Debug Drawing

jodavis42@gmail.com

Debug Drawing Shapes

Self Explanatory Shapes:

- Line
- Aabb
- Triangle
- Quad
- Frustum (with 8 points)

This will be a short presentation on debug drawing. The idea behind this debug drawing presentation is to figure out how to draw various shapes as a collection of line segments.

There are a few shapes that I won't go into as they should be self explanatory (just connecting various points with lines). These are Line, Aabb, Triangle, Quad, and Frustum (at least frustum where we have all 8 points).

Orthonormal basis

From a given direction how can we build a basis?

Use a world-basis (x, y or z) to compute a second vector.

One incredibly useful operation for the rest of the shapes is the ability to compute a basis from just 1 direction vector. With 2 vectors we can easily compute a 3rd as $\vec{w} = Cross(\vec{u}, \vec{v})$. With only 1 vector we need to choose some other vector to cross by, such as the z-axis. That is, given $\vec{u}$ we can compute $\vec{v} = Cross\left(\vec{u}, Vector3(0, 0, 1)\right)$ which just simplifies to $\vec{v} = \left(u_y, -u_x, 0\right)$. This isn't guaranteed to work though, as what if $\vec{u} = (0, 0, 2)$. In this case we'd get a result that is the zero vector. In this case we'd just need to choose another basis (either the x or y for instance).

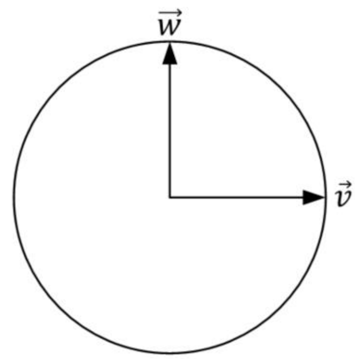
So as long as u_x or u_y is non-zero, then the z-axis is a valid axis to use, otherwise we can just use the x-axis.

Debug Drawing: Disc

Given a normal, point and radius, draw a disc.

Construct a basis from the normal

Paremetic equation of Circle:
$\vec{P}(\theta) = r(\vec{v}\cos(\theta) + \vec{w}\sin(\theta))$



Another important function for subsequent draws is the ability to draw a disc with a given normal vector. Using GenerateOrthonormalBasis as described before, we can compute a basis that spans the plane of the disc. From here the parameteric equation of a circle can be used to compute points on the disc to connect with line segments.

Debug Drawing: Ray

Line + Arrow-head

Draw a disc of a certain radius

Connect with lines to tip

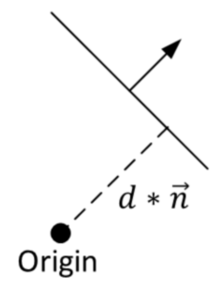
To draw a ray we just need a line with an arrow head. The easiest way to draw an arrow head is to draw a disc (with a normal in the direction of the ray) and connect it with 4 lines to the tip of the ray. To make the ray look nice, we can determine how far back to position the disc based upon the radius of the disc. A good amount is to position the disc at least twice as far back as the radius of the disc.

Debug Drawing: Plane

Vector4 method loses the original point

Remember $d = \vec{n} \cdot \vec{p}_0$

Add a ray to show plane normal

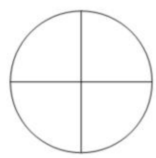


Unfortunately, the Vector4 version of a plane loses the original point, so how can we draw the plane? The simplest method is to compute some other point on the plane. If you remember from the SimpleIntersection slides, $d = \vec{n} \cdot \vec{\mathrm{p}}_0$. This distance is how far the original point was from the origin in the direction of the plane's normal. In other words, $d\vec{n}$ is a point on the plane.

From here you can use GenerateOrthonormalBasis to compute 2 vectors that span the plane in order to draw a quad. To make things even better we can draw a ray at the center of the plane in the direction of the normal.

Debug Drawing: Sphere

Construct 3 discs: x-axis, y-axis, z-axis



Not easily visible at certain angles

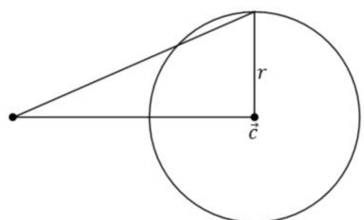
Draw the horizon disc of the sphere

The simple way to draw a sphere is to draw 3 discs centered at the sphere’s center, 1 for each of the world basis vectors.
This will unfortunately not look the best at certain angles as there won’t always be an outline of a circle visible to the camera.
The solution to this is to draw the horizon disc of the sphere.

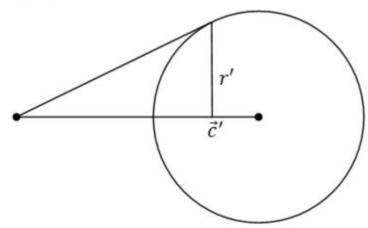
Debug Drawing: Horizon Disc

Idea: Draw a view aligned disc!

Unfortunately this is harder than it sounds.



Incorrect Horizon Disc



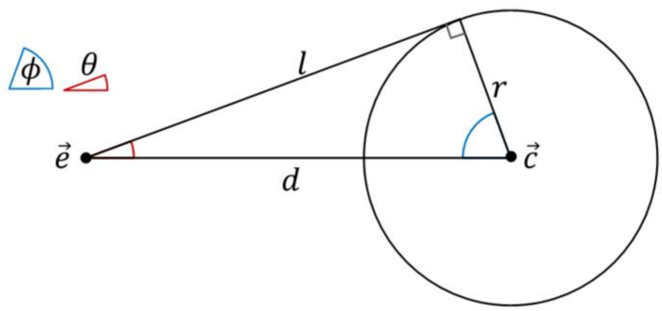
Correct Horizon Disc

Horizon disc should project to the sphere’s silhouette

Drawing a “horizon disc” for a sphere fixes visibility at odd angles. The idea of the horizon disc is to draw a disc that perfectly projects to the sphere’s silhouette. You might think you can just draw a view-aligned disc of the sphere’s radius but this is incorrect. If you look at the picture on the left you can see that the projection of the view-aligned disc will be smaller than the sphere’s actual silhouette. The actual horizon disc is pictured on the right. To do this we need to compute the new radius and position of the horizon disc.

Debug Drawing: Horizon Disc

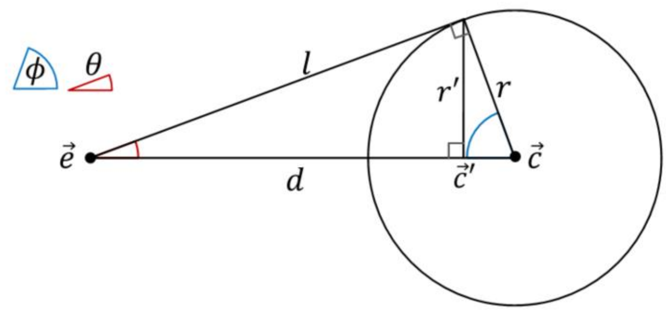
What do we know?



So to start off with, what data do we have? Well we have the sphere radius, the camera's position and the sphere's position. From this we can also compute the distance between the eye and sphere. Finally, since this is a right triangle and we know the fov we can compute all 3 angles of the triangle. With this we could actually compute the remaining side l.

Debug Drawing: Horizon Disc

How do we compute r' and $\vec{c}'$?

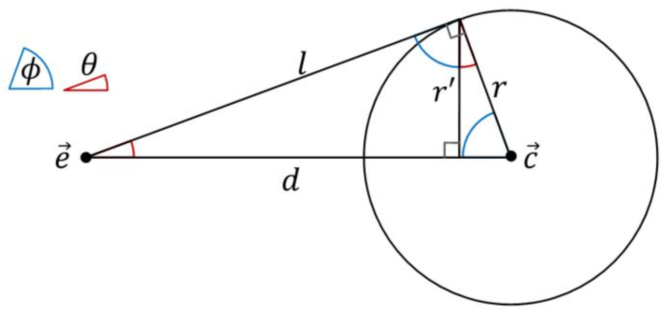


The question then is how do we compute our missing values? It's probably not obvious how, but it is doable.

Debug Drawing: Horizon Disc

Note that the two triangles are similar triangles.

We can now solve for the missing side lengths.



Knowing:

$$d = |\vec{c} - \vec{e}|$$

$$l = \sqrt{d^2 - r^2}$$

$$\cos(\phi) = \frac{r'}{l}$$

$$\cos(\phi) = \frac{r}{d}$$

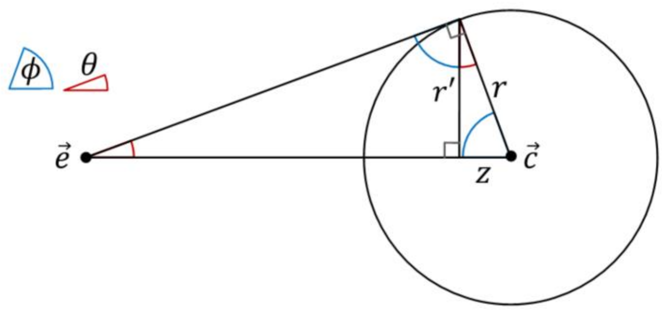
Solve:

$$r' = \frac{r * l}{d}$$

The important thing to realizes is that there is a similar triangle to our original triangle (the one we knew everything about). In particular, this is the small triangle with r' as one of its sides. With this we can now easily solve for r'.

Debug Drawing: Horizon Disc

Can now easily solve for $\vec{c}'$



$$r^2 = z^2 + r'^2$$

$$\vec{c}' = \vec{c} - z * \frac{\vec{c} - \vec{e}}{|\vec{c} - \vec{e}|}$$

Now it’s easy to solve for $\vec{c}'$. The easiest way to do this is to find the length of the missing side (z) of the smaller triangle. We can then use this length as a scalar the vector from the eye to the sphere center. This effectively is just pushing the center back a small distance to compute the horizon disc’s center. With that we now have all of the information needed to render the horizon disc!

Debug Drawing: Point

Just draw a small sphere

Finally, a point can be drawn by just drawing a small sphere where the point is.