Asteroids Overview

You are going to make a version of the 1979 classic arcade game Asteroids. If you haven't played it before, you can try it online at: http://games.cellbiol.com/asteroids.html

Some code has already been written for you. This code displays a black screen, on which you will make the spaceship and asteroids appear. It also currently updates a counter in the upper left-hand corner of the screen each time it updates the screen.

Look at the class `Game`. This class is an *abstract* class that is a subclass of the built-in Java class `Canvas` (http://docs.oracle.com/javase/7/docs/api/java/awt/Canvas.html). We will not change anything in the class `Game`.

Look at the class `Asteroids`. It is a subclass of the class `Game`. The `main` method for the game is in this class. First the `main` method creates a new instance of `Asteroids`, called `a`. Next the `main` method calls `a.repaint()`, which calls the method `a.paint(...)`, which you will fill in in the `Asteroids` class, over and over again. There are some details about how `a.repaint()` calls `a.paint()` which could be important for debugging, so they are explained in the next paragraph, but at a high level, you can think of what is happening in the code as:

```
Asteroids a = new Asteroids();
a.paint(...)
a.paint(...)
a.paint(...)
a.paint(...)
.
.
.
```

This paragraph will explain the details of what is actually happening when the main method calls `a.repaint()`, which is a method defined by Java in the class `Component`, which is the superclass of `Canvas` (which is the superclass of `Game`). The method `a.repaint()` calls the method `a.update()` over and over again. The method `update()` is defined in the superclass `Game` (overriding the original definition in `Component`), and it first calls the method `a.paint(...)`, which you will fill in in `Asteroids`. An instance of `Graphics` is passed into `a.paint(...)`, but this is not the instance that holds the game `Canvas`. Instead, this instance of `Graphics` acts as a buffer (and hence is called `buffer`). So when your code in `a.paint(...)` draws something, it is actually drawing it on this buffer `Graphics` instance. Everything in the buffer is then drawn to the screen all at once, making the animation less choppy. A schematic of what is really happening is below:

Asteroids Overview

```
Asteroids a = new Asteroids();
a.paint(...) -> draws to buffer
draw buffer on screen
a.paint(...) -> draws to buffer
draw buffer on screen
a.paint(...) -> draws to buffer
draw buffer on screen
.
.
.
```