University of Waterloo – Faculty of Mathematics  
CS 488 – Introduction to Computer Graphics

Final Project Proposal  
Revised

Richard Ye  
ID: 20418361

March 19, 2015

# Purpose

To create a ray racer that extends the ray tracing concepts learning in assignment 4.

# Topics

* Material and surface characteristics, such as refraction and reflection.
* Rendering techniques, such as soft shadows, depth of field
* Optimization techniques, such as adaptive anti-aliasing and uniform spatial subdivision.

# Outline

This project is an extension of the ray tracer completed for Assignment 4. The project will be implemented in JavaScript with the scene defined using JSON. The program shall run in any modern browser, though Google Chrome has a faster JavaScript engine that will reduce rendering time.

The scene to be rendered is a simple chessboard. The pieces will be modelled using standard geometric primitives and/or defined models. The pieces on one side will be transparent and will use refraction; the pieces on the other side will be opaque and will use reflection. The board itself will be a checkerboard textured via Perlin noise. There will be sufficient lighting to demonstrate material properties and soft shadows, as well as bump mapping.

**Objective 0: Supersampling**  
Naïve, trivial supersampling was implemented as the extra objective in A4.

**Objective 1: Mirror Reflection**Materials that have mirror-like reflective properties shall be implemented. The techniques for this are well-known and one example is described in (Hughes).

**Objective 2: Refraction and Transparency**Refraction is a common ray tracing technique described in (Hughes). Transparency is a prerequisite for refraction and shall be achieved by adding an alpha channel to all colors in the scene.

**Objective 3: Texture Mapping**  
Textures shall be implemented using mapping functions for available primitives. PNG images can be specified for non-mesh primitives in the scene. Meshes will only support generated textures, such as Perlin Noise (Objective 5).

**Objective 4: Bump Mapping**Bump mapping shall be supported for non-mesh primitives.

**Objective 5: Perlin Noise**  
Perlin Noise shall be implemented for textures on all primitives. Different examples of parameters passed into the Perlin noise function, and different based textures, will be used to demonstrate the range of textures that can be generated, for instance, wood and marble. (Hart)

**Objective 6: Soft Shadows**  
Light sources shall be allowed to be spheres defined by a point and radius, and not just points. The lighting engine will support realistic rendering of said lights using soft shadows using a naïve multi ray-cast approach. (Mo)

**Objective 7: Adaptive Anti-Aliasing**  
A basic edge detection algorithm shall be employed to find suitable candidate pixels for multisampling to reduce the effects of aliasing.

**Objective 8: Depth of Field Effect**  
A depth of field effect shall be simulated a post-processing effect where rendered pixels that are far from the focal length are blurred. This will require z values to be stored along with color for each pixel.

**Objective 9: Uniform Spatial Subdivision for Models**  
Models shall have their triangles divided using uniform spatial subdivision. Rays that are cast within the bounding volume for the model shall take advantage of this data structure to find potential intersecting triangles. (Teschner)

**Objective 10: Rendering of Final Scene**  
An original final scene shall be rendered that demonstrates all of the ray tracers capabilities.