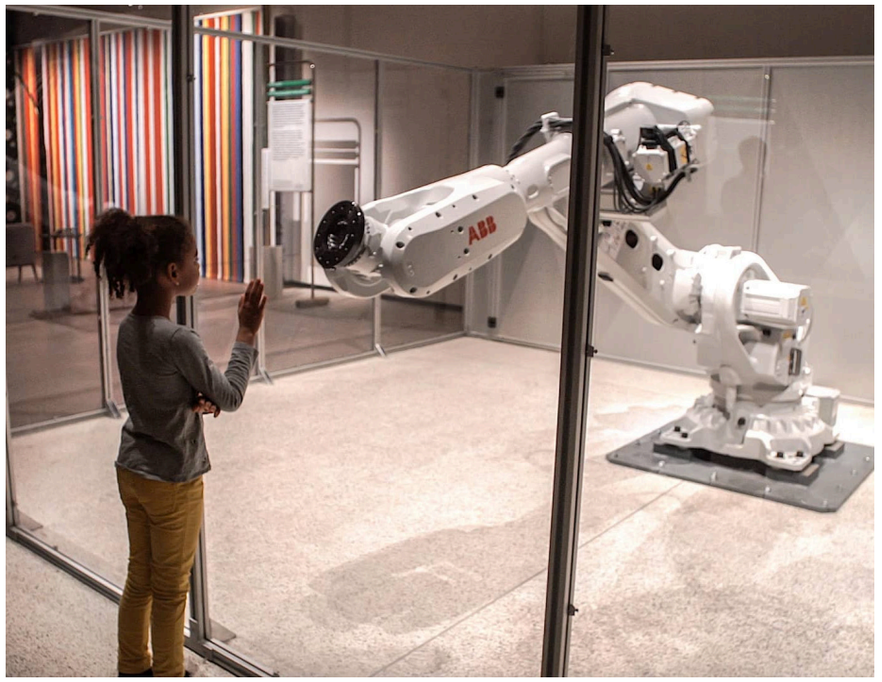
The Future Of Typography

Of Human and Machine

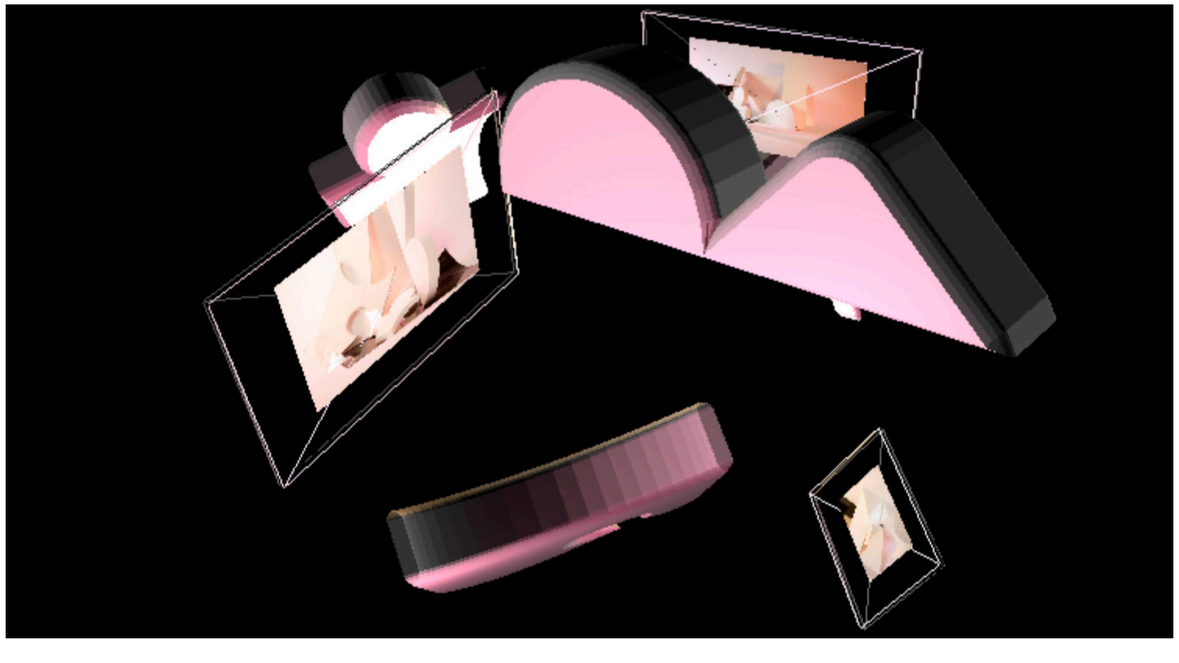
MINIMUS BY MADELINE GANNON

A robot created by Madeline Gannon that displays and exhibits behaviour that is intuitive in a way visitors do not anticipate. Its movement and demeanour feels human and visitors are able to understand and interact with it immediately and be drawn to it by these perceptive and human like interactions. Gannon believes that the robots of the future should not replace us but instead augment our own abilities and be designed with a focus on human interaction.



FEAR AND LOVE - DESIGN MUSEUM, LONDON

As Unbounded by Physical Rules

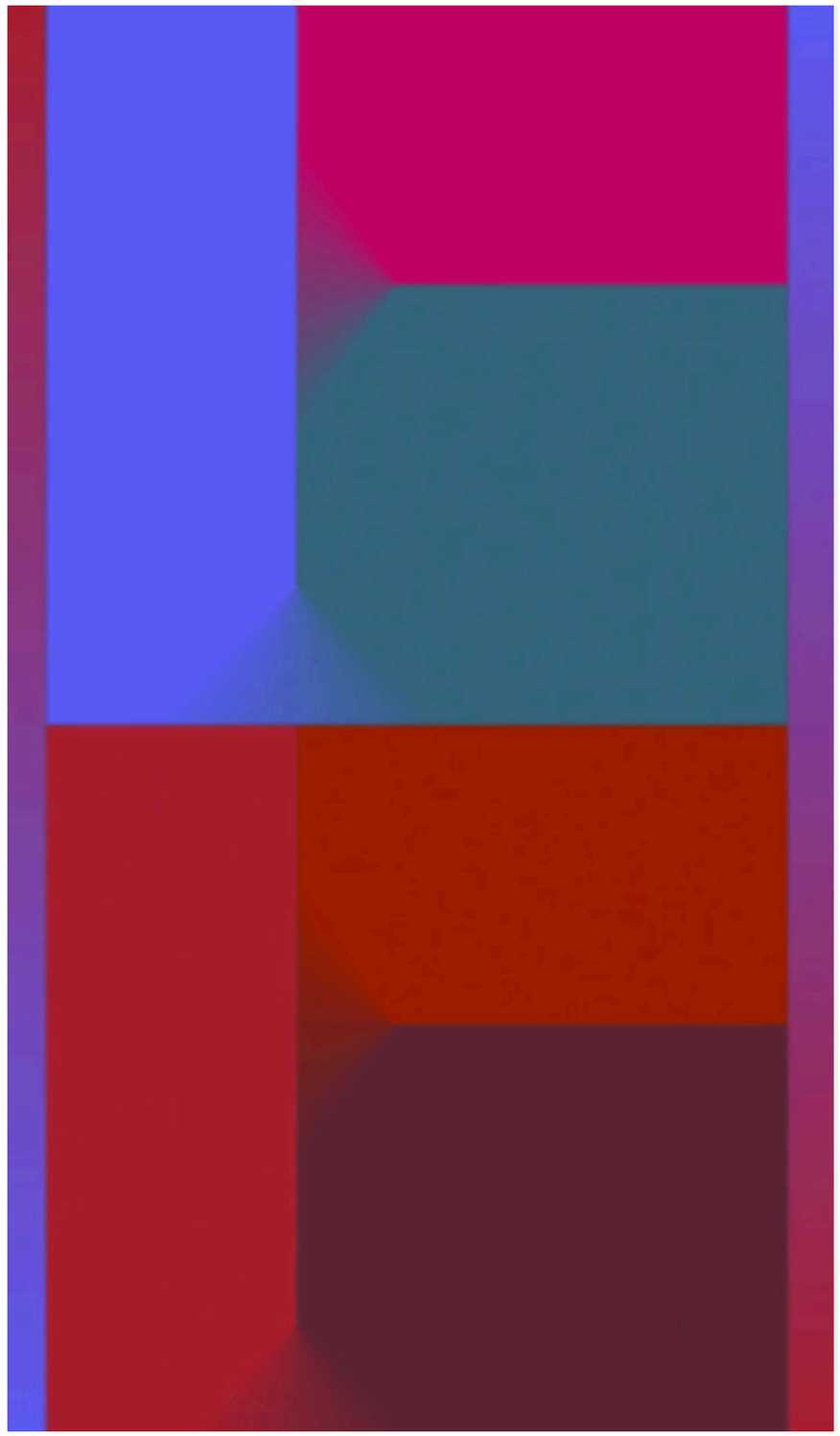


PROTOTYPE - 3 BY RAHUL SHINDE

An exploration of the ways in which the introduction of digital realities into daily has altered our methods for interacting with the world in which we live. In the past graphic design has functioned to bridge the gap between the digital and physical but limits have been constructed when digital spaces are modelled structures informed by physical reality.

The future of typography can embrace and create a world that bridges this gap but expands and explores new territory that is unbounded by the rules of our physical realities.

Never the Same Twice



REFLECTION BY BRIAN ENO

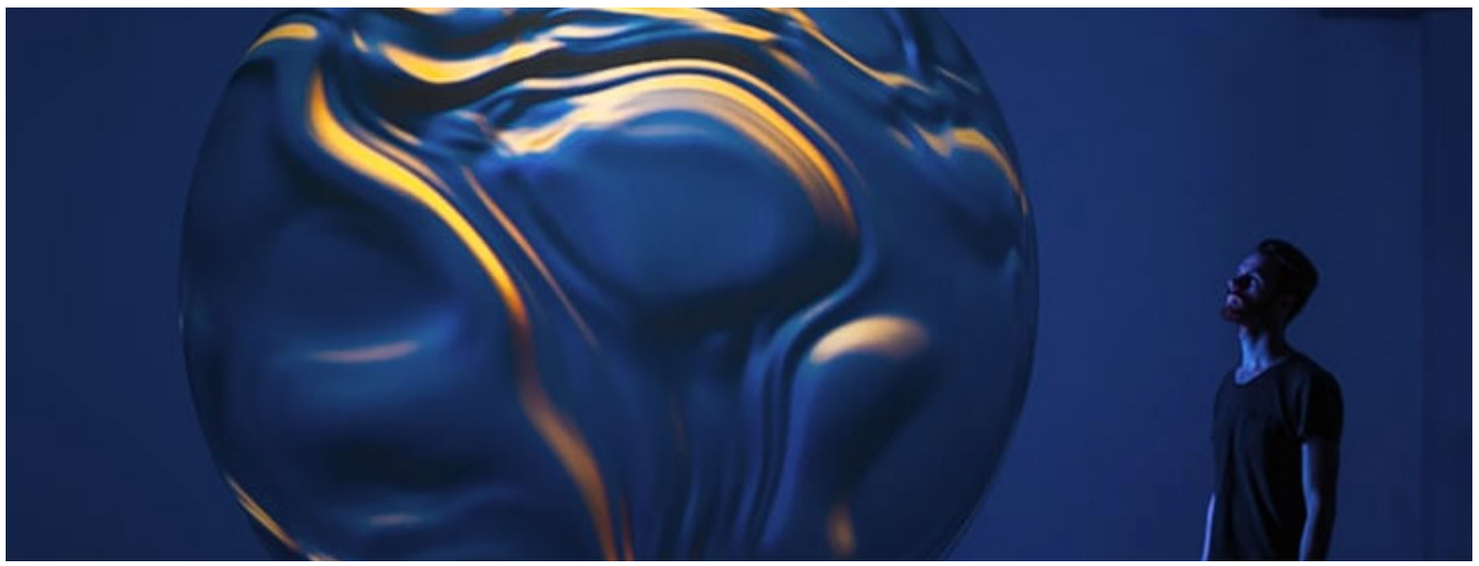
Light and Time: The Visual Art of Brian Eno - Interview

“Once you realise it is changing but it’s very slow, there’s a temptation to stick around to see what the next thing is going to be, and the next thing comes very slowly, gracefully and stays around a while”

Reflection is an album and app created by Eno. The sonic elements of the album play in entirely different formations every time you launch it. As with much of Eno’s work the visuals respond to and develop upon the sonic elements, creating a dialogue between the two.

“You start to experience a slower texture of time”

Unites Viewer and Atmosphere



ANIMA BY ONFORMATIVE STUDIO

A sculpture exploring the relationship between itself and its surroundings through the use of movement, texture, light and sound. The viewer is drawn in as the orb reacts to their presence. The orb is essentially “feasting on its environment to create an immersive and interactive experience of light and sound.” Here, there is a continuous exchange between the viewer and the entity which unites them in a infinite discourse through an interactive soundscape and visuals.

The future of typography is atmospheric and emotional.

The future of typography does not see human and machine as seperate entities.

The future of typography is the creator of new realities.

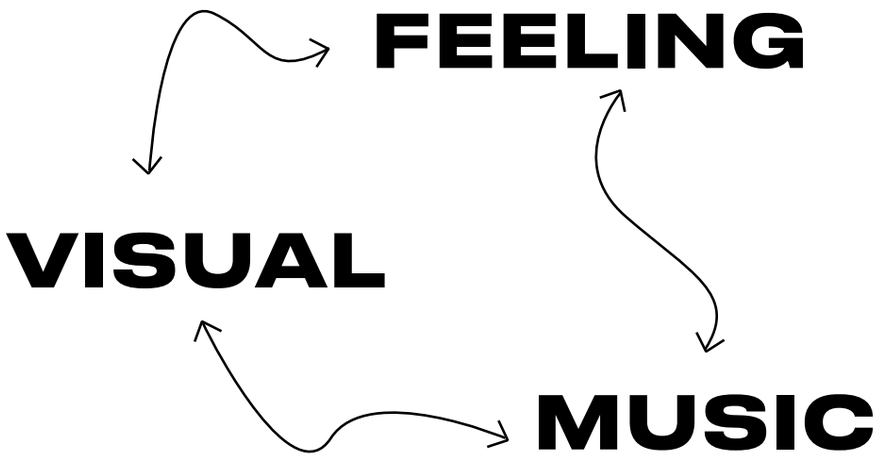
Design Proposal

THE FUTURE OF TYPOGRAPHY

The future of typography will be unmistakably neither human nor machine. It will be both at once. The combination of human intuition and the power of computational systems has created a typographic universe with capabilities far beyond anything letterforms have been capable of thus far. We are entering the age of type as a repository for and creator of meaning. It will be emotional, as a sensing, expressing, breathing organism that interacts with its environment in the same ways as us. The machine is not a cold thing, nor is it our enemy. It is part of us and we are part of it. The future of typography therefore, is found in these techno/human realities.

POSTER IDEA

The poster will act as a glimpse into this future of typography using music to bring together the user and the computer as one collaborative system continually in conversation. An infinite loop of interaction will be created in which the music playing will inform the visuals (type), feeling, and music will all constantly be informing one another. The continuous dialogue will be situated in a three dimensional environment that the viewer will be able to navigate and explore, moving through and around the typography and it shifts and changes.



Technologies / Parameters / Interaction

SOUND.JS

link sound to code

WEBGL IN P5.JS

reate a navigable 3D environment

PARAMETER		INFORMED BY
lighting color and brightness	———	spectral centroid of input signal
serifs of title font	———	amplitute of base frequencies
extenders on title font	———	aplitute of high-mid frequencies
rotation of small font orb thing	———	framrate and overall amplitute
speed of title pulse	———	overall frequency of song
colour scheme	———	spectral centroid (partially)

RANDOMLY* SELECTED

position of objects

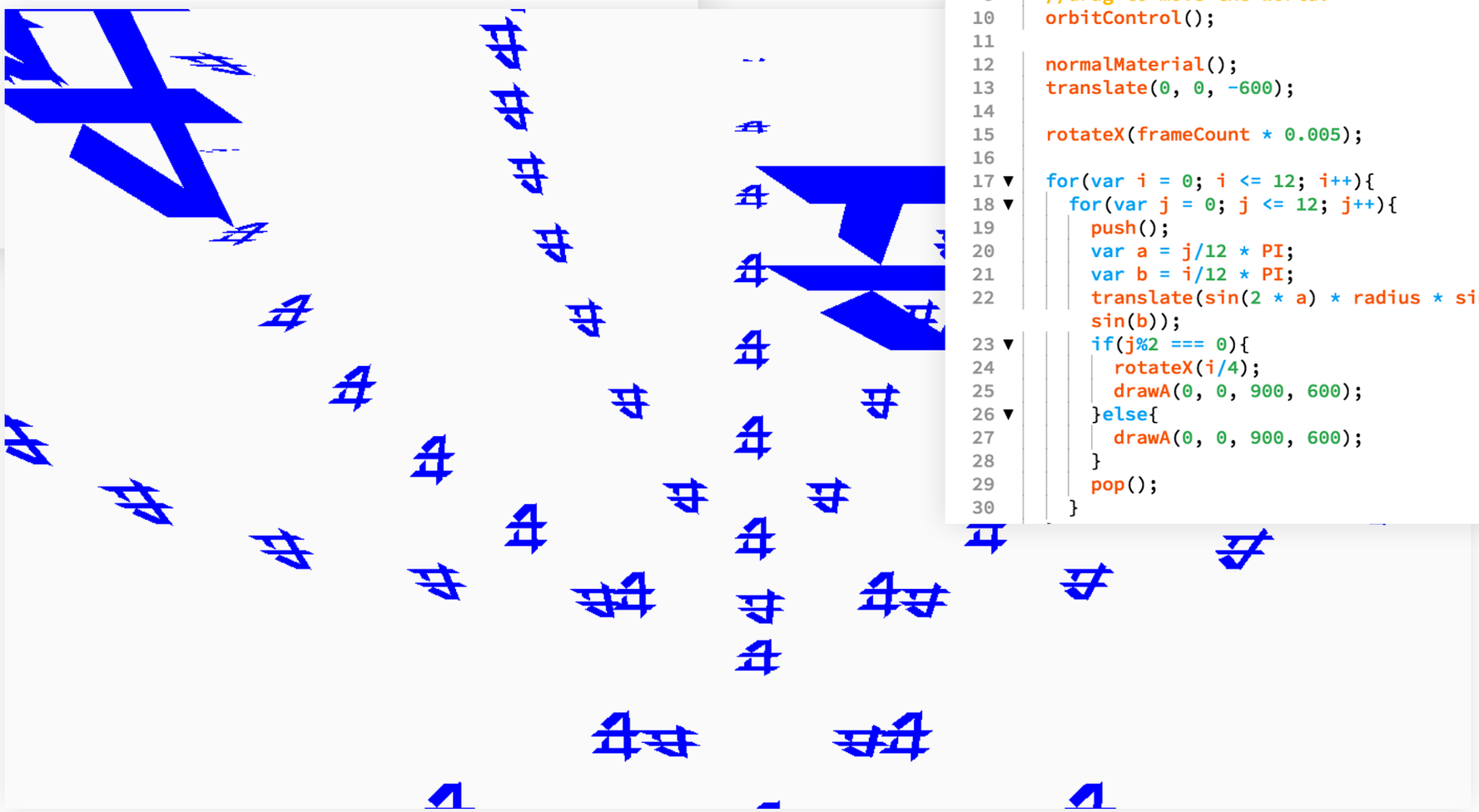
letter used

colour scheme (partially)

***ISH**

VIEWER SELECTED

frequncy of song	———	via slider
sound level of song	———	via slider
position of camera / view	———	zoom, mouse position and drag
colors	———	via slider (partially)



```
function setup(){
  createCanvas(710, 400, WEBGL);
}

function draw(){
  background(250);
  var radius = width * 1.5;

  //drag to move the world.
  orbitControl();

  normalMaterial();
  translate(0, 0, -600);

  rotateX(frameCount * 0.005);

  for(var i = 0; i <= 12; i++){
    for(var j = 0; j <= 12; j++){
      push();
      var a = j/12 * PI;
      var b = i/12 * PI;
      translate(sin(2 * a) * radius * sin(b), cos(b) * radius / 2 ,
      sin(b));
      if(j%2 === 0){
        rotateX(i/4);
        drawA(0, 0, 900, 600);
      }else{
        drawA(0, 0, 900, 600);
      }
      pop();
    }
```

```
song =  loadSound("assets/twiggles.mp3");
button = createButton("PLAY");
button.position(40, 40);
button.mousePressed(togglePlaying);
background(51);
fft = new p5.FFT(0.75, 1024);
fft2 = new p5.FFT(0.84, 1024);
amplitude = new p5.Amplitude();
angle = 0;
}

function draw() {
push();
graddy3();
pop();
orbitControl();
camera(camx, camy, zoom);

if (mouseX > width-width/10){
  camx = camx + 10;
} else if (mouseX < width-width/10){
  camx = camx + 0;
}
if (mouseX < 0+width/10){
  camx = camx - 10;
} else if (mouseX < width-width/10){
  camx = camx + 0;
}
if (mouseY > height-height/10){
  camy = camy + 10;
} else if (mouseY < height-height/10){
  camy = camy + 0;
}
if (mouseY< 0+height/10){
  camy = camy - 10;
} else if (mouseY < height-height/10){
  camy = camy + 0;
}

var spectrum = fft.analyze();
var spectrum2 = fft2.analyze();

var highs = fft.getEnergy("highMid");
var lows = fft2.getEnergy("bass");

var hey = map(highs, 0, 256, 50, height + 370);
var hey2 = map(lows, 0, 256, 0, width + 100);

var level = amplitude.getLevel();
var size = map(level, 0, 1, 0, 20);
var size2 = map(level, 0, 1, 0, 200);


var locY = (mouseY) * (-2);
var locX = (mouseX) * 2;
var posy = map(level, 0, 1, 0, 40);

translate(0, 0, 100);

push();
scale(1, 1.5, 1);
graddy();
pop();

push();
```