COSC 342 Assignment 3 – Render Engine

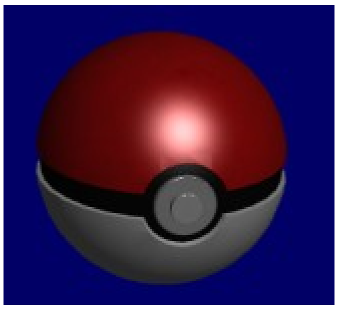
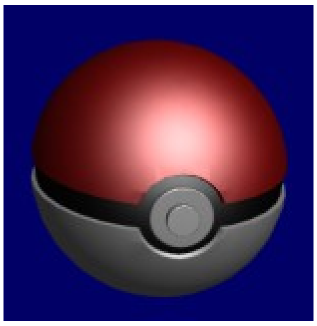
Report for Beckham Wilson ID #7564267

Functionality Added:

- Passed material parameters from the MTL file to shaders.
- Implemented Phong shading and Blinn-Phong reflection model.
- Implemented support for transparency.
- Implemented a dynamic “glass” effect via Render-to-Texture (RTT).

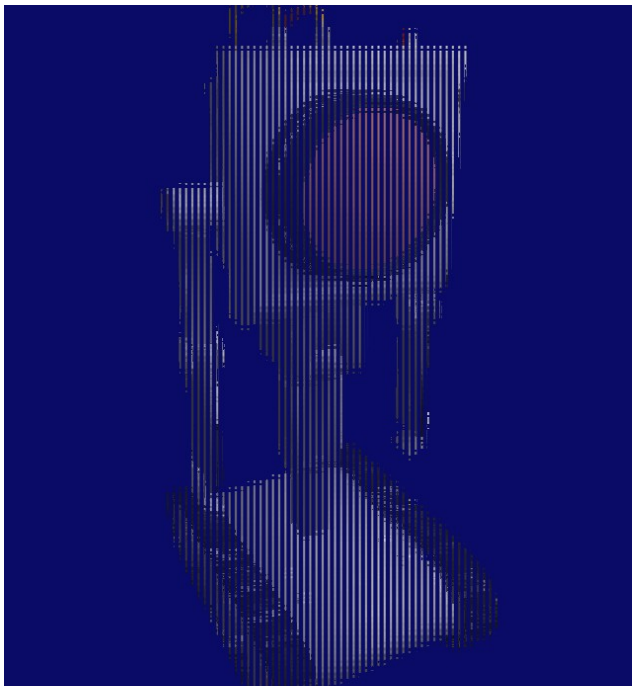
Phong vs Blinn-Phong reflection model:

When the shininess property (Ns) is low, there is a large specular area of light, making the sharp cutoff more apparent. In the Phong reflection model, the sharp cutoff between light and dark areas occurs because the dot product between the reflection vector and camera vector may be over 90 degrees, resulting in a specular component of 0 (since negative values are clamped to 0). However, in Blinn-Phong reflection model, by using the dot product between the halfway vector and the normal vector, we never exceed 90 degrees, thus preventing the sharp cutoff.

Phong reflection model	Blinn-Phong reflection model
	

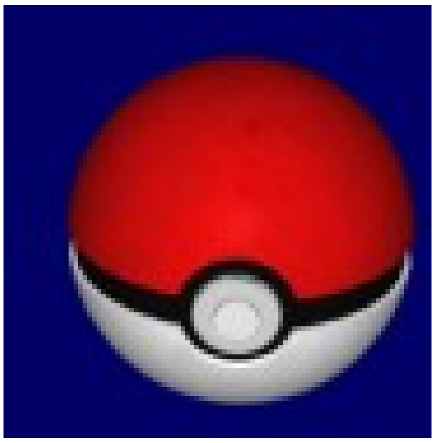
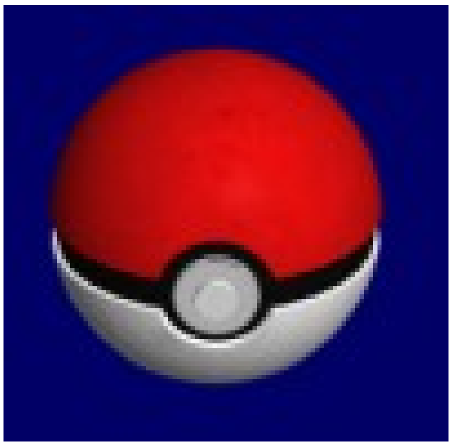
Transparency & Dynamic “glass” effect:

Transparency/Blending	Dynamic “glass” effect

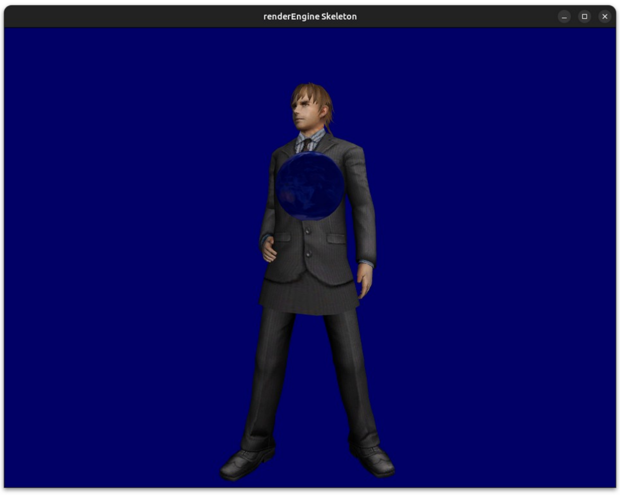
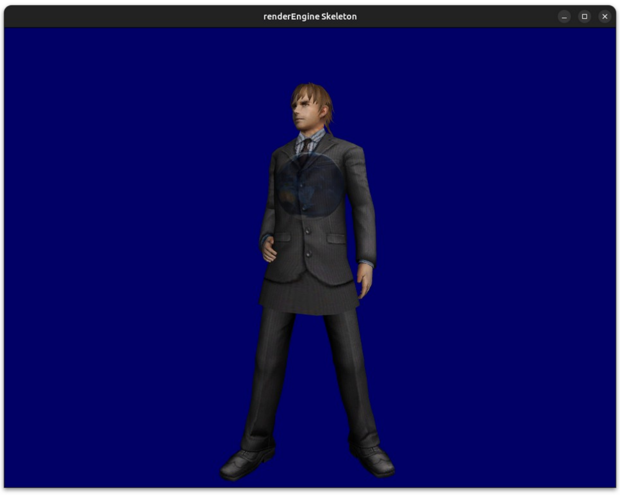
Bugs Fixed:

- The light source was previously inside the models, but now it has been moved outside the model, positioned above it to simulate ceiling lights or natural sunlight.

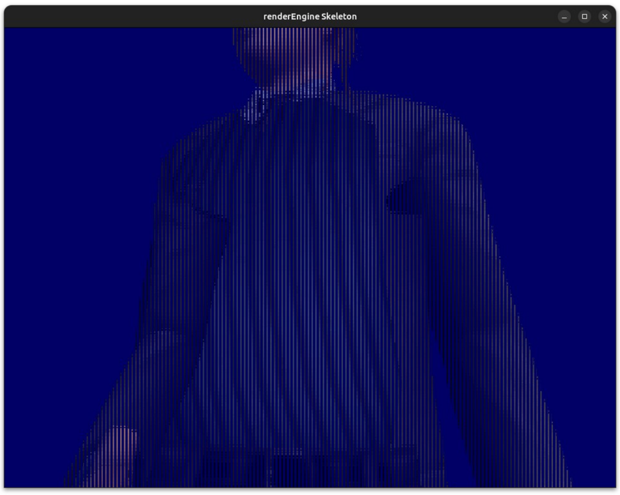
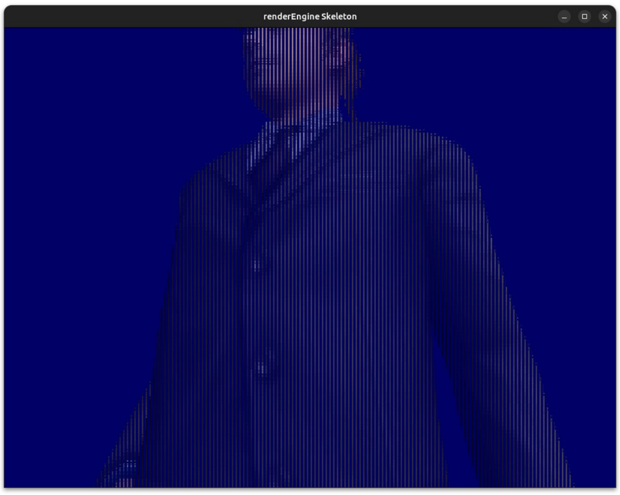
Before	After
	

- Originally, objects were drawn in the wrong order, preventing blending from working correctly. Now, by drawing objects from farthest to nearest, this issue is resolved, and blending and transparency work correctly.

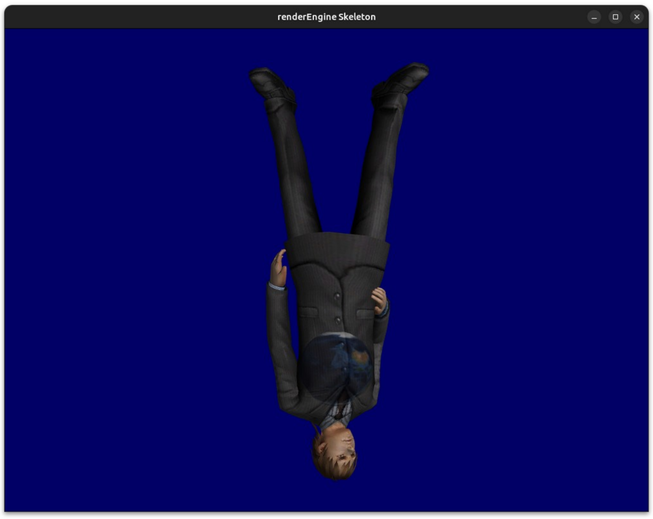
Before	After

- Artifacts due to lack of depth testing in the Render-to-Texture implementation were resolved by enabling depth testing in OpenGL through the creation of a render buffer containing the depth/stencil information.

Before	After
	

- If you rotate the camera excessively vertically (i.e., move the mouse too far up or down), the camera might flip upside down. To fix this, we check if the camera's "up" vector in camera space has a negative y-component, indicating it's upside down. If it is, we reverse the update to the camera vectors to ensure the "up" vector's y-component remains positive.



Testing:

I tested my program and ensured it ran correctly by keeping a series of reference images. After each change, I manually compared the new outputs to these references to check for any discrepancies, as shown in the example reference images above. Tested 4 different models, and even with all four models in the same scene still displays 16.666667 ms/frame, which is real time since its 60 fps. The only noticeable drop in performance only occurred in the first second which with 4 models only dropped to ~13ms/frame. Also didn't notice any impact on rendering performance with post-processing/special effects, illumination and polygon count/filesize.

File Formats Supported:

- obj/mtl
- stl (no texture, low polygons)
- ply (no texture)

File Formats Not Supported:

- x3d
- fbx
- gltf/glb

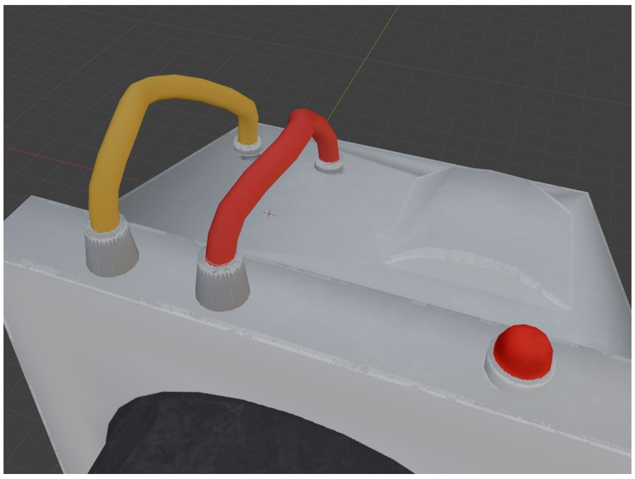
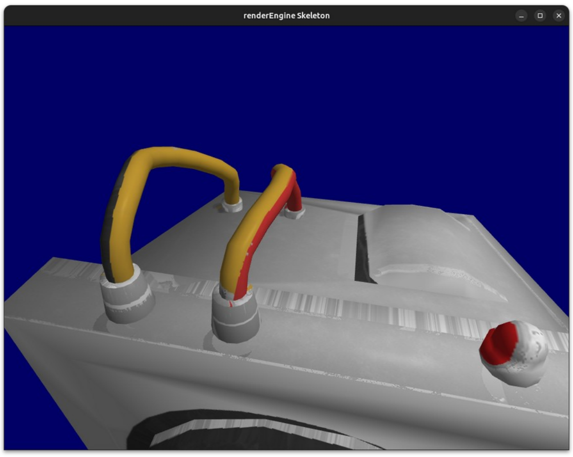
Image Texture Formats Supported:

- dds
- 24 bit bmp

Image Texture Formats Not Supported:

- png

Flaws:

Blender	Render Engine
	

- Noticeable mismatch for texture mapping when trying to use imported models might be related to the scale of the models.
- The program crashes/freezes if passing invalid paths into assimp and freezes the whole desktop GUI.
- The 3D models require a large amount of tweaking to scale model to fit the render engine.
- 3D file formats with embedded textures causes the textures to not display. Either assimp doesn't supported embedded textures or because Texture.hpp/.cpp only supports these external image textures via dds or bmp image formats.
- When sorting objects to fix the blending issues, I used the first point of the object. Instead, I should use the center of the model or the point of the object closest to the camera.

References:

Models:

- passthebutter.obj by klikker228666 https://sketchfab.com/3d-models/butter-bot-3bd1b48baa5846d19b564394277ee674, no changes were made. Licensed under CC BY 4.0 Deed.
- pokeball.obj by JMGameDev https://sketchfab.com/3d-models/pokemon-basic-pokeball-9b29539199c14ddea4de7776c4d758df, no changes were made. Licensed under CC BY 4.0 Deed.

Code snippets:

- https://learnopengl.com/Advanced-OpenGL/Framebuffers used render buffer snippet to enable depth testing for Render-to-Texture implementation
- https://learnopengl.com/Advanced-OpenGL/Blending used concept of sorting the objects when rendering to fix transparency issues.