Asteroids



# Step-by-step Instructions

1. [Download the asteroids scenario](https://drive.google.com/file/d/144zEjlyR0vfPRWNs97VtSQgAFXx-o1gQ/view?usp=share_link) to your *Greenfoot Projects* folder then unzip it and delete the zipped folder.
2. Rename the scenario folder so that it is named **asteroids\_yourname**.
3. Open the asteroids scenario located in your *Greenfoot Projects* folder.
4. Notice the various classes listed along the right side: World, Space, Actor, etc.
5. Double-click on the Space class to open the Java code for editing.

*This world does not use a background picture for the background. Instead it paints the background black and adds stars.*

1. Find this line of code:

**background.setColor(Color.BLACK);**

1. Change BLACK to RED. Click away from it to compile the class. If it doesn’t compile, see the comment below.

*Java is very case sensitive. RED and Red are not the same thing. Thus, you need to type RED with all capital letters or else you will get an error.*

1. Close the source code and Reset the project. The background should be red.
2. Open the source code for the Space class again.
3. Now change *Color.RED* to *new Color(145,23,244).*  The line of code that sets the color will now look like this:

**background.setColor(new Color(145,23,244));**

1. Close the source code and observe the color of the background. It should be purple.

*The three numbers used to make the new color represent an RGB (red, green, blue) color code. A new color can be made from mixing red, green, and blue and the first number is the amount of red, the second number is the amount of green, and the third number is the amount of blue. You can look up RGB color codes on charts like* [*this one*](https://www.rapidtables.com/web/color/RGB_Color.html)*.*

1. Open the source code for the Space class again.
2. Change the color again. This time pick your own numbers. The numbers must be between 0 and 255. If you like, you can use a [color code chart](https://www.rapidtables.com/web/color/RGB_Color.html) (scroll down, find the color you want, and then find the RGB color code for that color in the last column) or you can just choose any numbers between 0 and 255 and see what you get.
3. Compile the class, close the source code, and look at the color.
4. Change the background color back to black: Color.BLACK (If needed, see step #5 to see the line of code that makes the background color black.)
5. Run the project. Nothing happens. Click on the Reset button.
6. Right-click on the Rocket class and add a new Rocket to the world.
7. Right-click on the Asteroid class and add a new Asteroid to the world.
8. Hold the Shift key down and add a couple more asteroids to the world.
9. Run the program. You can fire the rocket with the Space bar and you can move the rocket with the arrow keys. If you turn up your volume and use headphones, you will be able to hear the sound effects.
10. Destroy an asteroid then stop the program.
11. Open the code for the Space class.
12. Find this line of code:

createStars(300);

1. Change the 300 to 1000 then compile the class and close the source code. Observe the difference. You may not notice a difference because it is a slight difference.
2. Now change the number of stars to 100000 and observe the difference. This time you will notice.
3. Change it back to 300.
4. Click on the Reset button to reset the project.
5. Add a rocket and two asteroids to the world, but don’t put the asteroids close to the rocket.
6. Right-click on the world and choose Save the world. (The world that you right-click on is the black background with stars on it.)

*Now the rocket and asteroids will always be at their current location when you click the Reset button. You don’t have to keep adding them.*

1. Run the program, fire a few shots then Reset.
2. Right-click on the rocket and choose *void setGunReloadTime(int reloadTime)*. When the box pops open, read the comment then set the reload time to 5 then click OK.
3. Run the program and fire a few shots. What is the difference?

*setGunReloadTime is a block of code - these are called methods in Java. The number (in this case 5) was an input into the method. In Java, these are called arguments or parameters.*

1. Open the code for the Asteroid class then find this line of code. It is near the top so it shouldn’t be too hard to find:

**private int size;**

1. Add the following code underneath the line of code shown in the previous step. You may copy and paste it:

**public static int numAsteroidsDestroyed = 0;**

*This code creates a variable named numAsteroidsDestroyed that is capable of storing integers. It is initialized to 0 meaning that its starting value is 0. This makes sense, because when you start playing the game, 0 asteroids have been destroyed.*

1. Find the following code in the breakUp() method. You will need to scroll down to find this:

if (size <= 16) {

// if we are already very small, just disappear

getWorld().removeObject(this);

return;

}

*If the size of the Asteroid is very small (less than or equal to 16 pixels) then the Asteroid is removed from the world. It has been destroyed.*

1. Add the following line of code to the code shown in the previous step. The code you need to add is shown in blue:

if (size <= 16) {

// if we are already very small, just disappear

getWorld().removeObject(this);

**numAsteroidsDestroyed++;**

return;

}

*We already made a variable named numAsteroidsDestroyed and initialized it to 0. The line of code that you just added increases this variable by one because an Asteroid just got destroyed and was removed from the World.*

*In my instructions, if you see code that is dark blue, that means that it is code that is intended to be copied and pasted somewhere.*

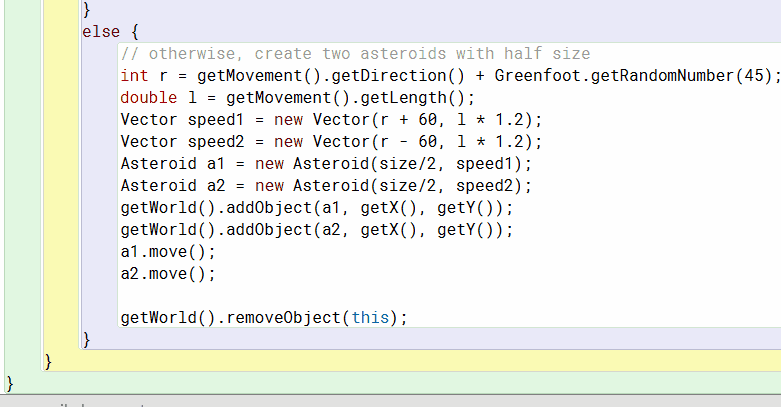
1. Scroll down to the very bottom of the Asteroid class and find the final closing curly brace - it looks like this **}**. Greenfoot is color coded and the curly brace that ends the scope of the class will be located on a light green background.
2. Paste the following code (copy and paste it) right **before** the last curly brace. Don’t paste it on the same line as the curly brace. Create a new blank line above the curly brace and paste it there (See the illustration below):

**public int getNumAsteroidsDestroyed()**

**{**

**return numAsteroidsDestroyed;**

**}**



1. Edit → Auto-layout. This will indent the code correctly. It doesn’t change what the code does but it makes it easier to read it.
2. Compile and close the code.
3. Reset the project and add three more asteroids to the world.
4. Save the world.
5. Play the game for a bit until you have destroyed at least one asteroid then pause the game. (The Run button changes to a Pause button while the project is running. Just click on the Pause button to pause the game.)
6. Right-click on any Asteroid and choose getNumAsteroidsDestroyed(). The answer should be in the yellow highlighted area.
7. Reset the project.

*Asteroids have an inherent stability. Each time they get hit by a bullet, their stability decreases. When it reaches zero, they break up. What is their initial stability value after you create them?*

1. Right-click on an asteroid and choose getStability() and it will tell you its initial stability.

*By how much does the stability decrease from a single hit by a bullet?*

1. Run the project and hit an asteroid with a single bullet then click on the Pause button.
2. Right-click on the Asteroid that you hit and choose getStability(). How much did it change?
3. Open the Asteroid class and look at the code for an Asteroid. Can you find where the stability is initialized? Can you find where it changes?
4. Make a very big asteroid and destroy it. You can make a very big asteroid by right-clicking on the Asteroid class and choosing a new *Asteroid(int size)*. When you click to place the asteroid in the world an input box will pop open and you can type the size - an asteroid of size 200 is very big.
5. Search for an asteroid image on the web. If you find a good one, download it to the images folder of this project. You might need to resize the image.
6. Right-click on the Asteroids class and choose Set Image and select this image. It should be on the list of images along the left side. Now add a new Asteroid to the world and it should use the image that you chose.
7. Reset the project.
8. Add more rockets to the world if you like.