

CMSI 371-01

COMPUTER GRAPHICS

Spring 2015

Assignment 0226 Feedback

Outcomes that ultimately cover both 2D and 3D max out at | for now because we are dealing only in 2D. They will expand to their full potential with the 3D course work.

Joaquin Loustau

joaquinloustau / joaquinloustau@gmail.com

Great to see that custom minion scene for your filters! Your single-pixel filters are fairly standard but serviceable, with the neighborhood filters (particularly “difference”) a little more interesting. As for the gradient circle, it certainly looks right; let’s see what the code reveals now...

1. Just remember that commas in code are like commas in text: they are more readable if a space comes after them. (4c)
2. Ah, interesting value from which to base a gray level. Good that you credit your source, too. (2c, 3c, 4d)
3. Yep, that is a typical sepia filter, but note a dash of inefficiency in your code: you are performing the same color computation twice! Instead, just compute the value; then, if it is over 255, set it to that. See the inline comment for the proposed refactoring. (4b)
4. Duuuude, are you indenting by 2 or indenting by 4? Make up your mind! :-P (4c)
5. This one is pretty neat huh? Did you discover it yourself or did you read about it on the web? (2c, 3c)
6. Remember to put spaces around binary operators, too. (4c)
7. Poor Pythagoras, reduced to a single one-line JavaScript function :)
8. This is functionally right, with the single point of improvement being further DRYness: observe that the bodies of your two `for` loops are virtually identical except for the y -coordinate that you are plotting. The bodies can thus be unified into a function `(i, row)` (more or less), turning both `for` loops into one-liners. Although taken into context, really not a big deal. (4b)

1a — +

2c (max |) — |

2d — +

3c — +

4a — +

4b — +

4c — | ...more of a reminder to be thorough than a real knock on readability.

4d — +

4e — +

4f — +