textures2D[1] : register(t0);

uniform SamplerState samplers2D[1] : register(s0);

#ifdef ANGLE\_ENABLE\_LOOP\_FLATTEN

#define LOOP [loop]

#define FLATTEN [flatten]

#else

#define LOOP

#define FLATTEN

#endif

// Varyings

static float4 \_vcolor\_Stage0 = {0, 0, 0, 0};

static float2 \_vTransformedCoords\_0\_Stage0 = {0, 0};

static float4 gl\_Color[1] =

{

float4(0, 0, 0, 0)

};

static float4 gl\_FragCoord = float4(0, 0, 0, 0);

cbuffer DriverConstants : register(b1)

{

float4 dx\_ViewCoords : packoffset(c1);

float3 dx\_DepthFront : packoffset(c2);

float2 dx\_ViewScale : packoffset(c3);

struct SamplerMetadata

{

int baseLevel;

int internalFormatBits;

int wrapModes;

int padding;

};

SamplerMetadata samplerMetadata[1] : packoffset(c4);

};

#define GL\_USES\_FRAG\_COLOR

float4 gl\_texture2D(uint samplerIndex, float2 t, float bias)

{

return textures2D[samplerIndex].SampleBias(samplers2D[samplerIndex], float2(t.x, t.y), bias);

}

#define GL\_USES\_FRAG\_COORD

void gl\_main()

{

float4 \_outputColor\_Stage0 = {0, 0, 0, 0};

{

(\_outputColor\_Stage0 = \_vcolor\_Stage0);

}

float4 \_output\_Stage1 = {0, 0, 0, 0};

{

float4 \_child = {0, 0, 0, 0};

{

(\_child = gl\_texture2D(\_uTextureSampler\_0\_Stage1, clamp(\_vTransformedCoords\_0\_Stage0, \_uTexDom\_Stage1\_c0.xy, \_uTexDom\_Stage1\_c0.zw), -0.5));

}

(\_output\_Stage1 = (\_child \* \_outputColor\_Stage0.w));

}

float4 \_output\_Stage2 = {0, 0, 0, 0};

{

float2 \_dxy0 = (\_uinnerRect\_Stage2.xy - gl\_FragCoord.xy);

float2 \_dxy1 = (gl\_FragCoord.xy - \_uinnerRect\_Stage2.zw);

float2 \_dxy = max(max(\_dxy0, \_dxy1), 0.0);

float \_alpha = clamp((\_uradiusPlusHalf\_Stage2.x - length(\_dxy)), 0.0, 1.0);

(\_output\_Stage2 = vec4\_ctor(\_alpha));

}

{

(gl\_Color[0] = (\_output\_Stage1 \* \_output\_Stage2));

}

}

struct PS\_INPUT

{

float4 dx\_Position : SV\_Position;

float4 gl\_Position : TEXCOORD2;

float4 gl\_FragCoord : TEXCOORD3;

float4 v0 : TEXCOORD0;

float2 v1 : TEXCOORD1;

};

@@ PIXEL OUTPUT @@

PS\_OUTPUT main(PS\_INPUT input)

{

float rhw = 1.0 / input.gl\_FragCoord.w;

gl\_FragCoord.x = input.dx\_Position.x;

gl\_FragCoord.y = input.dx\_Position.y;

gl\_FragCoord.z = (input.gl\_FragCoord.z \* rhw) \* dx\_DepthFront.x + dx\_DepthFront.y;

gl\_FragCoord.w = rhw;

\_vcolor\_Stage0 = input.v0;

\_vTransformedCoords\_0\_Stage0 = input.v1.xy;

gl\_main();

return generateOutput();

}