# Ray Tracing

Chin-Kai Chang

Yun-Yu Chen

Chien-Ming Liao

Chih-Jou Chang

Inspiration for our project:

The reason that we choose ray-tracing as the topic of our final project is that the reflections of the balls in all colors presented in professor’s slides looks amazingly beautiful. And we are so fascinated by how this simple algorithm can simulate the lights, reflections and shadows so naturally. Therefore, we decide to give it a shot to implement this amazing algorithm as our final project and savor the beauty of it while learning.

Language: JavaScript

We have chosen JavaScript for the advantages for following:

1. The professor highly recommends and mentioned a lot throughout the lecture.
2. The language itself is class free and 100% object-oriented language.
3. JavaScript is the most popular scripting language on the internet, and works in all major browsers, such as Internet Explorer, Firefox, Chrome, Opera, and Safari.

Design:

Just like a typical ray tracer, we shoot lights from a camera, shooting through the pixels of the screen plane. And determine the closest intersection with a primitive by testing all the rays. When a primitive is hit by a ray, the color for the ray can be calculated using the two lights we created. each light contributes to the color of the primitives. Illumination is determined by the light source by taking the dot product between the vector from the intersection point to the light source, and the primitive normal at the intersection point.

Scene: We build up our own scene to retrace, including spheres, planes and triangles.

Team collaboration:

Chin-Kai Chang : Scene, Output window, Ray tracer, textures

Yun-Yu Chen : Ray tracer, scene

Chih-Jou Chang : Ray tracer, scene

Chien-Ming Liao : Ray tracer

Challenges faced:

1. Unfamiliar to JavaScript: Most of the team members haven’t use JavaScript before. So it took longer than we expected to handle this language.
2. Finding the best / most accepted way to divide the work between team members: It is hard to divide the work and being able to compile for each programmers, due to each member could call the function the others are implementing.

How you would build on your program in the future:

1. More speed: Our code now is pure ray-tracing and would take a long time to render a sophisticate scene, to improve that, we are considering using hierarchical bounding volumes or kd-tree to improve the speed of rendering.
2. Soft shadows: Adding area lights to make the shadows looks more realistic.