Chapter One:
First Evaluation of Spatiotemporal Epidemiological Modeler (STEM)

Why We Chose STEM

The Spatiotemporal Epidemiological Modeler, or (STEM) for short, provides, as defined on the STEM FAQ page, a graph based spatiotemporal simulation engine. This allows researchers to produce models on infectious diseases. Such models can provide a plethora of information about diseases, which can assist scientists in learning more about them. Such pieces of information could include infection rates, potentially allowing for the ability to mitigate or prevent infection. It additionally, as stated on the About STEM page, allows for models to be made consisting of climate data.
Furthermore, there is extensive documentation surrounding all aspects of the application, from building it, to constructing models. This solidified our choice to select this project.

STEM FAQ
About STEM
STEM Main Page ~ Eclipsepedia (Hyperlinked Resources)

Thoughts on Building the Application

Because there is an abundance of documentation, it was fairly easy to set it up without issue. Following the documentation provided support throughout every step of the process. In fact, there are several YouTube videos available that detail how to get everything working. At no point did we feel that there was a lack of information available to us. While there were dependencies, it was not complex to address them. Regardless, for the scope of our project, we are more interested in specific methods in the application rather than the application as a whole.

STEM Installation Guide

Additional Considerations

While we were first investigating potential methods we might be interested in testing, we noticed that the first class we looked at was designed incorrectly, and would need to be rewritten in order to use it. As we delve deeper into the project, it is clear that we will need to commit resources to ensuring that the methods are in a testable state. Additionally, we need to only use files that are not dependent on the Eclipse framework that the project is interwoven with. So additional time will need to be considered to cherry pick the workable files, before we can even begin the meat of the project.

Final Thoughts

Ultimately, we are incredibly excited to jump into what is uncharted territory for all of us. While we have our work cut out for us, in not only implementing tests on methods, but fixing the methods into a testable state, we look forward to making progress as we resolve issues along the way.