1-1-2019

Emergency Evacuation Software Model For Simulation Of Physical Changes

Rachit Tomar
University of Windsor

EMERGENCY EVACUATION SOFTWARE MODEL FOR SIMULATION OF PHYSICAL CHANGES

By

Rachit Tomar

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2019

EMERGENCY EVACUATION SOFTWARE MODEL FOR SIMULATION OF PHYSICAL CHANGES

by

Rachit Tomar

APPROVED BY:

____

F. Ghrib
Department of Civil and Environmental Engineering

____

J. Lu
School of Computer Science

____

D. Wu, Co-Advisor

School of Computer Science

____

I. Ahmad, Co-Advisor

School of Computer Science

Dec 9th, 2019

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights. Any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Public space such as schools, cinemas, shopping malls, etc. must have an emergency evacuation system in place. Such places are also required to follow certain regulations and protocols for emergency evacuation to assure the safety of their occupants inside from any unpredictable incident. For nearly two decades, companies/organizations are using simulation models/software for evacuation planning. Researchers are working on these software models to improve the efficiency using latest algorithms. This thesis focuses on creating a base software model of evacuation systems for 3D indoor environments to simulate physical changes such as retractable chairs, movable walls etc., to evaluate their effectiveness before committing to those changes. This research tries to address various flaws and shortcomings of previous software. We are using tools like Unity 3D and Autodesk Maya to simulate suggested changes. It provides planners as well as researchers a new perspective to work on new recommended physical changes to design public venues.

DEDICATION

To my family, friends, my nephew Ojasv and my love…

ACKNOWLEDGMENTS

I owe a debt of gratitude to Dr. Wu and Dr. Ahmad, for the vision and foresight, which inspired me to conceive this thesis work. As my teacher and mentor, they have taught me more than I could ever give them credit for. I am also thankful to my thesis committee members, Dr. Lu, and Dr. Ghrib, for providing me extensive professional and personal guidance, which helped me learn a great deal about both scientific research and life in general.

Nobody has been more important to me in the pursuit of this thesis than my family members & colleagues. I would like to thank my parents; whose love and guidance are with me in whatever I pursue. They are the ultimate role models and the source of my inspiration. Most importantly, I wish to thank all the faculties and staff of the School of Computer Science and my friends who provided unending support and encouragement through my course work at the University of Windsor.

TABLE OF CONTENTS.

LIST OF TABLES

LIST OF ABBREVIATIONS/SYMBOLS

3D	3-dimensional
2D	2-dimensional
FDS	Fire Dynamics Simulator
CFD	Computational Fluid Dynamics
DXF	Drawing Exchange Format
DWG	Drawing Format
SFPE	The Society of Fire Protection Engineers
SGEM	Spatial-Grid Evacuation Model
CAD	Computer Aided Design
ACS	Agent Crowd Simulation
LED	Light Emitting Diode
EDS	Explosive Detection System
CO	Carbon Monoxide
DIFM	Dynamic Indoor Field Model
ABM	Agent Based Modelling
ABS	Agent Based Simulation
UC	User’s Choice
ID	Inter-Personal Distance
CA	Cellular Automata
MEL	Maya Embedded Language
API	Application Programming Interface
RVO	Reciprocal Velocity
CPU	Central Processing Unit
GB	Gigabyte

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

Every public space needs to be equipped with an efficient and functional system of emergency procedures/protocols, to ensure the safety of people who may get trapped inside in case of any unpredictable incident. As the phrase suggests, an unpredictable incident is the one that is difficult to predict in terms of its time of occurrence, form or means through which it may take place and in turn put human lives at risk. Therefore, there is a need to have a thorough and a well-tested evacuation system in place for every public space such as schools, cinemas, shopping malls, etc., cannot be understated. Evacuation systems are quintessential requirement, especially during such hazardous or fatal emergencies [1].

There are software models of evacuation systems simulating various kind of emergency scenarios in order to reduce number of casualties. If anyone wants to build a place that is safe, he/she needs to make sure that the hallways, width of doors, placement of exits or seating arrangement should be constructed in a way that no one gets hurt or injured during any emergency evacuation. It will reduce cost as well as time if the replica of the building/place designed in a computer works perfectly in a computer simulation before the actual construction starts. This replica of the building/place designed in a computer software is often called as 3D environment. Also, if a place is already built but needs some alteration in the environment, it should be tested on a computer simulated environment before committing to those changes. These changes in the physical environment such as changing width of doors, changes in seating arrangement or assembly of new set of chairs are called as physical changes. For example, if any civil engineer wants to increase the width of the hallways in a certain building by 1m, we should try to implement this first on a computer to check the effectiveness of this change, as process of increasing the width of the hallways on an actual building might cost a lot as compared to the changes in a computer simulated environment. However, most of the software models available work with the 3D environment that is predefined inside the software package. None of these software models provide a method/process to incorporate any possible physical changes in an indoor environment. Thus, in this thesis we develop a software model of evacuation systems for an indoor environment that can incorporate any possible physical change before they are actually constructed or built. Our software model helps in evaluation of the physical changes to check their effectiveness in case of an emergency for various scenarios based on incidents that happened in the past. Also, we can implement any possible physical changes in 3D indoor environment that might come in future using Autodesk maya and Unity game development engine.

Nevertheless, no one can predict incidents which can possibly happen anytime and anywhere. Thus, governments nowadays have strictly mandated all the indoor public places (i.e. schools, cinema halls, auditoriums, etc.) to have practice drills to avert harm in case of a fire. Also, they need to implement an emergency evacuation system to help everyone trapped inside to get out or at least protect themselves from any injury. Most reported incidents throughout the world are caused due to fire. Few of these incidents are presented in Chapter 2.

However, there are specific steps that are recommended in cases of a fire incident, which includes:

- Rescue
- Alert
- Alarm
- Extinguish
- Evacuate
- Gather at one point

Despite all these efforts, various fire incidents result in stampedes and human crushes. These incidents can be prevented by using some physical changes that we discuss in detail in subsequent chapters, such as:

- Retractable seats.

- Priority (Children and Physically challenged individuals).
- Extra emergency exits.
- Doors with panic bars.
- Extra staff for indoor environments.
- Glow in the dark paint.
- Movable walls.

Fire drills and other ‘mock evacuation exercises’ generally fail to accurately prepare for life-threatening actual evacuation scenarios since people tend to often ignore the danger or fail to access the situation all together [7]. Over the years many different software models have been developed to provide designers with ways of forecasting evacuation times for any type of indoor environment such as multi-storey buildings, concert venues, etc. A large number of software models for pedestrian simulation have been developed in many disciplines, such as robotics, computer graphics, evacuation dynamics, etc., that can be used to study various situations [8]. However, using computer graphics to simulate this general phenomenon is still a difficult job. Natural crowd is generally collective because a 'common purpose' or 'mutual destination' is shared by many individuals or animals [9]. These subconscious choices, however, are complicated to compute. However, a simulation generally costs substantial computational resources. Due to constraints such as computational costs, some early software models tend to concentrate on a crowd as one whole dynamic method rather than its movements.

In recent years, computer performance has exponentially enhanced. Current computers have the capacity to process crowd simulation programs and provide more advanced graphics and computation power. Thus, modern computers performance can support real-time simulations. Agent-based crowd simulation, a more advanced method for crowd simulation was first introduced as Helbing's model [13]. The principal idea of agent-based crowd simulation is to control the dynamics of crowd that is built autonomously. This model is considered as one of the best model’s to simulate crowd with realistic behaviors since it provides methods for controlling the movement of every agent individually. However, the cost of simulation is relatively high. Decreasing the computational cost while increasing the simulation performances such as the scalability, flexibility, and realism have remained in high demand in the area of agent-based crowd simulation.

For nearly two decades, companies are using simulation models/software for evacuation planning. Researchers are working on these software's to improve their efficiency using the latest algorithms. They are also working on building new software with various conditional parameters.

1.2 Thesis motivation

Even after application of various safety measures, failures in evacuations continue to occur. There are still hundreds of casualties due to stampedes and fires. In the year 2019 alone, 12 such events, consisting of fire and stampedes have been recorded around the world. We consider a wide range of software models that can provide evacuation timings as well as efficiencies. However, the existing software models have a major disadvantage of working with just the set of 3D environments available inside that software model. Furthermore, the possible physical changes that are not yet built or those changes that might come in the future, cannot be integrated in these software models. Hence, we anticipate reducing human casualties by a significant number by developing a software model incorporating any possible physical change inside an indoor environment before they are actually constructed/built or that might come in future, to simulate any real-world evacuation scenario.

1.3 Thesis objectives and contributions

The main objective behind this research is to create a base software model of evacuation systems for an indoor environment incorporating some possible physical changes before they are actually constructed or built using unity game development engine and Autodesk Maya. This base software model can be used for emergency planning and management as well as future research.

In order to reduce the time of evacuation and save life, companies work on physical changes on the respective indoor environment such as width of doors, seating arrangement, etc. Below are some of the features that our software model includes:

- We can import any file of .fbx format[1] incorporating a 3D indoor environment to our software model for simulation.
- The software model provides placement of animated agents such as men, women, children as well as wheelchair for physically challenged/slow moving agents.
- It also provides options to accommodate certain parameters that may be needed to run the simulation such as simulation start time, frame rate and running speed for agents.
- We can create customized parameters for any number of features to enable or disable them at specific times during the simulation for results and analysis.
- The software model also provides a navigation path to the nearest exit regardless of the number of exits available in the environment.
- We can simulate other features that provide a realistic simulation such as fire effects, smoke effects, exit blockage and counter-flow.

Other software models available to public incorporate few of these features and parameters. However, there is no single software model that can provide all of these features and parameters together. Apart from this, another disadvantage is that none of these software models provide a method/process to incorporate any possible physical changes in an indoor environment that might be added in future. Our software model helps in evaluation of the physical changes to check their effectiveness in case of an emergency for various scenarios based on incidents that happened in the past.

Thus, our software model can simulate any 3D indoor environment with any possible physical change, any number of exits and any number of seating. It provides an edge to engineers involved in planning and designing a new prototype of physical changes. Our software model provides a unique perspective to simulate a real-world evacuation scenario incorporating possible physical changes that can come in the future.

In this thesis we demonstrate a process to simulate certain number of features/physical changes that are new or missing from the existing software models that are used throughout the world. Examples for such changes include accommodation for physically

[1] .FBX file format is a 3D asset exchange format widely used by Autodesk Maya and other third party software such as AutoCAD, Unity 3D, etc.

challenged/slow moving agents, crowd distribution, counter-flow, simulation of any type of building, retractable seats, movable walls, extra staff and glow in the dark paint. These features are further described in detail in Chapter 3 and Chapter 4.

1.4 Thesis organization

The rest of this thesis is organized as follows: In Chapter 2, we introduce and review several typical crowd simulation models. In Chapter 3, we provide details about our proposed software model. Chapter 4 provides experimental details and the evaluation of our proposed methods. At last, we present our conclusions in Chapter 5.

CHAPTER 2

RELATED WORKS

In this section, first we look at some of the incidents that happened in the past in section 2.1. After that, we provide a critical review of some of the most pertinent related works.

2.1 Fire Incidents

Most reported incidents throughout the world are mainly due to fire. These incidents mostly happened as the safety policies were not followed carefully while constructing these indoor places. Below are some of the examples of these unfortunate incidents:

- The **1994 Karamay fire** (literally Karamay Big Fire) is considered to be one of the gravest civilian fires in the People's Republic of China. A fire broke out in a theatre hosting 1,000 children and teachers in Karamay on December 8, 1994. Students and teachers were asked to remain seated during the fire incident to allow Communist Party officials to walk out first. Furthermore, there were not enough doors for the crowd to move out. This resulted in deaths of 325 people, including 288 school children [2].
- The **Kiss nightclub fire** took place in Santa Maria, Rio Grande do Soul resulted in deaths of 242 people and leaving at least 630 others injured. This incident happened on January 27, 2013. Reasons for the high death toll include the shortage of emergency exits, and the front door was the only way out of the building. Additionally, the number of individuals inside exceeded the maximum capacity by hundreds [3].
  Approximately 90 percent of the victims died of choking due to excessive smoke. Many individuals died as they attempted to hide in the toilets out of panic. At least 180 corpses were removed from the toilets. The crush at the front gate and the rapid accumulation of smoke in the nightclub wounded more than 150 people.
- The **Station nightclub fire** in Rhode Island, United States, occurred on February 20, 2003, killed 100 people and injured 230. Although there were four possible exits, most of the people headed for the front exit through which they had entered. The ensuing

stampede led to a crush on the narrow hallway leading to that exit which quickly blocked the exit entirely and resulted in numerous deaths and injuries among the patrons and staff [4].

- The **Fire of Uphaar Cinema**, one of the worst fire tragedies in recent Indian history, took place at the Uphaar Cinema, Green Park, Delhi, on June 13, 1997. Fifty-nine individuals died by being trapped inside, mostly due to suffocation, and in the ensuing stampede 103 were severely wounded [5].
- **Garment factories of Karachi and Lahore** caught fire on September 11, 2012, killing 289 people and seriously injuring more than 600. When the blaze erupted, there were 1500 employees inside both the plants (Karachi and Lahore). Officials said all of the factory's exit gates were locked and many of the windows were covered with iron bars. It made hard for the employees to flee at the moment of the fire, killing many of them as a result of suffocation [6].

In order to reduce the number of casualties similar to the incidents discussed, many commercial entities or individuals around the world are using simulation models, to study and analyze evacuation timings and efficiencies during emergency. These software models provide a wide variety of features to simulate the required effects during emergencies and can be classified on the basis of techniques as: (1) flow-based models, (2) agent-based models, (3) activity-based models.

(1) **Flow-based models:**

Flow-based models focus on the entire crowd as a single unit, rather than on its components. This model is primarily used to estimate the movement flow of a large and dense crowd in a given environment [51].

(2) **Agent-based models:**

Agent-based modeling rests on the individual active component of a system. Active identities, known as agents, are recognized with agent-based modeling, and their behavior is defined. They may be individuals, homes, vehicles, machinery, products, or anything relevant to the system [14]. Agent-based models are the computer models that captures the behavior of individuals/agents in a certain environment. The most popular example of this technique is SIMCity computer game in which people or other entities interact with each other and/or their environment. This thesis mainly focuses on implementing a software model for evacuation using agent-based modelling technique, in which scenarios consist of different identities depicting several behavior.

(3) **Activity-based models:**

Activity-based models are based on people's daily pattern of activities. Activity based modeling predicts when, where, for how long, for and with whom each activity will be performed [51].

As we discussed, software models are classified on the basis of techniques. Also, these software models can be classified on the basis of availability for public use. For this thesis we consider 14 software models out of 28 available models, which we discuss in detail in section 2.2.

2.2 Related Software Models

In this section, we discuss various commercial and freely available software models for crowd simulation and the features/parameters that these models incorporate. These software models have various benefits as well as drawbacks. However, the major drawback of all these software models is that they can only work with the set of 3D environments available inside the software model. Also, none of these software models provide a method/process to incorporate any possible physical changes in an indoor environment before they are actually constructed/built or that might come in future.

EXODUS:

EXODUS is a freely available public domain software model in which the agents possess a fixed degree of familiarity with the building, agility, and patience. The model simulates the emergency situation of a large crowd from an enclosure and also accounts for the delay of movement or eventual cessation due to extreme heat or effect of toxic gases. The general model has been developed into different versions that vary according to several different contexts in which evacuations may occur, including ships (maritimeEXODUS), planes (airEXODUS), and buildings (buildingEXODUS) [15].

Since EXODUS is mainly an agent-based model, the movement of individuals in it is established by a fixed set of motion rules. The model as a whole is comprised of five interacting sub models: movement, behavior, passenger (agent), hazard, and toxicity. For instance, the hazard model generates values that correspond to a particular configuration of threat across the simulated environment. The toxicity model determines levels of exposure to toxic substances, which then affects the values of the variables associated with agent behavior, which in turn influences the calculations of the movement model. EXODUS is one of a group of models that have accumulated an impressive constellation of factors that inform more realistic evacuation scenarios. Even after so many sophistications, EXODUS doesn't provide simulation for physically challenged/slow-moving agents, which creates a significant problem in today's world [16].

WAYOUT:

WAYOUT is a freely available public domain program that has been developed to calculate traffic flow of building in emergencies from a multi-room and, possibly, multi-storey building. Only merging traffic flows are considered. The model considers several non-linear interactions of flows from different rooms, and it is based on movement speed - density dependence. Flows of only healthy people are considered. Psychological factors have not been included. Program WAYOUT considers up to 200 rooms connected in a merging network towards a single exit. The restriction in the room numbers has been introduced arbitrarily, to limit computer memory requirements, and can be extended if necessary, thus limiting the model to a merging network. This changes the problem from a probabilistic one to a deterministic one. WAYOUT considers its flows from door to door [17].

Program WAYOUT was used in Australia for several years and was applied in practical engineering situations to such cases as multi-storey buildings, stadiums, and malls, as well as miscellaneous more straightforward cases.

PEDROUTE:

PEDROUTE is a freely available public domain software model which uses a dynamic assignment method to simulate passenger movements through a station. The model also simulates station queue accumulation and its decay, and the blocking back effects of queues on adjacent regions. The model includes various methods of passenger assignment that can be used to simulate distinct situations of behavioral and physical layout. The techniques range from user-defined fixed paths to dynamic assignments using equilibrium or stochastic assignment algorithms. The model's route decisions consider the passenger's origins and destinations, available to him in the station, typically road entrances and platforms. The user defines these locations, as are the linkages between these locations. PEDROUTE takes into account walking speeds and delays for travelers during the process of simulating passenger demands through the station and summarizes the location of people during user-defined periods. The flow from one region to another relies on the number of individuals in that region and how long it takes to move through it. The model assumes that the time taken is a function of the density of passengers in that area [18]. PEDROUTE allows the user to look at levels of service and service factors for each area of the station layout, as well as the number of passengers present in each area during a given time.

SIMULEX:

SIMULEX Version 2.0 evacuation simulation program which is a freely available public domain that helps in analyzing the movement of groups. It provides a particular set of parameters to each "person" so that the walking speed is evaluated regardless of the average density of a group in a defined area. The model allows every person to choose their walking speeds. Beyond this improvement, the program also takes several other factors into account, which are included in derivation of motion algorithms, such as physical motions and gestures like body swaying and twisting, the proximity of other evacuees, shape of the building, structure and age, defined parameters for persons of 12-55 years of age, that are said to have social significance but that are not based upon ideas or information about social relations, culture, or group integration. Instead, the program presupposes the presence of a rational agent able to assess the optimal escape route and the agent's ability to avoid physical obstruction and overtake other persons that are conceptualized as impediments to movement [19].

FDS+EVAC:

FDS+Evac is a freely available public domain software model that simulates fire and evacuation processes simultaneously. It can also be used to only simulate the process of human egress without any fire impacts, such as a fire drill. FDS+Evac treats each evacuee as a distinct entity or 'agent' having their personal properties and methods of escape. The agents' movement is simulated using two-dimensional planes that represent building floors. The fundamental algorithm behind the egress movement resolves an equation of motion in a continuous 2D space and time for each agent, i.e., FDS+Evac performs some artificial molecular dynamics for the agents [20]. Fire Dynamics Simulator (FDS), the underlying fire simulation program, is a computational fluid dynamics (CFD) model of fire-driven fluid flow. FDS is published in the programming language of Fortran 90. It is written in C / OpenGL programming languages, along with the use of a companion program Smokeview, which represents the simulation outcomes graphically [21].

SIMWALK:

SimWalk is a freely available public domain pedestrian simulation software developed by Savannah Simulations AG, Switzerland. This software can be used for standard evacuation simulation of crowds, but the simulation of normal situations has limitations. It is only possible to define a start area, waypoint (middle point), and an exit point for standard mode simulations. SimWalk can model relatively large crowds and is used by several engineering and research institutes. Simwalk provides a SimDraw utility which can import AutoCAD designs, create and modify new environment designs. The software provides little difference between agents (movement delay and speed). Pedestrians use static paths, and therefore, it can be used for simulating the movements inside the buildings. However, this software cannot be used for a movement like Tawaf. Simwalk's underlying model is based on a social forces model. Movement trails, loads in each area, counters, the flow of information, level of service, speed and time taken by individual pedestrians to reach the destination, record and playback are some of the outputs of this software [22].

LEGION:

Legion is a freely available public domain software that is developed by Legion International Ltd., UK. This software has gone through years of studies on crowd movement and behavior. The software can simulate hundreds of pedestrians on a single PC, and it can be used for both evacuation scenarios and standard non-emergency studies. It can specify entrance, exits, and routes of movements. In Legion, individual pedestrians decide, based on their objectives and may possess a different level of knowledge, experience, and objectives. They also have a perception of their surrounding environment. This model uses a continuous "least effort path" algorithm for the simulation of movements. Throughout data, local density levels, congested areas, time taken for individuals to the destination, raw movement data output, video, and pictures are among the outputs that the software can provide [23].

MASSMOTION:

MassMotion is a freely available public domain simulation software model that is developed to enable design and planning professionals to rapidly test and analyze the movement of people in many kinds of environments. It does by providing users with a suite of tools for creating and modifying 3D environments, defining operational scenarios, executing dynamic simulations, and developing robust analyses [24].

MassMotion models real-world spaces by breaking those spaces down into parts and classifying the parts according to function. People in a MassMotion simulation know to walk around an obstruction because it has been marked as a barrier. Speed of movement is reduced when walking up a surface because that surface has been marked as a stair. The way in which classified objects are arranged can have a significant impact on how people navigate a space, affecting their speed, their movement patterns, and their route choices. The essential elements of a scene are the floor, link, stair, ramp, escalator, path, portal, and barrier [25].

Every person in a MassMotion simulation is an autonomous agent. Each agent has the ability to monitor and react to its environment according to a unique set of characteristics and goals. Agents are created and placed in the scene using events [25].

PATHFINDER:

Pathfinder is a commercial simulator for emergency egress with an embedded user interface and animated 3D results. It enables faster evaluation of evacuation models and more realistic graphics to be produced than with other simulators. Pathfinder supports the import of DXF and DWG files in AutoCAD format. The Pathfinder's floor extraction tool enables the imported geometry to be used rapidly to define the occupant walking space for the mode of evacuation. To represent the model geometry, Pathfinder utilizes a triangulated 3D mesh. As a consequence, Pathfinder can represent geometric information and curves accurately [26].

Pathfinder supports two simulation modes. In steering mode, agents, while avoiding other occupants and barriers, continue independently to their goal. Door flow rates are not defined but are the consequence of occupants' interaction with each other and with borders. Agents use behaviors that follow SFPE rules, in SFPE mode, with walking speeds depending on density and flow limits to doors. SFPE outcomes provide a crucial baseline for comparison with other outcomes, but SFPE calculations do not prevent multiple persons occupying the same space [27].

ALLSAFE:

AllSafe is a commercial software model which determines whether occupants are at risk depending on input data for the building, the building use, the occupants, and the design fire scenario. It has a global perspective. The model assigns behavioral characteristics to groups of occupants. The occupants have a global building perspective because they can only travel to one exit. The building is input into the model via a series of nodes. Attempts have been made to validate it with other models, such as Simulex. The model was designed to calculate the evacuation times when occupants are not aware of a fire until later in the scenario. The main calculation includes time delays prior to evacuation and estimating occupants walking time [28].

CRISP:

CRISP is a commercial model of allfire scenarios from Monte Carlo. The sub-models of CRISP represent physical objects such as rooms, doors, windows, alarms and detectors, furniture products, etc., layers of warm smoke, and individuals. The randomized elements include starting circumstances such as various open or closed windows and doors, number, type, and location of individuals within the building, fireplace, and type of burning product. CRISP's fundamental design is a two-layer zone model of smoke flow for various rooms combined with a comprehensive model of human behavior and movement. The Monte Carlo controller supervises all the physical objects, which makes each one performs for each timestep. The controller also manages all the input and output, initialization for each run, and each run starts automatically. Functions are included from any distribution to produce random numbers. Iteratively, calculations are performed for each run, with variable time intervals to ensure the effectiveness, precision, and stability of the program.

People are assumed to adopt distinct behavioral roles, either naturally or due to training. Depending on the state of the environment, their conduct can be defined in terms of actions that can be abandoned and substituted by fresh ones. Rational decisions are made based on current knowledge (which may be limited and/or incorrect) [29,30].

EGRESS2002:

EGRESS was a commercial software model, intended as an easy and efficient way to enhance our knowledge of the evacuation arrangements of a particular structure. It was intended as a generic simulation tool that offers the necessary facilities for efficiently modeling a wide variety of scenarios, from theaters to office buildings to railway stations to boats. EGRESS is especially useful for simulating evacuation from buildings and facilities where there is no data on evacuation drills. This involves new houses in the design phase, or buildings were owing to the nature of the activities or a large number of individuals engaged. In an EGRESS simulation, a structure's floor area is covered by cells equivalent to the minimum area which an individual would occupy. The EGRESS simulations advance by moving individuals (automata) on a hexagonal grid from cell to cell according to a weighted die throw. The method is intrinsic to the use of cells with a dimension equivalent to the minimum region that an individual occupies [31].

EGRESS has been intended as a simple and effective means of enhancing our understanding of and improving evacuation arrangements. It is ideally suited to modeling evacuation from a large office complex, shopping mall, underground station, airport terminal, or other complex building containing large numbers of people. It offers useful data for building designers, owners, occupiers, and emergency services, in simulating the evacuation process [32].

SGEM:

SGEM is a commercial model that takes the spatial geometry of a building (or set inside a building) and divides the setting into planar grids. A relative coordinate represents each notional occupant in the building, and the individual's movement pattern in the setting is modeled by solving the representing difference equations. Inter-Person influence and behavior can be taken into consideration. As the spatial layout of the building is represented by a grid system, the model has termed a spatial-grid evacuation model (SGEM) [33].

People in a fire escape from inner rooms, doors, in-corridor, door-halls, exit to a safe place step by step. Similar to other models, they define a building enclosure into a network system (nodes/arcs) in accordance with the people's possible direction of evacuation or the optimizing route of escape. In the model, each node (zone) may represent a room, a corridor, or a part of a hall. The nodes are connected to their neighbors by way of openings such as doors, exits, etc. The characteristic of each node can be described by recording the coordination of the setting. The flow direction for the nodes is also decided with respect to the positions of the final exit points. The choice of exit within a node is determined by various environmental stimuli, such as the distance to the exit, the presence of exit signs, visual accessibility, etc., and personal characteristics such as familiarity of routes, etc. In each node, the people are described as individuals located on grids that represent the buildings (or setting). It is considered that the behavior of each individual can be represented by his/ her spatial location and movement velocity vector. The velocity vector of each individual is a function of various environmental influences within the building, such as the influence of other peoples, the geometry of the paths, the behavioral reaction, etc. The spatial location of each individual can be represented by his/ her relative coordinates in the system. The records of the coordinates at various points of the time describe the movement trajectories of the people [34].

The SGEM can be used to predict the movement trajectory of each individual during evacuation. The basic principle is that it takes the geometry of a building and divides the setting into planar grids [34].

EXIT89:

EXIT89 is a commercial model that has shortcomings such as social interaction and emergent group response similar to SIMULEX. EXIT89 includes individual bodily dimensions (American, Soviet, or Austrian) and allows the specification of the number of physically challenged occupants, yet it does not incorporate bodily actions and gestures [35]. It also considers the counter-response of evacuees whose path during egress is blocked by smoke accumulation near an exit. The model determines travel time as a function of density and speed within a constructed network of nodes and arcs. The "shortest route" algorithm is combined with an individual perspective for each evacuee to track the path and progress of individual evacuees. However, all the occupants of a specific node initially traverse the same user-specified path, or shortest known path, to an exit. Moreover, the user is also able to set the percentage of occupants who are assigned a delay time. These dual functions (a particular path for an entire group and delay) implicitly mimic group behavior [36].

A significant drawback persists is that each individual evacuee exhibits particular physical characteristics which affect the flow of evacuation; however, these characteristics are devoid of interactive social characteristics such as monitoring others, directing, collective evaluation and collective agreement on the appropriate response [36].

Every software model discussed has its benefits as well as drawbacks on crowd simulation. However, all these software models have a major disadvantage of working with just the set of 3D environments available inside that software model. Furthermore, any possible physical changes that are not yet built or those changes that might come in the future, cannot be integrated in these software models. Also, external files cannot be imported to these software models to add a new 3D environment for simulation. We consider a set of parameters/features for thorough comparison of these software models. We discuss these parameters/features in section 2.2.

2.3 Parameter list for software model comparison

The parameters/features that we consider for comparison can be classified into two: main features and special features.

The main features for the model are the mandatory parameters that can define the working of a simulation. Below are the main features that we consider for comparison of software models:

- Availability to the public
- Visualization of the model
- Purpose
- Movement of agents
- Incorporation of fire data
- Computer-aided design (CAD)

The special features for the model are the optional parameters that the model can or cannot have. However, these features increase the level of scenarios or possibilities in a specific model.

Below are the special features that we consider for comparison of software models:

- Counter-flow
- Exit blocked
- Toxicity of the agents
- Disabled/slow-moving agents
- Effect of fire conditions on agents

We discuss these features in the chapter 3 during the comparison of these software models. Apart from these software models for crowd simulation, we also consider a number of research papers from various authors working in the field of agent-based modelling for crowd simulation. We discuss some of these works in section 2.3. All these research papers contain methods to evaluate the time results and efficiencies of evacuation.

2.4 Crowd Simulation : Literature Review

Several authors have proposed models for agent-based crowd simulation. Helbling et al. 's Social force model [13] settles the fundamental idea about how to apply the agent-based approach to a crowd simulation. In this model, authors apply the repulsion and tangential forces on each agent; therefore, the steering agent can avoid obstacles and other moving agents. However, the main downside of this method is that every agent shares the same attributes without individual identity; also, agents are navigated under the same moving speed unlike a real human crowd, which has individual differences regarding mobility. Besides, when Helbling's moving method is applied on a 3D model, the agents tend to 'shake' and 'vibrate' unpredictably with increasing density and non-priority rule applied. Overall, Helbing's model has introduced the fundamental conception of the agent-crowd simulation (ACS) study, and his research inspired us enormously.

A. Zavin, et al. [37] proposed an intelligent automated fire exit guidance system using A* search algorithm, where along with guiding the affected people through the safest optimal path, the system calculates least crowded and shortest path considering distance, endangered node, and crowd distribution mechanism. It also detects if any path is already compromised with fire and dynamically suggests the second optimal path to the fire exit. In the proposed system, a portion of a floor plan is given and represented by a unidirectional weighted graph. The weighted graph is represented as a grid where each grid cell is a square of unit cost. In the grid representation, some grid cells are considered as fire exits, and some are walls, whereas others are considered as rooms/hallways. Experimental results showed the proposed system executed accurately and effectively by handling distance and safety-related challenges to ensure minimum life and resource causalities.

However, there are some limitations to the proposed system, rather than designing any prototype; only a software simulation has been presented by creating a sample floor plan of a building. Some real-world factors are not been taken under consideration while guiding for the fire exit such as, electricity backup system, panic, and instant psychological state among fire-affected people, which might tamper the environment as well. Thus, as a future extension of this paper, a prototype developed based on the proposed system where the real-world factors are considered and implemented. Multiple fire exits, backup power supply, LED floor tiling system to show optimal paths, etc. are required to design such a fire-exit guidance system. Moreover, to implement the system for each floor of a building, sufficient sensors like smoke and temperature sensors, motion sensors and other facilities connected to central control.

S. Wong, et al. [38] proposed an idea to assist in the rapid evacuation. They presented an algorithm to calculate the optimum route for each local area with the idea to reduce congestion and maximize the number of evacuees arriving at exits in each time span. Their system considers crowd distribution, exit locations, and corridor widths when determining optimal routes. It also simulates crowd movements during route optimization. As a basis, they expect that neighboring crowds who take different evacuation routes should arrive at respective exits at nearly the same time. If this is not the case, their system brings up to date the routes or paths of the slower crowds. Since crowd simulation is non-linear, the optimum route is calculated iteratively. The system repeats until an optimum state is reached. The system permits the structure of the situation to be decomposed and determines the routes in a hierarchical way in relation to directly calculating optimum routes for a situation. Not only does this approach reduce computational costs, but it also allows crowds in distinct areas to evacuate with distinct priorities. This system can deal with different structures for the situation and obstacle positions in order to provide an optimal path. They used a mix and match strategy to provide results for various situation. They used various algorithms such as Djikstra's, A*, etc. on various environments such as a theme park, stadiums, etc. various crowd parameters are considered such as walking speed, the radius of an agent, etc. This system, however, lacks in 3d visualization and simulation of physically challenged or slower-moving agents.

A significant consideration is the emergency evacuation from airports, considering the ongoing occurrence of natural and human-caused disasters affecting these places. These incidents concentrated attention on the requirements of people with disabilities, who are more likely to suffer in cases of emergency. M. Manley, et al. [39] proposed the agent-based model that can be used by engineering in planning and designing efforts to estimate the evacuation performance of heterogeneous populations from airports. The model groups the environment according to accessibility characteristics, including various conditions that have a separate impact on the conduct of people with disabilities during an evacuation. The model architecture consists of four logical components: (a) environment, (b) population, (c) visualization, and (d) simulation. Also, an agent's direction choice is influenced by three essential forces: (a) attraction, (b) repulsion, and (c) friction (see Figure 1)

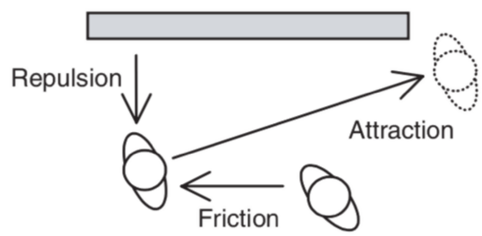


Figure 1: Essential forces for the agent's direction [39]

The objective of the simulation experiment is to estimate the impact of bomb placement on evacuation times for individuals with and without disabilities. The experiment is conducted using the map of an international airport from the intermountain west region of the United States. The airport complex is a two-story structure consisting of three terminals and five concourses patterned after the pier terminal design. Four simulation scenarios are specified based on the presence and location of the bomb. Scenario 1 is conducted without a bomb to establish a baseline for subsequent comparisons. Scenarios 2, 3, and 4 simulated a bomb discovery in one of the Explosive Detection System (EDS) devices found in Terminals 1, 2, and 3, respectively. Fifty simulations are conducted for each scenario.

The results of a simulation experiment with a densely populated airport demonstrated behavior consistent with expectations regarding individuals with and without disabilities. Additionally, several findings which inform both architectural engineering and emergency management practice were revealed, including (a) limitations of the pier airport design during emergency evacuations, (b) identification of the individuals most at risk or those with lower stamina, and (c) potential bottleneck areas at the stairways. Neither have they considered multiple fire exits or different kinds of agents such as children or overweight people.

Evacuation simulation in the airport domain needs additional characteristics beyond most simulations, including the distinctive behaviors of first-time visitors with incomplete area understanding and families who do not necessarily conform to frequently assumed pedestrian habits. Evacuation simulations that are not customized for the airport domain do not integrate the factors that are essential to it, resulting in inaccuracies when applied to it. J. Tsai, et al. [7] proposed ESCAPES - a simulation tool for multi-agent evacuation that includes four main features: (i) distinct kinds of agents; (ii) emotional interactions; (iii) informational interactions; (iv) behavioral interactions. Their simulator reproduces phenomena that have been observed during evacuation and the features that they integrate significantly impact escape time. They used ESCAPES to model the Los Angeles International Airport (LAX) International Terminal and obtain high commendation from safety authorities.

They discuss the multi-agent evacuation simulation system, ESCAPES, in two parts: individual agent types and agent interactions. ESCAPES include regular travelers, authority/security figures, and families, as these have been documented as having the most impact in an airport evacuation. ESCAPES agent interactions include three separate phenomena: the spread of knowledge, emotional contagion, and social comparison. The ESCAPES system is a two-part system consisting of a 2D, OpenGL environment based on the open-source project OpenSteer and a 3D visualization component using Massive Software. The scenario takes place in a generic airport setting consisting of 2 gates, 3 hallways, and 14 shops. The experiments for the scenario feature the following: 100 travelers which include 10 families, 10 airport staff. These entities exhibit the spread of knowledge and social comparison. ESCAPES was a tool that incorporates four key features: (i) different agent types; (ii) emotional interactions; (iii) informational interactions; (iv) behavioral interactions. These features are grounded in social psychology and evacuation research and tailored only towards the needs of an airport security official. ESCAPES can’t be used for multiple environments and large crowd.

Simulation of building evacuation provides designers with an effective manner to test a building's safety before construction. In a variety of fields (computer graphics, robotics, evacuation dynamics, etc.), a significant number of models are developed. N. Pelenchano, et al. [8] provide a review of models of crowd simulation and selected commercial software tools for simulation of elevated construction evacuation. The selected business tools (STEPS and EXODUS) are grid-based simulations that enable effective execution but introduce artifacts in the outcomes. They focus on describing the primary difficulties and limitation of these tools, in addition to explaining the importance of integrating human psychological and physiological factors into the models.

The basic model is to simulate generic flocking behavior consist of three simple rules which describe how individual boid maneuvers based on the positions and velocities of its nearby flockmates (see Figure 2):

- *Separation*: steering to prevent crowding local flockmates
- *Alignment*: steering towards the average heading of local flockmates
- *Cohesion*: steering towards the average position of local flockmates

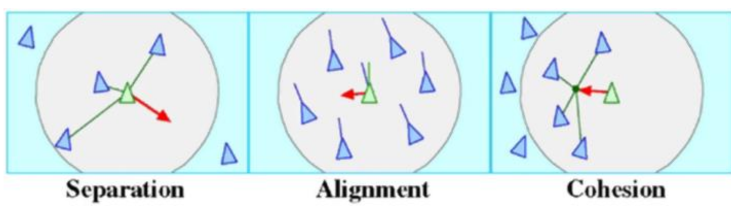


Figure 2: Generic flocking behaviors [8]

For the study, both software models (STEPS and EXODUS) have been fully tested and analyzed to determine their achievements but also their limitations. Models based on cellular automata lack realism because they are restricted to checkerboard configurations since the agents can only occupy discrete cells in a grid. Agents move based on some potential field calculated on the grid. Agents are therefore assumed to move within a static environment where the paths towards the exits do not change over time. Agents move from the current cell to the desired destination cell based on the occupancy of the space.

The paper concludes an overview of the basics that should be implemented to simulate human movement. In this simulation flow rates, densities and speeds of human is not a predefined value, however it changes dynamically for a limited grid size. One of the major drawbacks is that the system have not considered the factors resulting in crowd distribution, thus significantly reducing the escape time.

Evaluating the evacuation capacity of the metro station under emergencies such as fires, as well as improving the emergency protection capability of the rail transit system is an integral part of the rail transit operation management. On the basis of analysis of the impact of different factors on passenger's behavior under fire evacuation in metro station, the study by X. Zhang, et al. [40] models the facilities in metro station including gates, stairs, passageways, entrances and exits, and simulates the process of fire spread and the passenger evacuation scenarios by the model of cellular automata. A case study is brought out to illustrate the impact of different factors on evacuation capacity and evaluate the rationality of the station layout.

The study simulates a 2D visualization of the passenger evacuation process in a metro station, analyzes the main bottlenecks. It also stimulates the process of fire spread and the passenger evacuation scenarios by the model of cellular automata. Finally, they propose corresponding improvement measures. a) The model uses the concept of fire value, which represents a degree of danger during the process of fire evacuation simulation and in the path selection model of passengers, b) The simulation finds that the Cellular automata model has good practicality for evacuation process modeling of the underground building with a double-layer structure. Using computer simulations, the entire evacuation process of evacuees is visualized, and the distribution of evacuees at a different time is available so that the bottlenecks in a metro station which are not conducive to evacuation can be found out. The simulation result can provide a basis for a reconstruction of station facilities and layout design of the station.

In the last few years, agent-based modeling is used to simulate real-world business problems. In this paper by E. Bonabeau, et al. [41] provide fundamentals of ABM and benefits of it over other modeling methods. The benefits of ABM can be stated in the given three points: (i) ABM captures emergent phenomena which is resulted form the interaction of individual entities ; (ii) ABM describes a system with natural description which makes the model seem closer to reality; and (iii) ABM is versatile as it is easy to add more agents to it. It is evident, however, that what drives the other benefits is ABM's capacity to cope with emerging events.

Using real-world applications, its four application areas are addressed:

- *Flow simulation*: ABM seems ideally suited to provide valuable insights into the mechanisms of and preconditions for panic and jamming by incoordination. ABM captures that emerging phenomenon in a natural manner.
- *Organizational simulation*: It is feasible to model in a particular context or at a certain level of description the emerging collective conduct of an organization or part of an organization. At the very least, the simulation design process creates valuable qualitative insights. However, in some instances, semi-quantitative insights can also be generated. An excellent example of this is an agent-based model of operational risk.
- *Market simulation*: The dynamics of the stock market results from the conduct of many interacting agents, resulting in emergent phenomena that are best understood by using a bottom-up approach-ABM. To assess the impact of tick-size reduction, an agent-based model has been used by NASDAQ that simulates the effect of regulatory changes on the financial market under various circumstances. The model permits regulators to test and predict the results of different strategies, observe the conduct of agents in answer to changes, and monitor progress, providing warning of unintended consequences of newly executed regulations quicker than real-time and without risking early tests in the real marketplace.
- *Diffusion simulation*: ABM applies to cases where people are influenced by their social context; that is, what others around them do.

They had also discussed how ABM could bring significant benefits when applied to human systems. It is best to use ABM in the following contingencies:

- When agent interactions are complicated, nonlinear, discontinuous, or discrete (for instance, when other agents can dramatically, even discontinuously, alter an agent's conduct).

- When space is essential, and the positions of agents are not fixed. Example: theme park, supermarket, traffic, fire escape.
- When the population is heterogeneous, where each person is (potentially) different.
- When the interaction topology is heterogeneous and complicated. Example: when interactions are homogeneous and mix worldwide, there is no need for agent-based simulation, but social networks are rarely homogeneous, clustered, leading to deviations from normal behavior.
- When the agents display complex conduct, including learning and adaptation. Example: NASDAQ, ISPs.

ABM is not 100 percent effective at all times and has certain drawbacks. A general-purpose model cannot work thus the model must be constructed at the correct level of description, with just the correct amount of detail to serve its purpose. Another problem is the very characteristic of the systems one models with ABM in the social sciences: they most often require human agents, with potentially unreasonable behavior, subjective decisions, and complicated psychology— that is, soft factors, hard to quantify, calibrate, and sometimes justified.

The last practical problem in ABM persists that should not be ignored. By definition, ABM looks at a system at the level of its component units, not at the aggregate level. While computing power continues to rise at an impressive rate, ABM's high computational requirements persist as an issue in modeling large systems.

An agent-based system uses an autonomous interacting agent computational model in an environment to evaluate the group's emerging behavior. This paper by N. Wagnar, et al. [42] present a prototype of a computer simulation and decision support system that utilizes agent-based modeling to simulate crowd evacuation in a fire catastrophe and allows various disaster scenarios to be tested. The system aims to allow for multiple scenario testing and decision support for the planning and preparedness phase of emergency management with regards to fire disasters at concert venues. The system is designed for emergency managers, police, and any administrators who are charged with fire disaster mitigation planning for concert venues. Users of the system can benefit from evaluating the effects of potential safety measures.

The prototype is unique in the present literature as it is specifically intended to simulate a concert site setting such as a stadium or auditorium (see Figure 3) and is extremely configurable to allow user definition of concert venues with any arrangement of seats, routes, phases, exits and individuals.

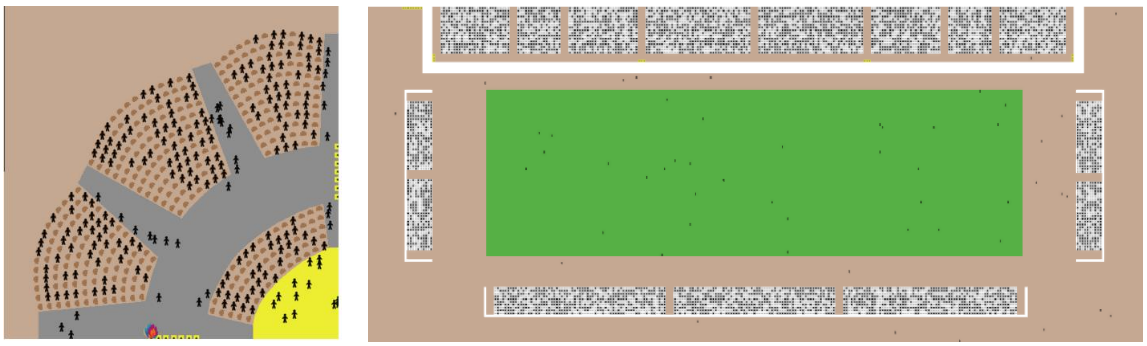


Figure 3: Venue setting in concert and stadium [42]

The ABS system described in this paper is a proof-of-concept system. As with any proof-of-concept prototype, several improvements can be made to the system, and several issues must be addressed to translate the system into a viable commercial system. The system could be improved by incorporating a more complex model of a person's decision-making process into each person agent. Another enhancement that could be made is in the area of fire dynamics. More accurate physical models of fire could include weather-related factors such as temperature, humidity, air pressure, and wind as well as topographical and fuel-related factors. Relevant topographical factors include the size, shape, and contour of the environment while fuel factors include the type of materials present in the environment.

Currently, the system limits the modeling of multi-level venues and it is always hard to simulate agents to move between the levels of the venue. Thus, adding this modeling capability is another relevant improvement that can be made and would allow for the simulation of more complex concert venue environments. Additionally, although the current system does allow for the specification of highly customized environments, this is accomplished via an input file which requires a minimum level of technical knowledge to create. An intuitive graphical user interface should be built for environment specification that makes this activity more user-friendly.

Another research by N. Pelachano, et al. [43] considers animating the evacuation of crowd from complicated buildings who may not understand the connectivity of the structure or who may accidentally find routes blocked. It requires simulated crowd behavior into consideration under two circumstances: agents communicate the understanding of building route, and agents take on distinct roles such as qualified staff, leaders, and supporters. Maces, without a centralized controller, is a distributed multiagent system. Each agent has its conduct based on natural variables of personality representing real psychological factors. The movement of agents is calculated at two levels. The high level corresponds to the process of finding a room sequence, whereas the low-level parallels to the local motion within a room. the input characteristics of the maze-like atmosphere are considered which are dimensions, number of exits, and number of hazards and the parameters required for simulation such as number of agents, percentage of trained agents, and percentage of leaders. Whenever two or more agents meet in a room, they share two pieces of data: locations of some of the risks that block routes, and sections of the building that other agents have thoroughly explored and found no available exit.

This evaluation has shown a significant improvement in evacuation rates when using inter-agent communication. They also observed the grouping behavior that emerges when the crowd contains a large proportion of dependent agents. The best evacuation rates are provided by only a comparatively smaller proportion of trained leaders. These results can be visualized in real-time with either 2D or 3D viewer.

Areas where there is room for improvement, including adding individualism into Helbing's model is that agents would have different local motions depending on their roles. The high-level wayfinding must be modified because people should be less likely to enter a full room when there are other possible paths available.

The building environment and the conduct of evacuation are crucial factors for the efficiency of building evacuation performance. Shifting to the use of agent-based models, many present studies have regarded the heterogeneous evacuation behavior arising from an individualized perception of the building sets, but few studies have integrated the consciousness of the evacuees about the predictable shift in spatial accessibility through activated fire safety facilities during emergency situations[44]. This study was intended to explore the specific influence of such spatial change on the efficiency of evacuation. L. Tan et al. [44] presented an agent-based building evacuation model in which the knowledge of the evacuee is considered, including both the spatial understanding of the stable environment during a typical scenario and the knowledge of the events of the predictable spatial change for firefighting purposes. Furthermore, when considering fire safety facilities, a semantic representation of a building environment is created to depict the alterable connectivity structure. Using the suggested model, during three specific fire situations, a series of evacuation simulations were conducted for groups of evacuees with different levels of knowledge.

To simulate the individualized route selection under the influence of a potentially changed connectivity structure, the evacuees' knowledge of the spatial environment, including the knowledge of the stationary spatial features and the awareness of the activated fire safety facilities, are considered. Each agent selects their escape route based on the assumed spatial accessibility. When congestion occurs, or the route is blocked, the agent updates its knowledge and adjust the escape route if there is another alternative route. To simulate the evacuees' local movement within each internal space, the building environment is geometrically represented as a grid consisting of small cells. In the proposed model, the building environment is represented by a semantic model at the macro level, where the connectivity relationship between the internal spaces, such as rooms and corridors, is updated according to the state of the fire safety facilities. At the micro-level, a grid-based representation is adopted to facilitate the simulation of the evacuees' movement within the internal space.

It is vital to note that the proposed model mainly concentrates on the influence of the predictable change in the spatial accessibility and does not include all of the factors that have or might affect the spatial accessibility of the building environment such as fire smoke, toxic gasses, etc.

In this study, they could have considered the evacuees' reaction when encountering the smoke in addition to their capability of avoiding the route that is blocked by activated fire safety facilities. Another issue is about the cell size of the grid-based representation. In this study, they used 40*40 cells given the common area occupied by a pedestrian and the computational efficiency. The cell size may influence the simulation results. The sensitivity of simulation results to the cell dimension needs to be investigated. Besides, when modeling the evacuees' movement at the micro-level, they have made simplifications, and some factors were not fully considered, such as the representation of the human body and the effect of sufficient dynamic width of exit. Given these facts, the model presented here is a prototype. More efforts are in need to validate the model and make it more applicable in the future.

In summary, there are large numbers of research modules on the subject of agent-based crowd simulation which have always tried to improve the existing software models and techniques. Still, none of these software models provide a method/process to incorporate any possible physical changes in an indoor environment before they are actually constructed/built or that might come in future. Also, external files cannot be imported to these software models to add a new 3D environment for simulation. However, our software model can simulate any 3D indoor environment with any possible physical change, any number of exits and any number of seating. Our software model helps in evaluation of the physical changes to check their effectiveness in case of an emergency for various scenarios based on incidents that happened in the past. It also proposes a vast range of parameters such as speed of agents, frame rate, etc., that can be used as a group or as an independent entity to simulate various sets of scenarios for multiple environments with multiple agents in order to reduce the significant number of casualties throughout the world.

CHAPTER 3

PROPOSED METHODOLOGY

In Section 2.1, we discussed 14 software models which we consider for comparison. Every software model has benefits and drawbacks. Previously, we only explained all these software models in detail. However, in order to explain the software model that we are designing, we need to compare existing software models also. These software models are compared on the basis of some parameters/features which we explain in detail in section 3.1. Then, we implement those missing features from every software model into our software model.

In beginning of this chapter, we compare 14 software models for simulation of emergency evacuation and later we discuss our method of designing a software model.

3.1 Existing Software Models

The 14 software models can be divided into 2 types according to their method of availability: available to the public and those available on a consultancy basis. In table 1 and table 2 software models are separated with a bold black line according to their method of availability. The models no longer in use are not considered for comparison.

Models available to the public:

- WAYOUT
- PEDROUTE
- Simulex
- FDS+Evac
- SimWalk
- BuildingEXODUS
- Legion
- MassMotion

Models available on a consultancy basis:

- PathFinder
- ALLSAFE
- CRISP
- EGRESS 2002
- SGEM
- EXIT89

we consider a set of parameters/features for thorough comparison of these software models. The parameters/features that we consider for comparison can be classified into two: main features and special features.

The main features for the model are the mandatory parameters that can define the working of a simulation. Below are the main features that we consider for comparison of software models:

- Availability to the public
- Visualization of the model
- Purpose
- Movement of agents
- Incorporation of fire data
- Computer-aided design (CAD)

Each main feature described here represents a column in Table 1. we compare 14 software models on the basis of main features in table 1.

The special features for the software model are the optional parameters that the model can or cannot have. However, these features increase the level of scenarios or possibilities in a specific software model.

Below are the special features that we consider for comparison of software models:

- Counter-flow
- Exit blocked
- Toxicity of the agents
- Disabled/slow-moving agents
- Effect of fire conditions on agents

Each special feature described here represents a column in Table 2. we compare 14 software models on the basis of special features in table 2.

Before the comparison, we explain every feature and label related to them in the table. Below, we explain each main feature in detail and label related to them.

Availability

Although the software models included in this study appeared in peer-reviewed journals, there are variations in how these models are made publicly accessible. In order to differentiate software model's availability, we separate them into two group. In the first group, software models are available for free in some cases. The second group includes models to be used on a consultancy basis by the development business. The consultancy firm uses the models in this group, and only the outcomes are provided to the user. In Table 1, software models that are available to the public for free are labeled with a "1," models that are available only through the development company which are paid are labeled with a "2".

Visualization

Visualization defines whether a software model enables the user to visualize the structure's evacuation output. Evacuation visualizations allow the user to see the location of the bottlenecks and congestion points within the room. Many of the models enable at least 2-D visualization and more lately have published versions or cooperate with other virtual programs to present 3-D outcomes. Visualization choices are often dependent on the audience. In Table 1, software models that provide 2-D visualization are labeled with "2D," software models that provide 3-D visualization are labeled with “3D,” and models which don’t have any visualization capabilities are labeled with “N”.

<u>Purpose</u>

Purpose defines the software model's use as it refers to particular kinds of buildings as determined by the developers of the model. Some of the software models emphasize a specific kind of building, and others can be used for many construction types. The main aim of using this as a feature is to know whether the model is meant to simulate the building design selected by the user. The software models as described in Table 1, include models that can simulate any building which are labeled with “1,” residential-specific models are labeled with “2,” government transport station-specific models are labeled with “3,” and models that only simulate 1-route/exit of the building are labeled with “4”.

<u>Movement of Agents</u>

The Movement of Agents relates to how the software models move occupants throughout the building. For most models, the user or modeling program generally assigns a particular unimpeded (low density) velocity to occupants. The more significant differences in the models occur when the occupants get closer, leading in queuing and congestion within the construction. Here are the various methods in which models that represent occupant motion and limited flow throughout the construction:

- <u>*Density correlation* (D):</u> The model assigns speed and flow depending on space density to people or populations.
- <u>*User's choice* (UC):</u> The user assigns values of speed, flow, and density to particular construction fields.
- <u>*Inter-person distance* (ID):</u> A 360 ° "bubble," surrounds each individual, requiring a minimum distance from other occupants, barriers, and construction elements (walls, angles, handrails, etc.).
- <u>*Emptiness of next grid cell* (E):</u> The occupant does not migrate into a grid cell that is already occupied by another occupant. Therefore, the occupant waits for the next cell to get empty. If there are more than one occupant waiting for the similar cell, then the model fixes the disputes that may occur while deciding which occupant moves first.

- *Conditional* (C): Movement throughout the building depends on environmental circumstances, structure, other evacuees, and/or fire condition with conditional models. Not much emphasis is put on congestion within the room for this designation alone.
- *Cellular automata* (CA): In this model, the occupants move through the simulated throw of a weighted die from cell/grid room to another cell. In this, all the agents behave as a single unit, e.g., if one agent dies, the others die due to the same effects such as a fire.

In table 1, various methods of agent's movement are labeled as follows: Density Correlation as "D," User's choice as "UC," Inter-person Distance as "ID," Emptiness of next grid cell as "E," Conditional as "C," and Cellular automata as "CA".

Fire Data

The fire data describes whether the model enables the user to integrate the fire impacts into the simulation of evacuation. The model can integrate fire information in the following ways: import fire data/results from another model, enabling the user to enter fire data throughout the evacuation at certain times, or the model can have its fire model running simultaneously with the evacuation model. If the model cannot integrate fire data, it runs all simulations in non-fire mode. The aim of evacuation models to include such information is to evaluate occupants' security, who move through degraded circumstances and to see how their conduct can be altered. In table 1, the software models that can import fire data are labeled as "Y1," the software models that has its fire element running simultaneously with evacuation are labeled as "Y2," and the model that runs all the simulations in non-fire mode are labeled as "N".

Computer-Aided Design (CAD) drawings

CAD feature defines whether the model enables the user to import documents into the model from a computer-aided design program (CAD) or other files containing the construction layout. In many instances, importing CAD files is time-saving and more accurate. If a user can depend on accurate CAD drawings instead of manually laying the building, there is less space for the building's input mistake. However, the user is required to be more cautious about properly connecting all the distinct components If the model has feature of importing the CAD model, label (Y) is used for those specific models in Table 1. On the other hand, when the model does not have that capacity, the label of (N) is used.

Below we mentioned every label that is used in the table 1:

Availability:

- 1: Free
- 2: Paid

Visualization:

- 2D : 2-D visualization
- 3D : 3-D visualization
- N : No visualization

Purpose:

- 1 : Simulate any type of building
- 2 : Only specializes in residences
- 3 : Specialize in public transport stations
- 4 : Only simulate 1-route/exit building

Movement:

- D : Density
- UC : User's choice
- ID : Inter-person Distance
- E : Emptiness of next grid cell
- C : Conditional
- CA : Cellular Automata

Fire Data:

- N : Model cannot incorporate fire data
- Y1 : Model can import fire data
- Y2 : Model has its own simultaneous fire element

CAD:

- N : Does not allow importation of CAD files
- Y : Does allow importation of CAD files

Software Model	Availability to Public	Visualization	Purpose	Movement	Fire Data	CAD
WAYOUT	1	2D	4	D	N	N
PEDROUTE	2	2D,3D	3	D	N	Y
Simulex	1	2D	1	ID	N	Y
FDS+Evac	2	2D,3D	1	ID	Y2	N
SimWalk	2	2D,3D	3	D	N	N
Building EXODUS	1	2D,3D	1	E	Y1	N
Legion	1,2	2D,3D	4	ID,C	Y1	Y
MassMotion	1,2	2D,3D	1	C	N	Y

PathFinder	2	2D	1	D	N	Y
ALLSAFE	2	2D	4	E	Y1	N
CRISP	2	2D,3D	1	E,D	Y2	Y
EGRESS 2002	2	2D	1	D(CA)	Y2	N
SGEM	2	2D	4	D	N	Y
EXIT89	2	N	3	D	Y1	N

Table 1: Main feature comparison

Table 2 is included in a manner to describe the more particular capabilities of each model. This table is included for users who want to simulate particular evacuation situations and/or for users to understand the variations in the sophistication of the software model. It can be seen that as the level of sophistication rises, the number of distinctive characteristics simulated by the model rises. The particular special feature included in the evaluation are as follows, many of which can be enabled or disabled by the user:

- Counterflow
- Exit block/obstacles
- Fire conditions affect Behavior
- Toxicity of the occupants
- Physically Challenged/slow-moving agents

Below, we explain each special feature in detail and label related to them in Table 2.

Counter-flow

During an evacuation as the evacuating occupants in the building move in one direction, emergency responders or other individuals may need to move in the opposite direction. This can lead to less width for the occupants of the construction. The counterflow defines those models that can simulate counterflow. Counterflow is the parallel movement of occupants in different directions which can be used if the desired simulation is to include either firefighters or other occupants that have duties to perform during the evacuation. If there is staff in the cinema hall that is assisting the agents out, counterflow mechanism applies to the assisting staff.

Exit Block

For many purposes, exit gates can be blocked. Exit block defines the models that enable the user to block exits to be used by simulated occupants. A software model's user could be interested in scenarios where not all paths can be used. Exit block identifies those models that allow the user to block exits from the use by simulated occupants in the scenario that have this capacity either allow the users to manually block the exit gate form use before or even during the situation, or the model blocks the exit from use, owing to certain circumstances (e.g., a falling debris from the roof can block the exit). Users should know how the occupants view the blocked exit and whether the exit can be blocked at a given time or only at the simulation start.

Fire Conditions affect Behavior

In a building fire, it is likely that the environmental conditions produced by the fire can affect the nearby occupants. Some software models that incorporate fire circumstances (see Table 1) have the ability to alter occupant conduct depending on these fire circumstances, including the following simulated situations: occupants selecting another exit because of a large fire and might get hurt by fire, occupants turning back and turning away from the fire. Fire conditions defines whether the model of evacuation is capable of changing occupant behavior owing to fire circumstances.

Toxicity

Material burning generates poisonous gases (narcotic and irritant gases), aerosols for smoking, and heat. Fires can spread smoke and warm gasses throughout a construction, presenting occupants with toxic gasses that can trigger symptoms of slight headaches, potentially leading to lower rates of motion, disability (i.e., where the occupants stop moving without help), or even death. Toxicity defines those models that mimic occupants' disability or death owing to toxic products from fire circumstances contained in smoke.

Physically Challenged/Slow moving Occupants

During an evacuation from a construction, not all occupants move at the same unimpeded velocity. The type of population in a given scenario must be considered by the software model used to determine whether the standard movement speeds are suitable for the entire population of the building. Physically challenged / slow-moving occupants defines those models that enable the user to assign reduced unimpeded speeds to a particular proportion or group of occupants to represent occupants that may move slower during evacuation. Just because the software model has this capability, does not mean that physically challenged occupants or slow-moving occupants cause bottlenecks or congestion in smaller spaces throughout the building.

For all of these features, the software model is labeled as either having this capability or not having this capability. If the model has this capability, a "Y" for yes is placed in Table 2. If not, an "N" for no is placed in Table 2.

Software Model	Counter-flow	Exit Block	Toxicity	Physically Challenged/ Slow-moving	Fire Conditions
WAYOUT	N	N	N	N	N
PEDROUTE	N	N	N	Y	N
Simulex	Y	Y	N	N	N
FDS+Evac	Y	Y	Y	N	N
SimWalk	Y	N	N	Y	N
Building EXODUS	Y	Y	Y	N	Y
Legion	N	Y	Y	Y	Y
MassMotion	Y	Y	N	Y	Y
PathFinder	N	N	N	N	N
ALLSAFE	N	N	N	N	Y
CRISP	Y	Y	N	Y	N
EGRESS 2002	Y	Y	Y	Y	Y
SGEM	Y	Y	N	Y	N
EXIT89	Y	Y	Y	Y	Y

Table 2: Special feature comparison

It should be noted that these models can only work with a set of parameters that are predefined inside the software package. These software models cannot import any file to add a new 3D environment into their software package. Therefore, if any possible physical change and new feature/parameter in technology comes up in the future, it cannot be implemented on the software model. However, our approach can work with any feature/parameter or any possible physical change, which can be easily imported and implemented with minor configurational handling which we discuss in section 3.2. We also propose a set of physical changes and various other features to make our model work in any scenario to overcome drawbacks of these software models.

Apart from all previously mentioned software models, we reviewed several other models that are being used by individuals and companies throughout the world (these software models also lack some or many functionalities as compared to the software model that we are working on):

- Oasys
- NET LOGO
- Anylogic
- Repast
- Jade
- MATSIM
- MicroPedSim and many others

3.2 Proposed Methodology

Our thesis involves the development of a base software model of evacuation systems for an indoor environment which can incorporate any possible physical change before they are actually constructed/built. Our software model helps in evaluation of the physical changes to check their effectiveness in case of an emergency for various scenarios based on incidents that happened in the past. Also, we can implement any possible physical changes in 3D indoor environment that might come in future using an agent-based crowd simulation technique. However, a comprehensive workflow can help understand our thesis' idea better.

Figure 4 illustrates the complete workflow of our software model. Each box in this workflow provides a wide understanding of our software model's development as well as the implementation of some possible physical changes that we propose.

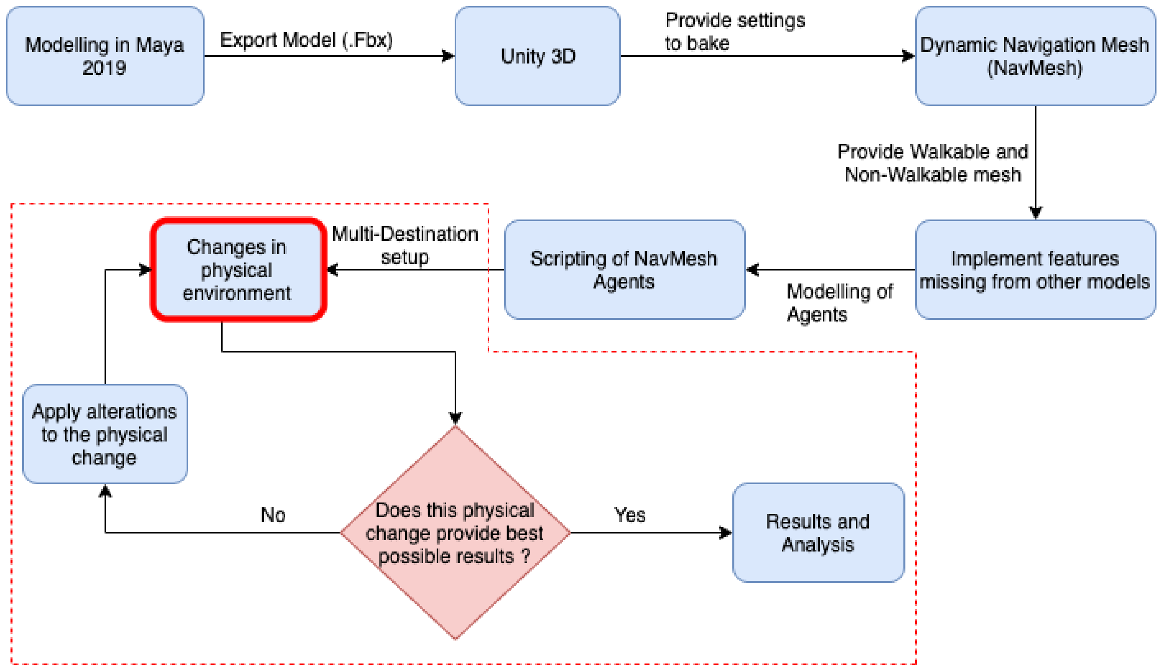


Figure 4: Flowchart of methodology

The flowchart in Figure 4 depicts various techniques and software's to design the whole evacuation software model. Also, the highlighted part depicts our main idea to incorporate any possible physical change which can help improving evacuation timing and efficiency. We discuss each component of the flowchart in details in further subsections. Also, in order to understand how our simulation software model works, we get insight from a generalised pseudocode for simulation of any physical change that can come in future in Figure 5.

```
Algorithm 1: Emergency Evacuation Software Model
  Input  : .fbx file from Maya
  Output: simulated model
define dynamicNavMesh; using unity 3D
if number(agents) > 1 then
    define NavMesh(Average size of the agents, Average radius of the agents);
    create walkable blue region && create non-walkable red region;
    return NavMesh
else
    Error
end
define variable thresholdCasualty;
define variable thresholdEvacuationtime ;
define list of physicalChanges;
while (numberCasualty ≤ thresholdCasualty) &&
  (evacuationtime ≤ thresholdEvacuationtime) do
    apply alteration to physical changes;
    define new physical changes ;
    add new physical changes to the list;
end
upload results and analysis
return results
```

Figure 5: pseudocode for simulating the physical changes

Hence, for simulation of any physical change the input must be a .fbx file incorporating any 3D indoor environment. Also, in start we have to define the parameters that creates the navigation mesh, threshold casualty and evacuation time to achieve expected results from our software model. We explain each component of the workflow and pseudocode in further subsection.

3.2.1 Modelling in Autodesk Maya

The first component to start creation is Modelling in Autodesk Maya 2019.

- **How Maya Works?**

  Maya is an application that is used to generate 3D assets, for use in film, television, game development and architecture, rendered in real-time. Maya provides for many characteristics, such as Volume Attributes, File Path Handling, Paint Effects Surface, Clip Matching, Inline Help, URI Support, Binding, and more.

- **How to create 3D models using basic Maya tools?**

  Basic Maya display consists of various tabs for modeling the different types of scenes. These tabs include curves/surfaces, polygons, rigging, animation, etc. We used polygons and curves to create models for cinema hall (Figure 6) from scratch. Polygons consist of geometry based on vertices, edges, and faces that you can use to create three-dimensional models in Maya.

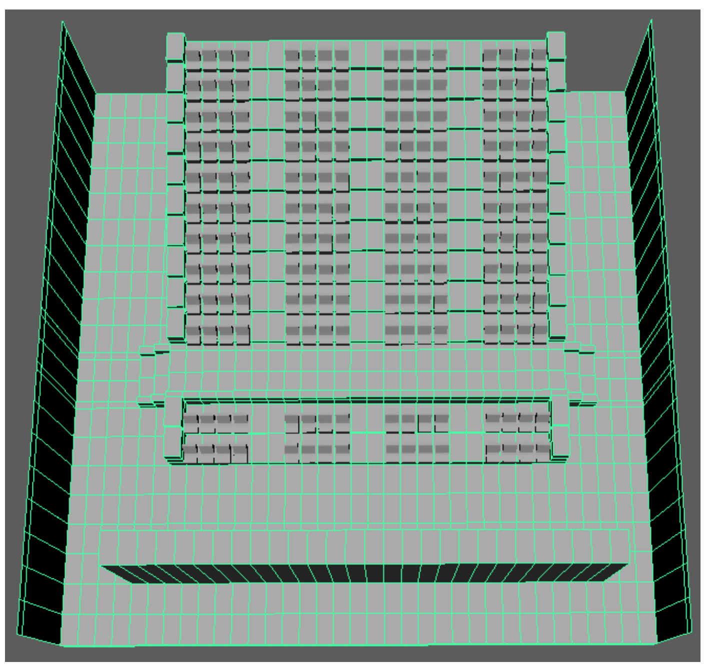


Figure 6: Cinema hall model in Autodesk Maya 2019

- **Why we chose Maya?**

  The launch of Maya 2014 has brought in some powerful modeling instruments into existence and the modeling toolkit that starts to level the playing field, with some excellent re-topology instruments and quicker workflow. Maya, with an enormous library of animation instruments, has been known to be a much stronger application when it comes to animation. Maya's proprietary scripting language MEL is extremely customizable, and that's one reason why Maya can be discovered a lot in the film industry because studios like to be able to produce their own set of characteristics that can assist them in accomplishing their work much easier.

- **Exporting files from Maya to Unity 3D**

  One of the other reasons why we choose Maya is that it provides a wide range of options while exporting our 3D model to any other software. As we were going to use Unity 3D after modeling, we exported our file in .Fbx format for better stability of the 3D model. The file got exported without any glitches and with all minor details intact. Also, this process of exporting the .Fbx file act as the input for our software model described pseudocode in Figure 5.

3.2.2 Modeling in Unity

The next component after exporting the file from Autodesk Maya 2019 is Unity 3D. The remaining components are designed and implemented in Unity 3D after successful importation of the .Fbx file.

- **How Unity works?**

  Unity is developed by Unity Technologies and is a cross-platform game engine. The engine can be used to create simulations, three-dimensional, 2-dimensional, virtual reality, and enhanced reality games. It is also adopted by non-video gaming sectors, such as film, automotive, engineering, architecture, and construction. Unity allows users to create both 2D and 3D games and experiences, and the engine provides a primary C # scripting API for both the Unity editor in the form of plugins and games, as well as drag and drop functionality. The navigation system in Unity allows us to provide powerful pathfinding to the characters/agents in games or simulations very easily. The system provides an intelligent surface called navigation mesh (Navmesh) for agents to move around.

- **How NavMesh works?**

  The surface on which agents move is stored as convex polygons by the **NavMesh**. Convex polygons are helpful as we understand that there are no obstructions within a polygon between any two points. In relation to the limits of polygons, we store data on neighboring polygons. This makes it possible for us to reason the entire walkable region.

- **Navmesh Agent**

  Navmesh agent is a component which takes some waypoints and makes an AI agent patrol in the map. Therefore, to move the agent, we need to tell navmesh agent to move the agent to a destination. To achieve this, we can tell an agent to start calculating the simple path by using NavMesh Agent and the destination property with the point we want AI agent to move. Once the calculation is finished, the AI agent begins to move along the path until it reaches its destination.

- **Animation and Rigidbody (Collider)**

  After deploying Maya models in Unity3D, we intend to deploy Agents (characters). To move an agent in simulation we used the following methods defined by unity:

- Rigidbody
- Animation Movements

o ***<u>Using Rigidbody</u>***

Rigidbody is a component which is used to control an object's position through physics simulation. Adding a Rigidbody component to an object put its motion under the control of Unity's physics engine. A gravity function is not required; as a Rigidbody objects gets pulled downwards by the gravity provided by Unity’s physics engine. To move a Rigidbody object in real-world space, a developer needs to apply add force property to that object. Force can be applied from anywhere like upward, downward, left, and right. Force is applied continuously along the direction of the vector.

Rigidbodies enable agents in simulated model to act under the control of physics. The Rigidbody can receive forces and torque to make the agents move realistically. Any agent must contain a Rigidbody to be influenced by gravity, act under added forces via scripting, or interact with other agents in the crowd through the NVIDIA PhysX **physics engine** [45].

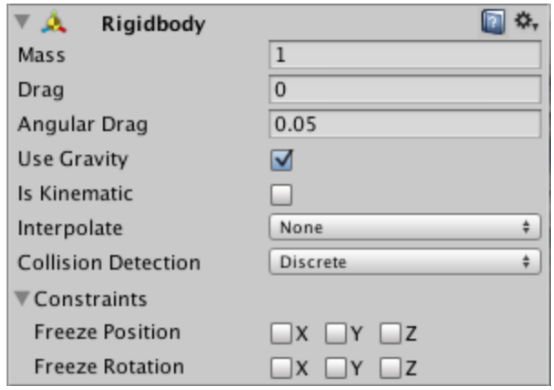


Figure 7: Rigidbody component values[45]

Rigidbodies must be explicitly added to our agents before they will be affected by the physics engine. We can add a Rigidbody to our selected agents from **Components->Physics->Rigidbody** in the menu. Now our agent is physics-ready; it will fall under gravity and can receive forces via scripting, but we may need to add a Collider or a Joint to get it to behave exactly how we want.

To control our Rigidbodies, we will primarily use scripts to add forces or torque. We can do this by calling **AddForce()** and **AddTorque()** function on the agents's Rigidbody.

Colliders are another type of element to be added together with the Rigidbody to allow collisions to occur. When two Rigidbodies crash into each other, a collision will not be determined by the physics engine unless a Collider is connected to both agents. Collider-less Rigidbodies will simply pass through one another during physics simulation [45].

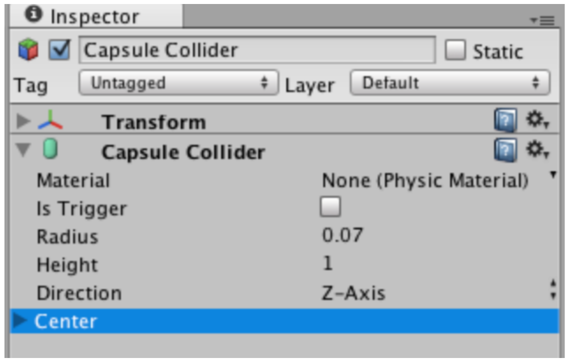


Figure 8: Capsule collider values[45]

There are various types of colliders available in Unity 3D such as Box Collider, Sphere collider, Capsule collider, Mesh collider, Wheel collider and Terrain collider. There are various specific values associated with each of them as shown in Figure 8 for capsule collider [45].

- ***<u>Animation Movements</u>***

To use more than one animation, the developer needs to use Animator component. An animator is actually a state machine. In the Animator component, animations need to be dragged in and use transitions to go from one state to another. To move from one state to another, transition conditions are used. In transition conditions, we need to make some parameters and add those on the transition condition. Different types of parameters can be set, to use transition conditions like Float, Boolean, Integer, and trigger. However, we didn't use Rigidbody in our simulation to move the player around as we used animation movement to move the player around the real-world space. Trigger and bool are actually the same as when we set the trigger, its values go from false to true, and when the task is done, its value goes back to false. There is another type of animator to control animation which was introduced in Unity5 called Blend Tree. To accomplish complicated animations, Blend Tree is used, which consists of multiple small states.

For some situations, mainly creating some specific effects, it is necessary to switch control of the agent between animations and physics. For this purpose Rigidbodies can be marked **<u>isKinematic</u>**. While the Rigidbody is marked isKinematic, it will not be affected by collisions, forces, or any other part of the physics system. This means that we have to control the agent by manipulating the Transform component directly. Kinematic Rigidbodies will affect other agents, but they themselves will not be affected by physics. For example, Joints which are attached to Kinematic agents will constrain any other Rigidbodies attached to them and Kinematic Rigidbodies will affect other Rigidbodies through collisions [45].

- **Physics**

  To explain physics, let's take the example of earth. In our daily life, when we throw something up in the air, that thing tends to come back on earth because of gravity. Unity3D provides some useful components to achieve similar behavior. To achieve gravity, Unity3D helps us by providing Rigidbody component which can be used to detect gravity, and developer can vary the value of gravity also. Its default value is however set at 9.8 m/s. There persists a permanent question as to how can we know if something collides with the surface or not? To detect collision Unity3D provides a component called the collider. There are different kind of colliders. If the simulation object is a sphere or a capsule, then capsule collider is better and if the mesh is too complicated, then mesh collider is a better choice.

  To have convincing physical behaviour, an agent in a simulation model must accelerate correctly and be affected by gravity, collisions and other forces. The integrated physics engines of Unity provide us with components that manage the physical simulation. With just a few parameter settings, we can create agents that behave passively in a realistic way (i.e. they're going to be moved by collisions and falls but don't start moving on their own). There are actually two distinct physics engines in Unity: one for 2D physics and one for 3D physics. The main concepts between the two engines are similar (except for the additional dimension in 3D), but they are implemented using different components. For example, there is Rigidbody component for 3D physics and a similar Rigidbody 2D component for 2D physics [45].

- **Why we chose unity?**

  Unity is one of the best game engines for developing games and simulations. The key attractions to why Unity is powerful, are the low learning curve for beginners, active ecosystem of asset and plugin creators, rapid development speed, and its cross-platform integration that supports 25 platforms. Unity Prefabs, which are preconfigured Game Objects, provide useful and flexible workflows that allow you to function confidently without worrying about making time-consuming mistakes. It also provides scripting in C#, which is one of the most popular and easily understandable languages to start.

3.2.3 Dynamic Navigation Mesh

Path planning and navigation play a significant role in simulated virtual environments and computer games. However, while designing an emergency situation, we need a dynamic environment, in which obstacles can fall or emerge at any point in time. In order to work it out, we need a dynamic navigation mesh on which agents can move and find a path to the nearest exit. Also, this process of defining the dynamic navigation mesh is described in step 1 of pseudocode in Figure 5.

- **How unity provides walkable and non-walkable mesh?**

  There are two problems which needs to be addressed if we want to relocate characters in the game intelligently. These problems are - First, how to discover the target and second how to move there. These two problems are closely linked, but in nature, are quite distinct. The issue of to discover the target is more global and static because it considers the whole scene. However, It is more local and vibrant to move to the target. Every Agent only looks at the direction to move and how to avoid collisions with other moving agents.

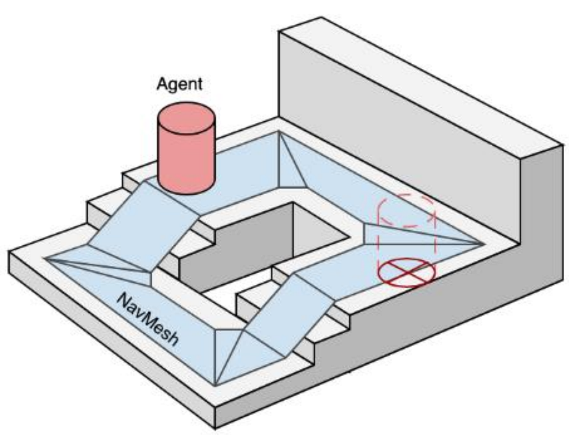


Figure 9: Walkable areas [45]

The navigation system is built automatically from the geometry of the scene to represent the walkable areas (where the agent can stand and move). Then the locations are connected to a top laying surface called the navigation mesh (NavMesh for short) (Figure 9). While defining the navigation mesh to provide walkable and non-walkable region, we must provide default parameters such as radius of agent, height of agent, walking speed, etc. This process of providing default parameters is described in steps 2-10 of pseudocode in Figure 5.

3.2.4 Implementing Features missing from other models

We took various features under consideration for comparison of existing software models in Table 1 and Table 2. Further, we discuss all the features that are missing from some of those software models and implemented them in our software model in detail.

- *Crowd distribution:* During high-stress situations such as emergency evacuation, every crowd employs a sheep herding behavior. Therefore, we implement crowd distribution in order to reduce casualties and improve evacuation time. We used the ray cast feature to implement this feature. In one frame an agent casts a ray to check how many other agents are going to that door and are there any path leading to that door or not. The default entry of the raycast feature for every exit is put to the value equal to the 10% population for that scenario. If more than 10% of people are going to the single exit, the next person takes another nearest exit.
- *Fire:* As we discussed the fire data feature in section 3.1, we can implement a fire component either by importing it or implementing into our software model. We chose to implement it in unity 3D simultaneously to get more accurate results. In this scenario, we can also check how many people died of fire (Fire also affects the behavior of the agent)
- *Exit block:* We used random blocks of concrete to drop from the top or any other effect to block the gates in certain situations. Thus, forcing the agents to take another exit, which is not blocked.
- *Toxicity:* Toxicity provides us with the overview of scenario during choking of agents due to smoke or any biodegradable invisible gas spread out. In this case, we can set the

timing for the agents to start choking, etc. If they are still inside after the start choking time, that leads to death by choking.

- *Physically Challenged/Slow moving*: This phenomenon was one of the most important phenomena to implement during evacuation as the least number of software models are working on it. To include that we used a wheelchair, in which we provided wheelchair-accessible doors and priority is given to wheelchairs as compared to normal running agents.

3.2.5 Scripting of Navmesh Agent

The Unity C# Job System enables users to write multithreaded code that interacts well with the remainder of Unity and makes writing right code simpler. Multithreaded code writing can provide advantages of high performance. These include substantial increases in frame rate.

- **Agent pathfinding**

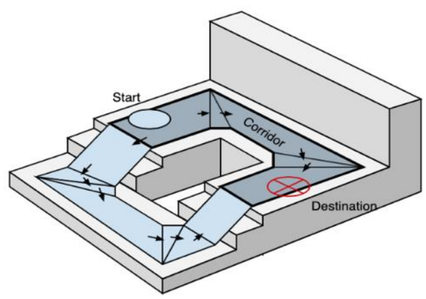


Figure 10: Agent pathfinding [45]

We first need to map the starting and destination places to their closest polygons to discover a route between two places in the scene. We then begin looking from the starting point, visiting all the neighbors until we achieve the polygon of target. Tracing the visited polygons enables us to discover the sequence of polygons which leads to the starting location (Figure 7). The algorithm that is used by unity for finding the route is A*.

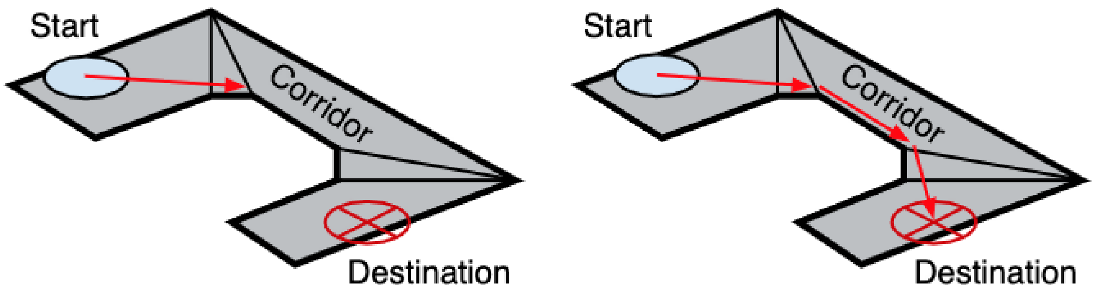


Figure 11: Destination tracing [45]

The polygon sequence that describes the route from the beginning to the polygon of the target is called a corridor (Figure 11). By providing guidance towards the next visible corner of the corridor, the agent reaches the target. If you have a straightforward match where only one agent is moving in the scene, finding all corners of the corridor in one swoop is okay and animating the personality to move along the line segments that connect the corners. While dealing with numerous agents moving simultaneously, when avoiding each other, they have to deviate from the initial route (Figure 12). It becomes challenging to attempt to correct such deviations using a route composed of line segments, and one can also be prone to committing common errors.

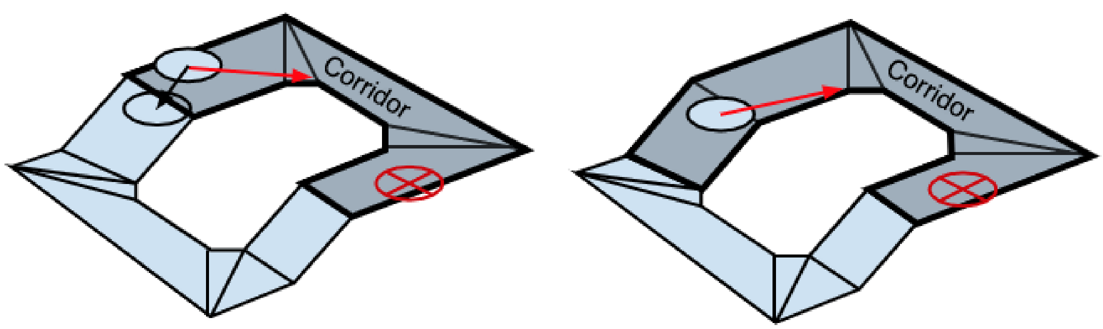


Figure 12: Multi-directional tracing to destination [45]

Since the movement of the agent in each frame is quite low, we can use the polygon connectivity to solve the corridor if we need to take a little detour. We then discover the next noticeable corner to be steered rapidly.

- **Avoiding obstacle**

  The steering logic requires the next corner position and a required direction and speed (or speed) required to achieve the target is based on those numbers. Using the required speed to move the agent may result in a collision with other agents. Obstacle avoidance selects a fresh speed that balances between moving in the required direction and preventing future collisions with other agents and navigation mesh edges. Unity uses barriers to predict and stop collisions with reciprocal velocity (RVO).

- **Moving agent to target**

  The final velocity is calculated after steering and avoiding obstacles. The agents are simulated in Unity using a straightforward dynamic model, which also requires acceleration into consideration to enable more natural and smoother motion. At this stage, you can feed the speed to the animation system from the simulated agent to move the character using root motion. The navigation system can also take care of that. After moving the agent using either technique, the position of the simulated agent is shifted and restricted to NavMesh. For robust navigation, this last tiny step is essential. Implementing Features missing from other models

- **Object destruction**

  Public static void Destroy(Object obj, float t = 0.0F);

  Destroy() function is used to destroy any object. If obj is a component, the element is removed and destroyed by the GameObject. If the obj is a GameObject, it destroys all the components and children of GameObject. The actual destruction of objects is always postponed until after the present update loop. Destroy is hereditary from the base class UnityEngine. Object.

- **Other features of Unity we used:**

  - <u>*RAYCAST*</u>

    Raycasting is the process of firing an invisible ray from a point in a stated course to spot whether there are any collisions in its path. By tracing the bullet or a laser from start to end we get to know how it behaves. In the simulation environment, we can physically observe and manipulate it. This should be helpful in finding a way to the closest gate or in checking if the obstacle is in front or not and also, in the event of retractable chairs that we're going to talk about in prosed modifications. Unity3D raycasting is divided into two separate parts, 2D and 3D. Both are objects of physics.

- *Coroutines*

  Coroutines assure that any action that takes place in a function must take place within a single frame update; a function call cannot be used over time to contain a procedural animation or a sequence of occurrences. Consider the process of gradually decreasing the alpha (opacity) value of an object until it becomes entirely invisible as an instance. By default, after it returns, a coroutine is restarted on the frame, although a time limit can also be introduced.

- **Multi-Destination Setup**

  After implementing all the agent finding techniques and modelling of agents to find the nearest exit, the next component in the flowchart is the multi-destination setup. Thus, in order to show the realistic emergency scenario, we need multiple exits for the cinema hall or any other model. If any exit is blocked, the agents go the next nearest exit.

Thus, we discussed our approach to develop a software model that can incorporate features that are missing in other 14 software models. In order to show that our software model can incorporate those features, we compare those 14 models with our software model approach. We can now compare which software models came close to our approach and what they lack in terms of features. Similar to the previous comparison (Table 1 & 2), we add a separate row at the end of the table 3 and table 4 to compare other 14 software models to our approach.

Table 3 provides comparison of other software models with our approach based on main features (labels of table 3 are same as labels of table 1).

Software Model	Availability to Public	Visualization	Purpose	Movement	Fire Data	CAD
WAYOUT	1	2D	4	D	N	N
PEDROUTE	2	2D,3D	3	D	N	Y
Simulex	1	2D	1	ID	N	Y
FDS+Evac	2	2D,3D	1	ID	Y2	N
SimWalk	2	2D,3D	3	D	N	N
Building EXODUS	1	2D,3D	1	E	Y1	N

Legion	1,2	2D,3D	4	ID,C	Y1	Y
MassMotion	1,2	2D,3D	1	C	N	Y
PathFinder	2	2D	1	D	N	Y
ALLSAFE	2	2D	4	E	Y1	N
CRISP	2	2D,3D	1	E,D	Y2	Y
EGRESS 2002	2	2D	1	D(CA)	Y2	N
SGEM	2	2D	4	D	N	Y
EXIT89	2	N	3	D	Y1	N
Our Approach	1	2D,3D	1	D,UC,ID,E ,C,CA	Y1,Y2	Y

Table 3: Main feature comparison with our approach

Table 4 provides comparison of other software models with our approach based on main features (labels of Table 4 are same as labels of Table 2).

Software Model	Counter-flow	Exit Block	Toxicity	Physically Challenged/ Slow-moving	Fire Conditions
WAYOUT	N	N	N	N	N
PEDROUTE	N	N	N	Y	N
Simulex	Y	Y	N	N	N
FDS+Evac	Y	Y	Y	N	N
SimWalk	Y	N	N	Y	N
Building EXODUS	Y	Y	Y	N	Y
Legion	N	Y	Y	Y	Y
MassMotion	Y	Y	N	Y	Y
PathFinder	N	N	N	N	N
ALLSAFE	N	N	N	N	Y
CRISP	Y	Y	N	Y	N
EGRESS 2002	Y	Y	Y	Y	Y

SGEM	Y	Y	N	Y	N
EXIT89	Y	Y	Y	Y	Y
Our Approach	Y	Y	Y	Y	Y

Table 4: Special feature comparison with our approach

Main features and special features shown in the comparison are explained in section 3.2. After carefully comparing these software models in Table 3 and 4. Our approach performs better than the software models available. The model that came closest to our approach and is in use for various projects is BuildingEXODUS (Column with "green"), nonetheless it also lacks some features such as physically challenged/slow-moving agents it can't provide, and it can only import fire data but cannot implement fire element running simultaneously with evacuation. It should be noted that these models can only work set of parameters that are predefines inside the software package. These software models cannot import any file to add a new 3D environment into their software package. Therefore, if any possible physical change and new feature/parameter in technology comes up in the future, it cannot be implemented on the software model. However, our approach can work with any feature/parameter or any possible physical change, which can be easily imported and implemented with minor configurational handling. Thus, we overcome all the drawbacks of other software models. Also, comparing our software model with previous agent-based crowd simulation models provided us with ideas to improve the simulation timings and efficiency using some possible physical changes. By implementing these physical changes, shows that our model can work in any scenario. We discuss all these physical changes in detail in Chapter 4.

3.2.6 Changes in Physical Environment

Once all main components of the workflow are designed, we can now import an external file designed in a CAD software (.fbx file format) or design a physical change inside our model to check if the physical change creates a difference in evacuation timing and efficiency. It helps engineers involved in planning and designing to create prototypes of the physical changes outside our software model and import them and check all these physical changes work. After implementing any physical change, Either the physical change does not provide appropriate expected results, or it does provide appropriate result.

If it does not provide appropriate result then with alterations in the physical change, the satisfactory results can be acquired. This process of applying alterations to existing physical changes until the software model provides appropriate expected results is described in steps 12-18 of pseudocode in Figure 5.

3.2.7 Results and Analysis

This component of the workflow provides the results and comparison with an actual event. We discussed this section in detail in Chapter 4, after simulating various physical changes in our cinema hall model to their effectiveness in case of an emergency.

3.3 Experimental Setup: Parameter Accessibility in Unity

In order to work with the parameters that the user can check to turn them on and off, we designed a Boolean valued pane from scratch (Figure 13). Whenever we need, we can add a new feature which can be easily accessible with vital values that can be changed accordingly to provide the best results for the simulation. Also, we can add list of physical changes in this Boolean pane which is described in step 11 of pseudocode in Figure 5.

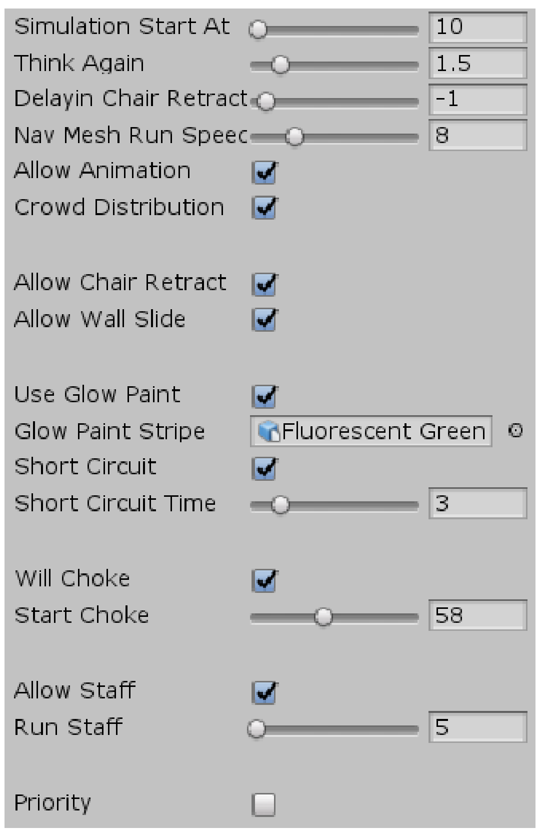


Figure 13: Parameter accessibility pane in unity

Below we discuss each parameter/change that we incorporated in the Boolean pane in unity for our simulation to run smoothly. Some of these parameters are only valid for specific physical change such as allow chair retract which can only work with retractable chairs. Similarly, use glow paint parameters only works with glow in the dark paint physical change. Some of the parameters are mandatory for all kind of scenarios such as simulation start time, navmesh run speed, etc.

- *Simulation start at:* This parameter provides you a time slider for the simulation start time. We can choose between 5 seconds to 1200 seconds/10 minutes.
- *Think Again:* This parameter works with the NavMesh AI. Thus, in order to reach the nearest exit, the agent has to think at every frame because the NavMesh is changing dynamically. The value provides the number of frames after which the AI starts calculation to find the nearest exit.
- *Delay in chair Retract:* this is the time taken by the chair after no agent is near to the chair.
- *NavMesh Run Speed:* it provides a default speed for an agent, nevertheless we can provide distinguishable speeds for different types of agents i.e., men (speeds between 4-8), physically challenged (speeds between 3-7), overweight agents (between 1-4), children (5-9), etc.
- *Allow Animation:* this parameter enables the animation sequence for running, moving, falling, directing, etc.
- *Crowd distribution:* this parameter enables the crowd distribution feature for the simulation
- *Allow Chair Retract & Allow Wall Slide:* After checking these features, it turns on the retractable seats and moving walls physical change into the simulation.
- *Glow Paint & Priority*: Similarly, we can allow the working of glowing paint on floor and priority given to physically challenged/children in our simulation. We also provided wheelchair accessible doors, that are in front of the cinema hall model and on the side of St. Denis center model, to provide an ease of access to them.
- *Short Circuit and time:* Using this parameter, we provide the scenario of an electrical short circuit in which nothing is visible to agents due to blackout, and we can also use the time slider to provide the specific time for a short circuit to happen.

- *Will choke and start choke*: We have implemented a smoke scenario in which the smoke rises after 20 seconds into the simulation. Smoke can result in choking of the agents; using this parameter, we can or cannot allow the agents inside to choke. We also have to assign the value using the time slider after which the agents start choking inside the simulation.

After detailed overview of our approach, we implement a set of physical changes to validate the working of our software model. In chapter 4, we provide results for various set of scenarios based on 3D model of Uphaar cinema fire in Delhi.

CHAPTER 4

RESULTS

In this chapter, first, we discuss various physical changes that we simulate in order to validate the working of our approach. Later we discuss, the hardware and software configurations that are used to run the simulation. At last, we provide results and compare those to the actual incident that happened in Delhi, India. These results contain calculation for time and efficiency of the simulation for that specific scenario after simulating different physical changes and other features.

4.1 Physical Changes considered for simulation

Below are some of the physical changes that are not available for commercial use and some are available but not used to their full potential. In section 4.1.1, we explain how various physical changes work. All the physical changes explained follows the flow of our methodology explained in Figure 4 in Chapter 3. First, the designing of the cinema hall model is carried out in Autodesk Maya 2018, then .Fbx file of the model including the physical changes is imported to Unity 3D for further improvements. In next step, dynamic navigation mesh is created on which all the occupants/agents move. Later, implementation of some specific features such as smoke, fire etc are done on the cinema hall model to provide a realistic scenario for evacuation. Then, scripting of the agents is done to direct them to the nearest door. This whole setup is a multi-destination setup. In the end, we simulate these physical changes separately as well as together to provide evacuation timing and efficiency for a specific type of scenario. Results for the simulation are explained in detail in section 4.3. These results provide the effectiveness of these physical changes in case of an emergency.

4.1.1 Retractable chairs

All around the world, basketball stadiums usually have retractable seats to reduce the space of the stadiums. Although, seats in the stadiums can only be retracted if the occupants have vacated the entire row or the seating area. Retractable seats in stadiums can't work even if a single occupant is sitting at any of the seats because entire section of seats is mechanically connected together to retract at the same time. However, retractable seats were never considered for cinema hall seating arrangement. Hence, for this thesis we incorporate these retractable seats in our cinema hall model with a separate set of mechanism in which it is not mandatory to vacate the entire row or the section of seats to reduce space. In this method, when a person stands up and is no longer in the proximity of the chair, it automatically retracts in the floor which gives another person to move freely around that area during the time of emergency. Due to this, as soon as the people in front row vacate, it gives extra space for people in the back row to move to their nearest exit point. We used the raycast feature for this scenario in which, the ray is cast from the chair at 45degree angle to see if some are in front of the chair if it is not it automatically retracts in the ground otherwise not (Figure 14). Thus, we designed our cinema hall model from scratch using Autodesk maya to simulate all the physical changes. First the model is designed then we imported .fbx file containing each physical change, in this case retractable seats. Hence, if any possible physical change comes in the future, our software model can incorporate that physical change easily to check its effectiveness.

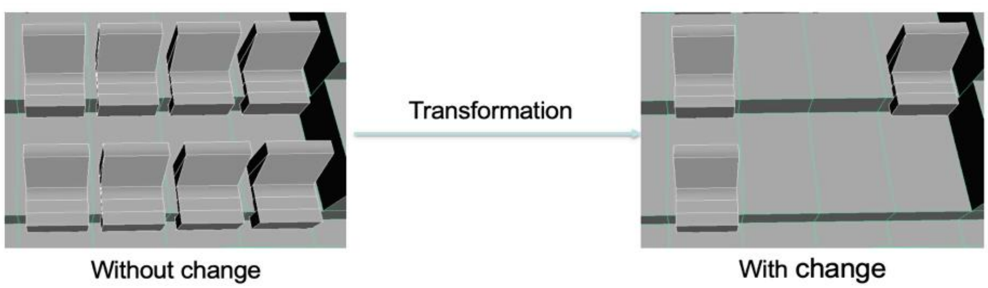


Figure 14: Retractable seats

4.1.2 Priority given to children and physically challenged people

Karamay Fire incident happened due to this, in which party officials were given priority over school children so that we can prioritize the movement of specific agents (children and slow-moving agents) over normal agents. As soon as the wheelchair reaches a door (Specific wheelchair accessible door), the normal agents move aside and give way to the physically challenged /slower moving agent or the children.

4.1.3 Extra emergency exits

Sometimes problems occur when most of the people use the same exit. Hence, by using extra emergency exits, stampede can be reduced. Similarly, for extra emergency exits we import a .fbx file incorporating the change and simulate it on the software model to check the effectiveness before it is actually built.

4.1.4 Doors with panic bars

Biggest reason behind some of the human crushes and stampedes is that the gates were not opening. In those situations, bar handles that open by just pushing can be helpful. Similarly, for doors with panic bars we import a .fbx file that incorporates the change and simulates it on the software model to check for efficiency before it is actually built.

4.1.5 Extra staff

In every cinema theatre, as soon as the movie starts, the staff scatters away leaving the hall unattended, and in certain cases, no one is there to help the people during an emergency. In such cases, staff can customers and direct them to the nearest exit (Figure 15). Likewise, for extra staff we import a .fbx file incorporating the change and simulate it on the software model to check the effectiveness before it is actually built.

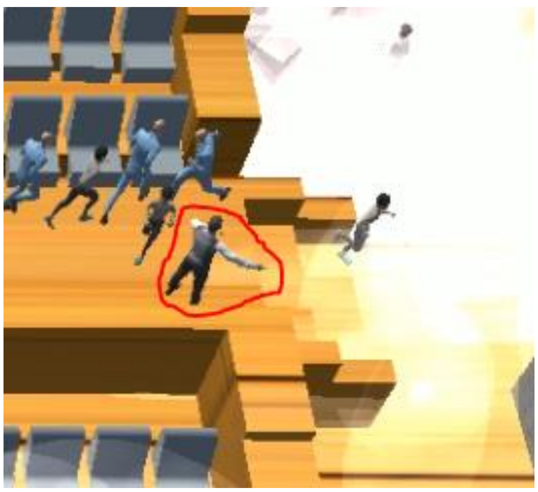


Figure 15: Assisting staff [47]

4.1.6 Glow in the dark paint

During a fire, at times the light sometimes gets cut off, which can lead to blackout in the hall. People cannot see their nearest exit. However, glow in the dark paint on the floor can allow people to find the path to exit (Figure 16). It reduces chaos and increases the efficiency of the system. Glow in the dark paint stripes can also be seen on the aisle of an airplane which directs the passengers to the nearest exit. Similarly, for glow in the dark paint we import a .fbx file that incorporates the change and simulates it on the software model to check for efficiency before it is actually build.

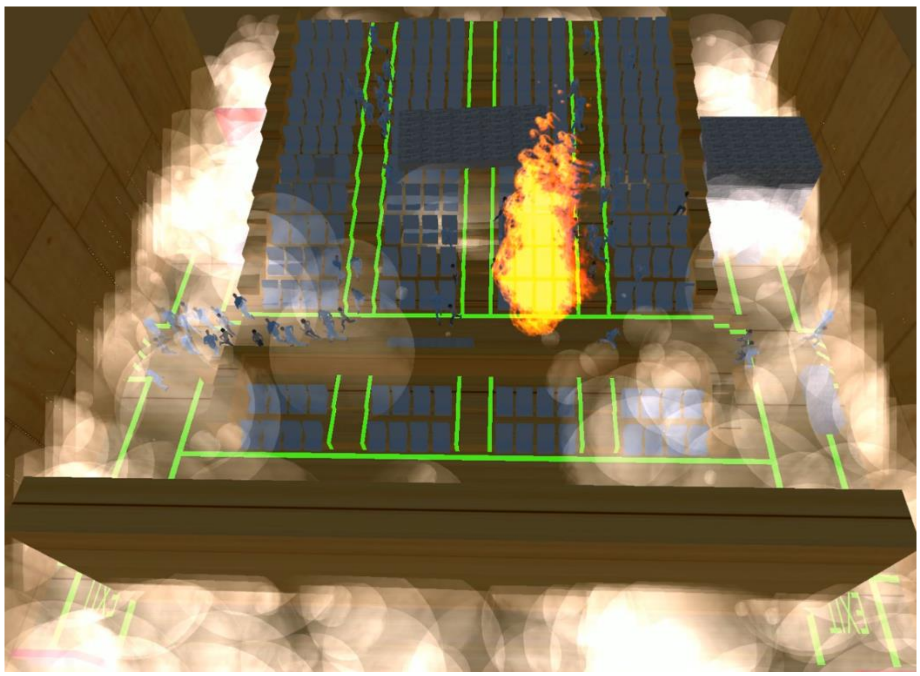


Figure 16: Glow in the dark paint [48]

4.1.7 Movable walls

During emergency evacuations most of the hallways and exits creates a bottleneck situation which leads to casualties due to stampedes. All the cinema halls around the world have maximum of 4-6 exits with a width of 2m, which is not enough for a crowd of 300-400 people rushing towards the exits at the same time. Also, the exits in a cinema hall are placed at the farther ends which increases the time of evacuation. Hence, for this thesis we incorporated set of movable walls due to which occupants don't have to rush to the find the exit and move towards it. In this process the sidewalls of the cinema halls slide from the middle creating a passage to escape the cinema hall, thus reducing the effort and time for occupants to reach to the exits even at farther ends. In order to reduce human crushes at that time, it is possible to make a structure with movable sidewalls so that people can move freely (Figure 17). Similarly, for movable walls we import a .fbx file incorporating the change and simulate it on the software model to check the effectiveness before it is actually built.

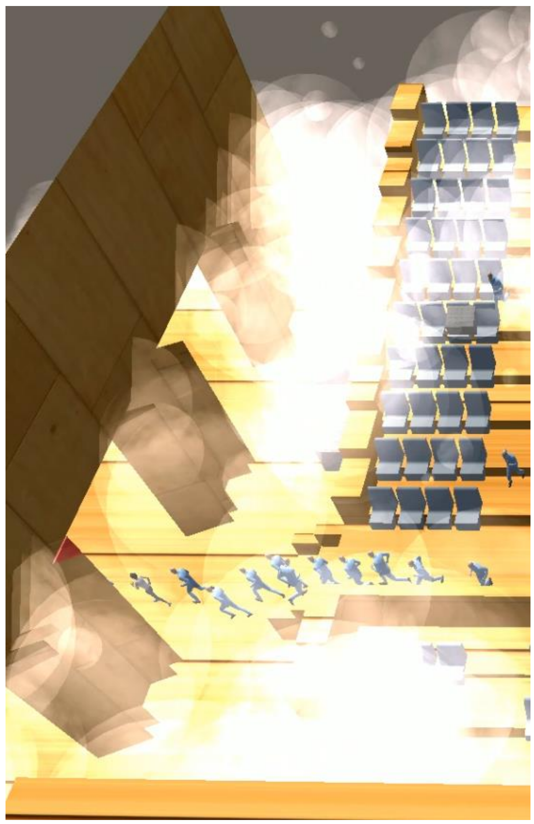


Figure 17: Movable walls [49]

After trying to implement these Physical changes with a smaller crowd, we tried to make a model to accommodate 3000-5000 people at a single time at a building depicting University of Windsor's St. Denis centre during convocation. We provided the environment with the same set of rules and protocols that we provided in the cinema hall model, such as fire, crowd distribution, wheelchairs etc. As the crowd increases the computation power as well as processing power increases.

Below are the images depicting the organization of St. Denis centre model (Figure 18 & 19). These include the alignment of chairs throughout the arena, placement of doors, stage, etc.

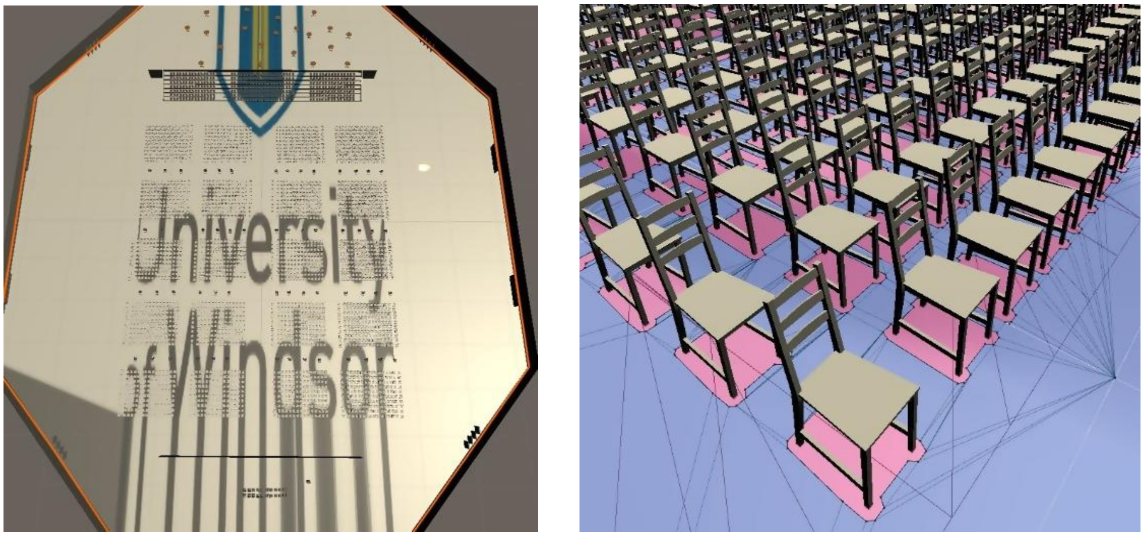


Figure 18: St. Denis arena and walkable mesh between chairs

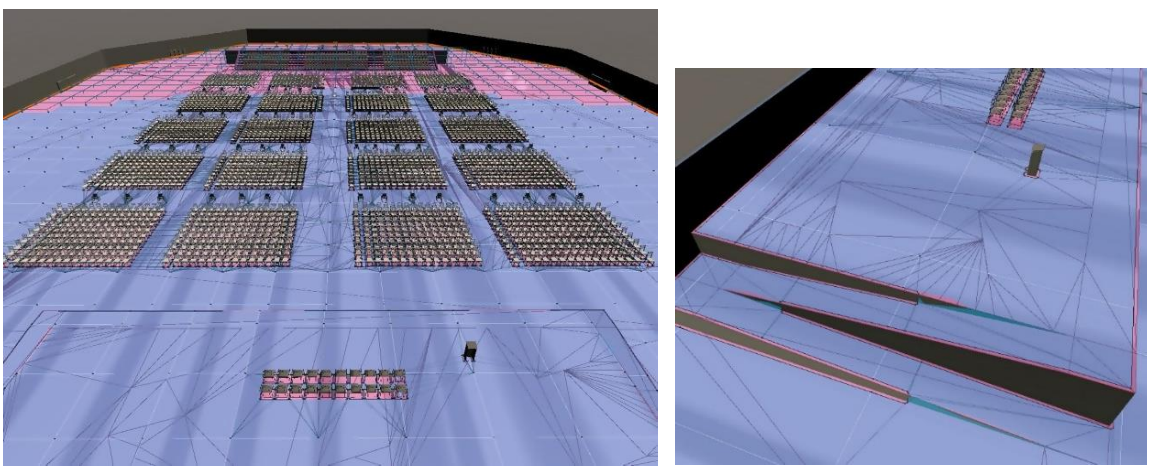


Figure 19: Walkable and non-walkable mesh in full scale St. Denis arena

It should be noted that other software models cannot import any file to add a new 3D environment into their software package. Therefore, if any possible physical change and new feature/parameter in technology comes up in the future, it cannot be implemented on the software model. However, our approach can work with any possible physical change before they are actually constructed/built or that might come in future, which can be easily imported and implemented with minor configurational handling. By proposing a set of physical changes and various other features to make our model work in any scenario to overcome drawbacks of other software models.

4.2 Hardware and Software configurations

We developed and tested our software model on a Windows computer. It contains 3.3 GHz Intel® Core i7™ CPU with 32 GB DDR3 memory and a 4 GB NVIDIA GeForce GTX 960 graphics card. Our software model's primary goal is to provide realistic crowd simulation for emergency situation under the display resolution of 2560 X 1440 pixels.

We developed our software model using certain software's which can work across multiple platforms:

- Unity 3.1.4f1 Personal (64 bit)
- AutoDesk Maya 2019 (Educational version)

4.3 Experiments and Results

In this section, we discuss the experiments we performed on our cinema hall model using a set of scenarios incorporating certain physical changes and comparing it with an actual event that happened in India. Later, we record the results and briefly conclude if the physical changes were effective or not. This section is arranged in the order below:

1) First, we discuss the arrangement of general cinema hall setting and results in case of no emergency.
2) We simulate a scenario similar to an actual incident that happened in India for comparison of results without taking electrical short-circuit into account. We consider this as scenario 1 for simulation.
3) We consider all the results of scenario 1 as default results for comparison.
4) We simulate working of retractable seats and results associated with it to show its effectiveness in case of an emergency.
5) We simulate working of movable walls and results associated with it to show the effectiveness in case of an emergency.
6) We simulate a mix and match strategy in which simulation depicts working of retractable seats and movable walls together, and results associated with it to show the effectiveness in case of an emergency.

7) If we compare to the actual event, there was a pitch-black environment on that day. In order to recreate the environment, we simulate a scenario with electrical short-circuit for comparison of our results. We consider this as scenario 2 for simulation and also the results of this scenario are considered default for further experimentation of other physical changes.
8) We simulate working of glow in the dark paint stripes and results associated with it to show its effectiveness in case of an emergency.
9) We simulate working of Extra staff and results associated with it to show the effectiveness in case of an emergency.
10) We simulate a mix and match strategy in which simulation depicts working of glow in the dark stripes and extra staff together, and results associated with it to show the effectiveness in case of an emergency

To provide results, we designed our cinema hall model from scratch using Autodesk Maya 2019 and Unity 3D. This model includes seats for 192 agents, with four exit doors. We created 101 agents for this seating arrangement, including five wheelchairs for physically challenged people in the front rows.

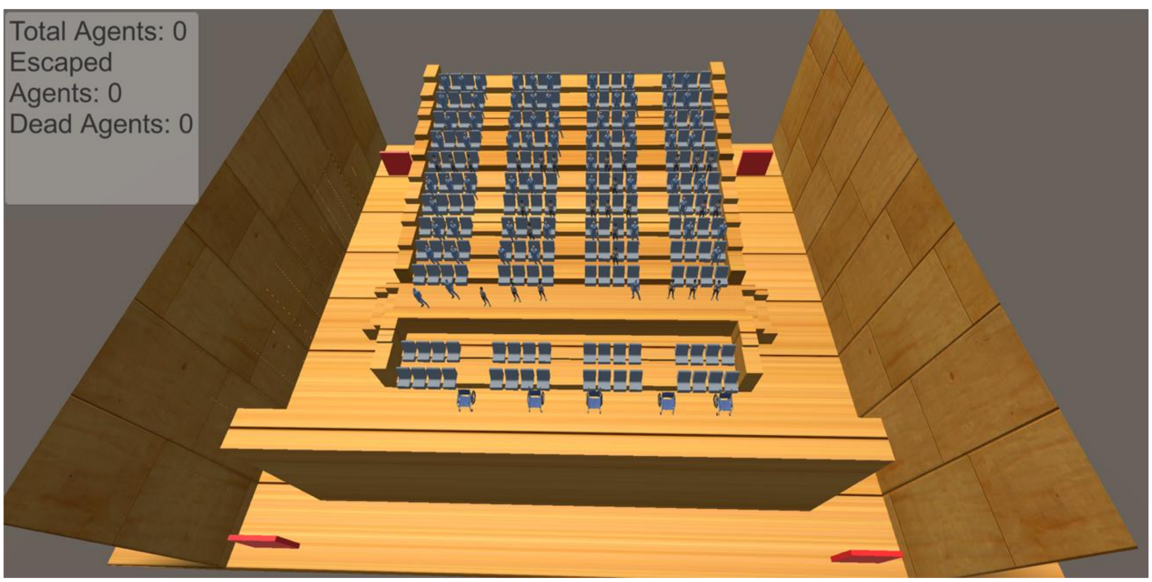


Figure 20: Default cinema hall's setting in unity

We recorded the evacuation time in the absence of any emergency situation such as fire, smoke, blocked exit etc. The results label (top left corner) provide various set of parameters such as total number of agents considered for simulation, number of agents that could escape from the environment without any harm, number of dead agents at the end of simulation and time in seconds till the last agent could escape or die based on the specific scenario. Also, there are 3 parameters to indicate cause of death for certain number of agents.

Death by fire: This parameter accounts for the deaths of agent who were seated close the fire or running near the fire in the simulation environment. This parameter simulates the death of agents caused by burning.

Death by roof (falling debris): This parameter accounts for the death of agent caused by a falling roof/debris directly on their head. Since blocks from the roof could fall anywhere inside the environment.

Death by choking: This parameter accounts for the deaths of agents due to excessive smoke inhalation or any other toxic material. This phenomenon leads to choking and agent dies within few seconds in the simulation environment.

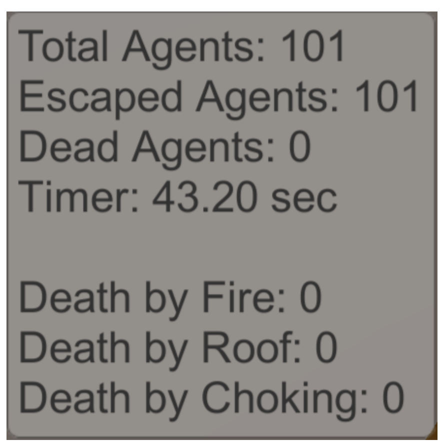


Figure 21: Default timing result for evacuation (No Emergency)

In our first experiment with no emergency situation, it took all of 101 agents 43.20 seconds to get out of the cinema hall. Taking this into consideration, we further check results by gradually increasing the complexity (adding different features one by one) of the simulation.

In order to provide results for simulation, it should be comparable to an incident that happened in the past. For our cinema hall model, we are comparing that model with the Uphaar cinema fire tragedy [7], one of the worst fire tragedies in recent Indian history that took place at Uphaar Cinema, Green Park, Delhi, on Friday, 13 June 1997. In this incident, 59 individuals who remained trapped inside died, primarily due to suffocation, choking, fire, blocked exits, pitch black, environment etc. To provide a similar perspective, we implemented various features and parameters that remain the same while applying our proposed methodology.

The features and parameters are:

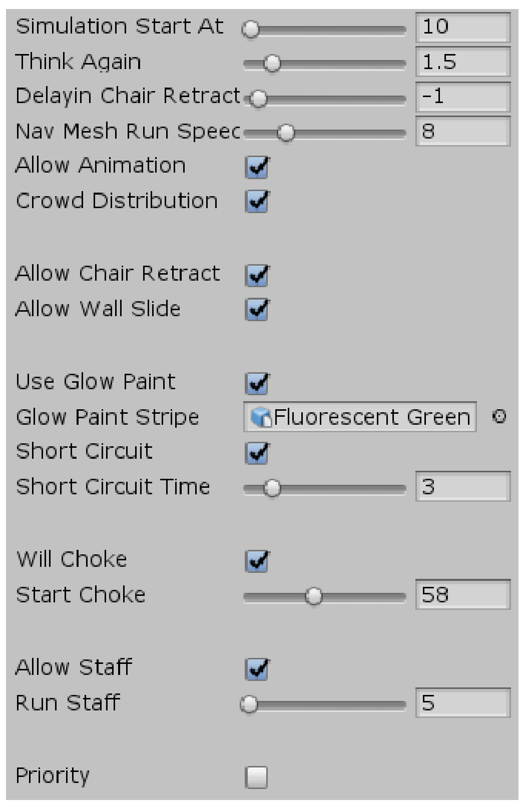


Figure 22: Parameter accessibility pane in unity

Think again time (the value provides the number of frames after which the AI starts recalculating the path to find the nearest exit): 1.5 frames

Delaying Chair retract (default time when the first chair retracts): -1sec

Allow animation (enables the animation sequence for running, moving, falling, etc.) : checked

Crowd distribution: checked

Think again time is take as 1.5 frames as agent calculates the path to the nearest exit after every 1.5 frames. we tried various values of think again between 0.4 – 2 frames, but 1.5 frames provided us with best results.

Similarly, in case of Delay in chair retract, the default time considered is -1 as it allows the empty chairs to retract before the simulation start. Thus, depicting a realistic scenario.

[We explained all of these parameters in detail in Chapter 3]

Other parameters and features such as simulation start at, allow chair retract, allow wall slide, use glow paint, short circuit, short circuit time, will choke, start choke, allow staff, run staff and Priority varies according to the scenario.

Figure 23 depicts similarities with Uphaar cinema on that day. There was smoke all around, fire at certain stages, some of the exits blocked and falling roof on top of people, debris everywhere.

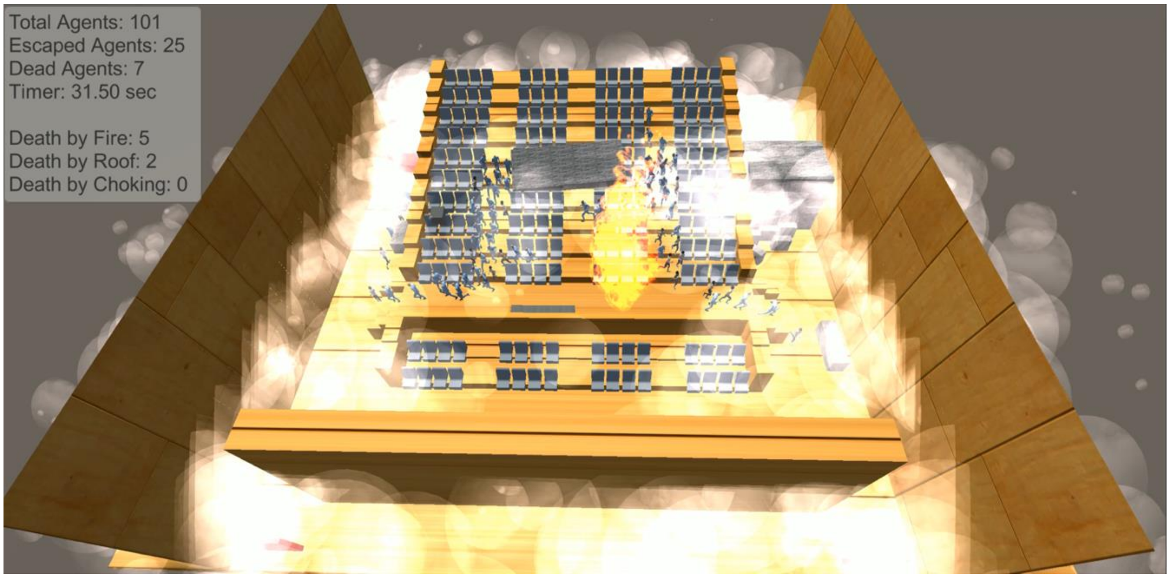


Figure 23: Simulation depicting similarities with Uphaar Cinema

As stated earlier, 59 people died on that day due to above-mentioned reasons. We have considered the similar percentage of alive agents to dead agents as compared with the scenario in Uphaar cinema. After setting all the exact parameters, our software model simulation provided similar results.

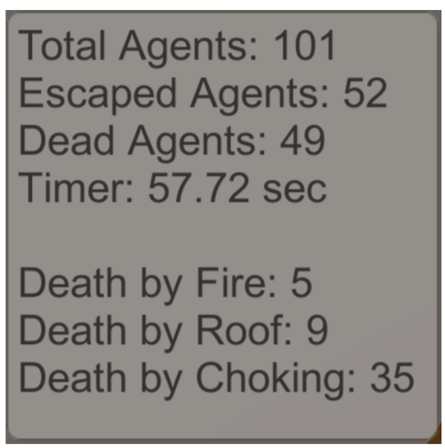


Figure 24: Results for default scenario without electrical short-circuit (scenario 1)

The total number of agents we used are 101, out of which 52 escaped within 57.72 sec. 49 agents died, which can be sorted by the type of death.

As shown in Figure 24, 5 agents died being in the vicinity of the fire. Also, 9 agents got crushed from the concrete blocks that fell out from the ceiling and 35 agents engulfed by smoke, did not get out on time and choked to death.

Now, we consider this as a default scenario for further comparison with our proposed methodology, including certain physical changes which might affect the timing and efficiency of evacuation. Thus, the timing to compare with other results is considered 57.72 seconds in which 49 agents died, helping us finding the efficiency of our proposed software model. We consider the scenario without electrical short-circuit as “scenario 1”.

Now, we consider each physical change or feature one by one that has been discussed in Chapter 3. First, we briefly explain how the change or feature works. Then, we provide evacuation timing and efficiency in the same format to reduce confusion. At the end of each result, we provide a conclusion for the same.

Retractable seats:

First Physical change that we consider is retractable seats.

It works on the basic idea that if no one is in vicinity of the seat, the chair will automatically retract in the floor, thus providing extra space for agents to move around in the cinema hall.

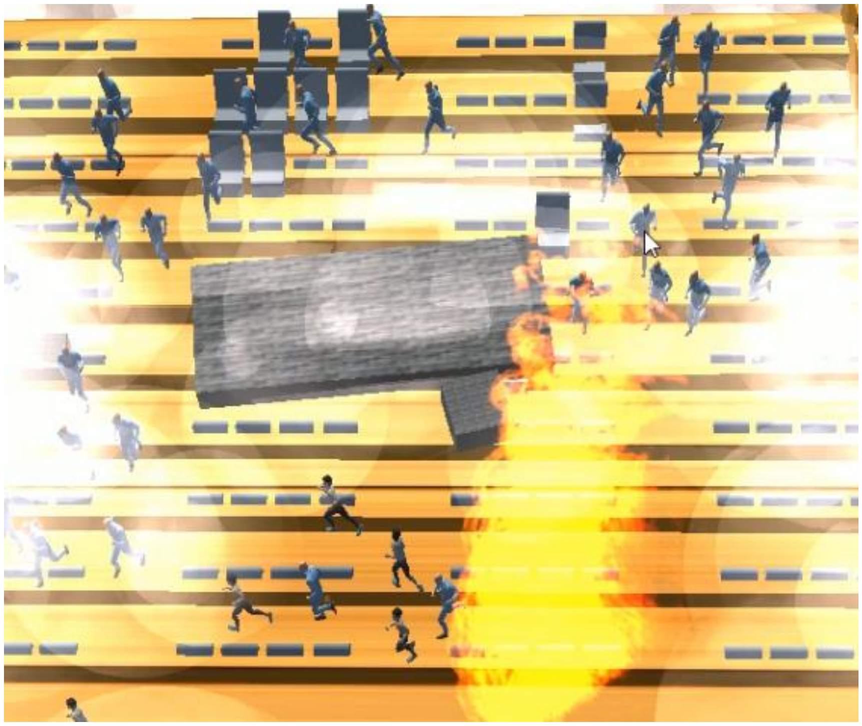
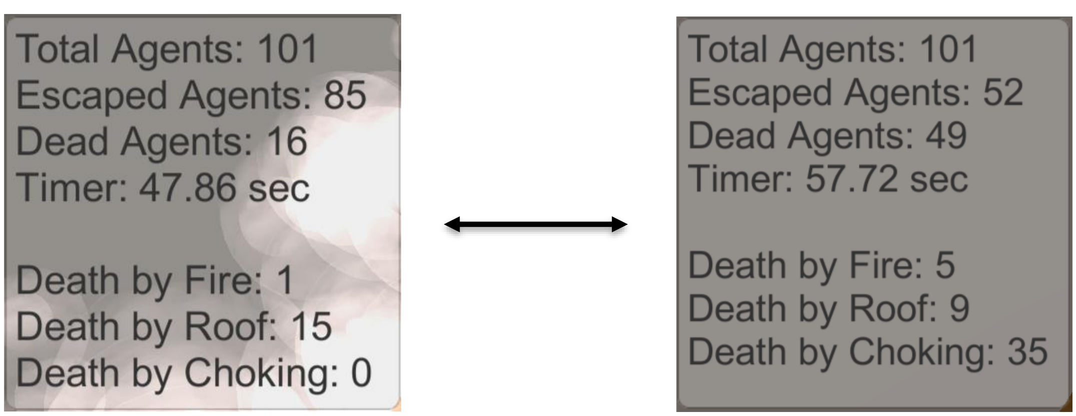


Figure 25: Simulation using retractable seats

As depicted in Figure 25, the agents are moving to the nearest exit and most of the seats have been retracted to the floor. However, some of the seats are still up because agents are still moving around them and even without the seats, the agents can still move diagonally, thus reducing the time to evacuate.



(a) (b)

Figure 26: Comparison of (a) results of simulation using retractable seats with (b) results of scenario 1

In this simulation, only 16 agents died, and remaining 85 agents took 47.86 seconds to get out safely, still some of the agents died due to fire and falling roof/debris. More importantly, the agents got out before they get chocked due to smoke, mainly due to time saving retractable seats.

By comparing results in figure 26, we can conclude that there is 17.08% reduction in evacuation time. Similarly, the efficiency of the system increased by 67% because 33 lives are saved compared to scenario 1 (scenario without electrical short-circuit).

Moving walls:

The next physical change that we implemented is the moving walls after unchecking the retractable seats. Therein, the sidewalls of the cinema hall slide from the middle creating a passage to escape the cinema hall, thus reducing the effort and time for agents to reach to the exits even at farther ends.

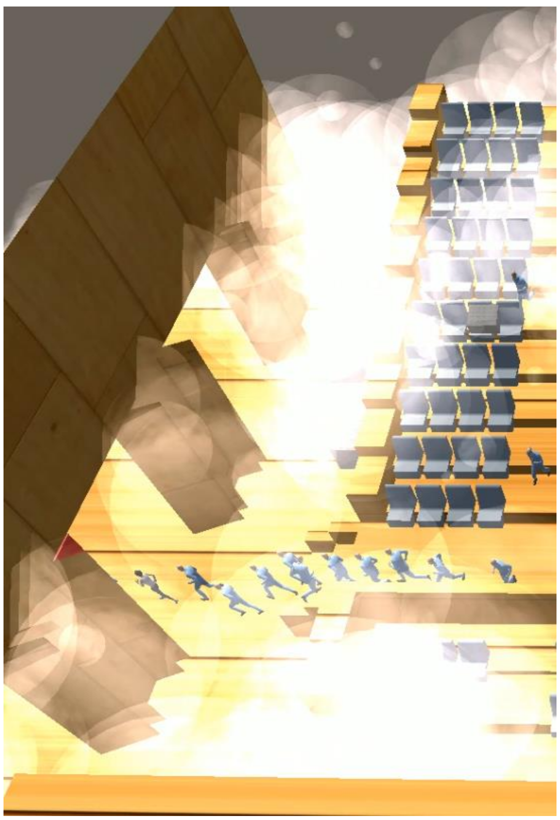
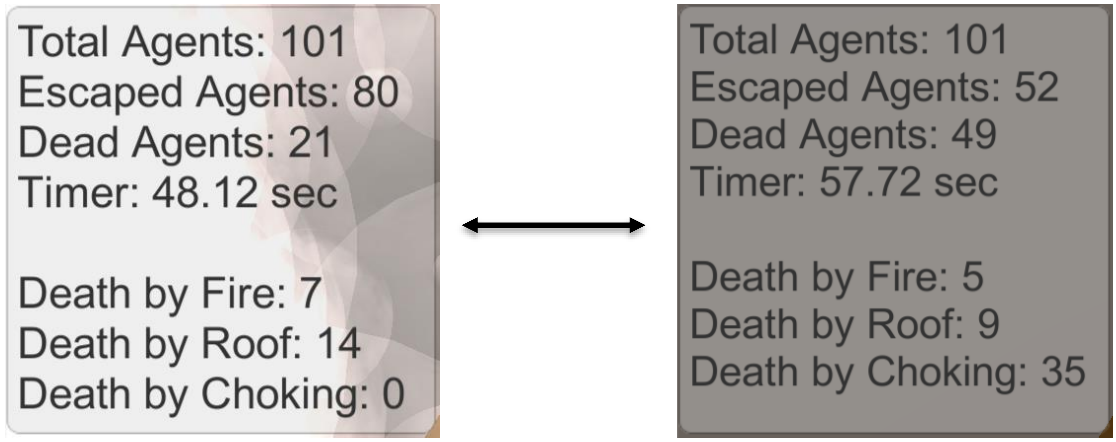


Figure 27: Simulation using Movable walls

As depicted Figure 27, the agents are moving to the passage created by the sliding walls. Thus, they can exit the cinema hall quickly, rather than searching the nearest exit.



(a) (b)

Figure 28: Comparison of (a) results of simulation using movable walls with (b) results of scenario 1

Simulation indicate that only 21 agents died in this case whereas, 80 agents were able to get out only in 48.12 seconds. Death of agents was primarily due to fire and falling roof. The sliding walls played a vital role in reducing time and effort, thus choking lead to 0 deaths as the agents got out before choking starts.

By comparing results in Figure 28, we can conclude that there is 16.63% reduction in time of evacuation. Similarly, the efficiency of the system increased by 57.14% because 28 lives are saved compared to scenario 1 (Scenario without electrical short-circuit).

After implementing these two physical changes, we tried a mix and match strategy using both the retractable seats and the moving walls.

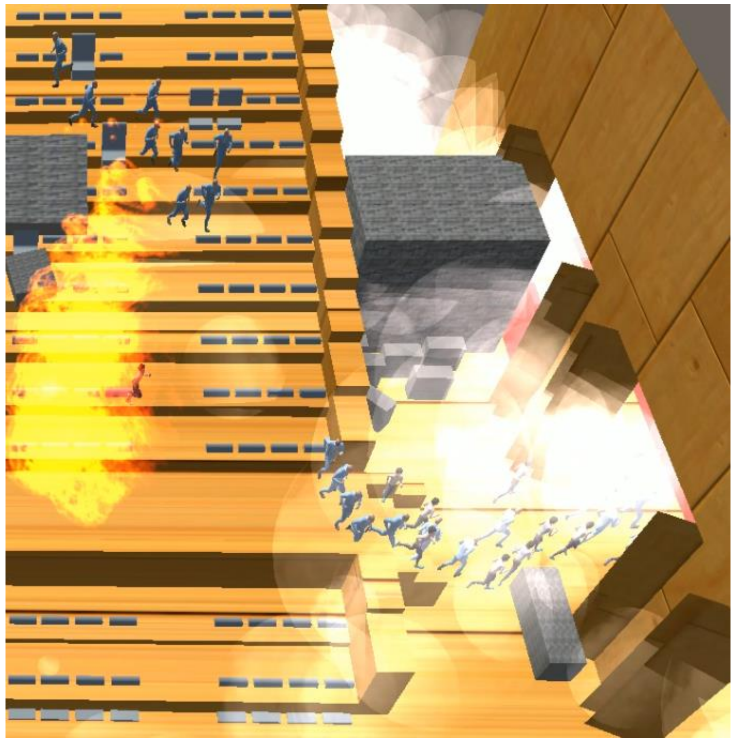
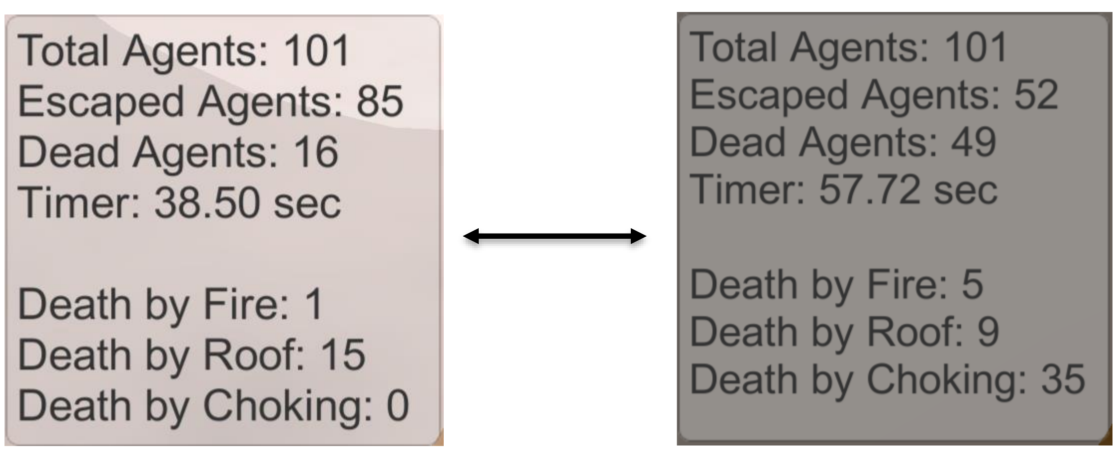


Figure 29: Simulation using both Retractable seats and movable walls

As shown in Figure 29, retractable seats and sliding walls both are working at the same time. This case provides us a more prominent results compared to the earlier discussed using only one change at a given time.



(a) (b)

Figure 30: Comparison of (a) results of simulation using both retractable seats and movable walls with (b) results of scenario 1

Combined result shows that only 16 agents died, and it took 38.50 seconds for 85 agents to get out safely, still some of the agents died due to fire and falling roof. By comparing results in Figure 30, we can conclude that by using both of the physical changes together, we can significantly reduce the time of evacuation, which came out to be 33.23 % less. The efficiency of the system remained equal as compared to the scenario where we used retractable seats only, which was 67%.

Results of simulation using retractable seats and movable walls as compared with scenario without electrical short-circuit are discussed in Table 5.

<u>Original scenario without electrical short-circuit (scenario 1):</u>

Physical Changes	**Time of evacuation (Reduced) %**	**Efficiency of system (Increased) %**
Retractable seats	17.08	67.00
Movable Walls	16.63	57.14
Retractable seats + Movable Walls	33.23	67.00

Table 5: Time and Efficiency results of scenario without electrical short-circuit

In Table 5, we provided a summary of reduction in time of evacuation and increased efficiency of the system for physical changes such as retractable seats and movable walls. We also managed to record results for our mix and match strategy for these two physical changes.

As mentioned earlier, there was a pitch-black environment at Uphaar cinema that day. In order to recreate the environment, we introduced a new scenario on the cinema hall model in which light went off due to electrical short circuit, which in turn made it hard for agents to find the nearest exit.

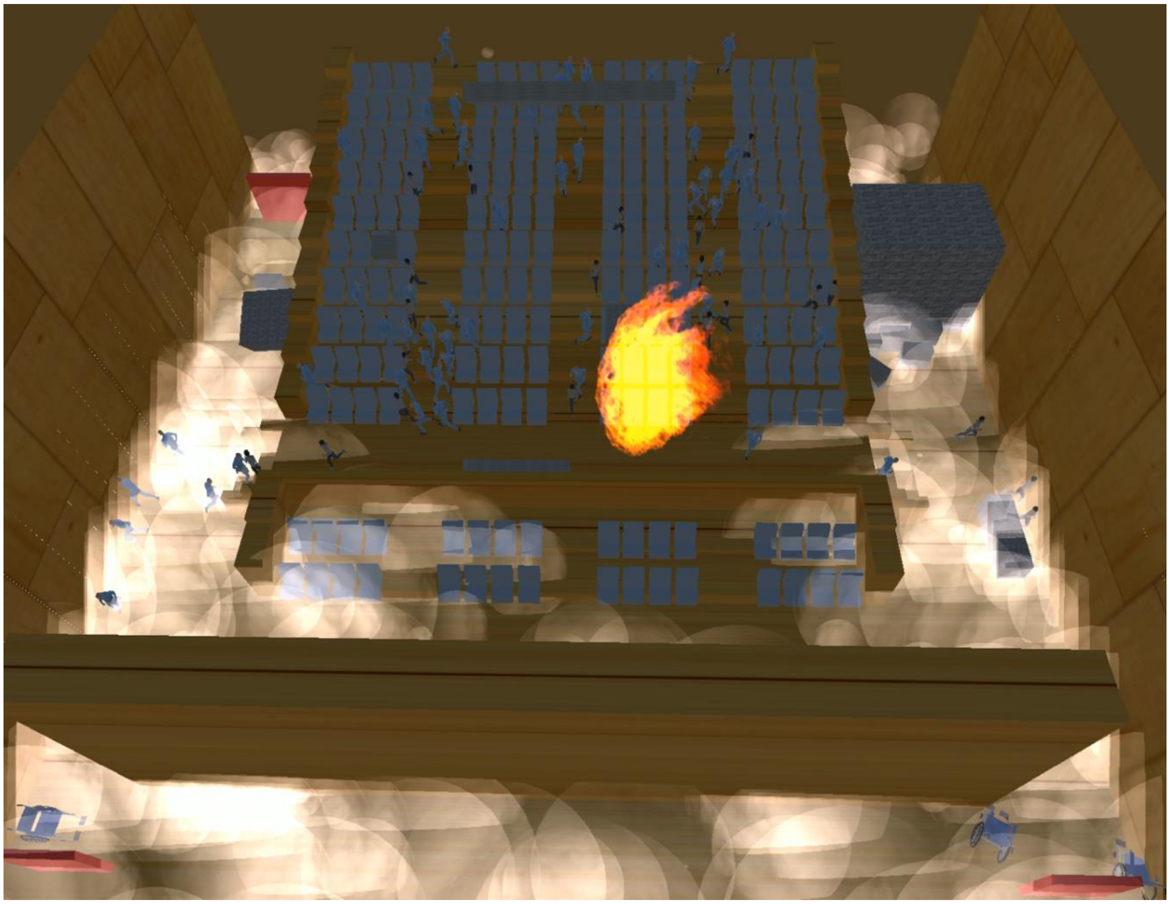


Figure 31: Simulation depicting similarities with Uphaar Cinema in case of electrical short circuit

We put the short circuit time to 3 seconds, so that the short circuit occurs after 3 seconds when the simulation starts. As Figure 31 depicts, a dark environment in addition to the previous scenario with fire, smoke, blocked exits etc.

If we notice carefully, the agents in this scene are moving randomly as they do not know the directions to the nearest exit, which is similar to the situation of Uphaar cinema. Shockingly only a very small number of agents were able to find their way out of the cinema hall, and most of them died inside.

Figure 32: Results for default scenario with electrical short-circuit (scenario 2)

Result provides information that 95 agents died and only 6 agents were able to get out safely. The agents died due to fire and falling roof. However, smoke made it worse, 76 agents died because of choking, due to their inability to find directions to the nearest exit.

Now, we consider this scenario with electrical short-circuit as scenario 2 and provide some physical changes to obtain better results in this type of situation. We considered a time delay of 3 seconds for the time when choking starts to provide a much more realistic scenario in a dark environment.

Glow in the dark paint:

First, we consider glow in the dark paint. The main idea behind this technique is that it illuminates the path to the nearest exit whenever there is a power outage. This physical change can be observed on the isles inside an airplane. Similarly, the painted stripes on the floor of the cinema hall glow as shown below.

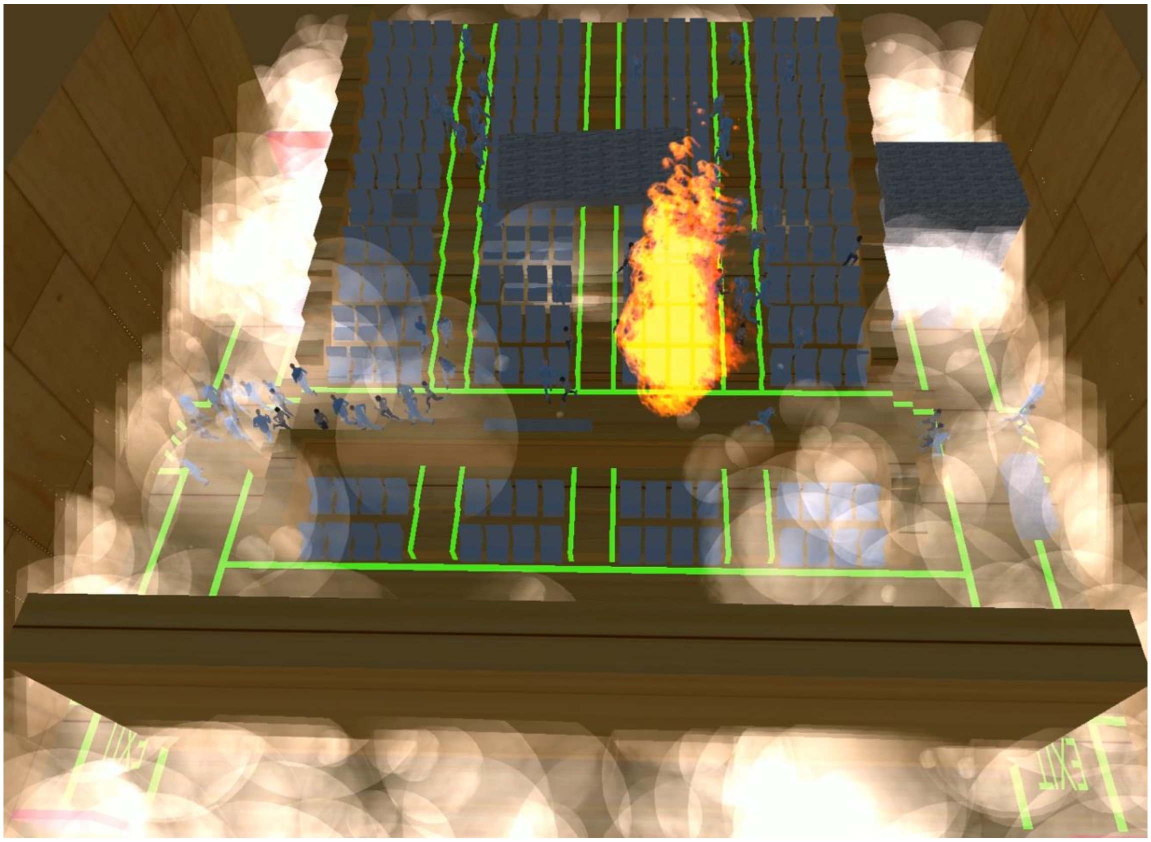
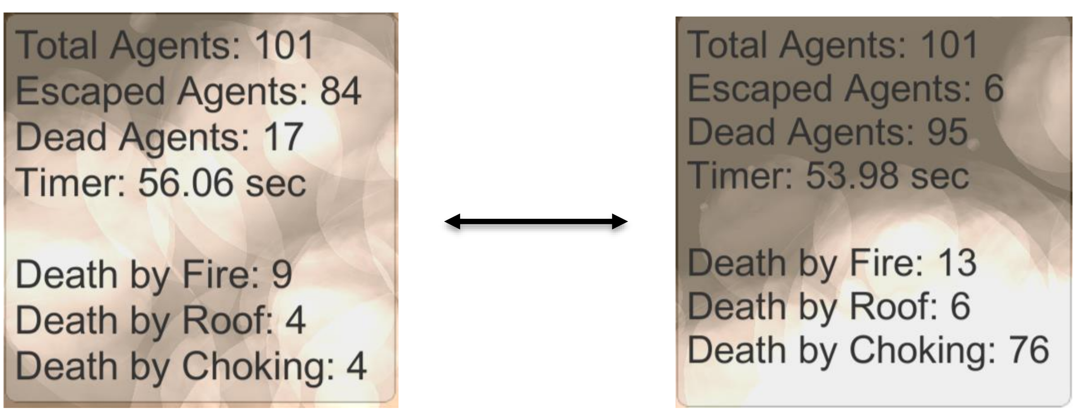


Figure 33: Simulation using Glow in the dark paint stripes

As Figure 33 depicts, the scenario remains the same, nevertheless the painted stripes are glowing on the floor. If we observe carefully, the stripes are directing the agents to the nearest exits, thus reducing their efforts to find the exit in a dark environment.



(a) (b)

Figure 34: Comparison of (a) results of simulation using glow in the dark stripes with (b)results of scenario 2

Results of this simulation indicate that only 17 agents died, and it took 56.02 seconds for 84 agents to get out safely. Still some of the agents died due to fire and falling roof. Glow in the dark paint proved vital for agents to get out safely, leading to only 4 deaths due to choking. The agents are able to get out rather than moving around randomly inside to find the exits.

By comparing results in Figure 34, we can conclude that there is 0.04% increase in time of evacuation as it took time to follow the path to the exit. However, the efficiency of the system is increased by 82.10% because 78 lives got saved compared to scenario 2 (scenario with electrical short-circuit)

Extra staff:

The next physical change that we implemented is extra staff after unchecking glow in the dark paint. As we know the staff of any cinema hall has better knowledge about the entry and exit points, etc. Thus, to direct the confused agents during a blackout, extra staff can be very helpful. Extra staff uses the feature of counterflow as the staff is moving in the opposite direction of the crowd movement which we explained in detail in Chapter 3.

We used two agents to move to specific spots and direct the agents to the nearest exit and leave the cinema hall after everyone gets outs. Thus, to provide a realistic scenario, this change also needs a time delay as agents have to listen to the staff and then run towards the nearest exit.

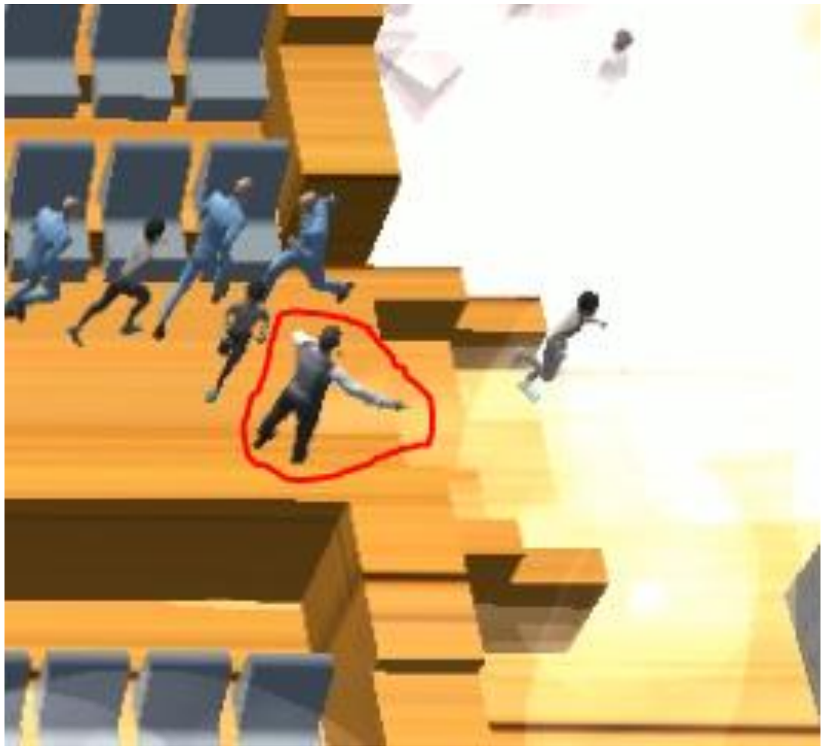


Figure 35: Simulation using Extra staff

As depicted in Figure 35, the staff is directing the agents out to the nearest exit by using animated hand gestures.

(a) (b)

Figure 36: Comparison of (a) results of simulation using extra staff with (b) results of scenario 2

Results indicate that only 38 agents died, and it took 59.84 seconds for 63 agents to get out safely. Some of the agents died due to fire and falling roof. As compared to glow in the dark paint, extra staff change is not that effective, although it reduces casualties with respect to the original scenario. In this scenario, 30 agents died because of choking. These results lead us to the conclusion that there is a 0.10% increase in time of evacuation as it took time to understand to the staff. Although, the efficiency of the system increased by 60% because 57 lives got saved compared to scenario 2 (scenario with electrical short-circuit).

After implementing these two physical changes, we again tried a mix and match strategy as it gave us much better results. For this scenario we used a combination of both glow in the dark paint and extra staff.

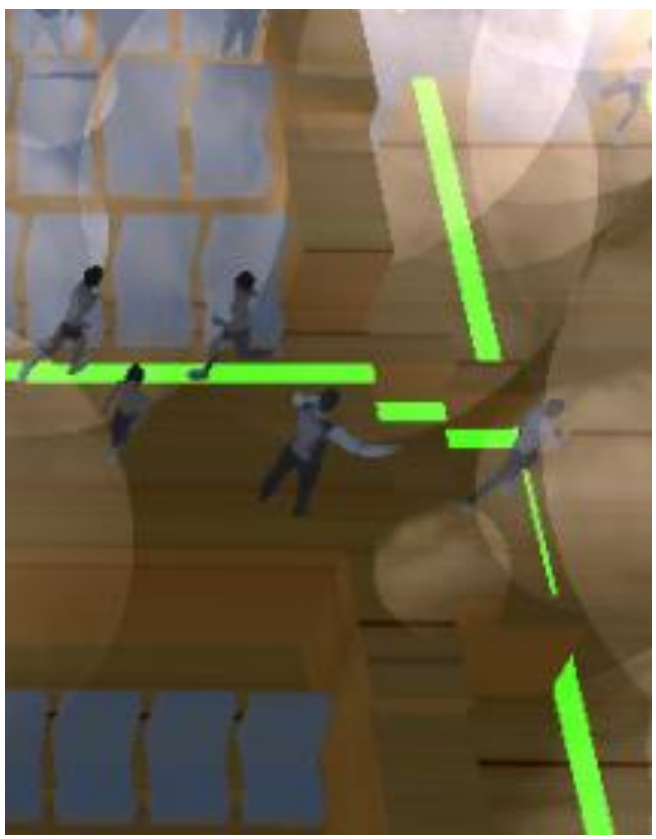


Figure 37: Simulation using both glow in the dark paint stripes and extra staff

As Figure 37 depicts, there is a staff member directing them to the exit and they are also taking help of the glow path.

(a) (b)

Figure 38: Comparison of (a) results of simulation using both glow in the dark paint stripes and extra staff with (b) results of scenario 2

Combined result shows that only 34 agents died, and it took 59.66 seconds for 67 agents to get out safely. However, some of the agents died due to fire and falling roof. By comparing results in Figure 38, we can conclude that by using both of the physical changes, we can increase efficiency of the system, which came out to be 64% more. The time of evacuation is 0.06% less.

Results of simulation using glow in the dark stripes and extra staff as compared to scenario with electrical short-circuit are discussed in Table 6.

<u>Original scenario with electrical short-circuit (scenario 2):</u>

Physical Changes	**Time of evacuation (Increased) %**	**Efficiency of system (Increased) %**
Glow in the dark paint	0.04	82.10
Extra Staff	0.10	60.00
Glow in the dark paint + Extra Staff	0.06	64.00

Table 6: Time and Efficiency results of scenario with short circuit

In Table 6, we provide a summary of increased time of evacuation and increased efficiency of the system for physical changes such as glow in the dark paint stripes and extra staff. We also managed to record results for our mix and match strategy for these two physical changes.

Apart from all these features mentioned, we also implemented a Priority system for physically challenged individuals and children. In the simulation when a normal agent encounters a child or a physically challenged agent, it moves aside and give a way to them.

Our software model also has a default crowd distribution parameter. It provides a distributed arrangement of moving crowd, if any exit is too much crowded for an agent, it finds a way to next nearest exit.

Additionally, both of these features help in reducing the time of evacuation and increasing efficiency of the system. However, change in result due to crowd distribution and priority system was not prominently visible in order to show them separately in tables. We can also show the toxicity of the environment without showing the smoke and still work with the choking feature. This scenario can work in case of biochemical hazard, in which gases are not visible, and still agents are affected in large numbers.

We also modeled a large arena based on University of Windsor’s convocation ceremony - a closed ceremony in the athletic auditorium, housing $3000 - 4000$ guests and students. Figure 39 is a general depiction of the scenario to accommodate such a large crowd.

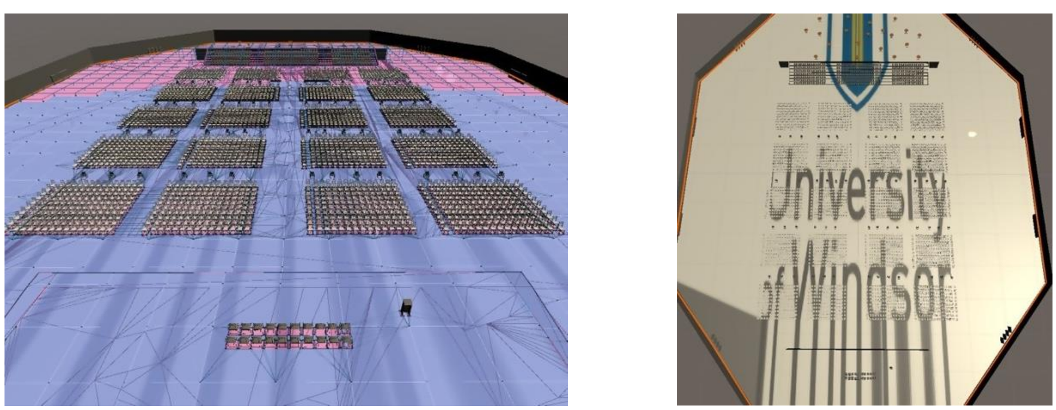


Figure 39: Walkable and non-walkable mesh in full scale St. Denis arena

All of these features or physical changes that we described, provided us with promising results that are missing from the models described earlier. Through a suitable computing environment, one can simulate large crowds and observe the effect of physical changes and environment. Our software is capable of reproducing phenomena that have been observed in the past in current evacuation scenarios for research and to observe the effects of new features or physical change for same scenarios to assess the impact of these physical changes before they are actually constructed/built.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this thesis, we worked on implementing a simulation of an advanced emergency evacuation system. Our implementation is based on the limitation of the existing models that are used throughout the world. We also considered a real-life scenario of Uphaar cinema fire happened in 1997 in Delhi, India. Thus, we designed a cinema hall model to provide a similar realistic effect using Autodesk Maya 2019 and Unity 3D. We applied various default features and parameters that happened during that time. After that, we simulated various proposed methodologies or physical changes inside the cinema hall model.

We worked with two sets of scenarios: scenario without short circuit and scenario with a short circuit. Both of the scenarios include some common features and some separate physical changes. While there was no electrical short circuit, we implemented the retractable seats and movable walls physical change, which reduced the time of evacuation exponentially. Also, in case of scenario with an electrical short circuit, we implemented glow in the dark paint and extra staff, which highly enhanced the efficiency of the system.

Both of these scenarios provided us with promising results and allowed us to conclude that if these physical changes were there on that day, many lives could have been saved. Thus, our software is capable of reproducing phenomena that have been observed in the past for research purposes and to observe the effects of new features or physical change for same scenarios to assess the impact of these physical changes before they are actually constructed/built. The software model is capable of integrating new features as well as possible physical changes in a 3D indoor environment by importing external files using .fbx file format. The software model can also simulate various level of details and crowd densities, even more massive crowds. The proposed software model is capable of dealing with a large number of parameters that can be used to simulate various scenarios and multiple environments with a large number of agents. This can allow researchers/planners to observe the effects of various situations in the design/planning phase without the investment of time and money and the effect of those physical changes to save human capital.

5.2 Future work

- Working with large crowds is always a time-consuming task. Thus, with more computation power, our system can handle crowd up to 3000-5000 agents inside St. Denis Athletic Arena during convocations. With large crowd, the complexity increase, and if we implement physical changes one by one, it will take more computation power to tackle the dynamic navigation mesh with all the obstacles and features.
- In our model, the rigging and animation of agents are done for the walking and running aspect only; thus, they are standing at a specific position. we can also work with agents sitting on seats and standing up later on to move the nearest exit.
- There can be any number of possibilities for features and parameters, however our software model can handle files imported from other platforms in any of the unity compatible format. Thus, providing flexibility to other fields to work with our software model flawlessly.
- As virtual reality is gaining heights in computer science, we can also implement a virtual environment and check the results with the help of that perspective. As both the systems can handle .Fbx file format, it would be great to see this advancement in the foreseeable future.

REFERENCES/BIBLIOGRAPHY

[1] Barham, R. (2006). Fire Engineering and emergency Planning: Research Application. Routledge.

[2] "China Orders Safety Study After Fire Kills 300". *The New York Times*. 10 December 1994. p. 3. Retrieved 12 February 2012.

[3] *"Fire rips through crowded Brazil nightclub, killing 233". CNN. Retrieved 28 January 2013.*

[4] Parker, Paul Edward (December 3, 2007). "Tally of a tragedy: 462 were in The Station on night of fire". *The Providence Journal*.

[5] Delhi cinema fire tragedy claims 59 *Indian Express*, Saturday, 14 June 1997.

[6] Zia ur-Rehman; Declan Walsh & Salman Masood (12 September 2012). "Pakistan Factory Fires Kill More Than 300". *The New York Times*. NYT Asia Pacific. Retrieved 16 September 2012.

[7] Jason Tsai, Natalie Fridman, Emma Bowring, Matthew Brown, Shira Epstein, Gal Kaminka, Stacy Marsella, Andrew Ogden, Inbal Rika, Ankur Sheel, Matthew E. Taylor, Xuezhi Wang, Avishay Zilka, and Milind Tambe. 2011. "ESCAPES: evacuation simulation with children, authorities, parents, emotions, and social comparison" In The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2 (AAMAS '11), Vol. 2. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 457-464.

[8] Nuria Pelechano, Ali Malkawi, "Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches" in Automation in Construction, Volume 17, Issue 4, 2008, Pages 377-385.

[9] Glynn, C. J., Herbst, S., Lindeman, M., O'Keefe, G. J., and Shapiro, R. Y. (2015). Public opinion. Westview Press.

[10] Guy, S. J., Chhugani, J., Kim, C., Satish, N., Lin, M., Manocha, D., and Dubey,P. (2009). Clearpath: highly parallel collision avoidance for multi-agent simulation. In Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 177{187. ACM.

[11] Weiss, T., Litteneker, A., Jiang, C., and Terzopoulos, D. (2017). Position-based multi-agent dynamics for real-time crowd simulation. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, page 27. ACM.

[12] van Toll, W., Cook, I., Atlas, F., van Kreveld, M., and Geraerts, R. (2016). The explicit corridor map: Using the medial axis for real-time path planning and crowd simulation. In LIPIcs-Leibniz International Proceedings in Informatics, volume 51. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[13] Helbing, D. and Molnar, P. (1995). Social force model for pedestrian dynamics. Physical review E, 51(5):4282.

[14] https://www.anylogic.com/use-of-simulation/agent-based-modeling/

[15] S. Gwynne, E.R. Galea, M. Owen, P. Lawrence, L. Filippidis, A comparison of predictions from the buildingEXODUS evacuation model with experimental data, in: J. Shields (Ed.), University of Ulster, 1998, pp. 711- 721.

[16] J. Parke, S. Gwynne, E.R. Galea, P. Lawrence, Validating the buildingEXODUS evacuation model using data from an unannounced trial evacuation, in: E.R. Galea (Ed.) Procedings of 2nd International Pedestrian and Evacuation Dynamics Conference, CMS Press, University of Greenwich, UK, 2003, 295-306.

[17] V.O. Shestopal, S.J. Grubits, S. J. (1994). Evacuation model for merging traffic flows in multi-room and multistory buildings, Fire Safety Science -- Proceedings of the 4th International Symposium, 1994, pp. 625-632.

[18] L.T. Buckmann, J. Leather, Modelling station congestion the PEDROUTE way, Traffic Engineering and Control 35 (1994) 373-377.

[19] IES, Simulex Technical Reference; Evacuation Modeling Software. Integrated Environmental Solutions, Inc., 2000. Generic.

[20] T. Korhonen, S. Hostikka, Fire Dynamics Simulator with Evacuation: FDS+Evac, Technical Reference and User's Guide, VTT Technical Research Centre of Finland, 2009.

[21] T. Korhonen, S. Hostikka, S. Heliövaara, H. Ehtamo, FDS+Evac: modelling social interactions in fire evacuation, in: 7th International Conference on Performance-Based Codes and Fire Safety Design Methods, Auckland, New Zealand, 2008, pp. 241-250.

[22] A. Steiner, M. Philipp, A. Schmid, Parameter estimation for a pedestrian simulation model, in: 7th Swiss Transport Research Conference Monte Verità, 2007.

[23] J. Berrou, J. Beecham, P. Quaglia, M. Kagarlis and A. Gerodimos, Calibration and validation of the Legion simulation model using empirical data, in: Pedestrian and Evacuation Dynamics 2005, 167-181.

[24] E. Morrow, MassMotion: simulating human behaviour to inform design for optimal performance, The Arup Journal (2010), 38-40.

[25] Cabinet Office, Understanding Crowd Behaviours Volume 1: Practical Guidance and Lessons Identified, The Stationary Office, United Kingdom, 2010.

[26] https://www.thunderheadeng.com/pathfinder/pathfinder-features/

[27] J. Cappuccio, Pathfinder: A computer-based timed egress simulation, Fire Protection Engineering, 8, 2000, 11- 12.

[28] A.W. Heskestad, O.J. Meland, Determination of evacuation times as a function of occupant and building characteristics and performance of evacuation measures, in: Human Behaviour in Fire -- Proceedings of the 1st International Symposium, 1998, pp. 673-680.

[29] CRISP 2.4, Computer models for fire and smoke, http://www.firemodelsurvey.com/pdf/CRISP_2001.pdf

[30] Koo, S., Fraser-Mitchell, J.N., Upadhyay, R. and Welch, S., 2009. Sensor-linked fire simulation using a Monte-Carlo approach Fire Safety Science 9: 1389-1400.

[31] A TECHNICAL SUMMARY OF THE EGRESS CODE ESRT/NOIL/27812001/002(R) Issue 1a.

[32] AEA Technology A Technical Summary of the AEA EGRESS Code, AEA Technology, Warrington, UK, 2002.

[33] S.M. Lo, Z. Fang, A spatial-grid evacuation model for buildings, Journal of Fire Sciences 18 (2000) 376-394

[34] S.M. Lo, H.C. Huang, P. Wang, K.K. Yuen, A game theory based exit selection model for evacuation, Fire Safety Journal 41 (2006) 364-369.

[35] R.F. Fahy, Verifying the predictive capability of EXIT89, in: 2nd International Symposium on Human Behaviour in Fire, 2001, pp. 53-63.

[36] R.F. Fahy, Calculation methods for egress predicition, in: Fire Protection Handbook 19th edition, National Fire Protection Association, Quincy, MA, 2003

[37] A. Zavin, F. Anzum, S. M. Faisal Rahman, M. N. Islam and M. Hoque, "Towards Developing an Intelligent Fire Exit Guidance System Using Informed Search Technique," *2018 21st International Conference of Computer and Information Technology (ICCIT)*, Dhaka, Bangladesh, 2018, pp. 1-6.

[38] Wong, SK., Wang, YS., Tang, PK. et al. "Optimized evacuation route based on crowd simulation" Comp. Visual Media (2017) 3: 243.

[39] Manley, M., Kim, Y. S., Christensen, K., & Chen, A. (2011). Modeling Emergency Evacuation of Individuals with Disabilities in a Densely Populated Airport. *Transportation Research Record*, *2206*(1), 32–38.

[40] X. Zhang, Q. Zhong, Y. Li, W. Li and Q. Luo, "Simulation of Fire Emergency Evacuation in Metro Station Based on Cellular Automata," *2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, Singapore, 2018, pp. 40-44.

[41] E. Bonabeau, Agent-based modeling: Methods and techniques for simulating human systems, Proceedings of the National Academy of Sciences May 2002, 99 (suppl 3) 7280-7287.

[42] Neal Wagner, Vikas Agrawal, An agent-based simulation system for concert venue crowd evacuation modeling in the presence of a fire disaster, Expert Systems with Applications, Volume 41, Issue 6,2014, Pages 2807-2815.

[43] Nuria Pelechano, Ali Malkawi, "Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches" in Automation in Construction, Volume 17, Issue 4, 2008, Pages 377-385.

[44] Lu Tan, Mingyuan Hu, Hui Lin, Agent-based simulation of building evacuation: Combining human behavior with predictable spatial accessibility in a fire emergency, Information Sciences, Volume 295, 2015, Pages 53-66.

[45] Unity Documentation : https://docs.unity3d.com/Manual/UnityManual.html

[46] Sun, Songqiao, "Agent-based Crowd Simulation Modelling for a Gaming Environment" (2017). *Electronic Theses and Dissertations*. 7399. https://scholar.uwindsor.ca/etd/7399/

VITA AUCTORIS

NAME:	Rachit Tomar
PLACE OF BIRTH:	Delhi, India
YEAR OF BIRTH:	1994
EDUCATION:	Bachelor of Engineering, 2012-2016 Indraprastha University, Delhi, India Master of Science in Computer Science, co-op, 2017-2019 University of Windsor, Windsor, ON