Project Overview *[Maximum 100 words]*

Our project is to create a simulation of the *Plague* game. The point of the game is fairly simple, infect as many people in the world with a new type of killer pathogen. The user is able to choose the starter country to spread his disease and then he is able to see how long his disease will take to infect the entire population.

Results *[~2-3 paragraphs + figures/examples]*

At the moment we have three different countries for the user to choose where his disease will propagate from (screenshot #1) Once this is done, the disease starts to spread and infect more and more people. There are a few counters in the bottom right hand corner which allow the user to see at a glance how many people are infected, how many are dead, and how many are left alive (screenshot #2)

We have added the ability to upgrade the disease's features to make it even more contagious. The user is therefore allowed to press T, S, or A to upgrade a different disease ability. The first one is the infection rate, the second one is the death rate, and the third is the airborne rate which makes it easier to reach other countries.

Implementation *[~2-3 paragraphs + UML diagram]*

Describe your implementation at a system architecture level. Include a UML class diagram, and talk about the major components, algorithms, data structures and how they fit together. You should also discuss at least one design decision where you had to choose between multiple alternatives, and explain why you made the choice you did.

- One design decision which impacted our overall project was how the game was going to function on the back end. We thought that we could have a doctor class and a virus class work alongside each other and that everything could be brought within a UI class. This brought in a whole realm of issues because we had to figure out if there were going to be rounds within the game, or if the game was going to function off of time. Yet we realized that it was easier to create a country object which

would be responsible for its own population and how quickly the infection spread within it.

Reflection ***[~2 paragraphs]***

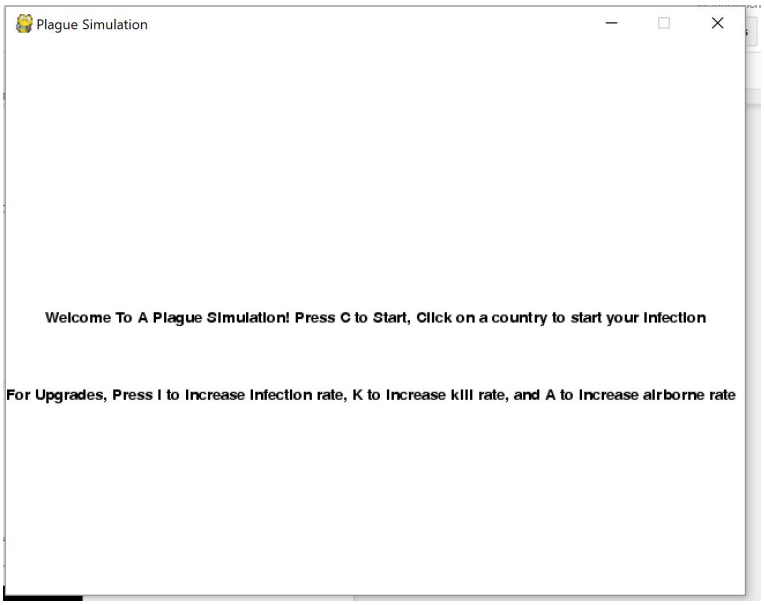
- Our project started from the idea that we wanted to replicate the game called *Plague*. The game was fairly easy to understand and we believed that it was well within our abilities to replicate and within the scope of the class.
- We started out the project by dividing up the code into three different parts; virus, doctor, UI. This proved to be a bad idea because we all write code differently and should have figured out the correct game mechanics first. Were we going to use time ? Or have multiple rounds ?
- These questions could have been solved by coding together instead of working separately.
- Besides a few issues of scheduling meetings times, our group worked well together. We managed to figure out the different components to each function and appropriately divvy up the work.
- Going forward, a huge benefit to this group project would have been a communal brainstorming in the beginning to lay out the inner workings of the program and how each piece was going to interact with the others.

From a process point of view, what went well? What could you improve? Other possible reflection topics: Was your project appropriately scoped? Did you have a good plan for unit testing? How will you use what you learned going forward? What do you wish you knew before you started that would have helped you succeed?

Also discuss your team process in your reflection. How did you plan to divide the work (*e.g.* split by class, always pair program together, *etc.*) and how did it actually happen? Were there any issues that arose while working together, and how did you address them? What would you do differently next time?

Screenshots:

1)



2)