Creating More Realistic Lens Effects

Mike Bailey
mjb@cs.oregonstate.edu
Oregon State University

Convex Lens Definitions

Eye
Center₂

Center₁

R₁ (>0)
R₂ (>0)

- Z
out vec3 vRefractVector;
uniform float uR1, uR2;
const float ETA = 0.66;
const vec3 EYE = vec3( 0., 0., 0. );
.
vec3 P = vec3( gl_ModelViewMatrix * gl_Vertex );
vec3 FromEyeToPt = normalize( P - EYE  ); // vector from eye to pt
vec3 Center1 = vec3( 0., 0., P.z - uR1 );
vec3 Normal1;
if( uR1 >= 0. )
    Normal1 = normalize( P - Center1 );
else
    Normal1 = normalize( Center1 - P );
vec3 v1 = refract( FromEyeToPt, Normal1, ETA );   // eta = in/out
vec3 Center2 = vec3( 0., 0., P.z + uR2 );
vec3 Normal2;
if( uR2 >= 0. )
    Normal2 = normalize( Center2 - P );
else
    Normal2 = normalize( P - Center2 );
RefractVector = refract( v1, Normal2, 1./ETA );  // 1./eta = out/in
gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
Convex Lenses (R1>0, R2>0)
Concave Lenses ($R_1 < 0, R_2 < 0$)