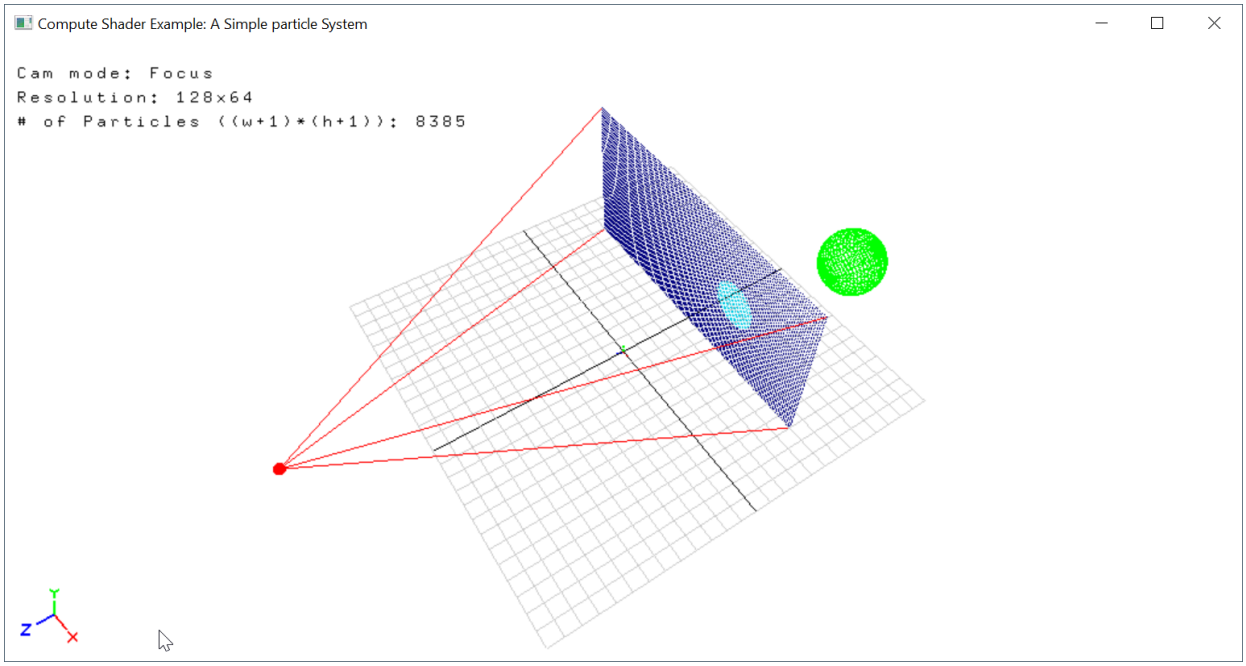
Assignment 4: Computer Shader Programming

1. Due Date

Assignment 4 is due on **Wednesday, 04/14, 11:59pm.**

2. Introduction

In this project, you are required to create a plane of particles based on the width and height of the resolution. A point is given to create rays from this point to the particles. In the scene there should be a wireframe sphere behind the plane of particles. If a ray through the particle intersects with the sphere, the particle should be rendered with cyan color; otherwise, it should be rendered with dark blue color. An executable of this assignment (*.exe*) created by the instructor has been uploaded to Mycourses. Please run it and get a feel of the work you should deliver. You are required to implement the ray-particle intersection and determine the color in a compute shader, and render these particles in the vertex and fragment shaders. You can start the assignment using the example code in Mycourses (*ComputeShader_ParticleSystem.zip*).



3. Requirements

1) **(25 pts)** Use the width and height of the resolution, and the min and max sizes to generate particles in the plane that is parallel to the xy plane. Initially, the resolution (W x H) should be set to 64 x 32. The min point is (-10.0, 0.0, -5.0), and the max point is (10.0, 10.0, -5.0). The generated particles should be evenly spaced in the plane based on the size from the min and max points. Note that, each row should have 65 (W+1) particles and each column should have 33 (H+1) particles. The initial color of particles is dark blue (0.0, 0.0, 0.5). Your code should store the coordinates and colors of these particles in arrays, and then map them to the shader storage buffer objects (SSBOs) in the compute shader.

2) **(15 pts)** Create a point, which acts as the "origin" of the rays to the particles. This point is at (0.0, 5.0, 20.0). Draw a red solid sphere (e.g., *glutSolidSphere(),* radius=0.3, color = (1.0, 0.0, 0.0)) to represent the location of this point. Also draw four red lines from this point to the four corner particles of the plane. This point needs to be sent to the compute shader.

3) **(15 pts)** Create a green wireframe sphere. The center of this sphere is (0.0, 5.0, -8.0), its radius is 2.0, and the color is (0.0, 1.0, 0.0). The center and radius need to be sent to the compute shader. Use the keyboard to move this sphere. For example, pressing 'w' and 's' moves it along the z axis, pressing 'a' and 'd' moves it along the x axis, and pressing 'u' and 'j' moves it along the y axis.

4) **(25 pts)** Implement the ray-sphere intersection in the compute shader. Refer to this Wikipedia page: https://en.wikipedia.org/wiki/Line-sphere_intersection, to find out the equation for the intersection test. If a ray intersects with the green sphere, the corresponding particle should be rendered with cyan color (0.0, 0.8, 0.9); otherwise, the color remains as dark blue.

5) **(20pts)** Use the '+' and '-' on the keyboard to change the resolution. The '+' should double the resolution (2W x 2H), and the '-' should reduce the resolution by half (W/2 x H/2). Set the maximum resolution to 8192 x 4096 and the minimum resolution to 16 x 8, so that the resolution cannot go beyond the maximum and minimum. Note that when the resolution is changed, the particles should be regenerated and remapped to the SSBOs. After finishing the implementation, you will experience the power of compute shader – which can process and render tens of millions of particles in real-time.

4. What to Submit

(Make sure your code can be compiled and run in Visual Studio.)
Submit the following items to *Mycoures*:

- A document explaining how to use your program.
- A zip file containing all source files (*.h* and *.cpp* files).