



# REPORT

Project 2, Ray Tracing

CAP 5705

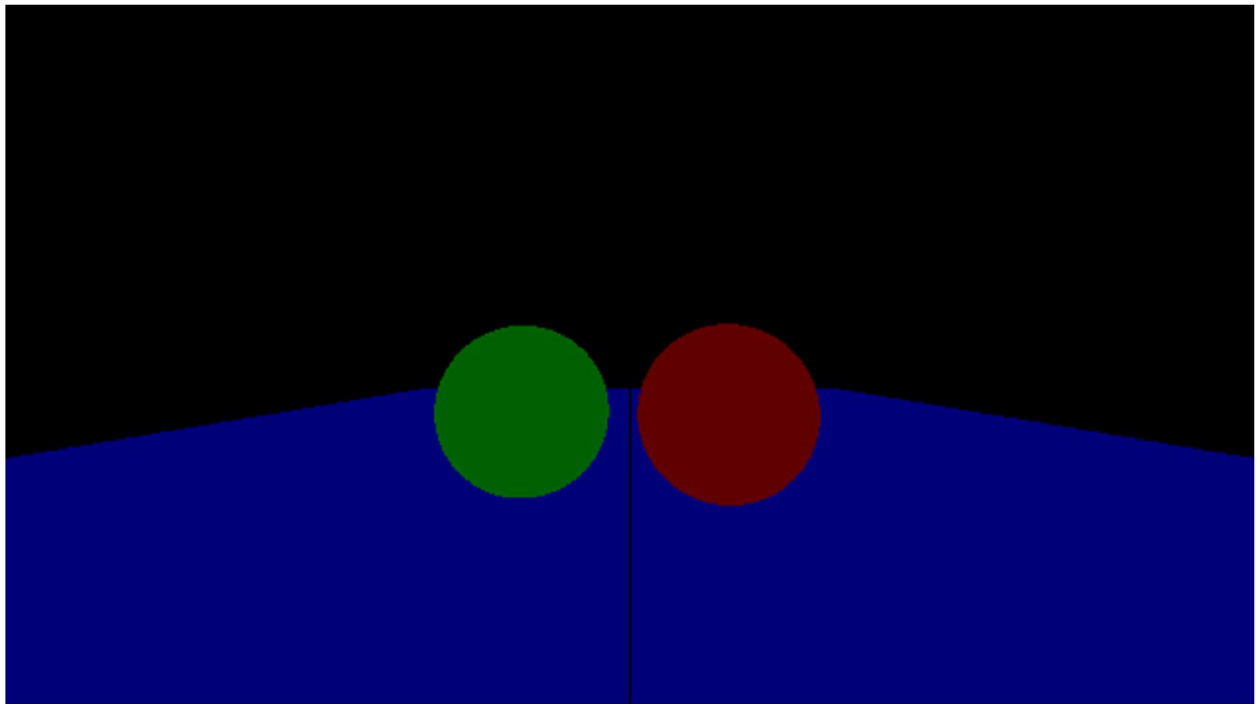
Chawla, Abhishek  
abhichawla@ufl.edu

#### A. Setup:

The scene consists of a camera, 3 objects (2 spheres and a plane made of two triangles), and a projection plane lying between the camera and the objects. Later on, two lights are also added to implement shading.

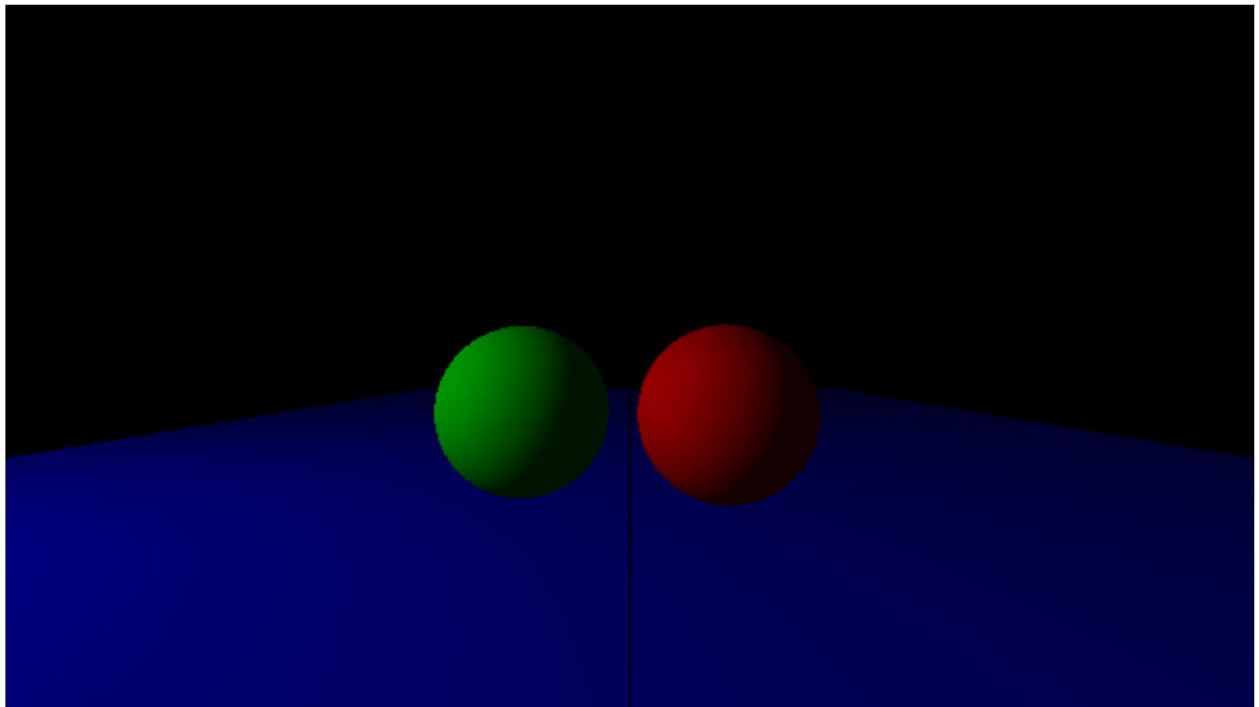
#### B. Intersections in scene:

The ray travels through the projection plane towards the objects, and if an object is intersected, the corresponding pixel on the projection plane is colored by the object's diffuse color. The closest intersection points are stored.

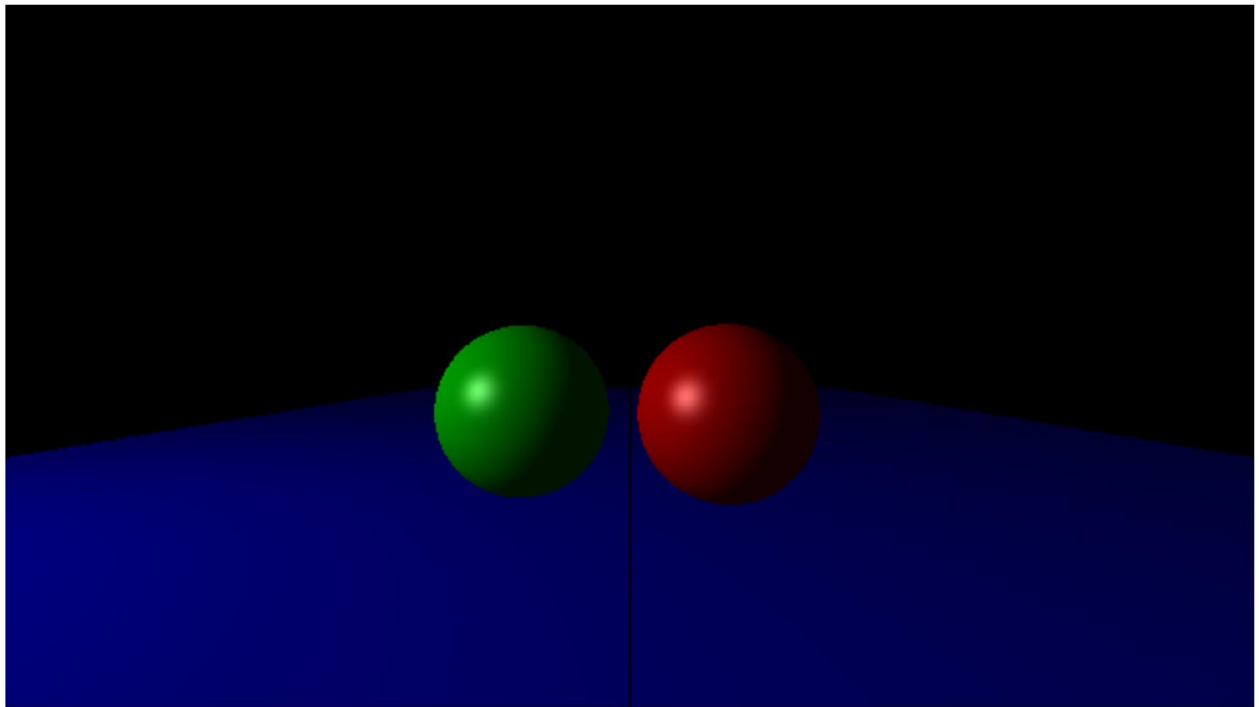


C. Shading:

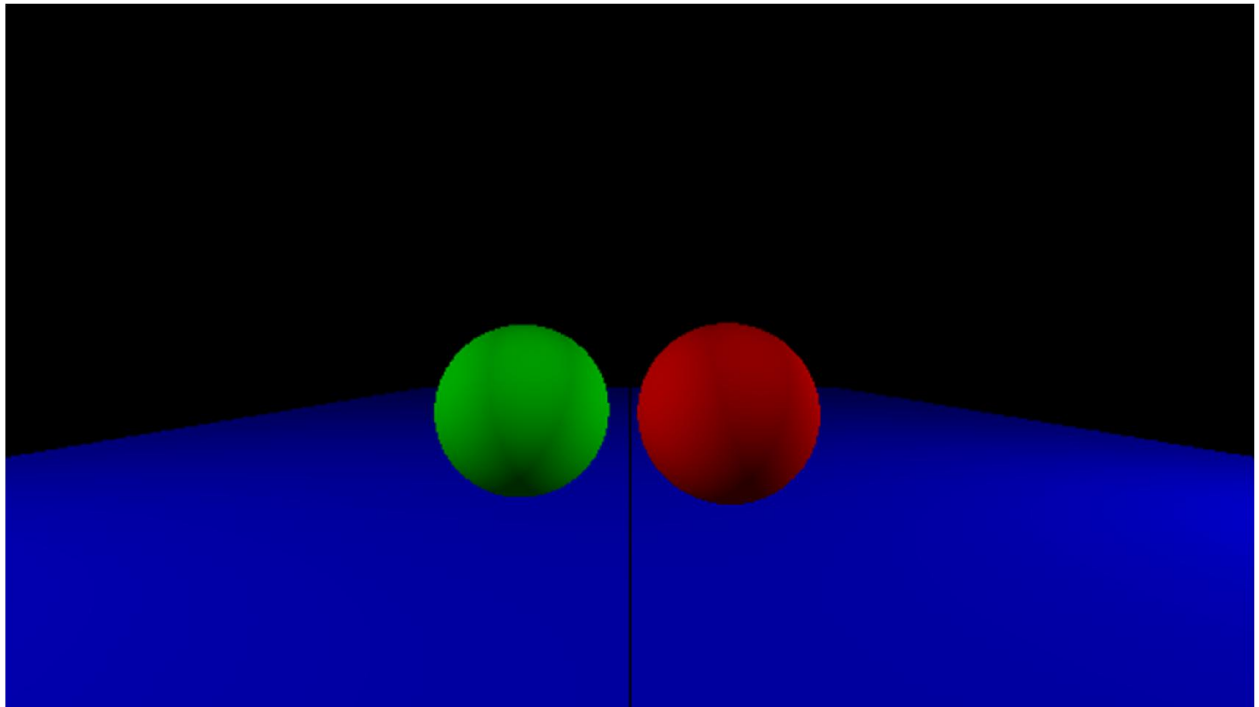
A light is added to the scene. Below is a picture showing Lambertian shading.



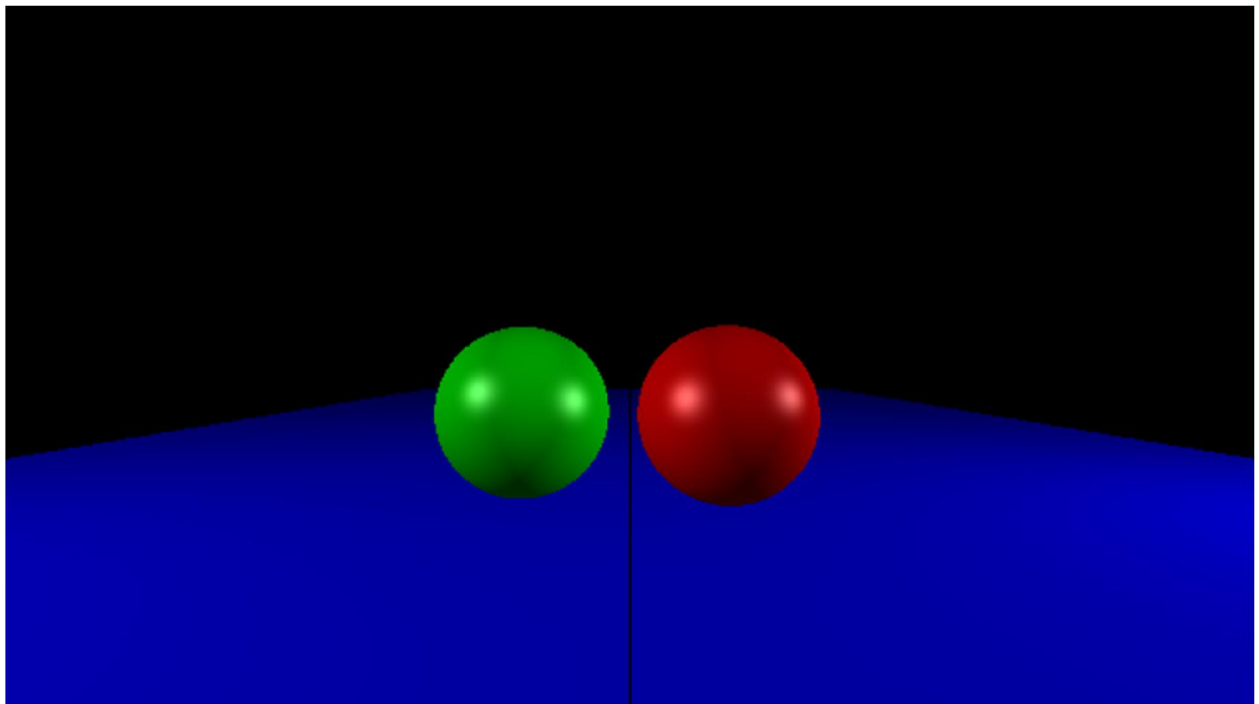
The Phong Shading with one light.



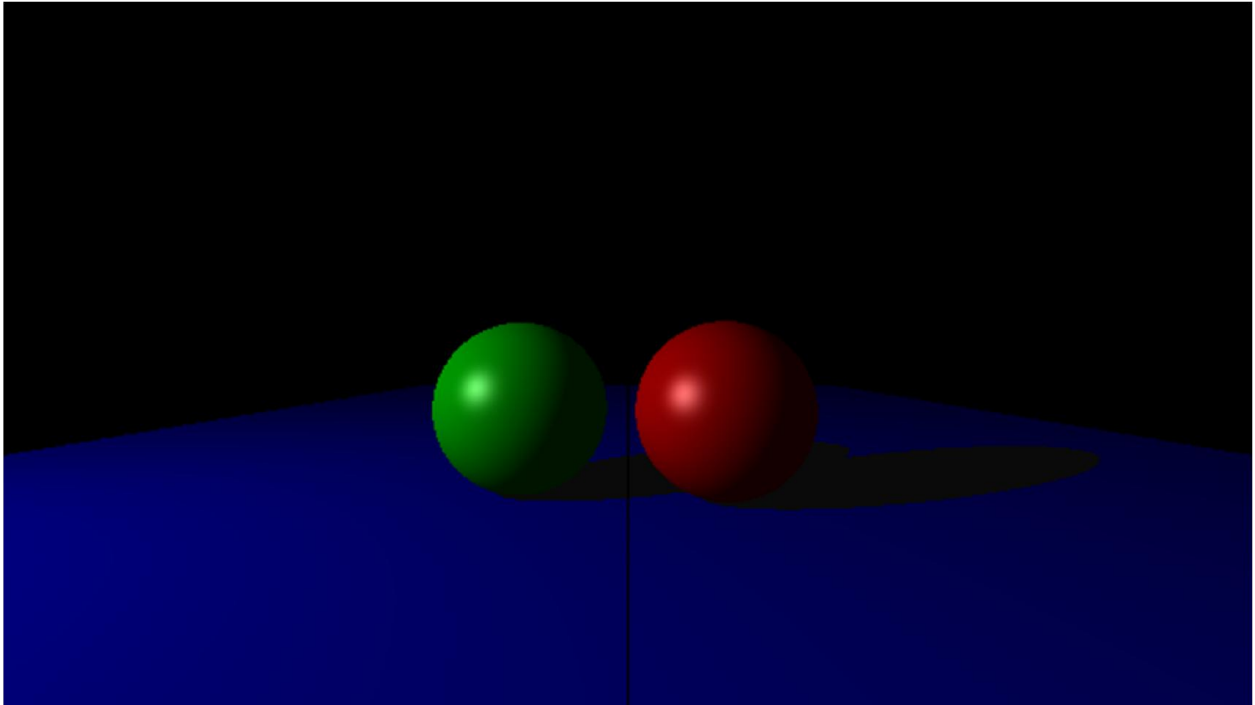
Another light is added. Below is Lambertian shading with two lights.



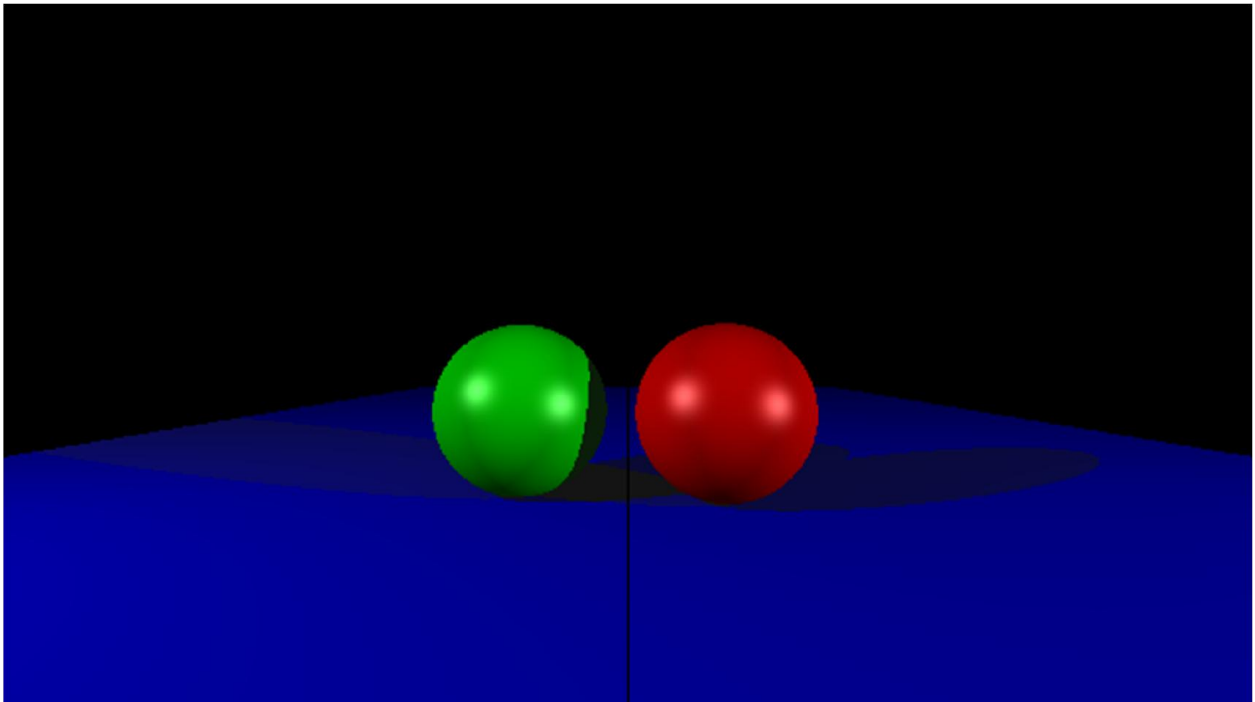
Phong Shading with two lights.



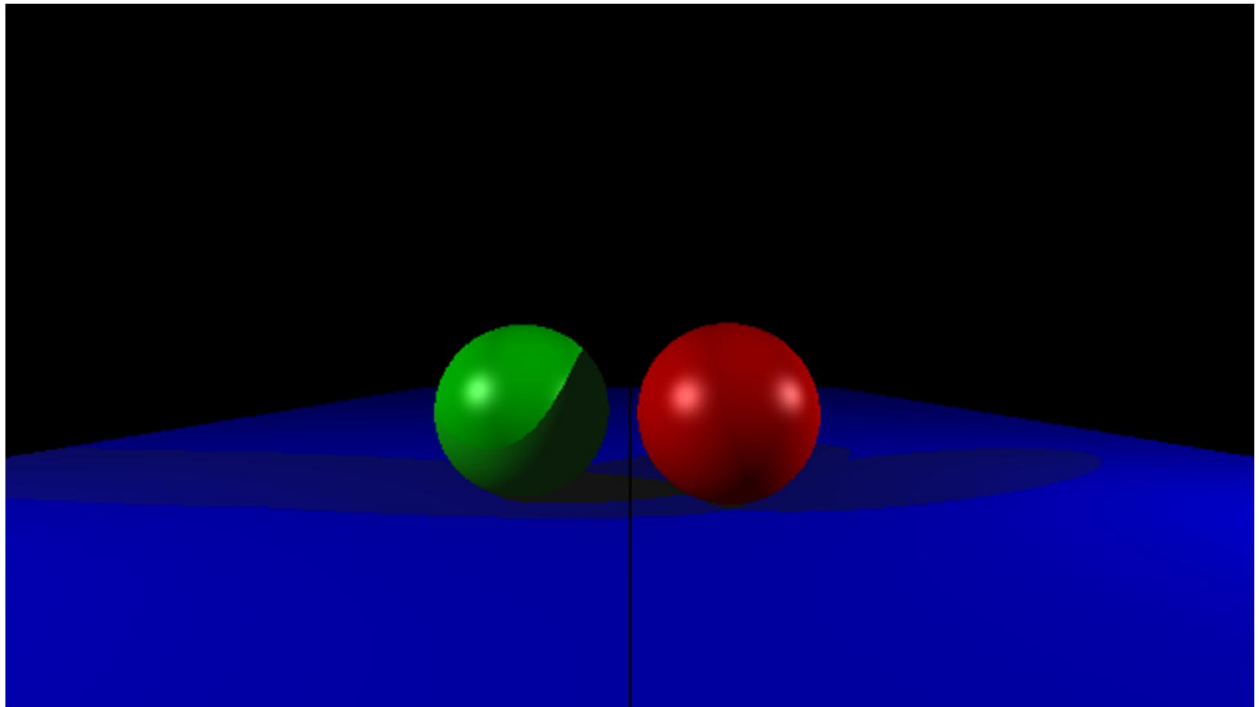
Shadows are added. Below is a scene implementing shadows with one light.



Shadows with 2 lights.

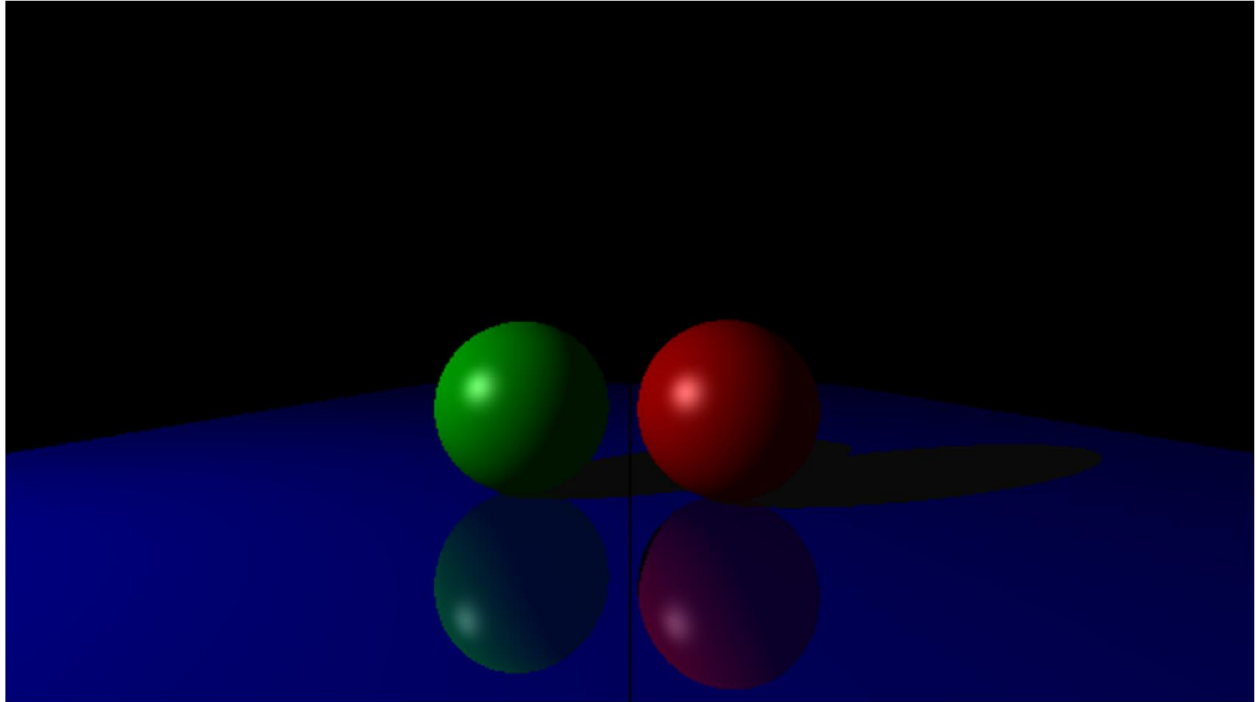


Position of lights changed.

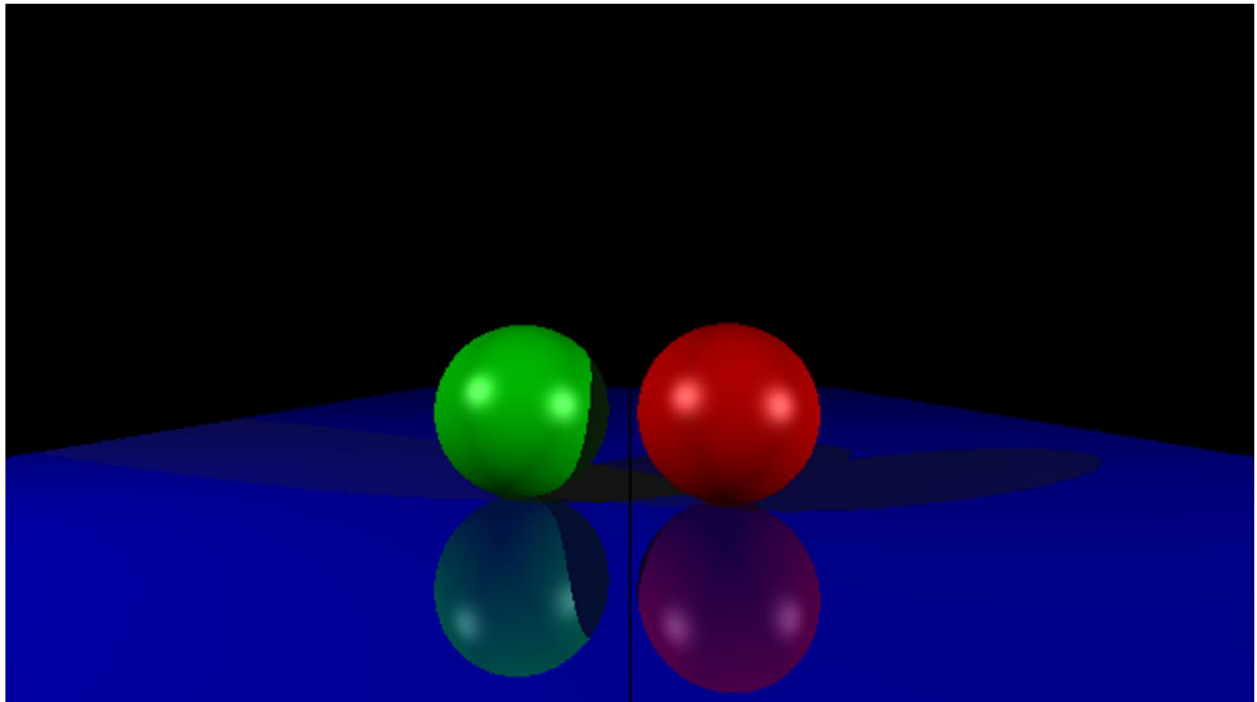


#### D. Special Effects:

The special effect I implemented in this project is reflection. In my setting, the blue plane (made up of two triangles) is reflective. The incident ray gets reflected from the plane and whichever surface the reflected ray strikes is seen on the plane.

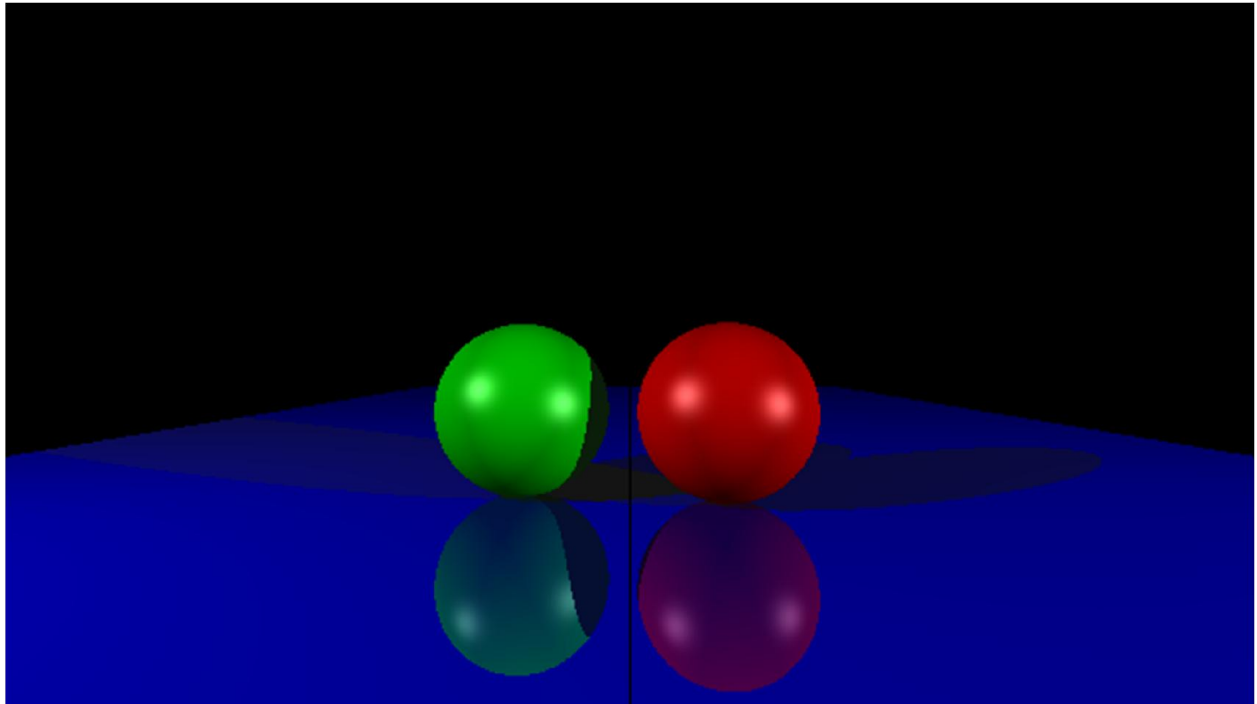


Reflection with two lights.

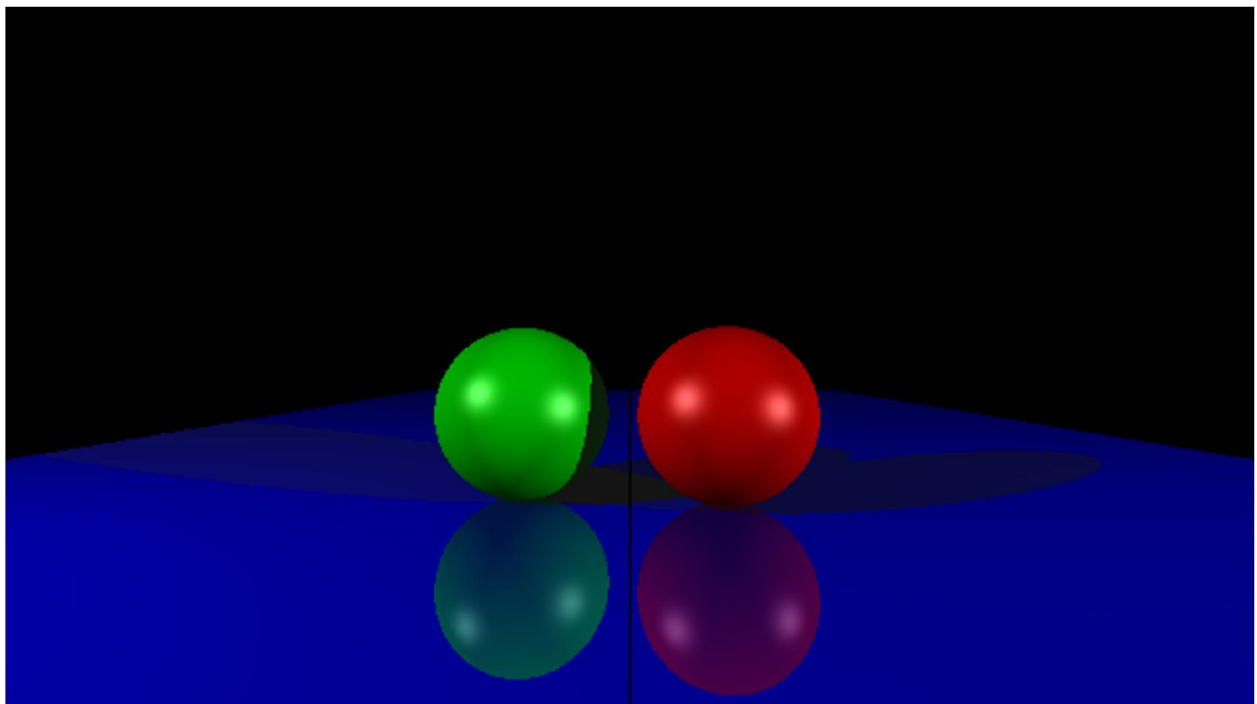


To show shadows in the reflected image, I checked that if the point intersected by the reflected ray is under a shadow. If it is, then a shadow was also shown in the reflected image.

Reflection with shadows

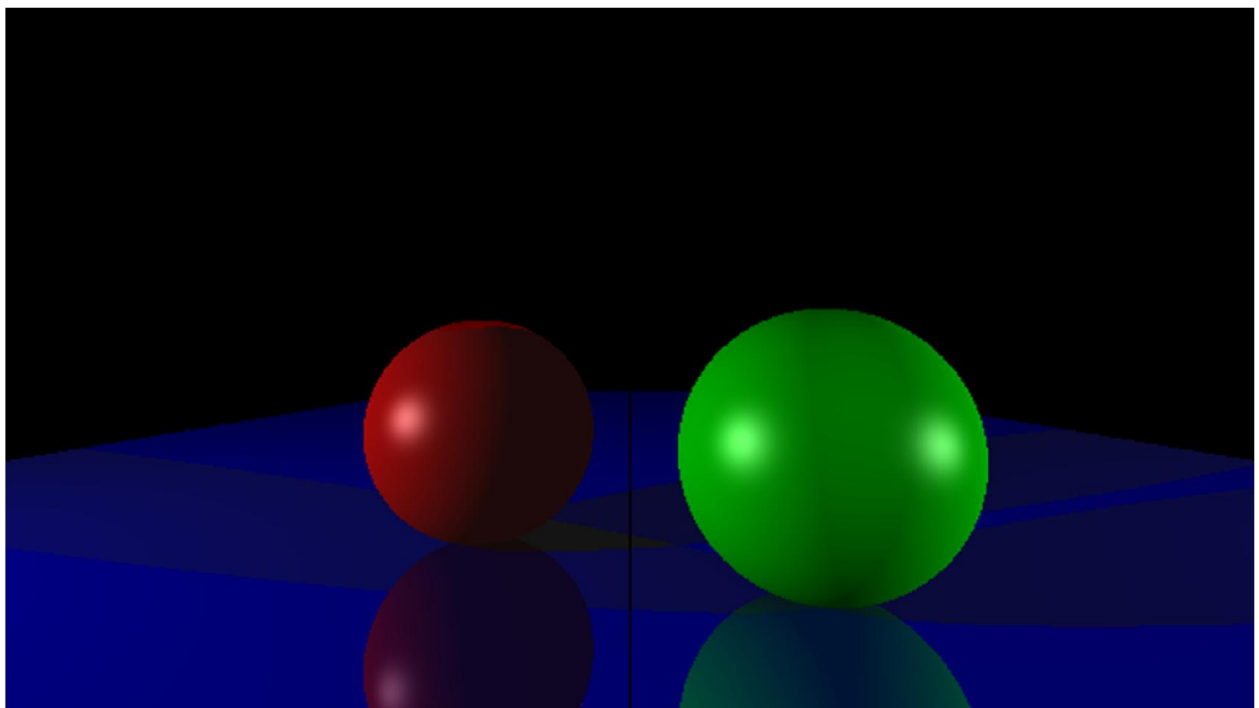
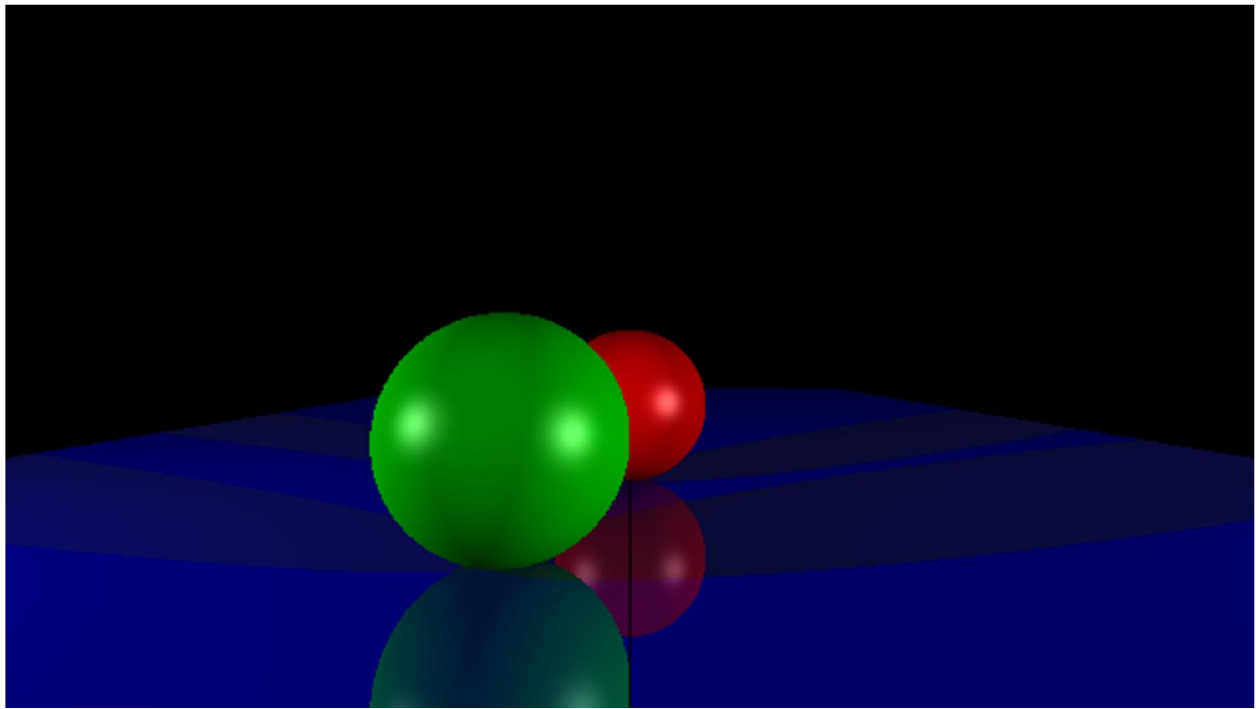


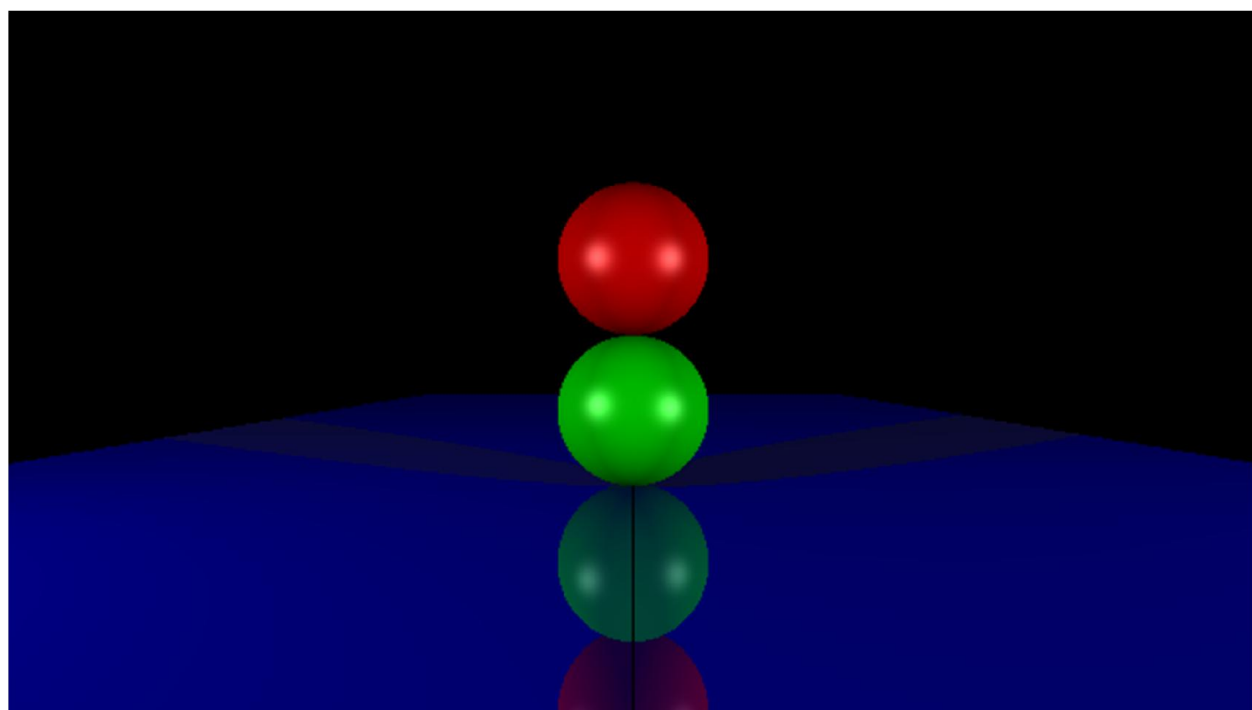
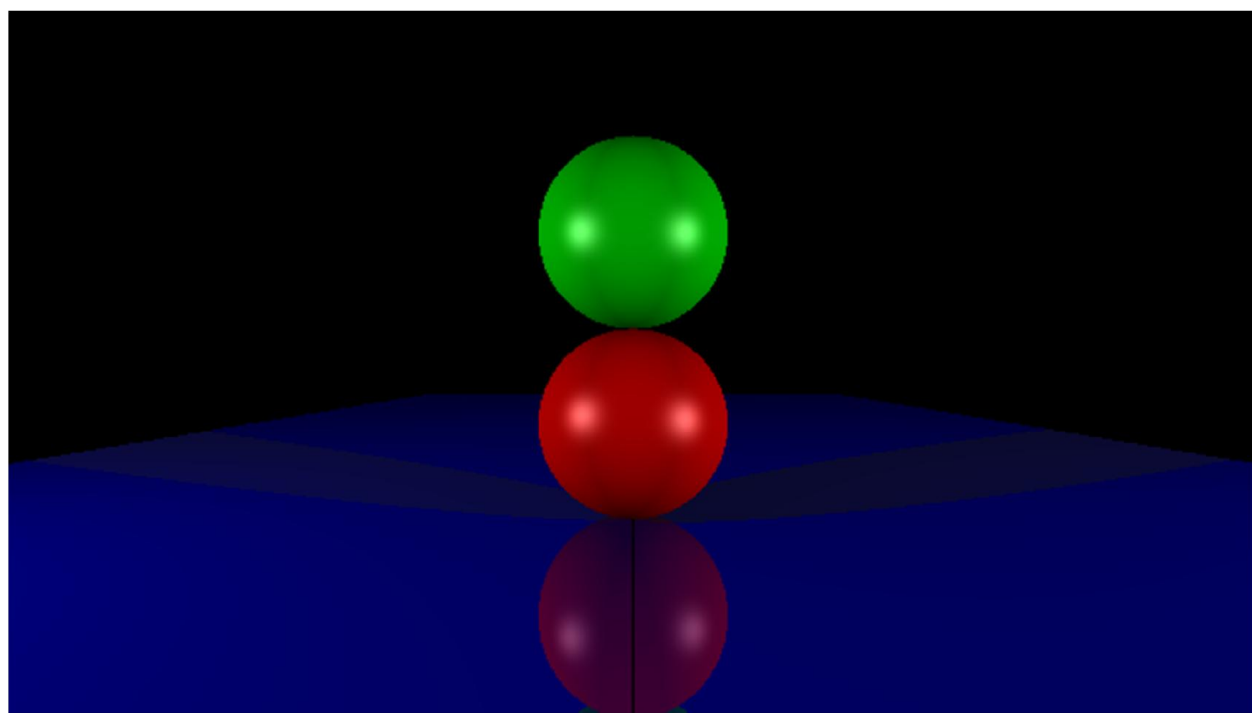
Reflection without shadows.





E. Shots with position of objects changed:





#### F. Linux Compilation:

The code was compiled on a Ubuntu machine hosted on a VMWare Virtual Machine. It was a freshly setup machine, so it came with only the default packages. I had to install “libglew-dev”, “freeglut3-dev”, “libglu-1-mesa-dev”, and “libsdl-image1.2” packages in order to compile the code in Ubuntu. I ran the “make clean” command before the “make” command. The code compiled correctly and the output was generated.

#### G. Issues:

There is some issue with the “LinearSolver.Run” method. For the all the pixels (x,y), where  $x = \text{screenWidth}/2$ , and y ranges from 0 to screenHeight, the method doesn’t return any solution. Same applies to all the pixels (x,y), where x ranges from 0 to screenWidth, and  $y = \text{screenHeight}/2$ ;

#### H. Other

I will be using the free 1-week extension for this project.