

# 15-102 Exploring Programming with Graphics

## Homework 3

**Start:** Thur 5.19.11  
**Due:** in class Mon 5.23.11  
**Goal:** Lost in Space...

### Course Web Site:

[www.andrew.cmu.edu/course/15-102](http://www.andrew.cmu.edu/course/15-102)

### Reading:

Posted on the calendar web page available from the link shown above

### Assignment:

You will continue your work with variables, assignments, arithmetic and expressions just as you did in Homework #2. You will also continue to explore some new functions.

### Specifications:

1. \_\_\_\_\_ Put the required information in a comment at the top of your program
2. \_\_\_\_\_ You will draw only one of your initials (the most interesting one) but you will draw it in three-dimensional space.
3. \_\_\_\_\_ The initial must be primarily composed of a series of spheres, boxes (using the **sphere( )** and/ **box( )** functions from the **3D Primitive** section ) and the **shape-2d** functions of the Processing API.
4. \_\_\_\_\_ The initial must be extend out into three planes. If all of the shapes that make up your initial lie in a single plane in the **x**, **y**, or **z** axis, you will lose points. Look at the demo below.
5. \_\_\_\_\_ The initial must always be centered in the graphics window. **You may assume that the window will always be square.**
6. \_\_\_\_\_ The arguments you use with the call of the **translate( )** function to move the origin to different places in space must use variables and expressions based on the Processing system variables **width** and **height** so **your initial grows or shrinks when Jim alters your window's size (heh...heh..heh...)**. The variable **width** is the horizontal number of pixels in the graphics window. The variable **height** is the vertical number of pixels in the graphics window. Do not hardwire the exact location of the elements that compose your initial. We will grade this by altering the size of the graphics window to see if your initial grows and shrinks appropriately and remains in the center of the window. Jim will demo this for you in class on Friday.
7. \_\_\_\_\_ You may use any of Processing's methods to control color, fill, etc. You may use the **2-D Primitives** if you wish to "enhance" your initial.
8. \_\_\_\_\_ The use of color is up to you
9. \_\_\_\_\_ The use of rotation is up to you – be careful if you use it. . .

### Exploring:

You may want to look at the **Lights** subset of functions. Some of the sample code in the API uses stuff that we have not covered. Unless you have written your own functions in

a previous life, do not add the code that starts with **setup( )** or **draw( )**. Those are functions similar to the ones you have been using, but they are not needed for this homework and putting them into your code could cause problems.

This Homework continues the exploratory nature of programming where you are asked to do a great deal with little or no help. You are **strongly** encouraged to form and work within a small group for this assignments.

**Tip:**

The order of operations is: translate → draw. Anything drawn before a translation is not affected by subsequent translations. If you are going to rotate stuff, then the order is: translate → rotate → draw. As before, old stuff on the screen is not altered or shifted or rotated by subsequent rotations.

**Grading:**

You need prints of the output and the code at the start of class Wednesday just as you did for the first two homeworks. Remember to use **saveFrame( "hw3.jpg"**) to make the picture of your output.

**Handin:**

This will be covered in class tomorrow.

**Sample:**

Another of Jim's attempt at being "*artistic*"...

