Eric Hao

EECS 372 – Uri Wilensky

June 8, 2018

Final Report

Emergency Evacuation

Introduction

Emergencies are inevitable. The aftermath can often be quite awful and horrendous. They can be so unpredictable yet so devastating. As a result, handling and preparing for emergencies are a key part of attempting to mitigate these outcomes. Emergency protocol is a very common aspect of American culture. As children growing up in the American education system, we experience at least annual, perhaps monthly fire drills that are both unrealistic and painfully boring. Evacuation plans are designed painstakingly to maximize efficiency, yet very few people know their building's evacuation plan at any given moment. Furthermore, in real emergencies, panic is a very real phenomenon that is often much stronger than reason. Therefore, instead of attempting to manipulate and control human behavior, it may be more productive and helpful to design architecture in a reactionary effort, given the panicked behavior most people devolve into when they feel their life is truly in danger.

Crowd behavior seems to change completely, almost incredibly, once the threat of an emergency looms. People tend to become selfish. They grow impatient extremely quickly. They push and shove and yell. They even trample others, sometimes literally killing to secure their own safety. This type of response is very natural and real, but is incredibly hard to study, as it basically required real humans facing real danger, or at least legitimately believing they are facing real danger. Thus, this paper describes an attempt to model this behavior heuristically.

Motivation

Much of the motivation behind modeling the emergent phenomenon of panicked evacuation comes directly from its importance. This phenomenon directly impacts everyone's lives during an emergency, and a greater understanding of it can always further prevention efforts for future emergencies. If we can understand crowd evacuation better, we can better understand what to expect during an emergency, how to handle it, and how to design buildings and towns to facilitate efficient evacuation given a crowd's unpreventable panic. In addition, there are numerous other applications for studying crowd behavior, including computer simulation, computer vision, swarm intelligence, and video games.

Background

Being a phenomenon with such important implications, there have been multiple related studies and research work in this area. Crowd Evacuation Simulation is an active field of research with a myriad of global contributions presenting interesting modeling and simulation techniques. Bakar et al. provides a survey of work done in this area in *An Overview of Crowd Evacuation Simulation*. A subset of this research has approached this topic from the perspective of Agent Based Modeling, which is especially relevant to this project. Almeida et al propose a framework for furthering development of evacuation simulations. According to them, "Multi-Agent Systems is stated as the preferred approach for emergency evacuation simulations." Work has also been done using flow-based modeling and cellular automata (Almeida, João E, et al.).

Within crowd evacuation, several specific behaviors have been discovered, identified, and studied. In emergency situations, people attempt to exit the building as fast as possible, resulting in higher velocities of movement. Comfort and convenience become unimportant, and sometimes, in panic, people end up moving toward exits using an inefficient path. Also, people may forget where the exits are located, leading to a herding behavior as they attempt to follow crowds to exit the building.

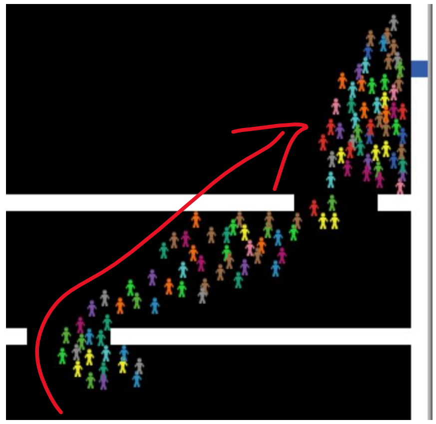

Figure 1: Herding behavior

Because everyone is desperately trying to move through a small exit at the same time, this results in a circular clogging phenomenon, which is depicted below.

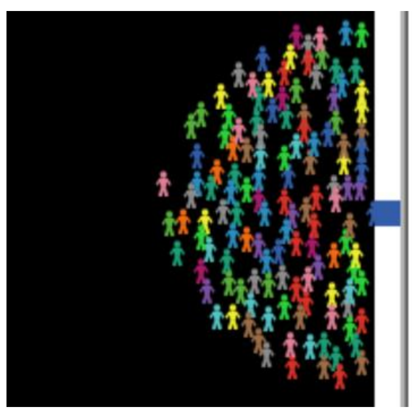

Figure 2: Clogging phenomenon

Design - Overview

The fundamental cornerstone of this model is a heuristic model that determines how people choose where to go next. It was written using NetLogo software, where patches represent parts of the building and turtles represent the people evacuating. Patches can represent one of three things: a wall, ground, or exit. The building architecture is initialized in setup, and patches are colored according to their type. After setup, patches are static and do not change.

Design – Model Rules

The rules listed may not seem that complex, but there are quite a few implementation details to consider, and the actual heuristic components also need to be introduced. The rules for the model are the following:

To Initialize:

- Create walls and doors based on BUILDING-TYPE
- Create NUMBER-OF-PEOPLE people at random ground locations

At each tick, each person does:

- Find a target:
    - Find and select a visible exit. If none, then
    - Find and select a visible checkpoint. If none, then
    - Find and select a nearby visible and vacant ground patch. If none, then
    - Do nothing.
- Select a nearby patch (immediate-target) based on target (if one was found).
- Move toward immediate-target.

At each tick, each patch does nothing.

Design – Visibility

Turtles need to know what is classified as "visible." For example, exits may be blocked by walls, and anything blocked out of view should not be considered as part of that turtle's potential options. The justification for this is intuitive. In the real world, people cannot consider options they cannot see. Even if they know about an exit they cannot see, they will not go directly toward it if something is in the way. They must first find a way around any obstacles. Visibility was implemented using ray-casting for each turtle. First the current state of the turtle is saved, then the turtle acts as the “ray” and move towards each potential target slowly, checking whether there are obstacles in the way. Afterward, the original state of the turtle is restored, so in the model, ray-casting is never visible to the observer.

Design – Exits and Checkpoints

As hinted at in the model rules, there are three sets of potential targets considered, in order of priority. Exits are prioritized above checkpoints, which are in turn prioritized above regular ground patches. Turtles are given pre-existing knowledge of the exist in the room. This is a slight simplification which is not always true in the real world, but it simplifies the model by eliminating the need to deal with memory. If turtles began with a non-perfect understanding of where exits were, they would also need to retain memory of exits they see as they move through the building. Checkpoints are invisible patches which mark potential paths around obstacles. They are initialized manually in setup and are commonly placed within gaps in walls as targets turtles can prioritize if they do not see an exit. Using checkpoints is an alternative to using a path-finding algorithm like A* or BFS, which is more complicated both conceptually and computationally. In addition, using checkpoints is more natural than a pathfinding algorithm, since in real evacuation scenarios, people are not able to run perfect pathfinding algorithms in their head since they cannot see the entire building layout. Instead, they see nearby opportunities like hallways and

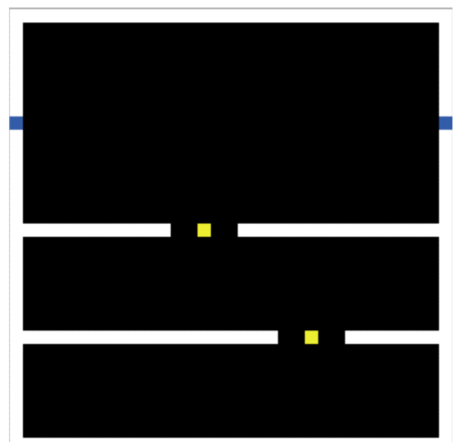

Figure 3: Yellow checkpoints

openings, which are conceptualized as checkpoints in the model. Finally, if no checkpoints can be found either, turtles choose from patches near them.

Design – Heuristic Selection

This is the most important part of the model and determines how turtles choose their targets each tick. There are five main heuristics used to aid in scoring patch desirability. Given a patch to consider as my target:

1. Crowdedness: How crowded is the patch?
2. Wall proximity: How many walls are near the patch?
3. Distance from me: How far away is the patch from me?
4. Distance from exits: How far is the patch from the nearest exit?
5. Crowd heading: How different would my heading be toward that patch compared to the nearby crowd's heading?

The rationale behind these heuristics is that each of them is crucially relevant to a person attempting to evacuate a building. Crowds might be bad if they are crowding around exits but might be good if someone doesn't know where the exits are. Moving directly toward walls is usually not helpful. Distance from me matters because having a target further away can indicate more progress hopefully toward an exit, if none are visible. Distance from exits is trivially relevant. Finally, considering the direction a crowd is headed is relevant especially if someone does not know where exits are. This is what produces the herding behavior mentioned previously.

It is very important to detail how these heuristics are applied, especially since they are considered differently when considering different types of targets, like exits as opposed to checkpoints. The application of these heuristics for each type of target is listed below in order of weight. Note that the heuristic values are not normalized, so even if the weight for one heuristic is greater than the weight of another, that does not mean the former heuristic is valued more heavily over the other.

Computing exit desirability:

- Crowdedness is bad (Heuristic 1 with weight -1)
- Closer exits are better (Heuristic 3 with weight -1)

Computing checkpoint desirability:

- Closer to exits is better (Heuristic 4 with weight -100)
- Going against the crowd is bad (Heuristic 5 with weight -5)
- Crowds are good (Heuristic 1 with weight 1)
- Closer checkpoints are better (Heuristic 3 with weight -1)

Computing regular patch desirability:

- Walls are very bad (Heuristic 2 with weight -3)
- Crowds are good (Heuristic 1 with weight 2)
- Further away is better (Heuristic 3 with weight 1)

The heuristics were developed based on conceptualizing what a real person in a real emergency would consider as progressing toward evacuation. Then the magnitude of weights was tuned manually until some degree of realistic behavior was realized.

Two remaining points to address are related to heuristic calculation. Most of the heuristics are calculated as one would expect. However, heuristic 4 is the sum of both the mean and min distances to exits. Incorporating both mean and min counters the possibility that different patches share the same mean distance to exits. Also, crowdedness counts the number of turtles in a certain radius but does not check visibility of those turtles. This is an unrealistic simplification but is countered by the unattractiveness of walls that might be blocking those counted turtles.

Results

Three experiments were run focusing on door proximity, hallway size, and number of rooms. Each experiment was run separately with 30 repetitions per configuration, while only modifying the parameter being tested. Measures collected are total evacuation time and average evacuation time. The door proximity experiment was run with the following configuration:

["door-proximity" 1 3 5 7 9 11 13 15 17 19]

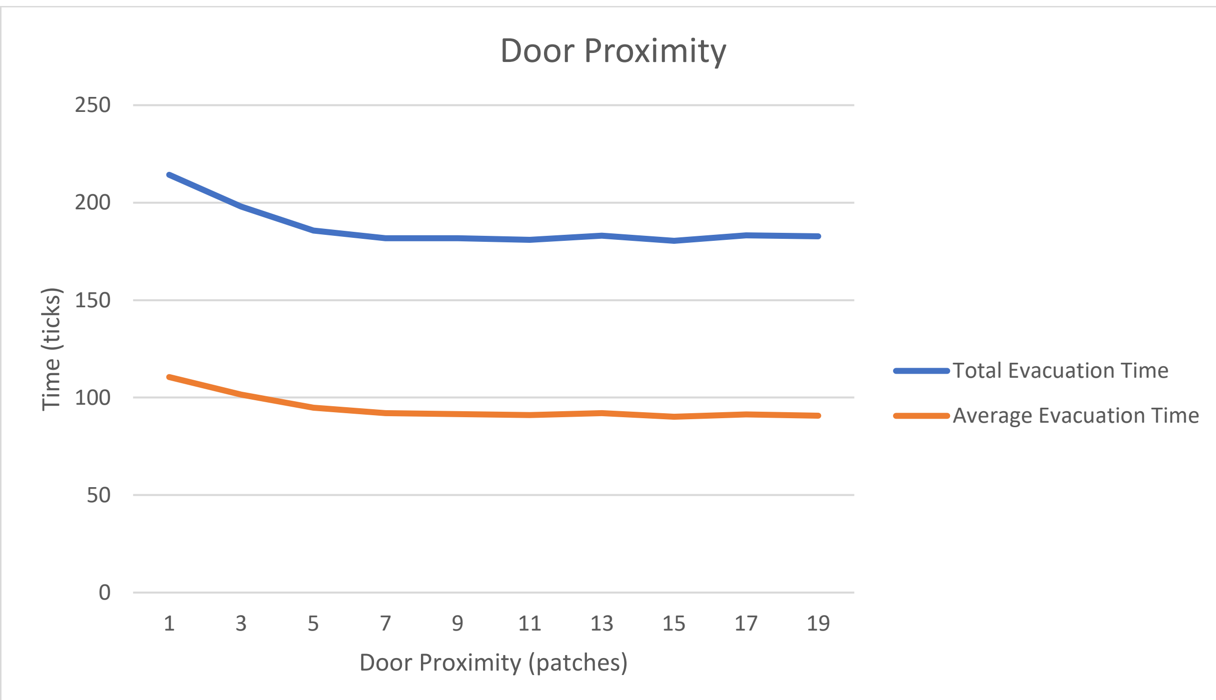

Figure 4: Door Proximity BehaviorSpace Results

The hallway size experiment was run with the following configuration:

["hallway-size" 3 5 7 9 13 15 17 19 21]

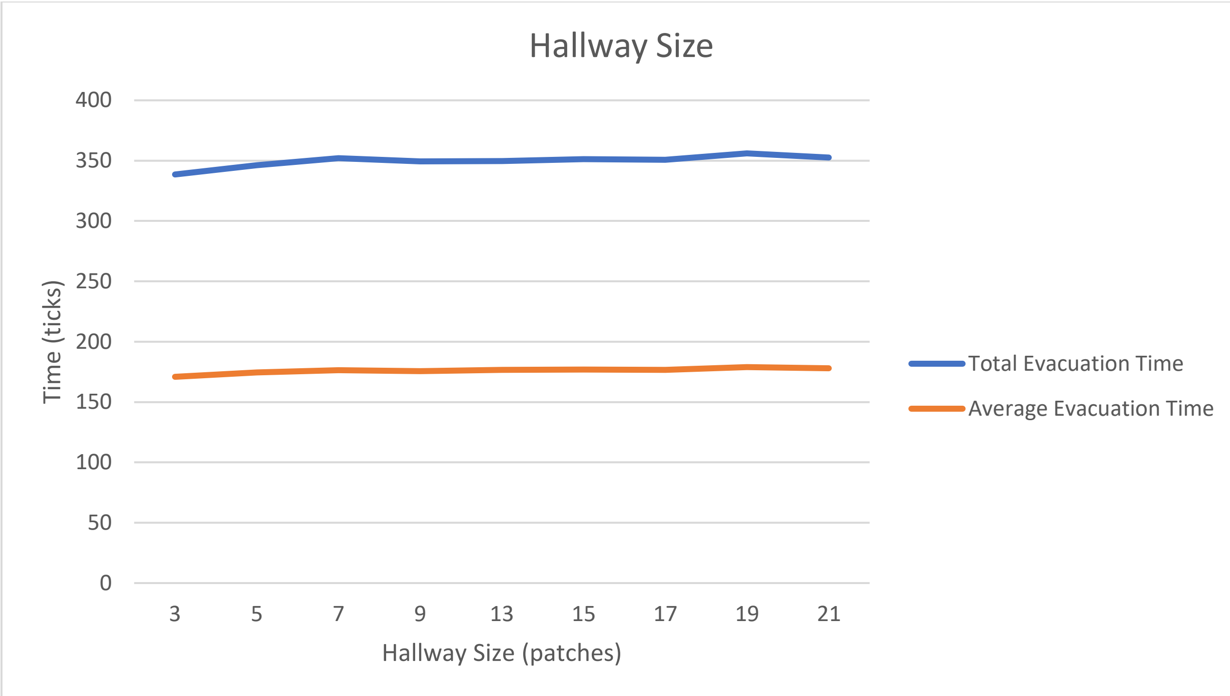

Figure 5: Hallway Size BehaviorSpace Results

The number of rooms experiment was run with the following configuration:

["number-of-rooms" 0 2 4 6 8 10]

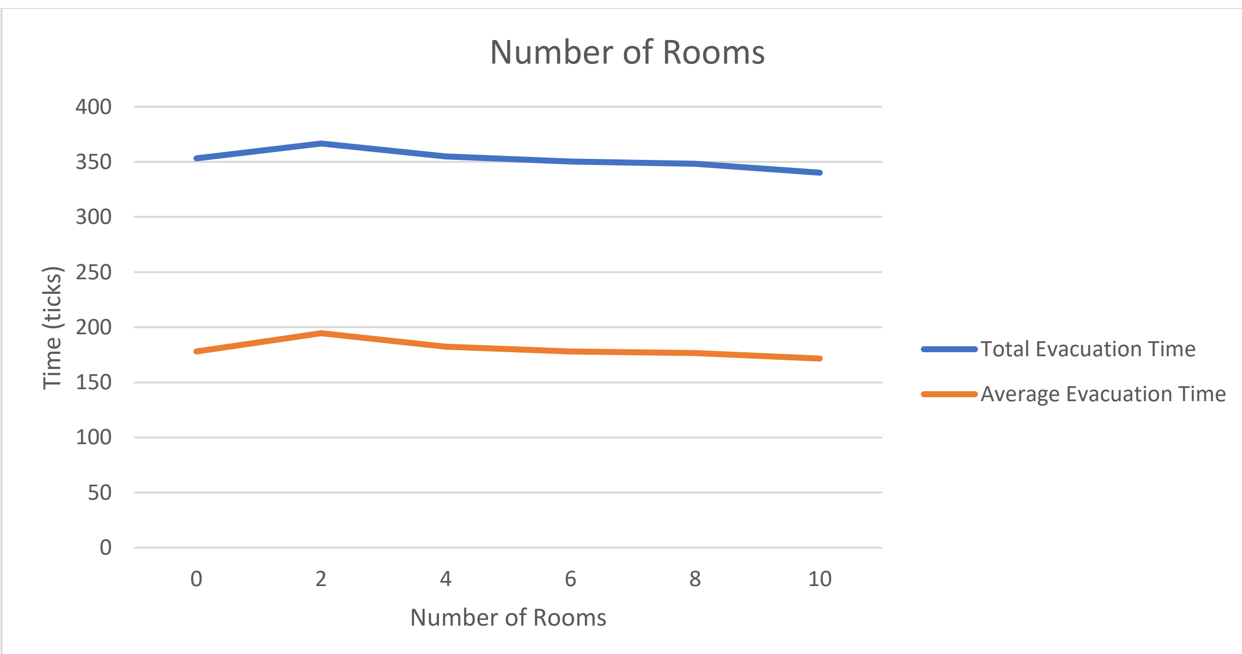

Figure 6: Number of Rooms BehaviorSpace Results

Discussion

The door proximity experiment shows a decrease in evacuation time as the door grows further and further apart. But more than around 7 patches apart, the evacuation time seems remain unchanged. This is probably because the circular clogging phenomenon forms a single circle around the exits when they are close to each other, but at some point, the exits are far enough apart with respect to the number of people that two separate circles form around the exits. 19 was the maximum DOOR-PROXIMITY value allowed by the model as restricted by maximum HALLWAY-SIZE, but analyzing the behavior with respect to the doors, one can observe that between two exit crowds, there happens to be a natural redistributing of people from one exit to another periodically, whenever one crowd gets significantly smaller than the other. As DOOR-PROXIMITY approaches 19, we begin to see more of this, but the crowds are still relatively close. If this pattern continues, it is possible that redistribution between the crowds takes longer and longer, potentially resulting in an increase in evacuation time for exits that are too far apart. A quick benchmark using the “open room” BUILDING-TYPE results in an average total evacuation time of 190.3 over 10 iterations, which may suggest a bit of that upward slant.

Hallway size seems to have little effect on evacuation time. In the model, smaller hallway-size results in larger rooms. These results suggest that waiting inside the room longer before joining the crowd is just as good as entering the hallway sooner, which is a bit unintuitive, as one might think that getting into the hallway as soon as possible is better and would result in a faster evacuation.

For the number of rooms experiment, there is a subtle but clear peak when NUMBER-OF-ROOMS is 2. When there are only two rooms (one on each side), there is only one opening from those rooms into the hallway, which means that people on the far left or right inside the rooms need to travel further to first exit the room, then make their way to the exit. This could explain the slight peak. There is also a downward trend in evacuation time as NUMBER-OF-ROOMS gets higher. This might be because more rooms require more walls, which contributes to more blocking.

Conclusion

The bulk of the work of this project was spent developing and implementing the basic functionality of the model itself. There are many ways it can or should be extended to increase the potential for more interesting experimental results as well as emergent behavior and phenomena. In conclusion, here are a few thoughts in that direction:

- Currently everyone has initialized knowledge of where exits are. What if they did not?
- There are plenty of other building types left unexplored. Some buildings have rooms in the center. Many are not square-shaped. Designing better building types also leads to better experimentation.
- Many buildings have tables and other miscellaneous obstacles that are not necessarily walls.
- Is it possible to build a complete heuristic model without using checkpoints that still exhibits realistic evacuative behavior?
- The weights for the heuristic model can be turned into tunable parameters, which also may facilitate interesting experiments.

Works Cited

Almeida, João E, et al. "Crowd Simulation Modeling Applied to Emergency and Evacuation Simulations Using Multi-Agent Systems." arxiv.org/ftp/arxiv/papers/1303/1303.4692.pdf.

Bakar, Noor Akma Abu, et al. "A Simulation Model for Crowd Evacuation of Fire Emergency Scenario." *2017 8th International Conference on Information Technology (ICIT)*, 2017, doi:10.1109/icitech.2017.8080027.

Bakar, Noor A, et al. *An Overview of Crowd Evacuation Simulation*. American Scientific Publishers, 2017.

Zheng, Xiaoping, et al. "Modeling Crowd Evacuation of a Building Based on Seven Methodological Approaches." *Building and Environment*, vol. 44, no. 3, 2009, pp. 437–445., doi:10.1016/j.buildenv.2008.04.002.