

Project Reflection

Kyle Dale

SNHU: CS-330-Comp Graphic and Visualization

Professor Enkema

June 18, 2023

Justify development choices for your 3D scene.

For the 3D scene I developed, I originally chose these objects because they presented a variety of different shapes, textures, and material properties to adjust and tamper with. I thought that, although challenging, these objects would provide me with ample opportunity to exercise the skills learned during the course, as well as create a scene I personally enjoyed.

I made certain design choices throughout development to better allow me to represent the objects from my original image. This included moving their position in the scene respective to the image, generally making them larger in perspective to the viewer or camera, maintaining a dimmer atmosphere to imitate an exhibit, as well as adding texture to and keeping the pyramid.

When concerning the real-world positioning of the objects, I have them arranged very closely together at the front of the shelf so they can be seen. The idea is that somebody would see something they are interested in, then pick it up to observe it or inquire about it. If they were sunken back further, the objects would not be visible unless a person walked directly up to the shelf. For my 3D scene, I thought each object benefited from an enlarged size and isolated position. Similar to my logic for the real-world positioning, I found this to be the best way to display and highlight each specific object, rather than cram them together like in the image.

I also chose to maintain a more dim atmosphere to mimic the setting of an art gallery or showpiece museum. As each of these objects is personal to me, it was

entertaining to be able to highlight and display them in such a way. I also think the ambience better sets the tone for the user, as to what the purpose and scene is, and should feel like.

Finally, my decision to keep the pyramid from our assignments, and apply more appropriate texture and material properties, was motivated by the objects in my original image, and what I foresaw as feasible to create. When we took those images in week one, I had no experience with object rendering or creating models of any kind. As such, I had not yet grasped the complexity of some of the objects in my image, or the complete scope of the assignment. Had I better understood or informed myself, I would have rearranged my shelves and included a few more of my simple shaped objects, in lieu of the terracotta soldier or gemstone penguin. As it stands, I believe I did a fair job of representing the clay pot, the llama, the dice, and the 9/11 memorial. However, this number of objects left an open space on my plane that I believe made the scene look awry. As such, I chose to include the pyramid, and model the texture based on an actual mini-pyramid displayed on the shelf above these objects.

Ultimately, I believe the scene does an accurate job representing the objects from my original image, with some artistic license employed to make the scene more appealing in the setting of a 3D scene.

Explain how a user can navigate your 3D scene.

The user can currently navigate the scene using a mouse and keyboard, specifically w,s,a,d for movement, and q,e for height changes. 'W' will move the user

forward in the scene, 'S' will move the user backward, 'A' will move the user to the left, while 'D' will move the user to the right; 'Q' will move the user upward, or the positive y direction and 'E' will move the user downward in the negative y direction. The mouse allows the user to freely view the scene from the first-person perspective, and 'P' allows the user to adopt an orthographic perspective.

Were I to expand this program and add support for additional devices, I would likely begin with support for the various game controllers on the market. The basic design of each is similar, so the variables needing to be changed would be minimal between the different controllers. Ultimately, I would stick to the same controller scheme that most every gamer is familiar with. The left joystick would control the w,s,a,d or movement function of the program, that is to say, forward, backward, left, and right, whereas the right joystick would substitute for the mouse control, allowing the user to manipulate the camera and view the scene.

When concerning the extended features, this would depend on what the program was intended for. If I were to adapt it into a game, perhaps I could lock the user in the y axis, and only allow certain movement in the positive or negative y depending on position in the scene such as falling or jumping. Ultimately, the design beyond the joysticks would depend on the purpose and intended functionality of the program.

Explain the custom functions in your program that you are using to make your code more modular and organized.

Unfortunately at the beginning of developing my program, I was inexperienced in this realm of programming and did not quite grasp the breadth our program could encompass. As such, I did not separate the functionality of the camera controls, movement controls, vertex shader, or fragment shader into separate .h files or otherwise. I similarly did not condense the generation of objects into a single function and apply transformations. This has caused my program to become one massive file containing the entirety of the code. Although this is not ideal, in order to meet the requirements of the assignment, I adopted a ‘don’t fix what’s not broken’ mentality.

Although I would need to rework the program for continued development, the program as it stands contains fantastic bones, and is adequately labeled for ease of navigation and understanding. Were I to remake the program from the ground-up, I could simply port the various functionalities into separate files, and reduce the clutter. Additionally, I was not wholly successful in applying transformations initially, and opted to brute-force the vertices for this assignment, because I knew I could make it work. This was less time-efficient and is much less repeatable, but the vertex data is there and can be adapted.

With a small amount of additional time and effort, one could simply create one of the feet for the llama, for example, and apply a transformation to recreate all four in the proper position upon rendering, rather than specify this in the vertices. Similarly, rather than define a function for each cylinder or sphere, I could define one and apply transformations upon rendering to set them in the proper position. Although my current approach was less efficient, it forced me to understand many of the principles about OpenGL object rendering in order to develop a successful program. Were I to rebuild my program, which I assuredly would before further

development, I now have the knowledge, experience, and more than adequate base code to properly condense and optimize this program.