Drake Seifert

CS475 Project 7

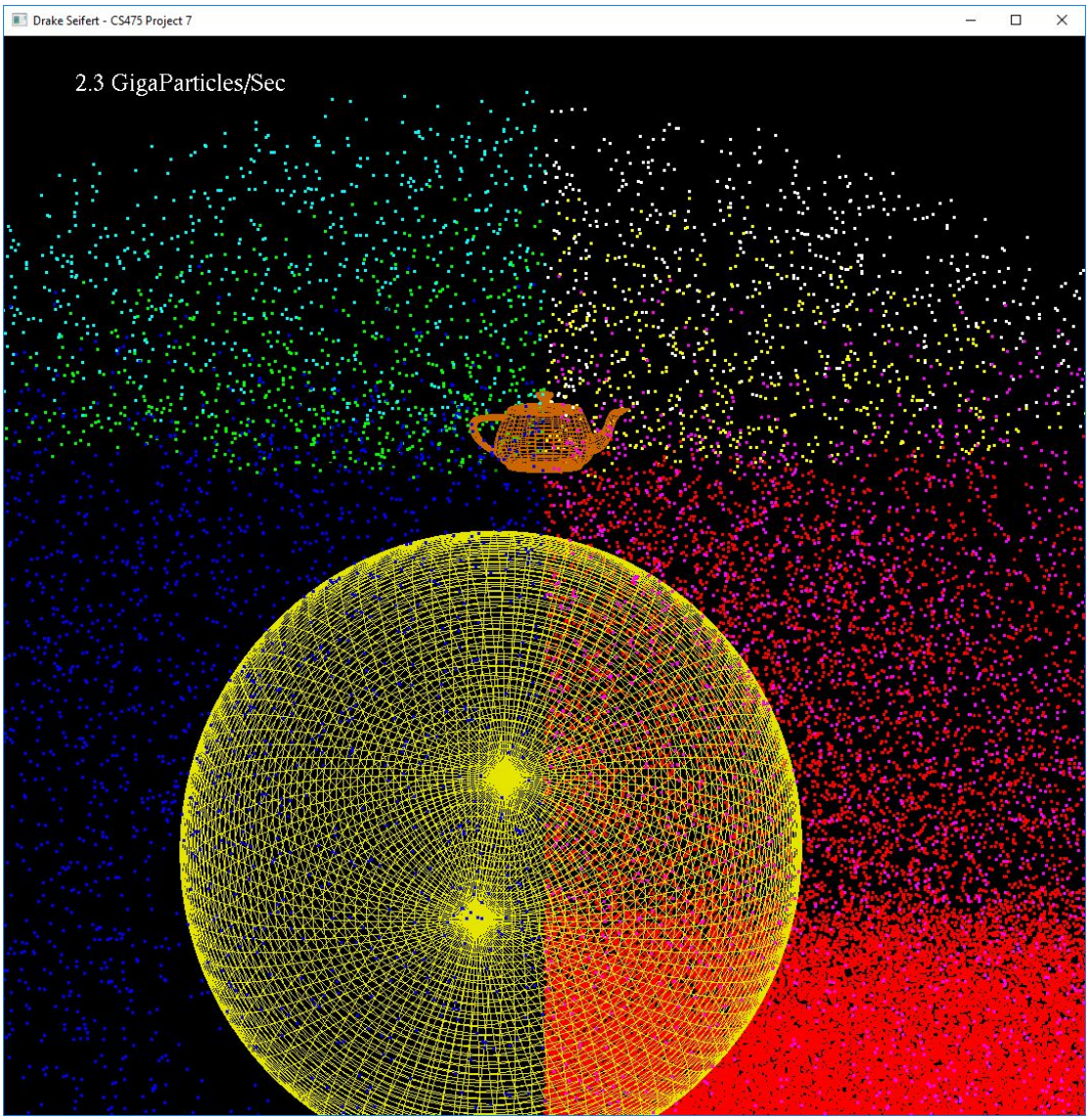
What machine you ran this on

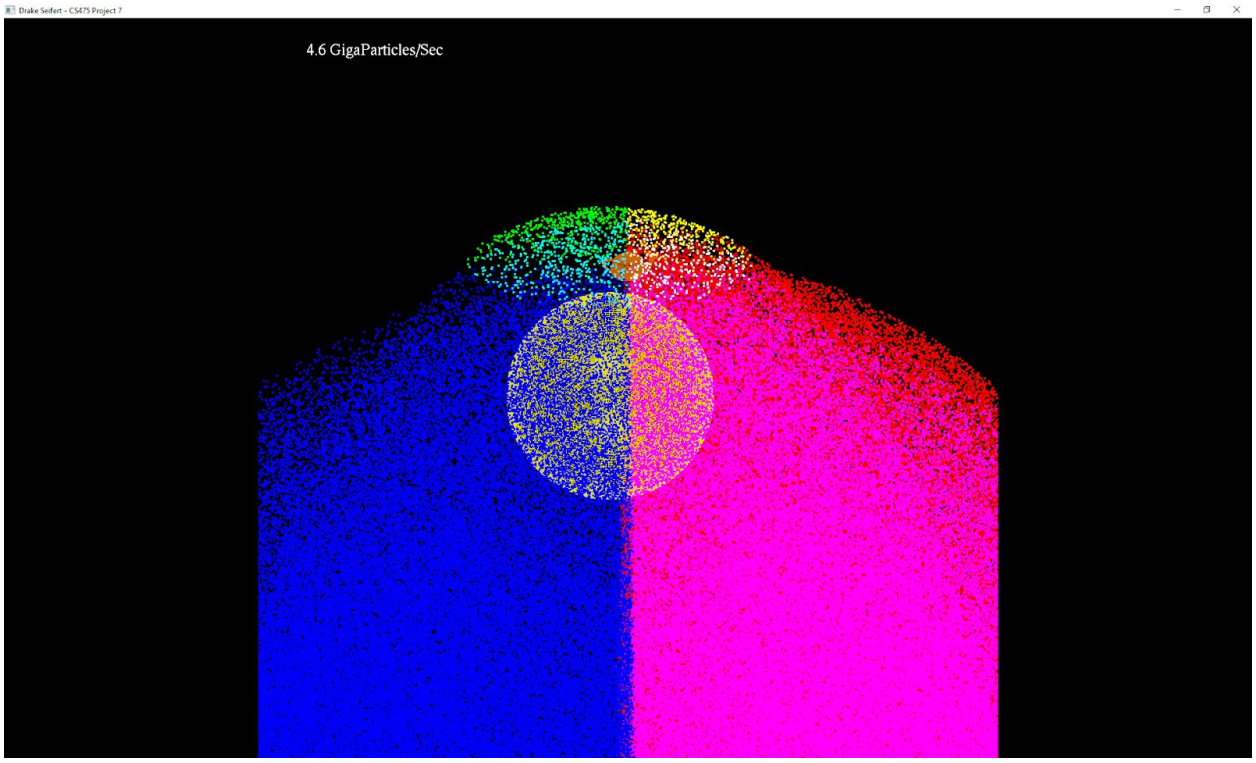
All tests were ran on the CGEL computers.

What dynamic thing did you do with the particle colors

I chose to change the particle color based off of position.

Include at least one screen capture image of your project in action



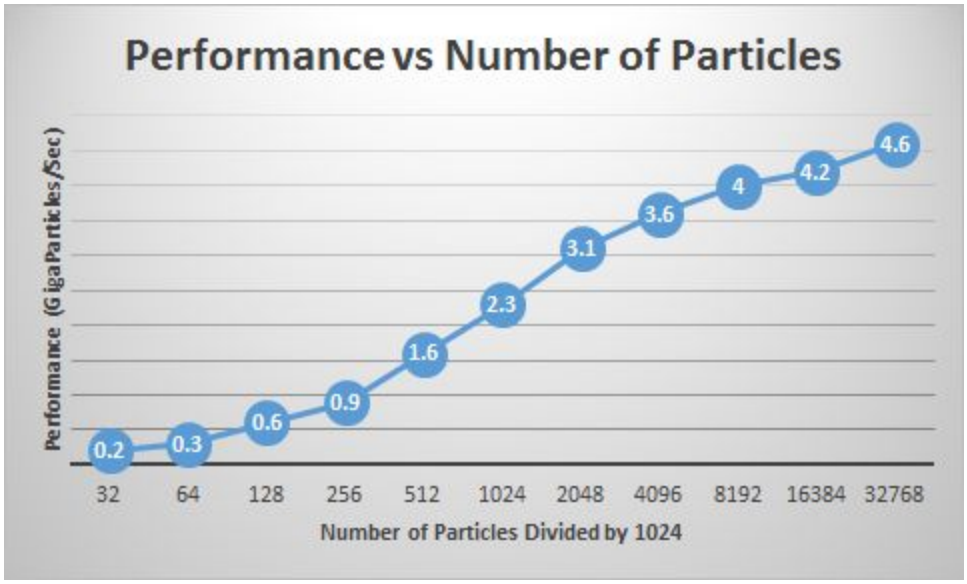


Show the table

Performance vs Number of Particles Divided by 1024 (for a local size of 32)

	32	64	128	256	512	1024	2048	4096	8192	16384	32768
32	0.2	0.3	0.6	0.9	1.6	2.3	3.1	3.6	4	4.2	4.6

Show the Graph



What patterns are you seeing in the performance curve?

Large performance gains occur for a larger number of particles. Tests were performed up until about the maximum integer value supported on the system being tested on. This pattern was consistently true for all tests.

Why do you think the patterns look this way?

The more particles there are, the more computations that must be performed. Given a fast enough system, this means that the performance can begin to “max out” its resources. When dealing with a smaller number of particles, all other particle calculations must be finished before continuing the next time step which severely hinders performance as the time step is set at a fixed rate.

What does that mean for the proper use of GPU parallel computing?

The proper use of GPU computing can be an incredibly powerful tool. As seen in this assignment, when used properly GPU parallel computing can begin to increase in performance up until there is an integer overflow error. That is pretty cool. This also puts a spotlight on time steps in the case of graphics and recognizing that a smaller number of particles can be computed at a lesser time step (if desired) with little to no performance loss. I was very surprised to see that such a large number of particles (1024 * 32768) managed to be calculated every tenth of a second without fail, but in hindsight video games manage to pull of tasks such as this all the time.