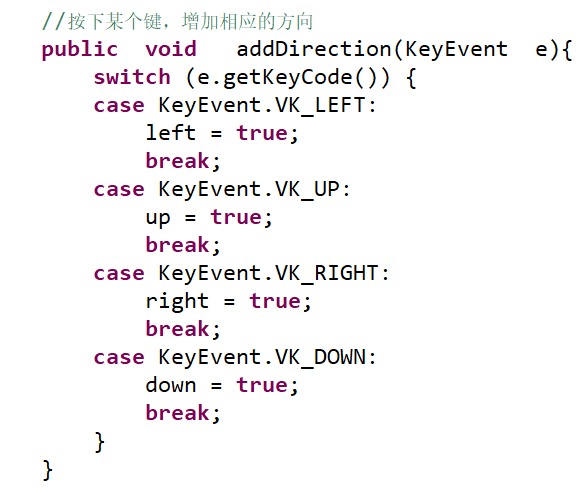
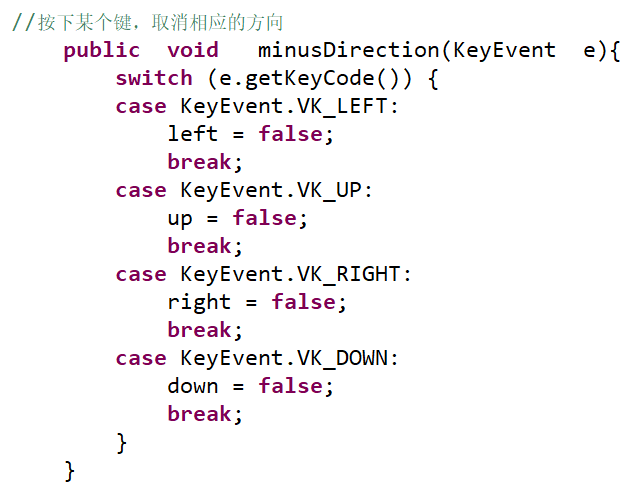
### 主要功能及算法设计

1. 核心功能：用键盘控制小鸡躲避炸弹。

算法设计：创造了方法addDirection和minusDirection实现通过键盘控制小鸡移动（包含上下左右四种方向）。



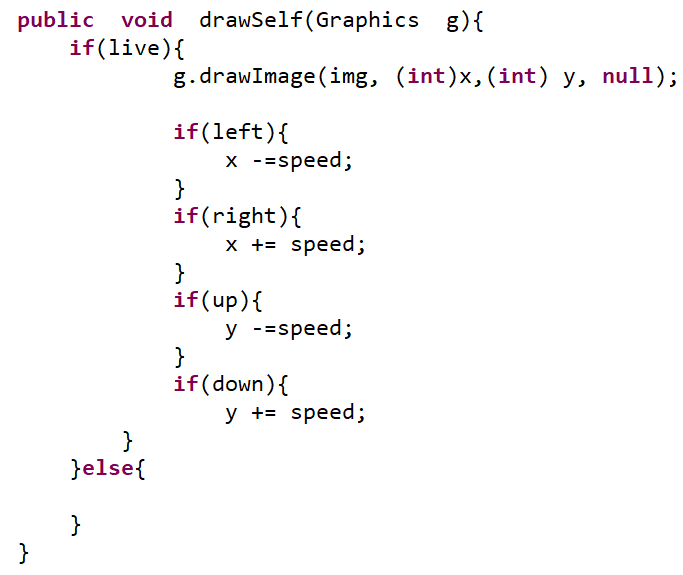


### 所有功能：

* 游戏角色（小鸡）的加载

算法设计：

小鸡类是游戏的主体角色类，通过drawSelf(Graphics g)进行绘制。同时通过判断用户的键盘操作来挪动自己的位置，基于用户的上下左右操作，位置挪动如下：

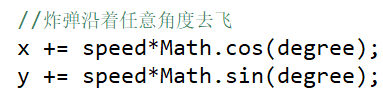


在MyGameFrame中将创建小鸡类的对象，实现游戏面板的小鸡初始化。在初始化过程中，将调用小鸡类中的各类属性参数和方法，实现和完成不同类之间的调用。

* 炸弹在面板上随机乱序飞行

算法设计：

在窗口固定位置（200,200）处生成炸弹，炸弹飞行的方向通过Math.sin(degree)定义，角度是随机的，速度定义为3，并且遇见边界会反弹。



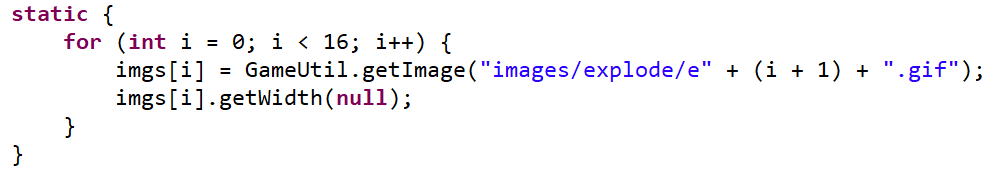
在定义炸弹的随机飞行时，我们调用了系统自带的Math.random函数，该函数可以随机生成数字，保证了炸弹飞行的随机性。



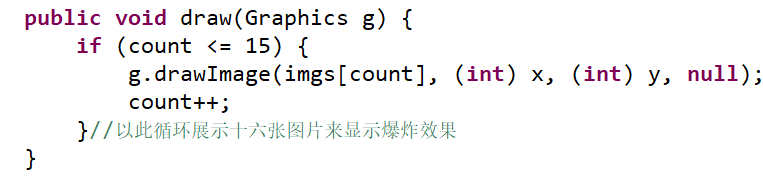
* 小鸡触碰到炸弹的爆炸效果显示

算法设计：

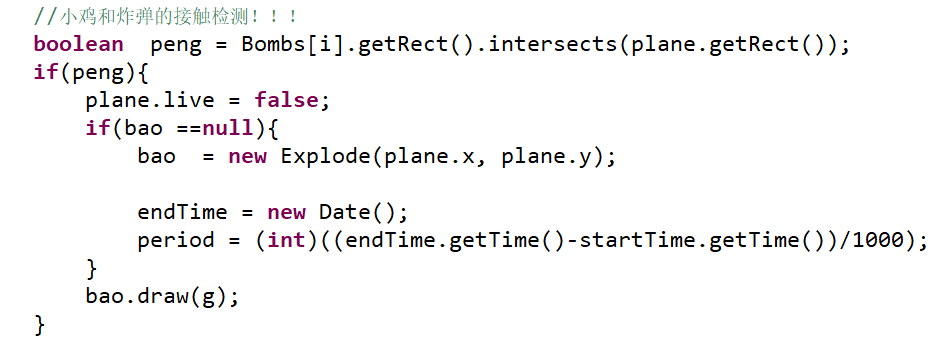
在小鸡触碰到炸弹后,导入爆炸类连播16张图片，并定义方法draw实现连播的动画效果。在设计过程中，我们定义Image[]来保存图片信息，并且使用static代码块，也就是在类加载时就加载这些图片，并且从属于类，通过此方法，我们不需要每次创建爆炸对象都加载图片，这样保证了游戏运行的效率。



同时，我们通过使用计数器count来控制到底画哪个图片，由于图片的命名规范，是按照顺序从1-16，这样程序只需依次读取这些图片对象即可。



判断小鸡是否与炸弹相碰，采用矩形检测原理，将物体抽象为矩形，判断；两矩形是否相交即可判断小鸡是否与炸弹相碰。



* 游戏结束时展现的时间显示

当游戏中的小鸡被麻雀捉住时游戏结束，在游戏界面中显示经过多少秒小鸡被麻雀捉到。此时页面中小鸡消失。

