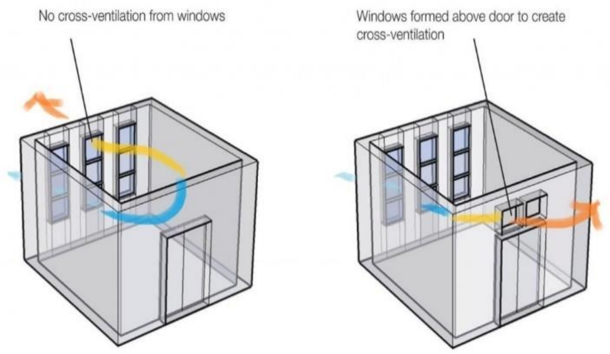

Figure 37. Graphic showing the movement of air in rooms without cross ventilation [61]

<u>**Implementation**</u>

The windows are opened using two independent switches, one to open and close the front windows and another to open and closed the side windows.

Air change rate

When both windows are open, the natural ventilation air change rate is implemented by adjusting the patches' airspeed to ensure that the aerosol particles are removed from the restaurant through the exhaust windows. When the concentration is transferred into the exhaust window patch, it is set to 0 to represent it being extracted to the outdoors.

If only either the front or side windows are open, the aerosols are moved from patch to patch in the direction of the airflow until the opposite wall is reached. At this point, to simulate the aerosols being carried up and along the ceiling, back towards to window, the aerosol concentration is reduced. In NetLogo, it would not be possible to have two different air currents overlapping each other in opposing direction as the simulation is 2 dimensional. Also discussed earlier, the aerosol concentration is only represented for the layer at head height.

The airspeed is implemented by moving a percentage of the concentration of a patch to the neighbouring path in the direction of the air current. The larger the percentage moved, the faster the airspeed. This was adjusted until is produced the correct air changes per hour.

An attempt to implement different airspeeds throughout the restaurant was made so that the airspeed of patches close to the windows would be higher than those in the middle of the room. While doing this, it was discovered to result in a build of virus particles on the boundaries between different speeds. This is because the higher speed patches move aerosols into the slower speed patches faster than they can move them to the next patch. Due to this issue, all patches must have the same airspeed.

Air paths

The air paths were implemented by dividing the seating area patches into a grid of 3 x 3 chunks and assigning all patches belonging to that chunk the same grid square number. Each patch in a chunk would initially be set the same air direction in the general direction of the airflow as a base. Much fine-tuning was required to capture the air paths as precisely as possible. This involves overwriting the initial direction with one that better represents the actual direction. Using this method allowed the large areas, where air direction was all in the same direction, to be set with one line of code instead of 9/one for each patch in the chunk.

The air direction is measured in degrees, with 90 degrees meaning the air is moving to the right. If the air direction of a patch was 90 degrees, every tick, a percentage of the aerosol concentration would be subtracted and added to the neighbouring patch to simulate the aerosols being blown in that direction.

Another method used to simulate the air direction more precisely was the use of 'push/pull' patches. These are patches that would allow the air to be dispersed more evenly in areas where a build-up would occur and make smoother turns in certain areas using two directions. The first would be the direction that the air is coming from, which would 'pull' a percentage of the aerosol concentration from that patch into the current patch. The second direction is where the air must go. This simultaneously pushes a percentage of its own aerosol concentration into the neighbouring patch in the 'push' direction.

Air direction must be a value from 0 to 360 degrees in a 45-degree increment, e.g., 0, 45, 90 etc. This is because of the grid layout of patches in NetLogo. Any air direction from 30 to 60 degrees and distance 1 will result in the same patch being selected.

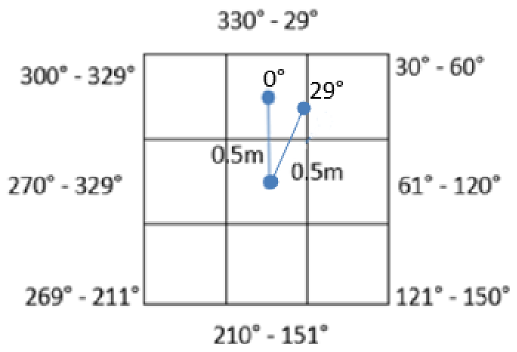

Figure 38. Diagram showing how air direction is calculated on a grid of patches in NetLogo using the angle of direction and distance.

The air paths in the restaurant and air direction of a patch will be changed depending on which windows are open, using the airflow control method. This method checks the state of the switches to see if the windows are open or not and sets each patch's air direction and speed.

4.6 User interface

The user interface went through many iterations as more features were added into the simulation. The initial design principles remained the same throughout. A wireframe of the user interface was used to plan the layout and aid its design:

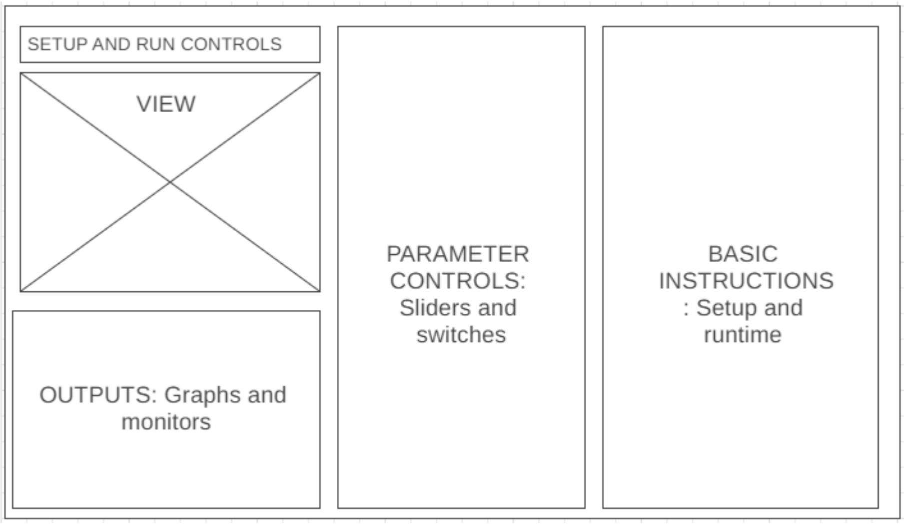

Figure 39. User interface wireframe

4.6.1 View

Design and Implementation

When designing and implementing features in the simulation, it must be ensured that some basic criteria is met, such as ensuring everything in the view is shown clearly, there are no contrast issues and that all components are well labelled.

Each component in the simulation has been given its own colour to differentiate it from other similar shapes, e.g., waiters and customers are the same shape as they both represent people, but waiters are coloured black, and customers are blue. Similar colours were avoided wherever possible.

To clarify what patches and agents are trying to represent, a legend was added into the view, which labels every patch and agent to show what they are.

4.6.2 Controls

Design and Implementation

The UI was made user friendly by organising the controls into categories where possible, such as infection controls or ventilation controls. The goal of this is to guide the user through the setup process. A set of instructions has been added for the user to read before using the simulation that walks them through setting up parameters and running the simulation. A brief explanation of each control and output graph/monitor has been provided if the user is ever unsure of what one of the controls does. This information is also found in the Info tab, but as the users were unfamiliar with NetLogo, they were not aware of this feature.

The UI allows the user to continuously run the simulation or step forward 1 tick/second, allowing specific outputs and actions to be observed in greater detail.

4.6.3 Outputs

Design and Implementation

Three output graphs are shown in a group at the bottom of the UI. They show the important data relating to the number of viral aerosols in the air, the fomite infection risk faced by agents and the aerosol and droplet infection risk. The output graphs and monitors are arranged into subcategories that should help the user find the output data they are looking for.

5 Testing

This section will cover validation testing to provide evidence that each requirement implemented functions as it should. This section also evaluates all safety measures that were implemented based on the number of probable infections produced and the average risk faced by the people in the restaurant.

Some aspects of the simulation can have quite a high degree of variability, but during the testing process, using 40 repetitions has produced consistent results. If there had been more time available, then a higher number of repetitions would be preferred.

5.1 Validation testing

5.1.1 General safety parameters

5.1.1.1. Capacity

To check that the capacity limit functions correctly, a test was run with the capacity set to 40 people tracking the maximum number of people within the restaurant at any given time. By looking at the data produced over multiple repetitions, we can see that the capacity never exceeds the set capacity of 40 people.

5.1.1.2. Length of bookings

When testing to ensure that customers leave once their booking slot is finished, we can track the average duration that a customer is in the restaurant for across multiple repetitions. If there is enough staff on shift to serve the customers in a timely fashion, the booking limit will not be exceeded. By setting the booking time to 30 mins, we can observe that the average time spent in the restaurant was 1830 ticks or 30.5 mins, over by the booking slot by 30 ticks to allow walking to and from the table. If there is not enough staff, customers will be forced to stay longer, just like in real life.

5.1.1.3. Length of service

To ensure that no customers enter the restaurant after the service time limit, we can run a test that stores the entry tick of the last customer to enter the restaurant. If the length of service is set to 3 hours, there should not be any customers entering the restaurant after 10800 ticks. The test found that the last customer enters the restaurant five minutes before closing time at 10500 ticks.

There is a minor bug in this part of the system that occurs exceptionally rarely and does not affect results in any way. All customers will leave the restaurant, but there will sometimes be one customer who will not move and stays seated.

5.1.2 Movement

5.1.2.1. Pathfinding

The node network pathfinding was tested using the manual customer spawning feature. This feature allows the user to select what table a customer is assigned to sit at. By going through each table in both seating configurations, it was possible to observe the paths that the customers, waiters and bartender took to the table and adjust the network accordingly by adding/removing links and nodes to improve the path the agent followed.

Figure 40 below shows an example improvement that was made. The red route was the old path travelled by the waiter to the table, and the blue route is the improved route made possible by adding a new link to the table and removing the link that caused the path to be diverted.

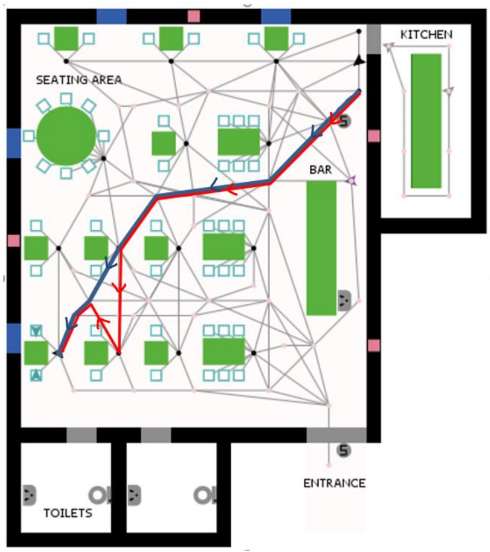

Figure 40. Image showing an improvement made to the node network after testing.

5.1.2.2. Scheduling

To ensure the schedule is working correctly, we can inspect the node of an occupied table. Verifying that the states are updating following the defined schedule confirms the scheduling is working as it should. This also confirms that the waiters will visit a table when the table state indicates that it should.

5.1.3 Virus emissions

5.1.3.1. Manual and automatic infection

Manual infection was tested by clicking on an agent in the simulation and observing if it became infected, which it did. The user testing of this is discussed in chapter 6.

Automatic infection can be tested by running the simulation for many repetitions and taking an average of the percentage of infectious people produced. This should be similar to the percentage set by the 'chance-of-being-infectious' slider. Using parameter sweeping, we can simulate a range of values from 0 to 100. From this, we can see if the percentage of infectious people produced matches the expected percentage.

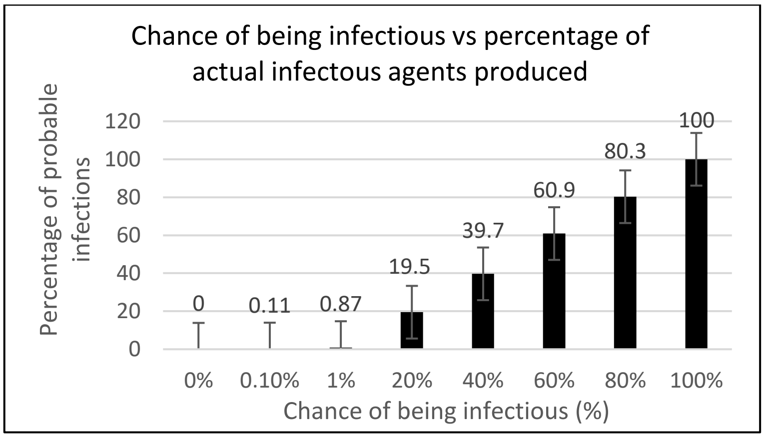

Figure 41. Graph showing an agent's chance of being infectious vs the actual percentage of infectious agents produced in the simulation.

Figure 41 shows that the chance of being infectious set by the user translates into the simulation as it produces a similar number of infectious agents. The percentage of infectious agents varies slightly from the chance of being infectious by a few fractions of a percent, but it is suspected that running the test for a higher number of iterations will result in a value closer to the one set.

5.1.3.2. Aerosol emission

Concentration

To ensure that the correct number of viral aerosols are being output into the air, we can ask the NetLogo Command Centre to colour all patches with a concentration greater than 0. This allows us to see which patches contain viral aerosols and ensure that the simulation is not mistakenly including the wall and outside patches in the virus emission calculations or projecting sneezes outside the restaurant walls. The result of this can be seen in Figure 20.

We can also confirm that the colouring of viral aerosol concentration in the view is correct by manually setting the concentration using the Command Centre to 0, 200, 400, …, 1000.

Figure 42. Patch colouring of different viral aerosol concentrations

Vocalisation

To test that the correct number of aerosols are being emitted from breathing and talking at the set volume, we can turn off coughing and sneezing, turn all ventilation off, manually infect one agent, and take note of the average concentration on the current tick (0.1857). The 'Step 1 tick' function can be used to go to the next step and note the average concentration again (0.1979). If we subtract 0.1875 from 0.1979, this results in 0.0104, which is the average aerosol concentration increase per patch.

We then calculate how much it should have gone up by. The restaurant volume was set to 50db. This will emit 36.315 viral aerosols into the air per tick, divided by the number of seating area patches 645 to give 0.0563 aerosols per patch, divide again by 5.4 to get the head height layer of air, giving 0.0104, which matches the value the viral aerosol actually increase by confirming that this is functioning as it should.

While testing this, a mistake was found in the viral aerosol concentration calculation and was fixed.

Coughing and sneezing

To test that the coughing and sneezing features are working, we can conduct an experiment that will simulate all possible combinations of coughing and sneezing on and off.

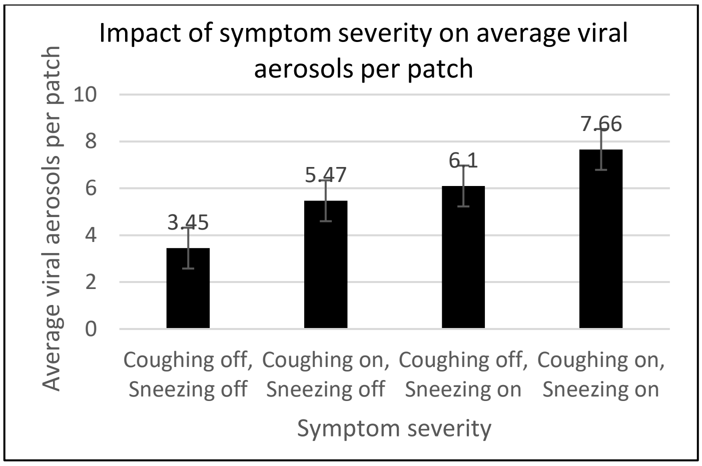

Figure 43. Bar graph showing the impact of coughing and sneezing on average viral aerosols per patch.

As shown by Figure 43, when coughing and sneezing are turned on, there is an increase in the restaurant's average number of aerosol particles. Interestingly, sneezing alone still produced more aerosols than coughing alone, even though a cough occurs far more frequently (in this experiment 1 cough per 5 mins vs 1 sneeze per 35 mins).

We can also vary the coughing and sneezing frequency to capture the severity of symptoms, impacting the number of viral aerosols. An example of increasing the coughing frequency is demonstrated below in Figure 44.

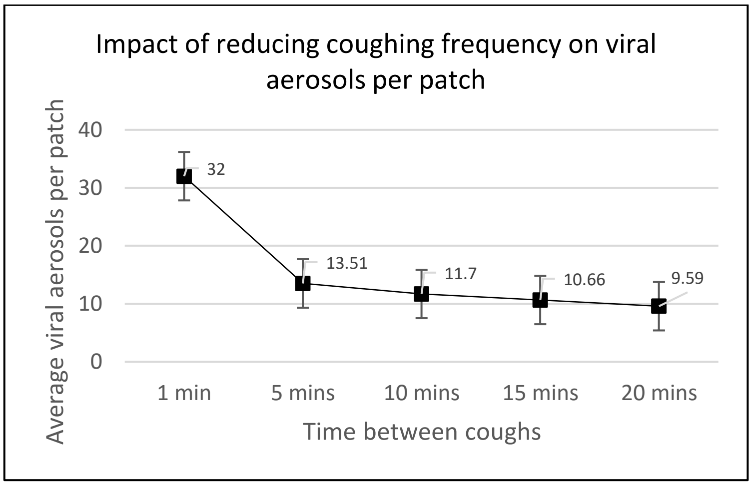

Figure 44. Bar graph showing the impact of reducing coughing frequency on the average viral aerosols per patch.

If we perform the same kind of average aerosol concentration increase calculations as we did for vocalisation, we can prove that the correct number of aerosol particles are output. The average concentration increase between ticks, before and after a cough, was 1.63 per patch. The expected increase was also 1.63, validating that the correct number of viral aerosols are being output.

5.1.4 Infection risk

5.1.4.1. Aerosols and droplets

To test that the inhalation of viral aerosols is working, we can run the simulation and observe that the average infection risk increases as the average viral aerosol concentration increases.

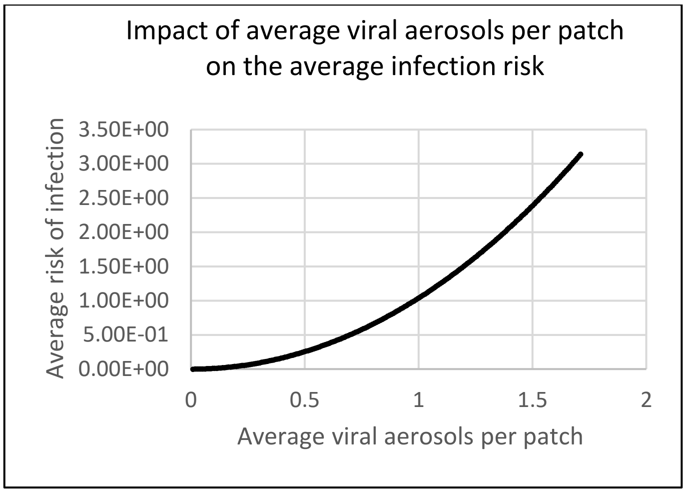

Figure 45. Graph showing the impact of average viral aerosols per patch on the average risk of infection over the first 250 ticks of simulation with one infected agent.

If we again use the step function to observe the increase in an agent's risk, we can compare this to the expected increase. The aerosol concentration of the patch the agent was on was 0.4719493. Over a single tick, the agent's infection risk increased by 0.0003808. Based on the patch's concentration and the amount of air an agent inhales per tick, it was calculated that the risk should have increased by 0.0003808, which is exactly what happened. This was using the calculation: (concentration of patch here / volume of patch layer) * ml of air inhaled.

5.1.4.2. Fomites

By inspecting table turtles and door patches, it can be seen that it becomes contaminated when an infectious person coughs or sneezes or contacts it. Customers hand contamination value can be seen increasing when they contact the surface, and their fomite risk increases when they touch their face.

5.1.4.3. Surface sanitisation

To ensure that the tables are being cleaned, we can observe that the colour of the table is being set back to green from red, and we can inspect the table and table node and observe that the 'fomiteInfected' Boolean is set back to false. We can inspect the door patches and make a similar observation to the tables.

5.1.5 Individual safety measures

5.1.5.1. Face masks

To confirm that the face mask feature is working correctly, a test can be run that shows the correlation between the average concentration of viral aerosols in the restaurant and the mask emission efficiency. As is shown by the line graph below, the viral aerosol emissions are extremely low when mask efficiency is 100% and unchanged when it is 0%.

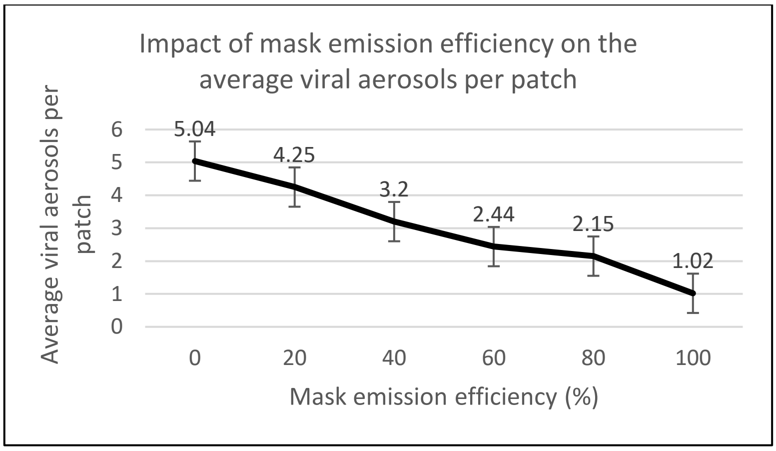

Figure 46. Graph showing the impact of mask emission efficiency on the average viral aerosol concentration per patch.

Similar can be shown for inhalation efficiency as when the mask inhalation efficiency is increased, the average infection risk decreases. This graph is shown in section 5.2.

5.1.5.2. Hand washing

Hand washing can be observed by inspecting a customer and watching their hand contamination value. Periodically it will be reduced by the set handwashing thoroughness.

5.1.6 Ventilation

5.1.6.1. Mechanical ventilation

The air change rate can be validated by infecting a random customer, allowing them to emit the virus for a short time, then using the Command Centre to 'un-infect' them by setting their infected Boolean to false. This will stop them from emitting virus particles and allow us to pause the simulation and inspect a random patches virus concentration. We can see below the viral aerosol concentration of the selected patch:

With an air change rate of 12 ACH, we can manually calculate using the equation above, that on the next tick, this value should be reduced by 0.003%, which is exactly what happens.

```
Reduced concentration = 0.05886233566934053 - ( 0.05886233566934053
* ((100 / (3600 / 12)) / 100)

Reduced concentration = 0.058666127882141
```

Figure 47. Example of reduced concentration calculation from mechanical ventilation

This can also be validated by looking at the line graph showing the relationship between air change rate and the average concentration of viral aerosols.

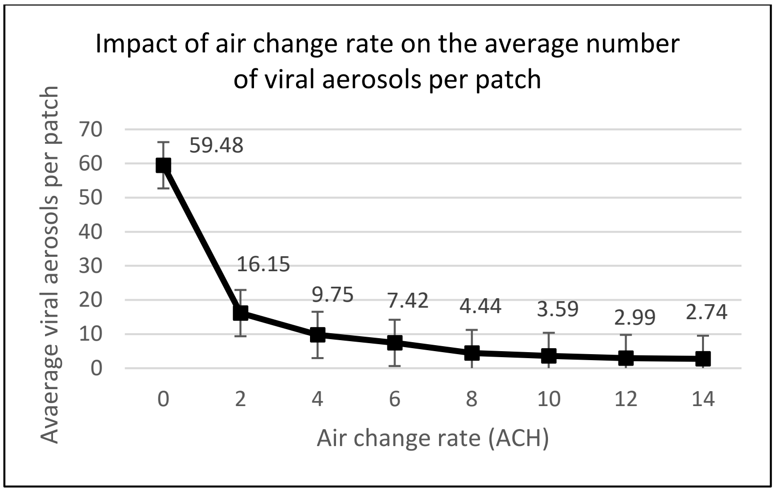

Figure 48. Graph showing the impact of air change rate on the average number of viral aerosols per patch.

Portable HEPA filters

We can test the impact of changing the number of HEPA filters on viral aerosol concentration. This will have the same effect as the air conditioning running 1, 2, 3, and 4 air changes per hour.

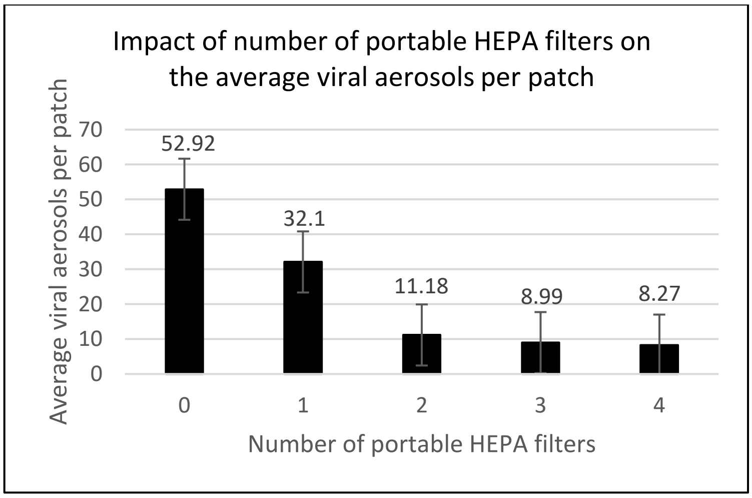

Figure 49. Graph showing the impact of increasing the number of portable HEPA filters in the restaurant on the average viral aerosols per patch.

5.1.6.2. Natural Ventilation

To test that each window configuration results in the correct air change rate, we can compare the average viral aerosol concentration when the air conditioning is on 0 ACH and the windows are open, to concentration level when the air conditioning is set to 5 ACH, which has been tested and validated to work as it should. The average viral aerosol concentration across multiple runs should be the same, which was confirmed to be the case, as shown by the table below.

	Front and side windows open	**AC on 5 ACH**
Average viral aerosols per patch	11.42792	11.7502

Figure 50. Table comparing the average viral aerosol concentration resulting from having all windows open vs AC on 5 ACH.

5.2 Safety measure evaluation

All tests (unless otherwise specified) will use the same basic input variables as listed below. The tests will involve changing one parameter at a time to investigate the impact it has on the output data. The reason for keeping all input variables the same, apart from the ones being investigated, is to give a fair comparison between each one. These specific values were chosen to give a balance between realistic and interesting results that can be analysed.

Input variables	Values		Input variables	Values
manual-customer-spawn	false		left-windows-open	false
noOfWaiters	3		top-windows-open	false
capacity	50		air-refresh-rate	5
diningTime	60		recirculation	false
length-of-service	4		noOfPortableHEPA	0
twoM-distanced-seating	false		masks	false
chance-of-being-infected	1		mask-effectiveness-inhalation	0
volume	60		mask-effectiveness-emission	0
coughing	true		sanitiser-on-tables	false
cough-frequency	5		handwashing-thoroughness	0
sneezing	false		doors	false
sneeze-frequency	--			

Figure 51. Table of base test parameters

The percentage of probable infections has been chosen as the primary gauge of effectiveness as it is unaffected by the number of customers served. Again, tests will be ran for 40 repetitions.

Output graphs will include the standard error to show the '*approximate standard deviation of a statistical sample population*' [62], because testing will only be using a limited number of runs, it is not representative of the entire population.

5.2.1 Physically distanced seating

To verify the effectiveness of a physically distanced seating configuration, a test can be run measuring the percentage of probable infections for both the physically distanced and non-physically distanced configurations. We can compare the simulation results and estimate how effective this safety measure is at reducing virus transmission.

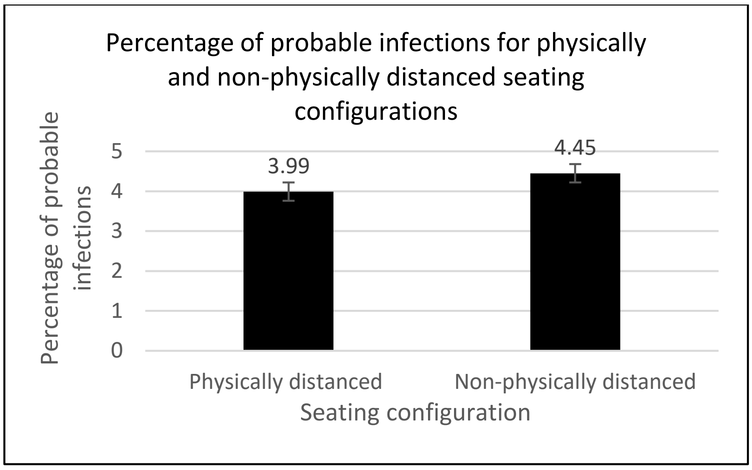

Figure 52. Bar graph comparing the percentage of probable infections for physically and non-physically distanced seating configurations.

Keeping all other parameters consistent across the runs, altering only the seating configuration between physically distanced and non-physically distanced, the percentage of customers likely to develop the disease dropped by 0.46 (10.3%). This suggests that a physically distanced seating configuration is likely to reduce virus transmission effectively.

5.2.2 Reduced capacity

To test the effectiveness of reducing the restaurants capacity, we can observe the impact of reducing the restaurants capacity on the percentage of people infected. The test will use parameter sweeping to run through all capacity values available using the slider in the user interface.

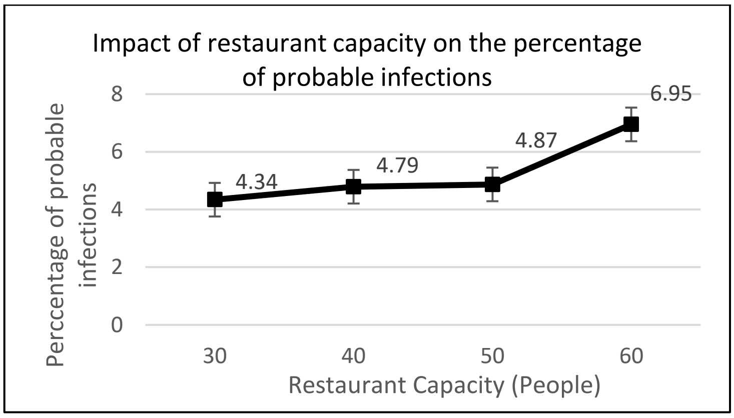

Figure 53. Graph showing the impact of restaurant capacity on the percentage of probable infections.

The simulation suggests that reducing the capacity by 10 people will reduce the percentage of probable infections by 0.88% on average.

Figure 53 shows how the percentage of infections reduces, but the total number of infections is also reduced as by reducing the capacity, there are fewer customers served and therefore less people exposed to the virus. To avoid losing customers, the service could perhaps be extended and booking time reduced slightly to counteract this.

These results are without any consideration for where in the restaurant the customer is sitting. A feature discussed in the 'Future work' section would be to seat the customer at the furthest possible table from any other occupied table, reducing the probable infections even further.

5.2.3 Reduced exposure time

If the length of service remains the same but the booking time slots are reduced, assuming available slots can be filled, this will result in more customers entering the restaurant. Therefore, the test will track the percentage of probable infections as a result of different booking lengths.

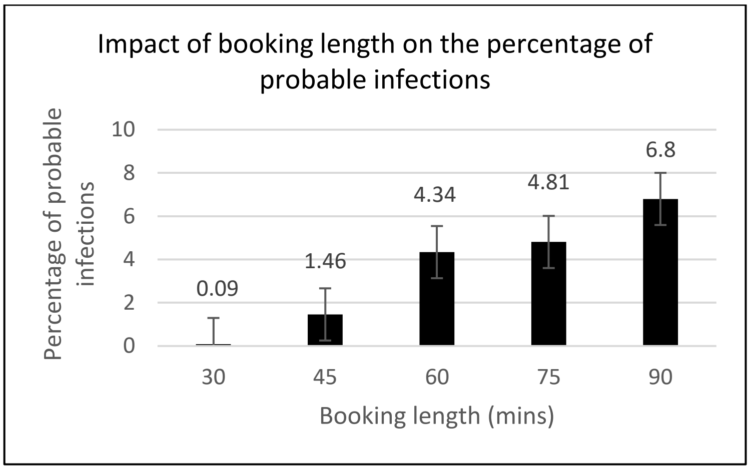

Figure 54. Graph showing the percentage of probable infections for different booking lengths (mins).

The graph in Figure 54 shows that reducing the length of a booking by 15 minutes reduces the percentage of infections by 1.68% on average. The simulation suggests that the greatest reduction in cases occurs by reducing the booking from 60 mins to 45 mins for the current parameters. Between these time slots is when most people have been exposed long enough to have inhaled an infectious dose of the virus.

With a booking slot of 60 minutes, we can test the effect of reducing service time. The results can be seen in Figure 55 below.

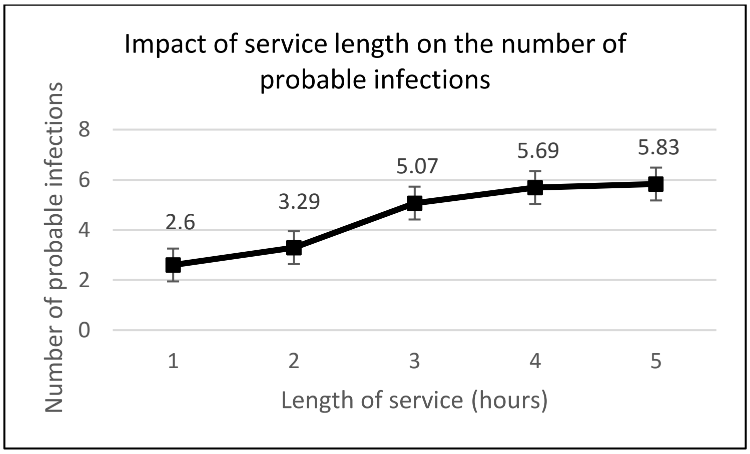

Figure 55. Graph showing the number of probable infections for different service lengths (hours).

The graph suggests that by shortening the service length, fewer probable infections will be produced as the restaurant is serving fewer customers, so less people are potentially exposed to the virus. In this case, reducing the service by 1 hour resulted in 0.81 fewer infections on average.

If both booking time and service length are reduced, this should result in an even larger reduction in cases. After running a test where booking time is reduced to 45 mins, from 60 mins, and service length is reduced to 3 hours, from 4 hours, it was shown that this would result in 0.48% of people infected on average, during a service of roughly 60 people; 3.86% fewer cases than the standard set up.

5.2.4 Volume

To investigate the impact of restaurant volume on the percentage of probable infections, we can simulate the restaurant being silent (50db), normal talking volume (60db), and loud talking volume (70db).

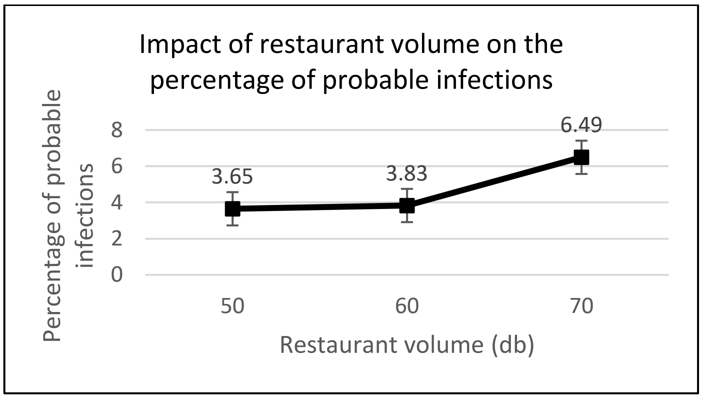

Figure 56. Line graph showing the impact of restaurant volume on the percentage of probable infections.

Figure 56 shows that the percentage of people infected in the restaurant increases with the volume. A 10db increase in volume will result in 1.42% more cases on average, highlighting the importance of lowering the restaurant volume by turning off the music and muting TVs.

The most significant increase appears to occur from 60db/normal talking volume to 70db/loud talking volume. The difference in probable infections produced between breathing and talking normally is minimal.

5.2.5 Face Masks

The test below shows the benefits of mask-wearing in the restaurant. It measures the percentage of probable infections if the average mask efficiency is 0%, 50%, and 90% to represent no masks, a mix/average of mask efficiencies, and N95 masks.

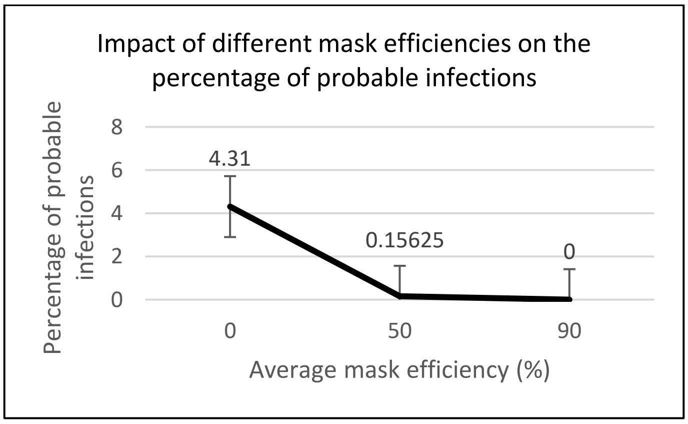

Figure 57. Graph showing the impact of different mask efficiencies on the percentage of probable infections.

As a result of an average mask efficiency of 50% the percentage of probable infections was reduced by 4.15%. N95 masks prevented all probable infections, but it is unrealistic to expect all customers and staff to wear these.

5.2.6 Hand washing

To test the importance of handwashing, we can run the simulation with handwashing thoroughness 0-100% in 25% increments and observe the fomite infection risk. As discussed earlier, new research suggests that the chance of infection for fomites is only 1 in 10,000 [10], but regardless, the results of this experiment can be seen below.

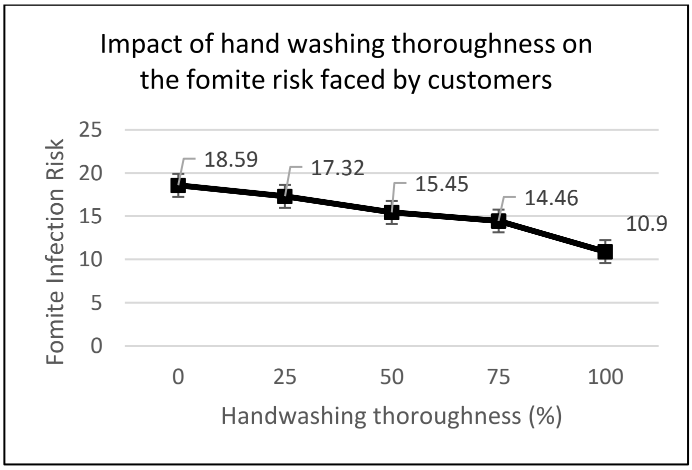

Figure 58. Graph showing the impact of handwashing thoroughness on customers fomite infection risk.

As suspected, the higher the handwashing thoroughness, the lower the fomite infection risk. Suppose customers were to achieve an average handwashing thoroughness of 75%. In that case, the simulation suggests that this could result in a 22.2% decrease in fomite risk, with no other behavioural changes such as less face and surface touching.

5.2.7 Increased mechanical ventilation

To investigate the effectiveness of increasing the air change rate coming from the air conditioning unit, an experiment can be conducted using the parameter sweeping technique to run the simulation with 0 to 14 ACH, in increments of 2 ACH. We can then observe the percentage of probable infections at each rate.

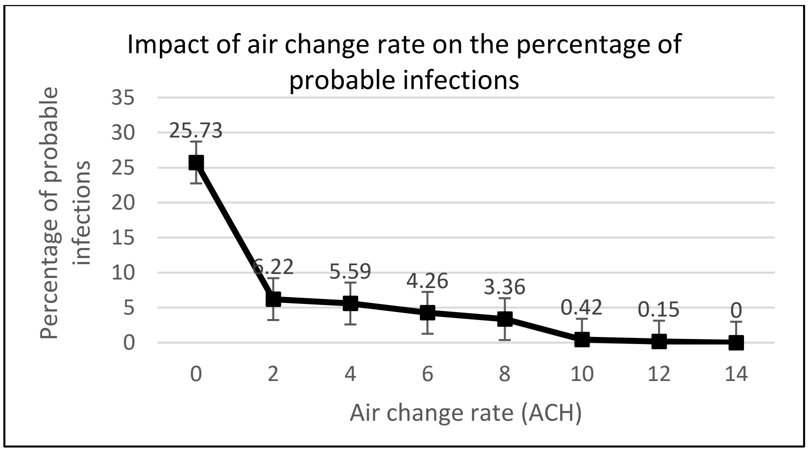

Figure 59. Graph showing the impact of air change rate on the percentage of probable infections produced.

As shown in the graph above, the most significant reduction in cases occurs at lower air change rates, and the impact lessens at the air change rate is increased. The average decrease in probable infections is 3.7%. With the recommended air change rate of 12 ACH, the percentage of infections is 0.15%. This highlights the importance of proper ventilation in the restaurant.

Portable HEPA Filters

The more portable HEPA filters placed in the restaurant, the greater the reduction in the number of viral aerosols per patch. There appears to be a fall-off of effectiveness when the number of filters exceeds 2, similar to what can be seen with the regular air conditioning as they essentially have the same effect.

5.2.8 Increased natural ventilation

	Front and side windows open	**AC on 5 ACH**
Average viral aerosol per patch	11.42792	11.7502
Percentage of probable infections	1.148037	3.772824

Figure 60. Table comparing the average viral aerosol concentration resulting from having all windows open vs AC on 5 ACH.

This test presents some interesting results. It suggests that even though the average viral aerosol concentration per patch is roughly the same, with the same air change rate, the percentage of probable infections produced with AC off and windows open (1.15%) is noticeably lower than the percentage produced with the AC on 5 ACH (3.77%). This is likely due to the lack of air currents produced by the AC as viral aerosols may remain in the area around the infectious person in higher concentration, resulting in high virus exposure to the surrounding people. With the windows open and AC off, the strong air currents from the windows will blow viral aerosols into areas where no customers are sitting. It will be spread more evenly throughout the restaurant rather than having areas with high concentration and low concentration. By looking at Figure 36, it can be seen that the air currents will carry air away from the seating area and over to the bar, where there will probably be fewer people. The people sitting on the side of the room with the intake windows will be at far less risk of infection as they are next to a continuous supply of outdoor air.

5.3 Conclusion

Based on the experiments conducted, it is now possible to rank the safety measures from most to least effective. Under these test parameters, the average percentage infection rate is roughly 4.5%. Safety measures will be ranked based on how much they reduced this value.

Rank	**Safety measure**	**Reduction of probable infections (%)**
1	Face masks (50% efficiency)	4.43%
2	Proper ventilation (12ACH)	4.35%
3	Reduced exposure time (booking length 60 to 45 mins, service length 4 to 3 hours)	4.02%
4	Reduced volume (70db to 60db)	2.66%
5	Reduced capacity (60 to 50 people)	2.08%
6	Open windows only (5ACH)	1.15%
7	Physical distancing (2m)	0.46%

Figure 61. Table ranking the how much each safety measure reduced the percentage of probable infections

The simulation suggests wearing a face mask is the most effective safety measure. The least is physical distancing, likely due to the airborne nature of the virus.

6 User Testing

The user interface was tested on five users unfamiliar with NetLogo and the topic of COVID-19 transmission. A group of 5 was chosen as '*the best results come from testing no more than 5 users*' [63]. They were provided with a test script which set them a series of tasks to complete. The test script can be seen in appendix 1 of this paper. Briefly, the tasks were to set up the restaurant environment with the given parameters, run the simulation, carry out some actions and observations, and note some of the results.

After having the first participant complete the test script, it was clear some things needed to improve to increase user-friendliness. The first comment from participant 1 suggested that the simulation controls could be labelled better and have descriptions where necessary and that the controls could be reorganised into more meaningful categories. To help solve this issue, I ordered parameters in the order they should be set up, numbering each heading and giving more descriptive heading names. All outputs were gathered into their own section. I then provided a 'read me' section containing instructions on how to set up and run the simulation.

The second comment was, '*I couldn't clearly see the people sitting at the tables. I had to try and chase someone who kept moving'.* This comment refers to the part of the test script that has the user click on an agent and infect them with the virus. The participant was trying to infect a waiter that would keep moving while they were trying to infect them. They also did not notice the stationary customer agents sitting at the tables. To help future users complete this action, some information was provided in the instructions, suggesting that the user first slows down the simulation slightly using the speed slider in the NetLogo interface and then clicking on the agent.

Regarding them not seeing the customers, previously the chairs were coloured cyan, and customers were light blue. This could have caused the participant not to notice the customer agents, so the colour of chairs was set to a lighter colour yellow, and customers set to a darker shade of blue to improve contrast. I created a legend describing what each agent/patch represents, which should also help users to identify the different parts of the simulation.

The final comment was similar to the first, stating that graphs could be labelled to better describe what they are showing and that the graphs themselves could be made larger. To help solve this problem I made the graphs bigger and gave each one a brief description of what it was showing.

The rest of the participants were asked to complete the user testing script on the improved user interface, where feedback was a lot more positive with some minor changes being made to the UI. The largest change was the addition of a section describing what each component on the user interface is responsible for. A minor change that was suggested was to have mask efficiencies automatically set by buttons. Rather than have the user set the inhalation and emission efficiencies using the slider, they could press a 'Surgical mask' button, and it would set the efficiencies for that mask type. To conclude and relate this to the objective, '3. Design a simple user interface', I would say that, in the end, this was achieved.

7 Example study

Each safety measure has been validated and evaluated independently, demonstrating how each can individually reduce virus transmission. A study can now be conducted using a combination of safety measures to show how effective they can be if used simultaneously. This will replicate a restaurant that is following all the relevant COVID-19 guidance.

7.1 Parameters

On the date of Scotland's highest 7-day positive rate per 100,000 population, 04/01/2021, the chance of a person being infectious was 0.3%. The simulation can estimate infection data if the restaurant was open on this day with and without safety measures implemented. The following input variables were used:

Input variables	**Value (unsafe)**	**Values (safe)**	**Reason for value**
manual-customer-spawn	False	false	Automatic spawn required for test.
noOfWaiters	3	3	No. of waiters needed to keep table states on schedule.
capacity	60	50	Recommended seating capacity. [51] Shown effective in Figure 53.
diningTime	60	45	15 min reduction shown effective in Figure 54.
length-of-service	4	3	As length of booking is shortened, the same no. of customers can be served in less time.
twoM-distanced-seating	false	true	2m distance recommended [33]. Shown effective in Figure 52.
chance-of-being-infected	0.3	0.3	Chance of a person being infectious at peak of pandemic [36].
volume	70	60	Reduction of 10db/loud talking over music, to normal talking [39]. Shown effective in Figure 56.
coughing	true	true	Symptoms must be kept consistent for fair comparison. A persistent cough is a symptom of COVID-19[43].
cough-frequency	5	5	Symptoms must be kept the same across runs [43].
sneezing	false	false	Symptoms must be kept consistent for fair comparison. Sneezing is not a main symptom of COVID-19 [64].
sneeze-frequency	-	-	
left-windows-open	false	true	Opening windows will improve ventilation, shown in Figure 50.

top-windows-open	false	true	Opening windows will improve ventilation, shown in Figure 50. All windows being open adds 5 ACH
air-refresh-rate	7	7	Air change rate of 12 recommended [54]. Shown effective in Figure 48. Value of 7 estimated as there is no way of telling what the air exchange was initially. 7 ACH from AC + 5 ACH from windows = 12 ACH
recirculation	false	false	The Kailyard does not have a recirculating air conditioner.
noOfPortableHEPA	0	0	Only required in extreme cases or poor ventilation.
masks	false	true	Shown effective in Figure 57.
mask-effectiveness-inhalation	-	50	Estimated average mask inhalation efficiency.
mask-effectiveness-emission	-	50	Estimated average mask emission efficiency.
sanitiser-on-tables	false	true	Hand washing will reduce hand contamination and fomite risk
handwashing-thoroughness	-	70	Estimated based on hand washing study. [50]
doors-open	false	true	Will reduce fomite transfer

Figure 62. Table of safe vs unsafe restaurant test parameters

7.2 Results

Over 100 repetitions of the experiment of both the 'safe' and 'unsafe' restaurant configurations, the following results were produced.

	Unsafe	**Safe**	**Difference**
Average probable infections	1.83	0	1.83
Percentage of probable infections	1.06	0%	1.06
Average infection risk	19.78157	3.003895	16.777675 (6.9x safer)
Average aerosol concentration	0.33	0.0132	0.3168

Figure 63. Table of test results from safe vs unsafe restaurant.

The numbers in Figure 63 include results from runs where there were no infections, therefore reflecting result similar to real life.

The simulation suggests that a restaurant implementing the proper safety measures resulted in 0 infections, whereas if the restaurant had not been, there would have been 1.83 probable infections on average, 1.06% of customer. With safety measures, the average infection risk was 6.9 times lower than without, and the aerosol concentration was significantly decreased.

There is no way of verifying these results as there have been no similar studies conducted on The Kailyard restaurant. A reasonable estimation would suggest that the data produced seems plausible.

8 Conclusion

8.1 Summary

This project's objectives were achieved successfully by creating a simulation that captures how COVID-19 is transmitted from person to person in a restaurant setting. This includes transmission through viral aerosol particles, respiratory droplets, and fomites. The simulation also allows for the implementation of COVID-19 safety measures such as physically distanced tables and face masks. The simulation can then output data which can be analysed to gauge how effective a safety measure was based on how significantly it reduced probable infections and infection risk. A combination of safety measure can also be tested to create the greatest reduction in infection risk.

The specific objectives for this project were:

1. To expand my knowledge of the problem and NetLogo. This was achieved through extensive research into the transmission of COVID-19 and the effectiveness of safety measures. The creation of many simulation prototypes grew my knowledge of NetLogo to a large degree.

2. Design a well-structured and expandable program. This was achieved as the code is structured to make it easily expandable, which has been proven throughout development as features were added incrementally.

3. Design a simple user interface. This was achieved through careful design and user testing. The feedback gathered from user testing was implemented into the user interface.

4. Create a 2D multi-agent simulation of a restaurant capturing:

- o Customer and staff movements and actions. Achieved using a best-first search path finding algorithms to navigate a network of nodes to specific locations controlled by an action schedule.
- o Virus emission and contraction through respiratory droplets, aerosols, and fomites. Achieved by using data gathered from relevant literature to provide an estimate of virus emission and intake of viral aerosols, larger droplets, and fomites.
- o Ventilation:
  - ▪ Natural ventilation (windows). Achieved by modelling the air change rate provided and the air paths through the restaurant when different combinations of windows are opened or closed, simulating how viral aerosols can be carried in air currents.
  - ▪ Mechanical ventilation (air conditioning): Achieved by implementing the ability to set the air change rate provided by the air conditioning, which will impact the concentration of viral aerosols. Also, the option to place portable HEPA filters around the restaurant to further reduce aerosols.
  - ▪ Air paths. Mostly achieved as this feature was included for natural ventilation only for reasons discussed in this report, but it was successfully implemented for all natural ventilation configurations.
- o A range of safety measures. Achieved through the implementation of face masks, hand washing, physically distanced seating, capacity, booking length

and service length limits, reduced restaurant volume, and increased ventilation.

5. An evaluation of safety measures based upon data produced. Achieved by gathering data using the NetLogo behaviour space to run the simulation under specific parameters. This data can be analysed in excel to investigate its effectiveness at reducing infection risk and probable infections.

6. An optimised restaurant configuration to reduce COVID-19 transmission among customers and staff. Achieved by setting up the restaurant simulation with a combination of safety measures such as those listed above and estimating the infection risk if these measures are implemented.

8.2 Evaluation

As the project reached all goals that were initially set, I would consider the project a success. A simulation of a restaurant was created, which captured the restaurant environment, the customer and staff movement, transmission of COVID-19, ventilation, and a multitude of safety measures.

The system will not be entirely accurate to the real world, but it can estimate infection risk and other related outputs. This was expected as such a system would be far beyond this project's scope, but it is a step in the right direction to research on the topic. Over the course of the project, spending countless hours researching the COVID-19 disease, I have learned a lot about it and how it spreads and the ways of mitigating its impact on people in enclosed spaces.

I have also learned about different computer modelling techniques, with most of that learning being centred around the agent-based modelling software NetLogo. The project's motivation was to expand my general programming knowledge into the computer simulation space. This project gave me the means to do just that by creating the restaurant simulation and researching similar simulation tools and virus transmission models.

In terms of any limitations of the results produced by the simulation, I would say there does not seem to be any obvious limitations at the current stage, but as there has been no similar research at The Kailyard, there is no way of verifying the results besides an intelligent estimation of what is expected from such a simulation. There are some implementation aspects I would approach differently if I were to do the project again. The first is to make the simulation run faster, as in some rare cases when there are many infectious individuals in the restaurant, execution slows down. Given more time, I could perhaps figure out a more effective way of dispersing aerosols among patches, but as far as I can see, the current implementation would be the only way to do this. Another improvement I would like to make is to tidy up the network of nodes as links appear to be all over the place. While there is reasoning behind the current node configuration, I think the number of links could be reduced slightly. This would also perhaps speed up the simulation somewhat as mobile agents would have fewer path options to consider when navigating the network.

There is one known bug in the program, but it occurs very rarely. As discussed in the validation testing section, the first is when customers are meant to leave the restaurant when it closes. However, occasionally, under a specific set of parameters, a customer will remain seated at the table. Thankfully, this error does not affect the results. NetLogo does not have a debugging feature, and the error messages are often slightly vague. The bug happens so rarely, and that NetLogo makes it tricky to track down, meaning that finding the root cause and fixing the errors has proven to be tricky.

8.3 Future Work

There are a plethora of features that I would have liked to add if given more time. The most impactful features were implemented, giving a good base for any further work to be completed.

In order of what I would deem most important to least important, the first improvement I would make would be to allow other restaurants to be either loaded in using a file or ideally allow the user to click and place tables and draw the walls but use predefined ventilation and movement mechanics. For this implementation, the pathfinding would have to be done using patches rather than nodes, perhaps using the A* algorithm. This would allow the tool to be utilised by the public as they would not require any coding knowledge to use it.

The second feature I would have liked to add is further physical distancing. This would make tables sit as far away as possible from other tables, as this would likely reduce probable infections even further.

The simulation could have a way of evaluating which seats are most dangerous. This would include running the simulation for many repetitions and assigning each table a risk score based on how often the customers at that table become infected. For example, the seats that are directly next to intake windows are likely the safest as they have a constant supply of fresh air. However, the seats at the exhaust windows could be less safe as all the potentially contaminated air is being blown past them.

An important factor for natural ventilation would be to set the wind direction and speed for a given day as this has the potential to impact the air change rate from the windows highly. Some days could be very still and others with high winds; one day, the wind may be blowing into the windows, another day away from them.

To list some additional features, we could implement a one-way system, protective screens, use of washroom facilities, save an order of customers for re-testing, portable fans for accelerating air currents, virus decay from temperature and humidity over long periods, variable time between booking arrival slots with a queuing system, reduced waiter table contact, and finally, more realistic dissipation of the virus from breathing, coughing, and sneezing caused by draughts of people walking around the restaurant. The way that the code is structured would simplify the implementation of these features.

References

[1] WHO, “WHO Coronavirus Disease (COVID-19) Dashboard | WHO Coronavirus Disease (COVID-19) Dashboard,” *Who.int*, 202AD. https://covid19.who.int/%0Ahttps://covid19.who.int/%0Ahttps://covid19.who.int/?gclid=CjwKCAjwnK36BRBVEiwAsMT8WJ3y00_BUzvrLsvbl3uthuoTH_Occ45gyEUbpYRyEqAzll3aZB6TYxoCcM0QAvD_BwE (accessed Apr. 01, 2021).

[2] Simon Read, “FTSE 100 suffers worst year since financial crisis,” *BBC News Business* , 2020. https://www.bbc.co.uk/news/business-55500103 (accessed Apr. 01, 2021).

[3] Office for National Statistics, “The impact of the coronavirus so far: the industries that struggled or recovered - Office for National Statistics,” *Office for National Statistics*, 2020. https://www.ons.gov.uk/economy/economicoutputandproductivity/output/articles/theimpactofthecoronavirussofartheindustriesthatstruggledorrecovered/2020-12-09 (accessed Apr. 01, 2021).

[4] SCOT GOV, “Coronavirus (COVID-19) update: First Minister’s speech 14 August 2020,” *Scottish Government*, 2020. https://www.gov.scot/publications/coronavirus-covid-19-update-first-ministers-speech-14-august-2020/ (accessed Apr. 01, 2021).

[5] Corey Bedford, “‘We can’t carry on like this’ - coffee shop owner warns hospitality industry won’t survive much longer,” *Grimsby Telegraph*, 2021. https://www.grimsbytelegraph.co.uk/news/grimsby-news/we-cant-carry-like-this-4848012 (accessed Apr. 01, 2021).

[6] D. Gursoy and C. G. Chi, “Effects of COVID-19 pandemic on hospitality industry: review of the current situations and a research agenda,” *J. Hosp. Mark. Manag.*, vol. 29, no. 5, Jul. 2020, doi: 10.1080/19368623.2020.1788231.

[7] A. Thamboo *et al.*, “Clinical evidence based review and recommendations of aerosol generating medical procedures in otolaryngology – head and neck surgery during the COVID-19 pandemic,” *J. Otolaryngol. - Head Neck Surg.*, vol. 49, no. 1, Dec. 2020, doi: 10.1186/s40463-020-00425-6.

[8] N. van Doremalen *et al.*, “Aerosol and Surface Stability of SARS-CoV-2 as Compared with SARS-CoV-1,” *N. Engl. J. Med.*, vol. 382, no. 16, Apr. 2020, doi: 10.1056/NEJMc2004973.

[9] CDC, “Science Brief: SARS-CoV-2 and Potential Airborne Transmission | CDC,” *Centres for Disease Control and Prevention*, 2020. https://www.cdc.gov/coronavirus/2019-ncov/science/science-briefs/scientific-brief-sars-cov-2.html (accessed Apr. 06, 2021).

[10] CDC, “Science Brief: SARS-CoV-2 and Surface (Fomite) Transmission for Indoor Community Environments | CDC,” *Centres for Disease Control and Prevention*, 2021. https://www.cdc.gov/coronavirus/2019-ncov/more/science-and-research/surface-transmission.html (accessed Apr. 08, 2021).

[11] E. Bonabeau, “Agent-based modeling: Methods and techniques for simulating human systems,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 99, no. 10, pp. 7280–7287, 2002, Accessed: Apr. 01, 2021. [Online]. Available: www.pnas.orgcgidoi10.1073pnas.082080899.

[12] G. Houzeaux *et al.*, “High-Performance Computing: Dos and Don’ts,” in *Computational Fluid Dynamics - Basic Instruments and Applications in Science*, InTech, 2018.

[13] Y. Li *et al.*, “Probable airborne transmission of SARS-CoV-2 in a poorly ventilated restaurant,” *Build. Environ.*, vol. 196, Jun. 2021, doi: 10.1016/j.buildenv.2021.107788.

[14] SAGE, “EMG: Simple summary of ventilation actions to mitigate the risk of COVID-19, 1 October 2020,” 2020. Accessed: Apr. 01, 2021. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/945754/S0973_Ventilation_Actions_Summary_16122020_V2.pdf.

[15] J. L. Jimenez and et al., “COVID-19 Airborne Transmission Tool Available,” *Cires*, 2020. https://docs.google.com/spreadsheets/d/16K1OQkLD4BjgBdO8ePj6ytf-RpPMlJ6aXFg3PrIQBbQ/edit#gid=519189277 (accessed Apr. 01, 2021).

[16] Javier Salas, H. TRADUTORGalloway, and L. VIDEOSAlmodóvar, “Aerosol transmission of Covid-19: A room, a bar and a classroom: how the coronavirus is spread through the air,” *El País - English Edition*, Oct. 2020. https://english.elpais.com/society/2020-10-28/a-room-a-bar-and-a-class-how-the-coronavirus-is-spread-through-the-air.html (accessed Apr. 01, 2021).

[17] M. Wei-Hass and K. Elliott, “Measure the risk of airborne COVID-19 in your office, classroom, or bus ride,” *National Geographic*, 2020. https://www.nationalgeographic.com/science/article/how-to-measure-risk-airborne-coronavirus-your-office-classroom-bus-ride-cvd (accessed Apr. 01, 2021).

[18] H. Stevens, “Why outbreaks like coronavirus spread exponentially, and how to ‘flatten the curve’ -,” *Washington Post*, 2020. https://www.washingtonpost.com/graphics/2020/world/corona-simulator/ (accessed Apr. 01, 2021).

[19] S. Maharaj and A. Kleczkowski, “Controlling epidemic spread by social distancing: Do it well or not at all,” *BMC Public Health*, vol. 12, no. 1, Dec. 2012, doi: 10.1186/1471-2458-12-679.

[20] N. Swift, “Easy A* (star) Pathfinding. Today we’ll being going over the A*… | Medium,” *Medium*, 2017. https://medium.com/@nicholas.w.swift/easy-a-star-pathfinding-7e6689c7f7b2 (accessed Apr. 01, 2021).

[21] A. Arooj, M. S. Farooq, T. Umer, G. Rasool, and B. Wang, “Cyber Physical and Social Networks in IoV (CPSN-IoV): A Multimodal Architecture in Edge-Based Networks for Optimal Route Selection Using 5G Technologies,” *IEEE Access*, vol. 8, pp. 33609–33630, 2020, doi: 10.1109/ACCESS.2020.2973461.

[22] Qiao, “PathFinding.js,” *Github*. https://qiao.github.io/PathFinding.js/visual/ (accessed Apr. 01, 2021).

[23] I. Lee, “Agent Based Modelling,” *SlidePlayer*. Dec. 2012.

[24] “NetLogo 6.2.0 User Manual,” *Northwestern University Center for Connected Learning and Computer-Based Modeling*, 2020. https://ccl.northwestern.edu/netlogo/docs/ (accessed Apr. 01, 2021).

[25] R. C. M. Cadiz, “OSC-NETLOGO: A TOOL FOR EXPLORING THE SONIFICATION OF COMPLEX SYSTEMS USING NETLOGO,” *Int. Comput. Music Conf. Proc.*, vol. 2012, 2012, Accessed: Apr. 01, 2021. [Online]. Available: http://hdl.handle.net/2027/spo.bbp2372.2012.069.

[26] D. A. Robertson, “Book and Resource ReviewsAgent-Based Modeling Toolkits NetLogo, RePast, and Swarm,” *Acad. Manag. Learn. Educ.*, vol. 4, no. 4, Dec. 2005, doi: 10.5465/amle.2005.19086798.

[27] M. Salgado and N. Gilbert, “Agent Based Modelling,” in *Handbook of Quantitative Methods for Educational Research*, Rotterdam: SensePublishers, 2013.

[28] S. F. Railsback, S. L. Lytinen, and S. K. Jackson, "Agent-based Simulation Platforms: Review and Development Recommendations," *Simulation*, vol. 82, no. 9, Sep. 2006, doi: 10.1177/0037549706073695.

[29] "R: What is R?," *R Project*. https://www.r-project.org/about.html (accessed Apr. 01, 2021).

[30] B. Farnsworth, "iMotions," *iMotions*, 2019. https://imotions.com/blog/statistical-tools/ (accessed Apr. 01, 2021).

[31] B. Vila and A. Lefton, "Here's the Standard Ceiling Height for Every Type of Ceiling - Bob Vila," *Bob Vila*. https://www.bobvila.com/articles/standard-ceiling-height/ (accessed Apr. 01, 2021).

[32] "What are the Dimensions of a Disabled Toilet Room? | Disabled Toilets | Commercial Washrooms," *Commercial Washrooms*, Apr. 2018. https://www.commercialwashroomsltd.co.uk/blog/what-are-the-dimensions-of-a-disabled-toilet-room.html (accessed Apr. 01, 2021).

[33] "Transmission of SARS-CoV-2 and Mitigating Measures EMG-SAGE 4 th," 2020.

[34] J. Atkinson, Y. Chartier, C. L. Pessoa-Silva, P. Jensen, Y. Li, and W.-H. Seto, "Respiratory droplets," *Nat. Vent. Infect. Control Heal. Settings.*, 2009, Accessed: Apr. 01, 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK143281/.

[35] Otter J, "Surface contamination and respiratory viruses with pandemic potential (SARS, MERS and influenza): an underestimated reservoir? | Reflections on Infection Prevention and Control," *Reflections on infection prevention and control*, 2015. https://reflectionsipc.com/2015/12/16/surface-contamination-and-respiratory-viruses-with-pandemic-potential-sars-mers-and-influenza-an-underestimated-reservoir/ (accessed Apr. 03, 2021).

[36] Public Health Scotland, "COVID-19 Daily Dashboard - PHS COVID-19 | Tableau Public," *Tableau public*, Apr. 2021. https://public.tableau.com/profile/phs.covid.19#!/vizhome/COVID-19DailyDashboard_15960160643010/Overview (accessed Apr. 04, 2021).

[37] C. Ang, "Zooming In: Visualizing the Relative Size of Particles," 2020. https://www.visualcapitalist.com/visualizing-relative-size-of-particles/ (accessed Apr. 03, 2021).

[38] HM Government, "Keeping workers and customers safe during COVID-19 in restaurants, pubs, bars and takeaway services," Nov. 2020. Accessed: Apr. 03, 2021. [Online]. Available: https://assets.publishing.service.gov.uk/media/5eb96e8e86650c278b077616/working-safely-during-covid-19-restaurants-pubs-takeaways-100920.pdf.

[39] M. Alsved *et al.*, "Exhaled respiratory particles during singing and talking," *Aerosol Science and Technology*, vol. 54, no. 11. Bellwether Publishing, Ltd., pp. 1245–1248, Sep. 17, 2020, doi: 10.1080/02786826.2020.1812502.

[40] W. G. Lindsley *et al.*, "Viable influenza A virus in airborne particles expelled during coughs versus exhalations," *Influenza Other Respi. Viruses*, vol. 10, no. 5, Sep. 2016, doi: 10.1111/irv.12390.

[41] P. Badia *et al.*, "Quality Improvement Initiative to Reduce Nighttime Noise in a Transplantation and Cellular Therapy Unit," *Biol. Blood Marrow Transplant.*, vol. 25, no. 9, pp. 1844–1850, Sep. 2019, doi: 10.1016/j.bbmt.2019.05.001.

[42] M. Jayaweera, H. Perera, B. Gunawardana, and J. Manatunge, "Transmission of COVID-19 virus by droplets and aerosols: A critical review on the unresolved dichotomy," *Environmental Research*, vol. 188. Academic Press Inc., p. 109819, Sep. 01, 2020, doi: 10.1016/j.envres.2020.109819.

[43] NHS, "Symptoms of Coronavirus (COVID-19)," *NHS*, Apr. 2020. https://www.nhs.uk/conditions/coronavirus-covid-19/symptoms/ (accessed Apr. 13, 2021).

[44] M. Van Beusekom, "Yet more data support COVID-19 aerosol transmission," *CIDRAP News*, 2020. https://www.cidrap.umn.edu/news-perspective/2020/08/yet-more-data-support-covid-19-aerosol-transmission (accessed Apr. 03, 2021).

[45] R. Dhand and J. Li, "Coughs and Sneezes: Their Role in Transmission of Respiratory Viral Infections, including SARS-CoV-2," *American Journal of Respiratory and Critical Care Medicine*, vol. 202, no. 5. American Thoracic Society, pp. 651–659, Sep. 01, 2020, doi: 10.1164/rccm.202004-1263PP.

[46] S. Basu, "Computational characterization of inhaled droplet transport to the nasopharynx," *Sci. Rep.*, vol. 11, no. 1, p. 6652, Dec. 2021, doi: 10.1038/s41598-021-85765-7.

[47] T. L. Lucas, R. Mustain, and R. E. Goldsby, "Frequency of face touching with and without a mask in pediatric hematology/oncology health care professionals," *Pediatr. Blood Cancer*, vol. 67, no. 9, Sep. 2020, doi: 10.1002/pbc.28593.

[48] J. L. Jimenez, "2020_COVID-19_Aerosol_Transmission_Estimator," *Https://Tinyurl.Com/Covid-Estimator*. 2020, [Online]. Available: https://tinyurl.com/covid-estimator.

[49] T. Dbouk and D. Drikakis, "On respiratory droplets and face masks," *Phys. Fluids*, vol. 32, no. 6, p. 63303, Jun. 2020, doi: 10.1063/5.0015044.

[50] J. M. Mumma *et al.*, "Variability in the Duration and Thoroughness of Hand Hygiene," *Clin. Infect. Dis.*, vol. 69, no. Supplement_3, pp. S221–S223, Sep. 2019, doi: 10.1093/cid/ciz612.

[51] D. O'Connor, "REPORT ON IMPACT OF OCCUPANCY NUMBERS ON LICENSED PREMISES AS A RESULT OF COVID-19 (UK Version)," May 2020.

[52] SCOT GOV, "Coronavirus (COVID-19): ventilation guidance - November 2020 - gov.scot," Scottish Government, Dec. 2020. Accessed: Apr. 09, 2021. [Online]. Available: https://www.gov.scot/publications/coronavirus-covid-19-ventilation-guidance---november-2020/pages/ventilation/.

[53] J. Atkinson, Y. Chartier, C. L. Pessoa-Silva, J. Paul, Y. Li, and W.-H. Seto, "Concepts and types of ventilation," *Nat. Vent. Infect. Control Heal. Settings. Geneva World Heal. Organ.*, pp. 7–15, 2009, Accessed: Apr. 09, 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK143277/.

[54] Axair, "Recommended Air Changes per Hour - Axair Fans," *AXAIR FANS*, 2020. https://www.axair-fans.co.uk/all-technical-information/recommended-air-changes-per-hour/ (accessed Apr. 09, 2021).

[55] D. A. Christopherson, W. C. Yao, M. Lu, R. Vijayakumar, and A. R. Sedaghat, "High-Efficiency Particulate Air Filters in the Era of COVID-19: Function and Efficacy," *Otolaryngology - Head and Neck Surgery (United States)*, vol. 163, no. 6. SAGE Publications Inc., pp. 1153–1155, Dec. 01, 2020, doi: 10.1177/0194599820941838.

[56] Just Fans, “Commercial Hepa Air Purifier | Just Fans Ltd,” *Just Fans*. https://www.justfans.co.uk/h14-true-hepa-commercial-air-purifier-p-3562.html (accessed Apr. 10, 2021).

[57] Window Master, “Air Change Rate Calculator | Estimate air change | WindowMaster,” *Window Master*. https://www.windowmaster.com/resources/calculators/air-change-rate-calculator/ (accessed Apr. 10, 2021).

[58] Weather Spark, “Average Weather in Dunblane, United Kingdom, Year Round - Weather Spark,” *Weather Spark*. https://weatherspark.com/y/38098/Average-Weather-in-Dunblane-United-Kingdom-Year-Round (accessed Apr. 09, 2021).

[59] timeanddate, “Weather in February 2021 in Dunblane, Scotland, United Kingdom,” *timeanddate.com*, Feb. 17, 2021. https://www.timeanddate.com/weather/@2650769/historic?month=2&year=2021 (accessed Apr. 09, 2021).

[60] DestiMap, “Dunblane Museum, Dunblane | DestiMap | Destinations On Map,” *DestiMap*. https://www.destimap.com/index.php?act=attraction&a=Dunblane-Museum%2C-Dunblane%2C-United-Kingdom (accessed Apr. 10, 2021).

[61] liz, “Natural Ventilation Tricks to Cool Off your Summer - Southern Athena,” *Southern Athena*, Jul. . https://southernathena.com/natural-ventilation-tricks-to-cool-off-your-summer/ (accessed Apr. 11, 2021).

[62] W. Kenton, “Standard Error Definition,” *Investopedia*, May 2020. https://www.investopedia.com/terms/s/standard-error.asp (accessed Apr. 14, 2021).

[63] J. Nielson, “Why You Only Need to Test with 5 Users,” *Nielsen Norman Group*, Mar. 2000. https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/ (accessed Apr. 12, 2021).

[64] Centers for Disease Control and Prevention, “Symptoms of Coronavirus (COVID-19),” *Cdc*, Apr. 2020. https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html (accessed Apr. 03, 2021).

Appendix 1

User testing questionnaire

1. Read the instructions before use.

2. Initialise the restaurant with the following parameters:
    a. 3 waiters.
    b. General safety measures of your choice.

    c. the chance of infection is 3%.
    d. the volume of the restaurant is 50db.
    e. coughing on or off with a frequency of your choice.
    f. sneezing on or off with frequency of your choice.

    g. masks on/off with efficiency of your choice.

    h. Hand sanitiser on tables with hand washing thoroughness at 80%.
    i. Set doors to be open.

    j. Natural ventilation of your choice.
    k. Air refresh rate of 5 ACH.
    l. Re circulation off.
    m. 0 HEPA filters.

3. Click set up to confirm the parameters.

4. Run the simulation.
    a. Infect a random person within the restaurant.

5. When the simulation finishes, find the following output:
    a. The number of probable infections:
    b. The percentage of probable infections produced:
    c. On the aerosol output graph, the rough average concentration of viral aerosol particles at 5000 second into the simulation:
    d. The total number of customers served:

Please respond to the following statements:

✓

1. Choosing the simulation parameters was simple:

1 (Strongly disagree)	2	3 (neither agree nor disagree)	4	5 (Strongly agree)

2. It was easy to understand what controls would do:

1 (Strongly disagree))	2	3 (neither agree nor disagree)	4	5 (Strongly agree)

3. It was easy to infect a random person in the restaurant:

1 (Strongly disagree)	2	3 (neither agree nor disagree)	4	5 (Strongly agree)

4. It was easy to understand what output graphs and monitors were showing:

1 (Strongly disagree)	2	3 (neither agree nor disagree)	4	5 (Strongly agree)

5. It was easy to identify which areas had high concentration of aerosol particles:

1 (Strongly disagree)	2	3 (neither agree nor disagree)	4	5 (Strongly agree)

6. If you answered disagree for any of the above, why?

7. Any other comments or improvements:

Appendix 2 – User guide / Installation Guide

Before running the simulation, it should be set up with the required parameters using the sliders and switches in the control sections. The parameters do not have to be set up in any particular order but to make the setup process easier to follow, the sections are numbered and ordered in the way that makes the most sense.

Setup begins with the 'General controls' in section 1. These are the general controls that do not fit into the other categories and more refers to the restaurant's general functionality. By setting the 'manual-customer-spawn' to true, automatic seating will be turned off, and the user will have to click on a table that they want to fill. If the switch is set to false, the simulation will handle the seating of customers automatically.

To set up the infection sub-model, we then proceed to section 2, the 'Infection controls'. The chance of being infectious can be set to the number of confirmed positive cases in the area currently or any other value relevant to the user's experiment. The restaurant volume can be set using the slider; 50db would be a silent restaurant, 60db is a normal talking volume restaurant, and 70db is a loud restaurant. Symptom severity can be adjusted by turning coughing and sneezing on or off using the switches, and the frequency can be set using the sliders.

The following section is to set up any personal protective measure which applies directly to the agent rather than the restaurant environment. Masks can be turned on or off using the 'masks' switch, and their inhalation and emission efficiency can be set using the sliders. Hand sanitiser use can also be turned on or off, and the thoroughness can be set using the slider. Doors can be opened, which would stop some potential fomite transfer.

The ventilation sub-model, section 3, is the last section that needs set up. Windows can be opened or closed using the switches, which will set up the restaurant's air currents and increase the air change rate. The air conditioning's air change rate can be set using the are-change-rate slider, AC recirculation can be turned on or off using the switch, and the number of portable HEPA filter can be set using its own slider also.

At this point set up is complete, so the user can click the 'setup' button in the top left to confirm the selected parameters. To run the simulation, the user can click the 'run simulation' button, which will begin execution. The execution speed can be set using the slider in the options bar at the top of the screen.

While the restaurant is running, the user can customise the view using switches in the view controls section to hide the links, chairs, and tables and make the aerosol concentration colouring more prominent. While the simulation is running, the user may wish to change some of the parameters, e.g., close the windows or remove masks. These changes will take effect from the moment they are changed. If the user has turned on manual customer spawning, they may now wish to click on tables to seat customers. They can also click on a customer, waiter or bartender to make them infectious.

Output data can be observed in the graphs and monitors at the bottom of the screen. The first section is showing data relating to the concentration of viral aerosols in the air. The second is data relating to the average fomite risk faced by the customers. The third, displaying data relating to the infection risk and the number of probable infections produced. The last section display general data such as the total number of customers served and the current occupancy on the seating area.