

Project herexamen New Media Devices

Astrid Vandevoorde

2016-2017

3 Multec A&T

Filmje met demo

<https://youtu.be/MBZiBKC3J6E>

Werking

We hebben 2 applicaties. 1 voor de android en 1 voor de pc. Beide applicaties zijn gemaakt in processing. De applicatie voor android is gemaakt met de modus "android" en de optie "VR" in processing 3.

Om de android (samsung galaxy s6) als VR apparaat te gebruiken heb ik google VR en google cardboard moeten installeren op de smartphone.

Op de computer heb ik de cardboard library en gradle gedownload voor processing en android.

De app van de android gebruikt het volume zijn eigen microfoon om een bolletje in VR van kleur(hsb) en grootte te veranderen. De app stuurt ook de gyroscope gegevens door naar de computer over zijn eigen netwerk via osc.

De app van de computer leest deze x,y,z gegevens om zijn cubus van vorm en kleur (rgb) te veranderen.

Het concept was een beetje gebaseerd op wat we gemaakt hadden in de les in het eerste semester. Daar werkte we ook met een interactieve cubus in VR.

Het idee was dan ook om abstract te blijven in de VR en op de app en de gebruiker wat te laten uitzoeken wat met wat correspondeert.

De microfoon te laten werken met de app was wel 1 van de moeilijke dingen van dit concept.

Minim kan je niet gebruiken met android dus heb ik een alternatief moeten zoeken om de gegevens te kunnen halen van de microfoon. Ik had ook nog nooit met android gewerkt dus ik heb toch wel wat bijgeleerd op het vlak van android app development, wat eigenlijk meer een onderwerp is voor App&web dan voor A&T.

Je kan de developers mode activeren in android door op je buildnummer in toestel – info 7 keer in te drukken. Dit is nodig om applicaties te kunnen installeren op je android via processing.

Code

Android app code

```
import oscP5.*;
import netP5.*;
import processing.vr.*;
import android.app.Activity;
import android.os.Bundle;
import android.media.AudioFormat;
import android.media.AudioRecord;
import android.media.MediaRecorder;
import android.hardware.Sensor;
import android.hardware.SensorManager;
import android.content.Context;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;

OscP5 oscP5;

NetAddress network;

PShape grid;
float r;
color newColour;

Context context;
SensorManager manager;
Sensor gyroscope;
GyroscopeListener listener;
float ax, ay, az;

short[] buffer = null;

AudioRecord audioRecord = null;
```

```

int bufferSize= 1024;

float volume = 0;

int buflen;


void setup() {
  colorMode(HSB, 255);
  fullScreen(STEREO);
  //osc connect to computer
  oscP5 = new OscP5(this, 7000);
  network = new NetAddress("192.168.43.199", 10000);


  //gyroscope
  context = getActivity();
  manager = (SensorManager)context.getSystemService(Context.SENSOR_SERVICE);
  gyroscope = manager.getDefaultSensor(Sensor.TYPE_GYROSCOPE);
  listener = new GyroscopeListener();
  manager.registerListener(listener, gyroscope, SensorManager.SENSOR_DELAY_NORMAL);


  //sound

  int freq =44100;
  int chan = AudioFormat.CHANNEL_IN_MONO;
  int enc = AudioFormat.ENCODING_PCM_16BIT;
  int src = MediaRecorder.AudioSource.MIC;
  buflen = AudioRecord.getMinBufferSize(freq, chan, enc);
  audioRecord = new AudioRecord(src,freq,chan,enc,buflen);


  audioRecord.startRecording();
  buffer = new short[bufferSize];


  //grid
  grid = createShape();

```

```

grid.beginShape(LINES);
grid.stroke(255);
for (int x = -10000; x < +10000; x += 250) {
    grid.vertex(x, +1000, +10000);
    grid.vertex(x, +1000, -10000);
}
for (int z = -10000; z < +10000; z += 250) {
    grid.vertex(+10000, +1000, z);
    grid.vertex(-10000, +1000, z);
}
grid.endShape();

}

void draw() {

    background(0);

    lights();

    translate(width/2, height/2);

    shape(grid);

    //get volume
    int bufferReadResult = audioRecord.read(buffer, 0, bufferSize);

    volume = 100;

    for (int i = 0; i < bufferReadResult; i++) {
        volume = Math.max(Math.abs(buffer[i]), volume);
    }

    text("" + volume, 100, 100);

    //draw a sphere if you blow the mic/ talk to the mic
    if(volume>300){
        makeSphere(volume);

    }
}

```

```

        fill(newColour);

        sphere(r);

        //print pos
        // println("X: " + ax + "\nY: " + ay + "\nZ: " + az, 0, 0, width, height);

        sendMessage();

    }

```

```

void stop() {
    audioRecord.stop();
    audioRecord.release();
    audioRecord = null;
}

```

```

void makeSphere(float volume){
    //make sphere

    noStroke();

    r = map(volume,300,10000,50,300);
    newColour = color(map(volume,300,10000,0,255),255,255);

}

```

```

class GyroscopeListener implements SensorEventListener {

    //get gyroscope values

    public void onSensorChanged(SensorEvent event) {

        ax = event.values[0];
        ay = event.values[1];
        az = event.values[2];

    }

    public void onAccuracyChanged(Sensor gyropscope, int pos) {

```

```
}  
  
}  
void sendMessage(){  
    //send positions to computer  
    OscMessage myMessage = new OscMessage("/position");  
    myMessage.add(ax);  
    myMessage.add(ay);  
    myMessage.add(az);  
    /* send the message */  
    oscP5.send(myMessage, network);  
  
}
```

Code applicatie Java

```
import oscP5.*;
```

```
import netP5.*;
```

```
OscP5 osc;
```

```
NetAddress network;
```

```
float x, y, z;
```

```
float s;
```

```
color newColour;
```

```
float ax, ay, az;
```

```
void setup() {
```

```
    size(1366,768,P3D);
```

```
    //osc verbinding met android
```

```
    osc = new OscP5(this, 10000);
```

```
    network = new NetAddress("192.168.0.255",7000);
```

```
}
```

```
void draw() {
```

```
    background(0);
```

```
    lights();
```

```
    translate(width/2, height/2);
```

```
    //update cube
```

```
    makeCube();
```

```
    rotateY(0.5);
```

```
    fill(newColour);
```

```
    rotateY(s);
```

```
    box(x,y,z);
```

```
    //print pos
```

```
    // println("X: " + ax + "\nY: " + ay + "\nZ: " + az, 0, 0, width, height);
```

```
}
```

```
void makeCube(){  
    //make sphere  
    noStroke();  
    //map the positions  
    x = map(ax,-1,1,50,300);  
    y = map(ay,-1,1,50,300);  
    z = map(az,-1,1,50,300);  
    s = az;  
    //make a new colour  
    newColour = color(map(ax,-1,1,0,255),map(ay,-1,1,0,255),map(az,-1,1,0,255));  
  
}
```

```
void oscEvent(OscMessage theOscMessage) {  
    //get osc messages from android  
    if (theOscMessage.addrPattern().equals("/position")) {  
        ax = theOscMessage.get(0).floatValue();  
        ay = theOscMessage.get(1).floatValue();  
        az = theOscMessage.get(2).floatValue();  
        println(ax + "/x " + ay + "/y " + az + "/z" + "message success");  
    }  
  
}
```


Literatuurlijst

<http://android.processing.org>

<https://forum.processing.org/two/>

<https://stackoverflow.com/>

<https://developer.android.com/index.html>