James Wilson (wilsjame)
CS 475 Spring ‘15
Project #7A

OpenCL / OpenGL Particle System

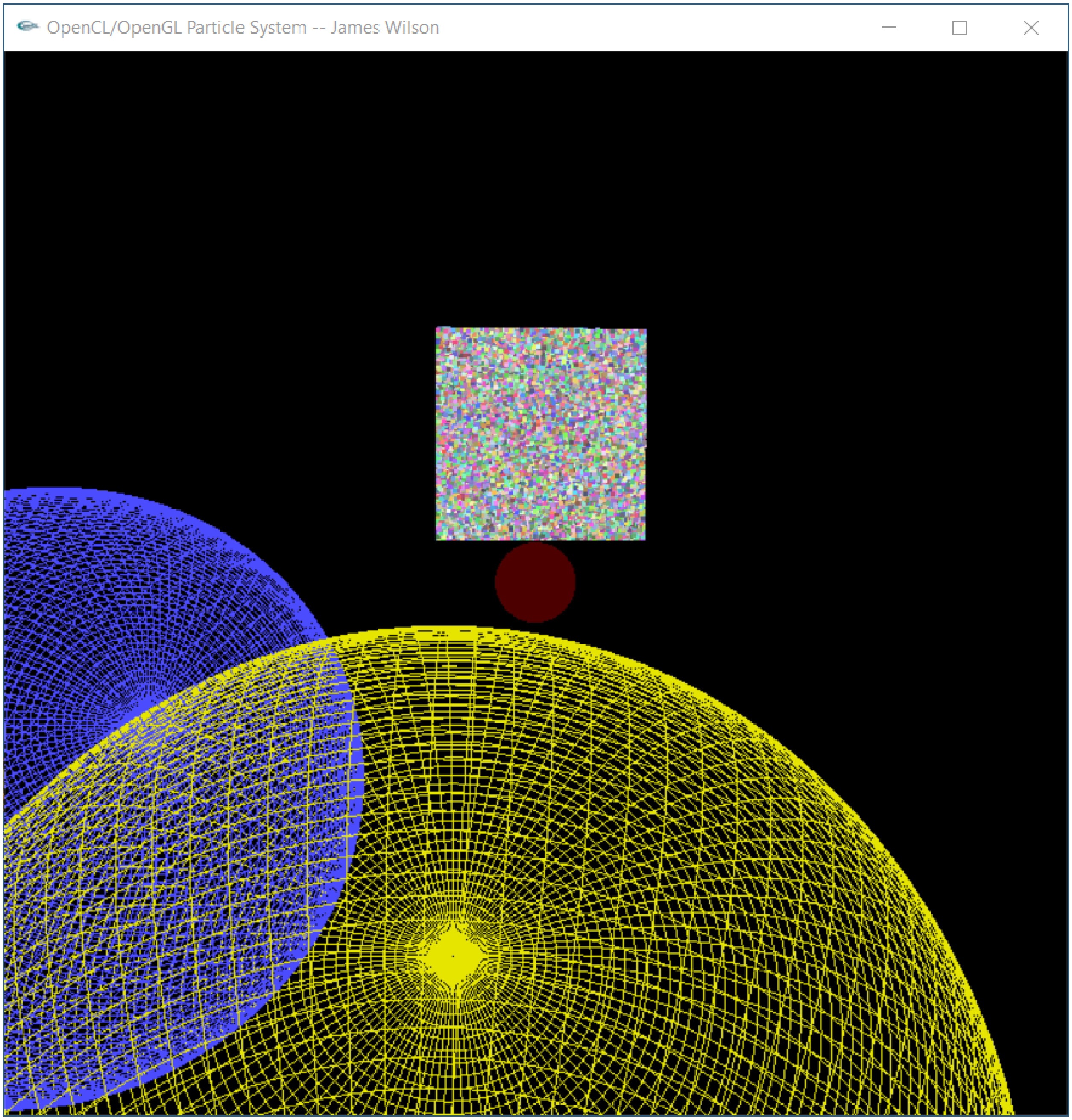
1. What machine you ran this on

   - OS Name Microsoft Windows 10 Enterprise
   - Version 10.0.17134 Build 17134
   - System Manufacturer Dell Inc.
   - System Model Inspiron 7577
   - System Type x64-based PC
   - Processor Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz, 2808 Mhz, 4 Core(s), 8 Logical Processor(s)
   - Installed Physical Memory (RAM) 16.0 GB
   - GPU Nvidia GeForce GTX 1060 with Max-Q Design

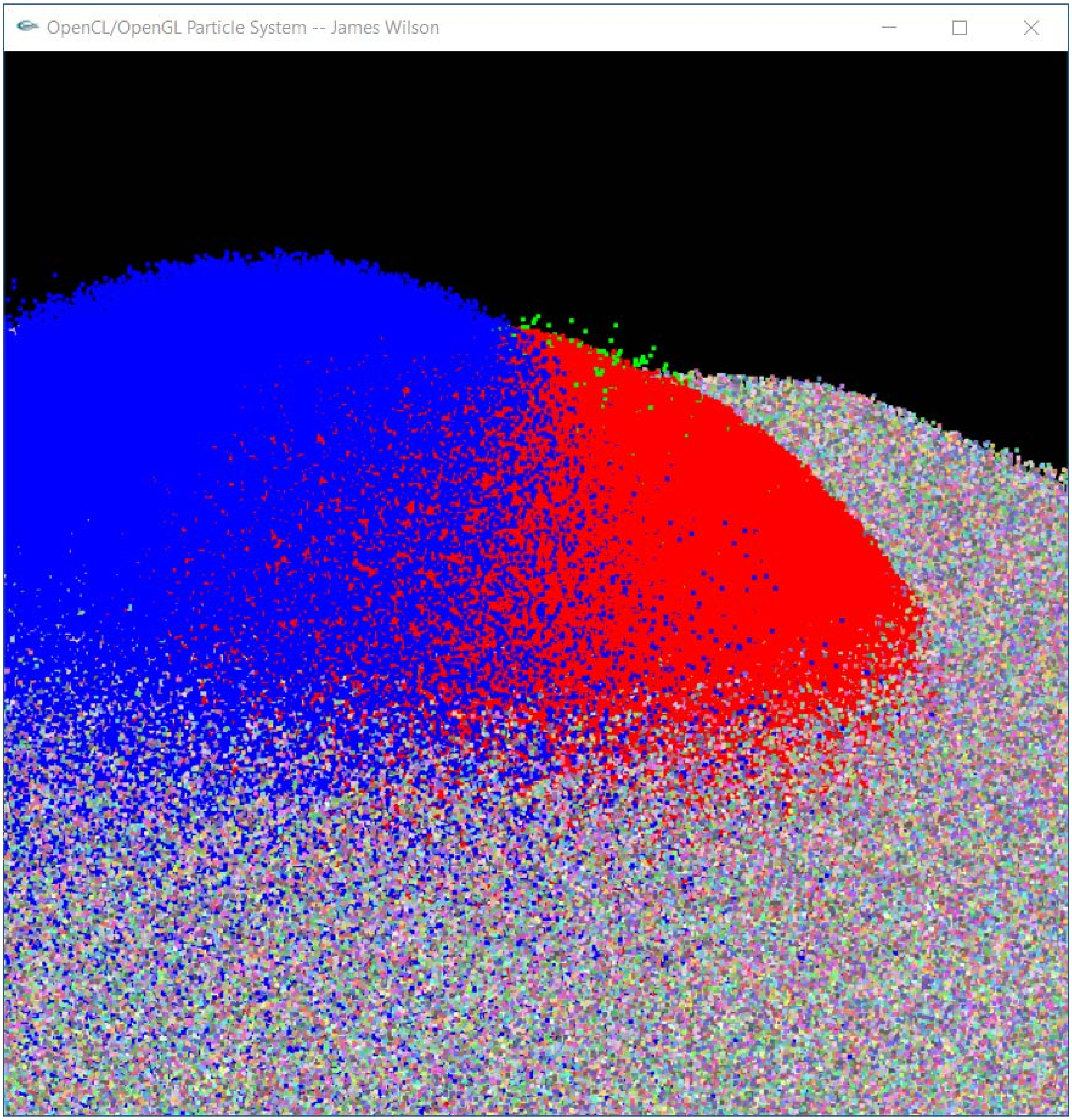
2. What dynamic thing did you do with the particle colors

   The particle’s color change to either red, blue, or green depending on which bumper they hit.

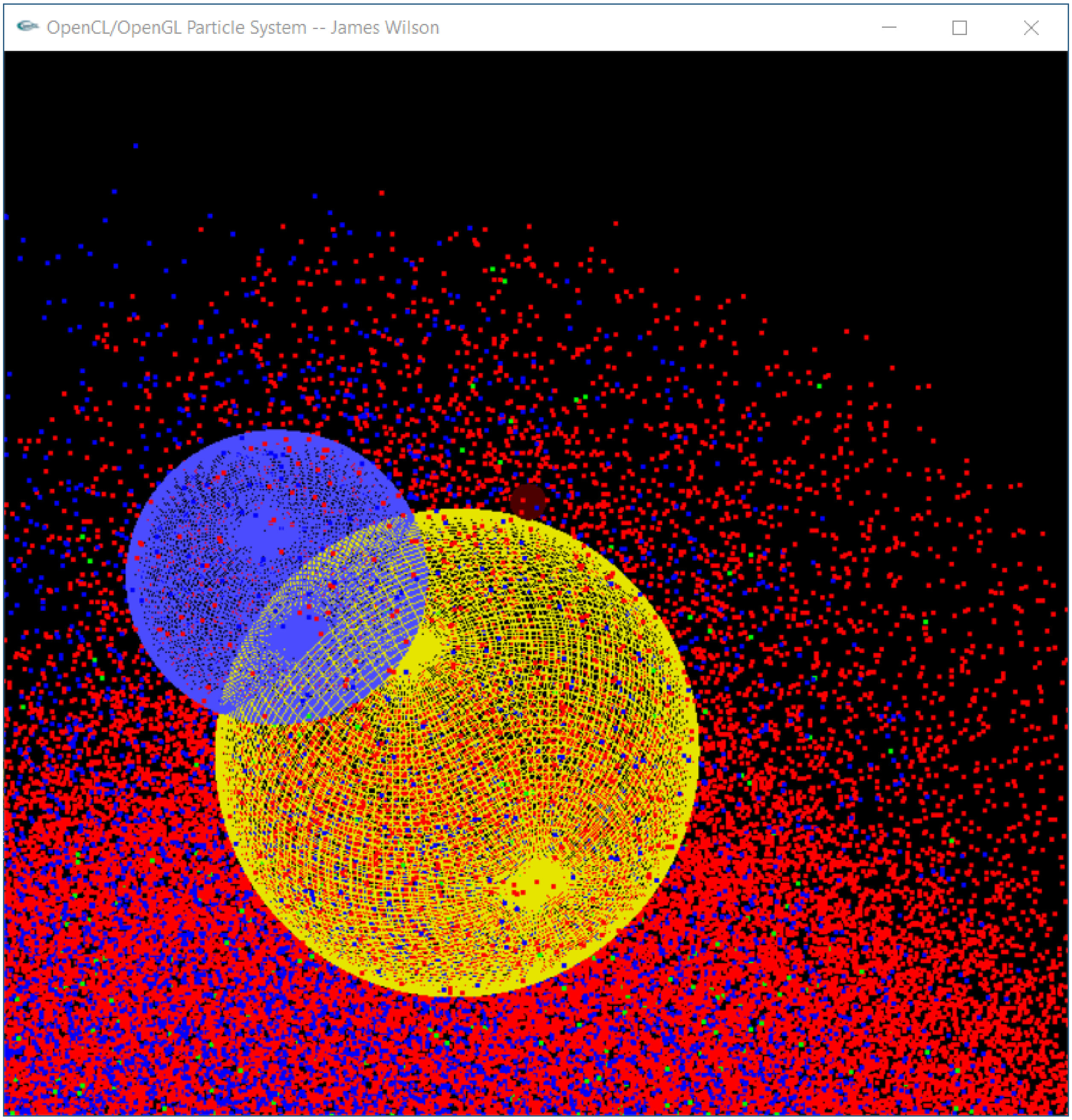
3. Include at least one screen capture image of your project in action

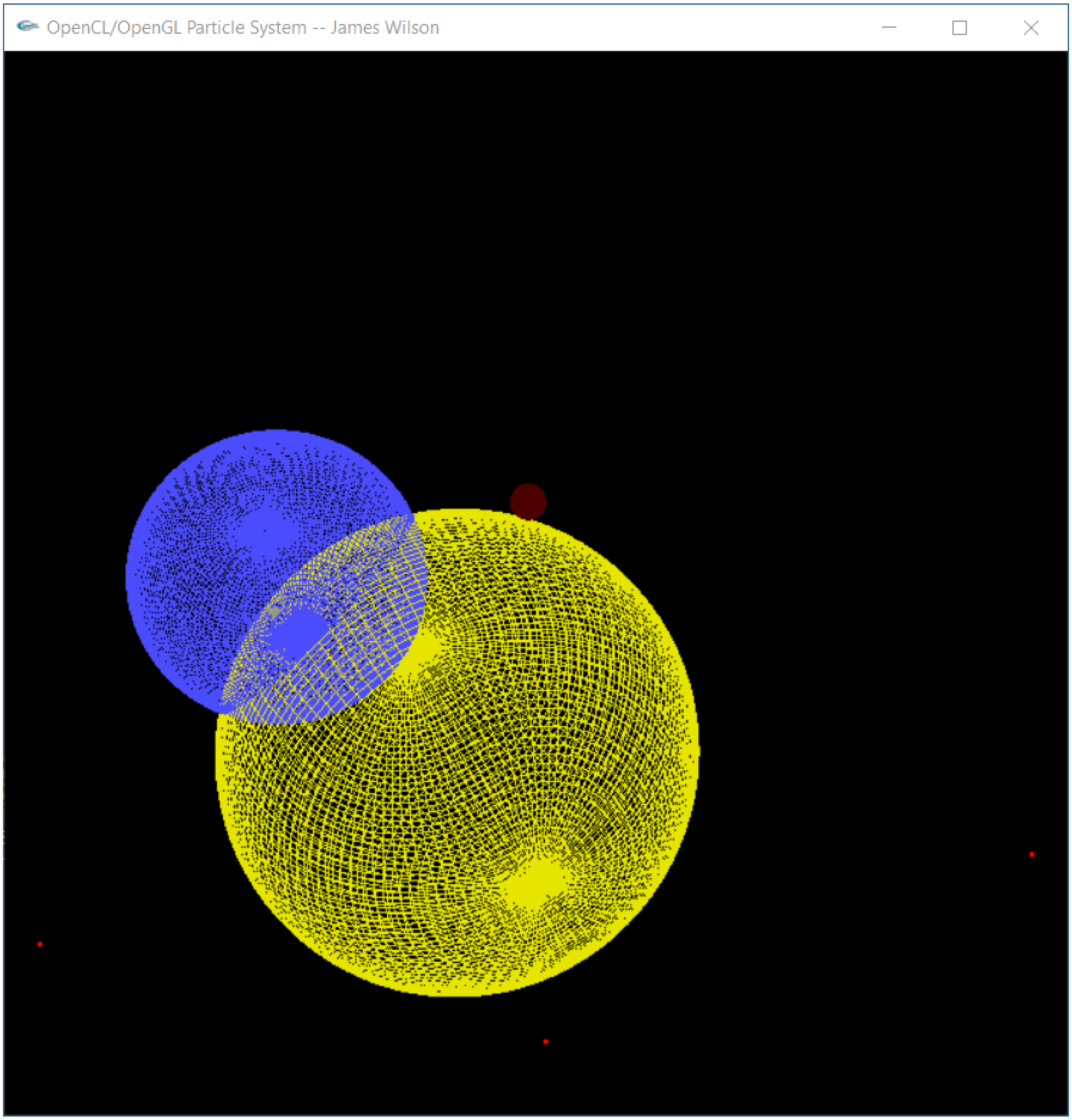


OpenCL/OpenGL Particle System -- James Wilson



OpenCL/OpenGL Particle System -- James Wilson

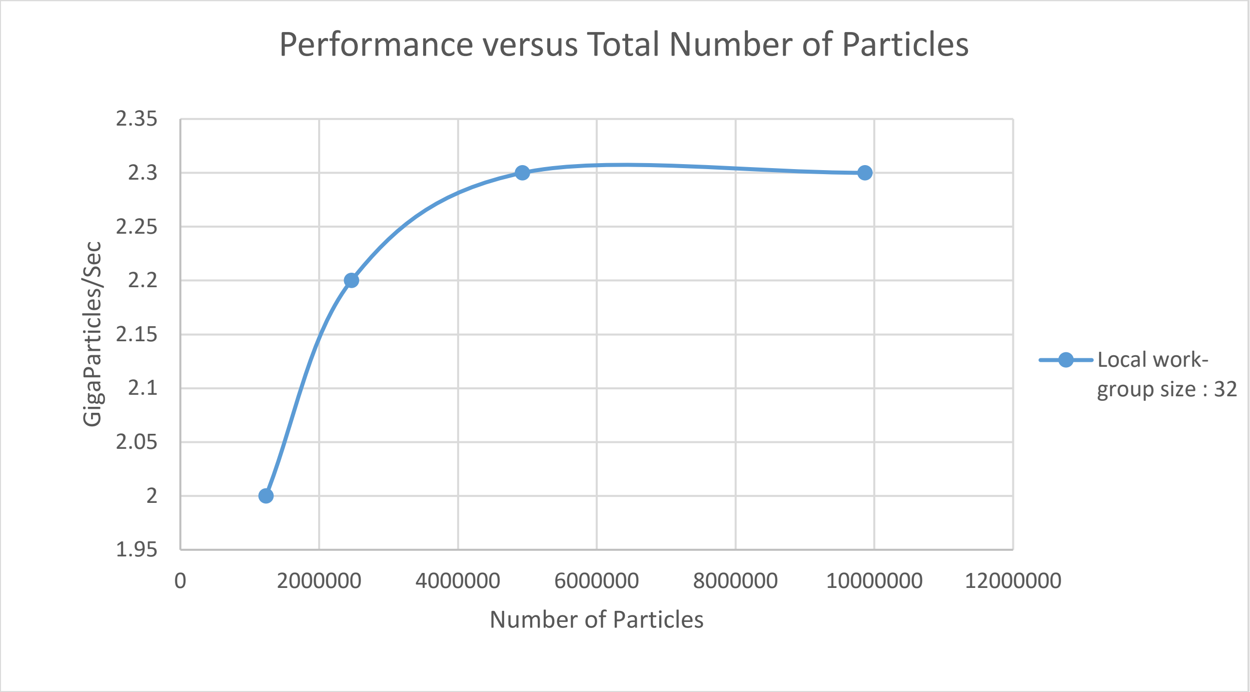

OpenCL/OpenGL Particle System -- James Wilson

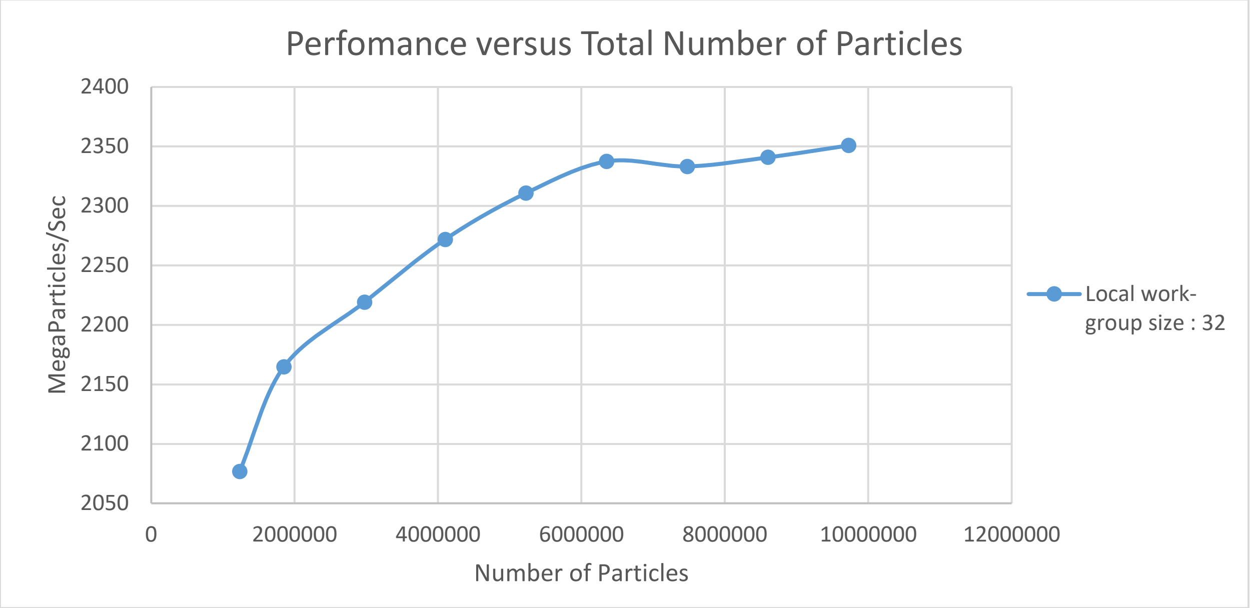


4. Show the table and graph

		Local work-group size : 32	
Global work size (Total Particles)	1232896	2	GigaParticles/Sec
	2465792	2.2	
	4931584	2.3	
	9863168	2.3	
	19726336	Crash	

		Local work-group size : 32	
Global work size (Total Particles)	1232896	2076.8	MegaParticles/Sec
	1849344	2164.7	
	2974344	2219.1	
	4099344	2271.9	
	5224344	2310.7	
	6349344	2337.4	
	7474344	2333.2	
	8599344	2340.8	
	9724344	2350.8	
	10849344	Crash	





5. What patterns are you seeing in the performance curve?

   The performance increases as number of particles increase until a horizontal asymptote is reached at high number of particle values. Eventually, the program crashes.

6. Why do you think the patterns look this way?

   As the number of particles increases the GPU is able to take more advantage of the global work-group size and in result performance increases. Eventually the return in performance gains reaches zero because the GPU has reached its max capacity of computing for this program (see Amdahl's Law!).

   The quick increase of performance from small numbers of particles to large numbers is because at low number of particles the overhead from the OpenCL/GL interaction is competing with the performance gains from parallel computing. It is at the higher values of particles that the gains in performance overcome this overhead. The program eventually crashes because my computer cannot handle the extremely high number of particles.

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

   In order to take advantage of the full potential of GPU parallel computing performance gains, one should use a large global work group size. To find this optimal value, one could test various global work group size inputs while keeps local work group sizes constant. If one uses value too high there is a risk of crashing the program.