Thoughts on first draft of MS project report:

* Is there an official **MS project report template** on the graduate studies website? Is so, you should be using it.
* This is a great first draft. You have the right content and I’m impressed with your overall grammar. There are some changes to be made, but overall, this is far better shape then almost every first draft I’ve seen from a master student.
* The main changes we need to make are in terms of organization. For my first edits, I’m going to give comments on just the first sentence of each paragraph as follows:

Understanding complicated crowd behaviors is essential to urban designers and architects. 加完

**1 Introduction**

The increasing proportion of people living in urban areas brings new challenges to urban planning and architecture.

A crowd forms when a large amount of people gathers in a limited space 移到implementation

More recent simulations of crowds of people use more complicated calculations. 移到 previous work

**2 Related Work**

Many simulations have achieved complicated behaviors [待做]

**3 Report Outline**

In this section, this report presents important components of this application which use the open source library Recast & Detour Outline 合併到 Intro [標題的**數字待改**]

**4 Resources**

This application uses an open source library Java Port of Recast & Detour navigation mesh toolset [7]. [標題的**數字待改 – 完8**]

**Recast**

Recast is a state-of-the-art navigation mesh construction toolset for games.

**Detour**

Detour is a spatial reasoning toolkit which accompanies with Recast to offer a simple static navigation mesh.

**5 Observations from Video Records**

In order to understand how a crowd moved, I study and analyze of recorded crowd videos at CenturyLink on different entrances during events such as concerts and a Disney on Ice Event.

**5.1 Pair Walking**

Normally, people attend events with their friends or family, thus people in the crowd are divided into numbers of small groups.

**5.2 Queue up Behavior**

When a waiting line is formed, people simply queue up and slowly move forward in the line.

**5.3 Security Checking Behavior**

For people who reach the security gates, they have two security processes to finish before entering the building.

Based on those videos, the scenes contain two lines of security gates. [提及agent到達的模式···**待做**]

**6 Foundation architecture and Scenario**

In this project, I developed a crowd simulation application which aims at creating realistic, dynamic and accurate crowd.

Scenario: Single agents or pair agents are randomly generated from the virtual entrance; each agent is initialized with a default start position and end position, and they will walk from start to end position.

**7 Environment Implementation and Setup**

**7.1 Input Data Initialization**

Input file allows us to determine the basic scene information such as agent id, agent start time, start position, end position and behavior mode. Instead of directly applying data from the real-life video, I need to manually generate agent data based on features I observed from the video.

In order to achieve more crowd features, I designed input data base on the following patterns: Agent id, enter time, start position, end position, behaved mode.

Start position determines where agent will appear. End position determines where agent

will exit, it also determines agent’s walking direction.

Below is the data input sample:

Blue is agent id; each agent has its unique id; green is agent’s enter time; 3656 means agent enters the scene at 3656 million second; orange is the start position (x coordinate, z coordinate, y coordinate); red is the end position (x coordinate, z coordinate, y coordinate); purple is agent’s behavior mode.

In the simulation, agents who stand next to their corresponding gate are representing security faculty in the real-life video.

Here is a demo to help us better understand how to determine range of start position, end position and security gate lines.

Each circle represents an agent, agents in black are security faculty, agents in grey are normal agents.

**7.2 Environment Initialization**

Before achieving the complicated crowd behavior features such as pair walking, queuing up and form single/pair waiting lines, I need to initialize environment for agents so that they could perform security check.

At the beginning of the simulation, environment is initialized based on the input data that I mentioned above.

The code below are always runs at the beginning of the simulation.

Methods above cover all the initializations required in the simulation. Description below shows more about details of the initialization:

**7.3 Appearance Feature of Upcoming Agents**

In the simulation, in order to create a realistic crowd, I intensively control the flow of crowd and let it follows certain pattern.

Charts above roughly illustrate the number of new agents appear every 1320 million seconds.

**8 Agent Behavior Implementation**

**8.1 Pair Walk Behavior**

In the real-life video, people in group are talking to each other while they are walking.

**How to Achieve Pair Walking?**

Agents in pair adjust theirs moving speed based on angle of two vectors: one is from leader’s current position to leader’s current destination. Another one is from leader’s current position to its follower’s current position.

Bases on the value of angle, agent’s speed has three states:

However, knowing the angle between two vectors is not enough to create the natural speed adjustment

**8.2 Queue up Behavior**

In order to enter the event, people need to do security check, they stop at the security gate for a few seconds.

State diagram figure\_state shows sequence of behaviors each agent in process of security check.

To simulate this queue up behavior, I allow each agent to have the following status:

In the simulation, I utilized 4 states to determine the current state of agents.

For agent that could directly reaches the security gate, agent starts to count the bag/ticket checking time.

Because agents in pair relationship will walk in pairs, they still stay side by side even after queuing up. Thus, inside the waiting line, conditions between agents could have the following statuses:

As I mentioned above, follower agent is set to stay next to its leader agent.

In this simulation, each agent needs to finish two security processes.

However, space between gate lines is limited. Each agent takes different time to finish the first security check

**8.3 Re-consider Behavior**

Once agents enter the scene, each agent is assigned a value to represents its anxiety level.

However, simply assigning anxiety level to agents are not enough to simulate the reconsidered behavior—I also have an anxiety monitor to adjust the anxiety degree based on the waiting lines conditions.

Anxiety degree updates in certain internal conditions:

Once the agent’s anxiety degree reaches the maximum value and have a shorter line on left or right side.

**9 Evaluation 待做**

**9.1 Input Data Evaluation**

The evaluation of the input data generated by QueueBehaviorApp follows the requirement of the simulation.

**9.2 Behavior Evaluation**

Based on observations of the simulation results, agents could achieve the following behaviors:

In the simulation, 260 agents are randomly generated which are distributed at one side of the scene.

**Simulation Results:**

Based on the observation, some agents walk in pair.

Once agents are generated from their start positions.

During the waiting process, anxiety degree of each agent in the waiting line could represent desire of line changing.

**10 Future Work**

In terms of future work, the application could only allow two agents walk in pair.

**11 Acknowledgements**

**12 References**