

Problem Description

The artist Josef Albers made the interaction of color the focus of his art and teaching. A key idea of his work is that the perception of color is a relative or contextual phenomenon. In particular, the color we perceive is highly dependent upon other surrounding colors. This week's homework assignment is to give you some experience with this concept, by encouraging you to experiment with how colors can be “shifted” by other colors.

Albers developed a number of problems to develop an understanding of color interaction. One of these is the Three Color problem. The idea is to make three different colors appear like four. Two colors are used as rectangular “background” fields and placed next to each other. In the middle of each field is a small shape of the third color. The idea is to choose the three colors such that the color in the middle of the left field appears different from the color in the middle of the right field, thus creating the illusion of four different colors.

Note that, very clever solutions to this can make the inner left color appear *identical* to the right background color while simultaneously having the middle right color appear to be the same as the left background. This is a “three-to-two” variant of the problem. For this assignment you will consider the “three-to-four” variant. Be creative with your experimentation, the provided code uses the HSV color representation and you can cause colors to shift in hue, saturation, and/or value.

To experiment, download a program called `albers`, included on the course webpage as `albers.zip`. This program can be unzipped and run on the SoC lab machines, and employs OpenImageIO (OIIO). One secondary purposes of this assignment is to give you more practice working with OIIO. After downloading, the program should compile and the executable program will be named `albers`. When you run this program, it gives you an interface to use to do your own Three Color problem. Help is available via the menu (right mouse) or just press the H key. When you have a solution you like, save your image to an image file and save the “project”. The project file is a text file that will contain the HSV colors that you chose. You should complete at least two example Three Color problems. Of course, further experimentation is encouraged! You will also be interested in knowing the innerworkings of this code for more examples using OIIO, OpenGL/GLUT, as well as how to convert between HSV and RGB color spaces.

Note: if you are building on a different system than the SoC Linux environment, you may have to modify the makefile accordingly. The makefile provided with the code should work with the SoC Linux environment, and potentially on OSX as well.

Requirements

For this assignment, 4040 students will be graded as follows.

You must create a simple webpage which displays your solutions. It does not have to be fancy, but it should include a saved version of the image as well as the specifications for the colors. The webpage should also include your name and the date. Your web page should be named `[your user name].html`. For example, mine would be `levinej.html`. The very best solutions may be put on display for the course webpage.

Be careful not to use absolute paths in your html file. When you submit, make sure that you include the image files.

Advanced Requirements for 6040 students In addition to the 4040 requirements, you must modify the albers code to support setting the color of the foreground shape independently for the left and right sides. This is a minor change; you'll need to allocate storage for the additional color in the Project class and draw the mirrored shape using this additional color. You must also modify the interface so that you can control this color. There is no need to submit the modified code.

After making the changes, you need to experiment with the inverse of the Three Color problem. You are to submit two example that show the variation of Three Color problem where four colors appear as three. Thus, instead of having identical foreground colors appear differently in the left and right, your task is to have different colors appear identical. By carefully select a different color for the left and right hand sides and adjust the background colors you can make the two middle shapes appear to be the same.

Submission

Submit using the handin procedure outline at <https://handin.cs.clemson.edu/>. You are welcome to use the commandline interface, but the web interface is sufficient. The assignment number is pa02.

Finally, since we are using multiple files (please include the image and project files when you submit), only submit a single file which has aggregated everything. You need not submit your modified code, only the output you've produced. This file should be named [username]_pa02.tgz where [username] is your Clemson id. To create such a file, follow the same directions outlined in Programming Assignment 01.

Rubric for Grading

This assignment will be graded on completion, not quality. Successfully completing two examples will earn you 3 points for each. For 6040 students, each of the four examples are worth 1.5 points each. Successfully submitting a simple webpage will earn you the remaining 4 points (provided you follow the directions for naming and include the correct contents as outlined above).

The expectation is that while it might be a little tricky to complete the Three Color problem, you will spend a significant portion of your time explicitly studying the code. In particular, understanding how I've used OIIO to write the data being drawn by OpenGL is key, and using this project as an template for future projects maybe be helpful.