

# Crowd Simulation

## Introduction

The increasing proportion of people living in urban areas brings new challenges to urban planning and architecture. Crowd simulation plays an important role in addressing these challenges. With the help of crowd simulation technique, urban designers or architect could determine the evacuation time of a massive crowd, detect the behavior of crowd flow inside the building and prevent overcrowded area during certain events.

Crowd is created when a large amount of people gathers in a limited space. Taking the whole crowd as a unit could help understand the behavior of the moving crowd. However, if we divide the crowd into numbers of groups that contains 2 to 3 people or single, the behavior of crowd can be more complicated. In the group, people know each might walk together. Previous researcher Reynolds [2] proposed steering approach Leader Following (LF). This approach is basically one of pair agents would be the follower who follows the leader and stay on its side. This approach contains one disadvantages in the simulation, in this basic steer approach, leader agent does not wait for its follower agent if distance between these two agents is too large, which is not realistic.

Simulating a crowd of people needs complicated calculation, previous approach [3] design agent as ellipse to have sense of environment and simply plan ahead their own path to avoid agent collisions. Unfortunately, the output of simulation is lack of realism and flexibility. Since it does not involve complex behavior such as allowing agents to move in and out of different group or line based on agent's desire, agent who has plan the path ahead might lead the agent directly queue up in the longest waiting line without hesitating. However, in reality, people do not just stay at their waiting line once they choose it, they might need to change waiting line if there has a better option.

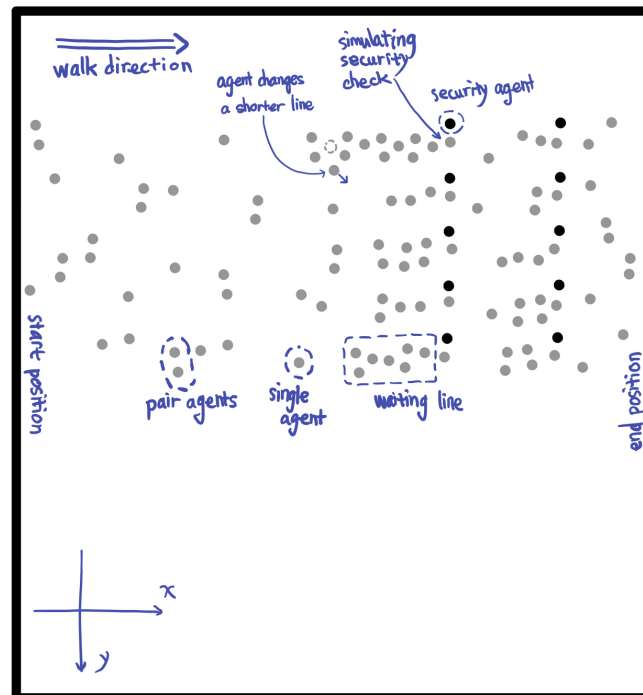
## Proposed project objectives

In this project, I will develop a crowd simulation application which aim at creating realistic, unique and accurate crowd.

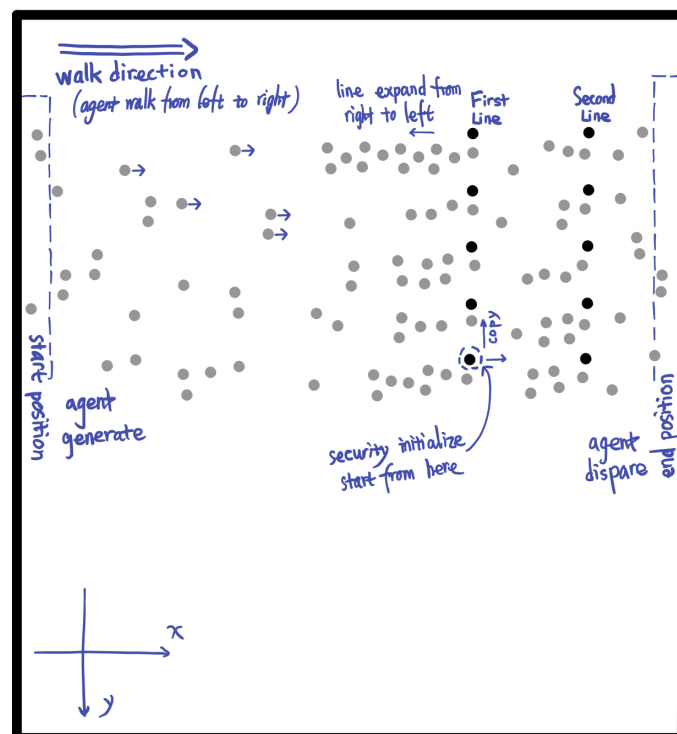
Scenario: Single agent or pair agent are randomly generated within the initialized range; each agent is initialized with a default start position and end position, and they will walk to their end position. However, before agent reaching their destination, every agent has to finish security check first. (people do ticket checking and security check during the concert event) Thus, every agent needs to stop near the security faculty to simulate the security process. Since the new agents is generating and security process takes time, the number of agents in the scene increase dramatically, the crowd created. However, instead of generating a massive chaotic crowd, agents in the crowd will queue up orderly and form several waiting lines, and each agent in lines will do security check one by one. After finishing the security check, agents will move to their default end position and despar.

To achieve this goal, I will use an open source state of art navigation mesh construction toolset Recast to achieve static avoidance and shortest path calculation. What's more, I will also utilize a path-finding and spatial reasoning toolkit Detour to achieve dynamic avoidance among agents in the path and to completed calculation of each frame of the simulation [4]. By taking the open source as platform to build the lower level of my approach - QueueBehaviorApp, I will simulate crowd behaviors such as pair walking, queue up and form single/pair waiting lines, agent switch from one waiting line to the others and do security check. Image below is the mockup demo that demonstrates the scene I will create. In this scene, agents generated and walk from left to right. Among each agent, some agents might know each other, so they walk with each other; some agents walk along. Once agent reach the security gate (place security agents stay), agent stops for a few seconds to simulate the security checking process. If the security gate is occupied, the upcoming agents wait behind. Thus, the waiting line created. Agents in the waiting line are not satisfied, they are seeking for chance to pass the gate as soon as possible. Therefore, agents who queue up near tail of the waiting line will look at

left of right side of line to find out if there has any shorter line to go, if it has, agent will leave its original line and queue up to a better line.



However, before doing that, I will initialize scene to generate input for the crowd simulation. Below is the mockup demo of the application interface with detailed requirement that could help understand the input and default requirements of the crowd simulation:



In this simulation, agents are initialized at the left side of the scene (*start position*) and vanish at the right side of the scene (*end position*). Agents in scene will move from left to right, this moving

direction determines the direction of the waiting line, the waiting line grow from right to left. In the simulation, there has two line of agents always standing at the same position in the whole simulation. These two lines of agents play roles as security faculties of the event in the simulation to simulate security check. The upcoming agents will stop in front of the security faculty for a few seconds to simulate security check (or ticket check) as we do in our real life. Once agents finish check, they continue moving to their final destination.

## Method

Program will implement an open source Recast/Detour in Java to achieve the lower level features. I will create crowd simulation by implementing knowledge of algorithm and data structure to generate output data file that contains information of every agent coordinate in every frame, and then utilize HTML canvas to create animation to demonstrate the result.

To make sure the simulated crowd behavior is realistic and dynamic, I will first collect different crowd behavior features from several rea-life crowd video recorded by Dr. Ricks research lab. Those videos recorded the walking crowd during several events such as concert, Disney On Ice, etc.

## Expected results

Agents single or in pair walk across the scene and pass through two lines of gates. When there has large amount of people appear, agents line up and create certain number of waiting line and each agent consecutively pass through the gate one by one. By comparing length of distance between agent's current position to the gate and length of nearby waiting line, agent in the waiting line might increase or decrease it anxiety level. Once agent's anxiety degree reaches the upper bounce, agent will leave its own waiting line and line up at a new line.

## Format of report

1. Application source code
2. Application animation
3. Application screenshot

## Project mentor

Dr. Brian Ricks

## Qualifications

Comparing the simulation result with the real-life video record to see how simulation result close to the video.

## References

- [1] Popelová, Markéta, et al. "When a couple goes together: walk along steering." International Conference on Motion in Games. Springer, Berlin, Heidelberg, 2011.
- [2] Reynolds,C.:Steeringbehaviorsforautonomouscharacters.In:GDC,pp.763–782(1999)
- [3] Baig, Mirza Waqar, et al. "Realistic modeling of agents in crowd simulations." 2014 5th International Conference on Intelligent Systems, Modelling and Simulation. IEEE, 2014.
- [4] Open source React and Detour. <https://github.com/ppiastucki/recast4j>

