

计算机图形学课程Project

祁佳薇 16307130293

Project 1 编程实现音乐节奏或旋律的可视化

- 原理：Processing编程艺术

Processing是一个为开发面向图形的应用（visually oriented application）而生的简单易用的编程语言和编程环境。Processing的创造者将它看作是一个代码素描本（Sketch）。它尤其擅长算法动画和即时交互反馈，所以近年来在交互动画，复杂数据可视化，视觉设计，原型开发和制作方向越发流行

Processing基于Java，其语法规则和Java一致，但Processing不同于Java，它更为简单，并且已经演化出了它自己的一套“工作习惯”，把常用的图形相关的操作都抽象封装好了，只需要简单调用就行：size(320,240)就可以建立一个320x240大小的画布；想在画布上画一个点，调用point(60,80)，就可以在(60,80)的位置画一个点了；想画一条线只需调用line(0,0,width,height)，就可以画一条从左上角到右下角的线；width和height就是画布的宽和高的尺寸，processing会自动把它转换成实际的画布尺寸

在新建项目时选择p5.js而不是java，导出.js文件。或者现在有基于它的js框架processing.js，通过该框架可以直接将processing的.pde程序转化成javascript，然后在HTML5的canvas中画出来。但是实践中企图通过这两种方式嵌入html时遇到众多问题，最终在Processing IDE中运行，导出项目见代码部分

- 实现：程序说明

1) 首先介绍p5.js方法，有两个默认函数function：

- setup() 里主要是初始化例如音频输入
- draw() 用于实现音乐可视化

这里取数据用到第三方库sound，loadSound加载指定的音频文件，sound.play()播放音乐。

```
function setup() {  
  sound = loadSound('Beautiful.mp3');  
  
  sound.play();  
  fft = new p5.FFT(0.3,16);  
  amp = new p5.Amplitude(0.5);  
  amp.setInput(sound);  
}
```

然后进行FFT分析得到频谱获取音频，同时通过振幅分析获取音量，方便可视化

```
function draw() {  
  // Get Spectrum  
  var spectrum = fft.analyze();  
  // Get volume  
  var volMain = amp.getLevel();  
  var volLeft = amp.getLevel(0);  
  var volRight = amp.getLevel(1);  
}
```

界面（布局、rgb值设置颜色）

红色波点高低反应频率

```
for (var i=0; i<spectrum.length; i++) {  
  var barwidth = width / spectrum.length;  
  var xPos = barwidth * i;  
  noStroke();  
  fill(255,0,0);  
  ellipse(xPos-(width/2)+100,spectrum[i],10,10);  
}
```

紫色圆圈大小反应音量

```
// circle  
ellipse(0,0,volMain * 1000, volMain* 1000);  
ellipse(-500,0,volLeft * 500,volLeft * 500);  
ellipse(500,0,volRight * 500,volRight * 500);  
  
noStroke();  
fill(106,90,205);  
  
ellipse(0,0,volMain * 1000, volMain* 1000);  
ellipse(-500,0,volLeft * 100,volLeft * 100);  
ellipse(+500,0,volRight * 100,volRight * 100);  
  
stroke(106,90,205);  
rotateX(frameCount*0.01);  
rotateY(frameCount*0.01);  
texture(img);  
sphere(volMain*400,20,20);
```

中间旋转的三个正方体增加趣味性

```
// 3D Boxes Rotation  
noFill();  
strokeWeight(2);  
stroke(106,90,205);  
rotateX(frameCount*0.01);  
rotateY(frameCount*0.01);  
box(200);  
  
noFill();
```

```

strokeWeight(2);
stroke(106,90,205);
rotateX(frameCount*0.015);
rotateY(frameCount*0.015);
box(200);

noFill();
strokeWeight(2);
stroke(106,90,205);
rotateX(frameCount*0.012);
rotateY(frameCount*0.012);
box(200);

texture(img);
rotateX(frameCount*0.01);
rotateY(frameCount*0.01);
sphere(100,50,10);
sphere(100,50,30);

```

2) 然后介绍Processing语言（即类似java语言做法）

该项目用到了processing的minim库

```

import ddf.minim.*;
import ddf.minim.analysis.*;

Minim minim;
AudioPlayer song;
BeatDetect beat;

minim=new Minim(this);
// song = minim.loadFile("Beautiful.mp3");
song = minim.loadFile("Huahuo.mp3");
// song = minim.loadFile("Draw Me A Sheep.mp3");
// song = minim.loadFile("Liangzhu.mp3");
//song = minim.loadFile("Without You I Am Dying.mp3");
song.play();

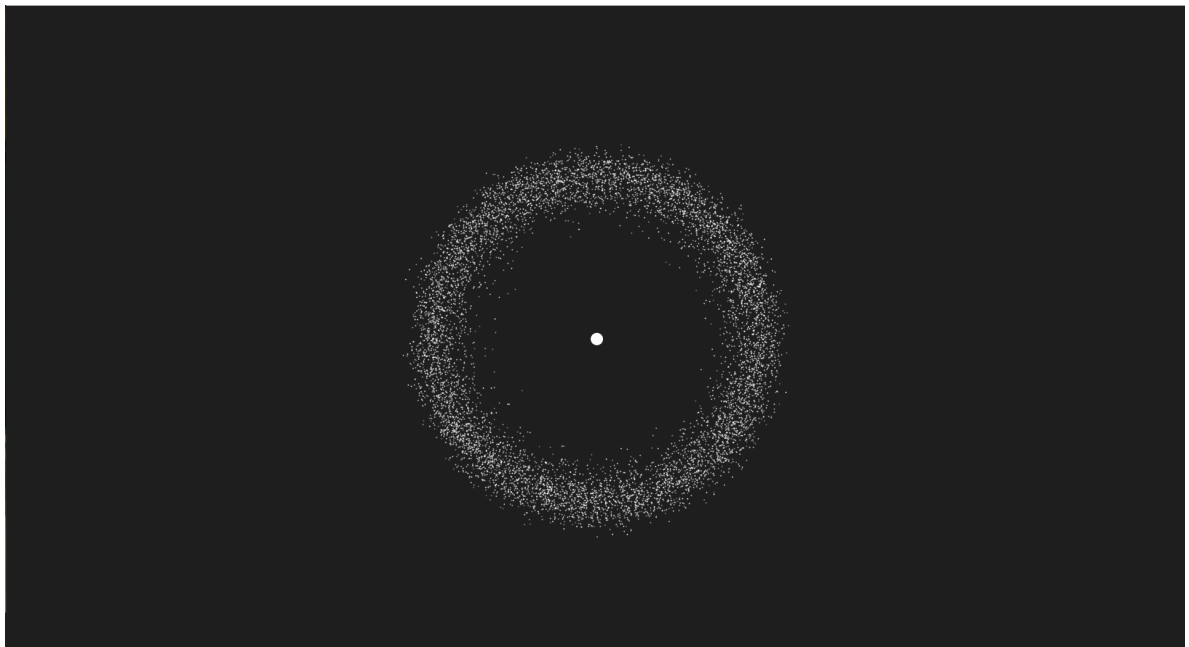
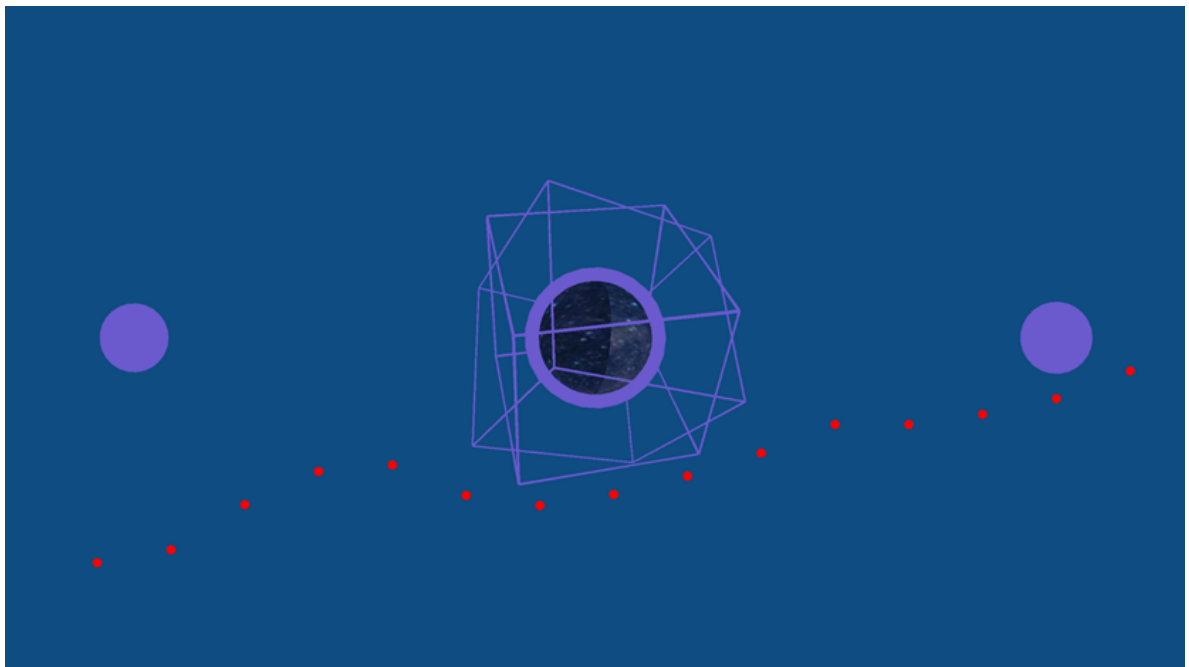
```

定义三个类

- PSystem：中心+所有粒子
- Particle：单个例子的位置、速度、加速度
- Repeller：粒子跟随鼠标移动方向、距离判断

界面是常见的粒子系统，单击后整个粒子系统跟随鼠标运动，再次单击固定，粒子系统距离中心的远度（圆的半径）体现了音量大小

- 运行效果



(具体实现效果见demo.html)

- 参考材料

[Processing编程艺术](#)

[Processing Sound library.](#)

[Processing Minim API](#)

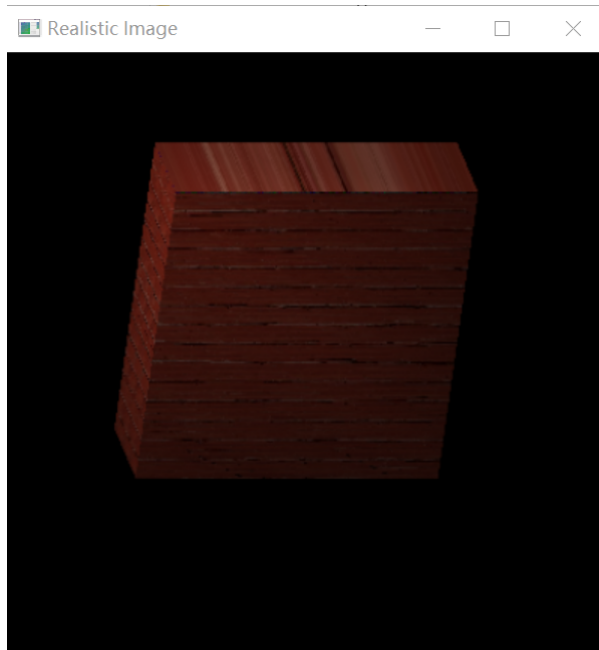
Project 2 编程画一个真实感静态景物

- 开发工具

python3+openGL

真实感图形需要考虑光源、材质：在画面左上角放置点光源；立方体表面使用原图纹理

- 效果展示



(具体见代码注释)

- 参考材料

图形学课本

[OpenGL基础图形编程（十）真实感图形基本概念](#)

Project 3 创作一个Flash动画

- 实现原理

制作工具：Adobe Animate CC 2020

首先绘制构图原件，插入关键帧，通过补间动画做出连贯效果。例如杯子里水的高度、摆动的电表和机器臂的上下移动

- 效果展示

