7-1 Final Project: Reflection

Andrew Wilson

Computer Science Department, Southern New Hampshire University

CS-330: Computational Graphics and Visualization

Dr. Ali Shaykhian

7-1 Final Project: Reflection

# Justification of Development Choices for 3D Scene

The objects in this rendering were chosen to closely match the original photograph. Based on the palette of available simple shapes, I used planes, boxes, cylinders, and a tapered cylinder. For example, planes were used for the floor and book covers and images. Boxes were used to give depth to the book covers themselves, for the aquarium pieces, and for the block of toy hay sitting atop the bowl. The tambourine was made of a large central cylinder circumscribed by six smaller cylinders to form the rattles. Finally, the bowl was made using a tapered cylinder.

Lighting was included by way of the Phong model, incorporating ambient, diffuse, and specular lighting. Two lights are included in the scene: a soft white light a good deal above the items, and an orangish light that is behind the camera and to the left. These two lights serve to illuminate the scene and prevent dark regions. One area that would warrant further tweaking is mitigating the “washed-out” look of the book cover textures. Additionally, each object was assigned a “material” type to help it interact with lighting in a more realistic way. The book covers are given a glassy look due to their glossy nature, while the pages themselves have a more matte surface.

# Navigation

The scene is navigable by way of mouse and keyboard controls. Specifically, the keyboard “WASD” keys are used to move the camera forward, left, backward, and right (respectively) relative to the front-facing direction of the camera. Additionally, the “Q” and “E” keys move the camera down and up (respectively), also relative to the camera’s direction.

The camera’s direction is controlled by the mouse position relative to the render window. As the mouse is moved from the center of the render window towards the top and left, the camera begins looking further up and to the left—and vice versa for opposite directions.

Finally, the mouse wheel is used to adjust the “speed” of these movements: scrolling “up” gradually increases the speed, and scrolling “down” gradually decreases it.

# Custom Functions

I developed several custom functions for modularity and ease of working. Most significantly, I incorporated an ImGui-based UI for adjusting the positions, rotations, and scalings of objects live during rendering. It also had the functionality of saving a C++ code block to save the positions and rotations and scales back into the source code.

I modified the vertex shader code to allow overlaying multiple textures atop each other, while maintaining the various lighting effects already present. I also refactored and abstracted the code for rending and transforming objects for increased modularity. These adjustments served to simplify the code as it grew in length and complexity.