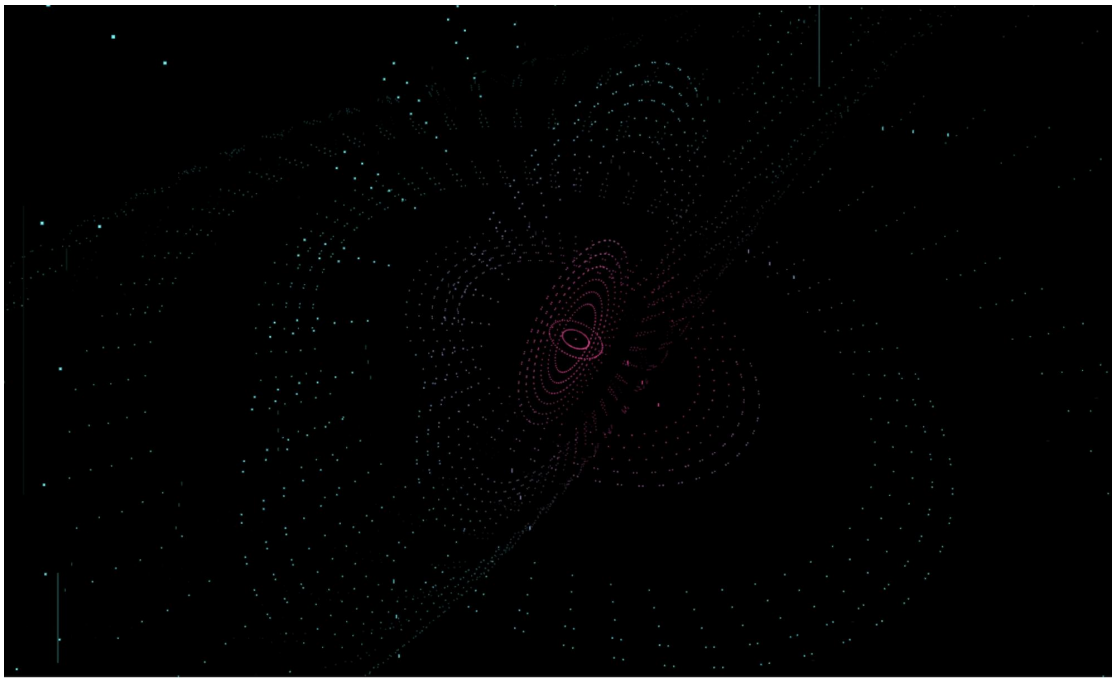
Project Description



MIMIC: https://mimicproject.com/code/2fc91290-133d-02e8-00de-3cff5bf5ab36

Please use Safari to open it (Google Chrome does not load properly).

Project brief

The project incorporates technology from Sound and Signal processing and 3D graphics. The 3D graphics engine was created using a simple 2D drawing context.

Sound Part

- Maximilian library is used.
- Three maxiSample objects are created using the Maximilian library, which are beat, loop, and chord.

```
audio.loadSample('loop.wav',loop);
```

```
audio.loadSample('beat.wav',beat);
audio.loadSample('chord.wav',chord);
```

- The maxiSample.play() function was used to create different condition triggers and so on to manipulate the sound.

```
var loopOut = loop.playOnce(0.2)+2;
var beatOut = beat.play(myOsc.sinewave(0.01))*10;
var chordOut = chord.play(1)*0.7;
out = loopOut + beatOut + chordOut;
```

- To make the sound more distinctive, the method sinewave() in maxiOsc is used.

```
var beatOut = beat.play(myOsc.sinewave(0.01))*10;
```

- Sets up a maxiClock object called 'myClock'.The clock speed is set to 120 by using setTempo(). And the number of beats is set to 4 by setTicksPerBeat().

```
myClock.setTempo(120);
myClock.setTicksPerBeat(4);
```

- The maxiClock system is also used. Run a test with a conditional statement to see if there is a clock tick. Call ticker() inside the play() function to advance the clock. The song() function is created and the conditional statement and modulus % are used in it to produce a more interesting rhythm.

```
function song()
{
    var out = 0;
    myClock.ticker();
    if( myClock.tick && myClock.playHead%beatPlay===0)
    {
        loop.trigger();
    }

    if(myClock.tick && parseInt(loopPlay/64)===1){

        beat.trigger();
    }
    if(myClock.tick && myClock.playHead%chordPlay===4){

        chord.trigger();
    }
 }
```

3D Graphics

- Basic perspective projection is used.
- The 3D model is a sphere of points with a ring number of 80.
- The x, y, and z coordinates are used to create the basic 3D

graphics.

```
var dim = 80; // This is the number of rings
// Each ring has as many points as there are rings8
// This is the spacing for each ring
var spacing = ((Math.PI * 5.5) / dim);
//var spacing = 200
//var spacing = Math.sin(dim)
// This is the total number of points
var numPoints = dim * dim;

// This is how big the sphere is.
var size = 500;
```

- CanvasRenderingContext2D.CreateRadialGradient() is used, and through in which access to the mouse parameters for interaction of graphics and the gradient effect.

```
    // Draw the point

    // Set the size based on scaling
      context.lineWidth = scale;
      var
grd=context.createRadialGradient(mouseX,mouseY,80,mouseX,m
ouseY,300);
```

```
            grd.addColorStop(0,"#FF3399");
            grd.addColorStop(1,"#33FFFF");
            context.strokeStyle = grd
            context.fillStyle = grd
          // context.strokeStyle = "rgb(0,255,255)";
            context.beginPath();
            context.moveTo(x2d, y2d);
            context.lineTo(x2d + scale, y2d);

            context.stroke();
```

The challenges of development projects

- The sound is sometimes distorted, which makes the mix often sound bad.
- Different browsers load different pages due to browser compatibility issues. My project was made in Safari, and the audio that had been uploaded could not load properly when I opened the link in Google Chrome.
- The sphere of points is so thin that it doesn't look obvious.

Future plan

I would like to create sphere of points that move to follow the rhythm of the sound. Like changing shape to follow the beat.