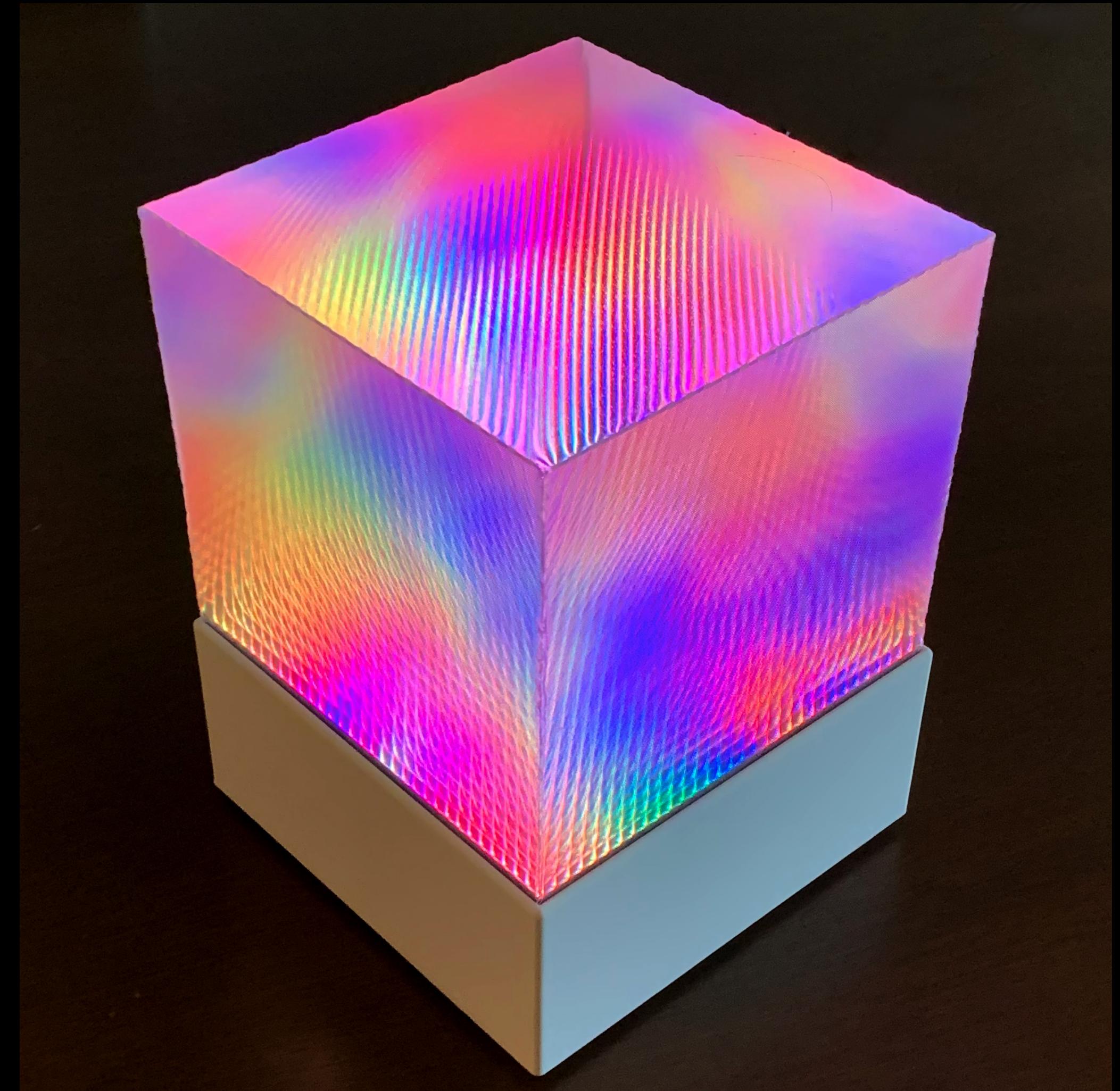


Lenticular Cube

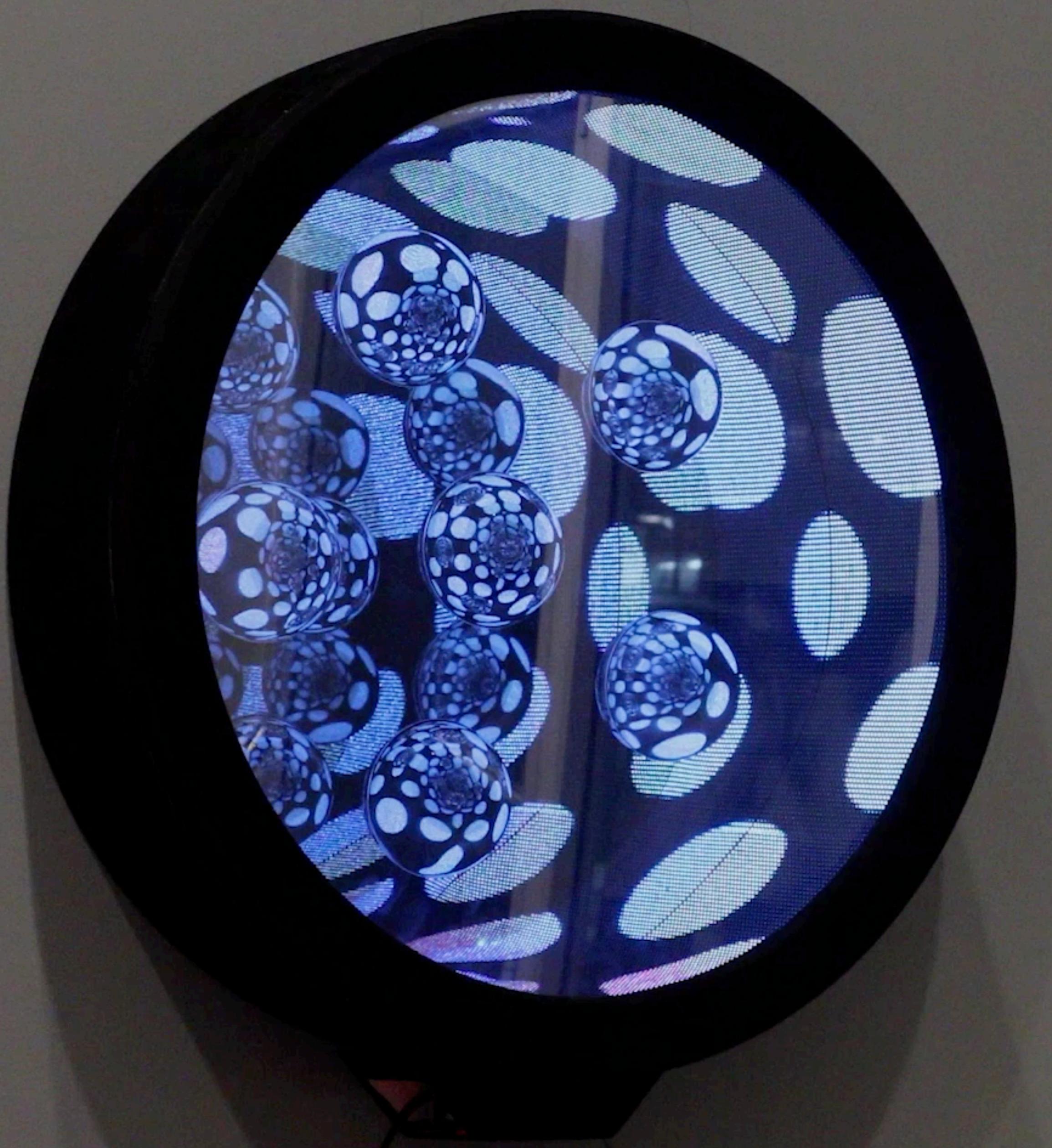
**Real time LED Matrix Control via
UDP, Touchdesigner Scripting,
and Raspberry Pi**



@aidanlincoln - May 2022

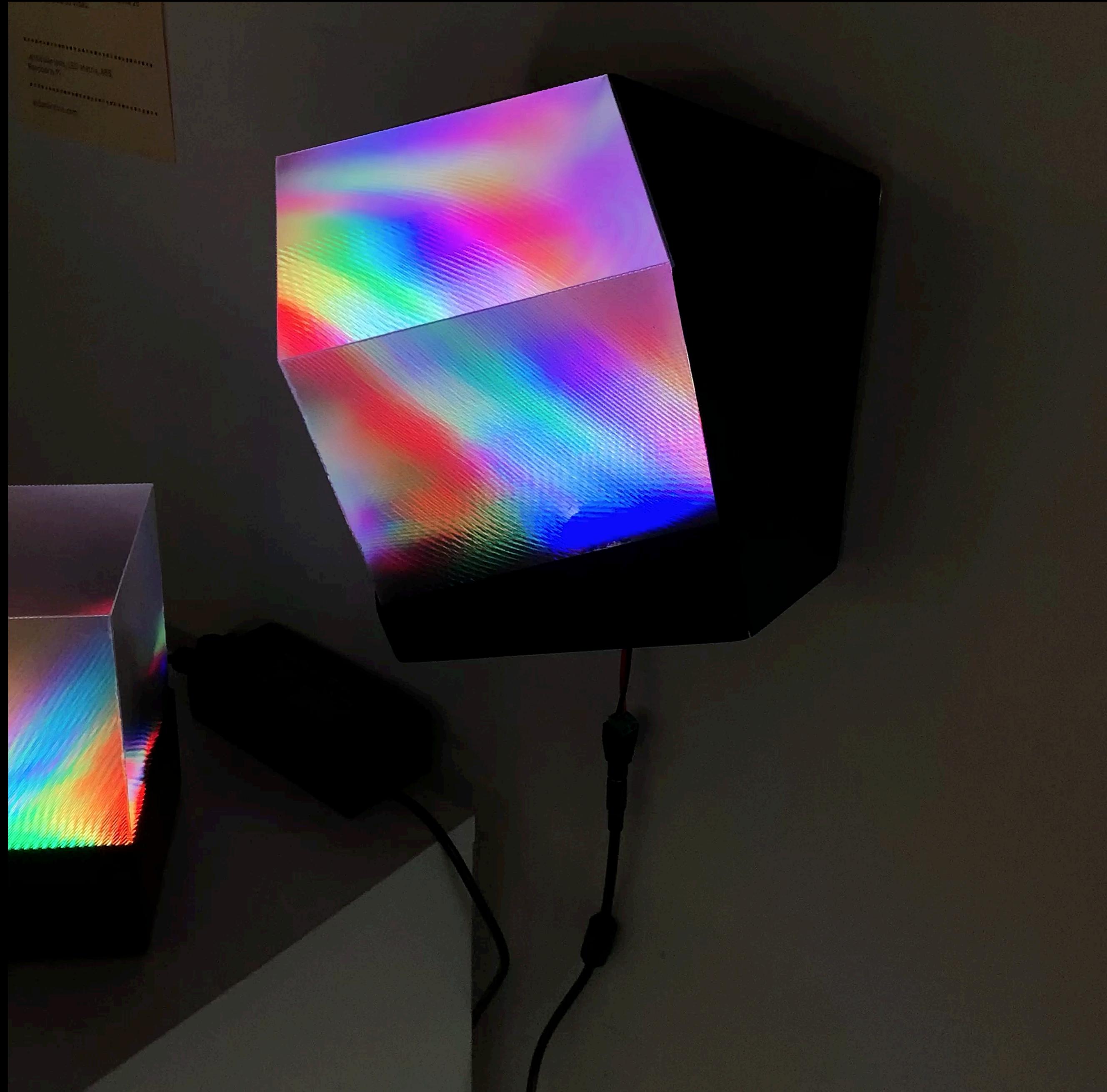
Background

- Brooklyn Based
- Installation Art
- Light Art
- Machine Learning
- Fabrication
- Software Architecture

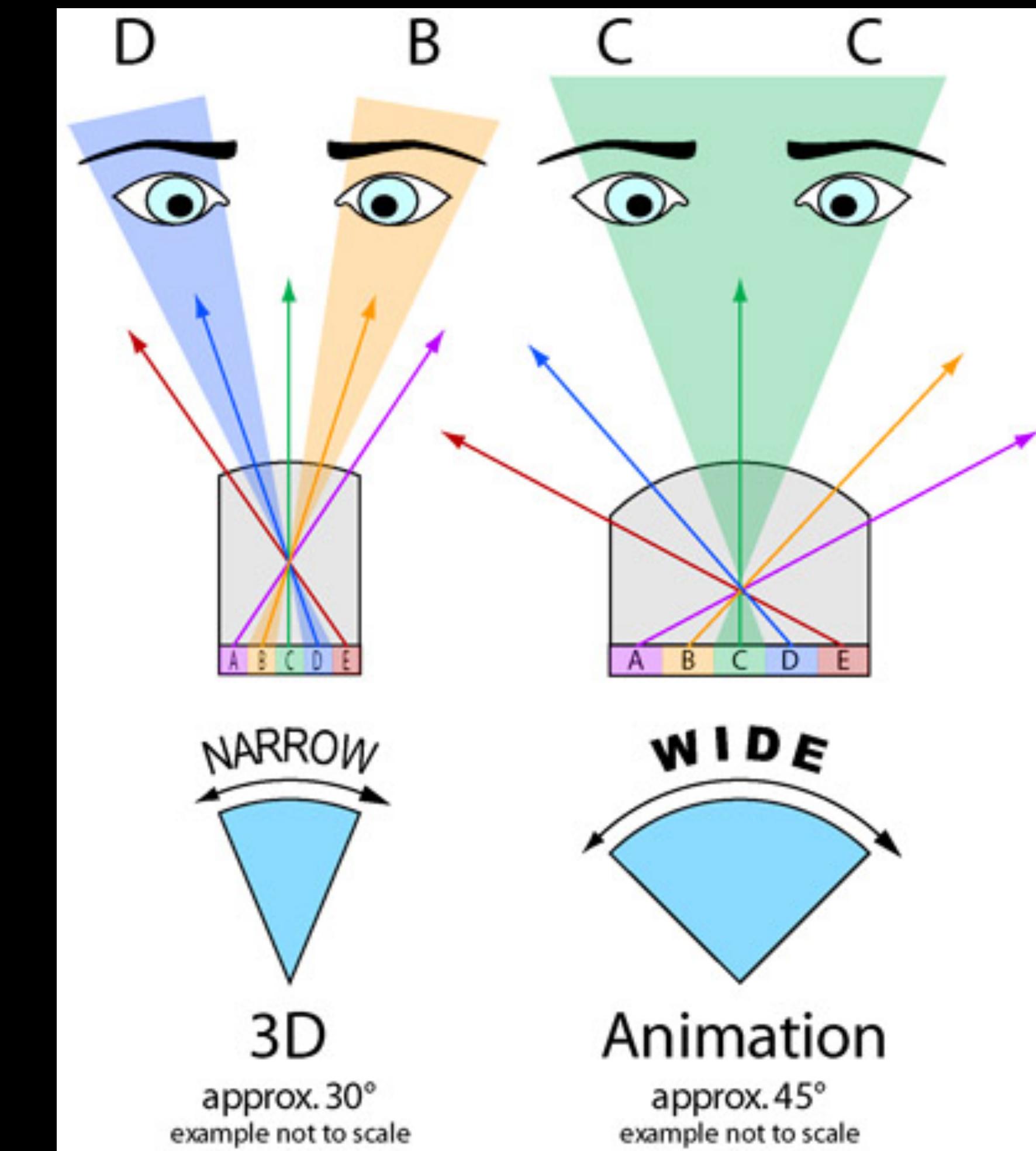
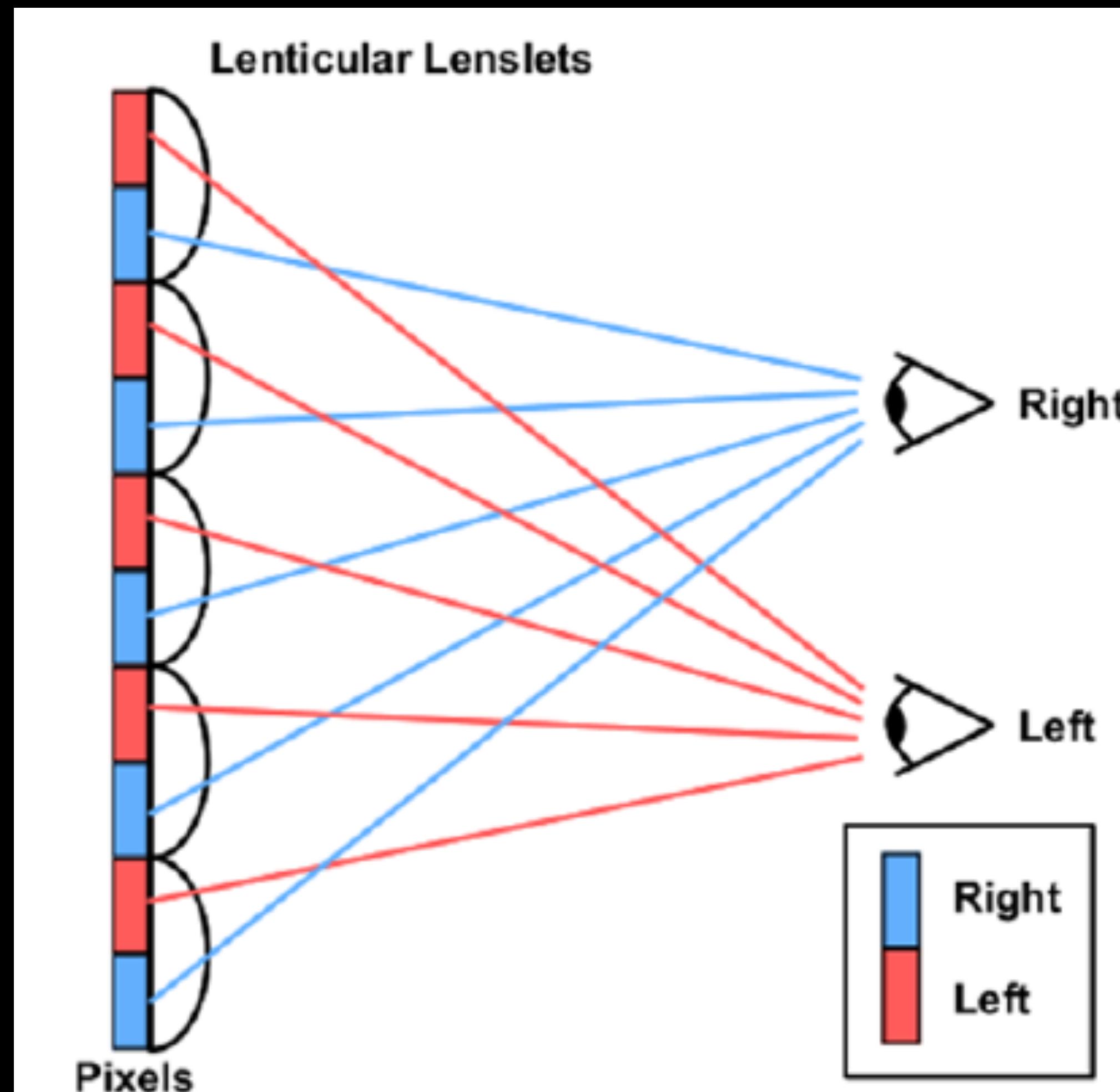


Lenticular Cube

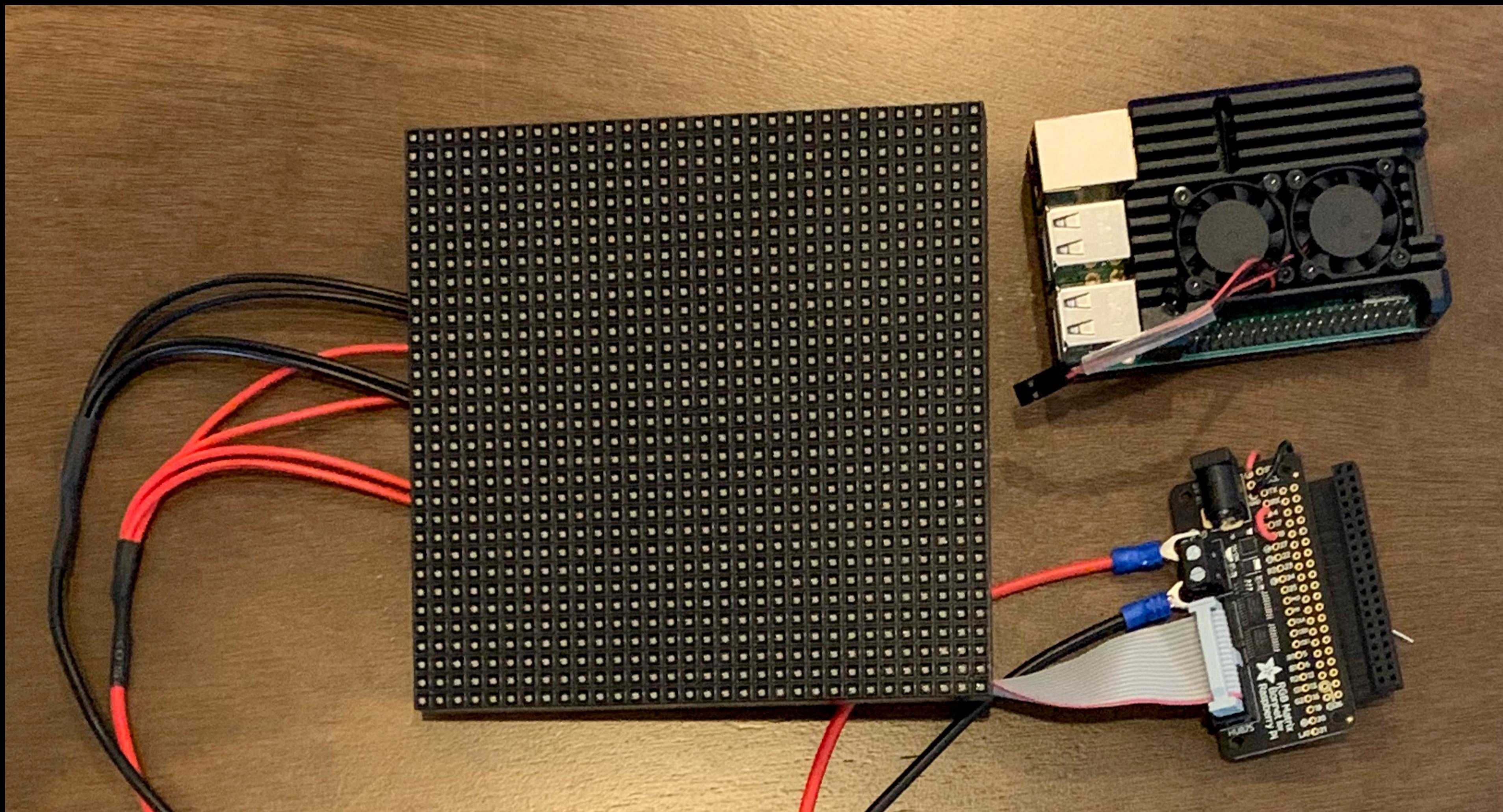
- Lenticular Lens Volume
- LED Matrix Pannel
- 3D Printed Enclosure
- Touchdesigner UDP Server
- Raspberry Pi UDP Client
- Adafruit RGB Matrix
Bonnet + Hzeller Library



Lenticular Lens



LED Matrix + Raspberry Pi +Bonnet

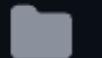
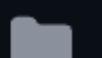
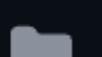
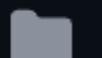
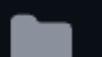


Hzeller Library For Led Matrix

hzeller / rpi-rgb-led-matrix Public

<> Code Issues 398 Pull requests 24 Actions Projects Wiki Security Insights

master ▾ 6 branches 0 tags Go to file Code ▾

commit	Message	Date
 39953b7	Allow for fonts with glyphs wider than 64 pixels. ...	on Apr 15 886 commits
 .github/workflows	Create c-cpp.yml (#1140)	2 years ago
 adapter	Some board-houses can't deal with Gerber X2 attributes yet.	3 years ago
 bindings	Generate Python binding with latest cython.	6 months ago
 examples-api-use	Remove unnecessary message.	7 months ago
 fonts	Compiling otf2bdf requires using some modern tools; provide assiste...	2 years ago
 img	Added V-Mapper:Z for flipped orientation panels chains (#1014)	2 years ago
 include	Expose runtime options to C API (#1141)	2 years ago

Raspberry Pi Client (Matrix Initialization)

```
from rgbmatrix import RGBMatrix, RGBMatrixOptions
from PIL import Image
import socket

options = RGBMatrixOptions()
options.rows = 32
options.cols = 32
options.chain_length = 1
options.gpio_slowdown = 3
options.pwm_bits = 11
options.show_refresh_rate = True
options.brightness = 95
options.hardware_mapping="adafruit-hat-pwm"
options.limit_refresh_rate_hz=100
matrix = RGBMatrix(options = options)
canvas = matrix.CreateFrameCanvas()

print("Matrix Initialized")
```

Raspberry Pi Client (UDP Socket)

```
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.connect(("8.8.8.8", 80))
localIP = s.getsockname()[0]
localPort = 51000
bufferSize = 3072 #for 1 led screens 32x32x3

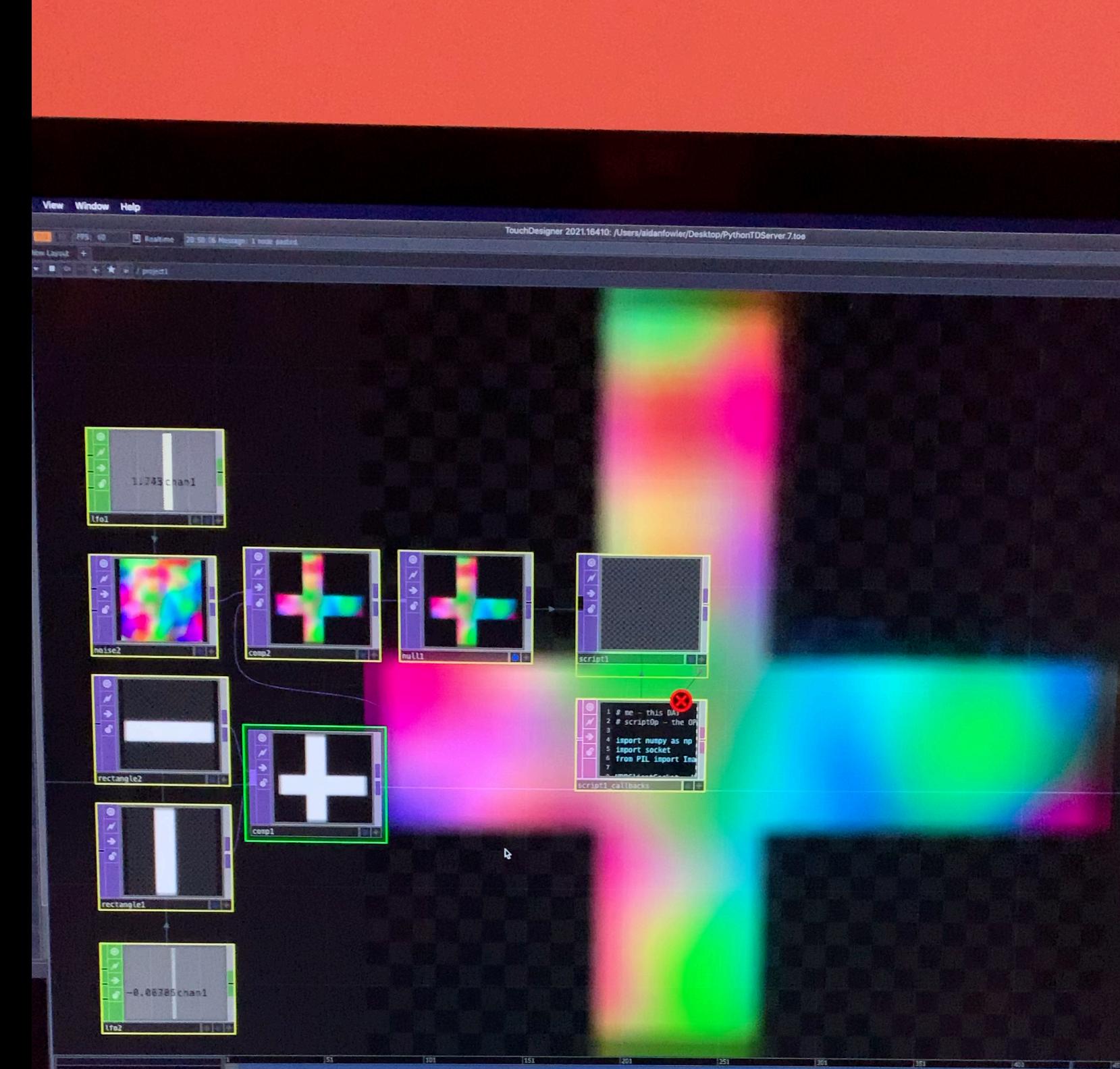
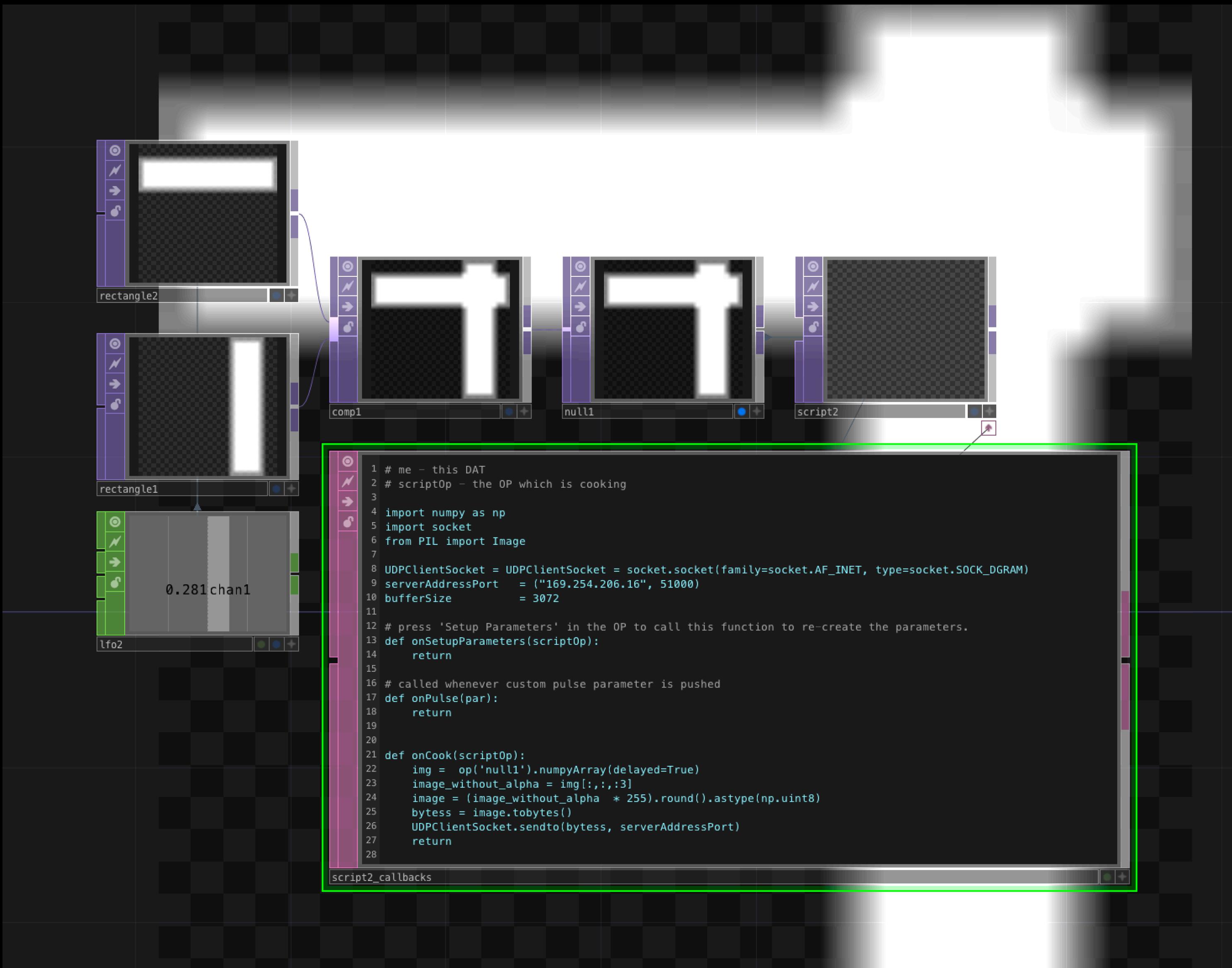
UDPServerSocket = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)
UDPServerSocket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
UDPServerSocket.bind((localIP, localPort))

print("RPi IP Address: ",localIP)
print("UDP Server Listening For Content On Port: ",localPort)

while(True):
    bytesAddressPair = UDPServerSocket.recvfrom(bufferSize)
    message = bytesAddressPair[0]
    im = Image.frombytes("RGB", (options.cols, options.rows), message, "raw")
    im = im.transpose(Image.FLIP_LEFT_RIGHT)
    canvas.SetImage(im)
    matrix.SwapOnVSync(canvas)
```

```
[aidan@ledmatrix:~/server $ sudo python3 matrixServer.py
Matrix Initialized
RPi IP Address: 169.254.206.16
UDP Server Listening For Content On Port: 51000
100.0Hz (lowest: 100.0Hz)
```

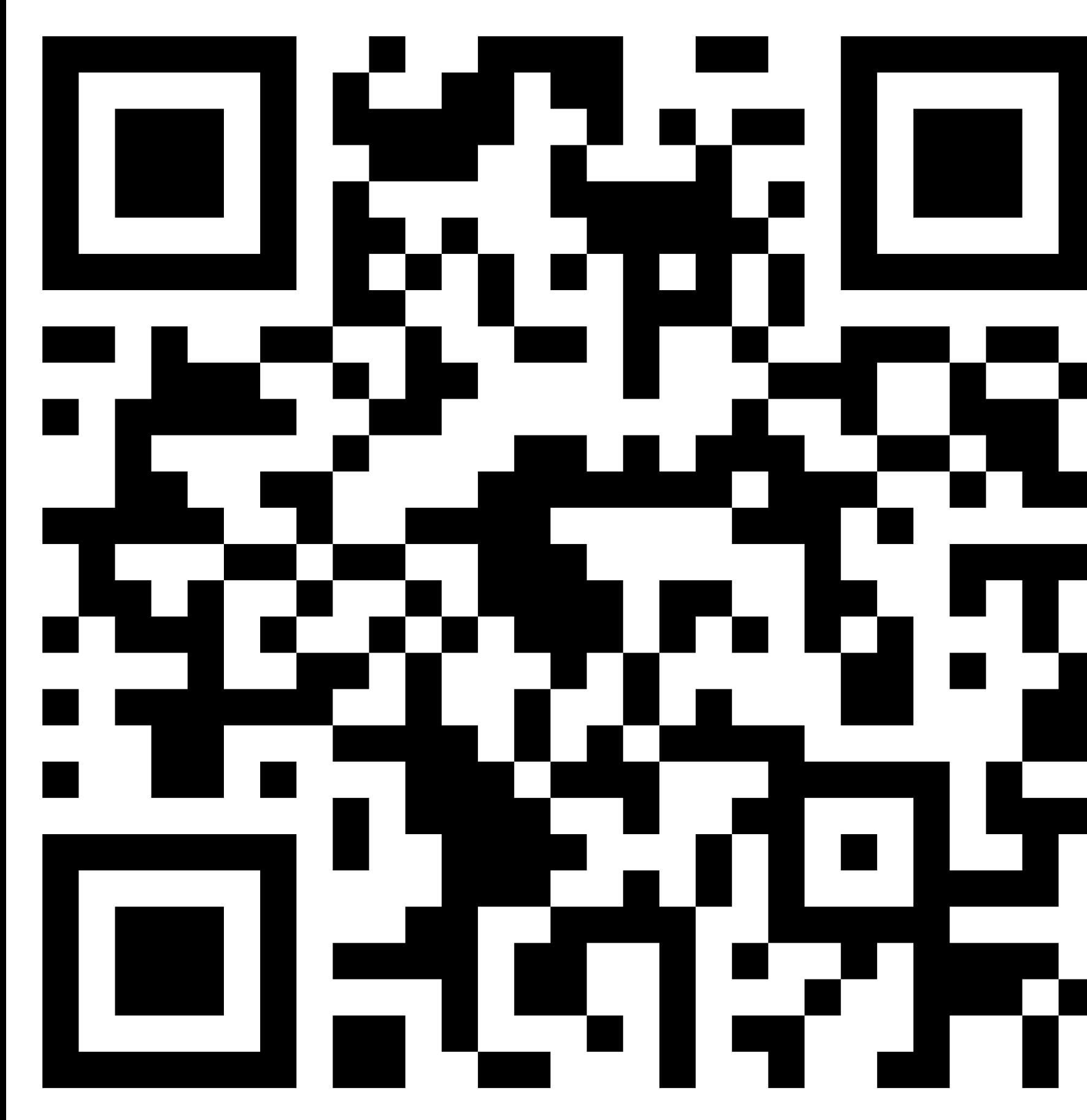
Touchdesigner UDP Server



Touchdesigner UDP Server

```
1  # me - this DAT
2  # scriptOp - the OP which is cooking
3
4  import numpy as np
5  import socket
6  from PIL import Image
7
8  UDPClientSocket = UDPClientSocket = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)
9  #serverAddressPort = ("127.0.0.1", 51000)
10 serverAddressPort = ("169.254.28.93", 51000)
11 bufferSize = 3072
12
13 # press 'Setup Parameters' in the OP to call this function to re-create the parameters.
14 def onSetupParameters(scriptOp):
15     return
16
17 # called whenever custom pulse parameter is pushed
18 def onPulse(par):
19     return
20
21
22 def onCook(scriptOp):
23     img = op('null1').numpyArray(delayed=True)
24     image_without_alpha = img[:, :, :3]
25     image = (image_without_alpha * 255).round().astype(np.uint8)
26     bytess = image.tobytes()
27     UDPClientSocket.sendto(bytess, serverAddressPort)
28     #scriptOp.copyNumpyArray(img)
29     return
30
```

Cube is open for live coding!



Feel free to reach out!

Aidan Lincoln Fowler

@aidanlincolnn

aidanlincoln.com

fowler.aidan@gmail.com