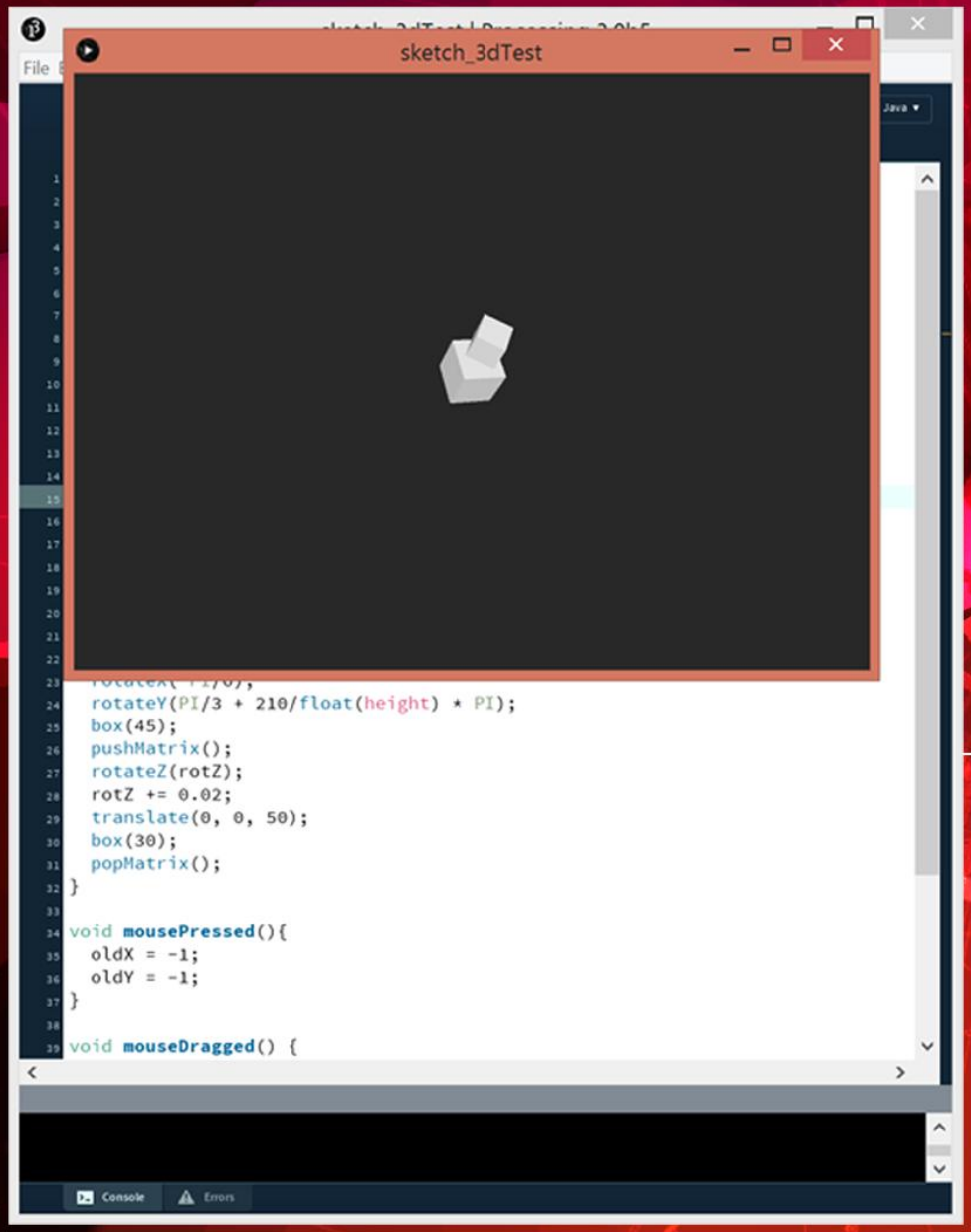
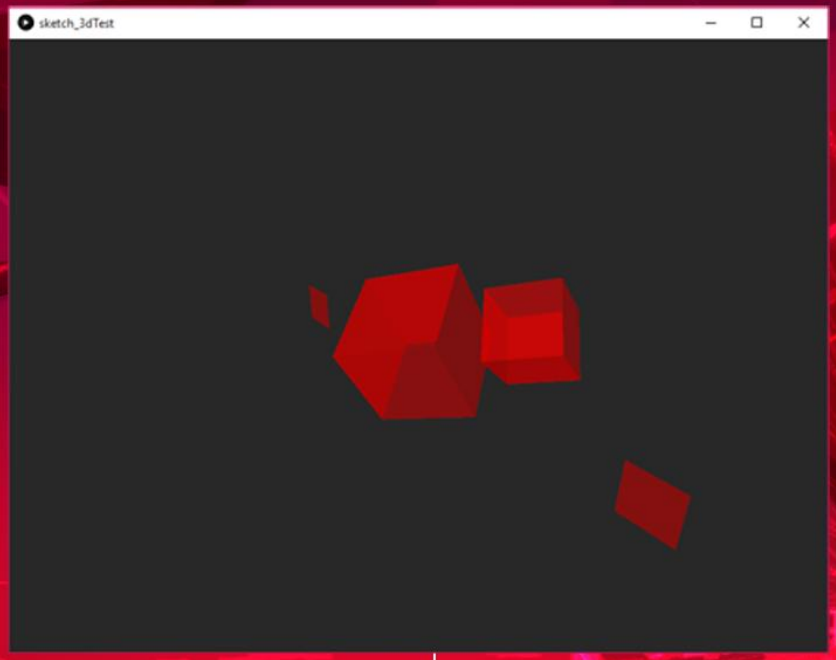

Processing.js Visualiser



WebGL In Processing

A quick screenshot of my first attempt at creating something from scratch using WebGL! I'd struggled to get my head around the structure of the code in the last SPA so I'd been wondering how I was going to be able to make something complex enough. I discovered yesterday though that Processing.js has built-in WebGL support, and since I'm familiar with both Processing and Java I was able to whip up something in 3D without any hassle - which hopefully bodes well for the rest of the project!

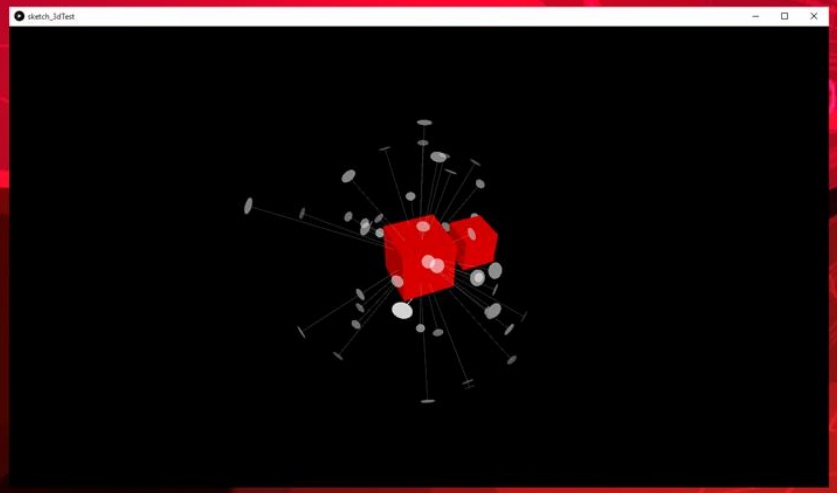


Moving on with the same Processing sketch I'd been working on, the first thing I added was proper, smooth camera controls. There's now a smooth interpolating camera rotate that works relative to whatever direction you happen to be viewing from, which was a serious pet peeve from the earlier version. There's also a very impromptu zoom function, which just moves the scene back and forward from the camera to simulate an actual zoom - this will probably have to stay like this until i get my head around the exact camera controls in Processing.

2D and 3D

I've also started investigating the possibility of integrating both 3D and 2D elements in a 3D scene. I found out that even when working in 3D, the 2D processing drawing tools still work, they just draw on a plane defined by the 3D rotations and transformations. This is both good and bad... Good in that elements drawn in 2D scale and hide behind other objects when appropriate according to the 3D coordinates. Bad in that I haven't yet been able to find an easy way of forcing the 2D elements to always face the camera, regardless of camera rotation. I though this would be a simple case of just reversing the global rotations but apparently not.

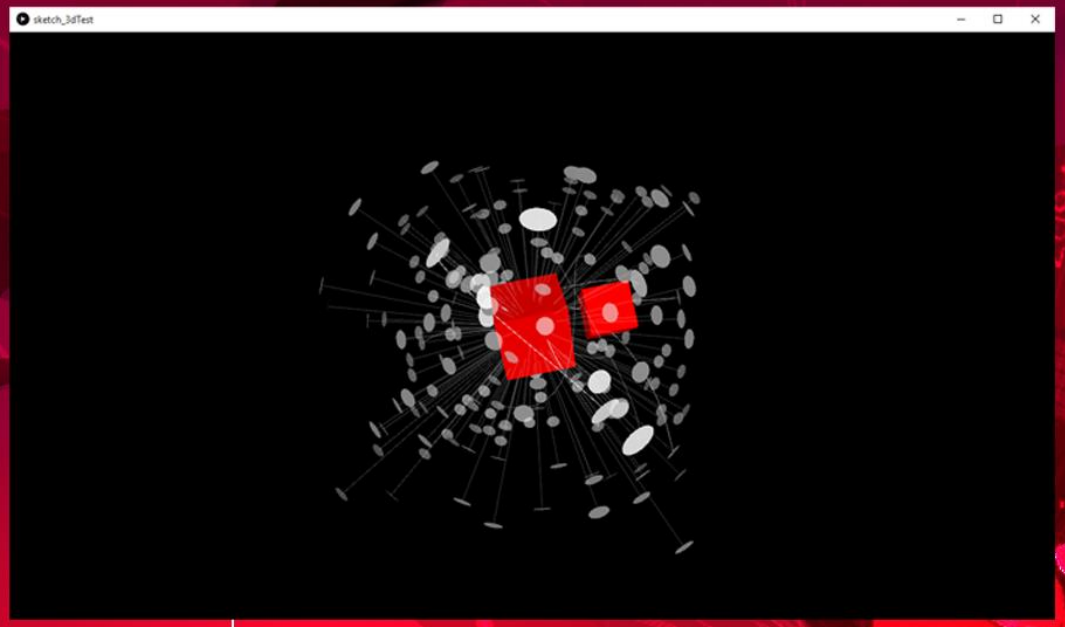
Further Experimentations



Since the last update I've done several updates to the code, the first of which was the most major. I added in 2D 'fireflies' into the scene to act as 2D objects in the 3D space. Initially these were just little glowing dots in space that were flat from one side, but I've been playing with several options for the rotations. Leaving them rotated relative to the scene is just not that interesting so I've been playing with either having them always facing the camera, or always facing outwards from the central point - essentially forming a sphere where you can see the edges around the outside. I've mostly settled on the latter, as it makes the 2D/3D juxtaposition much more obvious.

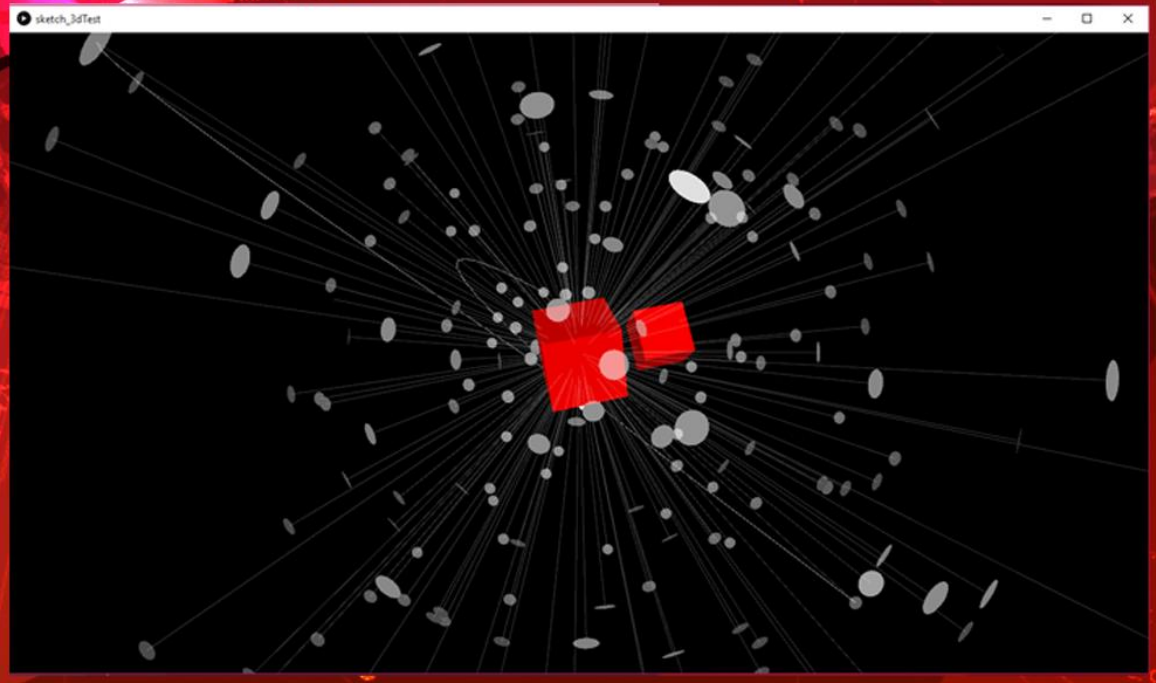
I've since had to change the way the fireflies were drawing to remove most of the transparency. Because of the way processing draws transparent objects you can only see through objects to ones that were drawn before them, meaning some objects are just cut out when they should be visible.

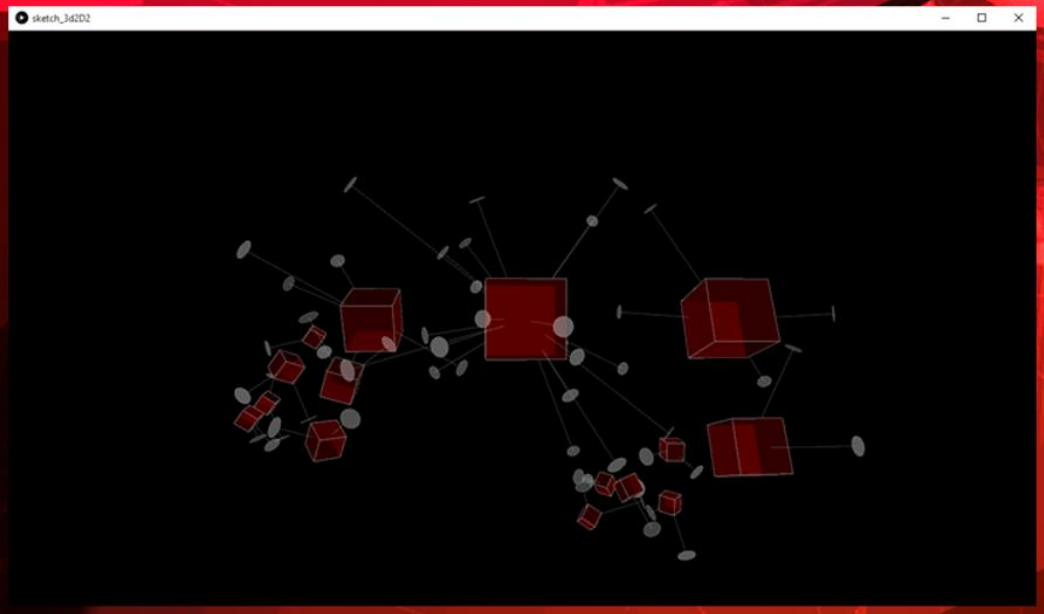
In another big update I've connected the fireflies to the central cube with 2D lines and I'm now drawing them as plain circles (not sure if they're going to stay like this but it's good for testing)



Lightning

To make the scene feel a bit more alive, I've added sporadic 'lightning' between fireflies, this causes them to flash and draws a curve between them. It's quite impressive in motion and i'm hoping to build a bit more of a tangible feel in with it :D



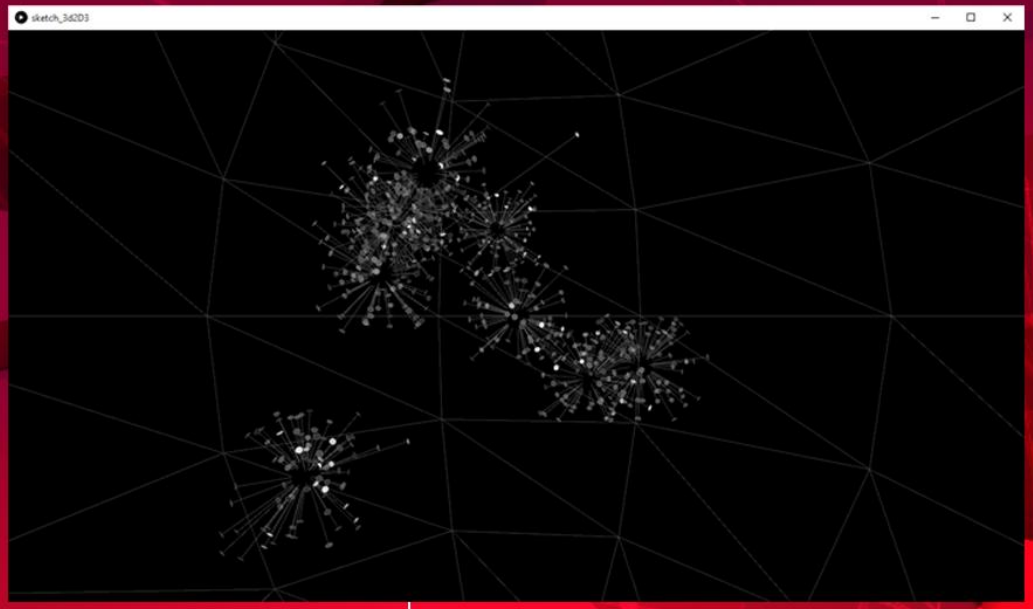


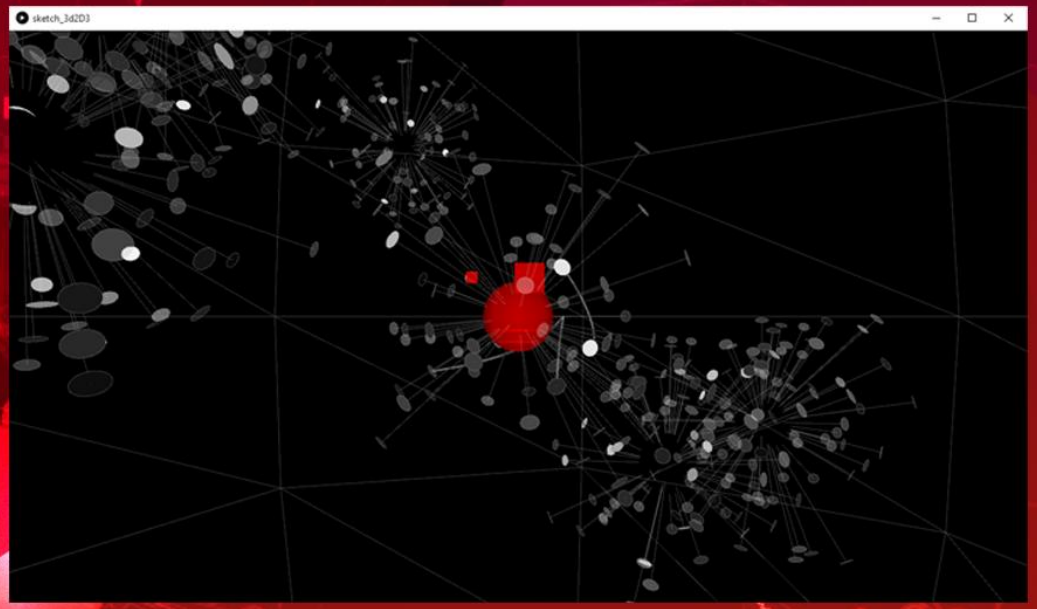
Arms?

I have a sneaking suspicion I'm heading in the wrong direction at the moment. I've set up a structure for the cubes, with each one linked to the position and rotation of a parent cube.

This way the whole structure rotates and bends and sort of has arms? it's not nearly as interesting as I thought it would have been, I was imaginging more of a giant imposing structure look, so if I don't make some improvement soon I might ditch this idea altogether.

I do like the progress with the rendering of the cubes though, the transparency looks great and I think I'll take that part further





Getting Somewhere Interesting

Big overhaul! Got rid of the linked cube rotation rubbish and started working on the environment as a whole.

There's now a skyboxcube-thing as a background - I added this as it was very hard to keep your bearings when things were rotating everywhere as well as the user-driven rotation.

I also changed the cubes to spheres, as they fit the rest of the scene slightly better (also because it looked more natural to have the lines connecting the fireflies coming out of a sphere).

There's now more spheres too, all rotating separately (although currently stationary in space). these spheres also flash in a manner similar to the fireflies, with the red flare and lightning connecting them for an instant

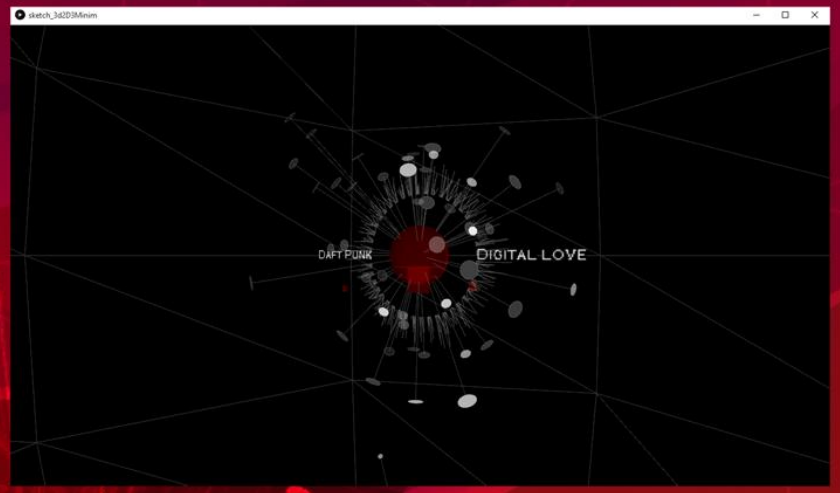
Visualiser Pt. 1

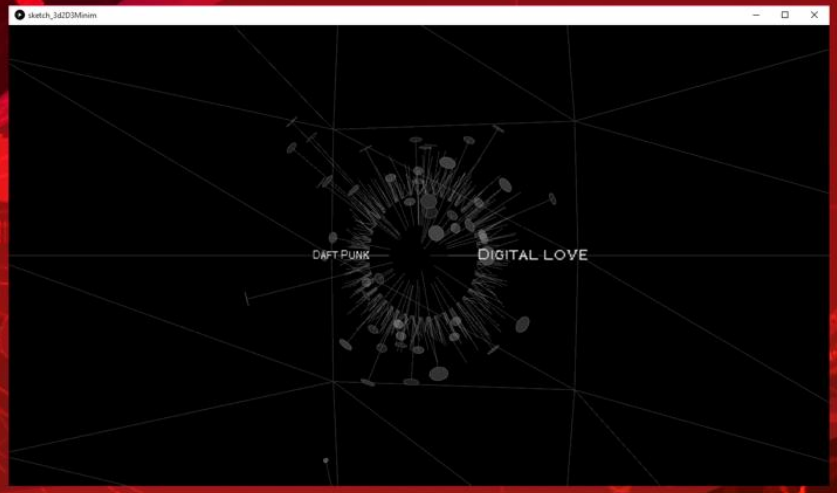
Been away for the weekend but I've still made a lot of progress! I've built the project into a music visualiser using Processing's Minim plugin.

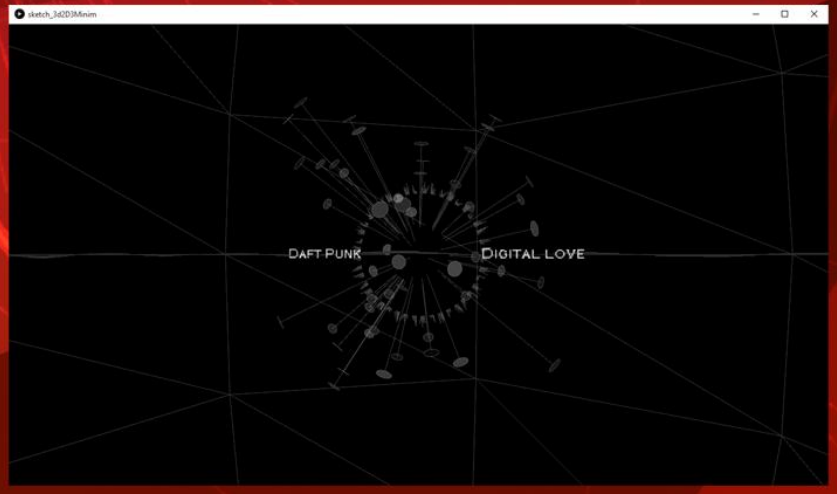
The first sound-controlled visual was the ring around the central sphere, this pulses with the levels in the current track (at this point it only played one track but I built on this later on). I also set up a display for the song meta-data beside the central sphere. Once again all of these effects are rendered in 2D in the 3D space, so rotating the view gives a different view of the display.

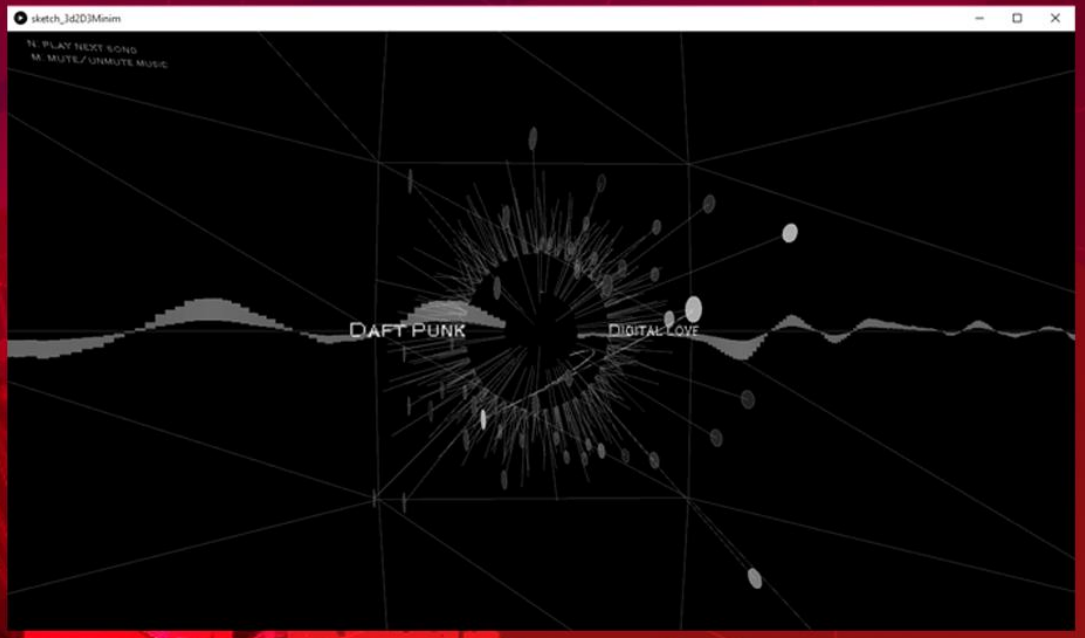
The second main improvement was adding the waveform display running through the center of the space, this displays the current waveform of the sound.

Next I started utilising the BeatDetection algorithms within Minim, and rebuilt a lot of the earlier code to accomodate this. The flashes of the spheres are now in time to the beats, and each beat causes another lightning/spark through the firefly dots. The fireflies have more behaviour now too, when they spark it gives them a temporary jump in speed - resulting in little flashes of energy as their colour is now determined by their current speed. The distance they travel is also determined by the volume of the music, quiet music causes them to sink into the sphere, barely appearing at all.





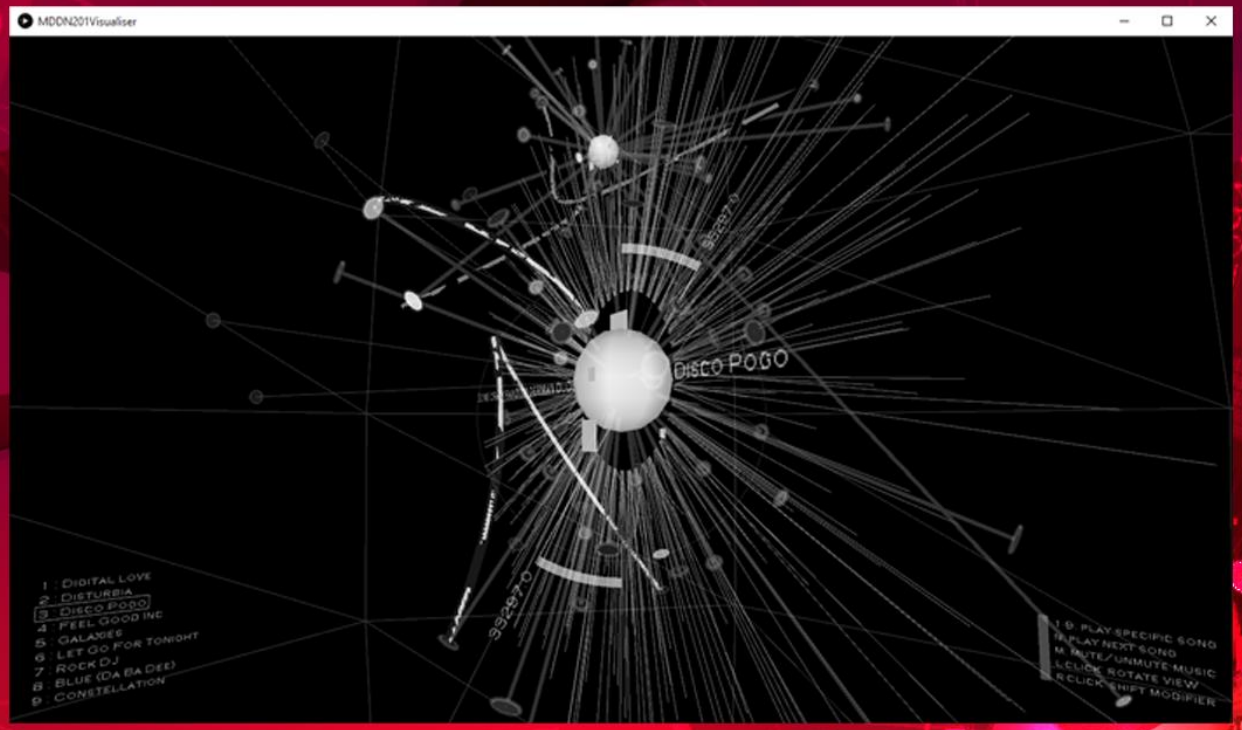


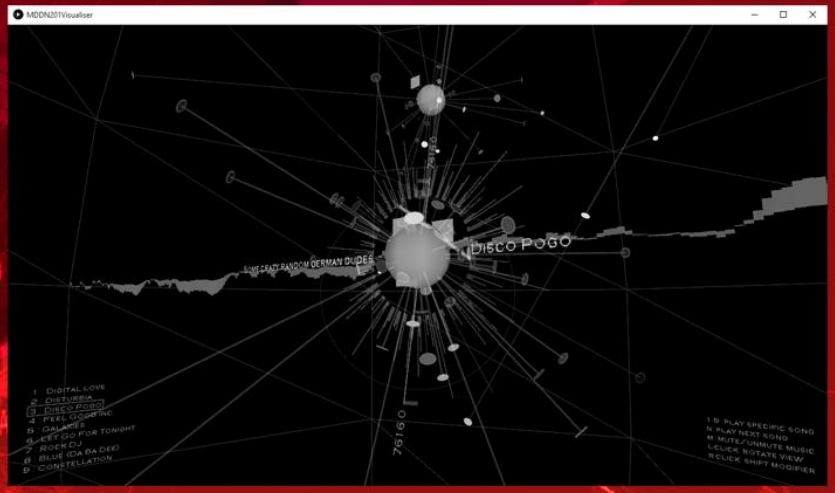


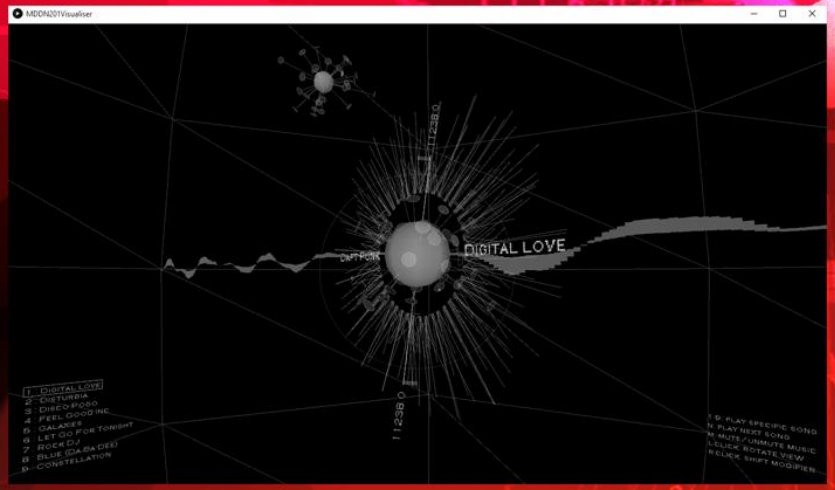
Visualiser Pt. 2

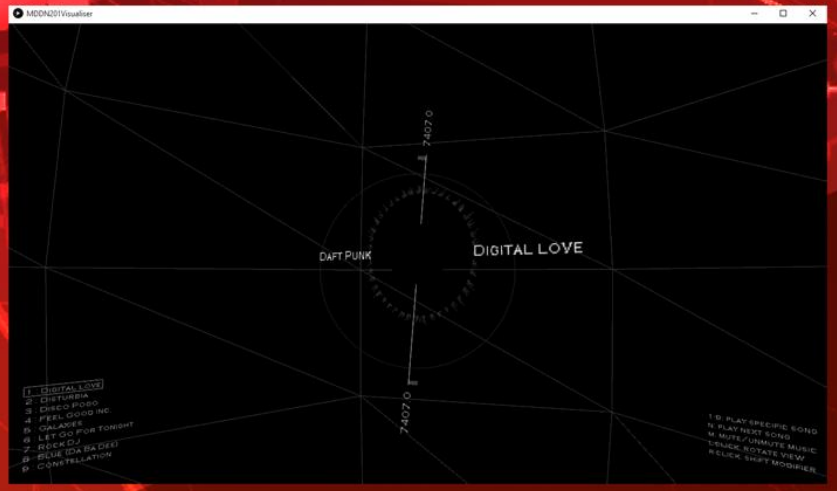
I've also built up several UI elements, all of which are rendered in 2D in the 3D space, and so are affected by the rotation of the view and can pass behind other objects in the scene.
First are the controls and song list displays at the bottom of the screen, these stay anchored to the corners of the screen but rotate vertically so they are always upright relative to the scene.
Then were the song completion rings that sit right at the back of the scene, these stay at a single height but rotate with the view around the y-axis so they're always behind the middle of the view. these display bars that show how far through the current song is.

Other improvements include the background sphere rotations, smoother camera movement (including a vertical rotation limit) and a slow y-axis rotation if the camera is not being manualy moved.









Further Development

Here's some more screenshots of the new features, showing off the waveform, 2D UI elements, beat effects and what the visualiser looks like with no sound

Ok so I'd finished working on features I wanted in the visualiser (new additions included a scalable set of visualiser bars and fixing the camera rotations) and went to embed it in a webpage following the tutorials on the Processing.js page. I had tested this much earlier in the project with a much simpler sketch and it had worked, but now all I'm getting is a grey rectangle where the sketch should be. After looking around in google the only conclusion I've been able to come to is that something in my code isn't able to be converted to Javascript - either because I'm accidentally using a java library somewhere (99% sure this isn't the case), or more likely because the latest version of Processing has features that aren't yet compatible with processing.js. I have no idea what I can do about this to be honest, and it's really stressing me out because pretty much the only requirement for this project was that it was an in-browser experience.... I'm hoping that there's someone in at Te Aro who might know about this, or at least be able to help me figure out which bits of the code are an issue. If I have to cut features to be able to hand it in then thats what i have to do

Ok after checking up on the Processing plugin I'd used (Minim, reasonably standard for dealing with audio in Processing), I found that Minim isn't able to be converted to Javascript... This is really huge as all of the Visualiser functionality runs off minim, It's integrated into every visual element of the design at this point and I can't extract it without taking away the entire point of the design

Major Technical Difficulties

I've done some final quality of life improvements to the visualiser, made the rightClickShift functionality permanent and added in the ability to pause the music (which really should have been there to start with). No solution to the Minim plugin issue has presented itself though. I've found partial Minim functionality online where people have put together a minim.js to help solve this issue, but it's using a much older (2012) version of minim and lacks 90% of the functionality I'm using. It's just implementing the music playing functionality, so It's missing all the beat detection and sampling code I'm relying on.

References

Processing
https://processing.org/

Processing Minim Plugin
http://code.compartmental.net/tools/minim/

Music:

Daft Punk	Digital Love
Rihanna	Disturbia
Die Atzen	Disco Pogo
Gorillaz	Feel Good Inc.
Owl City	Galaxies
Foxes	Let Go For Tonight
Robbie Williams	Rock DJ
Eiffel 65	Blue
The Firmament	Constellation