1 Basic Camera Model

The basic camera model is implemented inside the camera class. The camera class allows the creation of rays for a given pixel. The class first uses the Up and Lookat vectors to create an orthogonal basis w, u, v. An field of view is defined and then used along with the value for the width, hight and current pixel location to calculate the point on the camera screen that corresponds to the top left corner of the pixel. Once we have the point on the screen the viewing ray can be calculated and returned.

In order to implement anti-aliasing for each pixel 16 rays, these are aranged in a regular grid across the pixel where each ray is 0.25 pixel widths from the previous. The colour values for each of these rays are then averaged to give the anti-aliased value. Figure 1 shows a raytraced sphere with and without anti-aliasing turned on.

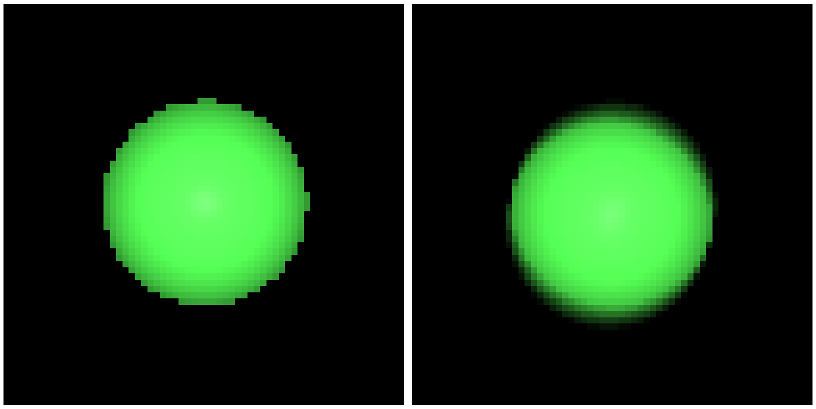

Figure 1: Sphere raytraced with and without anti-aliasing

2 Plane & Triangle Intersection

The triangle intersection is done in the triangle class. The class takes three vertexs as parameters, these are the three corners of the triangle. This first step in the intersection test is to calculate the three vectors which make up the sides of the triangle, these are used to get the normal by the application of the cross product [1]. Using the normal and a point the intersection between the ray and the plane the triangle lies on can be calculated then the value of t that the ray intersects at can be found. After this the cross product is used to check the intersection point is on the inside of each of the sides of the triangle.

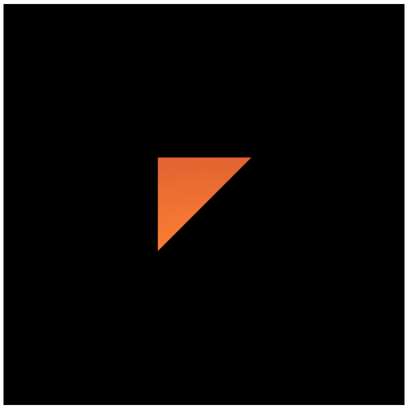

Figure 2: Triangle intersection

3 Quadratic Intersection

Quadratic surfaces are defined by 9 terms A - J in the equation below.

$$Ax^2 + 2Bxy + 2Cxz + 2Dx + Ey^2 + 2Fyz + 2Gy + Hz^2 + Iz + J = 0$$

In order to calculate the intersection point between a ray and the surface the ray equation $Dt + P = 0$ can be substituted into the quadratic. This substitution gives the quadratic equation below where dx, dy and dz are the x,y and z components of the direction and px, py and pz are the components of the initial ray position.

$$Aqt^2 + Bqt + Cq = 0 \text{ Where}$$
$$Aq = Adx^2 + Edy^2 + Hdz^2 + Bdxdy + Cdxdz + Fdydz$$
$$Bq = 2Apxdx + 2Epydy + 2Hpzdz + B(pxdy + pydx) + C(pxdz + pzdx) +$$
$$F(pydz + pzdy) + Ddx + Gdy + Idz$$
$$Cq = Apx^2 + Epy^2 + Hpz^2 + Bpxpy + Cpxpz + Fpypz + Dpx + Gpy + Ipz + J$$

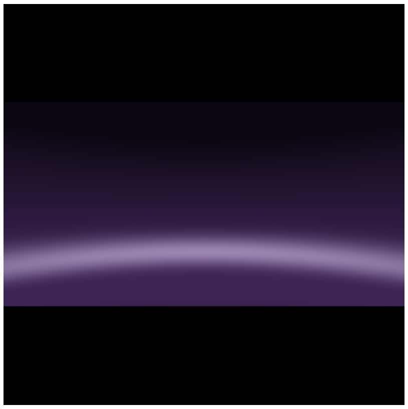

Figure 3: Quadratic Intersection

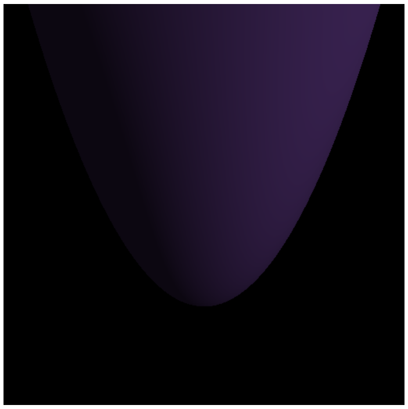

Figure 4: Basic Camera Model

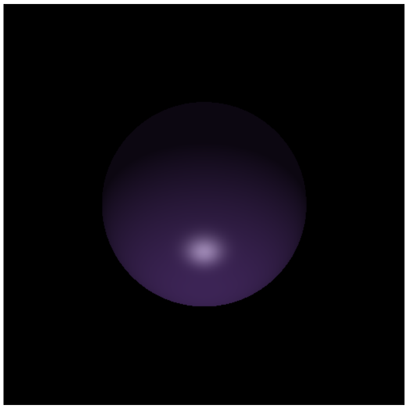

Figure 5: Basic Camera Model

4 Point Lights

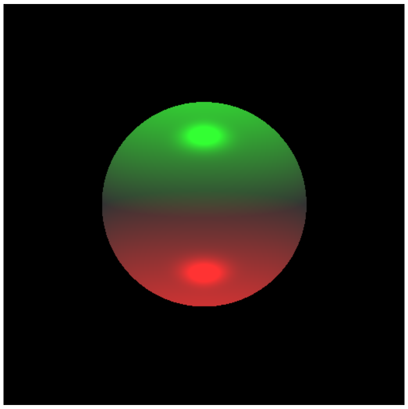

Figure 6: Basic Camera Model

5 Specular Material

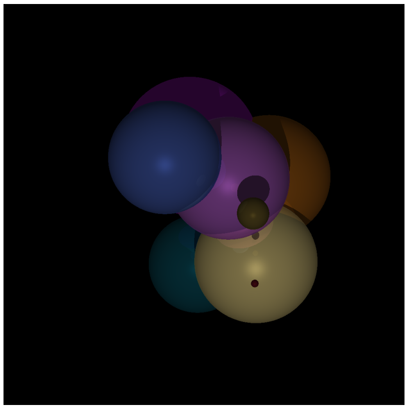

Figure 7: Basic Camera Model

0.

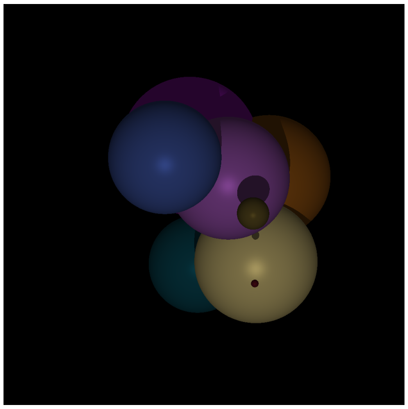

Figure 8: Basic Camera Model

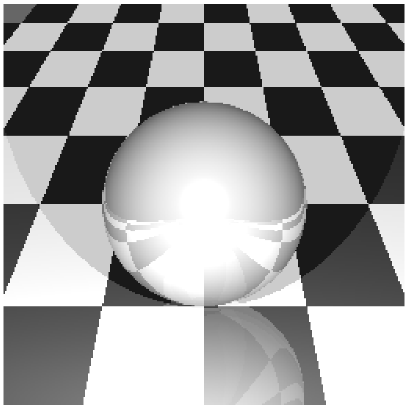

Figure 9: Basic Camera Model

6 Shadows

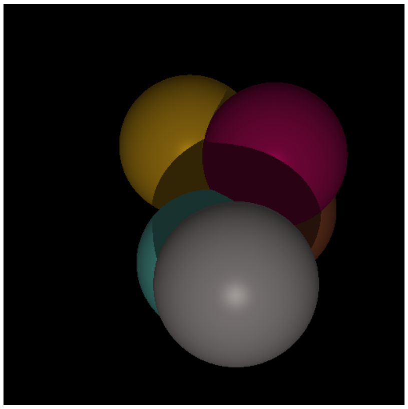

Figure 10: Basic Camera Model

7 Transparent Material

8 Octree

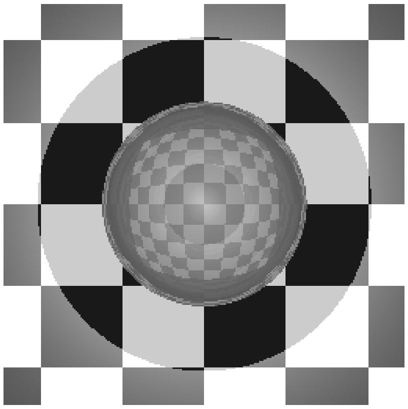

Figure 11: Basic Camera Model

9 seconds with octree, 53 without.

References

[1] Ray tracing: Rendering a triangle. `https://www.scratchapixel.com/lessons/3d-basic-rendering/ray-tracing-rendering-a-triangle/ray-triangle-intersection-geometric-solution`. Accessed: 2016-04-10.