准备，1.做雷霆战机之前，我们要现将它用到的父类放到一个包中，这样有利于我们进行进一步的研究，

我自己建立的有

2.有时候 构造器 建立吧，挺有用的

1.背景是死的，所以我们直接定死了就可以了

Image bg = JiazaiTupian.*getmage*("images/背景.jpg");

**public** **void** paint(Graphics g){

g.drawImage(bg,0,0,**null**);

}

2.但是飞机是会移动的，所以我们可以创建一个飞机类，让他自己画出

，就像我们在 建立太阳系的时候建立的那个 太阳是一样的。

**public** **class** Plane2 {

Image feiji;

**double** x,y;

**public** **void** draw(Graphics g){

g.drawImage(feiji,(**int**)x,(**int**)y,**null**);

}

**public** Plane2(String path,**double** x,**double** y){

feiji = JiazaiTupian.*getmage*(path);

**this**.x = x;

**this**.y = y;

}

}

3.调用飞机就可以了

不多说

**4.现在开始学习键盘的控制**

1》》需要一个新的内部类

内部类的名字随便，但是要继承一个类KeyAdapter

**class** Shubiao **extends** KeyAdapter{

}

2》》/定义一个方便使用 外部类的普通的属性

/

**class** Shubiao **extends** KeyAdapter{

**public** **void** keyPressed(KeyEvent e){

System.*out*.println("按下键"+e.getKeyCode()+e.getKeyChar());

} 输出数字 输出键盘属性字母

**public** **void** keyReleaseed(KeyEvent e){

System.*out*.println("弹起来"+e.getKeyCode());

}

}

3》》上面这个程序，我们想起来会像 关闭窗口那个程序是有点相似

现在要知道的也是，这个程序得写到 加载窗口那个方法中，所以在程序中重写那个方法

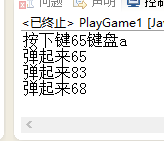
**public** **void** chuang(){

**super**.chuang();

addKeyListener(**new** KeyMonitor());

}

4》这样产生的结果是



6》现在我们开始使用键盘将图片动起来吧!

**class** KeyMonitor **extends** KeyAdapter{

**public** **void** keyPressed(KeyEvent e){

System.*out*.println("按下键"+e.getKeyCode()+"键盘"+e.getKeyChar());

**switch**(e.getKeyCode()){

**case** 37:

fei.x -=10;**break**;**case** 38:fei.y -=10;**break**;

**case** 39:

fei.x +=10;**break**;**case** 40:fei.y +=10;**break**;

}

}

7》现在我们思考这些都是飞机类的属性，看看能不能移动到飞机类中去

1>在飞机中建立一个类，

**public** **void** move(KeyEvent e){ 这个事传入的 参数，其实这个就是相当于是 画画的那个调用的方法一样的

**switch**(e.getKeyCode()){

**case** 37:

x -=10;**break**;

**case** 38:

y -=10;**break**;

**case** 39:

x +=10;**break**;

**case** 40:

y +=10;**break**;

}

}

2>在键盘监听的程序中直接调用就可以了

**public** **void** keyPressed(KeyEvent e){

System.*out*.println("按下键"+e.getKeyCode()+"键盘"+e.getKeyChar());

fei.move(e);

}

8》虽然现在飞机是可移动了但是方向比较单一不可以斜着移动（使用按下键和弹起键）

1》使用if语句讨论

**public** **void** draw(Graphics g){

g.drawImage(feiji,(**int**)x,(**int**)y,**null**);

**if**(left){

x -=speed;

}

**if**(right){

x +=speed;

}

**if**(up){

y -=speed;

}

**if**(down){

y +=speed;

}

}

2》在键盘监听当中是（最重要的是抬起键盘就是false）

//定义一个方便使用 外部类的普通的属性

**class** KeyMonitor **extends** KeyAdapter{

**public** **void** keyPressed(KeyEvent e){

System.*out*.println("按下键"+e.getKeyCode()+"键盘"+e.getKeyChar());

//调用飞机中的那个移动的方法，是图片动起来

**switch**(e.getKeyCode()){

**case** 37:

fei.left = **true**;**break**;

**case** 38:

fei.up = **true**;**break**;

**case** 39:

fei.right = **true**;**break**;

**case** 40:

fei.down = **true**;**break**;

}

}

**public** **void** keyReleased(KeyEvent e){

System.*out*.println("弹起来"+e.getKeyCode());

**switch**(e.getKeyCode()){

**case** 37:

fei.left = **false**;**break**;

**case** 38:

fei.up = **false**;**break**;

**case** 39:

fei.right = **false**;**break**;

**case** 40:

fei.down = **false**;**break**;

}

}

}

9.现在开始出子弹了（研究属性，点，坐标，生成好多不同的点）敌人的子弹是随机的所以，我们想起以前的东西 double a = Math.random();

是属于随机数字

**public** **void** draw(Graphics g){

Color m = g.getColor();

g.setColor(Color.*yellow*);

g.fillOval((**int**)x,(**int**)(y),(**int**)(a),(**int**)b);

g.setColor(m);

x = speed \* Math.*cos*(degree);

y = speed \* Math.*sin*(degree);

}

**public** **void** zidan(){

//设置一个子弹的随机产生

degree = Math.*random*() \* Math.*PI*\*2;

}

10.画出一些子弹，这时用到了容器

// 这个里面我们来存放子弹

ArrayList a = **new** ArrayList(); //泛型暂时为学，以后学了建议强烈加上

**public** **void** paint(Graphics g){

//检测是不是能用 g.drawLine(100,100,200,200);

g.drawImage(bg,0,0,**null**);

fei.draw(g);

**for**(**int** i = 0 ; i < 50 ;i++ ){

Zidan3 b = (Zidan3)a.get(i);

b.draw(g);

}

}

**public** **void** chuang(){

**super**.chuang();

addKeyListener(**new** KeyMonitor());

//这里画子弹

**for** (**int** i = 0;i < 50;i++ ){

Zidan3 b = **new** Zidan3(); //制造使之变成一些子弹类

a.add(b);

}

}

10.现在子弹是有了，但是我们得必须固定子弹的

1.位置，使他在中间

2.使他只能在我们的窗口中

**public** **void** draw(Graphics g){

Color m = g.getColor();

g.setColor(Color.*red*);

g.fillOval((**int**)x,(**int**)y,(**int**)a,(**int**)b);

g.setColor(m);

x = speed \* Math.*cos*(degree);

y = speed \* Math.*sin*(degree);

使他不能再窗口外移动

**if**(x > ChangYongFuLei.*Game\_Width* - 10||x < 0){

degree = -degree;

}

**if**(y > ChangYongFuLei.*Game\_Height* - 10||y < 0){

degree = -degree;

}

}

//初始化 使他在中间 放射子弹

**public** **void** zidan(){

//设置一个子弹的随机产生

degree = Math.*random*() \* Math.*PI*\*2;

x = ChangYongFuLei.*Game\_Width*/2;

y = ChangYongFuLei.*Game\_Height*/2;

}

11.子弹的碰撞检测

我们需要获得子弹 和飞机的 矩形

1.飞机类中

**public** Rectangle peng(){

//获得矩形的宽度

Rectangle r = **new** Rectangle((**int**)x,(**int**)y,feiji.getWidth(**null**),feiji.getHeight(**null**));

**return** r;

}

2.子弹类中

**public** Rectangle zhuang(){

Rectangle e = **new** Rectangle((**int**)x,(**int**)y,(**int**)a,(**int**)b);

**return** e;

}

3.对子弹碰撞之后进行真实性的检测（在画出子弹的时候进行的进一步的检测）

**for**(**int** i = 0 ; i < 50 ;i++ ){

Zidan3 b = (Zidan3)a.get(i);

b.draw(g);

//检测和矩形的碰撞

**boolean** zhuangshangle = b.zhuang().intersects(fei.peng());

**if**(zhuangshangle){

System.*out*.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

}

12.检测出来之后，我们可以很巧妙的时设置 死亡

**boolean** sihuo = **true**;

//将飞机画出来

**public** **void** draw(Graphics g){

**if**(sihuo == **true**){

g.drawImage(feiji,(**int**)x,(**int**)y,**null**);

**if**(left){

x -=speed;

}

**if**(right){

x +=speed;

}

**if**(up){

y -=speed;

}

**if**(down){

y +=speed;

}

}

}

2.设置死亡

**if**(zhuangshangle){

//System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

fei.sihuo = **false**;

}

13.双缓冲区（直接写在里面就行）

//窗口不会闪烁了

**private** Image offScreenImage = **null**;

**public** **void** update(Graphics g){

**if**(offScreenImage == **null**){

offScreenImage = **this**.createImage(*Game\_Width*,*Game\_Height*);

Graphics gOff = offScreenImage.getGraphics();

paint(gOff);

g.drawImage(offScreenImage,0,0,**null**);

}

}

14.现在我们开始时间的设置，比如 活了多久，死亡时刻

1.设置属性

文件开头设置属性

**public** **class** PlayGame1 **extends** ChangYongFuLei{

Image bg = JiazaiTupian.*getmage*("images/背景.jpg");

Plane2 fei = **new** Plane2("images/345.jpg",100,100);

Date start;

Date over;

2.文件开始执行，时间就已经开始了

**public** **void** chuang(){

**super**.chuang();

addKeyListener(**new** KeyMonitor());

//这里是容器

**for** (**int** i = 0;i < 50;i++ ){

Zidan3 b = **new** Zidan3();

a.add(b);

}

start = **new** Date();

}

结束时间

**if**(zhuangshangle){

//System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

fei.sihuo = **false**;

over = **new** Date();

}

时间的计算 long a = （start.getTime() – over.getTime()）/1000;

我们需要的是秒

14.设置爆炸效果（我们一般设置为static 为共享）

1.对于静态的初始化可能会不一样，以图片举例

2.爆炸其实是一些图片的

**public** **class** Baozha {

**double** a,b;

**static** Image *img*[];

**int** count;

//静态属性的初始化

**static** {

**for**(**int** i = 0;i < 16;i++){

//我们可以给图片设置一个好的名字，让他能够便于加载进去 例如 （i）.jpg "images/"+(i)+".jpg"

*img*[i] = JiazaiTupian.*getmage*("tupianlujin");

}

}

**public** **void** hua(Graphics g){

g.drawImage(*img*[count],(**int**)a,(**int**)b,**null**);

count++;

//让图片一张张的出现

}

**public** Baozha(**double** a,**double** b){

**this**.a = a;

**this**.b = b;

}

}