# Final Project Checkpoint 1

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

## Project Description

### An abstract summarizing the system and the goals of the project

The system we are developing will be a way to simulate the pedestrian flow of students across campus as they navigate between buildings throughout their day. The goals of the project are to identify how human crowd behavior emerges from pedestrian traffic rules and localized pedestrian decision making through the social force model. We will also be running the simulation on a map that resembles the campus more, which will help us gain insights about choke points and areas of high traffic during the day, and how people take different routes to reach certain locations based on a schedule that they follow.

### Description of the system being studied

The system will consist of particles that represent the students that attend Georgia Tech. Each one of these people has things to do or places to be at a specific time, and this agenda will be intrinsic to each particle, which we call a particle's "schedule". The schedule consists of a list of (location, time, duration) objects so that when a particle reaches its destination (location) at a specific time, it will remain there for a duration of time until it needs to go to the next location. People could start off their day at a certain building (dorm), and will visit locations in their schedule throughout the day, and end their day at their dorm. The destinations are the entrances of buildings, but the rest of these buildings also act as obstacles that particles need to avoid, since in the real world, you can only enter a building from defined points. Our simulation environment will begin at a smaller scale with 1-2 buildings and other small test scenarios, but we intend on expanding to a larger map of the main part of campus.

### A conceptual model of the system

We plan on implementing a model similar to the social force model described in the paper "Social force model for pedestrian dynamics" by Helbing and Molnar. The model defines several forces that act on each particle, which are attractive forces towards the destination (shortest path), repulsive forces from other particles and obstacles (personal space), and attractive forces from some particles and objects in the environment (friends, displays, etc.). The paper also describes directional weighting since things in the environment behind a person will not have as much of an affect on their behavior.

## Platform of development

Due to the ease of visualizing moving particles using web technologies, we will use a React website that displays the moving particles through a p5 sketch embedded in the page. The website will be hosted on GitHub pages. Much of the simulation code will be written using TypeScript, but for situations where more efficient code is needed to compute the different vector fields, we may resort to Python and Numpy, especially when doing precomputation tasks that only need to be run once to generate a base vector field that is a constant in the simulation. There is also a P5 module for Python, so it is possible to offload all computation to a Python/Numpy function, and only use P5 for visualization.

## Literature Review (pasted from previous submission)

### Modeling Flow of Pedestrians on GT Campus

For our topic, we want to simulate pedestrian traffic flow on campus at a microscopic scale, meaning that we would be simulating individual “students” moving between buildings. The students would be following a schedule, meaning that throughout the day they would travel between the buildings on campus, remaining at a building until the correct time. Using the simulation, one could identify points of congestion on campus, and how the students navigate around these points of congestion to reach their destinations. For microscopic modeling, we would use either cellular automata or particles in a vector field. With a generalized version of cellular automata, each cell would represent a student. The color of the cell would determine the destination building of the student. Once the student has reached the destination building, they would change to a different color to indicate that they are heading to a new destination.

However, there are limitations to typical cellular automata models. One paper by Sarmady, Haron, and Talib talks about some of them, as well as ideas to work around them. Typical models have difficulties with smoother and accurate movements due to coarse grain discretization of space, in addition to limited choice of speed. The Fine Grid Model utilizes small cells, with pedestrians having varying shapes and sizes (i.e. a pedestrian would take up multiple cells versus a single cell). Transition rules and next cell calculations only consider a center cell, with collision avoidance considering all cells of the pedestrian. Also, typical models only have pedestrians moving at a normal speed until reaching an obstacle or another pedestrian. This model uses perception of density, where an individual pedestrian assesses the local density around them and picks a specific speed correlating to that density from an empirical speed-density graph. In addition, typical models make pedestrians move the same speed in cardinal and ordinal directions. However, the displacement is different for each type of direction, which can skew the actual distance covered in time moving diagonally. This model has the pedestrian select cells which give optimal speed values that produce the least error from a desired movement speed, capping the movements of each pedestrian in each second of the simulation if the combination of the movements results in a displacement bigger than the free flow speed of the pedestrian. By implementing these ideas, the cellular automata model is improved to act more realistically regarding position, movement, direction, and speed.

In their paper on human mobility patterns, Serok N. and Blumenfeld-Lieberthal E. explain their spatio-temporal agent-based simulation techniques. Though this technique was not exactly a cellular automaton, it used a map divided into thousands of small cells, each with a certain classification (residential, employment, entertainment, public space, etc). On this map, there would be many agents with a unique set of parameters including age, marital status, employment status, etc. These parameters would determine where each agent wanted to be at different times of day.

These simulation techniques are quite analogous to our goals of simulating pedestrian flow on GT's campus. Different parts of the map will be either part of a walkway, class building, residence building, or something else, and there will be many different students with unique schedules that determine where they want to go and when.Another relevant detail from this paper is that different agents sometimes have non-routine destinations (going to an entertainment place at night), which may be analogous to GT students stopping somewhere to eat/study while waiting between/after classes.

With particles in a vector field, each particle would follow a vector field that directs them to their destination, but the presence of other nearby particles would affect their own field, meaning that areas of high congestion would naturally be avoided by newly arriving particles due to a large field surrounding the high congestion area pointing away from it. This is similar in concept to the social force model introduced by Helbing & Molnar [3] which describes how forces would model a person's motivations for certain movements, such as acceleration and velocity, keeping a distance from other pedestrians and borders, and being drawn towards other elements in the environment. In their paper, they found how a simulation following this model led to the creation of lanes based on walking direction, and people waiting for a doorway to be cleared by people going the opposite direction. Given that human behaviors can be observed using this model, it would be useful for modeling the movement patterns of students on campus without rigidly defining specific rules for movement.

Another model that could be used for our project is a queueing network, which was used by Løvås [4] to model pedestrian traffic in an evacuation simulation. In their paper, they simulated a building with multiple rooms and doorways, and a walkway network going through each doorway terminating at an exit node. Each room, represented by a node, has a capacity limitation, and each doorway has an "effective width" limiting the amount of people that can move through the door at a time step. The simulation tool they created, EVACSIM, models each person individually as a queueing network customer in order to incorporate things like reaction time and other human behavior. If we used a queuing network for our project, then in theory a network could be made from different patches of the environment, forming a graph upon which paths would be routes between buildings on campus. The capacity of each node would represent the number of people that could fit in that node, just as how the room capacity works in the evacuation simulator. The width parameter of connections between nodes would represent attributes like path width, which would limit the amount of students that could move between nodes. Paths in our graph would have to be bidirectional because not all people in the simulation would have the same destination.

# References


1. A Simulation Model for Intra-Urban Movements. Serok, Nimrod & Blumenfeld-Lieberthal, Efrat. https://doi.org/10.1371/journal.pone.0132576
2. Simulation of Pedestrian Movements Using Fine Grid Cellular Automata Model. Sarmady, Siamak, Haron, Fazilah, & Talib, Abdullah Zawawi. https://doi.org/10.48550/arXiv.1406.3567
3. Modeling and simulation of pedestrian traffic flow. Løvås, Gunnar G. https://doi.org/10.1016/0191-2615(94)90013-2
4. Social force model for pedestrian dynamics. Helbing, Dirk & Molnar, Peter. https://doi.org/10.1103/PhysRevE.51.4282

# Current Project State and Initial Results

## Show of Progress

Code for the simulation can be viewed at
https://github.gatech.edu/cx4230-sp22-group1/campus-flow/blob/main/src/components/main/sketch.
The current model has randomly placed particles that feel an attractive force towards their respective goal locations. There is also a randomly placed wall that repels particles. This will slowly be expanded to exactly reflect the buildings and walkways on GT's campus.

We have also planned out how we will construct the simulation environment. For simplicity, the map that the particles will move within is going to be drawn in a photo tool such as Microsoft Paint, and the color of each pixel corresponds to some object in the environment. There will be different colors for walls, grass areas, walking paths, and entrances to buildings that will be the destinations for different subsets of particles. Using this image, we will create a vector field for the environment through a process of gaussian blurring and then finding gradients for each color on the map, before combining the different fields into a single base layer. This means that the vector field of the environment is precomputed and only needs to be read by each particle when simulating. Each particle will also have its own repulsive field ("personal space") so when updating a particle's position and velocity, it will take into account nearby vector fields. "Nearby" is determined by the equation we use for the vector field, for example there is the inverse square law.

## Course Corrections

No major changes

## Division of Labor

Below is a list of divided-up tasks:

1. (Liam and Ausaf) Creating the environment for particles
    a. Determine color mapping for environment entities (path, grass, wall, etc.)
    b. Create maps in paint ranging from simple (1-2 buildings) to complex (campus)
2. (Mudit and Rahul) Precomputing vector field of map from image
    a. Separate out different colors to apply blurring per color
    b. Use blurred image to define vector fields, combine different fields
    c. Use Python for efficiency, export as JSON or something for JS to read in
3. (Mudit) Using the vector field(s) to determine the forces on the particles
    a. Potentially more efficient particle-particle interactions (non-major goal)
    b. Potentially make use of Python P5 package for speedup

# Git Repository

## GT GitHub link

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