Description
Our whole scene is centered around a Go Table in a traditional Japanese home setting. We picked a Go table because GPUs, a technological advancement in graphics hardware, enabled DeepMind’s AlphaGo to beat the Go world champion in 2016, so there’s a neat symbolical meaning behind the picture. More so, we find the Japanese home setting elegantly simple and calming to observe.

Models and Textures
The tatami mat, flooring, and back walls were self-created in Maya. The rest of the models, including the Go board, chair, shoji screen, lamp, playing pieces, bowls, and paintings were all found online through TurboSquid and Sketchfab. All of the textures were also found online, primarily through simple Google searches. However, the majority of models did not come with UV maps or applied textures. Thus, we had to create UV maps and apply textures manually in Maya. We also edited most of textures in Maya in order to get the “perfect” look for our scene.

Lighting, Layout, and Camera Perspective
We started with some point lights, though these led to quite sharp shadows. We then proceeded by using an area light, placed to the rear left of the frustum. We noticed that picking bigger sizes for the area light led to smoother shadows and a better ambient lighting effect in general, as most of the items are placed right in front of a wall. Given the camera angle we were working with, it was crucial to not leave any space feeling ‘empty’. The bottom right of the frame, in particular, was not occupied by any objects. To compensate for the lack of objects, we decided to dial up the reflectivity of the wood floor, which created a very nice projection of our objects onto the floor.

Technical Contributions
As suggested on Piazza, the main loop in the Run() method of RayTracer.cpp was quite easy to parallelize. To avoid aliasing, we found that using 32 samples per pixels and a sampling grid of 4 x 4 pixels was enough. We used the same adaptive sampler that was used in assignment 6. To understand exactly the position coordinates and rotation operations for the lights and camera, we had to experiment and tinker with the Maya built-in renderer. This was helpful in understanding whether the area light was pointed in the right direction and if it created enough light.
Work Breakdown

- **John:** UV mapping, texturing, shaders, smoothing in Maya
- **Luigi:** rendering, lighting, parallelization, anti-aliasing
- **Both:** theme and arrangement

(From top left to bottom right, in low resolution) the evolution of our scene

Variant A and Variant B of the Image, respectively