

Grundlagen im medialen Raum

Abschlusspräsentation

Christian Kolb, Lukas Geiger, Patric Richardson

Internet der Dinge 2.Semester

Hochschule für Gestaltung
Schwäbisch Gmünd

Sommersemester 2019

Music choice & information

Musikstück: The XX Intro



Der **Spiegel** schrieb „erhabener Melancholie, aber auch majestätischer Intimität“.

Der **Telegraph** bezeichnete Intro als „mysteriös“.

Music choice & information

Intro ist ein Instrumental-Lied der britischen Indie Band The XX.

Wurde von den Bandmitglieder Romy Madley Croft, Olivier Sim, Baria Qureshi und Jamie xx komponiert und produziert.

The XX wurde an der Londoner *Elliott School* gegründet.

Romy Madley Croft – Gitarre

Oliver Sim – Bass

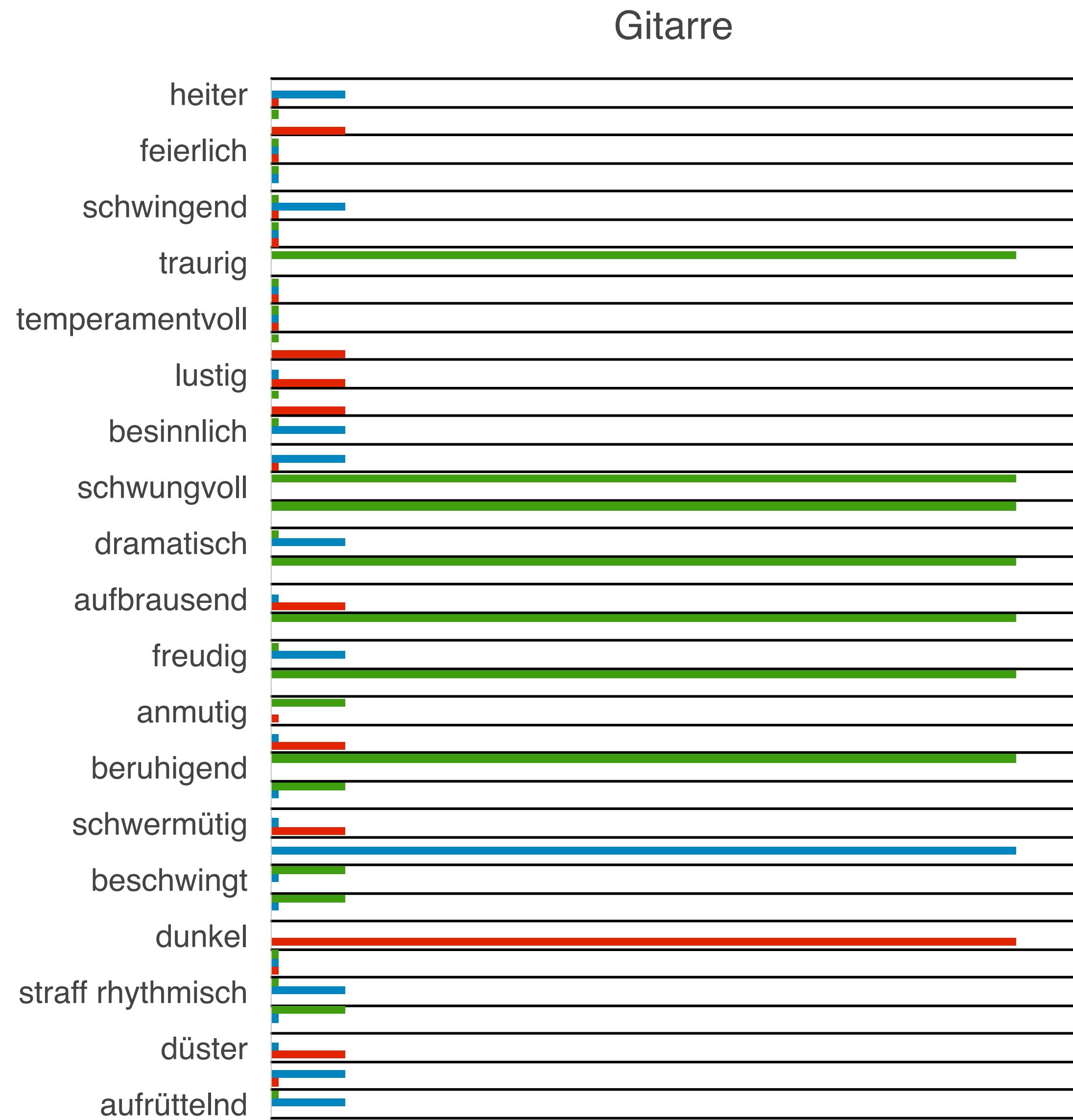
Jamie Smith – Beats, Abmischung, MPC, Musikproduzent

Baria Qureshi – Gitarre, Keyboards

Rodaidh McDonald – Abmischung

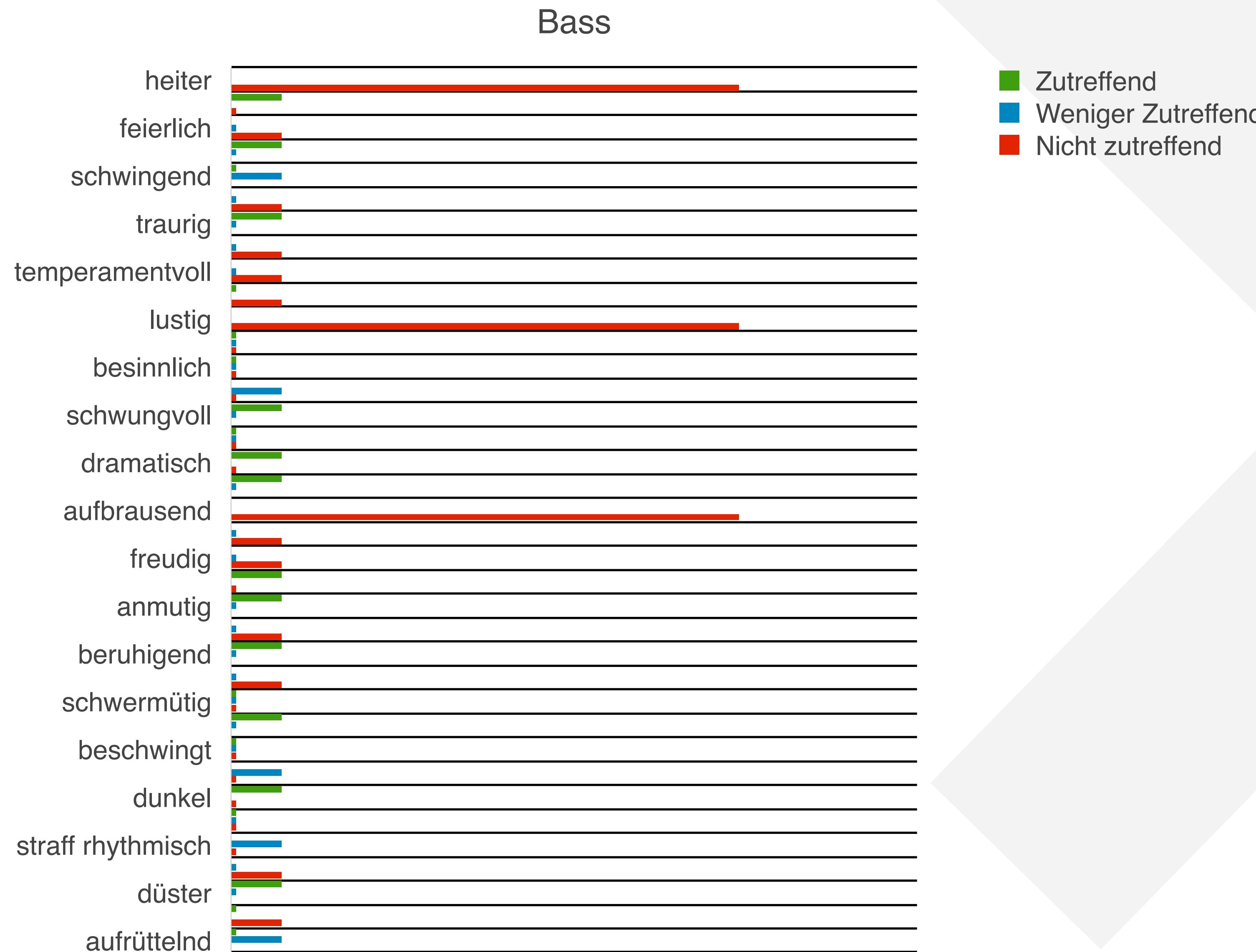
Nilesh Patel – Mastering

Semantic differential

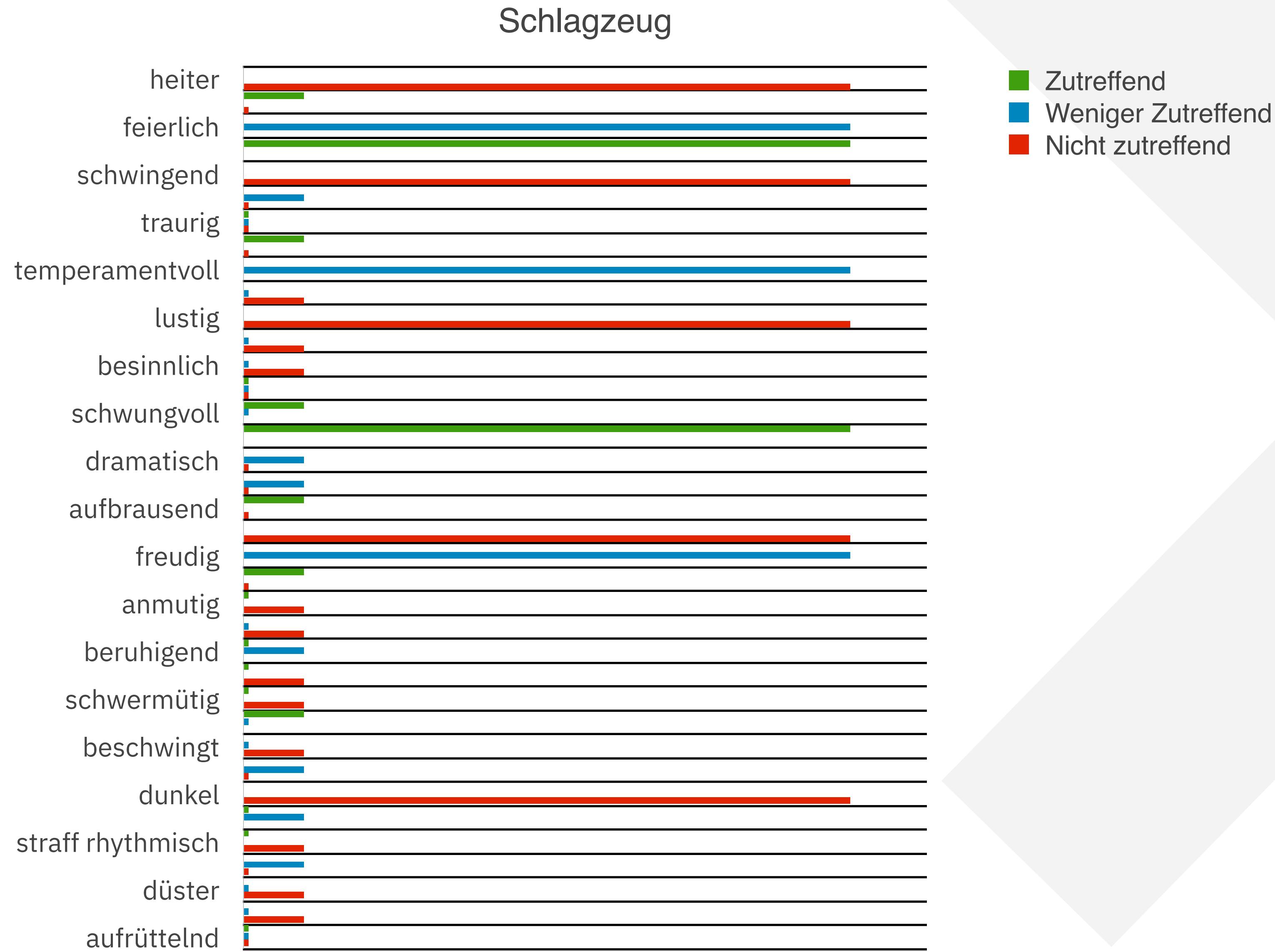


- Zutreffend
- Weniger Zutreffend
- Nicht zutreffend

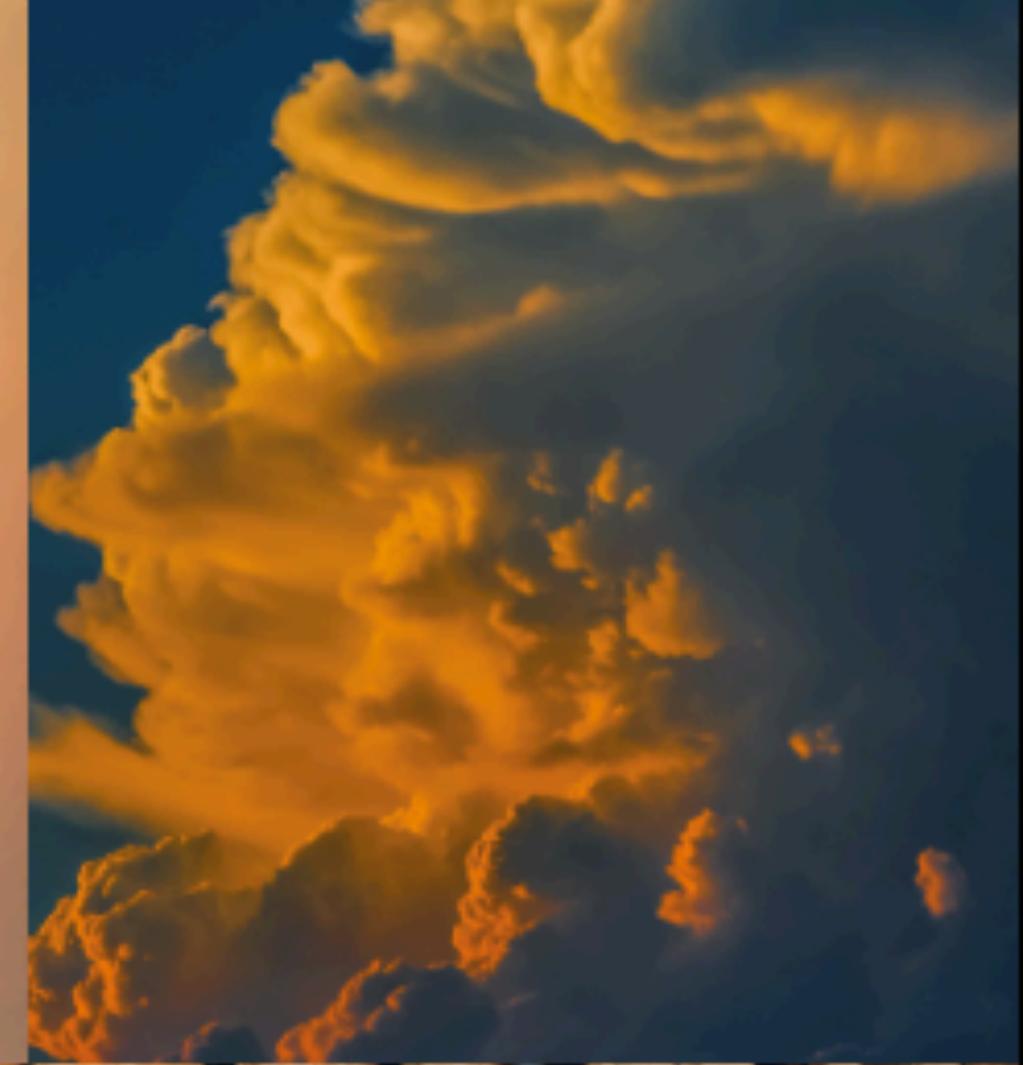
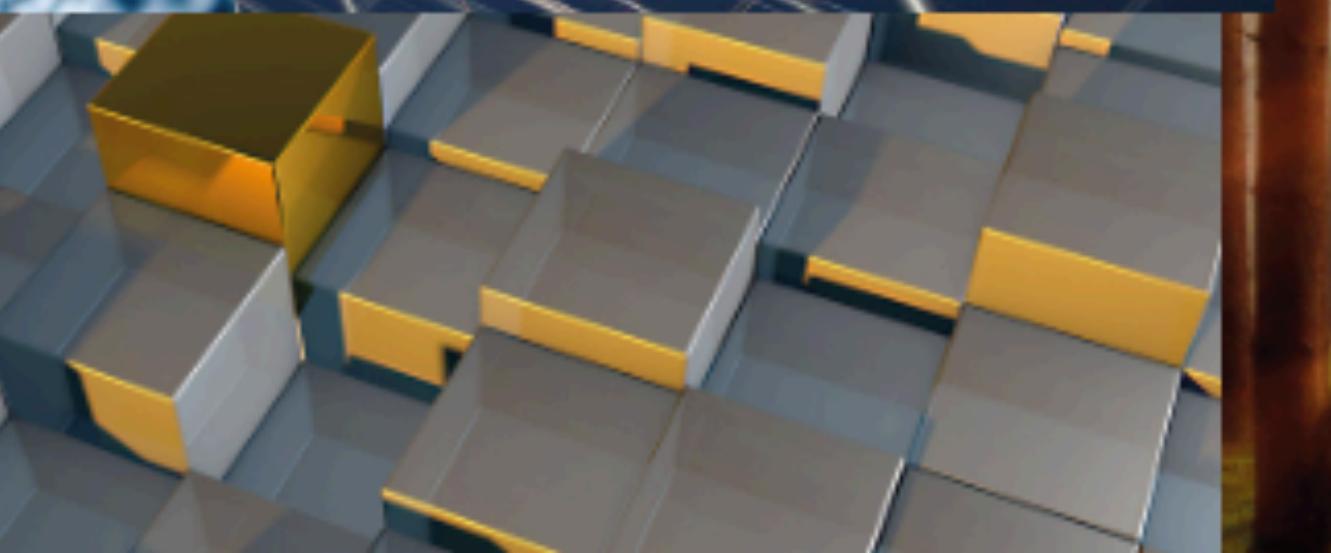
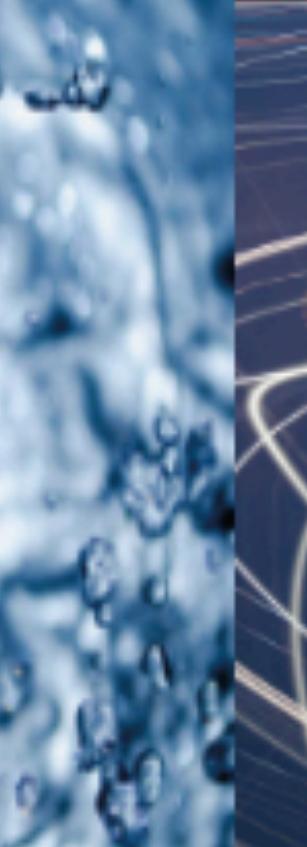
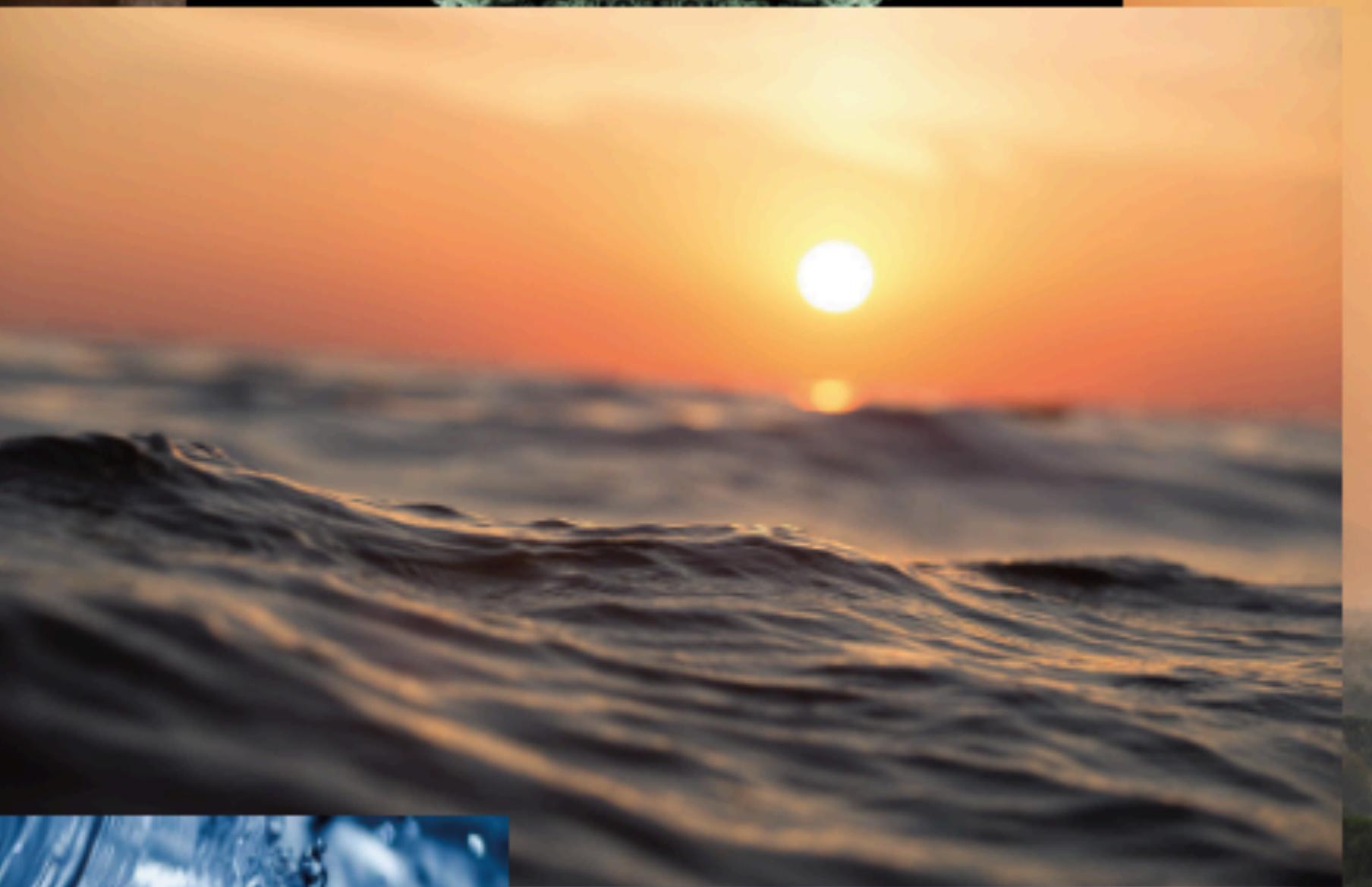
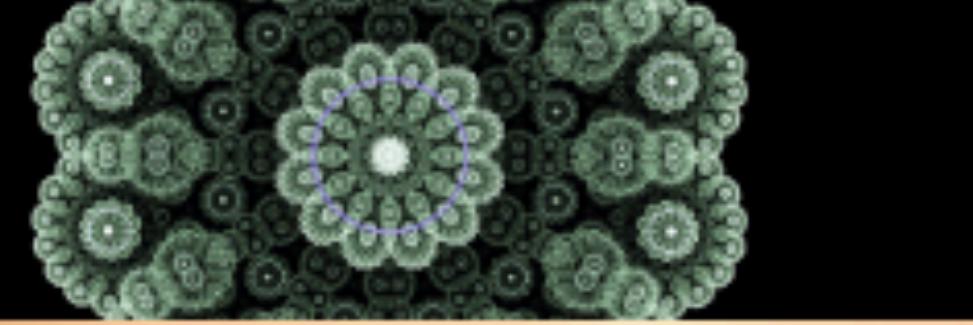
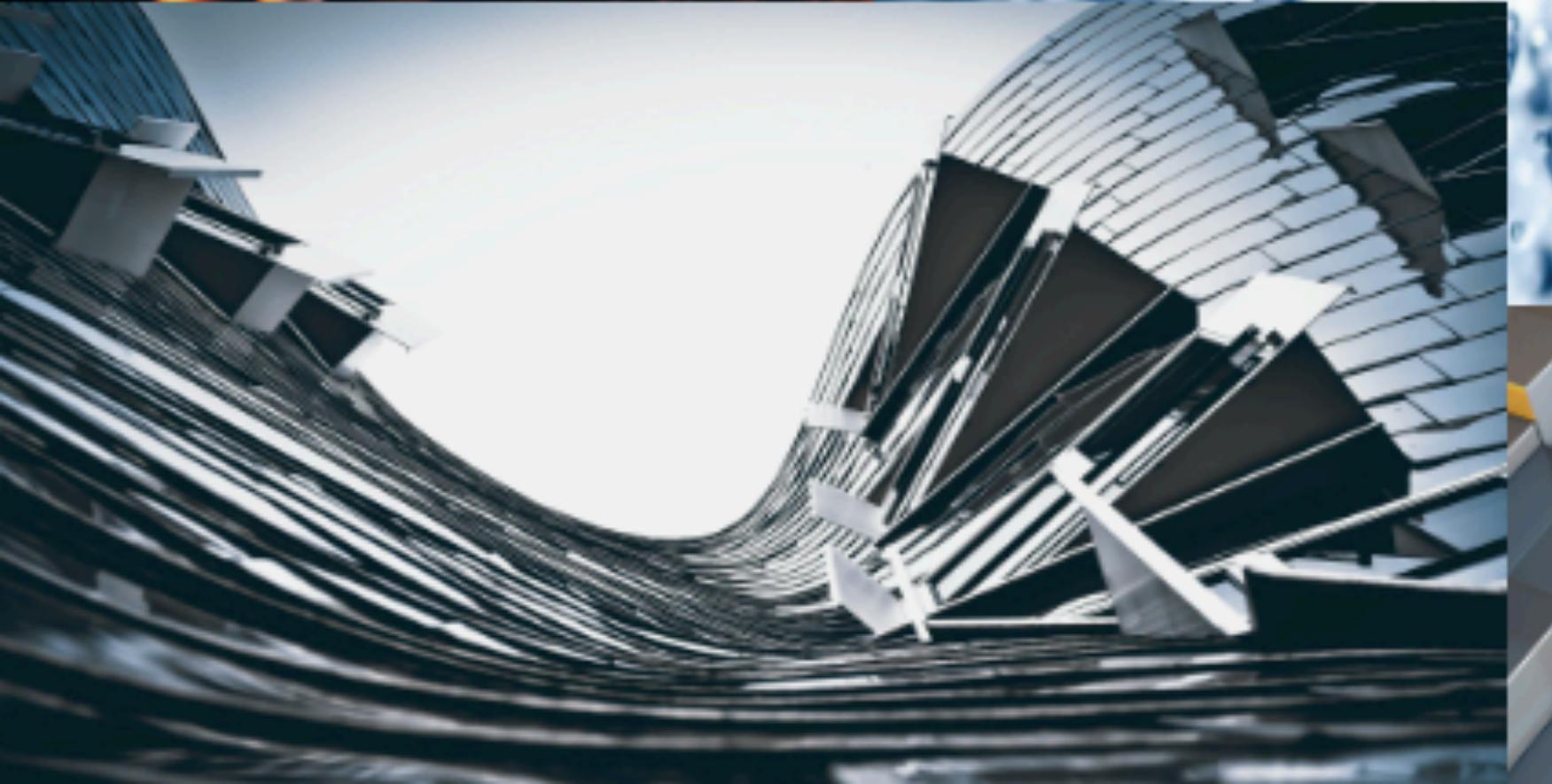
Semantic differential

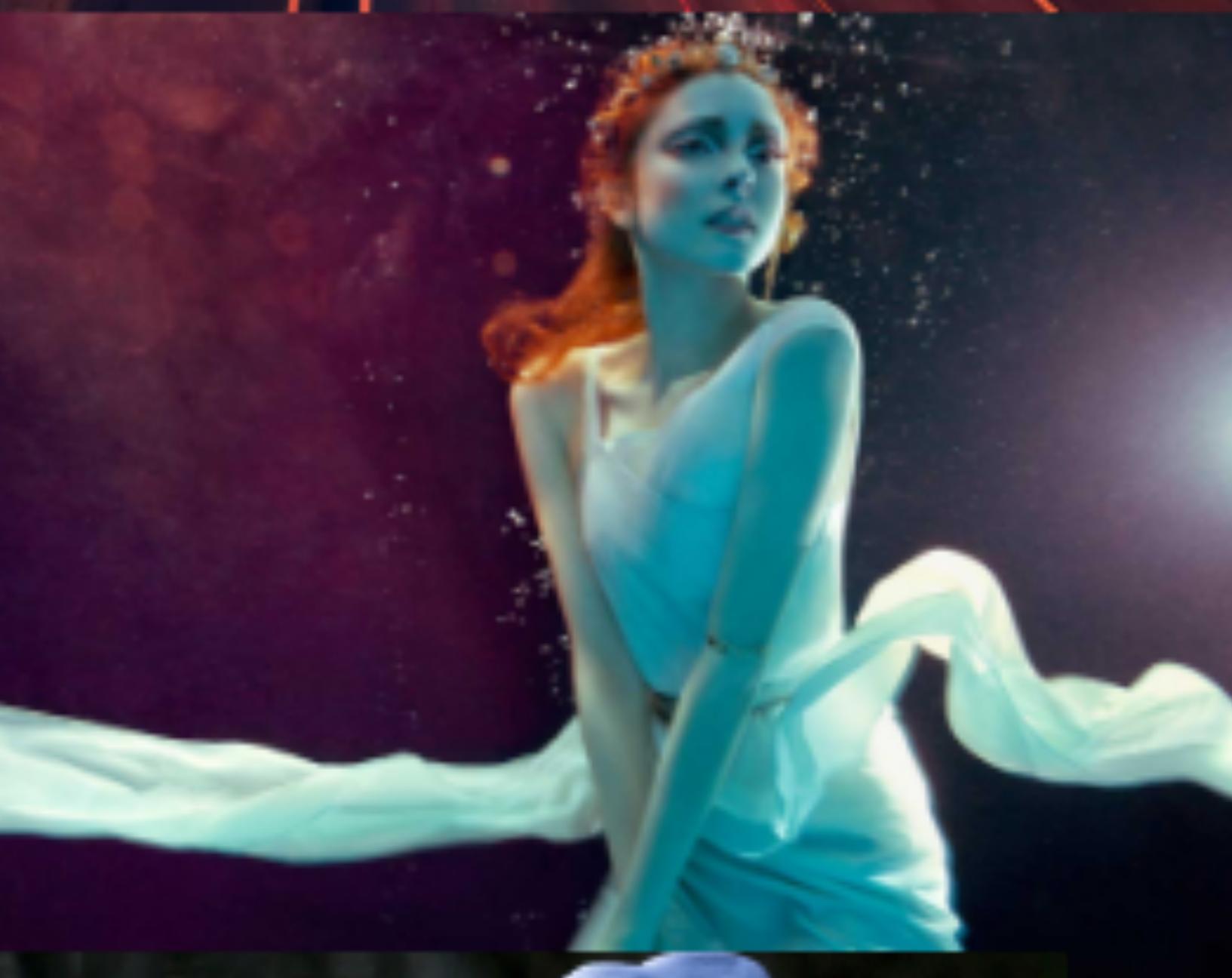


Semantic differential

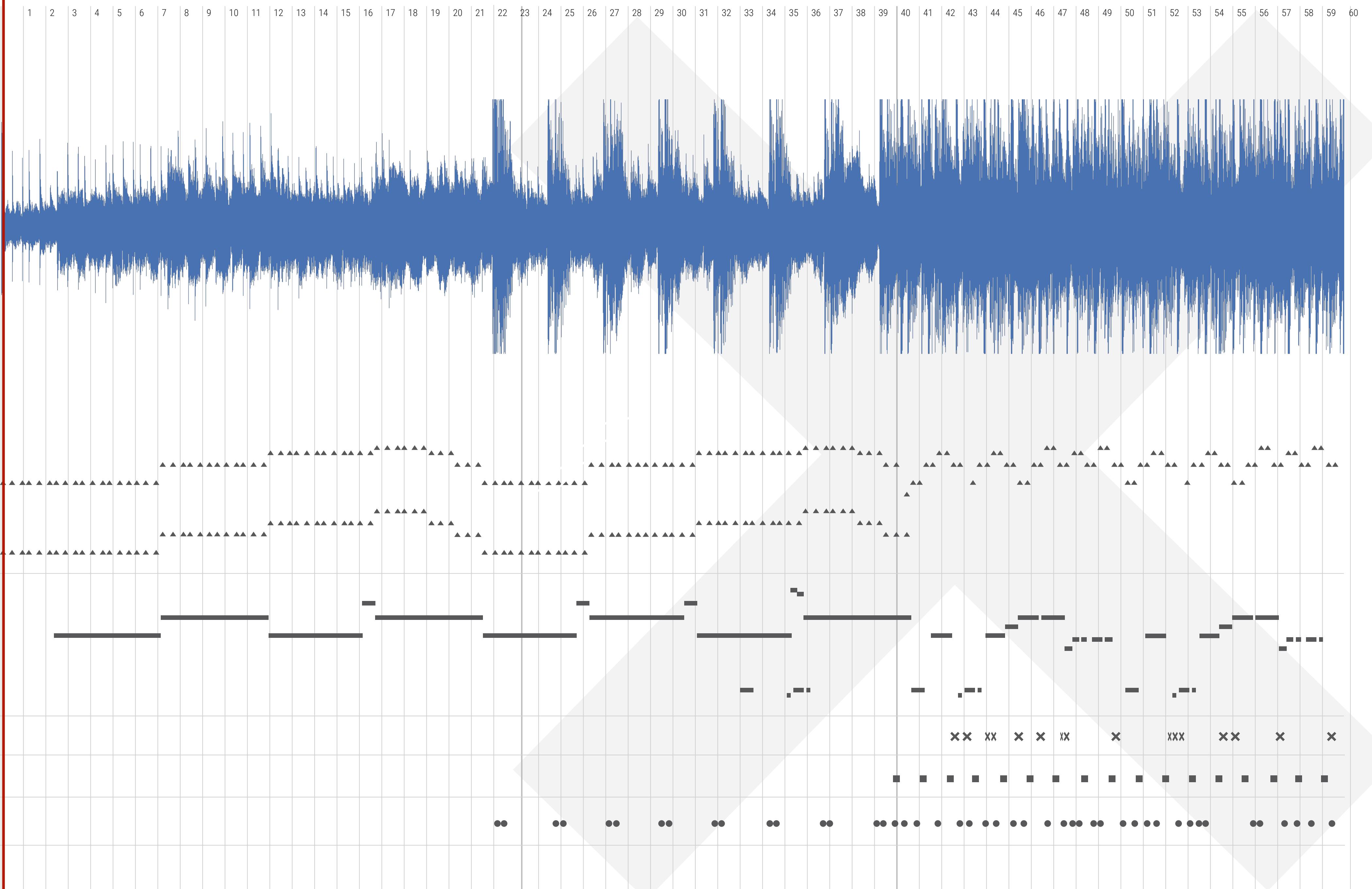




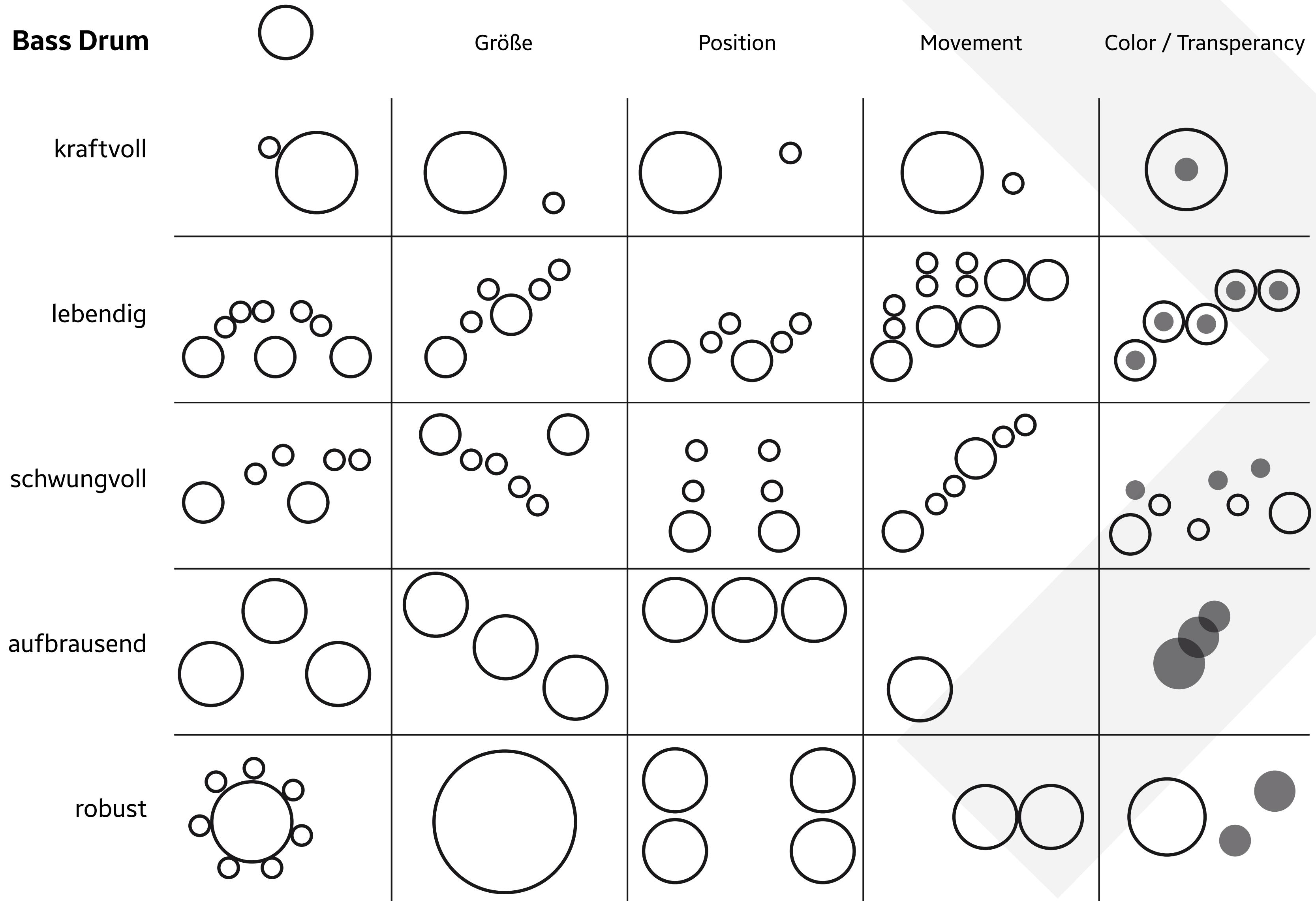




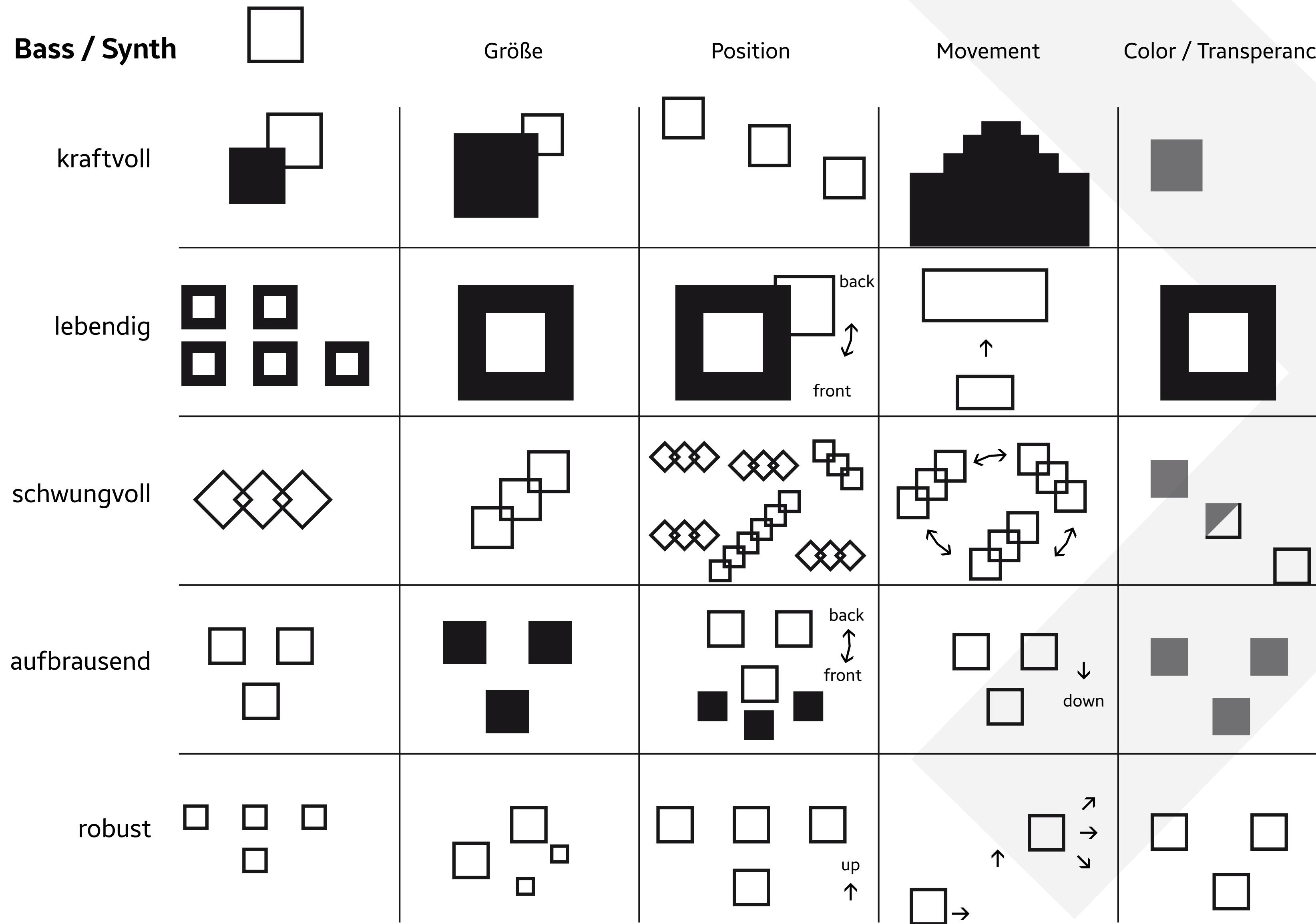
THE XX - INTRO



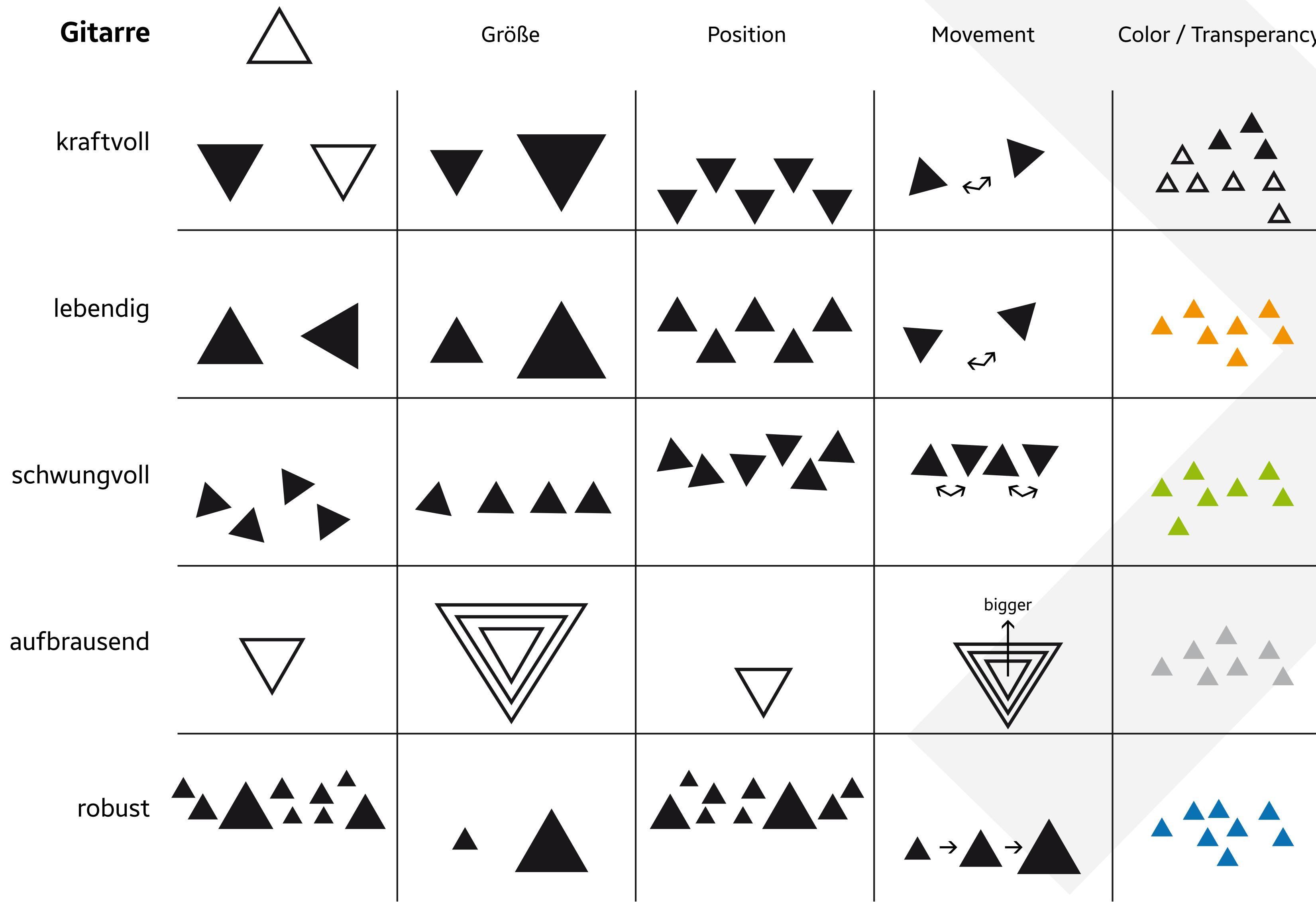
Morphological grid



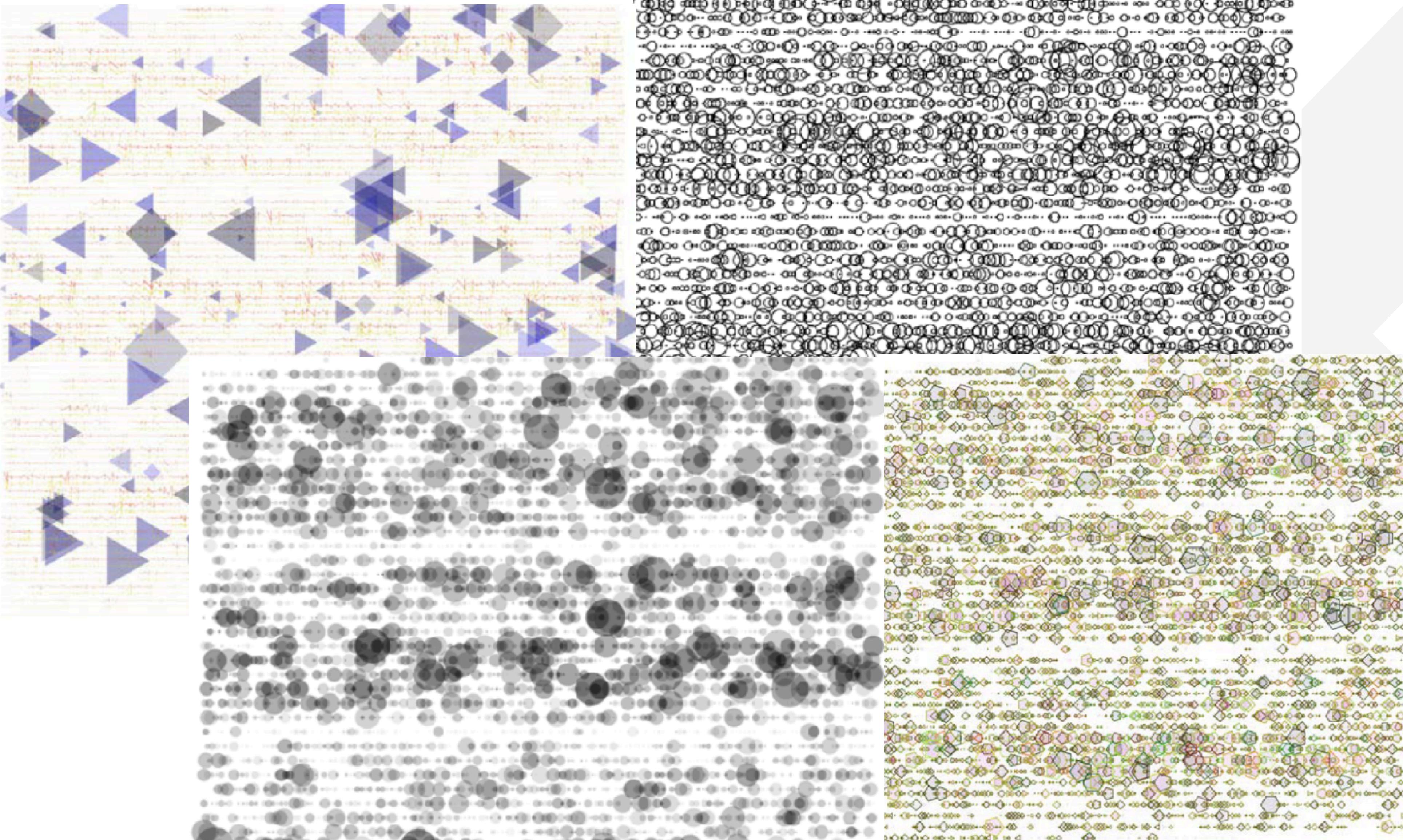
Morphological grid



Morphological grid



First sketches



Code

```
PImage back;
import ddf.minim.*;
import ddf.minim.analysis.*;

Minim minim;
AudioPlayer song;
FFT fft;
|
float[] _sample = new float[512];
float[] _freqBand = new float[8];
float[] bandBuffers = new float[8];
float[] bufferDecrease = new float[8];

int w = 10;           // Width of a single period of the wave
float theta = 0.0;   // Start angle at 0
float theta2 = 0.0;  // Start angle at 0
float amp = 10.0;   // Height of wave
float offy_1 = 600; //wave blau up or down
float offy_2 = 800; //wave gelb up or down

float x, y;
float dim = 80.0;

float movex = 960;
float movey = -1080;
float speed = 0.38;

float movex1 = 0;
float movey1 = 0;
float speed1 = 3;
```

Code

```
void setup()
{
    size(1920, 1080);
    back = loadImage("data/back-2.png");

    // always start Minim first!
    minim = new Minim(this);

    // specify 512 for the length of the sample buffers
    // the default buffer size is 1024
    song = minim.loadFile("introxx.wav", 512);
    song.play();

    // an FFT needs to know how
    // long the audio buffers it will be analyzing are
    // and also needs to know
    // the sample rate of the audio it is analyzing
    fft = new FFT(song.bufferSize(), song.sampleRate());
}
```

Code

```
void draw()
{
    background(back);
    fft.forward(song.mix);

    for (int i = 0; i < fft.specSize(); i++)
    {
        //line(i, _freqBand[2], i, _freqBand[2] - fft.getBand(i)*4);
        _sample[i] = fft.getBand(i);
    }

    int count = 0;
    for (int i = 0; i < 8; i++) {

        float average = 0;
        int sampleCount = (int)pow(2, i) * 2;

        if (i == 7) {
            sampleCount += 2;
        }

        for (int j = 0; j < sampleCount; j++) {
            average += _sample [count] * (count + 1);
            count++;
        }
        average /= count;
        _freqBand[i] = average;
    }
}
```

Code

```
void wave_1() //dunkelblau
{
    theta += 0.02;

    fill(0, 57, 82);
    stroke(0, 57, 82);
    strokeWeight(0);

    beginShape();
    vertex(0, height);

    for (int x = 0; x < width; x++) {
        vertex(x*w, sin(x/TWO_PI+theta)* _freqBand[1] + offy_1);
    }
    vertex(width, height);
    endShape(CLOSE);
}
```

Code

```
void circlesun()//strahlen und sonne
{
    push();
    strokeWeight(0);
    fill(239, 202, 132);

    movey = movey + speed;
    translate(960, - movey);

    circle(0, 0, 300);
    circle(0, 0, bandBuffers[0]*8);

    strokeWeight(5);
    fill(239, 202, 132);
    stroke(239, 202, 132);

    PVector center = new PVector(0, 0);
    for (int i = 360; i >= -360; i-=20) {
        float x = center.x + cos(radians(i))*_freqBand[5]*9;
        float y = center.y+ sin(radians(i))*_freqBand[5]*9;
        line(center.x, center.y, x, y);
    }

    if (movey > -300 ) {
        movey = -300;
    }

    pop();
}
```

Code

```
void wave_2() //hellblau
{
    theta2 += 0.02;

    fill(0, 81, 115);

    beginShape();
    vertex(0, height);

    for (int x = 0; x < width; x++) {
        vertex(x*w, sin(x/TWO_PI+theta2)* _freqBand[3]*2 + offy_2);
    }
    vertex(width, height);
    endShape(CLOSE);

    strokeWeight(0);
    stroke(255, 223, 0);
}
```

Code

```
void drei()//dreiecke
{
    //push();
    stroke(5, 122, 162);
    fill(5, 122, 162);

    //movey1 = movey1 + speed1;
    //translate( 0, movey1);

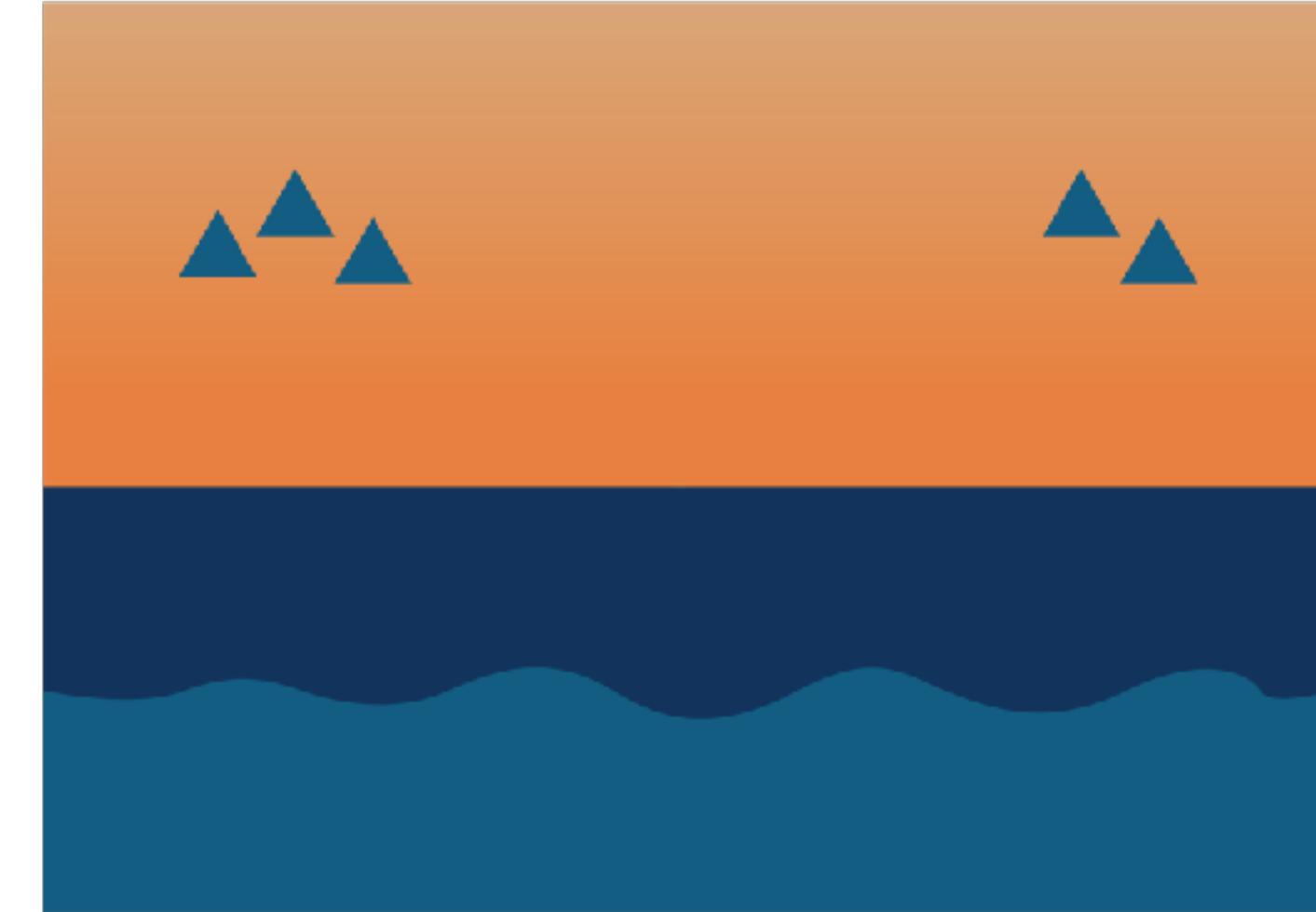
    triangle(189.699, 120.04, 257.849, _freqBand[3], 326, 120.04);
    triangle(325.699, 238.04, 393.849, _freqBand[3]+(118.04), 462, 238.04);
    triangle(462.199, 356.54, 530.349, _freqBand[3]+(118.04*2), 598.5, 356.54);
    ..
}
```

Code

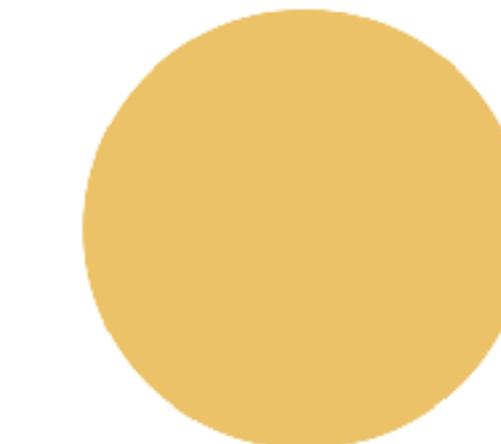
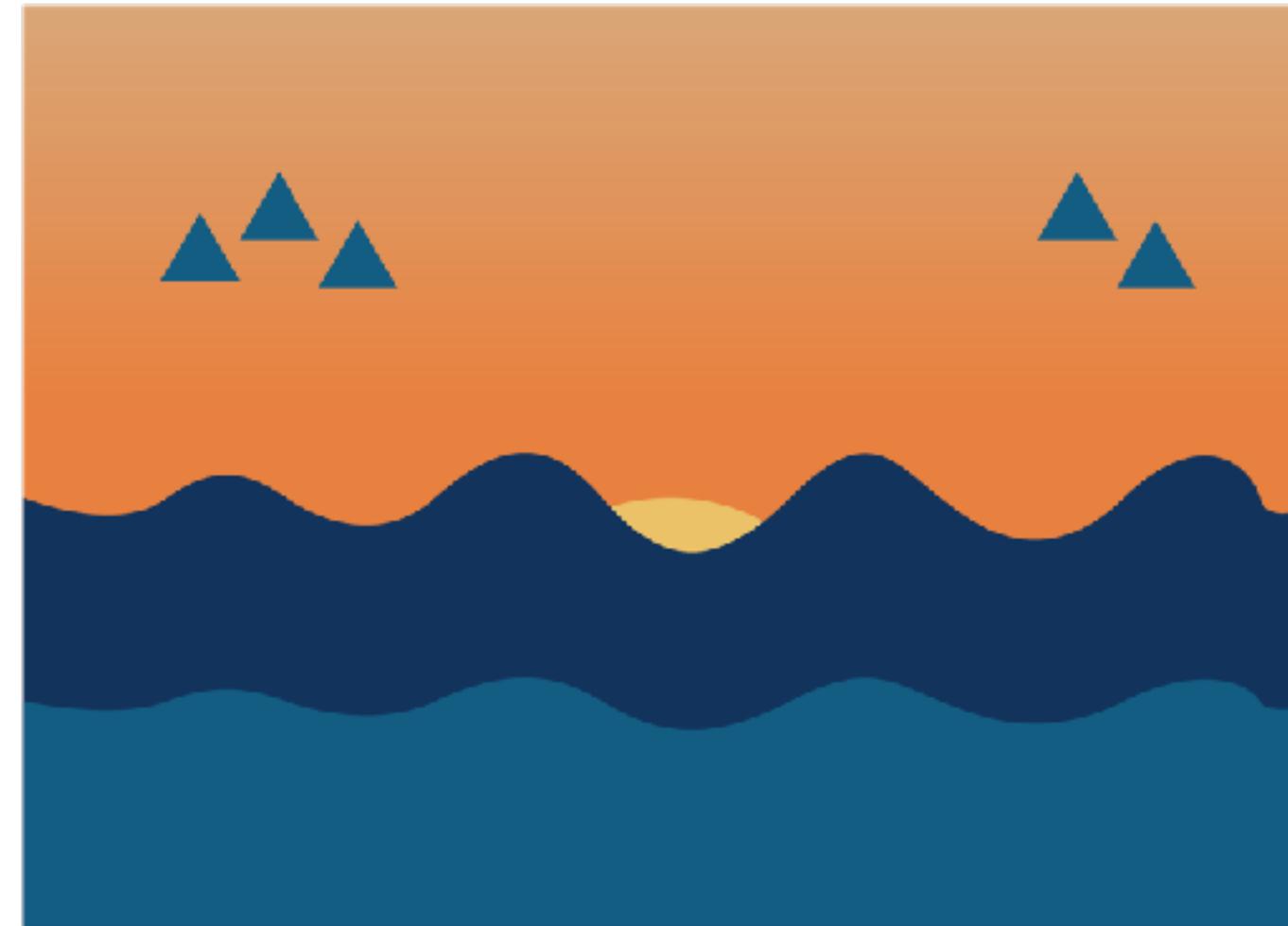
```
void bandBuffer()//buffer für flüssigen Übergang
{
    for (int i = 0; i < 8; i++)
    {
        if (_freqBand[i] > bandBuffers[i])
        {
            bandBuffers[i] = _freqBand[i];
            bufferDecrease[i] = 0.006f;
        }
        if (_freqBand[i] < bandBuffers[i])
        {
            bandBuffers[i] -= bufferDecrease[i];
            bufferDecrease[i] *= 10f;
        }
    }
}
```

Storyboard

Wasser und Dreiecke reagiren auf Gitarre.



Horizont bewegt sich zur Frequenz.



Bassdrum

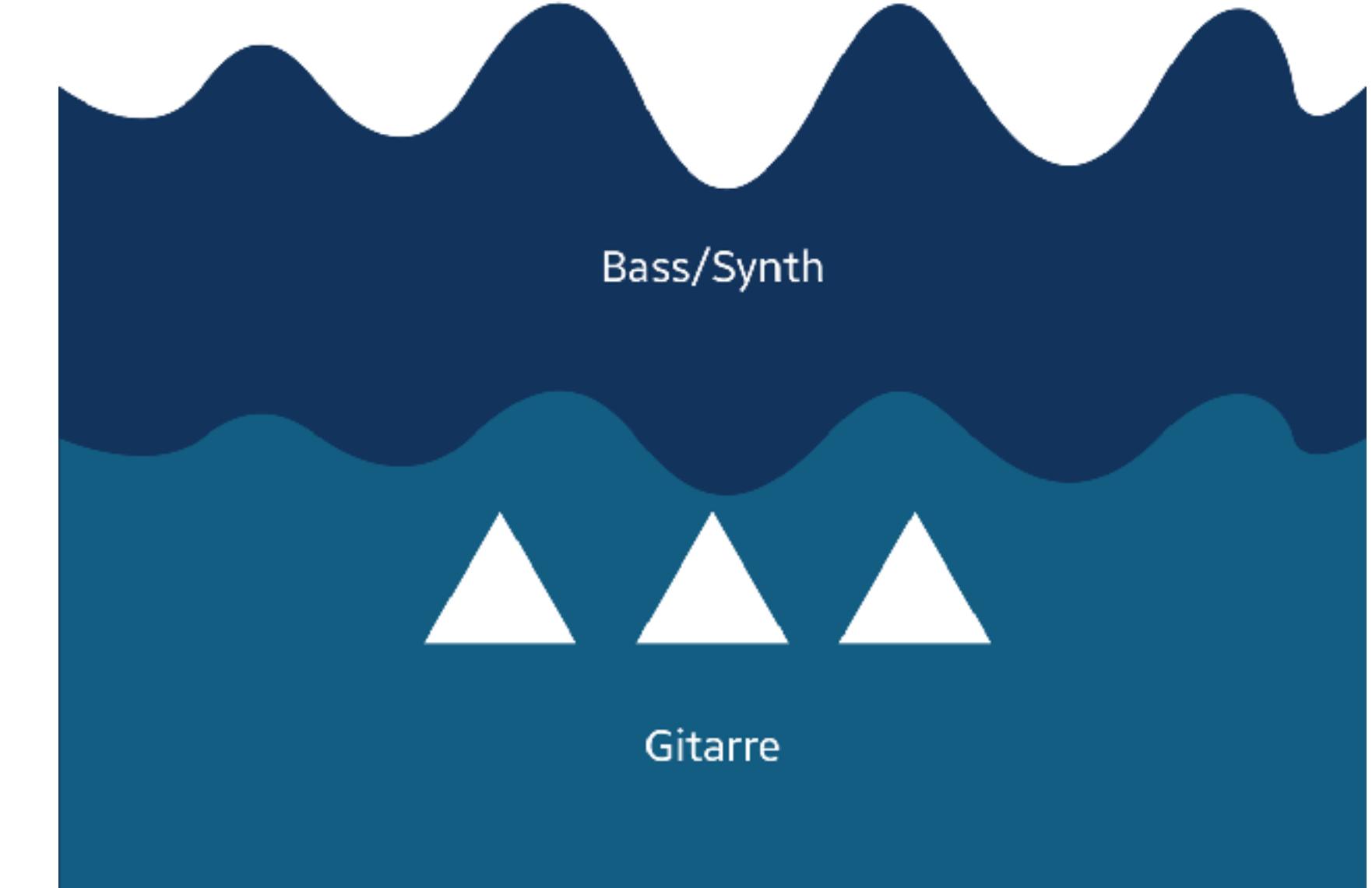


Snaredrum

Sonne dehnt sich zu den Schlägen der Bass-Drum aus.



Sonnenstrahlen ziehen sich in die Länge.



Bass/Synth

Gitarre

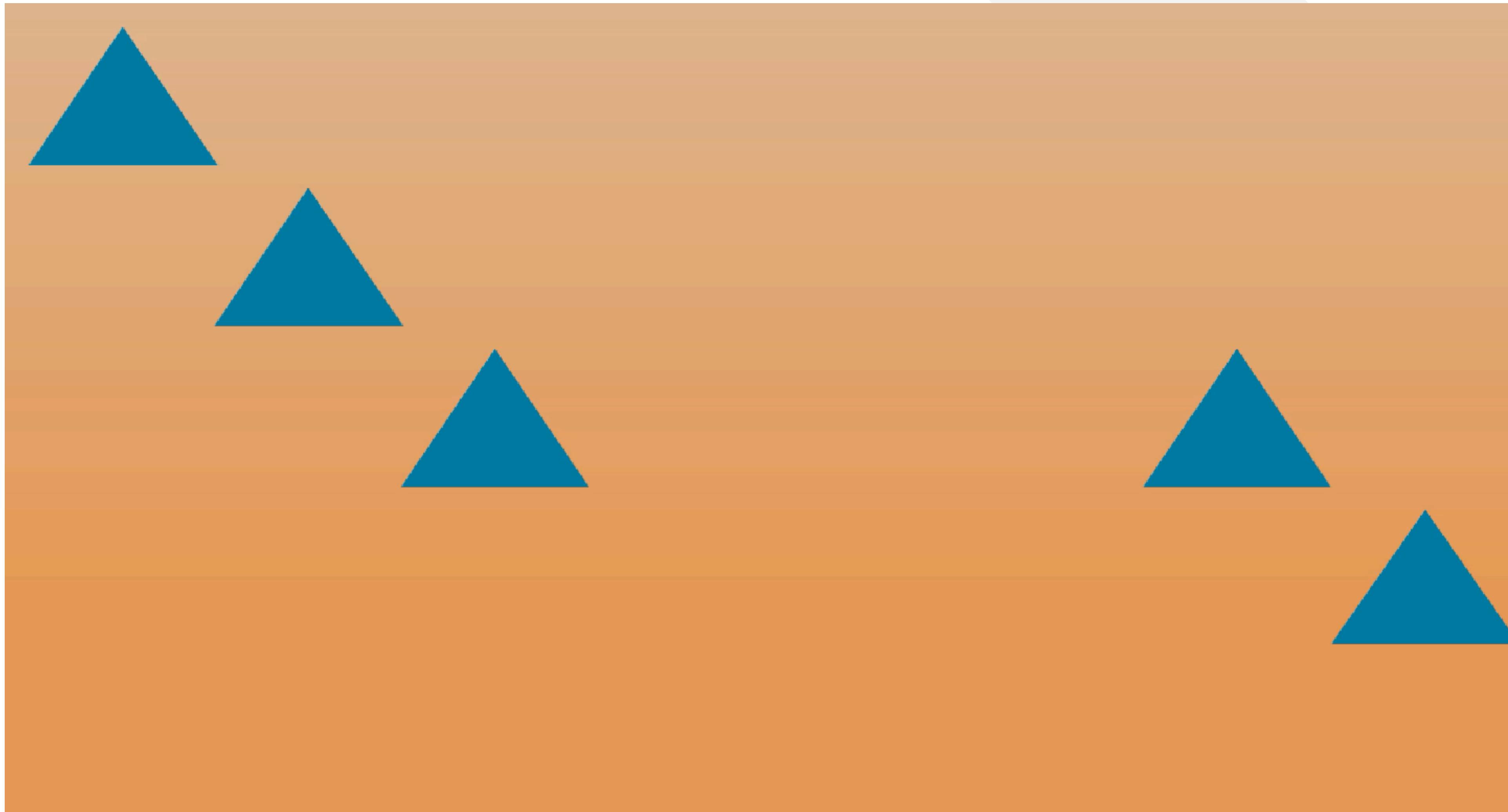
Frequenzbänder



Animation: Gitarre



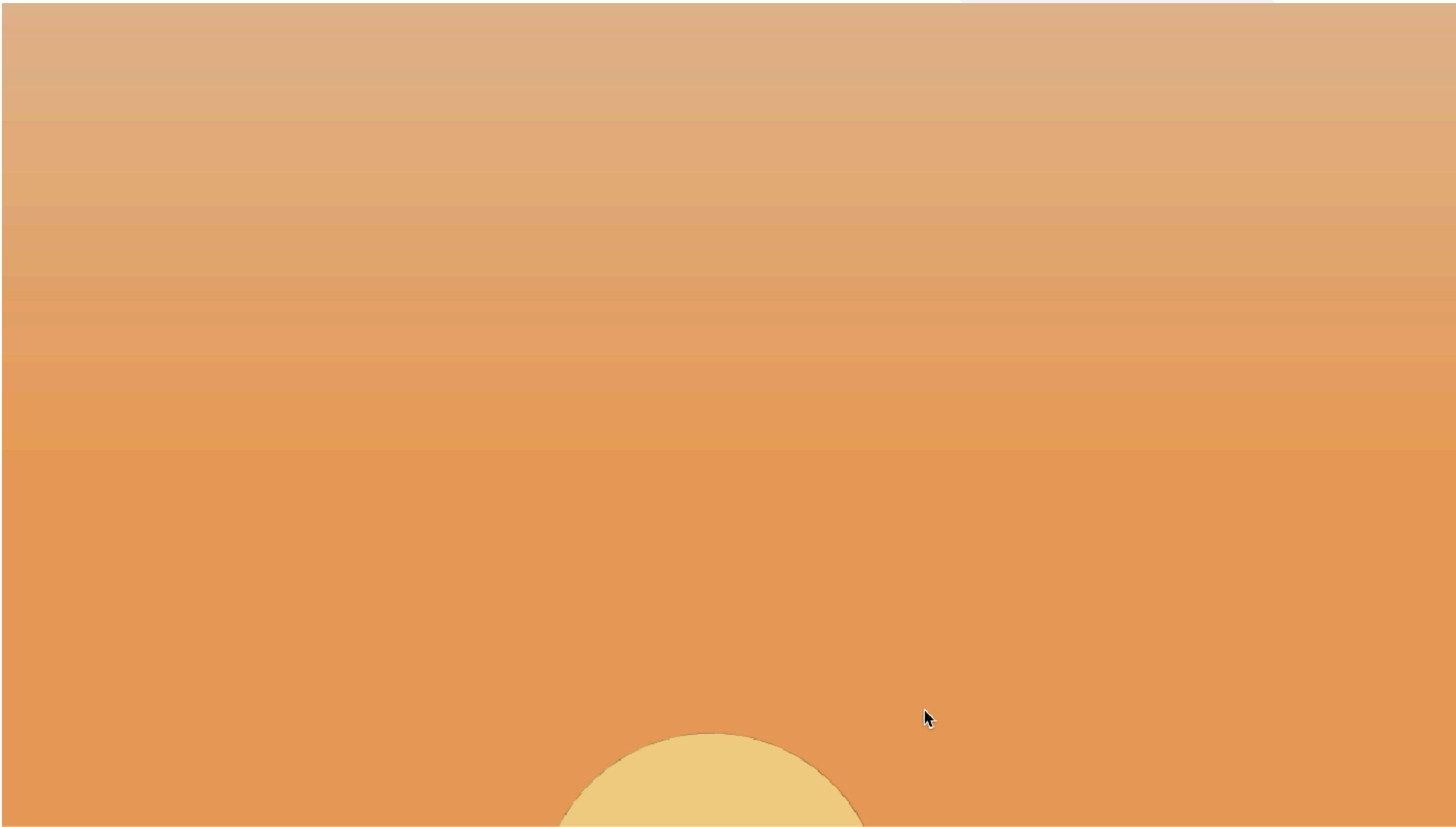
Animation: Bass / Synth



Animation: Bass / Synth



Animation: Bass Drum / Snare Drum



Vielen Dank für eure Aufmerksamkeit!