# Description of Changes:

CMakeLists.txt

Changed reference location to Texture.cpp and Texture.h

Not sure whether this will work correctly on your end or not.

Texture.cpp

Start of Program

Added

string GLMultiTexture::\_glsl\_names[4] = { "texture\_sunset", "texture\_face", "texture\_colorwheel", "texture\_blend"};

// Handles 3rd texture (texture\_colorwheel)

GLMultiTexture::GLMultiTexture()

Added

\_texture\_3 = 0;

\_textureIdx3 = -1;

int GLMultiTexture::loadAndCreateTextures

Added

unsigned int channels3;

unsigned int width3;

unsigned int height3;

unsigned char\* data3 = loadBitmapFile(path\_and\_file\_texture\_3, channels3, width3, height3 );

Handling for 3rd texture (labeled with #pragma region Texture 3)

Modified

if(data1 == NULL || data2 == NULL || data3 == NULL)return -1;

// Accepts 3rd image

bool GLMultiTexture::addVariablesToProgram

Added

\_textureIdx3 = glGetUniformLocation(program, \_glsl\_names[2].c\_str());

checkUniform(\_textureIdx3, \_glsl\_names[2]);

// glActiveTexture tells OpenGL which texture unit we want to use. GL\_TEXTURE0 is the first texture unit, so we will just use that.

glActiveTexture(GL\_TEXTURE2);

//We use glBindTexture bind our texture into the active texture unit.

glBindTexture(GL\_TEXTURE\_2D, \_texture\_3);

// Then we set the tex uniform of the shaders to the index of the texture unit. We used texture unit zero, so we set the tex uniform to the integer value 0.

glUniform1i(\_textureIdx3, 2);

// Handles 3rd image texturing

Modified

\_textureBlendModelIdx = glGetUniformLocation(program, \_glsl\_names[3].c\_str() );

checkUniform(\_textureBlendModelIdx, \_glsl\_names[3]);

Index changed from 2 to 3

Texture.h

class GLMultiTexture : public GLTextureBase

Added

const string \_glsl\_names[4] = { "texture\_sunset", "texture\_face", "texture\_colorwheel", "texture\_blend" };

// Added parameter for texture\_colorwheel

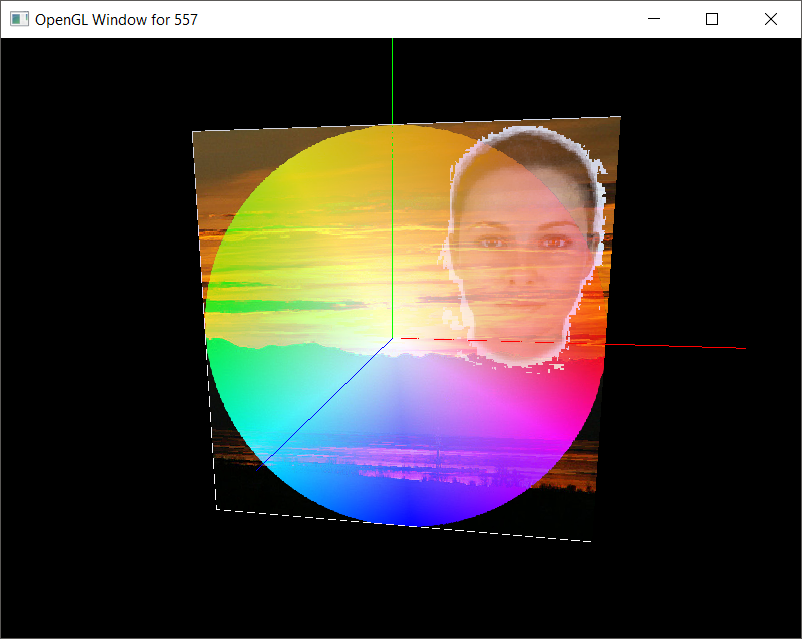
private

Added

GLuint \_texture\_3

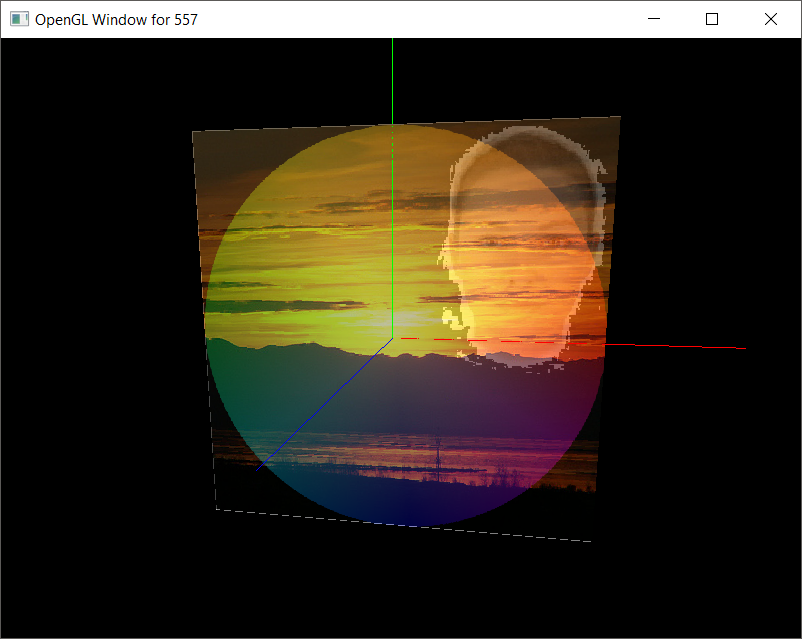
int \_textureIdx3

int \_textureBlendModelIdx2 // Not used



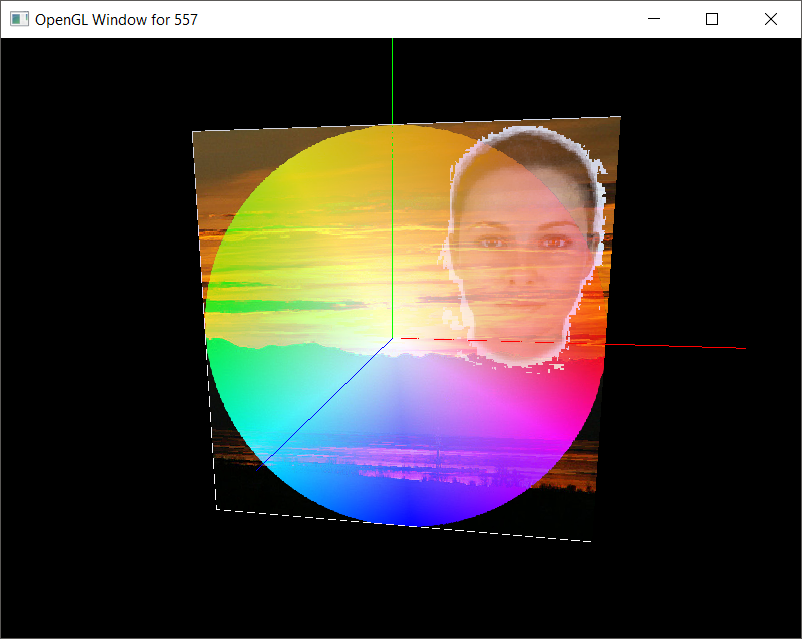
color = (tex\_face + tex\_colorwheel + tex\_sunset) / 3;

This scene is darker than I would like. The face has a black background, the gradient (outside of the circle) is black, along with the mountains and grasses in the sunset photo.



color = (tex\_face + tex\_colorwheel + 2 \* tex\_sunset) / 4;

This has a double weighting on the sunset. It becomes a bit overpowering compared to the gradient and the face. It seems darker than the first image.



color = (tex\_face \* tex\_face + tex\_colorwheel \* tex\_colorwheel + tex\_sunset \* tex\_sunset) / (tex\_face + tex\_colorwheel + tex\_sunset);

This one is a relatively balanced image. Things just look better than the other two images. The gradient is much more vibrant, and the center of the gradient is pretty close to the location of the sun, though the center of the image is not oversaturated as may have happened if we simply added the three images together without any division.