This sprint we planned on doing the stress testing using Jmeter, adding unit tests to the program as well as adding some minor things to the map itself such as a timer for the timed mode and detecting a win. We were able to do all of this in the sprint.

As a team we got together to work on the Jmeter and brainstorm ideas on how we could use that info to improve our program. Other than that, Josh added the things he needed to the map, Oneil modifed the CSS based on the feedback given by the professional, and Sean created the unit test for the users.

Next sprint we plan on documenting the code further, writing our paper over the project and preparing for the final presentation of the project.

As for scalability and performance after running JMeter we have a fairly wide deviation (1700) between our average and median as shown by the graph below. We were able to process 1,670 requests/minute. Which is fairly good considering we lack a dedicated server and were running this on a home machine. However we do believe there is some room for improvement in our program.

For things that could improve the performance of our program. Our program already caches a little bit in the GPS call. So I don’t need to ask for the location every single time we hit the function call, I set a minimum age for the position. This caches the users position for a set amount of time. The way this affected the performance of our program is before when it always asked for a new position every time, the program itself would start to update the view less frequently than desired. With the cache in place it tries to update the view in real time and doesn’t need to constantly fetch a new location which can be fairly expensive. We could always cache the user data we pull from fitbit so we don’t have to constantly go through fitbit.

Parallelism would probably help us out as it would allow us to run the line of code asking for the gps location separately from the code that updates the view. This way we wouldn’t have to check all the logic to draw on the map before we asked for a new location. The same could be said for us with asynchrony since our idea would be to run the gps pooling independently on a separate thread.

