```
import gameutil.*;
import javax.swing.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;

public class Play extends ListeningGameComponent
{
    Comet comet = null;
    Vector<Planet> planets = new Vector();
    Vector<Goal> goals = new Vector();
    boolean drag = false;

    //Vector<Comet> comets = new Vector();

    public Play()
    {
        super(640,480);
    }

    public void update()
    {

        if(planets!= null)
        {
            if(!isMousePressed(1))
                drag = false;

            if((comet != null) && (!drag))
            {
                if(isKeyPressed("UP"))
                    comet.move(Direction.NORTH);
                if(isKeyPressed("DOWN"))
                    comet.move(Direction.SOUTH);
                if(isKeyPressed("RIGHT"))
                    comet.move(Direction.EAST);
                if(isKeyPressed("LEFT"))
                    comet.move(Direction.WEST);

                for(int i = 0; i < planets.size(); i++)
                    comet.move(planets.get(i));

                comet.update();

                for(int i = 0; i < goals.size(); i++)
                    goals.get(i).testReached(comet);

                for(int i = 0; i < planets.size(); i++)
                {
                    if(comet.getDistance(planets.get(i)) < comet.radius+
                    planets.get(i).radius)
                    {
                        comet = null;
                        for(i = 0; i < goals.size(); i++)
                            goals.get(i).reset();
                        i = planets.size();
                    }
                }
            }

            if(win() && goals.size()>0)
                comet = null;

            if(isMousePressed(1))
            {
                if(!drag)
                {
                    drag = true;
                    //mousePressed1 = false;
                    comet = new Comet(5, mouseX, mouseY);
                }
                else if(comet != null)
                {
                    comet.xV = (comet.x-mouseX)/10.0;
                    comet.yV = (comet.y-mouseY)/10.0;
                }
            }
            if(isMousePressed(3))
            {
                mousePressed3 = false;
                planets.add(new Planet(25, mouseX, mouseY));
            }
            if(isMousePressed(2))
            {
                mousePressed2 = false;
                goals.add(new Goal(15, mouseX, mouseY));
            }
            if(isKeyPressed("S"))
            {
                try
                {
                    resetKeys();
                    ObjectOutputStream oos = new ObjectOutputStream(new
                    FileOutputStream(new File("levels\\" + JOptionPane.
                    showInputDialog(this, "Save As: "))));
                    oos.writeObject(planets);
                    oos.writeObject(goals);
                    oos.close();
                }
                catch(IOException ex)
                {
                }
            }
            if(isKeyPressed("L"))
            {
                try
                {
                    resetKeys();
                    Scanner kb = new Scanner(System.in);
                    ObjectInputStream ois = new ObjectInputStream(new
                    FileInputStream(new File("levels\\" +JOptionPane.
                    showInputDialog(this, "Load: "))));
                    planets = (Vector<Planet>)ois.readObject();
                    goals = (Vector<Goal>)ois.readObject();
                    ois.close();
                    comet = null;
                    for(int i = 0; i < goals.size(); i++)
                        goals.get(i).reset();
                }
                catch(Exception ex)
                {
                }
            }
            if(isKeyPressed("ENTER"))
            {
                try
                {
                    resetKeys();
                    File levels = new File("levels\\");
                    File[] level = levels.listFiles();
                    ObjectInputStream ois = new ObjectInputStream(new
                    FileInputStream(level[(int)(Math.random()*level.length)]))
                    ;
                    planets = (Vector<Planet>)ois.readObject();
                    goals = (Vector<Goal>)ois.readObject();
                    ois.close();
                    comet = null;
                    for(int i = 0; i < goals.size(); i++)
                        goals.get(i).reset();
                }
                catch(Exception ex)
                {
                    ex.printStackTrace();
                }
            }
            if(isKeyPressed("N"))
            {
                resetKeys();
                planets = new Vector();
                goals = new Vector();
            }
            if(isKeyPressed("R"))
            {
                resetKeys();
                comet = new Comet(5, mouseX, mouseY);
                for(int i = 0; i < goals.size(); i++)
                    goals.get(i).reset();
            }
            if(isKeyPressed("H"))
            {
                resetKeys();
                JOptionPane.showMessageDialog(this, "H: Help\nEnter: load a
                random level\nL: Load a level by file name\nS: Save a level by
                file name\nR: Restart the level\nN: new level\nArrows: apply
                thrust to the commet\nLeft Click: select the initial position,
                and volocity of the comet\nRight Click: Place planet\nScroll
                Click: Place Goal");
            }
        }

    }

    public boolean win()
    {
        for(int i = 0; i < goals.size(); i++)
        {
            if(!goals.get(i).reached)
                return false;
        }
        return true;
    }

    public void draw(Graphics g)
    {
        if(planets != null)
        {
            if(comet != null)
                comet.draw(g);
            for(int i = 0;i < planets.size(); i++)
                planets.get(i).draw(g);
            for(int i = 0; i < goals.size(); i++)
                goals.get(i).draw(g);
            if(win() && goals.size()>0)
            {
                g.setColor(Color.GREEN);
                ((Graphics2D)g).drawString("YOU WIN!!!", 310, 240);
            }
        }
    }
    public static void main(String[] args)
    {
        JOptionPane.showMessageDialog((new Play()).makeTestWindow(), "Press
        ENTER to get started or H for more help!");
    }
}
```