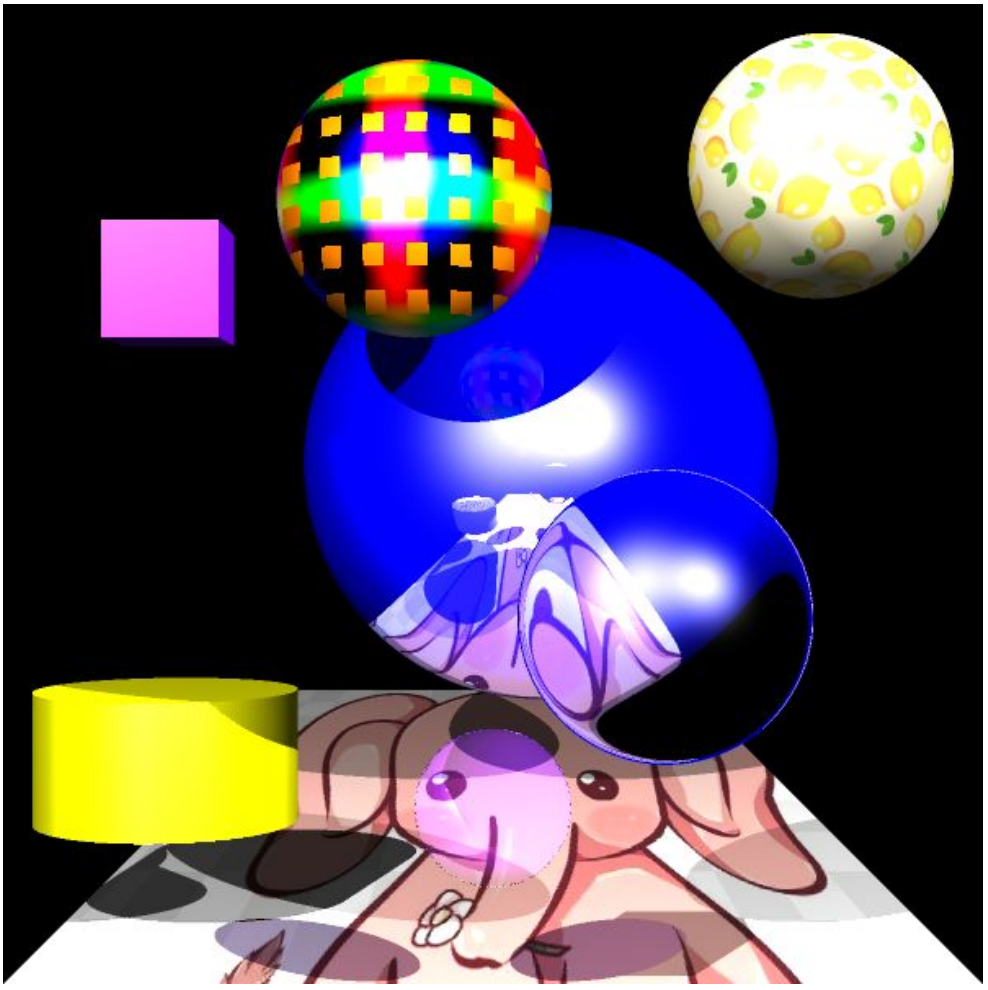
COSC363
Assignment 2 Ray Tracing Report

Claire Barnaby
Student ID: 52159849

31/05/2017

The scene



This has been rendered at 600 divisions with the recursion limit for antialiasing set at 10 and the threshold at 0.25. This took 30 seconds to render on lab machines. The scene has 5 spheres. The big blue is reflective, then one has a procedural texture, the one will lemons on it has been textured with a image. The semi pink one is transparent and the last sphere refractive. There is a yellow cylinder with caps and a pink cube. The cube is made of 6 planes of equal size. The is a ground plane that has been textured with a pink elephant. To texture this the equation is

$$t_s = \frac{x - p_{x\,\min}}{p_{x\,\max} - p_{x\,\min}}$$
$$t_t = \frac{z - p_{z\,\min}}{p_{z\,\min} - p_{z\,\max}}$$

Where x the ray intercept x coordinate and p_{xmin}, p_{xmax}, p_{zmin}, p_{zmax} are the minimum and maximum points on the plane for x and z.

Extra Features

Cylinder

There is a cylinder in the scene made from an intersect function and a normal function. Below a, b, and c come from the quadratic equation and t_2 and t_1 are the solutions to the quadratic equation. In equation for a, b, and c, d_x is the direction vector's x direction and d_z is the direction vector's z direction. The radius is r and the point position is represented by x, y and z. Finally z_c, x_c and y_c are defined by the center coordinate of the cylinder which is the center of the bottom circular face when the cylinder is standing up.

$$a = {d_x}^2 + {d_z}^2$$
$$b = 2(d_x(x - x_c) + d_z(z - z_c))$$
$$c = (x - x_c)^2 + (x - x_c)^2 - r^2$$

Using a, b and c in the equations for the t values will give you the intersect points for an infinite cylinder. To make it finite then the intersection point y coordinate should be between the lowest face and highest face for a vertical cylinder. So to check this the intersect point y coordinate is found for both t_1 and t_2 by

$$p_1 = y + t_1 * d_y$$

and

$$p_2 = y + t_2 * d_y$$

Then if either point is between the center bottom y coordinate and the center bottom y plus the height then the point will be seen as valid. If both t values are seen as valid then the minimum one that is closest to the point will be used. If one is not valid then the valid one is used. It is important to take into consideration that if the first t value is not valid then the second value might be valid because it could be going through the top of the cylinder and hitting the back.

To get end caps on the cylinder two more t values need to be calculated. These are described below where h is the height and c_y is the lowest point of the cylinder from the center coordinate.

$$t_3 = \frac{h - y + c_y}{d_y}$$
$$t_4 = \frac{c_y - y}{d_y}$$

The t_3 value is valid when t_1 is above the highest y coordinate of the cylinder (height + c_y) and t_2 is below the highest y coordinate. The t_4 value is valid when t_1 is below the lowest y coordinate of the cylinder (c_y) and t_2 is above the lowest y coordinate. If they are both valid then the lowest one should be used.

The normals the curved sides of a cylinder are calculated by a vector of

$$(\frac{(p_x - c_x)}{r},\ 0,\ \frac{(p_z - c_z)}{r})$$

Where the point is p and the center coordinate at the bottom of the cylinder is c and the radius is r. For the flat faces the normal vector goes straight up or straight down. If the point has the same y value as the highest point of the cylinder then the normal goes up, (0, 1, 0). If the point has the same y value as the lowest point of the cylinder then the normal goes down (0, -1, 0).
I found the cylinder especially difficult. It was the most difficult because I made mistakes in my equations and end caps were difficult to put on.

Multiple Light Sources

There are two light sources in this ray traced image. The effect of this is there are 2 specular highlights two shadows from each object. In real life there are more than one light source. The specular two reflections can be seen easily on the sphere that refracts light and the blue sphere. If the lights were further apart if would be easier to see. Each shadow can be in one in shadow from one light or both lights. If it is only in one light then the shadow is lightened by adding the light vector, normal vector dot product by the other light multiplied by the colour. The colour of the object is calculated taking into account of both lights by adding 2 lighting terms together where each term is the light vector, normal vector dot product added with the specular term multiplied by the specular colour which is white.

Refraction

One sphere refracts its surroundings, the effect of this is that the sphere looks similar to a glass ball. Refraction is where a light ray is slowed down when passing through and object so it bends. This is simulated in the ray tracer by changing the direction of the ray when it enters a sphere. When the ray hits the sphere, then the refraction direction vector is calculated by glm refract which used snell's law and a refraction index which is given to it. A new ray is created at this new direction that travels to the object outside object barrier. Then a new ray direction is calculated by the refract method with a refraction index given to be 1 over the refraction index of the sphere. This new direction will be the same as the first ray direction if it is going into the same median as it came into the object. This ray is then traced and the colour of the object comes from this ray. The shadows have also been lighten because not all the light is blocked by the object.

Transparency

Transparency is like refraction apart from the light ray is not bent. The effect of this is that you can see through an object. In the ray tracer the only thing that is different with the transparency and refraction is that the refraction index is 1 which means the light doesn’t bend and goes straight through the object. The shadows have been lightened and made to be slightly the colour of the transparent object that the ray went through much like real life.

Non planar Texturing

There is one sphere with a lemon texture in the scene. The equations I used to map the lemon texture to the sphere are

$$t_t = 0.5 - \frac{\sin^{-1}(n_y)}{\pi}$$

$$t_s = 0.5 - \frac{\sin^{-1}(n_x)}{\pi}$$

Where t is a texture coordinate and t and s correspond to x and y. The normal vector was used and represented by n. I found that these equations did not stretch the lemons out and that is not what I wanted. I also experimented with another equation for t_s

$$t_s = 0.5 + \frac{\tan^{-1}\left(\frac{n_y}{n_x}\right)}{2\pi}$$

But this stretched the lemons and was not what looked best.

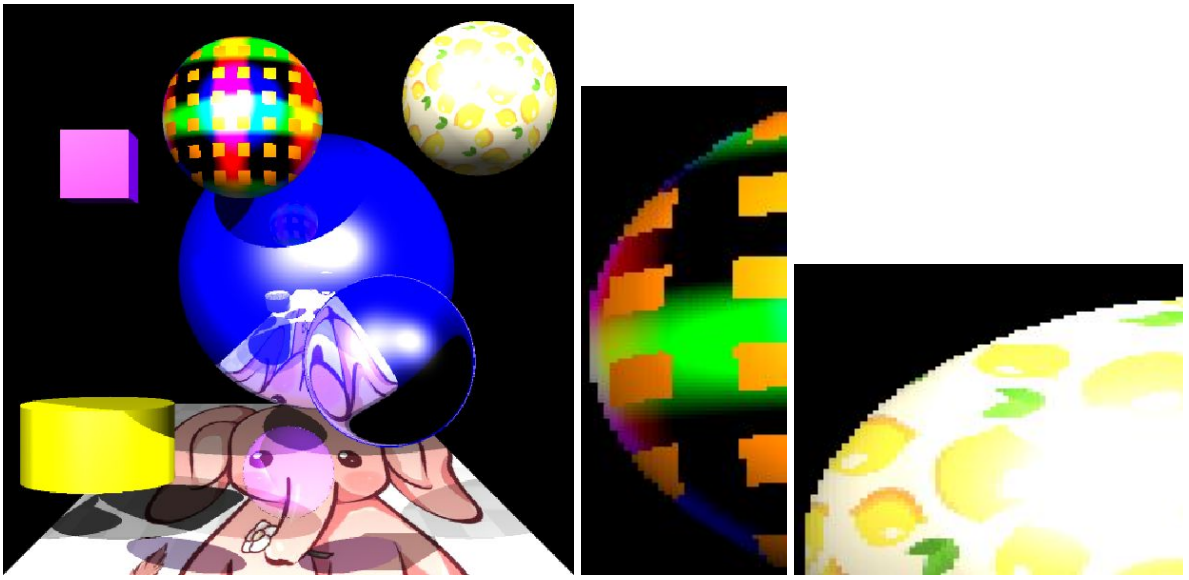
Procedural Texturing

The procedural texture is made by using sine and cosine. I used sin(3x) and cos(3y) to get the frequency of the orange squares higher. Where x and y were the x and y coordinates of the intersection point. Then if sin(3x) was greater than 0 and I made it a orange, so stripes would be procedurally generated. To make the stripes into squares then I restricted it further to only make it orange if sin(3x) and cos(3y) were both greater than 0 then make that point orange. The multicoloured background was a result of the colour being set to (sin(x), sin(y), sin(z)) where x, y, z where the x, y, z coordinates of the intersection point. This lead to a cool effect so if either sin(3x) or cos(3y) were less than 0 I made the colour (sin(x), sin(y), sin(z)). The effect of this resulted in orange squares with a multi colours background that is a kinda tartan pattern.

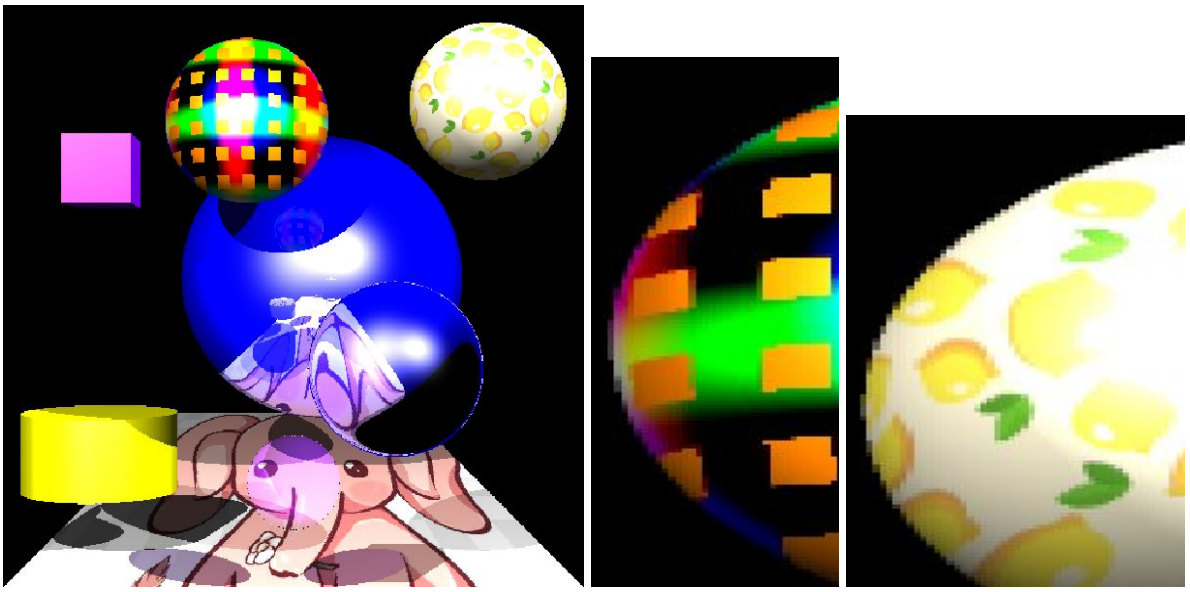
Anti-Aliasing

Anti-Aliasing is used to smooth out the the image and makes the edges of objects less jagged. It is does this by splitting the ray into 4 and directing them in the middle of each quarter of the cell. This means 4 colour values are generated the colour of the cell is then chosen to be the average of the 4 colour values. This results in a smoother look. To get an ever smoother look you can do recursive anti-aliasing which I have implemented. This means that if the 4 colour values are different enough from each other then recurse and split up the ray that was wasn't the same enough to the other 3 colours. That ray gets split into 4 then and the new colours averaged. The smaller you set the difference between the 2 colours the longer it will take.

The differences between anti-aliasing and without it are seen below
Without anti-aliasing 500 divisions:



With anti-aliasing 500 divisions recursion limit of 10 and threshold of 0.25 and time taken is 24 sec:



Resources

Lemons texture - http://wallpaperswide.com/lemons_pattern-wallpapers.html
Pink elephant texture - https://s-media-cache-ak0.pinimg.com/736x/ea/83/79/ea8379df2f8519192d1877fac8dfb449.jpg
Lec09a_MoreRayTracing.pdf by R. Muckundan of CSSE Univeristy of Canterbury
Lec09_RayTracing.pdf by R. Muckundan of CSSE Univeristy of Canterbury.