Geometric Morphing with the Vertex Shader

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

Morphing a Sphere into a Circle

Blend = 0.00
Blend = 0.25
Blend = 0.50
Blend = 0.75
Blend = 1.00

Morphing a Bunny into a Sphere

Note: the "face" in the spherical bunny is there because the normals were not morphed into sphere normals - they were left as bunny normals.

Morphing a Bunny into a Cube

Note: the "face" in the cube bunny is there because the normals were not morphed into cube normals - they were left as bunny normals.

out vec2 vST;
out float vLightIntensity;
out vec4 vColor;
const float TWOPI = 2.*3.14159265;

// original model coords (sphere):
vec4 vertex0 = aVertex;
vec3 norm0 = aNormal;

// circle coords:
vST = aTexCoord0.st;
float radius = 1. - vST.t;
float theta = TWOPI * vST.s;
vec4 circle = vec4( radius*cos(theta), radius*sin(theta), 0., 1.);

vec3 circlenorm = vec3( 0 0 1 );

vST += vec2( OffsetS, OffsetT );

// blend:
vec4 theVertex = mix( vertex0, circle, Blend );
vec3 theNormal = normalize( mix( norm0, circlenorm, Blend ) );

// do the lighting:
vec3 tnorm = normalize( vec3( uNormalMatrix * theNormal ) );
vec3 LightPos = vec3( 5., 10., 10. );
vec3 ECposition = vec3( uModelViewMatrix * theVertex );
vLightIntensity = abs( dot( normalize(LightPos - ECposition), tnorm ) );
if( vLightIntensity < 0.2 ) vLightIntensity = 0.2;

vColor = aColor;

geometric morphing with the vertex shader