Tony Lu (tianyua1)
Julian Tutuncu-Macias (jtutuncu)

Currently, we are behind. We revised our project proposal to make it more relevant to parallel programming, since our initial proposal to investigate a predictive rendering scheme is more suited to graphics itself than parallelism. We have looked through the source code of Craft and have found opportunities for parallelism aside from rendering. In the multiplayer version, client updates to the world are handled by a single server-side thread, resulting in a potential bottleneck. We've begun drafting schemes to multithread this process. We are also in the process of inserting code to benchmark performance.

Our primary goal has changed to successfully implementing a multithreaded server. We believe this is still possible, even with the goal of achieving nontrivial (> 2x) speedup. At the poster session, we will bring our laptops and connect them to a server. Interested students will be able to play and experience the performance improvements firsthand. We currently have no preliminary results.

We expect a learning curve when writing our multithreading schemes, since we will have to write and use threading libraries in Python as well as C/C++, which we are more familiar with. We are also not completely familiar with the sqlite database that is used to store the world data, so we will need to study databases as well.