Electronic instruments

20201116 Subin, Lee

Concept

In order to make music, you need to learn lots of stuff, such as way to play the instument, notes and ect…
This makes it the non-expert person in music hard to make one.
However, everyone know how to bounce back and forth along the rythme or tap with their fingers to make a simple beat.

Concept -KOR

음악을 즐기는 이들은 많지만 음악을 만들 수 있는 사람들은 극소수이다.
음악을 만들려면 전문적인 지식을 요하기 때문이다.

그러나 모든 사람들은 몸을 이용하여 소리를 만든다. 예를 들면 손가락을 탭하며 비트를 만들고 흥얼거리며 멜로디를 만든다.

이 생각에서 시작하여 몸의 움직임을 이용하여 음악을 만들 수 있는 장치를 만들고자 했다.

Functions

1. Make Music

   Put on the glove and control the sound simply by moving your hands around

   Make a drum hitting movent and drum will play

2. Sound Visualization

   Based on the music you are making, the sound wave will apear.

3. Control the mood

   By using the trackpad, change the mood by chaning the screen color

Functions - KOR

1. 음악 만들기

   장갑을 끼고 손을 움직이며 원하는 소리를 만든다.

   드럼을 치는 모션을 하면 드럼 소리가 난다

2. 소리의 시각화

   만든 소리가 시각화 된다.
   손으로 만든 소리에 따라 파도의 형태가 변하고 드럼을 칠 때마다 태양이 bounce한다

3. 무드 조정

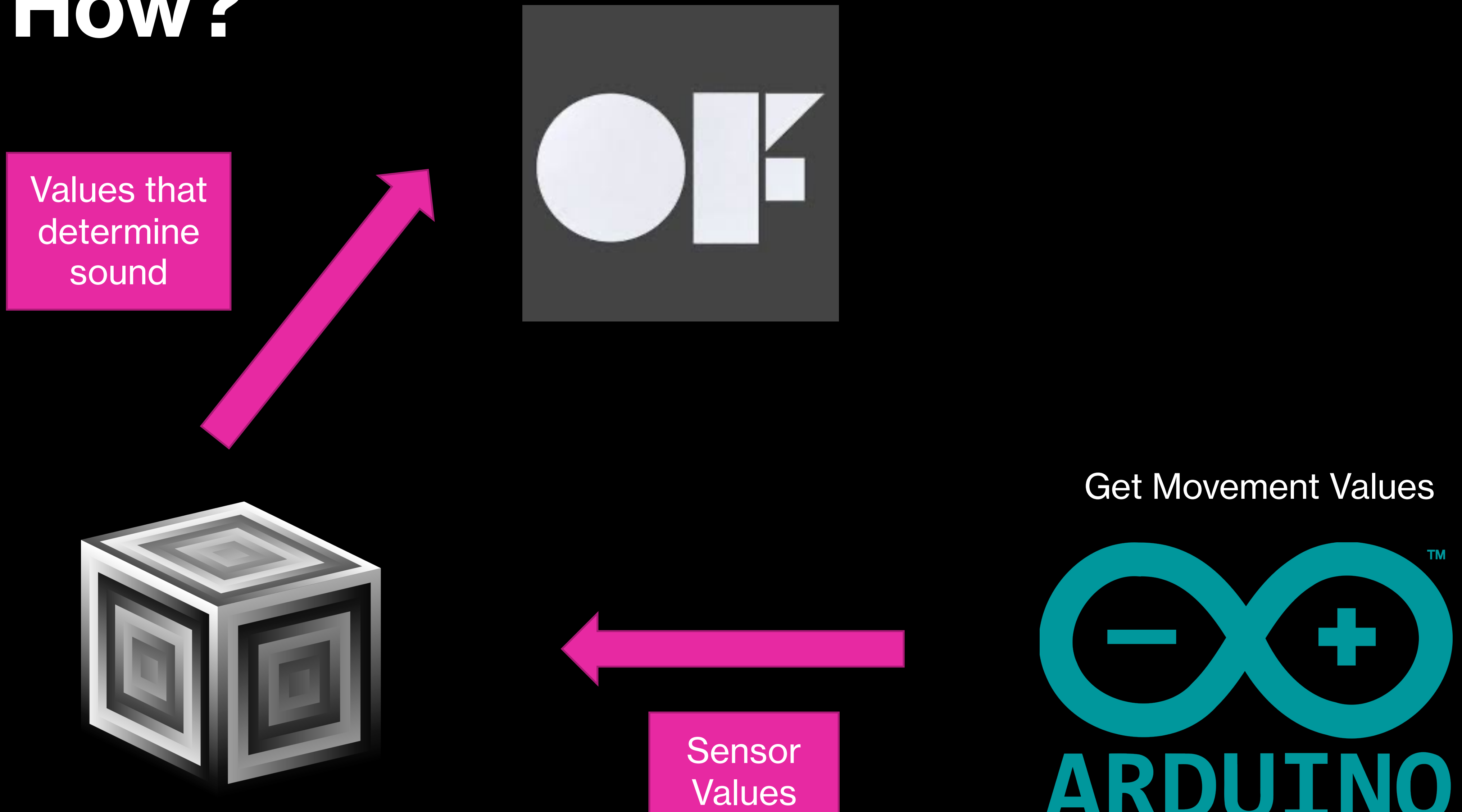
   마우스를 이용해 색에 변화를 주며 무드를 조정할 수 있다.

Target Users

It is a musical instrument for anybody , even for those who does not know much about music!

Anyone can easily make music with it!

How?



Sound Visualization

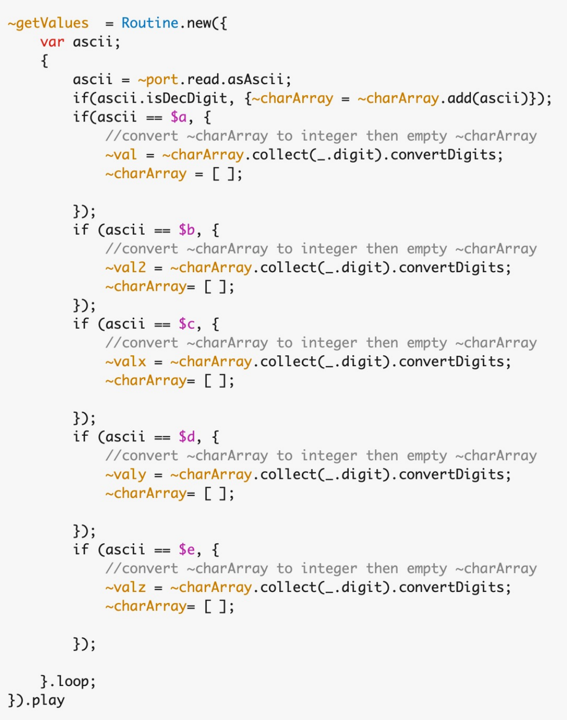
Values that determine sound

Get Movement Values

Sensor Values

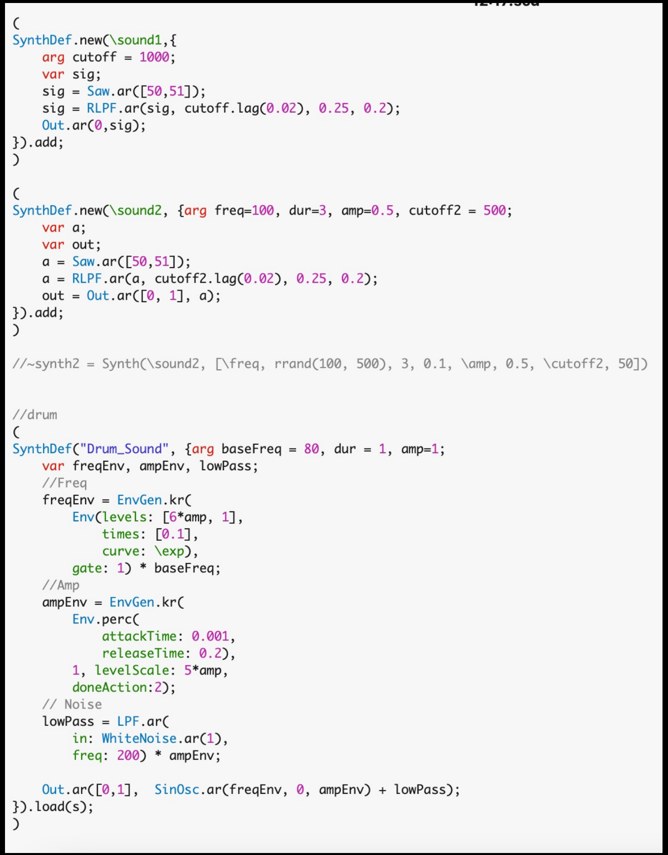
Convert the movement into music

Super Coillder



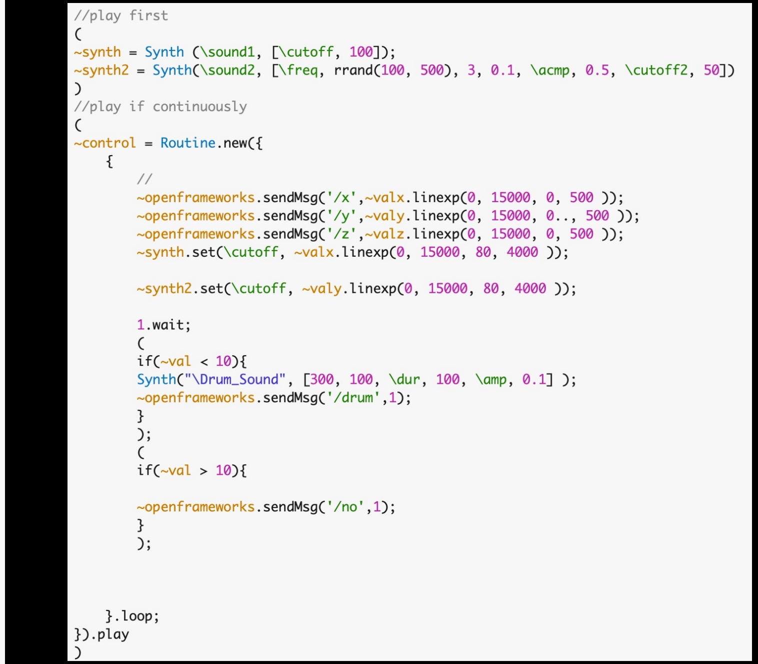
```
~getValues  = Routine.new({
    var ascii;
    {
        ascii = ~port.read.asAscii;
        if(ascii.isDecDigit, {~charArray = ~charArray.add(ascii)});
        if(ascii == $a, {
            //convert ~charArray to integer then empty ~charArray
            ~val = ~charArray.collect(_.digit).convertDigits;
            ~charArray = [ ];

        });
        if (ascii == $b, {
            //convert ~charArray to integer then empty ~charArray
            ~val2 = ~charArray.collect(_.digit).convertDigits;
            ~charArray= [ ];
        });
        if (ascii == $c, {
            //convert ~charArray to integer then empty ~charArray
            ~valx = ~charArray.collect(_.digit).convertDigits;
            ~charArray= [ ];

        });
        if (ascii == $d, {
            //convert ~charArray to integer then empty ~charArray
            ~valy = ~charArray.collect(_.digit).convertDigits;
            ~charArray= [ ];

        });
        if (ascii == $e, {
            //convert ~charArray to integer then empty ~charArray
            ~valz = ~charArray.collect(_.digit).convertDigits;
            ~charArray= [ ];

        });

    }.loop;
}).play
```

Get Values from Arduino



```
(
SynthDef.new(\sound1,{
    arg cutoff = 1000;
    var sig;
    sig = Saw.ar([50,51]);
    sig = RLPF.ar(sig, cutoff.lag(0.02), 0.25, 0.2);
    Out.ar(0,sig);
}).add;
)

(
SynthDef.new(\sound2, {arg freq=100, dur=3, amp=0.5, cutoff2 = 500;
    var a;
    var out;
    a = Saw.ar([50,51]);
    a = RLPF.ar(a, cutoff2.lag(0.02), 0.25, 0.2);
    out = Out.ar([0, 1], a);
}).add;
)

//~synth2 = Synth(\sound2, [\freq, rrand(100, 500), 3, 0.1, \amp, 0.5, \cutoff2, 50])


//drum
(
SynthDef("Drum_Sound", {arg baseFreq = 80, dur = 1, amp=1;
    var freqEnv, ampEnv, lowPass;
    //Freq
    freqEnv = EnvGen.kr(
        Env(levels: [6*amp, 1],
            times: [0.1],
            curve: \exp),
        gate: 1) * baseFreq;
    //Amp
    ampEnv = EnvGen.kr(
        Env.perc(
            attackTime: 0.001,
            releaseTime: 0.2),
        1, levelScale: 5*amp,
        doneAction:2);
    // Noise
    lowPass = LPF.ar(
        in: WhiteNoise.ar(1),
        freq: 200) * ampEnv;

    Out.ar([0,1],  SinOsc.ar(freqEnv, 0, ampEnv) + lowPass);
}).load(s);
)
```

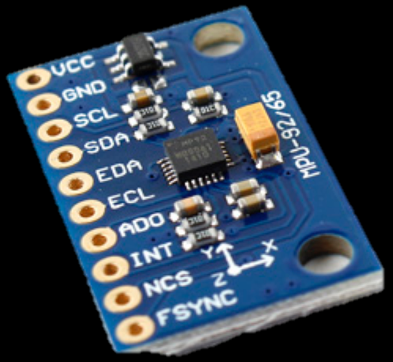
Define Sound

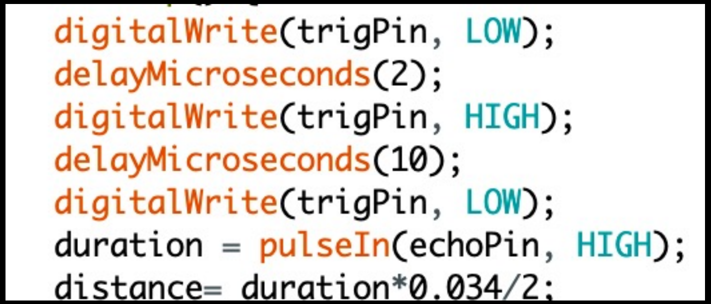


```
//play first
(
~synth = Synth (\sound1, [\cutoff, 100]);
~synth2 = Synth(\sound2, [\freq, rrand(100, 500), 3, 0.1, \acmp, 0.5, \cutoff2, 50])
)
//play if continuously
(
~control = Routine.new({
    {
        //
        ~openframeworks.sendMsg('/x',~valx.linexp(0, 15000, 0, 500 ));
        ~openframeworks.sendMsg('/y',~valy.linexp(0, 15000, 0.., 500 ));
        ~openframeworks.sendMsg('/z',~valz.linexp(0, 15000, 0, 500 ));
        ~synth.set(\cutoff, ~valx.linexp(0, 15000, 80, 4000 ));

        ~synth2.set(\cutoff, ~valy.linexp(0, 15000, 80, 4000 ));

        1.wait;
        (
        if(~val < 10){
        Synth("\Drum_Sound", [300, 100, \dur, 100, \amp, 0.1] );
        ~openframeworks.sendMsg('/drum',1);
        }
        );
        (
        if(~val > 10){

        ~openframeworks.sendMsg('/no',1);
        }
        );



    }.loop;
}).play
)
```

Make Music and
Send it to OF

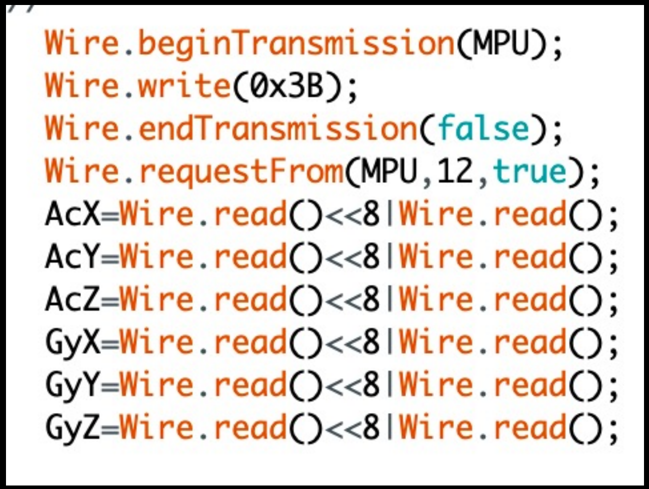
Arduino





```
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance= duration*0.034/2;
```

Get distance using Ultra Sonic Sensor

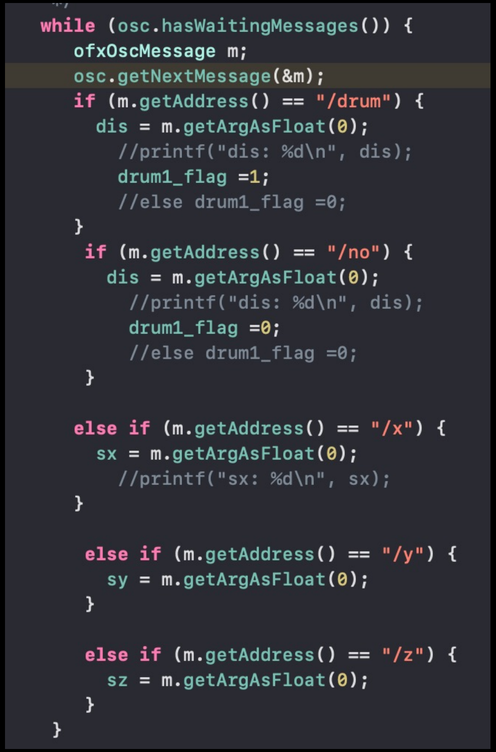


```
Wire.beginTransmission(MPU);
Wire.write(0x3B);
Wire.endTransmission(false);
Wire.requestFrom(MPU,12,true);
AcX=Wire.read()<<8|Wire.read();
AcY=Wire.read()<<8|Wire.read();
AcZ=Wire.read()<<8|Wire.read();
GyX=Wire.read()<<8|Wire.read();
GyY=Wire.read()<<8|Wire.read();
GyZ=Wire.read()<<8|Wire.read();
```

Get accelerometer values

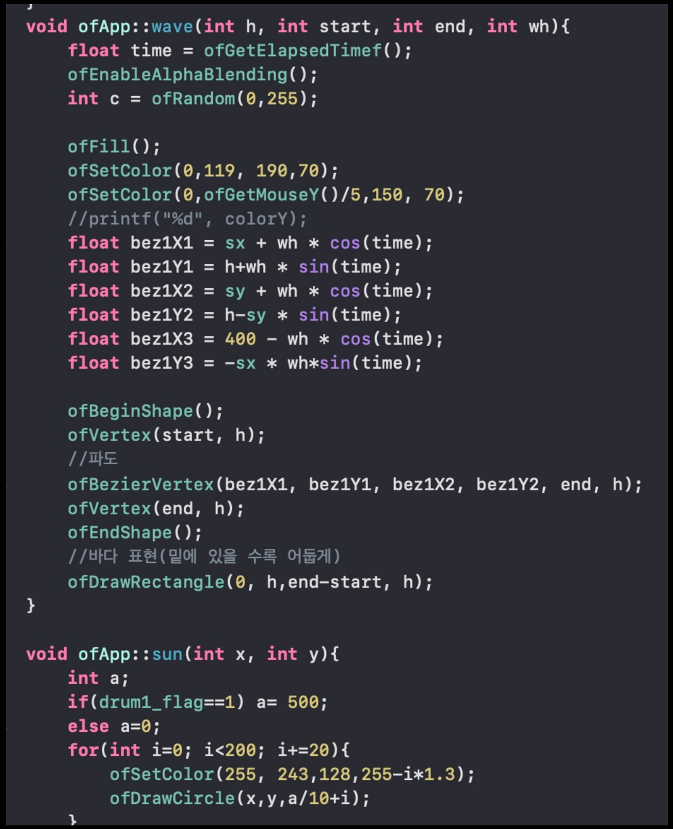
And send Data

Open Frameworks



```
 */
 while (osc.hasWaitingMessages()) {
    ofxOscMessage m;
    osc.getNextMessage(&m);
    if (m.getAddress() == "/drum") {
      dis = m.getArgAsFloat(0);
        //printf("dis: %d\n", dis);
        drum1_flag =1;
        //else drum1_flag =0;
    }
     if (m.getAddress() == "/no") {
       dis = m.getArgAsFloat(0);
         //printf("dis: %d\n", dis);
         drum1_flag =0;
         //else drum1_flag =0;
     }

    else if (m.getAddress() == "/x") {
      sx = m.getArgAsFloat(0);
        //printf("sx: %d\n", sx);
    }

     else if (m.getAddress() == "/y") {
       sy = m.getArgAsFloat(0);
     }

     else if (m.getAddress() == "/z") {
       sz = m.getArgAsFloat(0);
     }
 }
```

Get Data



```
}
void ofApp::wave(int h, int start, int end, int wh){
    float time = ofGetElapsedTimef();
    ofEnableAlphaBlending();
    int c = ofRandom(0,255);

    ofFill();
    ofSetColor(0,119, 190,70);
    ofSetColor(0,ofGetMouseY()/5,150, 70);
    //printf("%d", colorY);
    float bez1X1 = sx + wh * cos(time);
    float bez1Y1 = h+wh * sin(time);
    float bez1X2 = sy + wh * cos(time);
    float bez1Y2 = h-sy * sin(time);
    float bez1X3 = 400 - wh * cos(time);
    float bez1Y3 = -sx * wh*sin(time);

    ofBeginShape();
    ofVertex(start, h);
    //파도
    ofBezierVertex(bez1X1, bez1Y1, bez1X2, bez1Y2, end, h);
    ofVertex(end, h);
    ofEndShape();
    //바다 표현(밑에 있을 수록 어둡게)
    ofDrawRectangle(0, h,end-start, h);
}

void ofApp::sun(int x, int y){
    int a;
    if(drum1_flag==1) a= 500;
    else a=0;
    for(int i=0; i<200; i+=20){
        ofSetColor(255, 243,128,255-i*1.3);
        ofDrawCircle(x,y,a/10+i);
    }
```

Sound Visualization Part

Demo Video

https://drive.google.com/file/d/1wHxSc_pgSsewdMqeFM8NjHEnk60118iA/view?usp=sharing

Thank You