Project Report – Particle Simulation

Mark Randles
CS490
Dr. Hassan Rajaei

1. Introduction

The goal of this project was to extend work done during a previous project session. The actual problem, as defined had to do with “simulating” the motion of particles inside of a complex system. The major focus was implementing a algorithm that would limit the complexity of the math intensive part of the computation.

2. Problems

There were a variety of problems with completing this project in it's current form. Some can be contributed to error on my (Mark's) part and some to the complexity of both the algorithm and the problem.

Overall there were three major problems with the project. The largest problem was with the actual complexity of the problem as originally defined. One of my goals with this project was to explore the algorithm idea I had, and needed a simple problem to use as a basis for implementation. However, the particle problem proved to complex to implement. The overriding complexity was with the use of physics to simulate the particles. My experience with physics is limited and never before have I implemented a physics simulation or pseudo-simulation engine. These two factors combined to bog down the implementation of the arbitrary code, that is mostly unrelated to the algorithm.

Another problem was a lack of time on my part. Both class and social obligations and some procrastination combined to slow down the speed at which project work could progress. This tended to stall any productive work that got done.

The last major problem was the complexity of the study algorithm. The study algorithm not only depended on processing and sorting lists of items before the main computation. This presented a hassle as the program wasn't dealing with simple datatypes in a language that excels at using simple datatypes.

3. Solutions

The first solution to some of the problems with the project would be to dedicate more time to the project. However, this becomes increasingly harder as semesters progress, as motivation begins to

wane and time needed increases. This would be the easiest solution to most of the problems, however it would require more time, which is a precious commodity.

Another solution, which has been begun, is to simplify the problem being implemented. This can be a bit difficult as the problem must be able to be processed in the same manner as the original particle problem. However, this is easy, as the problem is pretty common in the abstract. Essentially the algorithm takes a algorithm that needs O(N^2) time to operate and reduces the N based on a criteria that is dependent on the problem. For the original particle problem is was if the ideal trajectories would intersect.

A new problem was defined that simplified the actual complexity of the calculations that the algorithm needed to do, while still retaining the necessary design of problem, so the new algorithm could be applied. However, at the time of this writing there are still some issues to be worked out with the implementation of the new, better problem, so concrete results can't be gathered.

4. Conclusions

Over all this problem has stalled, but could still be in a workable state, if more time and energy could be thrown at it. As of the time of writing development has stalled because of a lack of debugging facilities and clumsy development environments.

It should also be said, that there could be negligible speedup from use of this new method of algorithm, as it is highly reliant on paring down the N elements that must go through the O(N^2) part of the algorithm. If few elements can be excluded then the overhead would kill any speedup, and could lead to a negative speedup.

This project is still viable in my mind, but not as a study, rather a toy or diversion from other work. There is promise in the ideas, but in the current situation the time required to complete is preventative.