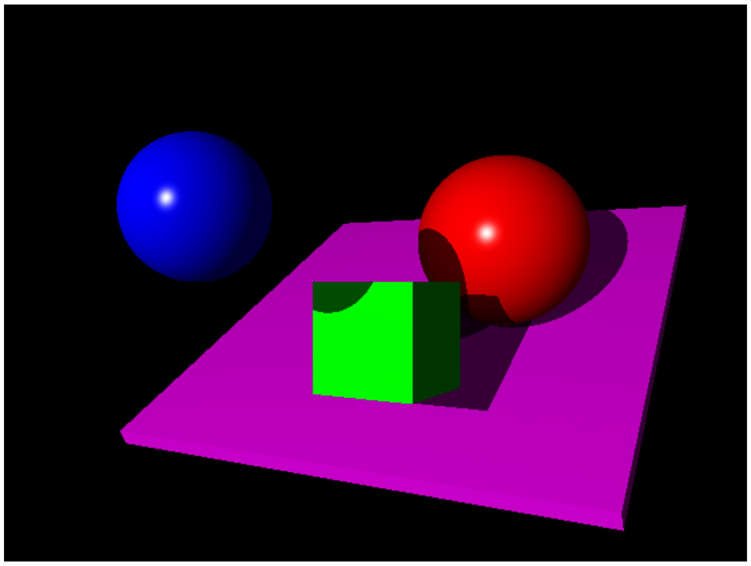
Assignment 5: Ray Casting

1. Due Date

Assignment 5 is due on **04/21 11:59pm**

2. Requirements

Ray casting is a simplified algorithm for ray tracing. It only casts primary and shadow rays, and ignore any reflection and refraction rays. In this project, you are required to write a simple ray casting program. You can write the program in C++ without shaders (the CPU-based implementation), or you can write the program using a computer shader (the GPU-based implementation) that casts and traces the rays in parallel. The GPU-based implementation will give you much better performance, but it's your choice which type of implementation you want to do. Either implementation is acceptable for this assignment.



Your program needs to ray cast the given scene (*geo.txt*) in the *geoData* folder. Your program needs to generate primary rays from the eye location through the near plane to the camera space, implement ray-sphere intersection and ray-box intersection (intersection with multiple rectangles), cast shadow rays, compute color for each pixel, and finally display the pixel buffer on the screen. The rendered image should be the same as the figure above. The image resolution is 640 x 480.

A basic code template is provided. It includes a loader to load the scene file, the basic sphere and box data structures with OpenGL drawing functions, and the camera control functions. Running the code template displays the wireframes of the scene. Feel free to use this basic code template in your project. You can also choose to create the project from scratch.

The ray casting method produces pixels. The pixels need to be stored in an array before they can be displayed. The code in *DrawPixels.cpp* is an example demonstrating how to store pixel values in an array and then send it to GPU memory to display as a texture on a quad.

3. Assessment

Your program will be evaluated using the *geo.txt* scene file. The scene has two spheres, two boxes, and one light source. Do not modify the scene setting.

3.1. (10pts) Your program should cast one primary ray per pixel through the center of the pixel. The image resolution is 640 x 480.

3.2. (20pts) Your program needs to calculate shadow ray color, and ignores reflection and refraction rays. Set the shadow color to black (0, 0, 0).

3.3. (40pts) Your program should calculate the hit points of rays correctly on the objects.

3.4. (30pts) When computing the color, if it's not the shadow color, use the Phong shading model for local illumination, like equations below.
Ia = ambient;
*Id = light_intensity * diffuse * s_dot_n;*
*Is = light_intensity * phong * pow (r_dot_v, 50);*
*color = (Ia+Id) * light_color * obj_color + Is * light_color;*

s is the normalized shadow ray (hit point to light); *v* is the inversed primary ray (hit point to eye), and they're normalized; *n* is the normalized normal vector at the hit point; *r* is the reflected ray of *s*, and it's normalized too.

4. Submission

Upload the following things to *mycourses*:
(1) A rendered image from your program.
(2) A *.zip* file containing all source files (*.h* and *.cpp* files).