

production-Copy1

November 21, 2024

```
[395]: import noise

from pdesign import canvas, shapes, lines
from pdesign import transforms as trans
from pdesign import smooth

import numpy as np
from shapely.geometry import MultiLineString, LineString, Point, Polygon,
↳MultiPoint
from shapely.geometry import box as Box
from shapely.ops import unary_union
import matplotlib.pyplot as plt

import importlib
importlib.reload(smooth)

from skimage.draw import line, circle_perimeter
from skimage import draw

[396]: from ipywidgets import widgets
from ipywidgets import interact, interact_manual, interactive

[397]: linewidth = 0.0275591*72
picture = canvas.Canvas(paper_size=(11,14), margin_percent=0.05,
↳origin='center')
#picture_bbox = Box(picture.bbox[0,0], picture.bbox[0,1], picture.bbox[1,0],
↳picture.bbox[1,1])

picture_bbox = Point((0,0)).buffer(4.5)

<Figure size 792x1008 with 0 Axes>

[398]: dp = {
    "alpha":0.7,
    "linewidth":0.3*0.0393701*72,
    "clear":False,
```

```
}
```

```
[399]: p1, p2 = 0.1, 0.14
l1, l2 = 2, 9
o1, o2 = 2,2

params = [
    {
        'octaves':o1,
        'persistence':p1,
        'lacunarity':l1,
        'base':0,
    },
    {
        'octaves':o1,
        'persistence':p2,
        'lacunarity':l1,
        'base':0,
    },
    {
        'octaves':o2,
        'persistence':p1,
        'lacunarity':l2,
        'base':0,
    },
    {
        'octaves':o2,
        'persistence':p2,
        'lacunarity':l2,
        'base':0,
    },
]
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[400]: """
dist = 0.01
dist2 = 0.005

dist = 0.005
```

```

dist2 = 0.0025

dist = 0.005
dist2 = 0.0025
"""
dist = 0.01
dist2 = 0.005

n_samples = 700

c1, c2 = 'dodgerblue', 'purple'
edge = 20

s1, s2 = 0.2, 0.25

to_draw = [
    [np.linspace(s1, edge, n_samples)[1::2], 2, c1],
    [np.linspace(s1+dist, edge+dist, n_samples)[1::2], 2, c1],
    [np.linspace(s1, edge, n_samples)[1::2], 1, c1],
    [np.linspace(s1+dist, edge+dist, n_samples)[1::2], 1, c1],

    [np.linspace(s2, edge+.05, n_samples)[::2], 1, c2],
    [np.linspace(s2+dist2, (edge+.05)+dist2, n_samples)[::2], 1, c2],
    [np.linspace(s2, edge+.05, n_samples)[::2], 3, c2],
    [np.linspace(s2+dist2, (edge+.05)+dist2, n_samples)[::2], 3, c2],

]

scale = (1/7)*(2/3)
theta = np.linspace(0, 2*np.pi, 500)

```

```

[401]: layers = []

alternate = False

for rad, pind, color in to_draw:
    all_lines = []

    rmat, tmat = np.meshgrid(rad, theta)

    for i in range(rmat.shape[1]):
        r, t = rmat[:, i], tmat[:, i]+np.random.uniform(0, 6*np.pi)

        x, y = r*np.cos(t), r*np.sin(t)

```

```

        z = np.array([noise.pnoise2(x[t]*scale, y[t]*scale, **params[pind]) for
↪t in range(len(x))])
        z += 2

        proj_line = np.vstack([x/z, y/z]).transpose()

        if alternate:
            proj_line = proj_line[::-1]

        alternate = not alternate

        all_lines.append(LineString(proj_line).simplify(1e-3).
↪intersection(picture_bbox))
        #all_lines.append(LineString(proj_line).simplify(1e-3))

        layers.append((all_lines, color))

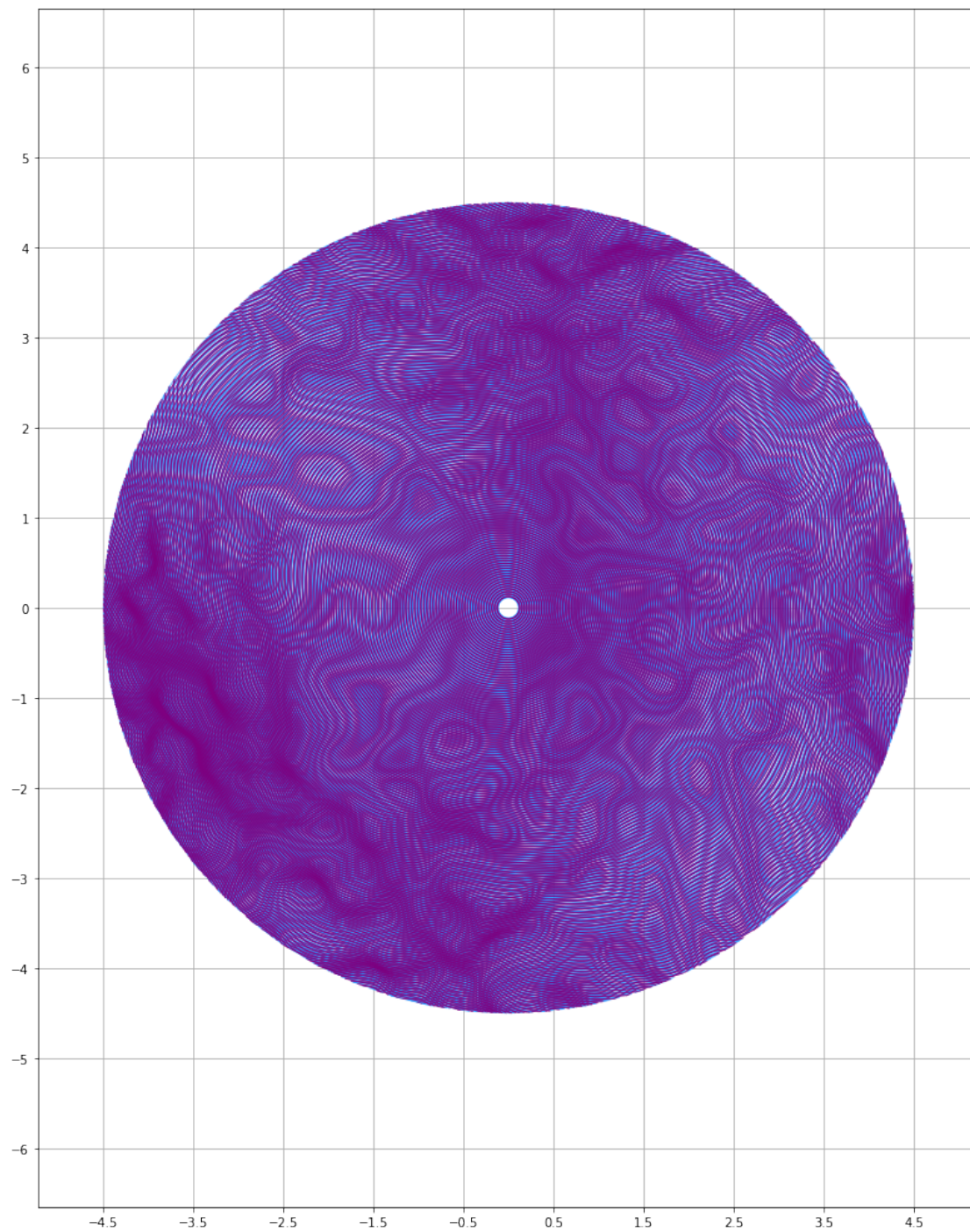
picture.make_canvas()
picture.add_grid(11,14)

for all_lines,c in layers:
    #picture.plot_shapes(all_lines, color='black', **dp)
    picture.plot_shapes(all_lines, color=c, **dp)
picture.fig

```

None

[401]:



[]:

[]:

```
[402]: alternate = False

layer_1 = []
layer_2 = []

for all_lines, c in layers:

    if alternate:
        all_lines = all_lines[::-1]
        alternate = not alternate

    if c == c1:
        layer_1 += all_lines
    else:
        layer_2 += all_lines
```

```
[ ]:
```

```
[ ]:
```

```
[403]: save = True
```

```
[404]: picture.make_canvas()

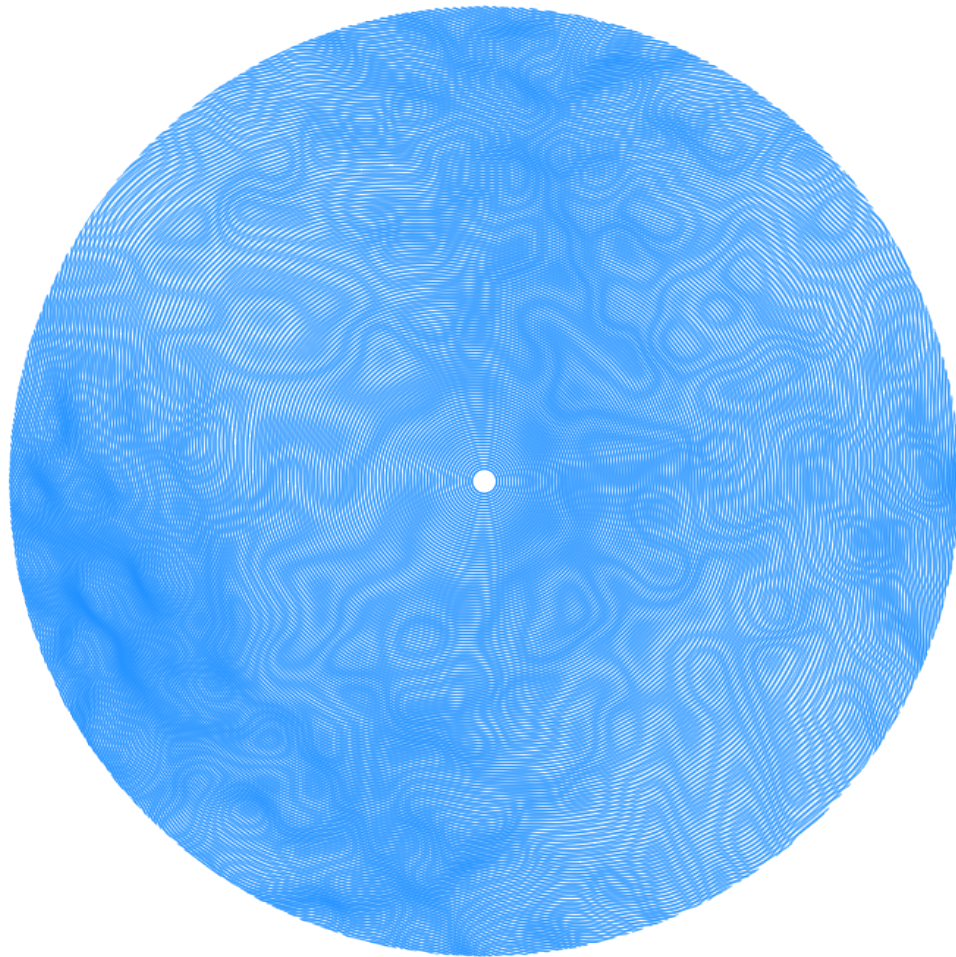
picture.plot_shapes([l.intersection(picture_bbox) for l in layer_1], color=c1, u
↪**dp)
picture.display_overlays(False)

if save:
    picture.fig.savefig("trippy_perlin_disk_video_l1.svg")

picture.fig
```

```
(11, 14)
```

```
[404]:
```



```
[405]: picture.make_canvas()  
  
picture.plot_shapes([l.intersection(picture_bbox) for l in layer_2], color=c2,   
↪ **dp)
```



```
picture.display_overlays(False)

if save:
    picture.fig.savefig("trippy_perlin_disk_video_l2.svg")

picture.fig
```

(11, 14)

[405]:




```
[39]: """
out=/Users/gnb/plotter_exps/printed/lavalamp.svg

f1=/Users/gnb/Projects/plotterart/pieces/2d_moiress/oil_slick/trippy_perlin_l1.
↪svg
f2=/Users/gnb/Projects/plotterart/pieces/2d_moiress/oil_slick/trippy_perlin_l2.
↪svg

vptype \
read -l 1 $f1 \
read -l 2 $f2 \
write --page-format 11x14 --center $out
"""
```

```
[39]: '\nout=/Users/gnb/plotter_exps/printed/lavalamp.svg\n\nf1=/Users/gnb/Projects/pl
otterart/pieces/2d_moiress/oil_slick/trippy_perlin_l1.svg\nf2=/Users/gnb/Projects
/plotterart/pieces/2d_moiress/oil_slick/trippy_perlin_l2.svg\n\n\n\nvptype read -l
1 $f1 read -l 2 $f2 write --page-format 11x14 --center $out\n'
```

[]:

[]:

[]: