Project B: Ray-Tracing

Yifei Ge Net ID: ygt052

User's Guide

Press H for help.

Press T for ray tracing.

JKIL to rotate the camera, and WS to move forward/backward, AD for Up/Down.

Press space to change Scenes.

Press Z for turn on/off headlight.

Press X to turn on/off Jittler antialiasing.

Press C to turn on/off area light.

Press arrows and ctrl and shift to move the moveable light.

Use the text areas to set the moveable light (the other one is bounded to the eye point).

Use the lowest text area to set the recursion depth for reflection rays.

Note:

This ray tracing program runs a bit slow due to the large calculation, (cost more than a minute), but please turn of area light, turn down the recursion depth and turn off the antialiasing if you want a higher calculation speed.

And when running the program, please open the inspect and log, because the program runs a bit slow, or the chrome may consider it as not reacting, and stop it..

Code Guide

The code structure is based on the lecture slides.

Projb.js is modified form a last quarter project, it contains all the WebGL codes for three D world and lighting and shading, it supports multiple shaders but only used the phong shading. "myScene" is the ray tracing objects, I did the responding act to the modelmatrix and the CGeom.world2model matrix to keep the two viewport matched, and set the viewmatirxs correspondingly.

Raytrace.js is the ray tracing part of the program. Its key components are:

CRay (): the object that describes the ray with an origin and a direction.

CCamera():It contains the information about eye point and look at information, and a function setEyeRay to set the rays.

CGeom(): the object that describes the different shapes and how to trace them, including cube, sphere and ground plane. It also contains the model coordinate system.

CGeom.tracegrid(): function used to traced the grid, with the function z=0;

CGeom.tracecube(): it works the same as tracegrid, but trace six times for the six faces of a cube, and picked the actually face that the ray hit first.

CGeom.traceSphere(): this function is based on the lecture slides, using the half-chord method.

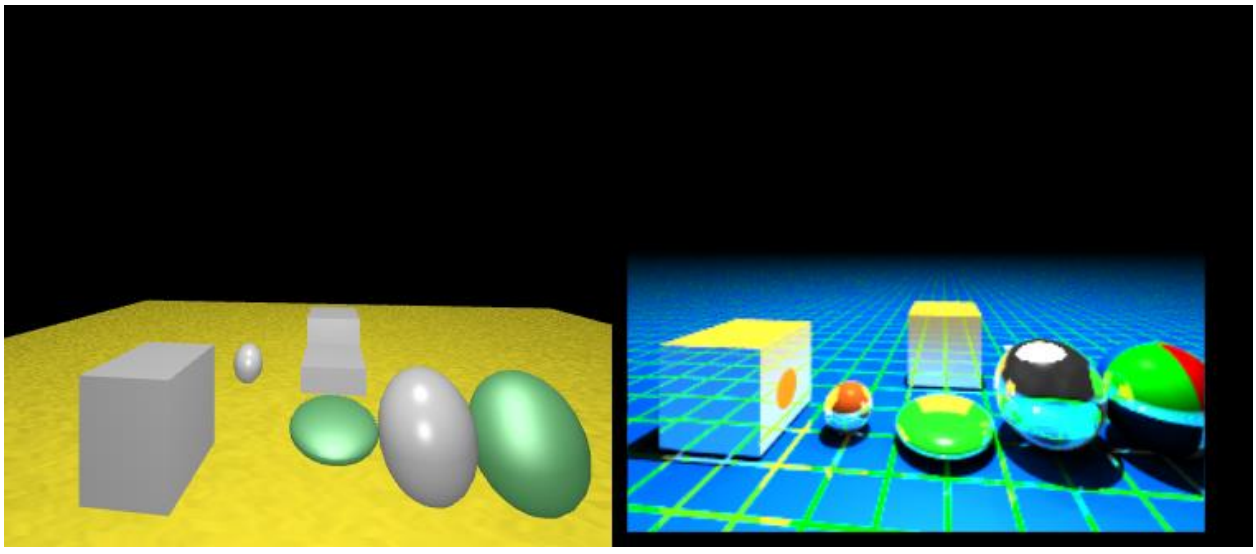
CHit() and CHitList(): Objects that contains the hit information of the ray.

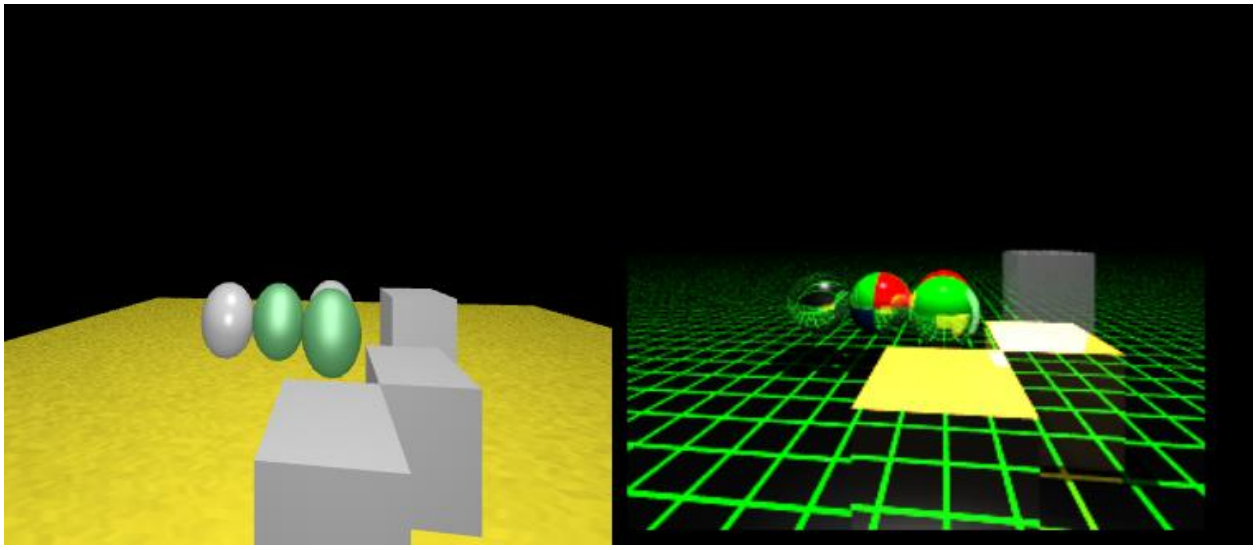
CScene(): The contains all the light information, the CGeoms, the materials and rays. By modifying the values in this object can directly effects the rays.

CScene().findShade(): The combined function that finish all color retrieve work for a ray, including the phong lighting color, reflection color and the refraction color if the material is transparent. (Thanks to my classmate Honghao for teach me a better way of refraction)

Results

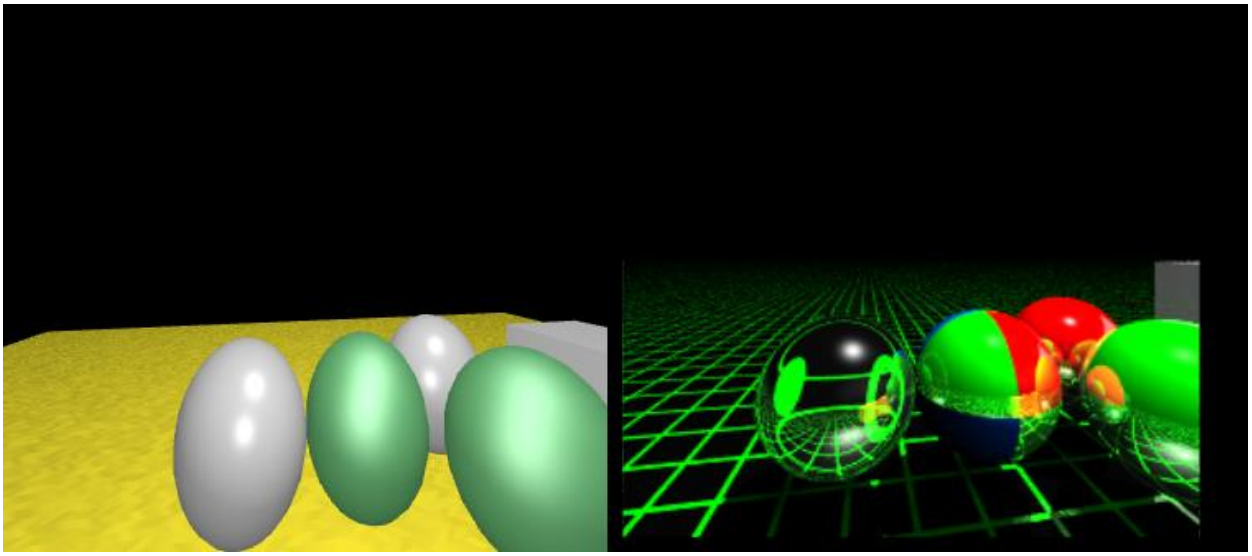
Two different Scenes:



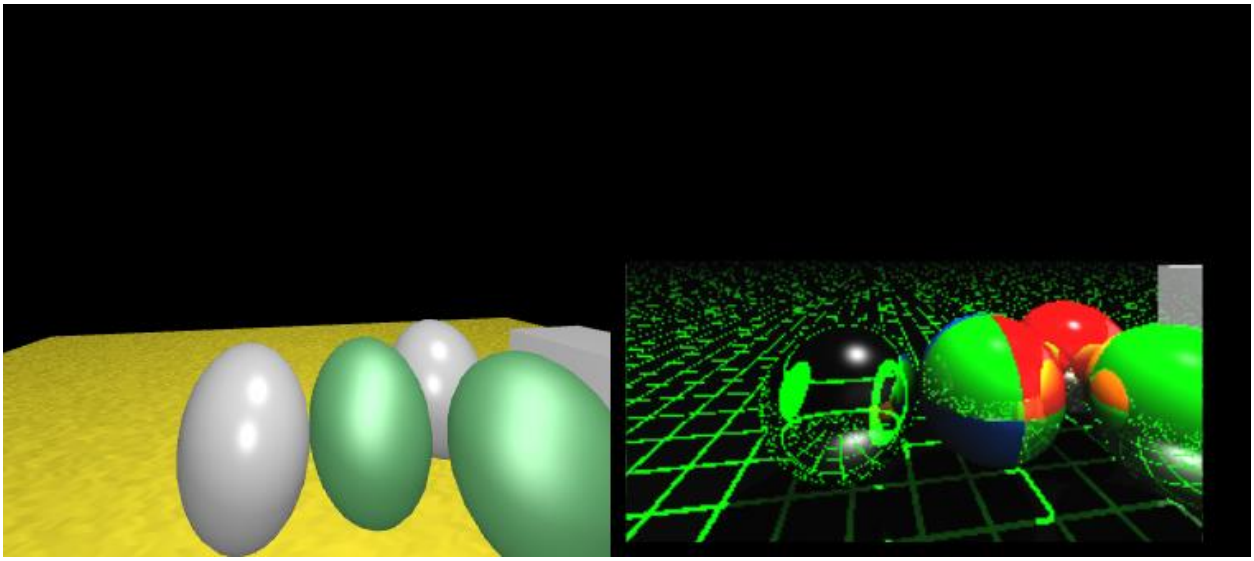


In the second scene the first two cubes are perfectly reflective, and you can see balls with different coloring using the checkerborard method.

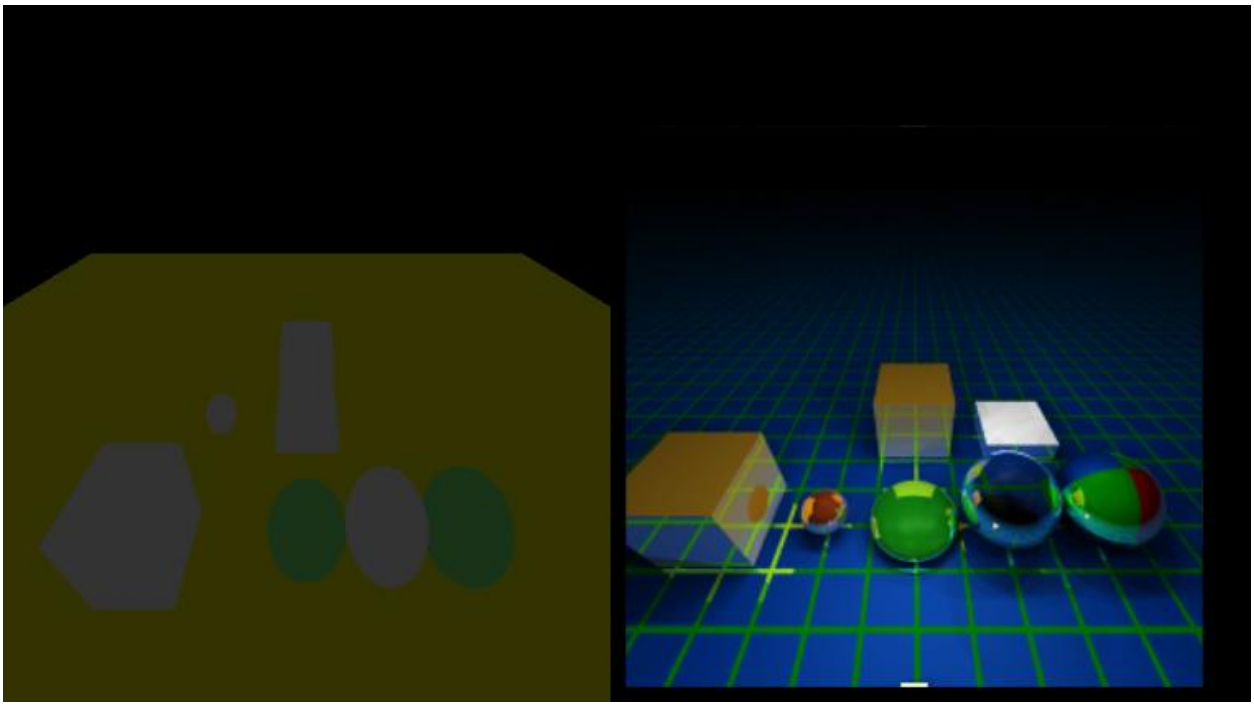
Free camera position& the transparent balls and multi layers of reflection &the shadow position:



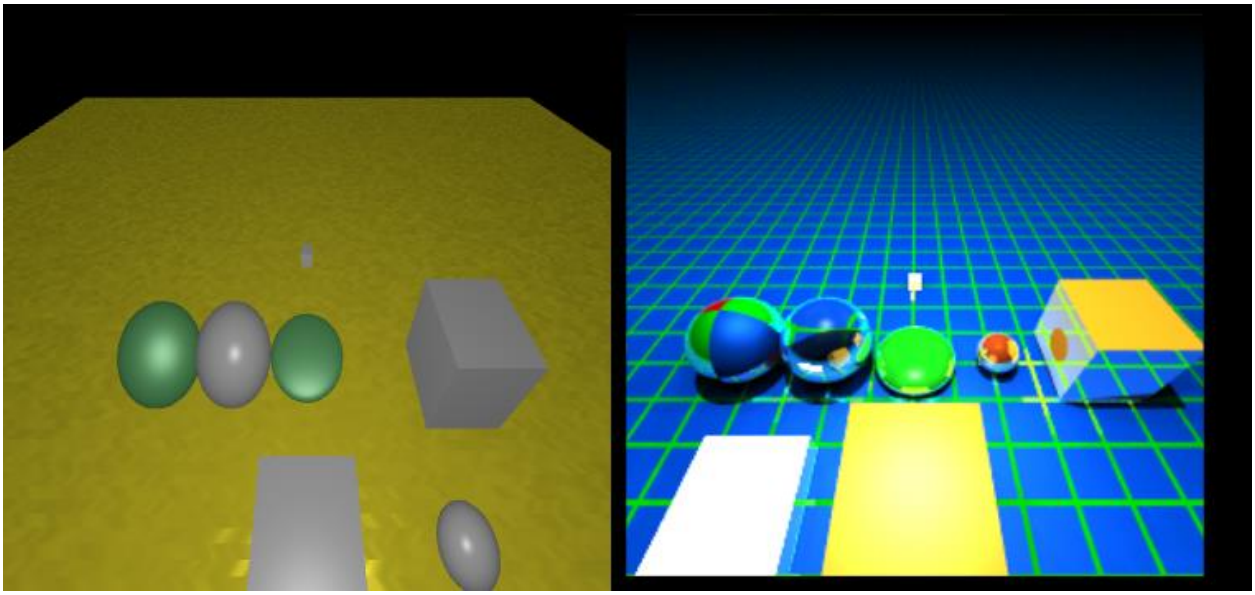
Turned off the antialiasing and the headlight:



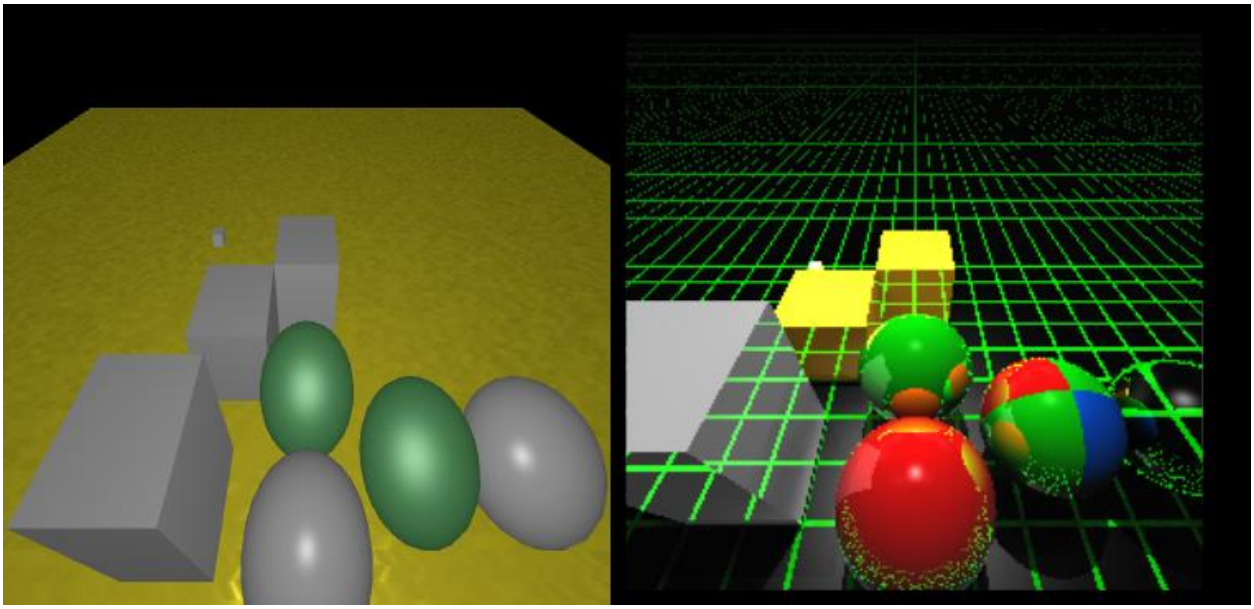
Turned off all light only left the area light(16 jitter assigned lights), in the webgl, I just used simple color to represent the area light, and the middle green ball is the non-uniform shape distortion of a ball(distorted to a pancake) :

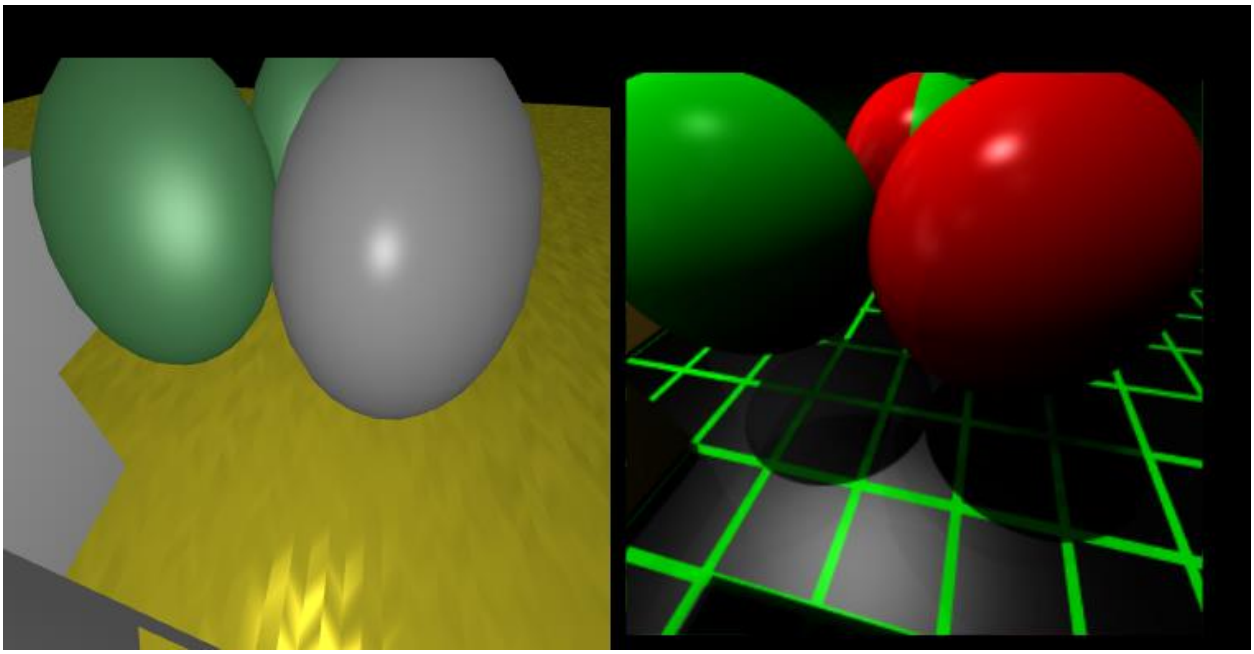


Move able headlight with its shades:



Some other Pictures:





(Reflection set to zero)