With the advent of social networks such as Facebook and Twitter, everyone is familiar with networks of people they know. However, the physical network of strangers you meet every day has been neglected. Unfortunately there are no currently no ways to discover how large your own physical network is. By taking advantage of the interactions between pathogens, we can finally answer these questions and find out so much more. Wouldn’t you want to know how many people you unknowingly interact with every day and how far your influence reaches?

Likewise, the user, or host, is the vessel in which their pathogen – and other users’ pathogens - can fight and infect other hosts. However, unlike typical role-playing games in which the combat system is highly active and requires turn-based or live-action decisions, the combat system of Pathogen is based entirely on passive encounters between players. Utilizing the geo-location technology available on most Android phones, Pathogen initiates interactions between users - and combat between their pathogens - whenever two users come in real-world, physical proximity with one another. In other words, Pathogen replicates the transmission process of real-world contagions to introduce a unique and interesting game dynamic to the ever-familiar strategic role-playing genre.

The Architecture is split into two parts: The app and the server. On the app side the player creates his pathogen while monitoring how far his pathogen has spread and mutated, and the phone sends periodic geolocation data to the servers. From there, the server stores all data to the noSQL Google Datastore, where it is processed by two Java processes hosted on the Google App Engine backend: the Geocell-based (hex strings instead of latitude/longitude, as pictured) location proximity algorithm, which runs periodically and powers the primary location sensing process of the application; and the battle algorithm, which executes ‘battles’ between two users in proximity and updates player data (health, infections, etc) accordingly. Data is pushed to user’s application via the Cloud Endpoint Messaging API, keeping users updated on their game progress.

Pathogen is a unique strategic mobile game for Android in which the user’s real-world experience is as essential to the gameplay as the decisions they make in-game. In Pathogen, users compete against each other through the creation of infectious, aggressive creatures called “pathogens.” Each user is only allowed a single pathogen. Much like real world contagions, the sole imperative of these pathogens is to infect as many hosts as possible and spread their numbers as far as they can. These pathogens are the vessels by which the user competes and interacts with other players in the world. Much like in tactical role-playing games such as Dungeons & Dragons, the specific traits of each pathogen can be tailored to however the users finds fit to facilitate the domination of their pathogen over others.

Take Away Lessons

Lesson 1: Start with blank IDE’s and sync them whenever possible.

A major problem we ran into was the applications we built individually would not work when brought together on one computer. We faced many crippling bugs and sleepless hours trying to make things work, and eventually we were able to make a working app by creating an entirely new project and copying our code over.

Lesson 2: Double the time you expect to get things done

We had an elegant and beautiful schedule to finish the project with plenty of time to debug and beta-test the app. The plan fell apart by the second week. No matter how fast you think you can code and debug the software, just plan to take more time. It can’t hurt.

Lesson 3: Don’t rely on the coolest and newest software.

We originally planned to use Amazon AWS and Skyhook APK for our server and location software. We couldn’t make them work in time. Instead we used the network/GPS service on the phone and Google App Engine for our server. Even then, making Google App Engine took a significant investment for a service that we did not take full advantage of.

References
