Erin Voss and Grace Zhao
CS148 Final Image Write-Up

For our final image, we wanted to have a scene that involved water in some way. We decided to have an infinity pool with rubber ducks looking out over a city at sunset. There are two wine glasses and a bottle of wine on the pool edge.

**Objects:**
We created the wine glasses and the pool in blender. Grace modelled the wine glasses and handled all the texturing. Erin helped make the pool and the body of water by using Blender’s DynamicPaint tool. We found the rosé bottle and the rubber ducks online. We also made a sphere, flipped the normals, and mapped a jpg image on the inside to create an environment for our scene. We positioned all the objects in blender first, so they would be in the correct locations relative to one another when we loaded them into the ray tracer.

**Textures:**
The textures of the wine bottle and the ducks were also found online, but we had to play around with them in blender to get them to texture right on our objects. We then also had to change setting in the code and in the mtl files of the objects to make the textures appear how we wanted them to look in the ray tracer. Ultimately, we wanted the ducks to appear to have a rubber texture and the wine glasses to look glassy. We also mapped an ambient image onto the ducks so that the rubber material seemed more realistic. We also did this for the labels on the rosé bottle. We found the tiles for the pool online and imported them to texture the pool in blender using the normal, displacement, and diffuse maps. We wanted to have an hdr environment map texturing the inside of the sphere (this worked in Blender), however that did not transfer over to work in the ray tracer. When trying to use the hdr in the ray tracer, the image just showed up as black. There was no error given by the raytracer, but doing further testing with print statements, we found the raytracer was not reading the hdr file. So we settled for a jpg version of the image and mapped it onto the sphere’s ambient, so the color is dependent from the lighting..

**Lighting:**
Since the hdr file was not supported in the raytracing framework and so the background is textured using ambient, we had to “fake” our light source from the sun. We did this using a directional light that came from the right towards the faces of the duck so that the right sides of the ducks would be lit up, like it would be in the real scene. We included a very dim area light from behind the camera so that we had more consistent lighting. We did not implement global illumination using photon mapping.

**Code Changes:**
We used Assignment7.cpp as the base for our code. We wrote a helper function to load in objects, so all you have to do it call the function loadObject and pass it the file name, the material object, and the scene. Then, for each object we do not have lots of repeated code.

We changed the code in several ways to get our project to work how we wanted it to. We added a function ComputeAmbient in Material.cpp and BlinnPhongMaterial.cpp that was very similar to
ComputeSpecular and ComputeDiffuse. They were dummy functions in Material.cpp that were actually implemented in BlinnPhongMaterial.cpp. We then added the ComputeAmbient functions return value to what was returned by ComputeNonLightDependentBRDF in Material.cpp for objects that had ambient.

We also changed code in BackwardRenderer.cpp to fix the shadows in our image. We changed the code so the shadow rays would not terminate when they hit a transparent object. If you hit an entirely opaque object in the scene, then we ignore this light ray and don't calculate the BRDF contribution from this light source. However, when you hit a transmissive object, then we continue tracing as if we didn't hit this object.

Lastly, we changed the code in rayTracer.cpp to make our program run faster. We used the std::thread function. We made 8 different threads, each handling a column of the final output. Each column was the width/8 pixels wide.

Collaboration:
We worked together for most of the project. Grace did more of creating the objects and texturing them while Erin did more of the coding changes. However, much of our time spent on the project was together, making changes to adjust our final image until we had something we liked and were happy with.