I did not implement any extra improvements to the ray tracer.

I did not attempt to implement bounding volumes.

I did not do Phong shading. However, I did do Phong lighting.

My interesting background fades from blue to red, from bottom-left to top-right.

I provided sample images from running my raytracer on nonhier.lua and simple-cows.lua. These (.png) images are located in the cs488/handin/A4 directory.

I added two classes, and used them extensively in my ray tracer. These are ray\_t and intersection\_t, and are located in ray.h/ray.cpp. Ray\_t represents a firing ray, and includes direction and initial position members. Intersection\_t represents an intersection point made by a ray, and includes the intersection point, the normal at the intersection point, the material at the intersection point, the distance from the intersection point to the ray origin, and a bool representing whether or not the intersection actually occurs (I use this in hierarchical ray tracing).

My overall method for ray tracing includes implementing an intersection\_check function for each of the primitives. This function gets called for each ray sent, as well as for all children in hierarchical ray tracing.

I used the intersection-of-halfspaces method to determine whether a ray was within a polygonal face or cube face boundary.

I implemented a very basic unique scene (sample.lua, output to sample.png, 500x500). This scene is basically a sphere, a box, and a dodeca. The dodeca is translated and scaled, and is a polygonal surface. The dodeca is a “shiny” surface. There are two point light sources, orange and white. Shadows are also visible.