

## VU\_Meter

```
//VU Meter
```

```
//example importing libraries
```

```
//introduces "arc" command of the form arc(center x,center y,width,height,start angle,end angle);
```

```
//Mr. H.
```

```
import processing.sound.*;
Amplitude soundAmplitude;
AudioIn soundIn;
```

```
int x = 300, y=300;
int count, peakCount = 0;
int radius = 140;
float angle= 0;
float maximumValue, peakValue = 0;
```

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

```
  soundAmplitude = new Amplitude(this);
  soundIn = new AudioIn(this, 0);
```

```
  // start the Audio Input
  soundIn.start();
  soundAmplitude.input(soundIn);
}
```

```
void draw() {
  background(100);
```

```
  angle = map(soundAmplitude.analyze(), 0, 1, 30, 270);
  soundAmplitude.analyze();
```

```
  //creates title of page
  textAlign(CENTER);
  fill(255);
  textSize(32);
  text("VU METER", width/2, 60);
```

```
  //creates rectangular frame for analog meter
  rectMode(CENTER);
  stroke(255);
  strokeWeight(1);
  fill(0);
  rect(x, y, 400, 260);
```

```
fill(255);
rect(x, y, 380, 240);
```

```
//draws "needle" for VU meter maximum peaks
```

```
stroke(220, 220, 255);
strokeWeight(1);
if (angle>maximumValue) {
  maximumValue = angle;
}
line(x, y+100, x-cos(radians(maximumValue))*radius, y+100-sin(radians(maximumValue))*radius);
count = count + 1;
if (count > 4) {
  maximumValue = maximumValue - 5;
  count = 0;
}
```

```
//draws "needle" of VU meter
```

```
stroke(0, 0, 255);
strokeWeight(4);
line(x, y+100, x-cos(radians(angle))*radius, y+100-sin(radians(angle))*radius);
stroke(0);
strokeWeight(2);
fill(255, 0, 0);
ellipse(x, y+100, 15, 15);
```

```
//draws both major and minor increments along arc
```

```
stroke(0);
for (int i = 30; i <= 150; i=i+15) {
  strokeWeight(2);
  line(x-cos(radians(i))*160, y+100-sin(radians(i))*160, x-cos(radians(i))*178,
y+100-sin(radians(i))*178);
}
for (int i = 30; i <= 150; i=i+3) {
  strokeWeight(1);
  line(x-cos(radians(i))*160, y+100-sin(radians(i))*160, x-cos(radians(i))*168,
y+100-sin(radians(i))*168);
}
```

```
//text on analog meter
```

```
fill(0);
textSize(12);
text("-35", x-cos(radians(30))*200, y+105-sin(radians(30))*200);
text("-30", x-cos(radians(45))*200, y+105-sin(radians(45))*200);
text("-25", x-cos(radians(60))*200, y+105-sin(radians(60))*200);
```

```
text("-20", x-cos(radians(75))*200, y+105-sin(radians(75))*200);
text("-15", x-cos(radians(90))*200, y+105-sin(radians(90))*200);
text("-10", x-cos(radians(105))*200, y+105-sin(radians(105))*200);
text("-5", x-cos(radians(120))*200, y+105-sin(radians(120))*200);
text("0", x-cos(radians(135))*200, y+105-sin(radians(135))*200);
text("+3", x-cos(radians(150))*200, y+105-sin(radians(150))*200);
textSize(24);
text("dB", x-cos(radians(90))*60, y+100-sin(radians(90))*60);
```

```
//displays label for audio channel
```

```
textAlign(CENTER);
textSize(20);
fill(255);
text("AUDIO LEVEL", x, y+170);
```

```
//creates black arc and red maximum arcs along edge of meter
```

```
//uses the form arc(center x,center y,width,height,start angle,end angle);
```

```
stroke(0);
strokeWeight(5);
noFill();
arc(x, y+100, 370, 370, radians(210), radians(330));
stroke(255, 0, 0);
arc(x, y+100, 361, 360, radians(315), radians(329));
}
```