Functional Lighting

Bruce Steinberg

Hi, I’m Bruce Steinberg and I’m here to talk about Functional Lighting.

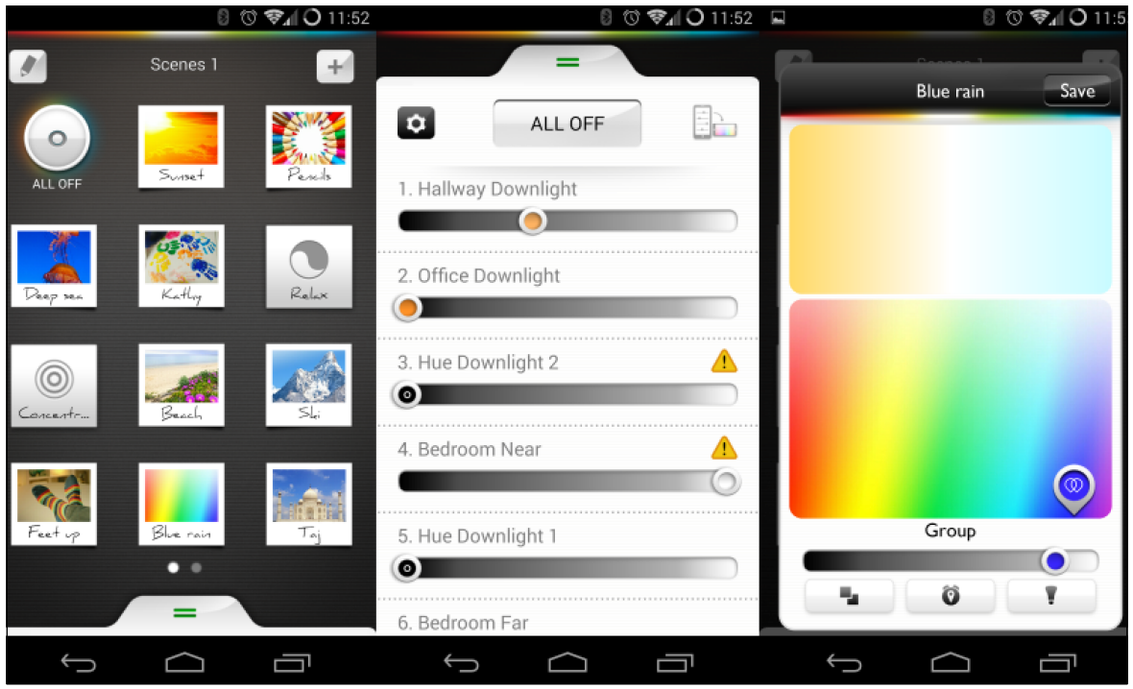
Philips Hue

A couple years back, I found these wonderful Hue light bulbs made by Philips. They can be controlled wireless to change color and intensity. Now, when one designs in an untraditional space—such as a gallery—one often has household lighting, maybe track lighting. In order to change the lights, one has to physically go to each, or with track, deal with sliders and grouped lights. With Hues, I could just put the lights up there and have individual control like I do in a theatre.

How Most People Use Hues

However, they were designed for applications like this. Making your living room super cool.

The Official App

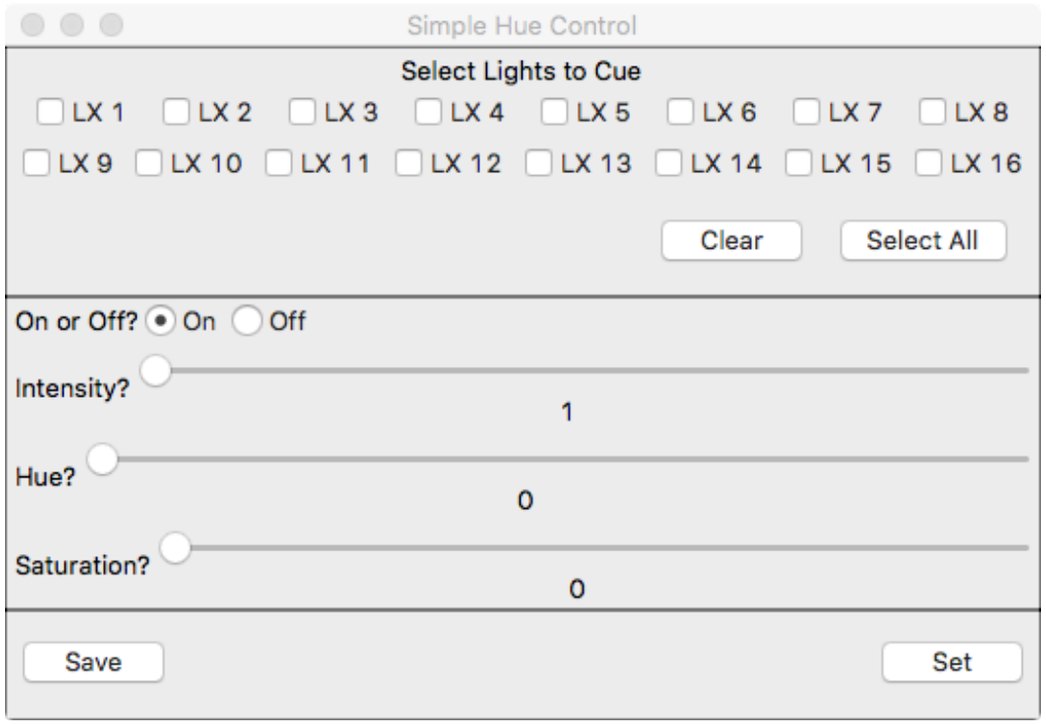


So the apps provided look like this. Imprecise color control, no control over timing, balky sliders.

How I Use Them

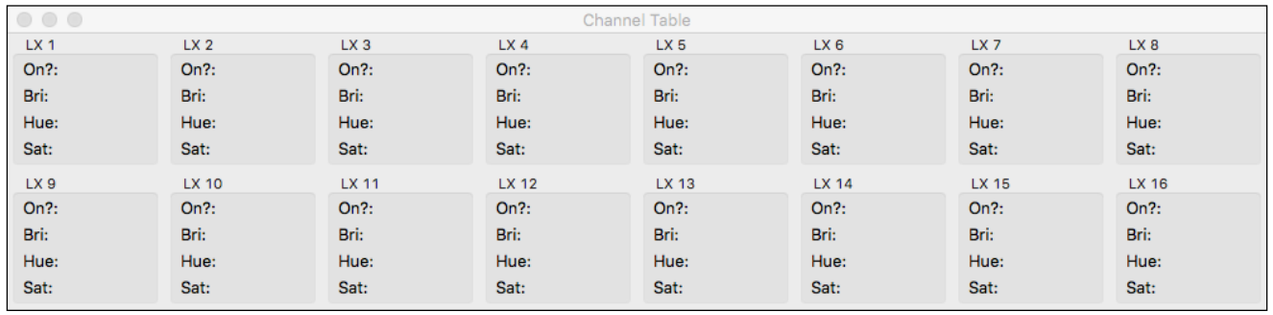
I wanted to do something more like this. At first I used AppleScripts executing curl commands, but it was a slow process.

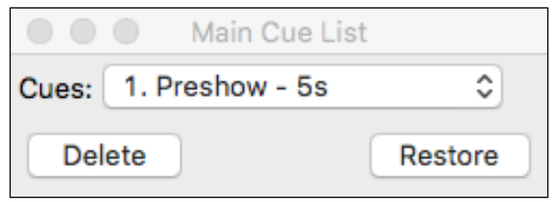
My Program (Control)



So I started making my own program in Racket, Simple Hue Control. A compact interface window that allows precise control over attributes.

Display & Cues





A screen that displays the state of each light, conceptualized as a “Channel” (something I will get back to later). And a window to run “cues”, specific lighting states that can happen over specified times.

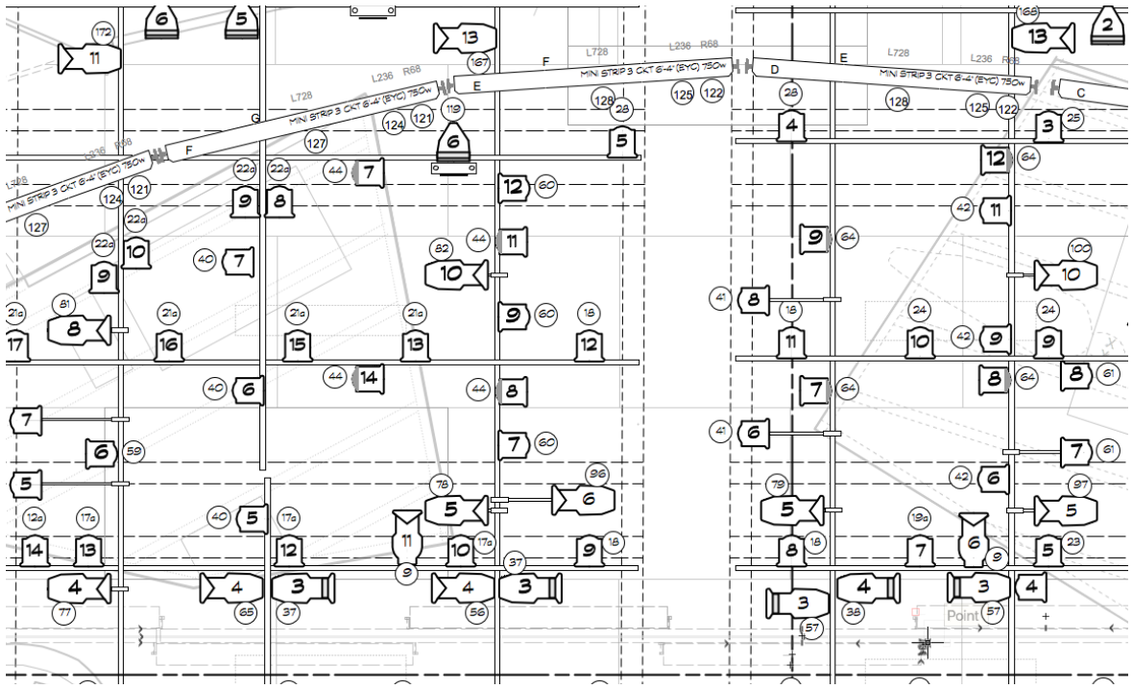
Right, Left, Right, Left

Now to take a step back and talk about the art of lighting design. One thing I love about it is consisting switching between sides of my brain. I'm going to take you on a *very* abbreviated walk thru of my process for a play called "Wonder". It was a new play about the end of history, people switching bodies, house plants turning into a jungle, and being stuck in the memory of the 1968 World's Fair.

Initial Impulse

After reading the script, I was especially interested in the shadows of things. We weren’t going to the actual World’s Fair, but a disassociated memory of it. This is one of the images I shared with the rest of the team.

Technical Documents



Then, in order to actually implement the design, I had to create a technical drafting of where the electricians should put the lights.

Programming

After that, we got into the theatre and I controlled the lights through a programming interface like this one.

Expression

And finally the expression of that idea on stage.

Aspects of Lighting Programming

Going back to the programming part, much of it grew out of idioms used to instruct burly men to move giant levers, sometimes using brooms. There's a concept called tracking, which implies a light stays at a specified intensity unless it is explicitly told to change. If you later go back and modify that intensity, it will not only change in the specific cue (or state), but in every cue following until it runs into another previously programmed instruction. This stems from the way the electricians would operate their boards: they would only move levers that they had been instructed to move. The others would remain static.

Most lighting program languages are far from Turing complete. But they definitely are imperative. The programmer sends commands telling the equipment what to do. Typical commands might be: “Channel 5 at 70%”, or “Group 7 at Color Palette 20”, or “Cue 6.2 Time 7.5”

Imperative & Fast

{Implicit} Number [Key]

- {Channel} 5 [at] 7{0%}
- [Group] 7 [at] [Color Palette] 20
- [Cue] 6.2 [Time] 7.5

  Or Even:

- {Cue 6.2} [Time] 7.5

They also have to be FAST. There's never as much time in the theatre as you want there to be. And a lot needs to happen. A partial list: the performers movement must be modified from what it was in the (usually smaller) rehearsal room, set changes must be practiced, lights must be set, and, most importantly, the performers must have time to organically grow into their roles while everything else is happening around them. So, the last thing a designer wants is for everyone to wait on their programming. Thus we get computers with specialized keyboards and the shortest possible syntax for inputing commands.

You can see here how a lighting board uses implicit commands to speed up programming. When you type a digit, it automatically puts the [Channel] command in front of it. If you want to modify the time of the active cue, one simply has to type the [Time] button and the board places the active Cue Number before it.

Verbal

There is a distinction between the designer and the programmer of the show. The designer wants to keep their eyes on stage as much as possible and looking at a screen gets in the way of this. Yet, in many cases, the designer also wants to be involved in the low level commands. They don’t want to simply say: “Make the lights on stage right brighter”. Rather, it’s make these specific lights this much brighter: “Channel 10 thru 12 and 18 thru 20 at 60 percent”. So, sometimes the designer is literally dictating the programming to another person who types it into the computer. This means the language must be optimized for verbal communication.

Hue Bridge

Here’s the little box that makes the magic happen. It connects to a wireless router and has a simple RESTful interface. It then uses the ZigBee Light Link protocol to talk to the individual bulbs, which create a mesh network to pass the commands along.

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"
  }
```


A little bit about the Hue A-P-I. One thing Simple Hue Control needs to know is the current state of the bulbs. This Get command returns far more information then we need (I've actually omitted some). Right now, Simple Hue Control only allows the hue/saturation colormode, so I've underlined the relevant attributes.

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}}
```

And here’s an example of the Put command I use to modify the lighting state. Again, relatively straight forward. There is a performance penalty for each attribute you set, so Simple Hue Control compares the light state you want with the current state of the light and only includes the attributes that have changed in the JSON.

Demo!

And now for a 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

As you can see, Simple Hue Control is pretty simple. There's a lot that I would love to do. The biggest (and probably the most challenging) is implementing a command line. While Simple Hue Control is faster than the official app, programming speed would increased dramatically if the programmer didn't have to take their fingers off the keyboard.

At the moment, when you record a cue, it is set in stone. To change it, you need to delete it and then re-record it. Updating it when it's onstage would make things simpler. Longer term, it would be useful to be able to modify cues that are not active. (This is called Blind mode on most lighting consoles).

As I mentioned before, Tracking is very useful. The programmer only needs to change an attribute once and it propagates through the cue list. Right now, each cue knows nothing about the others.

Obviously a color picker would be a useful option for specifying colors without having to remember what the corresponding 16-bit number is.

We didn't notice it today, because I only have two lights, but when many are changing, you can see the delay. They sort of update in spurts. I want to explore dynamically using other A-P-I commands to make the fades more seamless.

Fin

Code:
github.com/brucehs/simple-hue-control-racket

Portfolio:
www.BruceSteinbergLD.com

Hue Developer Site:
www.developers.meethue.com

My Email:
Bruce@Steinberg.lighting

That’s it! Any questions?