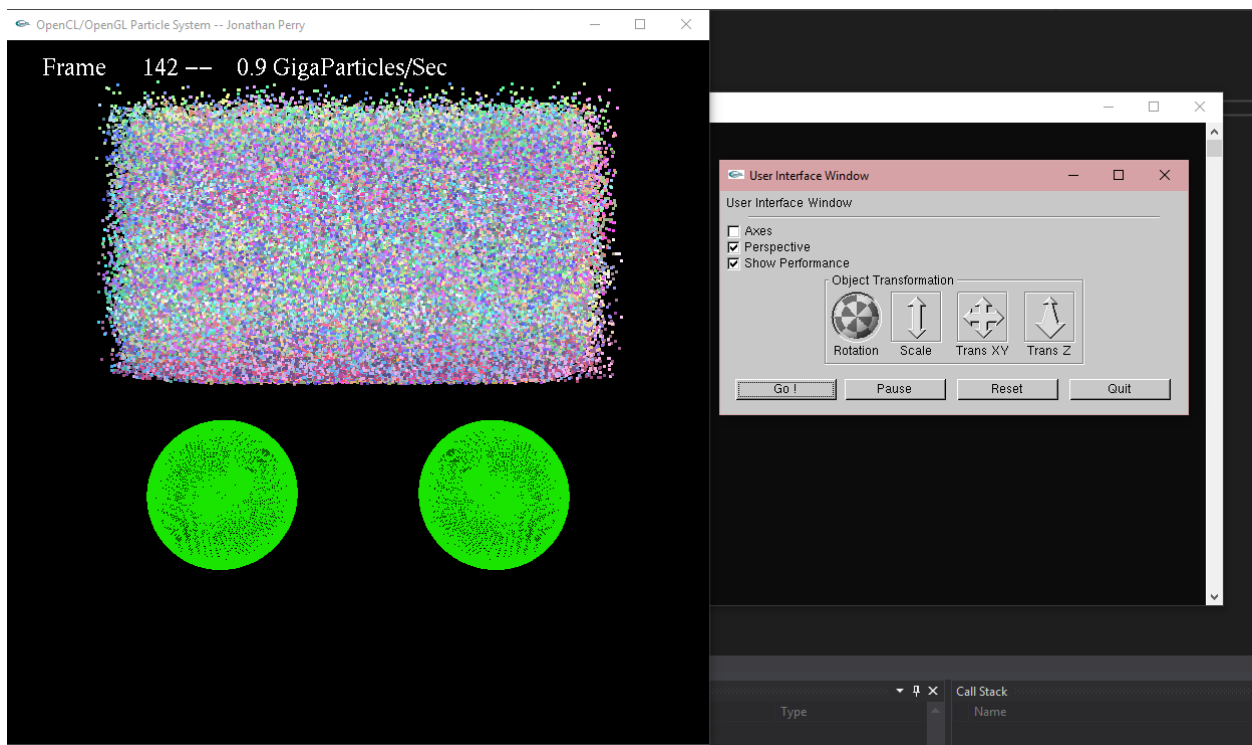


PROJECT 7A

1. I ran project 7a on my home computer, which is equipped with a gtx 770 4gb gddr5, an intel 4770k, 12gb of DDR3 RAM, and windows 10 installed on an SSD.
2. For the particle colors, what I did was have them change to red, green, or blue when they bounce off one of the two spheres. The following code worked nicely to make the colors change “randomly”, as accessing the RAND() function was not possible.

```
float r = gid % 10;  
  
float g = gid % 100 / r;  
  
float b = gid % 1000 / r / g;  
  
if(r > g){  
    c = (color)(r, 0., 0., 1.);  
}  
else if(g > b){  
    c = (color)(0., g, 0., 1.);  
}  
else{  
    c = (color)(0., 0., b, 1.);  
}  
}
```

3. Picture of project in action.



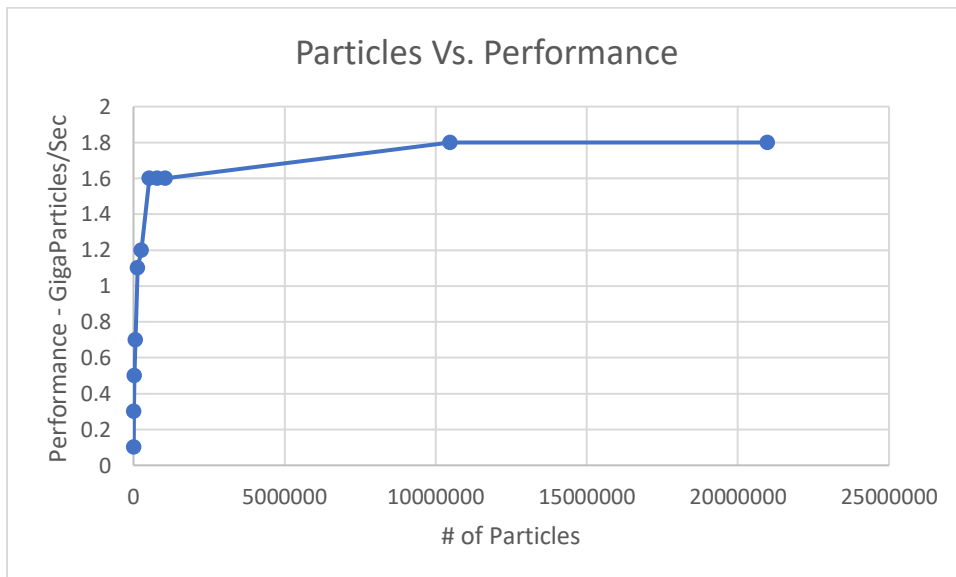
4. Table and Graphs

PROJECT 7A

- Table

| Input Size | Performance - GigaParticles/Sec |
|------------|---------------------------------|
| 8192 | 0.1 |
| 16384 | 0.3 |
| 32768 | 0.5 |
| 65536 | 0.7 |
| 131072 | 1.1 |
| 262144 | 1.2 |
| 524288 | 1.6 |
| 786432 | 1.6 |
| 1048576 | 1.6 |
| 10485760 | 1.8 |
| 20971520 | 1.8 |

- Graph



5. The Patterns I'm seeing is that performance increases heavily from using 8192 particles up to using 262144 particles in the program. After 262144, the number of particles doesn't have a direct effect on performance, as the performance curve flattens out.
6. I believe that the pattern looks this way because we have fully utilized every compute unit on the GPU for this OpenCL/OpenGL computation, performance cannot increase without the availability of more compute units to farm out each piece to different threads on the GPU.
7. This means that using GPU parallel computing for a large input size will reduce your overall execution time of your program versus using a CPU. For example, a CPU paired with OpenMP may not produce results anywhere close to using a GPU for parallel programming,

PROJECT 7A

as the CPU may split the program up to 8 or more threads, but the GPU has over a thousand possible threads active, where each thread executes the same kernel and they're all working on a small piece of the overall problem.