PROCESSING:
ASSIGNMENT C (due in wk 12)
create a sketch that illustrates a an element of the sketch you have chosen

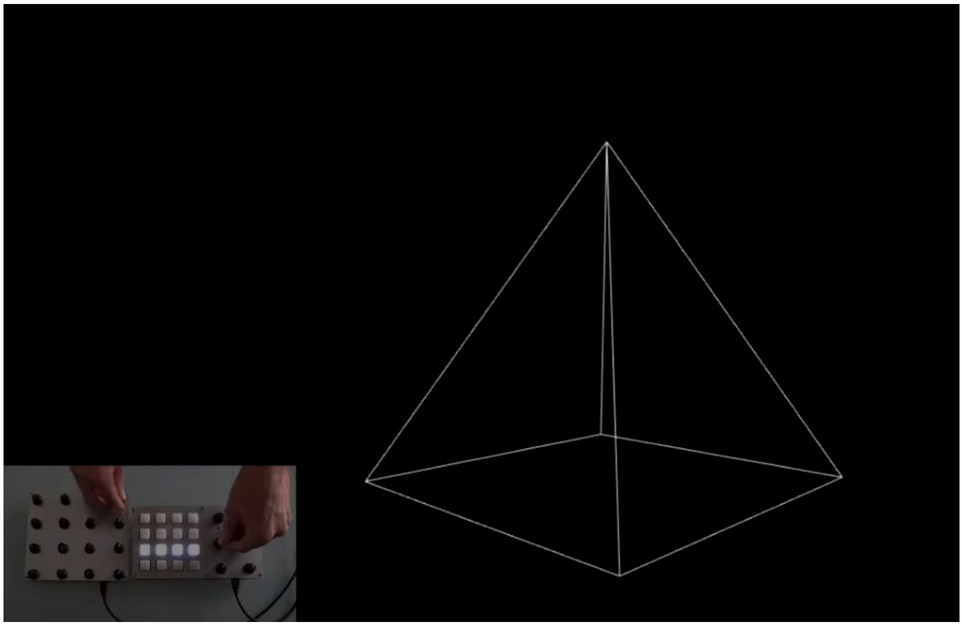
VIDEO EXAMPLE:
https://www.youtube.com/watch?v=gWU7VIrzYIQ&list=LLPcK9yGO3RsAzg17cPJV3WA&index=8

What is the concept?
The concept is based around monochromatic visuals. Shapes that graduate from 2D into 3D. For example a square will turn into a cuboid that rotates to show its dimension. That shapes will also move to the music. When the music gets to a higher intensity, a scribble is introduced and changes dimension/colour according to the music.

How was it created?
It was created using Processing and Ableton. It uses 2 midi controllers, both 4x4 controllers. One with just knobs and one with just pads.

How does it work?
One midi controller (the one with pads) will control the visuals, their dimensions, shapes, colours. The other midi controller (the one with knobs) will control the music, hooked up with Ableton.

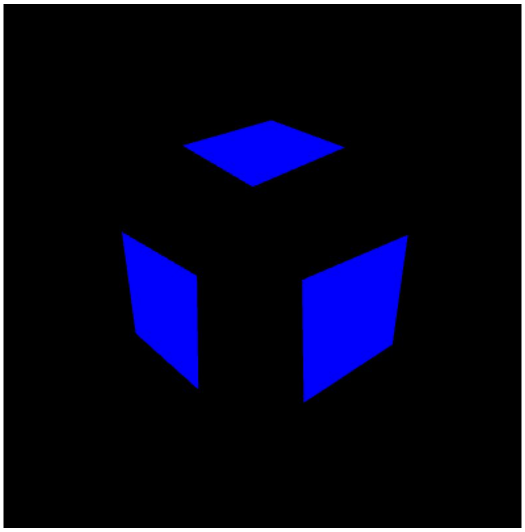
Why Do I like the Visuals?
I like these visuals because they are minimalistic yet effective. They remind me of the Arctic Monkeys MV visuals, which has a similar monochromatic feel. The visuals also offer density and texture. I feel it is also easy to watch and as a concept it can branch out into many different ways to create other visuals.



PDE:

Sketch 1:

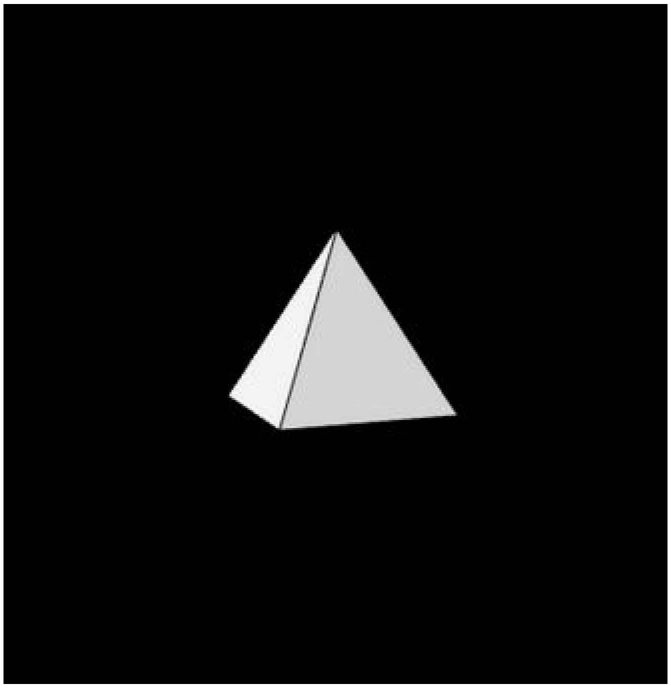
```
float xmag, ymag = 0;
float newXmag, newYmag = 0;
void setup()
{
  size(960, 540, P3D);
  noCursor();
  stroke(0);
}
void draw()
{
  background(0);
  pushMatrix();
  translate(width/2, height/2, -30);
   if (mousePressed){
  newXmag = mouseX/float(width) * TWO_PI;
  newYmag = mouseY/float(height) * TWO_PI;
 }
  float diff = xmag-newXmag;
  if (abs(diff) >  0.01) { xmag -= diff/4.0; }
  diff = ymag-newYmag;
  if (abs(diff) >  0.01) { ymag -= diff/4.0; }
  rotateX(-ymag);
  rotateY(-xmag);
  scale(80);
  beginShape(QUADS);
  fill(255-mouseX, 255-mouseY, mouseX); vertex(-1,  1,  1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex( 1,  1,  1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex( 1, -1,  1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex(-1, -1,  1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex( 1,  1,  1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex( 1,  1, -1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex( 1, -1, -1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex( 1, -1,  1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex( 1,  1, -1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex(-1,  1, -1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex(-1, -1, -1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex( 1, -1, -1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex(-1,  1, -1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex(-1,  1,  1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex(-1, -1,  1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex(-1, -1, -1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex(-1,  1, -1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex( 1,  1, -1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex( 1,  1,  1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex(-1,  1,  1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex(-1, -1, -1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex( 1, -1, -1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex( 1, -1,  1);
  fill(255-mouseX, 255-mouseY, mouseX); vertex(-1, -1,  1);
  endShape();
  popMatrix();
}
```



With reference to : “Flight With Dogma”
http://www.openprocessing.org/sketch/6181

Sketch 2:

```
void setup() {
  size(600, 600, P3D);
}

void draw() {
  background(0);
  pushMatrix();

  translate(width/2, height/2);
  rotateY(map(mouseX, 0, width, 0, 0.2*PI));

fill(255,200,200);
  beginShape();
  vertex(0, 0,0);//top
  vertex(50, 100, 50);
  vertex(-50, 100, 50);
  endShape();

fill(210,255,200);
  beginShape();
  vertex(0, 0, 0);
  vertex(50, 100, 50);
  vertex(50, 100, -50);
  endShape();

fill(200,255,255);
  beginShape();
  vertex(0, 0,0);
  vertex(-50, 100, 50);
  vertex(-50, 100, -50);
  endShape();

  popMatrix();
}
```



In reference to : “Joori Lee”
http://www.openprocessing.org/sketch/96935