Name: Minh Anh Le
Email: minh26@pdx.edu

Course Project Proposal

Project Name: Epidemic Spread Simulation
Project topic area: mathematical epidemiology, sir-model, disease-simulation

Project vision:

This project is about creating an interactive simulation that shows how diseases spread through a population using the SIR model (Susceptible-Infected-Recovered), and demonstrates how prevention methods can reduce disease transmission. When I researched the SIR model, I came across a YouTube video from 3Blue1Brown channel (https://www.youtube.com/watch?v=gxAaO2rsdIs&t=507s) and found it interesting how I can use simulation models to implement mathematical epidemiology concepts. Seeing the disease spread by watching simulations made me realize how serious diseases can be and how prevention methods can significantly reduce their spread.

My main goals for this project are:

- Show how diseases spread from person to person through contact. Each infected individual has an infection radius, and as they move around, anyone who comes within that radius has a certain probability of catching the disease. Users can adjust these factors to see how they affect the outbreak.
- I also want to try implementing prevention methods like social distancing and quarantine to see how they reduce the disease spread.
- If I have time, I'll show how diseases can spread between different communities through travel, and maybe even implement travel restrictions between these communities as an intervention.

My Plan:

I'll start with a single community where everyone falls into one of three categories:

- Susceptible (S): People who haven't gotten sick yet but could catch the disease
- Infected (I): People who are currently sick and can spread the disease to others
- Recovered (R): People who got better and now have immunity (or died/got isolated and can't spread it anymore)

A community will have several factors that users can adjust, such as:

- Number of individuals
- Initial percentage of infected people

- Infection radius
- Percentage chance that a susceptible person catches the disease per day when exposed

I'll make all individuals walk around the community randomly. Then, I will implement two graphs that update as the simulation runs:

First Graph: This will be a line chart showing how the three populations change over time.

- The X-axis will show days
- The Y-axis will show what percentage of people are in each group
- Three lines (blue for Susceptible, red for Infected, gray for Recovered)

Second Graph: This will show all individuals moving around in a 2D space.

- Each person is a colored dot.
- The dots will randomly walk around the area, like people moving around in a community
- When an infected person (red dot) gets close enough to a healthy person (blue dot), there's a chance the disease spreads

Optional Features:

- Social Distancing: I will have a variable to set the minimum distance between people, and another variable to set what percentage of people follow the social distancing rules. I'm thinking about using separation forces to implement this feature.
- Quarantine Zone: I will move infected individuals to a quarantine area. There will be a variable to set how many days after infection someone gets quarantined, and another variable to set what percentage of infected people actually go to the quarantine zone. People in the quarantine zone can't spread the disease anymore and will eventually recover (become R).
- Multiple Communities: I will implement several separate areas, each with its own population. Initially, only one or two areas will have infected individuals. People can travel between areas, and there will be a variable to control how restricted travel is between these communities.

Challenges:

I don't have any experience with Rust visualization libraries yet. I'll need to learn about the GUI library to create the 2D view. This will probably be the biggest challenge of the project. Time is also going to be a challenge. I have a busy schedule this term, so I might not be able to implement all the features. That's why I've organized the project with core features first - at least I'll have something working even if I don't finish everything.