Algorithmic Art

Nina Lutz, MIT

Prepared for Clubes de Ciencias ASU

Welcome!

Please go to yellkey.com/occur to fill out a pre-survey :)

Introductions

So who am I?

Nina Lutz

BSc. Computer Science and Engineering with Design (Architectural) MIT

Grad student at the MIT Media Lab

Numerical Simulation
Urban Planning
Algorithms
Optics
Colloidal Scattering in Architectural
Lighting
Architecture
Data Visualization
Projection Mapping
Computer Vision and Geometry

Math stuff

Glitter science

Algorithms for cities

Algorithms for story telling

Making computers make things better

And more.

People
Architectural Space
Math
Computers
Art

Make physical reality more digital.

Total Agents Rendered: 2000

rateScaler: 34

Spanish Speaking Amenity
French Speaking Amenity
Hotels
Restaurants
Attractions

11
5
1
0
6
7
St. Julia
Ordino
Massana

Andorra La Vella cirq, Hour: 23:00 – 0:00

Tourists | French Spanish Other

El Pas de la Casa
Canillo
Encamp

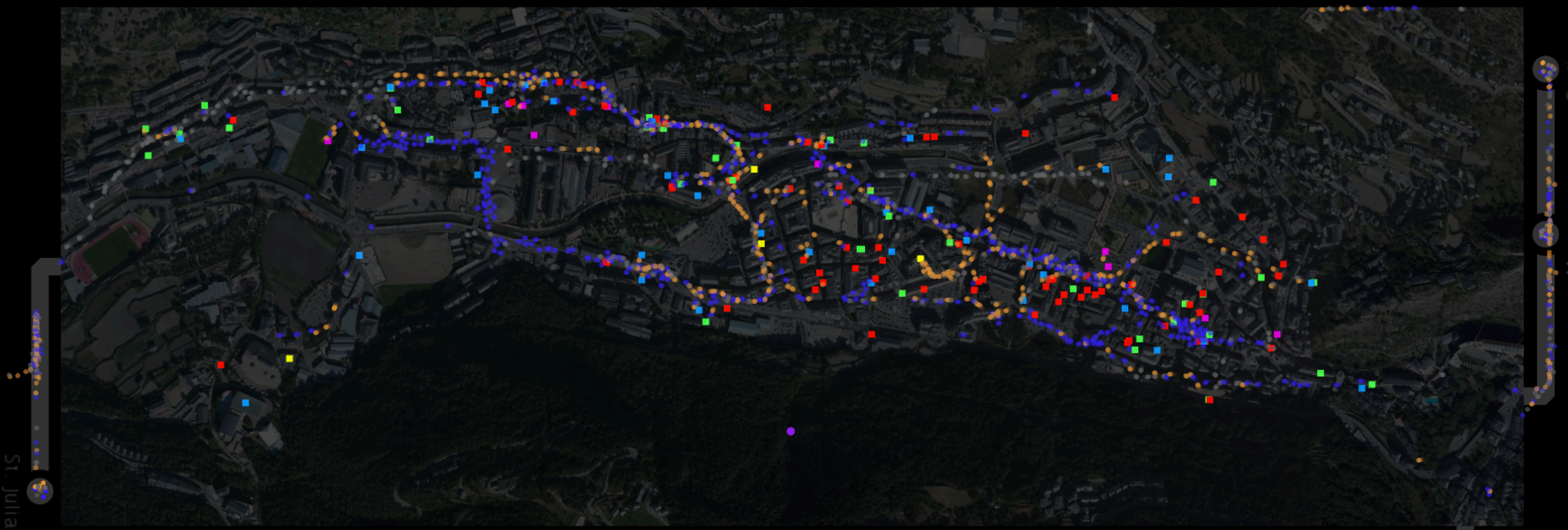
23:00 – 24:00
2015.07.23

Hr 0 3 6 9 12 15 18 21 0 3 6 9 12 15 18 21 0 3 6 9 12 15 18 21 0 3 6 9 12 15 18 21 0 3 6 9 12 15 18 21

Total Agents Rendered: 1978

rateScaler: 41

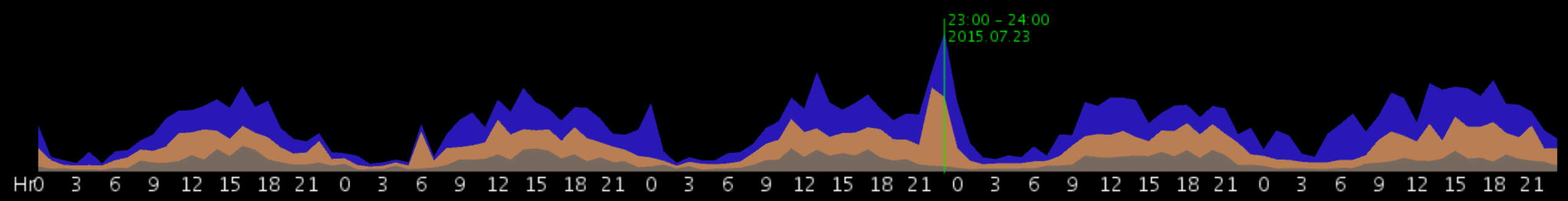
Spanish Speaking Amenity · Hotels
French Speaking Amenity · Restaurants
Attractions

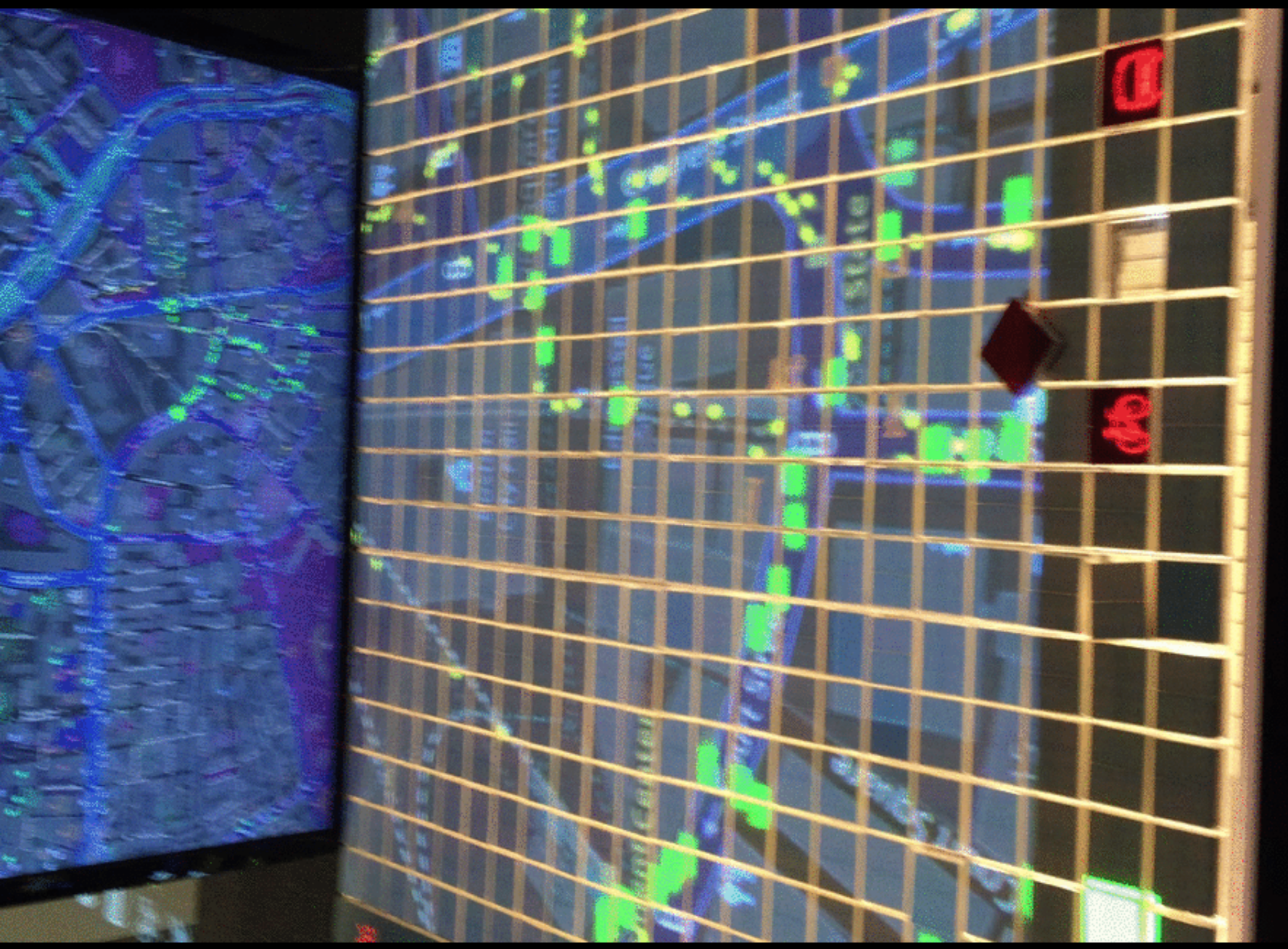


Andorra La Vella cirq, Hour: 23:00 – 0:00

Tourists | French Spanish Other

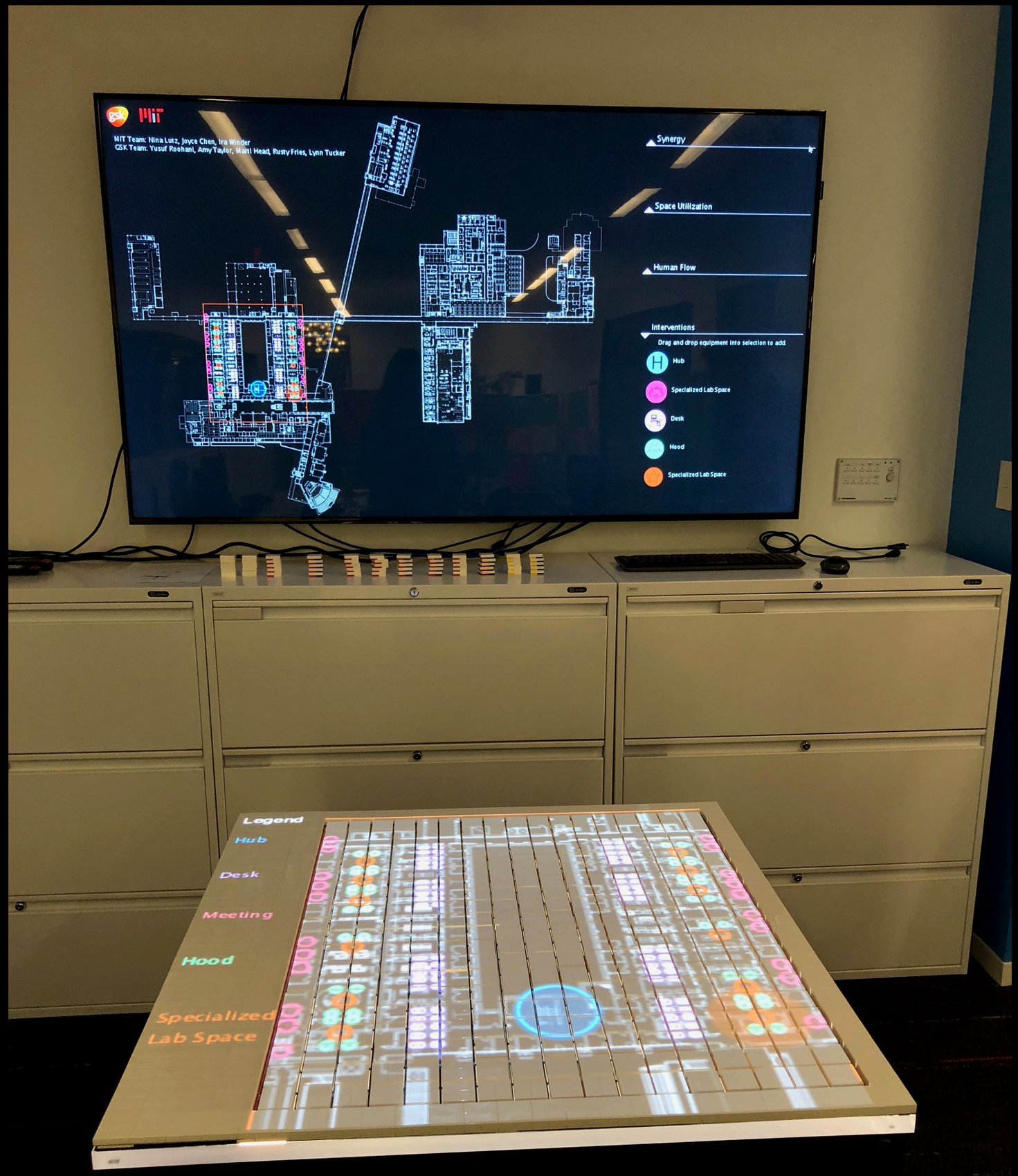
El Pas de la Casa · Canillo · Encamp

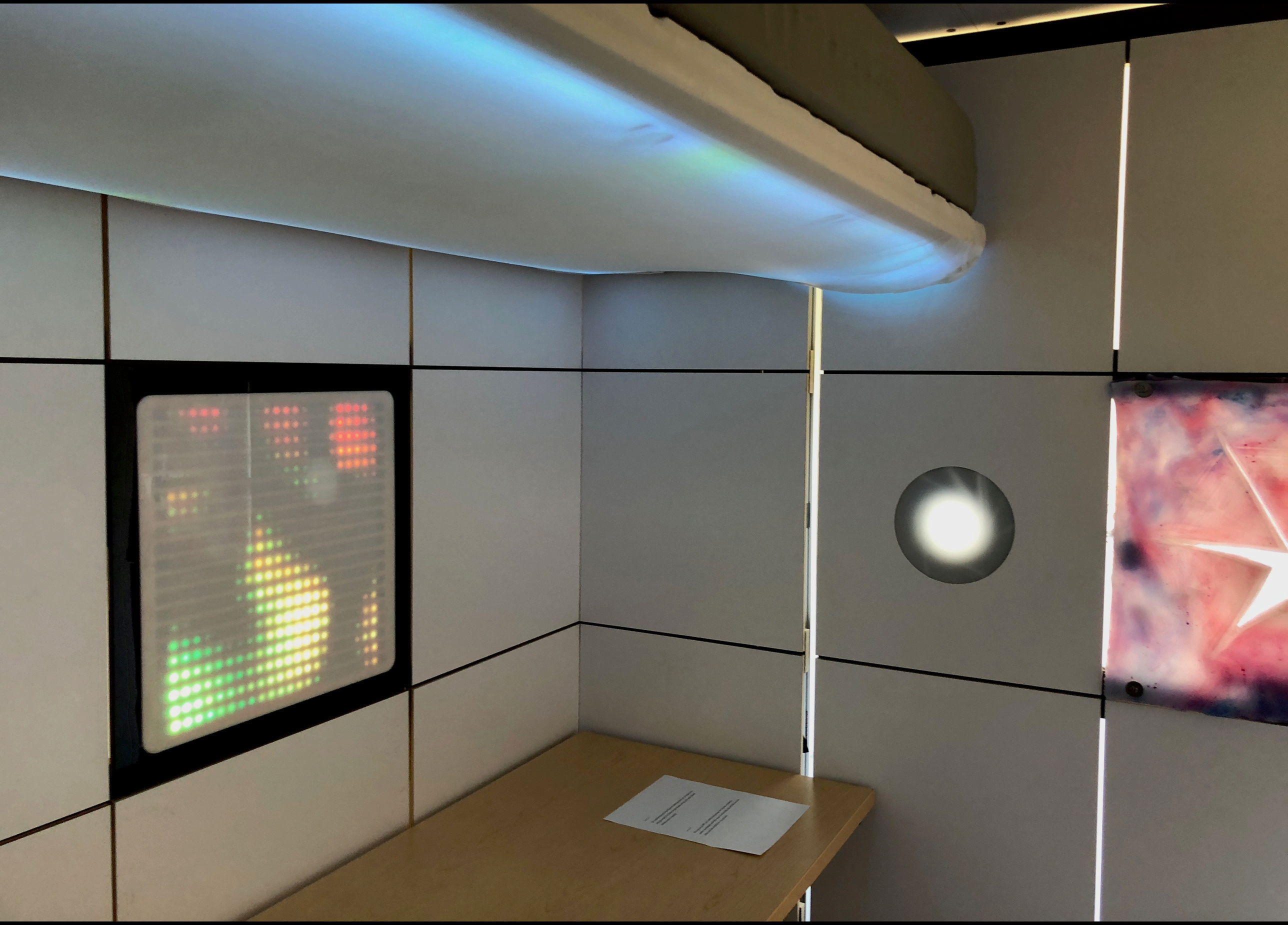


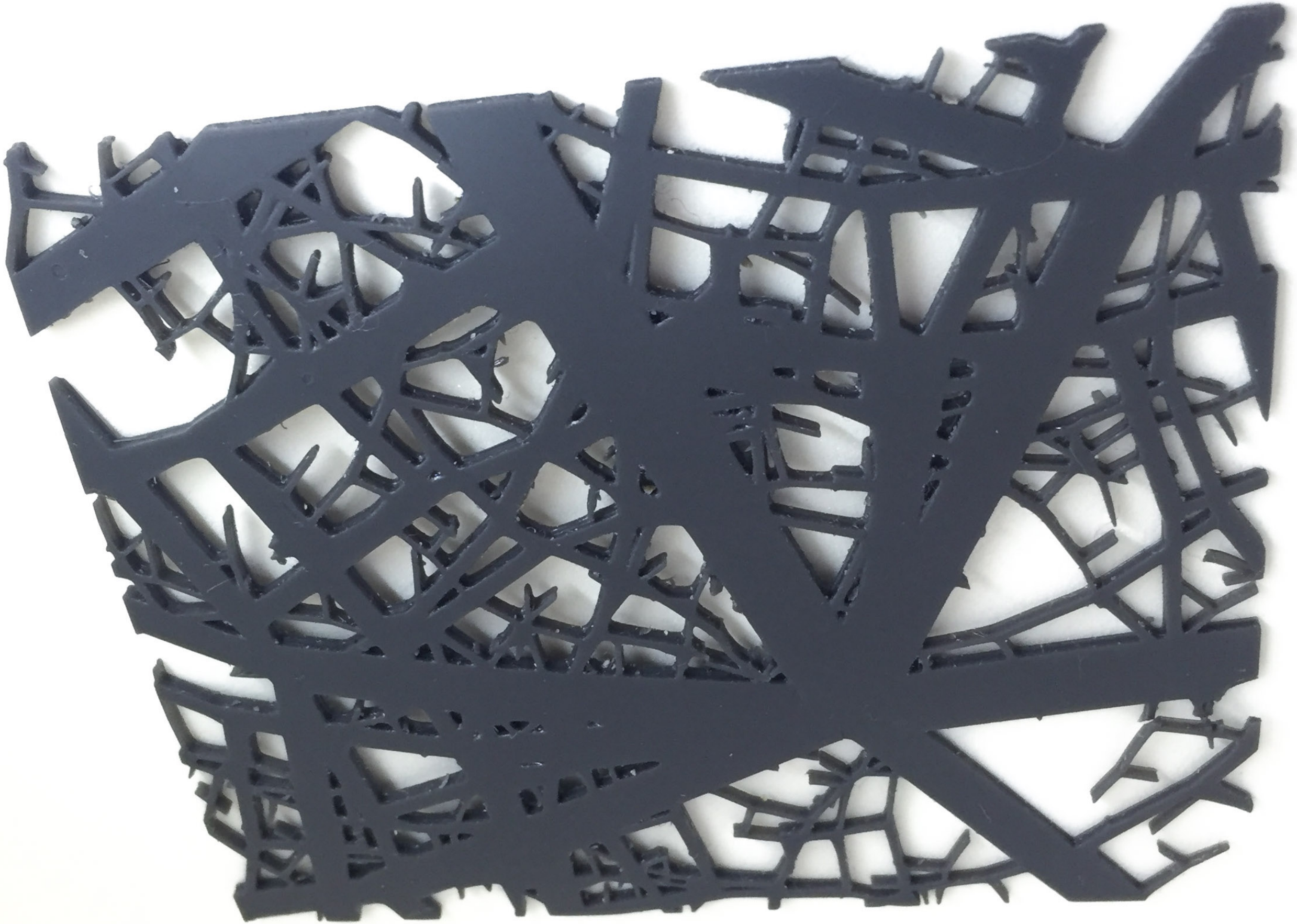


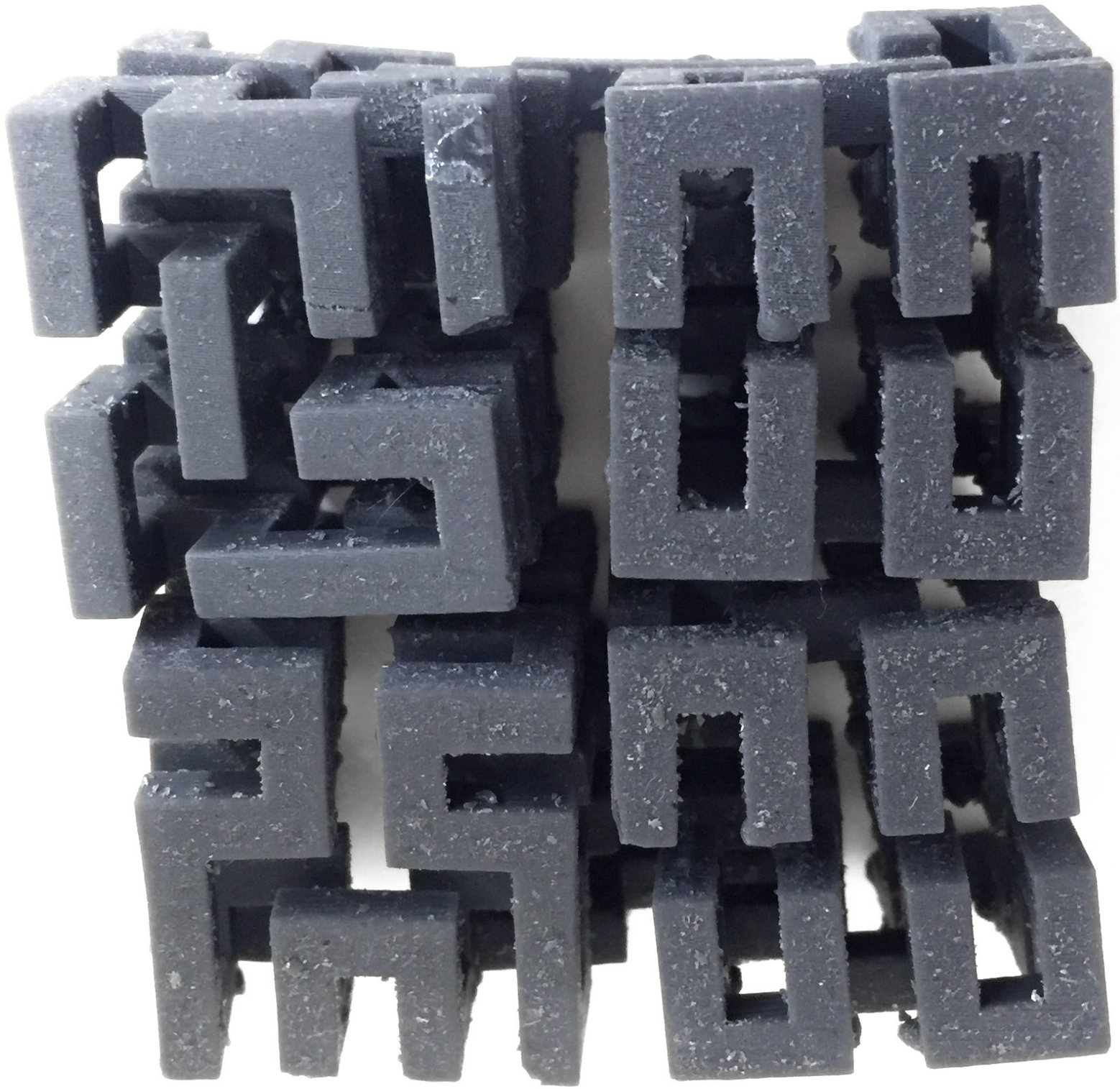
30%
15%
20%
35%
GEO
BITS
OPENSOURCE GEOMODELING

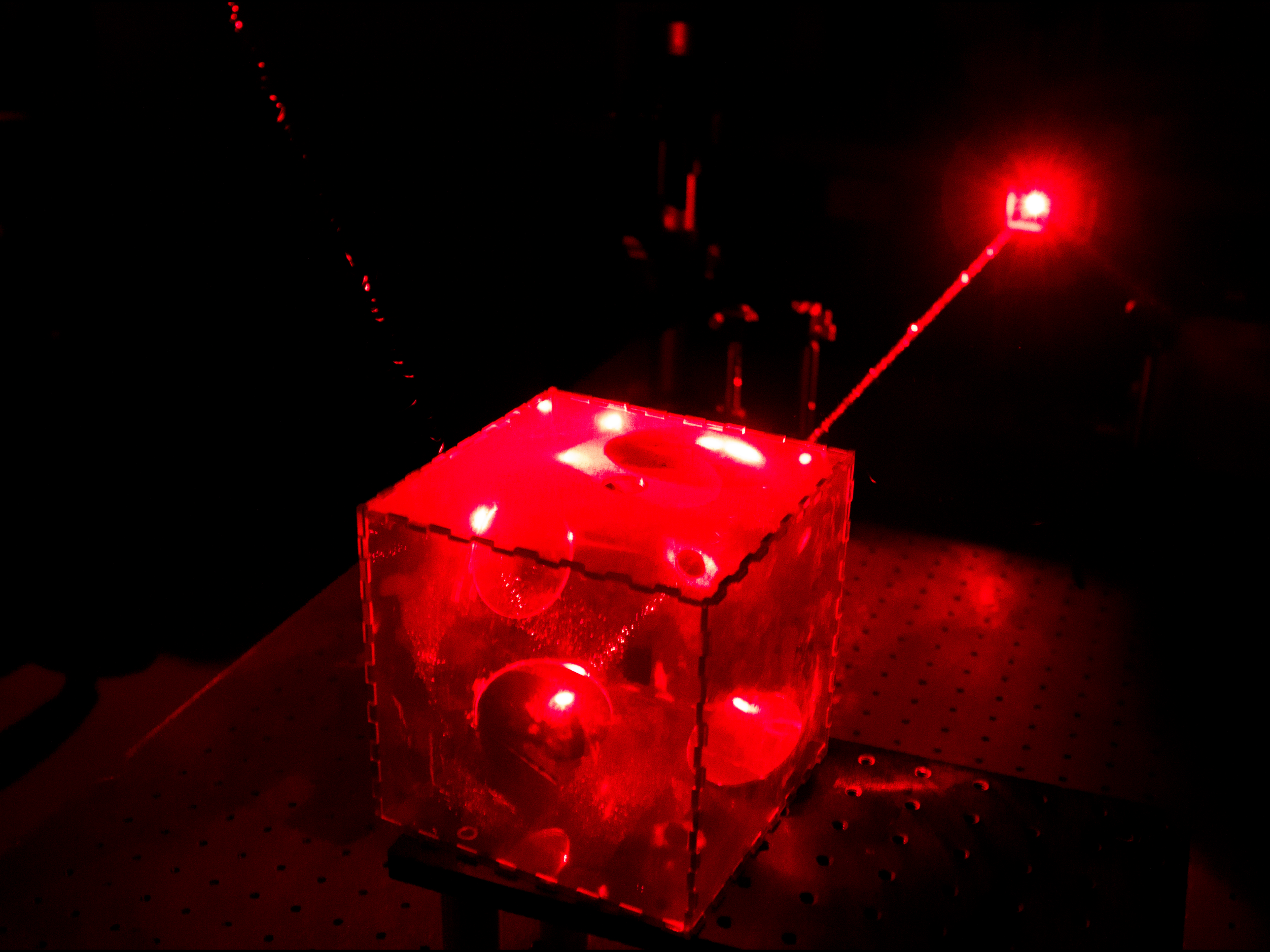

PharmaDSS BETA V1.0
MIT Media Lab + GlaxoSmithKline
Ira Winder, Nina Lutz, Giovanni Giorgio, Mason Briner, Joana Gomes
Operator
Fitter
Electrician
Manager
QA
Lab Staff
Candidate Selection
Launch
NCE Demand Profiles
2017
Actual
Forecast
Lead (Ph III)
End
Launch
Capacity
NCE 1. Double Forecast Demand
NCE 2. Competton
NCE 3. Blockbuster
NCE 4. Failure at phase IIB
NCE 5. AGSK Entity
NCE 6. Competton
NCE 7. Never Manufactured
NCE 8. Never Manufactured
NCE 9. Competton
NCE 10. Failure at phase IIB
Site Characteristics
Site 1
Built Capacity
Available Site
24t / 36t
Site 2
Built Capacity
Available Site
48t / 72t
R&D
MfG Capacity 'Chip'
Production Capacity 2 tons
Personnel:
Build Time: 3 y
Build Cost: 4.1£
Repurpose Time: 1 y
Repurpose Cost: 0.5£
Performance
Demand Met
Captal Expenses
Score
35.0 %
0.0 mil GBP
Year
2017
2022
2027
2032
2037
NCE
Shift
SAMSUNG

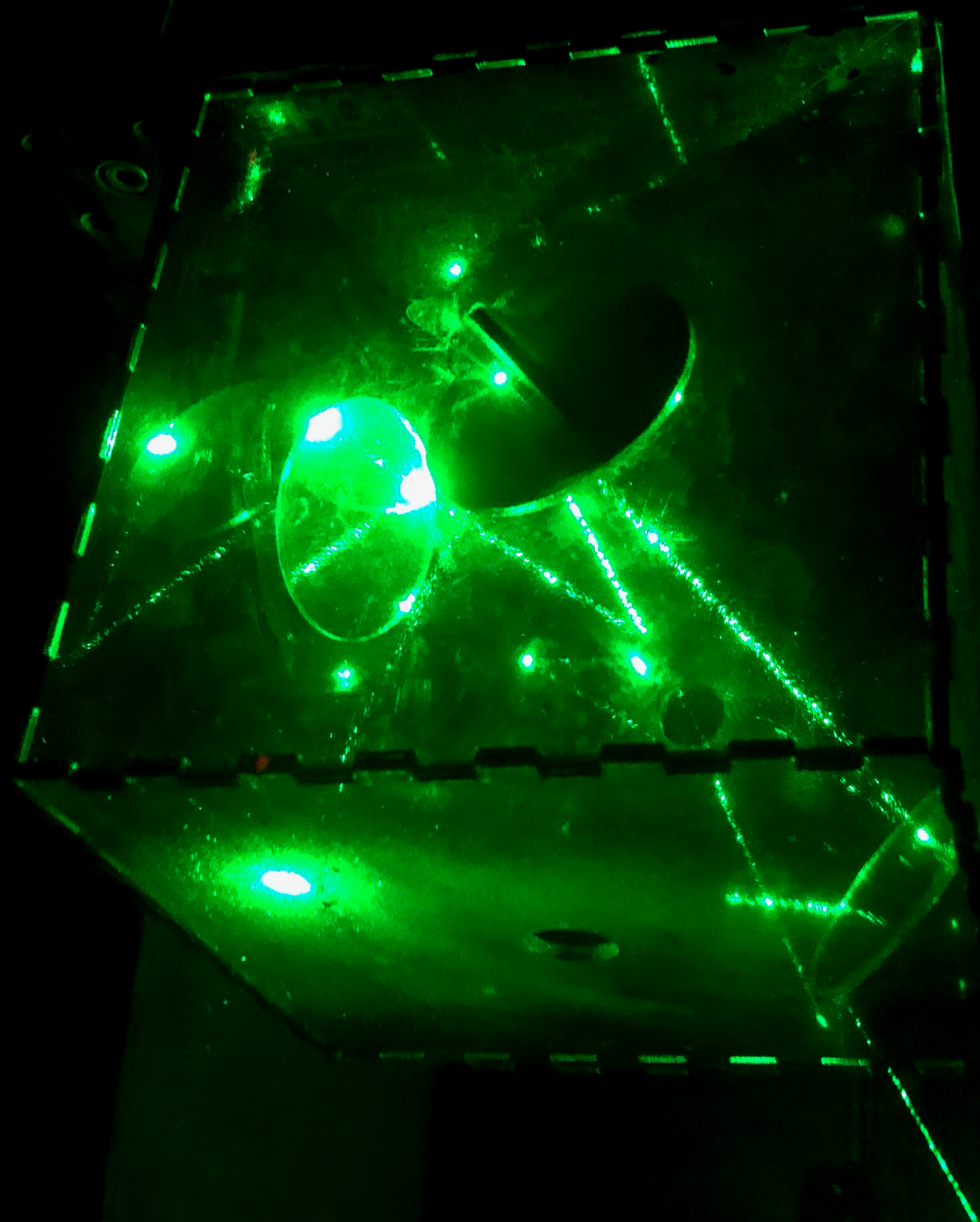

MIT Team: Nina Lutz, Joyce Chen, Ira Winder
GSK Team: Yusuf Roohani, Amy Taylor, Marti Head, Rusty Fries, Lynn Tucker
Synergy
Space Utilization
Human Flow
Interventions
Drag and drop equipment into selection to add.
Hub
Specialized Lab Space
Desk
Hood
Specialized Lab Space
Legend
Hub
Desk
Meeting
Hood
Specialized Lab Space

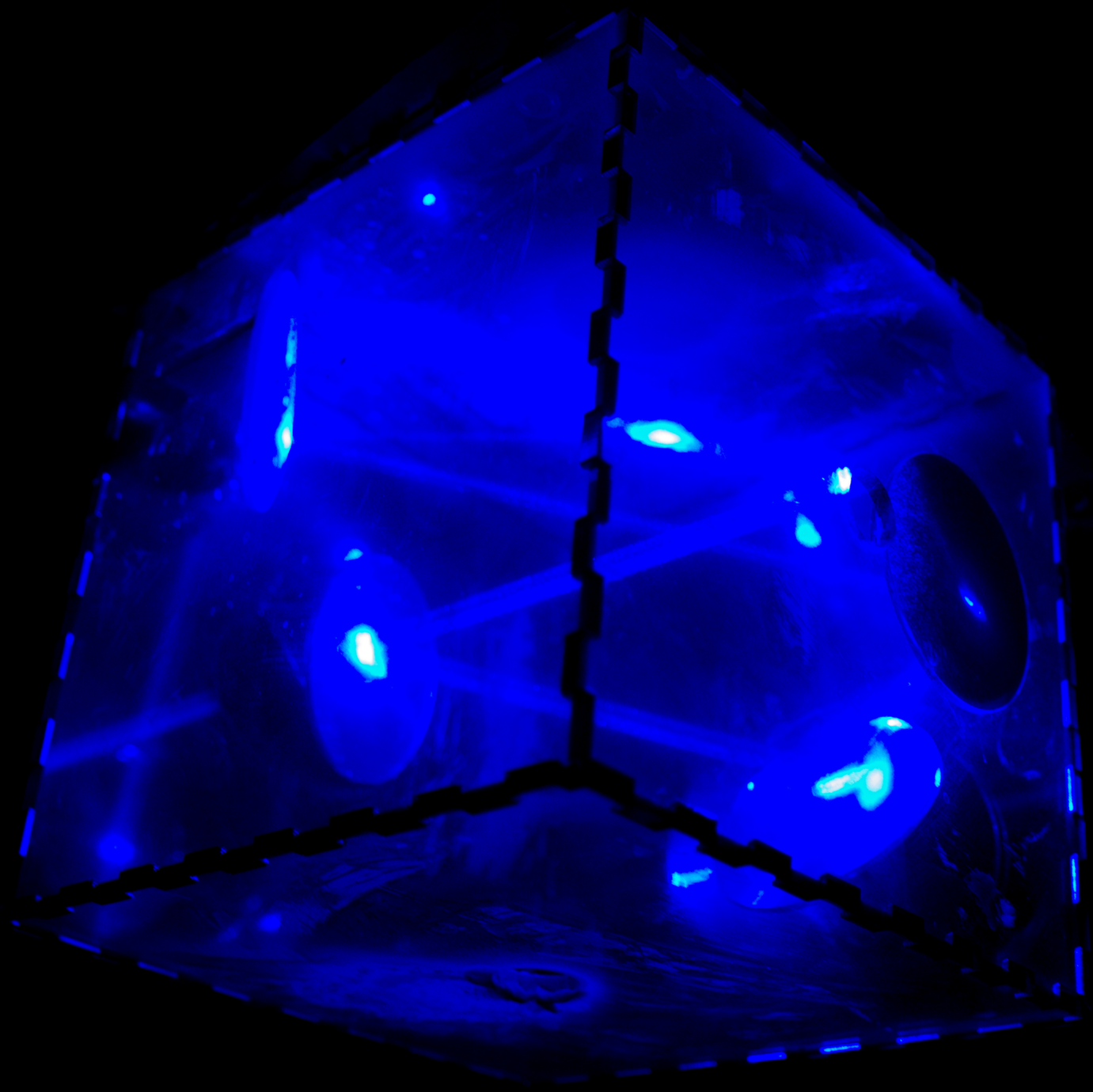


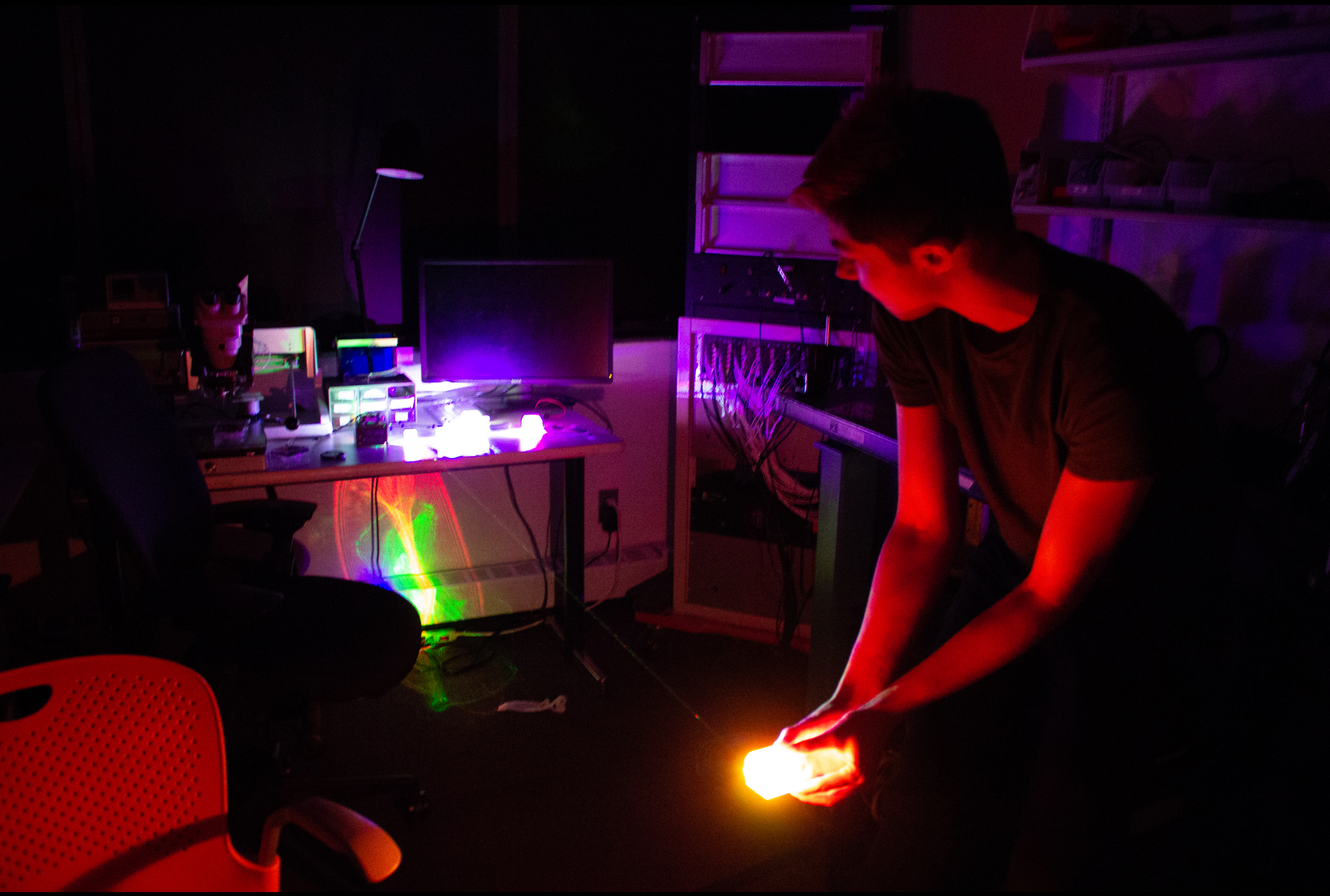












What will you learn this week?

Algorithmic thought

How to visualize algorithms

How to use algorithms in art with code and electronics

What is an algorithm?

What is an algorithm

A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

Essentially, how does a computer do xyz.

What an algorithm is NOT

Code - The algorithm is the instructions, not the software. However, some are often implemented in code and all can be written in **pseudocode.**

Math - some algorithmic paths can be considered a branch of mathematics but in reality they are **logic.**

Restricted to Computer Science — many disciplines have them.

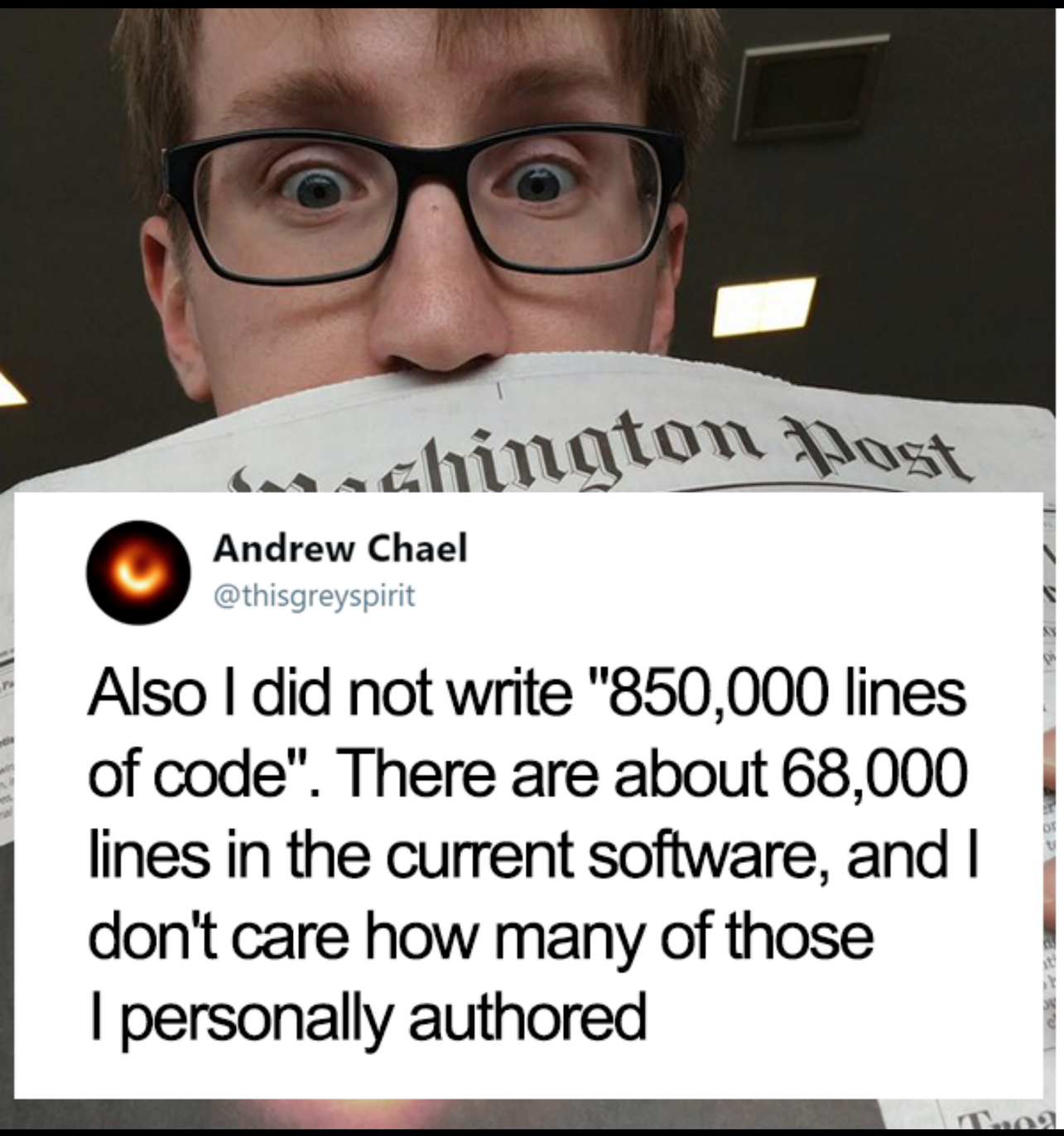
Computer Science vs Software Engineering

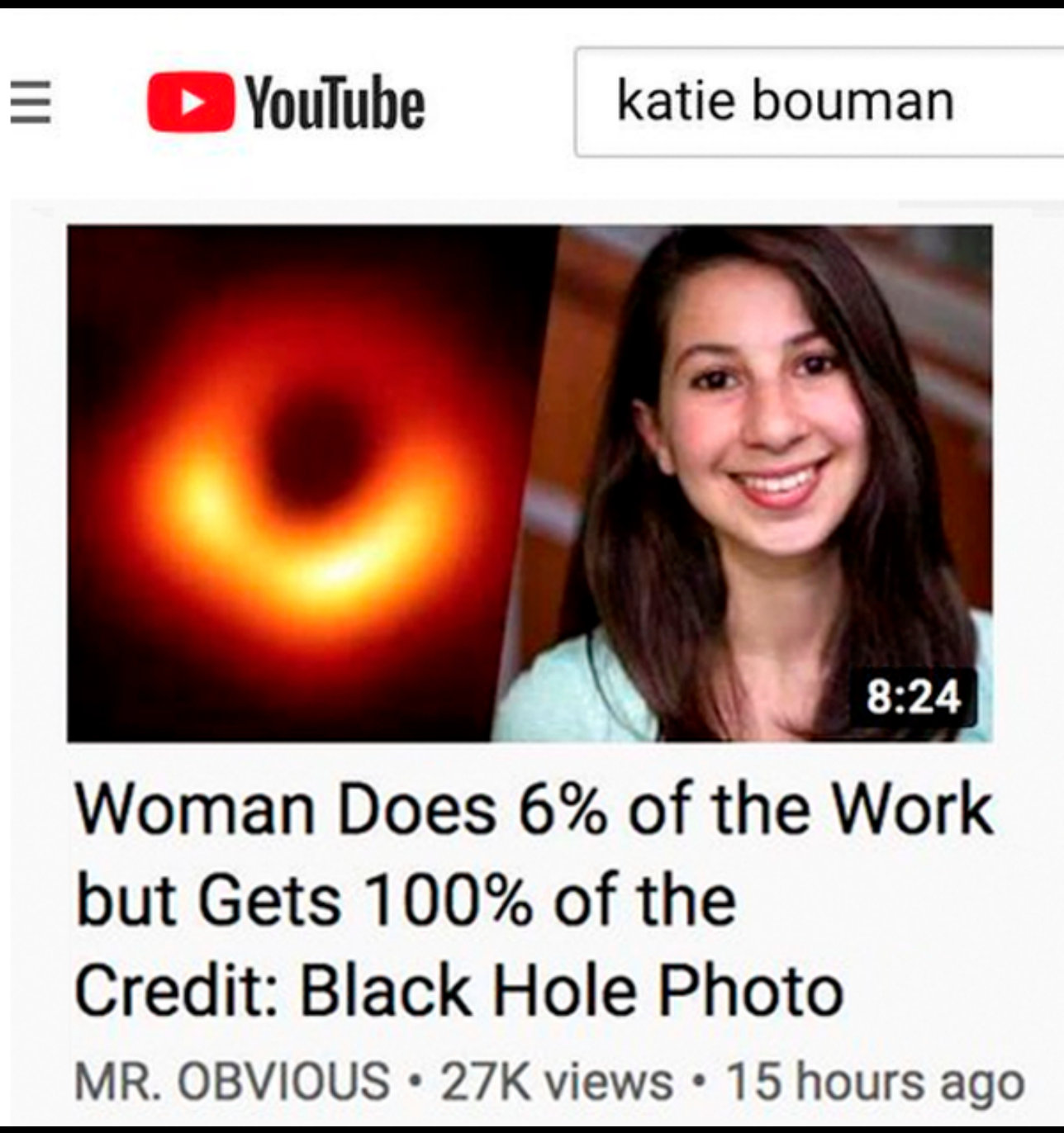
Software engineering — building working software for a variety of devices and purposes.

Computer science — the study of how data interacts in digital form.

Software engineering is part of computer science skillsets, but algorithms are not necessarily software engineering — but they help!

Thus, algorithms are not super verbose.
And in computer science line count is not
a good way to count anything.





Pseudocode

A way to write instructions that looks like code, but that probably wouldn’t be understood by a computer as code.

Example

- I have 12 cupcakes, I want to frost all of them.
- I want to alternate the colors of two frostings: red and blue.
- I can only frost one cupcake at a time and I have to wait 5 seconds between each frosting.

Pseudocode for cupcakes

```
count
for each cupcake:
   if count even: frost red
   if count odd: frost blue
   return cupcake
   delay 5 seconds
```

Example pseudocode:

```
dist[s] ←o
for all v ∈ V–{s}
      do dist[v] ←∞
S←∅
Q←V
while Q ≠∅
do   u ← mindistance(Q,dist)
     S←S ∪{u}
      for all v ∈ neighbors[u]
            do if  dist[v] > dist[u] + w(u, v)
                        then     d[v] ←d[u] + w(u, v)


return dist
```

Some Algorithm Families

Often divided by data structure or by method of the algorithm

There are so many algorithms go crazy!

https://en.wikipedia.org/wiki/List_of_algorithms

Types of Algorithms We'll Think About

Sorting

Searching

Pathfinding

Efficiency

Algorithms are great at instructions
— but they also need to be efficient
in timing and often in space.

Algorithmic Time

Big O notation
Time complexity
Runtime analysis

The computational complexity to run a process.

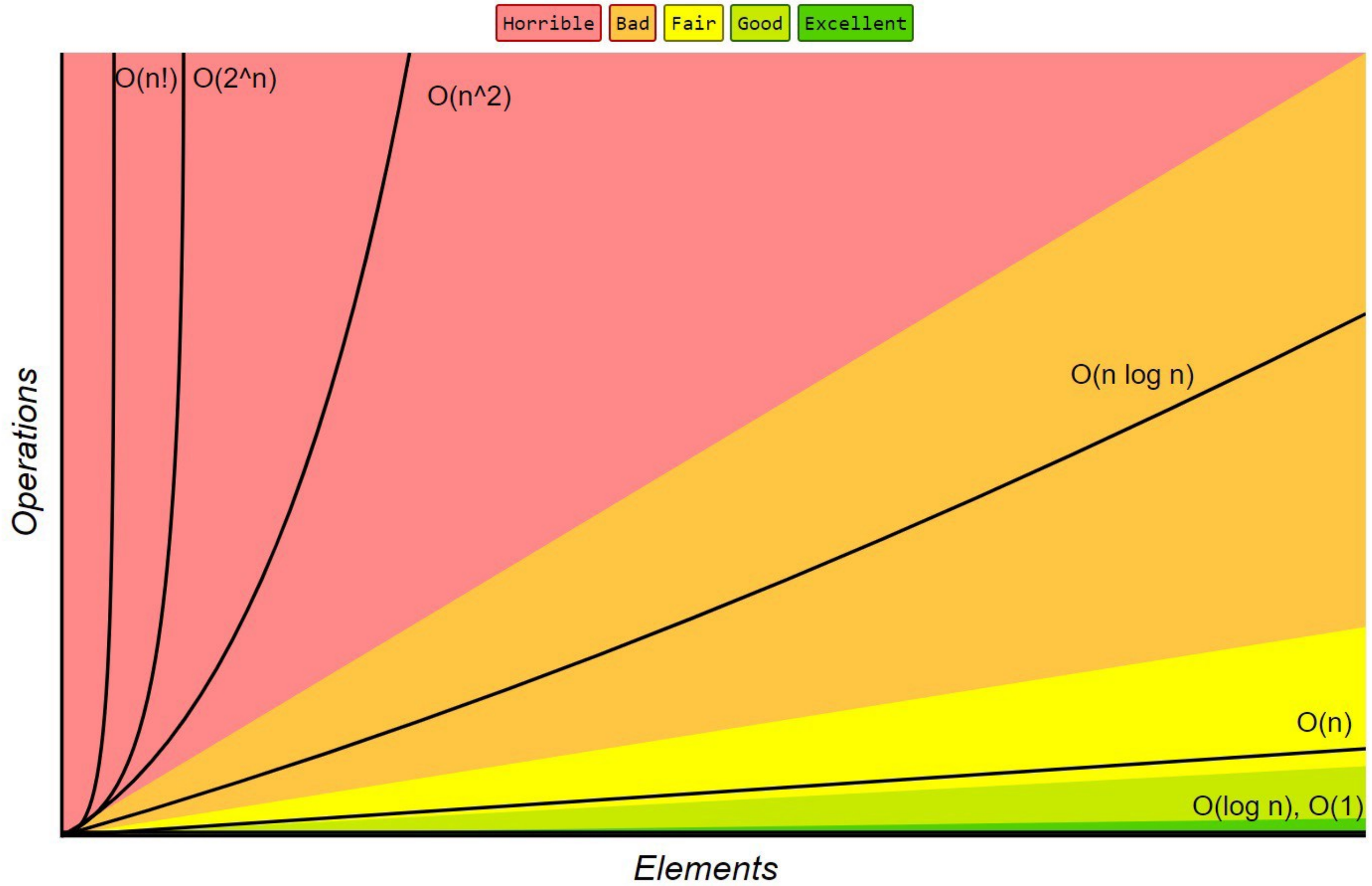
Big O Notation

- O(n) is a linear time algorithm that takes n bits of logic. n is usually the number of items and other constants have to do with the operations within it.
    - For example, our frosting algorithm takes n cupcakes with one process. Technically with our 5 second delay its O(5n) but it is still linear.

Ranking of Big O Notation

O(1)
O(lgn)
O(n)
O(nlgn)
O(n^2)
O(2^n)
O(n!)

Big-O Complexity Chart



Good rule of thumb

- You can usually denote how long an algorithm will take with its pseudo code based on:
  - How many elements you’re performing operations on
  - How long those operations take
  - Any comparisons or duplicities of these operations

Example of O(n^2)

- You have n cupcakes, each is frosted red or blue and has a label on the bottom.
- You want to frost all of them, this is O(n)
- Now you have to go through each cupcake and see if there are duplicates. A naive way of doing this is O(n^2) — for each cupcake's label you check it against every other cupcakes. You do n checks per cupcake, which is n*n or n^2.

Our Toolkit This Week

Processing

Arduino

Github