

# Lab\_1

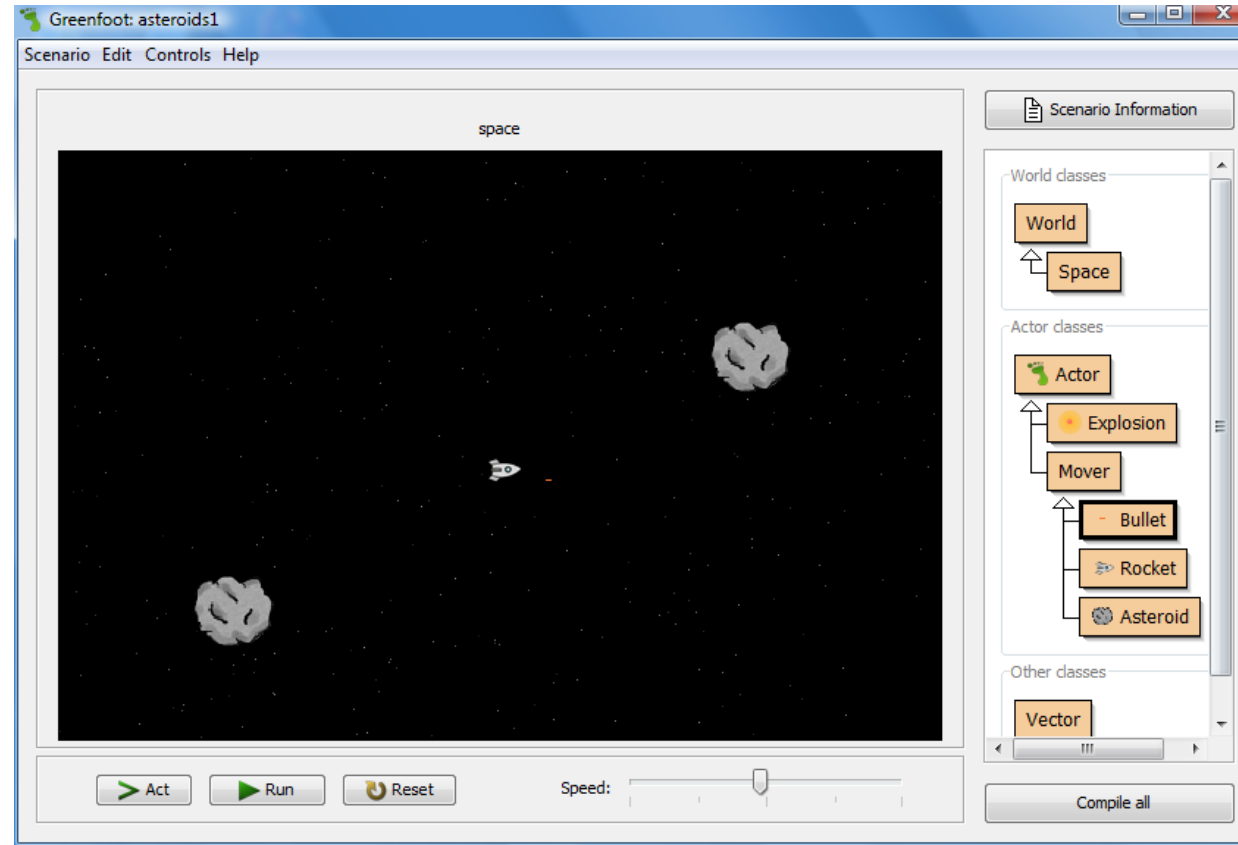
## Getting to know Greenfoot

---



420-141-VA - GAME PROGRAMMING 1 - VANIER COLLEGE

# Change the Scenario: Asteroids



# Step 0 - Setting up Asteroid Game

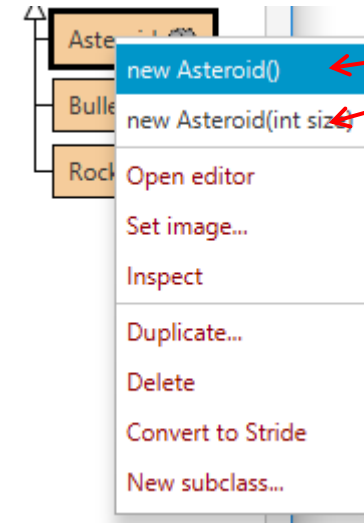
## Create 3 Asteroids Objects

- Try both Constructors
- Second Constructor specifies size of asteroid

## Create a Rocket Object (or many)

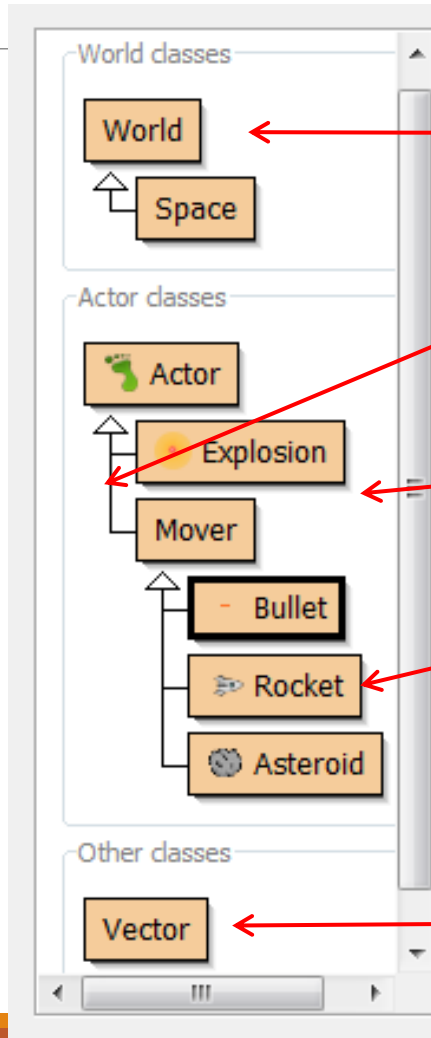
## Run the Scenario

- Control the spaceship with the arrows and space to shoot
- When you die, just create more Rockets!



**2 Constructors**

# Understanding the Class Diagram



**World Class** is always there in Greenfoot scenarios, it is built-in. Space represents a specific world for this scenario

Arrows show relationships

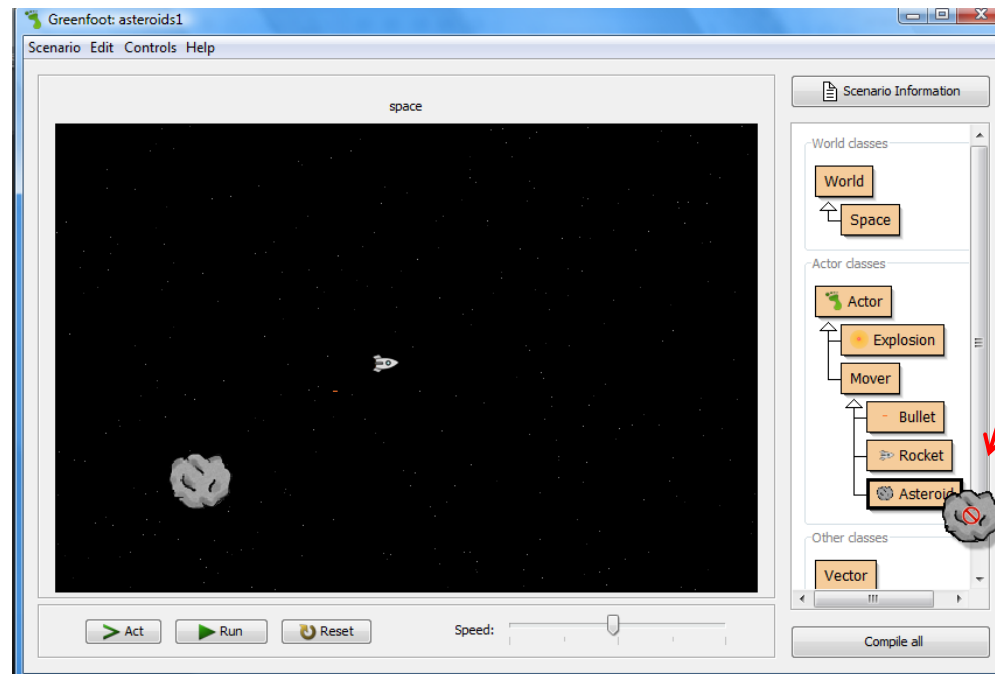
**Explosion and Mover** are subclasses of **Actor**

**Bullet, Rocket, and Asteroid** are subclasses of **Mover**.

**Vector** is a helper class

# Playing with Asteroids

**Start Playing by Creating Some Actor Objects (Objects of the Subclass of Actor). Create Objects for Rocket, Bullet, and Asteroid**



# Step 1

---

If you want to see what methods an object has access to, you \_\_\_\_\_ on that object.

- A. Left-click
- B. Right-click
- C. Rotate the scroll wheel
- D. Ask nicely

## Step 2

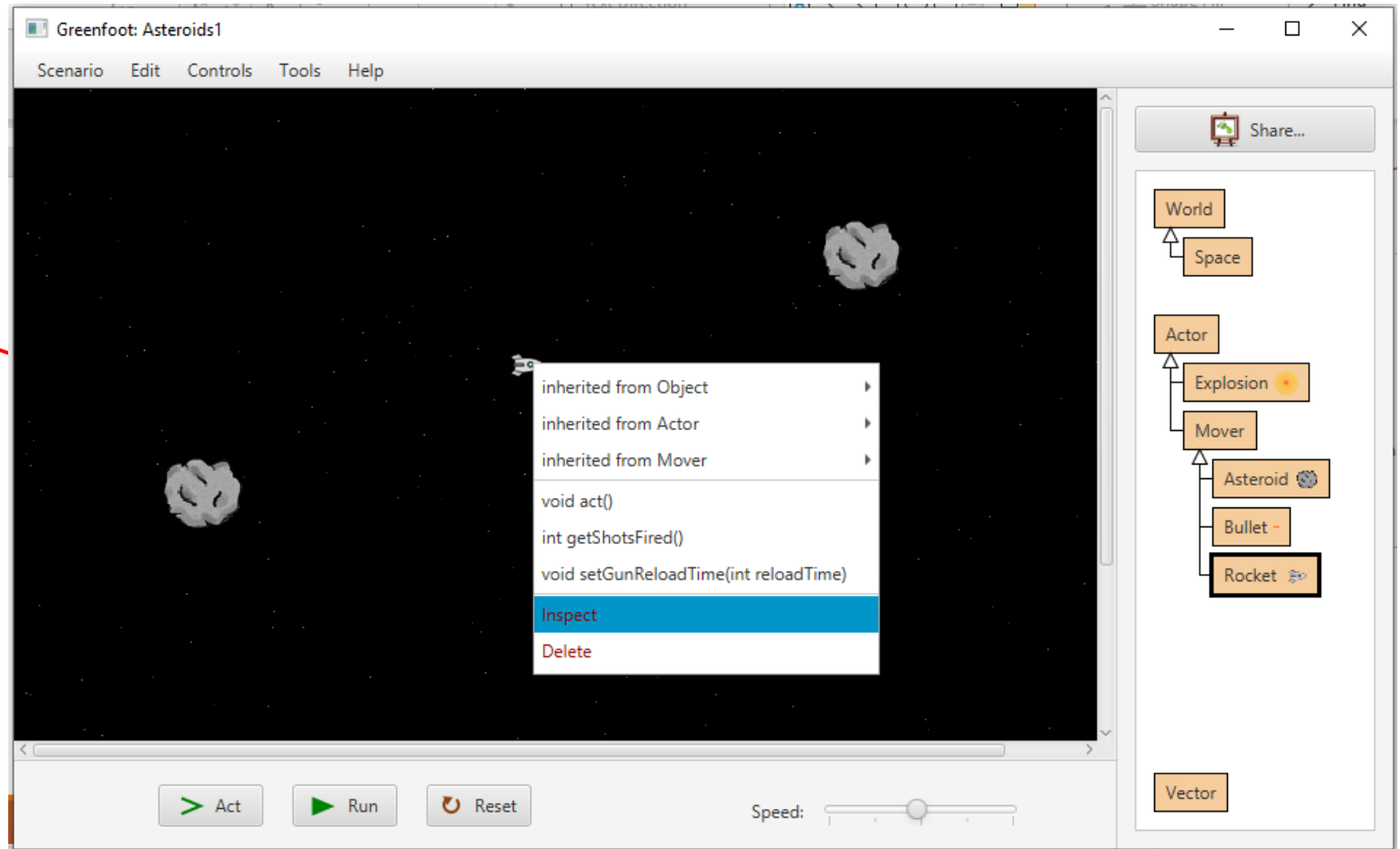
---

If you have played this game for a while, you will have noticed that you cannot fire very quickly.

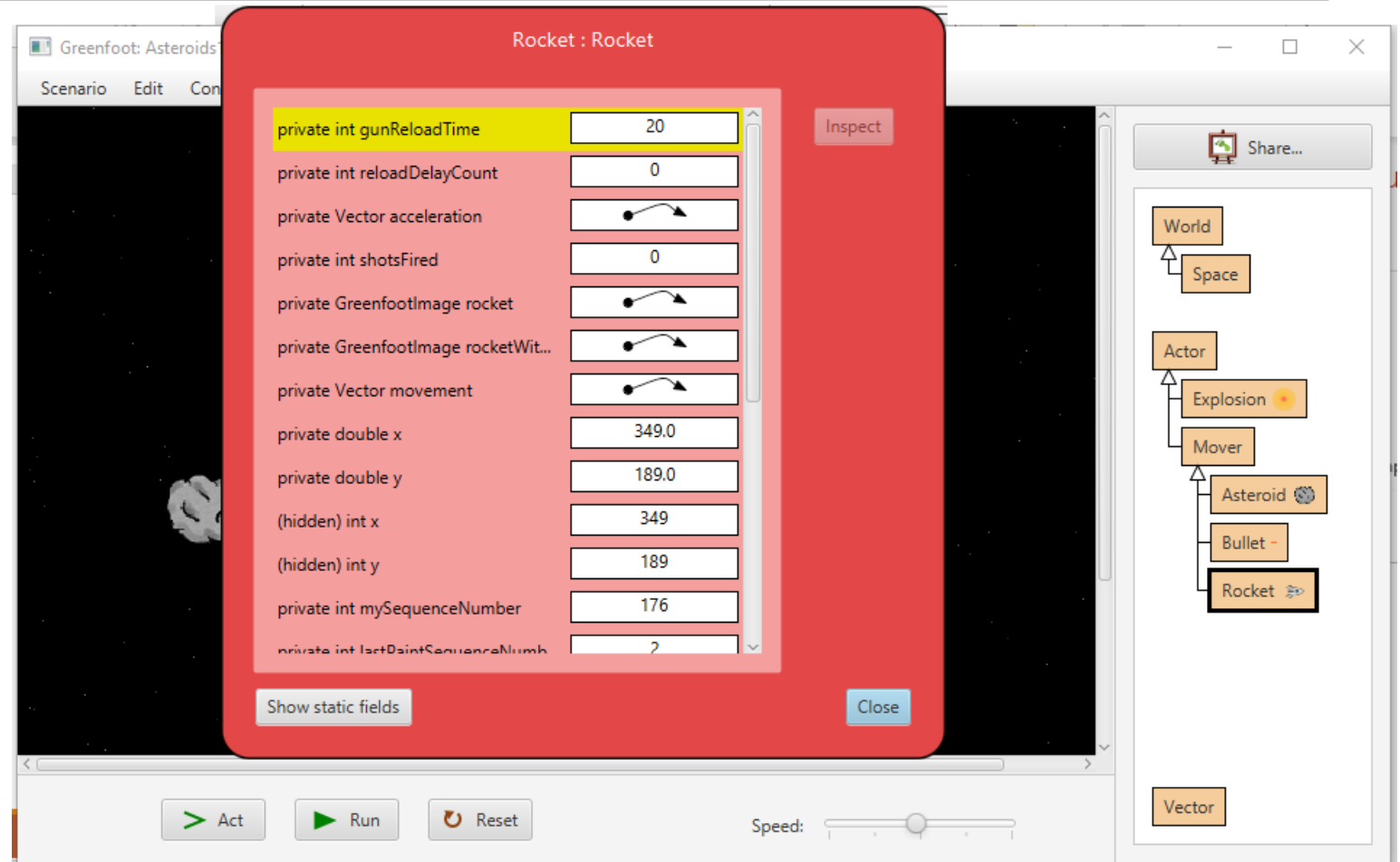
Let us tweak our spaceship firing software a bit so that we can shoot a bit quicker. (That should make getting the asteroids a bit easier!)

Place a rocket into the world, then invoke its **setGunReloadTime** method (through the object menu), and set the **reload time** to **5**. Play again (with at least two asteroids) to try it out.

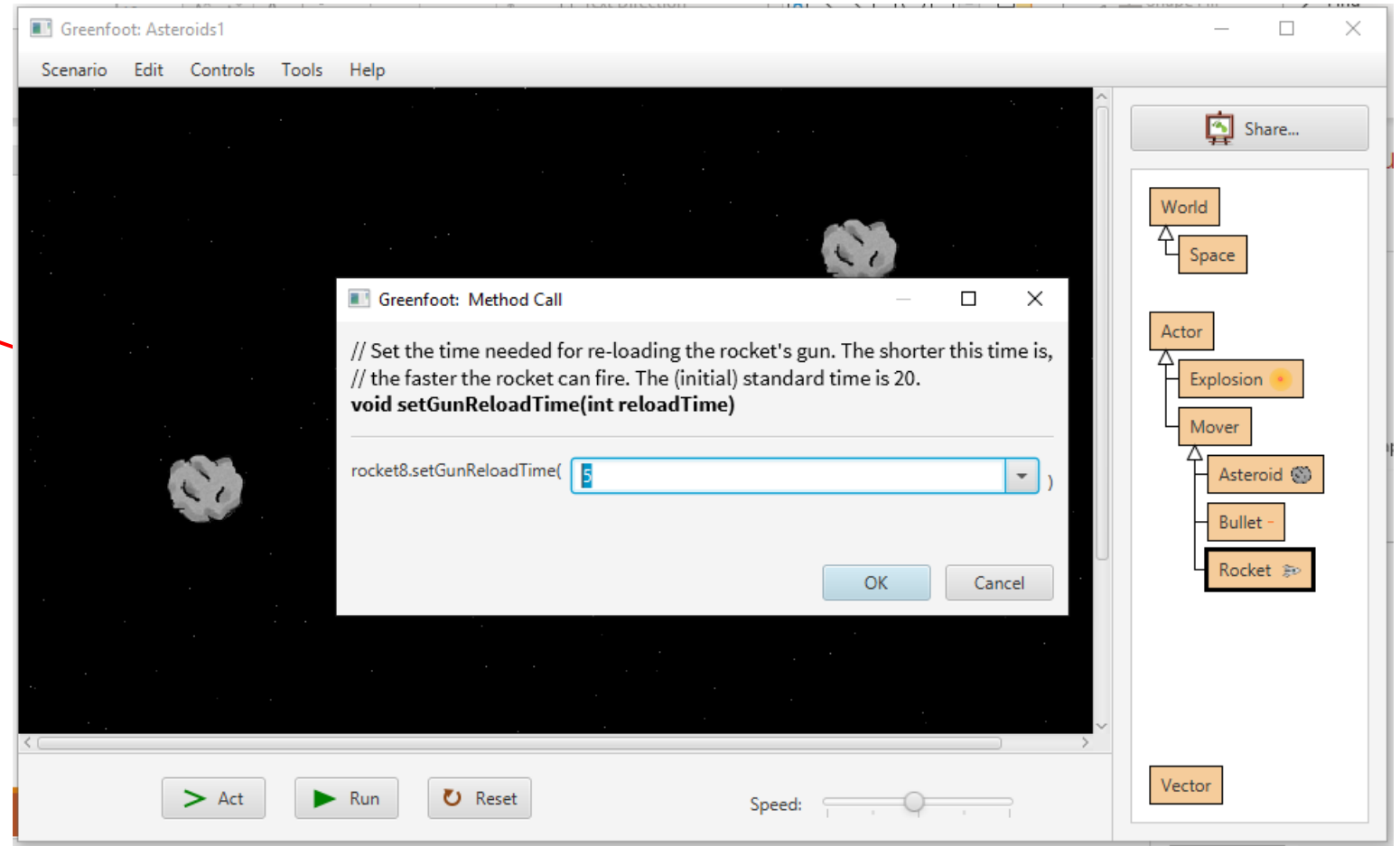
Right Click on the  
Object and Select  
**Inspect**







Click on  
**setGunReloadTime** and  
Type 5



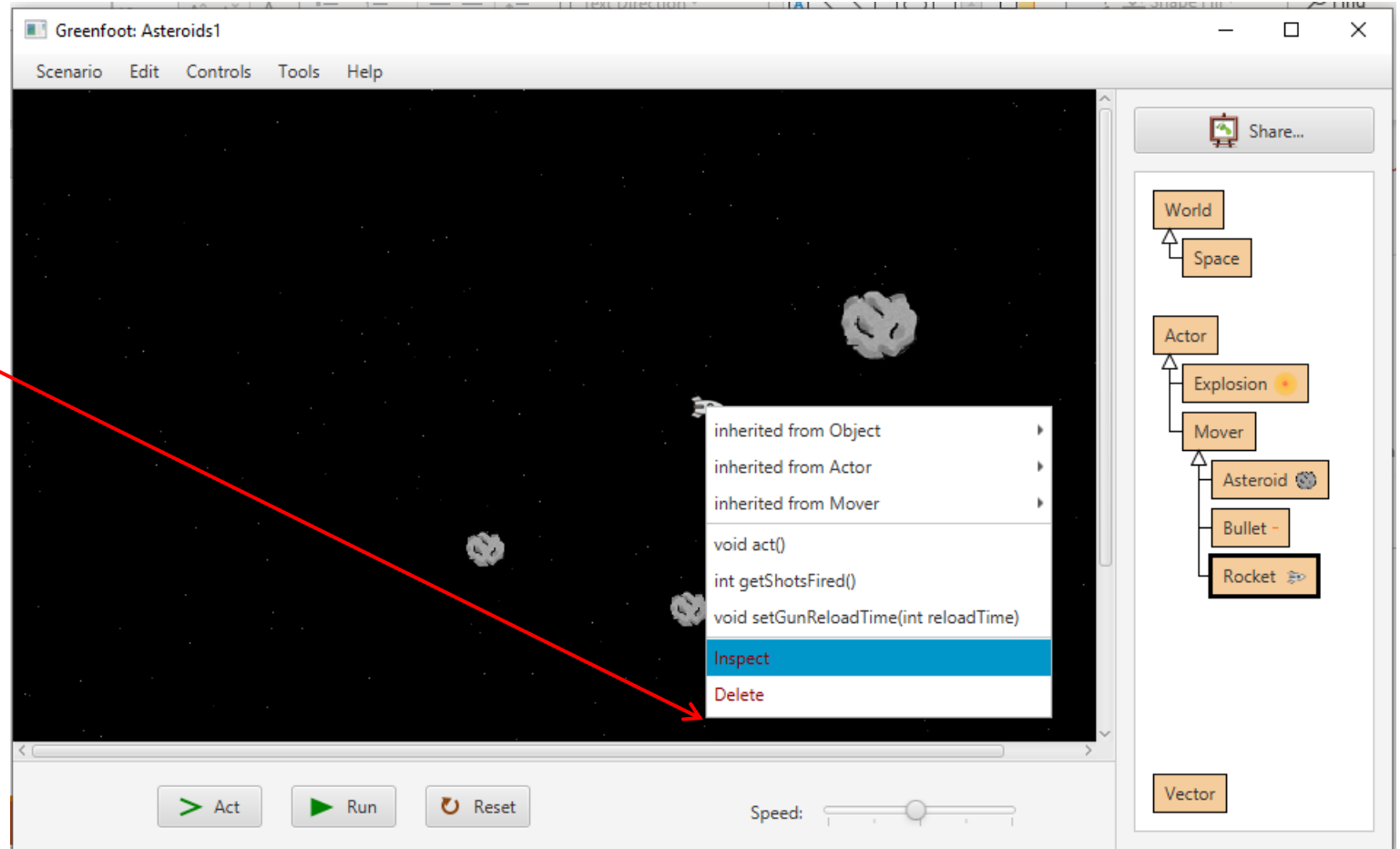
# Step 3

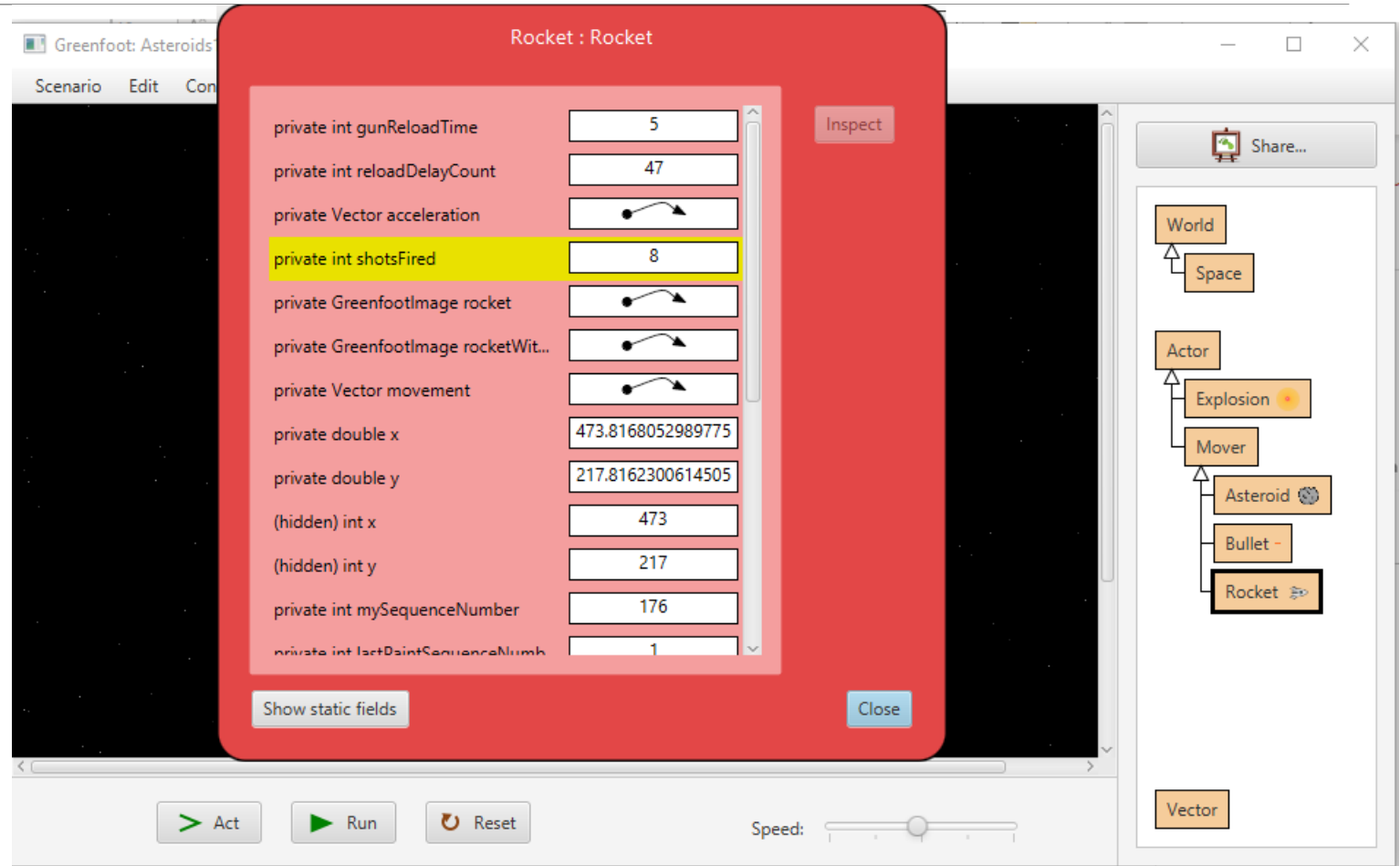
---

Once you have managed to remove all asteroids (or at any other point in the game), stop the execution (press Pause) and find **out how many shots you have fired**.

You can do this using a method from the rocket's object menu. (Try destroying two asteroids with as few shots as possible.)

**Right Click on the  
Object and Select  
Inspect**





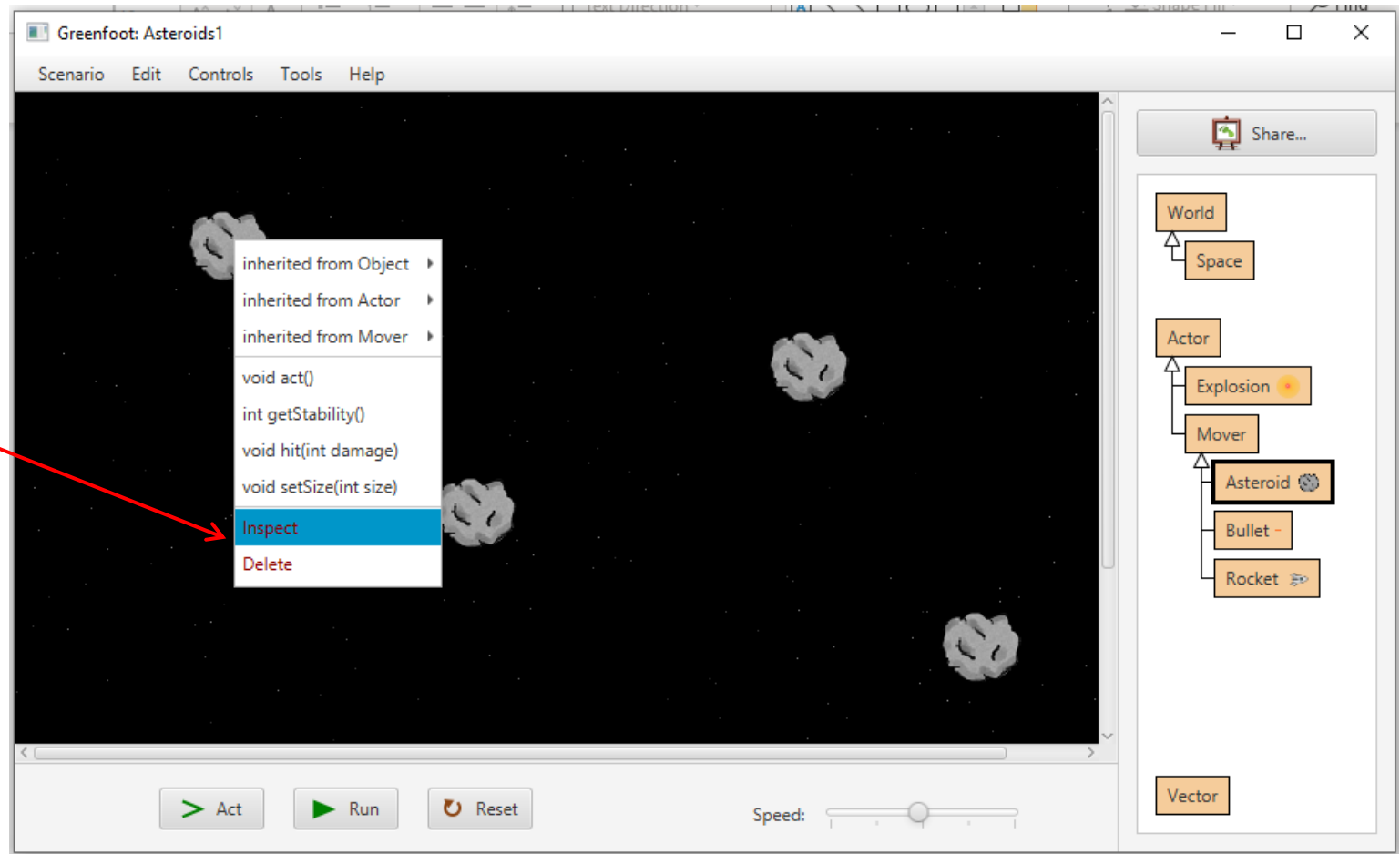
# Step 4

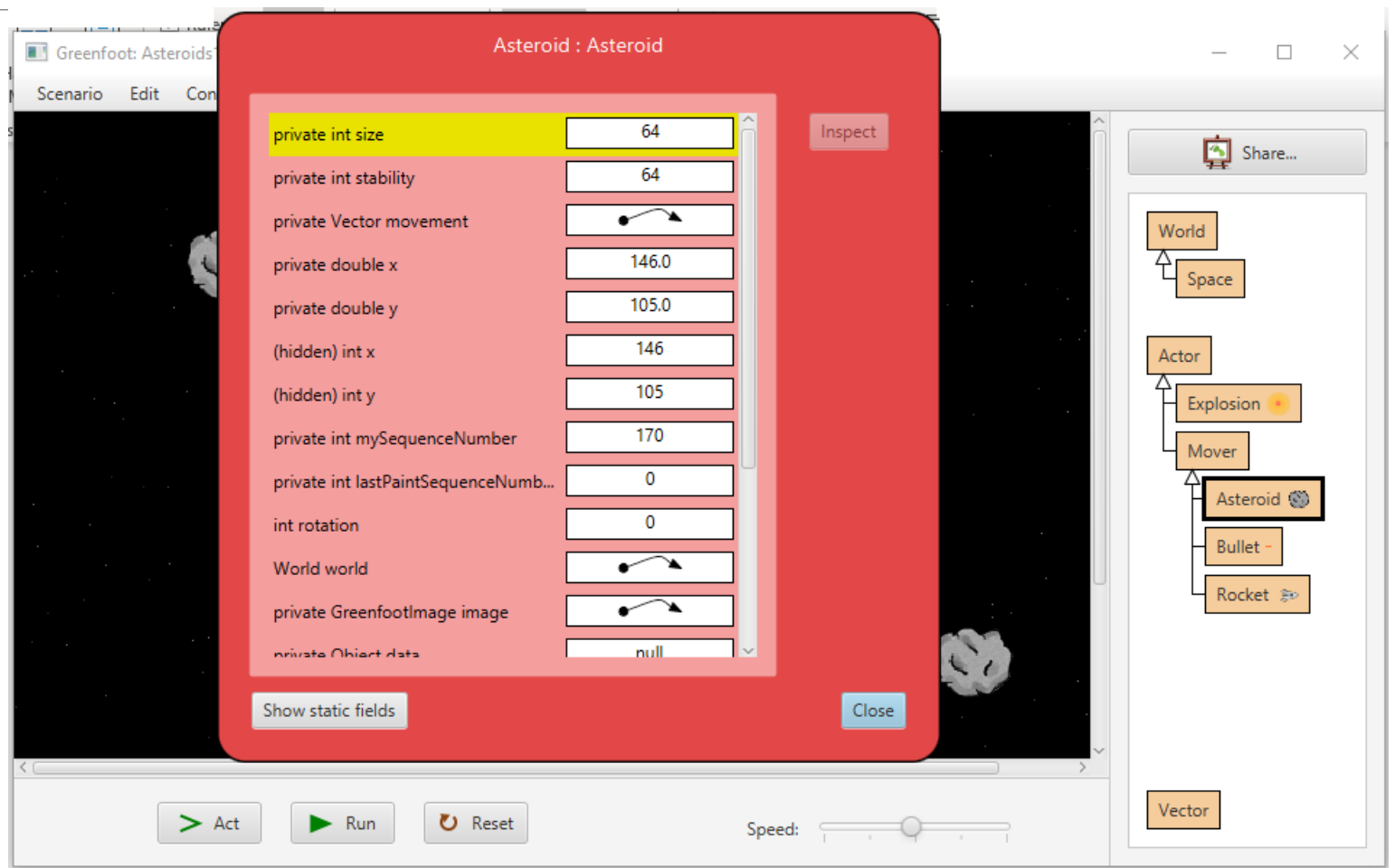
---

You may have noticed that the asteroids have the same **size** as soon as you place it into the world. What is its **initial size**?

Create 4 asteroids and change interactively (through method **setSize(int size)** ) their sizes respectively to **16, 32, 64** and **128**.

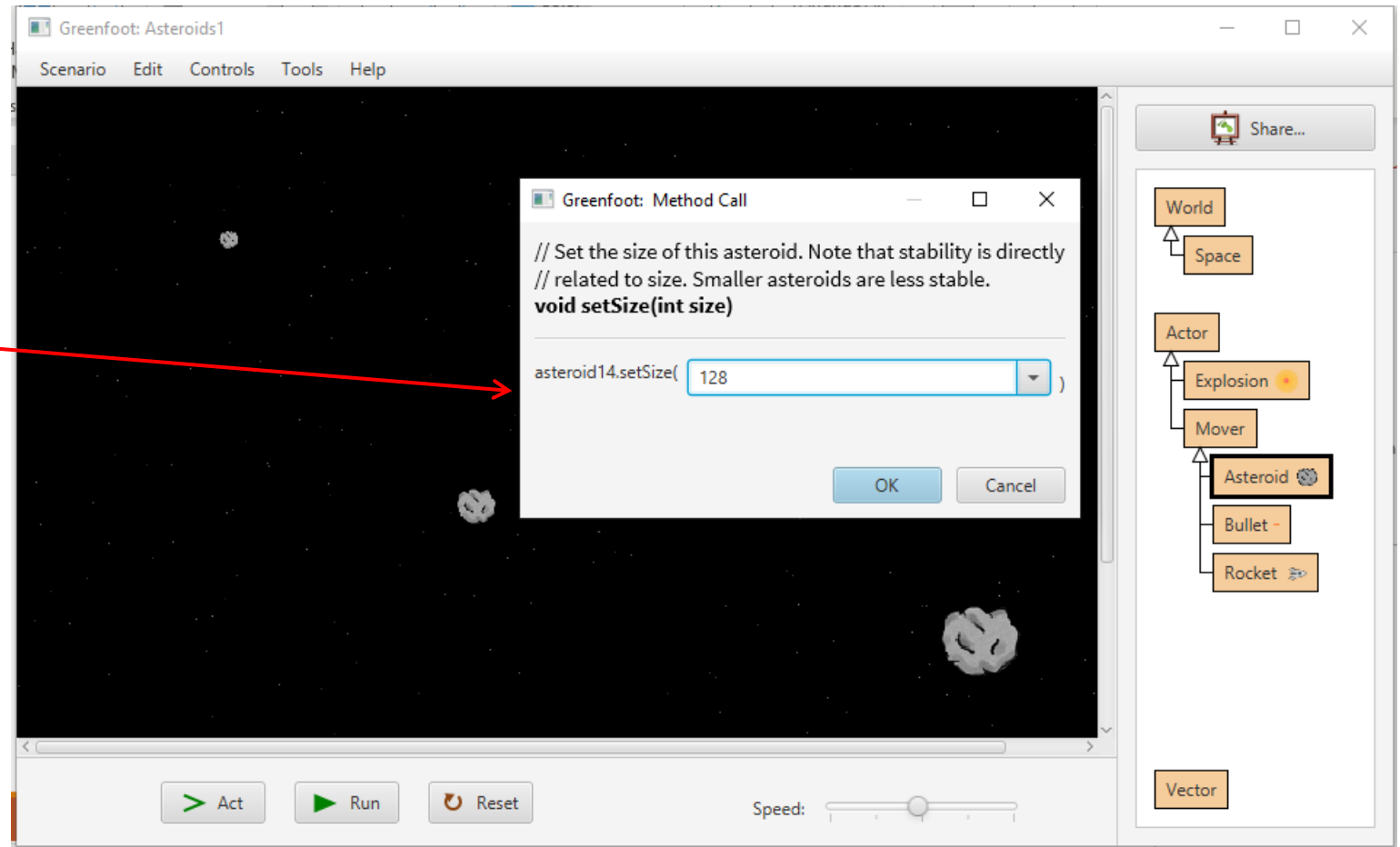
Right Click on the Object  
and Select **Inspect**







Right Click on the Object  
and Select setSize(int size)  
and type 128



# Step 5

---

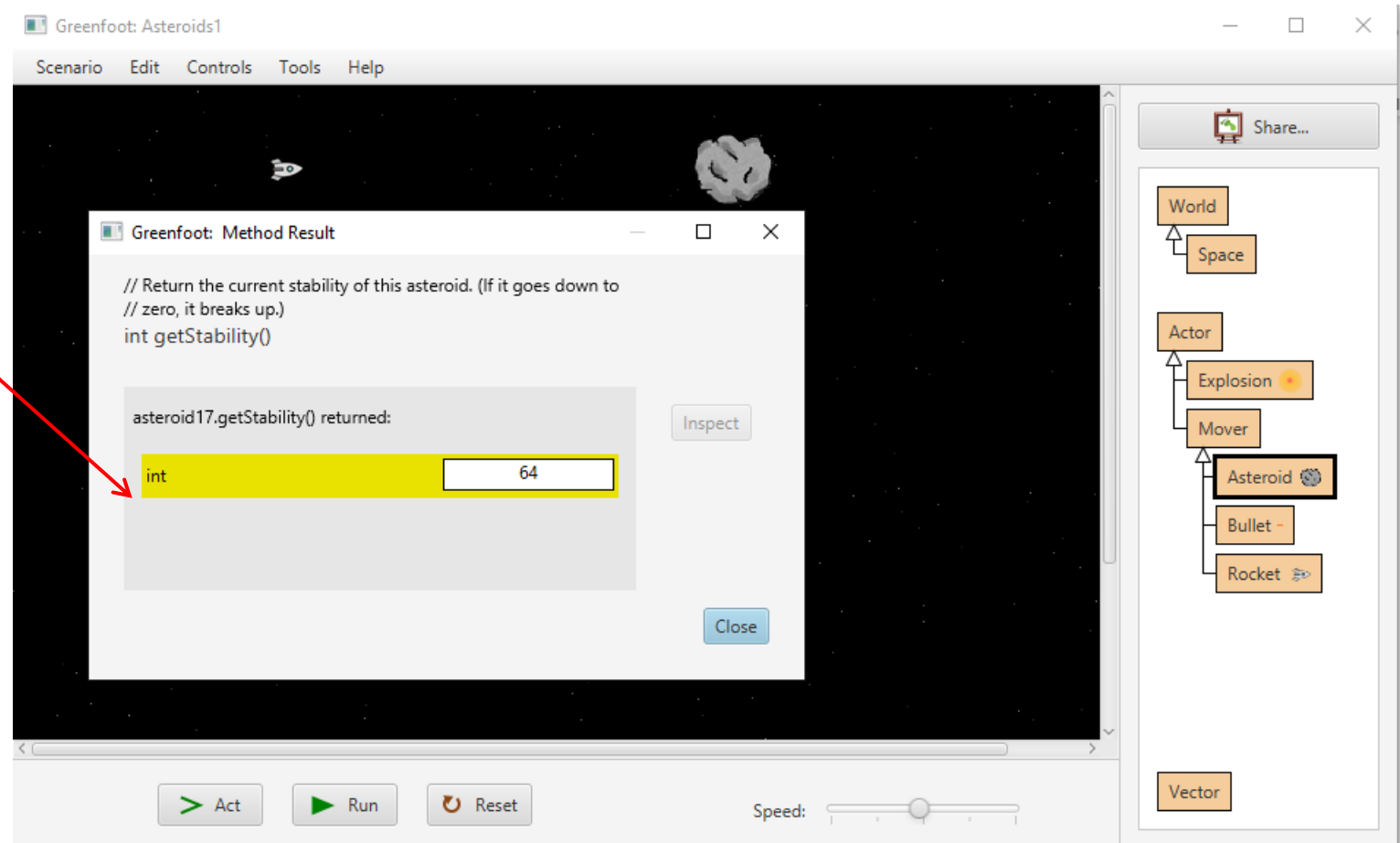
Asteroids have an inherent **stability**. Each time they get hit by a bullet, their stability decreases. When it reaches zero, they break up.

What is their **initial stability** value after you create them? By how much does the stability decrease from a single hit by a bullet?

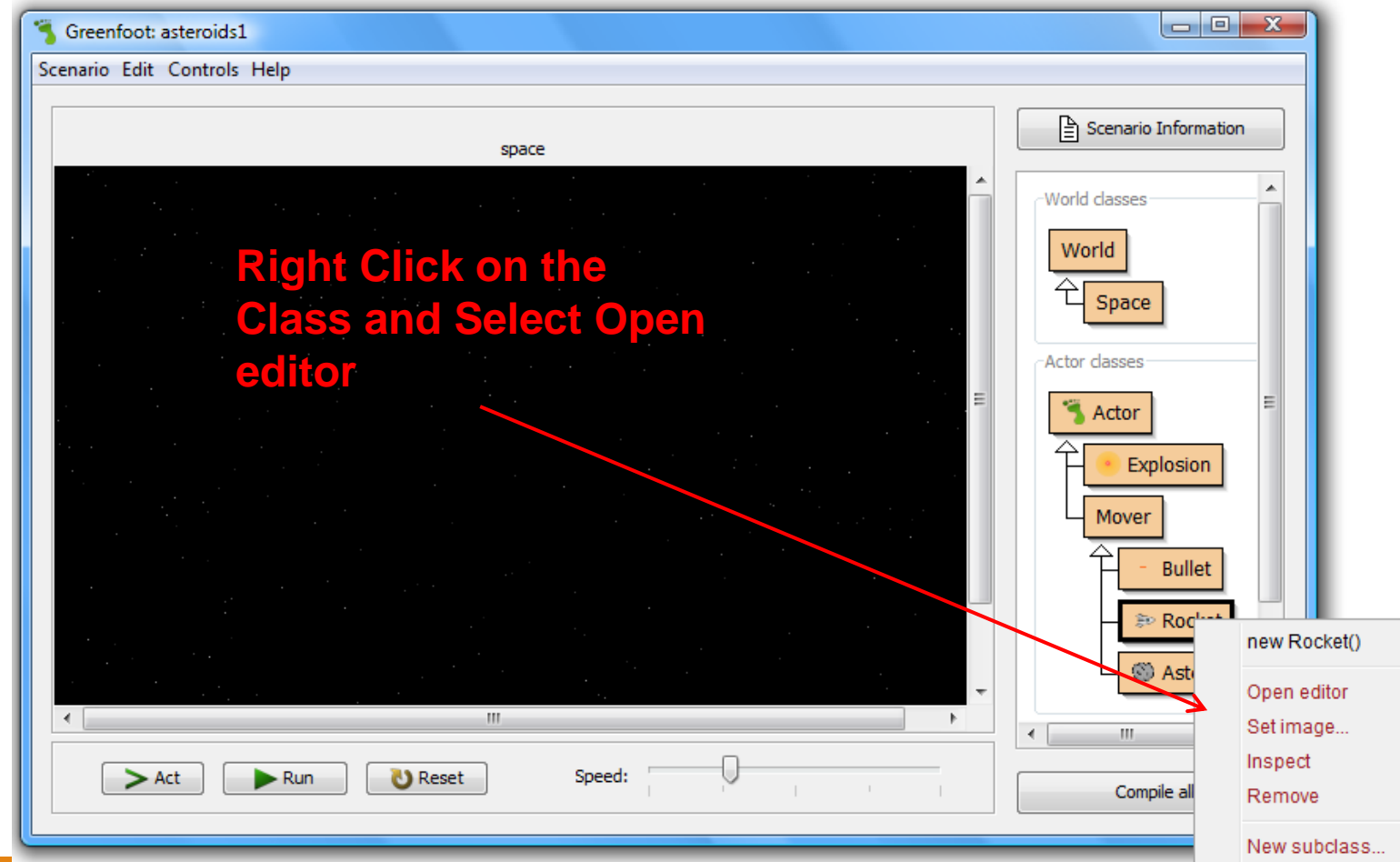
(Hint: Just shoot an asteroid once, and then check the stability again.

Another hint: To shoot the asteroid, you must run the game. To use the object menu, you must pause the game first.)

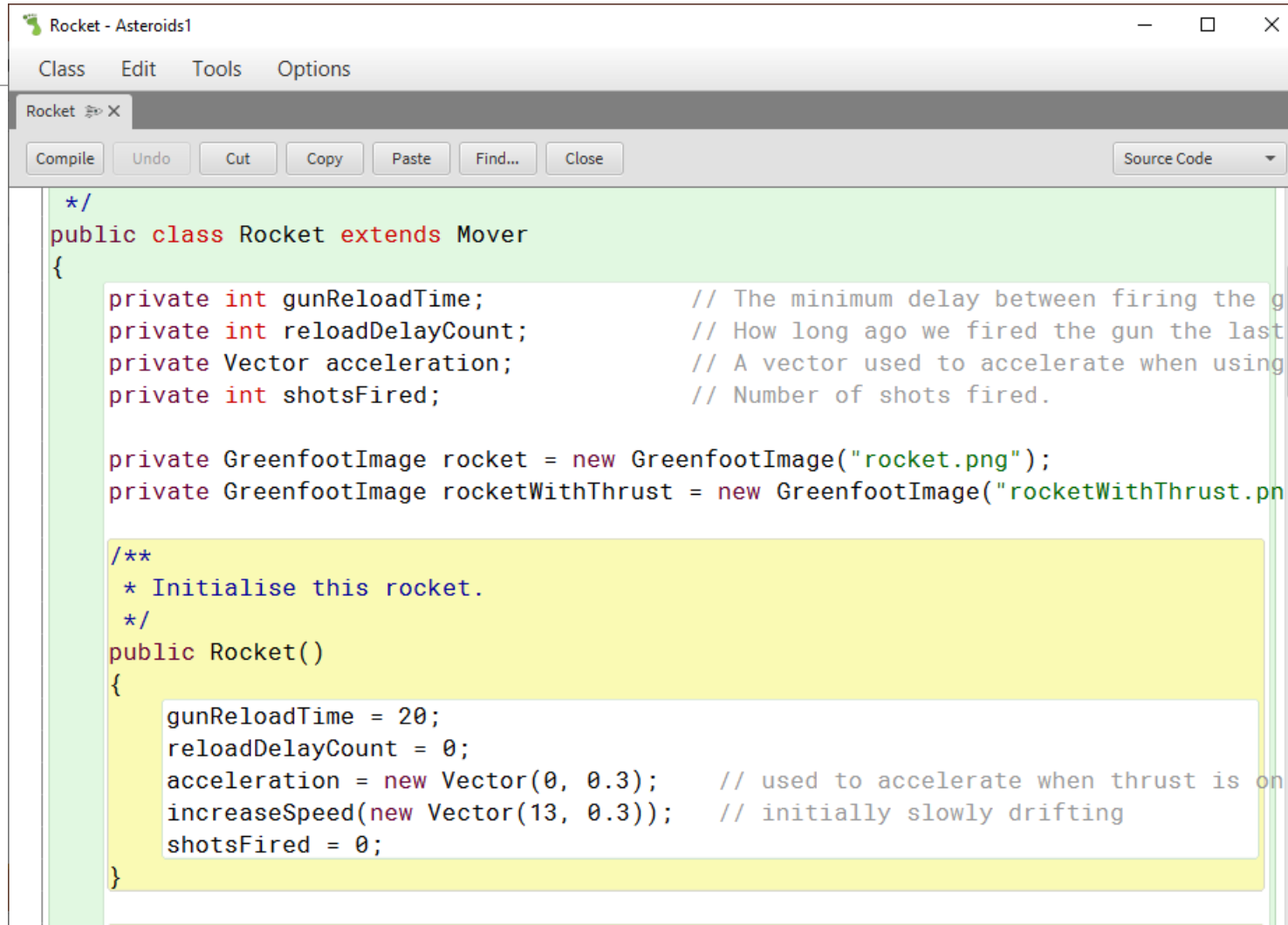
- initial **stability** is 64
- After first hit by a bullet, **stability** becomes ?
- After second hit by a bullet, **stability** becomes ?



# Source Code



# Source Code for Rocket



The screenshot shows a Java IDE window titled "Rocket - Asteroids1". The menu bar includes "Class", "Edit", "Tools", and "Options". The toolbar contains buttons for "Compile", "Undo", "Cut", "Copy", "Paste", "Find...", and "Close". A "Source Code" dropdown menu is visible on the right. The code editor displays the following Java code:

```
*/
public class Rocket extends Mover
{
    private int gunReloadTime;           // The minimum delay between firing the g
    private int reloadDelayCount;        // How long ago we fired the gun the last
    private Vector acceleration;          // A vector used to accelerate when using
    private int shotsFired;               // Number of shots fired.

    private GreenfootImage rocket = new GreenfootImage("rocket.png");
    private GreenfootImage rocketWithThrust = new GreenfootImage("rocketWithThrust.png");

    /**
     * Initialise this rocket.
     */
    public Rocket()
    {
        gunReloadTime = 20;
        reloadDelayCount = 0;
        acceleration = new Vector(0, 0.3); // used to accelerate when thrust is on
        increaseSpeed(new Vector(13, 0.3)); // initially slowly drifting
        shotsFired = 0;
    }
}
```

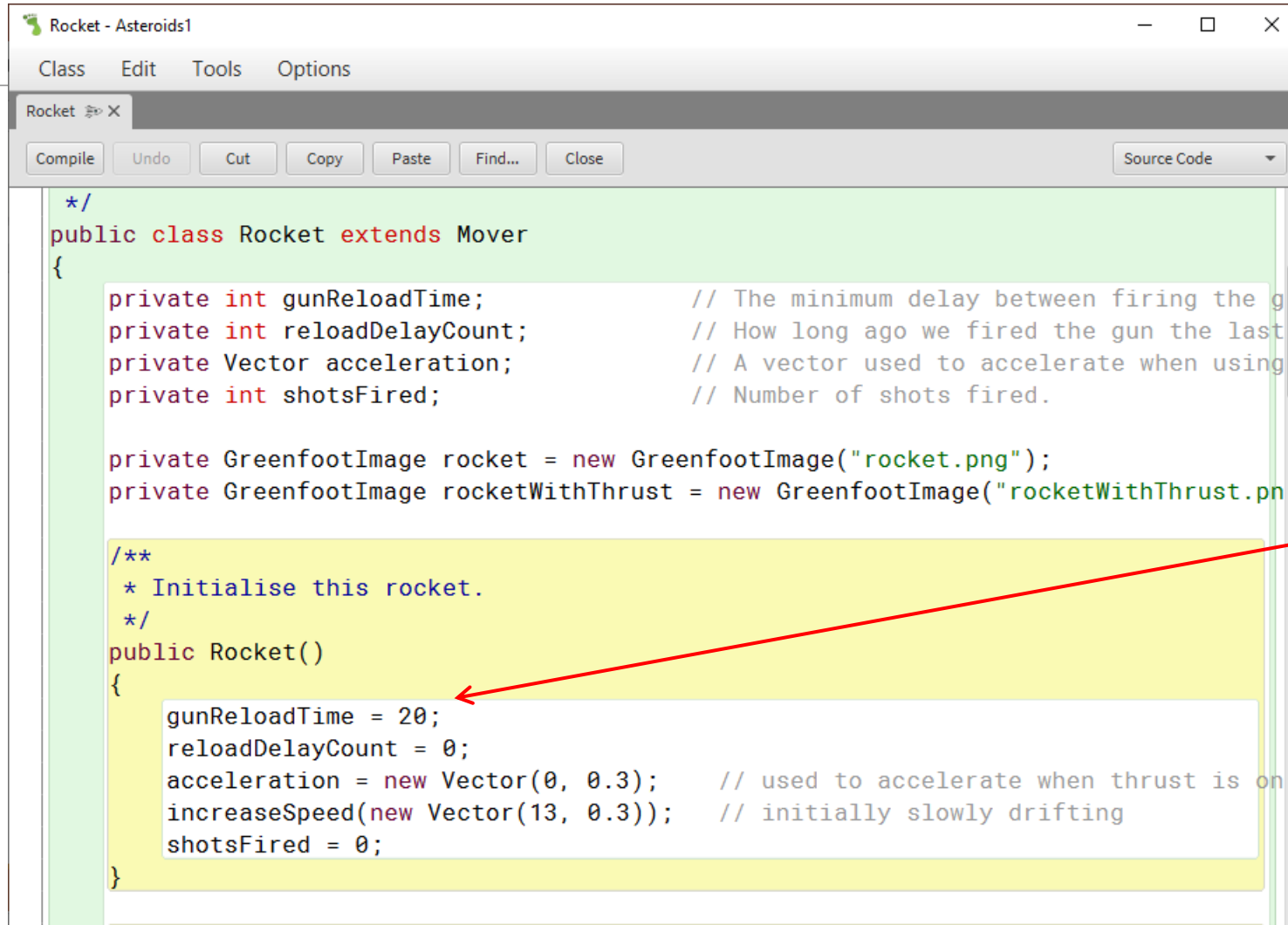
# Step 6

---

Make the **change to the Rocket class source** code as described in next slide. Close the editor and compile the classes.

Try it out: rockets should now be able to fire quickly right from the start.

# Source Code for Rocket



```
 Rocket - Asteroids1
Class Edit Tools Options

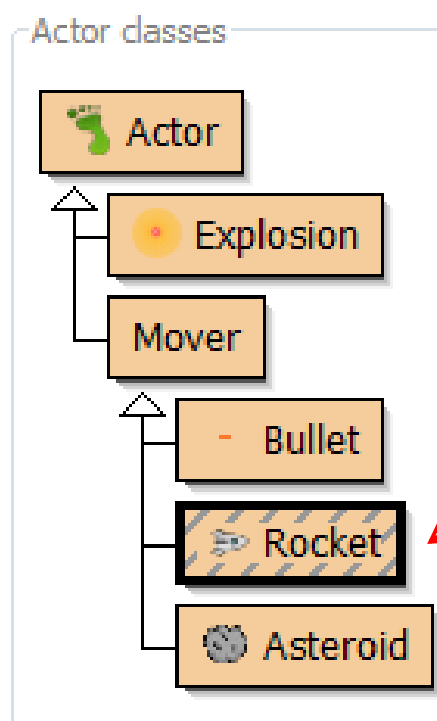
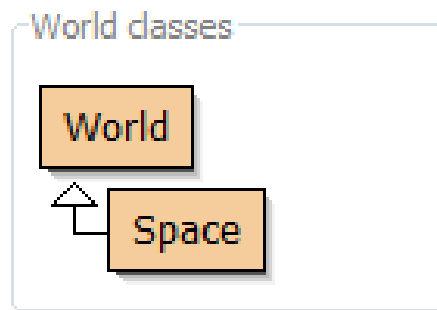
Rocket X
Compile Undo Cut Copy Paste Find... Close Source Code

*/
public class Rocket extends Mover
{
    private int gunReloadTime;           // The minimum delay between firing the g
    private int reloadDelayCount;        // How long ago we fired the gun the last
    private Vector acceleration;          // A vector used to accelerate when using
    private int shotsFired;               // Number of shots fired.

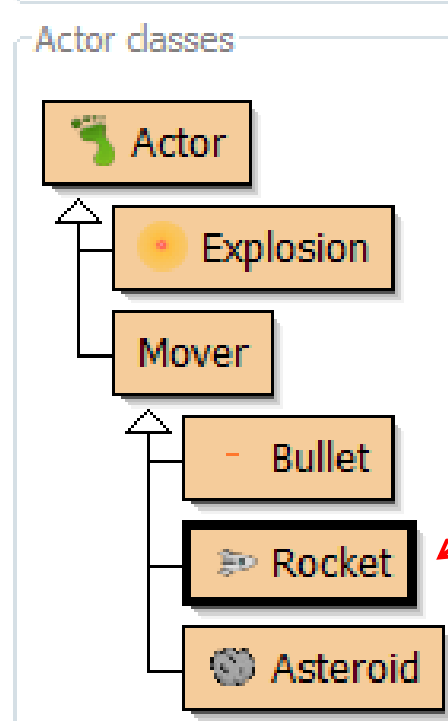
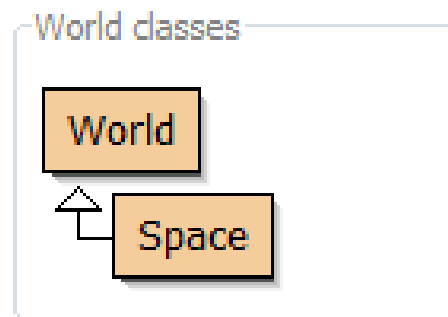
    private GreenfootImage rocket = new GreenfootImage("rocket.png");
    private GreenfootImage rocketWithThrust = new GreenfootImage("rocketWithThrust.png");

    /**
     * Initialise this rocket.
     */
    public Rocket()
    {
        gunReloadTime = 20;
        reloadDelayCount = 0;
        acceleration = new Vector(0, 0.3); // used to accelerate when thrust is on
        increaseSpeed(new Vector(13, 0.3)); // initially slowly drifting
        shotsFired = 0;
    }
}
```

**Change  
gunReloadTime from  
20 to 5**



**Class Changed**



**Class Compiled**



---

# Questions

# ?