1 OSL

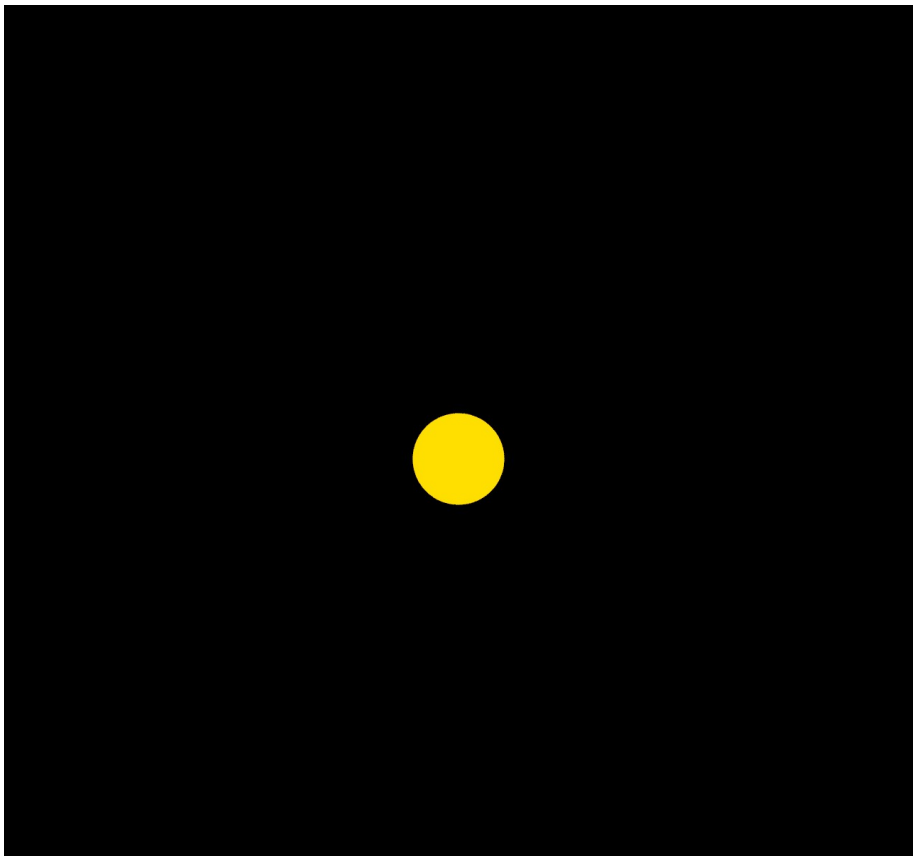
1.1 Circle shader (OSL) - dot.osl

Switch to Shading panel and create a plane and add a new material on it with a simple emit shader.
Add Script nodes.
Switch to Sript panel and create a sript named dot.osl
Add this script :

```
shader disk(
output color Factor = 0.5,
float Var_color = 1.0,
float Intensity = 1.0,
float r = 0.1,
float x = 0.0,
float y = 0.0)

{
//float x_weigth = 0.1;
//float y_weigth = 0.1;
//if(p[0] > x && p[0] < x + x_weigth && p[1] > y && p[1] < y + y_weigth) f = 1.
float f = 0.0;
if(((P[0] -x) * (P[0] -x) + (P[1] - y) * (P[1] - y)) - (r * r) < 0) f = 1.0;
float t = Intensity/f;
Factor = color("hsv", Var_color/10, 1, t);
}
```

Switch to shading panel select dot.osl script in the Script node.
Launch render with eevee or cycles.



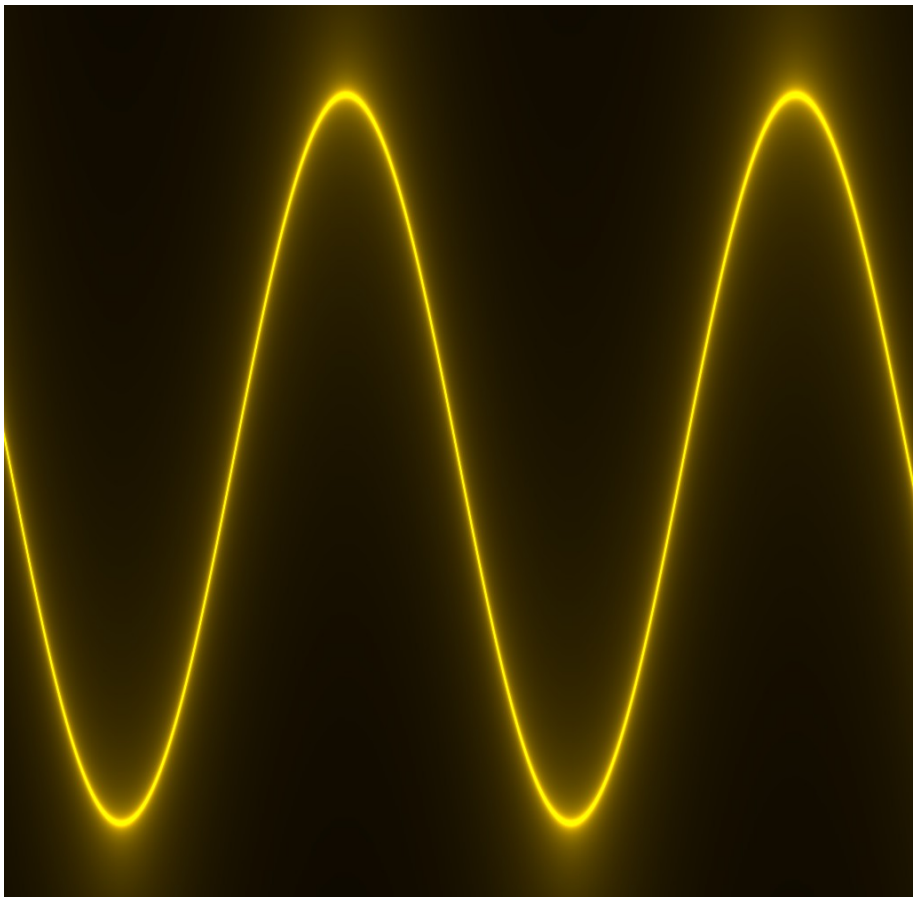
1.2 Sinusoide shader (OSL) - sinusoide.osl

Same method to create the shader and add this new osl script :

```
#include "stdosl.h"

shader pattern(
output color Color = color(0.8),
float Var_color = 1.0,
float Intensity = 0.1,
float Amplitude = 0.8,
float Periode = 0.1,
float X = 0.0,
float Y = 0.0,
point Pos = P)
{
float A = Amplitude;
float T = Periode;

float t = abs(( A * sin ( (1 /(T/2*M_PI)) * (Pos[0] + X) ) + Y) + Pos[1]);
float f = Intensity/t;

Color = color("hsv", Var_color/10, 1, f);
}
```

Switch to shading panel select sinusoide.osl script in the Script node.
Launch render with eevee or cycles.