Particle System: Fire Simulation

Sujata Tamang

***Abstract*—This report describes the project particle system for fire simulation that was implemented as part of Assignment04, project.**

I. Introduction

Particle System is a widely used technique in various real world applications such as game engine, physics simulation, movies etc. They are very useful to simulate phenomena such as fire, explosion, smoke or cloths.Particle system is comprised of one or more particles that may or may not interact with one another. In general we can describe the life cycle of Particle in three phases : Generation, Particle Dynamics, Extinction. [**?**] Particles may be generated randomly in a predetermined or a random location. They change over time. Depending on the application, particles may change color, indicate cool off, disappear etc. Eventually all particles die. The spawn rate, attributes(postion, color) is defined and updated in simulation phase. After the update is complete, each particle is rendered.

II. Approach

The approach used here for particle system is Instancing [**?**] which is creating a base mesh that has many instances of it. For this project, the base mesh is a quad of two triangles. This can be achieved by creating several buffers that describes the base mesh and the instances. In our case, Vertex Buffer Object is created to hold four vertices of particles which is shared by all particles. Also separate VBO is created for holding position, and color of particles. Attribute buffers are created for vertices, position of particle center, particle color. To draw the particles, glDrawArraysInstanced is used.

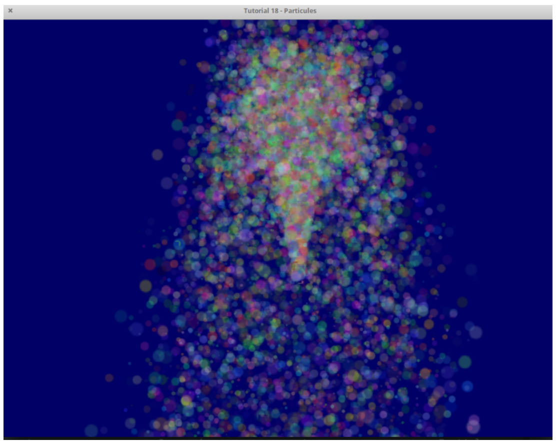
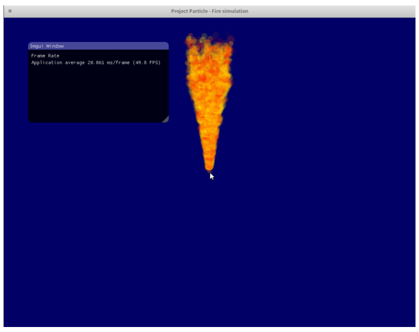


Fig. 1. Particles system

Particles born and die at a high rate. so, we need to constantly create new particles and delete old particles. We create particle container the size of MaxParticles (which is a constant that we set to 100000 initially). The Particle container is of type particle that has speed , position, life , color(r,g,b,a) , size , angle, weight for particles.We can access the particles from this container by its index. Searching through particle container, we can find the unused particle.If the life of a particle is less than 0, then we return it as unused/new particle. we can then add values for speed, life, color, position to ParticleContainer for that specific index. Besides generating new particles, we need to also make sure to delete old particles. As mentioned above, all particles eventually die. The particle container contains both living and dead particles. However, the buffer that we send to GPU should only contain living particles. So, we iterate through each particles and check if its living or dead. Depending on its state, we can update and finally copy it to GPU buffer.

The tutorial from [3] was followed for this project which looks like figure 1

For simulating fire, the color values for particles was changed to represent fire color palette. Also, the gravity was changed to represent fire as fire particles don't live too much like the one in 1 where the particles are also falling down. The resulting fire simulation is shown in figure 2.



```
Frame Rate
Application average 20.061 ms/frame (49.8 FPS)
```

Fig. 2. fire Simulation

III. Conclusion & Future work

By generating number of particles each millisecond in an up direction and using appropriate color to represent fire, we could simulate fire phenomena almost closely. Due to lack of sufficient time, it wasn't possible to do more work on it so as to make it look more real. Probably, by loading a 3D object like gargoyle, that looks like its spitting fire would be a good advancement.

References

[1] Particles/Instancing, opengl-tutorial.org. [Online]. Available: http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/particles-instancing/
[2] Tom Solarino, "ImGui Tutorial" [Online]. Available: http://www.tomsolarino.com/uploads/5/6/9/8/56982747/imguitutorial.pdf
[3] Allen Martin, "Particle Systems" [Online]. Available: https://web.cs.wpi.edu/ matt/courses/cs563/talks/psys.html