

**互动媒体设计课程报告**

**题目：** 艺术化游戏设计

数字媒体 学院 数字媒体技术 专业

学 号 1030515424

学生姓名 王林超

指导教师 律睿慜

# 摘 要

在玩了逆转飞行这个独立游戏后,尝试做一个雷电类型的飞行射击游戏,同时希望游戏能更有趣味和艺术表现力,能够让玩家在玩游戏的过程中，逐渐生成艺术效果最终呈现为一幅作品,于是设计开发了这个飞行射击游戏。  
 本游戏采用processing的开发框架,以计时器，碰撞检测为游戏的核心机制,分别用于生成和消灭飞机。在此基础上添加玩家的控制机制, 游戏UI面板系统,用于表现飞机尾焰的粒子系统,以及以生成分形体现玩家分数的分形树，未来希望可以继续丰富游戏中的关卡深度,艺术表现效果,更多种类的敌人和武器。

**关键词**：游戏设计；计算机动画；粒子系统；分形系统

# 目 录

[第1章 绪论 5](#_Toc517513634)

[1.1 引言 5](#_Toc517513635)

[1.2 相关游戏介绍 5](#_Toc517513636)

[1.3 主要研究内容 6](#_Toc517513637)

[1.4 论文结构 6](#_Toc517513638)

[第2章 游戏的总体设计方案 7](#_Toc517513639)

[2.1 游戏设计思路 7](#_Toc517513640)

[2.2 游戏详细设计 7](#_Toc517513641)

[第3章 游戏实现 9](#_Toc517513642)

[3.1.计时器 9](#_Toc517513643)

[3.1.1实现效果/功能简介 9](#_Toc517513644)

[3.1.2核心代码: 9](#_Toc517513645)

[3.2.向量 10](#_Toc517513646)

[3.2.1实现效果/功能简介 10](#_Toc517513647)

[3.2.2核心代码: 10](#_Toc517513648)

[3.3.粒子系统 11](#_Toc517513649)

[3.3.1实现效果/功能简介 11](#_Toc517513650)

[3.3.2核心代码: 12](#_Toc517513651)

[3.3.3Particle类 12](#_Toc517513652)

[3.3.4ParticleSystem类 13](#_Toc517513653)

[3.4.分形 15](#_Toc517513654)

[3.4.1实现效果/功能简介 15](#_Toc517513655)

[3.4.2核心代码: 16](#_Toc517513656)

[3.5.碰撞检测 17](#_Toc517513657)

[3.5.1实现效果/功能简介 17](#_Toc517513658)

[3.5.2核心代码: 17](#_Toc517513659)

[第4章 总结与展望 18](#_Toc517513660)

[4.1 本文内容的总结 18](#_Toc517513661)

[4.1.1 课题创新点 18](#_Toc517513662)

[4.1.2 工作总结 18](#_Toc517513663)

[4.2 对今后工作的展望 18](#_Toc517513664)

[参考文献 19](#_Toc517513665)

[致 谢 20](#_Toc517513666)

# 第1章 绪论

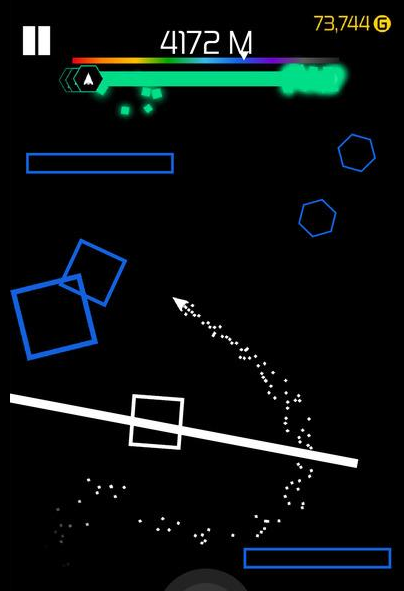
## 1.1 引言

近年来独立游戏日趋盛行,在上过本学期的游戏设计课程后,希望结合动画设计课程,设计一个体现2D艺术效果的艺术化游戏。

在玩过数款独立游戏后，我选择做以雷电游戏为主体，并通过processing易于展示代码艺术效果的特性，来实现一款能够让玩家在玩游戏的过程中实现艺术效果的游戏。

## 1.2 相关游戏介绍

1.（左图）微信飞机大战——简易的手绘风雷电游戏。

2.（右图）逆转飞行————通过向后发射粒子并用粒子攻击的创新雷电游戏

## 1.3 主要研究内容

1．实现游戏的核心机制[1],如时间控制,关卡及计分系统,玩家控制系统,敌机及子弹生成和消灭系统,碰撞检测。

2．实现多种不同的计算机动画效果,包括粒子系统，分形系统

**1.4 论文结构**

本论文的内容安排如下：

第一章为绪论部分，主要介绍了本设计的设计背景,想法来源，相关游戏介绍和主要研究内容，阐述了其内容以及意义．

第二章为游戏的总体设计方案，介绍了游戏设计思路、各部分的功能设计，并分析了涉及到的关键技术．

第三章为本游戏各模块的关键功能介绍及实现细节，针对主要功能的实现代码进行了详细说明，对于代码内最关键的几行也予以红线划出．

第四章为总结，分析了设计的创新点、对当前工作的总结以及对今后工作的展望．

# 第2章 游戏的总体设计方案

## 2.1 游戏设计思路

当前的人们对于游戏的要求日趋提高，将艺术效果融入游戏已经是很多独立游戏的选择，我也尝试进行这样的艺术化游戏开发。在本游戏开发中，使用这样一种方法实现游戏和技术的融合，玩家通过鼠标操作我方飞机击落敌机获取材料，材料越多，即可生成越复杂的分形效果。

## 2.2 游戏详细设计

通过上述分析,游戏的实现分成三个部分,如下图所示。

分形图案

键鼠交互

飞行射击游戏

数据输入

动画输出

游戏的输入也是游戏的交互方式[2],可以通过键盘控制游戏的开始,结束,鼠标来控制我方飞机的位置,鼠标是否按下来控制子弹的发射。

游戏的核心即飞行射击游戏，核心机制为

1按给定的参数控制生成敌方飞机的频率以及位置,

2按输入的参数控制我方飞机的位置以及在我方飞机对应位置生成炮弹

3.敌机与我方飞机碰撞事件,敌机与炮弹的碰撞事件

对于第一个机制,敌方飞机的生成可以用processing自带的draw函数每帧生成,但这样就无法控制生成频率来调节关卡难度，而且每帧生成一次的频率过高，不仅显示效果不好，而且对于计算机的负担也很大。所以选择用计时器来解决这个问题，并且计时器可以根据关卡参数来调节生成频率，同样的办法也可用于炮弹的生成。敌方飞机的运动模式相对固定,即由上到下，在敌机飞出屏幕时删除该对象，从而节省机器的计算量，炮弹同理。

对于第二个机制,我方飞机可以通过获取mouseX,mouseY参数来确定,并在相应位置实时绘制我方飞机,也可以在相应位置实时生成炮弹,并用计时器控制炮弹的生成频率。

对于第三个机制,采用相对简单暴力的距离计算来进行碰撞检测，即对于每一架敌机，计算其检测点与子弹检测点和我方飞机检测点的距离,距离小于某一参数视为碰撞发生，则可用代码控制相应事件发生如游戏结束，分数增加，子弹与敌机抵消。

在这三个核心机制上添加了，粒子系统实现的我方飞机尾焰[3]，通过将一张给定的图片作为粒子,实现了很逼真的尾焰效果，即在飞机朝某一方向运动时,尾焰也会受到相应的风力。

最后一部分为游戏输出的分形图案[4]，选择用分形树的形式展示，可以根据玩家分数的变换动态生成，玩家击落敌机越多，就可以生成越复杂，越细致的分形树，玩家在游戏结束时即可获得一幅分形树的艺术图案。

# 第3章 游戏实现

## 3.1.计时器

### 3.1.1实现效果/功能简介

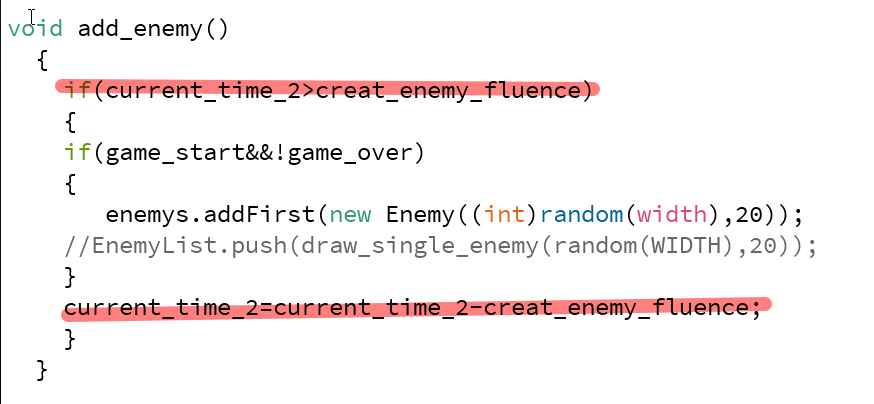
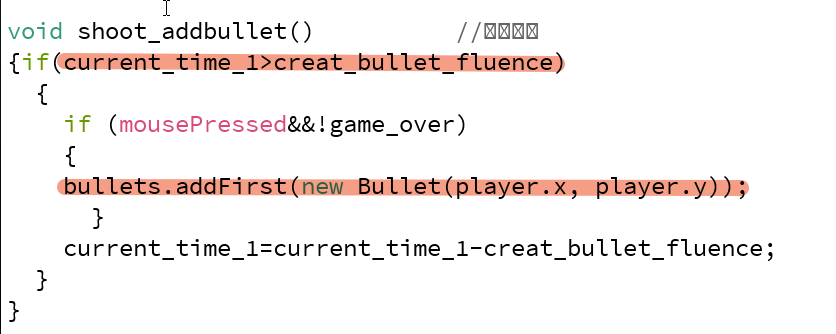
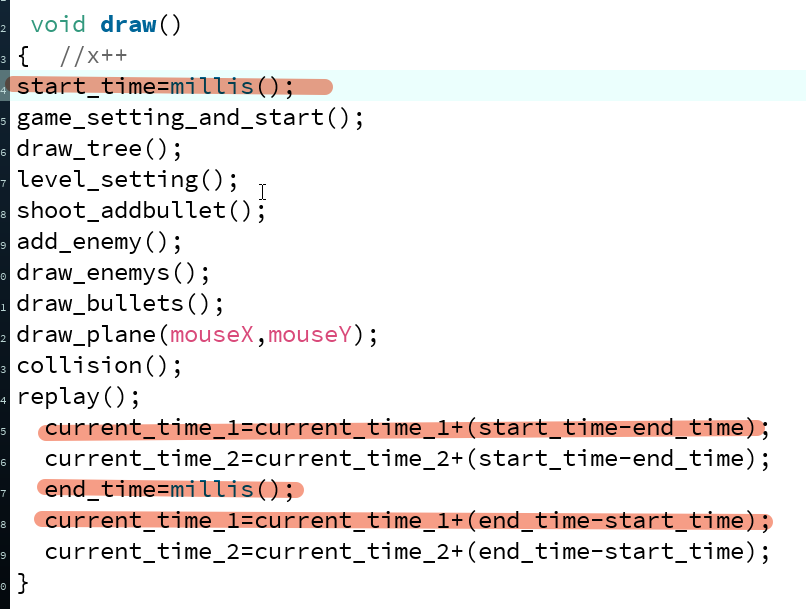
使敌机和子弹按给定的频率生成

通过计算叠加每帧时间,当合计时间大于生成频率规定时间，

合计时间减去生成时间,并执行相应代码,

如增加一架敌机或增加一颗子弹。

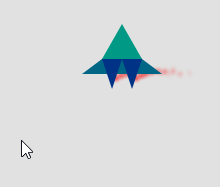
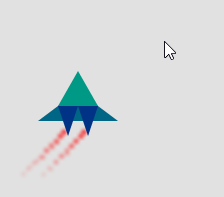
### 3.1.2核心代码:



## 3.2.向量

### 3.2.1实现效果/功能简介

通过向量[5]这一概念,实现对plane的控制,效果如下图所示。

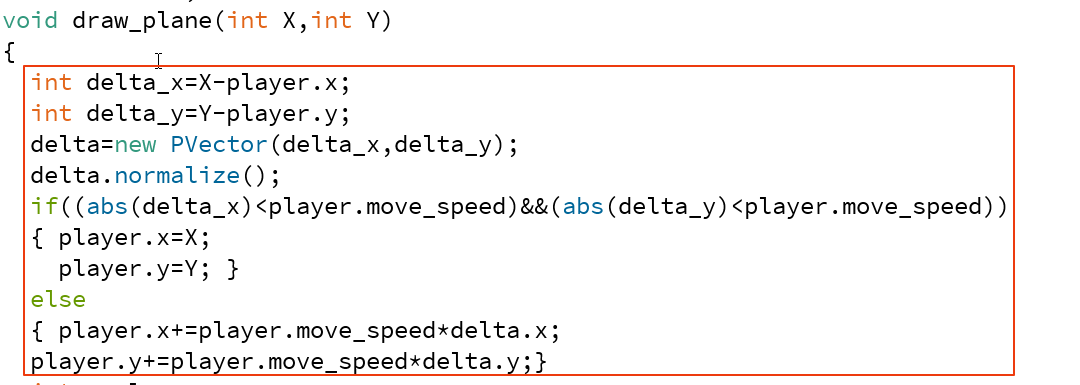


通过在player类里增加一个move\_speed变量,让飞机以一定速度朝鼠标方向移动，

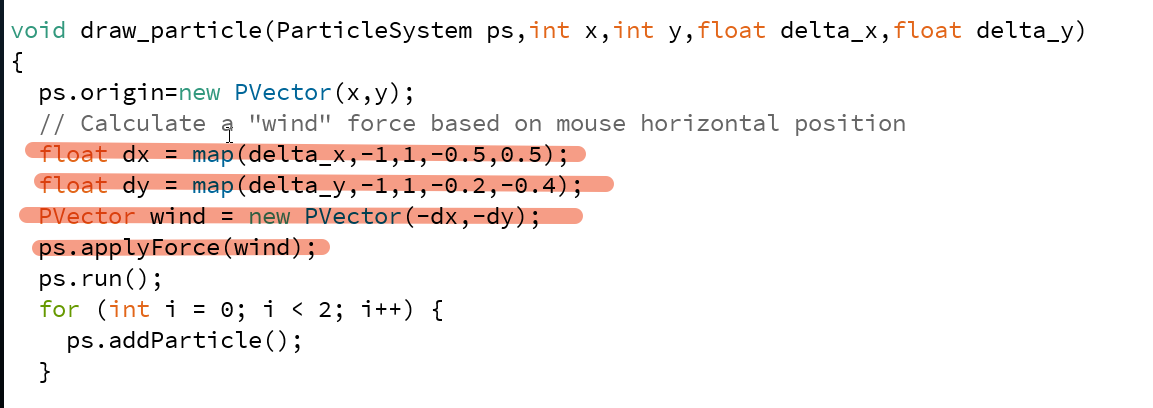
从而达到控制效果,并增加限制代码:当鼠标与飞机实际位置相差很小时,飞机实际位置就等于鼠标位置,可以防止出现飞机在鼠标反复摇摆的情况。

### 3.2.2核心代码:

飞机控制



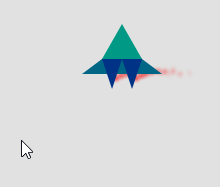
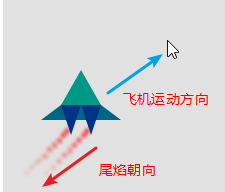
施加风力



## 3.3.粒子系统

### 3.3.1实现效果/功能简介

在飞机后模拟尾焰，,效果如下图所示。

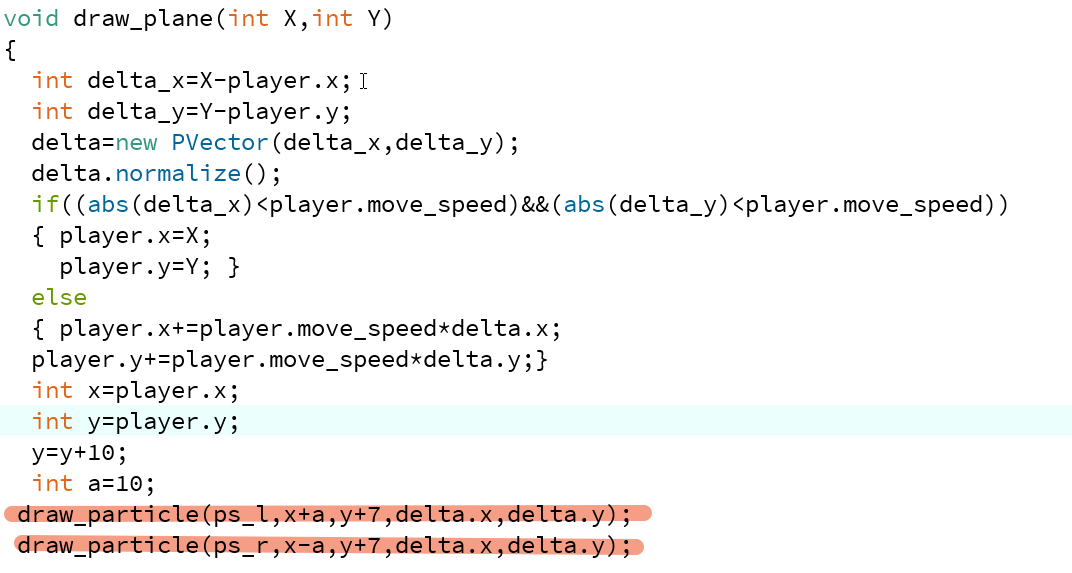
 



通过在photoshop中画出一个合适的红色半透明的雾状图,将该图片作为粒子在相应位置不断生成和消失构成一个粒子系统,在飞机的尾部合适位置添加两个这样的粒子系统,即实现的尾焰。

为了使尾焰更加逼真,向尾焰添加与飞行方向相反的模拟风力,并对参数进行控制,防止飞机后退时尾焰从上方露出，从而让玩家的游戏体验更加真实。

### 3.3.2核心代码:



### 3.3.3Particle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;

}

}

}

### 3.3.4ParticleSystem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));

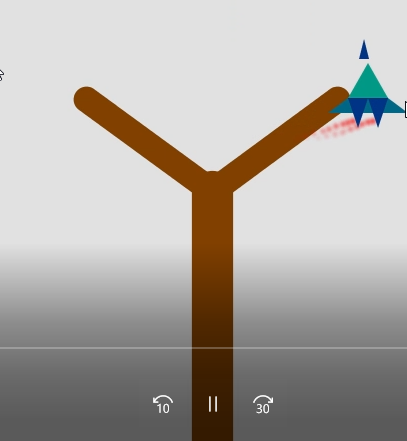
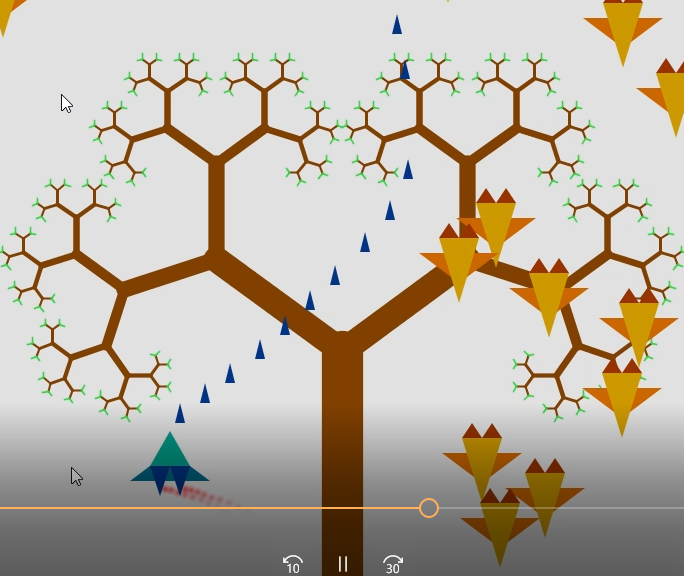
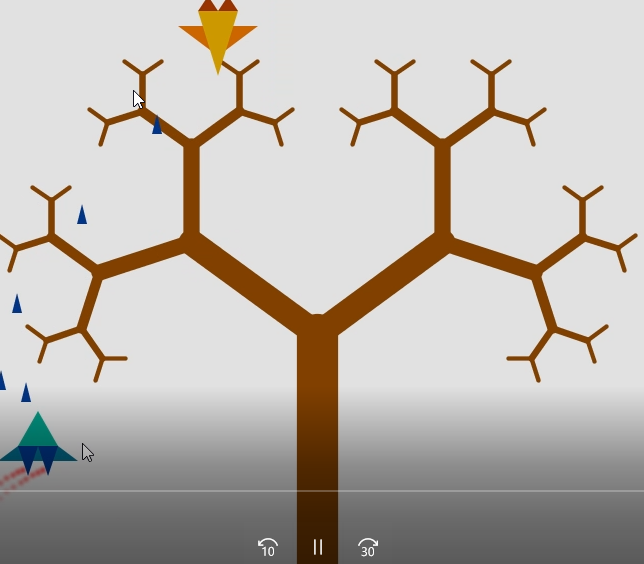
}

}

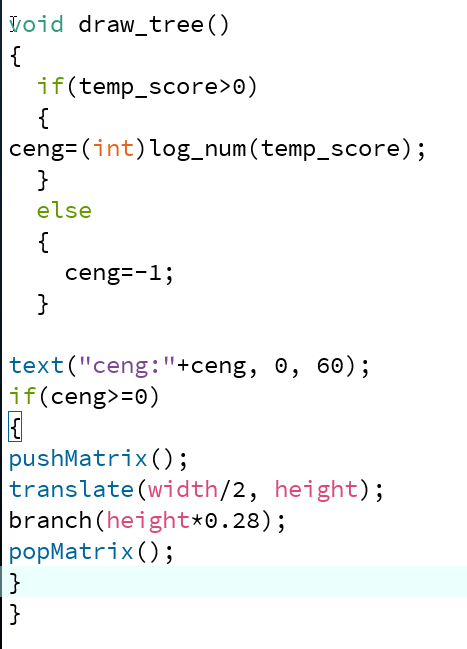
## 3.4.分形

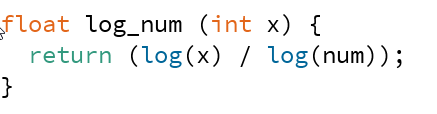
### 3.4.1实现效果/功能简介

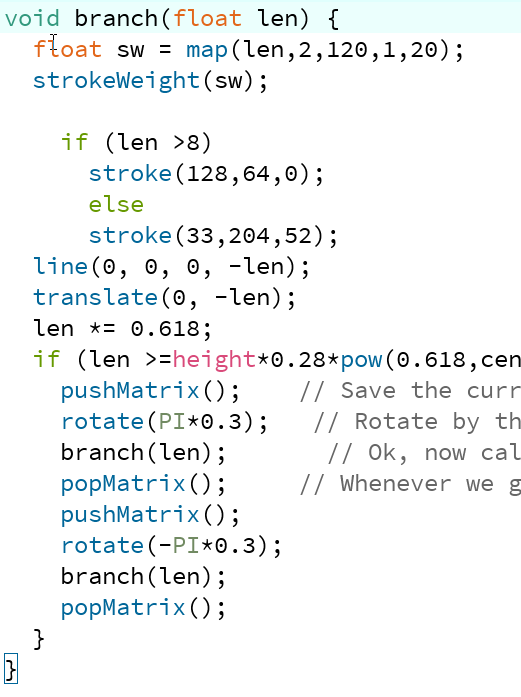
根据分数利用迭代的方法构造分形树，分数即为获取的材料数目,一个材料可以建造一段树枝。当树枝长度小于8是，颜色由棕色变为绿色，即为构造叶片。

### 3.4.2核心代码:







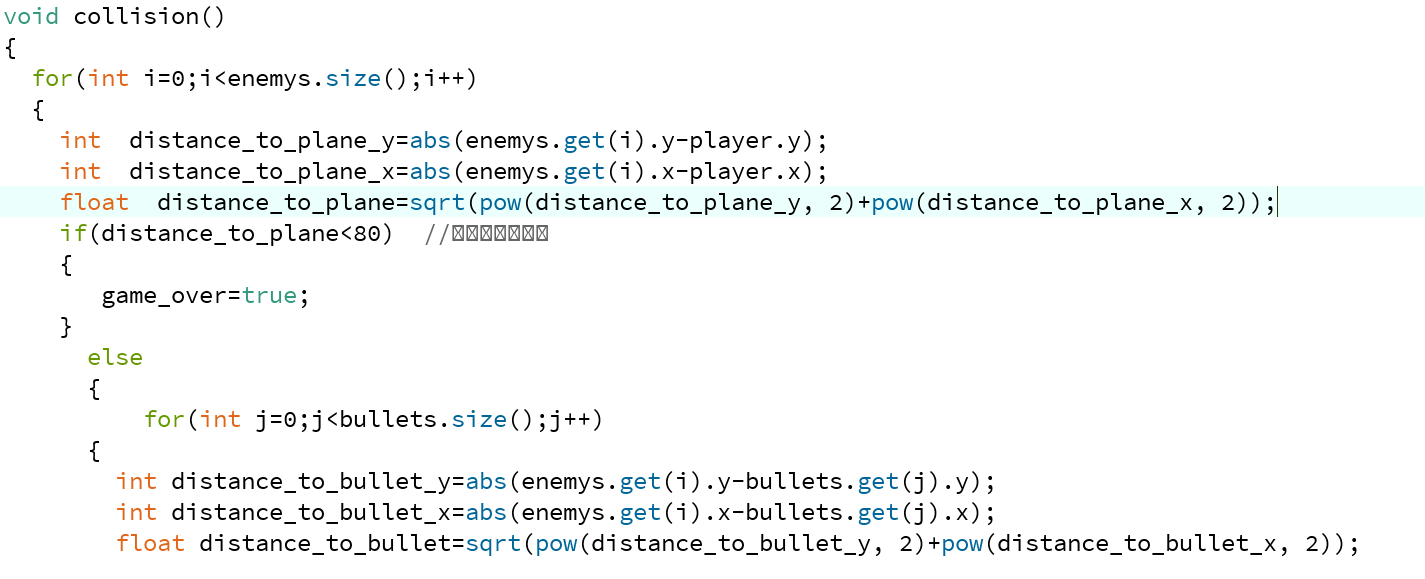
## 3.5.碰撞检测

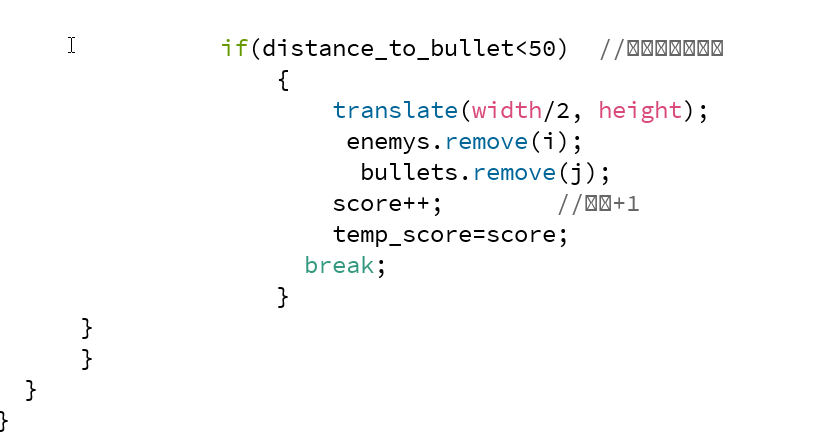
### 3.5.1实现效果/功能简介

判断子弹和敌机接触/得分,飞机和敌机接触/game over

具体为对于每一架敌机，计算其检测点与子弹检测点和我方飞机检测点的距离,距离小于某一参数视为碰撞发生，则可用代码控制相应事件发生如游戏结束，分数增加，子弹与敌机抵消。

### 3.5.2核心代码:





# 第4章 总结与展望

## 4.1 本文内容的总结

### 4.1.1 课题创新点

本游戏将游戏性与艺术性结合,让玩家在游玩的过程中了解到分形图案的规律性以及同自然界的紧密关系。

### 4.1.2 工作总结

大致工作步骤如下：

1. 确定主体:飞机、敌机、子弹，
2. 写每个类,以及其绘制函数，敌机和子弹数目多，

本来想用数组来表示，后来觉得还是用LinkedList增减方便；

1. 写计时器来按一定频率生成子弹,和敌机
2. 写碰撞检测机制来使子弹与敌机在一定距离内相消,

然后依据该机制来使飞机与敌机在一定距离内触发game over;

1. 写游戏逻辑，如game start.game over,level来控制游戏由简单逐渐变困难
2. 丰富游戏内容。在飞机尾翼加入两个以图片为粒子的系统，实现喷气效果
3. 改进飞机控制方式,由鼠标随移随到，改为朝鼠标运动,使游戏更合理。
4. 实现分形树，通过分数来控制分形树的生成，给玩家以反馈。
5. 总结:总代码量为350多行,游戏的可玩性和效果都还可以。

## 4.2 对今后工作的展望

本次的游戏游戏设计，还有许多功能、细节还有缺憾和不完善。如果有时间和经历还有很大的完善空间,在游戏性上可以添加新的游戏场景,对应新的分形图案,更多运动模式的敌机和炮弹,以及对玩家战机的升级系统设计等等,艺术性上也有很多的可改进之处,如更合适的游戏配色方案,更华丽的战机绘制方案,碰撞时可以生成一些带有艺术感的爆炸效果等。并且在碰撞检测上可以增加筛选机制，来减小计算量，使游戏不容易卡顿。

# 参考文献

[1] Adam Lake Game Programming Gems[M]COURSE TECHNOLOGY,2011

[2] Joshua Noble著．毛顺兵，张婷婷译．交互式程序设计[M]．北京：机械工业出版社，2014．

[3] 余国斌 基于粒子系统的火焰模拟[Z]中南林业科技大学，2010

[4] Mandelbrot The Fractal Geometry of Nature[M] W. H. Freeman and Company，1983

[5] Daniel Shiffman著．周晗彬译．代码本色[M]．北京：人民邮电出版社，2014．

# 致 谢

感谢《代码本色》作者Daniel Shiffman在github提供的各个艺术效果实现方案,给了我很多的启发，他在YouTube上发布的各个视频教程也让我获益匪浅。