

ASSIGNMENT: Fun with Fragment Shaders

Due May 16 by 10pm **Points** 20 **Submitting** a text entry box or a website url

Available May 9 at 8:15am - May 16 at 10pm 8 days

This assignment was locked May 16 at 10pm.

Assignment 4

Fun with Fragment Shaders

Task

Let's have some fun manipulating textures with our fragment shader to create some cool effects! You will use HTML's WebGL2 API.

Four Standard Filters (to earn a B-: 16 pts)

- Implement 4 effects as shaders:
 - Black & White: **4 pts**
 - Fish Eye: **4 pts**
 - Ripple: **4 pts**
 - Toon: **4 pts**

Additional Filter (to earn an A)

- Implement 1 additional effect as a shader: **4 pts**
 - Your choice! ...must consult with instructor prior to implementation.

Details

Your objective is to implement some fun (non-photo realistic) rendering techniques using the fragment shader. Given a texture, please implement the following filters using GLSL:

- Black & White
 - Luminance is calculated as $L = 0.299 * \text{Red} + 0.587 * \text{Green} + 0.114 * \text{Blue}$
 - Black and white images are created by assigning the luminance value to all 3 color components
- Fish Eye
 - This optical effect can be accomplished with a barrel distortion
 - In order to accomplish a barrel distortion, implement the following equations:

- scale and translate the texture coordinate such that it is in the range [-1.0, 1.0]
 - multiply by 2, then subtract 1
- calculate $\theta = \arctan(\text{texture_coordinate_y}, \text{texture_coordinate_x})$
- calculate radius = magnitude of texture coordinate, raised to the power of 1.5
- calculate fish eye texture coordinate = (radius * cos(θ), radius * sin(θ))
- re-scale final texture coordinate in range [0.0, 1.0]
 - final texture coordinate = 0.5 * (fish eye texture coordinate + 1.0)
- Ripple
 - This filter creates a ripple effect similar to dropping a pebble in a pond
 - In order to accomplish a ripple effect, implement the following equations:
 - scale and translate the texture coordinate such that it is in the range [-1.0, 1.0]
 - multiply by 2, then subtract 1
 - calculate radius = magnitude of texture coordinate
 - calculate a texture coordinate offset = texture_coordinate * (sin(radius * 30.0 - time * 5.0) + 0.5) / 60.0
 - calculate final texture coordinate = original_texture_coordinate + texture_coordinate_offset
- Toonify
 - This filter will round each color component to one of 5 levels (0.0, 0.25, 0.5, 0.75, 1.0)
 - Rounding can be done by multiplying by 4.0, then rounding, then dividing by 4.0
 - The result should be a cartoon-ish looking image
- Custom
 - Please consult with the instructor

Note: all modifications will be in the frag shader files.

Starter code is available on GitHub: <https://github.com/tmarrinan/cg-fragshaders> (<https://github.com/tmarrinan/cg-fragshaders>). Please **fork** your own version of the code, then enable GitHub Pages in the project's settings (change *Source* from *None* to *master branch*).

Submission

Code should be saved in a repository on GitHub while working on the project. In order to submit, enter the the project's live website URL for the assignment (in Canvas).

Also submit a checklist of what you feel you have accomplished from the rubric above, and include your total expected score. This can be made as a comment once you submit the URL.

Deadline

This assignment is due Monday, May 16 at 10:00pm.

Demonstrations

Demonstrations will be given during the first hour of the Final Exam time period (Section 01: T 8:00-9:00am, Section 02: F 8:00am-9:00am). Each person will have 2-5 minutes to present their project.

For the demonstration, you should use your own default video and showcase the shaders you've successfully implemented.