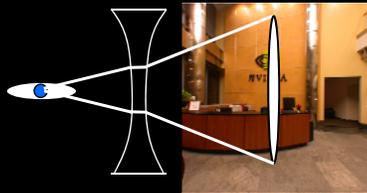


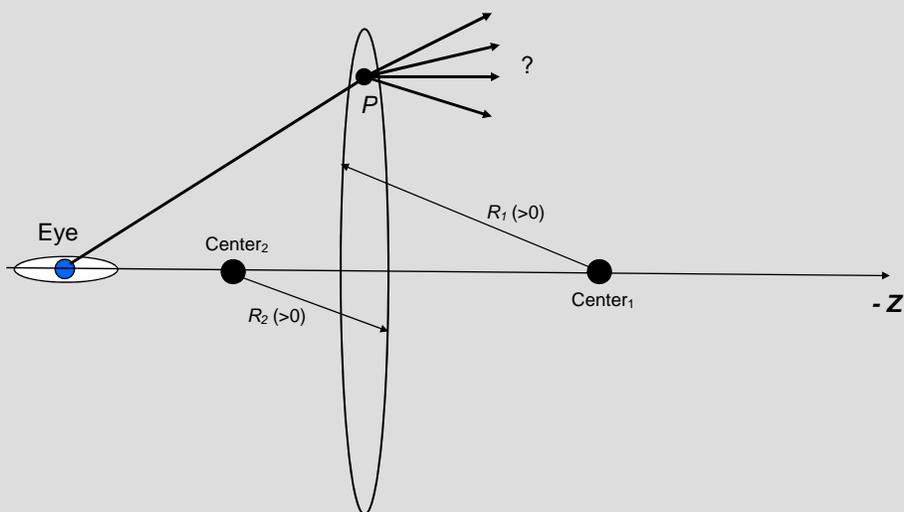
# Creating More Realistic Lens Effects

Mike Bailey

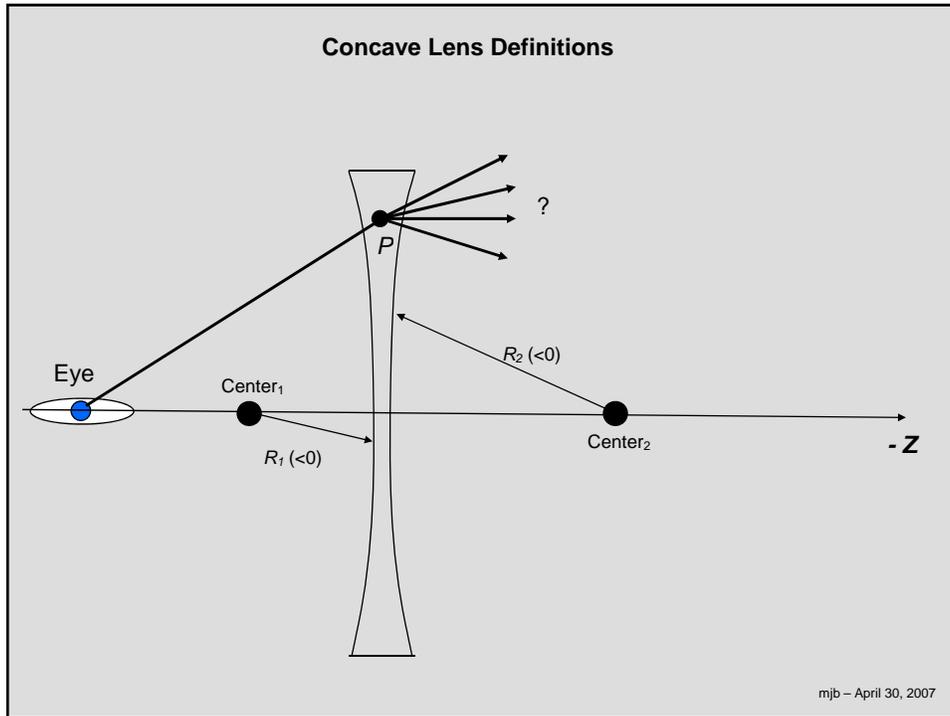
Oregon State University



## Convex Lens Definitions



mjb - April 30, 2007



### lens.vert - Setup for a Cube Map Texture

The diagram shows a lens with an eye on the left. A point  $P$  is on the lens. A vector  $FromEyeToPt$  points from the eye to  $P$ . A normal vector  $Normal1$  is shown at  $P$ . A refracted vector  $RefractVector$  is shown. A second normal vector  $Normal2$  is shown at the point where the ray hits the back surface of the lens. The lens is labeled "zero thickness". The optical axis is labeled  $-Z$ .

```

varying vec3 RefractVector;
uniform float R1, R2;

...
const float ETA_RATIO = 0.66;
vec3 P = vec3( gl_ModelViewMatrix * gl_Vertex );
vec3 Eye = vec3( 0., 0., 0. );

vec3 FromEyeToPt = normalize( P - Eye ); // vector from eye to pt
vec3 Center1 = vec3( 0., 0., P.z - R1 );
vec3 Normal1;
if( R1 >= 0. )
    Normal1 = normalize( P - Center1 );
else
    Normal1 = normalize( Center1 - P );

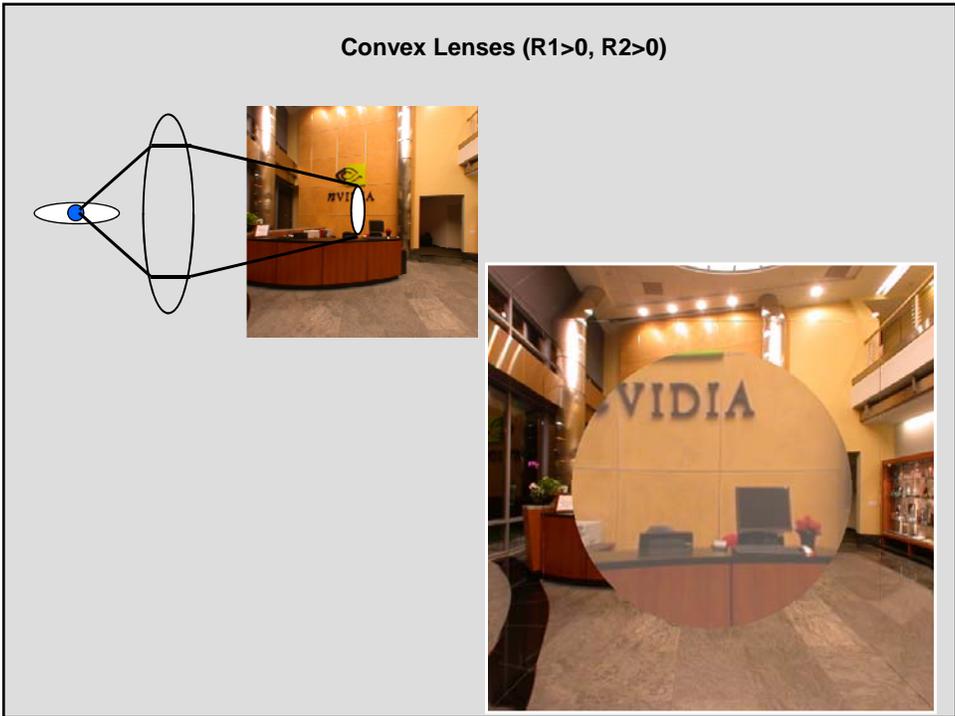
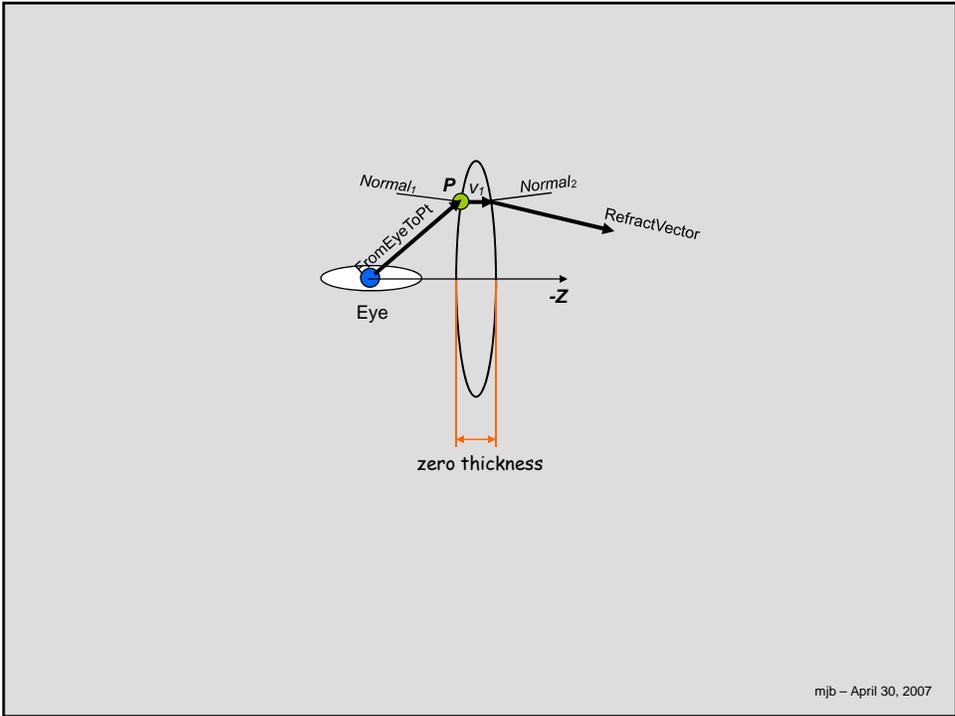
vec3 v1 = refract( FromEyeToPt, Normal1, ETA_RATIO ); // eta ratio = in/out
v1 = normalize( v1 );
vec3 Center2 = vec3( 0., 0., P.z + R2 );
vec3 Normal2;
if( R2 >= 0. )
    Normal2 = normalize( Center2 - P );
else
    Normal2 = normalize( P - Center2 );

RefractVector = refract( v1, Normal2, 1/ETA_RATIO );

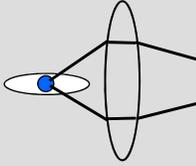
gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;

```

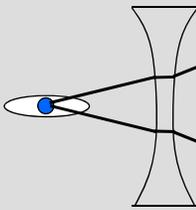
mjb - April 30, 2007



**Convex Lenses ( $R_1 > 0, R_2 > 0$ )**



**Concave Lenses ( $R_1 < 0, R_2 < 0$ )**





**Convex**



**Concave**

mjb - April 30, 2007