1. **Justify development choices for your 3D scene**.

I chose to make a scene where someone is about to paint with watercolors. For this scene, I chose to make a brush, a sketchbook, a water bowl, and a few watercolor half pans. This is everything someone needs to paint. The sketchbook is slightly rotated because the rings are propping it up. The three half pans are strewn about near the sketchbook, because the artist decided to not organize them well. The brush is lined up neatly next to the sketchbook, and the water cup is sitting behind it.

The most complex items are the water bowl and the sketchbook. It was difficult sizing both of the toruses to match the tapered cylinder for the water bowl, and I had to keep guessing and running the program until it was perfect. The sketchbook is the most complex in the number of shapes- there are four instead of three, and the paper (box) had to be rotated to accommodate the toruses.

One simplification decision I made was reducing the number of toruses involved in the rings for the sketchbook. The photo I used had two smaller rings for each hole, but I did not think it was necessary in this simple art style.

I used a paper texture for the notebook, a rough texture for the water cup, and two different types of wood textures for the paint brush (I had to substitute in wood instead of fur/hair because I couldn’t find a suitable hair texture with the parameters of the assignment). Everything else is smooth.

1. **Explain how a user can navigate your 3D scene**.

The user can use w to move forward, a to move left, s to move backward, d to move right, q to move up, and e to move down. These movements are relative to the camera angle, which the user can adjust by moving their mouse. They can also use the scroll wheel to adjust the speed of movement. Additionally, they can use p and o to toggle between perspective and orthographic views.

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

I added five custom functions: DrawBrush(), DrawPaintPans(), DrawNotebook(), DrawWaterCup(), and DrawPaper(). These functions break up the RenderScene() function into chunks, so that it’s easier to find what to edit. For example, the DrawNotebook() function draws the sketchbook, and the DrawBrush() function draws the brush. This way, I can just search for the specific object and edit parts of it without scrolling through all of RenderScene. With some adjustments, these could be reusable. I could edit all of the functions to accept float inputs for the x y and z coordinates, and draw the objects wherever they are.

Alternatively, I could use any of the Draw functions in a different OpenGL project, if I imported the same textures in the same way there too. If I was using different textures, I would have to alter the function to accept the texture/material nickname(s) as a parameter. This would not be a difficult adjustment- I would just have to use the parameter when setting them.