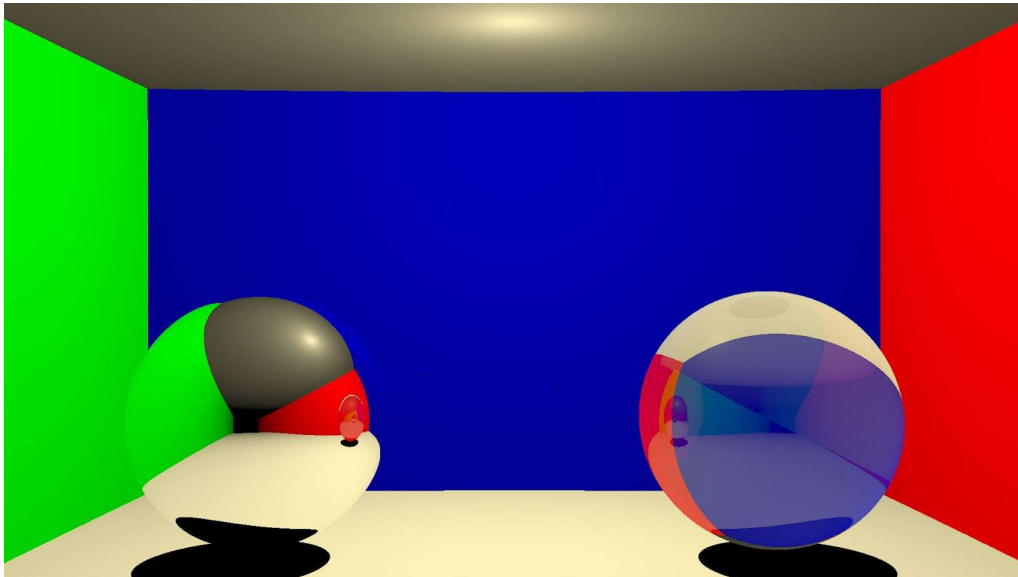


# Ray Tracing - Whitted Ray Tracing

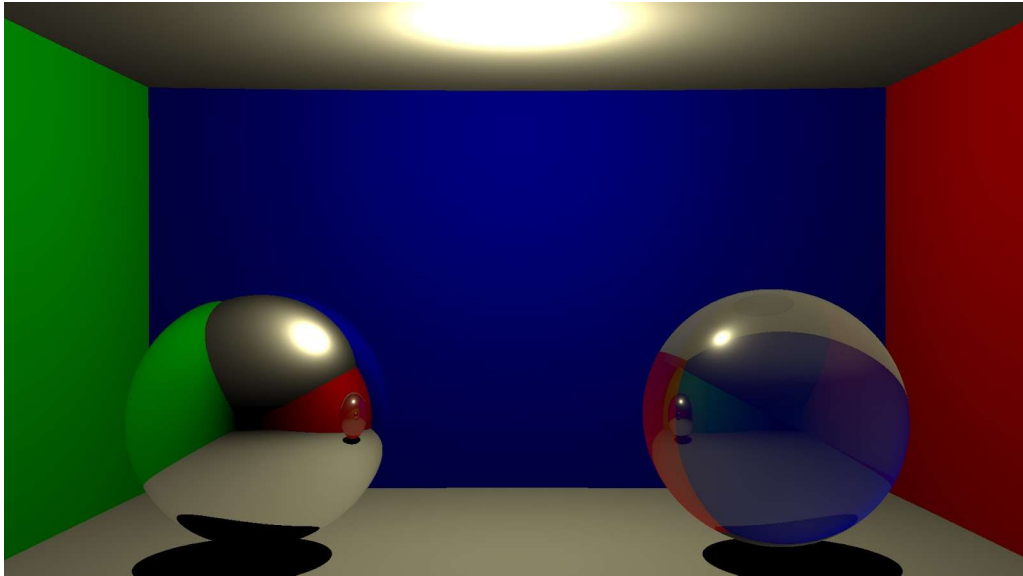
By Alan(200050005) & Rakesh(200050120)

- The ray tracing framework being used is called rt. We are supposed to implement a working version of the whitted ray tracing technique.
- Firstly, we completed the implementation of ***pointlight\_t***, in the ***direct*** function, diffuse and ambient lighting.
- Checking if ***hitpt*** was in shadow was done by checking intersection of a ray from ***hitpt*** to light position.



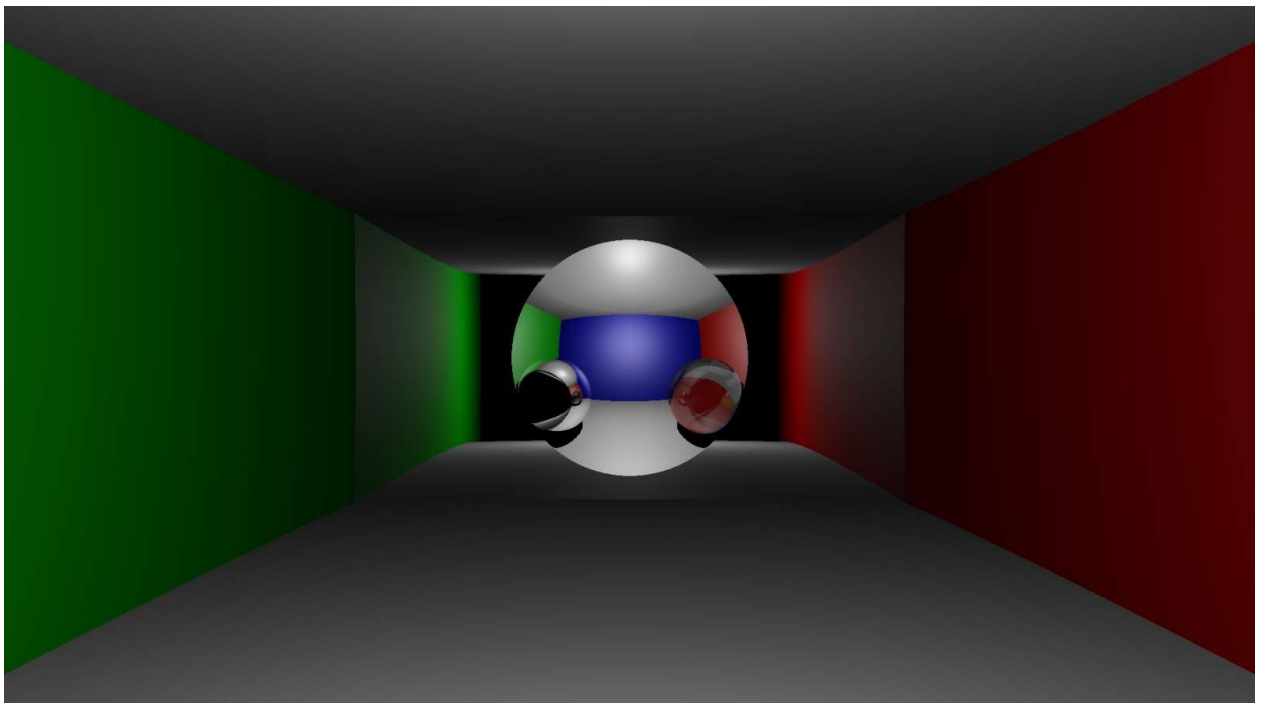
- To the ***radiance*** in ***whitted\_integrator\_t*** we added the recursion to decide how many times each ray can bounce around in the scene, specifically secondary reflected rays and transmitted rays.
- If the material is neither transmissive nor reflective we calculate the direct illumination.
- If it is transmissive we create a new transmitted ray through the ***ray\_t::transmitted*** function and recursively call ***whitted\_integrator\_t::radiance*** with the new ray.
- Similarly for reflected rays we use the ***ray\_t::reflected*** function.
- The ***sphere\_t::intersect*** function was edited to support rays that originate inside the sphere

- We created the simple version of a Cornell room scene using the sphere objects from the smallpt ray tracer.

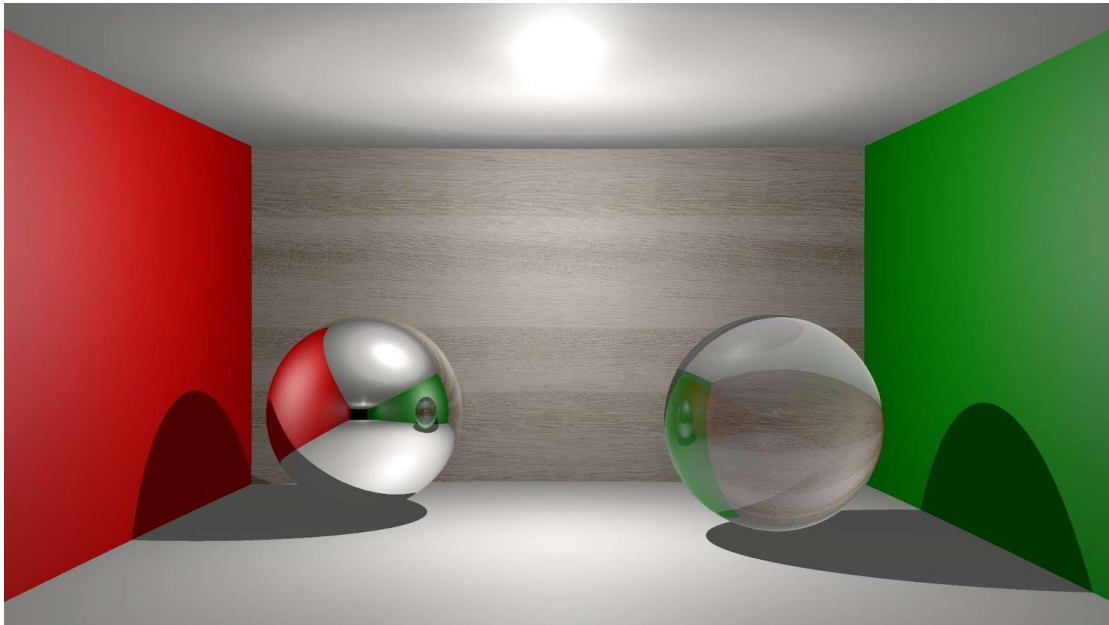


- In ***point\_light\_t::direct*** a specular component using the blinn-phong model was added.
- The ***ray\_t::transmitted*** function returns a reflected ray if Total Internal Reflection occurs.

In the below scene, the camera is inside a huge mirror with a very high coefficient of refraction. We can only see the inside of the room through a small circle where TIR does not occur.

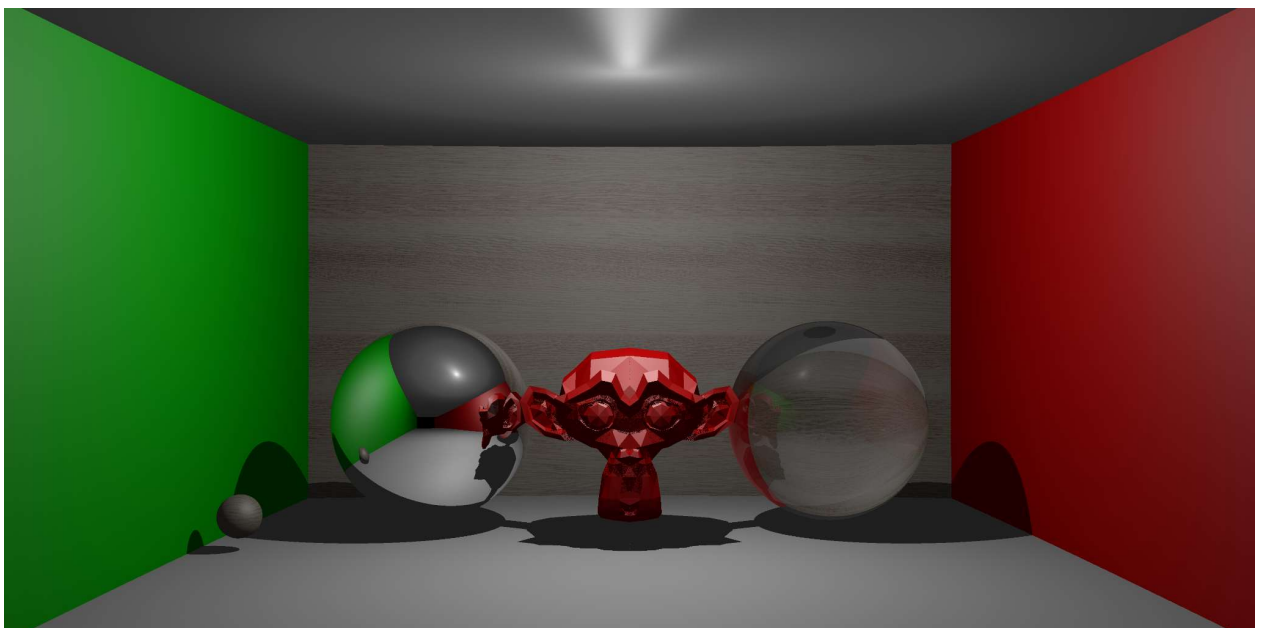


- We worked on the Bonus 1 of the assignment and added a neat little texture to the back wall as shown in the image.
- Cube Mapping is done for the spheres, where a 2d texture is used for each of the 6 faces of the cube. This texture can be chosen using the xml scene file.



- We also worked on Bonus 2 and added support for obj meshes by adding a ***mesh\_t*** class that derives from ***object\_t***.
- Ray Intersection is calculated by looping through each of the triangles in the mesh.

There is some noise due to precision issues in these calculations for meshes.



Instructions to run the code are:

The MakeFile creates an executable file which when run with a .xml scene renders an image of the scene.

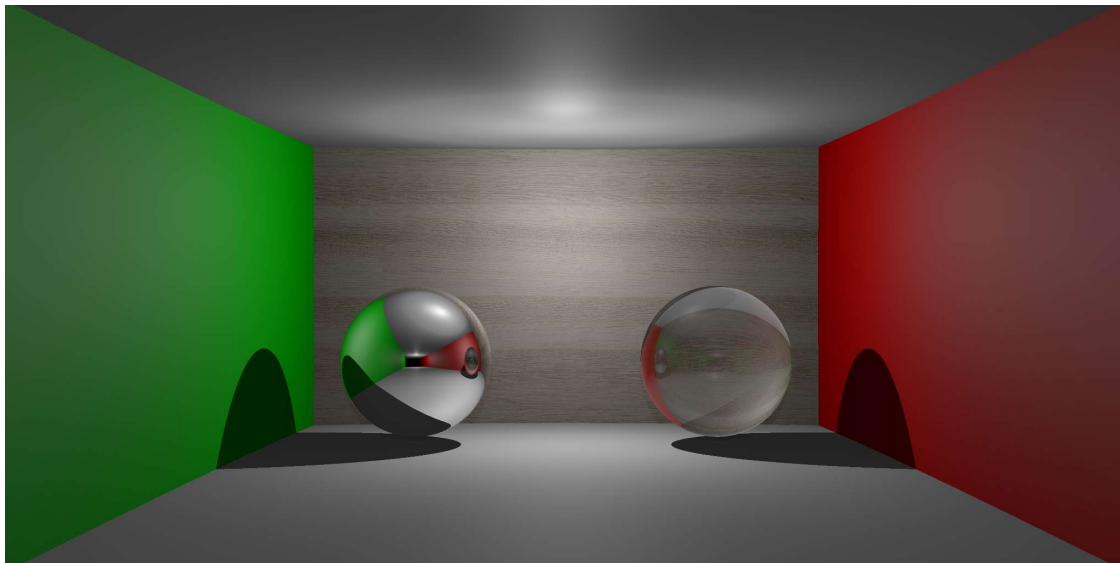
- make
- ./rt <path\_to\_xml\_file>

The Doxygen documentation can be run with:

- doxygen <configure\_file>

The corresponding documentation is present when you run the “index.html” file.

### **cornellscene.xml**



### **myscene.xml**

