

IniSphere 0.2.4  
by inimArt.com

Quick help guide

## iniSphere 0.2.4

iniSphere is a Quartz Composer custom patch that renders a sphere with some particular effects, ideals for VJ and realtime performances, ideals for VJ and realtime performances.

**more info:**

[inimart.com](http://inimart.com)

[Luca Palmili \(author\)](#)

**inimart**

## Sphere creation

Vertices, normals and texture coordinates are calculated every time you change the parameters for the resolution of the sphere or its opening. It is possible to render the sphere also in wireframe or in point mode. The sphere is created in „segments“, the opening of the sphere determines the rotation of the segments around one of the two ends. The vertices are sent to the GPU using „immediate mode“ GL\_QUAD\_STRIP: in later versions a VBO will be used, so to optimize the dialogue with the GPU.

# Plugin inputs

## SPHERE DIVISIONS

### HDiv

Sphere horizontal resolution

#### fromH ... toH

Draws horizontal sphere slices from fromH to toH.

### VDiv

Sphere vertical resolution

#### fromV ... toV

Draws vertical sphere slices from fromV to toV.

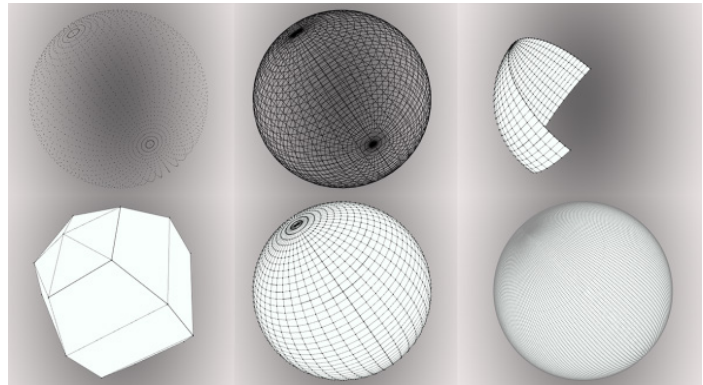
### opening

The opening of the sphere. If this value is 0, the sphere will be a unique object. As soon as this value became > 0, the sphere is rendered as a sequence of slices.

## SPHERE COMPONENTS

### drawFlat/Lines/Points

If **drawFlat** is enabled, the sphere is drawn using quads. If you want to add also wireframe mode or draw points on the vertices, turn on **drawPoints**.



Playing with sphere divisions (HDiv and VDiv) values.

## LIGHT

### useLight

If you want to use a light source.

- 0 don't use light (use a flat **diffuseCol** colour)
- 1 use light: The light source position is in (lightPosX, lightPosY, lightPosZ), has shininess **lighShininess**, specular colour **specCol**, and diffuse color **diffuseCol**.

### blending

Change OpenGL BlendFunc:

- 0 glBlendFunc (GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA);
- 1 glBlendFunc(GL\_ONE, GL\_ONE\_MINUS\_SRC\_COLOR);
- 2 glBlendFunc(GL\_DST\_COLOR, GL\_ONE\_MINUS\_SRC\_ALPHA);

# plugin inputs

## NOISE FUNCTIONS

### UseVNoise

It generates vertices perlin noise displacement, using VN\_ScaleOut / VN\_ScaleIn / VN\_Off parameters.

### UseENNoise (erosion noise)

It generates an additional perlin noise value EN\_v for each vertex, using EN\_ScaleOut / EN\_ScaleIn / EN\_Off parameters.

### EN\_Discard

For each vertex, if the generated EN\_v value is less than EN\_Tresh, the pixel is discarded from the pixelshader.

For example, if you use these values you can see a nice erosion effect (see the image):

*EN\_Discard* ON

*useENNoiseCol* ON

*EN\_ScaleOut* 10

*EN\_ScaleIn* 5

*EN\_off*: LFO with Amplitude 0.2 and offset 0.1

*EN\_Treshold* 6

### useENHeightMap

see HEIGHT MAP DISPLACEMENT

### useENNoiseCol

Transform EN\_v value assigned to each vertex into a colour.

### useTextureMap

If checked, you must provide an image in textureMap input AND in heightMap input (even if you don't use the heightMap), otherwise the sphere will be black.

## HEIGHT MAP DISPLACEMENT

iniSphere can displace his vertices using an heightMap as input.

### useENHeightmap

Use NE\_v generated from the erosion noise as height map. DOESN'T WORK IF UseHeightMap is turned on.

### useHeightMap

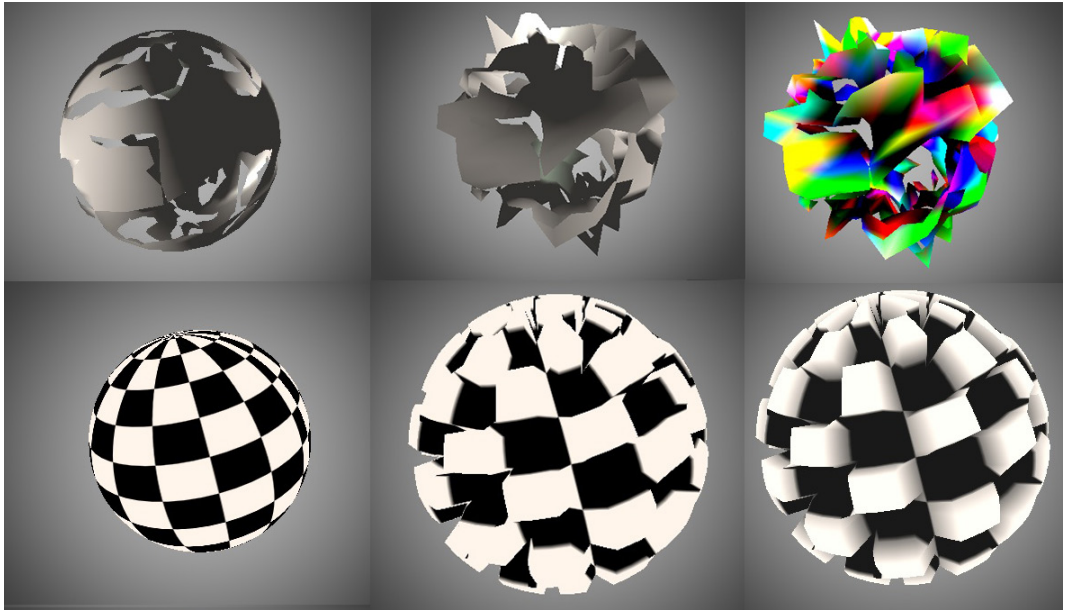
use heightmap input as displace map. The heightmap should be a grayscale image (light values extrude vertices). If you want to use the erosion noise as displace map, you have to turn off this value.

### displaceOffset

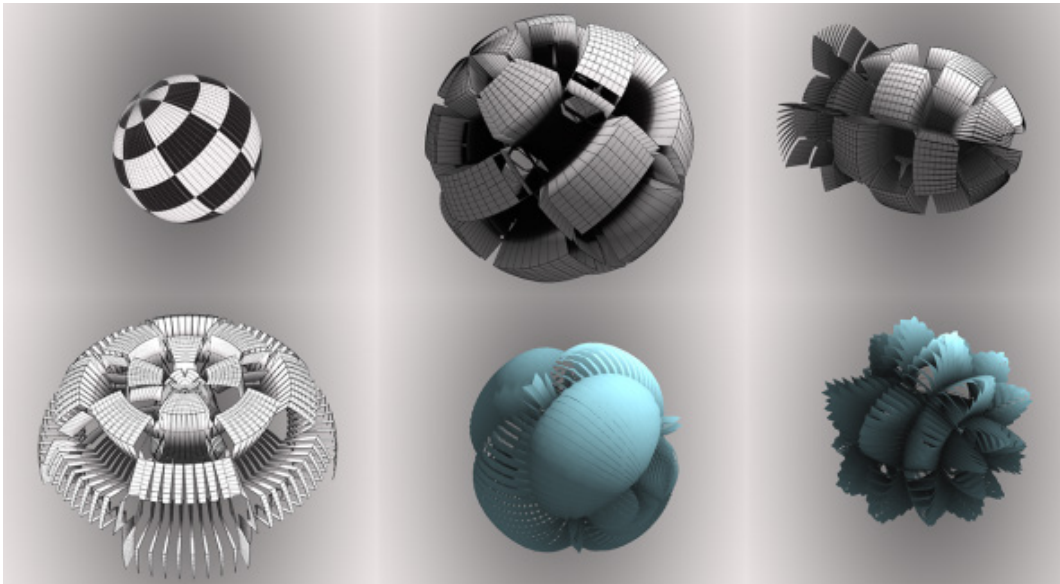
displacement value: it must be > 0 if you want to see some displacement.

### useDLighting

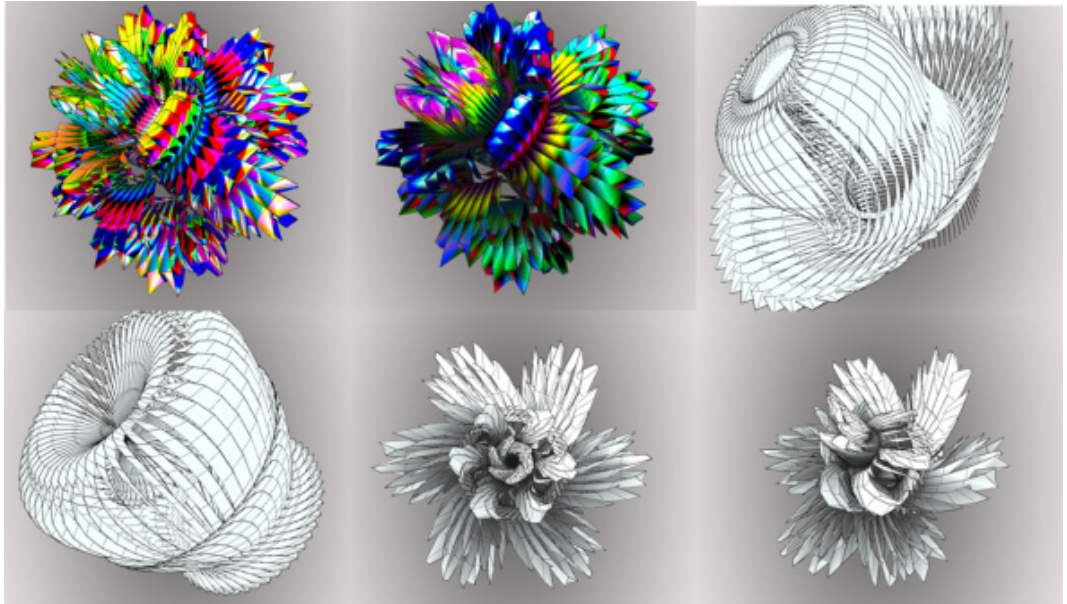
this kind of illumination helps to highlight the displacement extrusion (see the image), simulating a nice ambient occlusion effect. When use this kind of illumination, you must also use lightingMul value to adjust the intensity of the effect (if lightingMul is 0, the sphere will be black)



From left to right, top to bottom. Erosion noise: use EN\_Tresh value to change the amount of discarded pixels // Erosion noise + vertex noise // Erosion noise + vertex noise + color noise // Simple texturing with a chessboard pattern // Use of the chessboard pattern both for texturing and for displacement // Adding useDLighting flag to add kind of ambient occlusion effect to the displacement.



Checkboard displacement map / other displacement examples with different sphere opening



Vertex color noise and vertex position noise examples