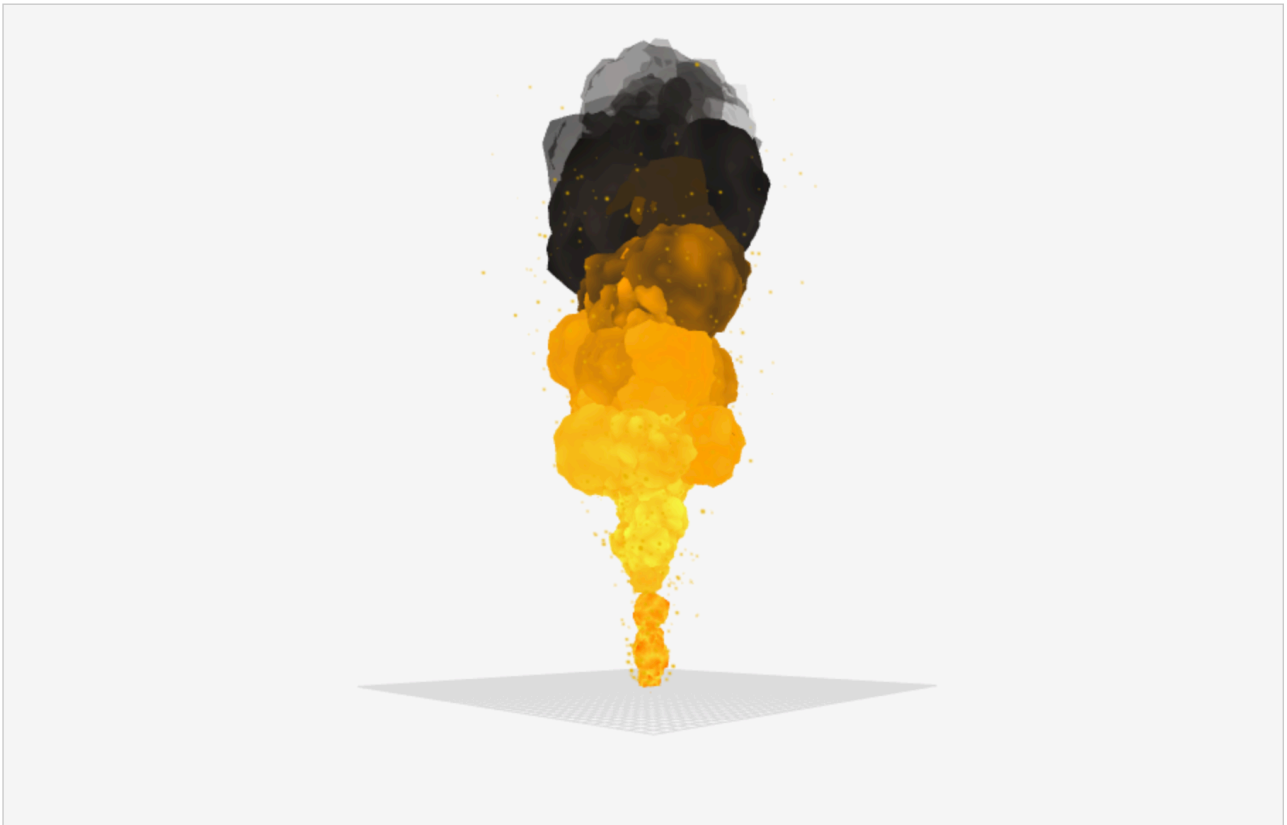

Fire Simulation and Customization

2110594 Computer Graphics and Physics Simulation

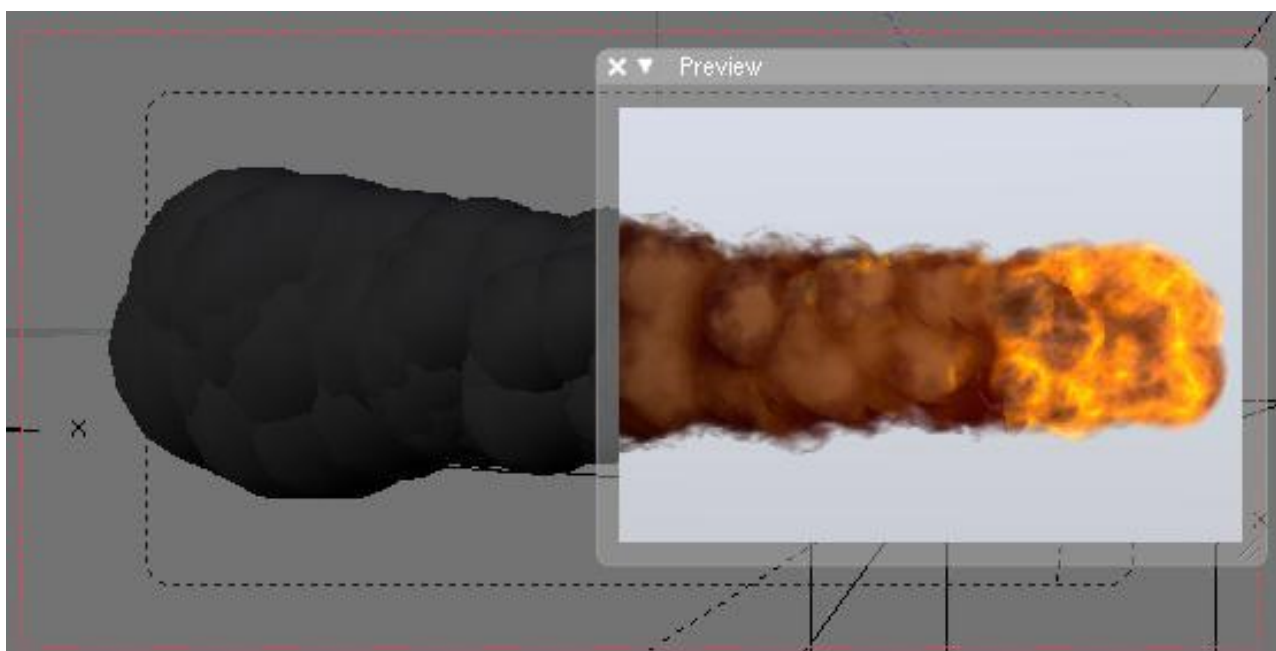
Kosate Limpongsa — 5731012721 — kosatelim@gmail.com



Introduction

Fire object are the popular material were use in the game application in the industry. But there some problem for simulating the fire because its has a way to hard to implemented for stunning realistic quality. If we ignore the quality, there are hundred of method can be used for fire simulation.

The method I used select one is volumetric fire as my approach to implement the fire simulation. Its use less consumption of hardware calculation and suitable for built on WebGL base application.



Volumetric Fire

Project Scope

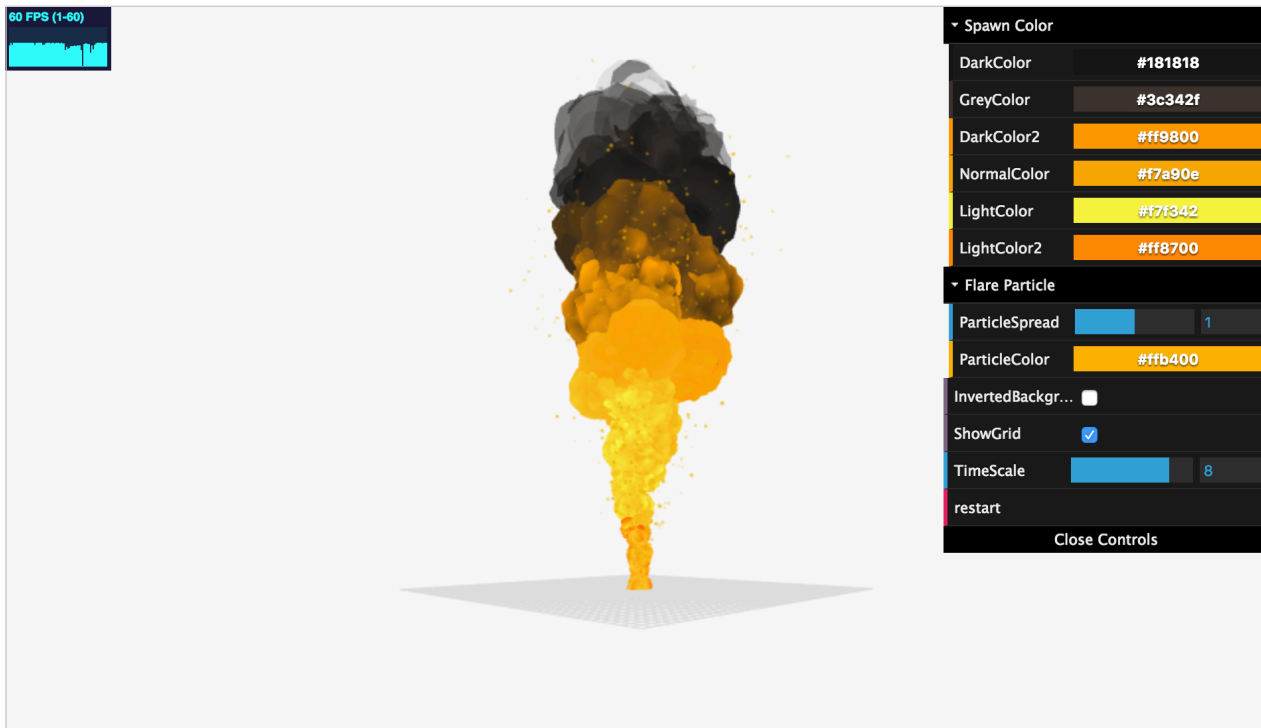
We propose a way of creating by using **THREE.js** for implemented this project. The fire material are implementation of vertex shader and fragment shader. Everything is done without the texture (Except the particle part). Also, run great in Google Chrome web browser and many customization of simulator with real-time monitoring statistic.

The project include two main parts. The simulation and customization. The first part I was introduced before. But the second part I use interface interactive tools base on JavaScript called **Dat GUI** and **stats.js** for monitor the simulation.

Finally, this project are written with **TypeScript** language, typed superset of JavaScript that compiles to plain JavaScript and detecting the type error.

Project Details

The project contains two main path. First is the simulation part, and second the customization part. Both part are work together the simulation will receive the information from customization panel. After you change the input, the panel trigger the change value to simulation and continue update so on.



Fire Simulation Preview

Part I : Simulation

I design to simulation fire using volumetric fire approach. The way I implemented is customize the vertex shader and fragment shader or sphere using Perlin noise^[1] algorithm to randomize the vertices of object.

This method were inspired by the lab during the class Vertex displacement using GLSL and THREE.js^[2]

After this process complete. Next step is combining of single fire ball to large flame. This process using some color transition and animation controller^[3] for generate the beauty fire to screen.

Furthermore, I also include the particle system to simulation for spark and combining everything to screen.

Part II : Customization

This part is not completed as before one. There are panel tools in the right side of the screen using Dat GUI library. You must mapping the event from triggering to some function for updating the real value.

The hard part is how to put the updated value to the simulation. First, we initialize some value in the class. Second, created a function for update value. Third, attach specific event and add callback function of second part. Then, put the value variable to the render update function that be call every 1000 / 60 seconds.

Optimization

I struggle with limitation of hardware resource. My computer is not fast enough for manage the data from simulation path.

What's happened ? The reason is when you create a fireball, there need several resource for allocate and process when fireball is gone. I just remove its from memory. These made Ping-Pong effect to memory.

The solution is using object pooling for fix this problem. When the resource were not use. Try to recycle its. After fix this issue, the simulation is now run perfectly with 60 fps.

Controller

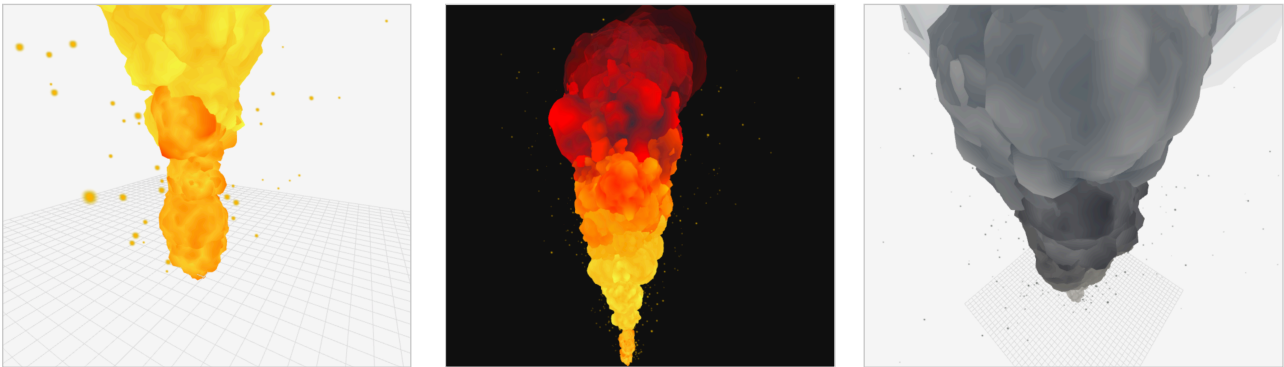
Use your mouse to play around the simulation.

- Single-touch or Three-touch to pan around.
- Double-touch for zoom in and out.
- Double-touch with dragging moving the screen.

For customization panel click anything you want to customize the parameter of simulator.

- Spawn Color — Color of fire
- Flare Particle — Spark customization
- Invert Background — Switch color background between white and black
- Show Grid — Show the grid
- Time Scale — Speed time of simulation
- Restart — Restart the experiment

Result



(Left) Fire with zoom in — (Center), (Right) Customize the color

The project run very great in 60 fps. Fully color customization of fire are work well. There are a little ugly with some color transition that happened in fire flame but you can fix it by change the color from panel.

Spark customization can spread out the radius of spark.

And you can customize the speed of simulator by changing value TimeScale in customize panel.

By object pooling implementation. The simulation can be running very wrong without frame drop. But it's expensive battery consumption.

Problems

The animation part is very huge problem of my project. Start from object transpose, there are many method for animation the object, there are many choice for interpolation to picking and apply to program. But the result looks ugly and look so fake in the first version. So, I must implemented all the animation with my own, with a little help from interpolation equation that I find from another website.

The another problems is color, because I use linear interpolation for transition the color of fire. The fireball which in between the transition of two color may be look like cartoon shade not same as realistic one. I try to switch another way but it's take a long time.

Further Development

- Find another way to implement color transition of flame.
- More accurate and realistic animation implementation.
- Add distortion effect to flame
- Add bloom post-processing effect to simulation
- Use texture for flame instead of solid color
- Extend customization option

Source Code & Video

My code are available in : <https://github.com/neungkl/fire-simulator>

Online demo : <https://neungkl.github.io/fire-simulator/>

All source code written by me. EXCEPT.

- src/js/* — Camera control file
- shader/vertexFlameShader.glsl — I customize a little bit of file but most of all are from website.

Video Presentation : <https://youtu.be/AmNtyLVsdu4>

Reference

Development Resource

[1] Palin Noise https://en.wikipedia.org/wiki/Perlin_noise

[2] Vertex Displacement with a noise function using GLSL and three.js <https://www.clicktorelease.com/blog/vertex-displacement-noise-3d-webgl-glsl-three-js>

[3] Easing Equation by Robert Penner <http://gizma.com/easing/>

[4] Object Pool Pattern https://en.wikipedia.org/wiki/Object_pool_pattern

Tools and Library

THREE.js — <https://threejs.org/>

Dat GUI — <https://github.com/dataarts/dat.gui>

Stats.js — <https://github.com/mrdoob/stats.js/>

TypeScript — <https://www.typescriptlang.org/>

Author

Kosate Limpongsa — 5731012721