






# <https://www.shadertoy.com/view/Id3Gz2>

Shadertoy

Search...

Browse New Sign In



34.09 60.1 fps 1024 x 576

REC

Snail

Views: 148922, Tags: procedural, 3d, raymarching, distancefield, sdf, snail

Created by iq in 2015-12-24

Procedural modeling and procedural lighting. Shading is a mix of procedural and textures. SDF rendered through raymarching.

Comments (74)

Sign in to post a comment.

eekee, 2021-07-04

Lovely! And I'm all for rendering speed improvements, especially on my old rig. ;)

tz, 2021-06-14

Amazing.

iq, 2021-04-29

I improved the rendering speed by 30% (1.3x) by adding a bounding volume to the snail

Mr\_Gu, 2021-04-16

It looks so beautiful!

oneshade, 2021-01-23

Amazing work!

+ Image

Shader Inputs

```
69 vec2 d20 = b2-b0;
70 vec2 gf = 2.0*(b*d21+d*d10+a*d20); gf = vec2(gf.y,-gf.x);
71 vec2 pp = -f*gf/dot(gf,gf);
72 vec2 d0p = b0-pp;
73 float ap = det(d0p,d20);
74 float bp = 2.0*det(d10,d0p);
75 float t = clamp((ap+bp)/(2.0*a+b+d), 0.0, 1.0);
76 return vec3( mix(mix(b0,b1,t), mix(b1,b2,t),t), t );
77 }
78
79 vec4 sdBezier( vec3 a, vec3 b, vec3 c, vec3 p )
80 {
81     vec3 w = normalize( cross( c-b, a-b ) );
82     vec3 u = normalize( c-b );
83     vec3 v = cross( w, u );
84
85     vec2 a2 = vec2( dot(a-b,u), dot(a-b,v) );
86     vec2 b2 = vec2( 0.0 );
87     vec2 c2 = vec2( dot(c-b,u), dot(c-b,v) );
88     vec3 p3 = vec3( dot(p-b,u), dot(p-b,v), dot(p-b,w) );
89
90     vec3 cp = getClosest( a2-p3.xy, b2-p3.xy, c2-p3.xy );
91
92     return vec4( sqrt(dot(cp.xy,cp.xy)+p3.z*p3.z), cp.z, length(cp.xy), p3.z );
93 }
94
95 // http://iquilezles.org/www/articles/smin/smin.htm
96 float smin( float a, float b, float k )
97 {
98     float h = max(k-abs(a-b),0.0);
99     return min(a, b) - h*h*0.25/k;
100 }
101
102 // http://iquilezles.org/www/articles/smin/smin.htm
103 float smax( float a, float b, float k )
104 {
105     float h = max(k-abs(a-b),0.0);
106     return max(a, b) + h*h*0.25/k;
107 }
108
109 // http://iquilezles.org/www/articles/smin/smin.htm
110 vec3 smax( vec3 a, vec3 b, float k )
111 {
112     vec3 h = max(k-abs(a-b),0.0);
113     return max(a, b) + h*h*0.25/k;
114 }
115
116 //-----
117
118 float hash1( float n )
119 {
120     return fract(sin(n)*43758.5453123);
121 }
122
```

Compiled in 0.0 secs 15627 chars