

**互动媒体课程设计报告**

**题目 TextObjects文字粒子特效**

**学院 数字媒体学院**

**专业 数字媒体技术**

**学号 1030515409**

**学生姓名 金卓群**

**指导老师 律睿慜**

**摘要**

现今，粒子特效有越来越广泛的应用前景，除了用于模拟现实世界的水、火、雾、气等效果，也应用于各种炫酷的特效，尤其是在网页端，对可视化的要求比较高，常会构建一些设计感很强的效果，或是模拟自然系统的粒子特效。

本次课程设计，主要是针对文字粒子特效这块，做了一些分析和设计。使用了processing作为开发软件，最终生成PC端的运行程序。

关键词：Processing、文字造型、粒子特效

**Abstract**

Nowadays, particle effects are more and more widely used, in addition to used to simulate the real world of water, fire, such as fog, gas effect, also applied to all kinds of cool special effects, especially in the web side, to the requirement of visualization is higher, often build some design sense is very strong, the effect of particle effects or simulated natural systems.

This course design, mainly aimed at the text particle special effects, did some analysis and design. Processing was used as the development software, and finally the PC running program was generated.

**Key words** : Processing, word modeling, particle special effects

目录

[第1章 绪论 5](#_Toc517605492)

[1.1 想法来源 5](#_Toc517605493)

[1.2 相关内容概述 5](#_Toc517605494)

[1.3 主要研究内容 5](#_Toc517605495)

[1.4 论文结构 6](#_Toc517605496)

[第2章 系统总体方案设计 7](#_Toc517605497)

[2.1 需求分析 7](#_Toc517605498)

[2.2 设计思路 7](#_Toc517605499)

[2.3 可行性分析 7](#_Toc517605500)

[第3章 系统呈现 8](#_Toc517605501)

[3.1 文字粒子形成 8](#_Toc517605502)

[3.2 粒子形态调节 9](#_Toc517605503)

[3.3 粒子变换动画 9](#_Toc517605504)

[3.4 粒子追随动画 9](#_Toc517605505)

[3.5 良好交互界面 9](#_Toc517605506)

[3.6 跟随音乐同频震动 11](#_Toc517605507)

[第4章 系统测试 12](#_Toc517605508)

[4.1 用户测试 12](#_Toc517605509)

[4.2 体验总结 12](#_Toc517605510)

[第5章 总结展望 14](#_Toc517605511)

[5.1 关于本系统的总结 14](#_Toc517605512)

[5.2 对未来发展的展望 15](#_Toc517605513)

[参考文献 16](#_Toc517605514)

# 绪论

## 想法来源

HTML5带来许多令人激动的新特性，在这之前的HTML中是不可能见到的。其中一个最值得提及的特性就是用于绘画的HTML Canvas，可以对2D或位图进行动态、脚本的渲染，必将是以后动画，游戏等应用首选的开发方案。[1]

而P5.JS（为艺术家设计师而开发的Javascript库）和Processing（在电子艺术环境下介绍的程序语言）的出现，更是让动画的实现更加简洁和人性化。本文所设计的粒子特效，就是用Processing重现一些canvas文字粒子特效，并在此基础上添加了自己的粒子动画设计。

## 相关内容概述

Processing是一种具有革命前瞻性的新兴计算机语言，它的概念是在电子艺术的环境下介绍程序语言，并将电子艺术的概念介绍给程序设计师。它是 Java 语言的延伸，并支持许多现有的 Java 语言架构，不过在语法 (syntax) 上简易许多，并具有许多贴心及人性化的设计。Processing 可以在 Windows、MAC OS X、MAC OS 9 、Linux 等操作系统上使用。目前最新版本为Processing 3。以 Processing 完成的作品可在个人本机端作用，或以Java Applets 的模式外输至网络上发布。

虽然图形用户界面(GUI)早在二十年前成为主流，但是基础编程语言的教学到今天仍是以命令行接口为主，学习编程语言为什么要那么枯燥呢？人脑天生擅长空间辨识，图形用户界面利用的正是这种优势，加上它能提供各种实时且鲜明的图像式反馈 (feedback)，可以大幅缩短学习曲线，并帮助理解抽象逻辑法则。举例来说，计算机屏幕上的一个像素(pixel) 就是一个变量值(the value of a variable) 的可视化表现。Processing将Java的语法简化并将其运算结果“感官化”，让使用者能很快享有声光兼备的交互式多媒体作品。

Processing的源代码是开放的，和近来广受欢迎的Linux 操作系统、Mozilla浏览器、或Perl语言等一样，用户可依照自己的需要自由裁剪出最合适的使用模式。Processing的应用非常丰富，而且它们全部遵守开放源代码的规定，这样的设计大幅增加了整个社群的互动性与学习效率。

## 主要研究内容

本文所论述的粒子特效，主要针对文字粒子特效的形成，变换，以及对Processing所提供的GUI库做一个具体介绍和应用，构建一个交互粒子系统。

## 论文结构

本论文的内容安排如下：

第一章为绪论部分，主要介绍了关于本课题的想法来源，相关内容概述，以及本文主要研究内容．

第二章为应用的总体方案设计，进行了本论文研究内容的需求分析和可行性分析，详细设计方面，介绍了应用的框架构成和功能、各模块的详细功能设计，并分析了涉及到的关键技术．

第三章为整个文字粒子系统的实现细节．

第四章是系统测试部分，介绍了测试结果和用户体验．

第五章为总结，分析了系统的创新点、对当前工作的总结以及对今后工作的展望．

# 系统总体方案设计

## 需求分析

根据本系统所要呈现的效果，需要满足如下需求：交互数据输入，控制粒子生成，对粒子字体的调节，设计变换动画，自治体跟随，良好的交互界面。

## 设计思路

现今对于图形图像的canvas应用已经很广泛，但对于文字的特效的实现会比较困难，一是各种语言符号之间没有相同的输入规律，二是不同语言也有不同的字体形态，本系统希望通过重构粒子的方式来解决这些问题，让粒子组成任意字符任意字体。

为了有良好的呈现效果和有趣的交互模式，还应用了一些数学公式，嵌入了对粒子特效运动、转换时的特效，以及粒子跟随自治体的效果。

## 可行性分析

Processing是一种具有革命前瞻性的新兴计算机语言，它的概念是在电子艺术的环境下介绍程序语言，并将电子艺术的概念介绍给程序设计师。它是 Java 语言的延伸，并支持许多现有的 Java 语言架构，不过在语法 (syntax) 上简易许多，并具有许多贴心及人性化的设计。简介的语言架构，友好的设计界面，都能够让我们更容易的实现出我们所设计的交互系统。

Processing的开源使其维持了良好的开发生态循环，许多设计师程序员投入其中，编写各种功能的库，来进一步满足用户的设计需求，包括本系统用到的P5Control——Processing的一个GUI库。

# 系统呈现

## 文字粒子形成

文字粒子的构成的主要思路是先获得画布上文字的每个像素点的位置，从而在此位置构造粒子。

第一步，获取画布上的像素点信息，在不同的工作环境下有不同的解决方法。例如在JS中，可以利用context.getImageData()[获取某点内存信息]；在P5.js中，在Image层面有一个get(x,y)函数[获取某点颜色值]，但这个函数十分的耗内存；在Processing中也有一个类似的get函数，表现效果要好于P5；

此外，对语言的支持上，P5.js要好于Processing，我们利用一些GUI控件进行文字输入时，Processing所提供的GUI库中的控件不支持中文输入，而P5则可以。

但从测试的流畅度来评估，我最终选择了Processing作为开发工具。

void createText(String s) {

text(s, rectWidth/2, rectHeight/2);

}

void startModule() {

int count = 0; //粒子个数

index = 0; //遍历粒子

for (int i = 0; i<rectWidth; i += gridX ) {

for (int j = 0; j<rectHeight; j += gridY ) {

color c = get(StartX+i, StartY+j);

if (c == color(255, 255, 250)) {

count++;

}

}

}

mods = new Module[count];

for (int i = 0; i<rectWidth; i += gridX ) {

for (int j = 0; j<rectHeight; j += gridY ) {

color c = get(StartX+i, StartY+j);

if (c == myTextColor) {

mods[index] = new Module(StartX+i, StartY+j, random(-3, 3), index, colors[int(random(0, 15))]);

mods[index].update(50, 5, 5);

mods[index].display(0, 0, 0, iseffects);

index++;

}

}

}

}

## 粒子形态调节

在构建粒子类的时候，我们提供了几个接口来控制粒子运动的周期相位半径大小：

chaUnit = (\_banjing+\_unit)\* sin(PI/\_zhouqi \* tt + \_index\*(xiangwei/(2\*PI)));

ellipse(0, 0, chaUnit, chaUnit);

## 粒子变换动画

粒子群在生成和刷新位置时，未避免过分僵硬而设置的一些转场效果。

效果一：将每个粒子的当前位置映射到另一块区域（缩小或扩大），类似一些群体运动。

map(mods[i].\_xOffset, a1, a2, (a2-a1)/random(2, 5)+a1, a2-(a2-a1)/random(2, 5))

效果二：粒子除了受到目标所带来的引力，在运动过程中还受到一个向心力，这个力的方向指向粒子群中央。当粒子已经靠近目标位置时，消去这个力。

applyForce(PVector.sub(PVector.sub(newPVector(\_xOffset+target.x-rectWidth/2, \_yOffset+target.y-rectHeight/2), \_position), velocity));

## 粒子追随动画

给当前粒子群设定一个目标，粒子群可以通过寻找目标，得到所需速度的方向和大小，直到到达目标。

applyForce(PVector.sub(PVector.sub(target, \_position), velocity));

acceleration.add(force);

velocity.add(acceleration);

velocity.limit(maxspeed);

\_position.add(velocity);

acceleration.mult(0); //与0相乘 重置

## 良好交互界面

P5Control库的应用——处理Canvas上构建图形用户界面的控制器,包括滑块、按钮、切换、旋钮、textfield、radiobutton、复选框等，可以很容易地添加到处理草图中。它们可以在单独的控制PGraphics上下文中进行安排，也可以在选项卡或组中进行组织.

void startUI() {

//文本输入框

cp5 = new ControlP5(this);

//创捷UI组

\_group = cp5.addGroup("")

……

;

RadioButton r\_effects;

r\_effects = cp5.addRadioButton("radioButton2")

……

.addItem("effect1", 0)

.addItem("effect2", 1)

.setGroup(\_group)

;

t1\_effect = r\_effects.getItem("effect1"); //设置按钮的图片

t1\_effect.setImages(loadImage("button\_yi\_F.png"), loadImage("button\_yi\_T.png"), loadImage("button\_yi\_T.png"));

t2\_effect = r\_effects.getItem("effect2");

t2\_effect.setImages(loadImage("button\_er\_F.png"), loadImage("button\_er\_T.png"), loadImage("button\_er\_T.png"));

r\_effects.activate(0);

cp5.addTextfield("\_input") //输入框

……

.setGroup(\_group)

;

cp5.addSlider("\_zhouqi") //滑块

……

.setGroup(\_group)

;

cp5.addSlider("\_xiangwei")

……

.setGroup(\_group)

;

cp5.addSlider("\_banjing")

……

.setGroup(\_group)

;

}

## 跟随音乐同频震动

ddf.minim库的应用——对声音文件进行分析，获得当前缓存下的声音信息，从而绘制波形，将波形与粒子位置进行联系，从而得到振动效果。

minim = new Minim(this);

kick = minim.loadFile("BD.mp3");

if ( kick == null ) println("Didn't get kick!");

int j = (int)map(i, 0, index-1, 0, kick.bufferSize()-1);

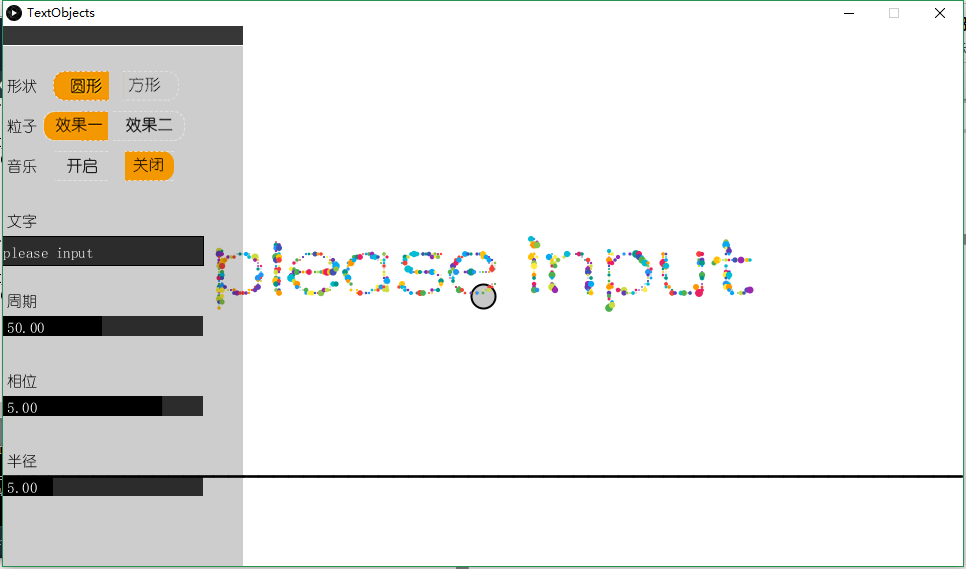
float kickmix = 20\*kick.mix.get(j/2);

translate(0, -wave);

# 系统测试

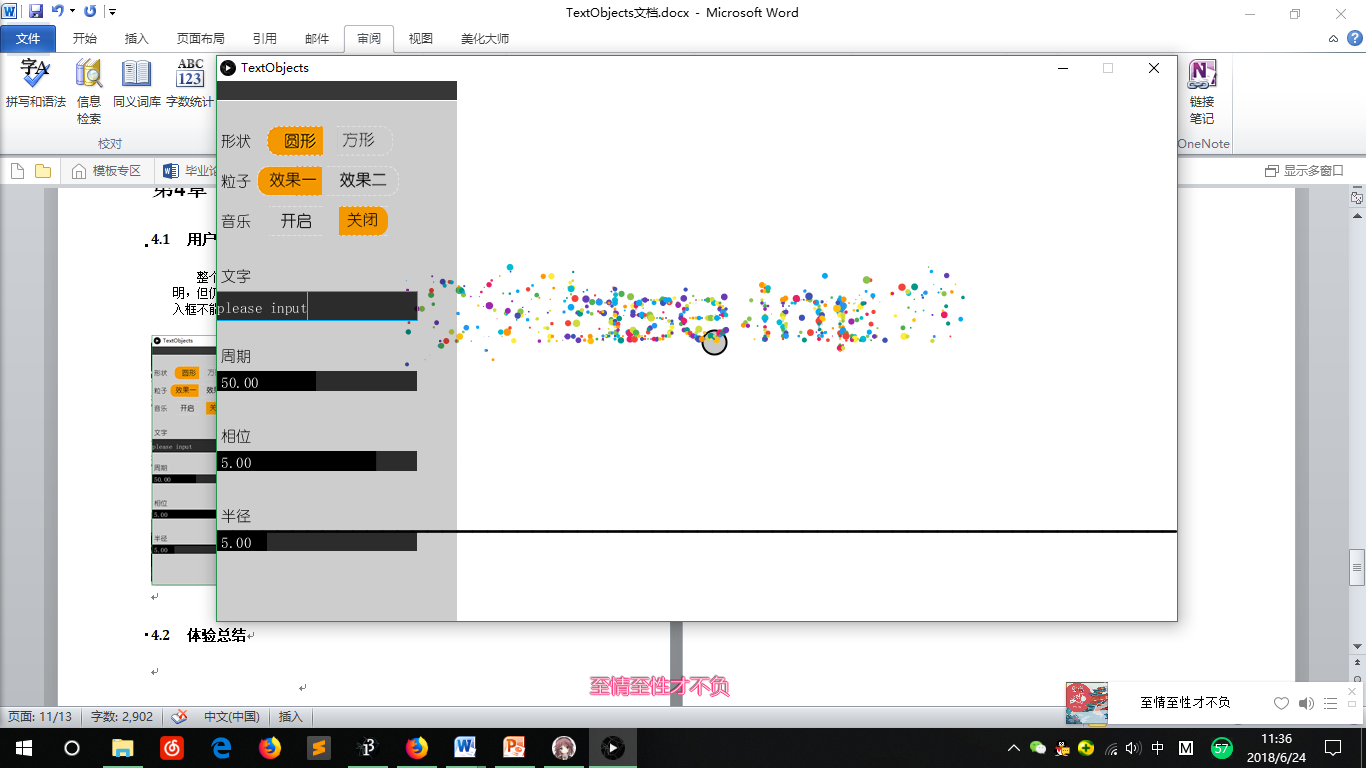
## 用户测试

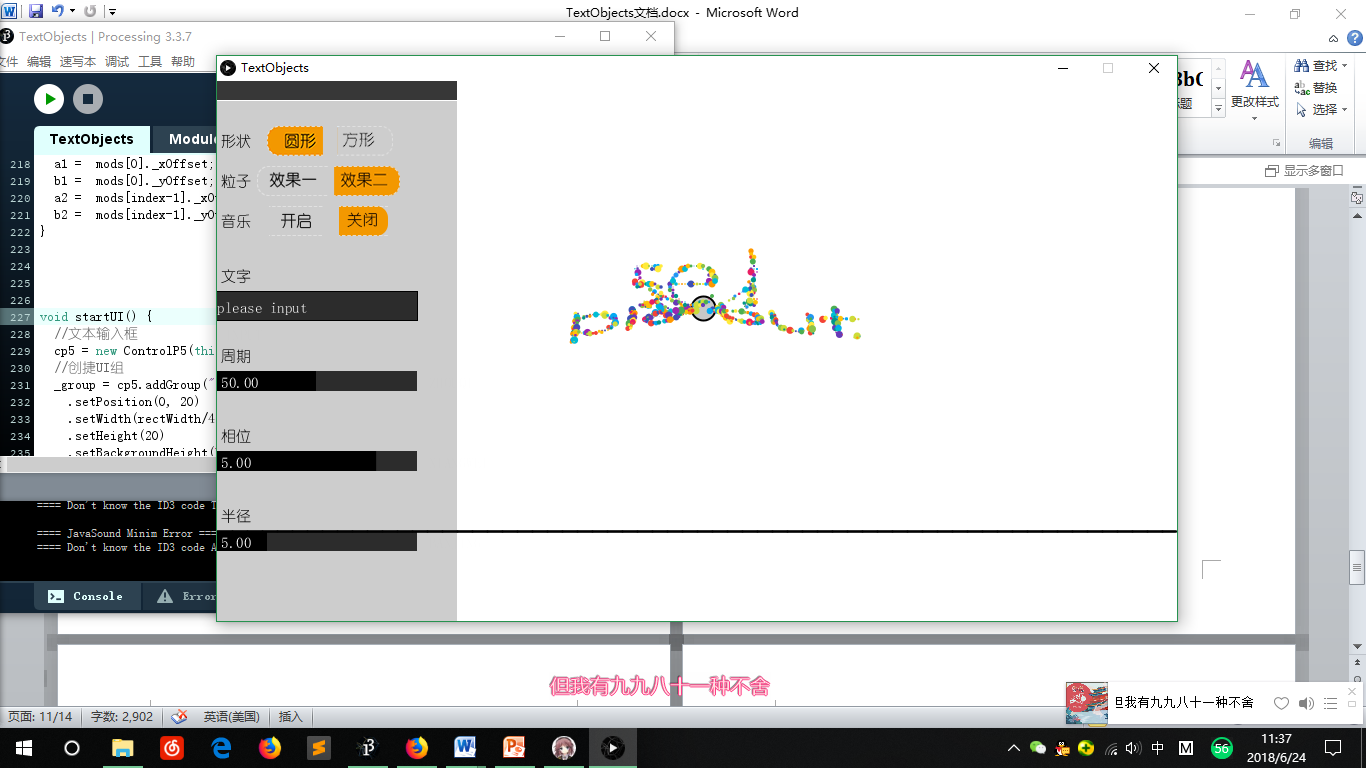
整个系统运行情况良好，没有什么BUG，所有功能也在侧边栏有交互说明，但仍有些设计需要改进：侧边栏收起后，自治体不能移动到该区域；输入框不能输入中文，以及呈现功能较少。

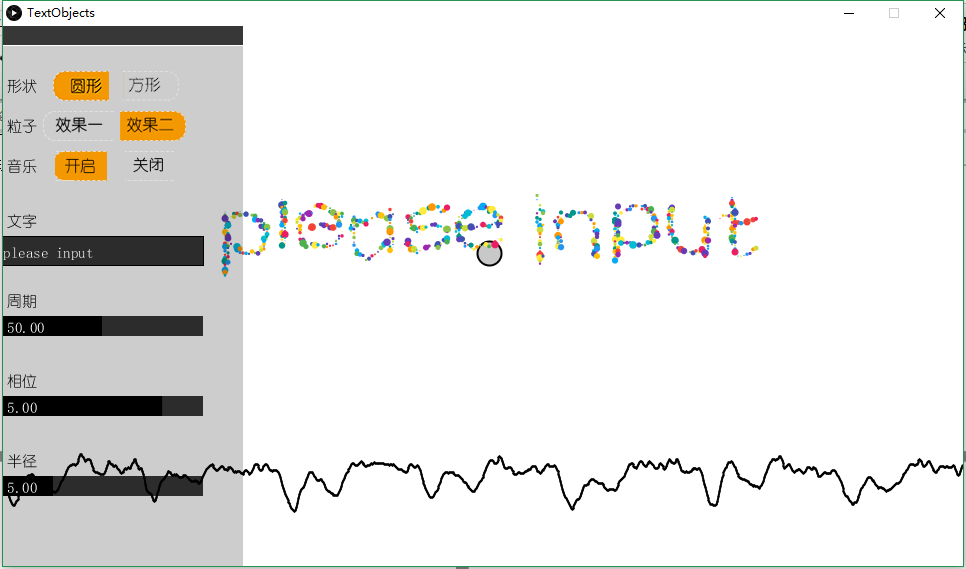


## 体验总结

粒子效果呈现效果良好，但内容较少，较单一，需改进。





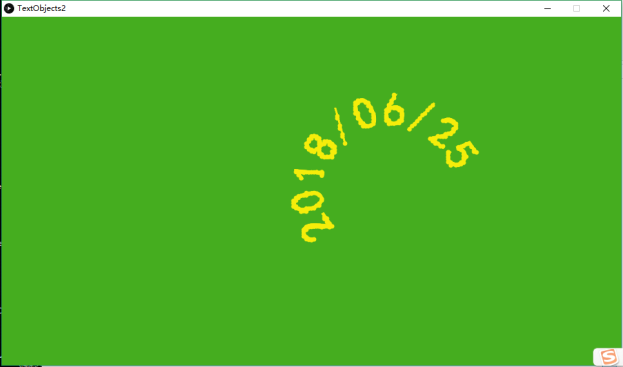


# 总结展望

## 关于本系统的总结

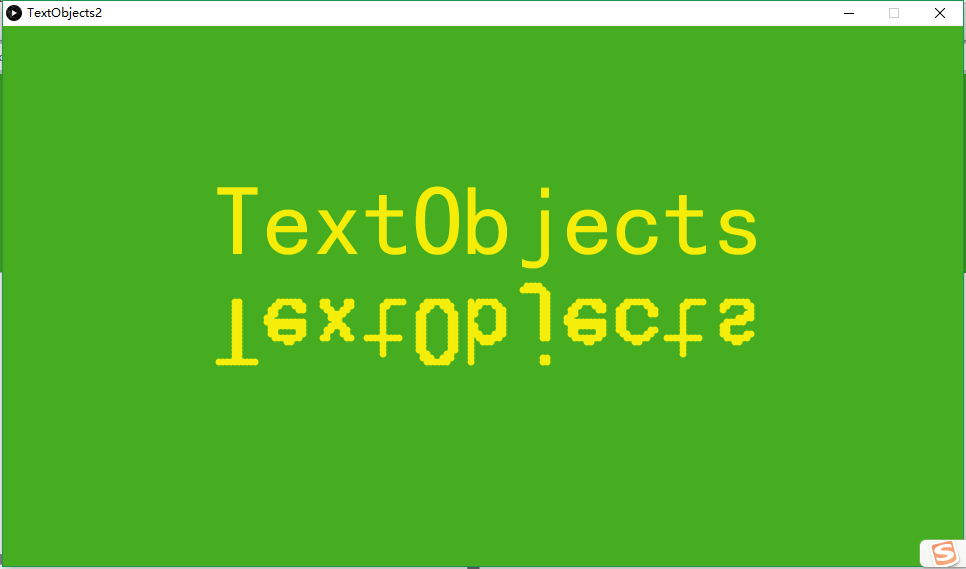
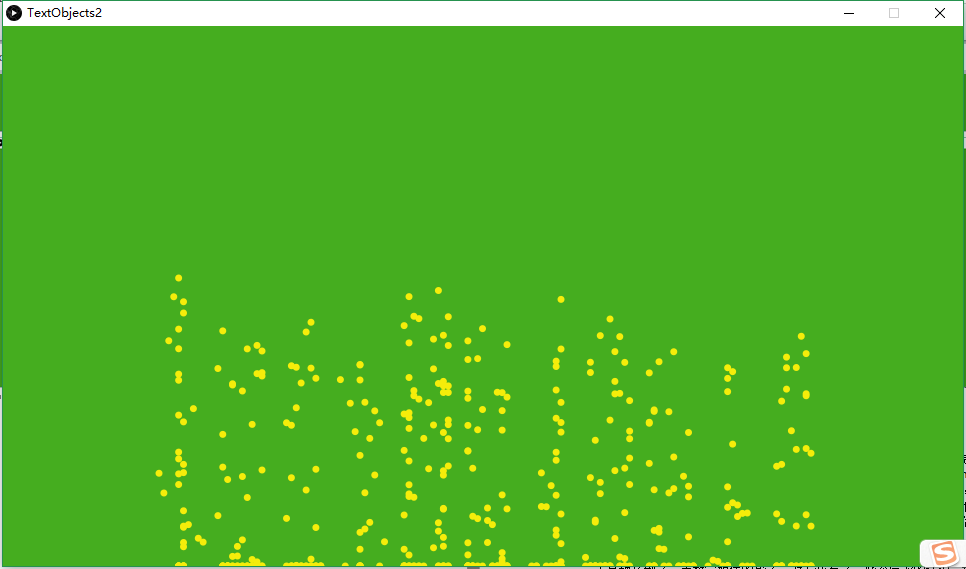
对Processing所提供的GUI库及声音文件分析库有了较好的应用，粒子动画的呈现效果也OK，但呈现的内容少可能较少，希望以后有时间能加以总结改进。

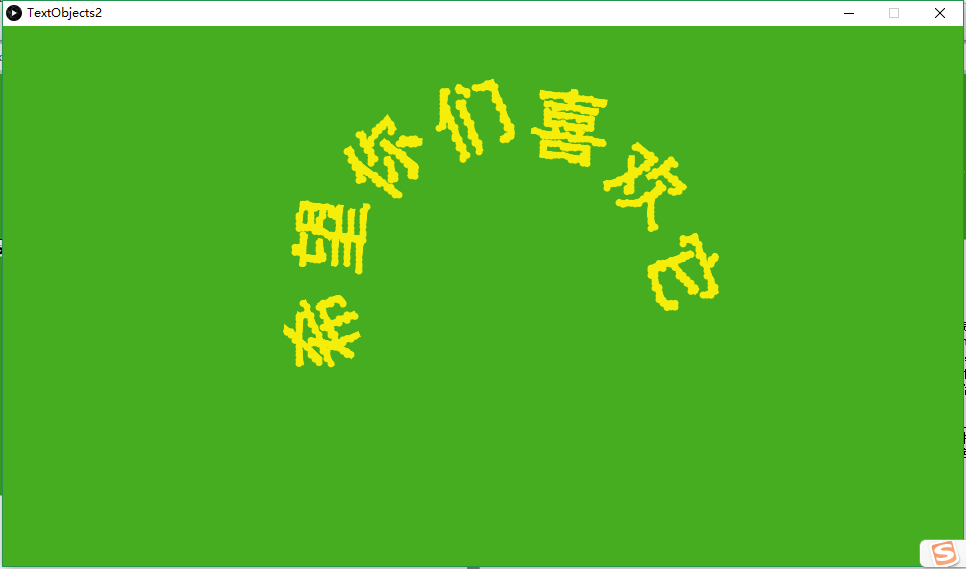
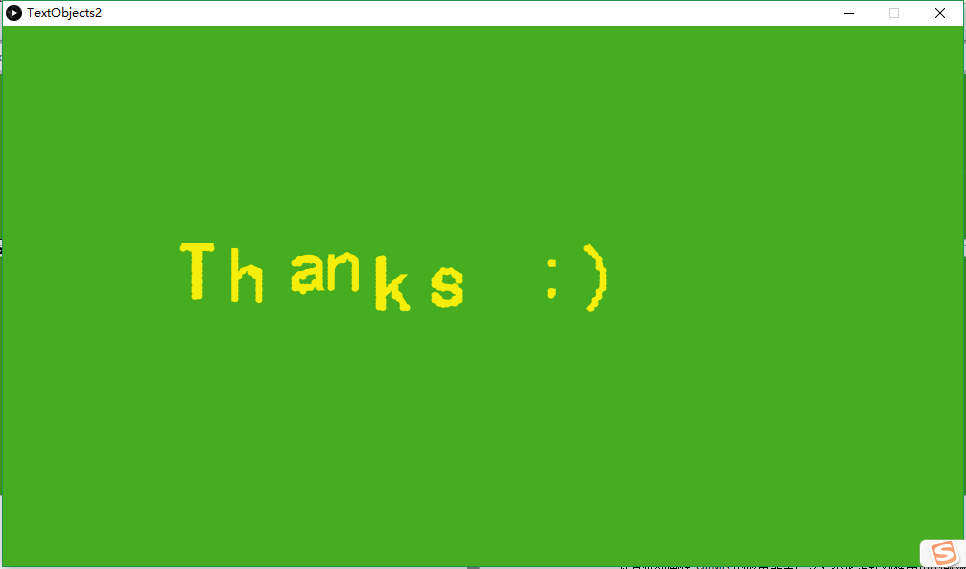
在此基础上也根据老师给的建议进行了视觉上的加强，在文字造型和动效方面做了改进。

## 对未来发展的展望

计算机动画在Canvas的应用非常广泛，不论是针对程序员的前端设计，还是设计师进行交互媒体设计，从大二开始也看了一些原生JS和Canvas之间应用的书籍，包括接触到计算机动画这门课、P5\Processing的学习，从最开始的模仿到渐渐有了自己设计的思路，希望未来走进这个行业的时候，能在这一方面有所发展，但在这之前也会不断的提高自身技术水平，为做一个有设计思想的前端。

迷茫的大三也终于要结束了，在互动媒体和计算机动画这两门课上，终于是触及到了一点数字媒体的影子，对专业有了一些深层次的认识，接触到了这个领域的人都在做着怎样有趣的事。有机会的话希望能做一个 #每天一个Canvas动画# 的系列，像这样有趣而生动的事真令人感到高兴。

# 参考文献

1. 王昆. 基于HTML5 Canvas的画图板设计与实现. 2013

[2] 陆汝钤,张松懋. 从故事到动画片——全过程计算机辅助动画自动生成[J]. 自动化学报. 2002(03)

[3] 张继生. 聊聊编程动画的两种类型[J]. 计算机与网络. 2017(14)

[4] 梁速,朱晟仁. 浅析粒子系统及其在游戏中的应用[J]. 电脑知识与技术(学术交流). 2007(04)