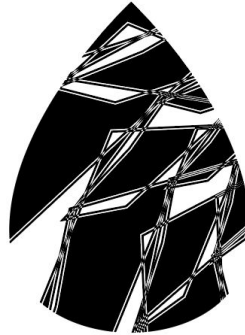


FNAR 264
DAVID COMBERG

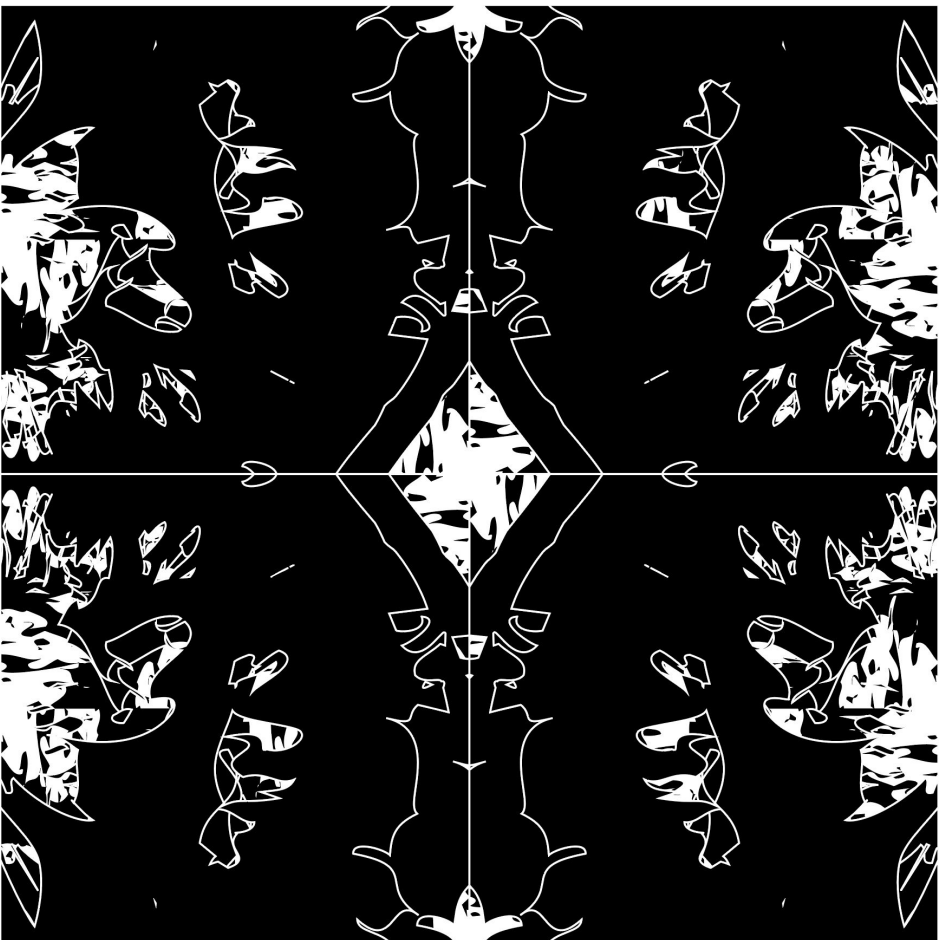


ELIZABETH POWELL
FINAL PORTFOLIO

C
R
I
C
K
E
T
S

O
N

L
O
C
U
S
T



C
A
P
T
U
R
I
N
G

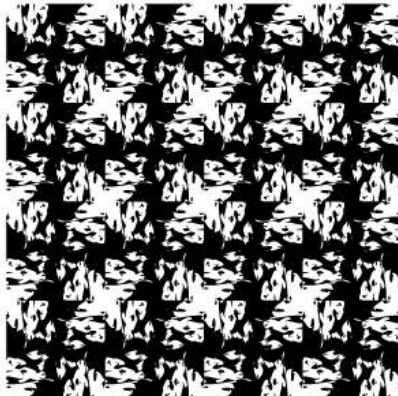
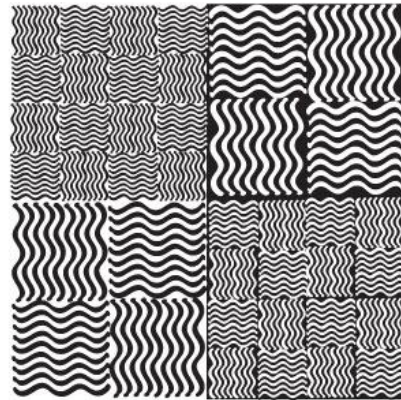
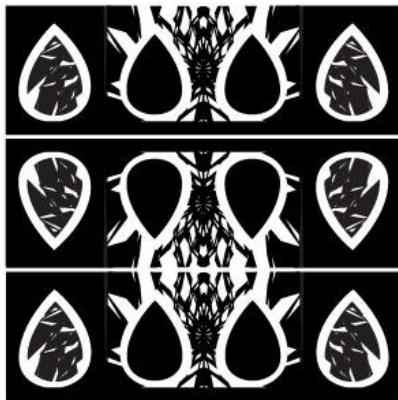
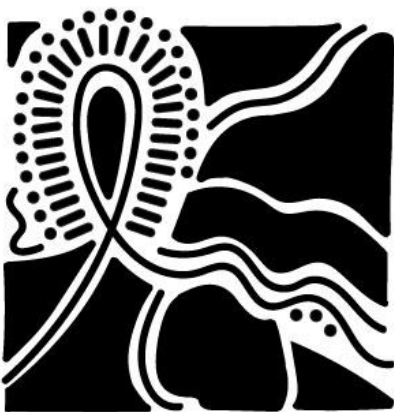
A

S
O
U
N
D

C
A
P
T
U
R
I
N
G

A

S
O
U
N
D



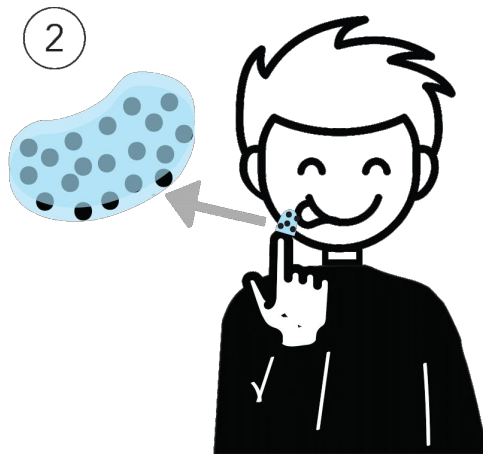
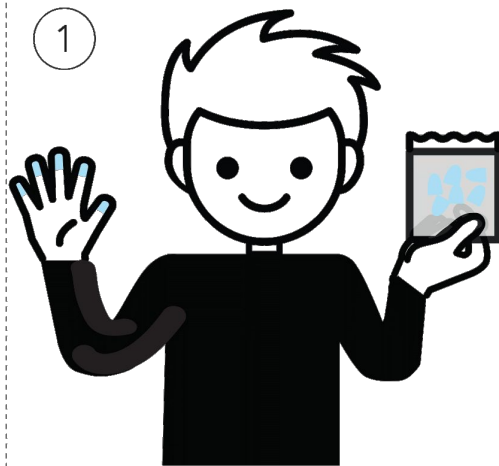
Elizabeth Powell
and the phonetic 'word' of the sound of crickets outdoor

C
R
I
C
K
E
T
S

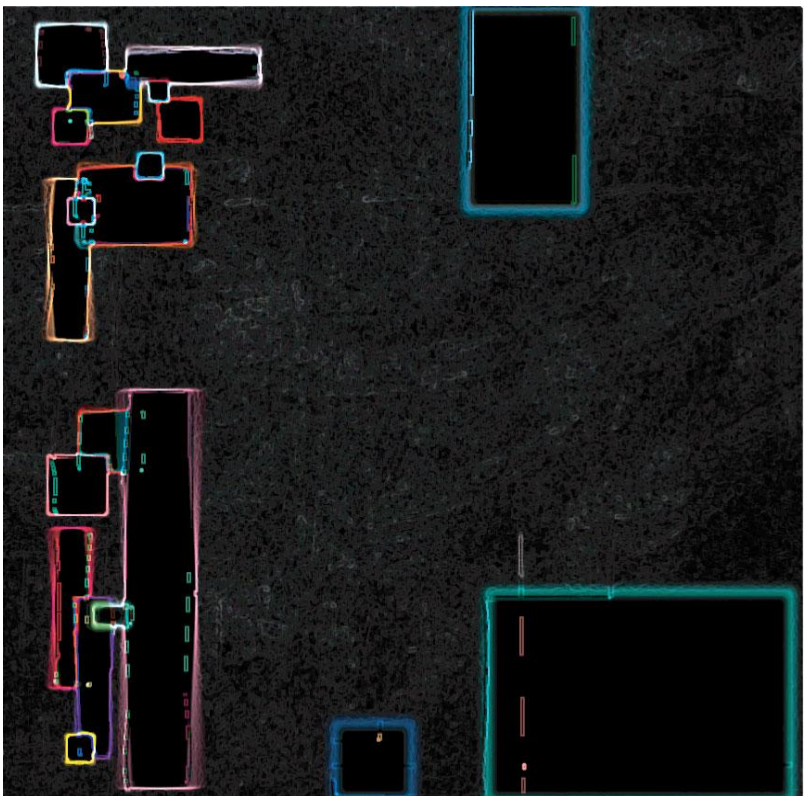
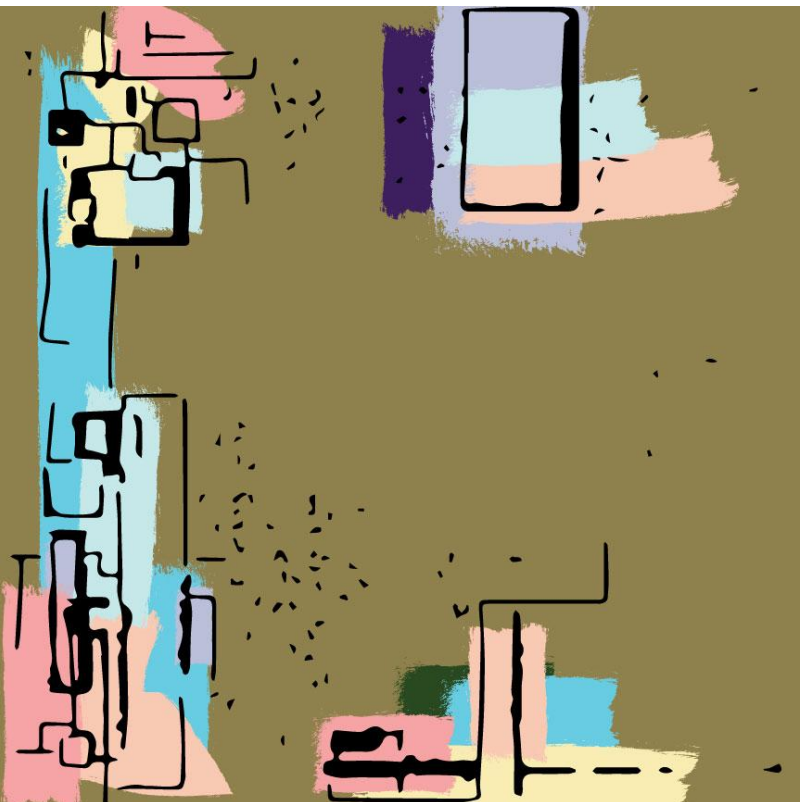
O
N

L
O
C
U
S
T

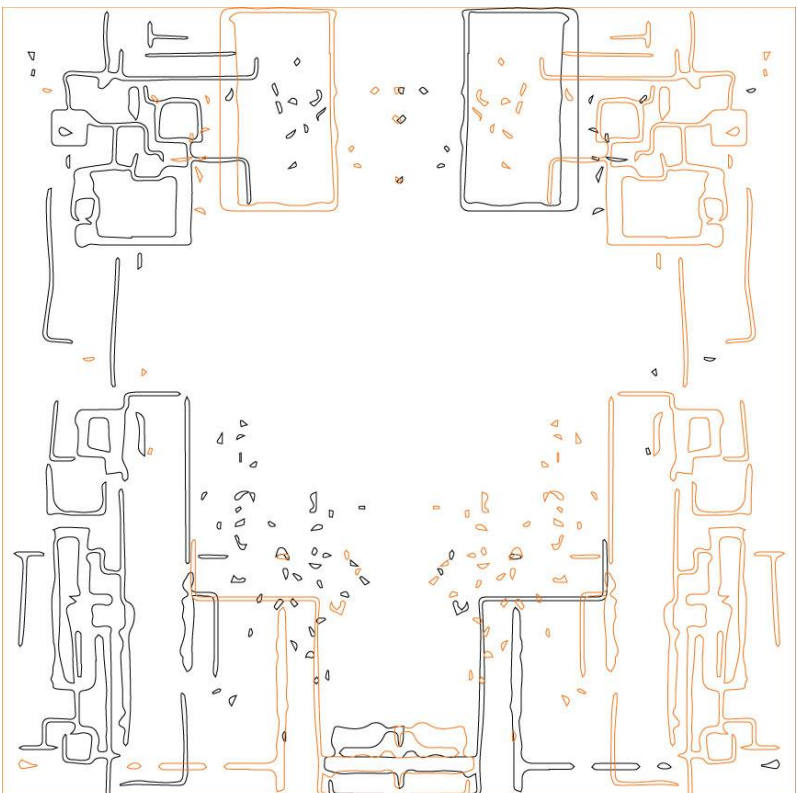
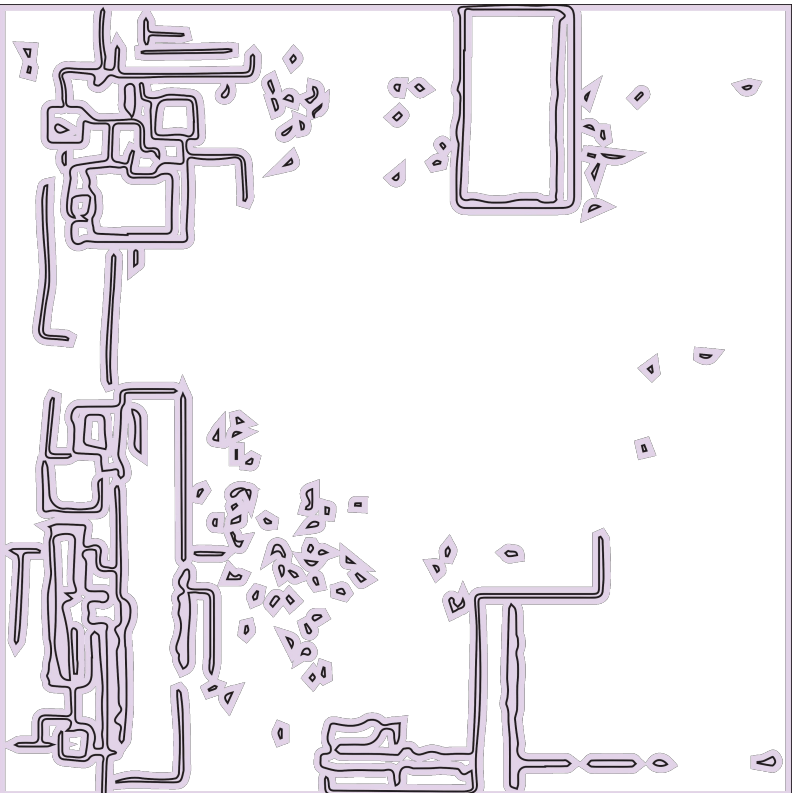
Snäck Shiëld



INVISIBLE CITIES

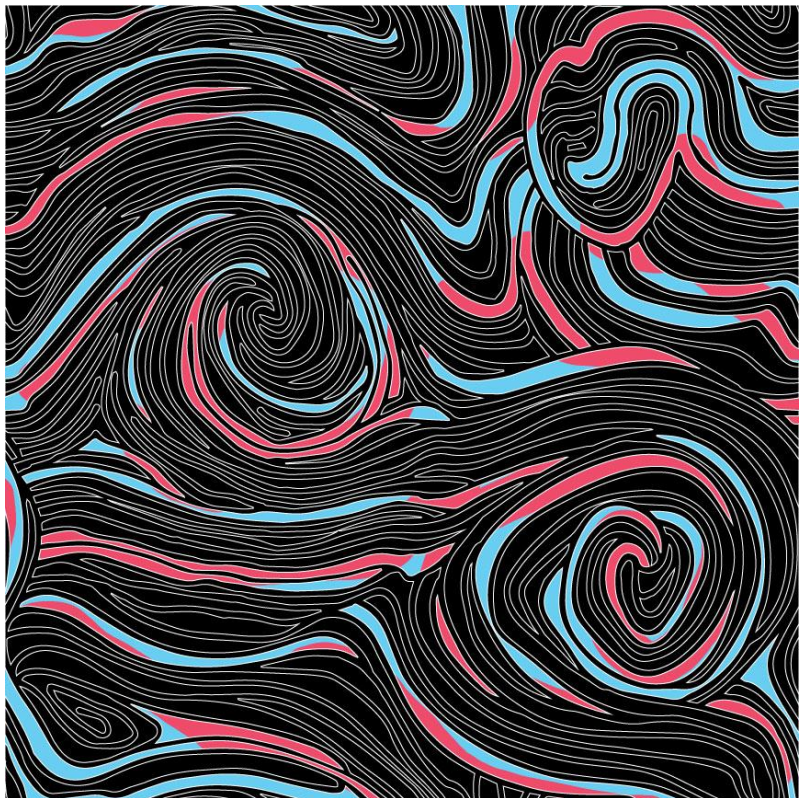


INVISIBLE CITIES

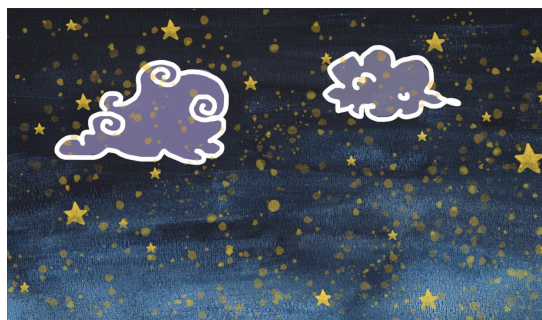
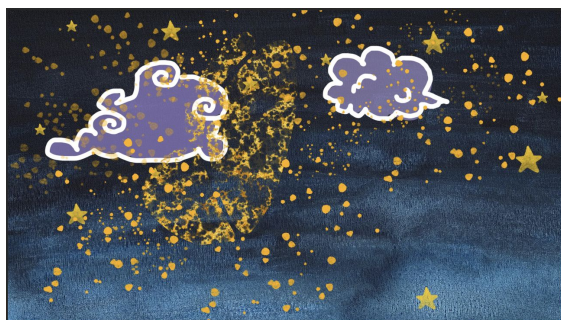
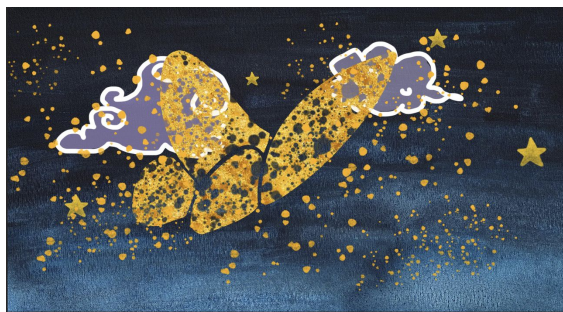
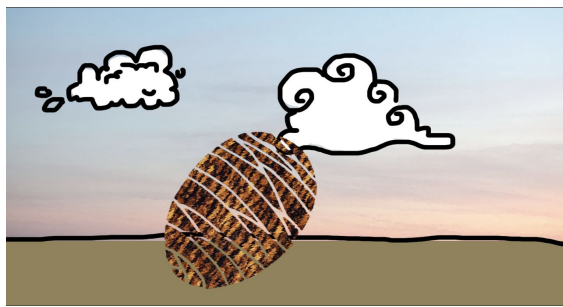
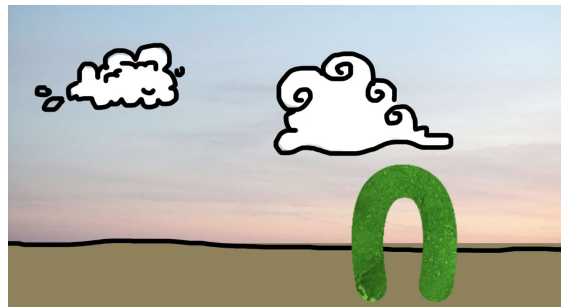


I
N
V
I
S
I
B
L
E

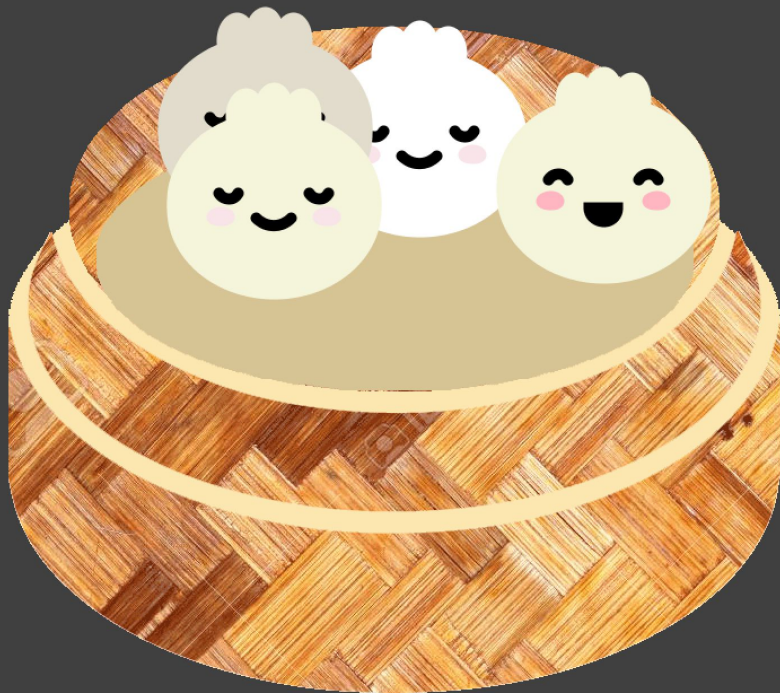
C
I
T
I
E
S



NOTHING GOLD CAN STAY



LEARNING PROCESSING



```
color whites[] = {color(245, 245, 220), color(226, 220, 205), color(255, 255, 255)};
color bodyColor[] = new int[numDumplings];
PImage img;

void drawSoupDumpling(int x, int y, int size, float rand, color c) {
  fill(c); //white
  stroke(c);
  ellipse(x, y, size + 30, size);
  ellipse(x, y + 7, size + 30, size);

  ellipse(x - 25, y - 75, 25, 40);
  ellipse(x, y - 80, 25, 40);
  ellipse(x + 25, y - 75, 25, 40);
  drawFace(rand, x, y);
}

void drawFace(float rand, int x, int y){
  if (rand <= 1.0) {
    drawHappyFace(x, y);
  } else if (rand <= 2.0) {
    drawShyFace(x, y);
  } else {
    drawLaughingFace(x, y);
  }
}

void drawHappyFace(int x, int y) {
  fill(0);
  stroke(0);
  ellipse(x - 40, y - 10, 13, 13);
  ellipse(x + 40, y - 10, 13, 13);

  fill(244,194,194);
  stroke(244,194,194);
  ellipse(x - 47, y + 14, 15, 7);
  ellipse(x + 47, y + 14, 15, 7);

  noFill();
  stroke(0);
  strokeWeight(10);
  arc(x, y, 45, 45, PI / 4, 3 * PI / 4, OPEN);
}

void drawShyFace(int x, int y) {
  fill(249,228,233);
  stroke(249,228,233);
  ellipse(x - 47, y + 14, 15, 7);
  ellipse(x + 47, y + 14, 15, 7);

  noFill();
  stroke(0);
  strokeWeight(10);
  arc(x, y, 45, 45, PI / 4, 3 * PI / 4, OPEN);

  arc(x - 40, y - 12, 20, 20, PI / 6, 5 * PI / 6, OPEN);
  arc(x + 40, y - 12, 20, 20, PI / 6, 5 * PI / 6, OPEN);
}
```

GENERATING WEARABLE DATA

```
void setup() {
  color[] colors = {#581845, #900c3f, #c70039, #ff5733, #ffc300, #f9564f}; //sunny, windy, rainy, cloudy, cold, beautif
  table = loadTable("small_data_set.csv", "header");
  background(#83E5FC);
  size(1000, 1000);

  rands = new float[table.getRowCount()];
  startPositions = new float[table.getRowCount()];

  for (int i = 0; i < table.getRowCount(); i++) {
    rands[i] = random(15);
    startPositions[i] = random(PI);
  }

  for (int i = 0; i < 200; i++) {
    wave(0, 0, colors[(int) random(6)]);
  }
}

void wave(int startx, int starty, color stroke_color) {
  stroke(stroke_color);
  strokeWeight(30);
  float inc = TWO_PI/250;

  float rand = random(8);
  rotate(rand);

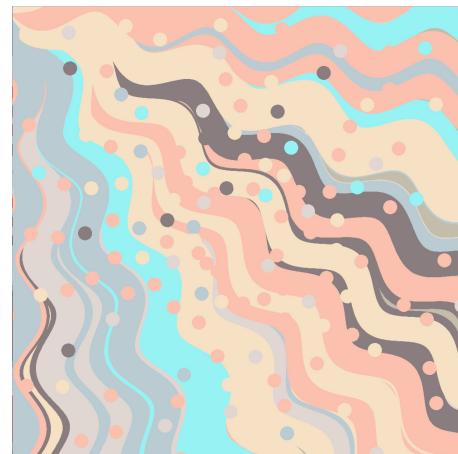
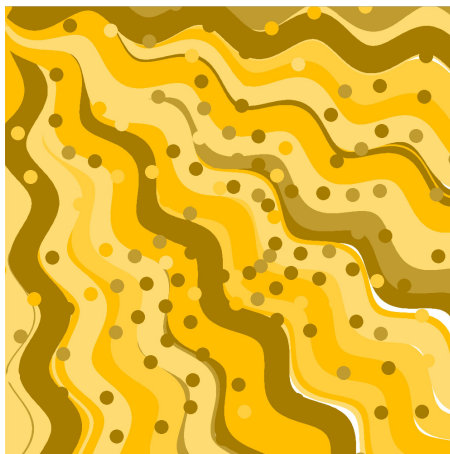
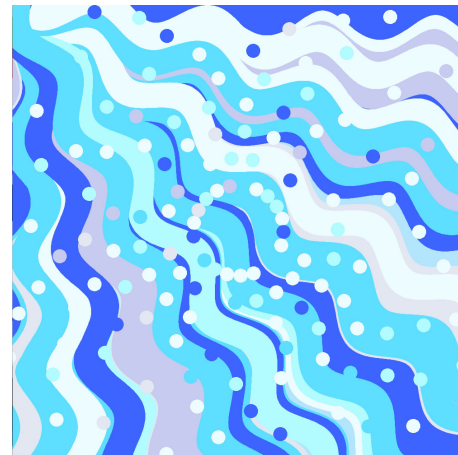
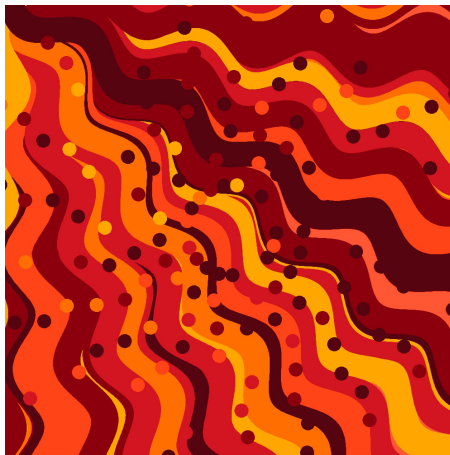
  float a = 0.0;
  for (int i = 0; i < 1500; i++) {
    line(startx + i, starty+sin(a)*30.0 - 0.1, startx + i, starty+sin(a)*30.0);
    line(startx + i, starty+sin(a)*30.0 - 10.1, startx + i, starty+sin(a)*30.0 - 10);
    line(startx + i, starty+sin(a)*30.0 - 20.1, startx + i, starty+sin(a)*30.0 - 20);
    line(startx + i, starty+sin(a)*30.0 - 30.1, startx + i, starty+sin(a)*30.0 - 30);
    a = a + inc;
  }

  rotate(-rand);
}

void draw() {
  data = new int[table.getRowCount()][3];
  color[] colors = {#581845, #900c3f, #c70039, #ff5733, #ffc300, #f9564f}; //sunny, windy, rainy, cloudy, cold, beautif
  strokeWeight(10);

  for (int i = 0; i < table.getRowCount(); i++) {
    TableRow row = table.getRow(i);

    int day = row.getInt("day");
    int miles = row.getInt("miles");
    double pace = row.getDouble("pace");
    String weather = row.getString("weather");
  }
}
```



GENERATING WEARABLE DATA

```
println(day + " " + miles + " " + pace + " " + weather);  
data[i][0] = miles;  
data[i][1] = (int) pace;  
switch (weather) {  
case "sunny":  
    data[i][2] = 0;  
    break;  
case "windy":  
    data[i][2] = 1;  
    break;  
case "rainy":  
    data[i][2] = 2;  
    break;  
case "cloudy":  
    data[i][2] = 3;  
    break;  
case "cold":  
    data[i][2] = 4;  
    break;  
case "beautiful":  
    data[i][2] = 5;  
    break;  
default:  
    break;  
}  
}  
  
float unit = 2 * PI / data.length;  
for (float radius = 100; radius <= 600; radius += 60){  
    float angle = startPositions[(int) (radius - 100) / 60];  
    for (int i = 0; i < data.length; i++) {  
        int miles = data[i][0];  
        int pace = data[i][1];  
        color c = data[i][2];  
  
        float x = (radius) * cos(angle);  
        float y = (radius) * sin(angle);  
  
        if (miles != 0) {  
            stroke(colors[c]);  
            fill(colors[c]);  
            circle(x + midx, y + midy, 20);  
        }  
  
        angle += unit;  
    }  
}
```



