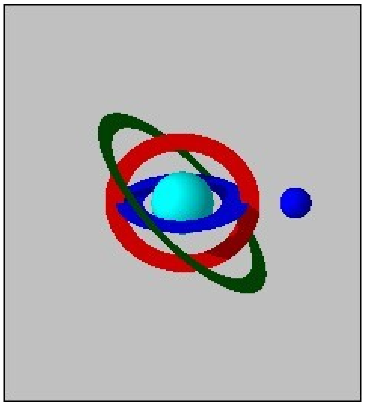
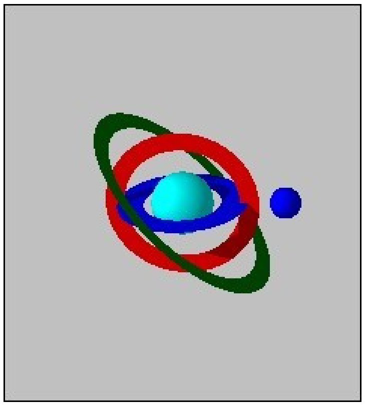
CMPSCI 119
LAB #11 – Wiggle Stereoscopy EXTRA CREDIT
Professor William T. Verts

The goal of this Python programming assignment is to create a simple animation from two images of the same scene from different viewpoints. By quickly switching back and forth between the two images, creating a "wiggle" effect, the relative motions of the objects in the images give an enhanced sense of three dimensions.

Download from the class site the file called **`Images.zip`** and unpack its contents into the folder for your Python programs. The contents are four files in two pairs: `Rings_Left.jpg` and `Rings_Right.jpg` form one pair, and `Scene_Left.jpg` and `Scene_Right.jpg` form the other pair, as shown below. There isn't much difference between the images in each pair but you would see a difference if you configured them to fit into an old-time stereoscope. Indeed, the second pair of images were extracted from an old-time stereoscope slide.





Your program (which does not depend on any previous code) must first ask the user to select each of the two image filenames, as well as a time-delay to wait after showing each image. Your program then goes into a loop that copies each image successively onto the visible canvas and waits for the amount of the delay after each repaint (usually somewhere in the range from 0.1 to 0.5 seconds). The loop should run for exactly 100 cycles before halting.

The JES routines that you will need include some that you have used before, but also some new functions as well. Here is a list:

`repaint(`*`Canvas`***`)`**

Updates and shows the canvas on screen.

`Filename` `=` **`pickAFile()`**

The `pickAFile` function brings up a dialog box that prompts the user for a file, and returns the name of the file as its result.

`Canvas` `=` **`makePicture(`***`Filename`***`)`**

The `makePicture` function is given the name of a file containing an image, and returns a new canvas containing the picture contained in the file. Crashes the program if the file is not of an image.

`Canvas2` `=` **`duplicatePicture(`***`Canvas1`***`)`**

This new JES function creates and returns a second canvas containing a duplicate of the first. Your solution can take advantage of this function, or it can be omitted in some approaches.

`copyInto(`*`Canvas2, Canvas1, X, Y`***`)`**

This new JES function draws `Canvas2` onto `Canvas1` with the upper-left corner of `Canvas2` at coordinate location `<X,Y>` in `Canvas1`. For this program `X` and `Y` are both zero. This function is critical to your solution; unfortunately it is also fairly slow.

Your final program using these functions will be quite short overall.

Finishing Up

Make certain that your name is in a comment at the beginning of your program.

Submit this assignment through the on-line form as Lab #11.