THEME PARK CROWD SIMULATION USING MULTI AGENT SYSTEM

D.M.L.N.P. GUNAWARDENA

(189380H)

Supervisor: Dr. Subha Fernando

Degree of Master of Science in Artificial Intelligence

Departments of Computational Mathematics

University of Moratuwa

Sri Lanka

2020–10-30

Declaration

I declare that this is my own research proposal and this proposal does not incorporate without acknowledgement any material previously published submitted for a Degree or Diploma in any other university or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Signature: .... Date:

I have read the proposal and it is in accordance with the approved university proposal outline. I am willing to supervise the research work of the above candidate on the proposed area.

Signature of the supervisor: .... Date:

Dedication

I am dedicating this thesis to beloved my family and many friends, who have meant and continue to mean so much to me.

A special feeling of gratitude to my loving father, Buddhadasa Matarage. Although he is no longer of this world, his memories continue to regulate my life. And my loving mother Sumana Gunawardena, whose words of encouragement and push for tenacity ring in my ears. My loving wife Deepani, children Yehen and Risandi have never left my side and are very special.

I also dedicate this dissertation to my many Msc friends and office friends, who have supported me throughout the process. I will always appreciate all they have done, especially my reporting manager Mr. Saliya Sajith for releasing me from work to achieve deadlines. And my co-workers sharing my work load to balance research work and office work.

Last but not least I am dedicating this to my late elder brother Samantha Matarage, gone forever away from our loving eyes and who left a void never to be filled in our lives. Though your life was short, I will make sure your memory lives on as long as I shall live.

Acknowledgements

I would like to express my gratitude to my supervisor Dr. Subha Fernando for the useful comments, remarks and support given through the learning process of this thesis. Also, I like to thank the panel of lectures who guided me throughout the master’s program.

Not least of all, I owe so much to my whole family for their undying support, their unwavering belief that I can achieve so much.

Unfortunately, I cannot thank everyone by name because it would take a lifetime but, I just want you all to know that you count so much. I would never have completed this thesis, if it not for your sincere love and help. So thank you all.

Abstract

Computer-based crowd simulation has become a dominant research topic today. Computer-based simulation applications are used in education, entertainment, training, theme park design and building evacuation. Among them, virtual crowd simulation has become a dominant topic in theme park industry. Limited research has been conducted in theme park crowd simulation using multi-agent system. Virtual simulations can be done changing the configurations, to decide the best-suited locations for stalls in the premises. Otherwise, it will cost a lot to change physically located items as experience and feedback.

In this research, Multi-Agent Technology has been used to simulate crowed behavior in Theme Park when an emergency is caused due to fire. NetLogo, a multi-agent simulation software, is used to build the modal. The crowd in the park is identified as agents. Different agents, children, parents, individuals and couples are programmed to behave as for social norms, defined under social science. The basic goal of every living agent, is to stay away from fire and evacuate from the closet exit as quickly as they can. But there are exceptional scenarios, unique to different agents. For instance, parents try to find their children, before existing from environment. We have defined coordinator agents to manage crowded areas and to help parent agents, who get lost while looking for their children. Logics, that governs each type of agent behavior are programmed in NetLogo.

The simulation is tested changing the number of agents and observed increment of evacuation time when the number of agents are increased.

In research simulation, few emergent phenomena were observed. One is, some areas get crowed while agents are evacuating the theme park. Another is, exits which are away from the fire location are getting crowded. And parent agents get lost in theme park while looking for their children.

Table of Contents

List of Figures

List of Tables

Abbreviation

MAS	-	Multi Agent System
FPS	-	Frames Per Seconds
SFM	-	Mathematical Model Social Force Model
OSM	-	Open Street Map
DAI	-	Distributed Artificial Intelligence

Chapter 01

Prolegomena

1.1 Introduction

Crowd simulation is becoming dominant in the research world for crowd evacuation simulation, pedestrian simulation, crowd formation simulation, traffic simulation and insect swarm simulation [1]. In this domain, peoples have used flow based, entity based and agent based [2] approaches as the simulation methodologies. Theme park industry is an emerged industry to entertain humans by interacting with activities. This research has been designed to develop a framework to simulate emergent crowd behavior in a theme park in an emergency, using multi-agent technology. The scope of this research is limited to simulate a situation, due to an emergency condition fire.

In this chapter, we present aims and objectives, background and motivation, introduction to theme park and crowd simulation. Further we will define the emergent feature. Finally, we present the structure of the remaining chapters of the thesis.

1.2 Aims and Objectives

Aim of the research is to use emergent behavior in Multi-Agent System and to simulate crowd behavior in a Theme Park when an emergency occurred due to fire.

Objectives

To achieve the aim, we identify the objectives as mentioned below.

1. To critically review of literature to identify the problem and the approach. And to identify the social norms, that govern crowd movements. And social inter-connections and interactions between different social groups.
2. To define the agent types and their properties that is required for proper simulation of the environment.
3. To Implement the solution using a simulation tool such as NetLogo.
4. Validating the emergent features of the system, crowd in theme park due to fire emergency.

1.3 Background and Motivation

Theme park crowd behavior, due to an emergency, requires to be analyzed to position the events and exits to minimize the damage to visitors. Further, this can be extended to optimize the design of theme park to evacuate the crowd from the premises as quickly as possible.

There are three reasons for crowd simulation to become so critical and interest - one to test design strategies. Second to test theories developed and third to create phenomena which to theorize [3]. Crowd behaviors are emergent phenomena, which is hard to capture in mathematical equations. Research approaches, to model crowd behavior can be categorized into one of the following.

1.3.1 Flow Based modeling

Researches has been conducted to simulate the analogue between crowd behavior with particle and fluid dynamics [4]. Crowd does not follow laws of physics but their desire and choice. Crow decide the direction as they wish, stop and start arbitrary. People are not utilizing available resources evenly. For example, when leaving a building crowd may herd and clogged at one exit while some other exist is not fully utilized.

1.3.2 Cellular Automata

People have discretized the flow are of interest for simulation into to a grid which can be represented by a Matrix [2]. And each cell is used to describe a free area or occupied area by individual. Decision Rules are defined to simulate the crowd movement, from one cell to another. Egress and Pedroute are two well know systems which follows the described architecture.

1.3.4 Agent-Based model

Emergent features due to social and psychological interaction with the environment and individuals are the main architectural phenomena of the crowd dynamic [4]. Legion is a well-known, such system. Legion works on four parameters 1. Goal point 2. Speed 3. Distance from others and 4. Reaction time. And least effort is the decision rule for the movement of the crowd.

1.4 Theme park concept

A theme park is a place with attractions made up of rides, such as roller coasters and water rides. A theme park also contains games and events for entertainment purpose along with shops, restaurants and other entertainment outlets. A theme park is a type of amusement park, that bases its structures and attractions around a central theme. Unlike mobile carnivals and funfairs, theme parks are stationary and built for long-lasting operations. Theme parks are made to be enjoyed by adults, teenagers and children. Park areas are themed to a particular area like water parks, parks modelled after movies, parked modelled after toys usually adventure or action.

Few of world-famous theme parks are

- Disneyland Park, California, USA
- Walt Disney World Resort theme parks, Florida, USA
- Universal Studios Florida, USA
- Fuji-Q Highland, Yamanashi, Japan The window of the World, Shenzhen, China

1.5 Crowd Simulation

Computer-based crowd simulation has become a dominant research topic today. Limited research has been done to simulate crowd behavior in theme park using multi-agent technology. Emergent feature of Agents due to social interaction and psychological status is not addressed in these researches in an emergency. Emergent feature of the Agents due to social interaction will add the novelty to the study.

To fully understand the crowd behavior requires, expose the actual crowd into the real field. For this, the environment need to be physically built. And the individuals with distinct psychological and social values need to be found. This is an impractical and costly approach. The simulation of crowd behavior has a significant advantage and benefit all the stakeholders like designers, planners and managers etc.

1.6 Definition of Emergence

Emergence can be defined as the new phenomenon arising in a system, that was not defined in the specification of the system. In crowd simulation, we define logic for individual agent. But we can observe new features as they behave as a whole. Crowding at the exit of the theme park, can be considered as an example for emergence feature.

1.7 Structure of the thesis

Rest of the thesis is structured as follows. Chapter 2 critically reviews the domain of crowd simulation, highlighting the technologies, advantages and limitations for defining the research problem. Chapter 3 describes the Multi-Agent technology and NetLogo simulation software framework for the research simulation. Chapter 4 includes the MAS approach to simulate crowd in theme park at fire emergency. Chapter 5 is on the design of the theme park crowd simulation. Chapter 6 presents the detail implementation of the theme park crowd simulation in an emergency. Chapter 7 discusses the evaluation of the research solution explaining the evaluation strategy, emergent features and analysis. Chapter 8 reports the outcome of the research and suggestions for future work.

1.8 Summary

In this section, we have described the features of a theme park and its purpose. Further, we have identified the importance of crowd simulation in a theme park when an emergency occurred. Additionally, we described the emergence phenomena. Finally, we have reported the structure of the thesis. In the next chapters, we will describe the exiting work done so far, along with their limitations and importance. And letter of the next chapter, we will discuss the research problem.

Chapter 02

Development and challenges in Crowd Simulation

2.1 Introduction

In the previous chapter, we identified the MAS terminology and crowd simulation terminology. And we further discussed, what a theme park and the importance of crowd simulation in a theme park for designing and decision making.

In this chapter, we will discuss the related work done by other researchers' along with the limitations and advantages.

2.2 Early Development in Crowd Simulation

Flow based modal has been used by researches for crowd simulation at the early stage of crowd simulation [2]. In this approach crowd flow is modeled based on fluid dynamics. The major drawback of the modal is, crowd does not follow laws of physics but act based on individual desire and choice. Crowd decide the direction as they wish, they stop and start arbitrary. People are not utilizing available resources evenly. Figure - 2.1 explain the flow based approach visually.

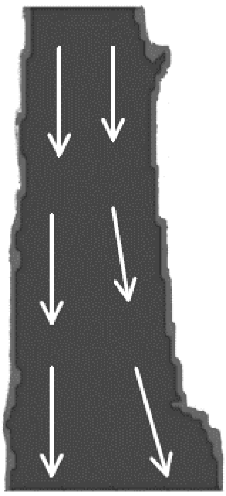


Figure - 2.1: Flow Based Modal

2.3 Modern Development in Crowd Simulation

Modern research is focused on using MAS and Cellular Automata for crowd simulation. Multi-Agent system has been used for crowd simulation in Emergency Situations [3]. The model flow chart is depicted in Figure-2.2. FDS (Fluid Dynamic Simulator) is used to generate hazard data. And ProSim is used to design building geometry. The framework is designed to define an existing path for each considering all interactions between agents and the environment. Individual behavior is fed to software, OpenGL(ModP). Agents are identified using attributes like Age, Sex, Reaction Time, State. The framework has not proposed a proper validation method to check the simulation for a scenario. And the framework does not consider social forces and emergent of human interactions.

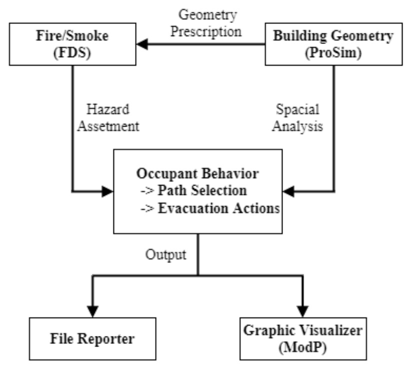


Figure - 2.2 FDS Modal Flow chart

Crowd simulation in a theme park is implemented using Multi-Agent System [5]. An agent represents any guest at the theme park. Agent design has done considering the decision making and social behavior. Agents are designed using the Layers of Cognition as described in Table-2.1. The agent brain is validated and refined using real time data. SpriOps AI editor is used to define agent behavior and Autodesk Maya is used for defining and running a simulation. The implementation does not integrate a realistic perception of the environment, features like vision, hearing. Agent behavior based on cultural differences is not addressed. People from different cultures react differently to the same stimuli.

Agent Design		Layers of Cognition
Queuing, crowding, grouping		Social
Decision-making	Opportunity	Rational
	Motivation	Cognitive
Collision avoidance		Reactive
Mesh and Animation		Biomechanical

Table - 2.1 Agent Attributes

MASSEgress framework model the individuals as agents, who interacts with the virtual environment and other agents via sensing, decision making and reacting [4]. Agents are different from each other by age, body dimension, mobility and personality. System maintains five human categories Median, Adult Male, Adult Female, Child and Elderly. When a situation is perceived, an agent activates a decision rule to produce an action. Choice of the decision rule is determined by psychological factors like the importance of the perceived information, uncertainty and urgency. The system architecture of the MASSEgress is shown in Figure-2.3. Individual behavior model is an iterative process of

1. Internally trigger for the decision
2. Perceive the information about the situation
3. Interpret and choose the decision rule
4. Conduct the collision check and execute the decision. The framework does not include physical models such as pushing, knocking. And also it is not considered accidental events.

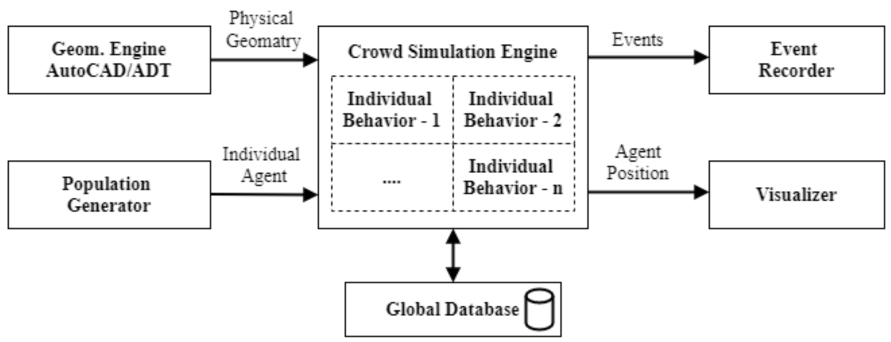


Figure - 2.3: MASSEgress System Architecture

Agent-Based Crowd Simulation in Airports Using Games Technology is conducted to simulate crowd behavior in Airport using Agent System and Gaming theory [6]. Hybrid architecture, consists of reactive behavior and deliberative reasoning is used for simulation. Deliberative reasoning uses a special database to store environmental and sociological data. Reactive control generates sophisticated agent behavior using PhysX. Environment is divided into cells. Cell contains environmental and social data. On top of grid four layers are implemented namely physical layer, navigation layer, occupancy layer and area search layer.

- A physical layer represents obstacles, such as walls, stairs, lifts, info screens.
- Navigation layer stores all path generated offline by Dijkstra algorithm.
- Occupancy layer represents all information about the occupancy levels of each cell saving information about which individuals are currently occupying which cell.
- Area search layer contains information about an unblocked line cells within a limited viewing arc in front of each agent.

Combination of Spatial Database and Physics Middleware is used for the framework. The system is able to simulate roughly 4,200 agents, maintaining an interactive frame rate of 30 FPS. The system has a major limitation on the need of pre calculating the navigation layer due to high computation required and size of the cell (5m*5m). And the agent in the simulation is not natural individual behavior in Airport but player-controlled. And the group behaviors, like family stay close together, is not addressed. Also situational event, that effect the agent emotional act is future work.

Crowd Simulation Incorporating Agent Psychological Models, Roles and Communication is conducted to simulate crowd behavior in evacuation using Multi Agent Technology and Communication phenomena [7]. The solution compromise of two levels, way finding, finding the path to exit and local movement with room using social forces. The agent has his own mental map, regarding the geometry of the building which will update as an agent explores the environment and communicating with each other. Agents are categorized into three roles.

1. Trained leaders who are well known to evacuation process and the exit path.
2. Untrained leaders who can handle the stress and search for path to exit.
3. Untrained leaders who are just followers.

PEMserv is used to configure psychological behavior of the agent which will affect the decision making nature and more realistic emergent feature. Fatigue, hungry, thirsty and injury are some of the such psychological features. MACES implements agent bodies, actions and results. For each agent, PMFserv would operate its perception and run its cognition to determine collective and individual action decisions and pass instructions back to MACES to carry out the resulting actions and emergent behaviors.

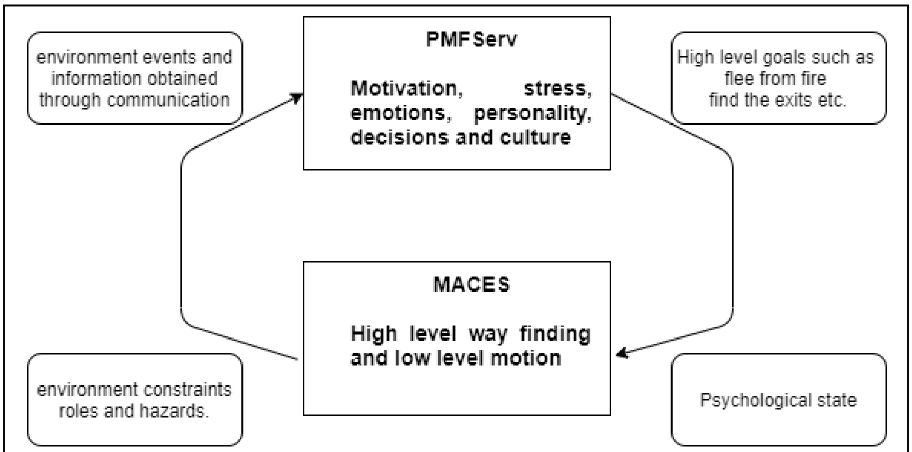


Figure – 2.4: PMFServ and MACES Interfacing

Research has been conducted to simulate Pedestrian Mobility in Theme Park Disasters [8]. Open Street Map (OSM) is used to extract the theme park map. Visitors are randomly positioned in the park. Each individual selects and exit gate. And move from one view point to other to reach exit. This selection depends on the individual knowledge of the environment, obstacles and red-zones along the way. Communication between the agents and the broadcasting of awareness messages are not considered. Mathematical model Social Force Model is used to predict pedestrian flow sharing the road.

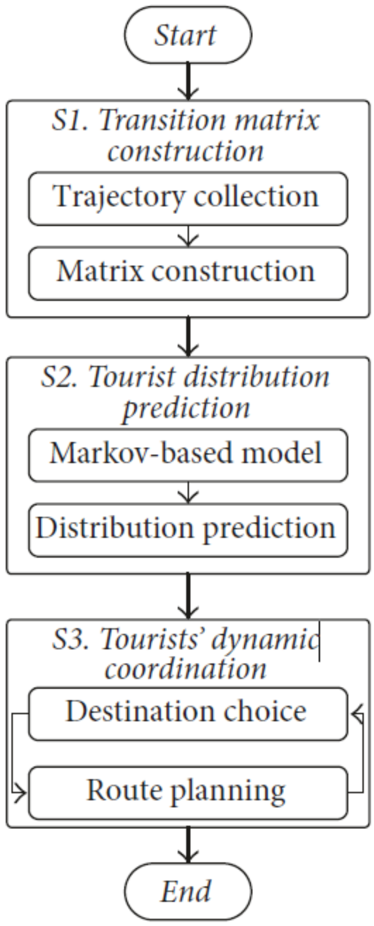


Figure - 2.5: Flow Diagram

Agent based modal to simulate emergent behavior of system of systems is modeled [9]. Agents follow one single rule, when agents come together one of them adopt speed and direction of other agent. Net effects is two agents move together. This will form group of agents move randomly together. Little groups will form large groups move together. Finally, a single group that moves together. Emergence is the dominant feature of agents in systems and has following four properties.

1. Emergence will happen only when the number of agents are more than single.
2. Emergent behavior will inversely proposal to the bond between the agents.
3. Emergence is nonlinear but due to interaction and feedback.
4. Self-organized so unpredictable and random.

Agent based modeling and simulation is conducted using SysML.

A model is developed to reflect the influence group of pedestrians [10]. The individuals interact inter group members and intra group members. As a result, collective walking patterns emerge. The model use the mathematical model derived by previous exponentially proved by previous work [11].

A framework for mitigating crowd in theme park has been proposed [12]. The methodology has three steps: transition matrix construction, predicting tourist distribution and coordinating tourists' movements. System flow diagram is shown in Figure – 2.5. System uses Marcov modal for predicting tourist distribution. Agent based simulation modal is constructed to validate the approach. Tourist movement patterns and interests are not considered in the research.

NetLogo Implementation of an Evacuation Scenario [13] describes the use of NetLogo as a tool for agent-based crowd evacuation simulation in closed space. Further, this modal can be used to assess building's architectural layouts in term of evacuation time. All the agent will have the same characteristics, and head to nearest exit. No overlapping is allowed. If other agent occupies the space ahead, agent will wait till his turn to move on. Closed space is represented with no obstacles. Arching and clogging phenomena is observed in simulation.

2.4 Challenges in Crowd Simulation

It is not practical to incorporate all the social norms of human in the agent characteristics. Social science is not matured enough to define all social interactions within crowds.

2.5 Problem Definition

Having considered the summary of issues for crow simulation in a theme park, we have identified, no research has done to simulate emergent feature of theme park crowd behavior when a fire emergency occurred using MAS. We also noticed many researchers have used MAS simulations to identify emergent featured in building when emergency situations occurred. Even though, the research is done to simulate agent technology to solve crowd behavior, emergent behavior due to social norms in social groups are not addressed in the context of theme park at an emergency. Based on our critical review, we have summarized important previous research efforts in table – 2.1. We define our research problem, to simulate emergent features of crowds in a theme park when a fire emergency occurred using MAS.

2.6 Summary of Literature Review

Research	Nature of Work	Technology	Issues Identified
Crowd simulation modeling applied to emergency and evacuation simulations using multi-agent systems [1]	Evacuation framework for evacuating crowd in a building at emergency	Multi Agent	Social forces and emergent of human are not considered. Validation methodology is not included.
A Multi-agent Based Framework for the Simulation of Human and Social Behaviors during Emergency Evacuations [2]	Human and Social Behaviors during Emergency Evacuations in Building	Multi Agent	Physical phenomena such as pushing, knocking not modeled. Accidental events not modeled.
Agent-based Crowd Simulation Tool For Theme Park Environments [4]	Crowd simulation in theme park	Multi Agent	Perception of environment and cultural differences are not included
Agent-Based Crowd Simulation in Airports Using Games Technology [5]	Crowd Simulation in Airport terminal	Multi Agent	Not natural individual behavior but player controlled Group behaviors and emotional act not considered
Crowd Simulation Incorporating Agent Psychological Models, Roles and Communication [6]	Crowd simulation for Building evacuation	Multi Agent	Only agent select exit and find way and evacuate.
Agent-Based Modeling the Emergent Behavior of a System-of-Systems [8]	Emergent behavior of system of systems	Multi Agent	Study only group emergent.
Toward a mathematical theory of behavioral-social dynamics for pedestrian crowds [12]	Crowd dynamics of pedestrians and social groups		Discuss only individual and group of individuals and interaction among them
How to Mitigate Theme Park Crowding? A Prospective Coordination Approach [11]	Framework for mitigating crowd in theme park	Mathematical framework, Marcov Modal	Tourists movement patterns and preference not considered.
NetLogo Implementation of an Evacuation Scenario [13]	Evacuation simulation in a closed space	Multi Agent	No obstacle in environment. All agents have same characteristics.

Table 2.1: Summary of Literature review

2.7 Summary

In this chapter we have given a critical review of crowd simulation. We have also identified various technologies used for crowd simulations. More importantly we have defined our research problem as how use MAS to simulate the emergent behavior of crowd in a theme park event at a fire emergency. In the literature review we have also identified multi agent technology as potential technologies for crowd simulation. In next chapter, we discuss our methodology, MAS to simulate the emergent behavior of crowd in a theme park event at a fire emergency.

Chapter 03

Methodology

3.1 Introduction

In previous chapter, we discussed a critical review of literature review of crowd simulation. The literature review identified the research problem and potential technology for solving the research problem. In this chapter we present the major technologies associate with the research. MAS is the AI technology used for simulating the crowd behavior in theme park. Crowd behavioral rules obtained from the social science are the biggest input to the system. NetLogo is used as the MAS simulation software.

3.2 Multi Agent Technology

Multi Agent Systems (MAS) emerged from the previous research efforts in Distributed Artificial Intelligence (DAI) in the early eighties [14]. in has been identified as the underlying technology for developing the simulation. DAI algorithms are classified into three categories [15] namely;

- Distributed problem solving (DPS).
- Parallel AI
- Multi Agent Systems (MAS)

3.3 Definition and characteristic of Multi Agent Systems

We will define the terminologies used in multi-agent system.

3.3.1 Agent

Agents in multi agent systems has following features in contrast to other programs [14].

- Agent perceives the environment where it is situated.
- Agent interacts with other agents.
- Agent is proactive and pursues its goal.

Research, Multi-Agent Systems: A Survey [16] has defined the term agent as autonomous entity, who sense the environment through the sensors connected to it. And act on the environment through the actuators. Further, researchers have used the term like softbots (software agents), knowbots (knowledge agents), taskbots (task based agents) to identify the agents, based on the domain where the agents are employed.

Research, Multi-agent systems: which research for which applications [15] has defined the term agent as an entity, which is placed in an environment. It senses different parameters that are used to make a decision based on the goal of the entity. The entity performs the necessary action on the environment based on this decision

3.3.2 Multi Agent Systems

Even though a single agent acts separately to solve a problem locale to that particular agent, a complex system is made up of different agents. Multi agent system can be defined as a collection of heterogeneous, computational entities, having their own problem solving capacities and interact with each other entities to solve the global goal. MAS reveals a kind of energy, that cannot be generated by summing the agents. System diagram of MAS is shown in Figure – 3.1.

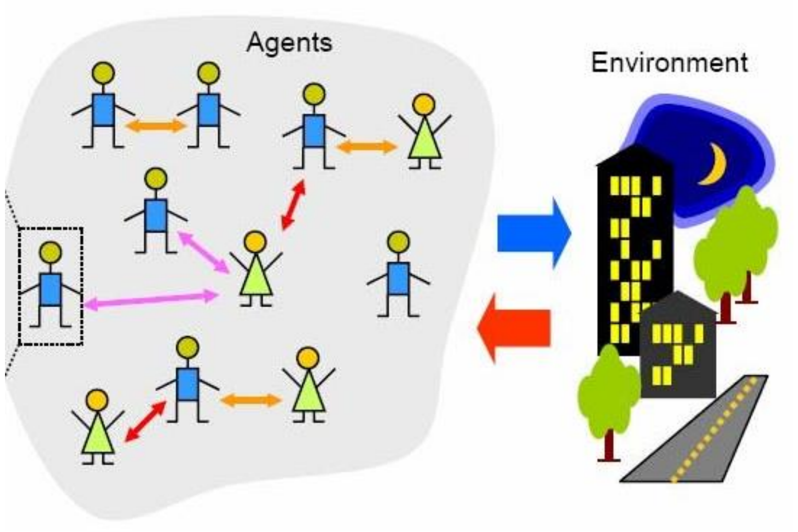


Figure – 3.1: Multi Agent Systems

3.4 Multi Agent Simulation Software NetLogo

NetLogo [17] [21] is a freely downloadable, agent based modal development software, that was developed by Center for Connected Learning and Computer-Based Modeling (CCL) at Northwestern University in United States. The project was directed by Uri Wilensky, professor of Learning Sciences, Computer Science and Complex Systems at Northwestern University.

NetLogo is used to develop modals to examine the emergent phenomena [19] [18]. Large scale patterns of the world are results large number of smaller interactions. Such large patterns that arises due to interactions among agents are called "emergent phenomena".

Netlog has three tabs interface tab, info tab and code tab. The interface tab is where we watch our modal runs visually. Additionally, we can use tool bar in this tab to create input components, to feed the variable values to the modal. For example, we can pass the number of agents to create as a variable value which will be fed using a slider. On the other hand, we can use a monitor component in interface tab to visualize how agents die against time that modal runs.

In info tab we provide an introduction to the modal. Here we describe, what is being modeled in the modal, how the modal is being built and how to explore and extend the modal.

The code tab is where the code of the modal is written. We store the code in code tab and run over and over. Once we click the "check" button, NetLogo examine the code written and report if the status of the code. The code tab will turn red, if there are any syntax errors. Error code that contains error will be highlighted and error message will be displayed. Figure – 3.2 is sample view of nelogo interface.

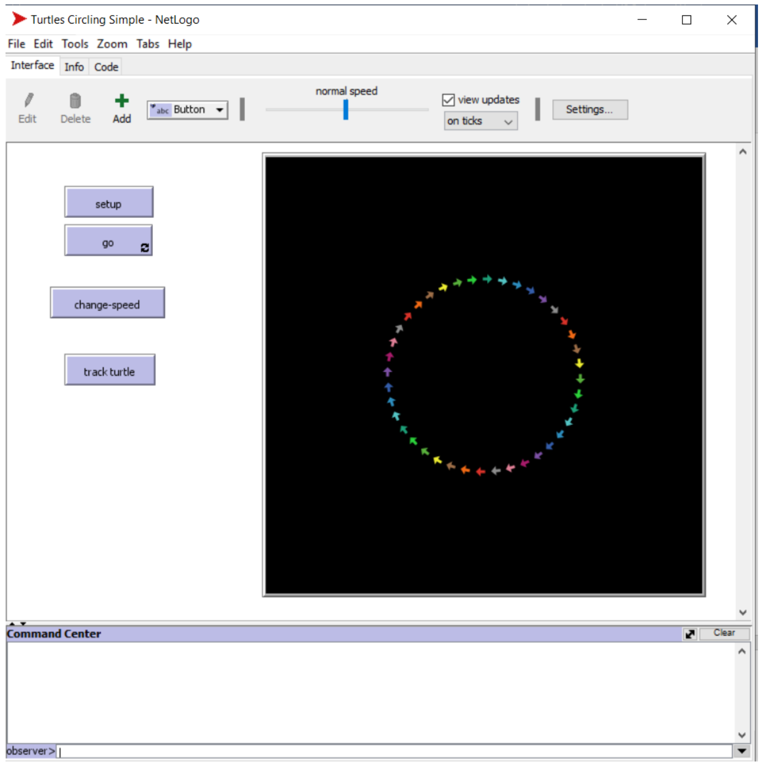


Figure – 3.2: NetLogo interface

3.4.1 Agents in NetLogo

NetLogo models are built based on four basic agent categories; turtles, patches, links and observer. Turtles are the agents that can move around in the world. World is composed by grid of patches. Patch is a square piece on the world on which turtles can move. Links interconnects two turtles. Observer doesn't have a location in the world, but can be considered as one who looks the world consists of turtles, patches and links

Turtles owns variables like size, color, shape which we can change as for the modal requirement. We can define different agents using the concept of breed in turtles. For example, we can create breeds to represent parents, children and animals. So we can give the commands to specific breed or all the breed using ask. "ask turtles" is used to give commands to all turtles and "ask parents" is used to run commands on specific breed parents only.

The NetLogo environment is built up with concept of grid. Each location in the x, y grid system is conceptualized as a patch. Patches are stationary agents in the modal. Patches have built in variables such as color, x coordinate and y coordinate. We can define variables to be owned by patches. For example, if we want to identify whether the patch belongs to a road or to forest, we can define a patch variable ‘meaning’ and assign values ‘road’ and ‘forest’ for distinguish. So we can use these values in the modal development code.

Link is used to build the connection between two turtles. Links also owns properties like shape, end1, end2 manage the link in modals.

```
create-turtles 10
ask turtles [
  fd 1
  set shape "person"
]
;; all turtles move forward one step
ask patches [ set pcolor 4 ]
;; all patches turn red
ask turtle 4 [ rt 90 ]
;; only the turtle with id 4 turns right
```

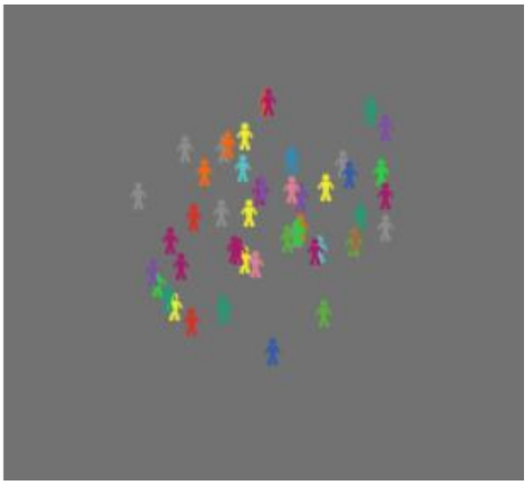


Figure – 3.3: Usage of ask to execute commands on agents and result

“Ask” command is used to tell a single agent or a set of agents which commands to execute. Figure -3.3 shows the syntax of passing commands to agents. Further, Figure – 3.3 shows how NetLogo interface reflect the result of commands execution.

3.5 Summary

In this chapter, we discussed MAS technology and NetLogo MAS simulation software. Further we discussed in depth the phenomena of MAS and terminology that governs the NetLogo modal development. In next chapter, we discuss the approach of the simulation. Further we will discuss the input, output, process features and users of the solution in next chpater.

Chapter 04

Approach

4.1 Introduction

In previous chapter, we discussed MAS as the technology used for crowd simulation. This chapter is structured to discuss approach of the simulation. And we will discuss the input, output, process, features and users of the solution.

4.2 Hypothesis

The hypothesis of the research is MAS can be used to identify emergent feature of crowd in a theme park event, when an emergency occurred due to fire situation.

4.3 Input

The inputs to the system can be identified as agents, geometry and fire notification alarm. The agents, we use in simulation are parents, children, couples, individuals and coordinators. Geometry is the design of the theme park, where the events are positioned and exits are located.

4.4 Output

The output of the system is emergent features arise from crowd in theme due to fire emergency.

4.5 Process

We have modal crowd simulation as MAS composing of five agents; parents, children, couples, individuals and coordinators. Features and governing rules of agents' behavior are tabulated in Table - 4.1.

Agent	Usual Behavior	Behavior in an emergency
Children	hang around a child events	Follow parent
Parents	Stay close to children. Hang around adult events.	Search for children. Find the closest exit. Stay away from the fire location. Evacuate from site.
Couples	Hand around couple events. Always stay together.	Find the closest exit. Stay away from the fire location. Exit from the site together.
Individuals	Hang around adult events.	Find the closest exit. Exit from the site, avoiding the fire location.
Coordinators	Random movements.	Guide the crowds.

Table – 4.1: Agents and Behavior Rules.

The types of events in the theme parks simulation is mentioned below.

- Children event
- Adult event
- Couple event
- Restricted areas
- Exits

Events and agents are configured in NetLogo for simulation.

4.6 Features

1. Emergent features of the crowd can be analyzed.
2. Geometry of the theme park can be changed easily.
3. Number of defined agents can be changed for analyzing.
4. Effected area dimensions and the location of the fire can be changed.
5. Can be extended to introduce new agent and events.
6. Required low CPU and Memory to run simulation.
7. Used open source simulation tool NetLogo. No licensing cost.

4.7 Users

The target end users of the simulation are designers of the theme park. Further, system can be used by safety and compliance managers to evaluate the design.

4.8 Summary

In this chapter, we discussed the hypothesis, input and output, process and features and users of the solution. In next chapter, we discuss to discuss the design of the research.

Chapter 05

Design of Theme Park Simulation

5.1 Introduction

In previous chapter; we discussed the hypothesis, input and output, process and features of the simulation. We also discussed the users of the research outcome. In this chapter we are going to discuss the design of the research.

5.2 System Architecture

The simulation consists of four major components.

1. Geometry Engine.
2. Agent Engine.
3. Simulation Engine,
4. Simulation Visualizer.

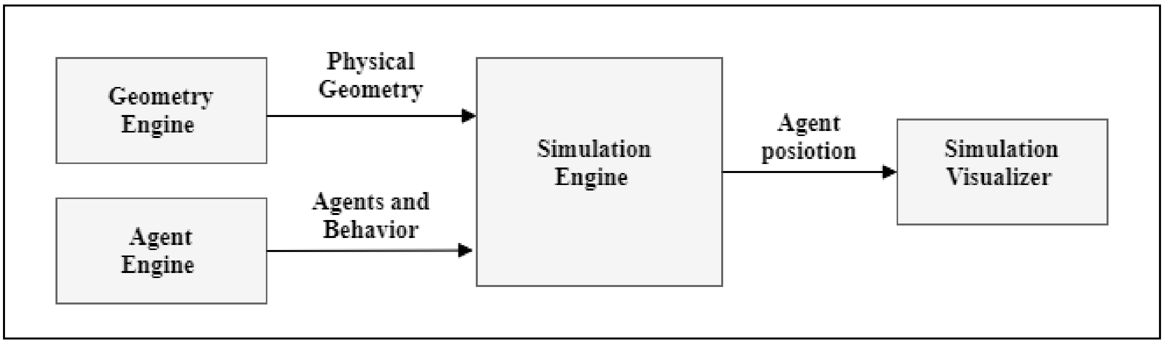


Figure – 5.1: High level architecture of the simulation

Architecture diagram of the system is shown in Figure – 5.1, with the interconnection within the modules.

5.2.1 Geometry Engine

Geometry engine is used to design the physical geometry of the Theme Park. NetLogo supports both, 2D and 3D simulation environments. We have used the 2D view for the research. NetLogo define the environment as a grid system as shown in Figure - 5.2. Each cell of the grid is defined as a patch. Patch has x and y coordinate, on which agents can move and stay. Patch having i,j coordinates and its neighbor patches are described in Figure – 5.3.

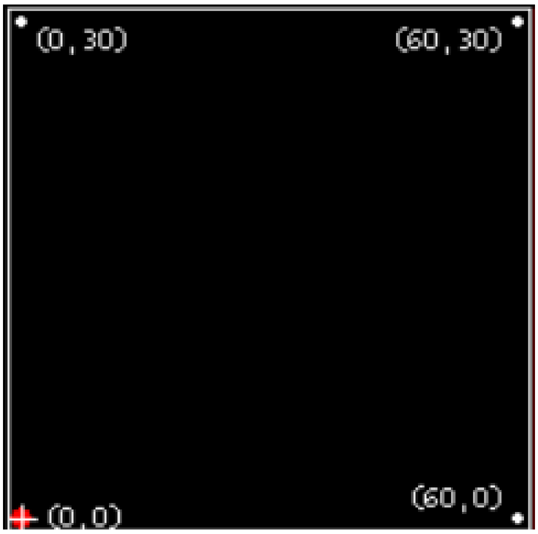


Figure - 5.2: Grid system of NetLogo

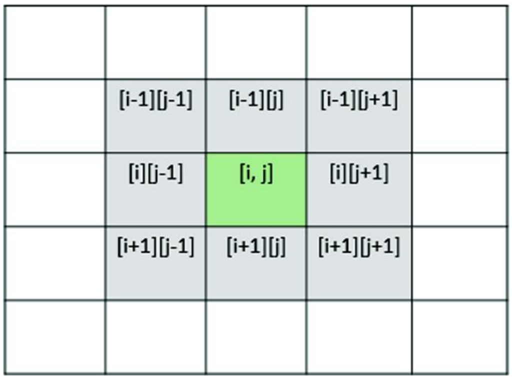


Figure - 5.2: [i,j] path and the neighbor patches

We have identified, followings items as the physical objects of the simulation.

- Child Event.

  Children are the target audience of these events.

- Couple Event.

  Couples are the target audience of these events

- Elder Event.

  Individuals, above age of 45 are the target audience of these events

- Exit.

  Exists are the physical locations of the park, where crowd in the park can exist from the park.

- Entrance.

  Entrances are the physical locations of the park, where crowd enter to the park.

- Restricted Area.

  The area in the theme park where movement of the crowds is prohibited.

- Fire location

  This is the location where the fire has happened.

- Effected area

  This is the area of the theme park, which is effected due to fire.

NetLogo does not support GUI for designing the geometry. We are using NetLogo programming, to design the geometry of the Theme park. NetLogo defines the environment of the simulation using grid system. In this research, we are using the 2D vie. NetLogo can be used, even for 3D visualizations.

5.2.2 Agent Engine

We define the agents and the behavior rules in agent engine. Agents in NetLogo are defined as turtles. Each turtle has a breed. Breed can be considered as type of the agent. We use a shape for agent breeds to distinguish them in the simulation visualization. We can define variables for all breeds or for individual breeds.

We define the agent behavior rules using NetLogo programming language. NetLogo has built in functions which can be used to define these rule. For example; NetLogo has in-cone function to get other agents in front of the agent, giving the vision angle and radius.

We have already identified the agents for the research in previous chapter and tabulated in Table - 4.1 along with the behaviors.

5.2.3 Simulation Engine

NetLogo includes an in build simulation engine to generate the movement of agents. NetLogo executes the programming code line by line sequentially. Behavior of an agent type is written as logic. For example, to move parent agent by one step is written as "ask parents [fd 1]". When NetLogo interprets this line of code, logic is executed on all the parent agents in parallel and wait till all parents completes the execution.

5.2.4 Simulation Visualizer

NetLogo has the capacity to perform 3D and 2D visualization. We are using the 2D visualization of the NetLogo environment for the simulation. In this simulation, time passes in discrete steps called "ticks". Tick is incremented after executing a single cycle of commands over all agents. Since the modal is configured to use tick based updates, after each tick, view is updated to visualize the output of command execution.

5.3 Summary

In this chapter, we discussed the design of the research. In next chapter, we discuss the implementation of the design in detail.

Implementation of Theme Park Simulation

6.1 Introduction

In previous chapter, we discussed the design of the simulation and how each module in the architecture are connected. In this chapter we discuss the implementation of each module in detail.

6.2 Geometry Engine

We have defined all the events using the combination of turtles and patches concept in NetLogo. Following elements are used when defining an element.

- Central coordinate (x,y)
- Width of the event area (w)
- Height of the event area (h)
- Shape of the event
- Event Boundary
- Identifier of the event
- Fire

These parameters are graphically defined in Figure - 6.1.

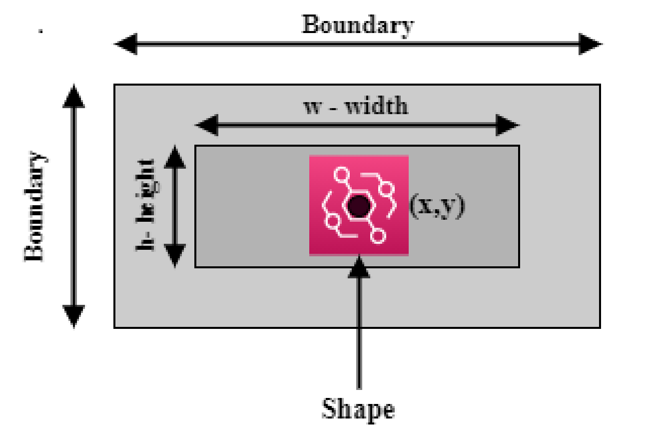


Figure – 6.1: Event parameters definition

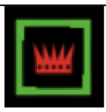
Event	Shape	Meaning Value	Has a crowd boundary	Crowd Boundary Value
Exist		exit	no	-
Child Event		childevent	yes	childeventBoundary
Elder Event		elderevent	yes	eldereventBoundary
Couple Event		coupleevent	yes	coupleeventBoundary
Restricted Area		restricted	no	-
Coordination point		coordination	yes	coordinationBoundary
Green area		green	no	-

Table – 6.1: Theme park events and identification

Two type of turtles shapes are used when marking the event edges, boundary and boundary corner. These two shapes are developed using the shape editor built in NetLogo tool. Horizontal edge is directly marked by the boundary shape. Vertical edge is defined by rotating the default boundary by 90 degrees. Boundary corner can be used directly to define the upper left edge of the event. By rotating the default edge, we define other three corners of the event. Figure - 6.2 describe this phenomenon clearly. We mark the patches inside the event by an identifier. For the this purpose we use the local variable of patches defined as meaning. Other than exists, other events have a boundary area where the target crowd hang around. We define the meaning value of these patches by concatenating the meaning of event patches with the string “Boundary”. This meaning values are used in the agent scope to identify the event they are dealing with. The setup of the world shown in figure - 6.3.

6.3 Agent Engine

Agents and their properties are defined in the agent engine. We have already defined the agent types and their behaviors in Table - 4.1. We will discuss the detail implementation of these behaviors in NetLogo environment.

6.3.1 Child Behavior

Children will hang around child events. They are reluctant to obey parents' directions when selecting an event. So naturally parents will follow children to children event.

But in an emergency situations children are in panic condition and not a decision maker to react. Behavior is defined to keep the children at their original location, until parent find them. Then children follow parent to evacuate from the site. Additionally, if parent died due to fire, child behavior is defined to keep where they are. In case, child is caught by fire, logic is defined to die child.

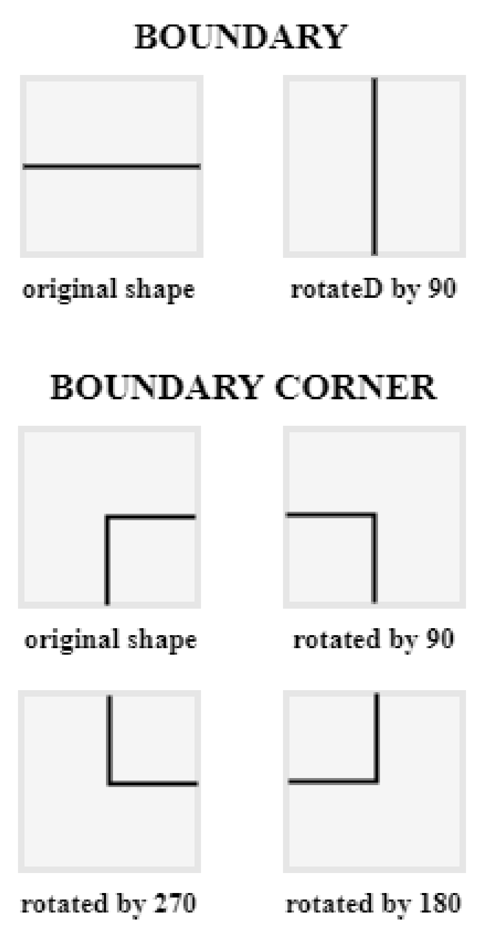


Figure – 6.2: Event parameters definition

6.3.2 Parent Behavior

In social science, parent behavior is defined as to take care of children. In simulation, parent will hang here and there, but keeping an eye on children.

But in case of fire situation, they first look for their children. Once the children are found, they look for the closest exit, avoiding the fire location. Then move to exit avoiding obstacles.

If they fail to find the children, wait for coordinator help. Coordinators reach to such lost parents by calculating the path using A * algorithm developed by Fernando Sancho [20]. Coordinators have the knowledge of the whole environment and not panic due to fire situation since they are trained. He translates the navigation path to parent to find child. The path is stored in a local variable of the parent agent for the purpose of finding children. Once child is found parent follow the usual evacuation procedure.

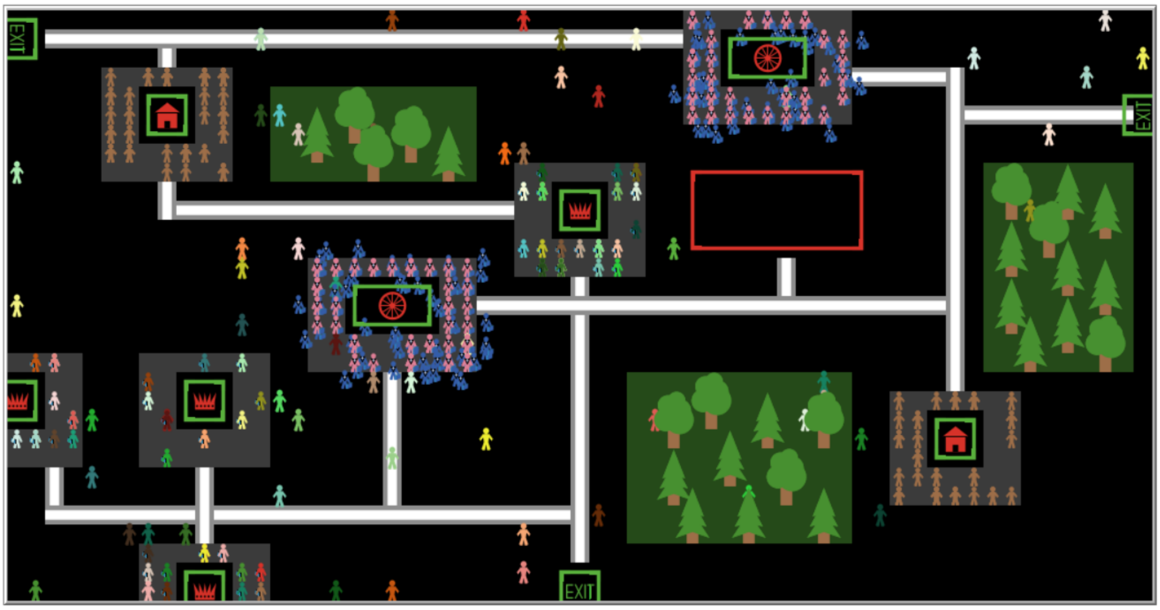


Figure 6 - 3: Modal Setup

6.3.3 Couple Behavior

In normal situation couples engaged in couple events and stay closely. Even in fire situations, they stay closely and follow the evacuation behavior. In simulation, male of the couple navigate to exist avoiding obstacles and staying away from fire. Female agent of the couple is programmed to follow the male. This is how the stay close behavior is defined.

6.3.4 Individual Behavior

Individuals are considered as elders. They engage in elder events in usual situations. But when the fire emergency occurred, they look for the closet exist, which is away from fire to evacuate from the site.

6.3.5 Coordinator Behavior

Coordinators are the agents, who are to help crowd to evacuate from the site when the fire situation occurred. They have the knowledge about the whole site and situations. But for the simulation, they are act on a single task.

If a parent calls for help, they will attend and coordinate to find the child. Parent asks for help, if he is blocked in the particular location more than 5,000 ticks continuously or fail to find the child with 10,000 tick counts.

Coordinator agents also handle the crowding scenarios. These agents calculate agent density in theme park environment. If the density of particular location exceeds the configured threshold, coordinator will move to that location. Then he changes the direction of crowd in the that area temporally to avoid crowding.

6.3.6 Obstacle avoidance behavior

All the agents we discussed so far, move in the environment without collisions with the obstacles. In this simulations, the patches which are not allowed to agents to move freely, are considered as obstacles. Theme park events and restricted areas are such obstacles. Even the fire location and the affected area is considered as obstacles. In NetLogo, we can get the patches, which are in the vision of agent using the in-cone function. We can pass the vision angle and the vision radius as parameters to the procedure. Figure – 6.4 illustrated this phenomenon clearly. We have defined the meaning variable in patches to hold the action of the patch. These values are mentioned in Table – 6.1. If one of the patches in agent vision cone has the meaning value of obstacles, agent rotate clockwise until the all the patches are clear to move.

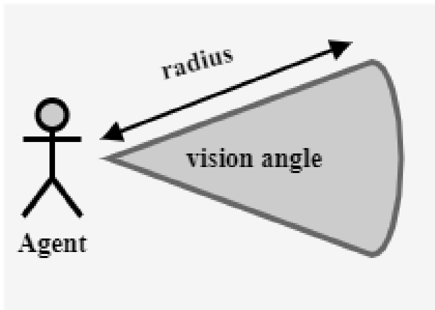


Figure – 6.4: Vision of agent in NetLogo

6.3.7 Agent collision avoidance behavior

Agents are not allowed to collide while moving. That is, no agent can move to a patch if there is any other agent on that particular patch. In such case, both forget their global target and change the direction randomly to avoid the agent ahead. The concept of collision between two agents and collision avoidance is shown in Figure – 6.5. After collision is avoided, agent face to global target and move towards it.

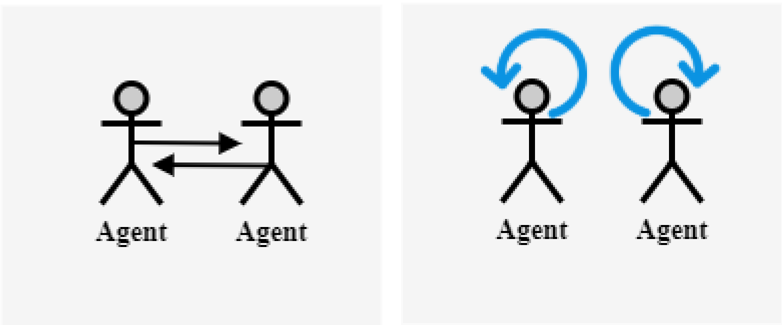


Figure – 6.5: Collision of agents & collision avoidance

6.3.8 Stay away from fire behavior

Every agent search for an exist to evacuate from the theme park when the fire situation is occurred to evacuate the site. There are few rules when searching the closet exist, to avoid agent from moving towards the fire.

- Agents on the left side of the fire find the closet exist in from left side.
- Agents on the right side of the fire find the closet exist in from right side.
- Agents on the top area not covered by above two scenarios, search the closet exist from top area.
- Agents on the bottom area not covered by first two scenarios, search the closet exist from bottom area.

This phenomenon is described in Figure – 6.6.

6.3.9 Exit and die behavior

When an agent reach to one of the patches having the meaning value as exit is considered as evacuated from the site. If an agent is on a patch having the meaning value as fire, when the fire situation is occurred, agent is considered as dead agent due to fire.

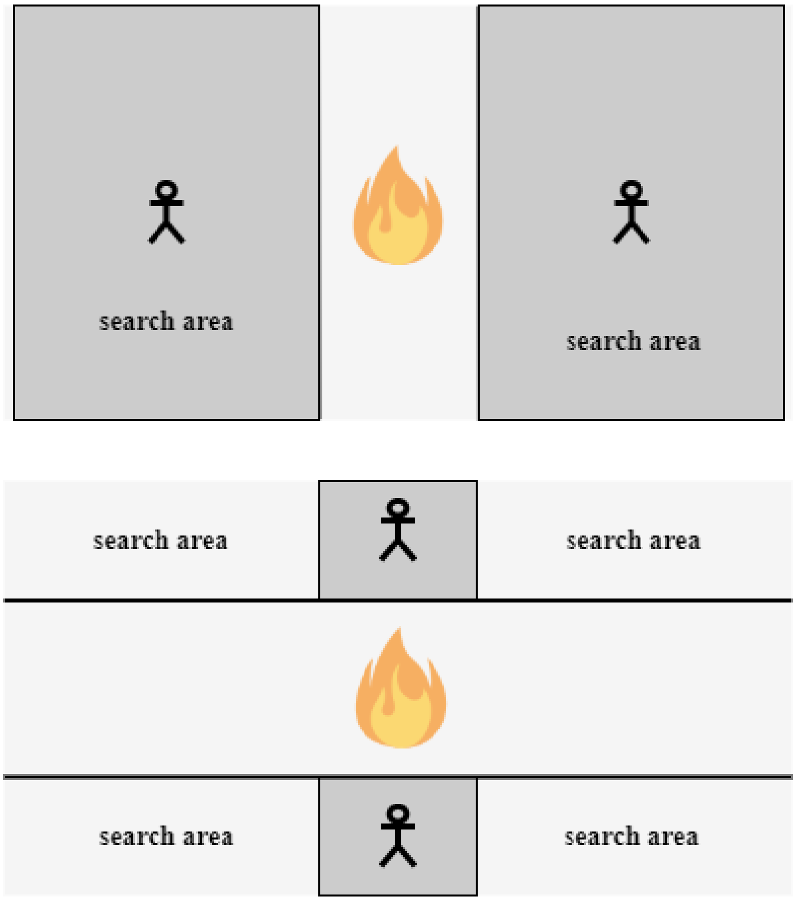


Figure – 6.6: Searching the exit avoiding the fire location

6.4 Summary

In this chapter, we discussed in detail the implementation of the simulation. In next chapter we discuss the evaluation of the simulation. mechanism we followed to evaluate our research on emergent behavior of crowd in a theme park event when a fire emergency occurred

Chapter 07

Evaluation

7.1 Introduction

In previous chapter, we discussed the system implementation of the theme park simulation. Further, we outlined behaviors of the agents in the simulation research. In this chapter, we discuss the tests and evaluation mechanism we followed to evaluate our research on emergent behavior of the crowd in a theme park event when a fire emergency occurred. We will discuss the following experiments and their results.

- Emergent features
- Agent Density experiment

7.2 Emergent Features of the agents

First, we run the modal with different agents and watch the phenomena of emergent.

7.2.1 Parents

It is observed that parents get lost while searching for their children. One reason for this is parents vision is blocked by the event buildings. Further we observed that number of parents get lost in searching their children increases when the affected area due to fire increases. We have introduced the coordinator agents to the modal, to help these parents to reach their children. This phenomenon is shown in figure – 7.1.

7.2.3 Crowding at exits and events

Agent get crowded at exits and nearby to events. All the agents try to reach the exit and evacuate from the theme park. And agents in events try to get the nearest exit to event. And parents also try move in the world to reach their children. Since this, agent meet each other and get collided. This phenomenon is shown in figure – 7.2.

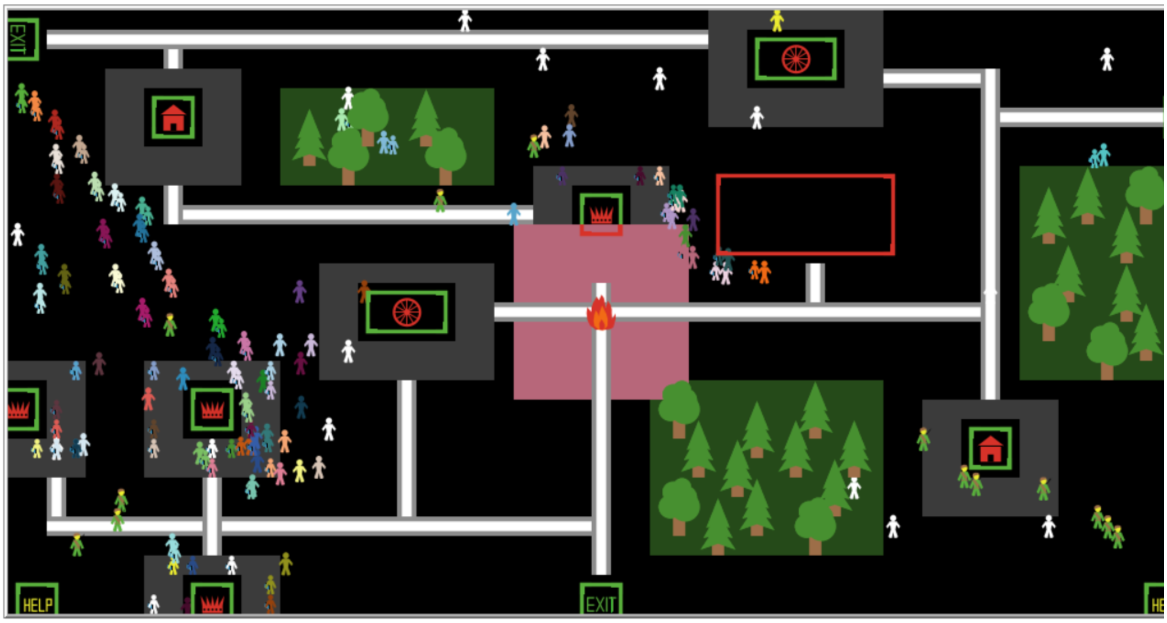


Figure – 7.1: Parent Lost in the world

7.2.4 Collision avoidance by negotiations

We observe agent get blocked when they move to the opposite directions. For example, let's consider parent agent is moving from top to bottom to reach the children. But that particular agent meets an elder agent moving from bottom to top to reach the closest exit. Both try to move opposite directions. At this scenario both agents are blocked. We have introduced

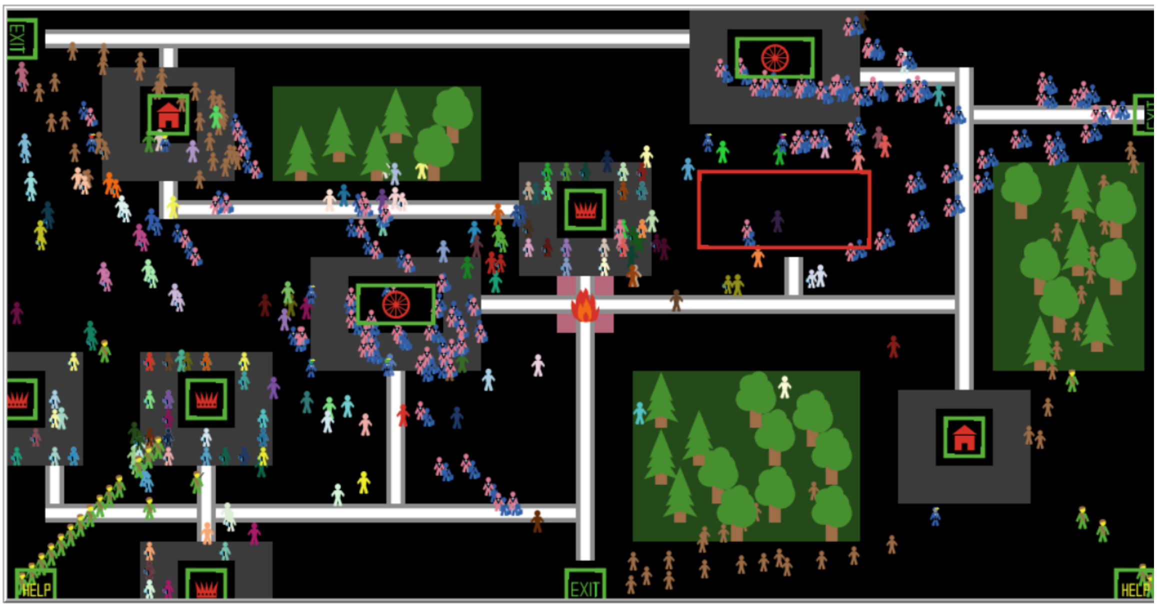


Figure – 7.2: Crowding at exists and collision with agents algorithm to deviate slightly from the original path to avoid this blocking. This can be considered as agent negotiation to avoid blocking. This algorithm is show in figure – 6.5

7.3 Agent Density Experiment

The next experiment we did was to check the relationship between the evacuation time to the population of agent. For this experiment, we run the modal with different population densities. And check the tick count required to evacuate all agents from the world. We start the testing with 10 agents from each type children, parents, couples and elders. And increased the number of agent did the experiment for several configurations, but keeping all the parameters remain as same.

7.3.1 Agent Density Experiment Results

The configuration parameters for the test and the agent evacuation tick count against number of agents are shown in figure – 7.3. We observe the number of tick required for evacuating whole population increase with the agent count.

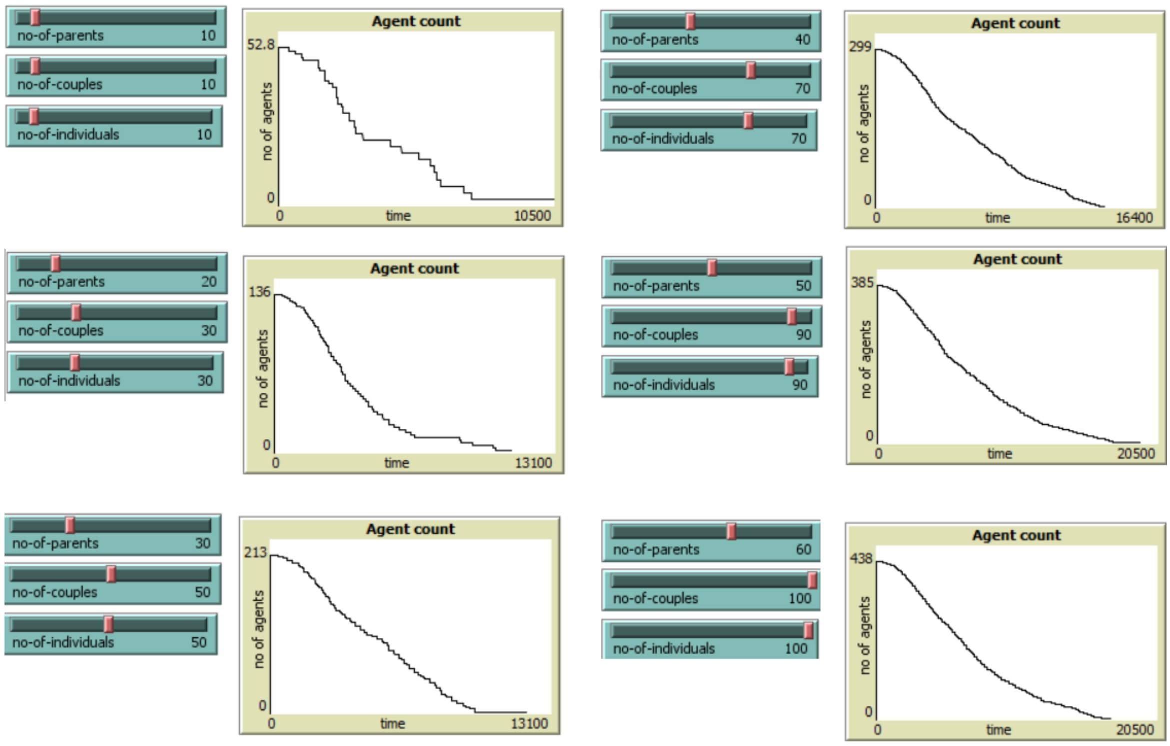


Figure – 7.3: Agent density and evacuation time result

7.3.2 Agent Density Experiment Results Interpretation

And it is also observed that, time period required to execute a tick cycle of the modal increases with the number of agents. This is since we increase the tick count by one, once all the agents completed executing asked command. The number of tick increases due to following reason.

- Collision among agents and events increases.
- Collision among agent increases.
- Number of parents who could not find their children increases and number of support required from coordinators increases.

7.4 Summary

In this chapter we discussed the and experiments we did to check evacuation time. Further we discussed the emergent behaviors from the crowd. In next chapter we discuss the finding of our research and the suggestions for future work.

Chapter 08

Conclusion & Suggested Further Work

8.1 Introduction

In previous chapter, we discussed the evaluation schema of our research on emergent behavior of crowd in a theme park event when a fire emergency occurred. In this chapter we discuss that MAS can be used to simulate emergent behavior of theme park crowd. Further we discuss the finding of the research and the limitations. Finally, we discuss the suggested future work for research topic.

8.2 Conclusion

Using MAS, we can simulate theme park crowd behavior when an emergency situation happened due to fire. Further we observed following emergent features from the simulation.

- Crowding at exits, which are away from fire.
  This is due to aim of the behavior of every agent is to evacuate from the park from the closet exit that is avoiding the fire.
- Parents get lost while searching for their children.
  These are the agents in the simulation, who have a different purpose before evacuating the theme park. They get lost trying to avoid the fire and event building in the path to child.
- Collisions are getting increased with the number of parents, searching for children.
  All the agents other than parents, find the closest exist and follow the path. So the chance to get collide with these agents are rare. But parents are moving, searching for their children. So they may get collide with agents in the movement of opposite direction.

Further we can conclude that, NetLogo can be used as the simulation environment for the simulating the crowd behaviors using MAS.

The aim of the research is to study the emergent feature of crowd behavior in theme park when an emergency situation occurred due to fire using Multi Agent System. This aim is achieved successfully.

Following objectives, which we defined at the beginning of the have met successfully during the execution of the research process.

1. To critically review of literature to identify the problem and the approach. And to identify the social norms, that govern crowd movements. And social inter-connections and interactions between different social groups.
2. To define the agent types and their properties that is required for proper simulation of the environment.
3. To Implement the solution using a simulation tool such as NetLogo.
4. Validating the emergent features of the system, crowd in theme park due to fire emergency.

Further we can use the modal to plan physical locations for the events, exits and entrances to avoid cost.

8.3 Limitations and Future Work

We have used MAS to simulate the crowd behavior in theme park due to emergency situation. We can extend our work to simulate crowd behavior in theme park without an emergency situation.

We are using the social behavior of parents, children, individuals and couples for the simulation. We can specialize the agents into deeper level of social categories like small families, disables, group of visitors coming together.

We can extend the modal to consider social status like religion, age and physiological level for the defined agents.

We can improve the algorithm used for collision avoidance between the agents. In the modal we are using arbitrary rotation of the path to avoid the agent ahead. This algorithm takes much time to avoid the collision. This can be improved using an advanced communication mechanism between two agents to avoid the collision.

8.4 Summary

In this chapter, we have discussed the findings of research and conclusions that we can finally derived from the research. Further we discussed the limitations and the suggestions for the future work for simulating emergent behavior of crowd in a theme park event when a fire emergency occurred.

.

REFERENCES

[1] M.-L. Xu, H. Jiang, X. Jin, and Z. Deng, "Crowd Simulation and Its Applications: Recent Advances," *J. Comput. Sci. Technol.*, vol. 29, pp. 799–811, Sep. 2014, doi: 10.1007/s11390-014-1469-y.

[2] S. Zhou *et al.*, "Crowd modeling and simulation technologies," *ACM Trans. Model. Comput. Simul.*, vol. 20, no. 4, pp. 1–35, Oct. 2010, doi: 10.1145/1842722.1842725.

[3] J. E. Almeida, R. J. Rosseti, and A. L. Coelho, "Crowd simulation modeling applied to emergency and evacuation simulations using multi-agent systems," *ArXiv Prepr. ArXiv13034692*, 2013.

[4] X. Pan, C. S. Han, K. Dauber, and K. H. Law, "A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations," *AI Soc.*, vol. 22, no. 2, pp. 113–132, Nov. 2007, doi: 10.1007/s00146-007-0126-1.

[5] S. Huerre, "Agent-based Crowd Simulation Tool For Theme Park Environments," 2010.

[6] O. Szymanezyk, P. Dickinson, and T. Duckett, "Agent-Based Crowd Simulation in Airports Using Games Technology," Jun. 2011, pp. 524–533, doi: 10.1007/978-3-642-22000-5_54.

[7] N. Pelechano, K. O'Brien, B. Silverman, and N. Badler, "Crowd Simulation Incorporating Agent Psychological Models, Roles and Communication:," Defense Technical Information Center, Fort Belvoir, VA, Jan. 2005. doi: 10.21236/ADA522128.

[8] G. Solmaz and D. Turgut, "Pedestrian Mobility in Theme Park Disasters," *IEEE Commun. Mag.*, vol. 53, pp. 172–177, Jul. 2015, doi: 10.1109/MCOM.2015.7158282.

[9] J. C. Hsu, J. R. Clymer, J. Garcia, and E. Gonzalez, "Agent-Based Modeling the Emergent Behavior of A System-of-Systems," *INCOSE Int. Symp.*, vol. 19, no. 1, pp. 1581–1590, Jul. 2009, doi: 10.1002/j.2334-5837.2009.tb01036.x.

[10] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The Walking Behaviour of Pedestrian Social Groups and Its Impact on Crowd Dynamics," *PLoS ONE*, vol. 5, no. 4, p. e10047, Apr. 2010, doi: 10.1371/journal.pone.0010047.

[11] M. Moussaïd, D. Helbing, S. Garnier, A. Johansson, M. Combe, and G. Theraulaz, "Experimental study of the behavioural mechanisms underlying self-organization in human crowds," *Proc. R. Soc. B Biol. Sci.*, vol. 276, no. 1668, pp. 2755–2762, Aug. 2009, doi: 10.1098/rspb.2009.0405.

[12] Y. Yuan and W. Zheng, "How to Mitigate Theme Park Crowding? A Prospective Coordination Approach," *Math. Probl. Eng.*, vol. 2018, pp. 1–11, 2018, doi: 10.1155/2018/3138696.

[13] J. E. Almeida, Z. Kokkinogenis, and R. J. F. Rossetti, "NetLogo Implementation of an Evacuation Scenario," p. 5.

[14] N. Bellomo and L. Gibelli, "Toward a mathematical theory of behavioral-social dynamics for pedestrian crowds," *Math. Models Methods Appl. Sci.*, vol. 25, no. 13, pp. 2417–2437, Dec. 2015, doi: 10.1142/S0218202515400138.

[15] E. Oliveira, K. Fischer, and O. Stepankova, "Multi-agent systems: which research for which applications," *Robot. Auton. Syst.*, vol. 27, no. 1–2, pp. 91–106, Apr. 1999, doi: 10.1016/S0921-8890(98)00085-2.

[16] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-Agent Systems: A Survey," *IEEE Access*, vol. 6, pp. 28573–28593, 2018, doi: 10.1109/ACCESS.2018.2831228.

[17] P. G. Balaji and D. Srinivasan, "An Introduction to Multi-Agent Systems," in *Innovations in Multi-Agent Systems and Applications - 1*, vol. 310, D. Srinivasan and L. C. Jain, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 1–27.

[18] E. Sklar, "NetLogo, a Multi-agent Simulation Environment," *Artif. Life*, vol. 13, no. 3, pp. 303–311, Jul. 2007, doi: 10.1162/artl.2007.13.3.303.

[19] U. Wilensky, “Modeling Nature’s Emergent Patterns with Multi-agent Languages,” p. 16.
[20] Fernando Sancho, “A General A * Solver in NetLogo” http://www.cs.us.es/~fsancho/?e=131 October 16, 2018.
[21] Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Appendix A

Agent and environment configuration in NetLogo

A.1 Introduction

This section intends to walk you through the configuration of theme park geometry and agents for the simulation modal.

A.2 Agent Definition

```
breed [elders elder]
elders-own [target]
breed [parents parent]
breed [children child]
breed [fire-alarms fire-alarm]
parents-own [my-child child-found? target]
children-own [my-parent parent-found?]
breed [exits exit]
```

A.2 Variable Definition

```
globals [setp-size fire? parent-colors]
```

A.3 Setting up the environment

```
to setup
  clear-all
  reset-ticks
  set crowded []
  set-default-shape fire-alarms "fire"
  set setp-size 0.01
  set fire? false
  set  parent-colors [
    12 13 14 15 16 17 18 19
    22 23 24 25 26 27 28 29
```

```
    32 33 34 35 36 37 38 39
    42 43 44 45 46 47 48 49
    52 53 54 55 56 57 58 59
    62 63 64 65 66 67 68 69
    72 73 74 75 76 77 78 79
    82 83 84 85 86 87 88 89
    92 96 94 95 96 97 98 99
    102 103 104 105 106 107 108 109
    112 113 114 115 116 117 118 119
    122 123 124 125 126 127 128 129
    132 133 134 135 136 137 138 139]
  draw-green-areas
  create-park-exits
  create-park-events
  draw-restricted-area
  create-agent-elders
  create-child-parent
  set-child
  create-agent-couples
  draw-roads
  draw-coordinate-points
end

to draw-green-areas
 let greenColor 52
 let treeColor 54
 ask patches with [(pxcor >= 33 and pxcor <= 44 and pycor >= 3 and pycor <= 11)
    or (pxcor >= 14 and pxcor <= 24 and pycor >= 22 and pycor <= 26)
    or (pxcor >= 52 and pxcor <= 59 and pycor >= 12 and pycor <= 22)] [
   set pcolor greenColor
  ]
```

```
  ask patches with [pcolor = greenColor] [
    if count neighbors with [pcolor = greenColor] = 8 and not any? turtles in-radius 2[
      ;if random 100 > 90 [
        sprout-trees 1 [
          set shape one-of ["tree" "tree pine"]
          set size 3
          set color treeColor
          stamp
        ]
      ;]
    ]
  ]
end

to create-park-exits
  set-default-shape exits "x"
  ask patches with [pxcor = 60 and pycor >= 28 or pxcor = 0 and pycor >= 28]
                 [set pcolor green
                  sprout-exits 4 ]
  ask patches with [pxcor = 30 and pycor = 0 ]
                 [set pcolor green
                  sprout-exits 4 ]
end

to create-park-events
  create-elders-events
  create-child-events
  create-group-events
end

to create-elders-events
  let meaningVal "elderevent"
  let shapeVal "house"
```

```
  create-event-and-boundary 50 8 3 3 20 meaningVal shapeVal true
  create-event-and-boundary 8 25 3 3 20 meaningVal shapeVal true
end

to create-child-events
  let meaningVal "childevent"
  let shapeVal "crown"
  create-event-and-boundary 30 20 3 3 20 meaningVal shapeVal true
  create-event-and-boundary 10 10 3 3 20 meaningVal shapeVal true
  create-event-and-boundary 0 10 3 3 20 meaningVal shapeVal true
  create-event-and-boundary 10 0 3 3 20 meaningVal shapeVal true
end

to create-event-and-boundary [x y l w shapeAngle meaningVal shapeVal event?]
  ask patch x y [
    sprout-events 1 [
      set shape shapeVal
      set heading shapeAngle
      set color red
      set size ( 1.5 )
      set meaning meaningVal
    ]
  ]
  draw-patch-boundary x y l w meaningVal green event?
end

to draw-patch-boundary [x y l w meaningVal lineColor event?]
  let leftX (x - (l - 1) / 2)
  let rightX (x + (l - 1) / 2)
  let upperY (y  + (w - 1) / 2)
```

```
  let lowerY (y  - (w - 1) / 2)
  ask patches with [leftX <= pxcor and lowerY <= pycor and  pxcor <= rightX and pycor <= upperY] [
   set meaning  meaningVal
  ]
  drawBoundaryCorner leftX upperY 0  lineColor
  drawBoundaryCorner rightX upperY 90  lineColor
  drawBoundaryCorner rightX lowerY 180  lineColor
  drawBoundaryCorner leftX lowerY 270  lineColor
  let startX ( leftX + 1 )
  let endX ( rightX - 1 )
  let startY ( lowerY + 1 )
  let endY ( upperY - 1 )
  while [startX <= endX] [
    drawBoundary startX  upperY 90 meaningVal lineColor
    drawBoundary startX  lowerY 90 meaningVal lineColor
    set startX startX + 1
  ]
  while [startY <= endY] [

    drawBoundary leftX startY 0 meaningVal lineColor
    drawBoundary rightX startY 0 meaningVal lineColor
    set startY startY + 1
  ]
  ask patches with [event? and pxcor > leftX - 3 and pxcor < rightX + 3 and pycor > lowerY - 3 and pycor < upperY + 2 and meaning = 0] [
   set pcolor 2
   set meaning word meaningVal "Boundary"
  ]
end
```

```
to create-group-events
  let meaningVal "groupevent"
  let shapeVal "wheel"
  create-event-and-boundary 40 28 5 3 20 meaningVal shapeVal true
  create-event-and-boundary 20 15 5 3 20 meaningVal shapeVal true
end

to draw-restricted-area
  draw-patch-boundary 40 20 10 5 "restricted" red false
end

to create-agent-elders
  while [count elders < no-of-individuals] [
   ask one-of patches with [meaning = "eldereventBoundary"] [
      sprout-elders 1 [
        set color brown
        set shape "person"
      ]
    ]
  ]
  ask elders-on patches with [(meaning = "exit" or meaning = "childevent" or meaning =
"elderevent" or meaning = "restricted" or meaning = "groupevent")] [
    set status "dead"
   die
  ]
end

to create-child-parent
  set-default-shape parents "person"
  set-default-shape children "person student"
  while [count parents < no-of-parents] [
```

```
    ask vacant-pathes [
      sprout-parents 1 [
        set size 1
        set color  red
        set my-child nobody
        set child-found? false
      ]
    ]
  ]
  while [count children < no-of-parents] [
    ask one-of patches with [meaning = "childeventBoundary"]  [
      sprout-children 1 [
        set my-parent nobody
        set parent-found? false
        set size 1
      ]
    ]
  ]
end

to set-child
  ask parents [
    if (my-child = nobody) [
      let tmp orange
      if empty? parent-colors = false [
        set tmp first parent-colors
        set parent-colors remove-item 0 parent-colors
      ]
      set my-child one-of children with [my-parent = nobody]
      set color tmp
      set size 1.2
      ask my-child [set my-parent myself
        set color tmp
      ]
```

```
    ]
  ]
end

to create-agent-couples
  let tmp 0
  while [tmp < no-of-couples ] [
    create-agent-couple
    set tmp tmp + 1
  ]
end

to create-agent-couple
  let vacant-patch vacant-pathes
  ask one-of patches with [ meaning = "groupeventBoundary"] [
    sprout-couples-male 1 [
      set shape "person business"
      set size 1
      set female nobody
      set color blue
      fd 1
    ]
    sprout-couples-female 1 [
      set shape "person business"
      set size 1
      set male nobody
      set color pink
    ]
  ]
  ask couples-female with [male = nobody] [
    set male one-of couples-male with [female = nobody]
```

```
    show male
    ask male [ set female  myself
     show female]
  ]
end

to draw-roads
  draw-road
end

to draw-road
  ask patches with [(pycor = 25 and pxcor >= 50 and pxcor <= 59) or (pycor = 27 and pxcor
>= 45 and pxcor <= 50)
    or (pycor = 15 and pxcor >= 25 and pxcor <= 49) or (pycor = 4 and pxcor >= 2 and pxcor
<= 29)
    or (pycor = 20 and pxcor >= 9 and pxcor <= 26) or (pycor = 29 and pxcor >= 2 and pxcor
<= 35)] [
    sprout-roads 1 [
      set shape "road"
      set heading 0
      set color grey
      stamp
    die
    ]
  ]
   ask patches with [(pxcor = 50 and pycor >= 11 and pycor <= 27) or (pxcor = 50 and pycor
>= 11 and pycor <= 27)
    or (pxcor = 41 and pycor >= 16 and pycor <= 17) or (pxcor = 30 and pycor >= 16 and
pycor <= 16)
    or (pxcor = 30 and pycor >= 2 and pycor <= 14) or (pxcor = 20 and pycor >= 5 and pycor
<= 11)
    or (pxcor = 10 and pycor >= 3 and pycor <= 6)  or (pxcor = 2 and pycor >= 5 and pycor <=
6)
```

```
    or (pxcor = 8 and pycor >= 20 and pycor <= 21) or (pxcor = 8 and pycor >= 28 and pycor
<= 28)] [
    sprout-roads 1 [
      set shape "road"
      set heading 90
      set color grey
      stamp
    die
    ]
  ]
end

to draw-coordinate-points
  draw-coordinate-point min-pxcor + 1 min-pycor  3 3
  draw-coordinate-point max-pxcor - 1 min-pycor  3 3
end

to draw-coordinate-point [x y l w]
  let meaningVal "help"
  ask patch x y [
    sprout-helps 1 [
      set shape "help"
      set heading 0
      set color yellow
      set size ( 1.5 )
      set meaning meaningVal
    ]
  ]
  draw-patch-boundary x y l w meaningVal green false
end
```

Appendix B

Generation fire emergency

B.1 Introduction

This section intends to walk you the NetLogo coding that generates the fire emergency in the simulation

B.1 Generation of fire emergency

```
to make-fire
  set fireXCor 30
  set fireYCor 15
  let t fire-thresshold
  ask patches with [pxcor > fireXCor - t and pxcor < fireXCor + t and pycor > fireYCor - t
and pycor < fireYCor + t] [
   set meaning "fire"
   set pcolor pink - 1
  ]
  if (fire? = false) [
    ask patch fireXCor fireYCor [sprout-fire-alarms 1 [
      set size 2
      set color red
      ]
    ]
     ask turtles-on patches with [meaning = "fire"] [
    set color red
  ]
  ask parents-on patches with [meaning = "fire"] [
    ask my-child [
       set color white
        set my-parent nobody
        stamp
        set status "dead"
```

```
      die
    ]
    set status "dead"
    die
  ]
  ask children-on patches with [meaning = "fire"] [
    ask my-parent [

      ;set breed elders
      set my-child nobody
      set color white
      stamp
        set status "dead"
      die
    ]
      set status "dead"
   die
  ]
    let selected turtles with [breed = elders or breed = parents]
    set-closest-exit turtles with [breed = elders or breed = couples-male]
    set fire? true
  ]
  ask turtles-on patches with [meaning = "fire"] [
    if (breed = elders or breed = children or breed = parents or breed = couples-male or breed =
couples-female ) [
      set status "dead"
      die
    ]
  ]
end
```

Appendix C

Program to exit agents from theme park

C.1 Introduction

This section intends to walk you through the NetLogo coding algorithms to exit agent from the environment.

C.2 Move to closet exit

```
to go-to-exit [ agents ]
  ask agents [ face target]
  ask agents [
    if breed = parents and child-found? = true  and  meaning = "exit" [
      ask my-child [
        set status "dead"
        die
      ]
      set status "dead"
      die
    ]
    if breed != parents and meaning = "exit" [
      set status "dead"
     die
    ]
    let i 0
    while [obstacle? and i < 4] [

      set heading (heading + i * 10)
      set i i + 1
    ]
    ifelse no-agent-met?  [
      fd setp-size
      set waitCount 0
    ][
      set waitCount waitCount + 1
```

```
    rt random 360
    fd setp-size
   ]
  ]
end
```

C.3 Finding closet exit

```
to set-closest-exit [ agents ]
  ask agents [
   let agentXCor xcor
   let agentYCor ycor
   let selected exits;
   ifelse fireXCor < agentXCor [
    set selected selected with [xcor >= agentXCor]
   ] [
    set selected selected with [xcor <= agentXCor]
   ]
   ifelse fireYCor < agentYCor [
    set selected selected with [ycor >= agentYCor]
   ] [
    set selected selected with [ycor >= agentXCor]
   ]
   set target min-one-of selected  [distance myself]
  ]
  ask agents with [target = nobody] [
   set target min-one-of exits  [distance myself]
  ]
end
```

C.4 Parents to search children

```
to find-child
  set total-search-count total-search-count + 1
  face my-child
  let found-neigh turtles-on patches in-radius 1
```

```
if any? found-neigh and member? my-child found-neigh [
 ask my-child [set parent-found? true]
 set child-found? true
]
if child-found? = false [
 ifelse path-to-child != 0 and path-to-child != false  [
  if next = 0 or patch-here = next and path-to-child != false [
   set next first  path-to-child
   set path-to-child remove next path-to-child
  ]
  ;show next
  face next
  ifelse no-agent-met? [
   fd setp-size
  ] [
    let i 0
    while [no-agent-met? = false and i < 360] [
     set heading (heading + 1)
     set i i + 1
    ]
   fd setp-size
  ]
 ] [
  let i 0
  while [obstacle? and i < 4] [
   set heading (heading + i * 10)
   set i i + 1
  ]
  if i > 8 [
   set seach-count seach-count + 1
```

```
      ]
      if no-agent-met? [
        fd setp-size
      ]
    ]
  ]
  ifelse patch-here = currentPatch [
   set waitCount waitCount + 1
  ] [
    set currentPatch patch-here
    set waitCount 0
  ]
end
```

C.6 Children to move with parents

```
to follow-parent
  ifelse  my-parent = nobody [
    set status "dead"
    die
  ] [
    face my-parent
    let gap distance my-parent
    while [gap > 0.5] [
      fd setp-size
      set gap distance my-parent
    ]
  ]
end
```

C.7 Coordinators to move help requested agent

```
to move-to-help-requester
 if patch-here = lostAgentPatch [
   set helped? true
   set helping? false
   show lostAgent
   if lostAgent != nobody [
     ask lostAgent [
       calculate-path-to-child
     ]
   ]
   set status "dead"
   die
 ]
 ifelse next2 = 0 or patch-here = next2 and not helped?  [
   if is-list? path-to-help [
     set next2 first  path-to-help
     set path-to-help remove next2 path-to-help
   ]
 ] [
   face next2
   fd setp-size
 ]
end
```

C.6 Coordinators to move crowded area

```
to move-to-crowded-area
   ifelse target != 0 and patch-here = target [
   let i xcor
   let j ycor
   set target 0
   ask turtles-on patches in-radius 3 [
     if breed = couples-male or breed = elders [
```

```
        let selected min-one-of exits  [distance myself]
        ifelse target != selected [
          set target selected
        ] [
          let from patch-here
          let t 0
          show word "patche here " patch-here
          let selectedExits exits
          set selectedExits selectedExits with [xcor <= pxcor]
          let selectedExit min-one-of exits  [distance myself]
          show selectedExits
          let x xcor
          let y ycor
          move-to patch (2 * i - x) (2 * j - y)
        ]
      ]
    ]
    ] [
      face target
      fd setp-size * 4
    ]
end
```

C.8 Coordinators to check crowded areas

```
to check-crowded
  let stepSize 3
  let i min-pxcor
  while [i <= max-pxcor - stepSize ] [
    let j min-pycor
     while [j <= max-pycor - stepSize ] [
      if is-area-crowded i j [
        let key word i j
        let used member? key crowded
```

```
        if not used [
          set crowded lput key crowded
          let station patch min-pxcor min-pycor
          if i > max-pxcor / 2 [
            set station patch max-pxcor min-pycor
          ]
          ask station [
            sprout-crowdcordinators 1 [
              set shape "person police"
              set target patch i j
            ]
          ]
        ]
      ]
      set j j + 4
    ]
    set i i + 4
  ]
end
to-report is-area-crowded [i j]
  let crowdedThresshold 10
  let stepSize 3
  let agent-set turtles-on patches with [pxcor >= i and pxcor < i + stepSize and pycor >= j and
pycor < j + stepSize  ]
  let density count agent-set
  if density > crowdedThresshold [
    let agent-list sort agent-set
    let movingAgents filter [ s -> [breed] of s = elders or [breed] of s = parents or [breed] of s
= couples-male or [breed] of s = couples-female ] agent-list
    if length movingAgents > crowdedThresshold [
      report true
    ]  ]
  report false
end
```

Appendix D

Simulation Output

Figure D.1 is the simulation view while exiting the agent from fire following the defined rules.

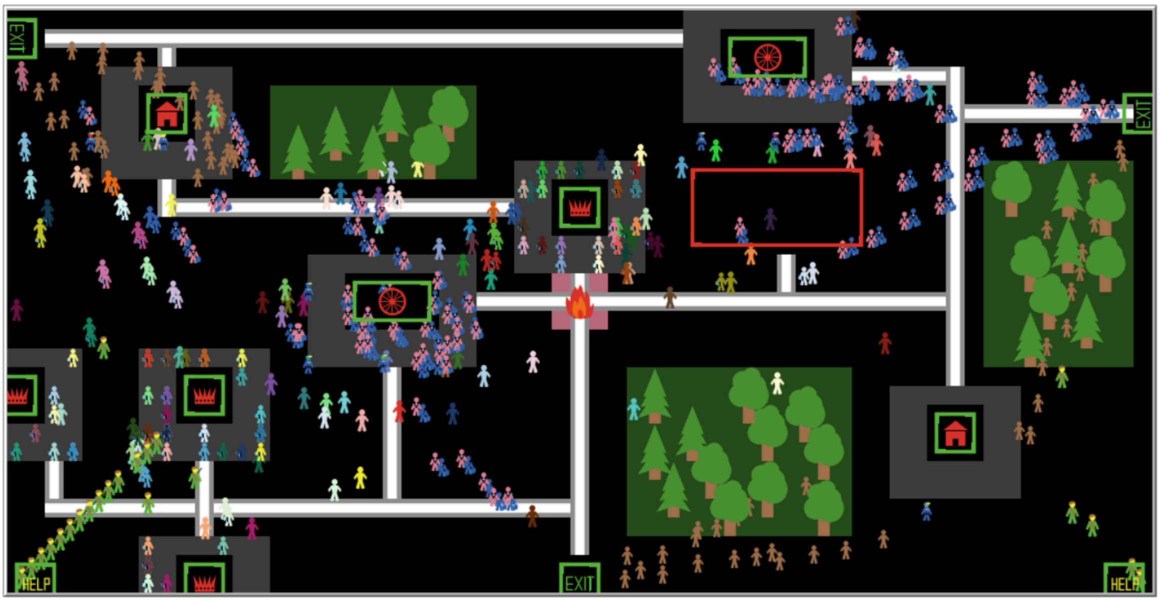


Figure – C.1 Simulation view