Computer Graphics Final Project

Oussama Soulimani S2439379

December 2021

1 Abstract

As a final project for the Computer Graphics course of 2021, we were asked to use the gained experience for building a firework simulation. As the term simulation states, realism is an important factor of this project. This documentation serves as a description of many aspects of my implementation.The different implemented forms of firework explosions are proposed as well as the way the firework raises to the sky. Moreover, there will be an explanation of the implementation and how I tried to make this simulation as real as possible in terms of lighting and explosions.

2 Description

For this simulation, the firework explosion was mainly divided into two phases: The first phase is for when the firework raises to the sky. The second one is the actual explosion of the firework and how it blows up until it disappears.

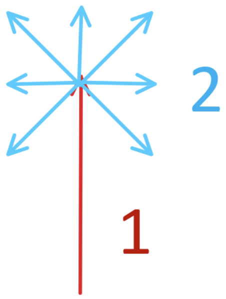


Figure 1: Firework explostion phases

3 Implementation

3.1 Raising style

Concerning the First phase, 3 styles were implemented. The first style is the basic one. Simply the firework raising straight up. Of course, gravity was also implement. When the firework raises, it slows down until the acceleration nullifies and becomes negative. This is also implemented for the styles discussed in the next sections.
The second style is the firework raising in a wavy style. The raising style is clearly demonstrated in this picture

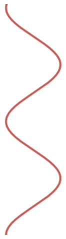


Figure 2: Second firework raising style

This style is implement with an inverted sinus function x = sin(y) with some small parameter modifications to make it realistic and match the size of the explosion.
The last raising style is a circular movement. The firework rotates clockwise about the up axis (y-axis) while it raises to the sky. This movement is clearly demonstrated in this picture.

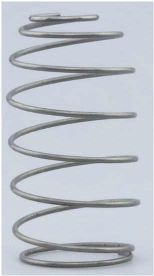


Figure 3: Third firework raising style

This style is implemented as follows: while the firework raises, the firework rotates around the up axis so the x value is the cosine of rotation angle and z value is the sines of the rotation angle.

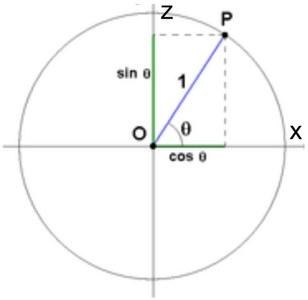


Figure 4: Third firework raising style

3.2 Explosion style

Concerning the explosion style, three styles were implemented. The first explosion style is a simple explosion where the particles move in the direction of a horizontal sphere. The angle of each particle is defined by the number of the particle multiplied by Pi divided by the the total number of particles. This gives us a circle where the angle between the particles is equal. Moreover, the x value is defined by the cosine of its angle and the z value is defined by the sines of its angle. These values are added to their initial values to make the particles moves out of the circle. This explosion is demonstrated below.

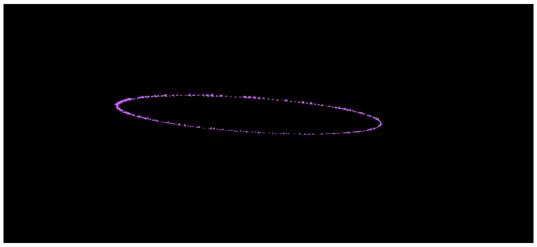


Figure 5: First firework explosion style

The second one is similar to the first one, but with three circles instead of just one circle. For this explosion specific particles where chosen to be placed on a specific circles. The particles are placed evenly on each circle to create a beautifully organized explosion.

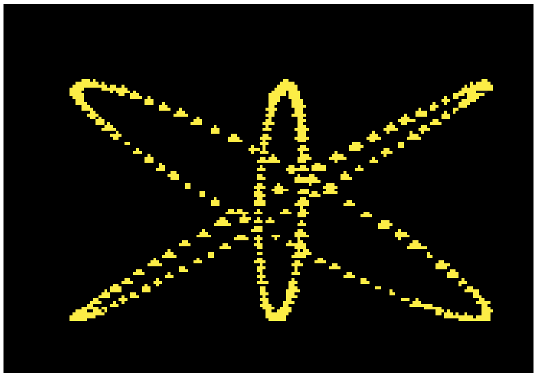


Figure 6: Second firework explosion style

The last explosion in the most common explosion in the real world, namely a spherical explosion. All particles move away from each other in a spherical form. This is implemented through multiple vertically placed circles. These circles circles are rotated about the up axis (y-axis) with an equal angle to placed the circle at the right position and form a sphere. The number of particles used in each circle of the sphere is the square root of the total number of particles. Square root particles are placed in each circle and there are square root particles of circles. The explosion is demonstrated below.

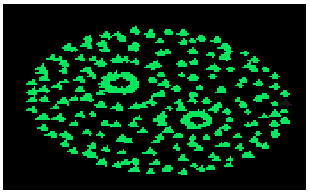
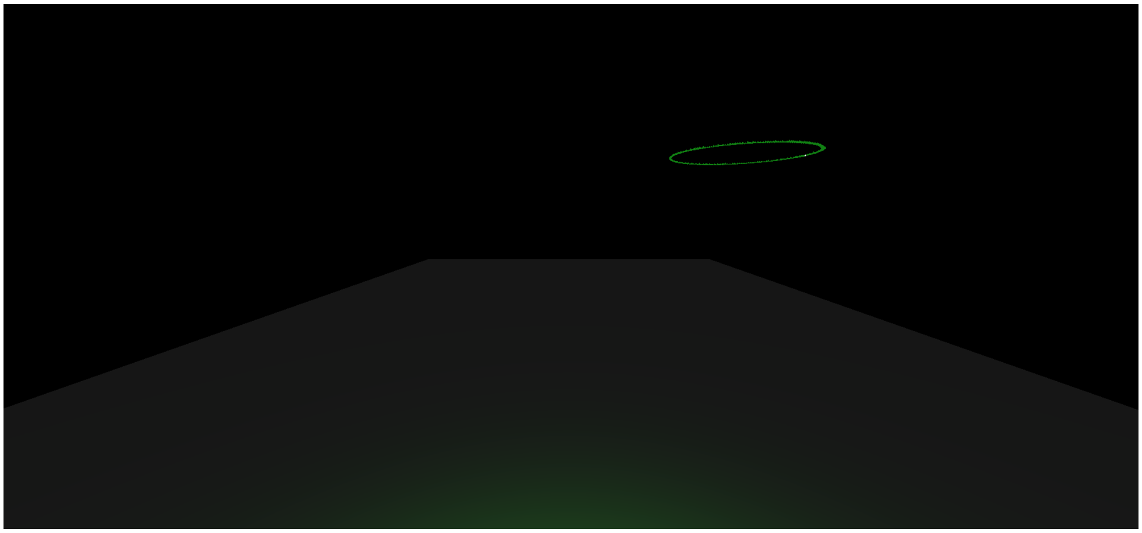
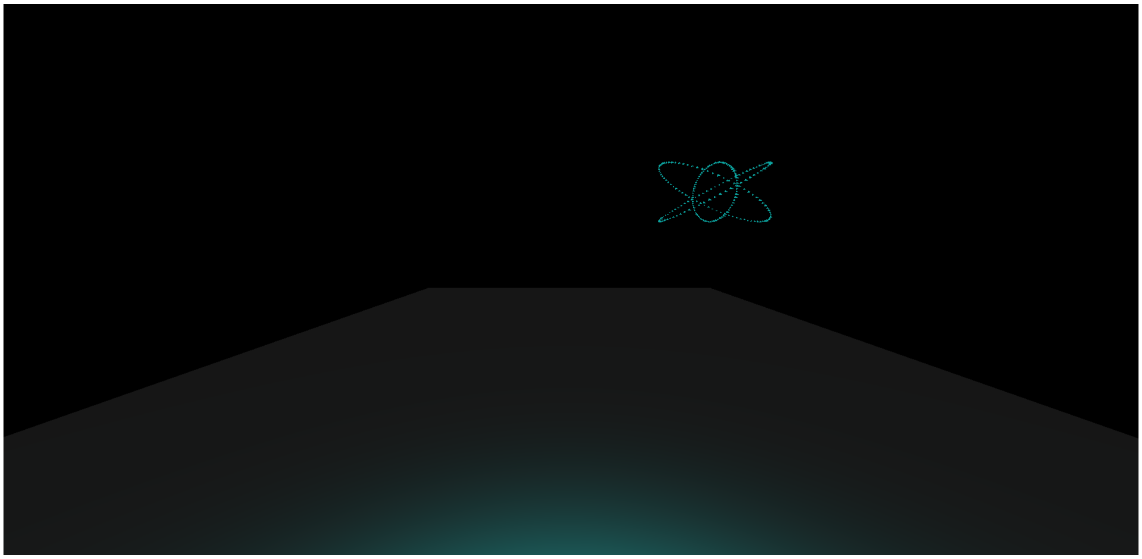


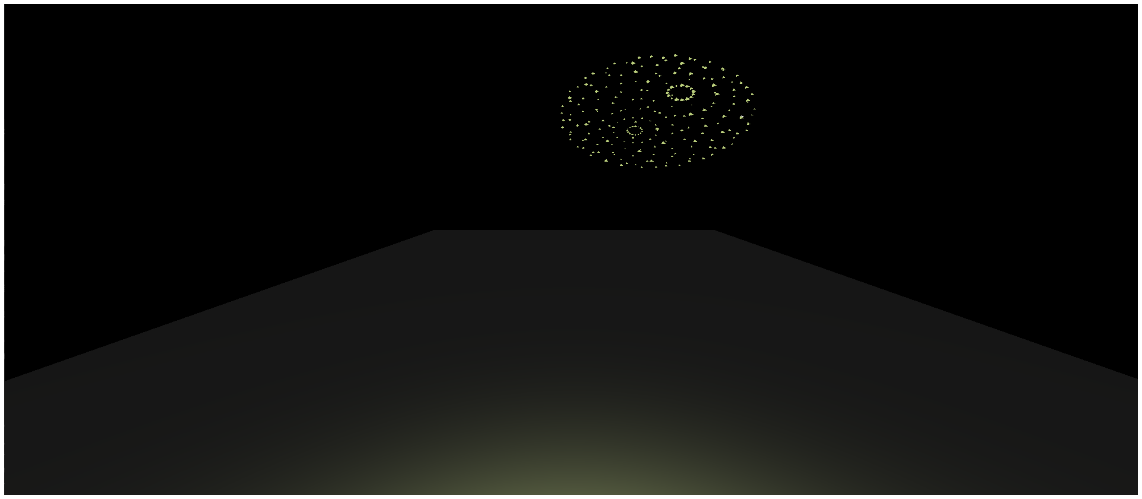
Figure 7: 3rd firework explosion style

3.3 Lighting

For this project lighting was the most difficult part. Light of the explosion is reflected to the surface/ground shown in the the executable. Because the surface is composed from four vertices, and the light is just perceived at these four vertices, the surface becomes brighter as a whole and dims as a whole. This was not realistic enough. The idea is to make just parts of the surface light up that are lightened up. To make this possible, the surface is divided into multiple parts / subsurfaces that lighten individually and dim individually. This serves as a implementation of the specular lighting. The lightsource however, is implement with a spot directional light, to brighten just the areas up that are near the explosion. A demonstration is shown below.







4 Summary

So far, multiple aspects of this project were explained and demonstrated. The style of the firework as it is raising, the style of the firework as it is exploding, as well as lighting. The latter was the most difficult part to make as realist as I had in mind. The surface is not lightened up in the right spots, at a specific time. The light dimming however, was quite realistic in my opinion. What concerns the originality of my code, except the audio effects, almost the whole code was written by myself and the start was inspired from the first workshop. That workshop was used as a starting point for my code. The discussed concepts in the lectures were also used. The audio effects are defined in the audio.h and audio.c files.

References

[1] Workshop 1
https://lcm.liacs.nl/courses/adisplay.php?fileid=cd43f893ca02cdd87d53ed61ab655687

[2] Lecture Math, Transformations, OpenGL
https://lcm.liacs.nl/courses/adisplay.php?fileid=e66fd0910904571baba4901789581ce1

[3] OpenGL PreWorkshop Slides
https://lcm.liacs.nl/courses/adisplay.php?fileid=a1fd8fff1a7a35040abebb87e54dd09d

[4] Lighting, Shading, Illumination
https://lcm.liacs.nl/courses/adisplay.php?fileid=3ddf2c754f5cd1837db7571dcfa98a73

[5] Audio effects
https://github.com/jakebesworth/Simple-SDL2-Audio