Particle System in OpenGL

Degree programme : BSc in Computer Science | Specialisation : Computer Perception and Virtual Reality
Thesis advisor : Prof. Marcus Hudritsch
Expert : Dr. Harald Studer (Optimo Medical AG)

Particle systems are tools to make scenes livelier without being expensive in terms of resources. They are a collection of particles that move in the world and change their size, color, transparency, and direction. This computer graphics technique can create various visual effects ranging from fire, smoke, and water to the sun.

Introduction

This bachelor thesis is the continuity of my project two, called „low-level particle systems" I have implemented a basic particle system in OpenGL working only on GPU. Particle systems already exist in many computer graphics tools, but they are all different. In my case, I have developed my own because I want to be independent of other solutions. I want to use as few resources as possible and be deployable to a maximum of platforms.

Goals

For this project, I had three main goals :

- Integrate my particle system in SLProject (A framework of the BFH).
- Create a user interface.
- Add features, such as size over life, gravity, etc.

The last goal was the one that took more time because there, the amount of work possible is enormous, and man can always find a new idea, a new feature for a particle system.

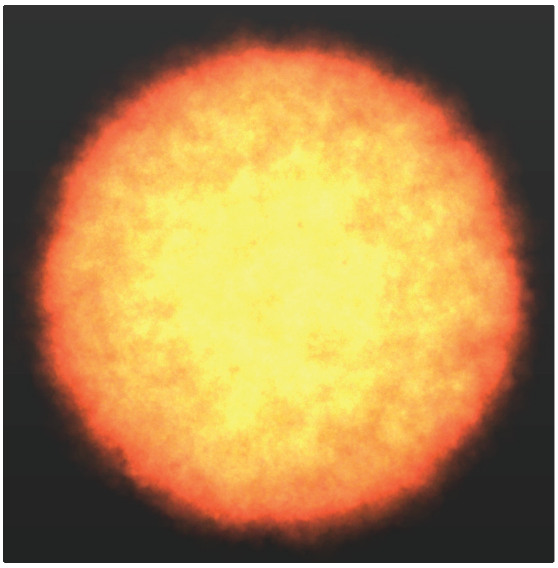
Implementation

The idea behind the functioning of a particle system is the following. There are two elements, a particle emitter and particles. The particles are spawned from the particle emitter and live for a specific time. Therefore, they are born and will live until they die. When the particles pass, they will be recycled and thus be emitted again from the particle emitter. The application draws the particles in each frame, but before it paints, it needs to update them. The CPU can do this task, but in my case, because I want to use fewer resources, the application updates them on GPU.
To update them on GPU, I use a transform feedback technique that allows me to write back information on a buffer to bypass the call between the CPU and GPU. The behavior of particles changes according to the features enabled. They can change the updating and the drawing or how the system generates the particle. These features are, for example, size over life, alpha over life, acceleration, gravity, etc. The more features, the more visual effects. In this project, I implemented many features, including the core ones. Because I want the particle system to use as few resources as possible, the shaders and the data running on the GPU are generated dynamically according to the currently enabled features, which saves memory and computation cost.

Conclusion

I managed to recreate physics effects such as fire, smoke, dust storm, and even the sun with my particle system. I fulfilled the objective of this project, but as said before, the features that one can add are tremendous. Some novelties are developed, like the Unreal Engine 5, which pushes for innovation. Therefore, this particle system can always be continued and enhanced.

Marc Fabien Affolter
marc.fabien.affolter@gmail.com



Sun created with my particle system in SLProject