QATAR UNIVERSITY

COLLEGE OF ENGINEERING

INTELLIGENT AGENT-BASED 3D SIMULATION OF COVID-19 FOR ANALYSIS OF SOCIAL CONTROL MEASURES

BY

ABDULRAHMAN ZAYED AL-KHAYARIN

A Thesis Submitted to

the Faculty of the College of Engineering

in Partial Fulfillment of the Requirements for the Degree of

Masters of Science in Computing

January 2022

COMMITTEE PAGE

The members of the Committee approve the Thesis of
Abdulrahman Al-Khayarin defended on 23/11/2021.

Dr. Osama Halabi
Thesis/Dissertation Supervisor

Dr. Sarada Prasad Dakua
Thesis/Dissertation Co-Supervisor

Dr. Zhihan Lv
Committee Member

Dr. Mohammad Saleh
Committee Member

Prof. Junaid Qadir
Committee Member

Approved:

Abdel Magid Hamouda, Dean, College of Engineering

ABSTRACT

ALKHAYARIN, ABDULRAHMAN, Z., Masters : January : [2022:], Masters of Science in Computing

Title: Intelligent Agent-Based 3D Simulation of Covid-19 for Analysis of Social Control Measures.

Supervisor of Thesis: Dr. Osama, Halabi and Dr. Sarada Prasad, Dakua.

The aim of this research is to model and simulate the recent and ongoing COVID-19 pandemic in terms of virus contagiousness among mixed groups of patients, carriers, and unaffected populations. The focus is on closed environments such as stores or schools that are typically ideal for the propagation of infectious pathogens. This research work utilizes real data from the State of Qatar to model and predict the behavior of COVID-19 as it spreads within human gatherings.

This work proposes the infection model SEIP (Susceptible-Exposed-Infected-Protected) developed to forecast the propagation of COVID-19 over time. The prediction model is applied to simulate different environments of human gatherings for the viral transmission under different biological factors. Applied machine-learning techniques based on reinforcement-learning algorithms trains smart agents who mimic the behavior of their human counterparts. Added 3D visualization, by harnessing the power of Unity 3D, further boosts the usability and appeal of the simulation.

The resultant simulation is customizable and extendable to simulate a myriad of possible pandemic scenarios and evaluate different potential safety control measures. Ultimately, we aim to equip authorities with a powerful tool to aid in decision making about future spreading of the virus under different controllable lockdown scenarios.

DEDICATION

To my dear family, who supported and encouraged me during this journey: this work is dedicated to you.

ACKNOWLEDGMENTS

I would like to offer my sincere thanks and gratitude to my advisor, Dr. Osama Halabi, who deeply encouraged, motivated, and assisted me during this project with his valuable insights, precious guidance, and unlimited support. It was truly such an honorable opportunity to get to work with you.

I would also like to thank Dr Sarada Prasad Dakua from the Department of Surgery at Hamad General Hospital for taking the time to review this paper and provide us with constructive feedback and comments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

The SARS-CoV-2 virus, commonly known as COVID-19, has been rapidly and violently spreading all over the world. At the time of writing this research, COVID-19 has affected more than 235 million people and caused more than 4 million deaths globally [1].

Consequently, governments all over the world have implemented several measures to restrain the propagation of the disease. Social control measures are the most utilized countermeasures, and they have achieved varying levels of success, depending on several environmental and social factors. Such factors may include imposing physical distancing, enforcing hygiene measures (such as wearing masks), and distributing vaccines doses among the population. Therefore, when implementing different social and safety control measures, it is of utmost importance to assess a priori the possible impacts of such measures before deploying them in the real world.

Hence, motivated by the necessity to analyze such measures beforehand, we aim to support government entities and stakeholders with a framework for simulating the propagation of the virus under different scenarios.

Thus, the ultimate objective of our research is to design and develop an integrated 3D environment for modeling and simulating the spread of COVID-19 among a sample of human populations under various settings and conditions. We developed a customizable infection model for the purpose of computing the possibility of viral transmission while factoring in a myriad of configurable parameters and variables. We employed an agent-based modeling approach to simulate the crowd behavior within a shopping environment in our system. Reinforcement-learning methods are combined with our agent-based approach to build interacting and

intelligent agents. The overall simulation is augmented in a 3D space environment and implemented on top of Unity technology.

Problem Statement and Research Questions

The COVID-19 virus is propagating globally at a rapid rate while mutations of the virus occur periodically producing new variants in the process. Stakeholders are continuously applying social control measures to control the propagation of the virus among the populations. The effectiveness of control measures is dependent on several elements such as the adherence of people to those rules and the nature of the virus. Consequently, it is crucial for government entities to evaluate such control plans under different contexts and scenarios. In this research, we aim to develop a simulation environment for simulating the spread of COVID-19 within humans' populations under in-door terrains when taking into account different social and human factors.

Based on our problem statement we can formulate the following research questions:

1. Given the nature of COVID-19, what kinds of biological infection models can appropriately describe the spreading phenomena when considering social aspects of the population?
2. Which non-biological factors contribute most to the propagation of COVID-19?
3. In terms of simulating crowd behavior, how can we exploit machine-learning techniques to produce computer bots that resemble real humans in terms of movement behavior within in-door environments?
4. Which possible approaches we can follow to introduce crowd behavior simulation into a biological infection model in order to study the spreading under social scenarios?

5. When using 3D space, what possible advantages we can gain by using 3D space (instead of 2D) in terms of visualizing the spreading of a disease within human populations?

Paper Structure

We organize this research as follows: Chapter 2 will discuss background concepts relevant to the research. Chapter 3 surveys the literature on related and recent works in the field, along with our explaining contributions to the literature. Chapter 4 explains our methodology that we followed in conducting this research. Chapter 5 will demonstrate the technical design of our solution. Chapter 6 presents the results of our work, along with analysis and discussion of those results. Chapter 7 evaluates some of the limitations of our work and possible future research directions. Chapter 8 will conclude our thesis.

During the course of this research, the State of Qatar is the main target of our hypothesis and experiments.

CHAPTER 2: RELATED BACKGROUND

In this chapter we will summaries some of the relevant background concepts relevant to our research and then present our final approach that we will follow during the rest of the research. We will start in the first subsection by introducing the biological infection model behind our research. Since we are concerned with controlling the spreading of COVID-19 within social measures, in the next subsection we will present the crowd-behavior model that we will utilize to simulate human gatherings.

SIR Infection Model

In the literature several models exist for modeling the spreading of infections among human populations. The most popular models are compartmental models, which divide the population into different *compartments* representing different stages of a disease. The simplest and most well-known compartment model is the SIR model, first proposed by Kermack and McKendrick [2]. This model divides the population into three disjointed groups such that each member of the populations can only occupy one such group at any given point. The groups are S = *susceptible*, representing people who have a chance of catching a disease; I = *infected*, representing people who are infected with the disease and have a certain probability of transmitting it to susceptible people; and R = *recovered*, representing people previously infected with the disease and now recovered from it. As shown in Figure 1, the model is implemented as a state diagram with probabilities Pi and Pr representing the probability of infecting a susceptible person and the probability of recovering from an infection, respectively. As shown in equations (1, 2, 3), the change over time of the population status is described as a set of ordinary differential equations (ODE):

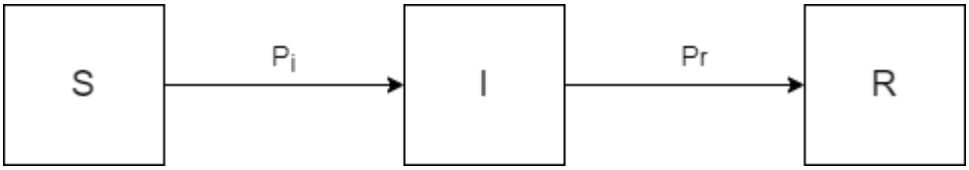

Figure 1. SIR Model

$$\frac{dS}{dt} = -\beta SI \quad (1)$$

$$\frac{dI}{dt} = \beta SI - \gamma I \quad (2)$$

$$\frac{dR}{dt} = \gamma I \quad (3)$$

The three equations (1, 2, 3) describe the rate of changes per time for susceptible, infected and recovered individuals respectively. S, I, R denotes the total number of susceptible, infected and recovered individuals in the population respectively. The parameter β represents the probability of an infection while the parameter γ represents the recovery rate (patients are assumed to recover following constant recovery rate γ).

Over the years, the SIR model has been extensively implemented and used to derive other models, such as the SIS (susceptible-infected-susceptible) model and the SEIR (susceptible-exposed-infected-recovered) model; SEIR adds the state *exposed* to represent infected individuals at a latent stage in which they may not show symptoms and may not transfer the virus to other people.

The main strength of the SIR-based models lies in their simplicity and their extensibility. Such models are deterministic and can be solved analytically or numerically. The main disadvantage of the compartmental models is that they ignore

the individual characteristics of humans (such as age) and deal with the population as one homogeneous unit. Those models output aggerated results that summarizes the whole population. However, the spreading of a pathogen can be influenced significantly by the individual characteristics of humans such as age or previous health records.

Agent-Based Modeling

Agent-based modeling (ABM) is a field of modeling that simulates a system as a set of interacting agents [3]. These agents are typically intelligent entities with decision-making capabilities that interact with other agents and with their environments based on predefined rules.

At the individual level, agent-based systems are simply composed of agents that move and interact according to certain rules. Those rules could be predefined rules set initially such that the agent will follow them during the course of the simulation. In other scenarios, the agents may display more stochastic behaviors by introducing random elements in the implementation of the agents. Agent-based systems are relatively simple to implement. However, such simple schemes are excellent at generating global patterns in a system, modeling the temporal evolution of a system, and showing the emerged complex behaviors of the interacting agents [4].

In terms of virus spreading, ABM has been deployed successfully in several real-life scenarios to accurately simulate pandemic spreading and inform policy decisions. One such example is the work of Eisinger and Thulke, which used ABM to predict the spread of rabies among a population of foxes [3]. Compared to classical differential equations, which predict that at least 70% of the population need to be vaccinated to control the disease, ABM predicted that a much smaller ratio of vaccination is required, thus saving millions of European euros in the process [3]. As

will be demonstrated in the next chapter, most recent works in the field of pandemic simulation (including COVID19) deploy some sort of agent-based scheme.

Reinforcement Learning

Reinforcement learning (RL) is a branch of machine learning that was invented in the early 1990s [5]. RL techniques are concerned with teaching intelligent agents how to make smart decisions within an environment. Unlike other machine-learning techniques, such as supervised learning (where agents learn from predefined labels or examples), in RL the agents interact directly with the environment by monitoring the environment and executing a set of actions accordingly to reach a predefined goal. By receiving appropriate rewards or penalties based on their actions, the agents learn to intelligently tackle the problem at hand using smart solutions. The agent may start with making random trial-and-error decisions and ending up with deploying optimal tactics to solve its problem. RL has been used in many fields, including AI in games, autonomous vehicles, and natural language processing.

In our project we will depend on compartmental schemes based on SIR to model the probability of infection between humans. Within the topic of modeling COVID-19 spreading, our main target would be evaluating social control measures under different scenarios. Therefore, it is crucial to model humans' behaviors. For simulating social movement of humans', we will utilize agent-based modeling to simulate the movement of humans inside an enclosed space. To further enhance the intelligence of our agents, we will exploit reinforcement-learning techniques in order to train agents that can mimic the behaviors of humans within closed spaces when taking into account social control measures.

Our Approach

In our project, we follow a hybrid approach combining stochastic ABM methodology with a compartmental infection model (based on SEIR) for simulating the

propagation of COVID-19 among humans. Compartmental models output macroscopic and aggregated results which can oversimplify the phenomena of infectious diseases. In reality, however, the distinct characteristics of agents, along with their behaviors and interaction, have enormous effects on the spreading of COVID-19 within a local population. Therefore, for a more informative description of COVID-19, it is crucial to model the viral transmission at the micro-level [6], at the level of local transmission between individuals. By employing stochastic models, we aim to closely mimic the real-life phenomena and produce more robust results. Despite those strengths, however, ABM may suffer from few issues. Mainly, the dynamic interactions of agents can lead to difficult-to-explain chaotic behavior, especially within stochastic models [7]. Consequently, the calibration of such model parameters is a challenging task, because the variation of a single parameter can have an unexplainable impact on the overall simulation results [7]. Therefore, we integrate a derivative of the SEIR model to lessen the impact of such stochastic factors and to help yield results that can be compared with real deployed compartmental models.

For a more effective implementation of an agent-based solution, we utilize RL techniques to produce autonomous agents equipped with intelligent decision-making capabilities for communicating with their environment and with other agents. Our ultimate objective is mimicking the behaviors of human movements within an enclosed environments in the presence of diseases. By integrating RL, our agents will gradually learn to move within an environment in a more organic matter while taking into consideration the presence of a virus among the neighbors' agents.

CHAPTER 3: RELATED WORK

In this chapter we review some of the recent works in the field of epidemic simulation with extra care given to works that deploy agent-based techniques. At the end of the chapter we outline our contributions to the literature.

Literature Review

Hassanat et al. [8] attempted to simulate the spread of the virus in the Kingdom of Saudi Arabia (KSA), taking into consideration the people's different movement behaviors and levels of hygiene compliance. The authors used the cities of Riyadh and Jeddah as simulation environments and scaled down the size and population densities of the cities to fit a simulation environment. The prediction model was verified using real infection data from KSA by comparing the output of the simulation against the historical real data within a set period.

Moein et al. [9] criticized the accuracy of mathematical SIR models, which are based on a set of differential equations, in forecasting future trends of the pandemic. The authors presented a case study using Isfahan in Iran, where an SIR model was deployed to forecast infectious cases of COVID-19 under different levels of social distancing restrictions. None of the scenarios had output results that conformed to the actual pandemic statistics.

In their work [3], Maziarz et al. discussed extensively the merit of using ABM for modeling the spread of COVID-19 from a theoretical and practical point of view. The authors presented an extensive argument in favor of ABM approaches compared with other differential equation models, and they advocate for the consideration of ABM as potential reasoning in the standard medical evidence hierarchies. Maziarz et al. argue that, although ABM does simplify certain real-world behaviors while discarding other features, the overall simulation still captures the core mechanisms of

the phenomena (the COVID-19 pandemic, in this case), provided that the model assumptions (e.g., incubation period, infection probabilities, reproductive number) are calibrated based on empirical studies. A modified version of AceMod (Australian Census-Based Epidemic Model) tailored to the characteristics of the COVID-19 virus (which was used to model the viral infection in Australia) was used as a baseline scenario to demonstrate the benefits of using ABM simulations.

Silva et al. [4] used an agent-based model to assess the economic effects of lockdown measures on the propagation of COVID-19 in Brazil. To model the spread of COVID-19 among the agents, Silva et al. argued that the SEIR model (susceptible-exposed-infected-recovered) is superior to the traditional SIR model due to the presence of the incubation period in COVID-19. In this experiment, the agents represented real-world entities (humans, households, businesses, governments) interacting in the presence of a contagious disease. The human agents were divided into different categories (employed, unemployed, and homeless), and their movements and behaviors were adjusted accordingly. The virus spreads when the distance between two human agents exceeds a certain minimum threshold based on an experimental contagion probability. Regarding the economic impacts, the economic environment was modeled as the transfer of wealth between the different agents involved in the simulation. People transfer wealth to business entities, and businesses use this wealth to pay employee salaries and government taxes. Governments then spend part of those taxes on health care institutes as part of the required expenses to run these hospitalization entities. Therefore, people's spending patterns are directly related to their mobility, which is affected by the lockdown measures, resulting in a lower number of transactions. The authors simulated several lockdown scenarios, ranging from partial lockdown measures to complete lockdown measures.

Chang et al. [10] used a highly modified version of AceMod calibrated with empirical COVID-19 parameters to assess different outbreak mitigation strategies across Australia, including home quarantine, school closures, and forcible social distancing measures. AceMod is a discrete-time and stochastic agent-based modeling system originally deployed in Australia in 2016 to simulate complex outbreak scenarios. AceMod consists of 24 million human agents, each with varying characteristics (age, gender, medical history) and social interaction behaviors within different contexts (households, schools, workplaces). Each scenario runs on a 12-hour cycle (representing day/night), where the infection spreads based on proximity and infection rate parameters.

Yang et al. [11] used a modified SEIR model that includes inbound and outbound parameters representing the flow of susceptible and exposed individuals across China. As a data source, the authors utilized official SARS statistics from 2003 and recalibrated the parameters (incubation, transmission, fatality) to account for the COVID-19 dynamics. The SARS data was used as a training set to train a recurrent neural network based on long short-term memory (LSTM) to forecast the spread of the epidemic in several provinces.

Guo et al. [12] designed a simulation of the outbreak of a respiratory disease within a military camp using a stochastic spread model. Within this model, the authors investigated the correlation between the attack rate and various epidemiological parameters (R_0, time of isolation, onset to isolation, and immunization rate). As an outcome of their work, the authors argued that a stochastic model is superior to traditional SEIR models because these traditional models assume the homogeneity of certain epidemiological inputs (e.g., contact rates), which does not realistically represent real-world phenomena.

In their work, Yang et al. [13] simulated an outbreak of influenza in Eemnes in the Netherlands to appraise the efficacy of various control measures. Their modeling approach was based on the Individual Space-Time Activity-Based Model (ISTAM), which is a novel bottom-up agent-based environment for mimicking epidemics among individuals at a fine space-time scale. The key concept of ISTAM is the *activity bundle*, which represents a spatial context where certain social interactions may occur. The activity bundle may represent households, schools, or shopping malls. The experiment simulates the infection among people existing within the same bundle, as well as people moving between different activity bundles.

In their paper, Chumachenko et al. [14] simulated an outbreak of the syphilis pathogen in Ukraine within an agent-based modeling environment. The key novelty of this work was splitting the infected state (I) of the SIR model into further sub-states (latent, primary, secondary, and tertiary) to represent the subsequent sub-states of the infected individuals.

Wong et al. [15] implemented a simulation of hepatitis C infection in Canada using a parallelized computing method based on a parallel sliding region algorithm (SRA). The outbreak simulation environment was the province of Saskatchewan and consisted of 45 interconnected sub-regions that were processed in parallel using multiple CPU threads. Two experiments were conducted to assess the effects of parallelizing the simulation process. One experiment utilized sliding region algorithm and the other processed the entire population simultaneously. The incidence rates were close enough between the different simulations, while the computing resources (e.g., computation time, memory consumption) were greatly optimized when applying SRA to the simulation.

Tabataba et al. [16] integrated beam filtering with an agent-based model to forecast the spreading of Ebola infections. The authors proposed an efficient particle filtering methodology called Smart Beam Particle Filtering (SBPF) in order to accurately calibrate the initial assumptions of the epidemic model (e.g., transmission rate, incubation period). At each time step of the simulation, the previous cycle's state vector (i.e., model parameters) and the observed values were used to predict the new state vector, which was then fed into the agent-based model to simulate the epidemic spread in the next simulation cycle.

In their work, Nakamura et al. [17] investigated the exponential growth of the complexity of agent-based simulations caused by the massive number of states the system can occupy, which represent the possible combinations of the infection states of the all the individuals in the experiment. In Nakamura et al.'s paper, the ABM method is modeled as a Markov process where the possible system configurations are represented by the different combination of infection states of the agents. The authors proposed two extended and computationally efficient methods based on the Markov process (Markov-Sym) and Monte Carlo (MC-Sym), respectively, that employ symmetry-based techniques to reduce the size of the transition matrix Tˆ (i.e., expressing the probability of transience between one system state to another).

In his work, Zhang [18] discussed the pitfalls of the traditional modeling of the spatial environments in agent-based simulations, which are usually implemented using a grid of discrete cells. Zhang argued that such spatial environments are limited to simulating a disease in closed or restricted spaces. Additionally, Zhang argued that grid-based systems are poor for representing public transportation carriers (such as buses and metros), which are ideal environments for spreading infectious diseases. Zhang proposed a new spatial model to organize "physical containers" in a hierarchal manner,

where each container can be composed recursively into smaller containers. Additionally, Zhang proposed the concept of “movable physical containers,” which represent crowded mobile vehicles where infectious contacts can take place.

Cuevas [19] explored the relationship between the mobility of humans and the probability of catching COVID-19 infection using ABS methodology within the context of indoor facilities. In this work, the probability of movement and infection rate are not set sectionally by dividing the population into distinct homogeneous groups based on age, for example; rather, these probabilities are completely controlled individually per agent in order to provide a detailed picture regarding the spread of COVID-19.

Hackl and Dubernet [2] used the MATSim framework to implement a large-scale simulation of an influenza pandemic within urban areas. In order to validate the proposed model, a numerical test was first run with varying model inputs in order to calibrate the model parameters, such as infection rate. Next, the calibrated parameters were used to simulate a seasonal influenza propagation in 2016–2017. The result was a good approximation of the real events considering the simplifications involved.

Kasaie et al. [20] designed a model to simulate the pandemic of tuberculosis using an agent-based methodology. One of the key aspects of their work was investigating the propagation of diseases when considering different layers of social networks, including households, neighborhoods, and local communities.

Zhang et al. [21] tackled the problem of modeling pandemics using agent-based methodology by adding a microscopic public transport infrastructure (e.g., buses and metros) to assess the effect of such systems on the infection rate within a large-scale metropolitan environment (Beijing, in this case). Different social groups of individuals (children, adults, elders) can choose to board a public transporter to reach different

destinations based on their daily schedules, such as going to school or work. Within a public transporter, healthy agents may catch infections from other infectious individuals.

Bobashev et al. [22] developed a hybrid epidemic model combining agent-based methodology with compartmental equation-based techniques (EBM). The key motivation behind this approach was to reduce the intensive computing time required by ABM by taking advantage of the relative simplicity of EBM under specific conditions. The simulation starts as a pure agent-based simulation (using a number of susceptible and infected individuals), and once the number of infections exceeds a configurable threshold, the simulation switches to a pure equation-based approach. The underlying justification for this hybrid approach is the assumption that, as the number of infected patients gets large enough, the overall dynamic of the disease spreading can be captured using compartmental methods, while providing adequate estimations in comparison with pure agent-based approaches.

Perrin and Ohsaki [23] presented a parallel approach for handling large-scale simulations of pandemics by combining an agent-based model with a social-network generation model. This hybrid model starts by generating a social network representing social contacts between agents, such as friends or family members, and the social network is modeled as a network graph where nodes represent agents and edges represent social links. To further optimize the performance of the simulation, the authors implemented parallel computing techniques to generate smaller sub-networks simultaneously before joining the sub-networks to produce the global social network. Next, they used the social network as an input in an agent-based environment to simulate disease propagation. Similar to social network modeling, parallelism was employed by running local simulations representing neighbors on separated clusters of

nodes. The novelty of this approach is that it provides a formal model of close social contacts between agents and models casual contacts in which both types of contacts are crucial in spreading an infectious pathogen.

Our Contributions

In terms of infection modeling, several works in the field depend on the SIR model or one of its derivatives. However, there is some lack of research that consider the status of vaccinated humans in their infection model, especially in recent works that deal with COVID-19. Given the increasing rate of vaccinations around the world, we argue that it is critical to consider the vaccination in our models to produce more accurate results. Therefore, we apply the SEIR model to derive the SEIP model. in this model, the R status is generalized to the P status representing all protected individuals. The word *protected* is a general term that refers to individuals who are immune to the virus because of a previous infection or vaccination, or from any other possible causes. Most of the recent works that discuss the spread of COVID-19 do not factor vaccination status into their calculation. However, a major part of our objective is to evaluate social control measures beforehand where the percentage of vaccinated people within a society can have major impact on the planning of such measures. Strict control measures can be softened in societies with high rates of vaccination. A major difference between the *protected* people within our model and the *recovered* population in the standard SEIR model, is the addition of partial immunity. In SEIP the immunity of an agent could be partial, in which case, a protected agent may get infected again depending on the level of immunity or efficacy of a vaccine. Each agent may have different immunity level depending on the source of immunity (recovering from an infection or by consuming vaccine doses) and the type of vaccine they consumed.

Additionally, the propagation of COVID-19 is a complex phenomenon that is shaped by a diversity of elements. Therefore, in the design of our model we incorporate a set of novel environmental, biological, and social elements, some of which we believe were never considered in similar works

From the literature survey, we can see that employing ABM is a common approach for simulating crowd behaviors. In agent-based simulations, the individual characteristics of the agents are critical when describing the transmission of the virus among human agents. One of the major shortcomings we encountered is the simplicity found within the implementation of agents. Many of the related works depend on static predefined rules to engineer the behaviors of their agents. Even works that use more stochastic models are still quite limited in terms of emulating real-life humans within crowd behaviors. To further improve the efficiency and effectiveness of crowd behavior in comparison with similar projects, we focused on incorporating machine-learning algorithms in our simulation. We utilized reinforcement-learning techniques when developing the agents for the ultimate purpose of constructing smart and competent agents. Within the context of our system, the term 'smart' refers to the ability of a computer agent to emulate its human counterpart in a reasonable manner. By integrating reinforcement-learning we aim to teach our agents how to mimic real humans' behaviors in terms of movement and interaction. As will be discussed in more details later in the methodology and implementation sections, our agents will learn to properly mimic real humans by observing the environment and receiving appropriate rewards according to their actions. In this sense, our agents will use minimum set of predefined behavioral rules.

Furthermore, in our analysis of the literature we found that most of the related simulation efforts are constrained to 2D agents within 2D grids. We argue that the

integration of a 3D plane can have positive effects on the visual and functional aspects of a simulation. Adding a 3D plane empowers us to properly visual objects and environments in terms of height, width, and depth. Functionality-wise, the integration of a 3D space support emulating extra scenarios that are difficult to simulate effectively within a 2D grid (such as navigating within stairs for example). Hence, we utilize the Unity platform [24] to provide an integrated 3D simulation environment for predicting and visualizing possible spreading of COVID-19 when considering a variety of infection and social factors.

Based on our proposed contributions, the next chapters will discuss the various aspects of the methodology of our research and the design of our system.

CHAPTER 4: METHODOLOGY

Simulation Environment

Our simulation environment resembles a grocery store where visitors or shoppers represent the agents. Such places are expected to be crowded with people buying items and traversing the store within narrow aisles, making them ideal environments for emulating the spread of a pandemic among a human population.

Simulation Agents

The agents of this simulation represent humans with shopping capabilities who navigate through a grocery store, pick up items, and line up in queues to pay for their items. Additionally, the agents are trained to honor social and safety control measures as set by the store policies. We used machine-learning techniques (based on reinforcement-learning algorithms) to properly train intelligent shopping agents to mimic the *behavior* of humans within shopping stores. In our agents' model, the agents *observe* the enclosed shopping environment as well as neighboring agents. Based on those observations, an agent makes proper decisions or *actions* regarding navigation and shopping, and the agent receives appropriate *rewards* based on its conformance with the desired shopping behavior. Similarly, the agent is penalized accordingly when engaging in undesired behaviors. In the context of our research. the term 'desired behavior' refers to realistically emulating behaviors of human within shopping environments. Unlike other fields such as games where a super-agent is desirable, our agents will be relatively smart to the extent it is appropriate to mimic the intelligence of real humans.

As shown in Figure 2, at a high level, the simulation proceeds as follows:

1. Shoppers or agents enter the store from one of the entrances. Some of the agents will be infected with COVID-19 and will spread the virus in the store.

2. Each shopper will have a shopping list and will start to navigate the store and pick up items from this list. The behavior of picking up items will be assumed to be uniform among different shoppers.
3. Once shoppers finish picking up items from their shopping list, they will search for a checkout counter to pay for their items. If an available counter is found, the agent will occupy it; otherwise, the agent will navigate the store randomly and visit random shopping points until a counter becomes free.
4. Finally, the shopper will exit the store through one of the store exits.

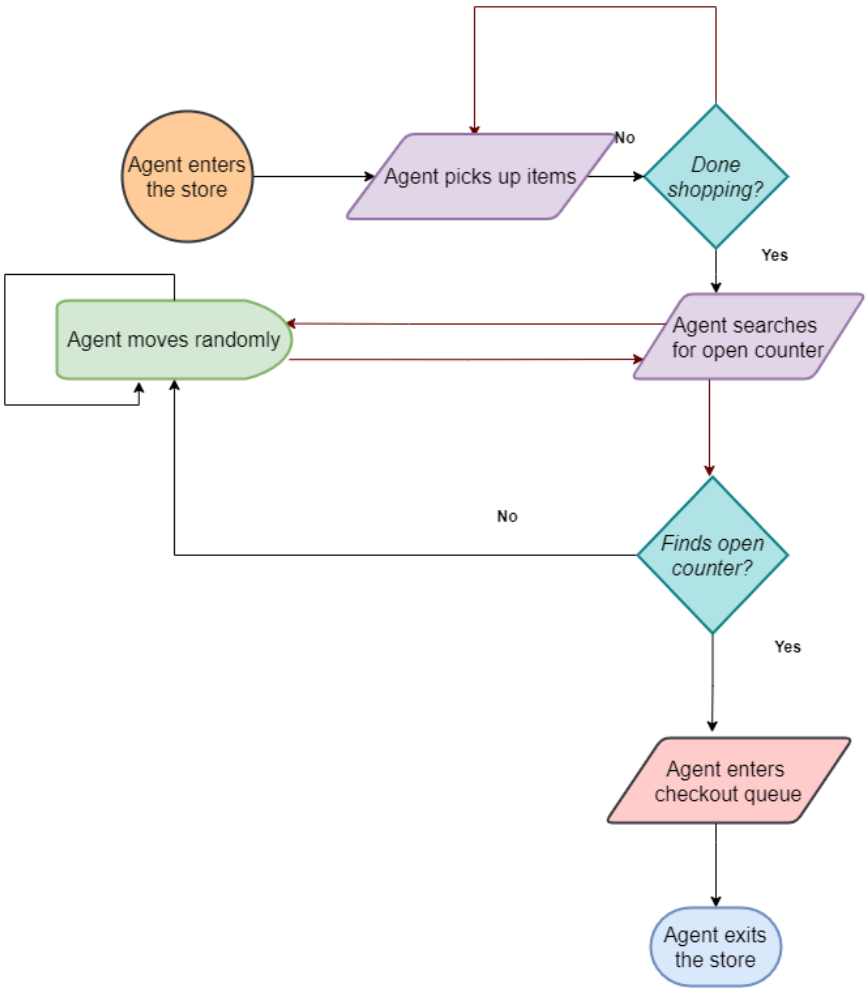

Figure 2. Simulation Flow

Infection Model

Infection Stages Scheme

In the implementation of our infection mechanism, we utilized the well-known SEIR model to derive the new model, SEIP, as visualized by Figure 3. When a healthy person in susceptible status S contacts an infected person I, this person may move into the asymptotic exposed status E. An exposed person will eventually start to show symptoms and reach fully infected status I (after a certain incubation period) where they start to infect other humans. Similarly, after a certain amount of time (based on a certain recovery rate). an infected person will either recover or die. The incubation period and recovery rate are outside the scope of our research so they will not be included in our calculations. The P status in our model stands for "protected" and comprises people who are resistant to the virus either because they have recovered from a previous infection or because they have been vaccinated. In our model, a protected individual can still catch the virus based on a probability P_{pe} but with a smaller chance than a susceptible person ($P_{pe} < P_{se}$). As will be explained shortly, the probability of infecting a protected individual is based on the efficacy of a vaccine. For protected people, our research will focus mainly on vaccinated agents; therefore, from now on, we will use the terms "vaccinated" and "protected" interchangeably to refer to the same concept. The term "healthy" will refer to all types of agents who can catch an infection (both susceptible and protected/vaccinated in this case). We will not deal with "death" status in this paper, as our focus is mainly on the rate of infection. One important point to consider is that for a person to be classified as completely vaccinated, they need to receive at least two vaccines doses. We will assume that the majority of people who receive one dose, will continue to receive a second one. Therefore, when dealing with

the total number of vaccine doses within a period, we will divide the number of doses by 2 to obtain the number of vaccinated people within this period.

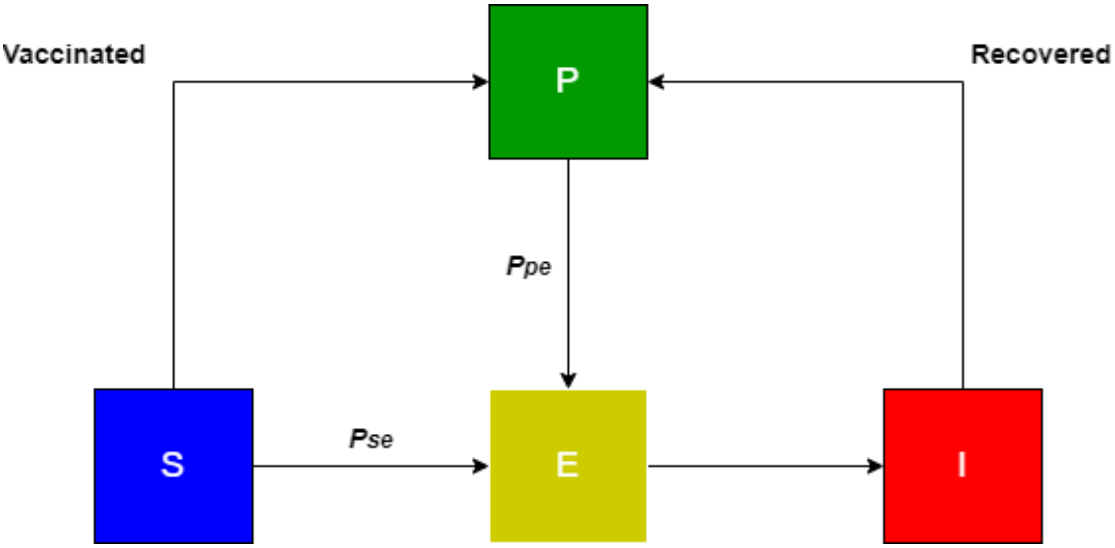

Figure 3. SEIP Model

Infection Formula and Parameters

At the simplest level, the spread of COVID-19 occurs when an infected person *I* encounters a healthy individual *H* where an infection probability function $P_I(H, I)$ is computed to determine the chance of infection. In reality, the possibility of exposure is dependent on a myriad of factors, including human-controllable factors (e.g., physical distance and mask usage), as well as biological factors related to the mechanism of the pathogen. In this research, we analyzed several possible parameters and classified them into the following categories:

1. Biological parameters ***B***: parameters related to the nature of the pathogen, such as the severity of the virus and the required physical distance for an infection to occur.

2. Human-social parameters ***S***: parameters related to the behavior of human agents within the simulation, such as the respect of social distance and usage of safety measures.
3. Environmental parameters ***E***: parameters related to the global simulation environment, such as initial number of infected individuals and size of the simulation space.

Table 1 summarizes the parameters we collected in our experiments, and in the following paragraphs we briefly explain those parameters.

Table 1. List of Simulation Parameters

Parameter	Abbreviation	Category
Simulation area size	Area	Environmental
Simulation max. time	ST_{max}	Environmental
Total number of agents	T_a	Environmental
Number of infected agents	I_a	Environmental
Number of vaccinated agents	V_a	Environmental
Number of open counters	C_{open}	Environmental
Social distance adherence ratio	SD_r	Human-Social
Respect social distance	$Flag_{sd}$	Human-Social
Hygiene measures adherence	HM_r	Human-Social
Respect hygiene measures	$Flag_{hm}$	Human-Social
Number of shopping points	C_{sp}	Human-Social
Distance between 2 agents	D	Human-Social
Is agent vaccinated?	$Flag_v$	Human-Social
Max. infection distance	D_{max}	Biological

Parameter	Abbreviation	Category
Infection probability at zero distance	IP_{zero}	Biological
Infection probability at max. distance	IP_{max}	Biological
Angle flag	$Flag_a$	Biological
Angle factor	AF	Biological
Min. vaccine efficacy	VE_{min}	Biological
Max. vaccine efficacy	VE_{max}	Biological
Min. face protection efficacy	FP_{min}	Biological
Max. face protection efficacy	FP_{max}	Biological
Min. face viral reduction	FVR_{min}	Biological
Max. face viral reduction	FVR_{max}	Biological
Surface-spreading flag	$Flag_{ss}$	Biological
Surface-lingering probability	SLP	Biological
Surface transmission probability	STP	Biological

Area refers to the size of the simulation space in m^2. ST_{max} indicates the maximum time in seconds after which the simulation will terminate. T_a, I_a, V_a represents the total number of agents, total number of infected agents, and total number of vaccinated agents within the simulation, respectively. C_{open} refers to the number of open counters available for checkout.

The bit parameter $Flag_{sd}$ indicates whether an agent respects the recommended minimum social distance or not (0 = no, 1 = yes). This flag is affected by another parameter, SD_r, which determines the average percentage of agents who keep physical distances when interacting. Similarly, HM_r signifies the ratio of agents who honor hygiene safety measures (wearing face masks in our research), which in turn affects the

hygiene flag $Flag_{hm}$. C_{sp} controls, on average, how many store shelves an agent will stop at to pick up items (as determined by the size of the shopping list). D refers to the distance between a healthy agent and an infected one. The last flag, $Flag_v$, indicates whether the agent is vaccinated or not.

D_{max} refers to the maximum distance in which the virus can be transmitted between two persons. The distance between two agents D must be <= D_{max} for an infection to happen. IP_{zero} and IP_{max} represents the probability of infection when D = 0 (two agents are very close to each other) and D = D_{max}, respectively. Intuitively, the chance of an infection is inversely proportional to the distance; therefore, IP_{max} <= IP_{zero}. As the two agents get closer and closer to each other, the probability of infection increases. The variable $Flag_a$ controls whether to take into consideration the angle between agents when calculating the overall infection probability. Our assumption is that there is a higher possibility of infection when two agents are facing each other than if they are facing in different directions. AF controls the effect of the angle on the final probability.

The variables VE_{min} and VE_{max} refer to the lower and upper efficacy of the vaccine against new infections if the agent is properly vaccinated. We will assume that given a particular COVID-19 variant, the efficacy of the vaccine against this variant will remain constant over time.

The variables FP_{min} and FP_{max} refer to the lower and upper protection capacities of a mask against COVID-19 transmission, respectively, when worn by healthy persons. FVR_{min} and FVR_{max} refer to lower and upper viral load reduction of a mask when worn by an infected individual. We decided to model the factors related to mask protection and vaccines as ranges of values rather than singular discrete values because

we found high variance in the reported values in the literature [25], [26]. Therefore, estimating a singular value would be challenging.

The last three parameters deal with the spreading of COVID-19 through aerosols and on solid surfaces. Based on recent research by Doremalen et al. [27], the virus can linger on surfaces for six-seven hours and then be transmitted to a healthy agent who comes into contact with it. $Flag_{ss}$ controls the consideration of the individual-surface spread within the simulation, while parameters SLP and STP control surface-lingering and transmission probabilities, respectively. SLP deals with the lingering of the virus on a surface, while SLP deals with the virus infecting a healthy agent who comes into contact with an infected surface.

Initially, we considered adding the parameter *Age* to study the spread of COVID-19 among different age groups. However, based on our findings [28], age seems to have no major effect on the probability of infection. Age has a bigger effect on the probabilities of hospitalization and death, which are not relevant to our project. Therefore, we decided to ignore age in our calculations.

Based on the above parameters, we can derive the final formula to calculate the probability of infection when a healthy agent encounters an infected one $P_{I,H}$. This probability depends on multiple inner factors. First, we define D_r as the ratio of the distance between agents D to the overall maximum infection distance D_{max}. Next, we define the base probability P_o(H, I) as follows:

$$P_o = IP_{zero} * (1 - D_r) + IP_{max} * D_r \quad (4)$$

Equation (4) interpolates the probability of infection based on the probability of infection at min distance (IP_{zero}) D_0 and probability of infection at max distance D_{max} (IP_{max}). When the distance between two agents = maximum infection distance D_{max}, the

final probability $P_o = IP_{max}$. Likewise, when D_r approaches zero, P_0 will converge to IP_{zero}.

Next, we need to consider the effect of the angle ∠ between the agents on the overall probability if the angle $Flag_a$ is set to 1 (consider the angle in the calculation as mentioned previously when discussing the model parameters). P_1 denotes the new infection probability given the angle flag $Flag_a$, angle factor AF, and base probability P_0.

$$P_1 = P_o + P_o * Flag_a * AF * \frac{(\angle - 90°)}{90°} \quad (5)$$

This equation maximizes the probability when two agents are facing each other (Angle = 0) and minimize it when the two agents are looking in opposite directions (Angle =180).

The next factor to consider is the effect of a vaccine V_{factor} if the healthy agent is vaccinated.

$$P_2 = P_1 * (1 - Flag_v * R(VE_{min}, VE_{max})) \quad (6)$$

The terms R() refer to a function that generates a random number in a certain range: in this case, the lower and upper values of the vaccine's efficacy. If the healthy agent is unvaccinated, $P_2 = P_1$ (from the previous step).

The final critical factor is the effect of the face mask F_{factor} if worn by either agent.

$$P_3 = P_2 * \left(1 - Flag_{hm}(H) * R(FP_{min}, FP_{max})\right) * (1 - Flag_{hm}(I) * R(FVR_{min}, FVR_{max})) \quad (7)$$

$Flag_{hm}(H)$ and $Flag_{hm}(I)$ refer to the hygiene statuses of the healthy and infected agents, respectively. The final probability is $P_{I,H} = P_3$. If none of the agents wear a mask,

$P_3 = P_2$ (from the previous step). Similarly, if only one agent is not wearing a mask, the corresponded term in the equation (7) will default = 0.

As discussed previously on page 22, we also consider the probability of individual to surface viral transmission. Therefore, we will define two extra probabilities: $P_{I,S}$ is the probability of spreading the virus from an infected individual *I* to a solid surface *S*, as given by equation (8).

$$P_{I,S} = SLP * (1 - Flag_{hm}(I) * R(FVR_{min}, FVR_{max})) \quad (8)$$

We can see that $P_{I,S}$ depends on the surface-lingering probability SLP and the viral reduction efficacy of a mask worn by the infected agent.

The second probability $P_{H,S}$, given by equation (9), deals with the transmission of the pathogen from a contaminated surface *S* to a healthy agent *H*. This probability is dependent on the surface transmission probability STP, as well as on the hygiene and vaccination status of the healthy agent.

$$P_{I,S} = STP * \left((1 - Flag_{v} * R(VE_{min}, VE_{max})\right) * (1 - Flag_{hm}(H) * R(FP_{min}, FP_{max})) \quad (9)$$

Dataset

The main dataset used in this research (pulled from [29]) consists of the daily COVID-19 statistics published by the Ministry of Public Health in Qatar, which spans the period from 2020-03-14 to 2021-09-10. The most relevant statistics for our research are as follows:

- Date
- Daily number of new positive cases (DNP)
- Daily number of new tests (DNT)
- Daily number of new vaccine doses (DNV)

- Total number of administered vaccine doses (TV)

We preprocessed the dataset to compute the daily infection rate (DIR) and the total vaccine rate (TVR) in Qatar based on the following formulas, respectively:

$$DIR\ \% = \frac{DNP}{DNT} * 100 \quad (10)$$

$$TVR = \frac{TV*0.5}{2572198} \quad (11)$$

As discussed in the infection stages scheme section, vaccination rate refers to people who receive at least two vaccine doses. Therefore, TVR represents the rate of getting two doses.

We assume that the ratio of new positive cases to new tests will provide approximation of the ratio of new infected people in Qatar. The number 2,572,198 in the second equation represents the total population of Qatar [30]. We divided the total number of vaccine doses by 2 because most people receive two vaccine doses over a period.

Simulation Algorithm Procedure

Based on the previous sections, we have developed a general pseudo-algorithm that describes the overall simulation workflow, as shown in Figure 4.

The algorithm takes as an input a list of configuration parameters *P* and outputs a list of simulation results *R* (e.g., the infection rate % and total simulation time). The simulation starts by executing the procedure *initSimulation* to prepare the environment, such as by creating an initial number of shoppers and preparing the counters.

At each time step of the simulation, new agents enter the shop via *enterShoppers*, and all agents are instructed via the procedure *doAction*() to perform associated actions, such as moving around the store or picking up items. For each infected agent, the simulation finds all susceptible and protected neighbors within the

distance D_{max}. For each neighbor agent, a probability of infection is calculated, and if the probability is satisfied, the agent status is updated to “exposed.”

At the end of each time step, some agents exit the store via *exitShoppers*, and the statuses of all checkout counter queues are updated. Once the elapsed time of the simulation exceeds the maximum allowed time ST_{max}, the simulation stops and returns the results of the simulation R.

```
BEGIN
    INPUT P ## P = Simulation Parameters
    OUTPUT R TO NULL ## R = Simulation Results
    SET simulationTime TO 0
    SET S TO NULL ## All Shoppers
    CALL InitSimulation(Parameters = P, Shoppers = S)
    WHILE simulationTime < maxSimulationTime
        INCREMENT simulationTime
        CALL enterShoppers()
        FOR s1 in S
            CALL s1.doAction()
            IF s1.Status == "INFECTED"
                FOR s2 in s1.Neighbors
                    IF s2.Status == "SUSCEPTIBLE" OR s2.Status == "VACCINATED"
                        SET shouldExpose = CALL calculateInfectionProbability(Infected = s1, Target = s2)
                        IF shouldExpose == TRUE
                            SET s2.Status = "EXPOSED"
                        END IF
                    END IF
                END FOR
        END FOR
        CALL updateQueues()
        CALL exitShoppers()
        CALL updateResults(Results = R)
    END WHILE
    RETURN R
END
```

Figure 4. Simulation Algorithm Pseudo-Code

CHAPTER 5: IMPLEMENTATION

Simulation Technology

We used the Unity [24] engine for developing our simulation and used C# as our language. Unity is a cross-platform game engine used for developing 3D games. It is an ideal choice for running computer simulations since it encapsulates most of the components required for building 3D graphic simulations, such as physics engines, collision detection, navigation mechanism, and 3D visualizations. Within the Unity sandbox, we utilized the ML-Agents package for programming the agents' behaviors. ML-Agents [31] provides capabilities for transforming any Unity game environment into a learning environment where game objects (or agents) can be trained using various training algorithms, such as reinforcement learning and imitation learning. Essentially, ML-Agents is an excellent toolkit for teaching intelligent agents complex behaviors. In the context of our research, our goal is to utilize the ML-Agents toolkit for programming agents with shopping behaviors that mimic real-life human shoppers.

The preprocessing, analyzing, and manipulating of the dataset and results are done using the R language within RStudio [32].

Base Implementation

We built our simulation on top of the Unity Grocery Store Simulation project [33], which is an open-source project built by the Unity team to demonstrate how Unity can power complex computer simulations. This project simulates COVID-19 propagation within a grocery store among a population of human shoppers. We extend this project to implement our infection model and customize the behavior of intelligent agents using machine learning. In the next few sections, we will discuss some of the implementation details and enhancements we added to the project.

Figure 5 shows a high-level architecture of the complete solution and depicts the different layers of the solution. The simulation is built on top of the Unity engine,

which houses the essential components of the infrastructure, such as collision detection, physical rendering, and game controls. The essence of the simulation is implemented through the simulation core layer. The core layer oversees setting up and terminating the simulation environments and managing the set of shopping agents. The layer of the shopping agents encompasses the collection of agents deployed by the simulation core within the environment. Each agent captures a set of properties expressing their social and health statuses, as well as an ML-Agents brain encapsulating the intelligent decision-making capabilities of the agent. The last component is the infection model, which implements the necessary logic of computing the infection spreading. This component is independent from the Unity engine framework and is initialized and queried by the core layer to find the probabilities of infection among the different agents. In the next few sections, we will discuss in detail the various layers, starting with the core layer.

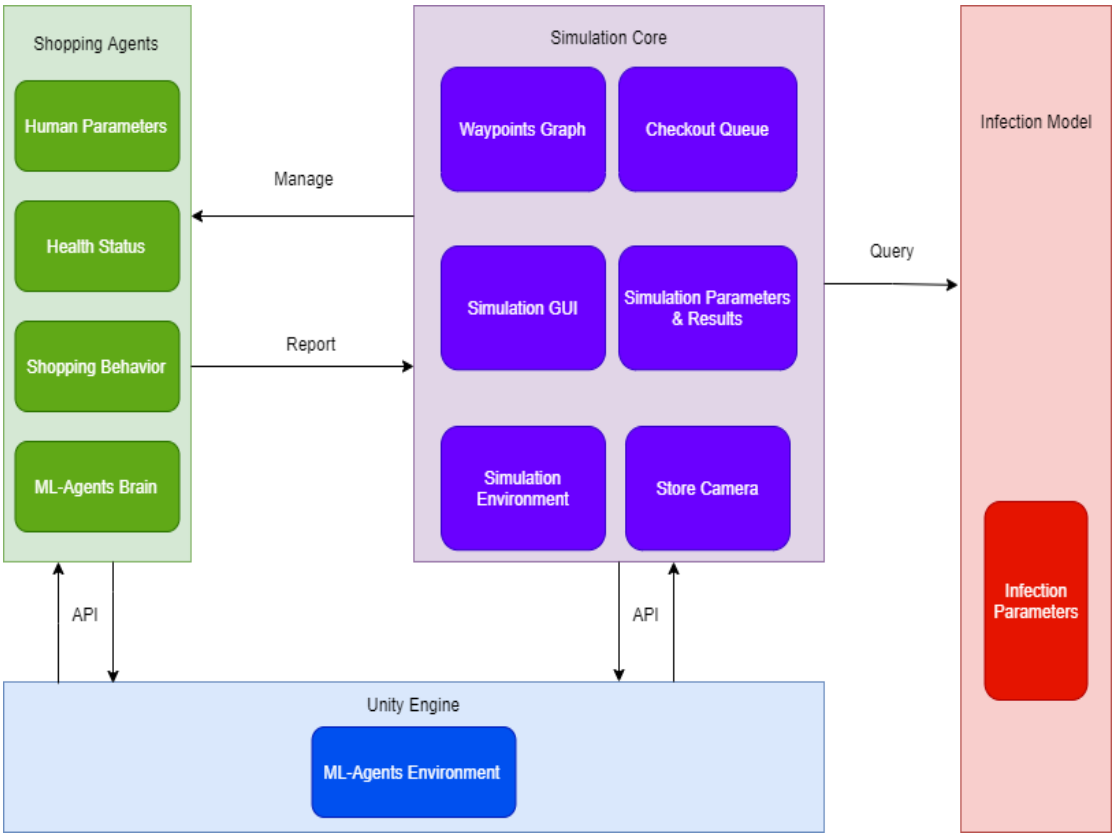

Figure 5. System Architecture

Simulation Core Layer

Waypoints Graph

The movement system within the store is powered by an implementation of a waypoints graph [34].

Figure 6 displays an example of the waypoints graph that consists of a set of waypoints or nodes (indicated by cyan squares) connected by edges (indicated by cyan arrows). The waypoints are classified into entrance nodes (indicated by green squares and only have outward-facing edges), exit nodes (indicated by red squares and only have inward-facing edges), and regular intermediate nodes (cyan squares). Among the regular nodes, a few nodes are annotated as “Registers” to indicate that those nodes

represent checkout queues where an agent must wait for the queue to finish before proceeding. Within this graph, an agent can traverse the store in any of the four directions (up, down, left, right) along any of the connected nodes.

In this scheme, two points, wp_1, wp_2, are assumed to be connected if the following criteria are satisfied:

1. A path $P(wp_1, wp_2)$ exists between the two nodes (distance $D(wp_1, wp_2) \neq \infty$).
2. $D(wp_1, wp_2) <= D_{wpMax}$. The parameter D_{wpMax} indicates the maximum allowed distance between any two pairs of nodes. By default, this variable is set to ∞, indicating that any node can connect to any other node in the graph.
3. The angle between the two nodes $\angle(wp_1, wp_2)$ must be within a certain configurable threshold A_{wpMax}.
4. There must not be any intermediate obstacles between the two nodes, such as shopping shelves.

In our implementation, an agent enters the simulation randomly through one of the entrances. The list of shopping items comprises a list of random intermediate regular nodes that an agent must traverse before moving to the cash registers. The agent finishes the experiment by exiting using one of the exits nodes. At every point in the simulation, the number of shoppers <= T_a (the maximum allowed number of agents.)

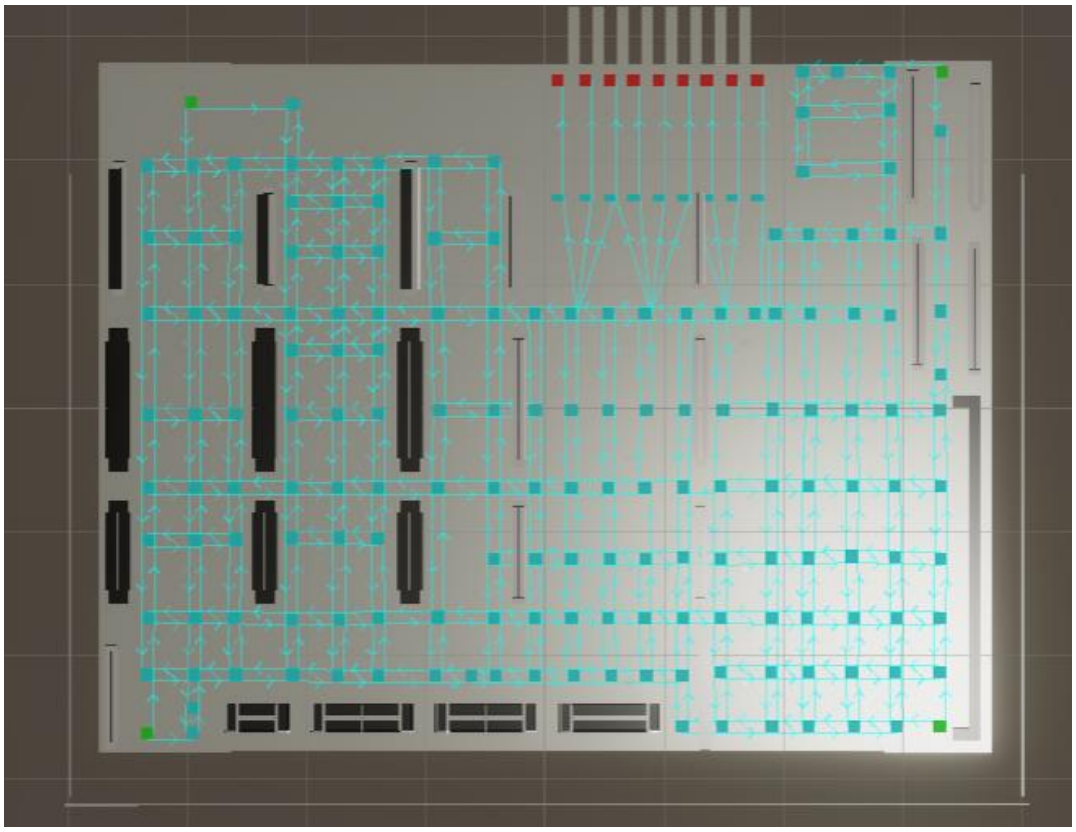

Figure 6. Graph of Waypoint Nodes

Simulation Environment

The environment is a 3D map of an imaginary grocery store with shopping shelves, aisles, and checkout counters. Figure 7 displays a screenshot of the map.

One important point to consider is that we are using a 1:1 scale in our simulation; therefore, a volume of m^3 in the real world translates into one unit in the game world (x = 1, y = 1, z = 1).

Figure 7. 3D Virtual Store

Store Camera

To implement the simulation, we developed a fully controllable 3D camera to better visualize the simulation at different angles and zoom levels; Figures 8 and 9 depict some different viewpoints.

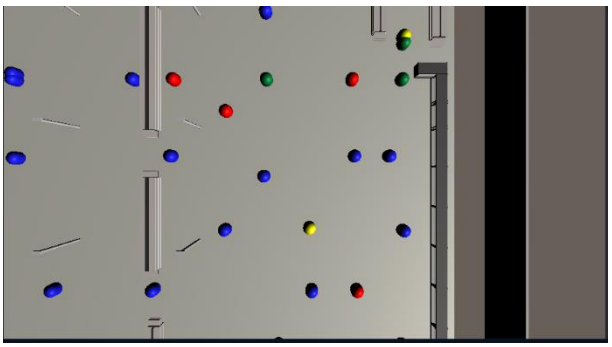

Figure 8. Top View Camera

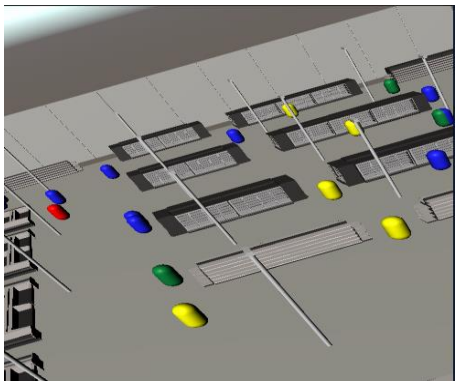

Figure 9. First-Person View Camera

Checkout Queue

The simulation queue component of our model emulates real-life billing for items at checkout stations. The number of available open counters is configured through the variable C_{open}, discussed previously in the Methodology chapter. When shoppers complete their shopping list, they search for a vacant counter. If an available counter is found, the shopper enters the queue; otherwise, the shopper roams the simulation randomly until an available queue is found. Figure 10 is a screenshot from the system depicting queues of agents at different checkout counters.

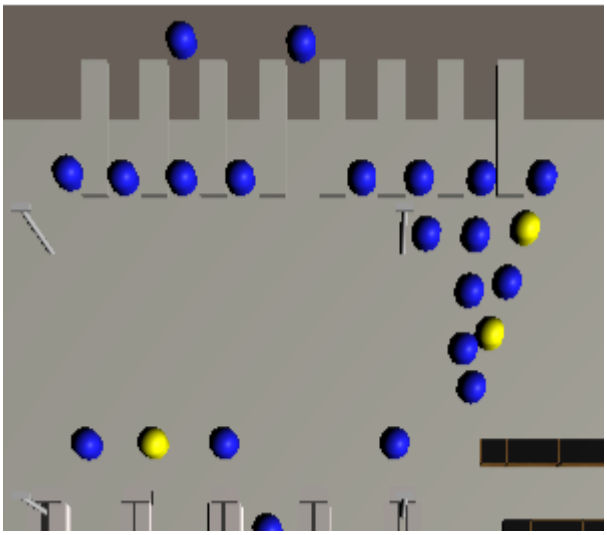

Figure 10. Checkout Queue

Simulation GUI

Finally, a GUI screen is provided for controlling all the different simulation parameters, as demonstrated in Figure 11.

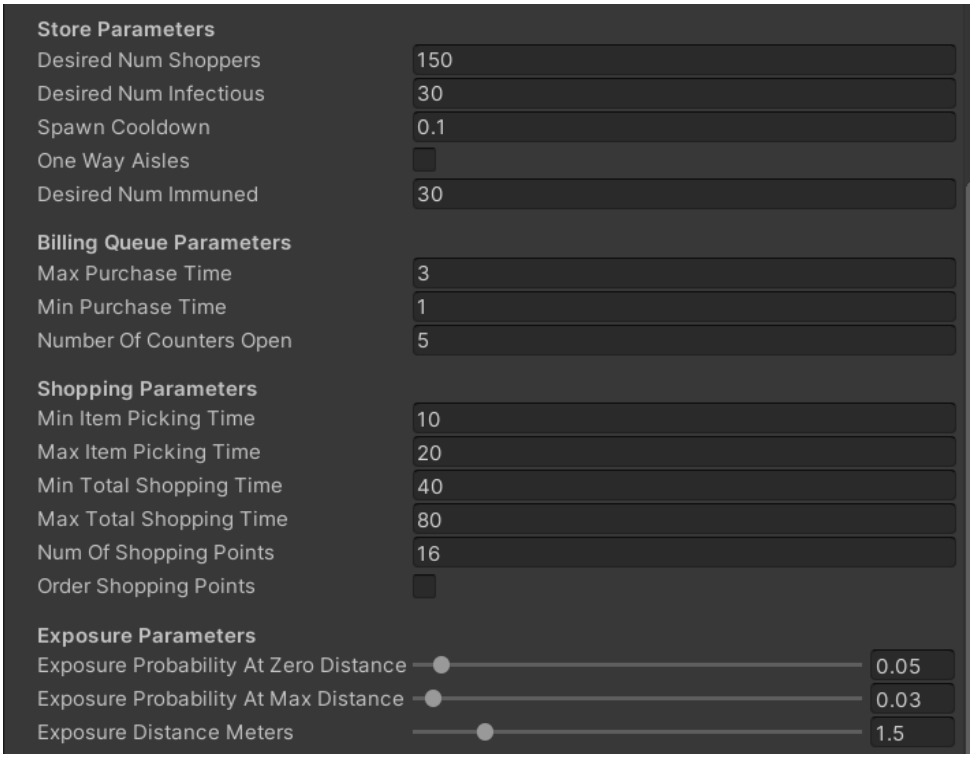

Figure 11. Parameter Configuration GUI

Infection Model

The shoppers in the simulation assume different colors based on their health or infection status (Susceptible Exposed, Infected, Vaccinated). Blue indicates susceptible, yellow indicates exposed, red indicates infected, and green indicates vaccinated. As explained before in the Methodology, an infection occurs based on a probability function when a healthy shopper gets close to an infected one within distance D_{max}. Within the context of the Unity engine, this probability is calculated once per frame per second. At each frame, a sphere or *OverlapSphere* [35] with radius D_{max} encompasses each infected agent. All healthy agents within this sphere have a chance of becoming exposed. Figures 12 and 13 demonstrate the infection behavior. Agent A1 is in the center of a circle with radius D_{max} surrounding several agents. All agents outside the sphere are ignored. For each of the inner healthy agents, an infection probability is computed. At the next frame, Agents A2 and A4 get infected and move

into the exposed category, while the rest of the healthy agents manage to survive this frame.

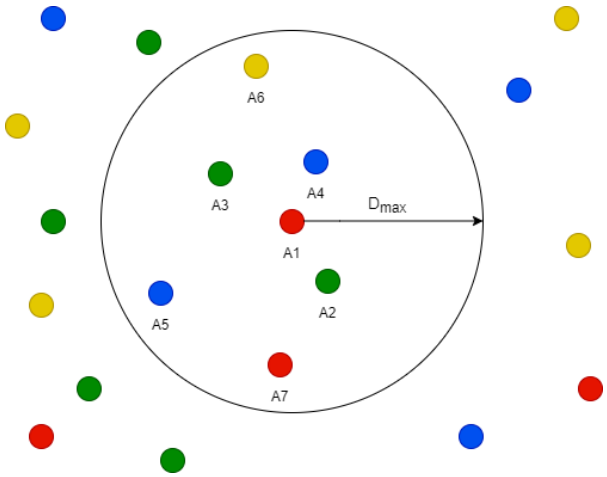

Figure 12. Agent A1 Infection Sphere (Before Infection)

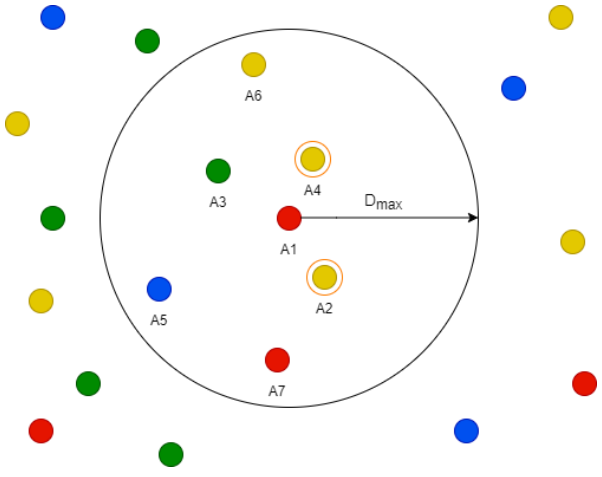

Figure 13. Agent A1 Infection Sphere (After Infection)

Shopping Agents Layer

The final component of the simulation is the agents layer encompassing the shoppers in our virtual store. The shoppers are visualized as spherical 3D objects with different colors illustrating different infection statuses. In addition to a healthy status, each shopper is also associated with other configurable real-life properties such as speed of movement, wearing of masks, and respecting of social distances. Those variables directly affect the spread of COVID-19.

As discussed in the subsection Waypoints Graph), the shoppers navigate through the store by means of a grid of waypoint nodes. A shopper arrives at the store using one of the entrance nodes, travels throughout several intermediate nodes (denoting shelves of goods), and exits using an exit node. In the base implementation of the solution, the shopper's trip around the store is determined beforehand when the shopper is first deployed to the environment. Using the nodes graph, the simulation core generates a path of nodes starting with a random entrance, passing through a list of intermediate goals ordered by distance, and ending with a random exit. The shopper follows this path with a constant speed until the end of the trip [34].

In our customization of the project, we re-engineered the movement system to harness the capabilities of the Unity ML-Agents package [31]. As discussed in the Methodology, we depended mainly on reinforcement learning to train our agents. The overall process is performed in two phases: training phase and inference phase. In the training phase, the agent is deployed to a smaller training environment to learn shopping behavior, with the overall goal of maximizing the received rewards. In the inference phase, the agent is equipped with a functional brain and is thus deployed to the real simulation environment. In the next few sections, we will discuss some of the details regarding the training of the agent.

Training Settings

We utilize proximal policy optimization (PPO) in our reinforcement-learning implementation. Table 2 presents some of the key configurations in our implementation, based on default toolkit configurations from [36]:

Table 2. RL Training Configuration

Parameter	Value
Trainer Type	PPO
Max. Steps	5,000,000
Time Horizon	64
Learning Rate	0.0003
Learning Rate Schedule	Linear
Hidden Units	64
Number of Layers	2
Batch Size	64
Buffer Size	12,000
Beta	0.001
Epsilon	0.3
Lambda	0.99
Number of Epochs	3

Environments

The agent will train within a grocery store environment to learn properly a shopping behavior. For the sake of speeding up the training process, the training will be done in a smaller environment. We aim to train the agent in an environment-independent manner.

Goals

The shopper agent will work on maximizing two subsequent goals:

1. Navigating the nodes graph with efficiency: The agent needs to learn to navigate the graph along a path with start and end nodes without prior knowledge of the graph.
2. Completing the shopping list: After training to achieve the first goal, agents should be able to go over all the nodes indicated by the shopping list and arrive at the checkout counter to pay for their items.

Observations

To properly observe the environment, each agent is equipped with two RayPerception Sensors [37]. The first sensor RPS_{env} is environment-specific and is responsible for monitoring the surrounding environment, mainly the nearest waypoint nodes. When an agent needs to move to a new position, the sensor is used to search for nodes connected to the current position and avoid choosing directions with dead ends. The second sensor RPS_{agt} is agent-specific and is responsible for monitoring the other agents. If an agent is parameterized to respect social distance (Flagsd = 1), this sensor is utilized to search for empty waypoints (i.e., no agent is currently associated with those points). If no available waypoint is found, the agent waits indefinitely in its current position until one is found. Figure 14 illustrates the different rays, where red rays indicate RPS_{env}, and green rays indicate RPS_{agt}.

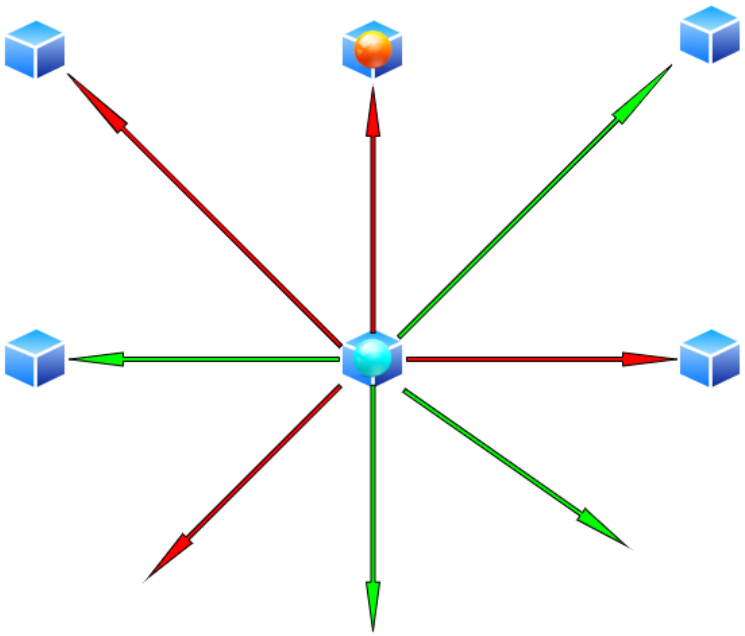

Figure 14. Agent Sensors

Actions

The agent decides randomly to navigate to a neighbor's node by selecting a movement direction from an array of possible directions (forward, backward, left, right). If the selected direction leads to a dead end (i.e., no node is connected to this direction), the agent selects a "wait" action and requests a new action from the brain. Similarly, if the selected node is occupied and the agent is set to respect social distance ($Flag_{sd} = 1$), the agent will also wait until the brain issues a new action. This process is repeated until a valid direction is found. Figures 15, 16, and 17 depict the different possible scenarios: In scenario A, the agent selects the action "forward" and navigates successfully to the northern node. In scenario B, the agent chooses the action "right," but because no node exists to the right, the agent waits until the next cycle to try a new direction. In scenario C, the blue agent selects the action "left", but the left node is

occupied by the orange agent, so the blue agent requests another decision from the brain.

Rewards

Based on the goals, observations, and actions of the agent, we can design our rewards scheme accordingly. We will reward our agent for successfully navigating the store and satisfying the shopping goals. Likewise, we will punish the agent for reaching dead ends and colliding with other agents. The full reward scheme is provided in Table 3. The rewards values were selected arbitrarily based on the difficulty and frequently of a task. Selecting a valid direction is considered an easy and frequent task therefore we reward the agent with small value (0.2). On the other hand, finishing all the shopping list is assumed to be a considerably challenging task, so a big reward is handed for completing it.

Table 3. Agent Action Rewards Scheme

Action	Reward
Selecting a valid direction	0.2
Reaching a dead end	-0.1
Selecting an occupied node	-0.1
Finishing a shopping objective	1
Finishing the shopping list	10

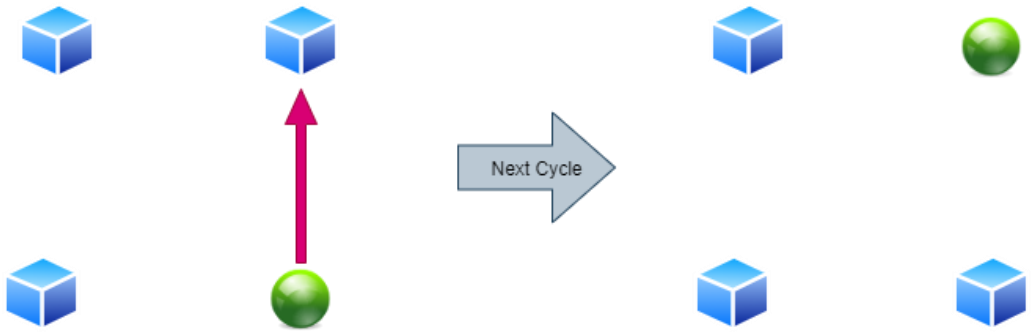

Figure 15. Agent Decisions, Scenario A

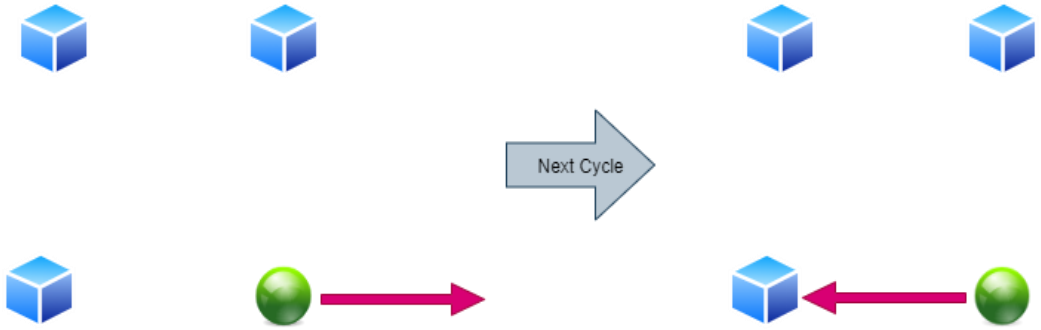

Figure 16. Agent Decisions, Scenario B

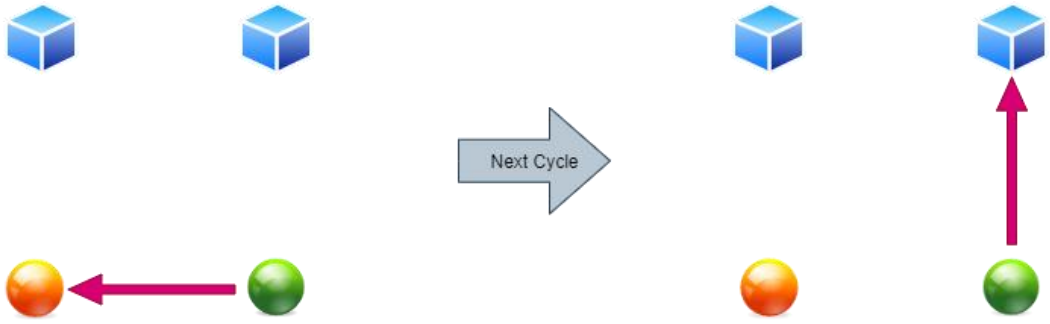

Figure 17. Agent Decisions, Scenario C

Results

To speed up the learning process, we deployed 10 agents to our learning environment (approximately 7% of the total number of agents 150 that will be used later in our final experiments) . For this research we allowed the agents to train for eight hours, after which we assessed the performance of the agent to be satisfactory. The final outputs of the training per agent per episode are highlighted in Figures 16 and 17. For both figures, the *x*-axis represents the number of elapsed training steps, where the peak is 800,000 steps (= 8 hours). Figure 18 shows the cumulative rewards per episode. The overall trend of the rewards represents a positive increase until exceeding 70 at the peak. Figure 19 displays the average episode length. In ML-Agents terminology, the episode length corresponds to the number of decisions (or steps) the agent took before completing its goal or ending the episode if the agent failed completely to learn. We can see that the episode length is constantly fluctuating at around 320 steps. This fluctuation can be attributed to the manner we are conducting the training. As explained previously, at the start of each episode, the agent is initialized with a random list of shopping nodes and asked to visit each node before stopping at a checkout queue. Consequently, when the generated shopping points are uniformly distributed across the store, we expect the agent to take a long time before reaching the target. Likewise, when condensed and closer nodes are chosen, the agent can finish its task quicker. Hence, the episode length oscillates considerably, which represents the randomness of our training scheme.

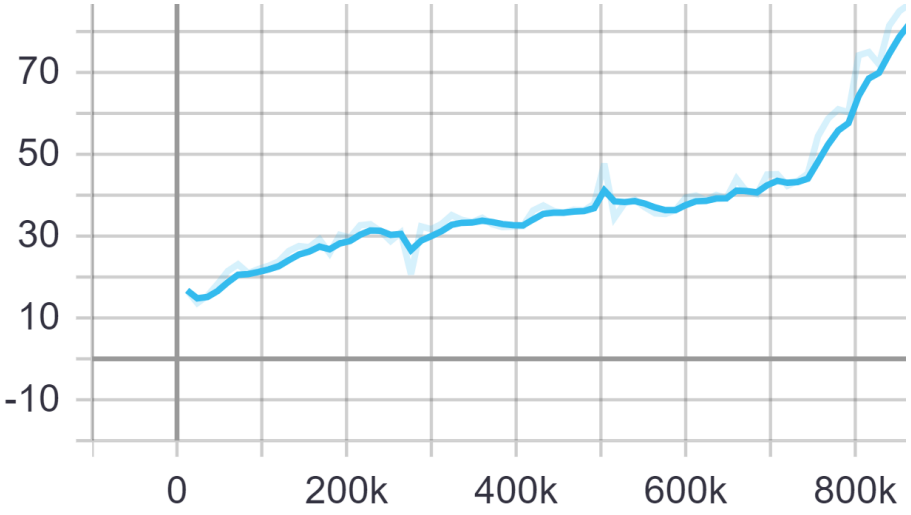

Figure 18. Cumulative Rewards Per Episode

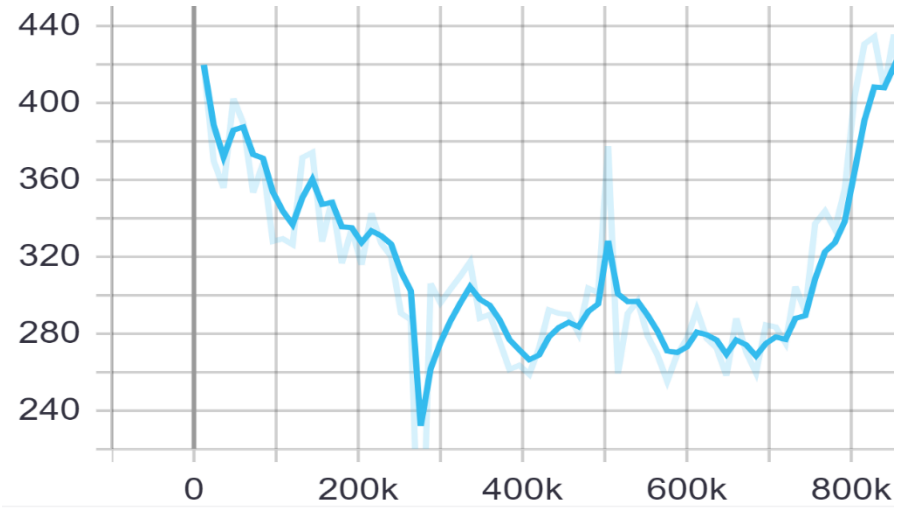

Figure 19. Training Episode Lengths

CHAPTER 6: RESULTS AND DISCUSSION

Model Validation

Validating the model was a challenging task due to the complex nature of COVID-19. Virus propagation is affected by a complex combination of parameters, many of which are still not well understood or are unknown. Additionally, the stochastic nature of our agent-based modeling further boosts the complexity of the problem.

We approached the problem from the angle of calibrating the parameters to produce simulations that approximate real-life statistics. The main criteria to maximize in the simulation are the simulation infection rate SIR%, expressed as follows:

$$SIR\% = \frac{E_c}{E_c + S_c + V_c} * 100 \qquad (12)$$

Where E_c, S_c, and V_c are the total number of new exposed agents, susceptible agents, and vaccinated agents, respectively. Each simulation corresponds to one real-life day of shopping; therefore, the SIR model is validated against the daily infection rate (DIR) of Qatar (see Methodology chapter for the calculations). We will compare our implementation against a subset of the dataset. We decided to select the first and last week of the dataset as comparison points. The first week represents the beginning of the pandemic in Qatar, so the number of cases is slowly growing. The last week represents a steady stage where the pandemic growth is relatively stable, largely owing to the vaccination campaign conducted by the Ministry of Public Health. The parameters and results vary considerably between the first and last weeks; therefore, we will be content if we can develop a model that satisfies the results of both weeks within certain acceptable criteria. For a more realistic implementation, we decided to create a simulation store that roughly approximates a real grocery store in terms of size and number of visitors. We based our virtual store on Qatar Family Food Center (FFC) [38]. The supermarket chain spans five stores totaling 220,000 feet^2; on average, each

store is 44,000 feet^2 or 4,088 m^2. Therefore, we designed our store area to fit a size of 72 × 56 (4,032 m^2). In terms of visitors, the FFC accepts 130,000 visitors weekly across all five stores. On average we expect each store to accept 150 visitors per hour per day (130,000 weekly = 18,571 daily = 3,714 visitors per store = 150 visitors per store per hour). Qatar has a policy regarding the maximum number of allowed visitors which is handled implicitly in our model by the variable T_a (maximum number of agents). A shortcoming of our work is that we are assuming a uniform number of visitors among different days and different hours of a day which is not very realistic assumption to take. However, we still claim that the final results of the simulation could approximate the actual results responsibly.

Error Measures

In evaluating our model, root mean squared error (RMSE), as given in the equation, is our main error measure:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Actual_i - Simulated_i)^2}{N}} \quad (13)$$

Where $Actual_i$ is the actual infection rate DIR% for day i and $Simulated_i$ is the simulated infection rate SIR for day i. We selected RMSE as our error indicator since it outputs results with the same measurement unit as our target variables (infection rate %); hence it is easier to interpret. For minimizing the error rate, our target criteria would be achieving a root mean squared error of ± 5%.

Calibrated Model Parameters

The list of parameters representing the base model is presented in Table 4. The base parameters are a mix of arbitrary assumptions and parameters taken from the literature. The main parameters we need to calibrate are IP_{zero} and IP_{max} (related to the infection probability between two individuals) and SLP and STP (related to the

infection probability through surfaces). The parameters T_a and V_a are independent variables that will be set individually at the start of each simulation.

Table 4. Model Base Parameters

Parameter	Base Value	Reason
Simulation area size - Area	72 × 56	FFC Average [38]
Simulation max. time - ST_{max}	36 min	Assumption
Total number of agents - T_a	150	FFC Average [38]
Total number of infected - I_a	Variable	
Total number of vaccinated - V_a	Variable	
Number of open counters - C_{open}	9	Assumption
Social distance adherence ratio - SD_r	80%	Assumption
Hygiene measures adherence - HM_r	80%	Assumption
Number of shopping points - C_{sp}	16	Assumption
Max. infection distance - D_{max}	1.5 meters	Recommended in [39]
Max Infection probability - IP_{zero}	0.04	Calibrated
Min Infection probability - IP_{max}	0.02	Calibrated
Angle flag - $Flag_a$	True	Assumption
Angle factor - AF	0.05	Assumption
Min. vaccine efficacy - VE_{min}	0.75	[26]
Max. vaccine efficacy - VE_{max}	0.91	[26]
Min. face protection efficacy - FP_{min}	0.45	[25]
Max. face protection efficacy - FP_{max}	0.97	[25]
Min. face viral reduction - FVR_{min}	0.25	[25]
Max. face viral reduction - FVR_{max}	0.25	[25]

Parameter	Base Value	Reason
Surface-spreading flag - $Flag_{ss}$	True	[27]
Surface-lingering probability - SLP	0.004	Calibrated
Surface transmission probability - STP	0.002	Calibrated

The simulation size area and the total number of agents T_a are based on the approximated values of family food center [38], as explained in the previous section. The number of checkout counters is set to 9, and the number of items in the shopping list is set to 16. The angle factor is set as a constant value of 0.05. The exposure distance is set to 1.5 meters based on the recommendation of the Ministry of Public Health [39].

In terms of safety measures, 80% of the population are assumed to respect social distancing and hygiene measures based on the regulations of the State of Qatar. Ideally, it should be 100%, but we set both at 80% to take into account people who do not respect those measures.

The factors related to the efficacy of vaccines (VE_{min}–VE_{max}) and face masks (FP_{min}–FP_{max} and FVR_{min}–FVR_{max}) are pulled from the literature [26], [25], respectively.

In order to calibrate the main parameters (IP_{zero} and IP_{max}), the simulation is configured with the base parameters and initialized with different values of infection probabilities until an acceptable result is found that closely fits the first and the last weeks of the real data (as explained in the previous section). The two parameters SLP and STP are set relative to IP_{zero} and IP_{max} such that they equal 10% of the two variables IP_{zero} and IP_{max}, respectively.

After running the simulation several times to arrive at the base values for IP_{zero} and IP_{max}, we simulated the spreading of the virus on each day of the selected two weeks. For each day, we varied the values of the parameters T_a and V_a to take into consideration the average number of infected and vaccinated agents in the past three days. Table 5 presents the results of our validation.

Table 5. Validation Results

Day	Week #	Real Value	Forecasted Value
2020-03-17	Week 1	1.7%	7.3%
2020-03-18	Week 1	3.1%	5.3%
2020-03-19	Week 1	1.4%	2.2%
2020-03-20	Week 1	2.2%	3.0%
2020-03-21	Week 1	2.5%	2.5%
2020-03-22	Week 1	$2.6\%_n$	3.1%
2020-03-23	Week 1	1.4%	2.6%
2021-09-04	Week 2	0.6%	0.2%
2021-09-05	Week 2	0.6%	0.3%
2021-09-06	Week 2	0.7%	0.3%
2021-09-07	Week 2	0.6%	0.5%
2021-09-08	Week 2	0.6%	0.5%
2021-09-09	Week 2	0.5%	0.2%
2021-09-10	Week 2	0.5%	0.6%
RMSE	**1.68**		

From Table 5, we can see that the RMSE equals 1.68, which is within our acceptable criteria. Figure 20 visualizes the predictions.

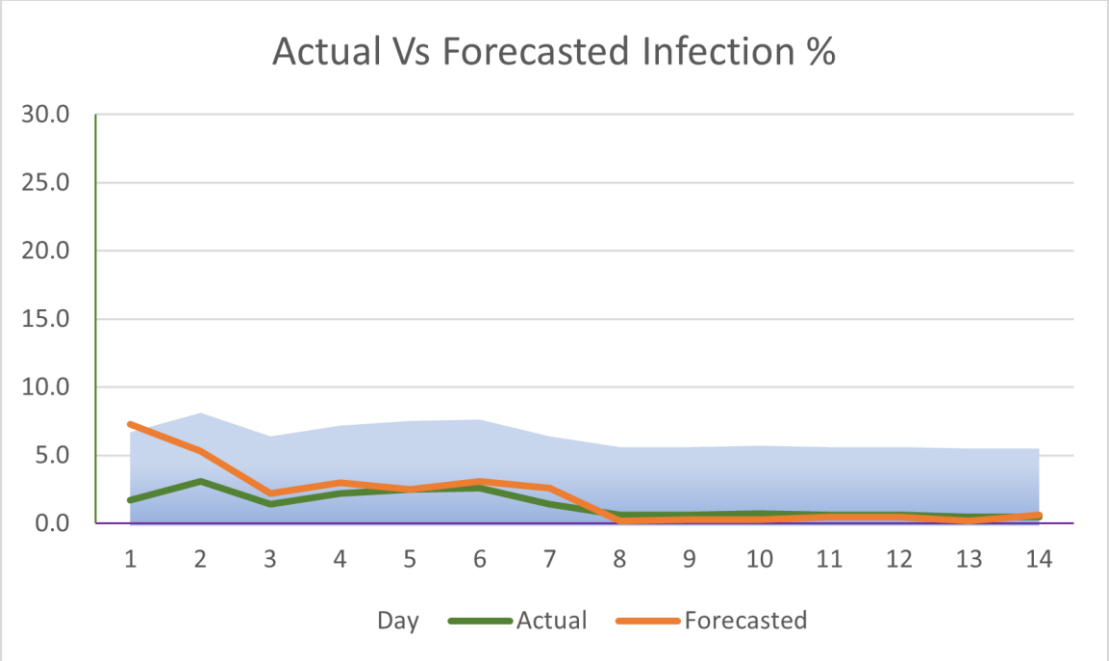

Figure 20. Validation Results

Sensitivity Analysis

After validating our model, we conducted a sensitivity analysis to assess the effect of each major parameter on the overall infection rate. Figure 21 is a tornado diagram demonstrating the sensitivity of 12 major parameters. In this experiment we assumed a base scenario of 100 people shopping in a store, in which 20 people are infected and 20 are vaccinated. Each shopper is allowed to pass by 16 shopping points and pay their bill at one of five operational checkout counters, and 50% of the shoppers are expected to follow social distancing and hygiene guidelines. The efficacies of the vaccines and mask protections are all set to 50%. The result of this configuration is an infection rate of 30%, as represented by the center of the tornado graph.

From this point onwards, we varied each of the 12 parameters to study the effect on the overall infection rate. The left side of the graph represents the minimum infection rate caused by one particular parameter, while the right-side value represents the maximum value. Going from green to pink represents a positive correlation with the infection rate, while going from pink to green represents a negative correlation.

Table 6 presents the ranges of values used in our sensitivity analysis.

Table 6. Sensitivy Analysis Parameter Ranges

Parameter	Min. Value	Max. Value	Base Value
T_a	50	200	100
I_a	10	40	20
V_a	10	40	20
C_{open}	1	9	5
C_{sp}	4	32	16
D_{max}	0.5	3.0	1.5
SDr	0%	100%	50%
HMr	0%	100%	50%
VE_{min}–VE_{max}	0.1	0.9	0.5
FP_{min}–FP_{max}	0.1	0.9	0.5
FLR_{min}–FLR_{max}	0.1	0.9	0.5

From Figure 21, we can observe the following interesting points:

1. The parameter that most affects the infection rate is the range of infection probabilities (IP_{zero}–IP_{max}). This is quite an intuitive result.

2. The numbers of infected and vaccinated people are crucial to the overall infection rate. Again, this is an expected result.
3. The number of open checkout counters has inverse correlation with the infection rate. Few open checkout counters lead to more congested queuing within small areas, which leads to more spreading of the virus. Interestingly, the effect of the number of open counters is asymmetrical: increasing the number of counters from one to five greatly reduces the infection rate, but further increasing the number of open checkouts has a negligible effect. One possible explanation for this is that a minimum of five checkouts is enough to satisfy a store containing 100 shoppers.
4. The number of shopping points seems to have the opposite relation with the number of open counters. When the shoppers visit a small number of shelves (C_{sp} <= 8), the infection rate is greatly reduced. However, the effect is greatly diminished after passing 16 shopping points. Our theory is that most of the infection occurs after some time within the simulation, when more people entered the store. When the size of the shopping list is small, people spend less time in the store; therefore, there is less chance of encountering each other.
5. The exposure distance has small but symmetrical inverse correlation with the infection rate. This is an expected result.
6. Increasing the percentage of social distance adherence and face mask adherence from 0% to 50% can greatly reduce the infection rate, as expected. Surprisingly, further increase of social distance adherence has less impact. For face masks, however, going from 50% to 100% adherence continues to greatly reduce the infection rate.

7. As expected, the efficacy of vaccines and masks have significant impact on the infection rate.

Figure 21. Sensitivity Analysis

Evaluation of Safety Measures

The final target of our data analysis is the effect of combining safety measures on the overall infection rate. In this section, we conducted two main experiments that will be presented in the next few subsections.

Experiment 1: Analysis of Social and Hygiene Measures

The first experiment analyzes the effect of social-distancing and hygiene measures on the overall infection rate. In this experiment, we use a store with 100 shoppers in which 25 shoppers are infected. The number of vaccinated is set to zero, as we need to focus on only the safety measures. We use different combinations of percentages for the values of social-distance and face mask adherence. We vary each

value of social-distance and face mask adherence by 0%, 25%, 50%, 75%, and 100%. The rest of the parameters are set according to the base values previously calibrated in the model validation section.

Figure 22 presents the results of our analysis, where the different colored bands represent different levels of infection severity: green, yellow, and red represent low, medium, and high severity, respectively. The *x*-axis represents social-distance adherence levels, and the five colored lines represent hygiene levels. From this figure, we can see that enforcing social-distance and face mask adherence of 50% will reduce the overall infection rate from 76% to 34% (> 50% reduction).

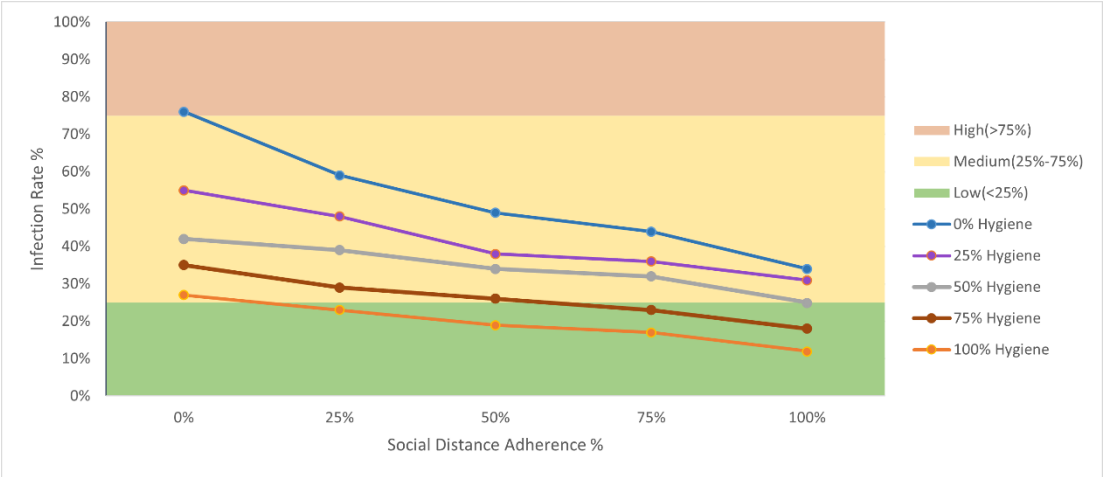

Figure 22. Safety Measures vs Infection Rate

Experiment 2: Infected vs Vaccinated

The second experiment studies the efficacy of the current vaccines against different COVID-19 variants. This experiment has two dependent main objectives. The first objective is to study the minimum number of vaccinations needed to avoid cases of high severity, where a 30% infection rate represents the maximum threshold for avoiding such danger zones when taking into consideration different infection levels.

The second objective of the experiment is to study the minimum percentage of vaccinations needed to reach a safe state (infection % <= 30%) when taking into consideration different COVID-19 variants.

In our research we assume the existence of four new COVID-19 variants. We represent these variants by altering the probabilities of infection (IP_{zero} and IP_{max}). Those variants are completely imaginary with arbitrary probabilities. The aim is to study vaccination strategies under hypothetical new mutations. Table 7 presents the probability values of the four different variants.

Table 7. COVID-19 Variants

Variant	IP_{zero}	IP_{max}
1	0.06	0.04
2	0.08	0.06
3	0.1	0.08
4	0.2	0.1

As in the first experiment, this experiment uses a store housing 100 shoppers. The number of infected people is varied within 25%, 50%, and 75%. The number of vaccinated is varied within 0%, 25%, 50%, and 75%.

Figures (23, 24, 25, and 26) present bubble charts representing the outcome of our experiments for each of the imaginary COVID-19 variants. In each of these graphs, the *x*-axis depicts the percentage of infected people, while the *y*-axis depicts the percentage of vaccinated people. In the graphs, green bubbles represent safe conditions

(infection % <= 30), while red bubbles represent danger zones. For the variants 1 to 3, we can reach a safe state if at least 75% of the population are vaccinated. For variant 4, our system predicts that the current vaccines could be ineffective in stabilizing the pandemic if the percentage of infected people reached 25%.

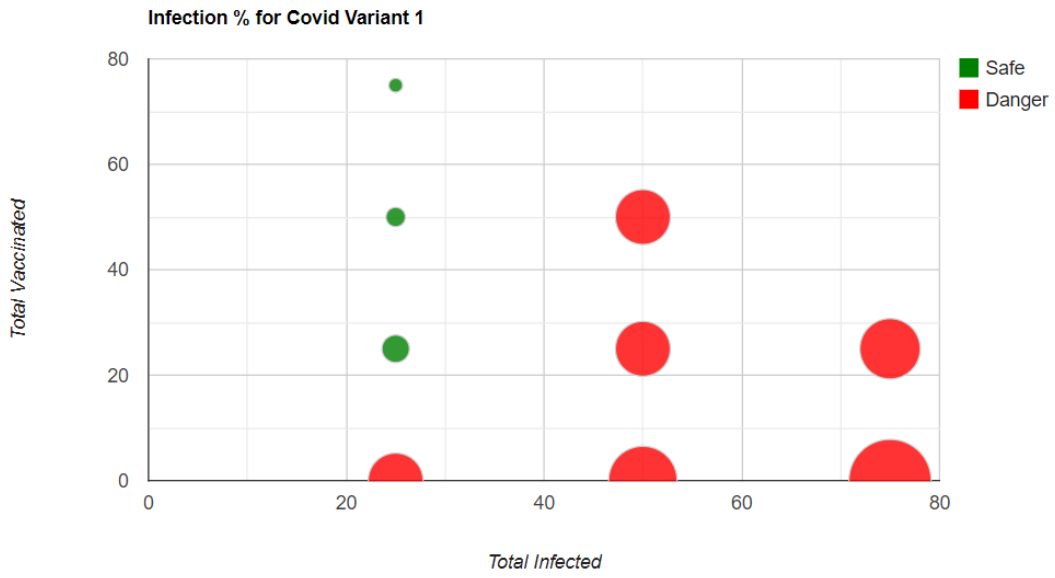

Figure 23. COVID-19 Variant 1 Results

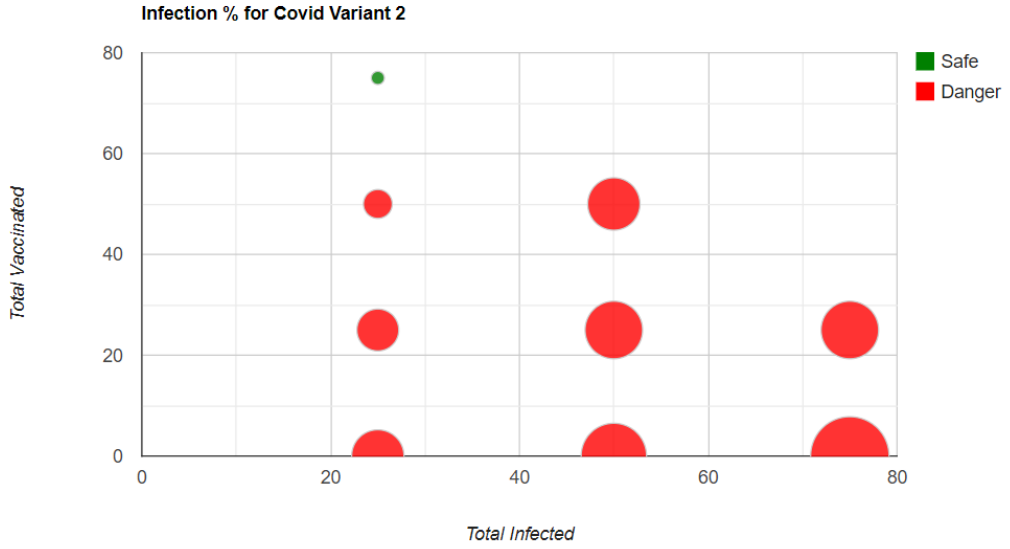

Figure 24. COVID-19 Variant 2 Results

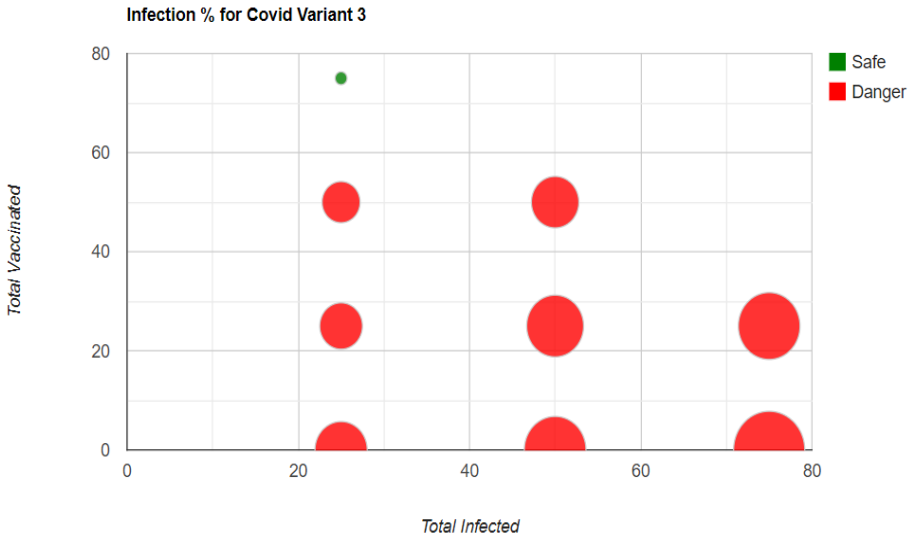

Figure 25. COVID-19 Variant 3 Results

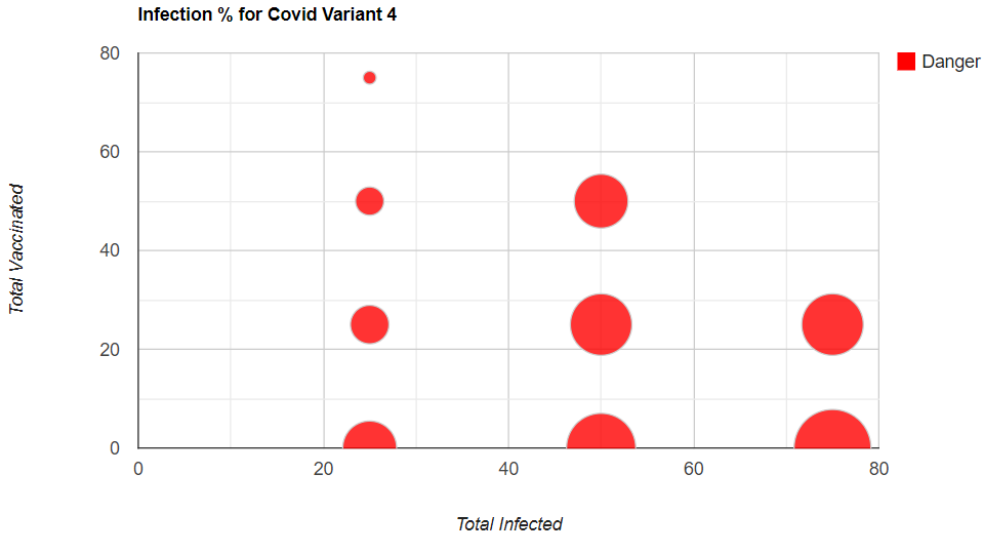

Figure 26. COVID-19 Variant 4 Results

CHAPTER 7: LIMITATIONS AND FUTURE WORKS

In this chapter we will discuss some of the limitations of our project and explore different directions to further extend the implementation.

Scientific Infection Model

The COVID-19 pandemic is complex and is manifested as a sophisticated combination of social and biological parameters. In this paper, we tackled the problem from a computing point of view; therefore, we had to make many assumptions in calibrating our model. Further model optimization using more scientific data analysis will lead to more accurate results.

Organic Movement System

The manner in which our agents navigate the store is limited to rectilinear movements because of the grid-of-nodes system we depend on for path finding. In real-life, however, shoppers move much more freely in a chaotic manner. One possible approach to implement more organic movement patterns is to integrate the NavMesh API [40], which presents functionalities for implementing more organic navigable zones and efficient path-finding methods.

Realistic Queuing System

The queuing component of our system is quite simple. First, it is limited in length to only four people, and second, the shoppers line up in a completely linear manner. In reality, the agents may queue in a more random way, especially when taking into consideration social distancing. Similarly, the distance between the shoppers in a queue is currently fixed, but in reality, the distance could vary depending on different store policies.

Varied Simulation Environments

The simulation is constrained to a small grocery store, but in reality, COVID-19 can spread rapidly in other closed spaces, such as schools, airports, and

transportation facilities. The solution can be extended to such simulation environments. Additionally, it would be informative to validate our infection model and agent movements against other environments and test whether the overall results are still satisfactory.

COVID-19 Variants and Vaccine Types

It would be educational to study the spread of different COVID-19 variants, as these variants require different model parameters. Similarly, it is crucial to compare the efficacy of the different vaccine types. Ultimately, we should study the effectiveness of the different vaccines against new virus variants to derive the best possible vaccination plans.

Simulation Visualization

Our project is currently limited to 3D models and animations. To improve visualization, we would like to import more realistic 3D models of human agents and use smoother animations. Similarly, it would be enlightening to integrate more realistic 3D environments—perhaps we could even mirror a real grocery store.

CHAPTER 8: CONCLUSION

In this research we offered an intelligent agent-based environment for simulating the spreading of COVID-19 among humans in the State of Qatar. We proposed the SEIP model as an extension of the SIR model to capture a more realistic picture of the pandemic. For more accurate estimations, the model was calibrated with a variation of novel parameters to better describe the phenomena of COVID-19. In terms of agent modeling, we followed a machine-learning approach and employed reinforcement-learning techniques to teach intelligent shopping behaviors and to infer optimal decisions. Smart shopping agents were developed using the ML-Agents toolkit, which is the Unity implementation of reinforcement-learning algorithms. The developed infection model and the smart agents were integrated using the Unity platform to design a 3D environment where the crowd behavior of shoppers and propagation of COVID-19 were controlled and visualized in real-time. We validated our model and parameters against real historical data from Qatar and derived different possible safety plans.

REFERENCES

[1] *COVID live update - Worldometer*. (n.d.). Retrieved October 3, 2021, from https://www.worldometers.info/coronavirus/

[2] Hackl, J., & Dubernet, T. (2020). Epidemic spreading in urban areas using agent-based transportation models. *Future Internet*.

[3] Maziarz, M., & Zach, M. (2020). Agent-based modelling for SARS-CoV-2 epidemic prediction and intervention assessment: A methodological appraisal. *Journal of Evaluation in Clinical Practice*, *26*(5), 1352-1360. doi:10.1111/jep.13459

[4] Silva, P., Batista, P., Lima, H., Alves, M., Guimarães, F., & Silva, R. (2020). COVID-ABS: An agent-based model of COVID-19 epidemic to simulate health and economic effects of social distancing interventions. *Chaos, Solitons & Fractals*, *139*, 110088. doi:10.1016/j.chaos.2020.110088

[5] Wang, Y., & Usher, J. (2005). Application of reinforcement learning for agent-based production scheduling. *Engineering Applications of Artificial Intelligence*, *18*(1), 73-82. doi:10.1016/j.engappai.2004.08.018

[6] Bobashev, G., Goedecke, D., Yu, F., & Epstein, J. (2007). A hybrid epidemic model combining the advantages of agent-based and equation-based approaches. *Proceedings of the 2007 Winter Simulation Conference*, 1532-1537.

[7] Bicher, M., & Popper, N. (2013). Agent-based derivation of the SIR-differential equations. *8Th EUROSIM Congress on Modelling and Simulation*, 306-311.

[8] Hassanat, A., Mnasri, S., Aseeri, M., Alhazmi, K., Cheikhrouhou, O., & Altarawneh, G. (2021). A simulation model for forecasting COVID-19

pandemic spread: analytical results based on the current Saudi COVID-19 data. *Sustainability*, *13*(9), 4888. doi:10.3390/su13094888

[9] Moein, S., Nickaeen, N., Roointan, A., Borhani, N., Heidary, Z., & Javanmard, S. (2021). Inefficiency of SIR models in forecasting COVID-19 epidemic: a case study of Isfahan. *Scientific Reports*, *11*(1). doi:10.1038/s41598-021-84055-6

[10] Chang, S. L., Harding, N., Zachreson, C., Cliff, O. M., & Prokopenko, M. (2020). Modelling transmission and control of the COVID-19 pandemic in Australia. arXiv preprint arXiv:2003.10218

[11] Yang, Z., Zeng, Z., Wang, K., Wong, S. S., Liang, W., Zanin, M., Liu, P., Cao, X., Gao, Z., Mai, Z., Liang, J., Liu, X., Li, S., Li, Y., Ye, F., Guan, W., Yang, Y., Li, F., Luo, S., Xie, Y., Liu, B., Wang, Z., Zhang, S., Wang, Y., Zhong, N., & He, J. (2020). Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions. *Journal of Thoracic Disease*, *12*(3), 165-174. doi:10.21037/jtd.2020.02.64

[12] Guo, Z., Xu, S., Tong, L., Dai, B., Liu, Y., & Xiao, D. (2020). An artificially simulated outbreak of a respiratory infectious disease. *BMC Public Health*, *20*(1). doi:10.1186/s12889-020-8243-6

[13] Yang, Y., Atkinson, P., & Ettema, D. (2011). Analysis of CDC social control measures using an agent-based simulation of an influenza epidemic in a city. *BMC Infectious Diseases*, *11*(1). doi:10.1186/1471-2334-11-199

[14] Chumachenko, D., Meniailov, I., Bazilevych, K., Kuznetsova, Y., & Chumachenko, T. (2019). Development of an intelligent agent-based model of the epidemic process of syphilis. *International Conference on Computer Science and Information Technologies CSIT 2019*.

[15] Wong, W., Feng, Z., & Thein, H. (2016). A parallel sliding region algorithm to make agent-based modeling possible for a large-scale simulation: modeling hepatitis C epidemics in Canada. *IEEE Journal of Biomedical and Health Informatics*, *20*(6), 1538-1544.

[16] Tabataba, F., Lewis, B., Hosseinipour, M., Tabataba, F., Venkatramanan, S., & Chen, J. (2017). Epidemic forecasting framework combining agent-based models and smart beam particle filtering. *2017 IEEE International Conference on Data Mining*.

[17] Nakamura, G., Souza, A., Souza, F., Bulcao-Neto, R., Martinez, A., & Macedo, A. (2020). Using symmetry to enhance the performance of agent-based epidemic models. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1-1. doi:10.1109/tcbb.2020.3018901

[18] Zhang, M. (2018). Modeling spatial environment for large-scale agent-based epidemic prediction and control. *The 3Rd IEEE International Conference on Cloud Computing and Big Data Analysis*, 537-541.

[19] Cuevas, E., (2020). An agent-based model to evaluate the COVID-19 transmission risks in facilities. *Computers in Biology and Medicine*.

[20] Kasaie, P., Dowdy, D., & Kelton, W. (2013). An agent-based simulation of a tuberculosis epidemic understanding the timing of transmission. *Proceedings of the 2013 Winter Simulation Conference*, 2227-2238.

[21] Zhang, M., Meng, R., & Verbraeck, A. (2015). Including public transportation into a large-scale agent-based model for epidemic prediction and control. *5 Society for Modeling & Simulation International (SCS)*.

[22] Bobashev, G., Goedecke, D., Yu, F., & Epstein, J. (2007). A hybrid epidemic model combining the advantages of agent-based and equation-based

approaches. *Proceedings of the 2007 Winter Simulation Conference*, 1532-1537.

[23] Perrin, D., & Ohsaki, H. (2011). A parallel approach to social network generation and agent-based epidemic simulation. *Proceedings of the Ninth Australasian Symposium on Parallel and Distributed Computing (Auspdc 2011), Perth, Australia*, 33-34.

[24] Unity Technologies. (2021). Unity. Retrieved October 6, 2021, from https://unity.com/

[25] Coclite, D., Napoletano, A., Gianola, S., del Monaco, A., D'Angelo, D., & Fauci, A. (2021). Face mask use in the community for reducing the spread of COVID-19: a systematic review. *Frontiers in Medicine, 7*. doi:10.3389/fmed.2020.594269

[26] Rosenberg, E. S., Holtgrave, D. R., Dorabawila, & V. (2021). New COVID-19 cases and hospitalizations among adults, by vaccination status — New York, May 3–July 25, 2021. *Morbidity and Mortality Weekly Report, 70*, 1306–1311. DOI

[27] van Doremalen, N., Bushmaker, T., Morris, D., Holbrook, M., Gamble, A., & Williamson, B. (2020). Aerosol and surface stability of SARS-CoV-2 as compared with SARS-CoV-1. *The New England Journal of Medicine*.

[28] Centers for Disease Control and Prevention. (2021). *Cases, data, and surveillance*. Retrieved October 10, 2021, from https://www.cdc.gov/coronavirus/2019-ncov/covid-data/investigations-discovery/hospitalization-death-by-age.html

[29] Coronavirus Disease 2019 (COVID-19) Statistics. (2021). Retrieved October 5, 2021, from https://www.data.gov.qa/explore/dataset/covid-19-cases-in-qatar/export/

[30] Monthly Figures on Total Population. (2021). Retrieved October 5, 2021, from https://www.psa.gov.qa/en/statistics1/StatisticsSite/pages/population.aspx

[31] Unity-Technologies/ml-agents: Unity Machine Learning Agents Toolkit. (2021). Retrieved October 6, 2021, from https://github.com/Unity-Technologies/ml-agents

[32] RStudio [Computer software]. (2021). Retrieved October 6, 2021, from https://www.rstudio.com/

[33] Unity Coronavirus Simulation. (n.d.). Retrieved October 6, 2021, from https:// https://github.com/Unity-Technologies/unitysimulation-coronavirus-example

[34] Unity-Technologies/unitysimulation-coronavirus-example. (2021). Retrieved October 6, 2021, from https://github.com/Unity-Technologies/unitysimulation-coronavirus-example/blob/master/docs/WaypointGraphAndMovement.md

[35] Unity Technologies. (2021). Unity - Scripting API: Physics.OverlapSphere. Retrieved October 6, 2021, from https://docs.unity3d.com/ScriptReference/Physics.OverlapSphere.html

[36] Unity-Technologies/ml-agents. (2021). Retrieved October 13, 2021, from https://github.com/Unity-Technologies/ml-agents/blob/main/docs/Training-Configuration-File.md

[37] Class RayPerceptionSensor | ML Agents | 1.0.8. (2021). Retrieved October 7, 2021, from https://docs.unity3d.com/Packages/com.unity.ml-agents@1.0/api/Unity.MLAgents.Sensors.RayPerceptionSensor.html

[38] Family Food Centre, Qatar. (2021). Retrieved October 7, 2021, from https://family.qa/ry/about

[39] Ministry of Public Health (2021). *Social Gatherings*. State of Qatar. Retrieved October 7, 2021, from https://covid19.moph.gov.qa/EN/Precautions-for-lifting-restrictions/Gatherings/Pages/Social-Gatherings.aspx

[40] Unity Technologies. (2021). Unity - Manual: Building a NavMesh. Retrieved October 9, 2021, from https://docs.unity3d.com/Manual/nav-BuildingNavMesh.html