

# CMSI 371-01

## COMPUTER GRAPHICS

Spring 2016

### Assignment 0308 Feedback

Outcomes that eventually cover both 2D and 3D continue to max out at | for now because this assignment remains in 2D.

Dustin Kane

ranneyd / dustinpkane@gmail.com

*Notes while running (high-priority notes are marked with \*\*\*):*

- Filters all look good. Spotlight is particularly fun.
- The circle gradient looks fine too, though it seems to take longer than I'd expect. Will have to check on that in the code.

*Code review (refer to <http://lmucs.github.io/hacking-guidelines/> for code-review abbreviations):*

1. Single-pixel filter code works out nicely. The computations used for spotlight are actually very close in spirit to our 3D light calculations.  $+(2c, 3c)$
2. All good on the neighborhood filters too. Fun for you to adapt the online sharpen filter to build on the basic edge detector.  $+(2c, 3c, 4b)$
3. I appreciate what you did in the primitives portion, generalizing the gradient code for reuse by both the rectangle and circle. You can see the performance cost though: this approach scans all of the pixels twice, once when building the gradient then again when drawing. Of course that doesn't mean that this takes twice as long, but there will still be appreciable overhead, mostly from additional function calls. With the circle, you *do* get concrete excess computation because the circle won't touch the entire rectangle, and thus, some pixel computations are wasted (around 21.5%, actually:  $1 - \pi/4$ ). I'll call it a  $4b$  push:  $+4b$  for the reusable design,  $-4b$  for the decreased efficiency.

$1a$  — +

$2c$  (max |) — |

$2d$  — +

$3c$  — +

$4a$  — +

$4b$  — + ... $4b$  push in effect.

$4c$  — + ...Spacing is still glitchy, but didn't bother me as much this time.

$4d$  — +

$4e$  — Sufficiently descriptive messages alongside good commit frequency plus time management. (+)

$4f$  — Submitted on time. (+)