CISC 110 Lab 6: Fireworks

Complete the following during your lab time.

1. Write a version of the function you wrote in a previous lab for generating a random integer. This version uses the Math.random and Math.round methods to generate a random integer between 0 and the function's integer parameter. Here is the header line for your function:

   ```
   // Generates and returns  a random integer >= 0 and <= n
   function randomVal( n: int ): int
   ```

2. Test your randomVal function to ensure it's correct by writing a for loop that iterates 20 times, calls the randomVal function with 10 as the argument, and displays the result. Note that one should always test a function definition one writes to ensure it's correct before including it in a larger program.

3. Draw a line with the exact following code. Copy this into your script and then try changing some of the arguments to see what changes occur.

   ```
   // Create a new Shape object
   var lines: Shape = new Shape( );
   // Specify line thickness and color
   lines.graphics.lineStyle( 6, 0xff0000 );
   // Set current drawing position
   lines.graphics.moveTo( 100, 200 );
   // Specify line end point & reset current drawing position to that end point
   lines.graphics.lineTo( 200, 350 );
   // Add line to stage
   addChild( lines );
   ```

4. Draw three more lines that start at the same center point, but have three different end points. You can have more than one line in a Shape object, so just add to the drawing script above: move the drawing position back to (100, 200) before drawing each additional line.

5. Change your script to use the randomVal function to choose three random end points for the three lines. Call the randomVal function six times, two for each point.

6. Write a for loop that draws 15 lines with the same center point, but each with a random end point. Only the moveTo and lineTo instructions will need to be inside the for loop. You will only have one call to each, since they will be repeated 15 times by the for loop. Now your display should look like a firework (sort of …)

7. Define a function that will create a firework with a specified color and a specified center point, and a range of random end points. To do this, you will place your firework-drawing script inside of the function definition, and then change some of the numbers to be the parameter names instead.

   However, you will not have the call to the addChild method inside your function. Instead you will return the firework Shape object you have created without displaying it.

   Here are comments and the header line for your function:

```
/* Create a firework with a given center point and color,
*  and with random length lines within a specified coordinate range
*  Parameters:
*        centerx:  the x coordinate of the firework's center point
*        centery:  the y coordinate of the firework's center point
*        color:  the color of the firework's lines
*        xRange: the upper limit on the x-coordinate of the end point of each line
*        yRange: the upper limit on the y-coordinate of the end point of each line
*  Returns:  A Shape object that contains the firework graphic
*/
function createFirework( centerx: int, centery: int, color: Number, xRange: int,
yRange: int ) : Shape
```

8. Create an array. Call your createFirework function several times with different argument values to create several different firework Shape objects and assign each of these Shape objects to a different position in your array. Then you will have an array of Shape objects (instead of an array of integers or Strings or other types).

9. Use a for loop to display all of the fireworks in your array by calling the addChild method for each of them.

The step-by-step instructions in this and other labs are intended to show you the process of developing a script incrementally:

- Start with a simple version of what you want to do and implement and test that.
- Incrementally add more complexity or additional steps, testing each before going on to the next.
- Major steps or steps that will be repeated several times should be defined as functions. All function definitions should be tested separately before incorporating them in your script (unless the script is a simple script that essentially does test the function, as with the second function definition in this lab).