Pandemic Simulator Documentation

Introduction

This document outlines the theoretical foundation, mathematical modeling, and implementation details of the Pandemic Simulator, which is built using Python and Pygame. The goal of the simulator is to model the spread of a disease within a population based on probabilistic and mathematical principles.

Mathematical Model: The SIR Algorithm

The SIR (Susceptible-Infected-Recovered) model is a standard epidemiological model used to simulate the spread of infectious diseases. The simulator is loosely based on this model.

SIR Model Equations:

$$\frac{dS}{dt} = -\beta SI \frac{dI}{dt} = \beta SI - \gamma I \frac{dR}{dt} = \gamma I$$

Where:

- S represents the number of susceptible individuals.
- I represents the number of infected individuals.
- R represents the number of recovered individuals.
- β is the infection rate (probability of transmission per contact).
- γ is the recovery rate (rate at which infected individuals recover).

Implementation Details

Variables and Constants:

- **WIDTH, HEIGHT**: Dimensions of the simulation window.
- **POPULATION_SIZE**: Number of individuals in the simulation.
- **INFECTION_RADIUS**: The radius within which infection can spread.
- **INFECTION_PROBABILITY**: Probability of infection upon contact.
- **RECOVERY_TIME**: Duration after which an infected individual recovers.
- **MORTALITY_RATE**: Chance of an infected person succumbing to the disease.
- **SPEED**: Movement speed of individuals.

Person Class

Each person in the simulation has:

- A position (x, y)
- A status (Healthy, Infected, Recovered, Deceased)
- A movement pattern (randomized)
- Infection mechanics based on proximity and probability

Simulation Class

Handles:

- Updating movement and interactions
- Checking for infections
- Managing recovery and mortality
- Rendering the graphical elements
- Displaying a live graph for infected, recovered, and deceased counts

Enhancements

1. Visible Infection Field

Infected individuals emit a visible field indicating their infection radius.

2. Increased Population Density

Higher population density to simulate real-world urban settings.

3. Decreased Recovery Rate

Slower recovery mimicking real-world disease behavior.

4. Live Graph

A graph dynamically plotting the number of infected, recovered, and deceased individuals.

Conclusion

This pandemic simulator provides an interactive way to visualize the spread of diseases using fundamental epidemiological principles. Future improvements may include vaccination, social distancing, and quarantine mechanics.