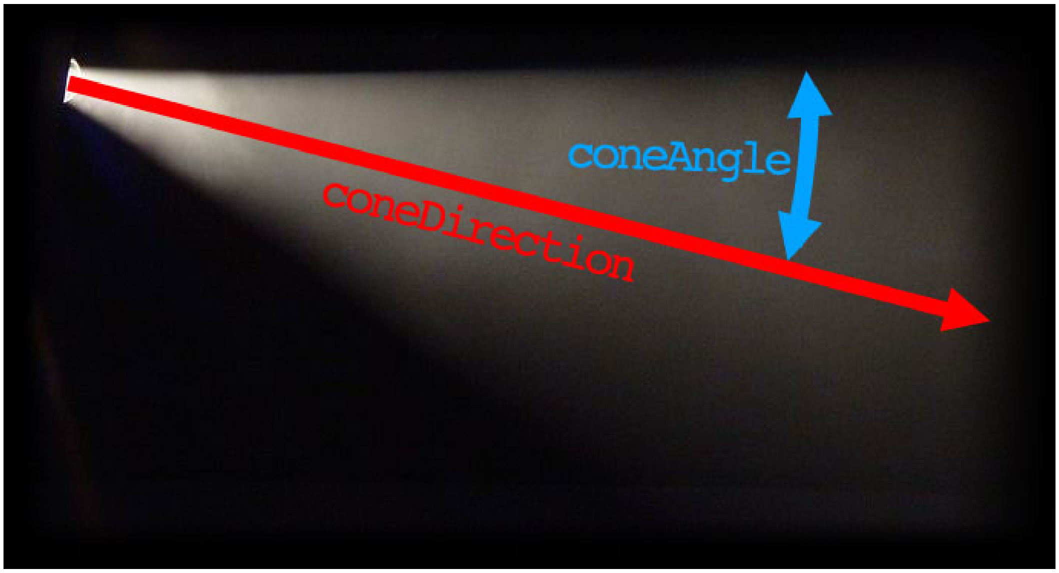
Graphics Final Process:

Brandon L’Abbe and Olli Machina

1. Get directional light source working

Once we have the light source- we can easily begin to create the glass fragments and create the demonstrations and shaders to have the light source shine through the glass pane.

Using a raycast, we can create a raycast from the origin of the light source to the fragment and check if the ray is passed through the glass object and must be changed with the shader.



2. Add glass and transparent effect

The glass needs to have a transparency with the color to simulate the look of stained glass and display a window effect. This should be quite simple by utilizing the alpha value for the object and adjusting as needed to get the intended glass look and effect for the lights.

3. Implement multi-pass rendering:

1: Implement phong shading for the texture applied on the object

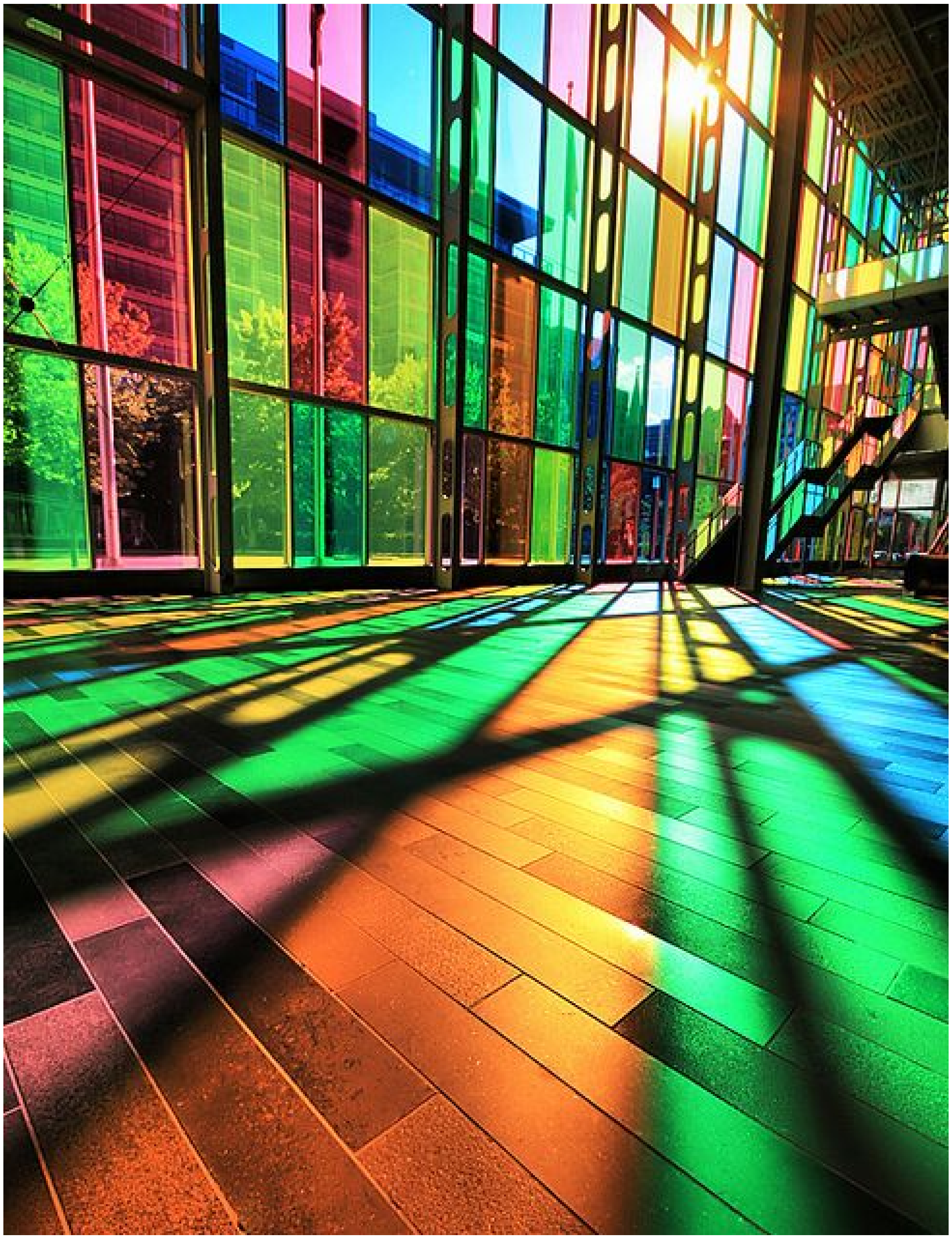
2: additive blending with the lights

To create desired lighting effect and interaction with the textures

4. Implement light movement and animation sequences

A highlighted feature for our project is the ability to move the lights around the scene and display the manipulation of light and color through the shaders. We must create the intended effect from light movement- as well as implement some sort of animation to display a moving effect without user input.

We are able to calculate the movement in our shader- but in order to communicate the data to the lighting shader- we need to use a C# script to actively pass the data from one shader to another.



5. Random generation of glass

As an added feature of displaying the shader capabilities- we can add the detail of glass shards being randomly colored and placed in the scene. It can show the extravagant abilities of color manipulation as well as the versatility of the shader itself.

This can be achieved with a simple C# script that will generate the color and location of the pieces of glass, including random sizes around the scene as well.