

# Functional Lighting

Bruce Steinberg

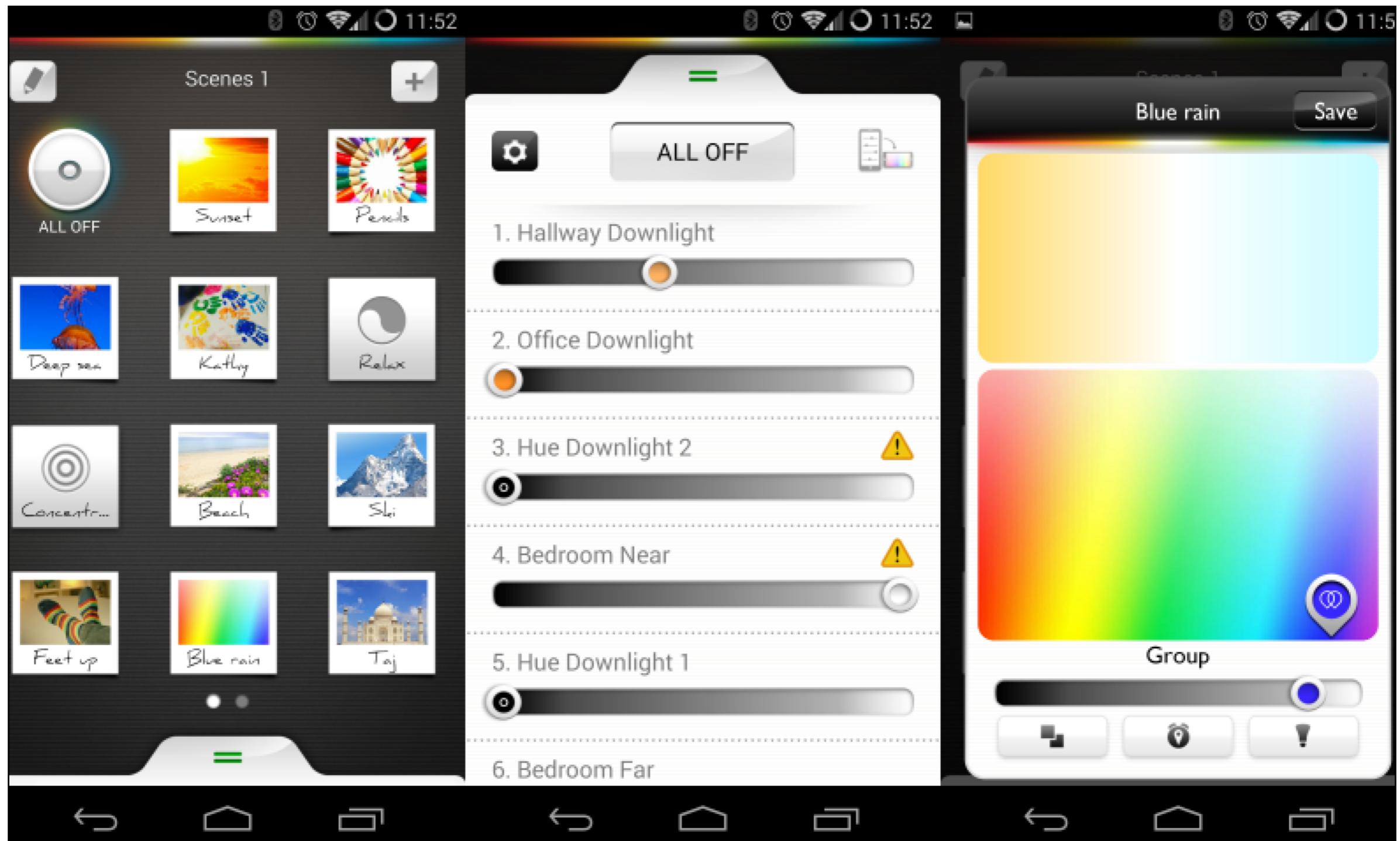
# Philips Hue



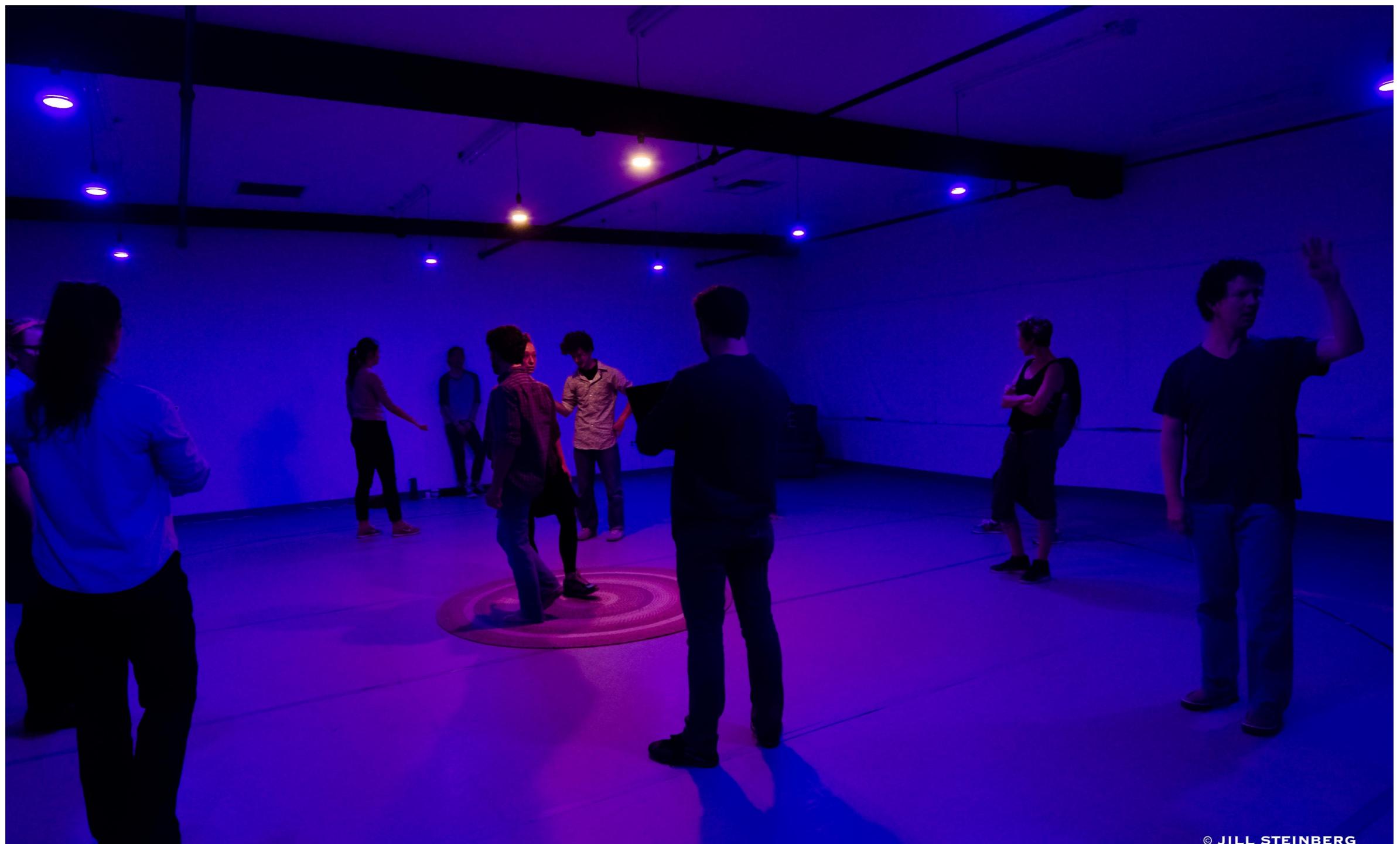
# How Most People Use Hues



# The Official App



# How I Use Them



© JILL STEINBERG

# My Program (Control)

Simple Hue Control

Select Lights to Cue

LX 1    LX 2    LX 3    LX 4    LX 5    LX 6    LX 7    LX 8  
 LX 9    LX 10    LX 11    LX 12    LX 13    LX 14    LX 15    LX 16

[Clear](#)   [Select All](#)

On or Off?  On  Off

Intensity?

Hue?

Saturation?

[Save](#)   [Set](#)

# Display & Cues

Channel Table							
LX 1	LX 2	LX 3	LX 4	LX 5	LX 6	LX 7	LX 8
On?:	On?:	On?:	On?:	On?:	On?:	On?:	On?:
Bri:	Bri:	Bri:	Bri:	Bri:	Bri:	Bri:	Bri:
Hue:	Hue:	Hue:	Hue:	Hue:	Hue:	Hue:	Hue:
Sat:	Sat:	Sat:	Sat:	Sat:	Sat:	Sat:	Sat:
LX 9	LX 10	LX 11	LX 12	LX 13	LX 14	LX 15	LX 16
On?:	On?:	On?:	On?:	On?:	On?:	On?:	On?:
Bri:	Bri:	Bri:	Bri:	Bri:	Bri:	Bri:	Bri:
Hue:	Hue:	Hue:	Hue:	Hue:	Hue:	Hue:	Hue:
Sat:	Sat:	Sat:	Sat:	Sat:	Sat:	Sat:	Sat:

Main Cue List

Cues: 1. Preshow - 5s

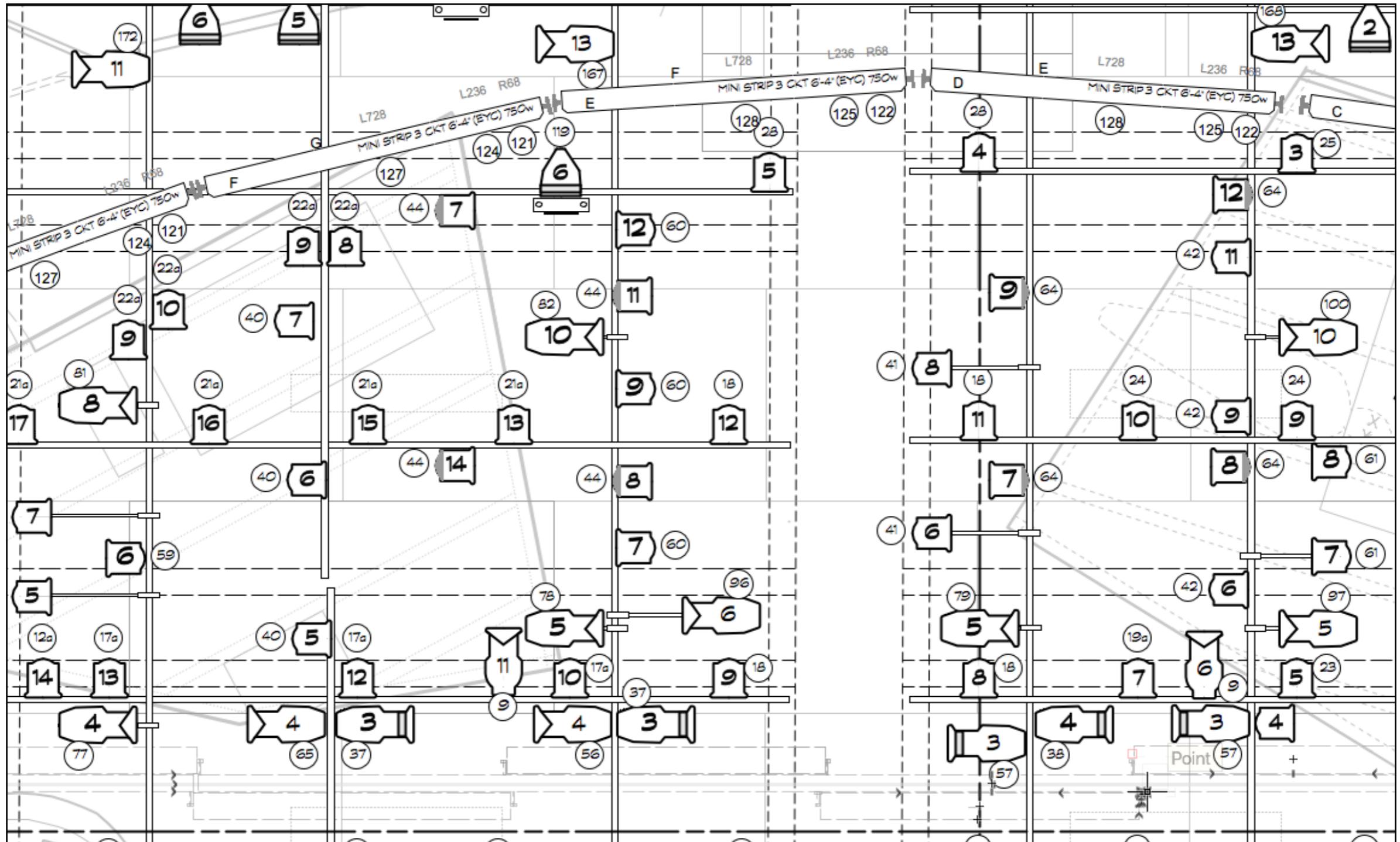
**Delete**      **Restore**

**Right, Left, Right, Left**

# Initial Impulse



# Technical Documents



# Programming

ColorSrc PAR		Color			Beam
Ch		Red	Green	Blue	Shutter Strobe
51	40	:G L200	:G L200	:G L200	
52	40	:G L200	:G L200	:G L200	
53	40	:G L200	:G L200	:G L200	
54	40	:G L200	:G L200	:G L200	
55	40	:G L200	:G L200	:G L200	
56	40	:G L200	:G L200	:G L200	
57	40	:G L200	:G L200	:G L200	
58	40	:G L200	:G L200	:G L200	
59					
60					

ColorSrc PAR		Color			Beam
Ch		Red	Green	Blue	Shutter Strobe
61	80	:G R5	:G R5	:G R5	
62	80	:G R5	:G R5	:G R5	
63	80	:G R5	:G R5	:G R5	
64	80	:G R5	:G R5	:G R5	
65	80	:G R5	:G R5	:G R5	
66	80	:G R5	:G R5	:G R5	
67	80	:G R5	:G R5	:G R5	

# Expression



# Hue Bridge



# Hue API

Get Lighting Attributes & State:

<bridge ip address>/api/<username>/lights/<id>

**Method:** GET

**Reply:**

```
"state": {  
    "hue": 50000,  
    "on": true,  
    "effect": "none",  
    "alert": "none",  
    "bri": 200,  
    "sat": 200,  
    "ct": 500,  
    "xy": [0.5, 0.5],  
    "reachable": true,  
    "colormode": "hs"  
}
```

# Hue API

Set Lighting State

/api/<username>/lights/<id>/state

**Method:** PUT

**Body:**

```
{  
    "hue": 50000,  
    "on": true,  
    "bri": 200,  
    "sat": 200,  
}
```

**Reply:**

```
{"success": {"/lights/1/state/bri": 200}},  
{"success": {"/lights/1/state/on": true}},  
{"success": {"/lights/1/state/hue": 50000}},  
{"success": {"/lights/1/state/sat": 200}}
```

# Simple Hue Control

- **Sets Lights, Saves Cues, Restores Cues**
- No Main Event Loop. Each Command to the Bridge is discrete.
- The GUI buttons fire off procedures that set the lighting state, save cues, and restore cues.
- Objects are used to store information about lights and cues.

# Light, Patch, Cue, Cue List

- Lights are Objects contained in a Patch Object.
  - “Patching” relates the physical address of the bulb to a conceptual “Channel”.
- Cues are Objects contained in a Cue List Object.
  - The json-value field of a Cue Object contains a hash of the lighting states returned from the bridge.

# Setting Lights

- **get-attributes** takes the settings from the GUI and passes them into the selected Light Objects.
- **set-lights!** takes the selected lights and:
  - Retrieves the bridge status
  - Compares state of the referenced bulb to the Light's state
  - Creates a hash containing the state & whether there are changes
  - Edits the hash to just the changed attributes
  - Converts that hash to json & sends it to the bridge

# Saving Cues

- Pressing “Save”:
  - Grabs the Cue Number, Cue Name, and Time from the GUI
  - Passes the to a new Cue Object
    - Which uses **retrieve-bridge-status** to grab a hash containing current state of all lights

# Restoring Cues

- **restore-cue**
  - Takes the index of the cue from the cue-list-display choice%
  - Iterates over all the light states within the stored bridge state and:
    - Creates a hash for each light containing the stored state and which attributes change
    - Uses the same process as **set-lights!**
    - But uses user defined timing.

# Demo!

# Things I'd Like to Improve

- Implement a Command Line
- Modifying Cues
- Implement Proper Tracking
- A Color Picker
- Optimize the Commands to the Bridge

# Fin

Code:

[github.com/brucehs/simple-hue-control-racket](https://github.com/brucehs/simple-hue-control-racket)

Portfolio:

[www.BruceSteinbergLD.com](http://www.BruceSteinbergLD.com)

Hue Developer Site:

[www.developers.meethue.com](http://www.developers.meethue.com)

My Email:

Bruce@Steinberg.lighting