Yahiya Hussain
Professor Haehn
CS 460
12/20/19

Final Project Report

This semester I have embarked on a journey to explore both liquid simulation and Vanilla WebGL. Both it turns out were exceptionally difficult, but I have learned so much while doing this project, even though I was not able to achieve the exact results I yearned to in September. I started out by messing around with the original code for the ship game in one of the earlier assignments. It was so interesting to play with such low-level code and seeing how much power was in my hands when I coded simple things in javascript. I decided to commit to doing the full project in Vanilla WebGL without the help of a framework, which was quite a perilous journey it turns out. I spent countless hours with a black screen and red arrows running down the chrome web tools console. I was willing to settle for a half-rate simple advection simulation if time ran out. With the help of the textbook, "Programming in WebGL," I was able to start with a spectacular though modest achievement of setting up a Quad geometry and texture, which I would continue to use and basically not change for the remainder of the project. I learned how to switch shader contexts and how to upload values to shaders and I quickly came up with cool designs as you can see if you look at https://yahtzeerage.github.io/Basic-Fluid-Simulation/folder/primitive/randomArray.html, but it wasn't enough for liquid simulation, I need to go far beyond just inputting parameters into shaders. I needed to get my hands nice and dirty with textures. I can't show off my early efforts with textures since github is not working properly, but if you run my folder in a python http server and go to **Basic-Fluid-Simulation/folder/basics** then you will find them and see the power of blurring. Once I could make the GPU not only see one pixel, but also see all neighboring pixels, now I was talking, because I could simulate trajectories of things moving over time and not require a closed-form math equation which determines where something is at each time-step. If you look at https://yahtzeerage.github.io/Basic-Fluid-Simulation/folder/roughdrafts/Advection_Angles.html you will see primitive and buggy but really interesting and pretty rainbow colors in a sort of runny egg grid pattern. It was really fun playing around with these kinds of things and seeing the interesting art I could come up with just by messing around. I also had to learn quite a lot of math and theory in order to even start to tackle these things. Fluid simulation requires vector fields if you want nice and performant designs and this makes the equations unintuitive and weird to the uninitiated because they represent fluid flowing past each pixel rather than actual particles. Advection is of course the tendency of fluid to carry both materials and its own velocity with it as it moves along. I used the Navier-Stokes equations using many different tutorials online and they all agreed that in order to simulate liquid, not only must you have a velocity at each pixel changing the colors on screen and their velocities in time, but also you must have a pressure of velocities impacting each other so that you don't have a compression onto a single-point, since in nature water generally doesn't collapse into a black hole. Thus I had to learn an algorithm called the jacobi iteration method in which I solve systems of linear equations

(which include the pressure of all pixels in the grid all as variables being represented) one step at a time. The computer cannot do linear algebra the same way as a human without getting massive rounding errors and slowdown, afterall and also programming such a thing would be a catastrophic mess. I've learned so much in this project, that it is hard to really state how much. I've written tens of thousands of lines of code, though scrapped most of it. I didn't full achieve the results I wanted at the beginning, but I am passionate at this point and will definitely work towards learning more and refining my methods and code to get rid of the bugs within in. There is a tendency towards hard lines in my final project, which indicates it has no yet reached the status of liquid at this point, but it is leaps and bounds beyond what I thought myself capable of at the beginning and I am proud of what I have accomplished for this project this semester. I have submitted a pull request of code to the cs460 github, but I also have my own page in case it doesn't work or the website is buggy. I think the python server does a better job preventing bugs, if you want to look into my other spin-off art projects in the folders full of happy accidents of pretty graphical art bugs. https://github.com/YahtzeeRage/Basic-Fluid-Simulation My final project submission is the index.html file in the github repo, but it has some quirks to the link so I posted the link to the website for viewing on the repo page. The folder named folder contains folders called: FinalDraft, basics, primitive, and roughdrafts, which are all much easier to view on the python http server, but are also neat things I made along the way. I heavily commented the final project javascript file to make it easier to follow along and understand, though there are limits to understanding Vanilla WebGL and being sane simultaneously.
https://yahtzeerage.github.io/Basic-Fluid-Simulation/folder/roughdrafts/MuscleCells.html