

# CS 432 Final Project Report

Sarah Kushner

June 8th 2015

## 1 Idea and Why

For the final project in this class, I wrote a very basic particle system that simulates fire (or a phenomenon similar to it, whatever that may be). It is not as detailed or realistic as I would have liked, but I chose a relatively difficult project for me in the time constraints.

I wanted to explore simulation in general, and what better way to do that than to build a particle system. Also, fire is pretty and interesting to look at. I love things that share natural beauty and elegant mathematics.

## 2 Interaction

The user is able to stop and start the fire's motion using a drop down menu, and also control the parameters of the fire. Using certain keys, the user is able to increase and decrease the speed and size of the fire as well.

## 3 Preparation

To observe the motion and color in fire, I watched YouTube videos and looked at pictures, as well as other already implemented simulations from others.

## 4 Issues

I was planning on having a few more features than I ended up with. I originally wanted the particles to interact with objects using collisions, but it was agreed upon that the idea was too ambitious for the time I had.

In the spirit of trying to get the effect I wanted, I textured the particles as point sprites, but could not completely get the alpha channel working as expected. As a result, it is easy to tell that particles are in front and behind each other when I really aimed for a layering of transparent particles.

## 5 Sources

Using an example from chapter 9 in the textbook, I modified and picked out pieces I liked of it. It helped me understand the gist of particle systems.

I also used these links:

<http://buildnewgames.com/particle-systems/>

Even though it is in JavaScript and pseudo-code, the explanations of the life of a particle definitely helped.

<http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/particles-instancing/>

The above strengthened my understanding of instancing.

## 6 Algorithm

I have a Particle struct based on the one from the textbook. It is modified to my needs. To initialize the particles, I loop through the max number of particles I allow and give each one its properties. I give each one a semi-random location. I make sure it is “born” in the bottom middle portion of the screen by setting its position. I also give it a life which can be thought of as its lifetime. It can only live as long as this variable. Each particle also receives a color.

To add a touch of realism, each particle is textured with a radial gradient going from dark in the middle to light around the outside edges.

To change the speed and size of the fire, the speed variable and the number of particles variable is increased or decreased accordingly.

Like one of our usual assignments, an array of vertices and colors is passed down the pipeline through a buffer and drawn using `glDrawArrays`. This time, the idle function does a lot more. It decreases the life variable and creates new particles every time an older one “dies”. This is beginning to sound morbid.

Unlike the example in the textbook, my particles do not collide and bounce off the side of the screen and are not affected by gravity.

## 7 Instructions

To run the program, type “make run”. This compiles and runs the code.

Text instructions are printed to the console but here they are again: Keys: 'm' - more particles 'n' - fewer particles 'f' - faster particles 's' - slower particles

## 8 Demo

Pictures are on the next page. It looks way better when its moving.

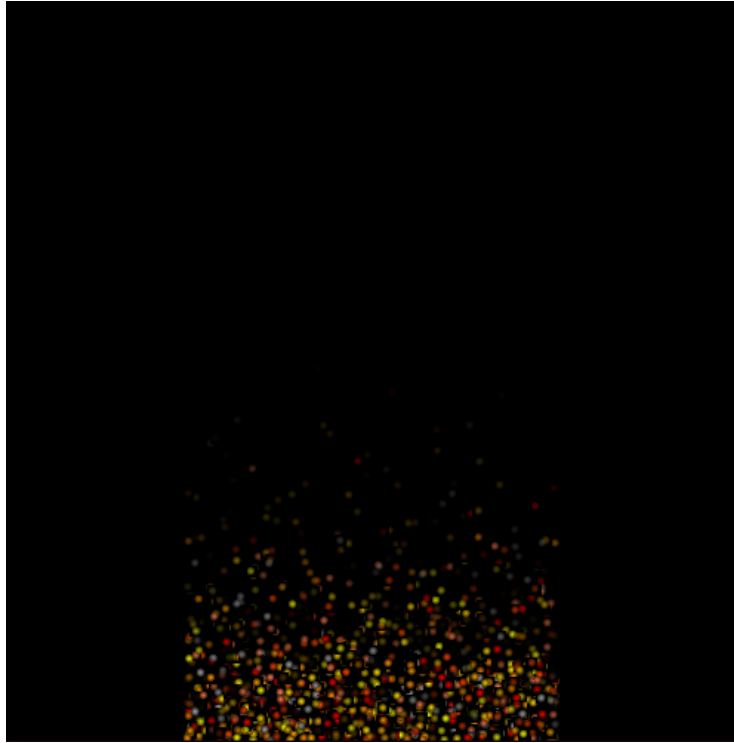


Figure 1: The full shot of the fire.

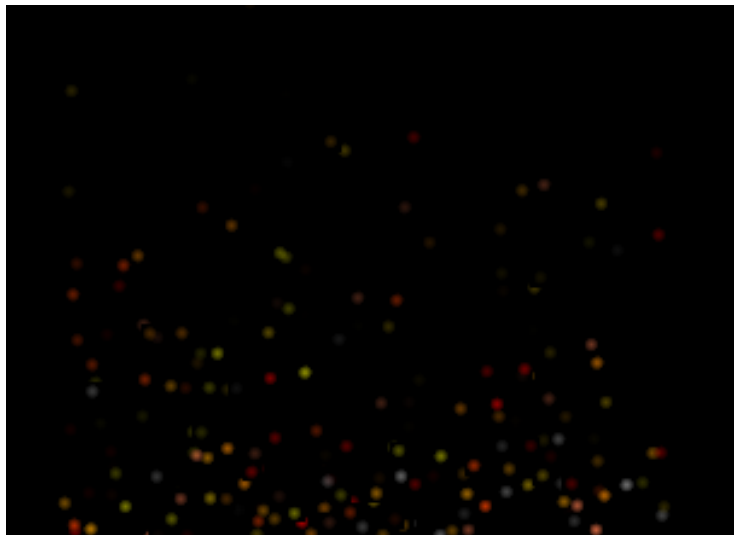


Figure 2: Close up of the particles fading because of their texture.