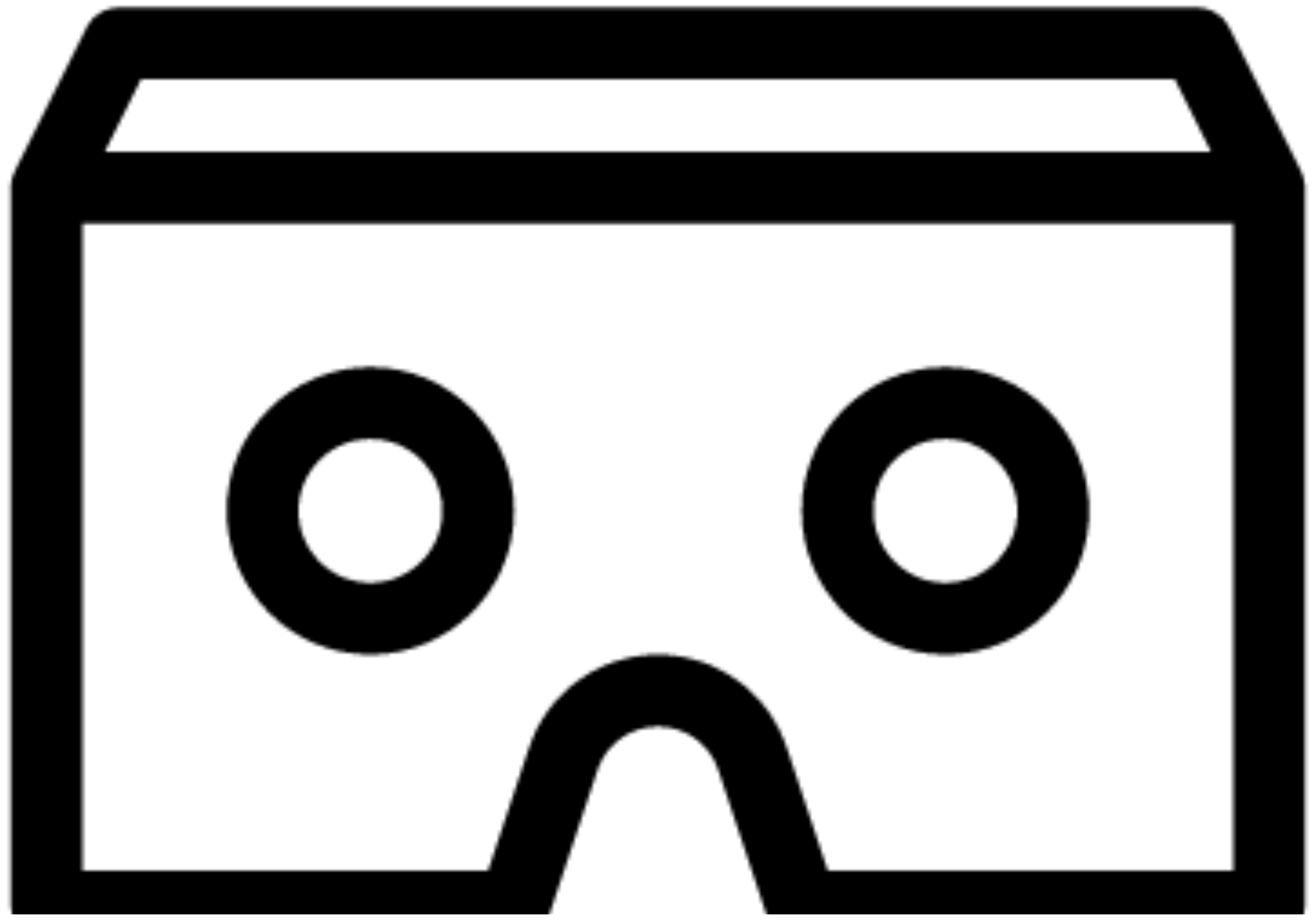
2016



I M M E R S I V E D A T A V I Z

For the Web

Alex
Norton

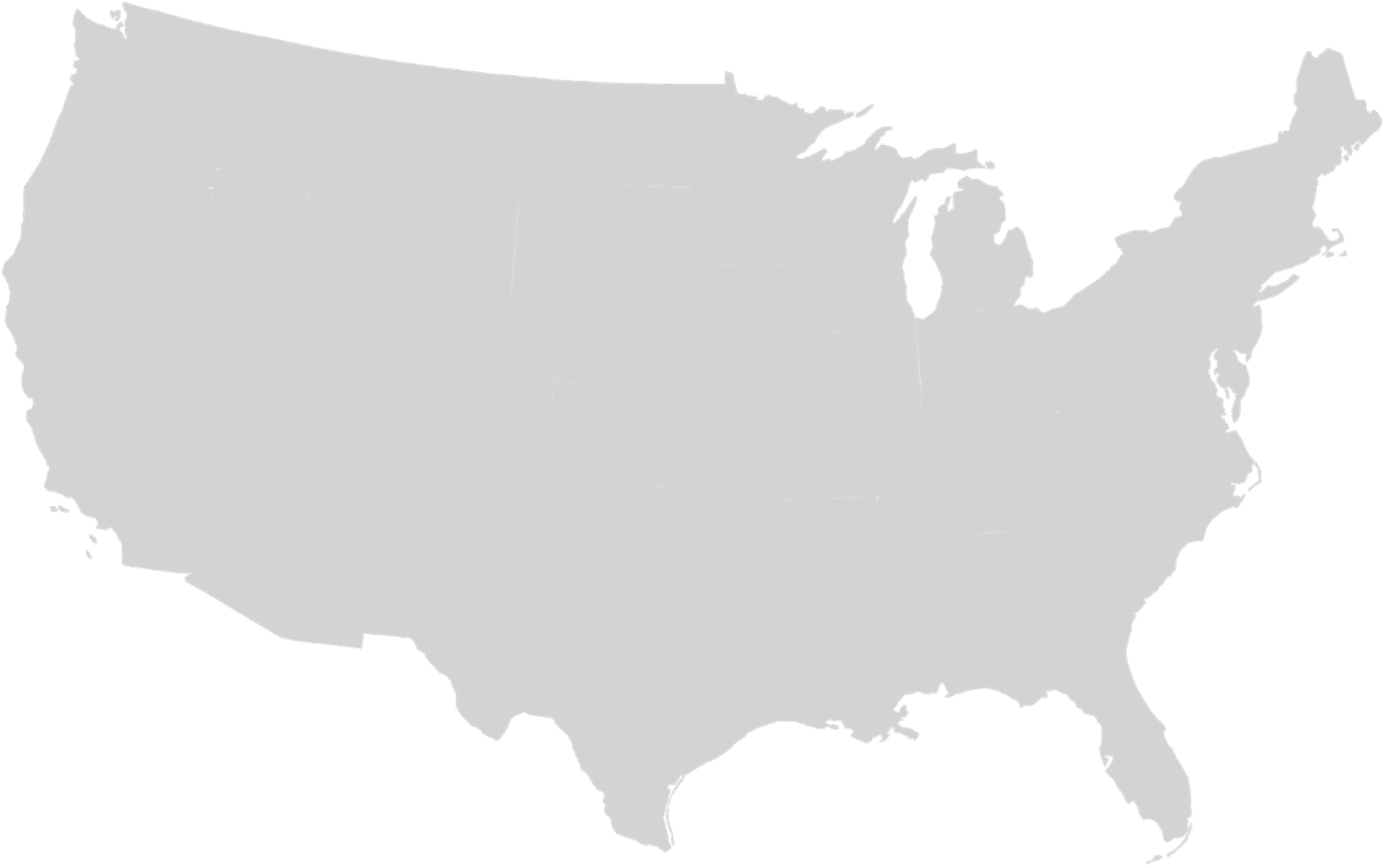
follow along!

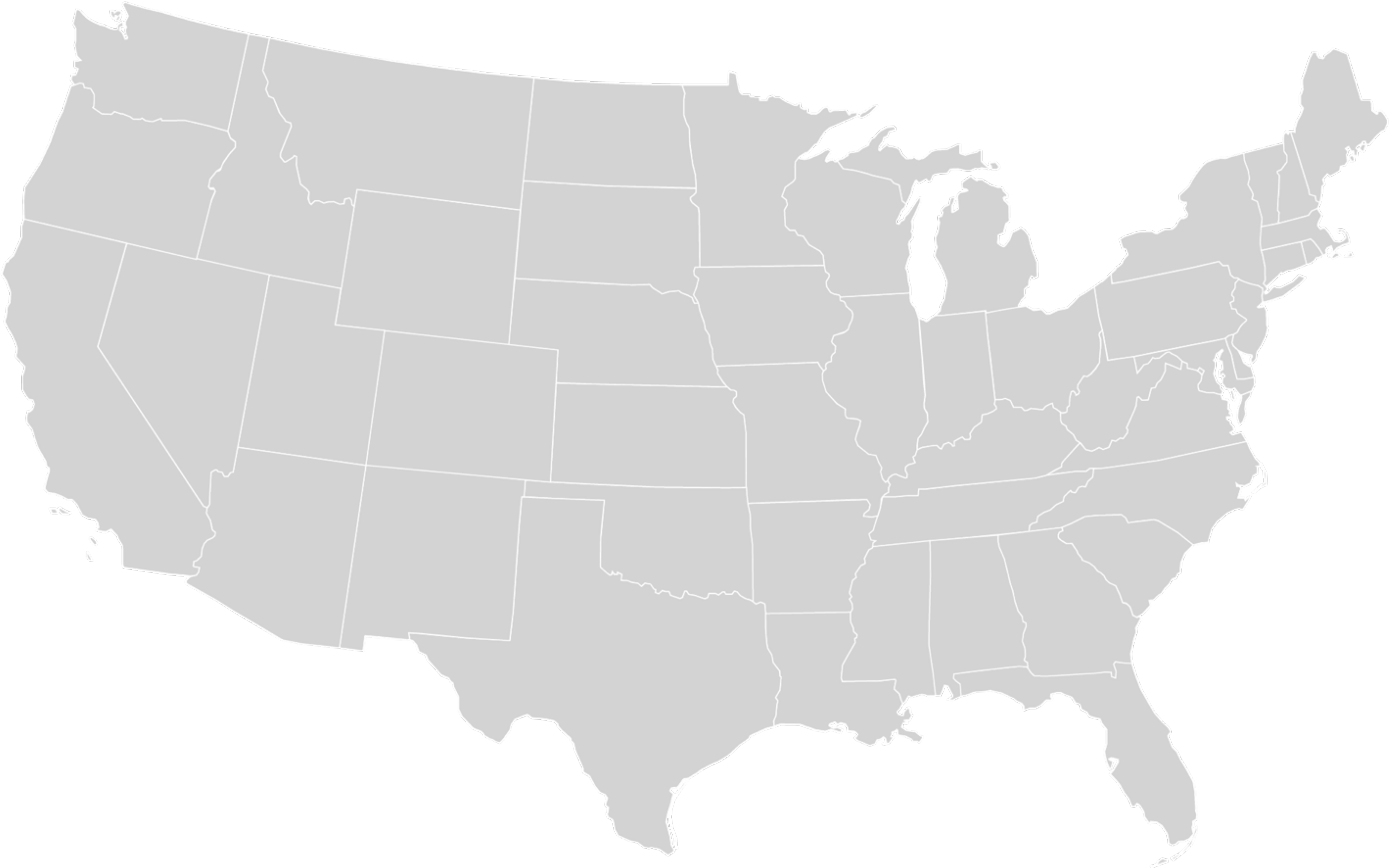
alexnortn.github.io/webVR/

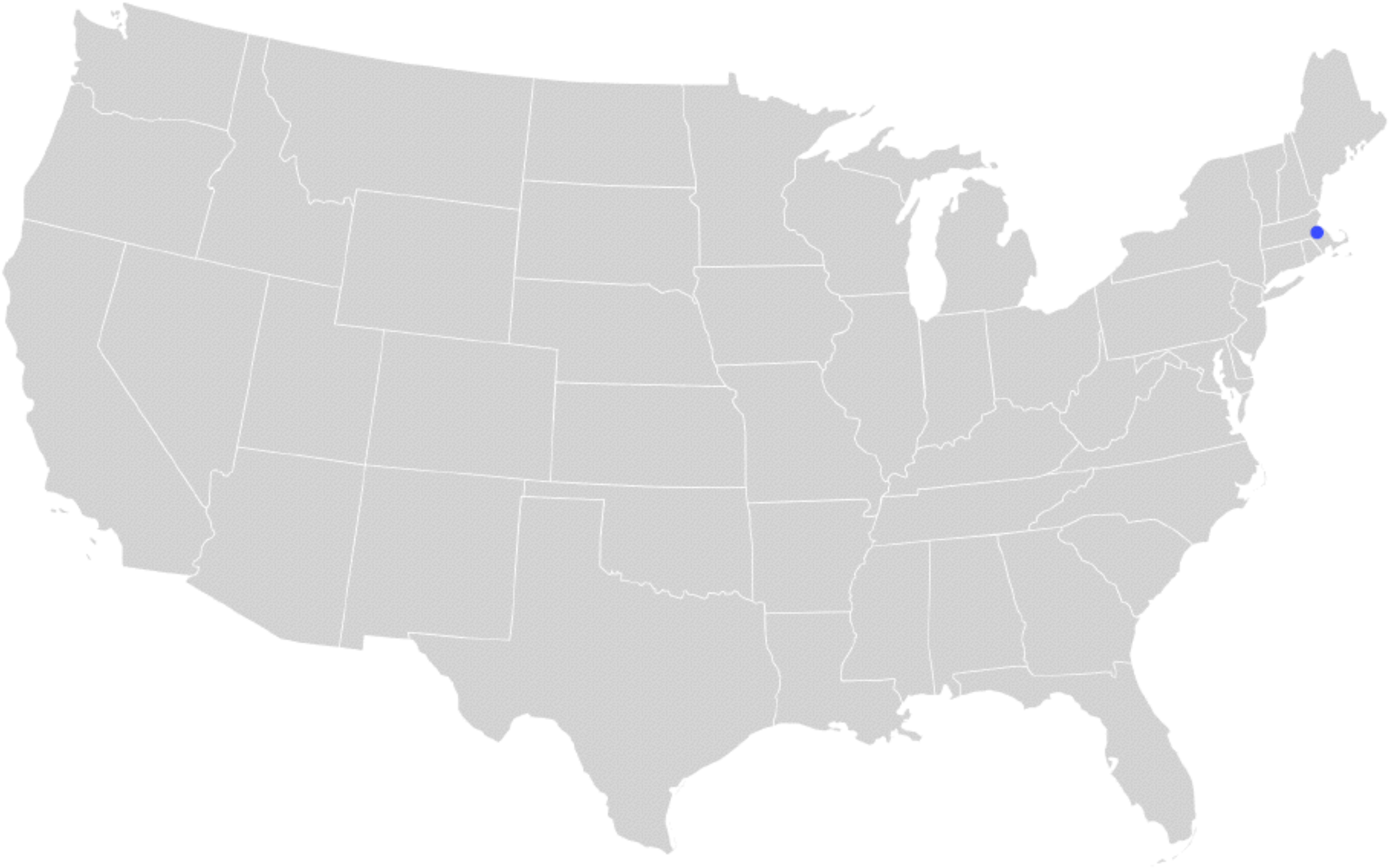
@alexnortn

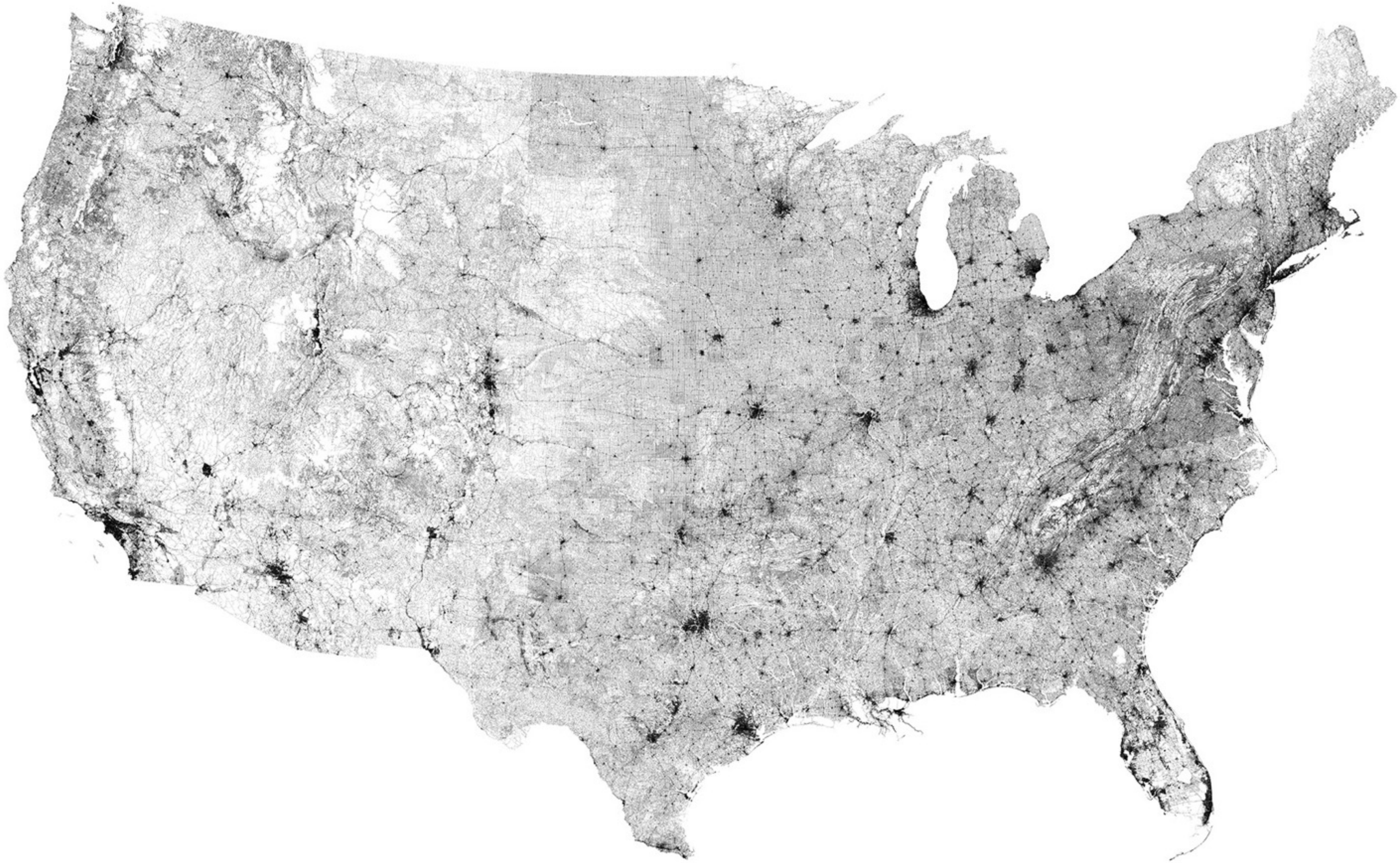
Alex
Norton

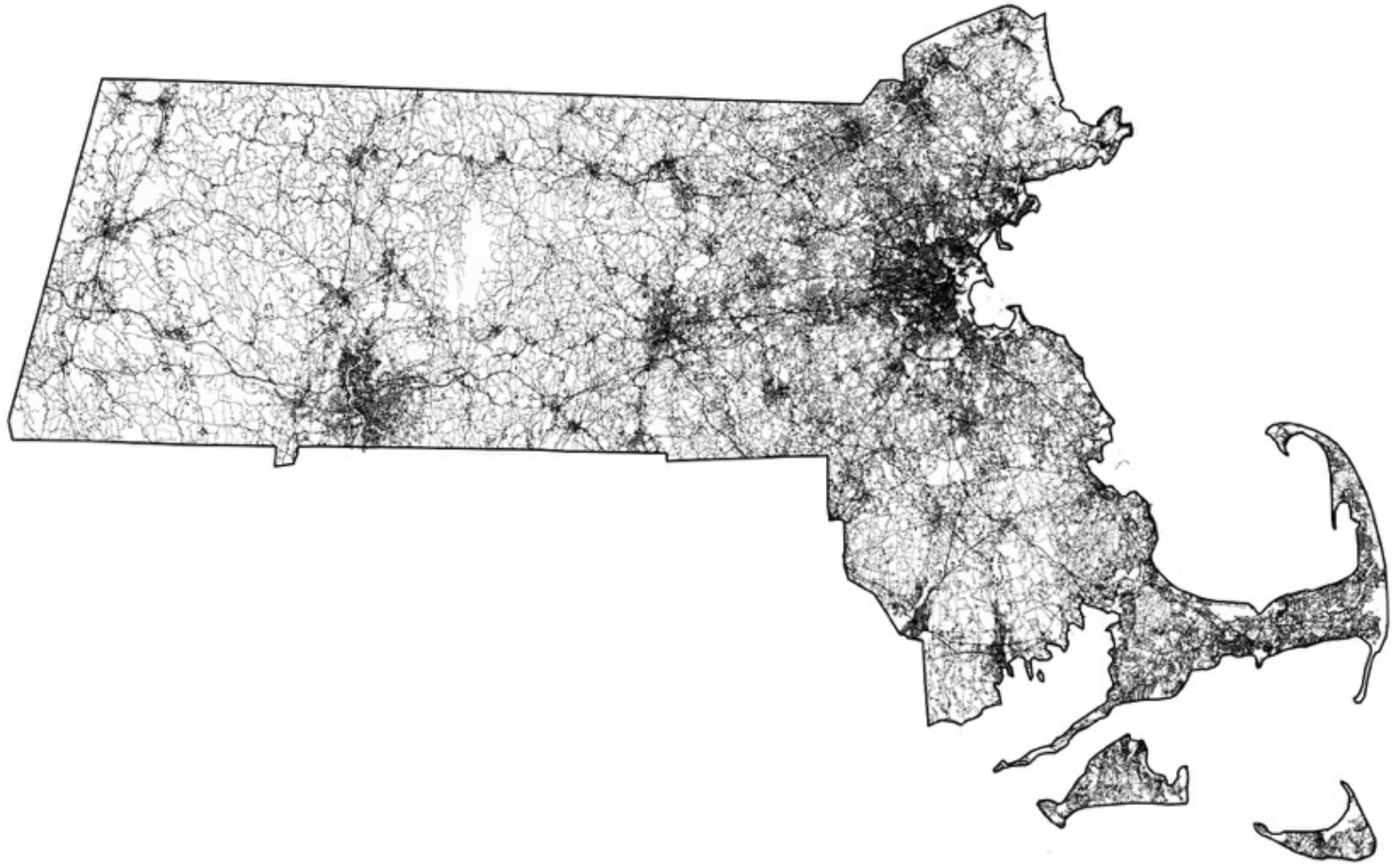
Interaction Designer
Creative Coder











Flight Patterns

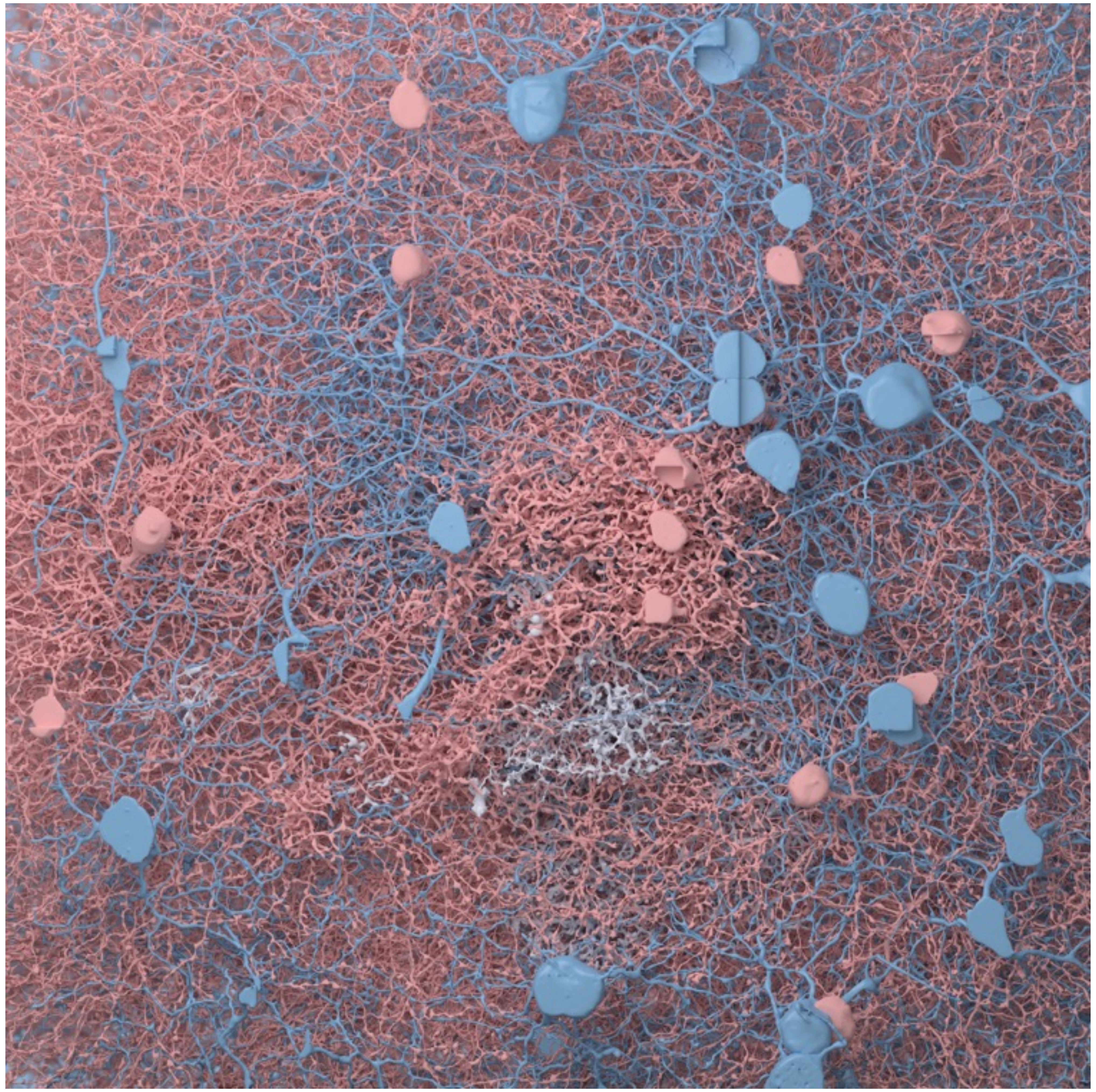
Aaron
Koblin

MY ROLL:

Make complex ideas + data accessible to a broader audience.

WHY

The complexity of our challenge demands collaboration.

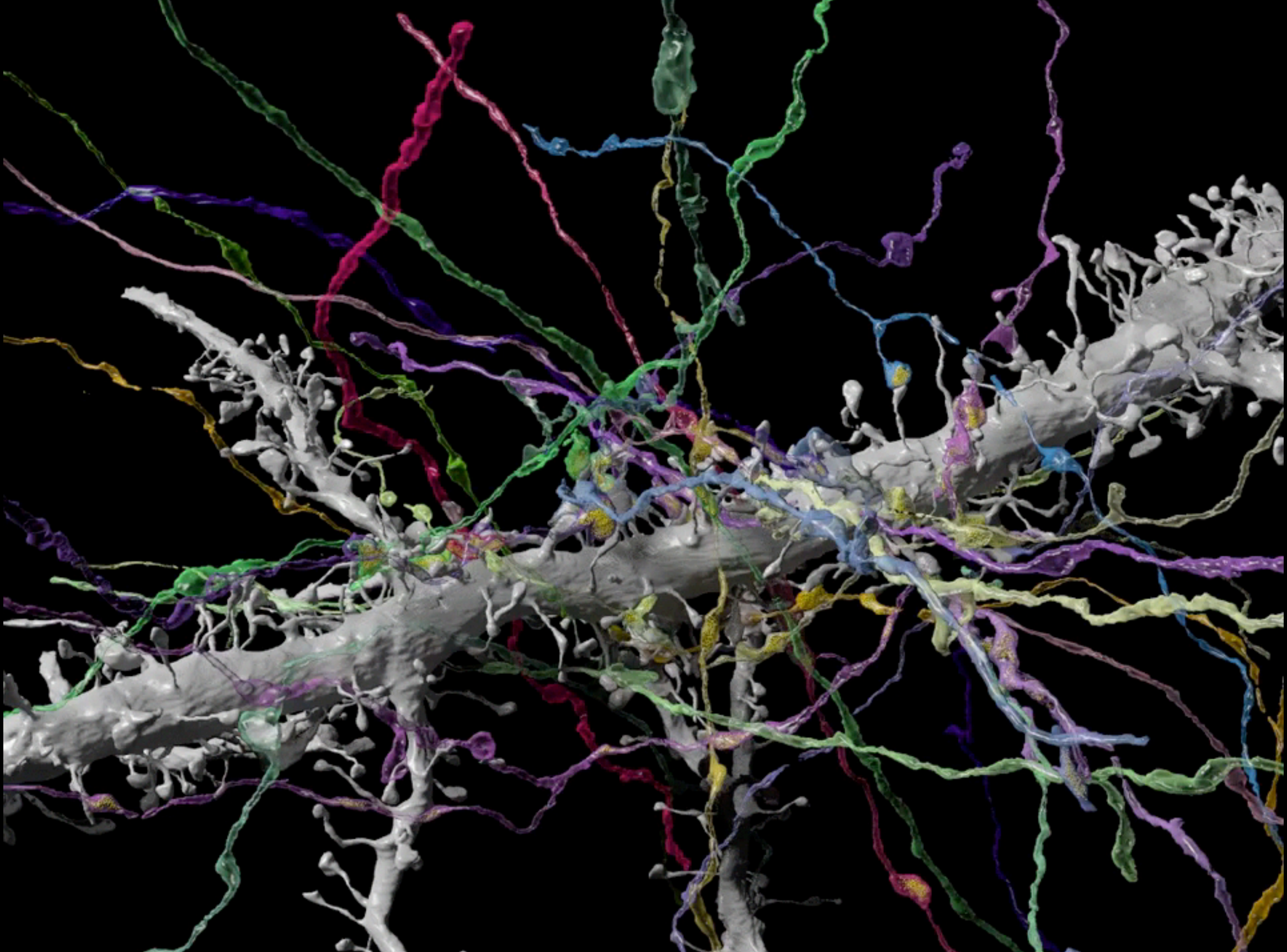


Eyewire

SHARE

A GAME TO MAP THE BRAIN

PLAY NOW



Neuron Museum | Seung Lab @ Princeton

ENTER MUSEUM

Digital Neural Reconstructions

The Seung Lab uses techniques from machine learning and social computing to extract brain structure from light and electron microscopic images. Eyewire showcases our approach by mobilizing gamers from around the world to create 3D reconstructions of neurons by interacting with a deep convolutional network.

This cell museum is where we present our finished reconstructions in an accessible format. In our viewer, you'll be able to see these finished cells represented in 3D space along with data pertinent to interpreting their significance. The neurons currently displayed here are mouse retinal cells from a dataset known as e2198.

You can read more about our lab here.

LINKS

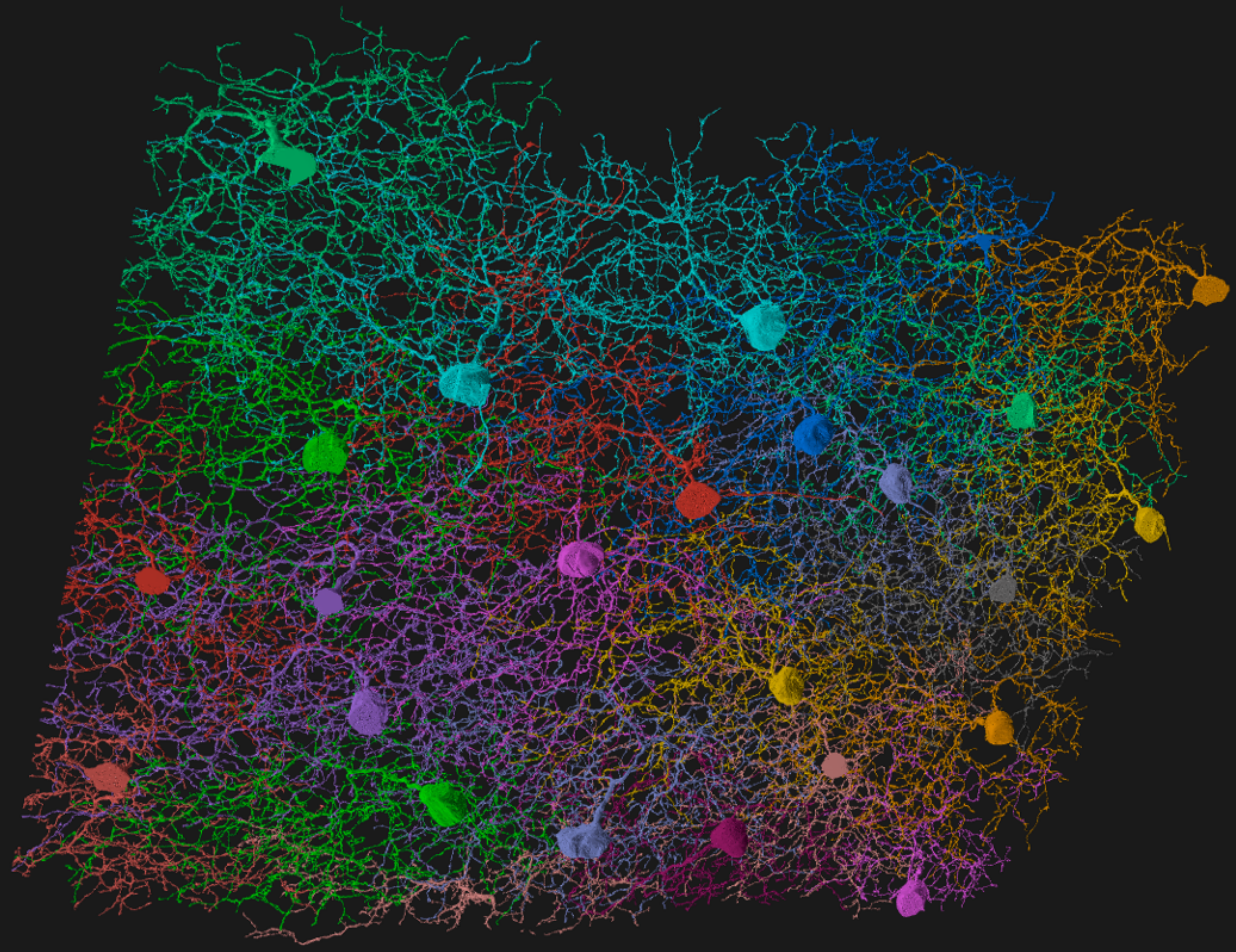
Seung Lab Website

Cell Wiki

Eyewire

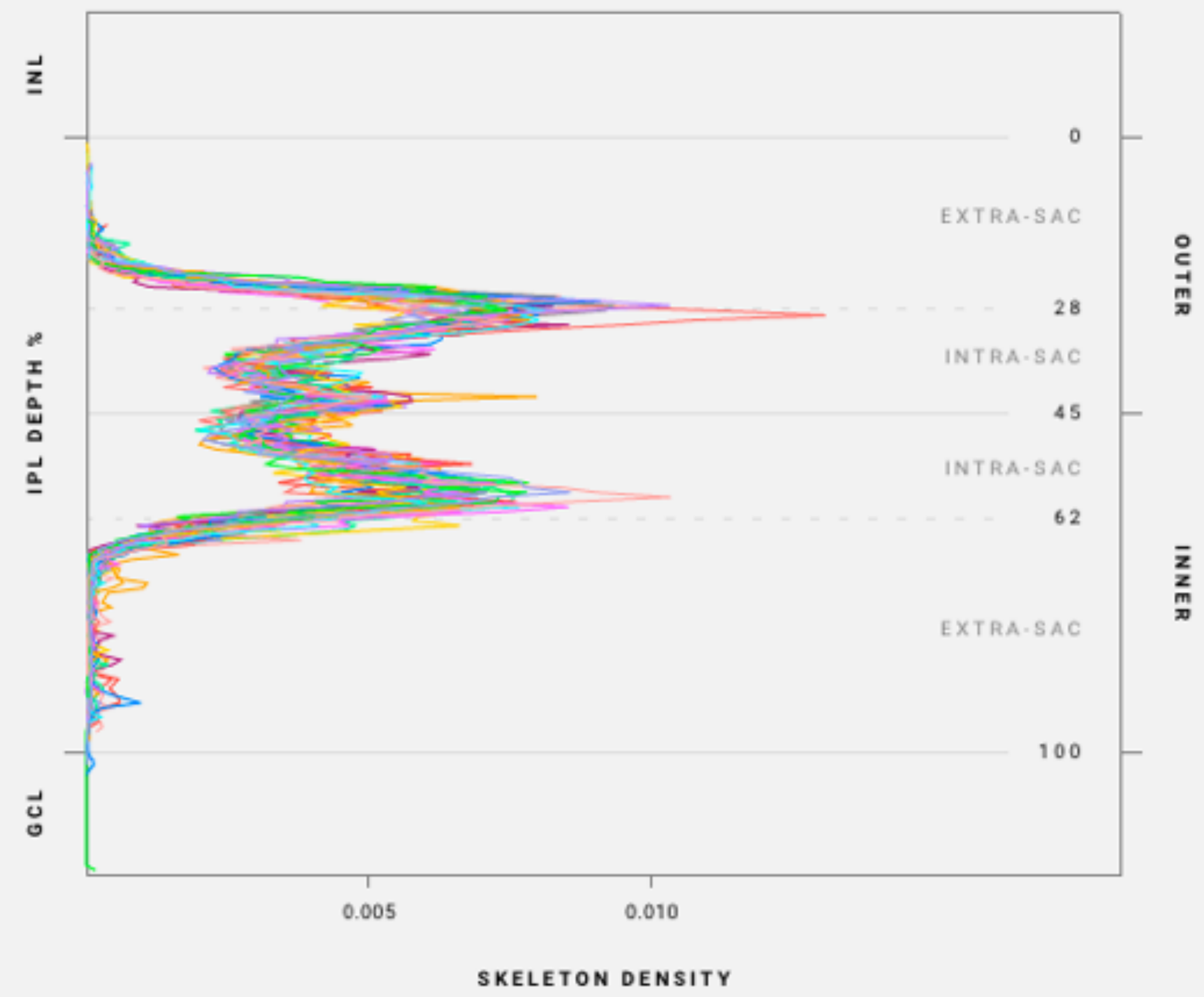
Classification System

Museum Search

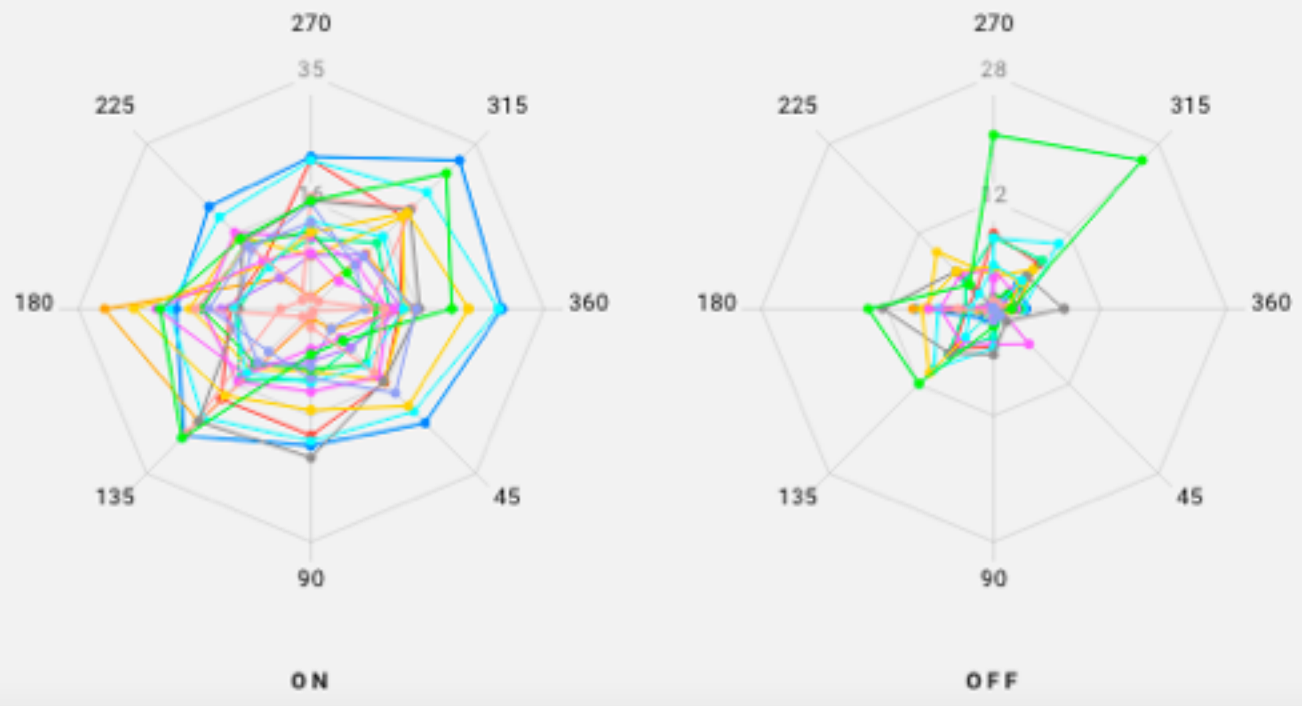


MOUSE RETINAL NEURONS

Characterization



DIRECTIONAL RESPONSE +



20208 20129 20140 20181 30003 20178 20005
20011 20019 30002 26028 26027 26057 26023
26068 26191 26089 26125 26141 26148 20071
17084 17097 17114 17140

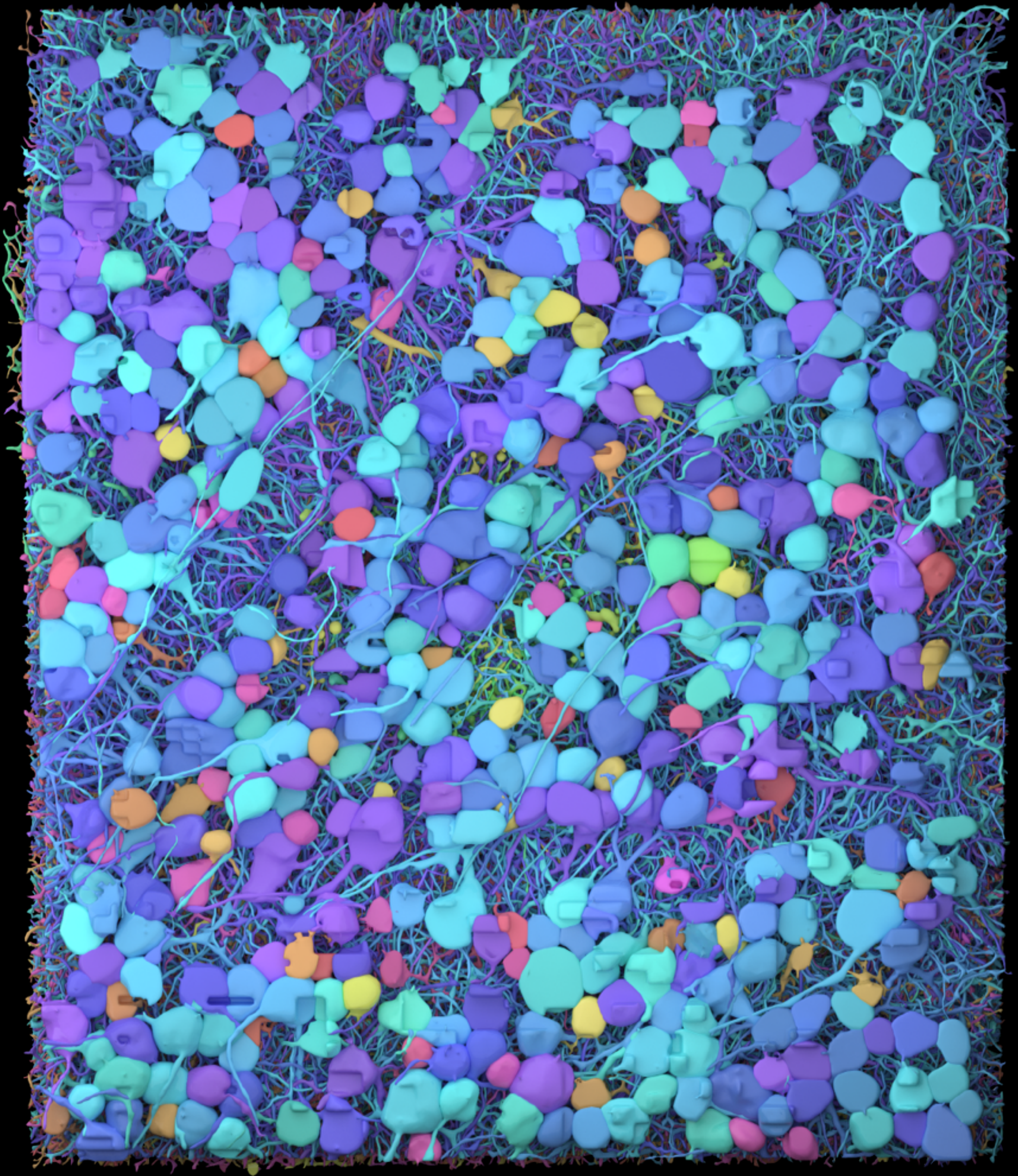
PROCESS

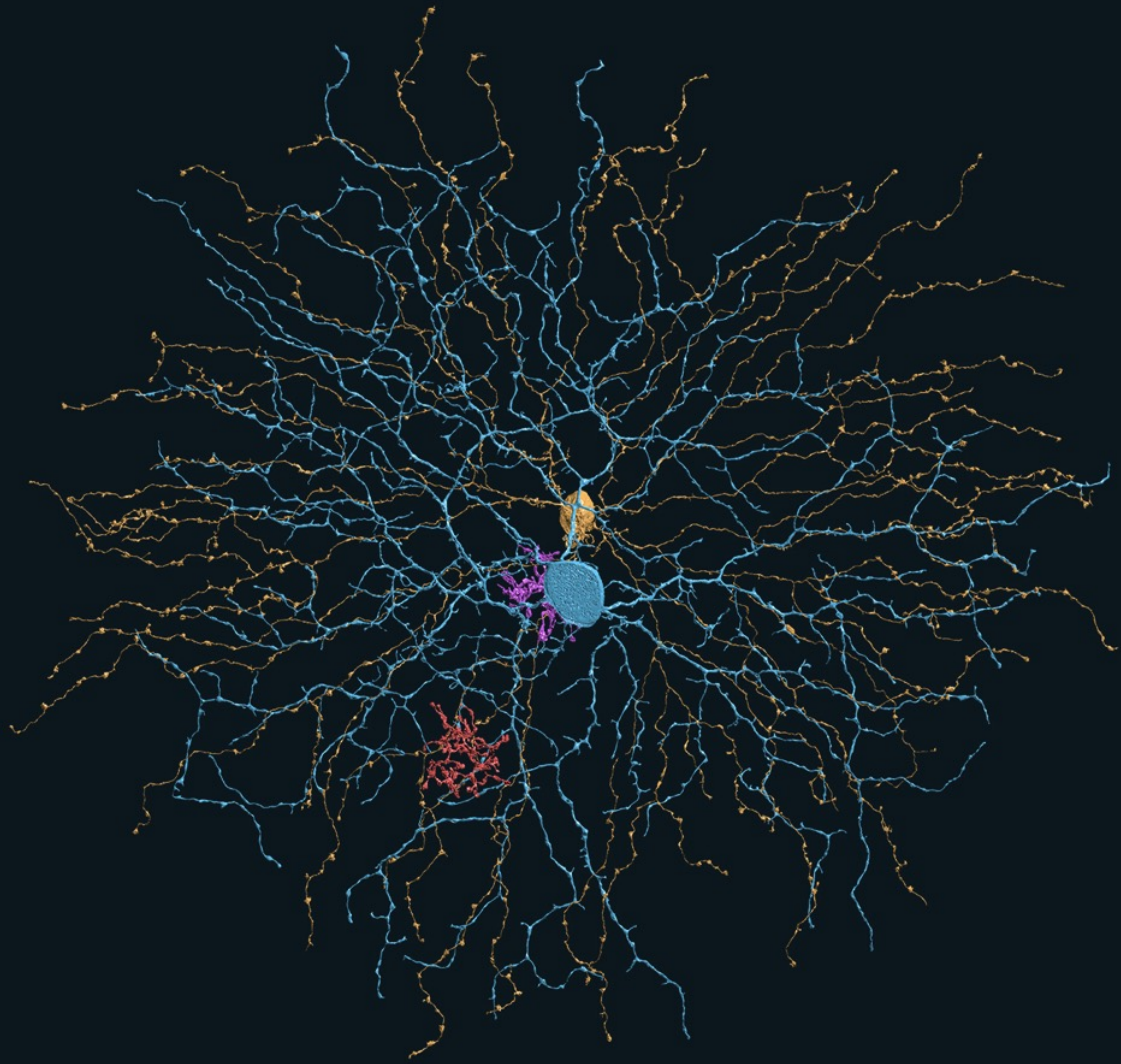
Design Process for Virtual Reality


BrainVR
Indicated

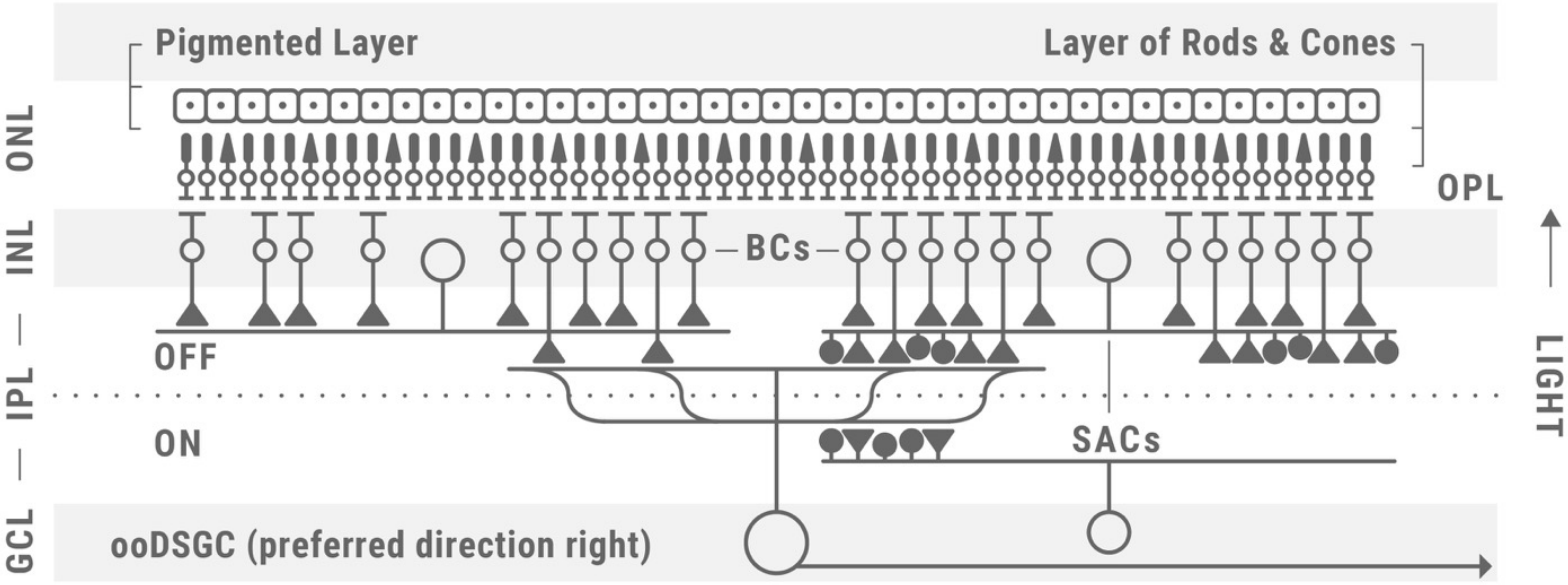
PROCESS

Design Process for Virtual Reality

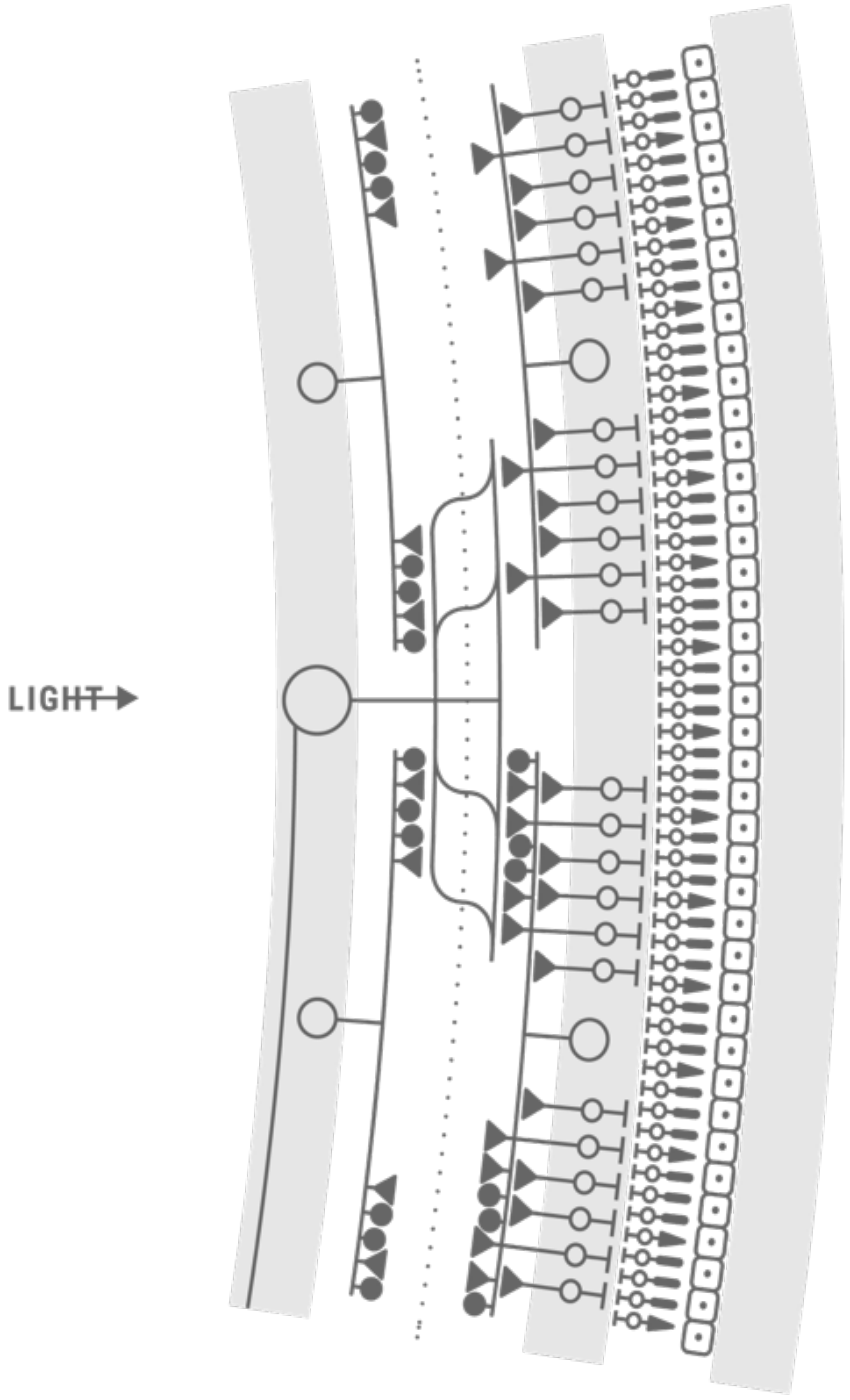





LIGHT

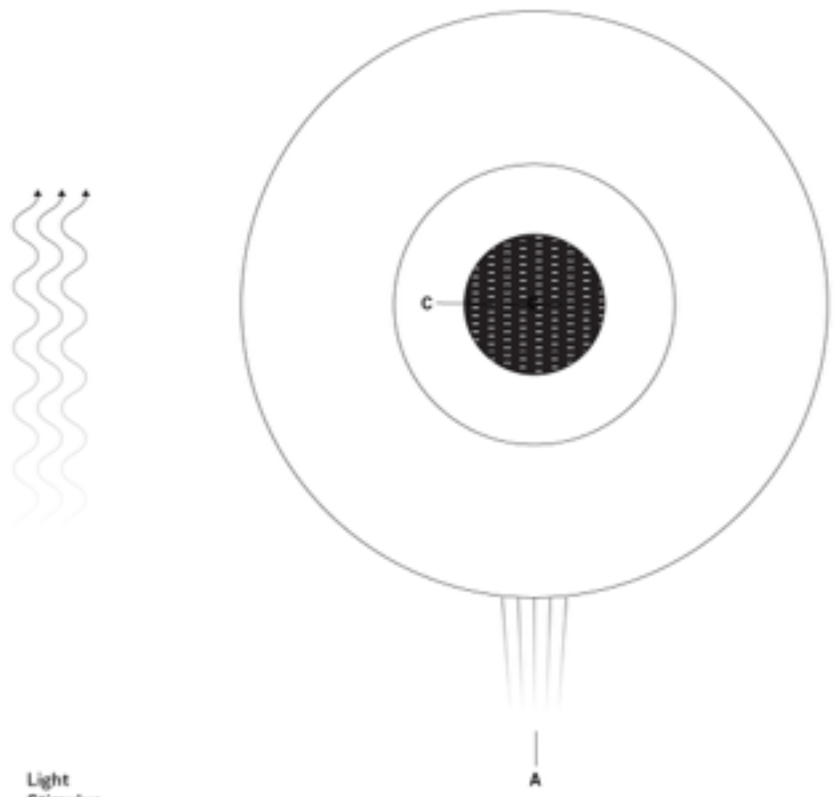


Circuitry of ooDSGC. BCs feed excitatory inputs (▲) to SAC and ooDSGCs. SAC feed inhibitory (●) inputs to ooDSGCs. ooDSGCs get more SAC inhibition ofrom the direction they prefer.


LIGHT

EYE

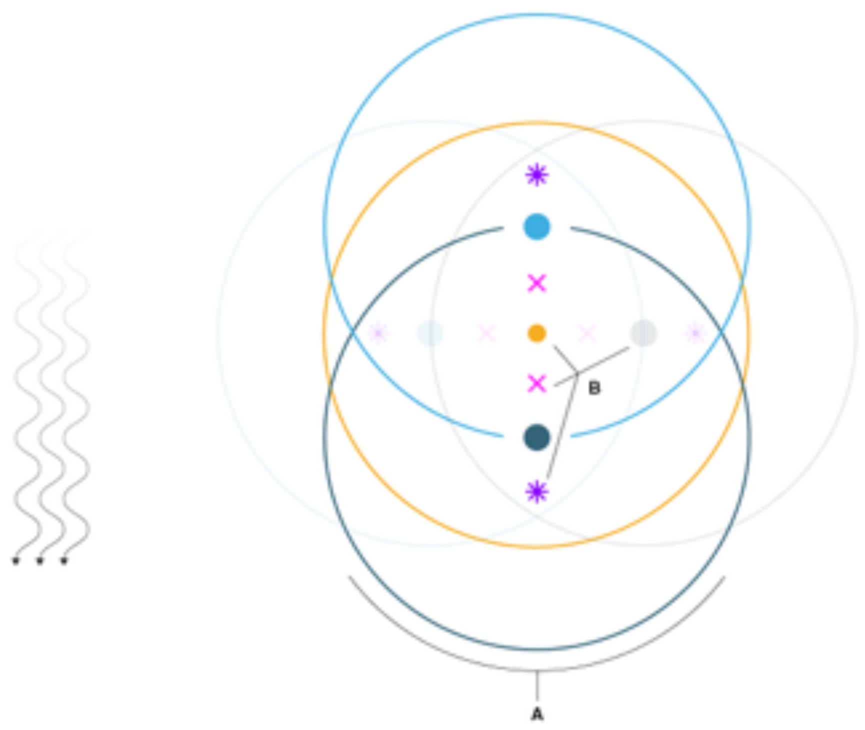
A | Optic Nerve
B | Pupil



Light Stimulus (Motion)

SEUNG DETECTOR (FRONT)

A | Receptive Field
B | Cell Bodies



Light Stimulus (Flipped)

RETINAL CIRCUITRY + MOTION DETECTOR

*Preferred Direction with respect to external light stimuli.

Starburst Amacrine Cells

- OFF SAC

Photoreceptors

- Rods + Cones

Bipolar Cells

- Bipolar Cell (Type BC2)
- Bipolar Cell (Type BC3a)

ON/OFF Direction Selective Ganglion Cell

- ooDSGC (Preferred Direction Up)
- ooDSGC (Preferred Direction Down)
- ooDSGC (Preferred Direction Left)
- ooDSGC (Preferred Direction Right)

Synaptic Connections

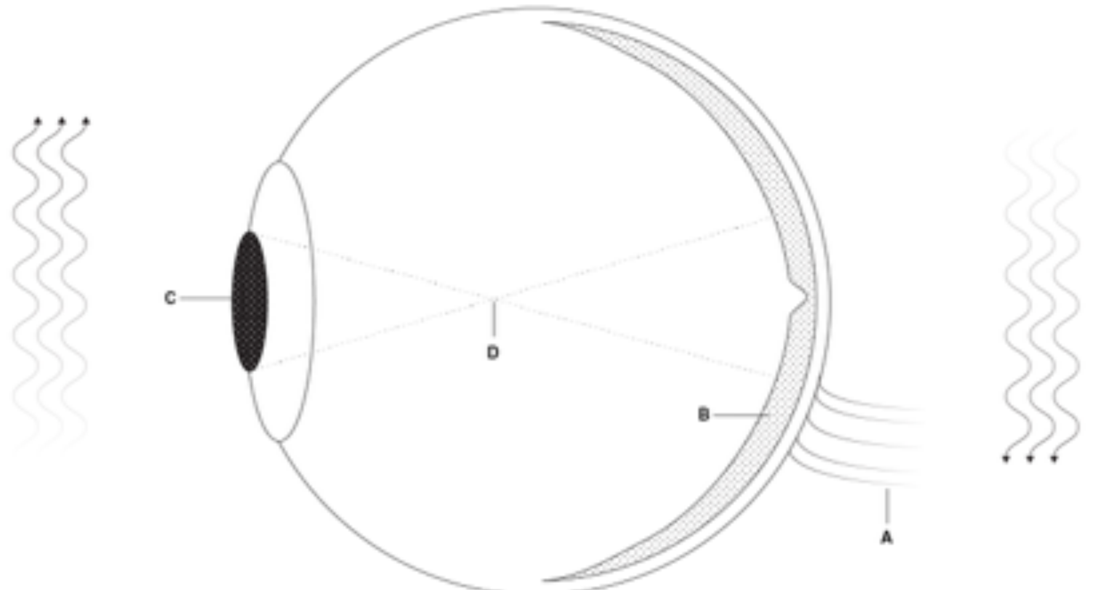
- Excitatory (Continue)
- Inhibitory (Stop)

Signal Flow

- Excitatory (Continue)

EYE

A | Optic Nerve
B | Retina
C | Pupil
D | Lens Induced Flip

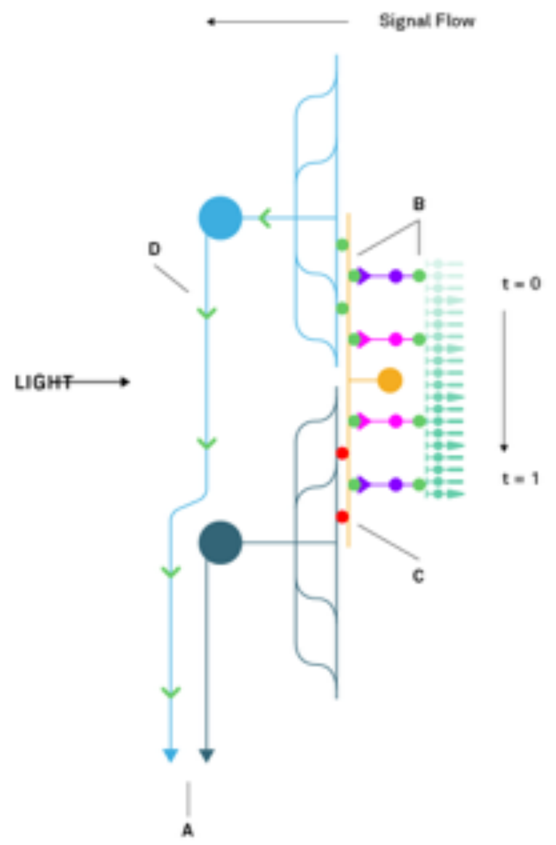


Light Stimulus (Motion)

Light Stimulus (Flipped)

SEUNG DETECTOR (SIDE)

A | Optic Nerve (to Brain)
B | Excitatory Synapse
C | Inhibitory Synapse
D | Cell Fires

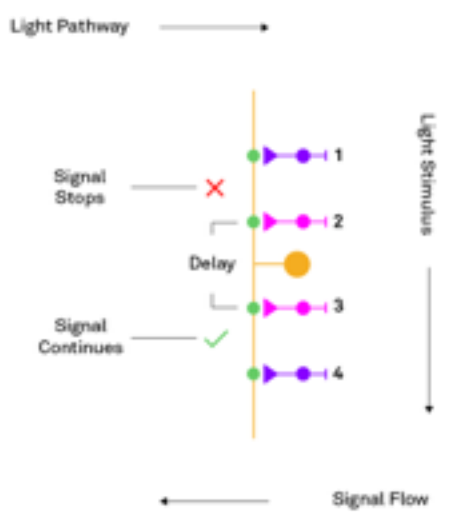


Light Stimulus (Flipped)

Direction Selective Circuit

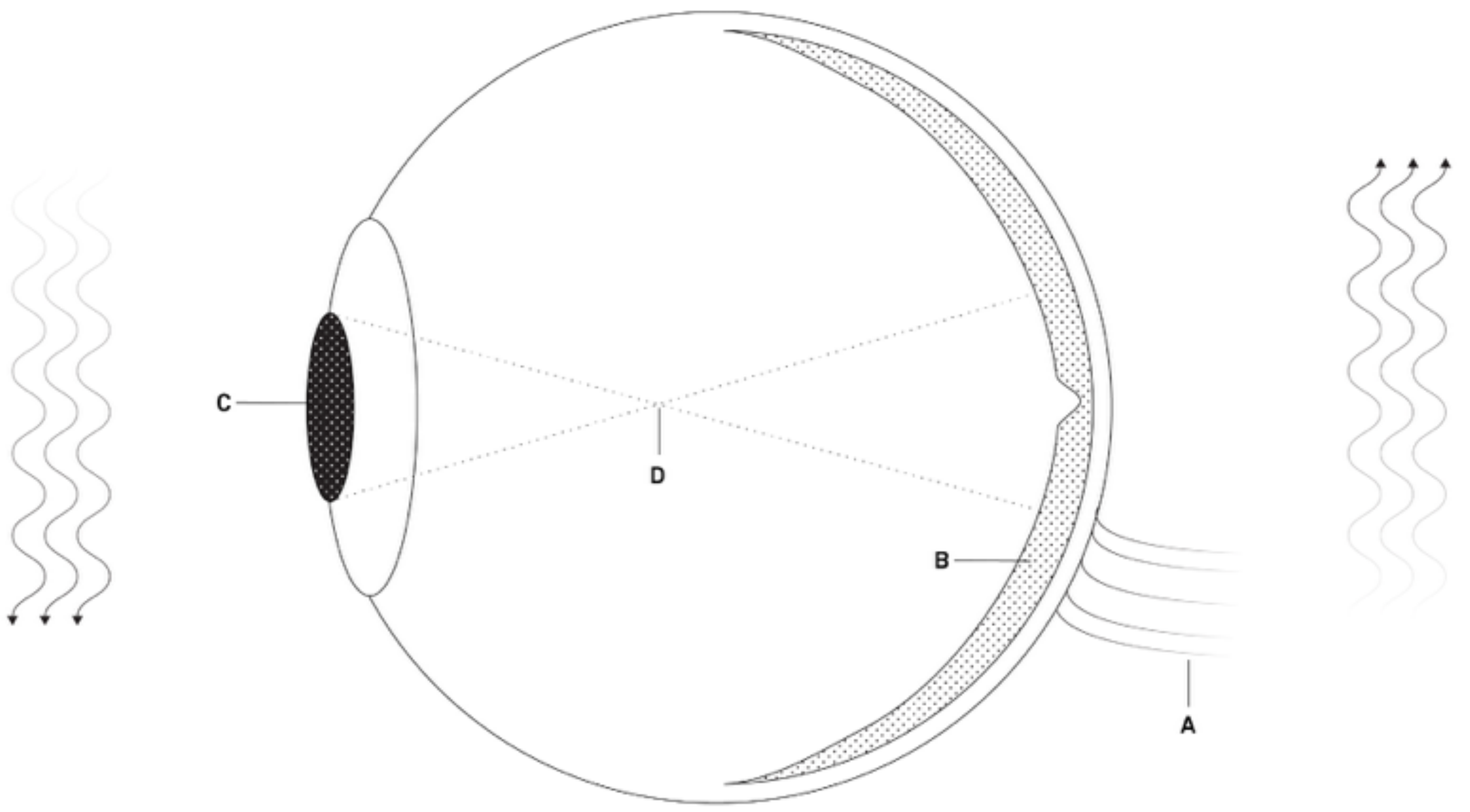
SAC dendrites are wired to pathways with time lags of visual response that differ by an amount (t) (BCs 2+3). This circuit amounts to an elegant motion detector.

Generally speaking, starburst amacrine cells (SACs) respond to a visual stimulus that moves from the soma (cell body) towards the distal dendrites (centrifugal or CF motion) but not to a visual stimulus moving in the opposite direction (centripetal or CP motion).



EYE

A | Optic Nerve
B | Retina
C | Pupil
D | Lens Induced Flip

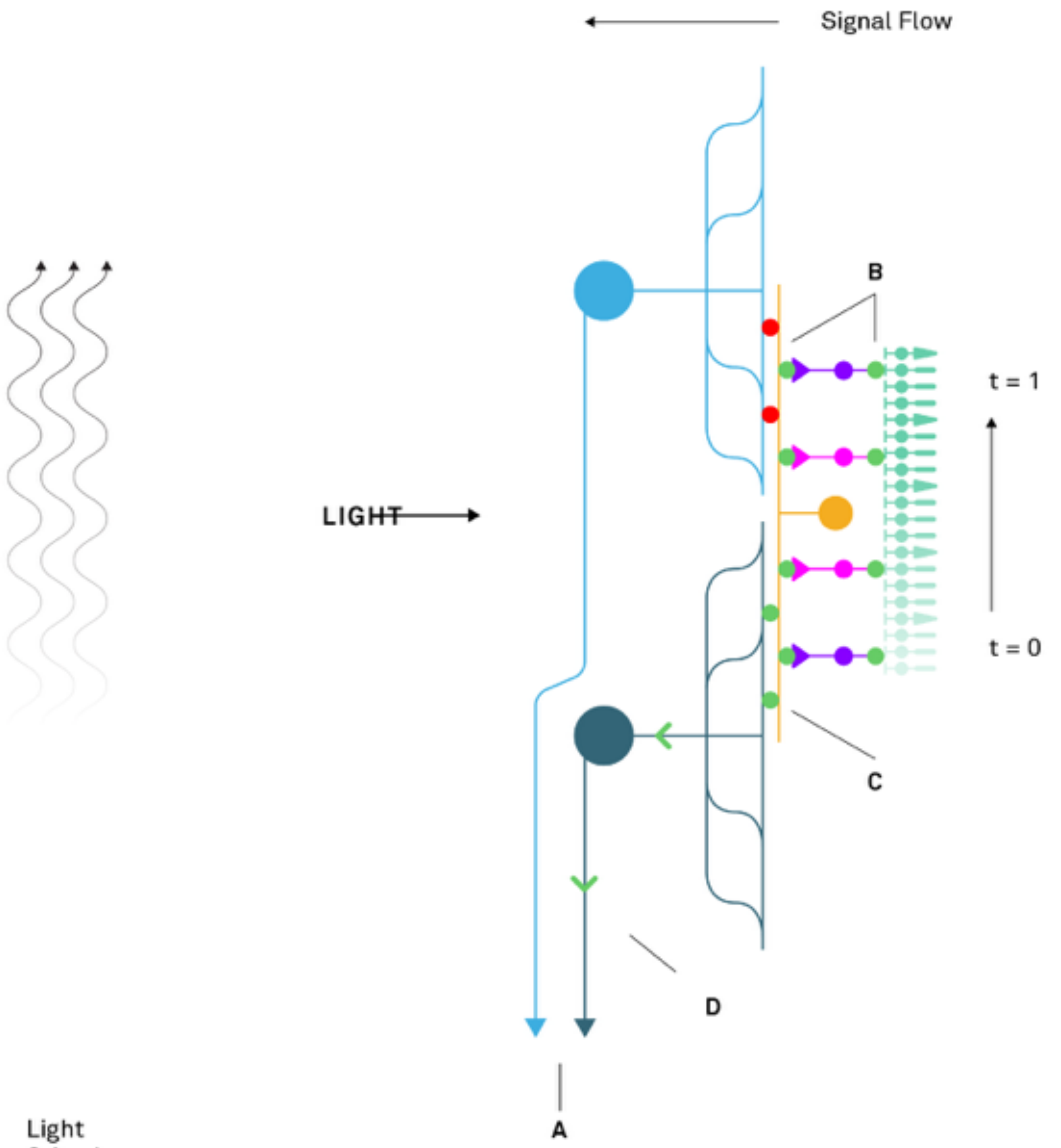


Light
Stimulus
(Motion)

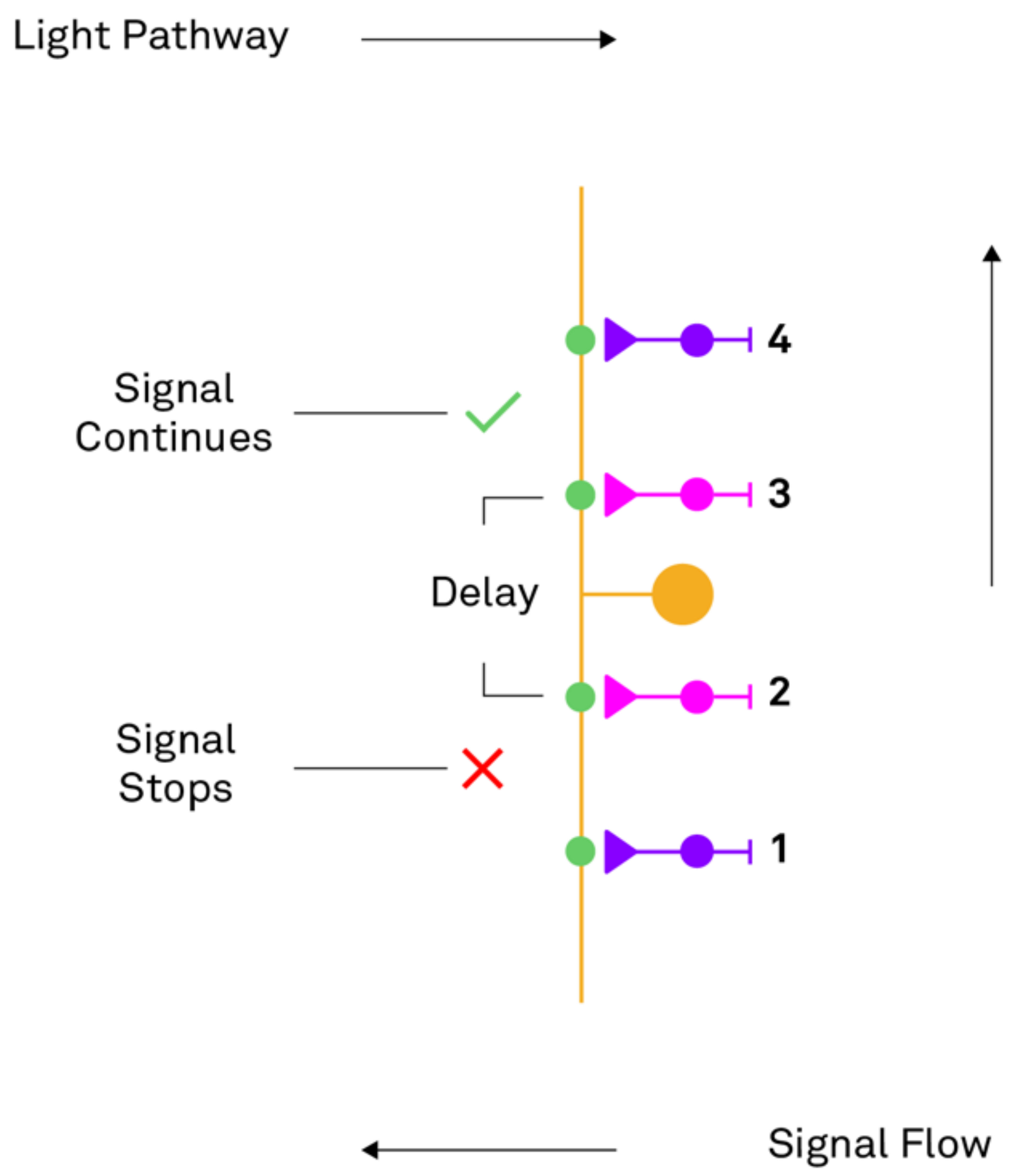
Light
Stimulus
(Flipped)

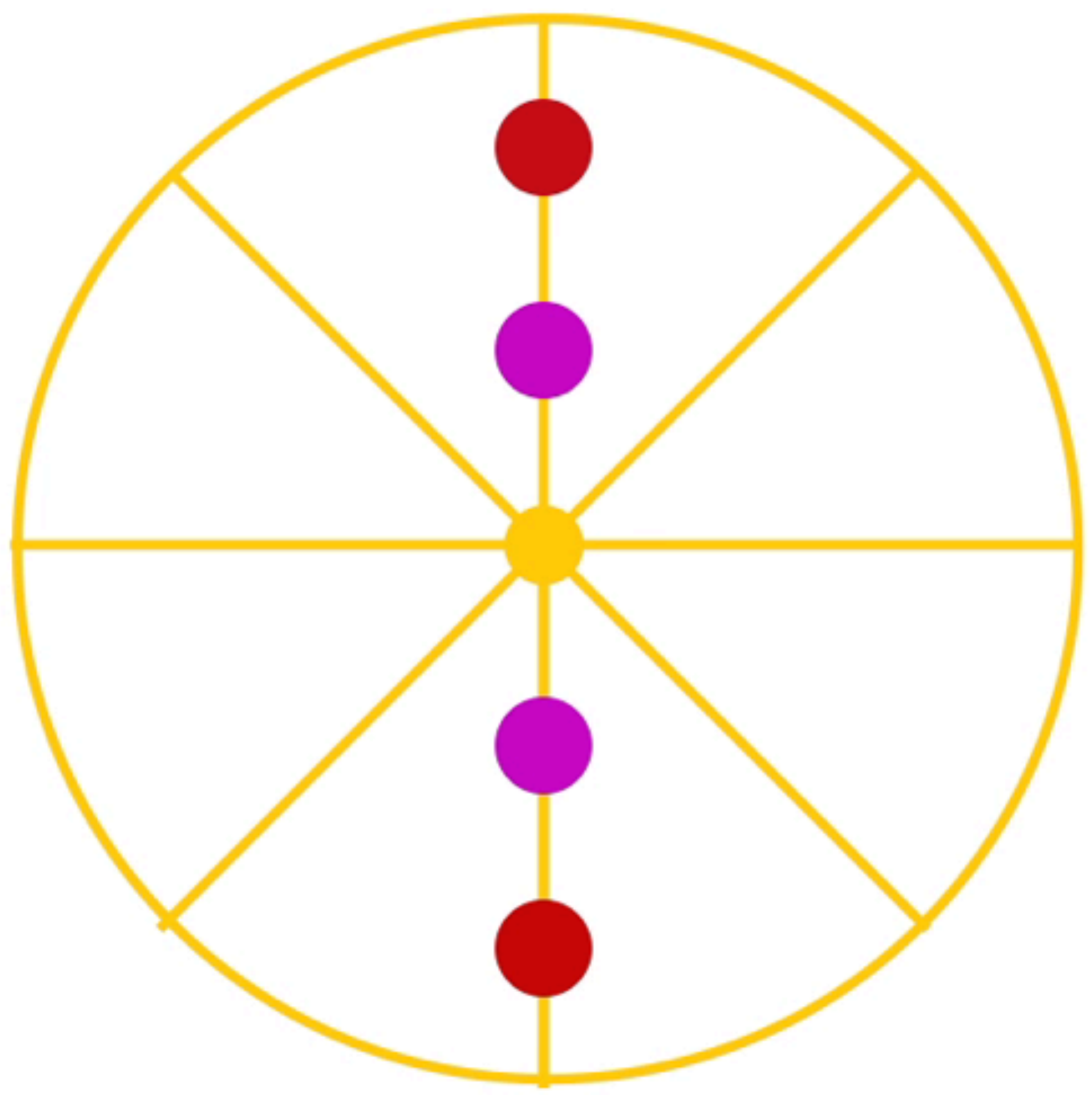
SEUNG DETECTOR (SIDE)

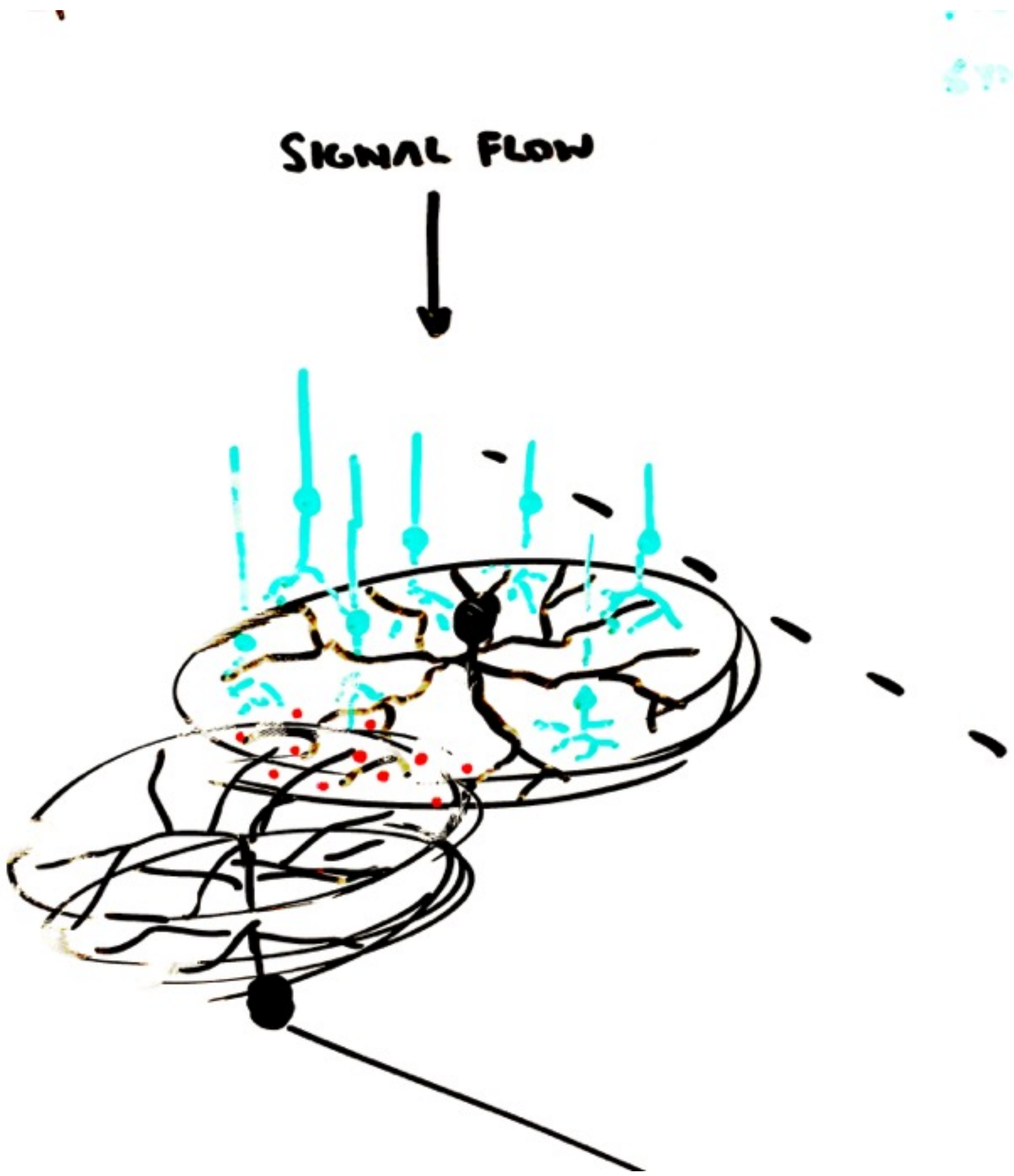
A | Optic Nerve (to Brain)
B | Excitatory Synapse
C | Inhibitory Synapse
D | Cell Fires

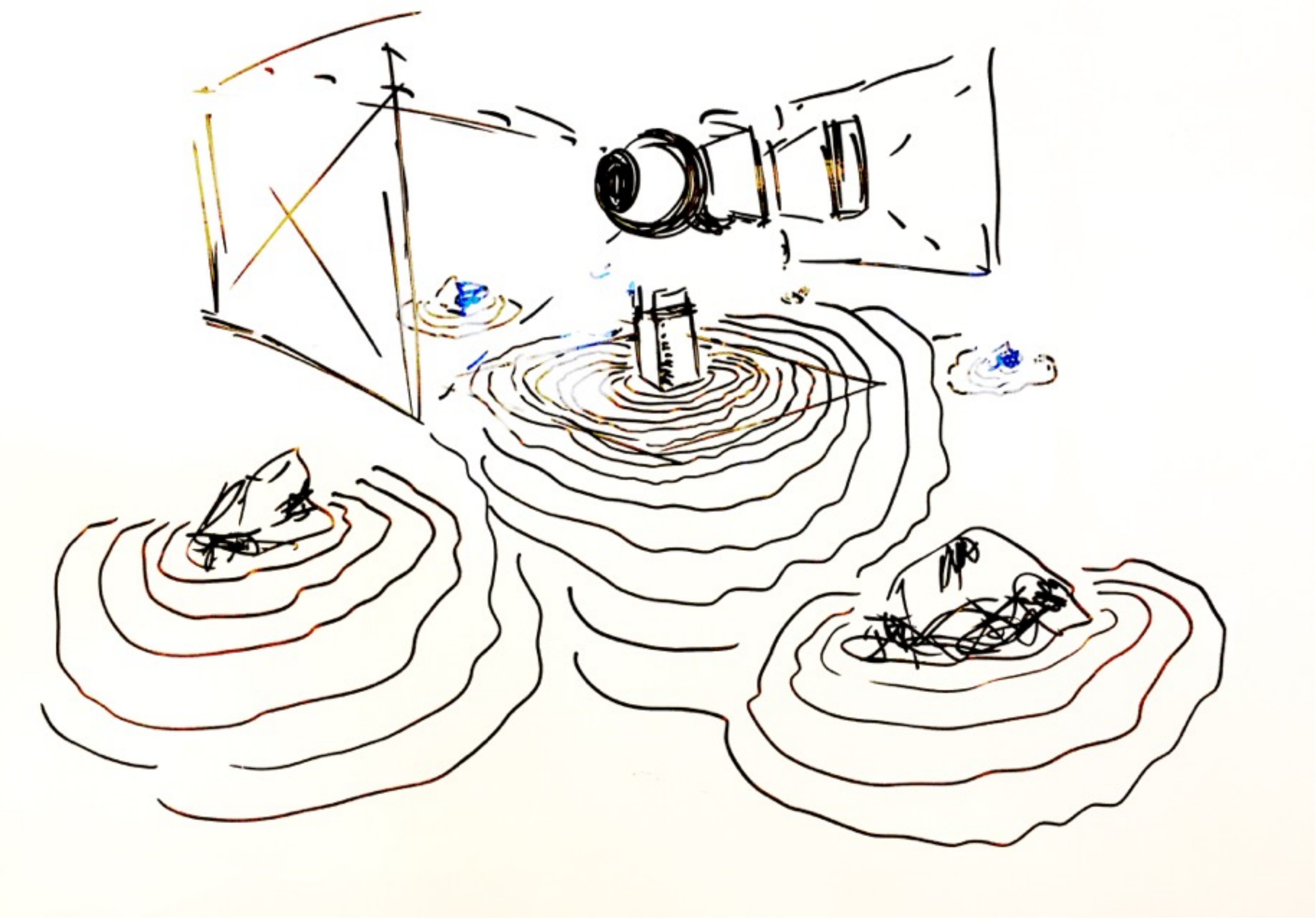


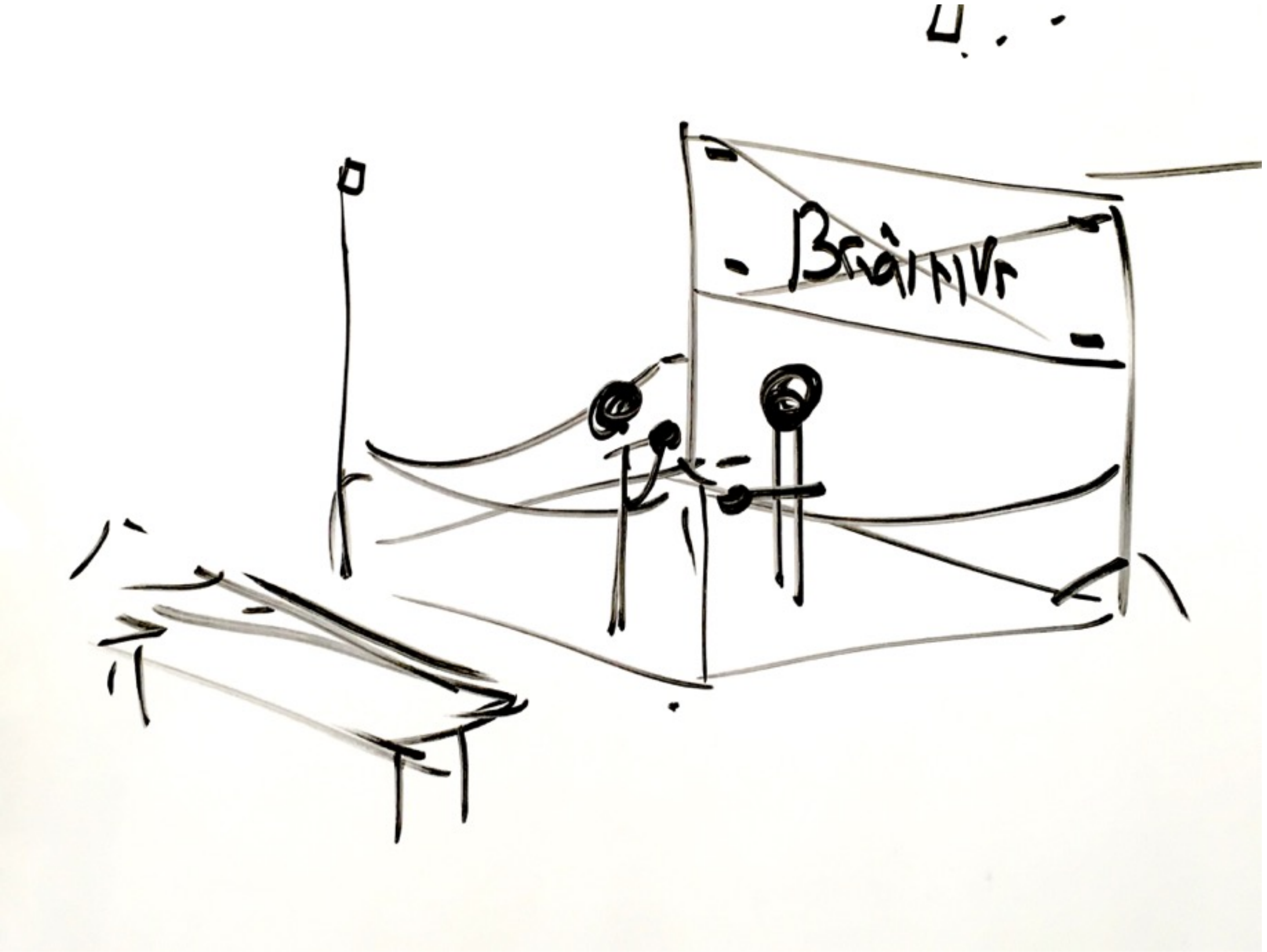
Light
Stimulus
(Flipped)

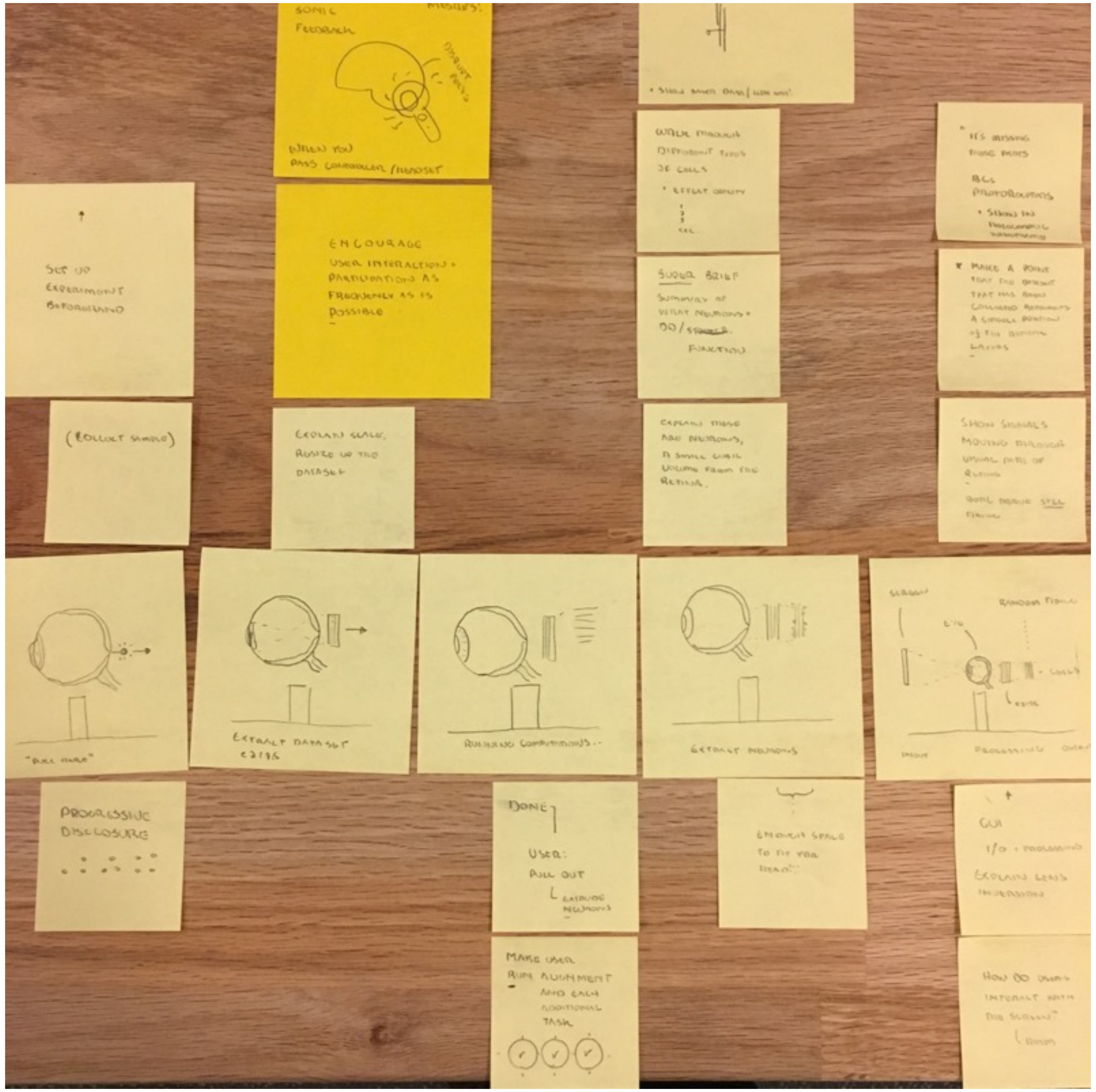

Light Pathway
Signal Continues
4
3
Delay
2
Signal Stops
1
Signal Flow



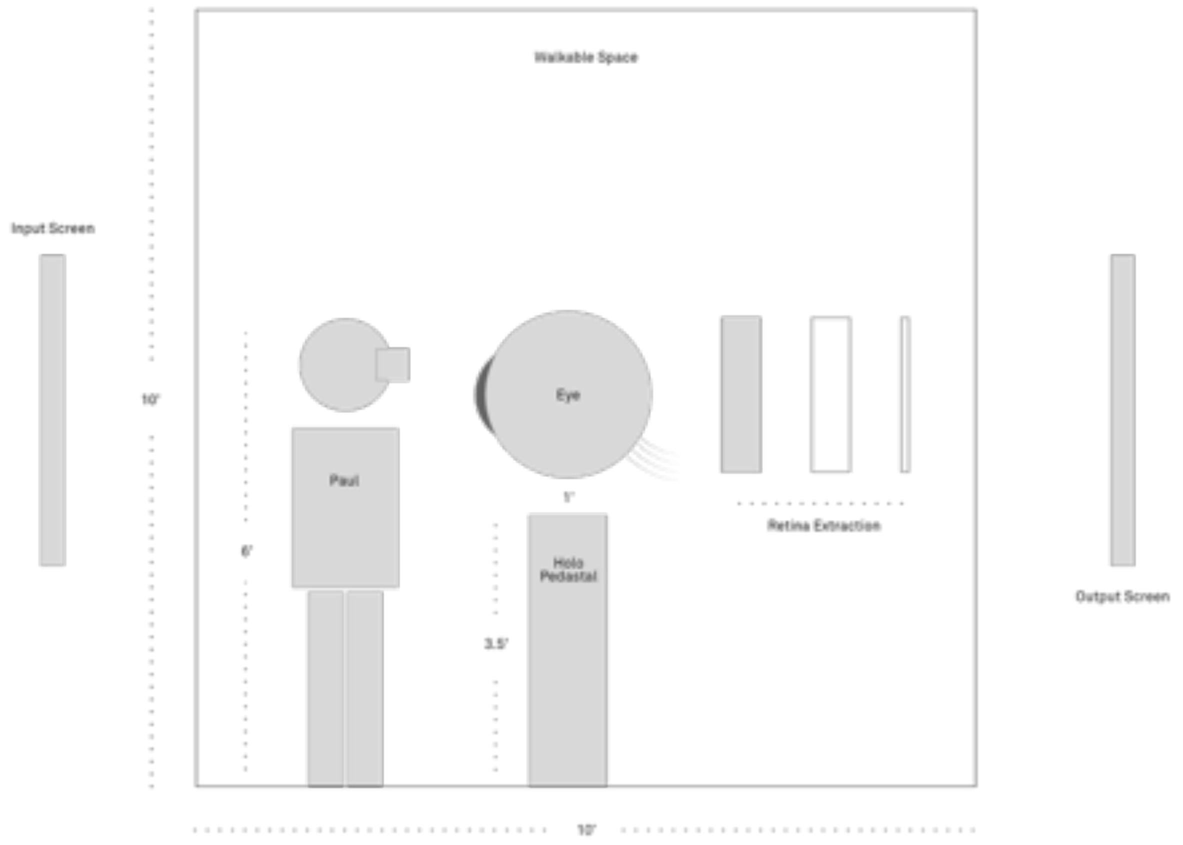

SIGNAL FLOW



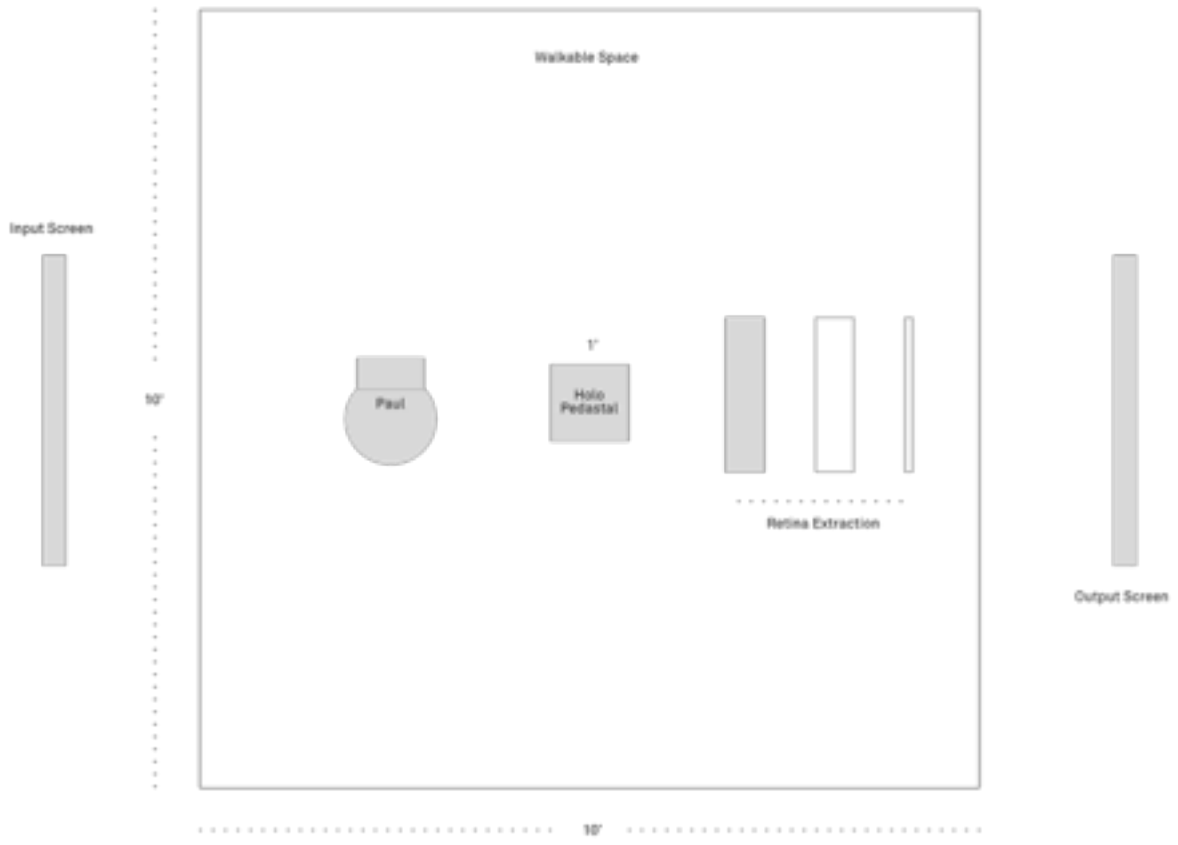

BrainVr


ENCOURAGE
USER INTERACTION + PARTICIPATION AS FREQUENTLY AS POSSIBLE
PROGRESSIVE DISCLOSURE
DONE
USER:
PULL OUT
MAKE USER
RUN ALIGNMENT
AND EACH ADDITIONAL TASK

SIDE VIEW



TOP VIEW



DETAIL

Overview

The VR Lab will be enclosded within a dimension of 10ft per side.
We'll use a holographic pedestal in the center of the space to give a bit of context to the brain + retina models. Instead of the 3D objects simply floating in a surreal space, we use the metaphor of a Virtual Reality laboratory and holographic pedestal to ground the space.

Dialogue

n/a

Top + Side | View

The space is symetrical, this affords equal usability in all directions.

Perhaps the next() moments are created by interacting directly with the holopedestal. Now we don't technically have to do this, but it gives us the promise of a fairly consistent location for our users to experience whatever we choose to put before them.

SCREENS

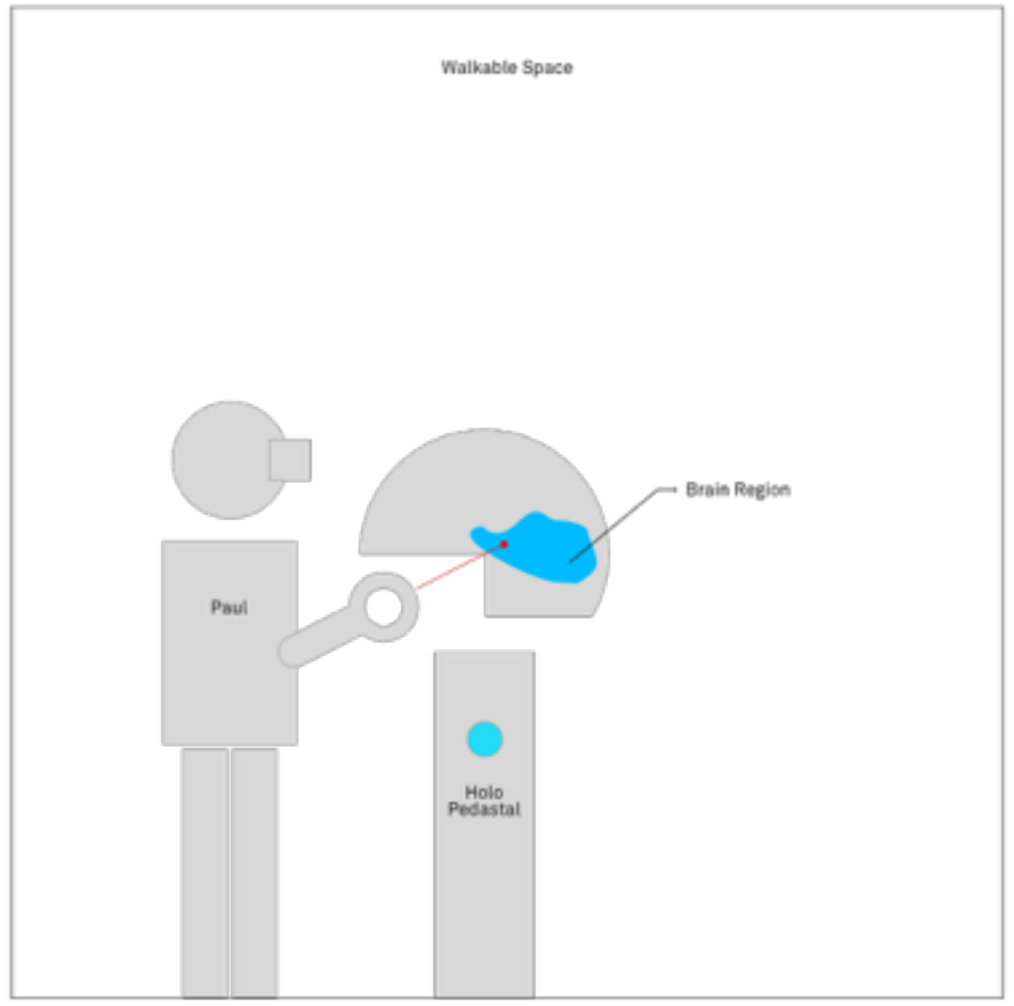
Input Screen

The input screen has two uses. First, it displays supporting images and animations (cinema). Second, it displays simple light impulses which are in turn perceived by the eye.

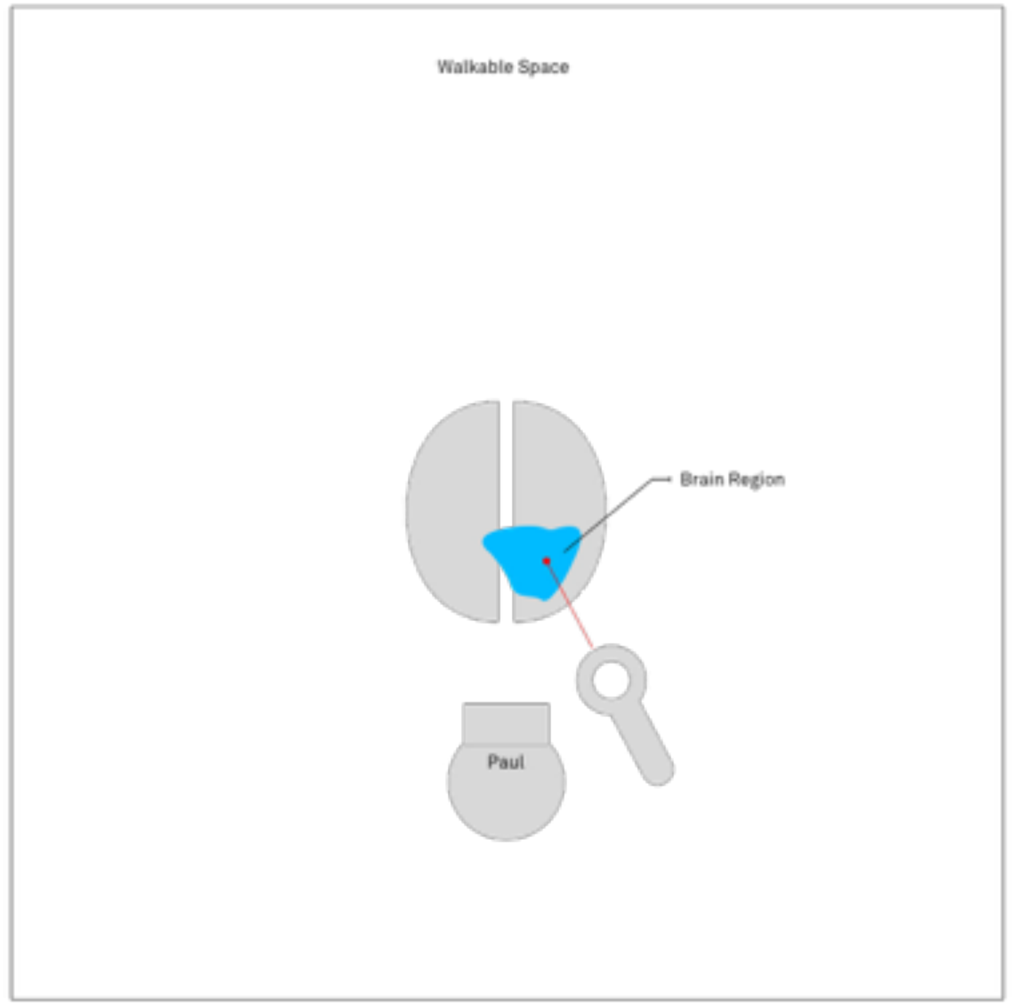
Output Screen

The output screen serves to inform the users on the internal mechanics of the neural circuit. It is a simplified representation of direction selectivity, showing circuit outputs as cardinal directions.

SIDE VIEW



TOP VIEW



DETAIL

Overview

Brain highlights region if controller points toward it. Controller emits a sort of laser beam to help with orientation.

Dialogue

Your brain has many specialized regions. Should we enurmerate the main ones?

Top + Side | View

Persistent floating GUI projecting off the brain showing. When user begins to point at the brain, all other HUD GUIs fade away.

SCREENS

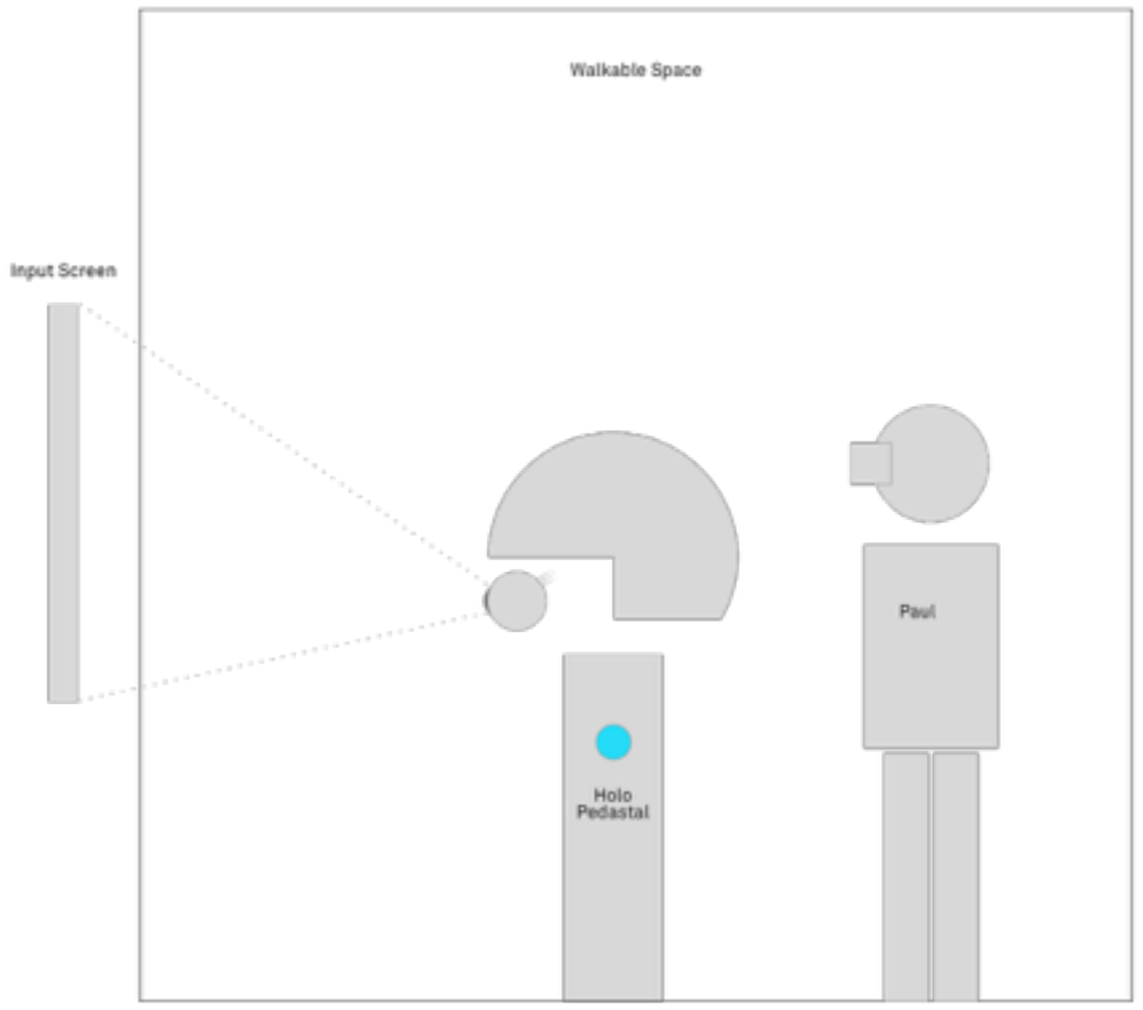
Input Screen

n/a

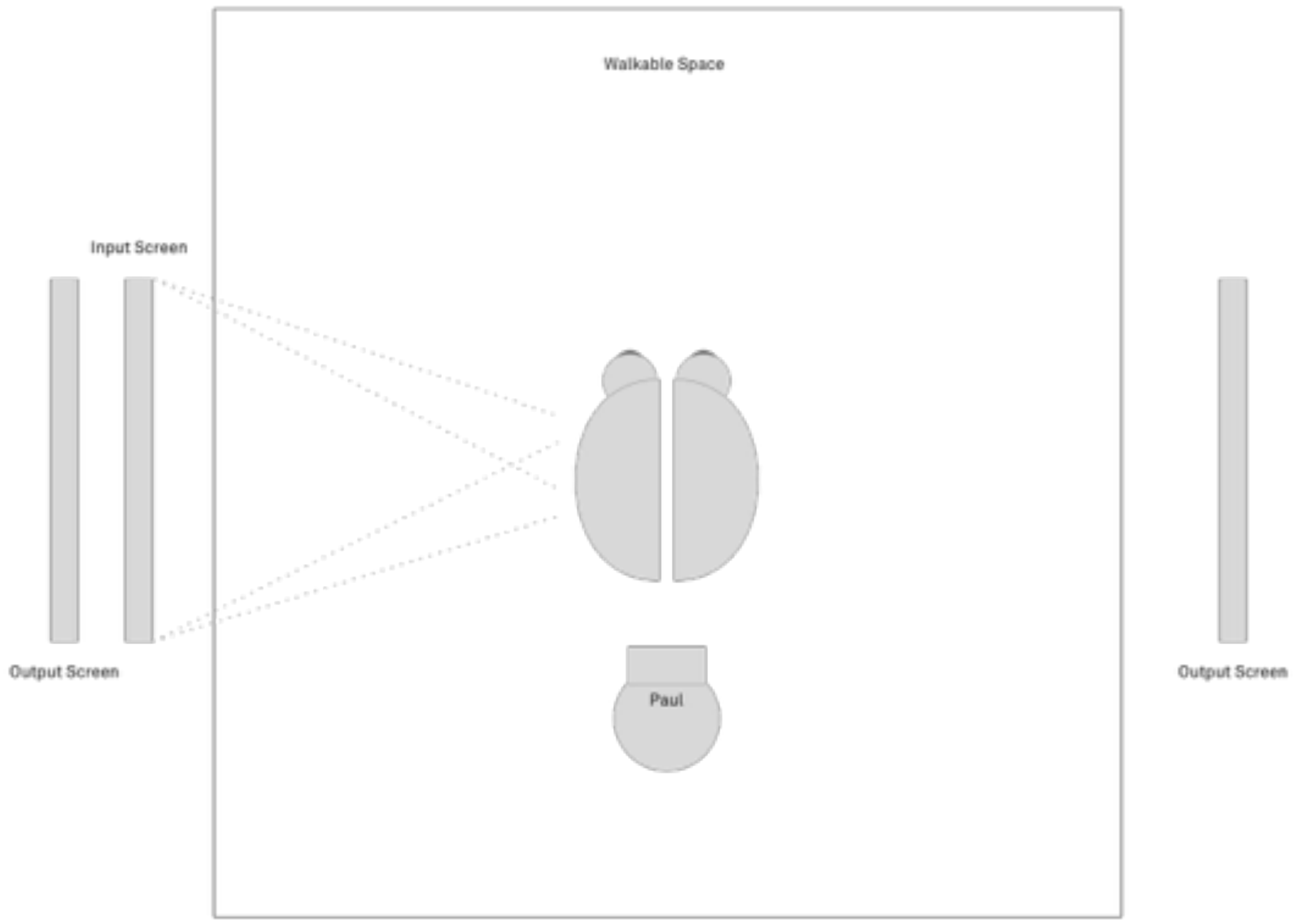
Output Screen

n/a

SIDE VIEW



TOP VIEW



DETAIL

Overview

The Muybridge gallop is a classic example of all three of our cinematic factors. The experiment was performed under the pretense of a bet, do a horses completely hooves leave the ground when it is at full gallop?

The experiment consists of an array of some 20 still cameras arranged to capture a galloping horse in rapid sucession.

If the resulting images are

Dialogue

Begin framing "vision" and the cinematic experience.

Cinema:
Light, Movement, Time.

Top + Side | View

The user is drawn to the newly apparent input screen where following the countdown, now plays a looping animation of the Muybridge gallop.

Users are able to control the animation speed, pause and play at will.

The relative speed of animation will elicit a differential response in the brain + eye combination.

SCREENS

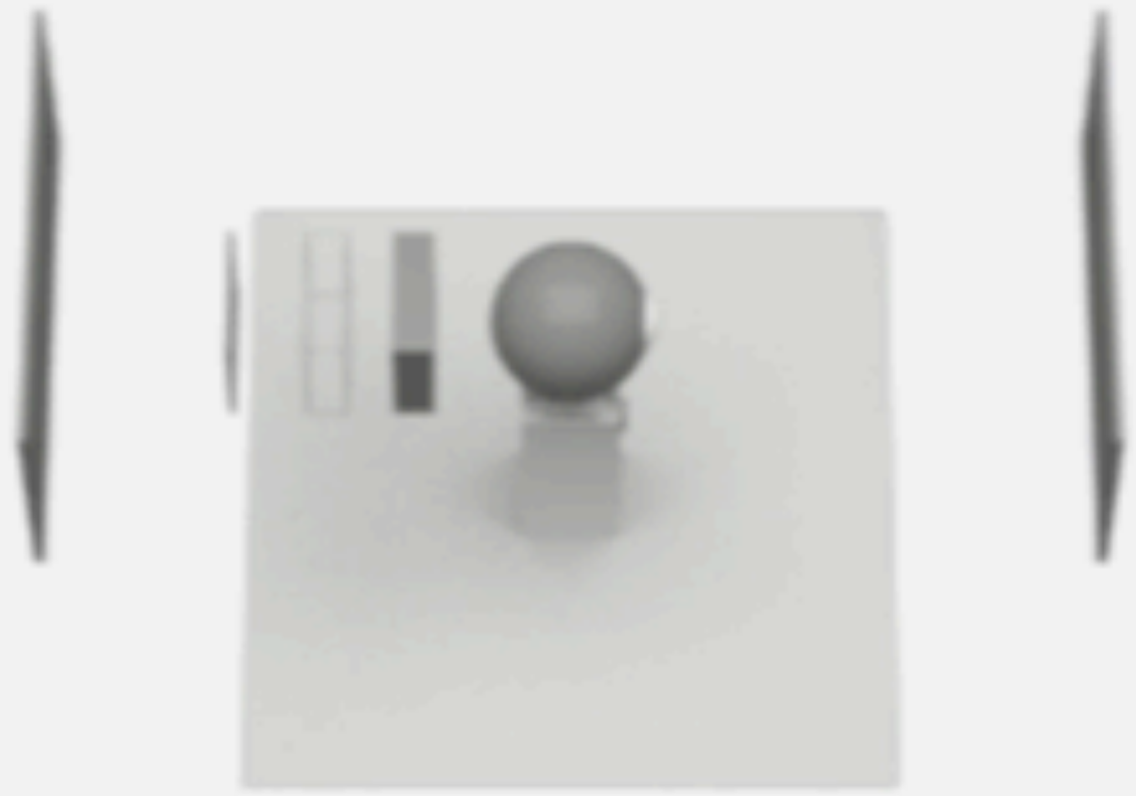
Input Screen

Screen now switches to show one of thte most canonical examples of animation + cinema. The Muybridge Gallop.

The user is able to control the play stat of the images.

Output Screen

n/a



Painting from a new perspective

Tilt Brush lets you paint in 3D space with virtual reality.

Your room is your canvas. Your palette is your imagination. The possibilities are endless.

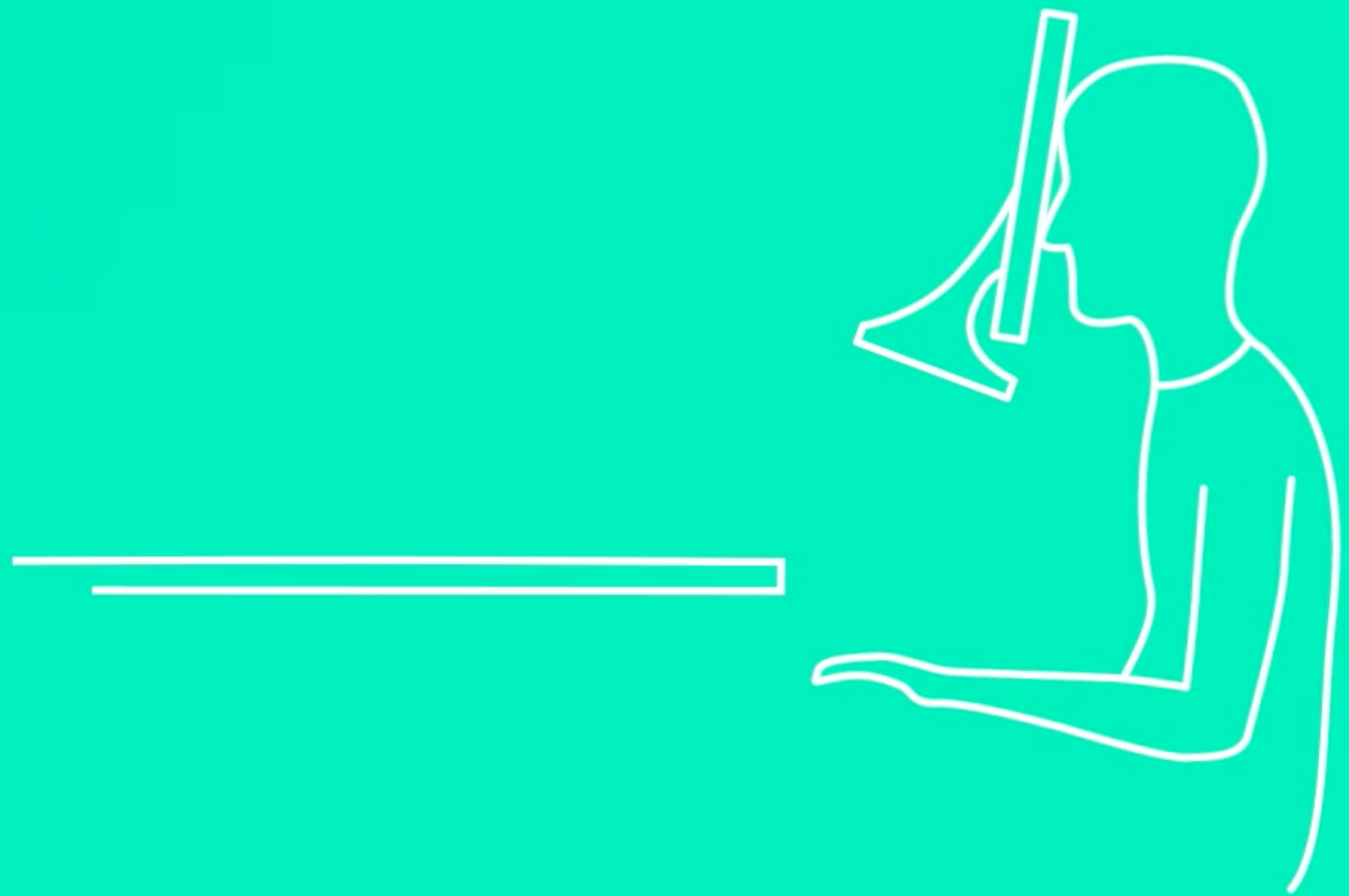

Tilt Brush
RECENT NEWS
Tilt Brush


BRainV

DESIGNING

Mike
Alger

Interaction Design Patterns



HMD

Head Mounted Display

drag and drop 

blue hyperlink 

clickwheel 

pinch to zoom 

INTRODUCING

HMDs

Oculus Rift

HTC Vive

Samsung Gear VR

Project Morpheus

Google Cardboard

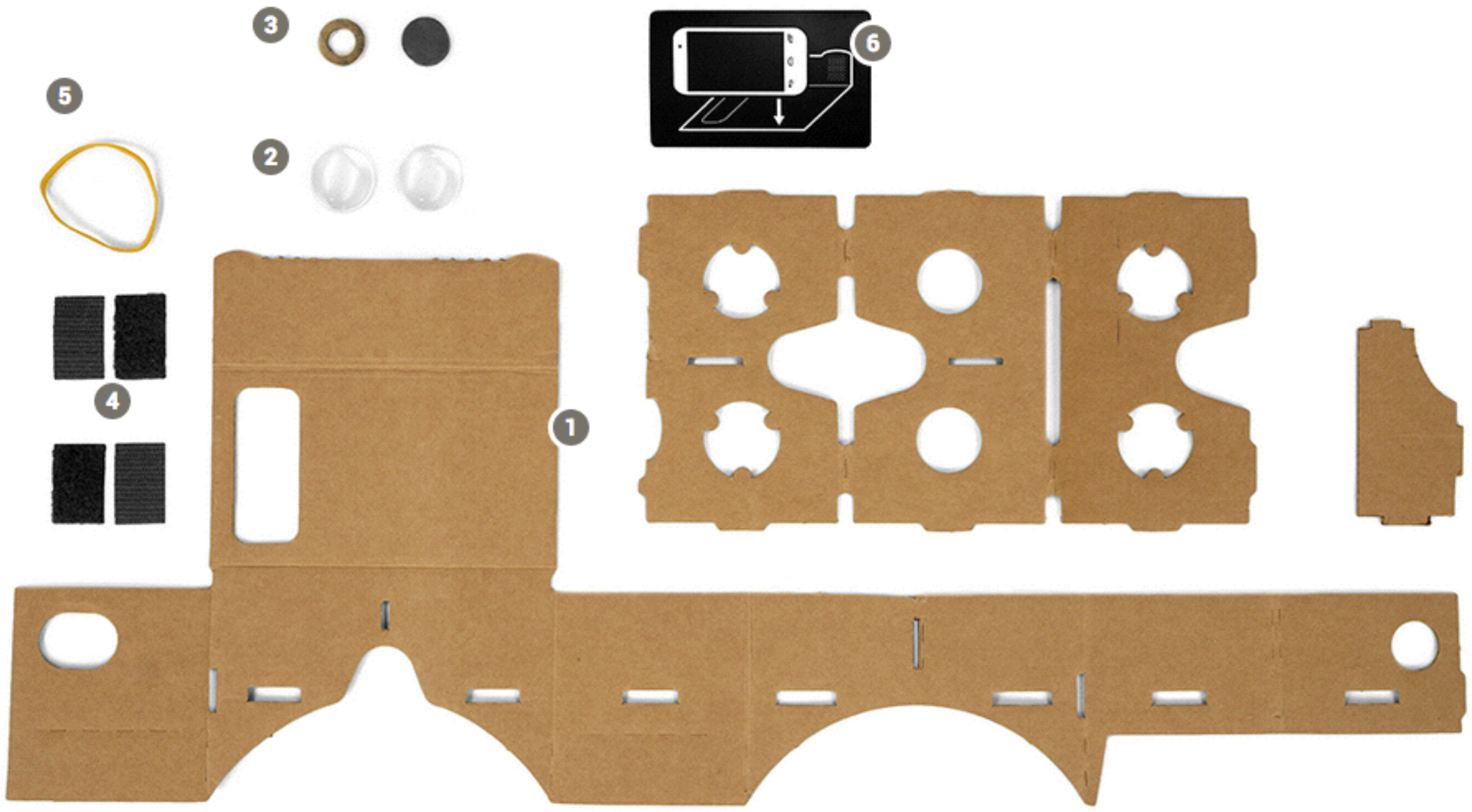
Virtual Boy

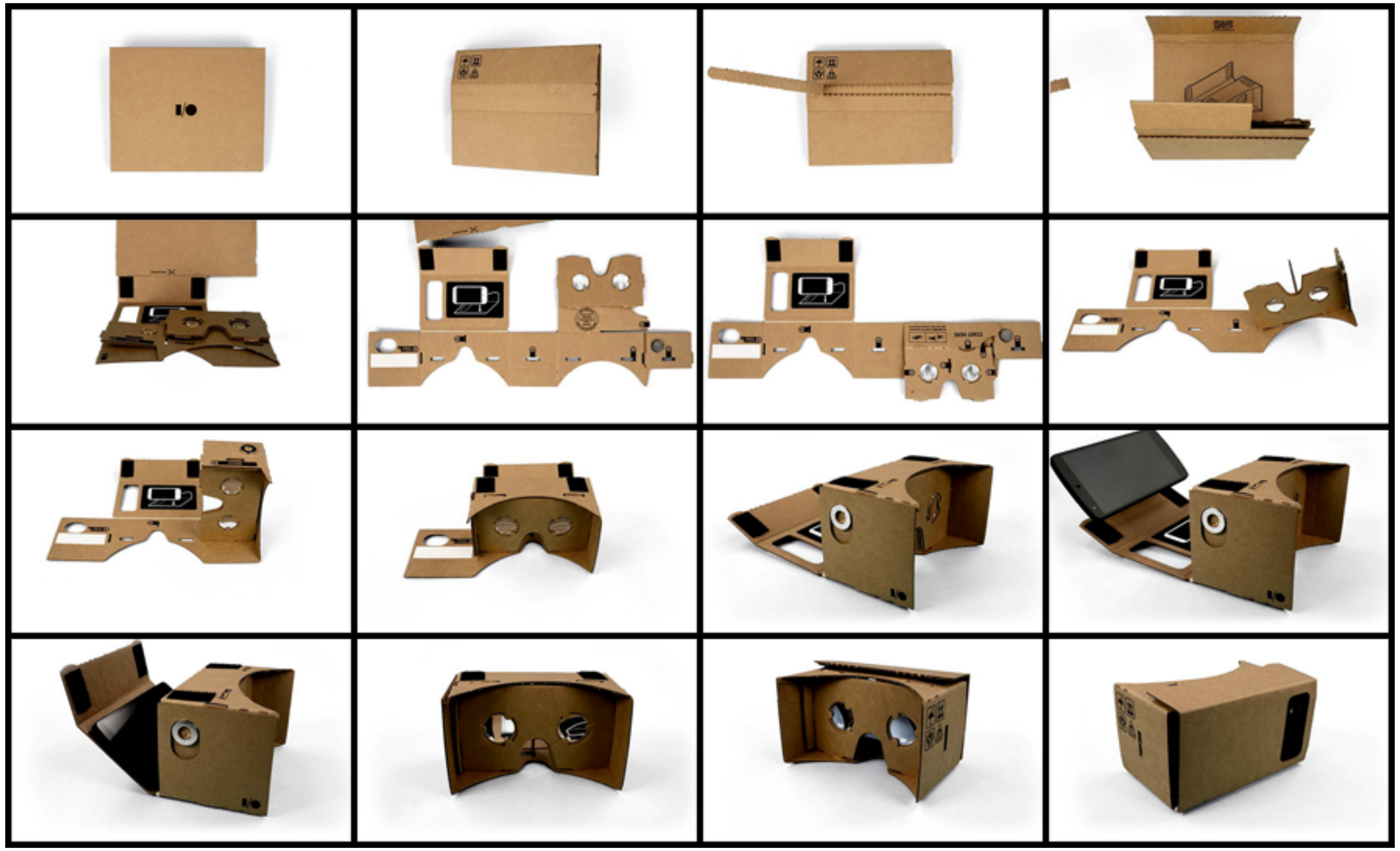
Famicon 3D System

VictorMaxx StuntMaster


htc

Here's what you need to get started:





INTRODUCING

Development Environments

unity

UNREAL
ENGINE

INTRODUCING

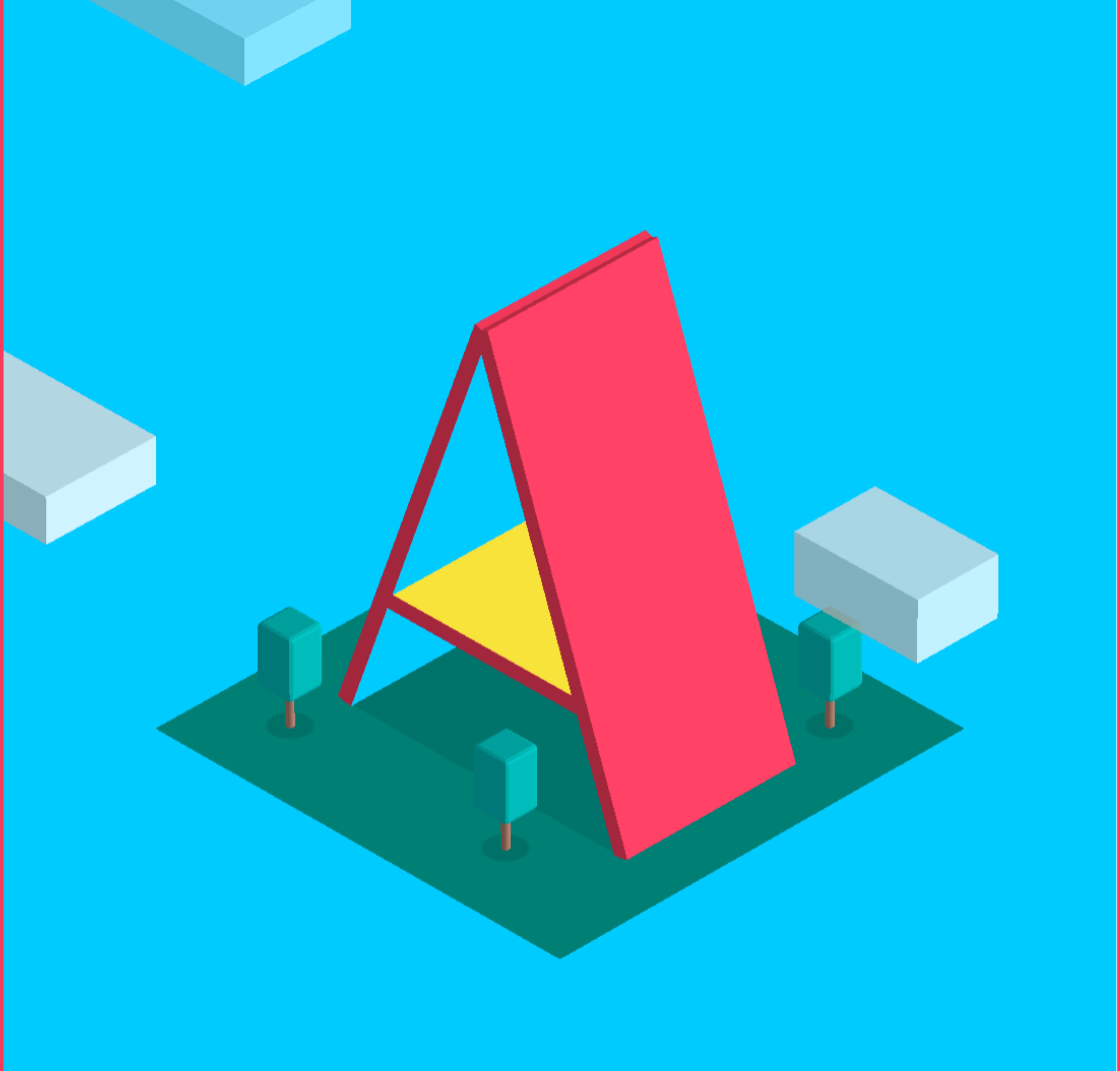
webVR

A-FRAME

Examples

- Hello World
- Anime UI
- Composite
- 360° Video
- Curved Mockups
- Spheres & Fog
- Shopping
- Warp
- Logo
- Unfold
- Panorama

← →



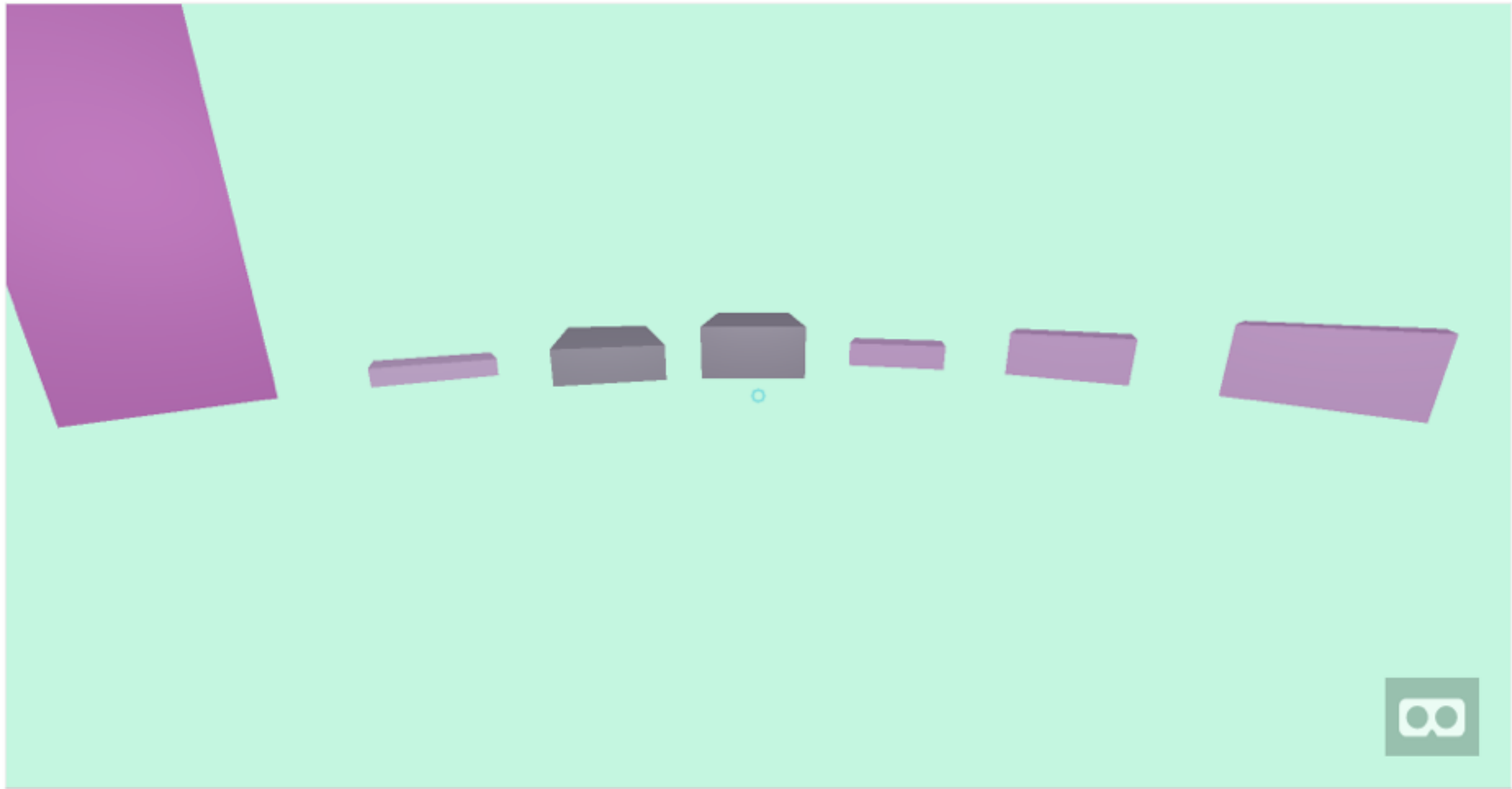
DOCS FAQ BLOG COMMUNITY GITHUB

Building blocks for the virtual reality web

Use markup to create VR experiences that work across desktop, iOS, Android, the Oculus Rift, and the HTC Vive.

GET STARTED

aframe + d3 test



Testing out aframe.io with d3.js.

Open

Since AFrame works with DOM elements, you can use d3's selection API to generate and modify 3D elements, as well as handle "mouse" events (including the VR friendly "fuse cursor") as if it was SVG.

Built with blockbuilder.org

index.html

```
<!DOCTYPE html>
<head>
  <meta charset="utf-8">
  <script src="https://cdnjs.cloudflare.com/ajax/libs/d3/3.5.5/d3.min.js"></script>
  <script src="https://aframe.io/releases/latest/aframe.min.js"></script>
  <style>
    body { margin:0;position:fixed;top:0;right:0;bottom:0;left:0; background-color: white;}

  </style>
</head>

<body>
```

Thanks!