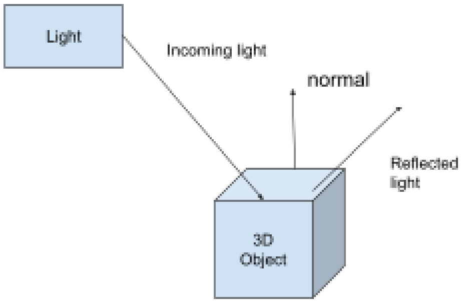
Kishan Mohanakanth 100921637

Intro to Computer Graphics 2025-11-21

COMPUTER GRAPHICS BONUS

Questions:

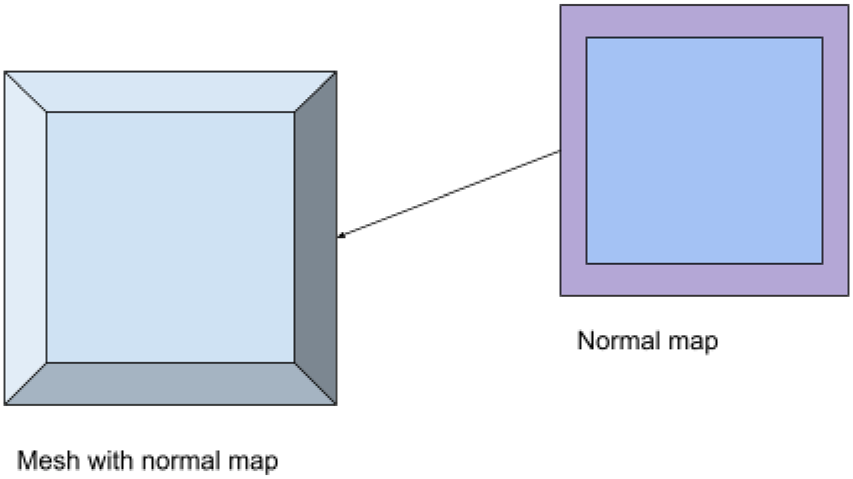
1. Rendering Pipeline fundamentals:
    a. Rendering pipeline is how the shader graphics are applied in the game models/engine from your CPU to your GPU, it's where they can be processed and bring visual effects to enhance the game. Some components are vertex processing, rasterizing, and fragment processing. Vertex processing is how the shader identifies the vertices in the mesh to either apply a transformation or animation. Rasterizing makes the game object more pixelated, it uses 2d pixels to visually enhance the 3d model. Fragment processing is the calculation to get the output of the shader.
    b. CPU -> GPU -> screen
2. Fundamentals of deferred and forward rendering:
    a. Forward rendering is when the object does its own lighting calculations and leads to performance problems. Deferred rendering is when lighting information goes through a single pass to improve performance.
    b. Forward rendering: geometry -> vertex shader -> geometry shader -> fragment shader and lighting -> frame buffer
    c. Deferred rendering: geometry -> vertex shader-> geometry shader -> G-Buffer -> lighting -> frame buffer
3. Fundamentals of vertex and fragment shaders:
    a. Vertex shaders work with identifying the vertices on the game object. Fragment shaders are the OpenGL rendering pipeline, with advanced effects processed on the game object. Some typical applications of vertex shader are real time 3d rendering, animations, and high performance graphics. Fragment shaders can be seen in texture mapping, lighting effects, post processing effects.
4. Lighting model:
    a. A lighting model is the representation of how light intensity is calculated, reflected on the surface of the object.
    b. 

    c. Lambert shader = C x L max(0,N.L)
        i. C = base color, L = Light color or Intensity, N.L = cos angle between surface direction and light direction.
5. Normal and Bump Mapping:
    a. Normal mapping and bump mapping are shaders used to make the 3d mesh's normals dent in or out. Normal mapping shader relies on the object's normal map

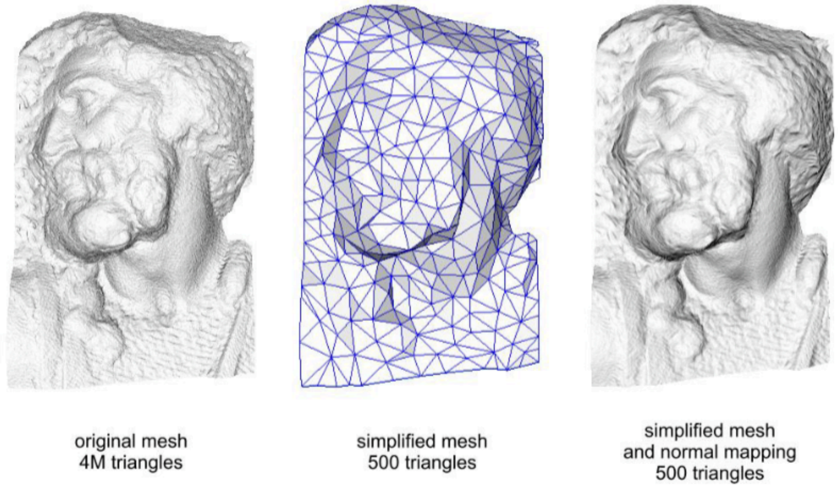
to calculate the number of bumps and dents as the bump mapping shader relies on the grayscale texture to calculate the dents and bumps.

b. Bump map uses information from the texture map to modify a flat or smooth shaded normal. A normal map uses a texture map to retrieve a distinct normal vector.

c.



d.

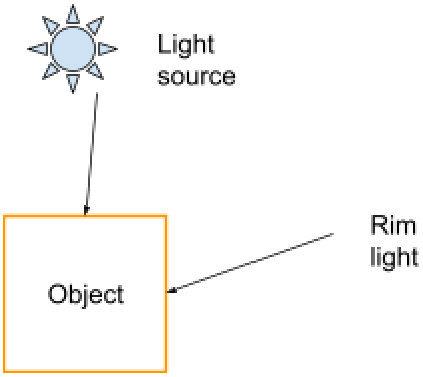


http://theslyd.tumblr.com/post/161020478636/tutorial-normal-bump-map-baking-in-blender

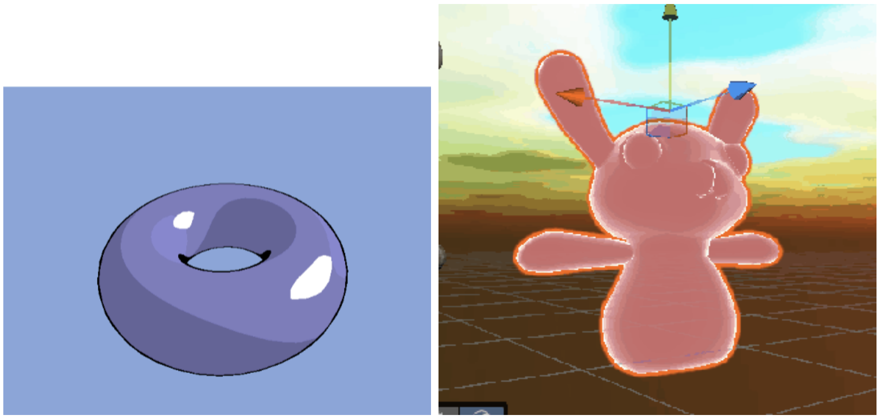
e.

6. Rim lighting and toon shading:

   a. Rim lighting gives the object a highlight or a glow edge around a 3d object. It highlights the outer edges rather than across the surface. Toon shading otherwise known as cel shading creates a cartoon or anime effect on your objects, it makes your 3d objects look like flat 2d surfaces. Renders shadows using flat colour toon bands.
   b. An example of rim lighting can be used for a character standing on a beach sunset, the sun is behind them and its orange colour light goes with the edges of the character mesh. An example of toon shader would be in games like legend of zelda, character dont have smooth shadows only bold colours.
   c.

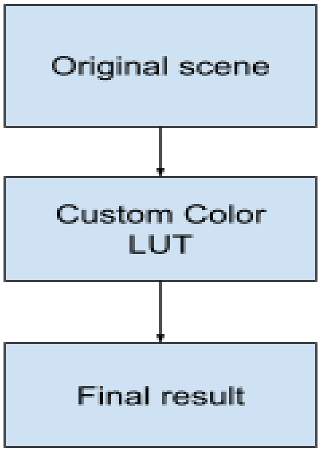


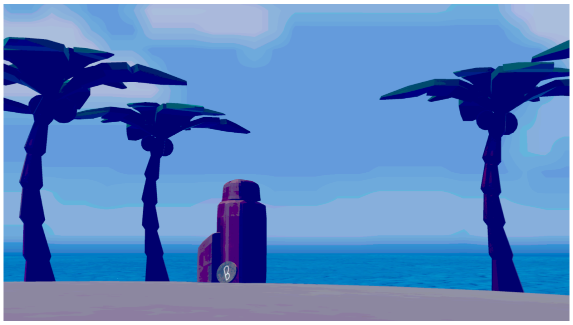
   d.



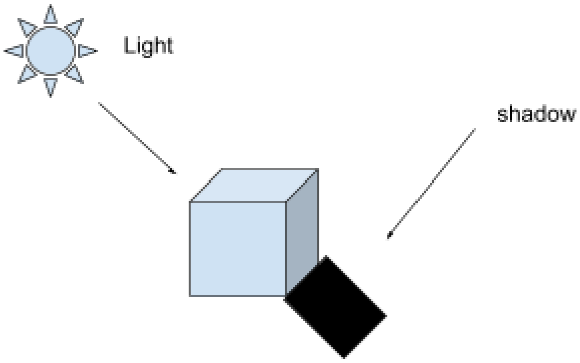
   e. Rim lighting can be useful for vr, stealth, photo mode. Toon shading can be used in low poly games.

7. Color grading:
   a. Color grading is the process of remapping the games color tones by using LUT (look up tables)
   b. You can use a blue color grade to show the night skylight to feel more dramatic or depressing.

c. 

d. 

e. You can use a black and white colour grading in games where you have to find certain objects in a puzzle

8. Visual Effects:
    a. Transparency, calculates how much of the surface can be seen through, determines the opacity of the mesh
    b. Culling, determines which side of the mesh is drawn
    c. Stencil, a mask that controls where something should or should not be rendered.
    d. Hologram, simulates a projected glowing object
    e. They can be used as quest markers, sci fi doors, ghost characters
9. Shadows:
    a. The darkened areas that appear when an object blocks off light from reaching the surface. It gives depth and realism.
    b. Imagine a palm tree blocking off the sunset, a shadow can be casted depending on the angle of the sun.
    c. 

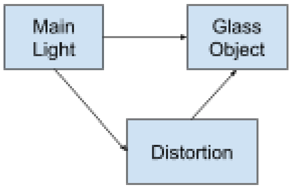
d.

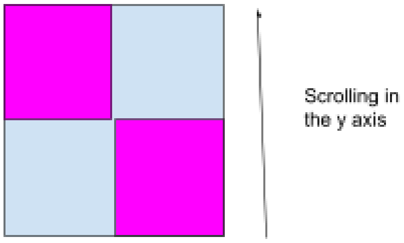
e. Shadows can be used to make a scene in the game more realistic and dramatic like in horror games.

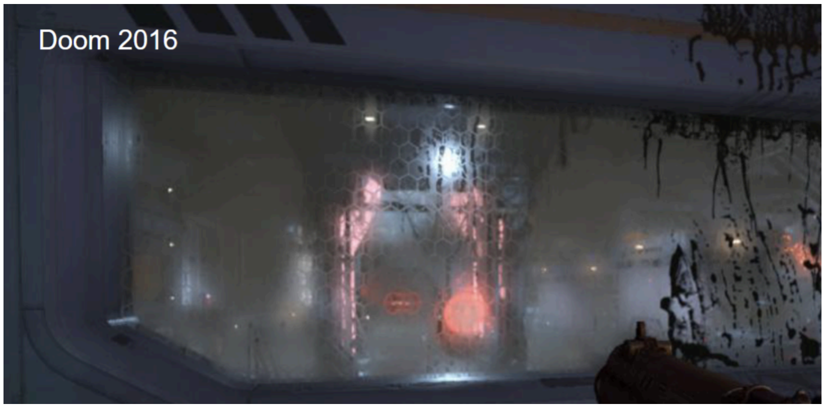
10. Glass and Scrolling shaders:

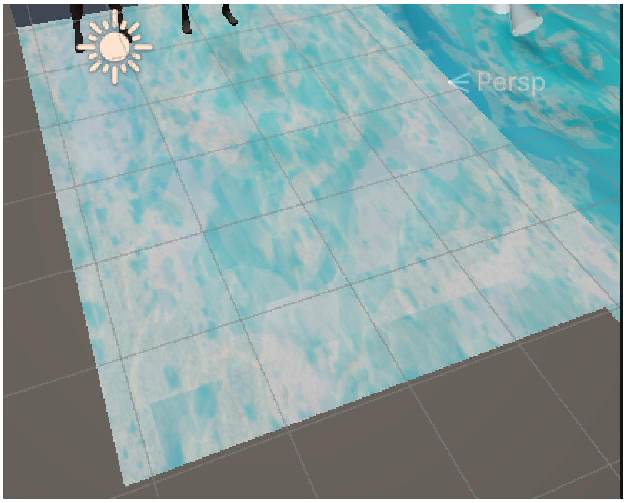
a. A glass shader is to make a transparent or semi transparent object such as windows, mirrors, and water. A scroll shader makes it so that the texture animates along the UV coordinates.

b. An example of glass shader can be a glass block in minecraft, in the game the glass is transparent and you can see through the object. An example of scrolling textures can be the water waves texture moving in the plane.

c. 



d. 



e. Some examples we can see of glass shaders are in portals and ice or crystal objects. For scrolling shaders we can see them in a conveyor object the belt texture can scroll to show that the object is animated.

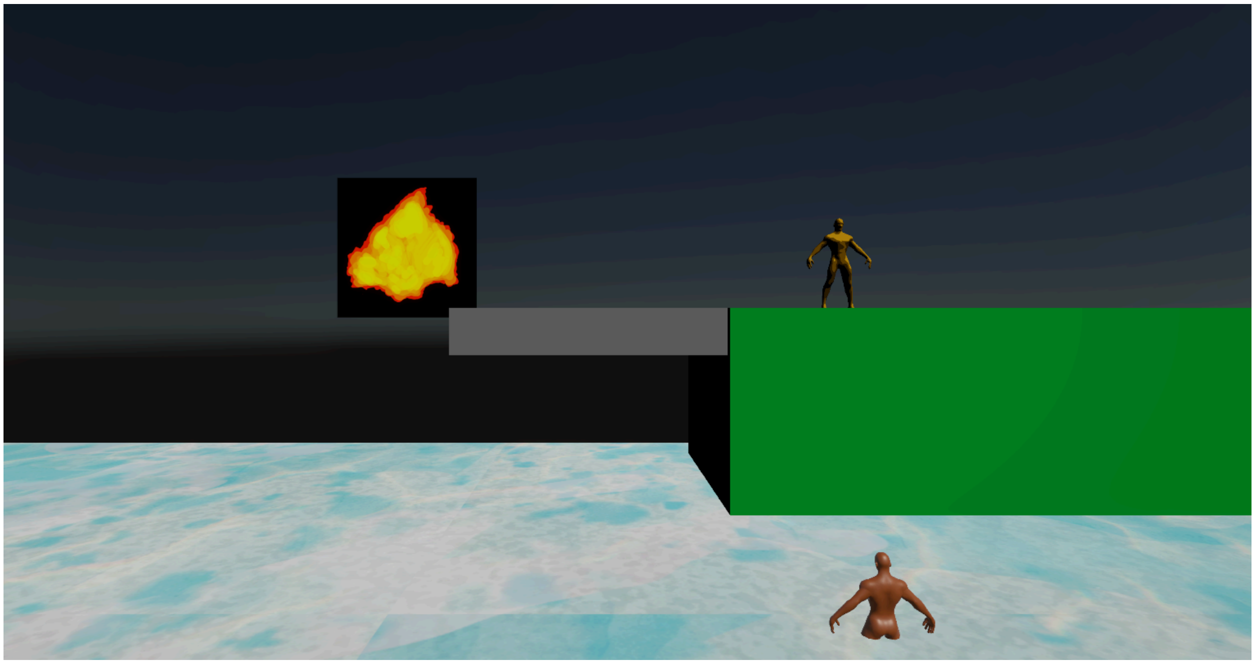
f.

Bonus Review challenge PRACTICAL:

Following 2 shaders implemented: Water scrolling and toon shading

Kishan Mohanakanth 100921637

Intro to Computer Graphics 2025-11-21



All BONUSReflection:

1. Theoretical Reflection:
    a. Overall the theoretical part of this bonus really tested my knowledge of computer graphics, trying to recollect all the definitions and diagrams. One thing that I noticed when doing this part is that I realized that I was on the right track but I was struggling to get the right words to explain and describe the shaders using examples and diagrams. On the other hand I was not confident if I got all their definitions right. So in the future, when I do my final exam for this course I will focus on studying very carefully and making sure I can get all the definitions and diagrams done confidently.
2. Practical Reflection:
    a. Overall the practical part of this bonus tested how fast I can implement shaders in a short amount of time. I challenged myself like if I were sitting in the final exam building the practical requirements. So far I feel like I'm not bad at implementing shaders fast but I feel like I need more practice and to get faster in implementing the rest of the shaders. So in the future when I do the final exam I will focus on practicing implementing all the shaders done in class as fast and comfortable as possible.

COURSE FEEDBACK:

Kishan Mohanakanth 100921637

Intro to Computer Graphics 2025-11-21

Ontario Tech University - Student Feedback Questionnaire

Your responses were successfully submitted.

At Ontario Tech University, we strive to deliver a great education to each of our students in every one of our courses. There are many elements that contribute to improving the quality of a

Ontario Tech University - Student Feedback Questionnaire

Your responses were successfully submitted.