Midterm by Benjamin Cooper & Lansingh Freeman

Overview: The project implements object movement, mirror reflection textures on objects, and a pixelation post processing effect.

Contributions:
Lansingh Freeman:
- Mirror Shader
- Some pipeline work

Benjamin Cooper:
- Object movement
- Pipeline work
- Pixelation Shader

Goals: We set out to make a number of smaller shader operations which would show out competence in many aspects, while doing more advanced operations. We would achieve this with object movement, a mirror reflection texture, and a pixelation shader.

Justification: As stated, we attempted to show proficiency in a number of aspects and doing more advanced shaders. We have most of the requirements, save for one overarching category that was presented.

Achievements:
- Unique feature: A good effort at generating a reflective mirror texture without the use of a cube map
- Use of a lighting program: Phong was used in the mirror shader, giving a shine to the mirrored textured objects
- Post-processing effect: The pixelation shader was a post processing effect.
- Curve interpolation algorithm: The movement of the object uses a beizer interpolation, and the implementation of the equation is in the geometry shader.
- Used vertex and fragment shaders for rendering: yes, this was achieved.
- All rendering off screen: yes, pretty sure we weren't drawing directly to the front buffer.

Code that was relevant:
Mirror Shader:
mirrorTexture_vs4x.glsl
- Note: Had to be implemented into passTangentBasis_transform_instanced_vs4x.glsl due to an unknown error with texcoord. Code still exists in this shader, which was copied into this other shader to be viewed.

mirrorTexture_fs4x.glsl
- Note: Had to be implemented into drawPhong_multi_forward_mrt_fs4x.glsl due to an unknown error with texcoord. Code still exists in this shader, which was copied into this other shader to be viewed.

Pixelation Shader:
drawTexture_pixelation_fs4x
passTexcoord_transform_vs4x

- Note: Nothing here was modified for pixelation, this is just listed for reference on where the data comes from

Animation:
a3_Demo_Curves_idle-update

- Lines 86-101 were added in, allowing animal3D to handle the interpolation (this was not done as part of the bonus lab)

Pipeline:
a3_DemoState_loading

- Lines 511,550-551(loading of the shaders)
- Lines 735 -747 (creation and activation of new shader programs)

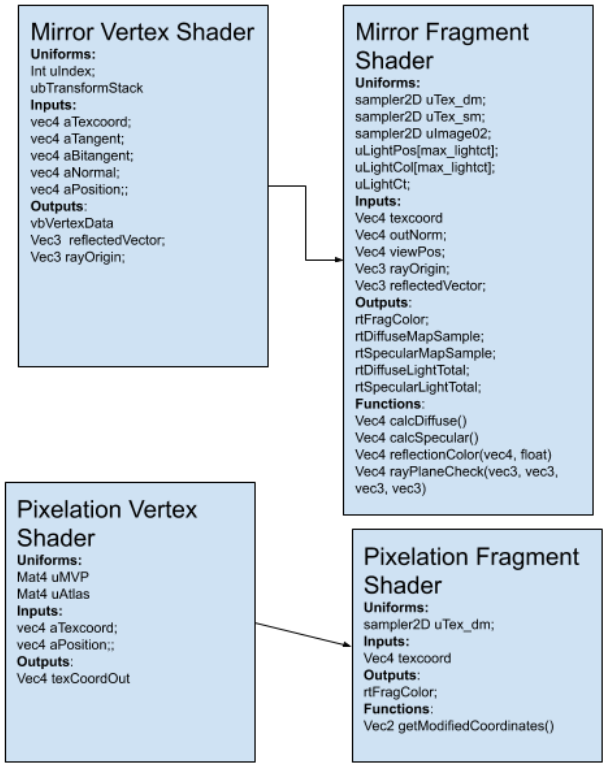
a3_Demo_Curves

- Lines 61 & 71 (adding mirror and shader to the renderer and display)

a3_Demo_Curves_idle-render

- Lines 62 and 69 (adding mirror and pixelation to renderer and display names)
- Lines 260 and 268 (adding mirror and pixelation to programs)
- Line 442 (Binding the skybox texture)
- Lines 713-718 (Activation and setup for pixelation)

UML:


Mirror Vertex Shader
Uniforms:
Int uIndex;
ubTransformStack
Inputs:
vec4 aTexcoord;
vec4 aTangent;
vec4 aBitangent;
vec4 aNormal;
vec4 aPosition;;
Outputs:
vbVertexData
Vec3 reflectedVector;
Vec3 rayOrigin;
Mirror Fragment Shader
Uniforms:
sampler2D uTex_dm;
sampler2D uTex_sm;
sampler2D uImage02;
uLightPos[max_lightct];
uLightCol[max_lightct];
uLightCt;
Inputs:
Vec4 texcoord
Vec4 outNorm;
Vec4 viewPos;
Vec3 rayOrigin;
Vec3 reflectedVector;
Outputs:
rtFragColor;
rtDiffuseMapSample;
rtSpecularMapSample;
rtDiffuseLightTotal;
rtSpecularLightTotal;
Functions:
Vec4 calcDiffuse()
Vec4 calcSpecular()
Vec4 reflectionColor(vec4, float)
Vec4 rayPlaneCheck(vec3, vec3, vec3, vec3)
Pixelation Vertex Shader
Uniforms:
Mat4 uMVP
Mat4 uAtlas
Inputs:
vec4 aTexcoord;
vec4 aPosition;;
Outputs:
Vec4 texCoordOut
Pixelation Fragment Shader
Uniforms:
sampler2D uTex_dm;
Inputs:
Vec4 texcoord
Outputs:
rtFragColor;
Functions:
Vec2 getModifiedCoordinates()