Project 1 Final Report

CSE6730 2016Spring (Group 12)

Jianli Chen, Yuzhi Guo, Kai Lu, Xin Cao

1. Project Description

The project is intended for simulation of pedestrian egress of Georgia Tech Bobby Dodd Stadium after a football game. An efficient evacuation plan should be come up with the simulation to optimize the evacuation time. A stochastic cellular automata model would be developed incorporating the individual behavior of pedestrians. In the scope of this project, since it is too time and memory consuming to model the whole campus with all possible destinations, the geographic range of modeling and simulation is defined as the red rectangular shown **Figure 1** below.

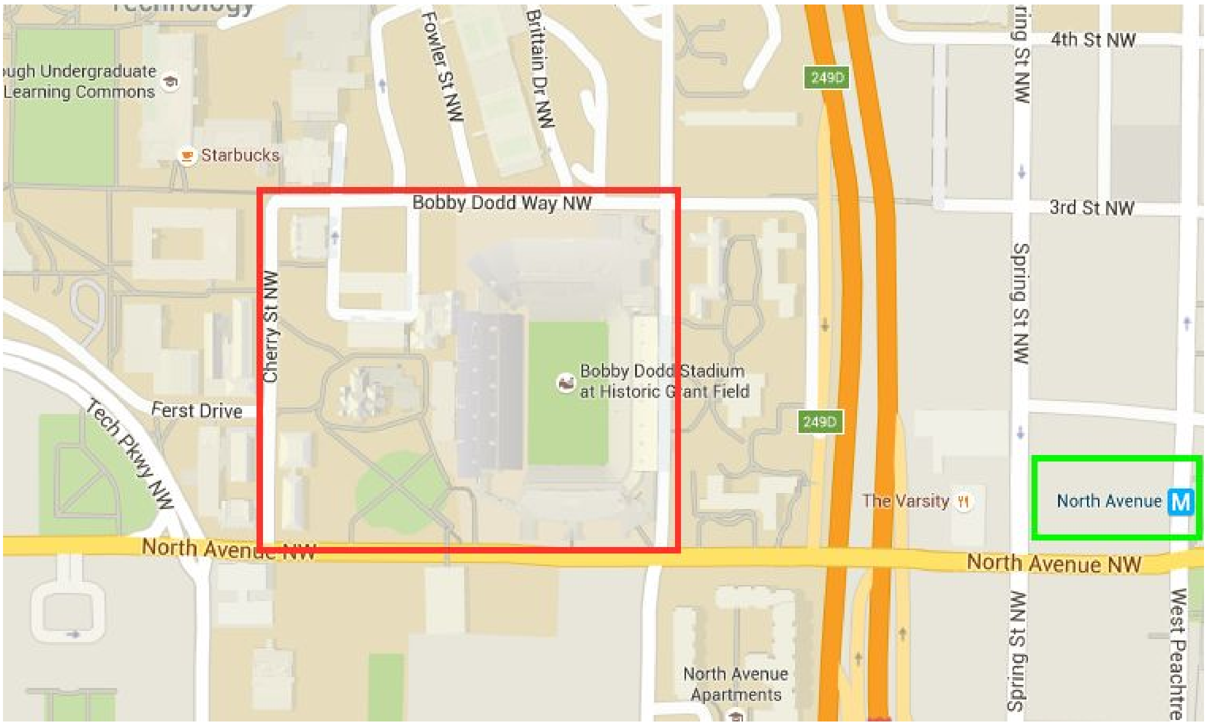


Figure 1. Geographic range of modeling and simulation

Several exits related to people's final destinations will be assigned on specific locations along the red circle, these exit points are refered as notional destination. Once people from the stadium have reached the notional destination, that people instance will be removed from the modeled area at the next step.

The final destinations could be divided into three categories: North Avenue MARTA Station, Vistor paking lots around the campus and student residence halls.

North Avenue MARTA Station is shown in the picture above with a green box.

Gameday vistor parking lots are highlighted in the **Figure 2** below by green boxes.

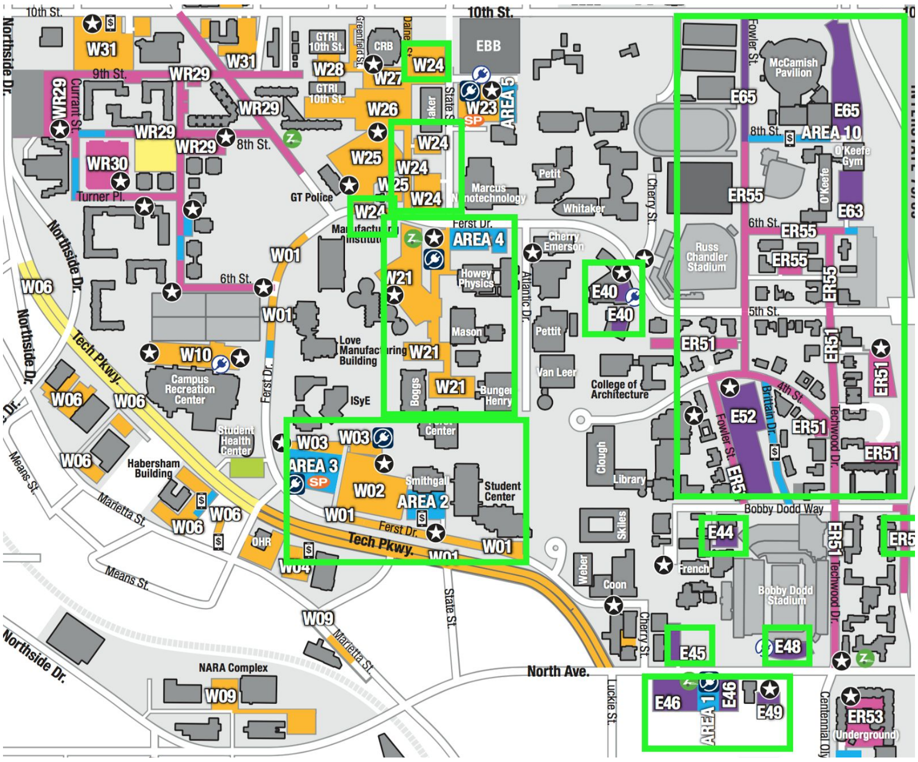


Figure 2. Map of campus parking

Students residence halls are marked as yellow blocks in **Figure 3** below.

Figure 3. Campus map of Georgia Tech

The stadium has 10 exits as shown in **Figure 4**:

Figure 4. Map of Georgia Tech Bobby Dodd Stadium

2. Literature Review

2.1. Pedestrian behavior model

As for how to model the pedestrian behaviors and dynamics, we found that two models are the most popular based on our research. The first is social force model. Based on [1], the first force needed to be defined is the acceralation. The pedestrians are always assumed to go on the fastest road to his/her destination. Secondly, to model the interaction between pedestrians, each pedestrian is assigned an ellipse based on his speed and the density of floor. That ellipse represents the occupied space, which has a repulsive force to other pedestrians. Thirdly, attractive force springs from other persons (friends, street artists, etc.) or objects (e.g. , window displays) is defined as well. Lastly, a fluctuation term is imported to model the random behavior of pedestrians. Later, this model is further expanded, such as in [2], special characteristics (e.g group evasion) of pedestrian crossing behavior is further taken into account to better model condition of pedestrian in signalized crosswalk. In addition to the social force model, the other classic model for modeling pedestrian dynamics is the floor field model, which we think is very suitable to use with cellular automation. In [3], it says that the transition probability of a pedestrian is composed of three parts -- the pedestrian's preference, static floor field and also the dynamic floor field. The pedestrian's preference is determined by the pedestrian's relative position to his destination. The static field represents the attractive spots for pedestrians, such as the exits. Lastly, the dynamic floor is used to model the long and short distance interaction between pedestrians. It's evolving with time. Similar to social force model, the floor field model is also extended and updated later to simulate the evacuation process [4] and also movement of pedestrians in 3D dimension [5]. The transitional matrix including static and dynamic floor model is improved in order to match the reality better without explicit matrix of preference [6].

2.2. Flow distribution of crowd

The previous study about the collective crowd behavior reveals the dynamics of the pedestrains has a chracterastics that entering the scene of pedestrains can be modeled as poisson distribution[6]. Many event data showed that the poisson distributioin is a good approximation for the collective crowd behavior. The typical scene of the applications includes the airport, the train stations, the stadium egress or the event evacuation. Therefore, introducing poisson distribution into our model is significant to approach the real situations of those events. In other words, a pedestrain emerging into each exit of the stadium follows a poisson process in time, the potential distribution of which is:

$$P(k,\lambda) = \lambda^k e^{-\lambda}/k!$$

where k is the number of people which go through the exit of the stadium towards their desired destinations, and λ is the expected number or average amount of the people going through the exit of the stadium during a certain time period (or unit time as we set in the model). queuing theory usually satisfy the poisson distribution for the behavior pattern [7], and the average effect during a time period is meaurable and significant in macroscopic time scale, the poisson can bring a congestion to form queues if we concern the capacity of a queue is finite, because it is possible that there is a shape increase of the poeple arriving at one stadium exit during a short time, even through such a possibility is usually not high, but the frequency of such an event is repeatable during the process of evacuation. The arriving time of the people arriving at the exit of the stadium follows the corresponding poisson process:

$$P[N(t) = k] = (\lambda t)^k e^{-\lambda t}/k!$$

The egress system is associated to and affected by the ingress system which has a pattern of poisson distribution. Therefore, the ingress will directly decide the sequential crowd distribution in the whole area. we estimate the average number of the people arriving at each exit of the stadium every unit time (0.3 second in the model) by an ideal assumption that the stadium could be emptied out with 10 minutes if there is no congestion situation occuring in each exit of the stadium. The average number of the people arriving at each exit of the stadium is about 17 per second (5 person per second). This number is in the range of many data which has been validated in real events.

3. Conceptual Model

For this project, we are using cellular automata (referred as CA model) as the prototype for conceptual model. In CA model, our grid cell size would be 40x40 cm^2, square shaped, with the capacity of one single people in each cell. The analyzed square area is approximately 295x335 meter^2, which is scaled basicly using Google Map distance calculator API, with some in field measurements for some details.

The visualized map of CA model is shown below:

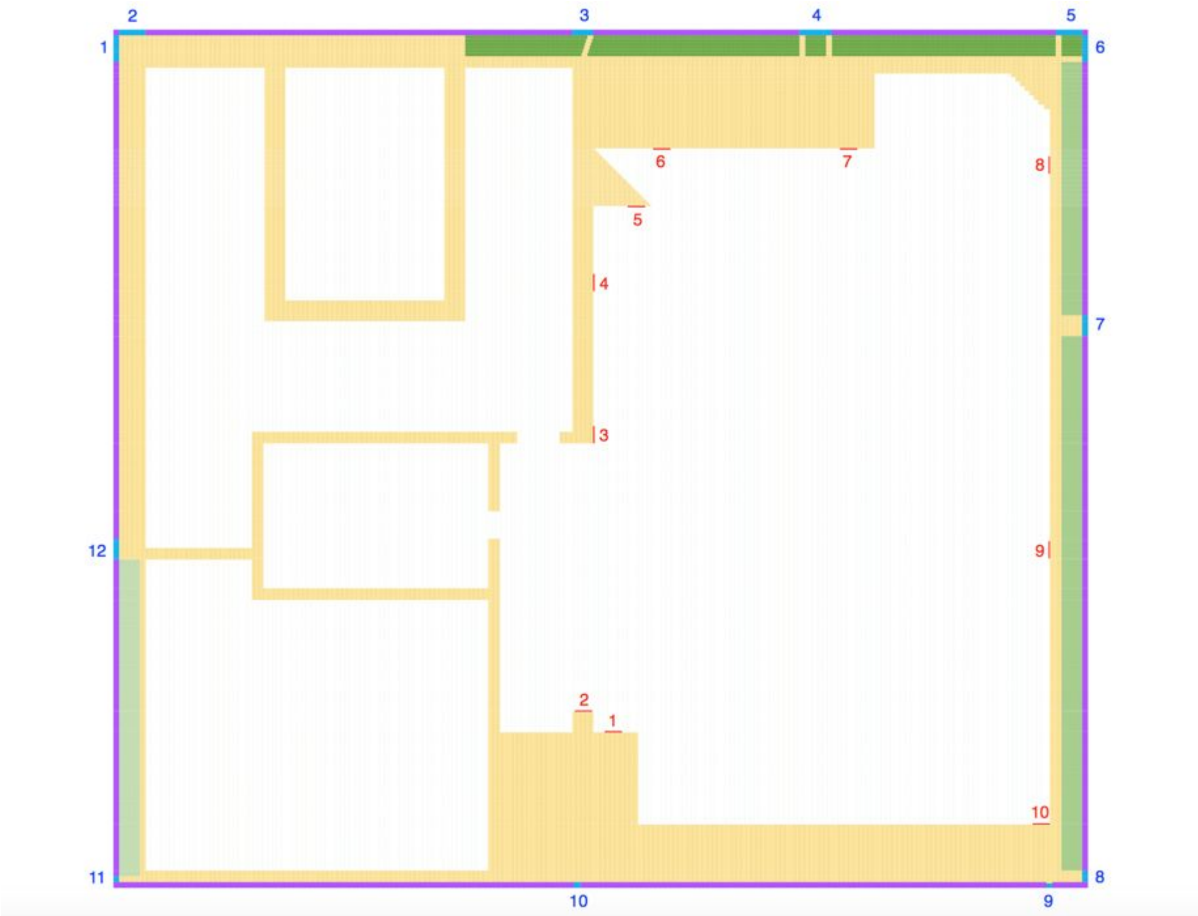


Figure 5. Simplified geographic simulation model

Note:

- ➔ White area represents non-walking zone, i.e. buildings, trees, etc.
- ➔ Yelloe area represents walking zone.
- ➔ Green area represents roadway that could either stay open, i.e. non-walking, or be closed off, i.e. become walkway to accelerate egression process.
- ➔ Purple lines are the boundaries, approaching which represent successfully evacuate.
- ➔ Red short lines, with red numbering, represent the 10 stadium exits.
- ➔ Blue short lines, with blue numbering, represent the 12 notional destinations.
- ➔ Orange numbers denote the two traffic signals within the analyzed area.

With the goal of simulating people leaving Bobby Dodd Stadium after a football game, and of minimizing the total egression time, we adjust our conceptual model by changing the state, i.e. stay open or close off, of three surrounding street: Cherry Street NW, Bobby Dodd Way NW, and Techwood Dr NW. And North Avenue NW is considered main road, and will always stay open, such that pedestrian could only cross the street using crosswalk with obeying to the traffic signal, or using the foot bridge located between Cherry St and Techwood Dr.

Inputs and outputs, entities and attributes, events and activities, assumptions and simplifications will be discussed in the following contents.

3.1. Input

3.1.1. Map contents

Within the analyzed area described above, a simplified geographic map is measured and modeled, showing pedestrian's walking areas, non-walking areas, street's area (open or closed), traffic signals, stadium exits and destinations around Georgia Tech Bobby Dodd Stadium. The visualized map is presented above.

3.1.2. Flow distribution.

A distribution of the flow over time at stadium exits after the end of football game would be considered

3.1.3. Destination assignment

Each pedestrian is stochastically assigned a true destination, and then randomly associated with a possible notional destination.

3.1.4. Evacuation plan

Change the open/close state of the three surrounding street to simulate evacuation process under different scenarios.

3.1.5. Signal state

Traffic signal state would change alternately. Signal interval measured

3.2. Output

3.2.1. The evacuation time

The most important output would be the evacuation time, which denote the egression efficiency. We ran the experiments several times, obtained each evacuation time, which would be presented in detail in the experiment result analysis part. Further more, the evacuation time is compared at diferent scenarios, i.e. the streets' open/close state. With different scenarios comparison, the effect of closing surrounding strees on the evacuation time could be easily analyzed. The detailed analysis about this would also be in the analysis part.

3.2.2. The visualized dynmaic evacuation process

Our experiment would output a visualized dynamic evacuation process using python GUI. With the visualized output, the whole egression process could be better understood. The verification process could also be visually analyzed.

3.2.3. Congestion analysis

If pedestrians stay in the same cell for a relatively long time, or wandering aroung the same area, a congestion is considered happening. At the end of the experiment, the congestion evaluation, including the location congestion happen, how many congestions take place, and how long the congestion last, would be output for analysis purpose.

3.3. Entity {Attributes}

3.3.1. Pedestrian{
ID(No.),
Coordinates,
Speed(associated with the cell density),
Destination,
Preferred exit(associated with destination),
Size of Group,
preference matrix
}

3.3.2. Cell{
Coordinate,
State(<0 when no walking is permitted, =1 when occupied, =0 when available),
density
static field value (the number of static field is same with the number of possible exit)
dynamic field value(increased with people step on)
}

3.3.3. Stadium Exit{
Coordinates,
flow distribution
}

3.3.4. Pedestrian Signal{
Coordinates(on a crosswalk where vehicles are permitted),
state(stop/pass),
ControlCell
}

3.4. Event and Activity

3.4.1. Pedestrian:
- Walk(either direction or stop)
- Exit(out of the borders)

3.4.2. Stadium Exit:
- Stadium discharge flow

3.4.3. Crosswalk Signal:
- State change

3.5. Assumptions and Simplifications

3.5.1. Assume that the expected number of egress people from the stadium is 30 per unit time step (0.3 seconds) for 10 exits of the stadium.

3.5.2. People’s movement inside the stadium is not modelled. Assume each stadium exit has the same flow distribution over time when there is no conjection at the exit.

3.5.3. Assume the group size is alway one

3.5.4. Assume all entities have two levels of walking speed, when the density of pedestrians are lower than a threshold (details will be described later in 4.2.2), the pedestrian tends to walk as twice as faster than the condition with a high pedestrian density.

3.5.5. Once a congestion take place at the stadium exits, assume people who are inside the stadium about to come out, are temporarily paused from evacuating process.

3.5.6. People are randomly assigned a final destination (exit) with equal probability (the possible destinations are limited to Marta station, several on campus dormitories and several campus parking lots).

3.5.7. Assume people always obey crosswalk signals; Once traffic signal turns red, pedestrains who on the crossing area will reach the other side immediately, and the crossing will be cleared.

3.5.8. Assume that the transitional probability of each possible cell (the matrix of preference) is combined with the static floor field and the dynamic floor field. The implicit assumption is the choice of pedestrain depends both on the distance from the exist and on the social influence by the surrounding population.

3.5.9. All the stadium exits are modeled as vertical or horizontal lines.

3.5.10. All pedestrians are familiar with the fastest route to their own destinations.

4. Simulation Model and Program

4.1. Simulation Working Flow chart

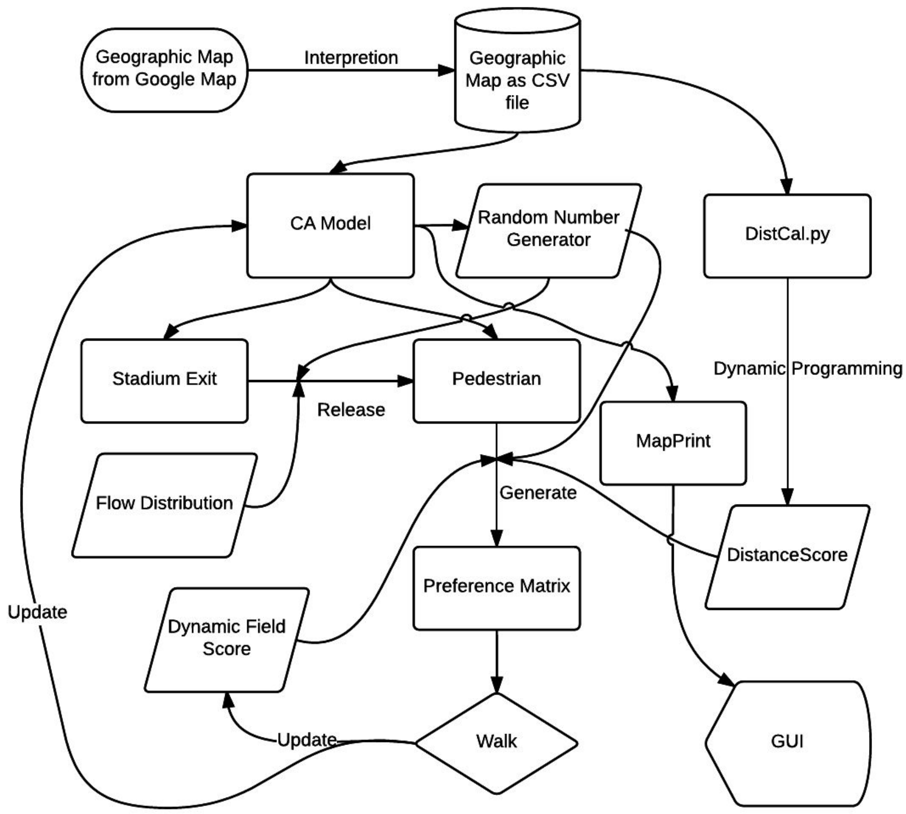


Figure 6. Working flow chart of simulation program

The picture above shows the simulation working flow of our program.

First of all, we refer to the geographic information of the simulation range around the football stadium, which is from Google Map. Based on real geographic distances, objects and coordinates, we build a simplified map in form of a CSV file, in which the walking-zone, no-walking-zone and roads(could be either close or open for traffic) are all marked with specific number. Other important traffic information such as destinations, stadium exits and crosswalk signals are embeded in the code.

The next important action we take is the calculation of the Distance Score, which is known as an attributes called “Static field score” of entity Cell in our conceptural model shown above. The Distance Score actually indicates the distance from a cell to a specific destination. The calculation algorithm we employed in the code of DistCal.py is dynamic programming(DP) with a Breadth-First-Search approach.

Then, we could build our CA model with the geographic data we have got. Two critical entities are constructed as two classes called StadiumExit and Pedestrian. The instances of StadiumExit have a method called “egress” for generating instances of Pedestrians. The number of people released to the field yield our flow distribution function. The destinations of pedestrians are randomly assigned.

The behavior of pedestrians is based on the Perference Matrix, which is a linear combination of the Distance Score we have and the Dynamic Field Score indicating the popularity of a cell. In addition, we also add some stochasticity to the behaviors by introducing some random numbers. Since we use a priority queue to proceed the walking simulation, we assume that the score for the occupied cells would always be the lowest.

With the Perference Matrix calculated, we could decide where to go for each of the pedestrians. And with new location of each pedestrain updated, the CA model state would be refreshed and so as the dynamic field score of each cell.

At the end of each timestep, the program would send the state of CA model to MapPrint.py and print the map to our GUI.

4.2. Preference Matrix Calculation Using Static and Dynamic Field

Calculation of preference matrix is the most critical component in our simulation since it directly determine the behaviors of pedestrians. Overall, as shown in **Figure 7** and **Figure 8** below, our preference matrix could be either 5 by 5 (when the speed is 2) or 3 by 3 (when the moving speed is 1) based on the density of its neighborhood cells. The basic formula for calculating the preference matrix is shown in **Figure 9** below. In the formula, the Si,j is the static field score that measures the distance decrease if the pedestrian moves from its standing point to i,j cell. Si,j is only positive when the pedestrian move to the cell that is closer to his destination (exit) compared to his standing cell. For the other cases, this Si,j is set to 0 thus the preference matrix is purely determined by random number. As for the dynamic field score, it's for modeling the interaction between pedestrians -- the pedestrians later tend to follow the former pedestrians' movement if they intend to move in the same direction (e.g the same exit or destination). Dynamic field score will increase by one if it's occupied at that time step. And this score decays with a factor of 0.5 at each time step. ni,j in the formula represents the state of the target cell, which is 1 when occupied and 0 when unoccupied. Finally, for the random number, when the static field score is larger than 0, the simulation program will generate a random number from 1 to 2 to ensure that if the pedestrian could walk a cell that is closer to the exit, he will move there. In all the other cases (move to the cell that has distance to exits equal to standing point or larger than standing point), a random number from 0 to 1 will be generated so that the pedestrian can move randomly.

$M_{1,1}$	$M_{1,2}$	$M_{1,3}$	$M_{1,4}$	$M_{1,5}$
$M_{2,1}$	$M_{2,2}$	$M_{2,3}$	$M_{2,4}$	$M_{2,5}$
$M_{3,1}$	$M_{3,2}$	$M_{3,3}$	$M_{3,4}$	$M_{3,5}$
$M_{4,1}$	$M_{4,2}$	$M_{4,3}$	$M_{4,4}$	$M_{1,1}$
$M_{5,1}$	$M_{5,2}$	$M_{5,3}$	$M_{5,4}$	$M_{1,1}$

Figure 7. 5 by 5 Preference Matrix

$M_{1,1}$	$M_{1,2}$	$M_{1,3}$
$M_{2,1}$	$M_{2,2}$	$M_{2,3}$
$M_{3,1}$	$M_{3,2}$	$M_{3,3}$

Figure 8. 3 by 3 Preference Matrix

$$M_{i,j} = S_{i,j} * D_{i,j} * \left(1 - n_{i,j}\right) + \ random\ number$$

Figure 9. Formula for Calculating Transition Probability Pi,j in Preference Matrix

$$S_{i,j} = \begin{cases} Decrease\ of\ distance & ,if\ cell\ i,j\ is\ closer\ to\ exit\ compared\ to\ standing\ point \\ 0, & otherwise \end{cases}$$

Figure 10. Si,j Function

4.3. Outstanding features

4.3.1 Dynamic programming for distance calculation

Distance calculation is one of the most critical aspect in the simulation program because it has a large impact on the movement direction of pedestrian and efficiency of the whole simulation program. In our simulation program, we utilize dynamic programming to precompute the distance from one cell to all exits during the initialization phase. For each exit, starting from the cells that have distance to that exit as 1, we use breadth first search (BFS) to find all the neighborhood cells and set their distance to the exit as 2. Recursive call is used to calculate the distance to that exit for all the cells in the map. The distances of one cell to all possible exits are stored after initialization. Later, when calculating the preference matrix for each pedetrian, the distance could be easily retrieved for his surrounding cells, which improves the efficiency of our simulation program a lot. An example of distance calculation result is shown in **Figure 11** below. The darker the color, the further the distance on the map to exit, which is at the top of the map.

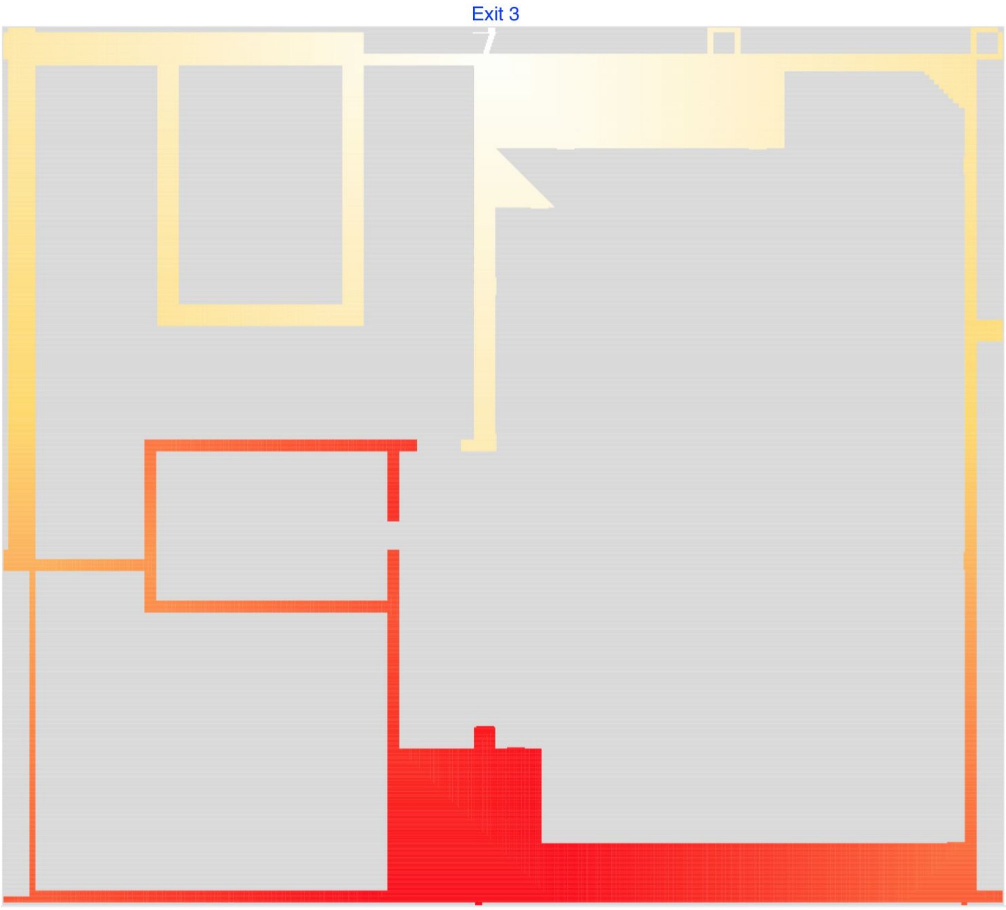


Figure 11. A visualized map of Si,j for exit #3(darker color means larger distance)

4.3.2 Variable Speed based on real time density

To model the real scenario of pedestrian evacuation after football game, the pedestrian is set to be able to adjust his speed according to his surrounding environment. when the density of pedestrian around him is high (more than half of his neighborhood space (5 by 5 cells around) are occupied), the pedestrian has speed of 1. While the density of pedestrian is low (more than half of his neighborhood is unoccupied), the peredstrian can walk in the speed of 2, which means that the pedestrian could directly move 2 cells in that timestep.

4.3.3 Randomization to model real pedestrian behavior

In our calculation of preference matrix, we add some randomization process to make the pedestrians in our simulation program behave similarly to the pedestrians in real world. Since our cells are small (40cm by 40cm) and the street could become very crowded in the real world evacuation after football game, the pedestrian walking could hardly follow any strict patterns (e.g pedestrian always go straight etc). Thus, some random numbers are added to the preference matrix after calculation (details have been provided at 4.2 in this report).

4.3.4 Accurate And Flexible Map Establishment

An accurate map is significant for making our simulation be realistic. Our simulation map is established based on distance measurement from Google map and on site investigation. The map is established in a CSV file first and then imported into our simulation program so that it could be easily changed for the test of different scenarios.

4.3.5 Pedestrian Signal Modeling

The pedestrian signals are modeled near the exits on the North Avenue. Since the pedestrian signal is a very possible source for aggravating congestion, its modeling could be important to study the congestion phenomenon that happens during evacuation of a football game.

4.4 Program Architecture & Interfaces:

File and class hierarchy is shown in the **Figure 12** below:

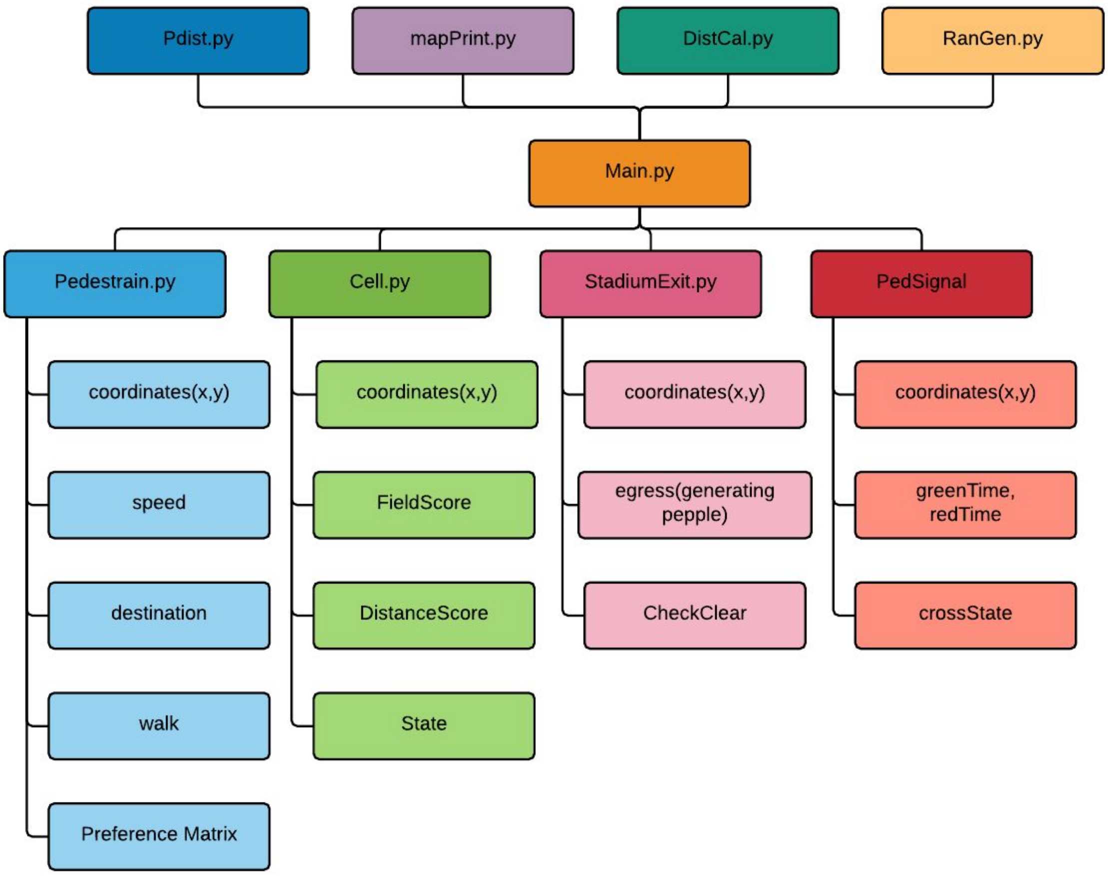


Figure 12. File and class hierarchy

—Pedestrain.py includes the definition of class Pedestrian, which has attributes and functions including: coordinates, speed, destination, groupsize, destination,preference matrix and walk

—Cell.py includes the definition of class Cell which has attributes and functions include: State, score, updateState, updateScore.

—StadiumExit.py includes the definition of class StadiumExit, which has attributes and functions including: coordinates, egress and checkClear

—PedSignal.py includes the definition of class PedSignal, which has attributes and functions include: coordinates, greenTime/redTime period, crossState.

—MapPrint.py includes the function MapPrint for print the simulation results to GUI

—drawWin.py includes the definition of class drawWin for the creation of GUI

—Pdist.py includes the definition of class Pdist for the flow distribution generator

—DistCal.py includes the function DistCal for calculating the static field score with Dynamic Programming

—RanGenNew.py, which generates random numbers for StadiumExit and Pedestrians

—findDistance.py, which is used by DistCal for calculating the minimum distance

Graphic User Interface(GUI)

The GUI Interface we create for displaying the evacuation process is show in **Figure 13** below.

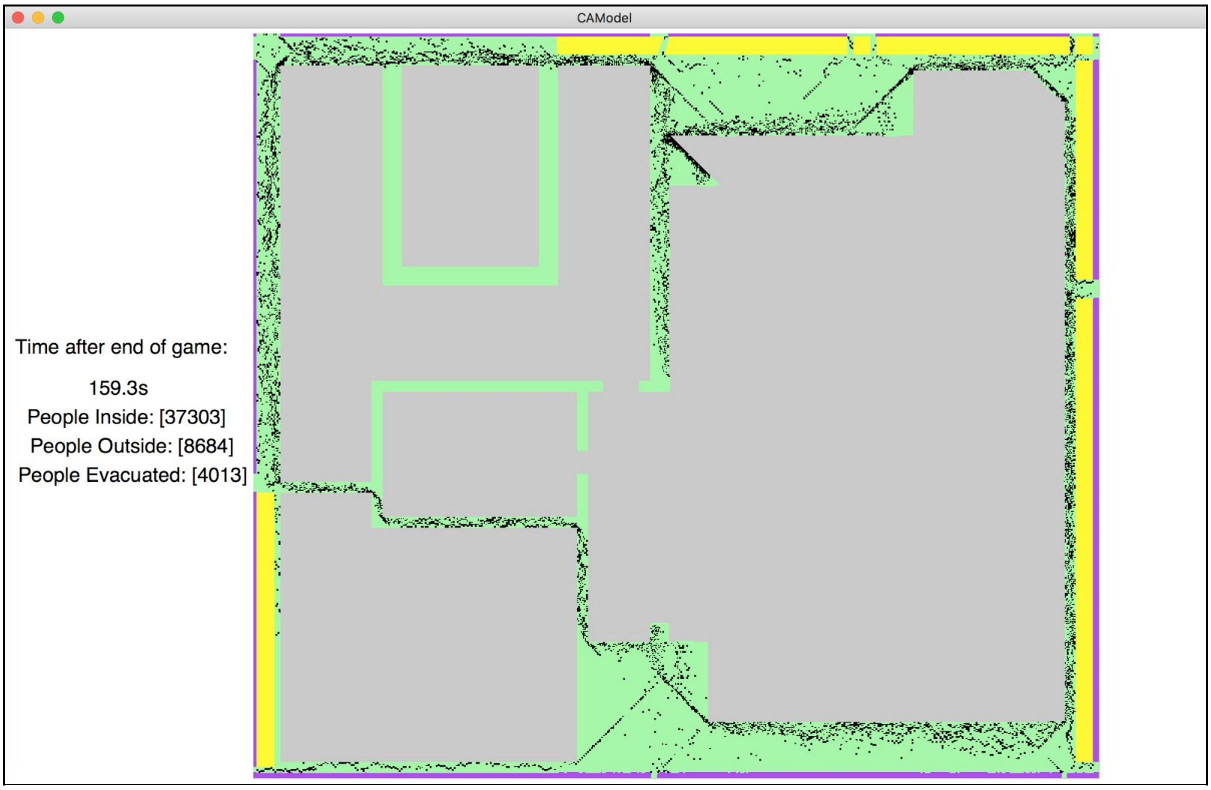


Figure 13. Sample GUI

On the interface, the most eye-catching part is a real-time updated image of the simulated stadium neighborhood. Every pedestrian on the field would be marked as a 1X1 pixel black dot, so that their behavior could be easily monitored. Other information of the event shown here includes the time passed after the end of football game and the numbers of people inside, outside and evacuated.

5. Verification and validation

5.1. Random number generator verification

We have develpped a random generator which has a probability of uniform distribution. The corresponding code file is RanGenNew.py. We used lehmar algorithm, the details of which has been discussed in the lectures. The parameter we used is: a=16807, c=0, m=2147483647, which is recommended by Parker and Miller in 1988 [9].

$$N_{k+1} = (a\,N_k + c)\ mod\ m$$

we also introduced the anti-overflow mechanism, which we turn the iterated number which approaches the upper limit of the long int type back to the intial seed of the long array, and then make the number keeping iterating in the new loop. By validation, the period of this array is very long, at least longer than tens of millions' iterations. We did not touch to the end of the iteration loop since it will take a long time to iterate the calculation, Another pretective mechanism in the model is set a lower limit to avoid from underflow to zero or less, in which we just simply set the number back to the initial seed number if any iteration results in a negative number, even if it is impossible from the iteration equation. We also used a strategy to use the system time whose accuracy is microsecond inorder to get a seed stochastically. We used the random number generator to generate 10000 random number, and do a statistics about the frequency of the numbers generation in different range in order to test and validate the generator can get numbers in the uniform distribution.

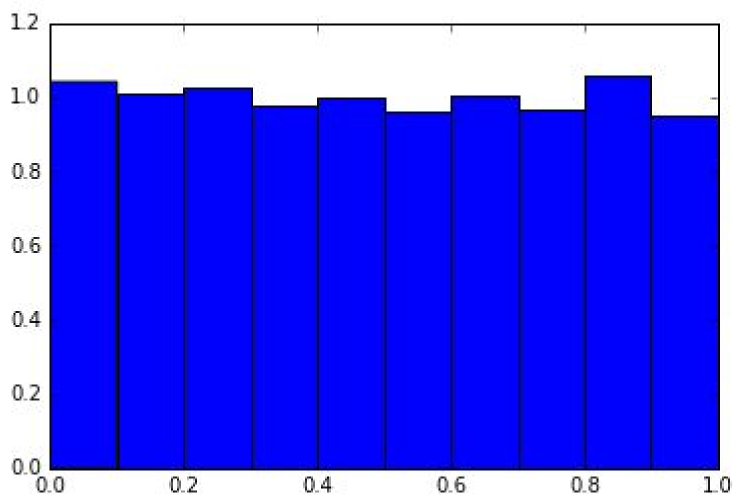


Figure 14. The histogram of the normalized frequency of random numbers in each bins vs the range of bins

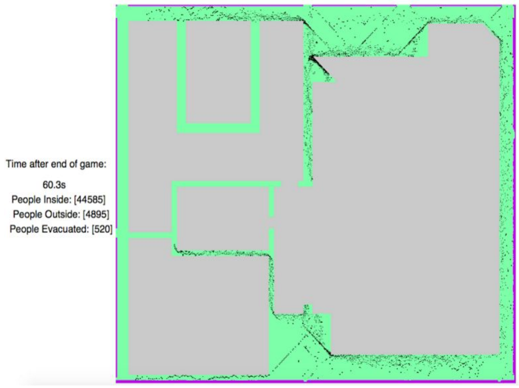
This histogram shows that the numbers generated by the random generator almost follow the uniform distribution during 0~1, and the y axis the normalized frequency of different random numbers whose value falls to one of the bins shown in the x axis. Each bins has about 1000 samples and the total number of bins are 10. From the histogram plot, we have validated that the random generator can generate numbers with a uniform distribution. The random number generator is applied to two process, the first is to assign one destanation randomly to the people who arrives to any exit of the stadium. the second senario is when we calculate the scores of the preference matrix, we introduced random numbers to weighted scores of the preference matrix.

5.2. Poisson distribution validation

As we discussed before, the entering of crowd into the event area follows the poisson distribution. We validated the arrival time interval of the pedestrians and found that it appoximately follows a poission process, which corresponds with the underlying poisson distribution of the pedestrian outflow through the exits of the stadium as we assumed. And this is a key point of a poission distribution. The distribution of pedestrian outflow directly decides the initialization of the entering events and the dynamics of the crowd. A good distribution assumption can lead the simulation close to the reality while a bad distibution assumption could lead the simulation deviate from the reality and in even worse cases could result in some unreal conjestions. We tested our code a couple of times with parametered poission distribution and observed the results, combind with the other assumptions of our model, we found the total egress time is comparable with the real situations and would result in a real randomicity locally. Based on the papers we read about the evacuation of a event at a relative limited area such as the airport, the train station or other public fields, the collected data reveals a very good consistance with a poisson distribution, which reveals that the poisson distribution is a very general characteristics of the crowd dynamics.

5.3. CA model and Preference Matrix Verification

For verification of our simulation program, visualization method is used. More specifically, we take screenshots of our simulation result approximately every 60s to monitor the evacuation process and pedestrian walking process to see whether it reflects our thoughts in the conceptual model properly. The screenshots in the graphs below shows as an example. It’s our simulation under the condition that roads are all closed for pedestrian evacuations (the vehicles are not allowed to enter).

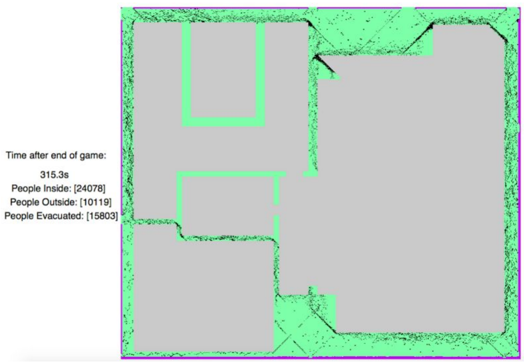


60s screenshots

120s Screenshot

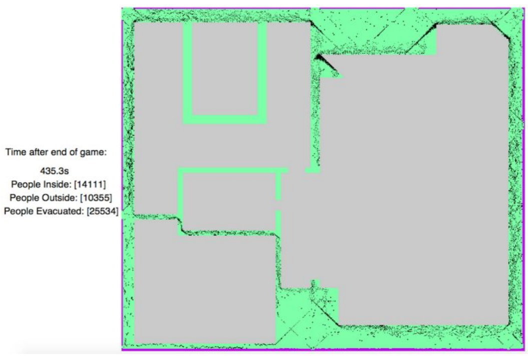
180s screenshots

255s screenshots

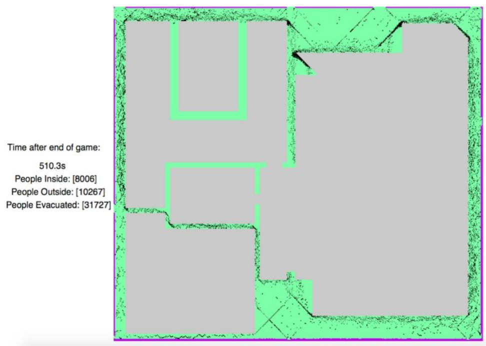


315s Screeshot

390s Screenshots



435s Screenshots



510s Screenshots

600s Screenshots

From the graphs above, we could see that at the beginning, the people who is walking on the road increase very fast. However, as the time passes by, the number of walking people has become a relatively stable number. It is expected since after all the roads are closed for pedestrian walking, we have already got enough capacity to evacuate all the pedestrians at the same time (we will not have congestions during simulation). Also because the map we model is relatively small, both the rate of pedestrian egress from stadium exit and the rate of pedestrian evacuation out of map could become stable after a period of time. Thus, through the observations of pedestrian walking for a continuous period of time, it's concluded that our model match our conceptual model and our expectation very well.

5.3 Simulation Validation

The method we used for validating our model is to see whether our simulation result will match the realistic evacuation process. For the first place, as shown in **Figure 15** below, we can see obviously that the pedestrians disperse quite well on all the walking zones (expecially roads). Since we could have bi-direction flow on the roads in our simulation model and in realistic evaculation process the pedestrian walking on the roads will definitely avoid each other and find vacant space to walk, our simulation model reflect the real case scenario very well. The only problem is that in the large square on the top and bottom of map, the pedestrian tend to follow some patterns since they always find the fastest route to his destination. But it's still reasonable in our simulation since it's obvious that our square is large enough to accomodate all the pedestrians egressing from exits nearby, thus the pedestrians actually have the space to walk on his fastest route to the destination they want. Secondly, based on our simulation result for different scenarios presented in the section below, it shows that as we close more and more roads for vehicles so that pedestrian can have larger space to walk, the evacuation time will decrease. Finally, the box plot (**Figure 16**) from 10 times of our simulation shows that our mean and median total evacuation time is about 874. Although it seems a little bit small, but considering that our map scope, it's still an acceptable and reasonable result.

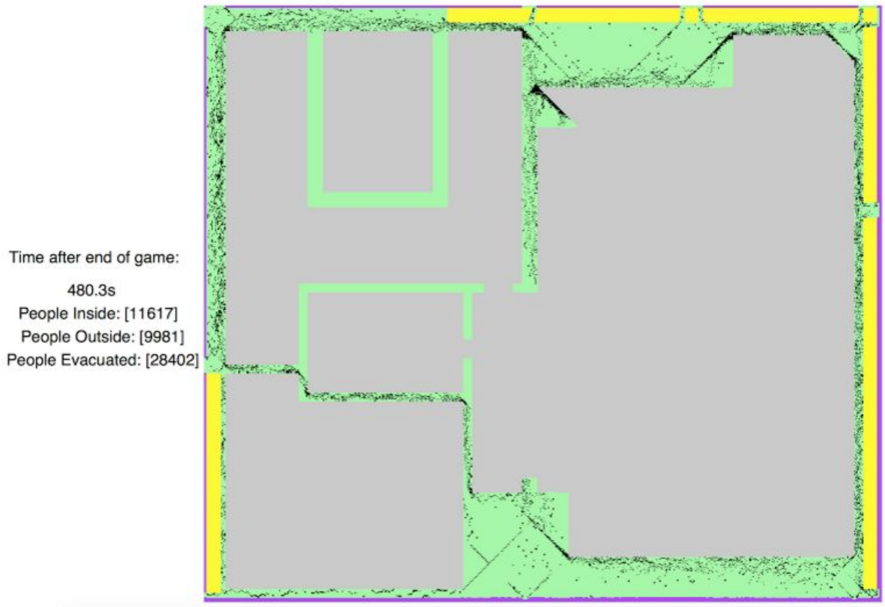


Figure 15. Disperse of Pedestrian In Our Simulation Process

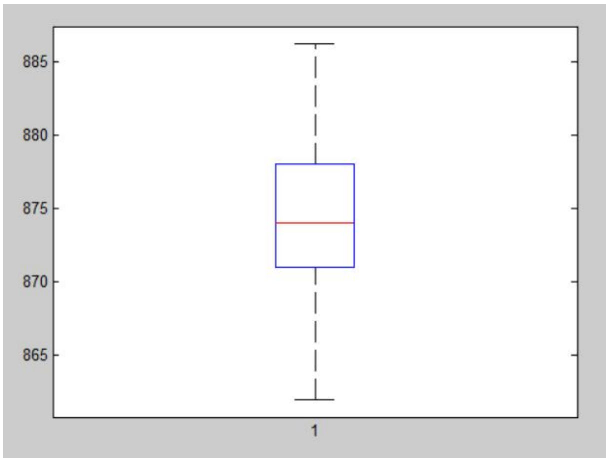


Figure 16. Box Plot for total evacuation time for scenario 1

6. Experimentation result analysis

6.1. Conjestion analysis

To run the congestion analysis, we increase the discharge rate of each stadium exit to a very large number (expected 10 persons per 0.3s) to see the most probable place that could have congestion. From the **Figure 17** below, it’s easy to see that we have 4 places that is most probable to have the congestions. However, except for the right corner, we can’t do anything to the other three places since the roads are already all at their largest capacity to evacuate pedestrians. Thus, during the scenario tests, we close down more roads (vehicle is not allowed) especially on the right and upper part of the map to see whether we could have better evacuation performance.

Figure 17. Congestion Analysis Screenshot

6.2. Scenarios Tests

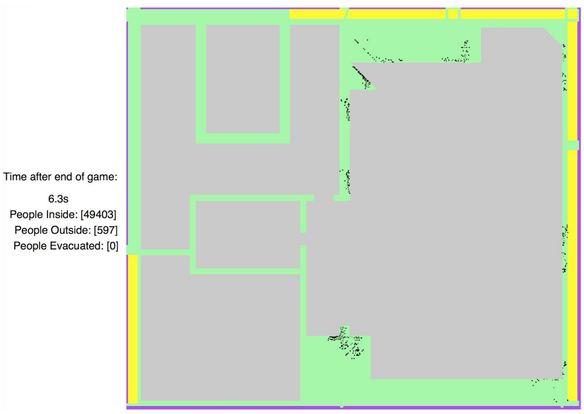
We employed three scenarios as follows:

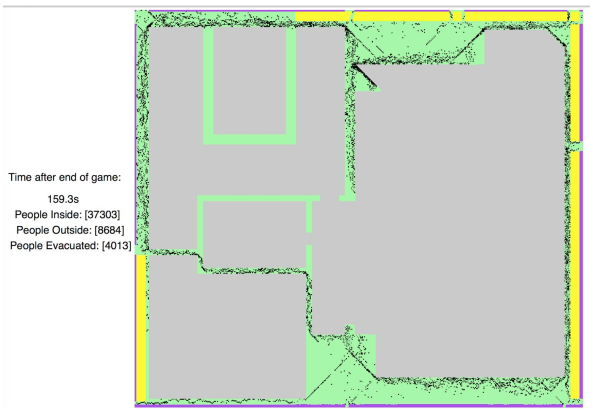
- All roads stay open, i.e. pedestrian only use side walk.
- All roads closed off for pedestrian to evacuate.
- Bobby Dodd Way NW closed off for pedestrian to evacuate.

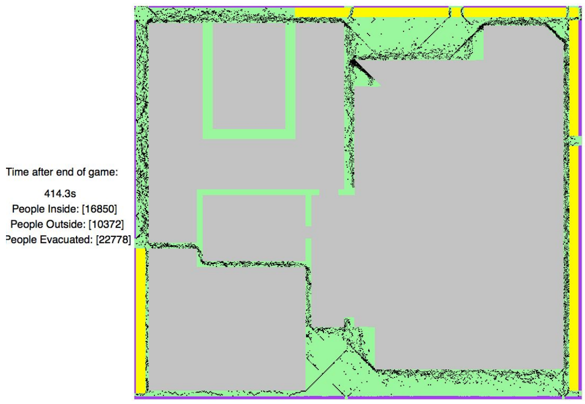
Each scenario would be analyzed in detail in the following contents:

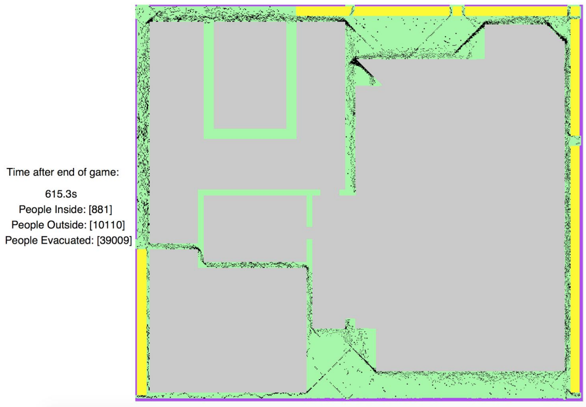
6.2.1. Scenario 1- All road stay open, i.e. pedestrian only use side walk

During the visulized simulation process, we produced some screen shots:

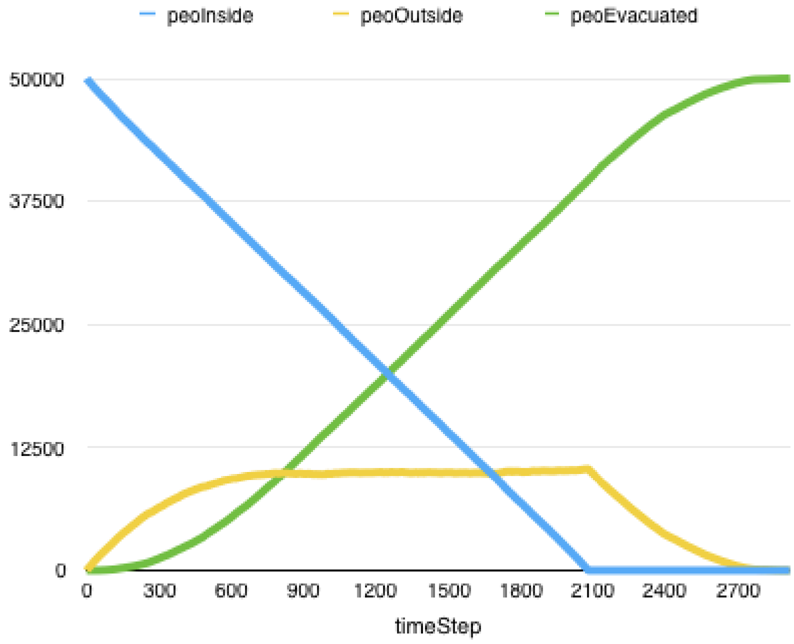








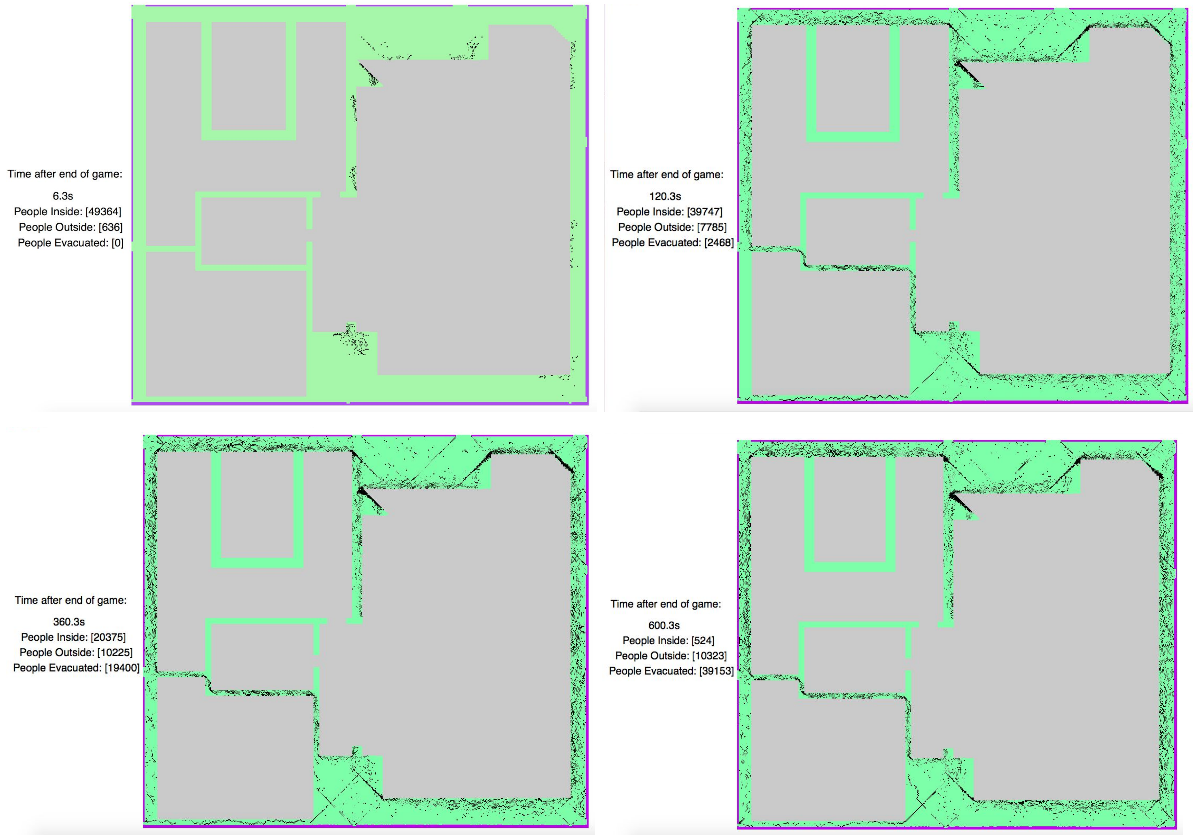
The following diagram shows the relationship between the number of people inside the stadium, the number of people evacuated outside the stadium but still remain in the system under investigation, the number of people successfully egressed, and the timeStep.



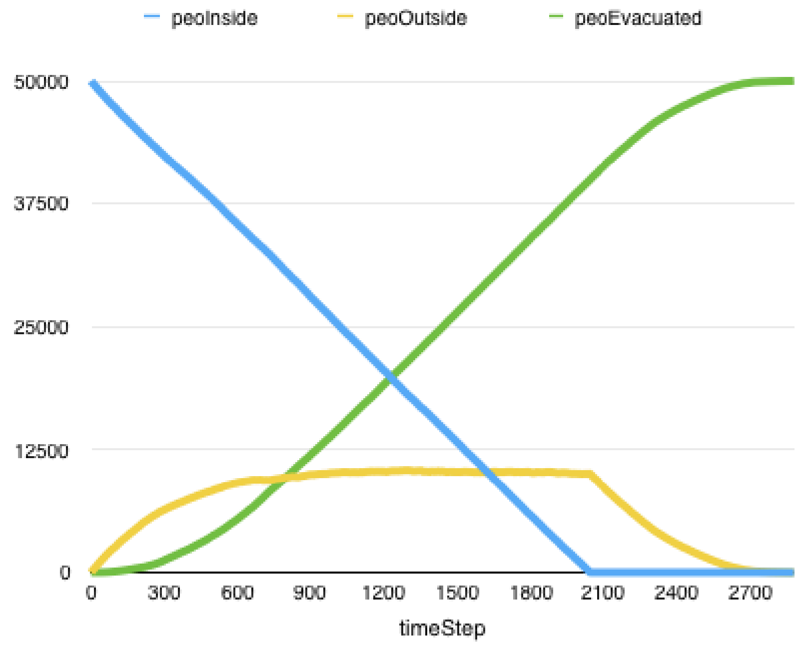
Total egression time: 874.2s
Stadium egression time: 623.7s

6.2.2. Scenario 2- All roads closed off for pedestrian to evacuate

During the visulized simulation process, we produced some screen shots:



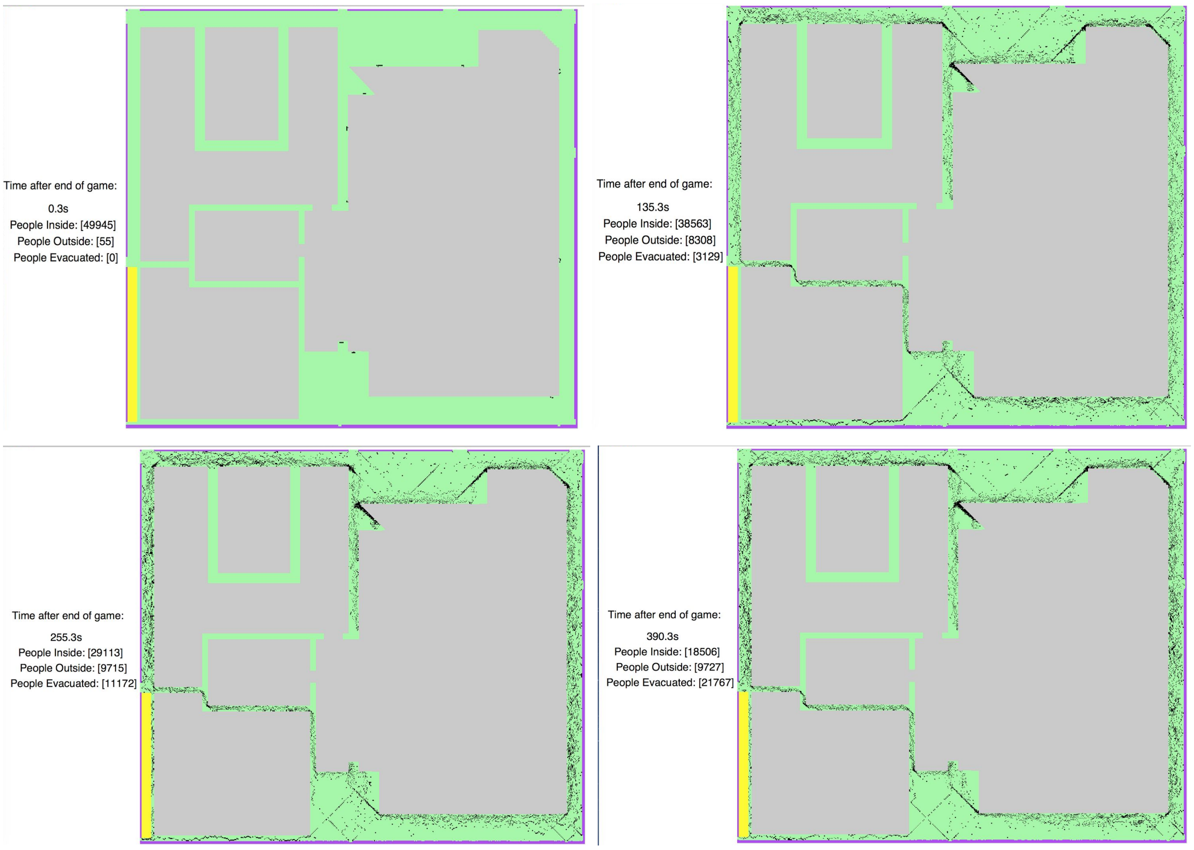
The egression number of people vs timeStep diagram is as followed:



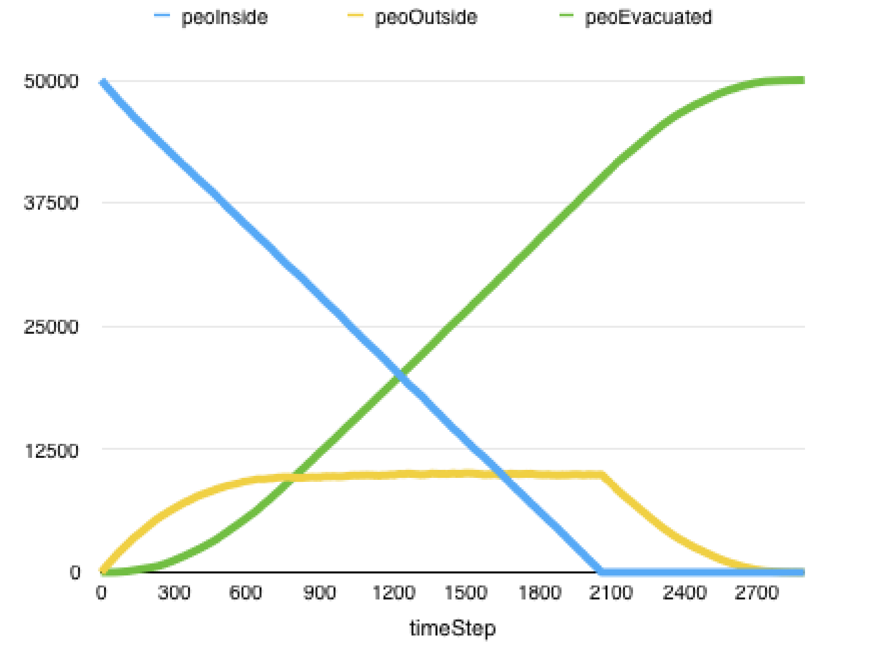
Total egression time: 863.1s
Stadium egression time: 612.9s

6.2.3. Scenario 3- Bobby Dodd Way NW closed off for pedestrian to evacuate

During the visulized simulation process, we produced some screen shots:



The egression number of people vs timeStep diagram is as followed:



Total egression time: 867.0s
Stadium egression time: 616.8s

6.3. Scenario Comparison and Analysis

The following chart shows a comparison between simulation results between the three scenarios analyzed above about the egression time as one of the most important output data:

	Scenario 1	Scenario 2	Scenario 3
Total egression time (s)	874.2	863.1	867.0
Stadium egression time (s)	623.7	612.9	616.8

From this chart, it is obvious that both total egression time and stadium egression time are longest in Scenario1, shortest in Scenario2, and mediocre in Scenario3. This result is pretty reasonable considering that closing off surrounding streets would speed up the evacuation process, which in return help with the model validation. As for the comparison between the three scenarios' screen shots, we may visually notice that the without closing off the streets, pedestrian tend to get congested on the narrow streets, especially on Techwood Dr NW. While closing off streets would mitigate the congestion phenomenon. Notice that people on Techwood Dr NW tend to walk further from each other, resulting in shorter egression time, higher egression efficiency.

7. Future improvement

7.1. self-feed back system coupled outflow distribution and crowd behavior

The parametered poisson distribution we assumed is fixed during the whole simulation process, whose underlying assumption is the collective behaviors of the pedestrians egress is relative consistent in a long period of time evolution. This is a good assumption for a event with large scale population, but the limit is that if the simulation touch the end period, the fixed-parametered poisson distribution might not be able to describe the collective outflow from the stadium very well, because the total number of the pedestrian inside the stadium has decreased to a small number, which make the basic assumption of poisson distribution fail. An alternative way to improve it is that we can assume the expectation parameter λ varies with the variation of population inside the stadium or with the evolution time. For example,

$$\lambda = \lambda(N_{inside}) \text{ or } \lambda = \lambda(t) ,$$

in which we can track the variation of λ or the gradient of λ with time. If we plot a figure about variables of our model vs the gradient of λ, we can obtain more underlying relationship during the crowd behavior and the real egress distribution. Such a nonlinear system is complicated time evolution system coupled with the decision distribution and the system evolution. In other words, the outflow distribution will decide the dynamical behaviors of the crowd, while the state of the crowd will affect / feed back the outflow distribution. Therefore this is a coupled system whose characteristics frequently appears in the real world.

7.2. traffic guidance factors

An important factor is traffic guidance, we will introduce the traffice guidance into our model in the future and study its influence on the dynamics of the crowd. This factor can provide more freedom to optimize the stategy about how to assist the egress of the pedestrians. Especially in the case including the vehicles flow in the main road which might significantly affect the egress of the pedestrians. Basically, we can divide the traffic guidance into two types: local and global. The local guidance use the strategy based only on the local traffic situation, while the global guidance can get the traffic information of the whole area to make a global optimized strategy but not necessary to be locally optimized. The comparison of these two strategy is an interesting topic to discuss in the future.

7.3. Group behavior of the pedestrians

In our conceptual model, we initially talked about people’s possible group behaviors during evacuation process, i.e. people within a group would stay together for the whole period and form a block, or they would stop if some group member stays behind. However, such a model would result in a complete congestion situation if we do not realign the group member in high density area. Due to the limited time and energy, we would leave this point for future improvement.

References:

[1] D. Helbing, A mathematical model for the behavior of individuals in a social field, Journal of Mathematical Sociology, 19 (3) (1994), pp. 189–219

[2] Weiliang Zeng, Hideki Nakamura, Peng Chen, A Modified Social Force Model for Pedestrian Behavior Simulation at Signalized Crosswalks, Procedia - Social and Behavioral Sciences, Volume 138, 14 July 2014, Pages 521-530, ISSN 1877-0428, http://dx.doi.org/10.1016/j.sbspro.2014.07.233.

[3] C Burstedde, K Klauck, A Schadschneider, J Zittartz, Simulation of pedestrian dynamics using a two-dimensional cellular automaton, Physica A: Statistical Mechanics and its Applications, Volume 295, Issues 3–4, 15 June 2001, Pages 507-525, ISSN 0378-4371, http://dx.doi.org/10.1016/S0378-4371(01)00141-8.

[4] Xiao Yang, Binxu Wang, and Zheng Qin, “Floor Field Model Based on Cellular Automata for Simulating Indoor Pedestrian Evacuation,” Mathematical Problems in Engineering, vol. 2015, Article ID 820306, 10 pages, 2015. doi:10.1155/2015/820306

[5] Jun, Chulmin, and Hyeyoung Kim. "A 3D Indoor Pedestrian Simulator Using an Enhanced Floor Field Model." Agents and Artificial Intelligence. Springer Berlin Heidelberg, 2010. 133-146.

[6] Kirchner A, Schadschneider A. Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics[J]. Physica A: Statistical Mechanics and its Applications, 2002, 312(1): 260-276.

[7] Ministry of Haj - Workshops, http://www.crowddynamics.com/Company/Ministry of Hajj Workshops.htm

[8] Moussaïd M, Perozo N, Garnier S, et al. The walking behaviour of pedestrian social groups and its impact on crowd dynamics[J]. PloS one, 2010, 5(4): e10047.

[9] Stephen K. Park and Keith W. Miller (1988). "Random Number Generators: Good Ones Are Hard To Find". Communications of the ACM 31 (10): 1192–1201 .doi:10.1145/63039.63042.