

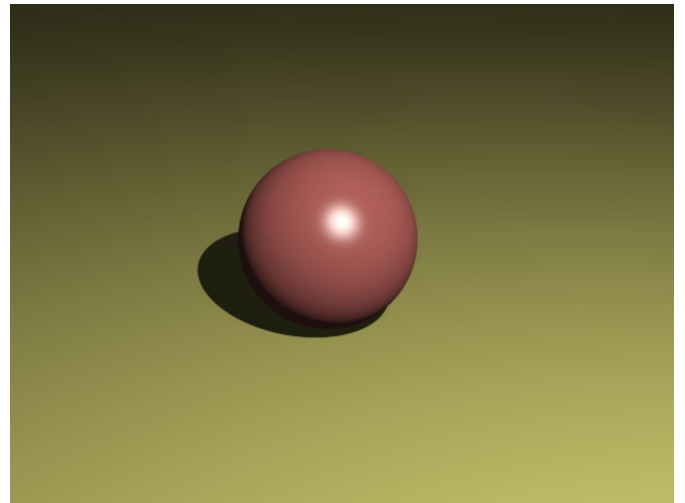
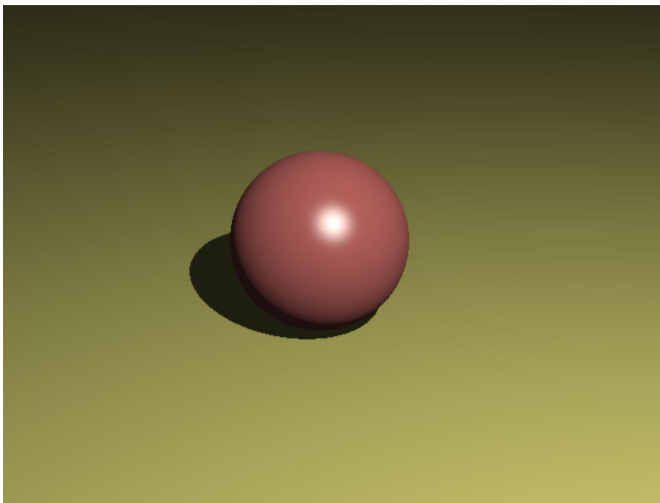
Homework 7: Ray Tracing Extensions

I implemented the following extensions:

- Anti-Aliasing (2 pts)
- Acceleration Structure (14 pts)

Anti-Aliasing:

In the render function, I've supplied a parameter (sampleSize: line 35) to modify the sample size, default should be 1. I've included a comparison between the stock ball pointlight scene and 4x4 super-sampled scene. The edges of the sphere and shadow are particularly more smooth.



Acceleration Structure:

The majority of the source code for this can be found in the “bounding.js” file. As for my implementation, I modified the shapes to all have a function that defines a bounding box. Since I used Box3, all of these boxes had to be axis-aligned and thus the dimensions were just the largest occurring values for each coordinate component. I also created a node class for the hierarchy tree that, given an array of shapes, creates a bounding box for that array. At every level, I split the shapes into two groups, representing the shapes that will define the left and right subtrees of the tree. Eventually, when there is only a single shape remaining, I create a leaf node. After the tree is constructed, for every ray, I check for intersection against the tree. If at any level, the ray does not intersect with the current node, then I stop recursion. Otherwise, I continue until I reach a leaf node and append that shape to a list of shapes that I must further check for a more precise intersection point. I modified rayIntersectScene to only consider the

array of shapes produced by this method, instead of always checking the ray against the entire shapes list: thus reducing the total number of intersections.

I tested some of the more novel scenes with more geometry and here are the results.

Teapot (Firefox):

accel **ON**: 4000 - 4100 ms \approx 4 seconds

accel **OFF**: 106516 - 123763 \approx 110 seconds (1.8 minutes)

accel **ON (4x4 SS)**: 30658 ms \approx 30 seconds

accel **OFF (4x4 SS)**: Too Long

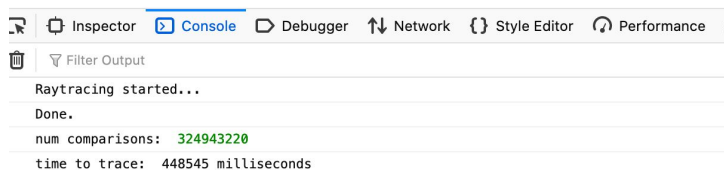
Cow (Firefox):

accel **ON**: 15197 - 15296 \approx 15 seconds

accel **OFF**: 426836 - 633008 \approx (7 - 10 minutes)

accel **ON (4x4 SS)**: 448545 \approx 448 seconds (7.5 minutes)

accel **OFF (4x4 SS)**: Too Long



There's a little bit of set-up that goes into making this work though. If you want to enable acceleration in a scene, make sure that the following globals are defined in the scene html file:

- Planes: array for all plane objects since they don't have a bounding box and are considered by every possible ray.
- numComparisons: a simple counter that measures the efficacy of optimization. Tree structure should reduce the number of ray-scene intersection calls.
- Two variables for time stamps (t1, t2). A more precise measure over numComparisons
- Boolean accel set to true

Notes: some of the less geometrically complex scenes see no benefit from the acceleration structure, in fact experiencing an increase in run-time from the tree's construction (presumably). However, as showcased above, the scenes that import an object file with a lot of triangle geometry are accelerated a fair bit.

Thanks for a great semester! :)