159.261 Lab 5
Asteroids – Part 2

In this lab we are going to continue work on the asteroids game by upgrading the graphics from primitive shapes to sprites taken from a spritesheet.

1) Drawing with Sprites

1. Compile and run the starting code `Lab5.java` (you will need `GameEngine.java` as well). This code should be similar to your solution to *Lab 4* except that the spaceship is now drawn using a sprite rather than a triangle.
2. Open the file `spritesheet.png` and look at the images it contains. These sprites are 240x240 pixels each and are stored in a grid.
3. Read through the `init()` and `drawSpaceship()` functions to make sure you understand how spaceshipImage is loaded from the sprite sheet and drawn on the screen. Note that although the sprite is of size 240x240 pixels, it is drawn on the screen with a size of 60x60.
4. Currently the laser is drawn as a single line and the asteroid is drawn as a circle. Add code to the `init()` function to load the laser and asteroid images from the sprite sheet into the variables `laserImage` and `asteroidImage` - you will need to use the function `subImage(Image source, int x, int y, int w, int h)`.
5. Modify the functions `drawLaser()` and `drawAsteroid()` to draw these objects using the sprites you have just loaded from the sprite sheet rather than with primitive shapes.
6. If you get stuck with any of these exercises, check the spaceship functions to see how they have been written.

2) Adding Rocket Engines

This exercise adds visual feedback to the player about the motion of the spaceship by drawing extra images over the spaceship when the left, right or up arrow keys are held down. These sprites represent the exhaust of the rocket engines that accelerate and rotate the ship.

1. The sprite sheet contains images for the rocket engines of the spaceship to make it accelerate forwards and turn left and right. Work out what the coordinates for these images are in the spritesheet.
2. Add code to the `init()` function to load these three images. You will need to create three new `Image` variables to store the sprites.
3. Add code to the `drawSpaceship()` function to draw these images if the player is holding down the up, left or right arrow keys. This information is already stored in the booleans `up`, `left` and `right`.

   Note: the left and right sprites have been carefully made so that drawing them directly on top of the spaceship should make them line up correctly. However, the main engine sprite will need to be offset slightly, find a suitable distance to offset the sprite so that it lines up correctly with the back of the spaceship.

3) Extra Lasers

This exercise involves increasing the number of lasers the spaceship can fire at any one time. To do this you will need to replace the main set of variables for the laser with arrays.

1. Create a variable called `maxLasers` that will control the maximum number of lasers that can be active at any given time. The following variables that represent properties of a laser will need to be replaced with arrays (or similar data structure).

    - `laserPositionX`
    - `laserPositionY`
    - `laserVelocityX`
    - `laserVelocityY`
    - `laserAngle`
    - `laserActive`

    Note: all the lasers appear exactly the same so we do not need to make an array of laser images.

2. Modify the code to use these new laser arrays. For example, the `fireLaser` function will need to iterate through the array of lasers to find one that can be fired. The `drawLaser` and `updateLaser` functions will need to iterate through the arrays to draw and update any active lasers.
3. Finally, you will need to edit the code that checks for collisions between the laser and the asteroid to check for a collision between the asteroid and any of the active lasers.