Project: Interactive Simulation of Disease Implementation

Report: I finished the first part of this project by building a library called weave-3d that will handle the rendering for this simulation. It has similar functionality to THREE.js but a lot simpler since I made it from scratch. I based the concept off Unity, THREE.js, and BABYLON.js. It abstracts away a lot of low-level graphics by allowing the user to create Cameras, Scenes, Meshes, and Objects in a scene hierarchy. The scene hierarchy is then used to distribute update calls and draw calls.

All objects in the scene are represented by the GameObject data structure which handles positions, parent-child hierarchy, optional meshes, and allows the user to define custom behavior through the update() method. Every frame the main loop calls update() in all the scene objects similar to how Unity game engine works.

The library currently only supports basic models and no shading, but I want to add more in the future. However, for now it's more important to focus on the actual simulation part, then I can go back and add fancier features later. Currently the library is around 1000 lines long.

I have used this library to make a simple demo video that I have included in the files. The demo is quite simplistic, but it is just to show that the library is working. I was able to create this demo in only a couple minutes using the library functions.

```
WEAVE.init();

let scene = new WEAVE.Scene();
WEAVE.setActiveScene(scene);

let camera = new WEAVE.Camera(WEAVE.Camera.PERSPECTIVE);
WEAVE.setActiveCamera(camera);

let earth = new WEAVE.Sphere();
earth.mesh.color = new Vec4(0,0,1,0.5);
earth.update = () => {
    earth.transform.position = new Vec3(Math.sin(Date.now()/1000)*10,0,Math.cos(Date.now()/1000) * 7);
    earth.dirty = true;
};

let moon = new WEAVE.Sphere();
moon.transform.scale = new Vec3(0.3,0.3,0.3);
moon.mesh.color = new Vec4(1,1,1,0.8);
moon.update = () => {
    moon.transform.position = new Vec3(Math.sin(Date.now()/300) * 2,0,Math.cos(Date.now()/300) *2);
    moon.dirty = true;
};
earth.add(moon);

let sun = new WEAVE.Box();
sun.mesh.color = new Vec4(1,1,0,0.5);

camera.update = () => {
    camera.transform.position = new Vec3(Math.sin(Date.now()/5000) * 20,Math.cos(Date.now()/5000)*15,Math.cos(Date.now()/5000)* 20);
    camera.lookAt(new Vec3(0,0,0));
    camera.dirty = true;
};

WEAVE.start();
```

Current Status: I am a bit behind schedule since it took a lot longer than I thought to get stuff actually rendering. I wanted to have the crowd already simulated by now but I think, given how difficult making the library was, this is not a bad spot to be in. In retrospect I should have just used THREE.js to make my life a lot easier but I think it was a good learning experience and hopefully I can continue working on this and use it for other projects in the future. So far, I'm still on track to finish the project in time.