**期末作业 多种动画技术的组合应用**

**课程名称： 计算机动画编程技术 实验日期： 2018－06－05**

**班级： 数媒1504 成员： 王林超（单人组） 学号： 1030515424**

**一、概述**

组队完成一个运用多种动画技术的应用

**二、要求：**

1. 组队完成，每组不超过3人；

2. 必须用从参考教材中0~4，6~10章节中选取多章，章节数最少为（2+队员人数），运用这些章节讲述的技术，其中必须有第6章及之后的技术；单个组员必须至少负责一项动画技术。如果感到确实难以运用第六章后的技术，则可以不直接运用，而是每个组员自行完成3份动画程序及对应的实验报告，对应6、7、8章。

3. 从后续”选题列表“中选定主题，程序必须符合该题目的特别要求；

4. 开发工具可选用Unity，Processing，P5，Openframeworks，若想选用其他语言或工具，须14周内向老师申请；

5. 需要区分出各个组员的工作内容（在演示和设计报告中描述）；

**三、总体构思**

个人比较喜欢以游戏交互的形式来体现技术。

所以在选题列表里选择了“艺术化游戏”。

希望通过游戏来体现分形艺术。

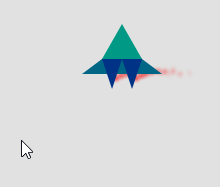
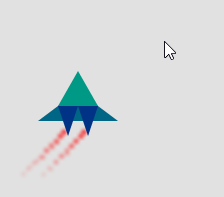
于是就做了这个飞机游戏。

**四、动画技术**

运用了三项动画技术。

1.（第1章）向量

示例图片:

通过计算鼠标与飞机类之间的向量，来确定飞机的移动方向。

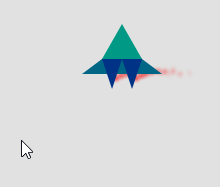
（移动方向可以通过尾部粒子系统看出）

并且使用向量归一化，使得飞机的移动速度固定，使游戏更加合理。

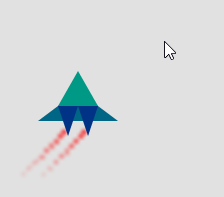
（一开始是使用鼠标位置来直接确定飞机的位置，不够合理）。

2.（第4章）粒子系统

示例图片:







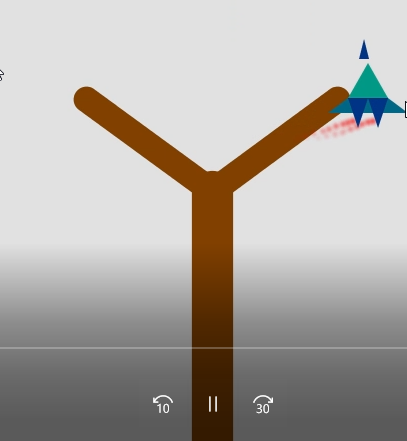
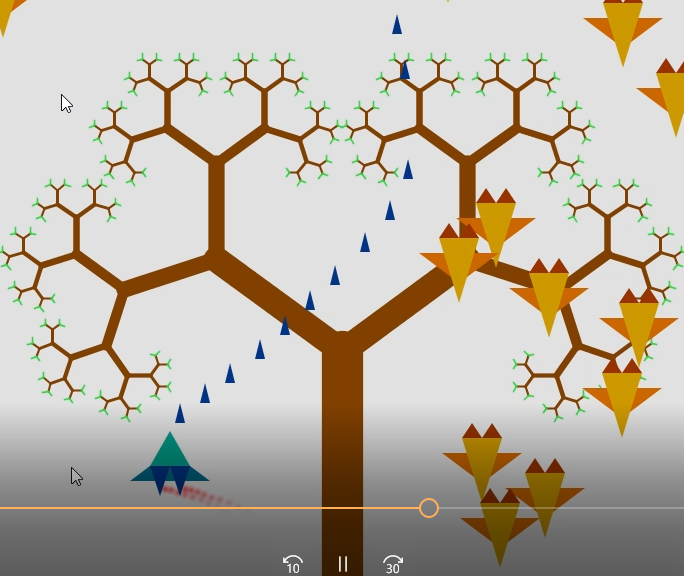
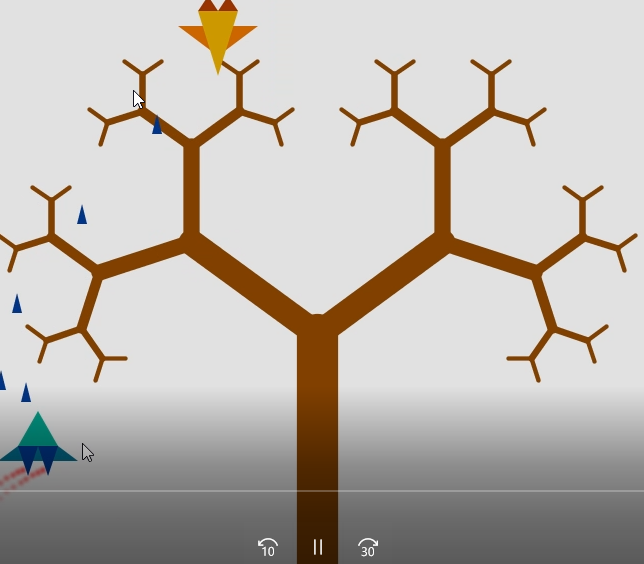
为了使玩家的飞机更加生动

在飞机的尾部添加了两个粒子系统。

粒子使用的是红色的纹理图案，模拟出红色的气流。

并且飞机往哪运动，粒子就会受到相应的风力，使效果更真实。

3.（第8章）分形

通过射击得分来获取材料构造分形树。

**五、系统规划**

用pocessing做的，所以得利用好java类

一共附加了5个class

分别是plane(飞机本体),bullet(子弹),enemy(敌机),

particle(粒子), particleSystem(粒子系统)

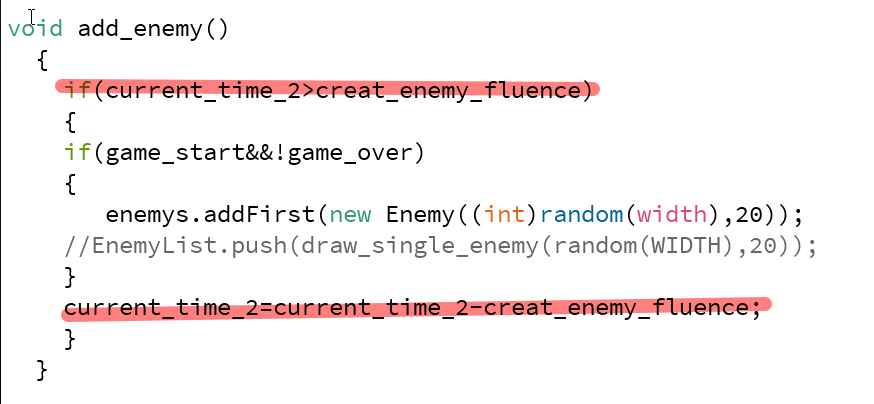
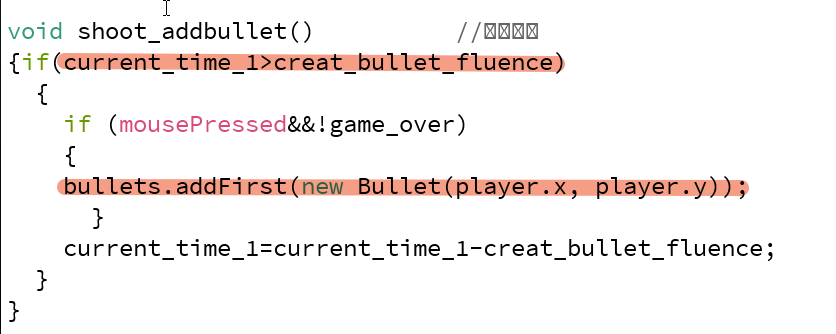
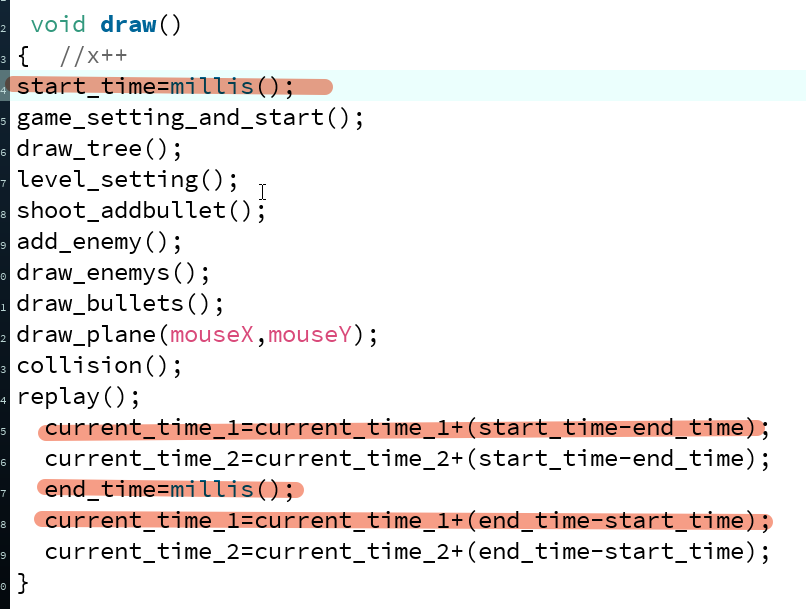
并把与之相关的函数写在类下,

**六、技术介绍、**

**1.计时器**

**1.1实现效果/功能简介:使敌机和子弹按给定的频率生成**

**1.2相关代码:**



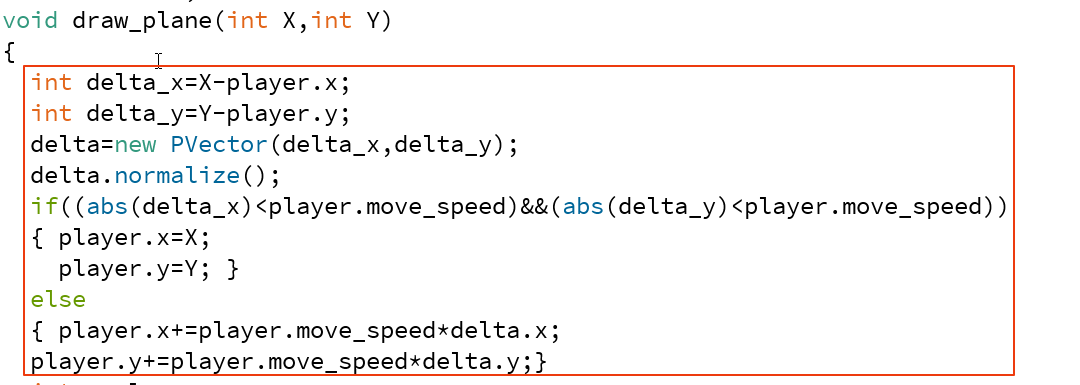
**2.向量**

**2.1实现效果/功能简介:实现对plane的控制,**

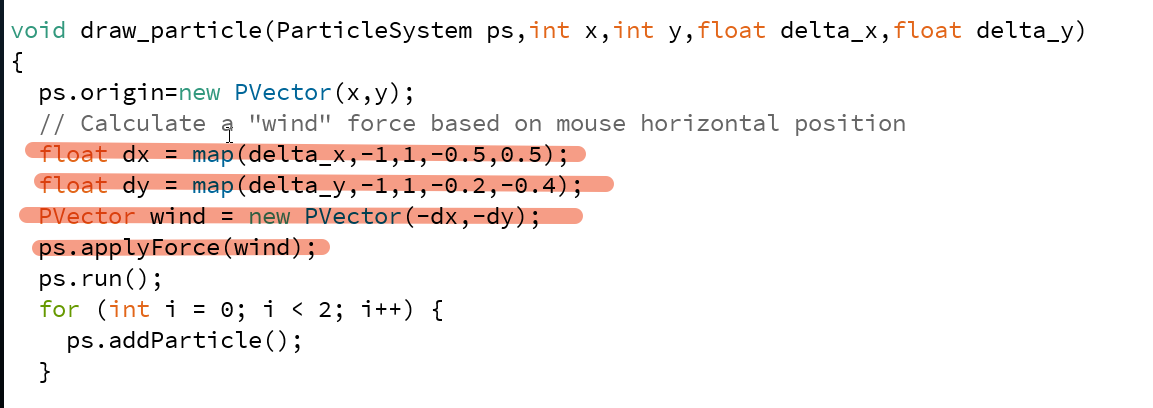
**以及对粒子系统施加风力详见第2、3页的图**

**2.2相关代码:**

飞机控制



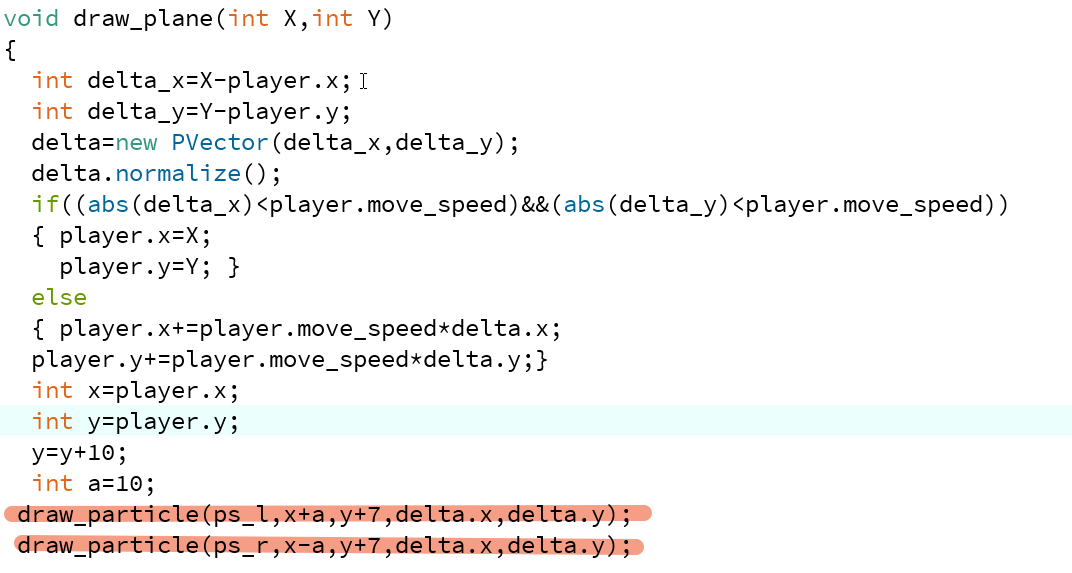
施加风力



**3.粒子系统**

**3.1实现效果/功能简介:在飞机后模拟喷气，详见第3页的图**

**3.2相关代码:**



**Particle类**

class Particle {

PVector pos;

PVector vel;

PVector acc;

float lifespan;

PImage img;

Particle(PVector l,PImage img\_) {

acc = new PVector(0,0);

float vx = randomGaussian()\*0.1;

float vy = randomGaussian()\*0.1 ;

vel = new PVector(vx,vy);

pos = l.get();

lifespan = 100.0;

img = img\_;

}

void run() {

update();

render();

}

void applyForce(PVector f) {

acc.add(f);

}

void update() {

vel.add(acc);

pos.add(vel);

lifespan -= 10;

acc.mult(0); // clear Acceleration

}

void render() {

imageMode(CENTER);

tint(255,lifespan);

image(img,pos.x,pos.y);

}

// Is the particle still useful?

boolean isDead() {

if (lifespan <= 0.0) {

return true;

} else {

return false;

}

}

}

**ParticleSystem类**

class ParticleSystem {

ArrayList<Particle> particles; // An arraylist for all the particles

PVector origin; // An origin point for where particles are birthed

PImage img;

ParticleSystem(int num, PVector v, PImage img\_) {

particles = new ArrayList<Particle>(); // Initialize the arraylist

origin = v.get(); // Store the origin point

img = img\_;

for (int i = 0; i < num; i++) {

particles.add(new Particle(origin, img)); // Add "num" amount of particles to the arraylist

}

}

void run() {

for (int i = particles.size()-1; i >= 0; i--) {

Particle p = particles.get(i);

p.run();

if (p.isDead()) {

particles.remove(i);

}

}

}

// Method to add a force vector to all particles currently in the system

void applyForce(PVector dir) {

// Enhanced loop!!!

for (Particle p: particles) {

p.applyForce(dir);

}

}

void addParticle() {

particles.add(new Particle(origin, img));

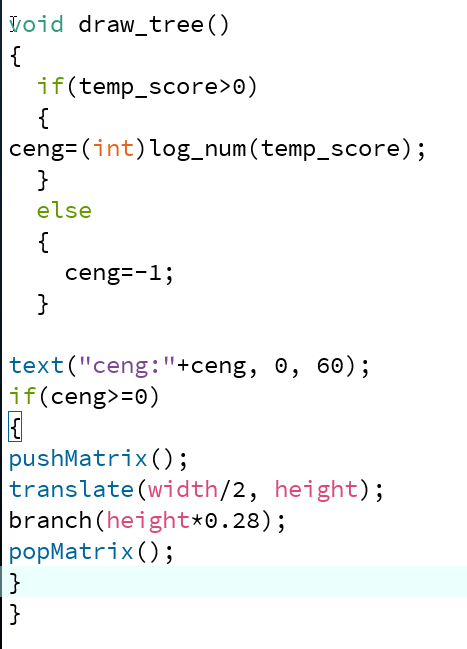
}

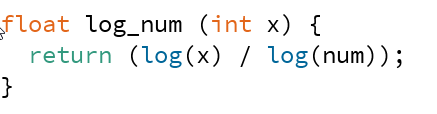
}

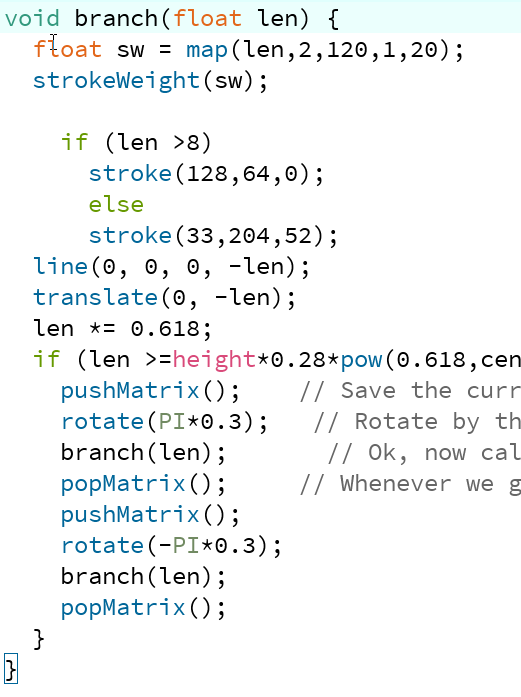
**4.分形**

**4.1实现效果/功能简介: 根据分数构造分形树**

**4.2相关代码:**







**5.碰撞检测**

**5.1实现效果/功能简介:判断子弹和敌机接触/得分,飞机和敌机接触/game over**

**5.2相关代码:**

