Final Project Checkpoint 2

CX 4230 - Dr. Rich Vuduc
Group 1 - Ausaf Ahmed
Liam Frank
Mudit Gupta
Rahul Katre

Significant progress in simulator development

Code: https://github.gatech.edu/cx4230-sp22-group1/campus-flow

Summary of accomplishments

Ausaf	I completed part of the CV object detection with Liam. The goal of this module is to parse the raw map files and extract an object representation of the map elements so that they can be fed to the program and form a high level representation of the environment. The CV object detection will take the map and find the shapes and colors of the objects. The colors represent different types of objects, and the shapes represent individual objects or parts of objects. In this way, attributes can be assigned to different types of objects so that they can interact properly in the simulation. I also created an initial map for the simulation to use with Liam. Currently, this map contains 2 buildings, each with 2 doors, as well as a walkway between the buildings. Using shape recognition algorithms, we detect rectangles, then perform a proximity based search for different colored rectangles in order to enumerate a building's entrances. This is then stored in JSON format which Mudit's React code will read and use to build the potential field for the map.
Liam	I created a Python script to generate the schedules for the particles to follow each day, which is stored in a JSON file for the simulation to input. Examples of schedule variables include start and end building location, building locations for each event (such as attending a class), and start and end times for each event. This includes randomly selecting certain variable values, which allows for better simulation runthrough and results. The generator also weights certain variable values, which better reflects the real-life patterns of actual people on the Georgia Tech campus. For example, most students start and end their day in a residential building, so those buildings have higher weights for the start and end building location variables. I also worked with Ausaf to get part of the CV object detection done and create an initial map for the simulation to use.
Mudit	• Setup React website with p5.js sketch to visualize the particles and their motions based on the vector field • Work with Rahul on generating a vector field based on an input map image by using gaussian blurs and gradients • Tune parameters in simulation for attraction force towards the goal, friction force on particles, and momentum of particles

	● Created a python script to take in a directory that contains a map.png and will generate a field.ts within the same directory ○ This field file will be automatically imported by the simulation to ● Add randomness to the field vectors such that particles are less likely to get stuck in local minima within the field.
Rahul	● Worked on adding vector field normalization to the field preprocessing. ● Also worked with Mudit on finding good values for parameters like sigma for Gaussian blur, and different field strengths. ● One of the issues we ran into was that particles would cut through obstacles, which required lowering the attractive force towards the goal.

Remaining Work

Ausaf	The rest of the CV object detection needs to be finished with Liam. The map design also needs to be finished with Liam, but we need to wait until the simulation works on a basic level with the initial map. Otherwise, if the simulation's interaction with the map changes, then there would be no use to create more advanced maps.
Liam	There is a schedule for the particles to follow. However, the schedule variables from the Python script are not completely in sync with the schedule variables for the simulation. As the pieces of the simulation come together, there will be a clearer picture for what specific schedule variables are needed, as well as their spaces and constraints. I will need to go back and update the Python script so that they match once this happens. Additionally, I still need to finish the CV object detection and map design with Ausaf.
Mudit	● Continue working on tuning parameters for simulation such that particles will more optimally navigate around obstacles to reach their goals ● Integrate the CV object detection for building entrances into the simulation code. This will involve adding new fields to the generated map files that encode the building locations and each of their entrances.
Rahul	● Currently, the particles no longer cut through obstacles, but it takes them a while to get unstuck. ● Imagine you were trying to get to the other side of a building from the outside but instead of walking parallel to the wall you just walked head on and slowly slid left till you reached the corner and then walked straight. ● Resolving this will require some more tuning with the field and how the particle responds to it. ● So far, we have tried adding slight randomness to the field, but the original paper uses directional weighting so we will try that.

We have also uncovered a difference/improvement made in our implementation of the social force model compared to the model proposed in the paper that introduced the social force model. The original model computes the force from obstacles and attractive things based on distance to the closest point on the edge/border of each obstacle/object in the environment. For the simple environments showcased in the paper, where there are only 1-2 borders, this is reasonable. But for our project, where there are multiple obstacles and many other objects in the environment (as our environment is supposed to be a model of campus), this can become inefficient. Instead, our solution is to process an image and compute a gradient vector from each pixel in the image. The resulting vector field only needs to be computed once per scene, and is constant for the duration of the simulation; particles only need to read the field.

We use masks and brightness values as a potential function, by using Gaussian blurring to smooth out the masks/potentials. The Gaussian/normal distribution function is monotonic and decreasing for our purposes since the particle only interacts with the closer side of the normal distribution “curve”. This is an important detail since the paper also specifies a monotonic decreasing potential function. Our simulator also incorporates the idea of attractive objects in the environment, which in our model is walking paths between buildings.

Original paper: arXiv:cond-mat/9805244v1 [cond-mat.stat-mech] 20 May 1998

Future plans

- Add pause button for simulation
- Disable clicking inside obstacles