CX 4230 Simulation Project - Rohan Varshney, Dheeraj Bandaru, Rishab Mitra - Spring 2021

Git Repo: https://github.com/rohanvarshney/EvacuationModSim

Abstract

In creating an ideal movie theater, one must not just account for how many seats and how much revenue each theater screen brings in - there must also be safety guidelines and evacuation plans set. Thus, the goal of this model is to show an optimal movie theater that's able to have the max amount of seats while abiding safety guidelines**.** The scope of this project will be limited to one type of evacuation - fire. The amount of casualties in case of fire must be zero. However, a goal for a movie theater is to maximize the amount of revenue, as well. Therefore, the movie theater must have the max amount of seats, but also enough exits or space to create a safe evacuation with no casualties.

Feature Description

The features that will be used and manipulated to get our output of evacuation time and total casualties for our simulation and modeling of the movie theater will be selected to best encapsulate all the major factors that go into evacuation routes in any big congregation of people. The main features that will be used will be the capacity of the room and the number of exits present in the room. This is a very key detail that needs to be manipulated with and determined to keep with reality we will limit the number of exits relative to the capacity of theater as it is unrealistic to have 1:1 ratio of people and exits although that would be optimal for evacuation times and reducing the number of casualties in the event of a tragedy. Another feature that is essential to our simulation is the percentage of seats occupied in the theater. This is a very important feature that will determine the flow rate of movement within the theater as well as through the exits. Another feature that will be used that will affect the flow rates for movement will be the average walking speed of a patron of the movie theater. With inclined surfaces and darker lighting than other public venues walking speed can vary and be less than the average walking speed and so should be accounted for and be a factor in the simulation to get more realistic results. The location of the exits will also be used as a feature and we will create a set of locations to experiment and test the set of locations for the different number of exits. For example if we have 4 exits they can be placed either in the midpoints of the room or the corners of the room and we test both instances and see which one is most effective in increasing the evacuation rate. We will also have a feature to manipulate the rate at which the threat is approaching the patrons and the starting point of the threat. This feature can have a very direct impact on the evacuation path and the rate at which the evacuation occurs.

Data For Features

In order to create the features listed above and get the best possible results and simulations we will need to work on creating the best possible data protocols for the creation of the data for the features. The data that will be created to get the capacity of the room and the number of exits present in a room will a set of tuples that provide an integer ranging between 50 and 100 to represent the maximum capacity of the movie theater and the second part will be an integer of the capacity mod 4 to get a value for the number of exits in a given theater. We will also create the percentage filled of a movie theater by using a random number between 1 and 100 as the percentage filled.

We will create the flow rates for the movement by using the chart indicated from the "Modeling and simulation of large crowd evacuation in hazard- impacted expected" by Datta and Behzadan which is also presented below. Using those averages we will assume 2 different ratios of male to female to child. The first case is that of a children movie where the ratio will be 1:1:3. The other case is an adult movie which will be 2:2:1. Using these ratios and the average velocities of the population we will determine the flow rate of movement.

Table 1 Velocity distribution of different evacuee class (Shi *et al.* 2009)

Evacuee Class (attributes)	Mean (m/s)	Std. Deviation (m/s)
male, child, able	1.08	0.26
male, child, disabled	0.92	0.34
male, adult, able	1.24	0.45
male, adult, disabled	1.06	0.26
male, elderly, able	1.05	0.15
male, elderly, disabled	0.91	0.13
female, child, able	1.08	0.26
female, child, disabled	0.92	0.34
female, adult, able	1.30	0.38
female, adult, disabled	1.06	0.26
female, elderly, able	1.04	0.16
female, elderly, disabled	0.89	0.14

And finally we will determine the location of the exits using a binary system where 0 would indicate the exits are located in the corners and the 1 would indicate that the corners are located at the midpoints of the boundaries.

Furthermore, one other agent that must be modeled in the simulation is the hazard itself. There are many hazards that can occur in a movie theater, but the project will focus on only one type of hazard - fires. This is because fires follow a certain mathematical model in terms of rate of spread. The mathematical model is stochastic due to the fact that the conceptual model is a cellular automaton. The model would be similar to the rules given in Conway's Game of Life, where probabilities are given if a cell is consumed by fire.

Data Creation

While there are ways to procedurally generate maps in the form of two dimensional rooms, unfortunately a movie theater must be three dimensions due to the multiple floors found in a movie theater. Also, movie theaters have a specific constraint in terms of the layout of how many seats there are per row. Thus, it would be easier to create 3 custom movie theater maps, one small size, one regular, and one large. However, the population of the movie theater will be randomly generated given the demographics in the previous section. The only attribute that the population will have is each individual's velocity. The hazard will always be a fire, but the fire's placement will also be random. However, the fire will always have the same rate of spread, but the direction a fire goes will be random. Using all of these agents and parameters, one way to determine the most optimal movie theater in terms of safety is through grid search. A performance metric that can be used to determine if a movie theater has a high safety score is through a metric called Likelihood of Survival (LS). This metric is defined as the following:

$$LS = Number\ of\ survived\ agents / Number\ of\ total\ agents$$

The highest LS score will be used to determine the most optimal movie theater. The adjustment of parameters can be done using a Grid Search method with a max and min amount for each parameter.

Behavioral Information

Some behavioral information related to this simulation has to do mainly with the population that fills up the theater. We also have to consider the behavior of the hazard and what characteristics might affect how we model those behaviors.

Below we have created our own table of velocity distributions based off table 1 from the previous section we have condensed the amount of evacuee classes to reduce computational costs for our simulation and we have used a 90-10 weighting to represent both the able bodied and disabled populations in the mean and we took the average standard deviations between the populations for our experiments. We do not account for elderly due to our considerations about computational costs and the behavioral characteristics between elderly and adults being fairly similar. We also do not consider gender when looking at the child class as the mean and standard deviation for a child is the same regardless of gender in the original data.

Table 2 Velocity Distributions of different evacuee classes

Class	Mean (m/s)	Std. Deviation (m/s)
Male, Adult	1.22	.36
Female, Adult	1.28	.32
Child	1.06	.30

The hazard we want to model will be a fire. Fire often starts in different locations and moves at varying speeds so we will create an array of starting positions for our simulation and an array of different starting speeds and rates at which the fire speeds up. Some other characteristics that we want to incorporate into our simulation include the direction in which the fire spreads; we want it to go in multiple directions and we will use the starting location to determine which directions it will proceed initially.

Platforms for Development

For the development of the conceptual model of the system, currently the model is being built in Python in a notebook environment. Currently no external libraries are being used to create the simulation of a movie theater as we are currently using 2-D arrays to represent the movie theater. However, for evaluation metrics we may have to use matplotlib to create linear graphs or bar charts.

Conceptual Model

The conceptual model of the movie theater can be defined as the following. The movie theaters can be represented as 3D arrays, and each layer of the array can be considered as a single floor of the movie theater. There can be specific values for each element in the array depending on what type of space the element in the array represents. For example, a 0 can represent that a space is open, a 1 can represent an exit, a 2 can represent a wall, and a 3 can represent the fire hazard. Thus, the agents in the room will want to go to a 1 to escape the hazard. An assumption that is being made for the project is that the entire population in the room will want to go to the nearest exit, and that each agent will take the most optimal path in doing so. Also one other note is that no two agents can occupy the same space, so an agent must wait if another agent is blocking its path. As for the agents, the only attribute that each agent has is their velocities which represents how many array elements can one agent traverse in each time step. The hazard will also have the same rate of spread throughout all simulations. However, where the hazard first starts will be an arbitrary location. Also, the directions in which the hazard spreads will also be random. In terms of evaluation, to rate each simulation there will be two metrics that will be used - the Likelihood of Survival score mentioned in the previous section and the time step where all agents have escaped or been lost to the hazard. If two simulation runs share the same Likelihood of Survival score, then the tie-breaking metric will be the time step. The run with the smaller time step will be considered the most optimal.

Current State of the Project

Currently we have made an initial modeling attempt of the evacuation and escape of people given a certain room. We have created two rooms for testing and debugging so far. Each person moves at the same pace and hazards have not been implemented yet. However, we are able to successfully randomize the placement of people in the room and simulate the exit of all people in a given room. The time it takes for all people to evacuate from a room is outputted. Some code snippets of the simulation and its output are pasted below.

```
In [164]: # ROOM 1
          # One exit in a walled empty room.
          room_1_map = [
              [1, 1, 1, 1, 1, 1, 1, 1, 4, 1],
              [1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
              [1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
              [1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
              [1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
              [1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
              [1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
              [1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
              [1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
              [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
          ]
          room_1 = Room(room_1_map)
          room_1_exits = room_1.get_element_coordinates_of_map(RoomElement.EXIT)

          paths = room_1.paths_to_a_goal(Coordinate(3, 7), room_1_exits[0])

          print('Paths: ' + str(paths))
          print('Best Path: ' + str(room_1.best_path(paths)))


          room_1_empty_spots = room_1.get_element_coordinates_of_map(RoomElement
          .EMPTY)
          room_1_empty_spots_subset = random_elements_of_list(room_1_empty_spots
          , len(room_1_empty_spots), int(len(room_1_empty_spots)/10))
          print('Character Spots: ' + str(room_1_empty_spots_subset))

          simulation1 = EvacuationSimulation(room_1, room_1_empty_spots_subset)
          simulation1.simulate()
```

```
    def simulate(self):
        personList = []
        for coordinate in self.peopleCoordinates:
            personPath = self.get_path_based_on_current_coordinate(coo
rdinate, self.room)
            newPerson = Person(coordinate, personPath)
            personList.append(newPerson)

        print('Initialize simulation')
        print(self.room.print_current_map_spots(self.peopleCoordinates
))
        print(personList)
```

388/nbconvert/html/Google%20Drive/GitHub/EvacuationModSim/CX4230_Evacuation_ModSim.ipynb?download=false Page 5 of

n_ModSim 4/8/21, 3:06 P

```
        time = 0

        # Start Loop

        while not self.everyone_has_exited(personList):
            locationDict = {}
            for person in personList:
                if (person.next_spot_in_path() is None):
                    continue
                nextPlaceInPath = person.next_spot_in_path()
                print("Next spot: " + str(nextPlaceInPath))

                if nextPlaceInPath in locationDict and locationDict[ne
xtPlaceInPath] is not None and len(locationDict[nextPlaceInPath]) > 1:
                    oldList = locationDict[nextPlaceInPath]
                    newList = oldList.append(person)
                    locationDict[nextPlaceInPath] = newList
                else:
                    locationDict[nextPlaceInPath] = [person]


            print(locationDict)

            for key in locationDict.keys():
                print(key)
                print(locationDict[key])
                if locationDict[key] is None:
                    continue
```

```
Time: 11
Coordinates
[<0,8>, <0,8>, <0,8>, <0,8>, <0,8>, <0,8>]
[[    1    1    1    1    1    1    1    1 9999    1]
 [    1    0    0    0    0    0    0    0    0    1]
 [    1    0    0    0    0    0    0    0    0    1]
```

38/nbconvert/html/Google%20Drive/GitHub/EvacuationModSim/CX4230_Evacuation_ModSim.ipynb?download=fals

n_ModSim

```
 [    1    0    0    0    0    0    0    0    0    1]
 [    1    0    0    0    0    0    0    0    0    1]
 [    1    0    0    0    0    0    0    0    0    1]
 [    1    0    0    0    0    0    0    0    0    1]
 [    1    0    0    0    0    0    0    0    0    1]
 [    1    0    0    0    0    0    0    0    0    1]
 [    1    1    1    1    1    1    1    1    1    1]]
Done with simulation
```

Remaining Work

We currently have the following tasks slated to be completed before the completion of our project, however, if these tasks do not take as long as anticipated we will continue to work to add more features into our model to ensure the best outputs possible in the timeline given

1. Implement a hazard (Rohan)
    a. Fire can expand at random rates (stochastic).
2. Different rates of people (Dheeraj)
    a. Define each type of person. Each person has a different average walking speed and standard deviation. Subclasses for people each with different statistics all based on the parent class of person which has already been implemented . Will factor into behavior for collisions.
3. Randomized map generator (Rishab)
4. Best optimal room evaluation (evaluation score) (Rohan, Dheeraj, Rishab)
    a. Will require an extensive set of models and will output the best rooms for evacuation processes by using validation metrics like the amount of people escaping the hazard

References

https://www.vttresearch.com/sites/default/files/pdf/tiedotteet/2010/T2562.pdf
https://www.sciencedirect.com/science/article/pii/S1877042814041287?via%3Dihub
https://www.sciencedirect.com/science/article/pii/S0957417413008270?casa_token=qgPT7dpDak0AAAAA:HQjIsgK3yXwxwEIXDS-0j62cL8kx2kzjmkZ3wba8-sy5TgPsx4Es3h8esYyFv2jJnf6lxTI