PROGRESS REPORT4

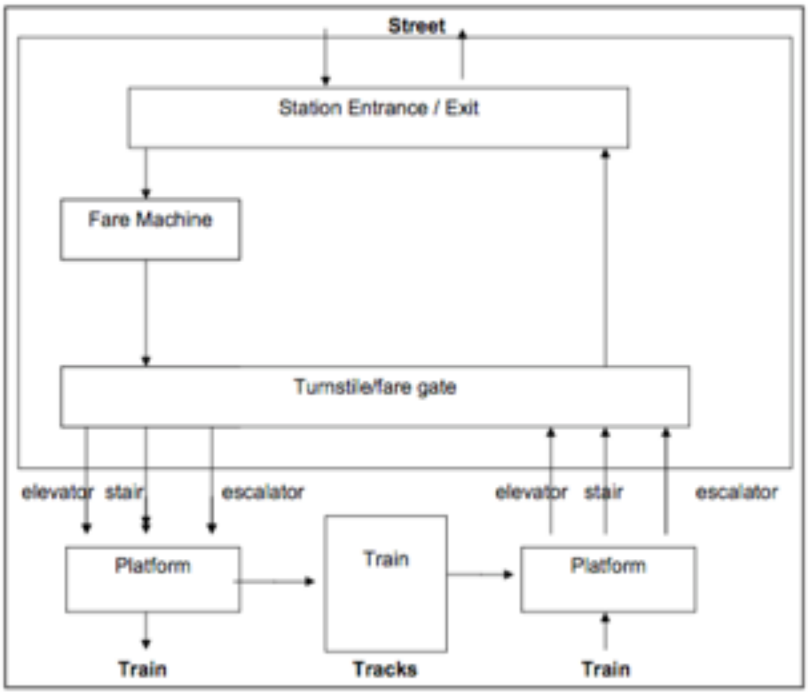
Agent behaviour:

Agents start from left side of the world, which is outside of the station, move through the entrance and get on the train. The entrance is represent by a white blank. The black area is the wall and agents can not move through the wall.

System behaviour:

Number of turtles start from outside the entrance and move to the train. If there are too many turtles trying to get through the entrance, they might stop and wait. They should never run into other turtles.
I made the patches the same size as agents and allow only one turtle on each patch. This is how turtles avoid bumping to other turtles.
Agents are always trying to move towards the train, so if there is other turtle on the patch ahead an agent, it will move to neighbours to find new path. If it can't move anywhere, it stop and wait.

```
facexy dest-x dest-y                          ; dest-x dest-y is the coordinate of the train
if not can-move? 1 or patch-ahead 1 is wall   ; avoid run into walls
    [ find-nearest-patch ]                    ; find the neighbour which is nearest to destination
ifelse not any? other turtles-on patch-ahead 1  ; avoid bumping to others
  [ fd 1 ]
  [ set wait-time wait-time + 1
    move-to-neighbour ]
```



Now the train arrives periodically. Turtles will wait outside the train until it's arrival.
The structure of the station could be changed by varying inputs, e.g. stair-spacing and width. Almost all the subway stations have a similar structure as shown in above figure. I'm modelling the flow from fare gate to the train. Thus what I draw could represent most station structures. Usually, platforms on two sides of the tracks are symmetrical. Therefore, we only need to model one side. To be specific, now I'm modelling how the location and width of stairs (or passenger way) influence the flow speed.
Add three ways to get to platform, stairs, escalator and elevator.
Add time to whole system. 400 ticks represent one hour in real world.

Rationale for agent rules

Agents find the shortest path to get on the train and avoid collide. I give the agents these rules to make them behave like a real person.
Turtles rule is as shown in the code above.
Turtles get on the train at random cars. The distribution is similar to reality since random numbers are dense at the middle and rare at both ends. In reality, people are more likely to boarding at middle of the train.
Turtles now find the nearest passenger way (or stairs) instead of randomly finding passenger way.
I have create another model, which is a neural network, taking population in different directions and choosing the best direction. In next version, I'll connect these two models. Agents in "crowded station" will input current platform information to neural network model and take the output of the neural network as the decision about where to go.
Turn neural net on and let agent making decisions based on the result of the neural network. Turning neural net on leads to a longer waiting time, which is reasonable. Because neural net simulates a decision from a real person—walk as less as possible and get in a near queue whose size is acceptable. This is not so efficient as randomly choose a car.

Model output

I didn't come up with new measures. I think number of people getting on trains in a certain period and mean waiting time can reflect the performance of a station structure. Because a good design of the station structure is to make more people move through this area in the same time. Longer waiting time shows the flow didn't move smoothly, which might lead to less people getting on the train in the same time.
Plot waiting time per hour. Add measurements about population outside the entrance, on stairs or elevators and on the platform.

Questions:

The output of the neural network is not always correct. I need to figure out how to train it to achieve a higher accuracy.

Next steps:

Improve the neural network

Model Analysis:

Increase width of entrances reduce the waiting time. Because people enter the station from two ends of the station, if the stairs (entrances) are near two ends, it will achieve a faster flow speed. Model shows that this claim is true.

Test model on the real flow data. Waiting time increases as population increases. There are two peaks in the diagram of waiting time in a day, which is consistent with the flow population in a day.

Experiment shows escalators are more efficient than stairs and elevators.

Advanced feature:

Inspired by the sheep model Bryan showed last week, now I plan to use LevelSpace to give each agent a “brain” that they can “think” and make decisions to wait or finding other ways and so on. For example, suppose there are two entrance to the platform, and an agent has moved forward only 2 patches in 10 ticks, it may think to use another entrance. But changing queues is a time-cost action, so it may evaluate the waiting time for these two queues. Sometime after making a wrong decision, the feedback of the neural network could help it make better future decisions.

My neural network have three inputs: population on the left side, population in front of the agent and population on the right side. The only output is turning left, right or going straight. The specific area that in front of an agent is a square whose width is 3 and length is the distance to the train. Area above the area that in front of an agent is considered as the left side of the agent. Then the right side is the area under the area that in front of it.