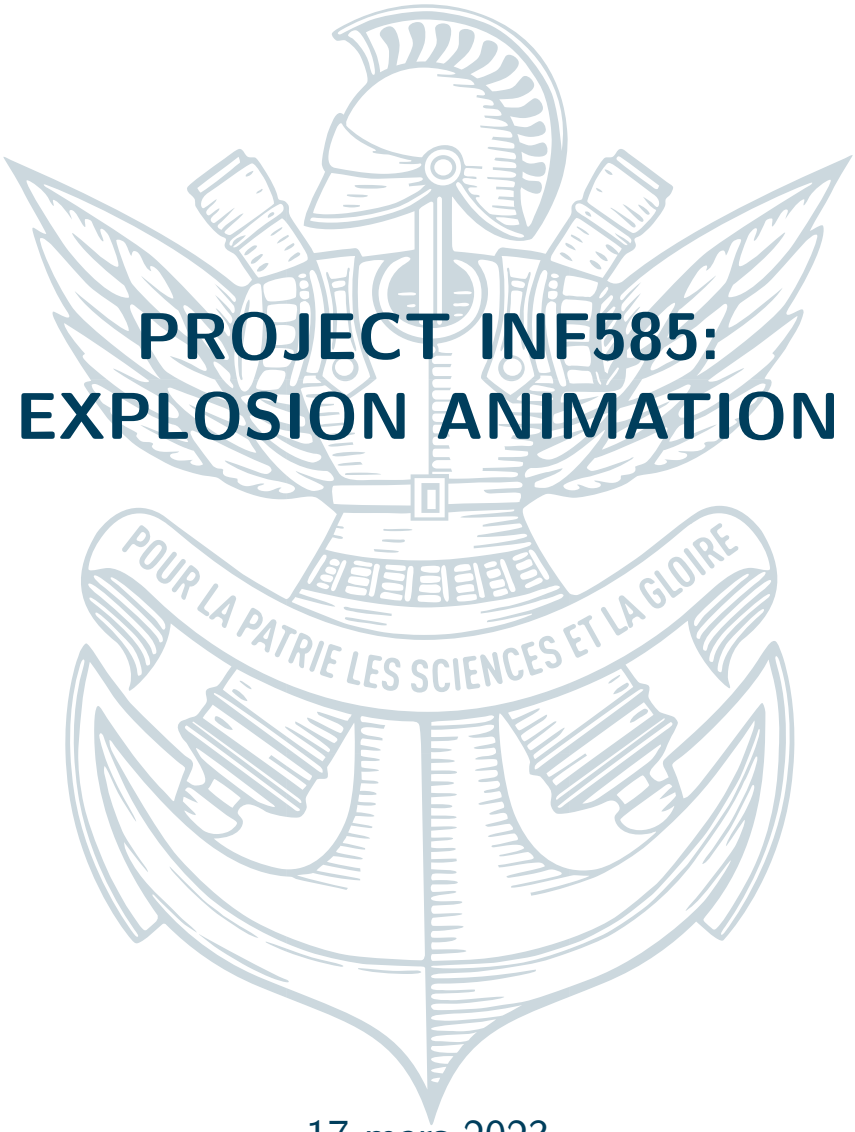

PROJECT INF585: EXPLOSION ANIMATION

17 mars 2023

Nikita DUDOROV, Vannvatthana NORNG
X2020

ÉCOLE POLYTECHNIQUE

IP PARIS

1 INTRODUCTION

In this project, we would like to make a simulation of explosion animation. In particular, in this scene, we make each window of the building explodes with the broken pieces and dust blown towards the outside direction of the building (figure 1).

FIGURE 1 – Similar example of the effect to implement (from The Matrix(1999))

2 IMPLEMENTATION

In order to facilitate the implementation of the project, we use the GUI template given during the practical sessions. The simplicity of this interface made it easy to navigate and user-friendly. It also helps us save times from trying to create another user interface and focus only on the objective tasks for creating the effect. There are two main parts in the implementation : the windows of the building and the whole building (except the windows).

2.1 WINDOW PANEL

Window panel is the most important part in this implementation as it contains all the animation effect for the simulation. Suppose that it is made of glass, there are three different components that are combined for its behavior : the breaking of the glass panel, the dust from the force of explosion, and the explosion.

We calculate the broken glass effect using Voronoï diagram [1] to partition the panel into small pieces (Voronoï cells), and glue them together before the explosion event. In addition, we know that in real life, the force of the explosion also brings the dust particles and debris to the direction where the glass is blown. We have consider applying the motion behavior for each dust particle, but it would be extremely expensive in computational cost since the particles are too small, so we would need a lot of them to clearly see the effect. Therefore, we decide to use dust billboards as an alternative solution. Each billboard act as it is a single particle, but it launches a bunch of particles at once instead of one, which is significantly more efficient in terms of cost. The force of the explosion for the window panel and the dust particle is calculated using the equation

$$F_{\text{explosion}} = C * S_{\text{particle}} * \exp(-\gamma t) \tag{1}$$

where C is some coefficient (different coefficients for the window panel and the dust), S_{particle} is the characteristic area of the particle.
The friction force for both components is computed using the equation

$$F_{\text{friction}} = k * S_{\text{particle}} * v^2_{\text{particle}} \tag{2}$$

where k is the friction coefficient (also different values for both components).

For the flame animation, we use the animation sprite. Each frame of the sprite are cropped to be used as billboards, then all billboards are assigned to appear as a very fast slideshow and to synchronize with the breaking of the glass panel. There is no more computation involved in this step.

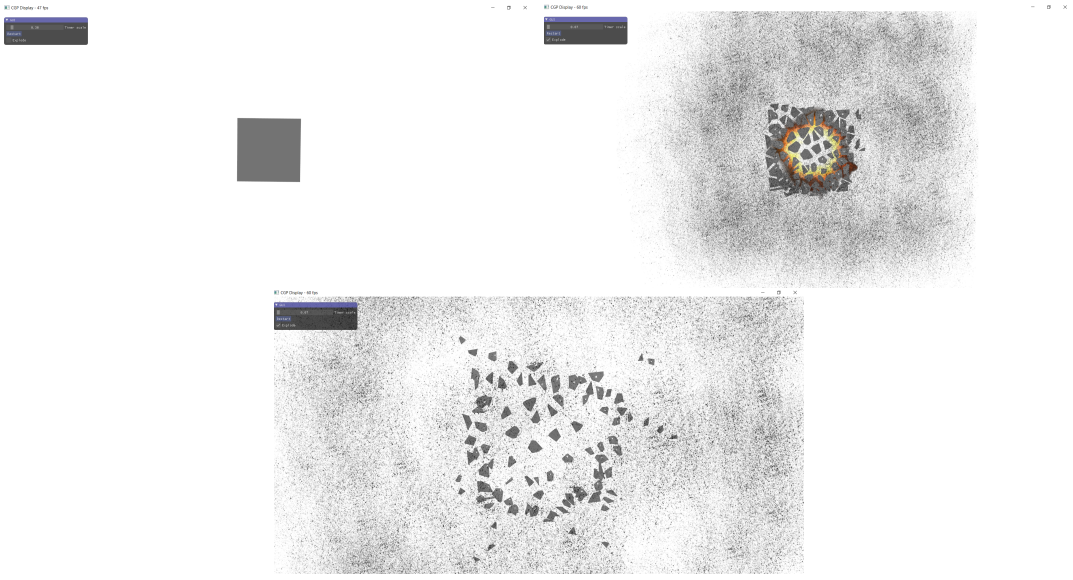

FIGURE 2 – Explosion of one window panel

1. In a plane (2D), Voronoï diagram is a partition of this plane into small regions (called **Voronoï cells**). Given a set of points, each point has it corresponding Voronoï cell, where each cell contains all the points in plane that are closest to that point than the others in the given set (For more information, see reference [3])

2.2 Building

The building where the explosion occurs does not interact with the rest of the scene, only the windows do (it is basically static). Therefore, we only need to find a building model (.obj file) with only frames for placing our implemented windows and import it to the code. However, the models available online always come with window objects, some are separated from the main body of the building, some are attached. So, we did a bit of modification (using Blender) on a model that we found, by deleting the window objects before importing it to the code. In addition, we also use some textures to apply to the building to try to find a better texture than the default white one.

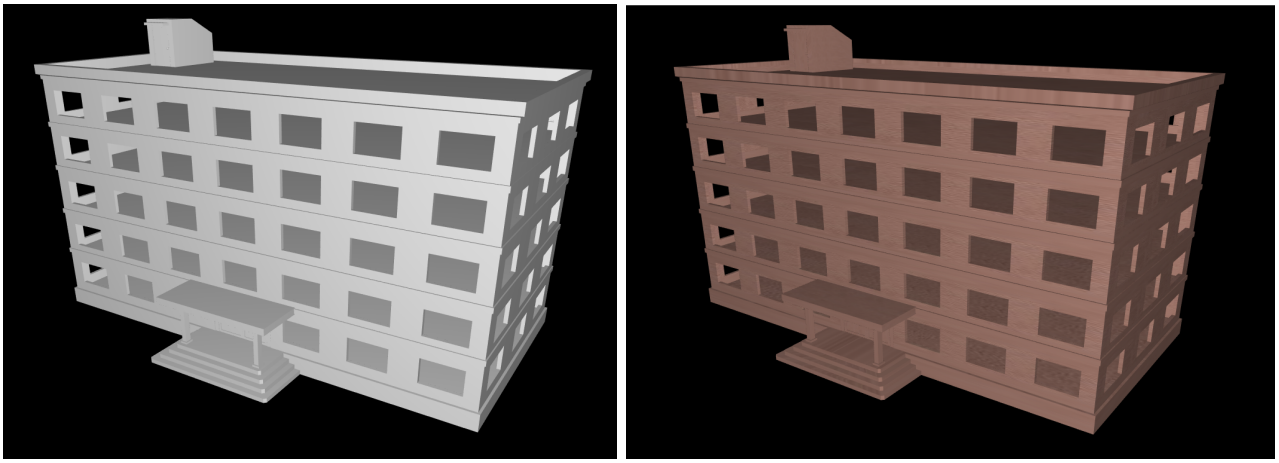

Figure 3 – Building model : without texture(left), with texture (right)

In the example shown in the figure 3, the building does not look really much better with the applied texture, it just looks different. Finally, we decide to keep this building at the default white texture.

After implementing the behavior of the window and importing the building, we need to put them together. The glass window panel is multiplied to be put on the building. In order for the panels to fit in the building, the size and coordinates of corners of each window frame are calculated by hand.

3 RESULT

In the final result, the animation happens in the center of the scene where the explosion is inside the building, and it causes the front face's window panels to break and to get blown away

in pieces one by one [2] (see figure 4).

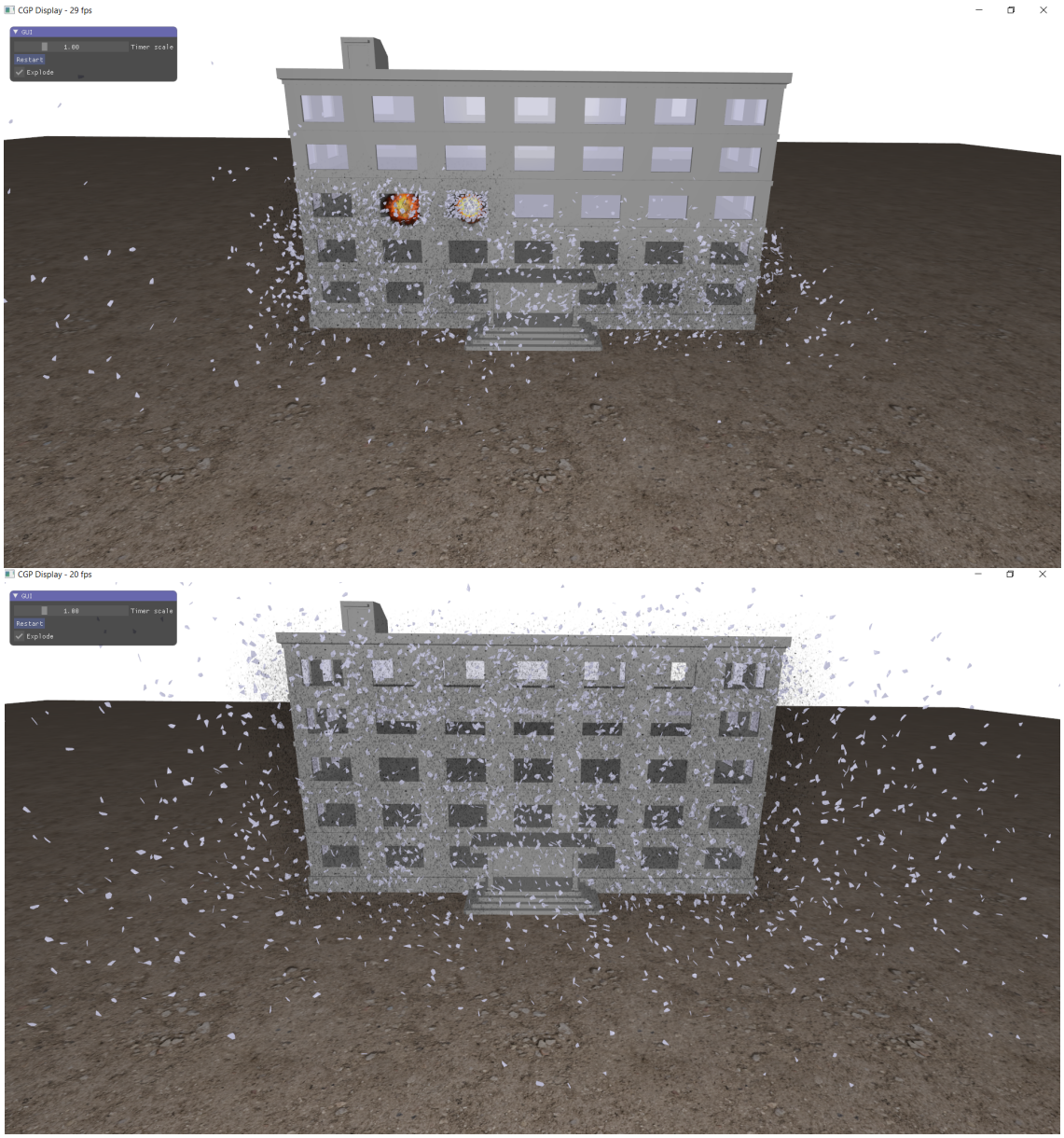

FIGURE 4 – Explosion effect : during (top) and after the explosion finished (Bottom)

2. The user needs to zoom out the scene before checking the "Explode" to start the animation, since the scale of the building is bigger than the screen size

4 CONCLUSION

We have successfully simulated the explosion effect with our implementation. The effect looks beautiful, it looks as good as it should be in this scenario. However, the algorithm can be improved in term of some computations. In particular, there can be more efficient ways on finding the coordinates of each corner of every window frame of the building so that we can fit the animated windows more easily rather than calculating by hands.

RÉFÉRENCES

[1] Voronoï software library : https://math.lbl.gov/voro++/

[2] TD's GUI template of the course INF585 for implementation (please click here).

[3] Voronoï diagram's information : https://en.wikipedia.org/wiki/Voronoi$_d$*iagram*