

Coding for Concerts

Chris Piech + Alex
Green + Kyle
Fleming

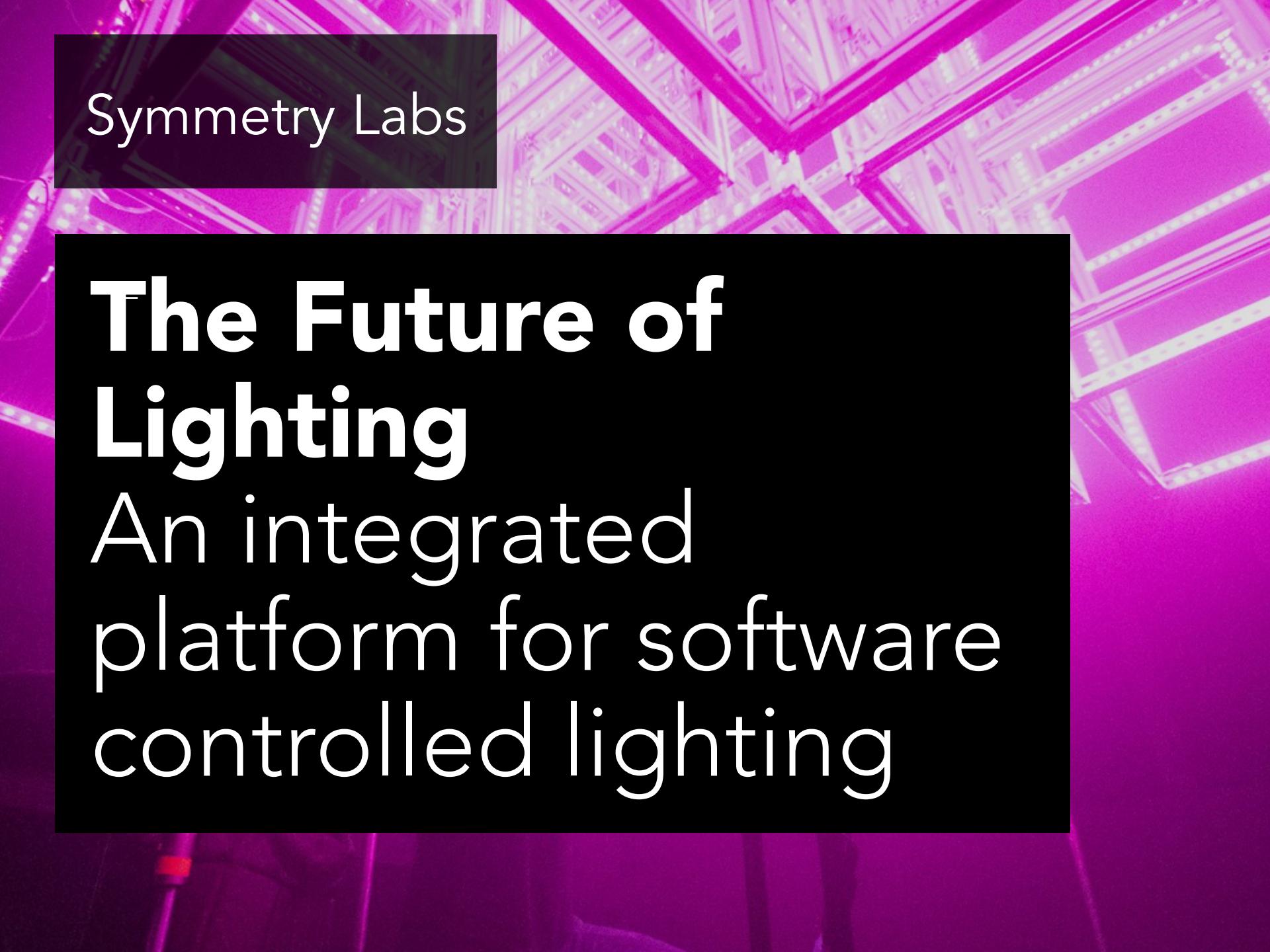
CS 106B
Lecture 24
Mar 9, 2016

Today's Goal

1. See “real” C++
2. Be inspired



Special Guests

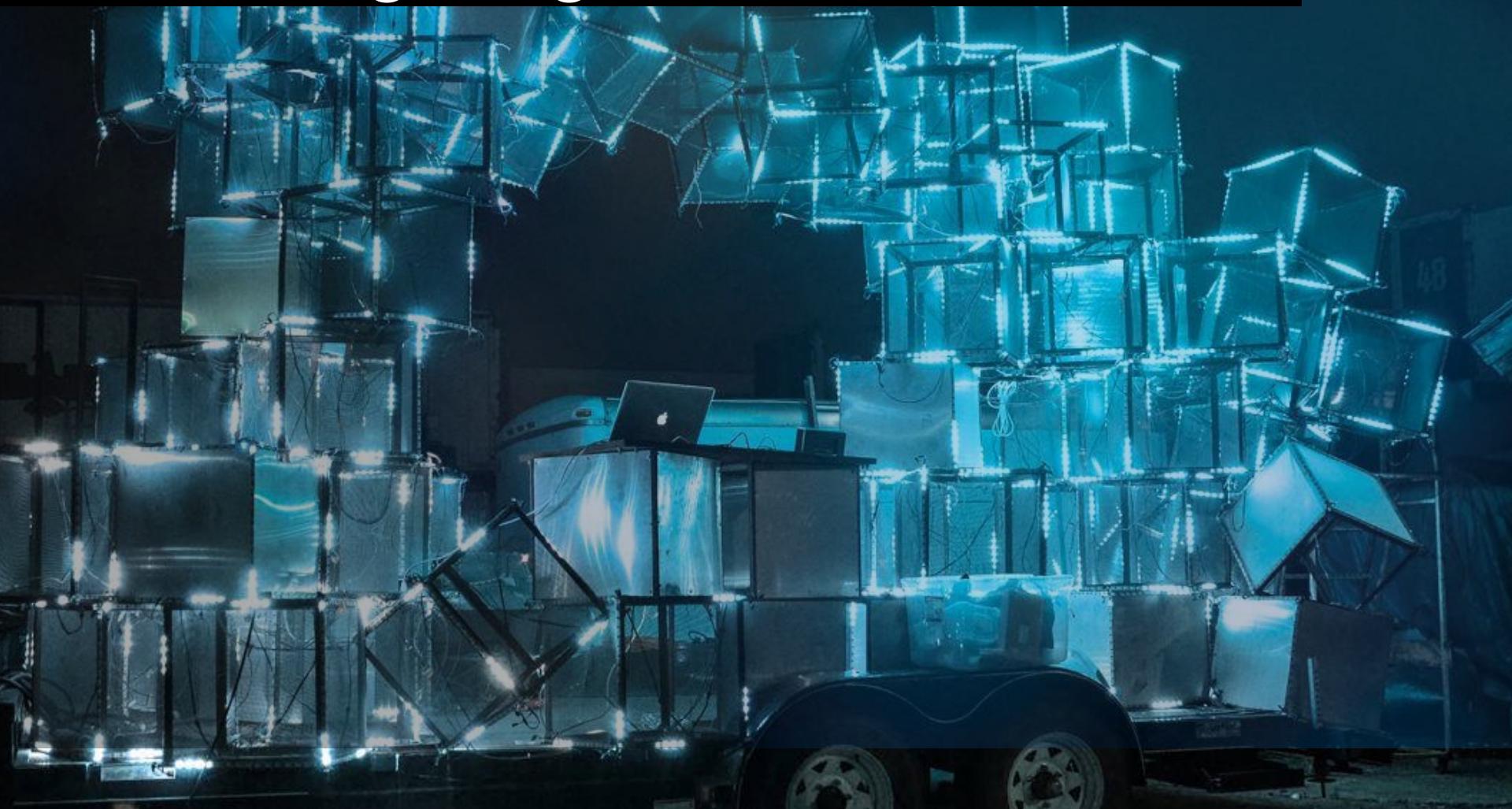


Symmetry Labs

The Future of Lighting

An integrated
platform for software
controlled lighting

It started with a simple idea: to make lighting more like music



- Lighting today isn't really controlled by algorithms
- It's either video or it's “programmed” by sequences of switches, essentially hard coded in, using something like this

Abstractions are incredibly powerful!

```
/*
 * Model of a single cube, which has an orientation and position on the
 * car. The position is specified in x,y,z coordinates with rotation. The
 * x axis is left->right, y is bottom->top, and z is front->back.
 *
 * A cube's x,y,z position is specified as the left, bottom, front corner.
 *
 * Dimensions are all specified in real-world inches.
 */
public static class Cube extends LXModel {

    public final static int FACES_PER_CUBE = 4;
    public static final int POINTS_PER_STRIP = 16;

    public final static int STRIPS_PER_CUBE = FACES_PER_CUBE*Face.STRIPS_PER_FACE;
    public final static int POINTS_PER_CUBE = STRIPS_PER_CUBE*POINTS_PER_STRIP;
    public final static int POINTS_PER_FACE = Face.STRIPS_PER_FACE*POINTS_PER_STRIP;

    public final static float EDGE_HEIGHT = 24;
    public final static float EDGE_WIDTH = 24;
    public final static float CHANNEL_WIDTH = 1.5f;

    public final static Face.Metrics FACE_METRICS = new Face.Metrics(
        new Strip.Metrics(EDGE_WIDTH, POINTS_PER_STRIP),
        new Strip.Metrics(EDGE_HEIGHT, POINTS_PER_STRIP)
    );

    public final int boardNum;
    public final String macAddress;
    public final String id;

    /**
     * Immutable list of all cube faces
     */
    public final List<Face> faces;

    /**
     * Immutable list of all strips
     */
    public final List<Strip> strips;

    /**
     * Front left corner x coordinate
     */
    public final float x;

    /**
     * Front left corner y coordinate
     */
    public final float y;
```

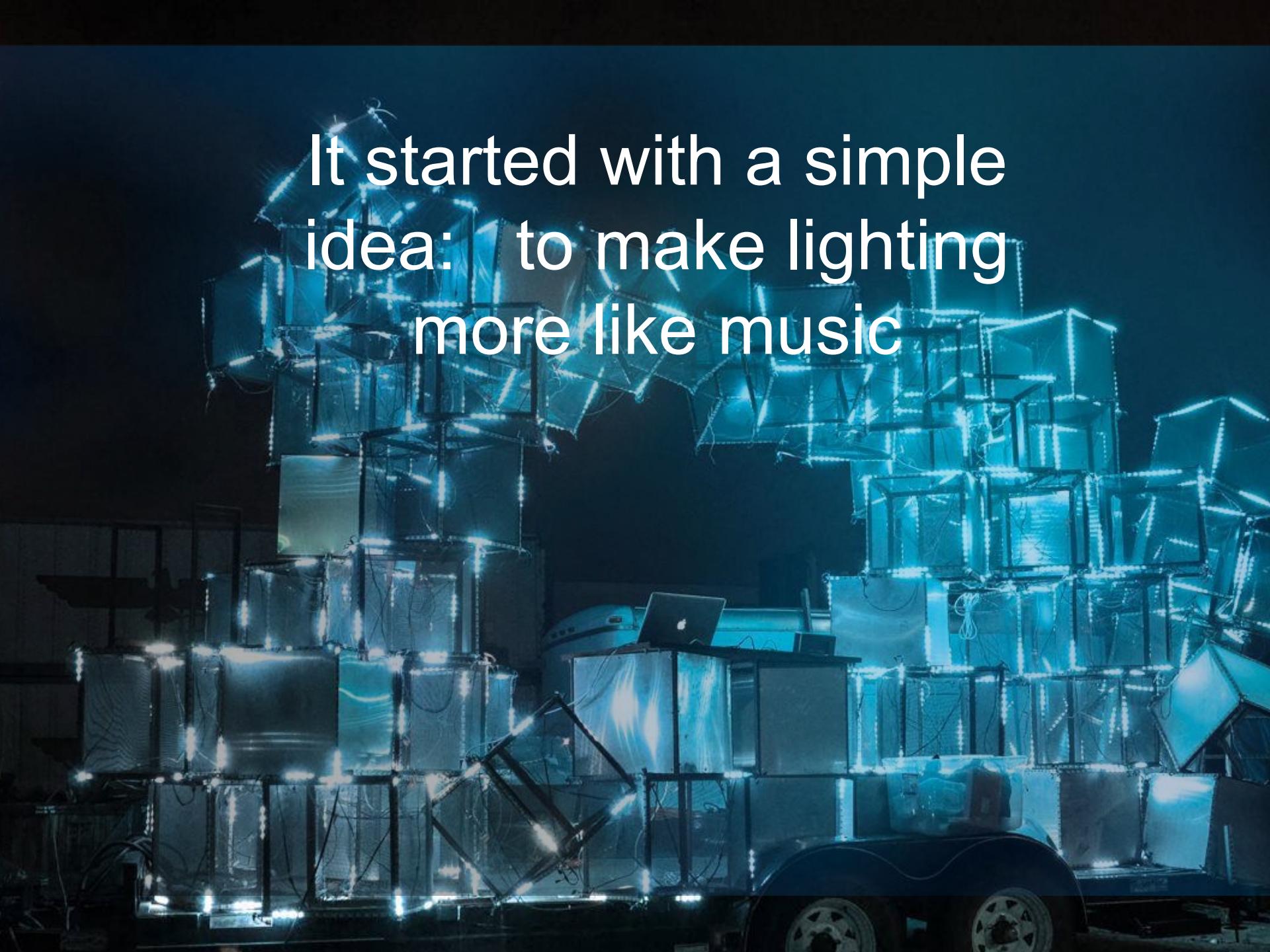
- Lighting today isn't really controlled by algorithms
- It's either video or it's “programmed” by sequences of switches, essentially hard coded in, using something like this



57 Band Practice
108 Pro Service
138 Drama
200 Outreach
281 Band Warming

SENDING DRY

Pan Out: 57 Cen: 10 Rec: Cen: 100
• 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115
• 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135

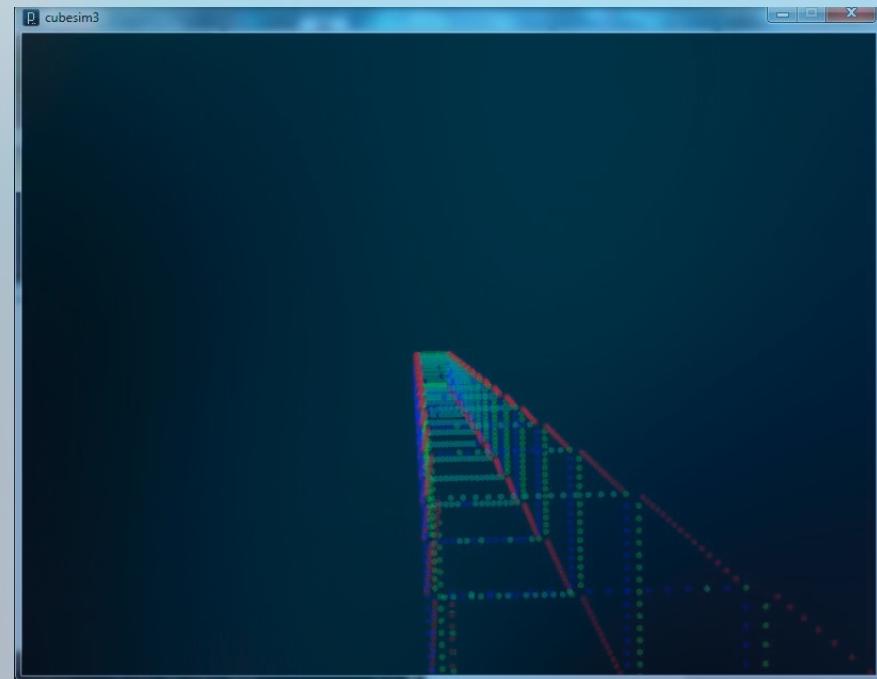


It started with a simple
idea: to make lighting
more like music



In reality, there was a lot of
this

and this



and this (Customs is a pain)

Track DHL Express Shipments

DHL's online tracking is the fastest way to find out where your shipment is. No need to call Customer Service when we can offer you real-time details of your shipment's progress as it speeds through the DHL Network on the way to its destination.

Result Summary

Waybill: 5115121745	Thursday, August 16, 2012 at 12:43				
 Clearance processing complete at LOS ANGELES GATEWAY - USA	Origin Service Area: > HONG KONG - HONG KONG - HONG KONG				
> Sign up for shipment notifications	Destination Service Area: > SAN FRANCISCO, CA - SAN FRANCISCO - USA				
<hr/>					
Thursday, August 16, 2012					
11	Clearance processing complete at LOS ANGELES GATEWAY - USA	Location	LOS ANGELES GATEWAY, CA - USA	Time	12:43
<hr/>					
Wednesday, August 15, 2012					
10	Clearance delay	Location	LOS ANGELES GATEWAY, CA - USA	Time	12:49
9	Please contact DHL	Location	LOS ANGELES GATEWAY, CA - USA	Time	00:59
8	Processed for clearance at LOS ANGELES GATEWAY - USA	Location	LOS ANGELES GATEWAY, CA - USA	Time	00:47
7	Clearance delay	Location	LOS ANGELES GATEWAY, CA - USA	Time	00:40
6	Arrived at Sort Facility LOS ANGELES GATEWAY - USA	Location	LOS ANGELES GATEWAY, CA - USA	Time	00:27
<hr/>					
Tuesday, August 14, 2012					
5	Departed Facility in HONG KONG - HONG KONG	Location	HONG KONG - HONG KONG	Time	14:45
4	Processed at HONG KONG - HONG KONG	Location	HONG KONG - HONG KONG	Time	14:31
3	Arrived at Sort Facility HONG KONG - HONG KONG	Location	HONG KONG - HONG KONG	Time	05:45
2	Processed at HONG KONG - HONG KONG	Location	HONG KONG - HONG KONG	Time	03:11
<hr/>					
Monday, August 13, 2012					
1	Shipment picked up	Location	HONG KONG - HONG KONG	Time	20:25

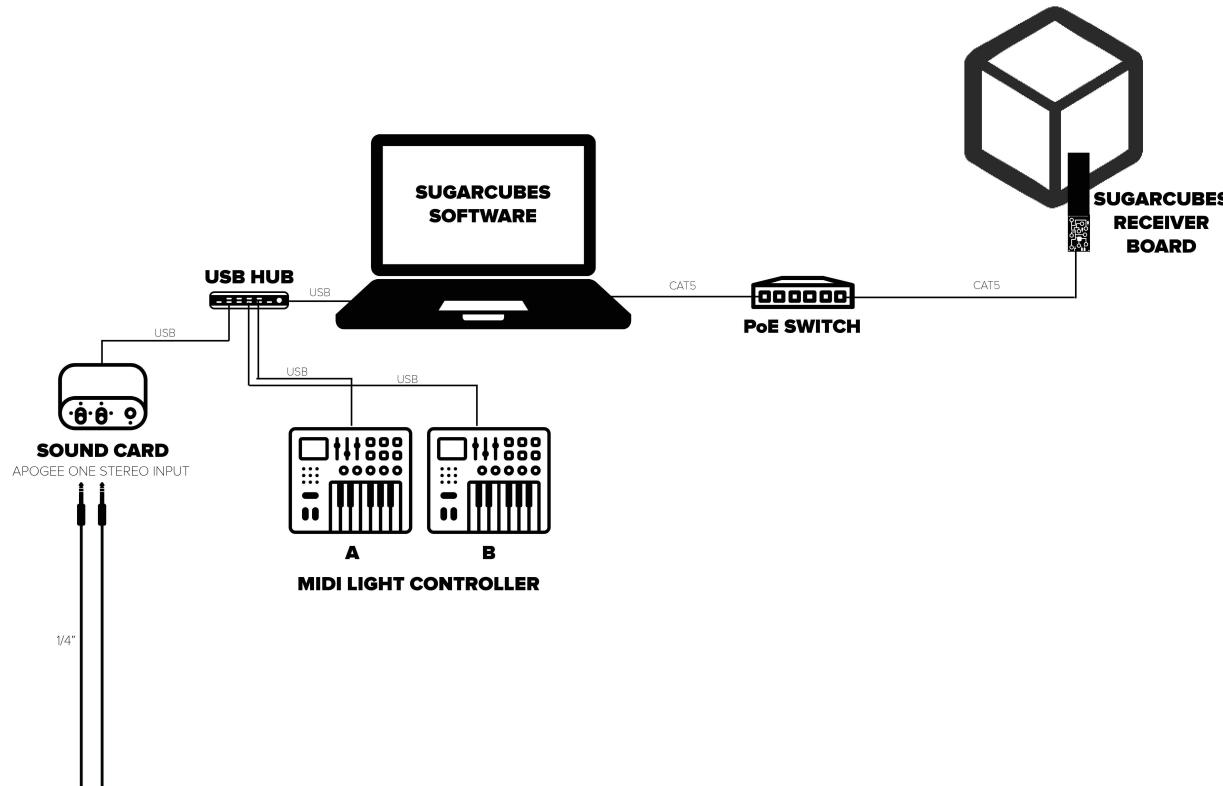
and this



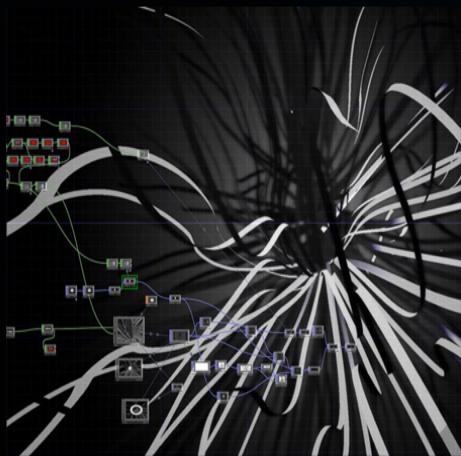
System Overview

SYMMETRY
LABS

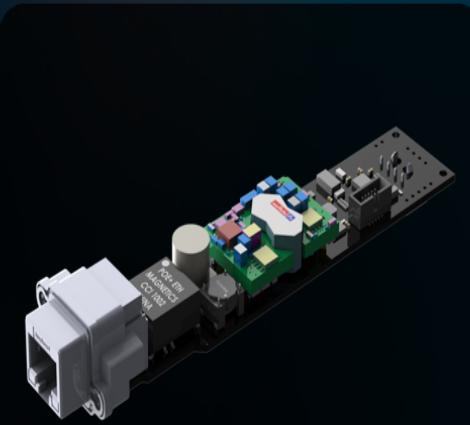
SUGARCUBES INSTALLATION GUIDE



We are innovating across a range of technologies to build a modern platform for dynamic lighting



3D content platform
+ control software



lighting controller
hardware + firmware

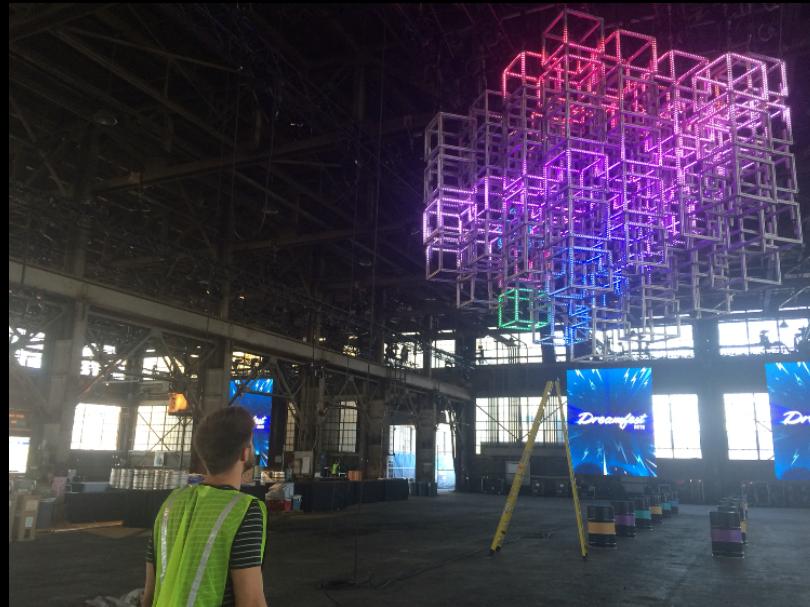


algorithmic physical design

digital ← → physical



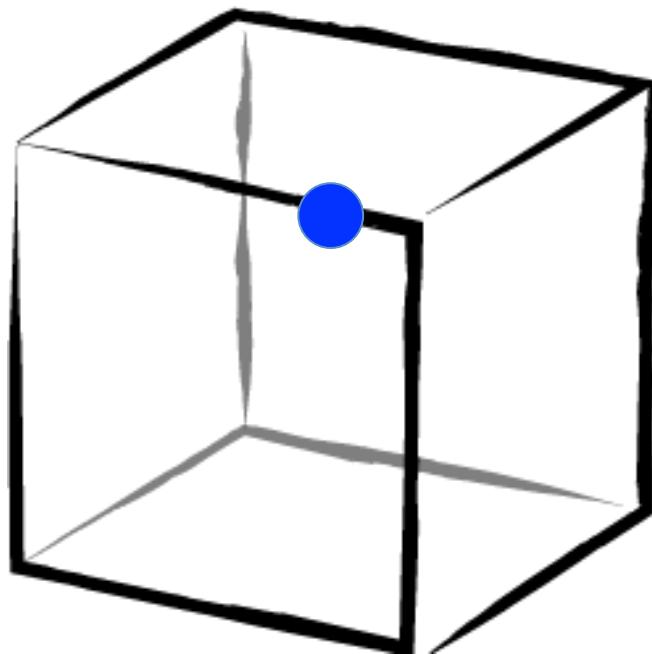




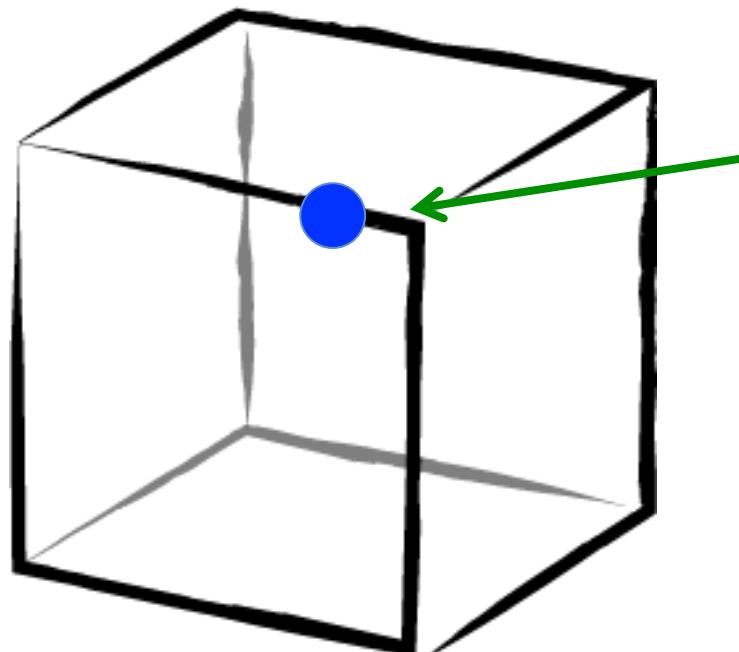
Switch to Kyle

Pipeline

Sugar Cubes



Job is to Set “Pixels”



LED is like a pixel

Updated 60x per second

The Pipeline

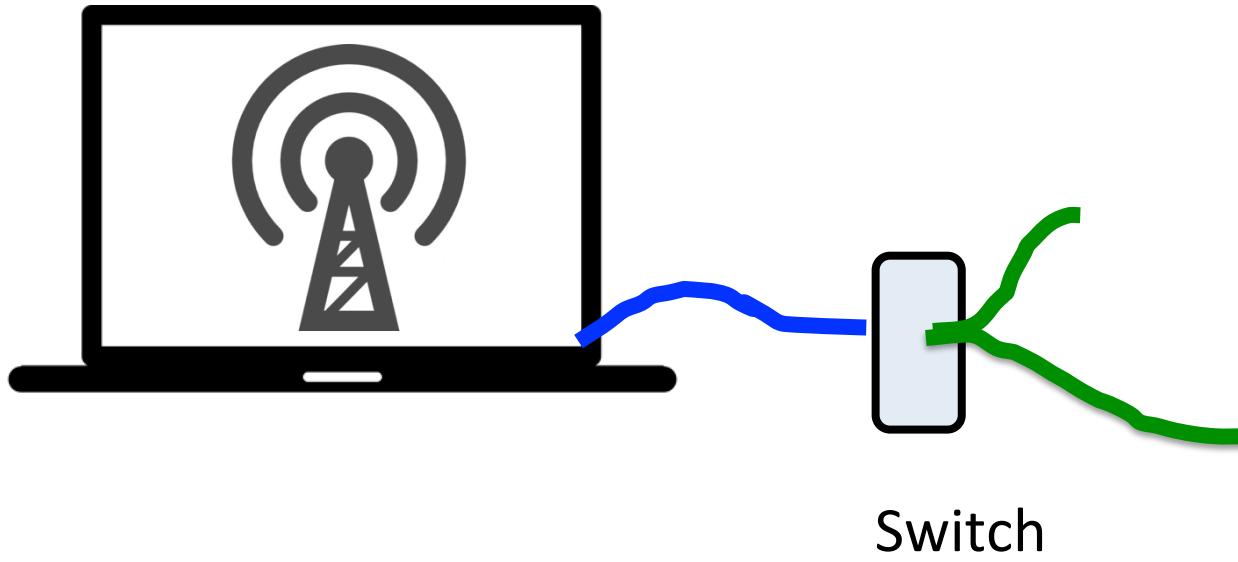
Your Program



Computer -> Switch -> Pixel

The Pipeline

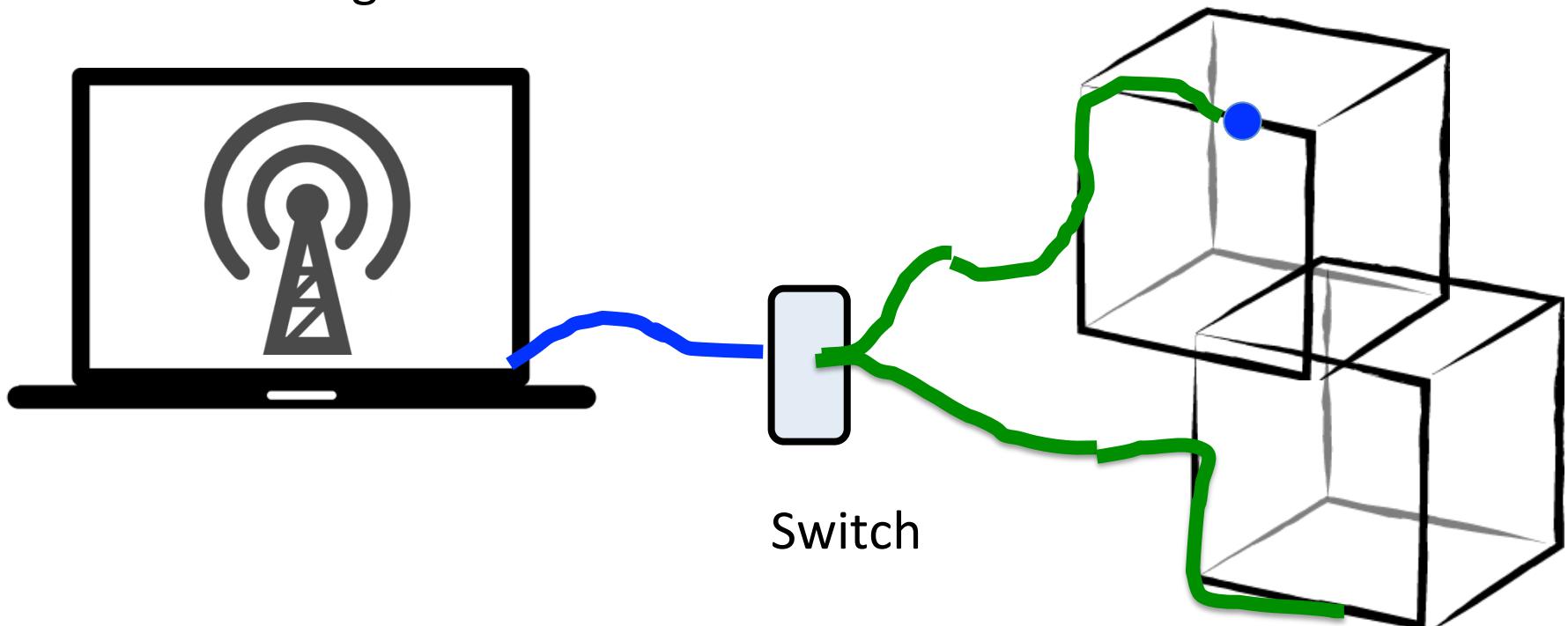
Your Program



Computer -> Switch -> Pixel

The Pipeline

Your Program



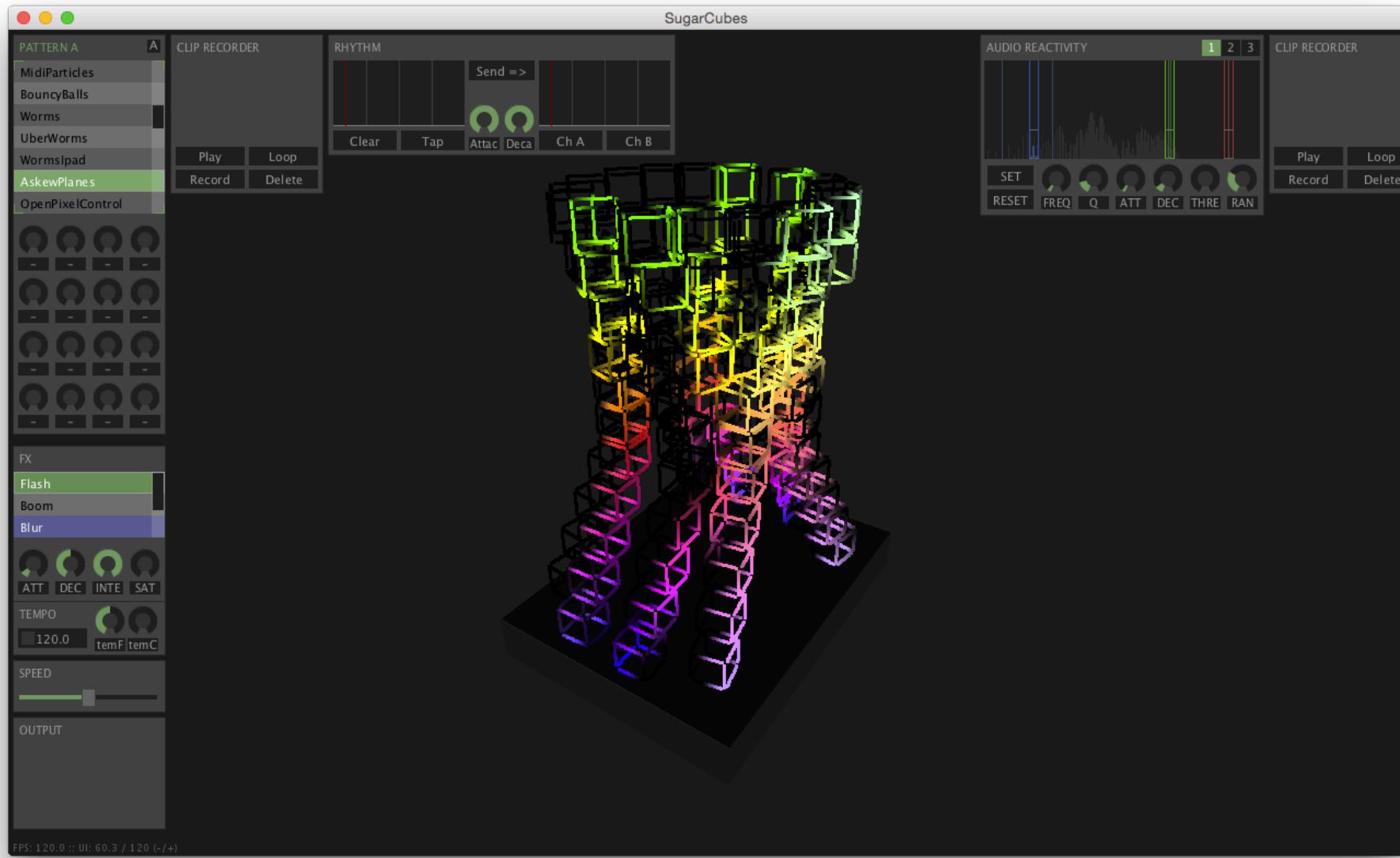
Computer -> Switch -> Pixel

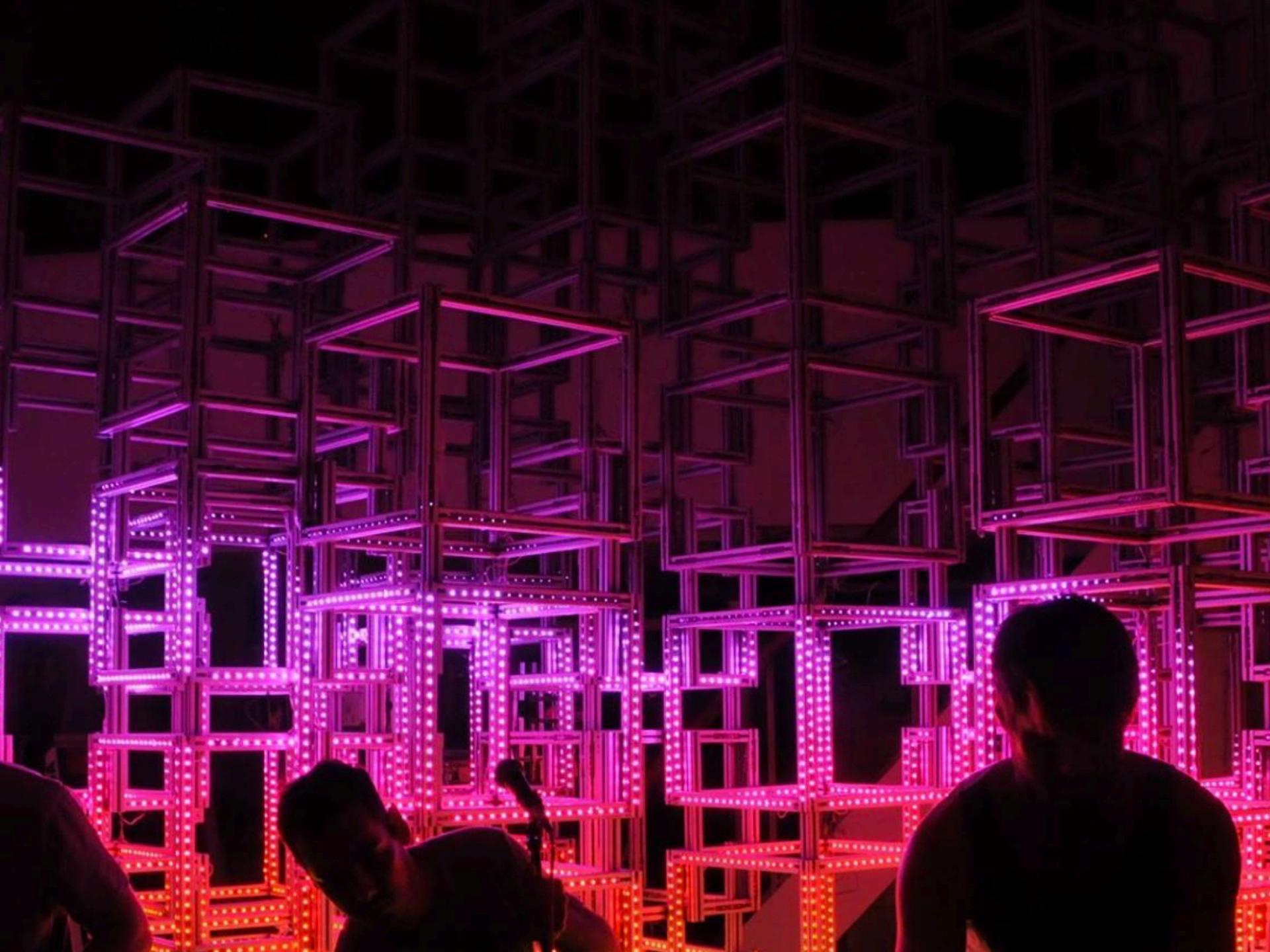
Colors



In Performance (Artistic Inputs)

The Interface





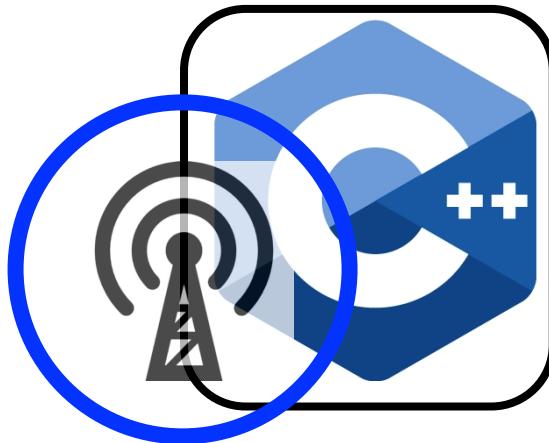
Switch to Chris

Alex + Kyle agreed to let us build in their codebase!

Where we left off...

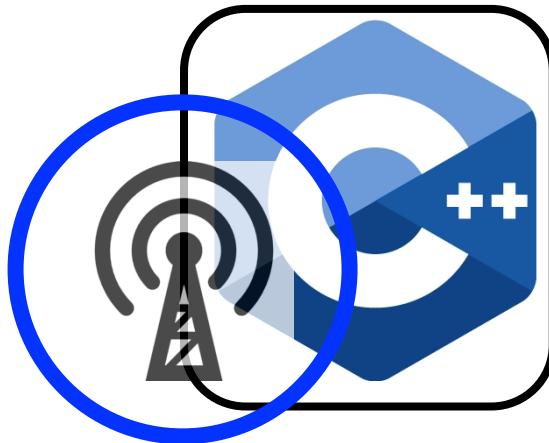
Listener Model

Artistic Program



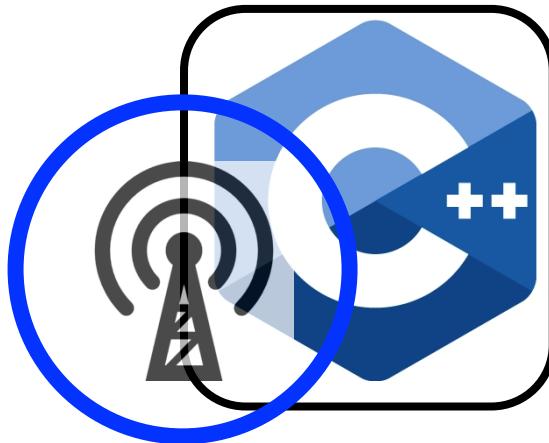
Listener Model

Artistic Program



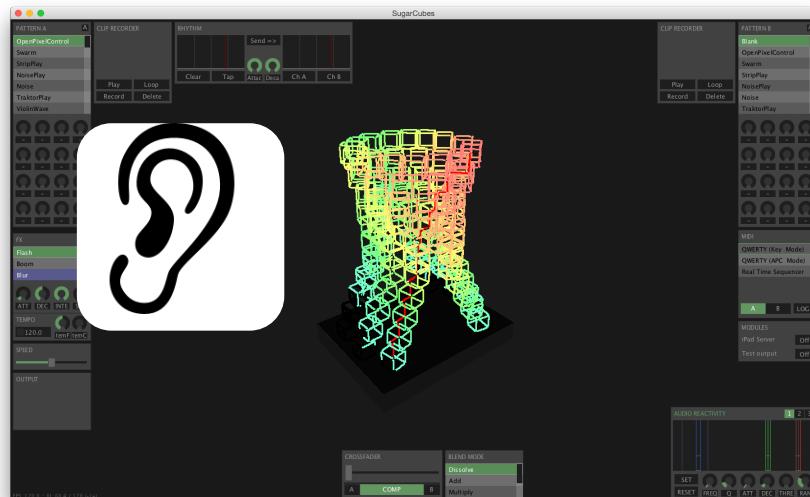
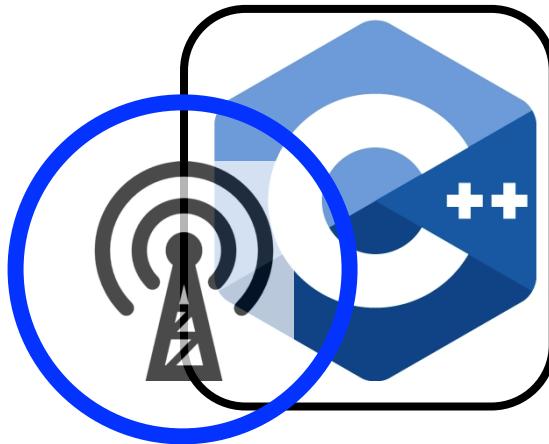
Listener Model

Artistic Program

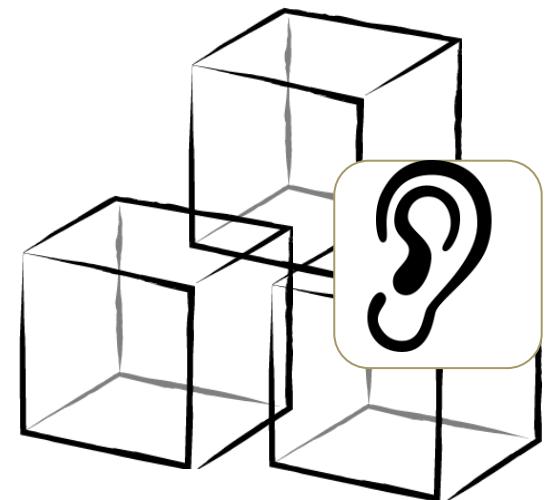


Listener Model

Artistic Program



Visualizer Program



Actual Cubes

Effect Class

Artistic Program

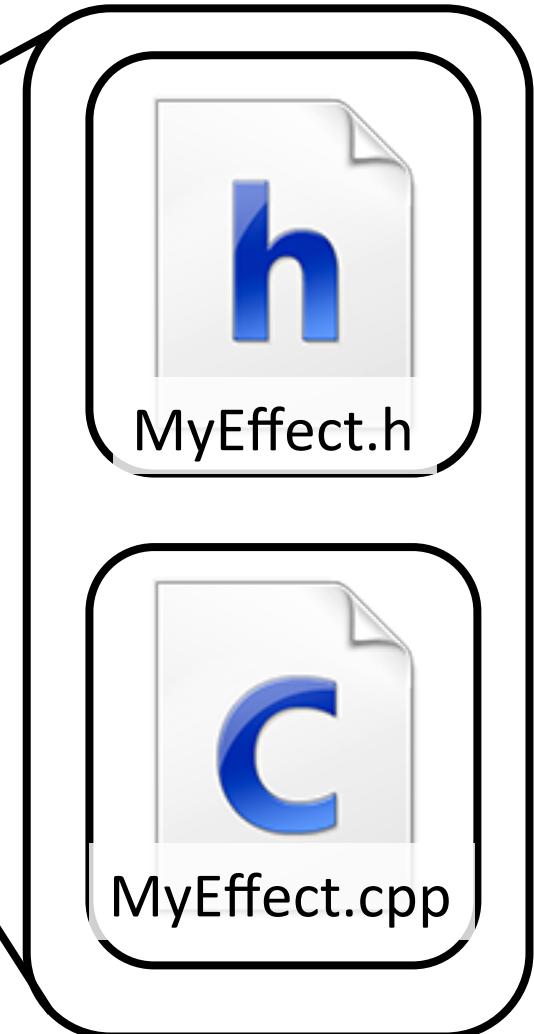


Effect Class

Artistic Program

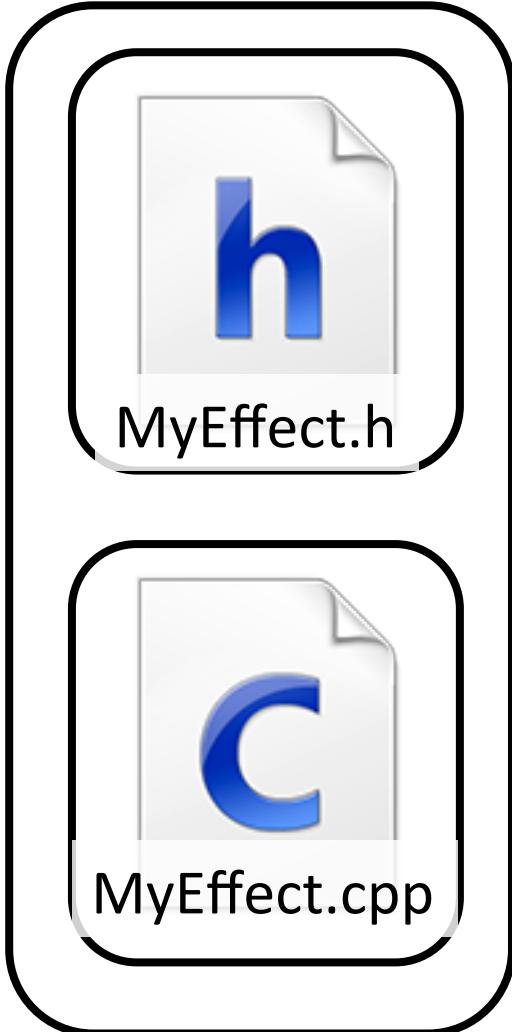


New Effect Class



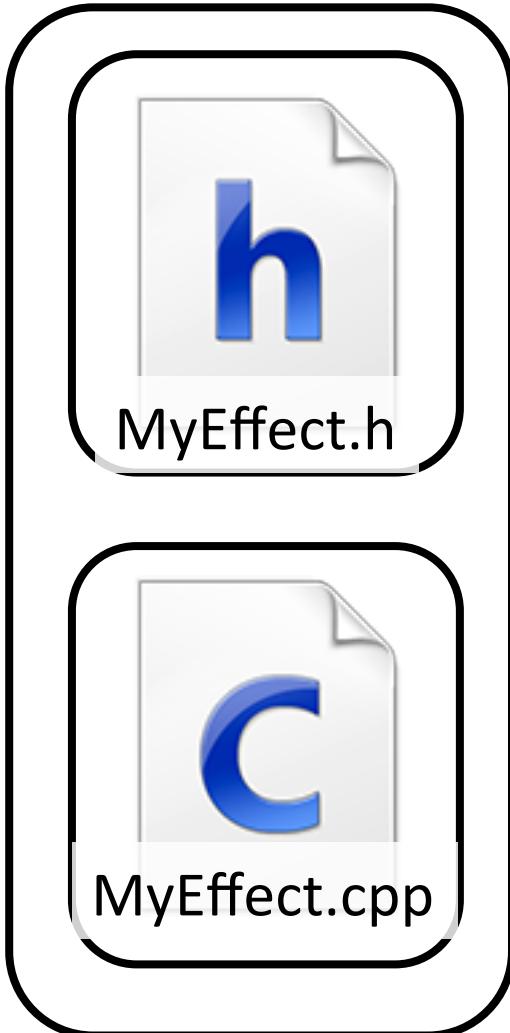
Effect Class

New Effect Class



Effect Class

New Effect Class



```
void onStart(allPixelData)
```

```
void beginFrame()
```

```
void shader(rgb, pixelInfo)
```

Passed by reference

Lets do it

A photograph of a woman with blonde hair tied back, wearing a tan short-sleeved shirt and tan shorts, crouching in a lush green jungle. She is handing a small chimpanzee a piece of food from her hand. The chimpanzee is dark-furred and is looking up at her. The background is filled with dense tropical foliage.

C++ in the Wild

vector - C++ Reference

Chris Piech

www.cplusplus.com/reference/vector/vect...

cplusplus.com

Search: Go

Reference <vector> vector

Standard Template Libraries

class template

std::vector

`template < class T, class Alloc = allocator<T> > class vector`

Vector

Vectors are sequence containers representing arrays that can change in size.

Just like arrays, vectors use contiguous storage locations for their elements. Elements can be accessed using offsets on regular pointers to its elements, and just like arrays, the size can change dynamically, with their storage being automatically handled by the container.

Internally, vectors use a dynamically allocated array to store their elements. This allows the vector to grow in size when new elements are inserted, which implies a reallocation of memory. This is a relatively expensive task in terms of processing time, and it is something that happens every time a new element is added to the container.

Instead, vector containers may allocate some extra storage to accommodate growth. The container may have an actual **capacity** greater than the storage strictly required. Libraries can implement different strategies for growth to balance between space and time efficiency. In general, reallocations should only happen at logarithmically growing intervals. Inserting elements at the end of the vector can be provided with **amortized constant time complexity**.

Therefore, compared to arrays, vectors consume more memory in exchange for growth, but they can grow dynamically in an efficient way.

Waiting for lax-46.lax-rtb1.rfihub.net...

C++ Reference

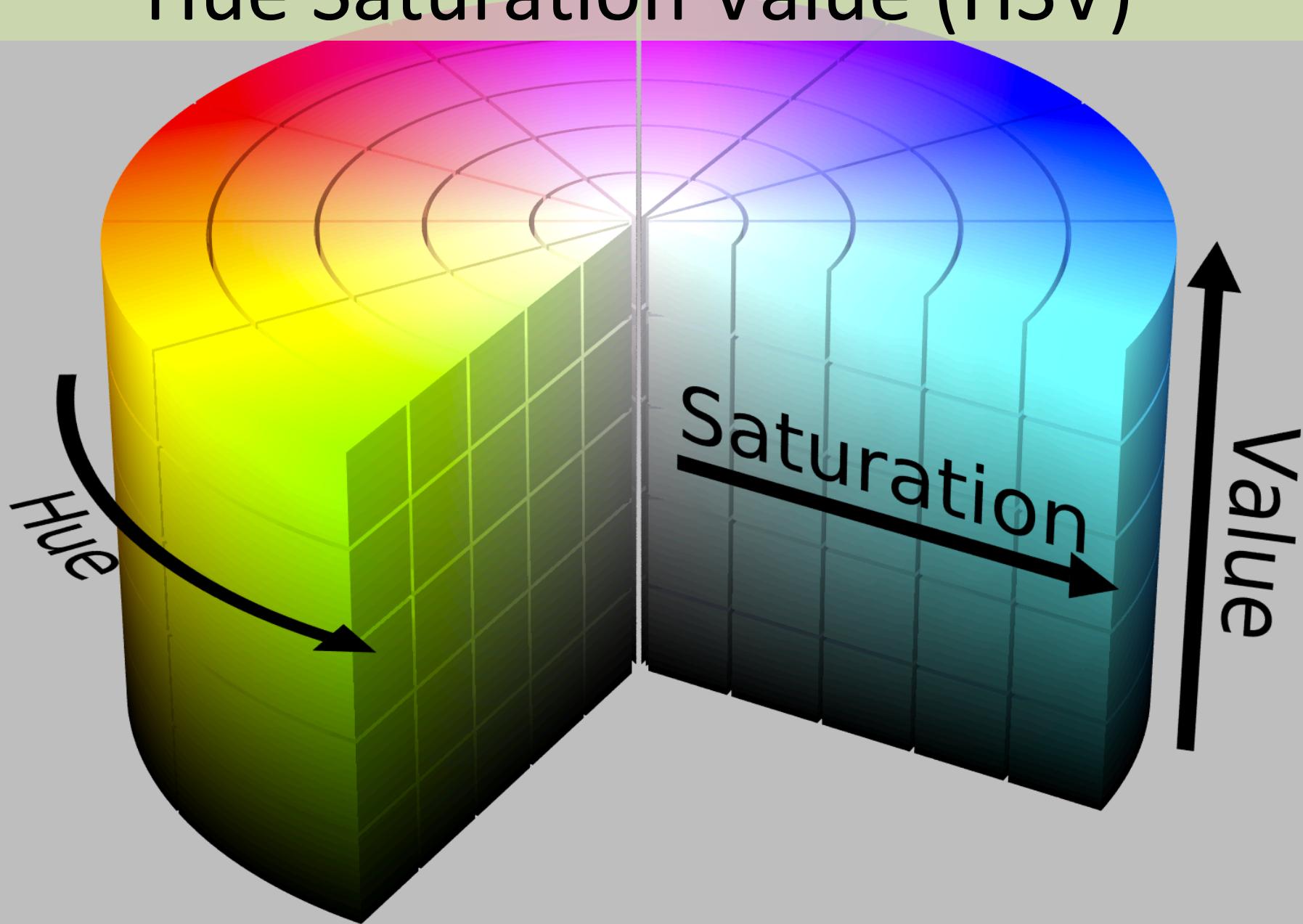
- C library:**
- Containers:**
 - <array>
 - <deque>
 - <forward_list>
 - <list>
 - <map>
 - <queue>
 - <set>
 - <stack>
 - <unordered_map>
 - <unordered_set>
 - <vector>
- Input/Output:**
- Multi-threading:**
- Other:**

<vector>

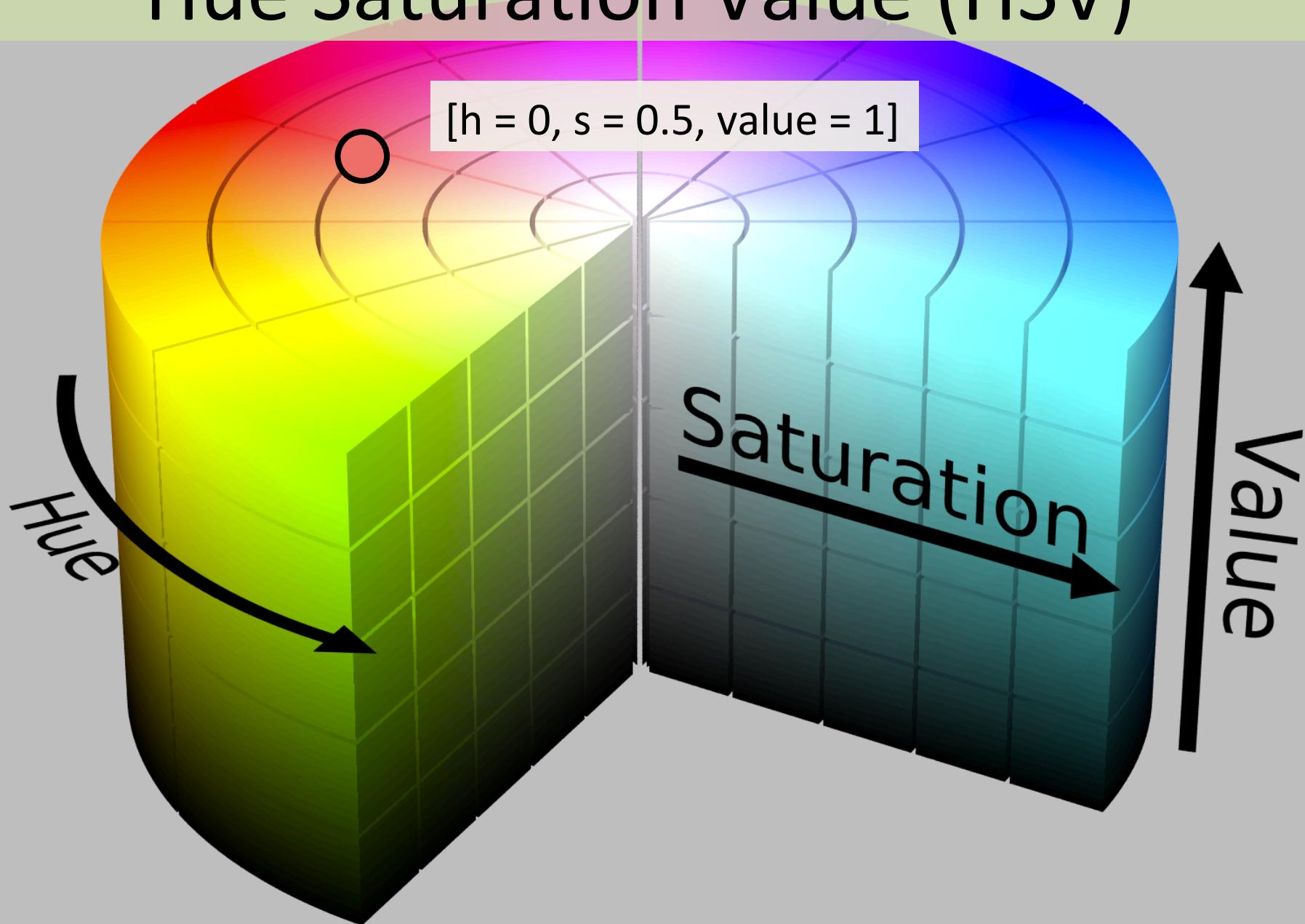
vector
vector<bool>

vector

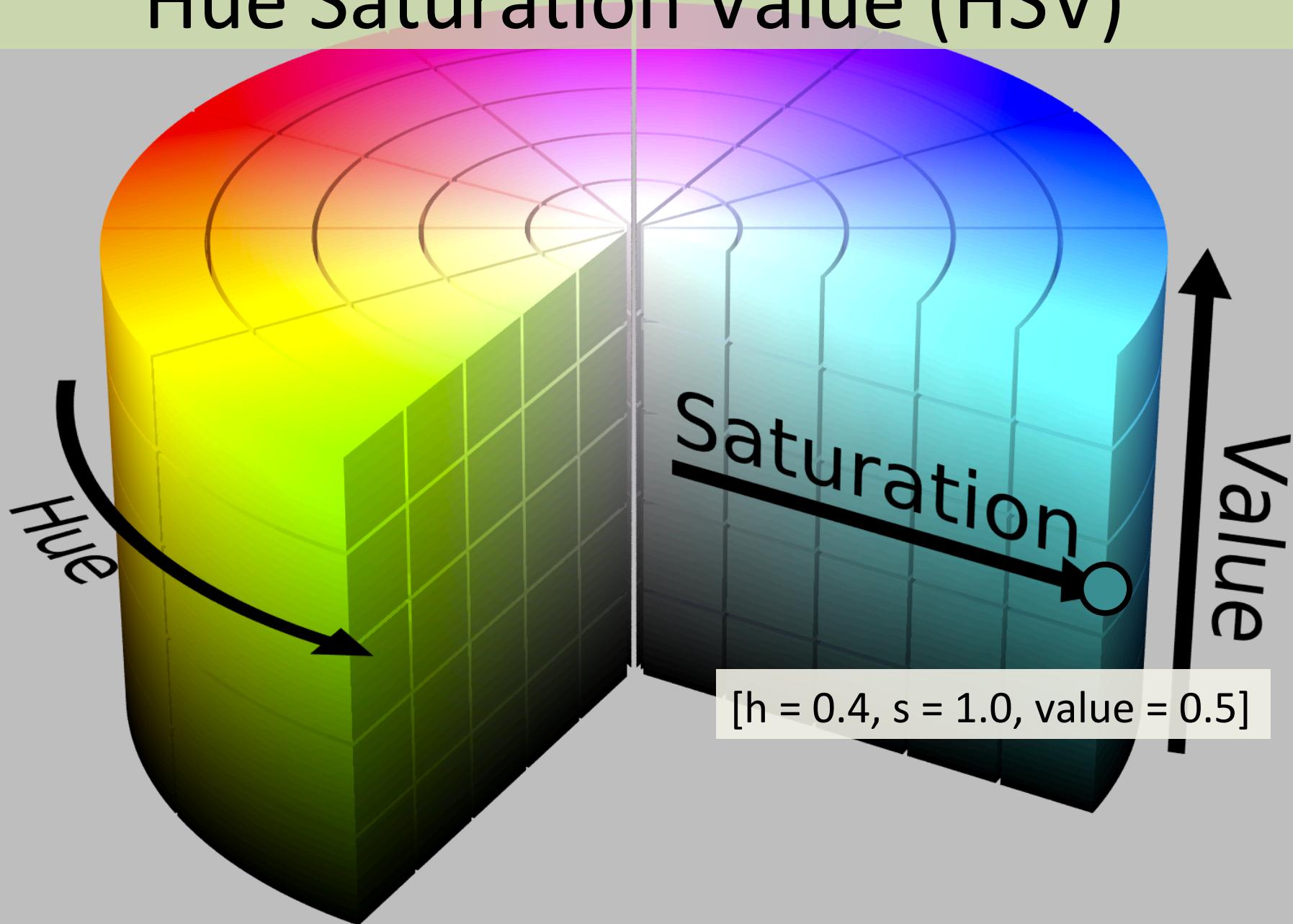
Hue Saturation Value (HSV)



Hue Saturation Value (HSV)



Hue Saturation Value (HSV)



To the Code!

Today's Goal

1. See “real” C++
2. Be inspired

