**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 (Big goal).

Crowd is created when a large amount of people gathers in a limited space.

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. [*(1) Other researcher’s limit*]

Simulating a crowd of people needs complicated calculation, previous approach [3] design agent as ellipse to have sense of environment and plan ahead their own path to avoid agent collisions. Unfortunately, past crowd simulation approach is lack of realism and flexibility because 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. In reality, people not only just directly walk from one place to the other, but also queue in line to buy ticket or do security check before enter concert. [*(2) Other researcher’s limit*]

[*what and how I could improve those disadvantages.*]

**Proposed project objectives**

In this project, I will develop a crowd simulation application which aim at creating realistic, unique and accurate crowd. Implemented open source state of art navigation mesh construction toolset Recast to achieve static avoidance and shortest path calculation by generating simulation data. Utilized 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. (<https://github.com/ppiastucki/recast4j>).

***To be continue….***

(Briefly talk about what you observed from video we recorded…)

Features I plan to achieve:

* Determine shortest gate
* Pair walking
  + Adjust speed to walk side by side
  + Stay pair even queued already
* Queue up behavior
* Form a single/pair waiting line
* Agents reconsider and change line for shorter pass
* Agents walk through gates and line up behind another line
* Agents do two-phase security check
* Agent wait at first gate if there has no room to queue up in second phase
* Agent who finished check wait for partner to finish final check
* Anxiety update

Utilized Java and implemented algorithm and data structure to simulate real-life crowd movements based on annotated crowds to achieve simulated behaviors such as pair walk, change queued up line, single and pair queue up mode, passing intermediate gates and change to shorter line action.

**Method**

Data Structure

Algorithm

**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

Project mentor

Qualifications????

**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]