```
package cn.sxt.game;
import java.awt.Color;
import java.awt.Font;
import java.awt.Frame;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.util.Date;
import javax.swing.JFrame;
/**
 * 飞机游戏的窗口
 * @author 江
 *
 */
public class MyGameFrame extends Frame {
	int period;//游戏进行的时间;

	Image bg=GameUtil.getImage("images/bg.jpg");
	Image planeImg=GameUtil.getImage("images/plane.png");
	Explode bao;

	Shell shell=new Shell();
	Shell[] shells=new Shell[50];

	Plane plane=new Plane(planeImg,250,250);

	Date startTime=new Date();
	Date endTime;
	@Override
		// TODO Auto-generated method stub
```

```
                //super.paint(g);

        public void paint(Graphics g) {      //自动被调用.g相当于一支画笔
                g.drawImage(bg,0,0,null);
                plane.drawSelf(g);//画飞机


                for(int i=0;i<shells.length;i++) {
                        shells[i].draw(g);//画炮弹
                        boolean
peng=shells[i].getRect().intersects(plane.getRect());
                        if(peng) {
                                plane.live=false;
                                if(bao==null) {
                                bao=new Explode(plane.x,plane.y) ;
                                endTime=new Date();
                                }
                                period=(int)(endTime.getTime()-
startTime.getTime())/1000;

                                bao.draw(g);
                        }
                        if(!plane.live) {
                        Color a=g.getColor();
                        g.setFont(new Font("宋体",Font.BOLD , 50));
                        Font b=g.getFont();
                        g.setColor(Color.red);
                        g.drawString("时间: "+period+"秒",(int)
plane.x,(int) plane.y);
                        g.setColor(a);
                        g.setFont(b);
                        }
                }
```

```
	}
	//帮助我们反复的重画窗口
	class PaintThread extends Thread{
		@Override
		public void run() {
			while(true) {
				//System.out.println("窗口被画！");
				repaint();

				try {
					Thread.sleep(40);
				} catch (InterruptedException e) {
					// TODO Auto-generated
catch block
					e.printStackTrace();
				}
			}
		}
	}
	//定义键盘监听的内部类
	class KeyMonitor extends KeyAdapter{
		@Override
		public void keyPressed(KeyEvent e) {
			plane.addDirection(e);
		}
		@Override
		public void keyReleased(KeyEvent e) {
			plane.minusDirection(e);
		}

	}

	/**
```

```
     * 初始化窗口
     */
    public void launchFrame() {
            this.setTitle("江大师");
            this.setVisible(true);
            this.setSize(Constant.GAME_WIDTH,Constant.GAME_HEIGHT);
            this.setLocation(300,300);

            this.addWindowListener(new WindowAdapter() {

        @Override
        public void windowClosing(WindowEvent e) {
            // TODO Auto-generated method stub
            super.windowClosed(e);
            System.exit(0);
        }
             });

             new PaintThread().start();//启动重画窗口的线程
             this.addKeyListener(new KeyMonitor());//给窗口增加键盘
的监听
             for(int i=0;i<shells.length;i++) {
                      shells[i]=new Shell();
             }


    }
    public static void main(String args[]) {
    MyGameFrame f=new MyGameFrame();
    f.launchFrame();
    }
    private Image offScreenImage = null;
```

```
	public void update(Graphics g) {
		if(offScreenImage == null)
			offScreenImage = this.createImage(500,500);//这是游戏窗口的宽度和高度

		Graphics gOff = offScreenImage.getGraphics();
		paint(gOff);
		g.drawImage(offScreenImage, 0, 0, null);
	}

}
```