

Project 1: 音频可视化

16307130158 李政

使用方式

- 浏览器打开 `index.html`，将mp3格式的音乐拖入；
- 自动播放，可以看到有金属光泽的立方体跟随音乐律动。

技术栈

- 绘图：Three.js
- 音频处理：Web API Audio Context 接口

代码逻辑

第一次尝试使用Three.js进行绘图，主要就渲染器、相机控制、灯光控制等进行了学习和使用：

- 首先在 `scene` 中放置45个立方体，调用 `THREE.MeshPhongMaterial` 设置立方体颜色为红色，将参数 `specular` 定义为0xfffff可以得到有金属光泽的渲染效果，按照一定间距将它们依次排列即可；
- 设置相机 `THREE.PerspectiveCamera`，同时注意调用 `THREE.OrbitControls` 绑定在相机上进行视角控制；
- 为了保证音频的播放，需要单独设置源缓冲，这里借助于自带的Web API——AudioContext，设置缓冲区读取并解析音频的内容；
- 核心可视化代码：

```
//script.js:91
this.js_node.onaudioprocess = function () {
    //从解析器中获取数据
    var array = new Uint8Array(that.analyser.frequencyBinCount);
    that.analyser.getByteFrequencyData(array);

    visualizer.renderer.render(visualizer.scene, visualizer.camera);
    visualizer.controls.update();

    var step = Math.round(array.length / visualizer.numberBars);

    //每次得到的音频按照频率梯度依次影响各个立方体的形状，由近及远
    for (var i = 0; i < visualizer.numberBars; i++) {
        var value = array[i * step] / 4;
        value = value < 1 ? 1 : value;
        visualizer.bars[i].scale.z = value;
    }
}
```

- 参考了w3schoolhttps://www.w3school.com.cn/html5/html_5_draganddrop.asp 的拖放脚本，设置了页面对于mp3文件的拖拽和读取。
- **总结：**整体而言，任务一主要进行了Three.js的初步学习（相机、灯光、渲染等），结合Web API的音频读取接口，由于只是做形状的变化，因此较为容易。

