Light Show: Intuitive Control for Complex Lighting Systems

Isaac Salier-Hellendag
UC Berkeley iSchool
102 South Hall
Berkeley, CA 94720 USA
dish@ischool.berkeley.edu

CHI 2008, April 5 – April 10, 2008, Florence, Italy
ACM 1-xxxxxxxxxxxxxxxxx.

Abstract
The aim of this project is to create a simple, intuitive interface for designing and controlling a multicolored light show.

Keywords
Light, control board, lighting design, color wheel, infrared, Wii remote, rear projection, remote control, color picker, color dropper, palette

ACM Classification Keywords
H5.2. User Interfaces: Input devices and strategies.

Introduction
To use a lighting system comprised of multiple arrays of multicolored lights, a user generally needs a high level of expertise. A typical light control console includes multiple dimmer sliders, used to manage the intensity of the lights. By consolidating light intensity controls at the console, the lighting designer can produce color combinations across multiple sets of lights. [1]

However, the console interface is neither simple nor intuitive for a novice user. Additionally, the prohibitive cost and expertise required for console control are obstacles for users wishing to experiment with muticolored light displays. This project was created to

explore an alternative interface concept, namely a simpler, more intuitive and compelling console for lighting control.

Core Design Decisions

The following key design decisions served as the framework for the implementation of the concept:

- A single-surface interface provides maximum simplicity. For example, a painter's palette contains all of the colors he or she will require for painting. Similarly, this interface should provide all of the possible color values that a user might apply to a set of lights.

- To choose a color, the user should not need to perform any color combination calculations to identify the correct color value. Simple visual selection of a color is ideal.

Following these principles, the basic interface required:

- a table or easel with an image of a color wheel or other color continuum

- a selection tool to choose a color – for instance, tokens placed on the interface surface

- a method for matching a selected color to a selected set of lights

Design Process

The concept of direct interaction with a single surface bears a resemblance to a touchscreen. With that in mind, it was logical to implement the design in a manner similar to a touchscreen device. Rather than use one's fingers to interact with the screen, however, the selection tool is used.

Following the assumption that users familiar with digital photo manipulation or graphic design would be comfortable making color selections with a color wheel, it was assumed that users would also be comfortable with the concept of the "color dropper". This tool is used in software applications to perform color selections. A physical version of the dropper mechanism could thus be created to serve as an intuitive selection tool for this interface.

A crucial step in the design process was to ensure appropriate feedback upon the selection of a color. In order to provide consistent feedback, a changeable screen display would be necessary. A projection system was a logical choice, as it would allow a projection of the color wheel image upon the interface surface as well as a panel to display immediate feedback during color selection.

Figure 1 shows the implementation of the screen portion of the interface, including the rear projection method.

Figure 2 shows the implementation of the dropper mechanism, and Figure 3 depicts the dropper tool in use with the screen.

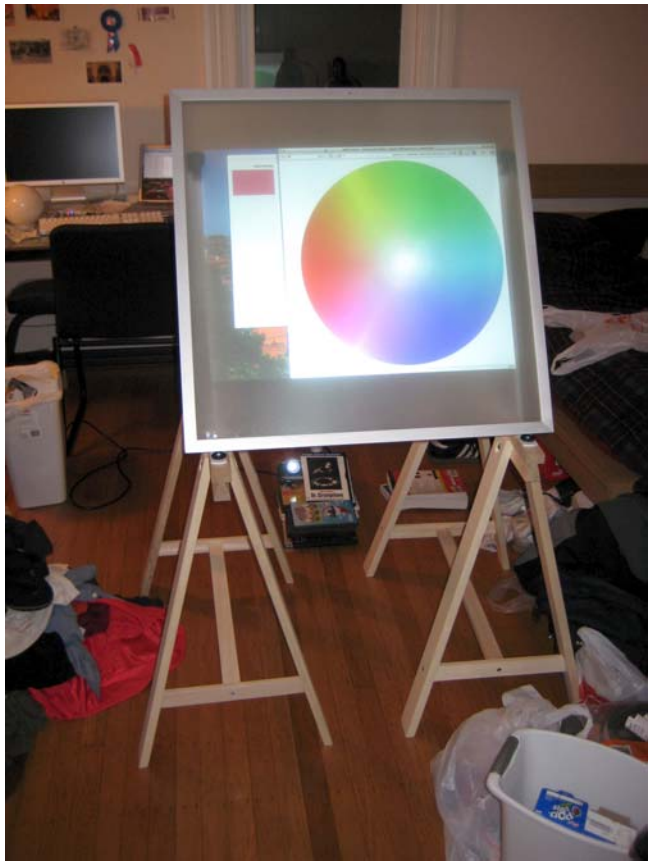

figure 1. The screen portion of the interface, comprised of a pane of frosted glass on a wooden easel, with the color wheel display provided via rear projection.

figure 2. The dropper tool, used for selecting colors on the screen interface (fig. 1). Pushing the button activates color selection – this implementation is discussed later.

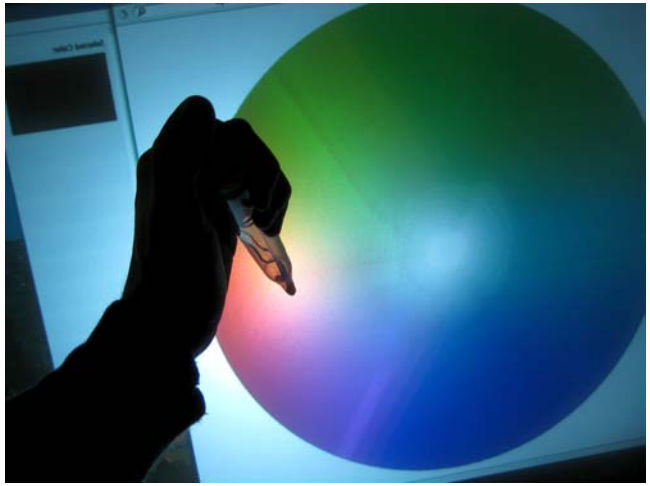

figure 3. The dropper tool in use, selecting a color. Note the button being pushed to signify a color choice.

Following the selection of a color, the next step is to apply that color to the desired set of lights. In an ideal implementation, a user could use the dropper tool to point in the direction of the desired lights and push the dropper button again (similar to a remote control), thereby assigning that set of lights with the chosen color. The lights would then shift to the color selected. In this manner, the user and the interface would not need to be directly connected to the lights themselves. Rather, remote communication would allow the lights to change, essentially turning the physical space of the venue into a canvas for the user to manipulate.

Additionally, the user would have the option of creating patterns of color by dragging the dropper tool across the color wheel display. By selecting a sequence of colors in this manner, the user could pre-program patterns for later use. He or she could then arrange for sets of lights to loop through the selected colors, possibly to synchronize the lights to music or cues in a stage performance.

Implementation Specifics

The projection screen was built from the following:

- A frosted glass cabinet door (Ikea)
- Two wooden trestles for support (Ikea)
- Two shelf braces to prop the glass up at an angle
- A Dell projector, providing rear projection

The dropper tool was built from the following:

- A modified turkey baster
- Two 3V watch batteries
- 3 infrared LEDs plus one red LED as a pilot light
- a button switch

To select a color, the user places the dropper tool onto the screen on the selected color and presses the button. It is important to note that no actual color detection is performed by the dropper tool itself. The button turns on the 3 infrared LEDs, which pass infrared light through the glass screen. On the other side of the screen sitting above the projector was a Wii remote. The Wii remote is a Bluetooth device with an infrared camera on the front. The camera is capable of identifying up to four points of infrared light and translating those points into x-y values – the dropper tool provides enough infrared light to be detected as a single point.

When the Wii remote receives infrared input, it passes the location data to a Cocoa application created for the purposes of this project. Based on an open-source Wii remote Cocoa framework [2], the application translates the x-y value into an RGB value that matches the color selected on the screen, assuming correct calibration between the screen and the Wii remote.

When the RGB value is calculated, the application displays the resulting color in a “Selected Color” display next to the color wheel. This provides immediate feedback to the user, indicating that their color choice has indeed been recorded.

In the prototype created for presentation at the iSchool Tangible User Interfaces Open House in December 2007, the RGB value was written via serial port to an Arduino board connected to red, green, and blue LED’s.

An Arduino program modified for this project changed the brightness of each LED to match the appropriate RGB value selected by the user. In this way, the user's color selection directly changed the light display. However, the remote light control feature was not yet implemented.

Partial Solutions and Future Work

The prototype created for the December open house succeeded as an accessible, intuitive design for lighting control, but it did not contain certain features that were partially implemented or could be added in future versions of the interface:

- Forming/programming color patterns by dragging the dropper tool along the screen
- Saving colors and patterns to a custom palette for simple re-use
- Dimming and brightening a single color
- Display of the exact RGB value on-screen to write down for later use outside the scope of this interface
- Light set synchronization
- Synchronization to music
- Pointing first at the target light set, then at a color to change the color displayed by the set
- Allowing multiple users to use multiple droppers – collaborative lighting system manipulation
- Controlling physical movement of light sets
- Using pre-programmed gestures to produce certain behaviors in the lighting displayed
- Controlling other aspects of a light/music/theatrical production, including special effects, fog, strobes

The most vital step to perfecting this interface is calibration. While the projector display and Wii remote were relatively well-matched, users were not able to be extremely specific with the lights they desired. This is because the scope of the Wii remote's infrared camera was a different size than the color wheel on the screen, and because the center point for the Wii remote was difficult to align perfectly with the center of the color wheel. In the future, designing a method for creating a quick, simple calibration procedure would be very helpful in providing greater precision for the user's color selections.

Notes

The code written for the software portion of this project is freely available online.

- LightShow Cocoa application [3]
- WiiRemote framework (for gathering data from the Wii remote infrared camera) [2]
- AMSerialTest (for writing to the serial port) [4]
- Modified Arduino code for LED manipulation [5]

Acknowledgements

Many thanks to my Tangible User Interfaces course instructors and classmates, and Ms. Tina Lin, whose knowledge and experience helped tremendously in the production of this prototype.

Citations and References

[1]Lighting Control Console.
http://en.wikipedia.org/wiki/Lighting_control_console

[2]SourceForge.net: DarwiinRemote
http://sourceforge.net/projects/darwiinremote/

[3]LightShow Application Code
http://www.salier-hellendag.com/lightshow/

[4]harmless Cocoa – http://harmless.de/cocoa.php

[5]LightShow Arduino Code – http://www.salier-hellendag.com/lightshow