计算机图形学lab1 编程实现音乐节奏或旋律的可视化

马成 20307130112

程序说明

1. 这是一个html文件，其核心代码是使用JavaScript编写的。
2. 不需要任何的配置，直接双击使用浏览器打开即可

算法原理

1. 使用了html自带的插件进行音乐播放，并准备画布让后续js在上面操作

```
<input type="file" style="color:red;" name="" value="" id="musicFile"/>
<input type="button" name="startStop" value="暂停" id="startStop"/>
<p id="tip" style="color:red;"></p>
<canvas id="canvas"></canvas>
```

2. 核心代码即解析音频部分
   - 实例化一个解析器，并设置解析的目标为当前的音乐，设置fftsize为4096
   - 得到解析后的数据之后进行可视化的绘制，遍历每一条数据画出一个宽度为2高度为对应数据的矩形，使用循环绘制就可以得到一个条形图
   - 对于快速傅里叶变换的实现细节有js内部库封装

```
// 解析音乐
var audioBufferSourceNode = audioContext.createBufferSource();
var analyser = audioContext.createAnalyser();
analyser.fftSize = 4096;
audioBufferSourceNode.connect(analyser);
analyser.connect(audioContext.destination);
audioBufferSourceNode.buffer = buffer;
audioBufferSourceNode.start();
var bufferLength = analyser.frequencyBinCount;
// 可视化绘制
var dataArray = new Uint8Array(bufferLength);
var oW = canvas.width;
var oH = canvas.height;
var color1 = canvasCtx.createLinearGradient(oW,oH-10, oW , oH - 150);
color1.addColorStop(0, '#000000');
function draw() {
    drawVisual = requestAnimationFrame(draw);
    var barHeight;
    // 自定义获取数组里边数据的频步
    canvasCtx.clearRect(0, 0, oW, oH);
    for (var i = 0; i < bufferLength; i++) {
        barHeight = dataArray[i]*2;
        analyser.getByteFrequencyData(dataArray);
        // 绘制向上的线条
        canvasCtx.fillStyle = color1;
        canvasCtx.fillRect(oW  - (i * 2), oH, 1, -barHeight);
    }
```

```
};
draw();
```

参考资料

http://blog.csdn.net/xh888zw/article/details/124409016