**Objective:**

To implement particle Engine using Compute Shader.

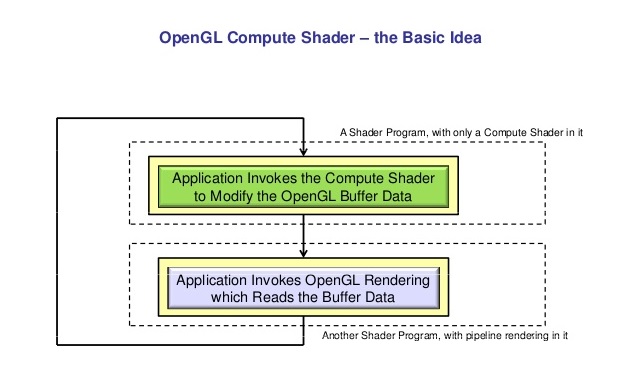
**A brief write-up about the program:**

**Partical System :**

A particle system is a technique in computer graphics that uses the large number of very small sprites to simulate certain kind of phenomena, which are very hard to reproduce with conventional rendering techniques.

**Compute Shader :**

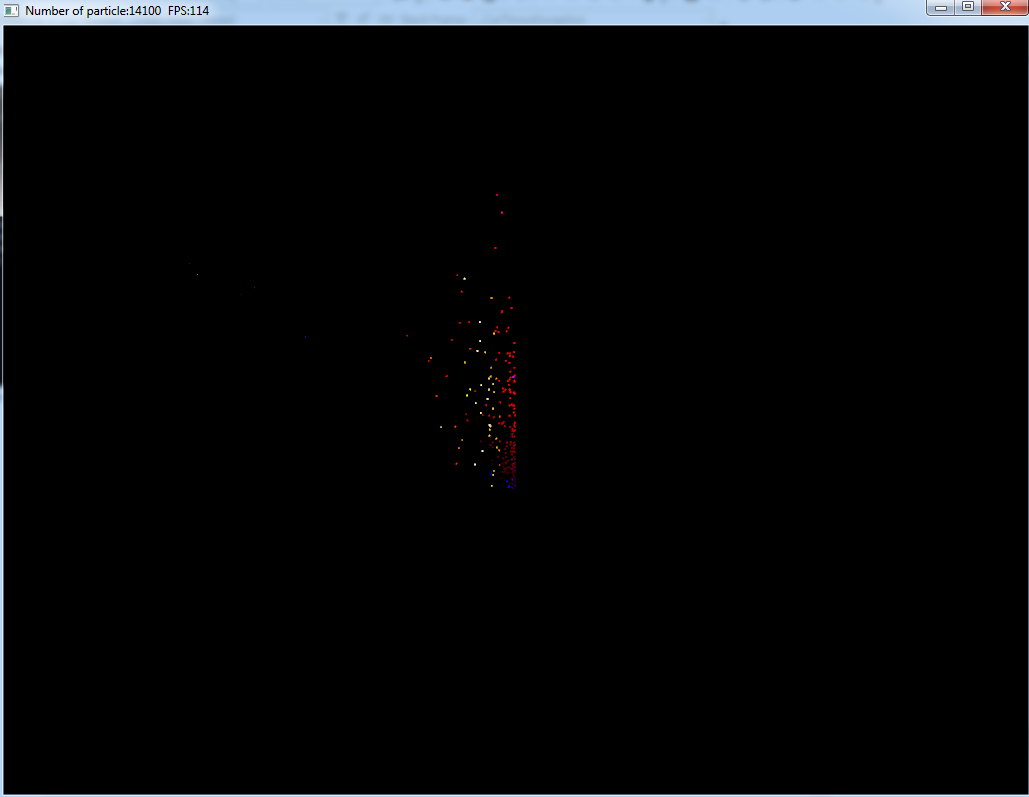
* Compute shaders work very differently. The "space" that a compute shader operates on is largely abstract; it is up to each compute shader to decide what the space means.
* The number of compute shader executions is defined by the function used to execute the compute operation.
* Most important of all, compute shaders have no user-defined inputs and no outputs at all.

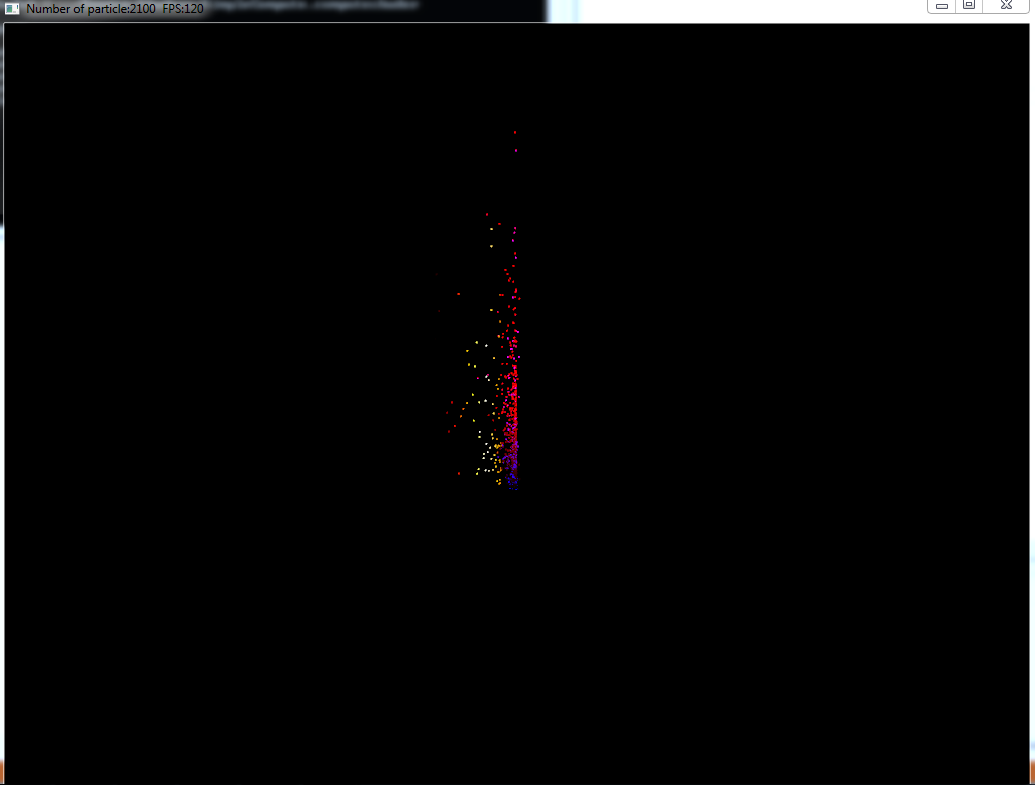
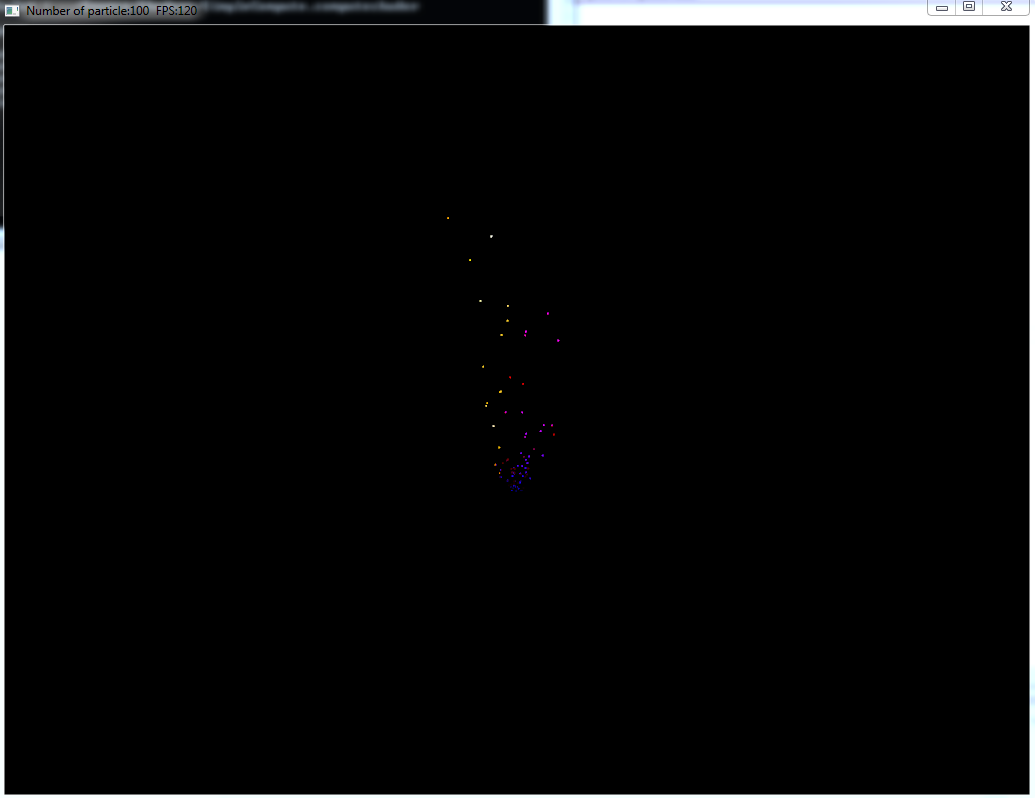


* The important is that while the other shaders have a fixed execution order, compute shaders can essentially alter any data anywhere.
* Shader objects within a program object are implicitly pipelined after another, and a program object is "ready to go" as it is.
* Compute shader generates two shader storage buffer objects (SSBOs).

One for particle positions and one for particle velocities, and the compute shader then accesses these SSBOs to calculate particle velocities depending on an attraction point and accordingly updates the particle positions. So the whole particle system gets calculated on the GPU, which should be a lot faster than doing it on a CPU.

**Screenshots:**





**Usage Instructions:**

* Open the executable file of the assignment from the bin folder.
* Press UP to Increase the Number of particles.
* Press Down to Decrease the Number of particles.
* Press NUM+1,NUM+2,NUM+3,NUM+4 to change the texture.
* Press Esc to quit.

**References:**

1. Compute Shader Tutorial

(https http://www.saschawillems.de/)

1. Partical System and instancing

(http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/particles-instancing/).