

Algorithmic art and laser cutter

NTA 404, 2020
Aistis Raudys

Summary

What is Algorithmic art?

Fractals is one example

Generative design, Parametric design,

Laser cutter and Python

Illustrator

Computer graphics

Architecture

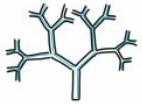
Generative



PHYSICS



CELL AUTOMATON



RECURSIVE



CHAOS



REACTION



SWARM



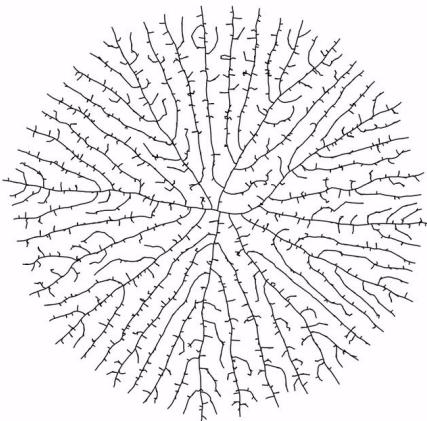
Zaha Hadid / Heydar Aliyev Cultural Center Baku

Lightning

I was not able to find

One in python

FOUND: space colonization algo



Diffusion Limited Algorithm / Brownian Tree

What is popular

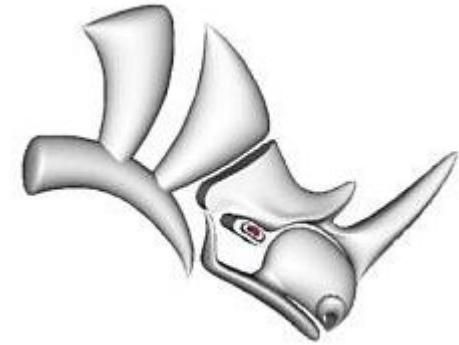
Rhino3d + Grasshopper

Rhinoceros 3D - design software (CAD)

Grasshopper - algorithmic modeling for Rhino

More for 3D

Visual programing - limited if you go very complex



File Edit View Curve Surface Solid Mesh Dimension Transform Tools Analyze Render Panels Paneling Tools V-Ray Help

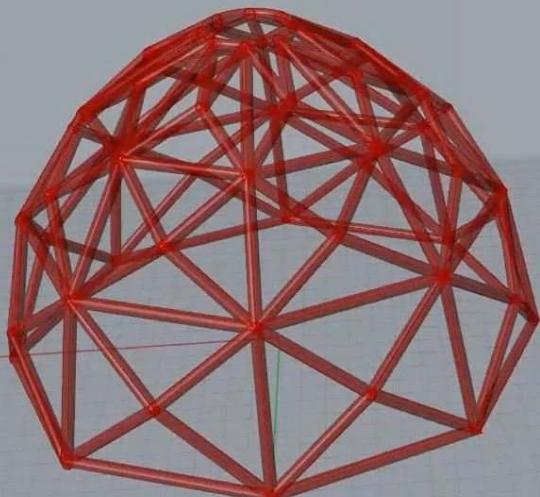
Command: Grasshopper

Command: |

Standard CPlanes Set View Display Select Viewport Layout Visibility Transform Curve Tools Surface Tools Solid Tools Mesh Tools Render Tools Drafting New in V5



Perspective |



Perspective Top Front Right +

Points Curves Surfaces Polysurfaces Meshes Annotations Lights Blocks Control Points Point Clouds Hatches Others Disable

End Near Point Mid Cen Int Perp Tan Quad Knot Vertex Project Disable

CPlane x-176.14 v-45.94 z 0.00 Inches Default Grid Snap Ortho Planar Osnap SmartTrack Gumball Record History Filter Minutes from last save: 46

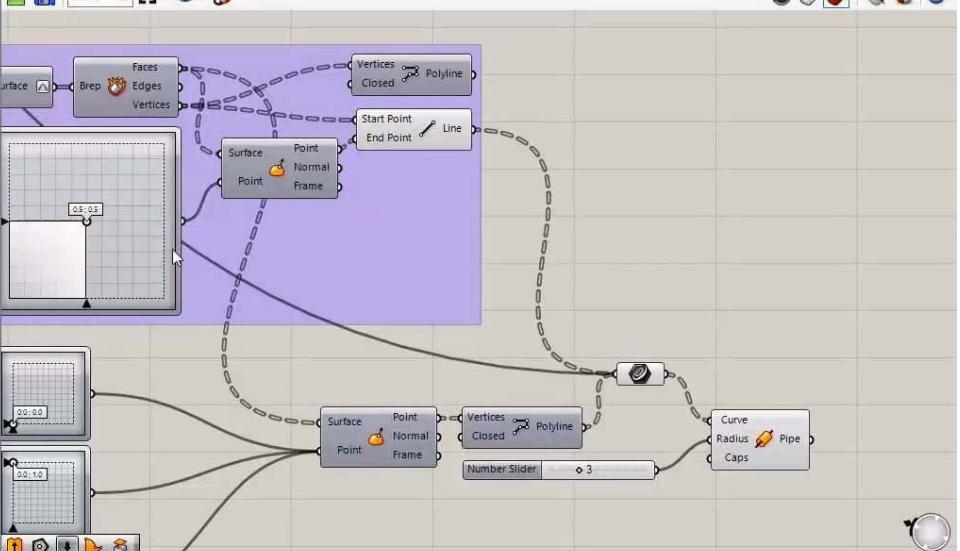
Grasshopper - unnamed

File Edit View Display Solution Help

Params Maths Sets Vector Curve Surface Mesh Intersect Transform Display PanelingTools V-Ray



100%



0.9.0076 ..

Other tools

Sidefx Houdini

More for animation and games

Procedural Lake Houses
by Anastasia Opara



www.anastasiaopara.com



Example of the generated content

Houdini
3D ANIMATION TOOLS

Back to school

- Geometry
- Pythagorean theorem
 - Minimal distance (too fragile | laser cutter)
- Distorted lines
- Smoothing
 - Smooth distorted lines
- How to draw a circle
 - Sinus
 - Cosinus
- Many more

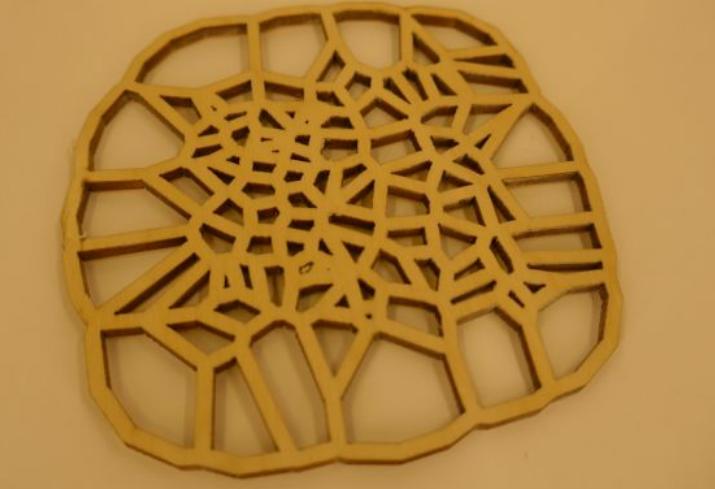
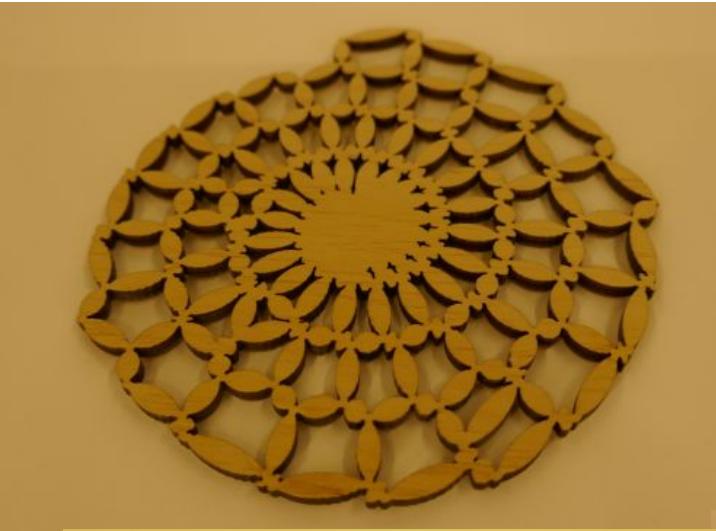
Laser cutter

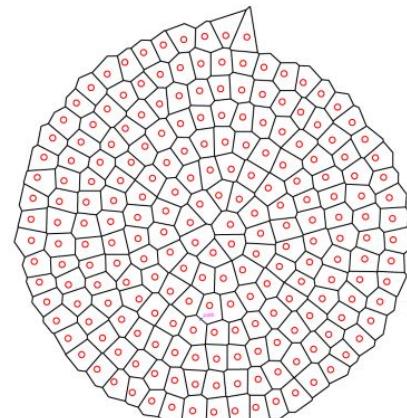
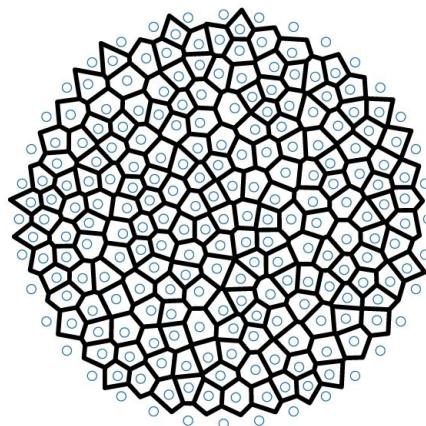
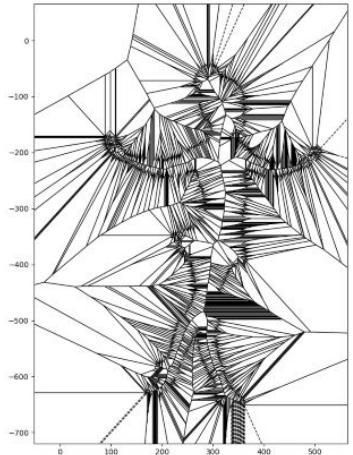
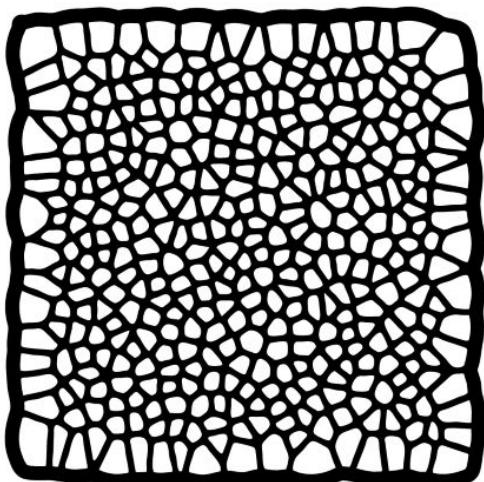
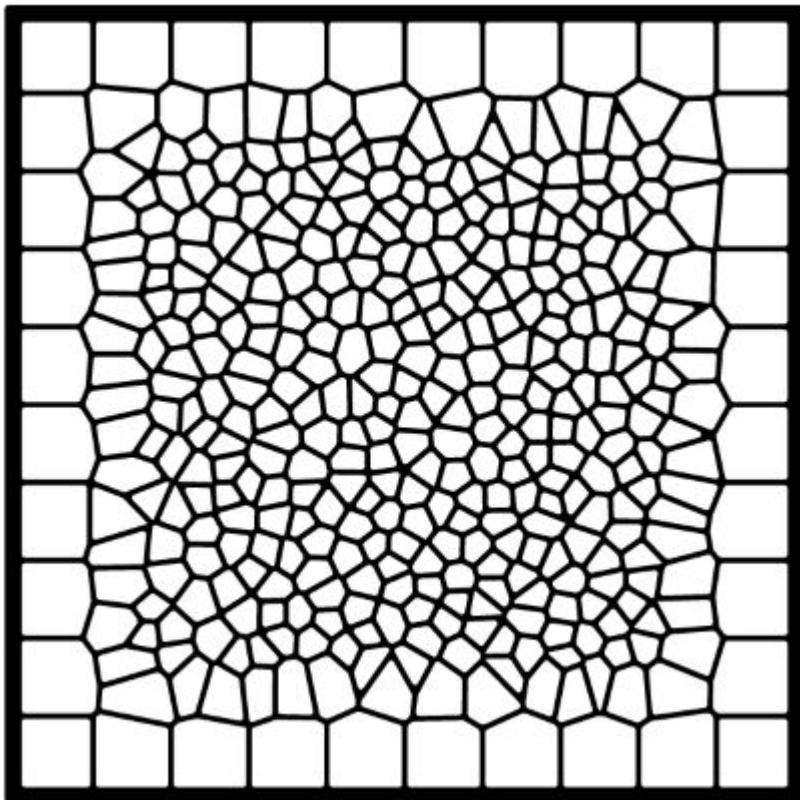
Things fall out

Things are too fragile

Voronoi diagrams

Really fantastic invention





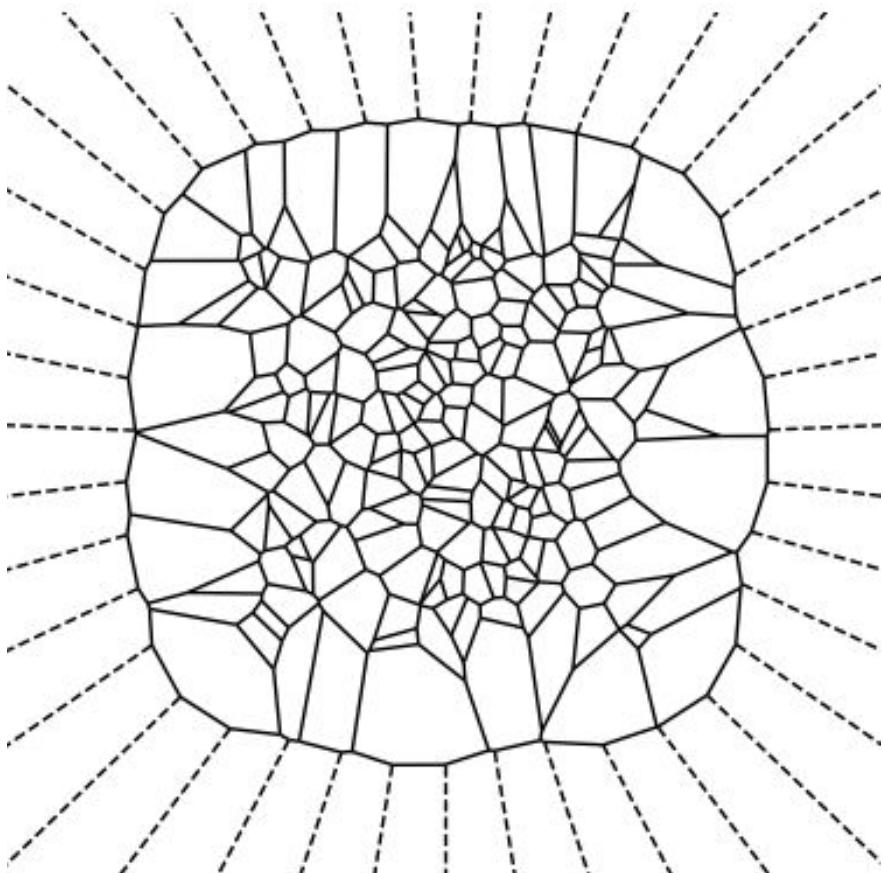
Code

```
import matplotlib.pyplot as plt
import numpy as np
import math
from scipy.spatial import Voronoi, voronoi_plot_2d

# plot circle (-1, 1)
z = np.linspace(0, math.pi * 2, 40)
x = np.sin(z)
y = np.cos(z)
points = np.vstack((x, y))

# plot random dots inside circle
for i in range(200):
    x1 = np.random.rand(1) - 0.5 # (-0.5, 0.5)
    y1 = np.random.rand(1) - 0.5
    points = np.hstack((points, [x1, y1]))

# plot Voronoi using dots
vor = Voronoi(points.T)
voronoi_plot_2d(vor, show_points=False, show_vertices=False, line_width=1)
plt.axis('off')
plt.axis('equal')
plt.xlim(-1, 1)
plt.ylim(-1, 1)
plt.savefig('voronoi_circles.svg')
plt.show()
```



More code

```
#init array
points2 = np.empty((2,0))
points_grid = np.empty((0,2))

size = 5
step = 5
hstep = step / 2

for s1 in np.arange(0,size*step,step):

    temp1, temp2 = distorted.line(s1-hstep, -hstep, s1-hstep, size*step-hstep, 50 * size, 0.1, include_nan=True)
    zzz = np.vstack((temp1, temp2)).T
    points_grid = np.vstack((points_grid, zzz))

    temp1, temp2 = distorted.line(-hstep, s1-hstep, size*step-hstep,s1-hstep, 50 * size, 0.1, include_nan=True)
    zzz = np.vstack((temp1, temp2)).T
    points_grid = np.vstack((points_grid, zzz))

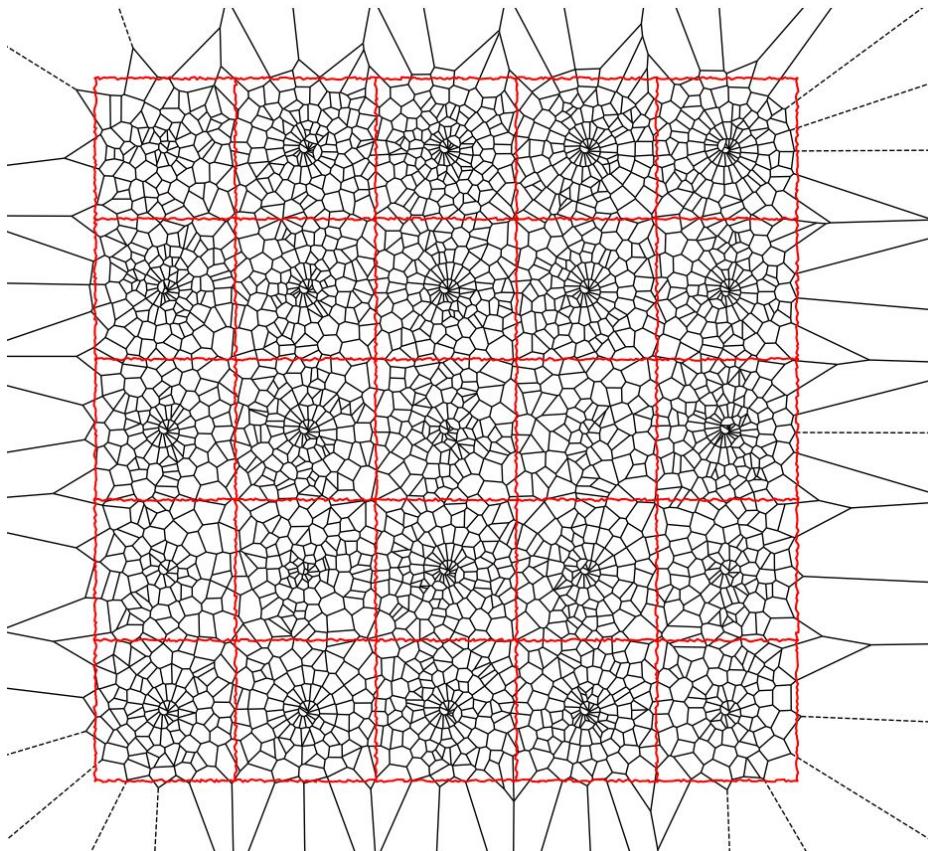
for s2 in np.arange(0, size * step, step):
    points = np.empty((2, 0))
    space_points = np.random.randint(40,110)
    rand_lvl = np.random.rand()*0.2 # 0.1
    z = np.linspace(0,math.pi*2*5,space_points)
    z = z + np.random.rand(z.shape[0])*rand_lvl
    x = np.sin(z)*z/10
    y = np.cos(z)*z/10
    points = np.vstack((x,y))
    rprintsc = np.random.randint(10,40)

    for i in range(rprintsc):
        x1 = np.random.rand(1)*4-2
        y1 = np.random.rand(1)*4-2
        points = np.hstack((points, [x1, y1]))

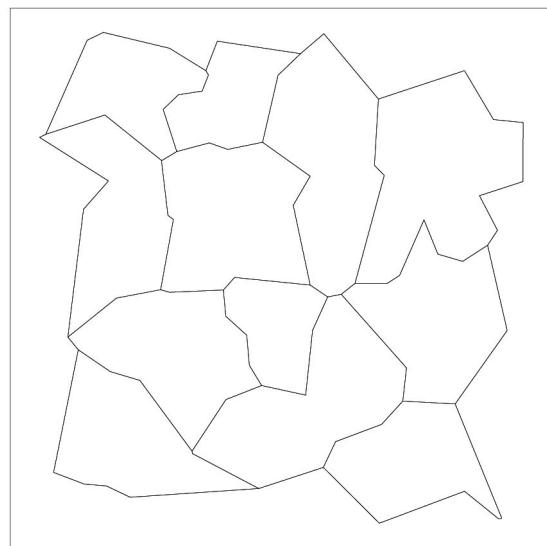
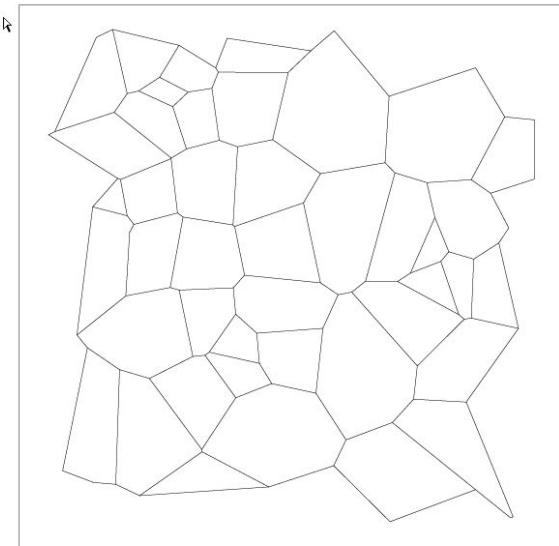
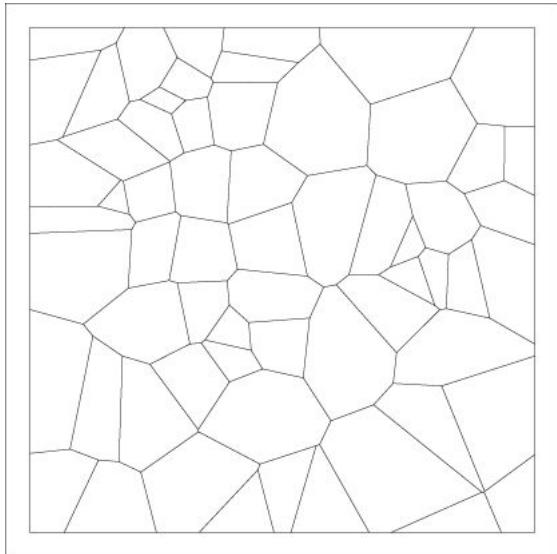
    points[0, :] += s1
    points[1, :] += s2

    points2 = np.hstack((points2, points))

#draw border lines
points_grid = np.vstack((points_grid,
                        distorted.line(size * step - hstep,-hstep,size * step - hstep, size * step - hstep, 50*size, 0.1, 1),
                        distorted.line(-hstep, size * step - hstep, size * step - hstep, size * step - hstep, 50*size, 0.1, 1),
                        ))
```



Puzzle Fails



Mix art + algo

+ Added value :



Transparent



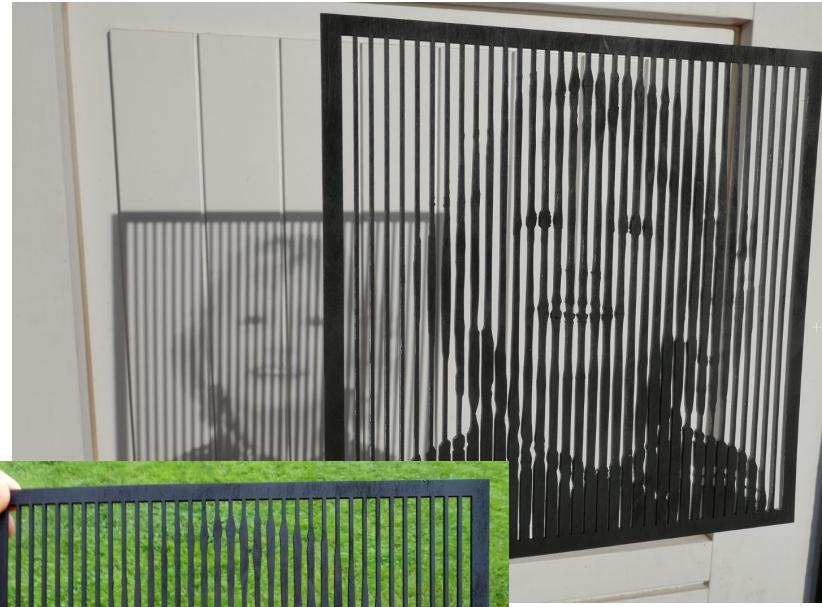
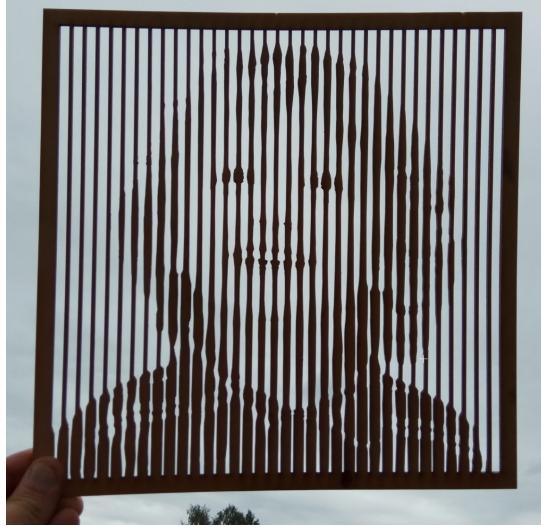
Pictures

Standard photoshop algorithms

Things fall out



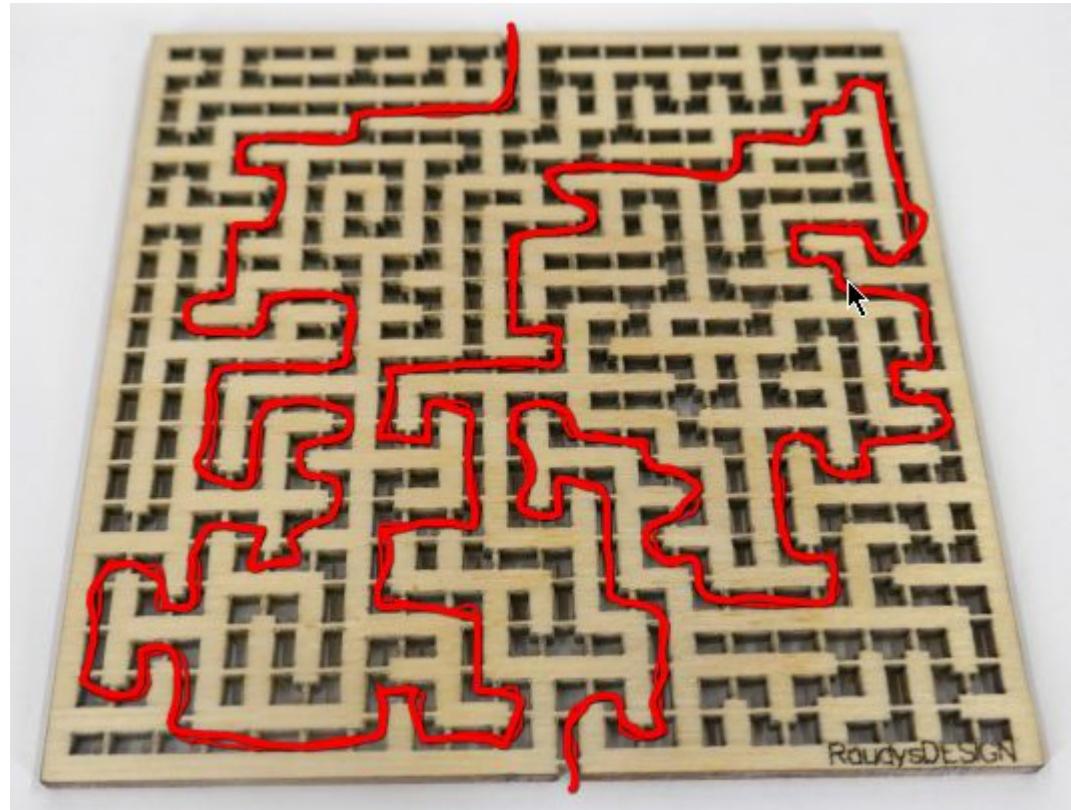
Picture processing



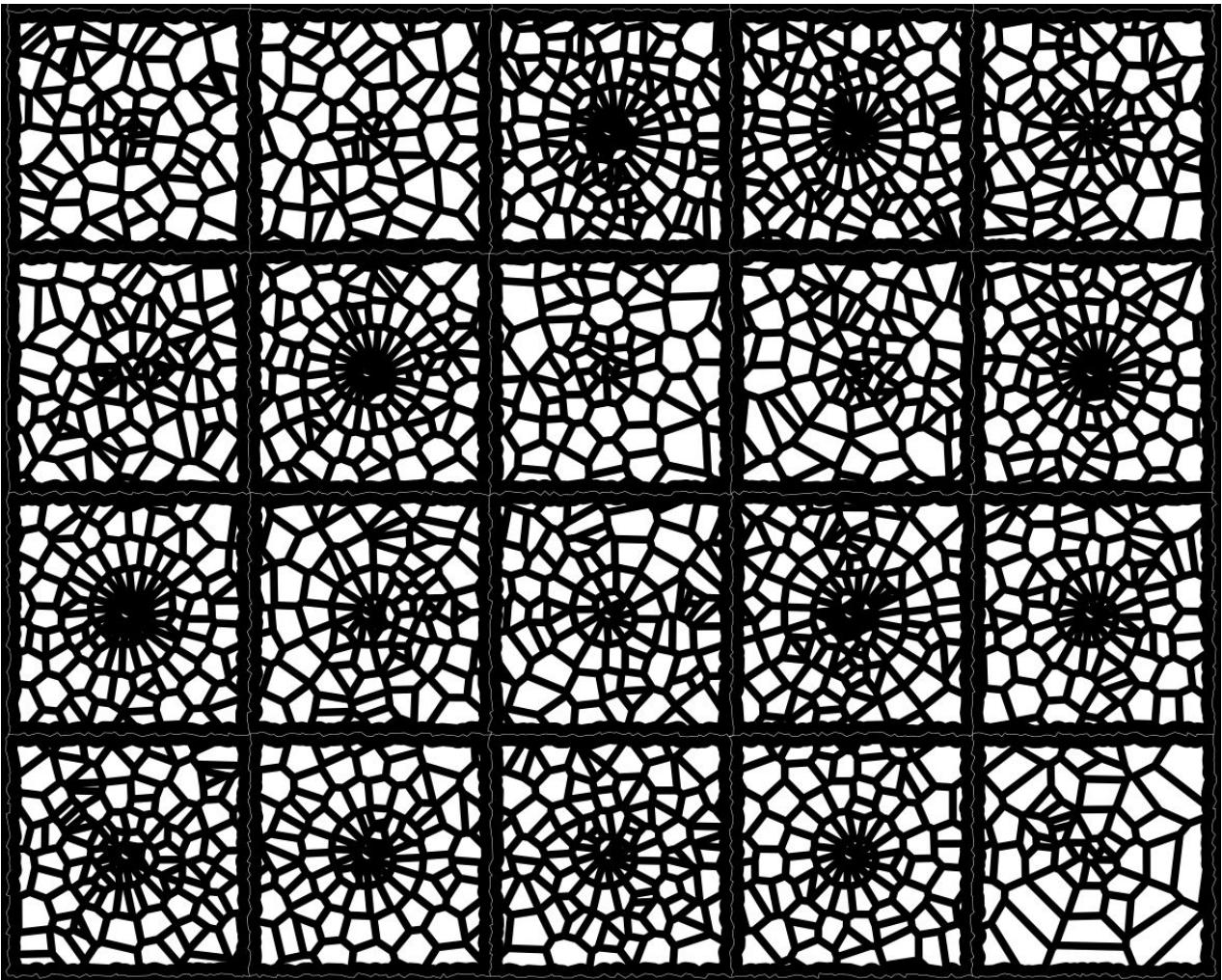
Python

A lot of ready made stuff

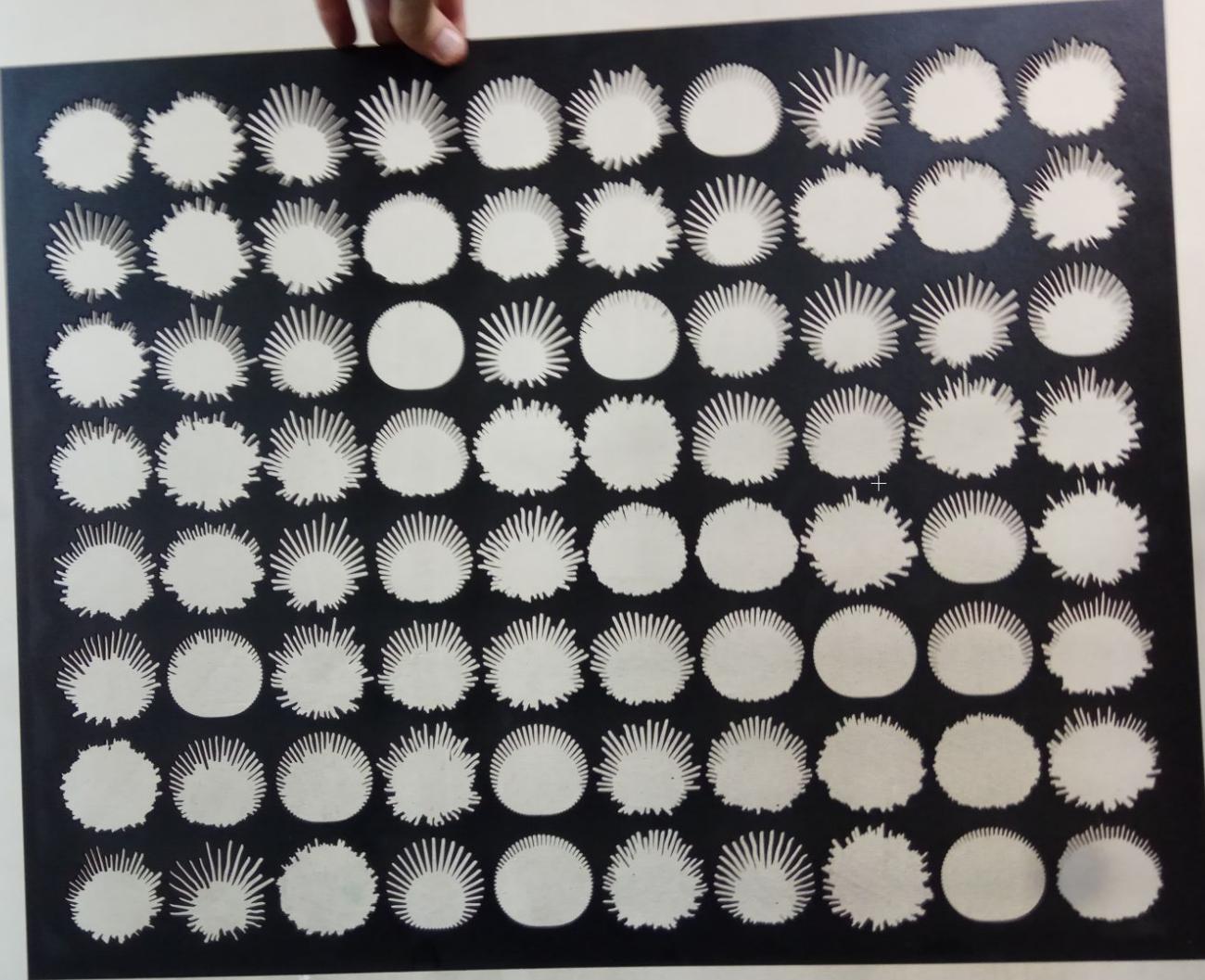
Like maze generators



Pick
What
You
Like



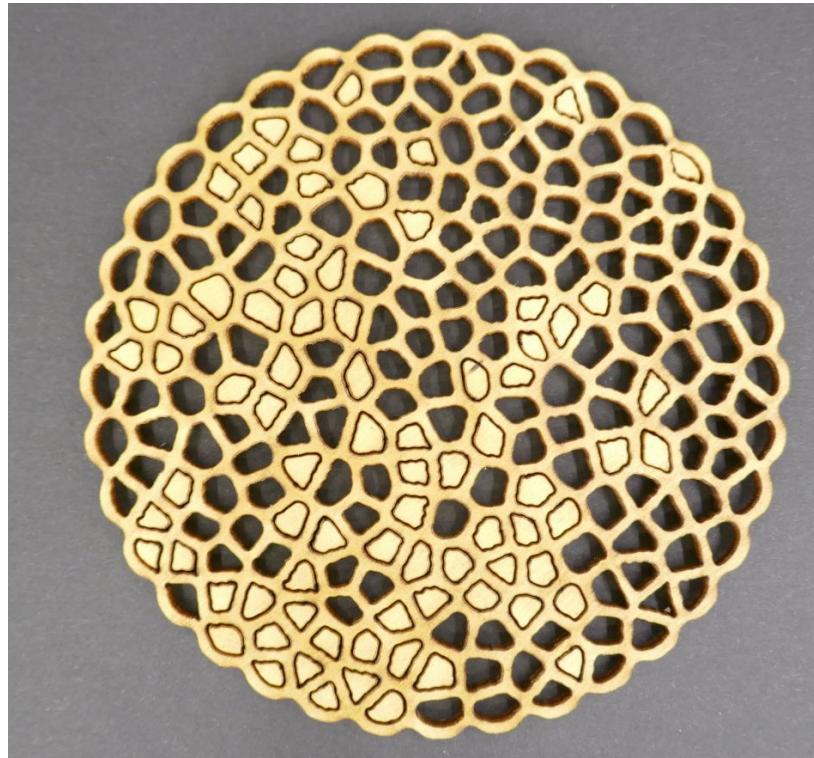
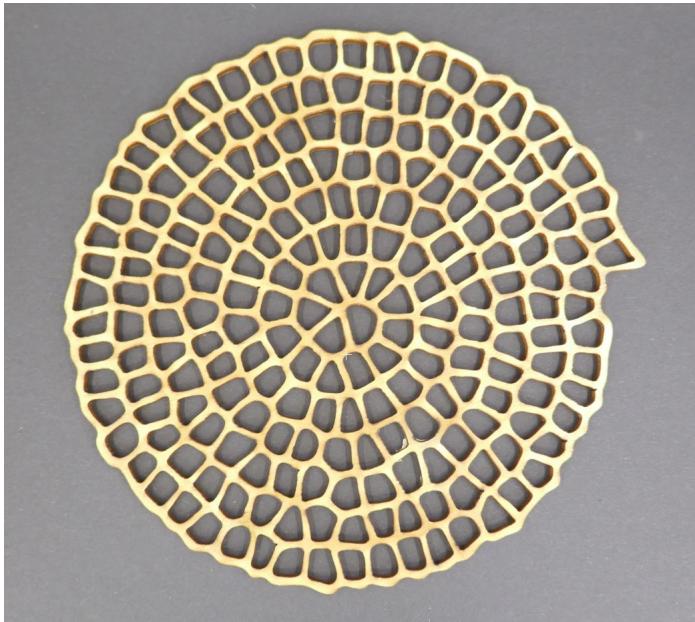
Pick
What
You
Like



Fails -
things
fall
out

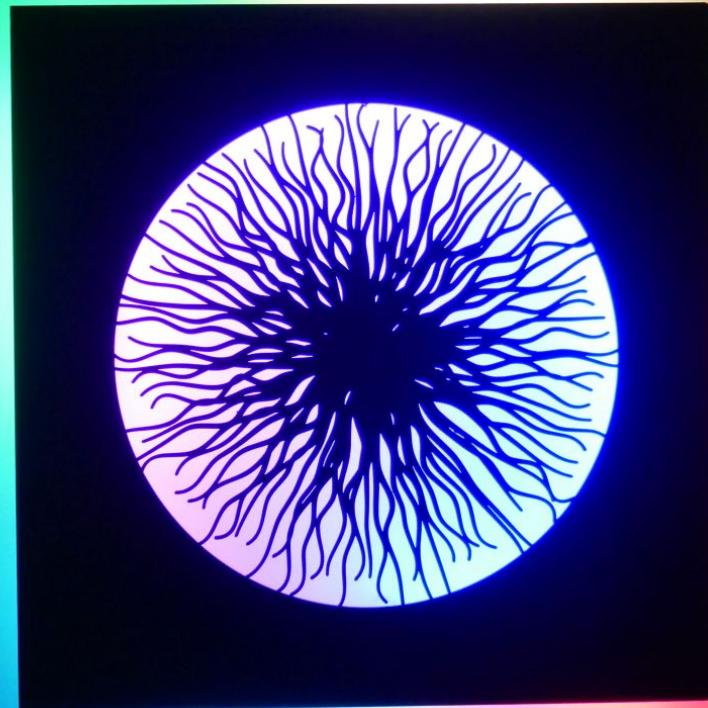


Spiral stuff

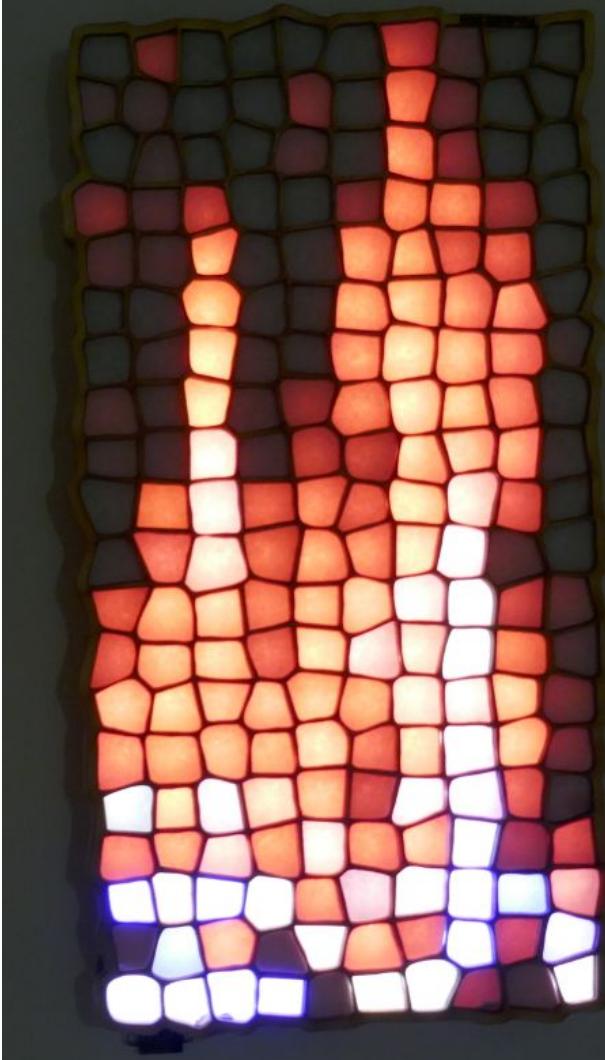
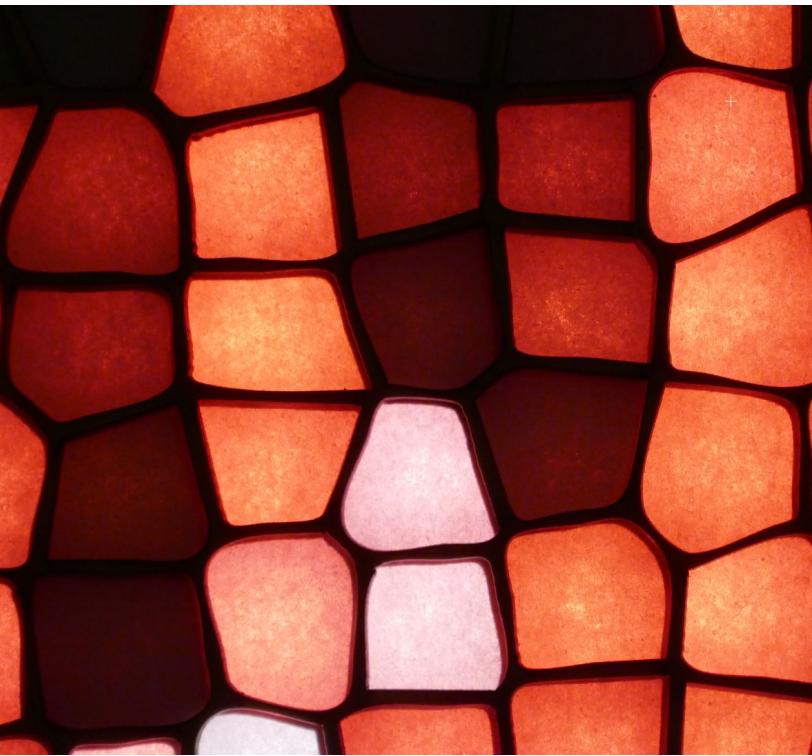


```
f line(x_start, y_start, x_end, y_end, steps=100, dist_level=0.02, include_nan=False, two_d_output=False)
f line_smooth(x_start, y_start, x_end, y_end, steps=100, dist_level=0.02, include_nan=False, two_d_output=False)
f circle(centre_x, centre_y, diameter, steps, distort_level)
f circle2(centre_x, centre_y, diameter, steps, distort_level, two_d_output=False)
f line_end(x_start, y_start, x_end, y_end, steps=100, dist_level=0.02)
f rectangle(xcor1, ycor1, xcor2, ycor2, steps=100, dist_level=0.02, two_d_output=False)
f text(position_x, position_y, text_string, text_size=1)
f save_to_svg_file(data, file_name)
f save_to_svg_file_polygon(data, file_name)
```

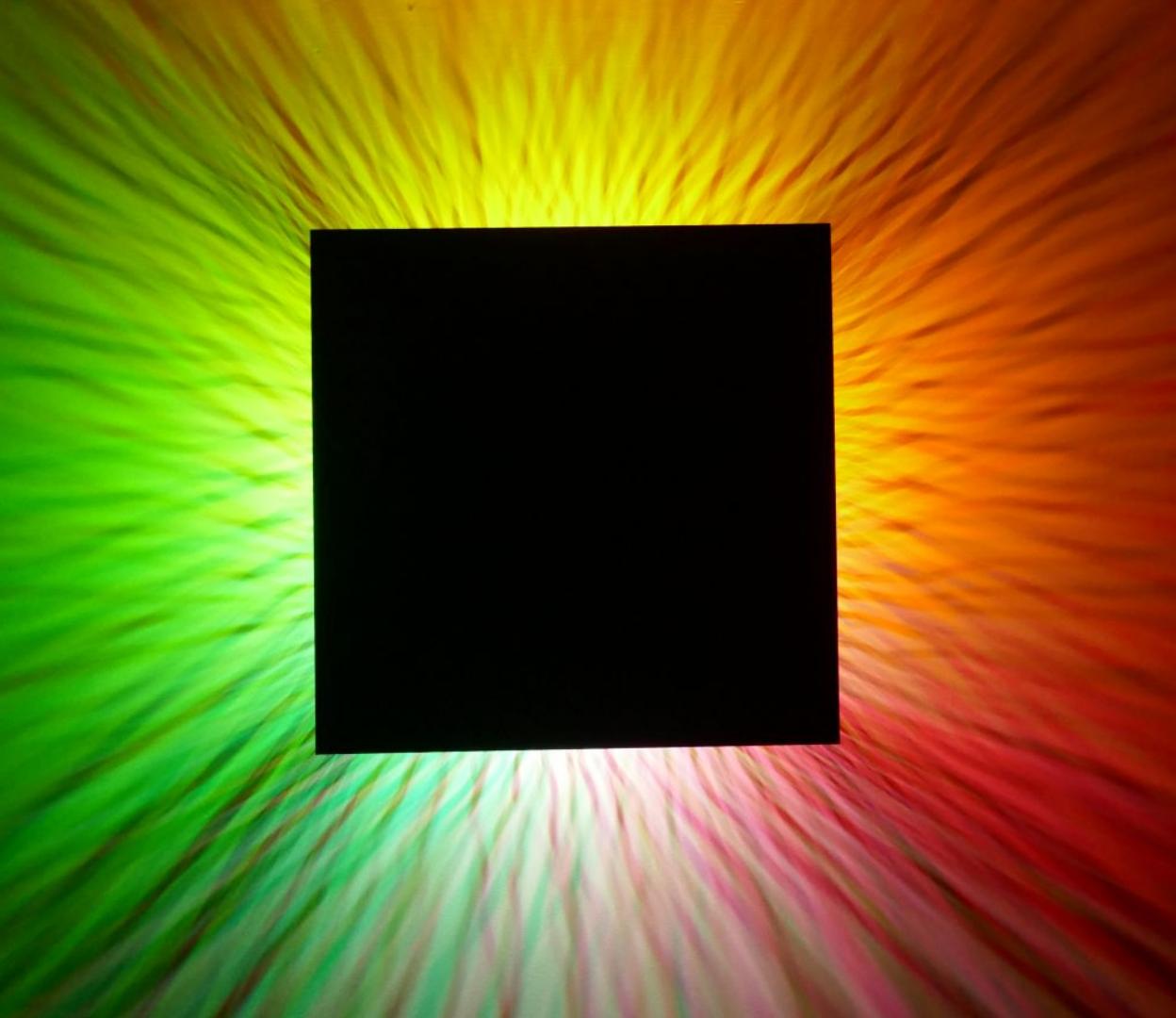
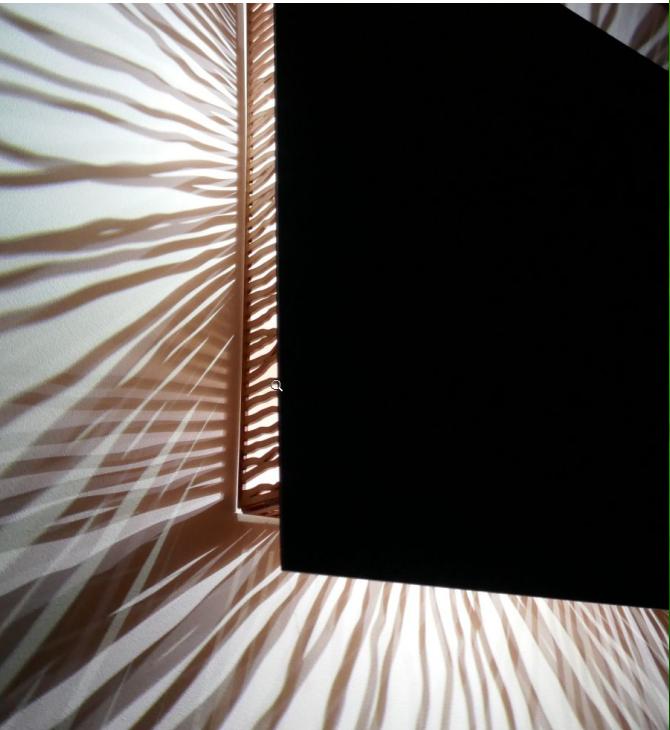
Algo + RGB

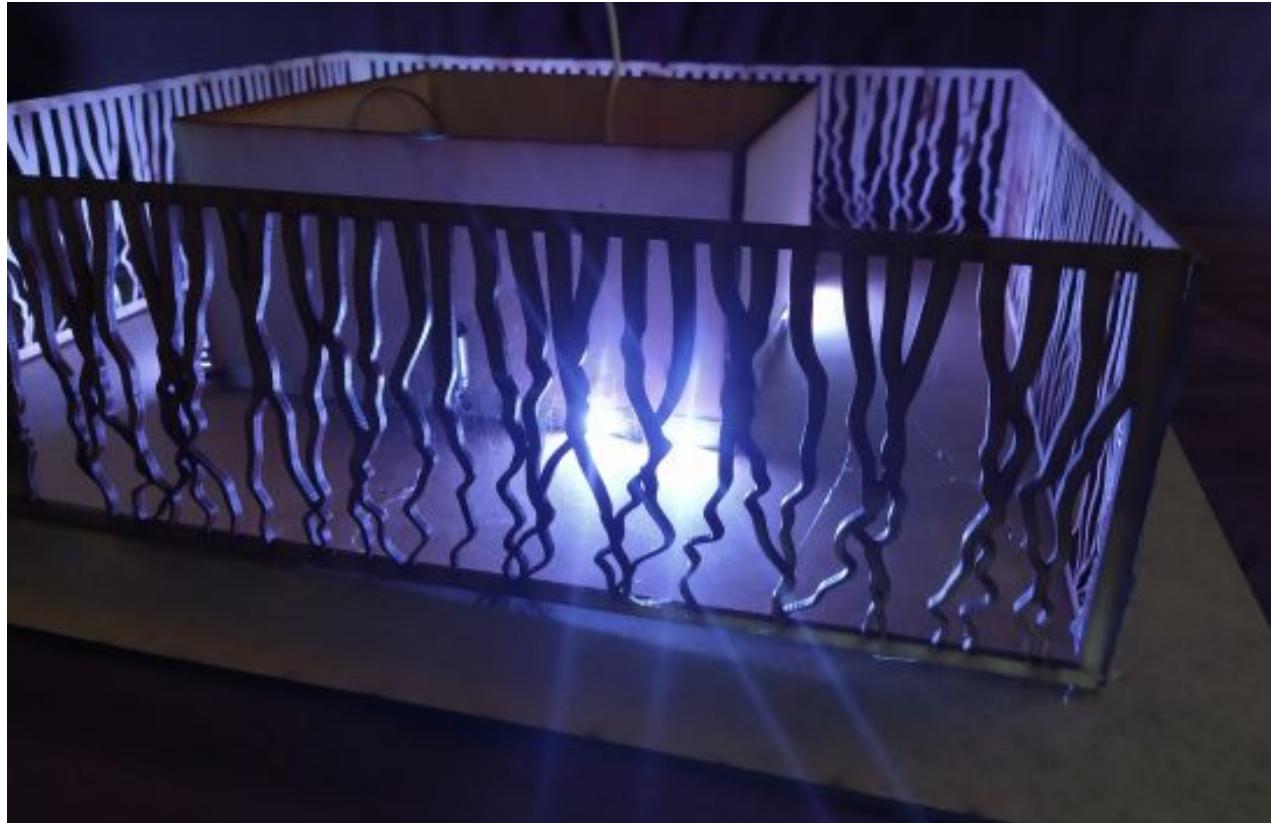


Algo + RGB+
FireAlgo



Shadows





Multi layer

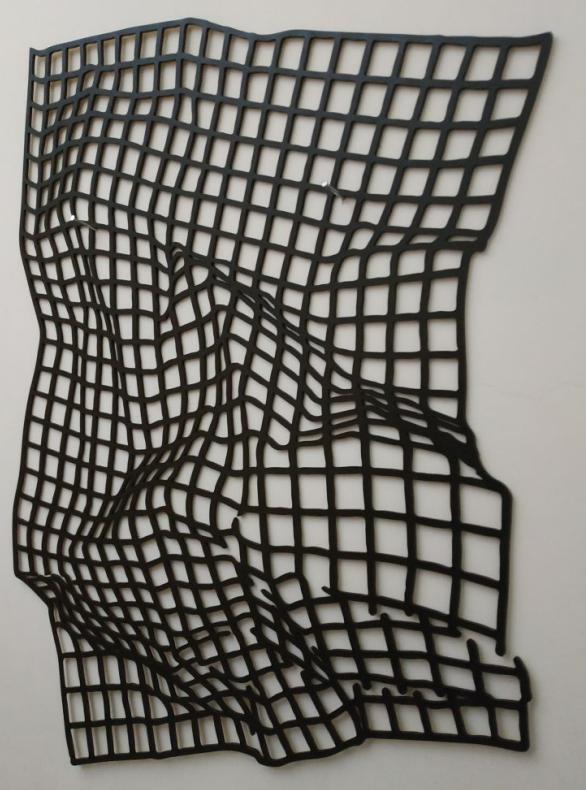


Then you get bored with algorithms

Or Algos

Gets too

Complex





Circle packing

Apollonian gasket

