15-418 Project Milestone Report

Aimee Feng and Jackson Romero

So far, our progress has generally been slow but steady. Both of us have had other classes with notably large assignments (like Compiler's Lab 5) and Carnival which has created less progress than we initially planned in our schedule. However, we have restructured our schedule since we now have a clearer view of our availability for the remainder of the project. While we haven't been able to follow our original proposed schedule, we haven't run into any major roadblocks and are able to spend significantly more time in the following weeks. Our progress so far includes:

- Sequential implementation of Barnes-Hut
    - We decided to not create a visualization yet for the sequential implementation as it seemed like a relatively high workload for a non-essential portion of our project. Thus, we are pushing this display feature to be implemented towards the end. Instead, we currently only support outputting snapshots to ppm files.
    - We test correctness by comparing results against a brute-force sequential implementation, and check that the results are approximately the same.

- Some work on initial CUDA implementation and some work on our CUDA visualizer.
    - We are still in the process of debugging our CUDA implementation, so it hasn't been fully tested yet. We test correctness by comparing results against a brute-force sequential implementation, and check that the results are approximately the same. Similar to the paper[1] on which we are basing our project off of, we have separated the Barnes-Hut algorithm phases into different kernel launches. We have decided to parallelize along points so that each CUDA thread is responsible for inserting a point into the shared quad-tree during the Barnes-Hut tree construction and each CUDA thread is responsible for calculating the net force for a point.
    - Similarly, with the visualizer, the scaffolding is created but there's no output yet as the implementation isn't complete

[1]https://iss.oden.utexas.edu/Publications/Papers/burtscher11.pdf

We still believe that we'll be able to meet our deliverables, although the "nice to haves" are looking unlikely. We are currently a few days behind the original proposed schedule. Given that the amount of time we are able to dedicate towards the project during the second half is significantly more than we were able to in the first half, we are confident

in producing our deliverables. Implementing the sequential version of Barnes-Hut gave us a better understanding of implementation details, so our lessons learned from debugging the sequential version has helped us in debugging our parallel implementation. The “nice to have” features involving CPU-only implementations seem difficult to achieve given time constraints, but I believe we will get a better feel for timing once we start our optimization implementations.

Our updated goals for the poster session are:

- Sequential CPU-only implementation of Barnes-Hut algorithm that allows for a tunable parameter θ for approximation accuracy
- CUDA implementation of Barnes-Hut that allows for a tunable parameter θ for approximation accuracy
- Speedup graphs for CUDA vs. CPU-only implementations with varying number of points (5000, 50,000, 500,000, 5,000,000, and 50,000,000) at fixed θ
- Accuracy graphs for the CUDA implementation across varying values of θ (0.05, 0.5, 1, 5, 50, and 500), where accuracy is measured by the sum of the squared distance between approximated points and ‘accurate’ points calculated at θ = 0. These measurements will be taken across varying problem sizes (5000, 50,000, 500,000, 5,000,000, and 50,000,000 points).
- Visualization of bodies and their movement across timesteps from the Barnes-Hut algorithm for both CPU-only and CUDA implementations

Of the extension goals, extra parameters and importing initial positioning data are the most achievable and will be what we prioritize if we have the time.

Our biggest concern at this point is that we are unsure about the difficulty for optimizing the CUDA implementation as we have yet to start optimizing anything. Right now we’re working under the assumption that it will be roughly similar in difficulty to the circle rendering assignment. Based on our progress with the sequential implementation, this feels like a reasonable assumption, and we have extra buffer time in our schedules to allot to this project if needed. But, on the whole, we don’t have any outstanding unknowns and hope finishing this project is a matter of just doing the coding.

Note: We have revised the schedule on our main proposal doc to reflect our progress so far and assign tasks to specific team members. That updated version is available at 418.jtromero.com.