

## Update F12

I have made considerable progress in the project. Although I am currently a week behind schedule (according to proposalM9), I think I will be able to catch up fairly quickly. The most updated VPython code can be found on the webpage under the link *double\_slit\_F12.txt* under *Week 12*.

This program, in its current state, sets up a laser and two screens. The laser is merely ornamental. Fringes show up as yellow bands on the second screen. As the screen1-to-screen2 distance ( $L$ ), slit distance ( $d$ ), light wavelength ( $w$ ), and screen2 size ( $S$ ) change, the appropriate number of fringes are shown. Although slit width ( $a$ ) can be varied, it doesn't have an effect on the experiment's result. As the initial comments explain, when the program is run in VPython, the scene must be rotated to see it properly. To see the tiny slits, the scene must be zoomed extremely. In the final version, maybe another window will be zoomed in on the slits so they can be seen more easily.

I have also begun to start programming the calculations of the intensity. In the current code, the intensity is calculated only for band  $m=1$  (where  $m$ , as explained in proposalM9, labels the individual bright fringes). This intensity is in  $W/m^2$ .

### Future goals:

In the next week, I plan to include individual intensity calculations for each bright fringe. The intensity will decrease as  $m$  gets larger, and in turn the opacity of the bands will decrease. I also want to include the factor of *peak width*, or the width of the bright fringes. Currently, the bands appear to—incorrectly—have a fixed width (or height, in this case) of .1. I also may start to write a program to exhibit Rayleigh's Criterion. For this, I would only have one slit of varying width. This would change the apparatus' ability to resolve two far away objects.

### VPython experimenting...

I have also been experimenting a bit with how to code things in VPython (Python). Specifically, I am coming up with different methods of programming the calculation of Fibonacci's Sequence. Some of the ways are more efficient than others. Currently, there is a short file that has two ways of calculating the sequence. Maybe in the future I can visualize the sequence in an appropriate spiral.