Imaginative Python

Morgan Visnesky

What is Art?

Dictionary

Search for a word

art[1]

/ärt/

noun

1. the expression or application of human creative skill and imagination, typically in a visual form such as painting or sculpture, producing works to be appreciated primarily for their beauty or emotional power.
   "the art of the Renaissance"

   Similar: fine art | artwork | creative activity

What is Art?

Dictionary

Search for a word

art[1]

/ärt/

noun

1. the expression or application of human creative skill and imagination, typically in a visual form such as painting or sculpture, producing works to be appreciated primarily for their beauty or emotional power.
   "the art of the Renaissance"

   Similar: fine art | artwork | creative activity

Process Oriented Art

- “emphasizes or focuses on **processes**, systems, or procedures rather than results or underlying causes” – Oxford Dictionary
- Algorithmic art?

Dictionary

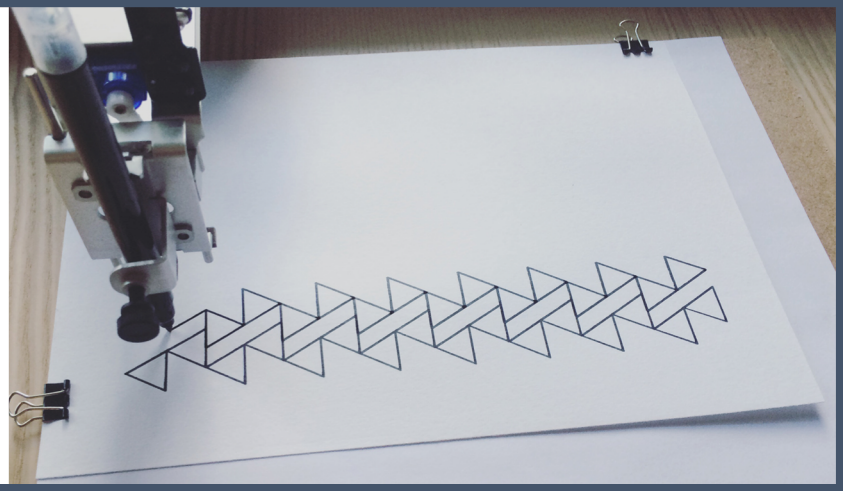
Search for a word

al·go·rithm

/ˈalgəˌriT͟Həm/

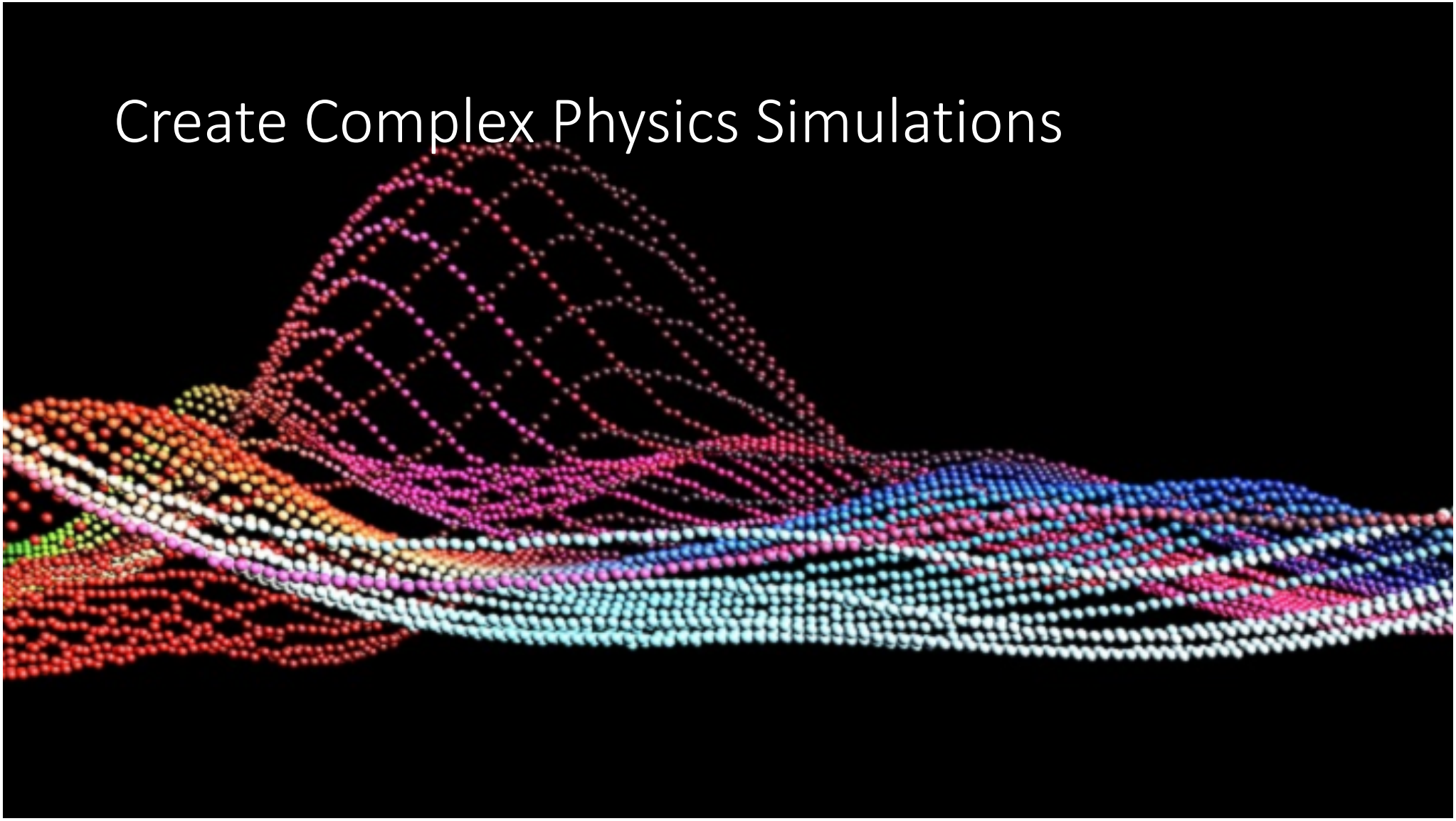
noun

a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.
"a basic **algorithm for** division"

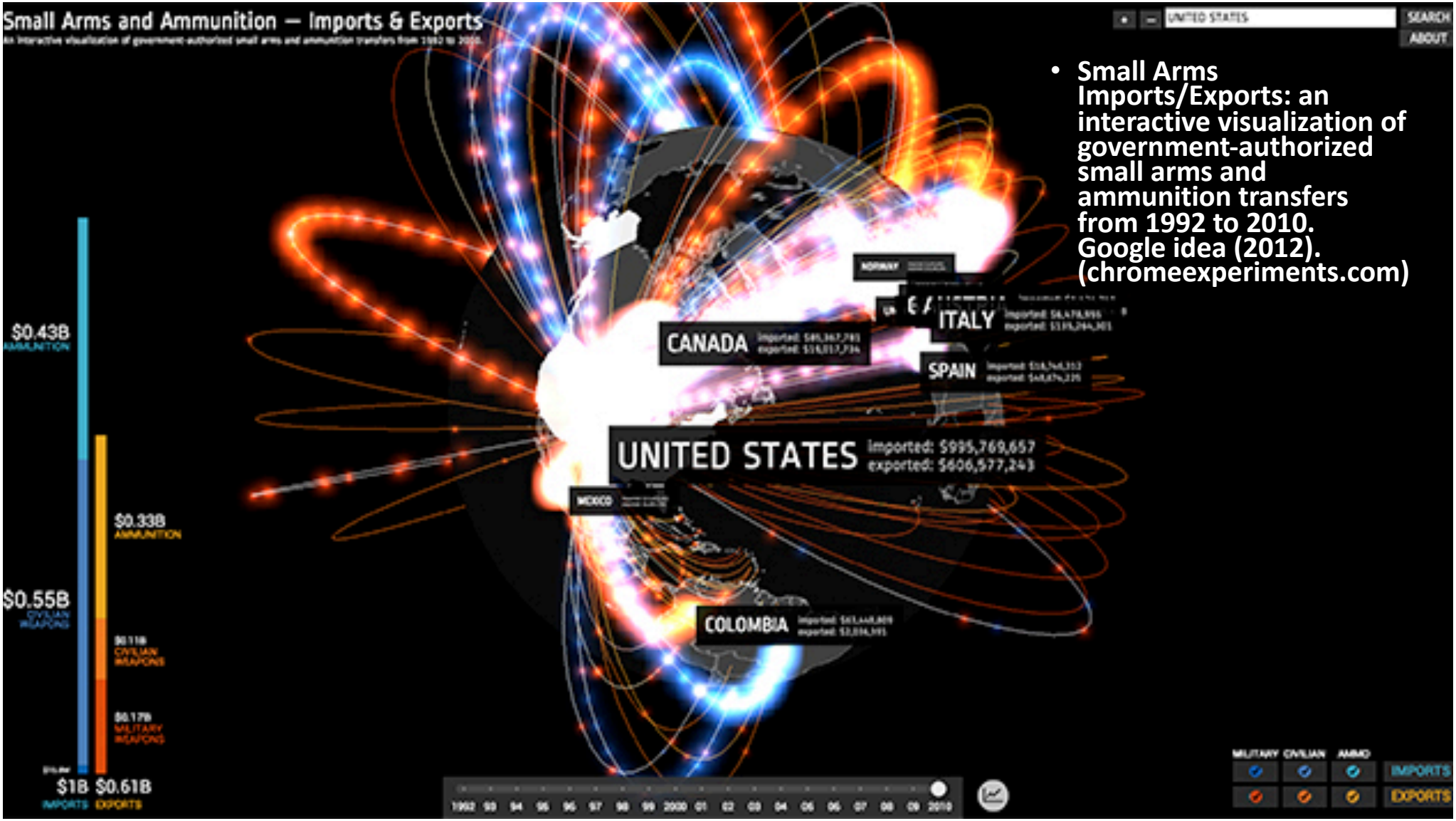



SO WHATS POSSIBLE?

Create Complex Physics Simulations



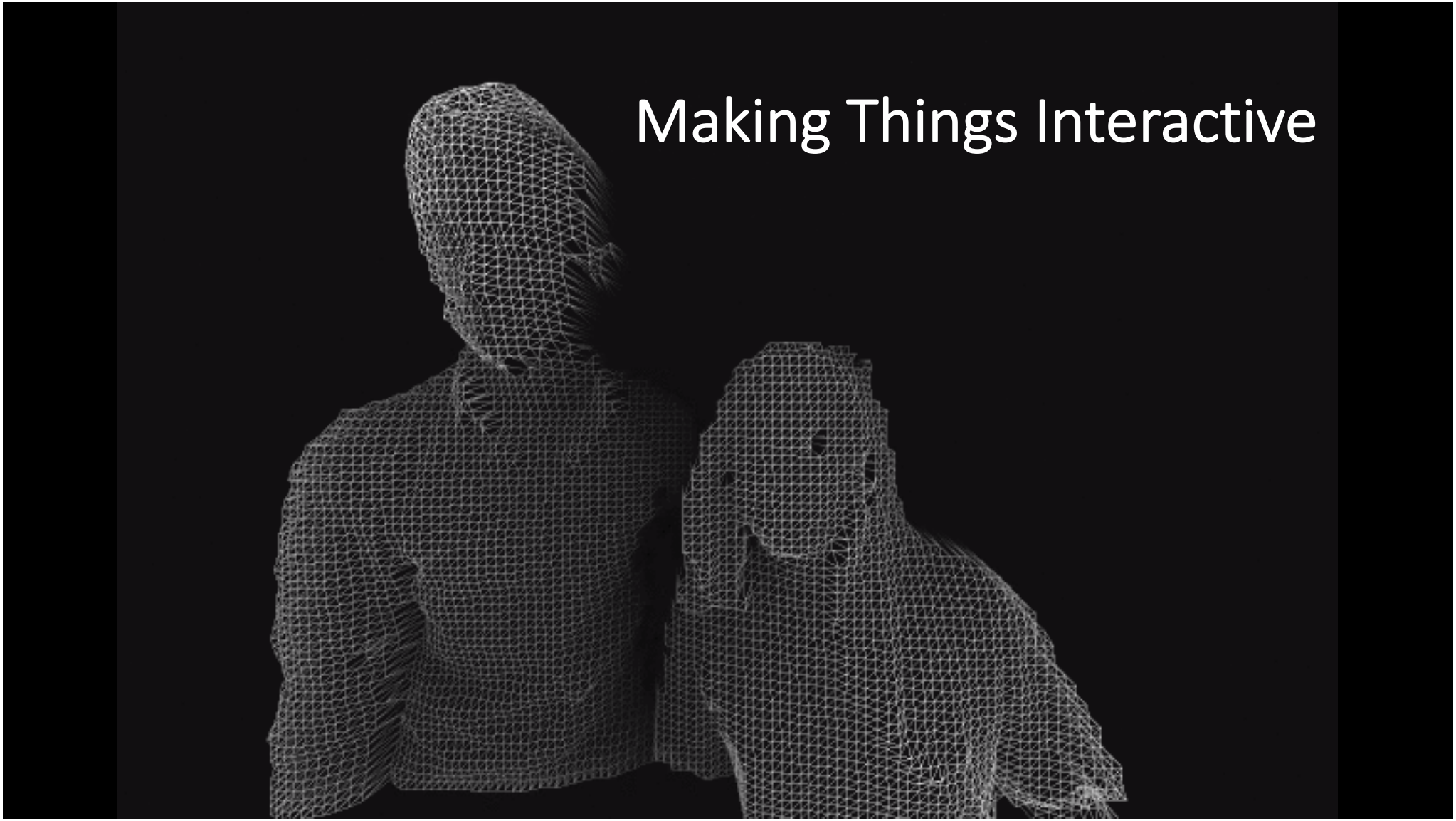

Creatively Visualizing Data sets
January 2003
Friday
01-17
01-16
01-04



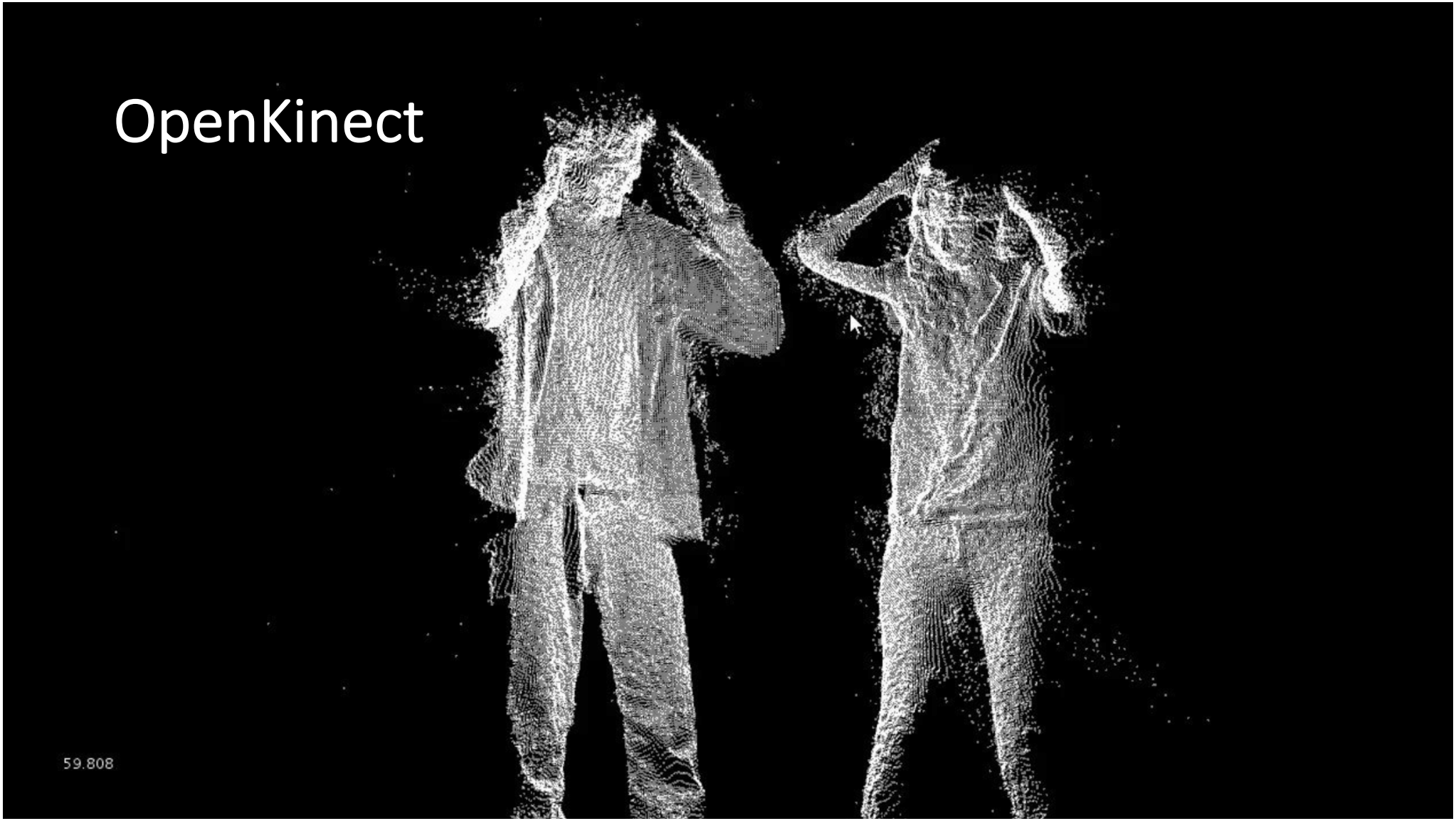
- **Small Arms Imports/Exports: an interactive visualization of government-authorized small arms and ammunition transfers from 1992 to 2010. Google idea (2012). (chromeexperiments.com)**

Concepts that translate

- .OBJ file export/import supported by processing and blender along with a ton of other 3D design software.
  - Can also use these with 3D printing
- Graphics math and programming is really similar across environments.
- Interactivity
- OpenGL / GLSL

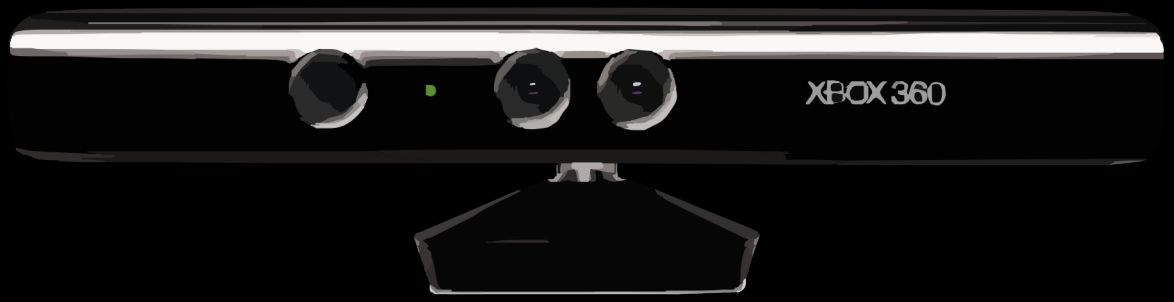

Making Things Interactive

OpenKinect

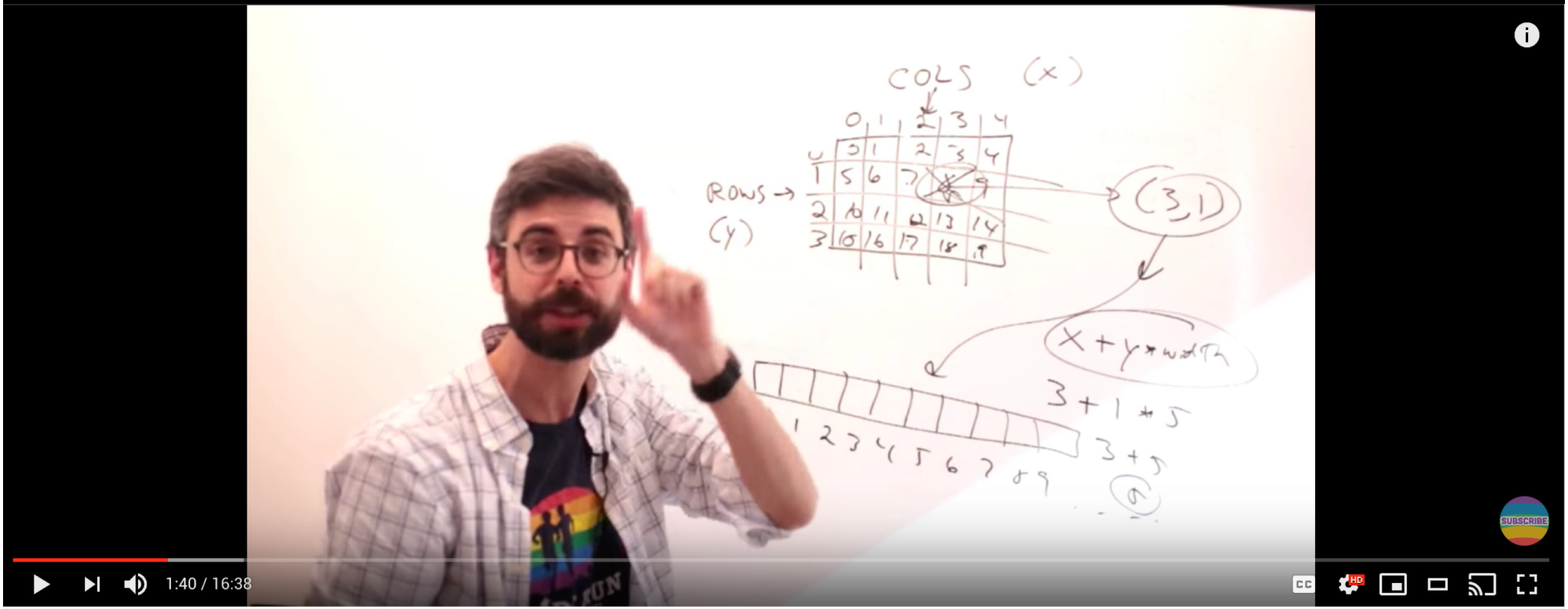


Point cloud Data From Xbox Kinect

- Processing.py
- Python
- Blender
- Maya
- TouchDesigner
- OpenFrameworks
- Unity
- Cinema 4D



Point Cloud Data Explained



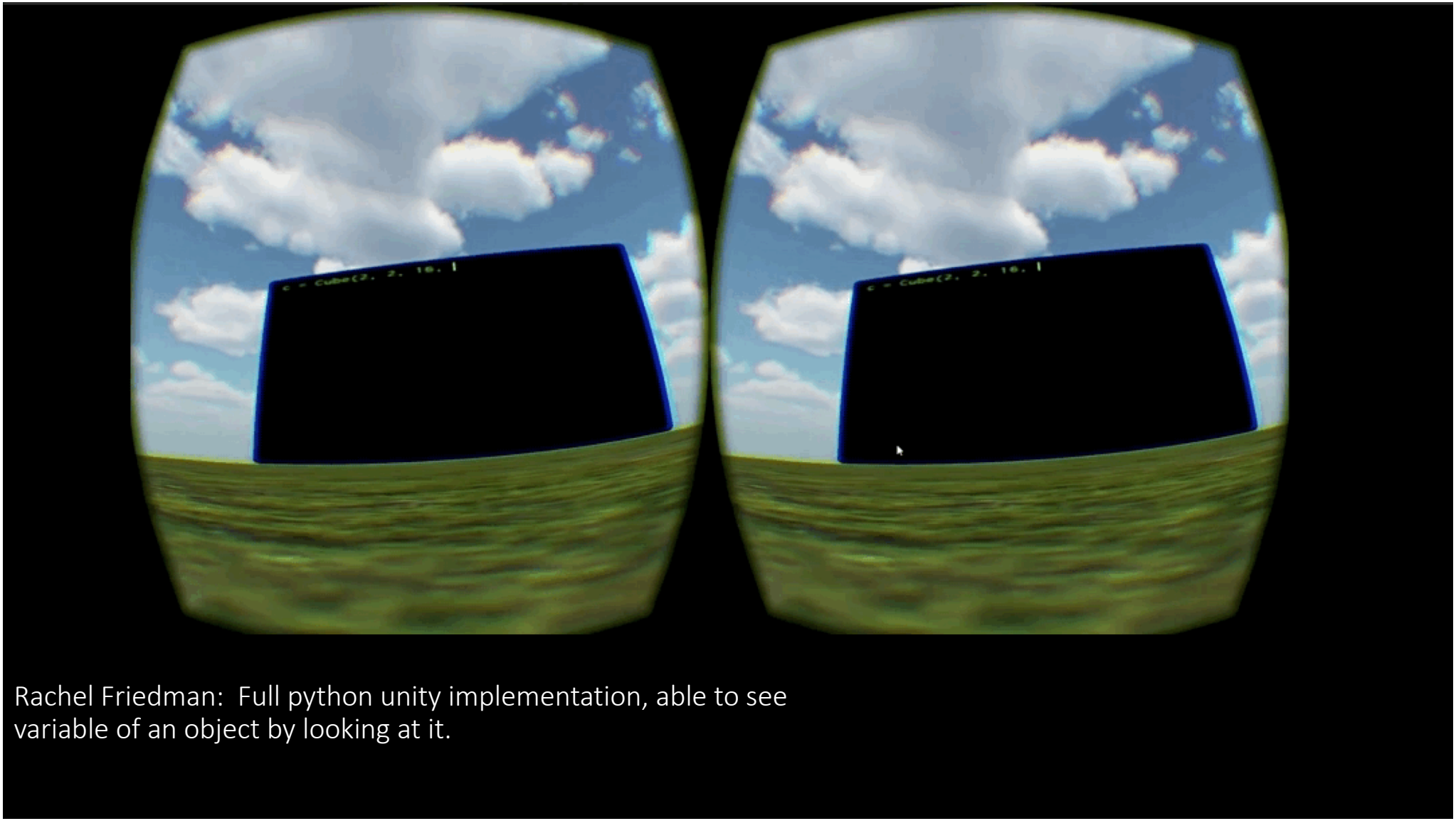
12.3: Raw Depth Data - Point Clouds and Thresholds - Kinect and Processing Tutorial

VR with python

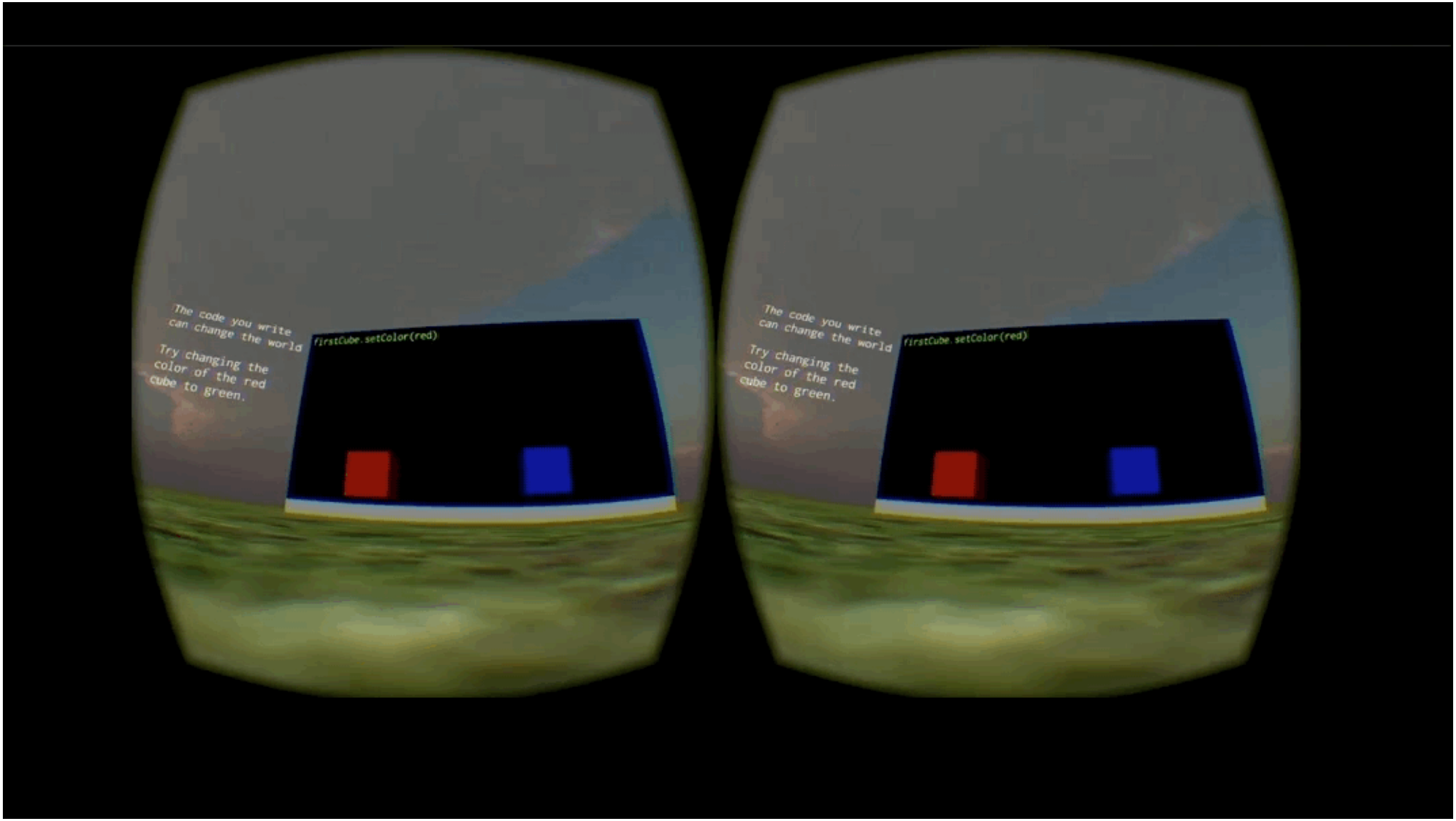
- Blender
- Maya
- OpenGL

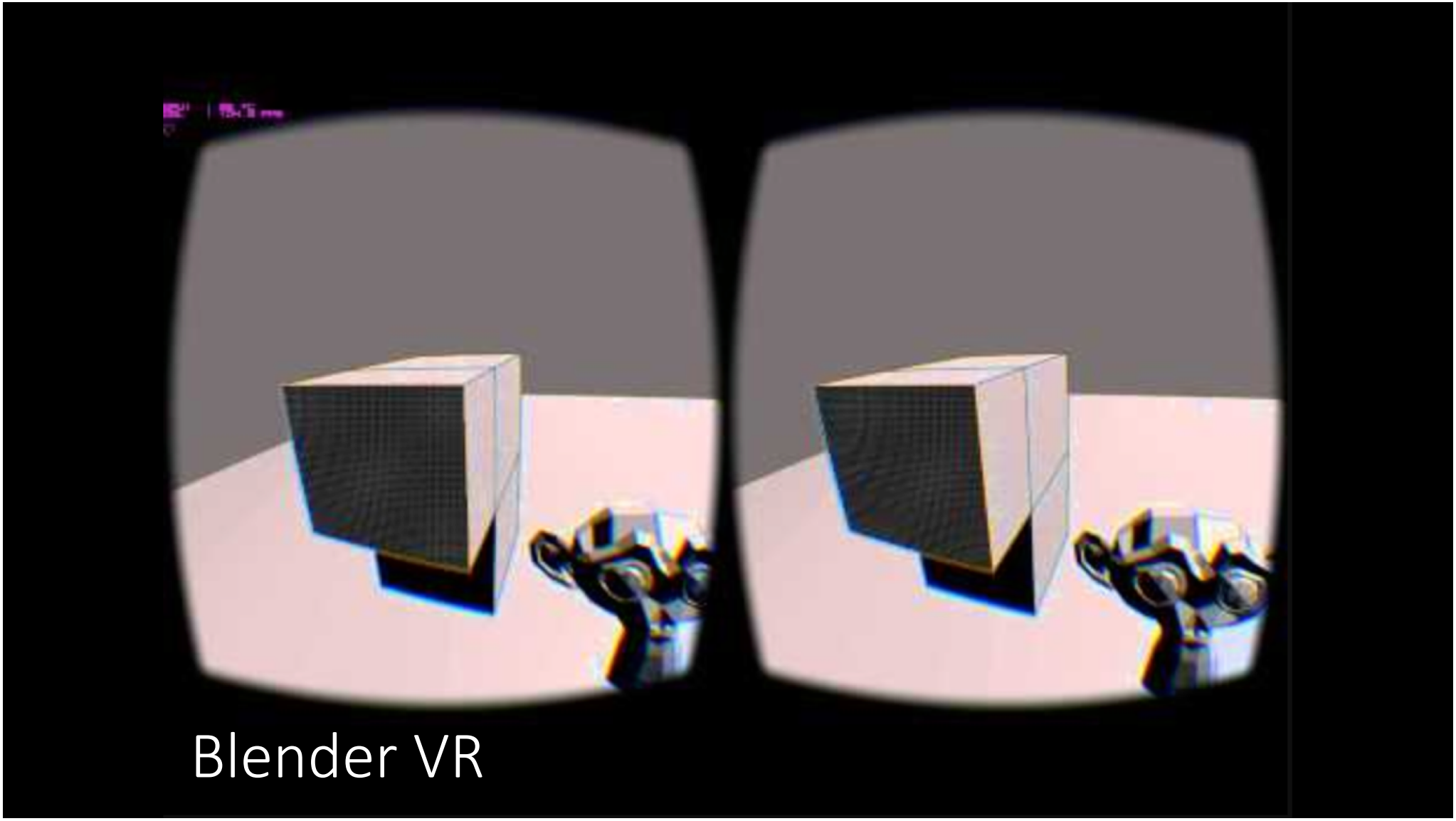
NOT IN PYTHON

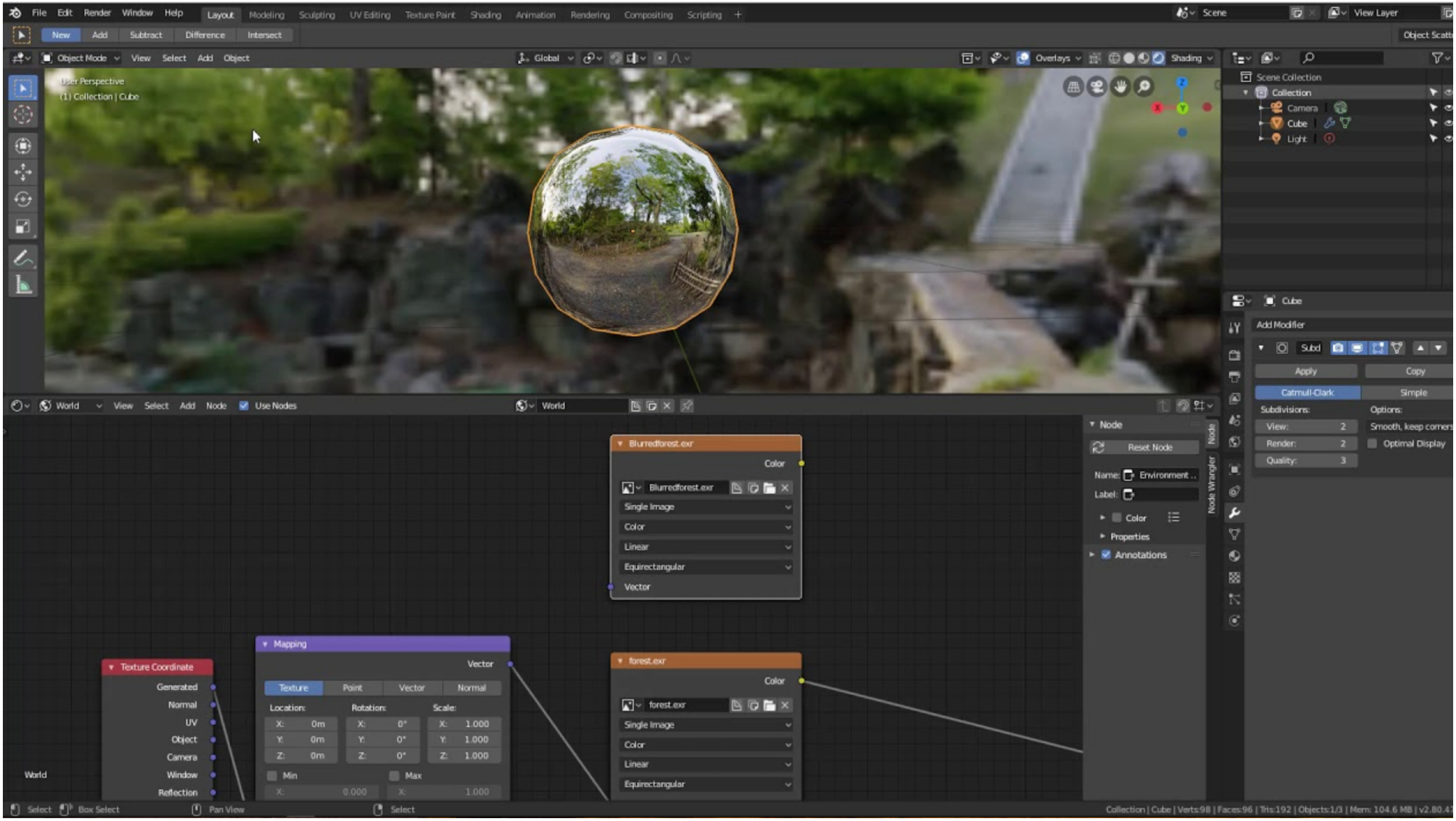
- Processing.vr for android
  - Uses Google VR
  - *this only supports Java ATM*
- Unity, uses C# (I frequently use python to send control signals to this)



Rachel Friedman: Full python unity implementation, able to see variable of an object by looking at it.


The code you write
can change the world
Try changing the
color of the red
cube to green.
firstCube.setColor(red)
The code you write
can change the world
Try changing the
color of the red
cube to green.
firstCube.setColor(red)

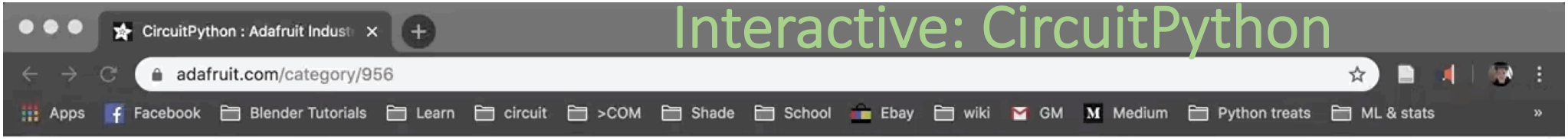

Blender VR


File Edit Render Window Help
Layout Modeling Sculpting UV Editing Texture Paint Shading Animation Rendering Compositing Scripting
Scene
View Layer
New Add Subtract Difference Intersect
Object Mode View Select Add Object
Global
Overlays
Shading
User Perspective
(1) Collection | Cube
Scene Collection
Collection
Camera
Cube
Light
Add Modifier
Subd
Apply
Copy
Catmull-Clark
Simple
Subdivisions:
Options:
View: 2
Render: 2
Quality: 3
Smooth, keep corners
Optimal Display
World View Select Add Node Use Nodes
Blurredforest.exr
Color
Single Image
Color
Linear
Equirectangular
Vector
Node
Reset Node
Name: Environment...
Label:
Color
Properties
Annotations
Texture Coordinate
Generated
Normal
UV
Object
Camera
Window
Reflection
Mapping
Vector
Texture Point Vector Normal
Location: Rotation: Scale:
X: 0m Y: 0m Z: 0m
X: 0° Y: 0° Z: 0°
X: 1.000 Y: 1.000 Z: 1.000
Min Max
forest.exr
Color
Single Image
Color
Linear
Equirectangular
World
Select Box Select Pan View Select
Collection | Cube | Verts:98 | Faces:96 | Tris:192 | Objects:1/3 | Mem: 104.6 MB | v2.80.4

Interactive: Bot art

- Input data into your project from bot interactivity.
  - Webserver that outputs art based on:
    - user input
    - Many different types of twitter data
- Use tools like:
  - Natural language processing & sentiment analysis
  - Social media and other web API’s
    - Twitter
    - Slack
    - Robinhood (Stock Market Data)
    - OpenWeatherMap

Interactive: CircuitPython



CIRCUITPYTHON

CircuitPython is Adafruit's branch of MicroPython designed to simplify experimentation and education on low-cost microcontrollers. No compiler, linker or IDE required! Now you can code directly on the device itself. CircuitPython makes it easier than ever to get prototyping by requiring *no upfront desktop software downloads* - open up **main.py** in any text editor and type away.

It is great for beginners: With CircuitPython you can write clean and simple Python code to control hardware instead of having to use complex low-level languages like Arduino, C or C++. The simplicity of the Python programming language makes CircuitPython an excellent choice for beginners who are new to programming and hardware.

Experts love it too: skip the annoyances of long-compile cycles, pointers and memory management - get right to the programming you want to do. CircuitPython is also quite full-featured and supports all of Python's syntax (Python version 3.4) and implements some useful parts of the Python standard library so even seasoned Python veterans will find CircuitPython familiar and fun to use.

Pick up a Circuit Playground Express if you want everything-and-the-kitchen-sink-too, Gemma M0 for small wearables, Trinket M0 for compact breadboard-friendly designs, Metro M0 Express for shield-compatibility or Feather M0 Express to take advantage of the Feather system

Raspberry pi

- Python libraries to leverage GPIO pins for electronics projects
- Runs processing and Blender
- Easy device to use for artistic installations
- Can update and interact with wirelessly


moving_3DCUBE_gamecontroller
Switch Controller

Game Controller Setup

Game Controller

Connects to pi via bluetooth

Raspberry pi
Running evdev script

Parses bluetooth messages with evdev & performs conversion

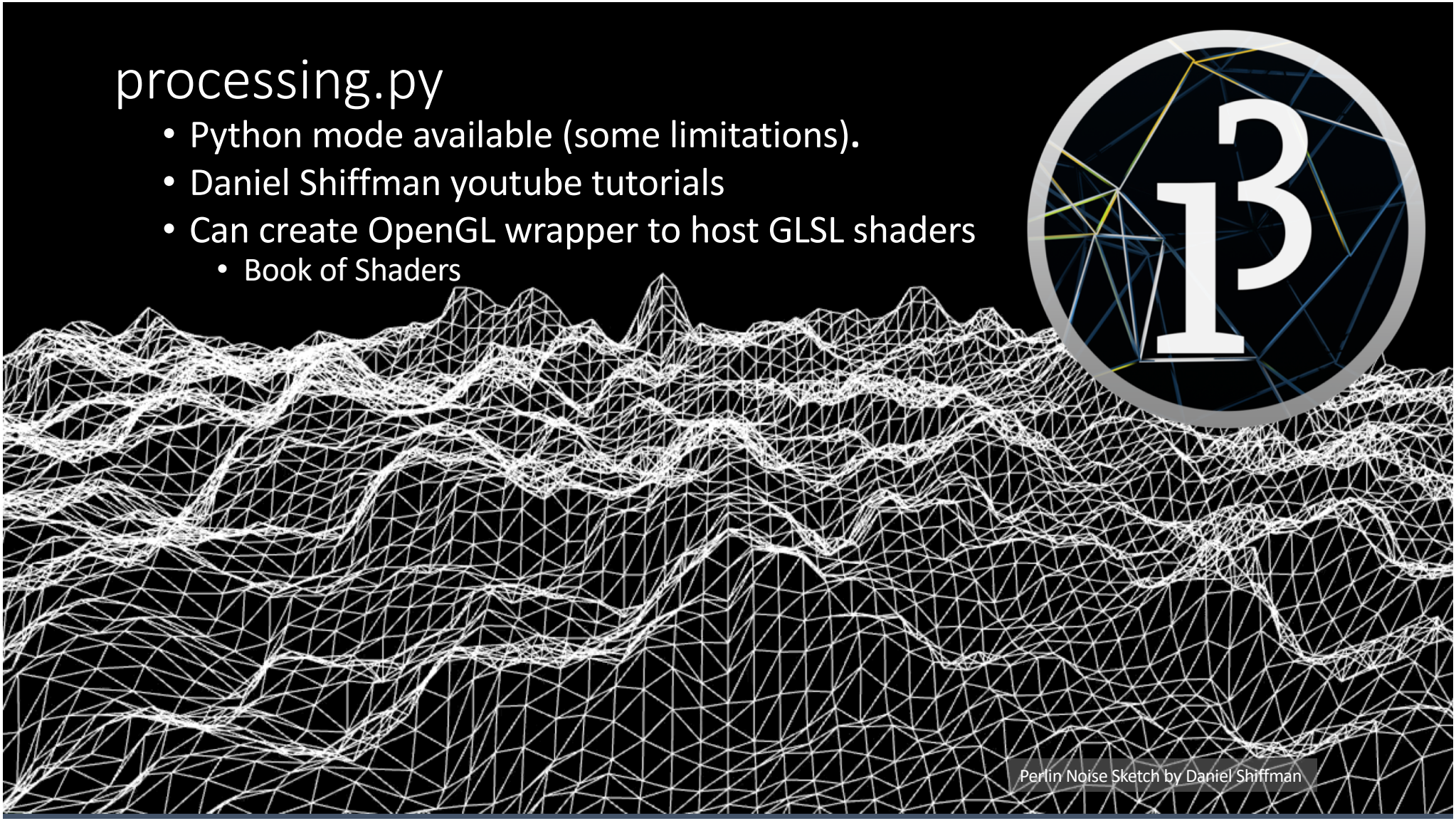
Sends converted messages as UDP

Macbook Pro
Running processing

Performs logic based on UDP

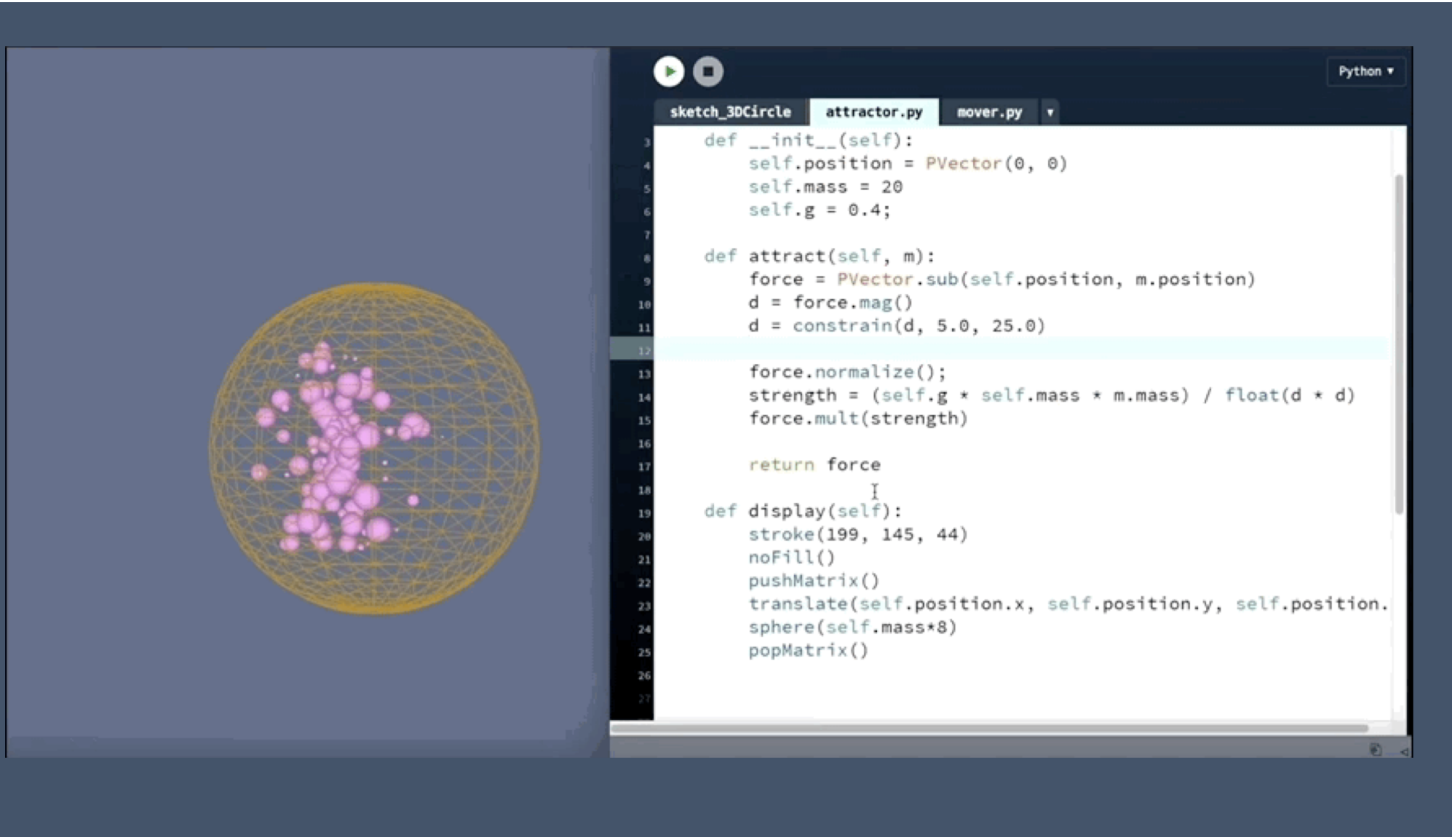
processing.py

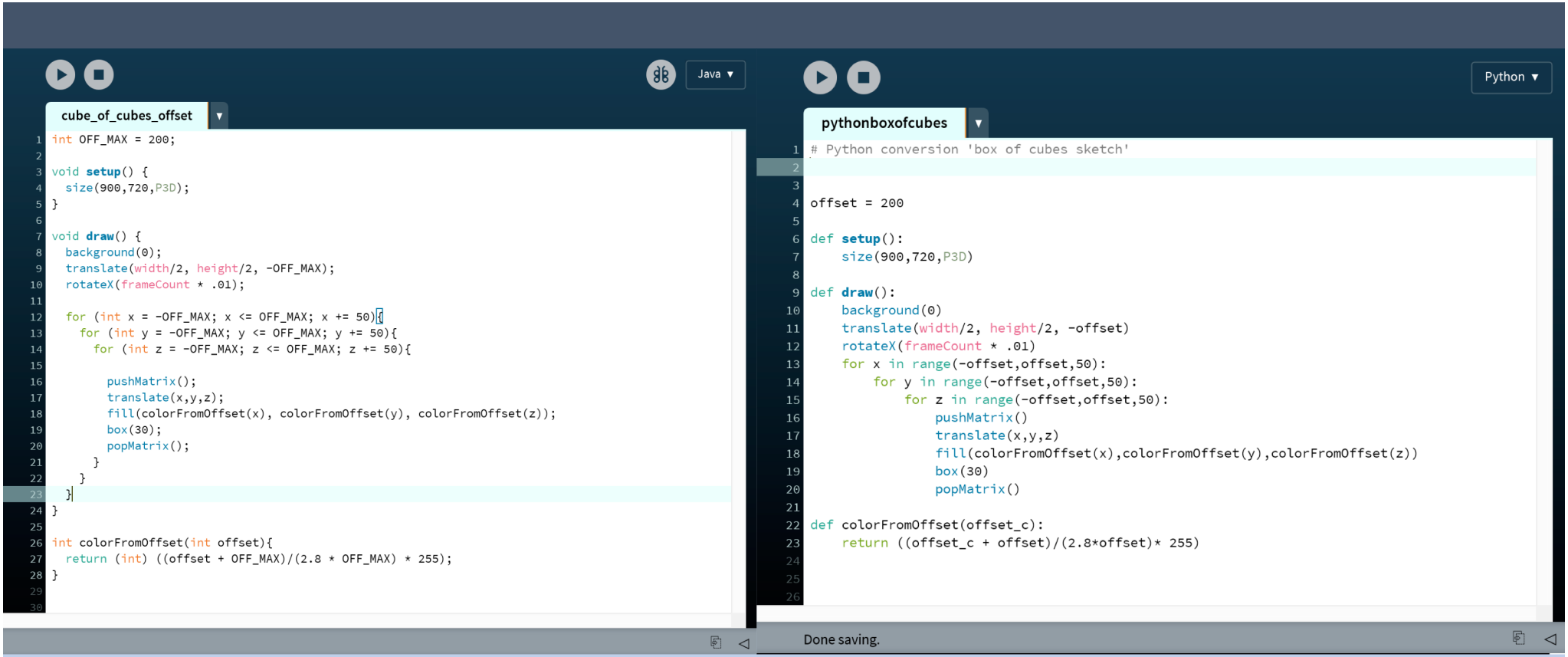
- Python mode available (some limitations).
- Daniel Shiffman youtube tutorials
- Can create OpenGL wrapper to host GLSL shaders
  - Book of Shaders



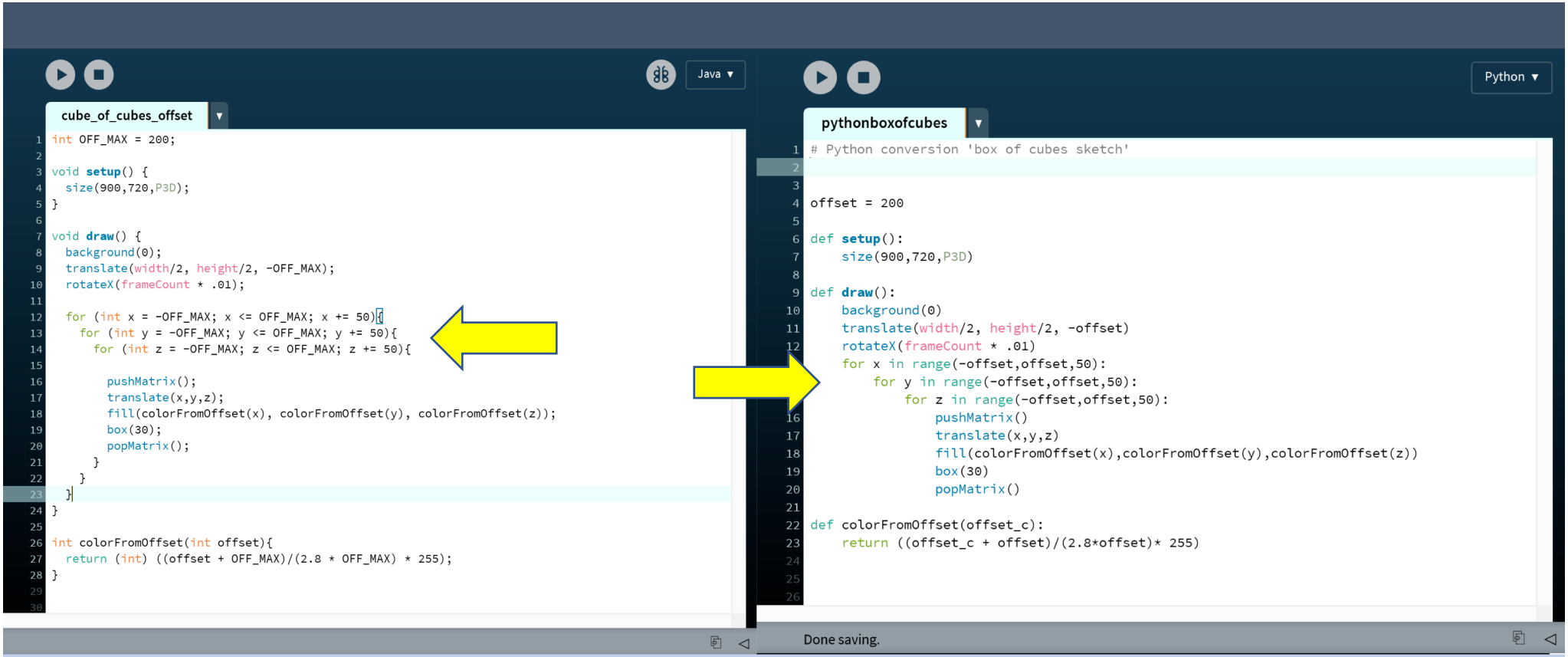
Things that wont work in processing.py:

- Numpy, scipy, and scikit-learn will not work in Jython, so will also be unavailable in processing.py.
- Libraries that depend on anything listed above will not work
  - ** A lot of these DO work in blender**


Python
sketch_3DCircle
attractor.py
mover.py
def __init__(self):
self.position = PVector(0, 0)
self.mass = 20
self.g = 0.4;
def attract(self, m):
force = PVector.sub(self.position, m.position)
d = force.mag()
d = constrain(d, 5.0, 25.0)
force.normalize();
strength = (self.g * self.mass * m.mass) / float(d * d)
force.mult(strength)
return force
def display(self):
stroke(199, 145, 44)
noFill()
pushMatrix()
translate(self.position.x, self.position.y, self.position.
sphere(self.mass*8)
popMatrix()

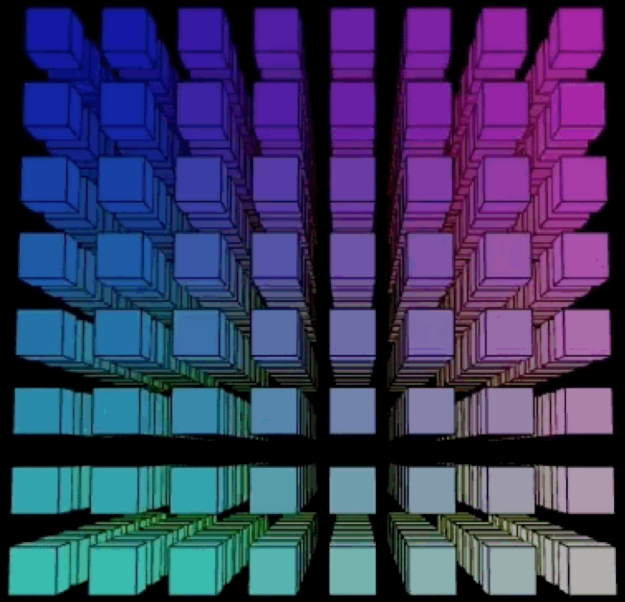


Easy to convert existing examples and code from java into python.



Easy to create java-like for loops using range function.

Both sketches have
the same output.



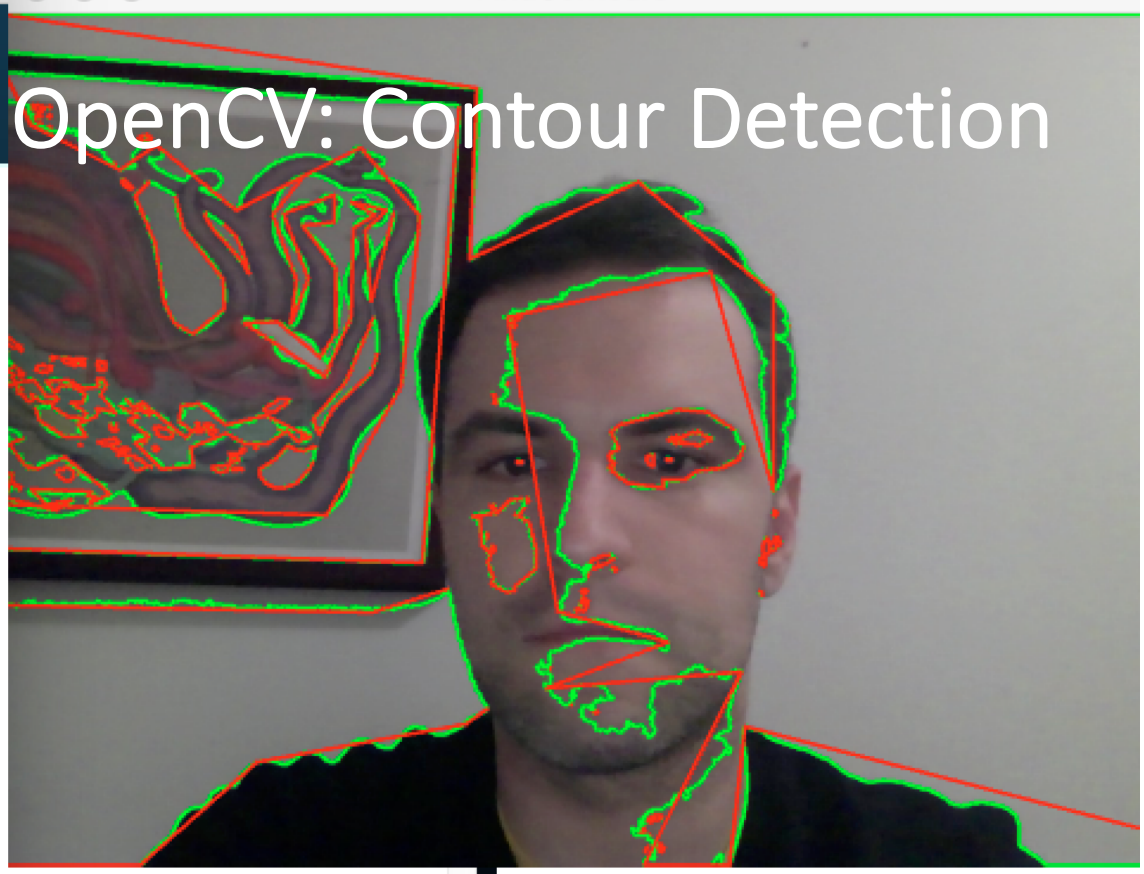
OpenCV: Contour Detection

FindContours | Processing 3.5.3

FindContours

```
1  add_library('opencv_processing')
2
3  src = loadImage("test.jpg")
4  size(480,480, P2D)
5  opencv = OpenCV(this, src)
6  opencv.gray()
7  opencv.threshold(70)
8  dst = opencv.getOutput()
9  contours = opencv.findContours()
10 print "found %d contours" % contours.size()
11
12 scale(0.5)
13 image(src, 0, 0)
14 image(dst, src.width, 0)
15 noFill()
16 strokeWeight(3)
17
18 for contour in contours:
19     stroke(0, 255, 0)
20     contour.draw()
21
22     stroke(255, 0, 0)
23     with beginShape():
24         for point in contour.getPolygonApproximation().getPoints():
25             vertex(point.x, point.y)
26
27
```

FindContours



OpenCV: Face Detection

Python ▾

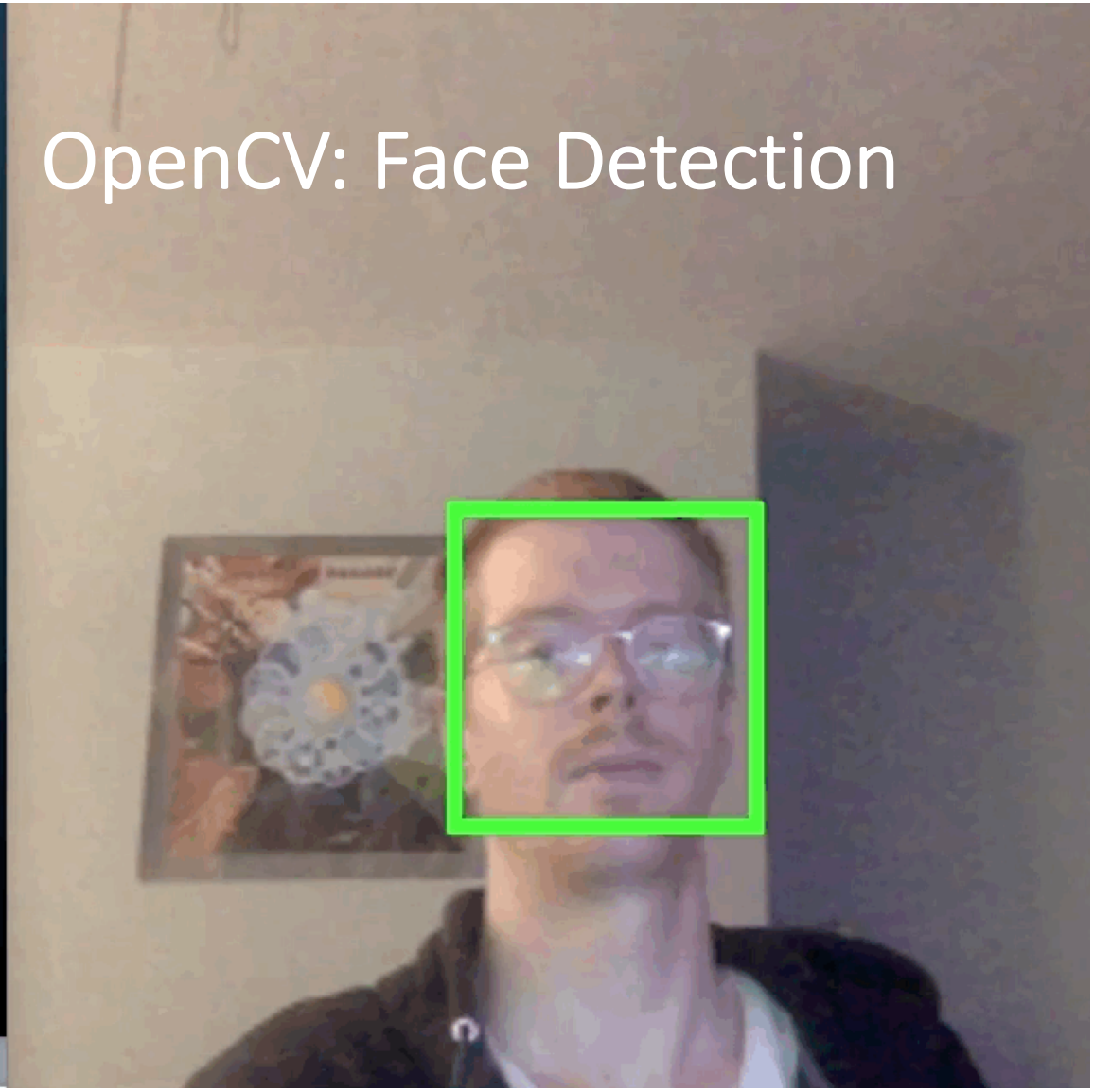
LiveCamTest ▾

```
    opencv.loadCascade(OpenCV.CASCADE_FRONTALFACE)
    video.start()
    frameRate(60)


def draw():
    scale(3)
    opencv.loadImage(video)
    image(video, 0, 0)
    noFill()
    stroke(0, 255, 0)
    strokeWeight(3)
    #background(back)
    faces = opencv.detect()
    if back:
        background(0)
    for face in faces:
        rect(face.x, face.y, face.width, face.height)
    if show_fps:
        text("%d fps" % frameRate, 20, 20)


def captureEvent(c):
    c.read()

def mousePressed():
    global show_fps
    show_fps = not show_fps
    global back
    back = not back
```

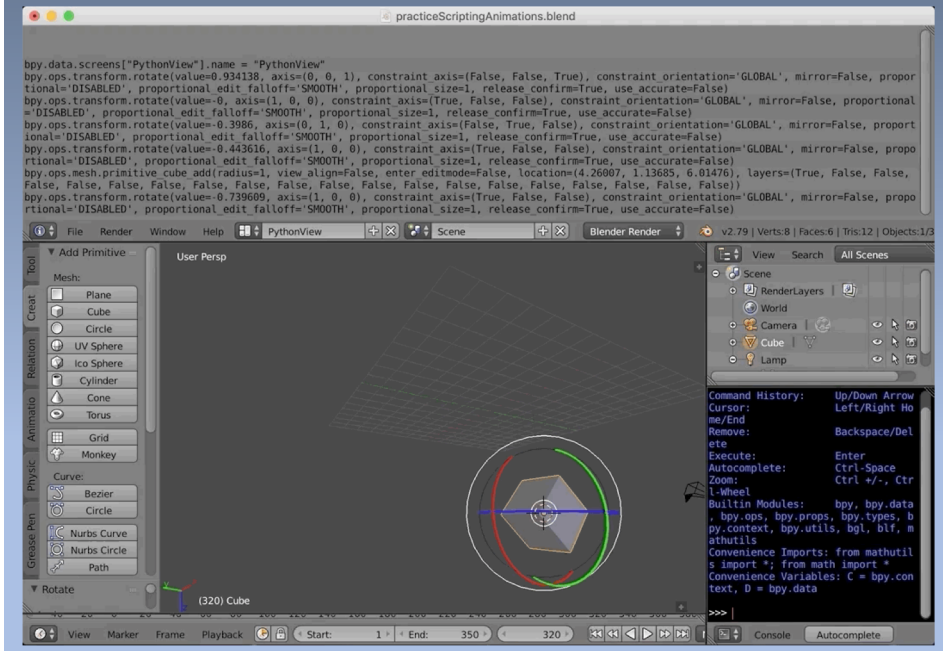


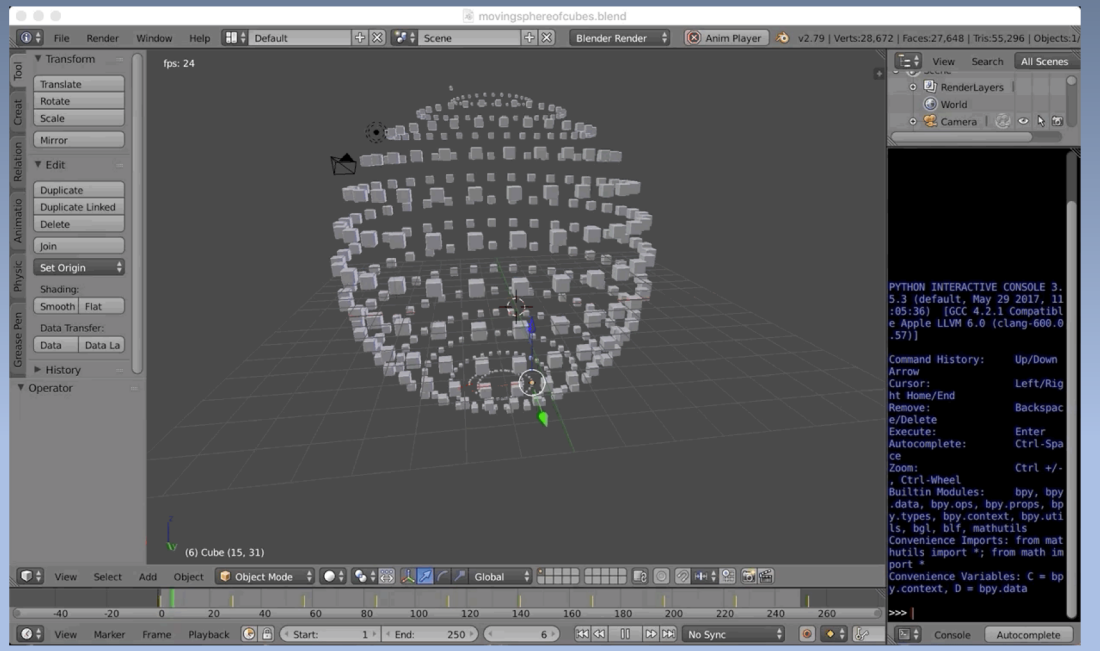
Blender

- Open source 3D design suite
- Jeremy Berheandt “Creative Coding in Blender: A Primer”
- “Code-Generator” feature
- Python API for BlenderVR

Motion Capture in Blender with OpenCV

What's possible in blender?



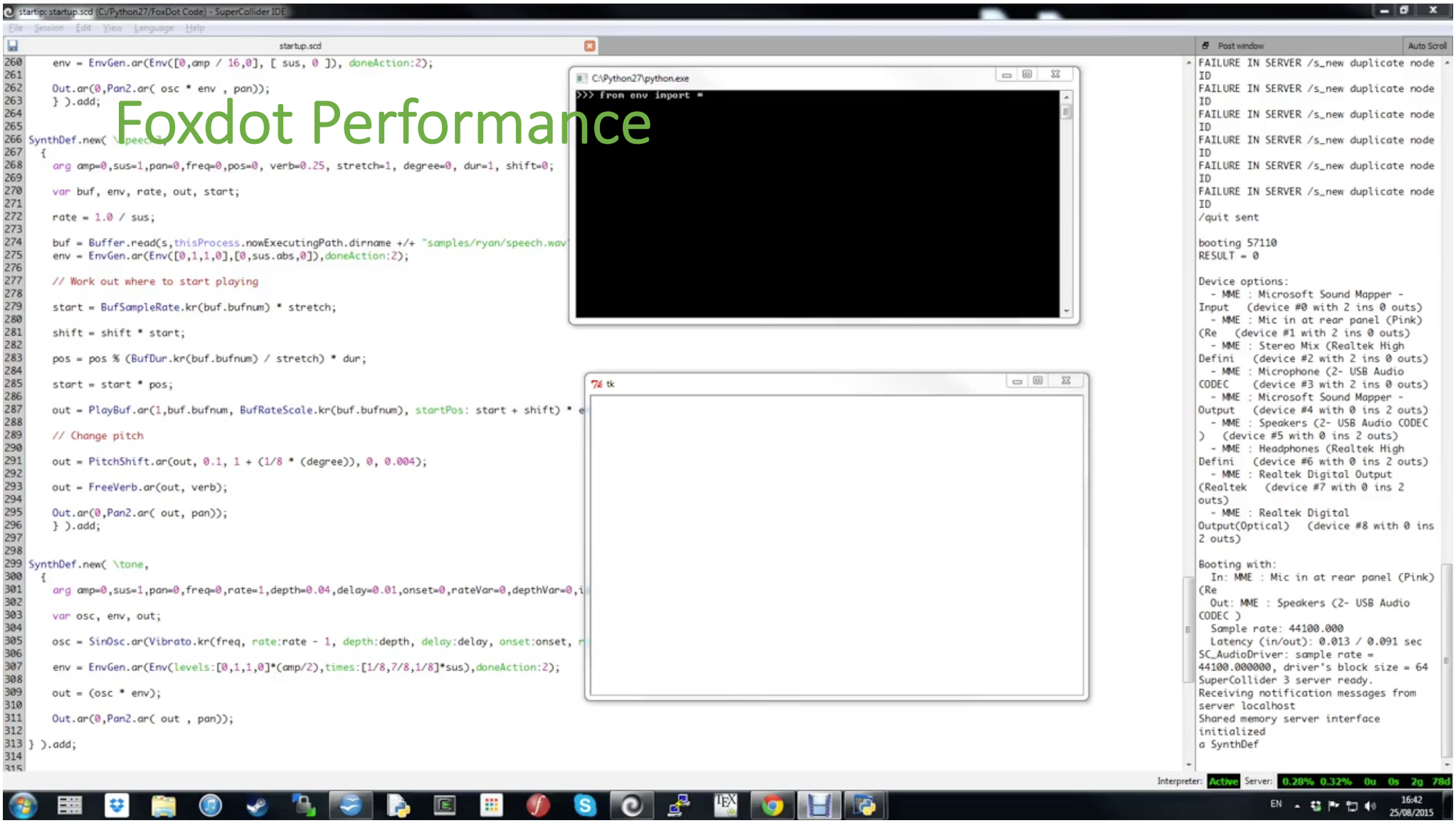


Creative Hacking & Making Connections

- Embrace programming in general not just programming in python, most important concepts are similar or the same in may programming languages, just small changes in syntax.

FoxDot

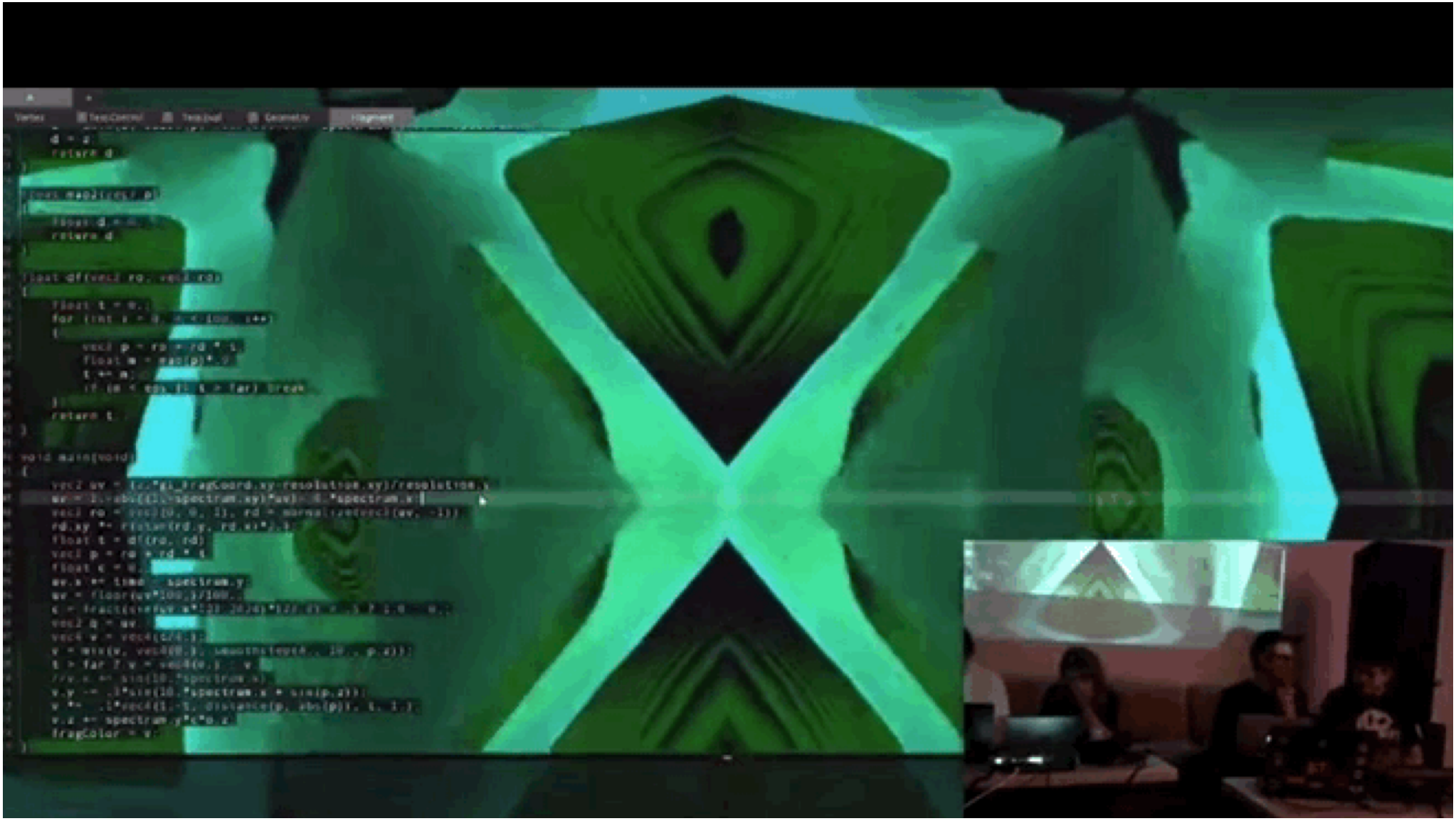
- Created and maintained by Ryan Kirkbride, Leeds UK
- Tool for Live coding audio patterns
- ** Would be a useful tool to send control messages to something like processing, or another visual platform you'd like to use.
- What is live coding?
- Using core python to do strange things in FoxDot


Foxdot Performance
startup.scd (C:/Python27/FoxDot Code) - SuperCollider IDE
File Session Edit View Language Help
startup.scd
env = EnvGen.ar(Env([0,amp / 16,0], [sus, 0]), doneAction:2);
Out.ar(0,Pan2.ar(osc * env , pan));
}).add;
SynthDef.new(\speech,
{
arg amp=0,sus=1,pan=0,freq=0,pos=0, verb=0.25, stretch=1, degree=0, dur=1, shift=0;
var buf, env, rate, out, start;
rate = 1.0 / sus;
buf = Buffer.read(s,thisProcess.nowExecutingPath.dirname +/+ "samples/ryan/speech.wav
env = EnvGen.ar(Env([0,1,1,0],[0,sus.abs,0]),doneAction:2);
// Work out where to start playing
start = BufSampleRate.kr(buf.bufnum) * stretch;
shift = shift * start;
pos = pos % (BufDur.kr(buf.bufnum) / stretch) * dur;
start = start * pos;
out = PlayBuf.ar(1,buf.bufnum, BufRateScale.kr(buf.bufnum), startPos: start + shift) * e
// Change pitch
out = PitchShift.ar(out, 0.1, 1 + (1/8 * (degree)), 0, 0.004);
out = FreeVerb.ar(out, verb);
Out.ar(0,Pan2.ar(out, pan));
}).add;
SynthDef.new(\tone,
{
arg amp=0,sus=1,pan=0,freq=0,rate=1,depth=0.04,delay=0.01,onset=0,rateVar=0,depthVar=0,i
var osc, env, out;
osc = SinOsc.ar(Vibrato.kr(freq, rate:rate - 1, depth:depth, delay:delay, onset:onset, r
env = EnvGen.ar(Env(levels:[0,1,1,0]*(amp/2),times:[1/8,7/8,1/8]*sus),doneAction:2);
out = (osc * env);
Out.ar(0,Pan2.ar(out , pan));
}).add;
C:\Python27\python.exe
>>> from env import *
tk
Post window
Auto Scroll
FAILURE IN SERVER /s_new duplicate node ID
FAILURE IN SERVER /s_new duplicate node ID
FAILURE IN SERVER /s_new duplicate node ID
FAILURE IN SERVER /s_new duplicate node ID
FAILURE IN SERVER /s_new duplicate node ID
FAILURE IN SERVER /s_new duplicate node ID
/quit sent
booting 57110
RESULT = 0
Device options:
- MME : Microsoft Sound Mapper - Input (device #0 with 2 ins 0 outs)
- MME : Mic in at rear panel (Pink) (Re (device #1 with 2 ins 0 outs)
- MME : Stereo Mix (Realtek High Defini (device #2 with 2 ins 0 outs)
- MME : Microphone (2- USB Audio CODEC (device #3 with 2 ins 0 outs)
- MME : Microsoft Sound Mapper - Output (device #4 with 0 ins 2 outs)
- MME : Speakers (2- USB Audio CODEC) (device #5 with 0 ins 2 outs)
- MME : Headphones (Realtek High Defini (device #6 with 0 ins 2 outs)
- MME : Realtek Digital Output (Realtek (device #7 with 0 ins 2 outs)
- MME : Realtek Digital Output(Optical) (device #8 with 0 ins 2 outs)
Booting with:
In: MME : Mic in at rear panel (Pink) (Re
Out: MME : Speakers (2- USB Audio CODEC)
Sample rate: 44100.000
Latency (in/out): 0.013 / 0.091 sec
SC_AudioDriver: sample rate = 44100.000000, driver's block size = 64
SuperCollider 3 server ready.
Receiving notification messages from server localhost
Shared memory server interface initialized
a SynthDef
Interpreter: Active Server: 0.28% 0.32% 0u 0s 2g 78d
EN
16:42
25/08/2015

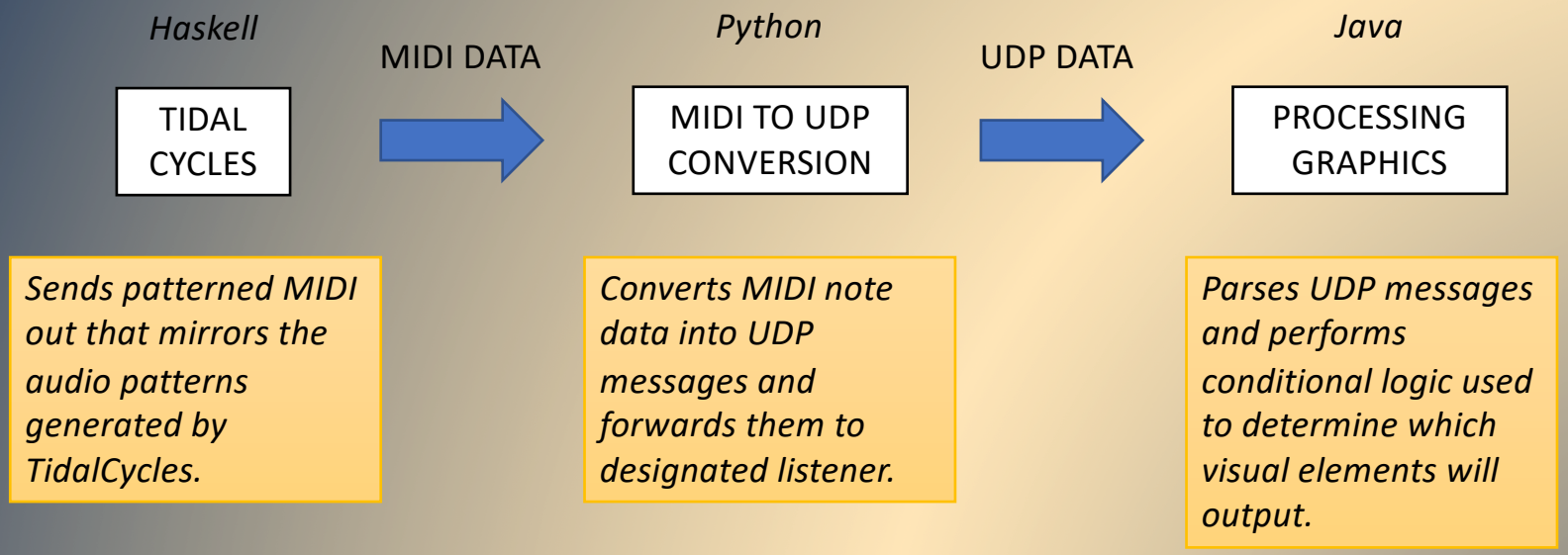

Local artists:
Danielle Rager
Kevin Bednar
Lukas Herman
Char Stiles
Austin Marcus
Chris Palmer
Samir Gangwani
Slow Danger
Adrienne Cassel
Projectile Objects

SLOWDANGER

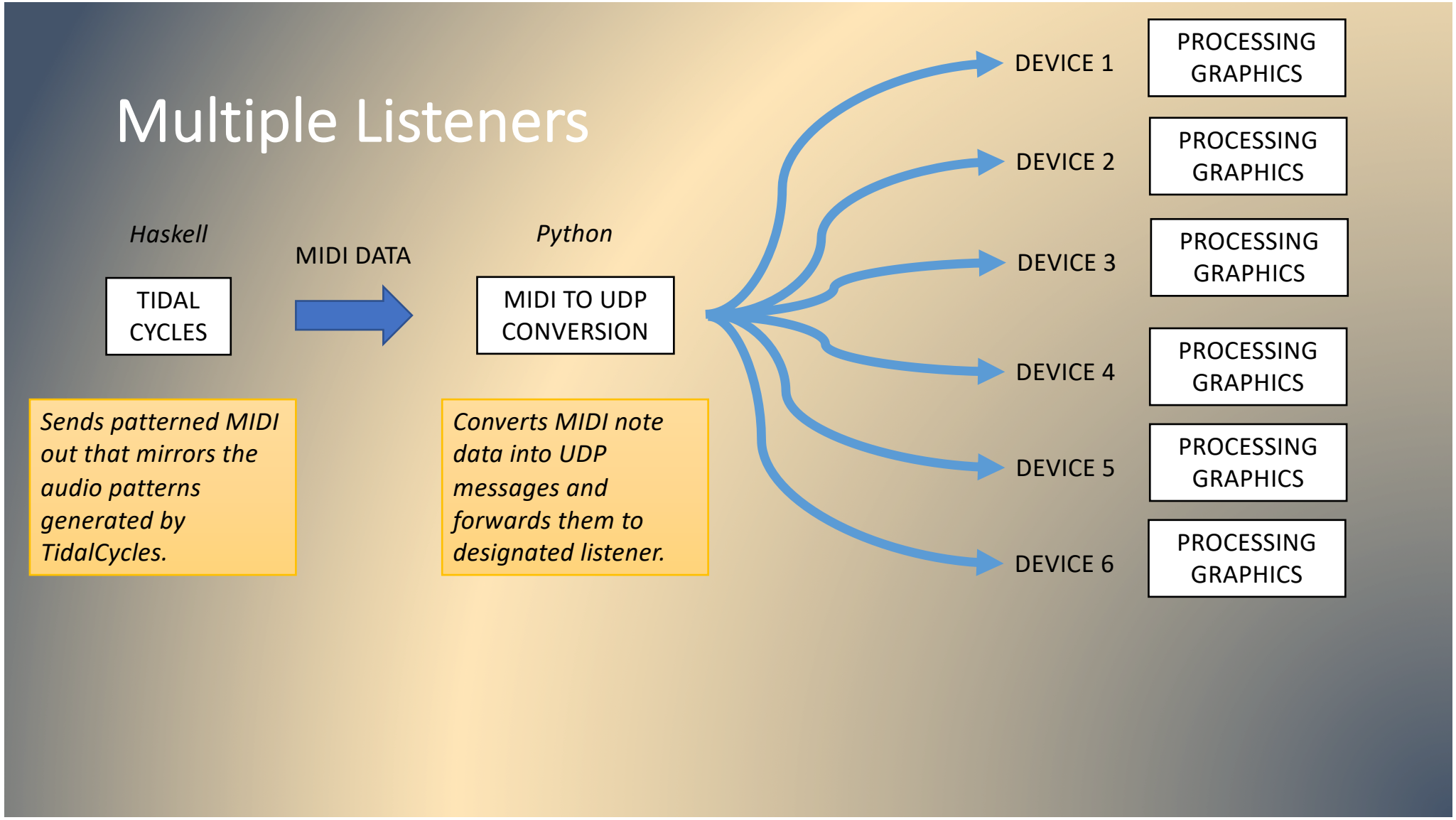
- Frequent collaboration with Projectile Objects & Char Stiles to bring a technology element to their live performance.
- Blog write-up's available here: https://projectileobjects.com/

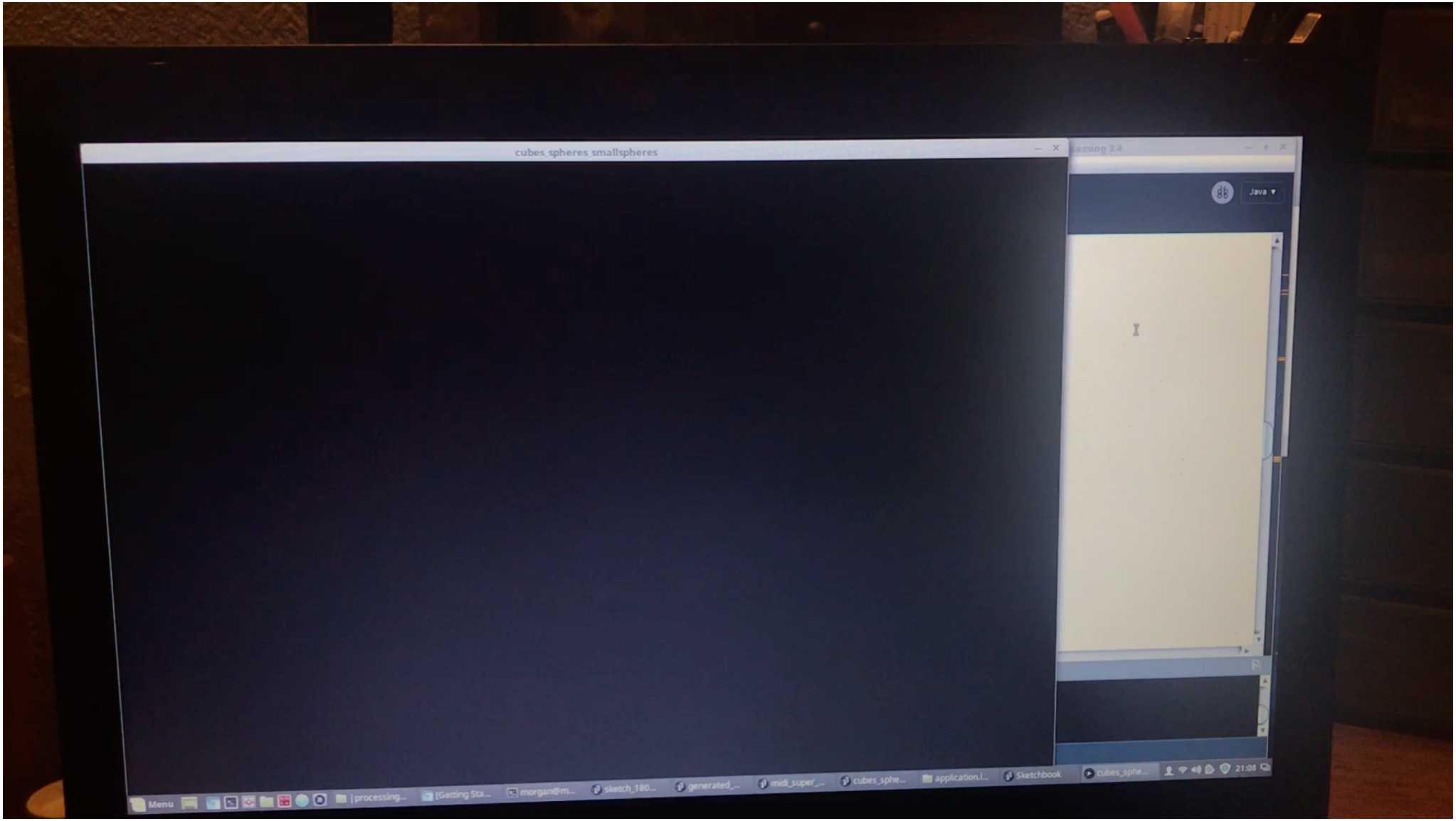


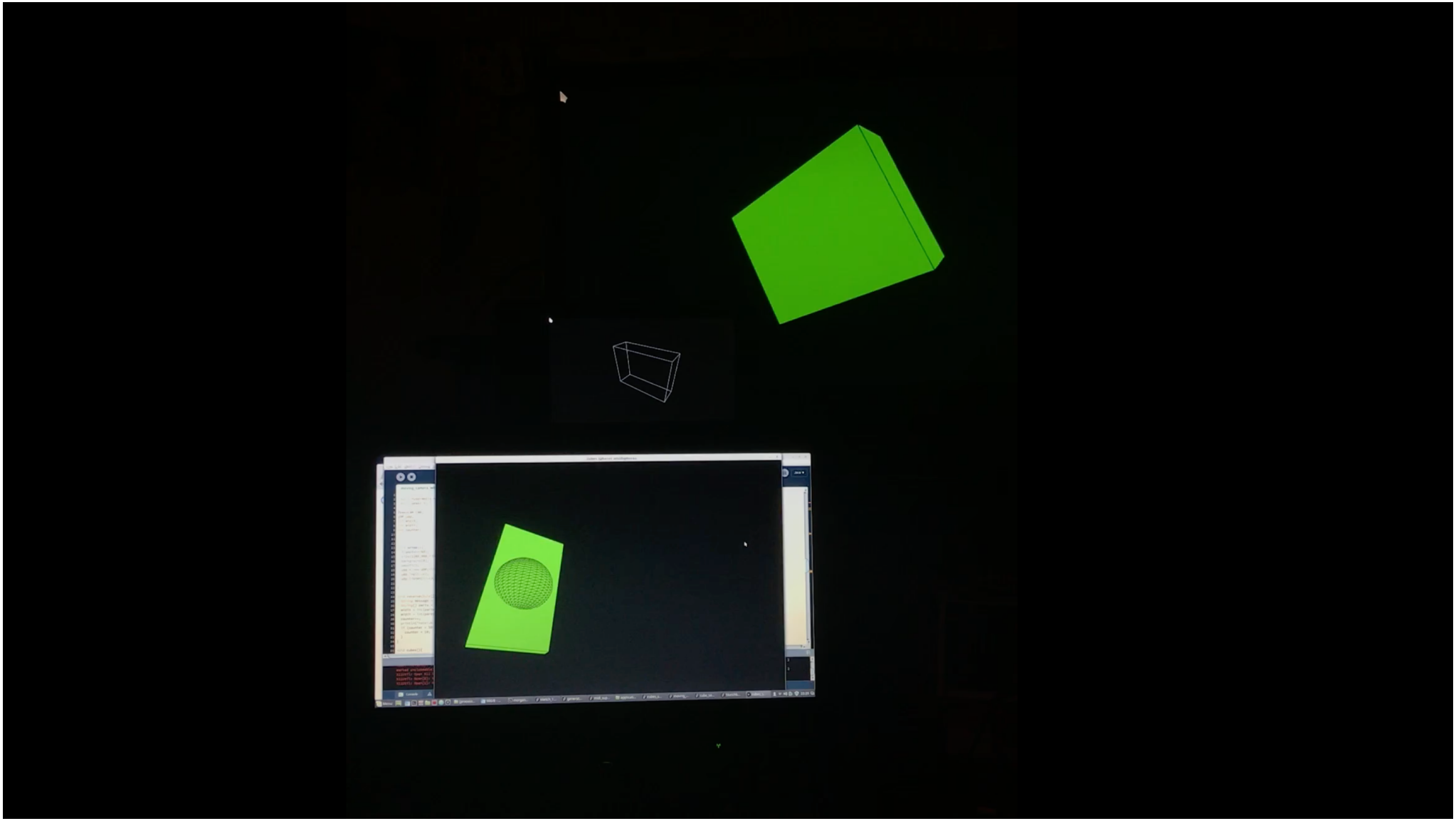
Personal Configuration

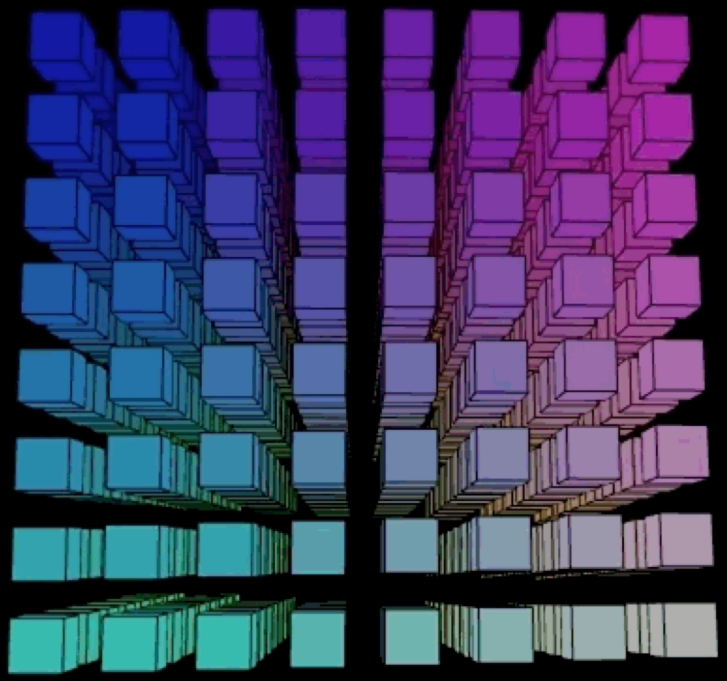


Multiple Listeners




cubes_spheres_smallspheres
Java
Menu
processing...
[Getting Sta...
morgan@m...
sketch_180...
generated_...
midi_super_...
cubes_sphe...
application.l...
Sketchbook
cubes_sphe...
21:08





MORE RESOURCES:
https://github.com/vism2889/imaginative_python

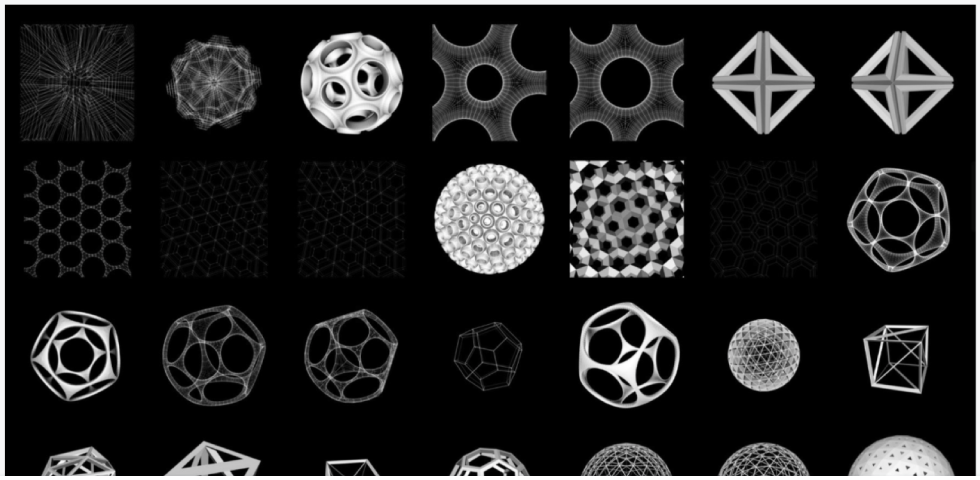
Meet-up: Generative Art and Adversarial Networks

Thursday, November 21, 2019

Generative Session

Hosted by Matt Turnshek and Hazem
From Generative Art and Adversarial Networks
Public group

Share



Thursday, November 21, 2019
6:30 PM to 9:30 PM
Add to calendar

STACK
5740 Baum Blvd. 3rd floor · Pittsburgh, PA

How to find us
After entering the building, head down the hall and then up the long staircase to the very top.

https://github.com/vism2889/imaginative_python