Ian Moon
IGME 230 Section 2

Project Type:
Interactive Visual Art

Description:
This project uses simple physics to simulate fireworks. The user can click or tap on the screen to fire off a firework, which is given off by a visual and sound queue. The firework is assigned a random velocity-component and is acted upon by gravity. Once the firework reaches it's apex, the firework explodes, which is again given off by a sound queue and an eruption of 50 particles. The particles are assigned a random velocity vector and a lifespan. As their lifespan decreases, the particles begin to fade from existence until they are completely gone.

Process:
My process began with deciding on my topic, which was originally a side scroller. However, after submitting my proposal and beginning work on it, I realized that it was really generic and, honestly, not very interesting to work on. So instead, I opted for improving at two key things in this course that I have felt uncomfortable working with. These two things are Javascript and working inside external libraries. Javascript I was really unfamiliar with and didn't really understand the syntax and my experience with Bootstrap showed me just how much I needed to work on my ability to work within someone else's framework.

Starting off, I decided on a new topic. I decided I wanted to go with something more art/simulation based and I wanted to tackle more of the vector based physics that have stumped me in the past. After a while of thinking about it, I figured that fireworks would be neat, as it deals with all of these aspects and would show me how to handle simple particle systems.

Starting off with this topic in mind I discovered, mostly on accident while looking around for Javascript tutorials, p5.js and decided it would be the external library I would use.

Breaking the project down, I realized I would need three basic scripts. The base script required by p5.js for various aspects related to their canvas, a firework script, and the particle system script.

After figuring out what was needed, I focused on the base behavior that I wanted accomplished. This was essentially three steps at first:

1. The firework flies up from the bottom of the screen, getting slower as it flies
2. The firework explodes when it reaches the top of it's flight
3. Particles erupt from where the firework used to be

While there were a few hiccups along the way, such as immense lag due to improper object handling and particles that behaved like fireworks, this stage went fairly smoothly and gave a decent but boring look and ran fairly slow when even a small amount of fireworks were fired. In addition to the aforementioned, other issues popped up as well during this stage. While, the project behaved properly on my computer, I noticed that as the screen size got smaller the fireworks would travel off screen and the particle system eruption was massive among other issues.

Naturally, given the above, the next stage would be debugging and refining. During this stage I focused on making sure performance worked well before focusing on alternate screen sizes, and tackled issues such as particle handling before moving on to optimizing the code for multiple screen sizes.

Finally, I worked on polishing up the user experience. This began with a visual rehaul. While efficient for contrast, a plain black background with a white firework bursting into white particles was not very visually appealing. In order to combat this visual boredom, I decided that fireworks would have a random color assigned to them and would pass this color onto their particle system. This allowed the graphics to remain simple, but added enough variety to keep the page interesting for longer. Also, an interesting bug arrived during this time within the p5.js library. When I switched my color mode to improve the visuals and add the random colors to the fireworks, I realized that I had lost the alpha channels that I was using to simulate the particles fading away and the background became a weird gray. In order to solve this bug I found a workaround where I change the color mode right before I draw the firework and particles and switch it back immediately after. The only side effect of this is a lingering path left on the background on my computer screen but not my phone. I could not figure out how to remove them and retain my alpha channels.

Next, I worked on the interactivity of the project. At this stage the project behaved pretty simply following these basic steps each frame:

1. Roll a random number between 0 and 1, if that number is between 0 and .4 create a new firework beginning at a random x-value on the canvas
2. Update all fireworks and projectiles
3. Draw all fireworks and projectiles

There really wasn’t much in terms of interactivity. To solve this, I changed the spawning of fireworks to something user controlled, which I noticed an instantly better reaction from people who I had test the page (as they oftentimes tried seeing how many fireworks they could spawn in an attempt to crash their phone/tablet). However, the page still didn’t feel interactive enough, so I decided to allow the user to also control the x-value the firework began at. Finally, I added sound effects, as at this stage this was the most requested feature by those I had test.

Resources Used:

The project was completely done in Javascript utilizing the p5.js library. I did minor sound editing in Audacity to lower the volume of the two sound effects I used as well as shorten one. Throughout the course of the project Youtube, StackOverflow, and WC3Schools were used as guides to help with many facets of Javascript and refreshing on general coding practices.

New Skills:

Going into this project, I did not feel overly confident in my ability to program in Javascript. However, thanks mostly to the fact that this project was entirely done in Javascript, I now feel I have a solid grasp of the syntax and the basics on how to accomplish the tasks I would normally do in C#. Also, I was working with I library I had never seen before. This caused me to have to delve deep into their references and watch numerous tutorials on how to properly utilize various aspects of my project. Among these was ensuring that my project behaved properly across various screen sizes.

Overall, after this project, I feel vastly more confident in my ability to program inside Javascript and work with external libraries in order to accomplish my goals. As such, I will most likely be utilizing Javascript with the p5.js library in future projects down the road.

Future Improvement:

While I think the program is good as a simple simulation and is enjoyable for a few minutes, there are definitely some areas of improvement and I've brainstormed a few aspects that could be added in the future to make the interactive, visual, and technical side more engaging and smooth:

- A mouse drag or touch drag feature that would allow the user to curve the last spawned firework
- A better particle system that would produce a similar effect without so many particles needing to be spawned
- The chance for the firework particles to form shapes when the firework explodes

Sources:

I edited two public domain sound effects to add a bit more immersion to the fireworks for this project:

1. The firework firing sound (http://www.freesound.org/people/northern87/sounds/88532/)
2. The firework explosion sound (http://www.freesound.org/people/Werra/sounds/244394/)

Other Notes:

As a result of using p5.js, all work is done within Javascript on an interactive canvas and did not require any CSS.