

LightFactory

By: Rob Halliday



Figure 1: Startup screen

Interesting thought: Once we got past the first generation of memory lighting control, which required a massive investment in computer hardware, each new generation of innovation in the field has come from a small player, an upstart company with a bright idea, a point to prove, and none of the burden of history that comes with an existing range and user base. To pick just two examples, ETC pioneered the microprocessor-based lighting control, and just about every console now on the market owes a debt to the Wholehog I and II.

Physically, though, these products still resembled “traditional” lighting consoles to some degree—a box with the familiar interface of faders and wheel and buttons on the front, and connections on the back to talk to the dimmers and lights (usually via DMX, a great enabler for a generation of new control manufacturers). Of course, internally they were just computers running software, but you didn’t really care, except when a free update fixed bugs or added features.

Some companies tried to skip the expensive hardware, with software-only solutions running on standard computers, but, while they offered low-cost access to lighting control, they never really became industry standards, in part because established users rejected the notion of controlling lights using just a keyboard and mouse.

Now, though, we seem to have a new wave of software control solutions—and, in many cases, it is this sector that is innovating in the field. Why now? Partly because the interface from PC to lights has become easier through the power of the network. Partly because new users are familiar with windowing interfaces; they want their controls to work that way rather than through arcane keystrokes. Partly because the software can now run on innovative, yet readily available, hardware that exceeds what specialist lighting manufacturers have been able to dream up (or, perhaps, afford to implement): That big, multi-touch screen on Apple’s iPad is a delightful experience begging for a lighting app to take advantage of it. All of this means that the barrier to creating a lighting controller now is time and program-

ming ability, not the direct—and enormous—expense of getting hardware designed and built.

Which is exactly the background of this month’s subject: LightFactory. Its creator, Martin Searancke, decided—foolishly, some might say!—to take the money he’d made from a non-entertainment company he’d founded, and use it to work full-time at his hobby, stage lighting. In New Zealand, designers often program their own consoles, so he set out to buy one for himself—quickly discovering just how much a “full-featured” console would cost. Instead, with the independent spirit that seems to be part of antipodean culture, he decided to make his own.

The result is a fascinating product—occasionally quirky, sometimes untraditional, but packed full of options that just make you go, “Mmm, that’s so obvious; why has no one done that before?” Its origins as a personal project are clear in senses both bad (little annoyances like windows that occasionally don’t refresh their content properly) and good (while using the product on shows, its creator thinks of useful new features, then goes ahead and implements them). The question is, does the good outweigh the bad?

Hardware requirements

LightFactory is a PC-only package, citing as its minimum requirements a Pentium III processor, 256Mb of RAM, 300Mb of free disc space, video output at 1024 x 768 or higher, and Windows XP, Vista, or 7; the software will support dual monitors if available. It also ran fine on a Mac via VM Fusion; we tested various builds of V2.4. The software is priced according to the number of DMX outputs you need, from half a universe (256 DMX slots) at \$199 up to the full 64 universes at \$4,999. The software works without a license, but with no data output, except to visualization software (from which it will also receive selection and autofocus information). But—the first of the why-has-no-one-done-this moments—if you’re doing a show, you can rent instead of buy—\$39 gets 12 hours of four universes—with options up to \$259, giving you the same outputs for five days. Only the time the software is actually running counts.

Getting data out of LightFactory is most easily accomplished over a network—it will output the ArtNet, ENTTEC DMX over Ethernet, ZeroWire DMX, or Pathway PathPort protocols, for which a wide variety of network-to-DMX interfaces exist; RDM support is coming soon. Alternatively, Cooper Controls, which distributes LightFactory, makes a variety of single-universe Ethernet-to-DMX and USB-to-DMX interfaces, and the software will work with similar interfaces from others. Cooper also now makes a playback wing (10 faders and go/back keys) and a programming wing (keyboard and three rotary encoders) for the console, but we are not looking at those here.

Software

Downloaded from the LightFactory website, the software installed and ran without any hiccups. If you don't have a license code (which you can either enter into the software, or install in certain of the external DMX convertors so that the registration travels with that hardware whichever computer you connect it to), you get plenty of warning that no data will be output. The startup screen also reveals one interesting tidbit when it says, "Starting database connection" [Figure 1]. It turns out that LightFactory is using an SQL database as its back end for storing cue data. If you don't care, you don't need to care, but it does open the possibility of other software connecting to LF's data—imagine Lightwright getting patch changes automatically, or new cues magically appearing in a designer's Excel spreadsheet. One LightFactory user has already started offering utilities to assist with some common tasks—these are at lfutilities.co.uk.

Start-up is quick, and LightFactory always resumes exactly where it left off, same show loaded, channels sitting at their last levels. The interface consists of a "control bar" of large buttons that can be locked to the top of the Windows task bar [Figure 2.]. The left-hand button, "Options," is the equivalent of the standard "File" menu, with options for loading or saving shows, changing preferences, or exiting LF. The next nine buttons, each of which has a function key shortcut, open separate windows for different functions—a channel display window, a groups list, a cue list editor, the patch, and so on. The last button, "Extras," is the "ran-out-of-buttons" button, letting you get at other windows.

The windows that these buttons open are standard Windows, and so can be positioned and resized to suit the operator's taste, minimized, and so on—though bigger screens will be helpful for serious users! The window backgrounds have been grayed down in newer versions of LightFactory, and its colorings can be customized, though perhaps a single "dark mode" toggle would be useful [Figure 3].

You can get a good overview of LightFactory's approach, strengths, and weaknesses even through the console's basic features. Patching is a drag-and-drop affair, pulling fixtures from a comprehensive library containing information from both LightFactory and Zero88 (which distributes Lightfactory), and dropping them onto an address—this not as well error-checked as it could be, since you can drop fixtures partially onto other fixtures—then a channel [Figure 4]. The LightFactory patch interface is beautifully clear, the left-hand side showing channels, the right-hand side a grid of DMX addresses. For each channel, LightFactory can store a lot of information—weight, power, phase, color, and more— together with a picture of the fixture that is very useful for quickly finding lights when it appears in the channel display. While it's disappointing that the library is inconsistently incomplete, you can just fill in the blanks directly in the patch if you need to, and it tries to auto-complete anything



Figure 2: Control bar

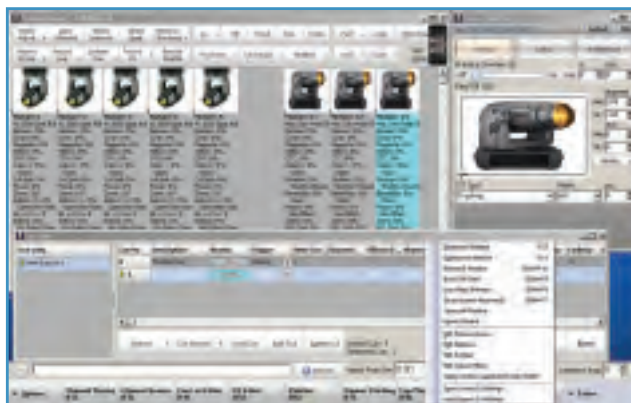


Figure 3: Console Windows

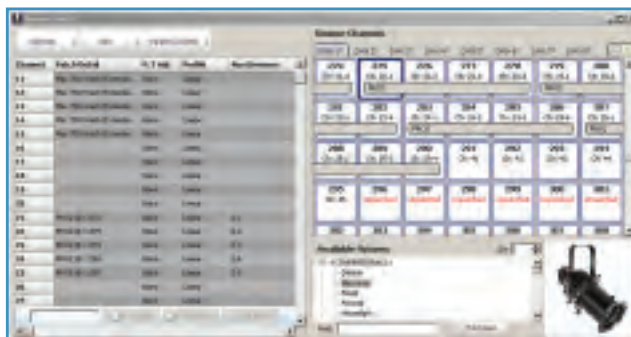


Figure 4: Patch

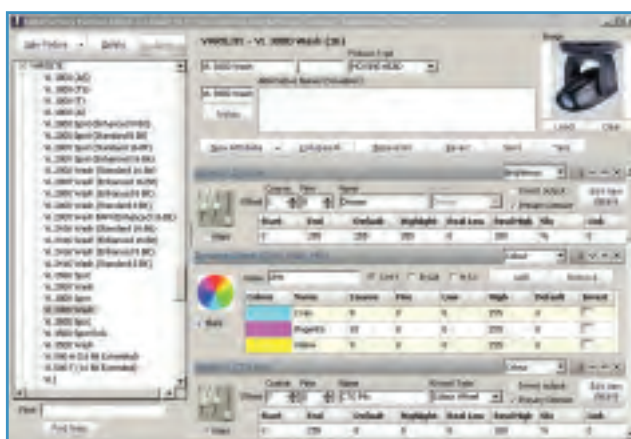


Figure 5: Fixture editor

it already knows. The fixture library is also editable, in a quite user-friendly way [Figure 5]; sadly, you need to do this to customize moving lights with particular gobos or to make your own scrolls.

Examples of real-world useful features start right here—for example, being able to clear part of the patch, which is fabulous when you need to clear just the front-of-house universe each week on tour, or the column that shows DMX addresses as dip-switch setting. More importantly, there is the first of the software's many search boxes: Start typing and it will find matches across any field in the patch, quickly

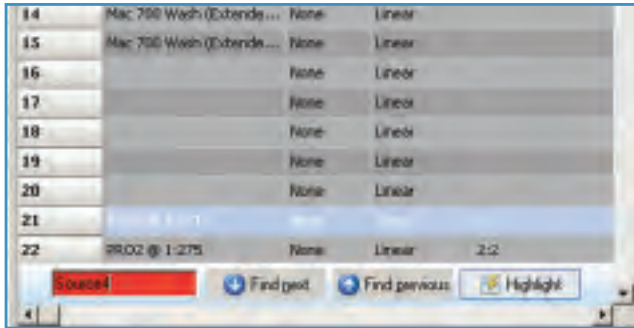


Figure 6: Find in patch



Figure 7: Channel window



Figure 8: Compact



Figure 9: Moving lights

rewarding you for adding extra information either by hand or by importing it from external files by letting you find lights by name or color or position or more [Figure 6].

Patched lights appear in the Channel Display window, this arranged with related buttons for selecting lights and



Figure 10: Character display



Figure 11: Plan display

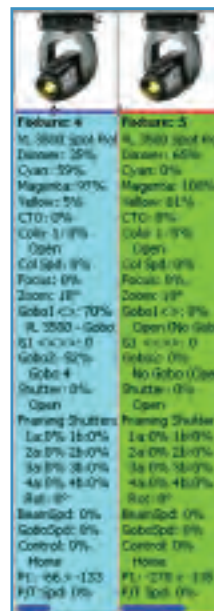


Figure 12: Selected channels

recording cues, groups, or palettes across the top [Figure 7]. Channels can be displayed in a variety of ways—as various grids showing different amounts of information (just channel number and level [Figure 8]; all attributes for moving lights [Figure 9], both of these showing a picture of the fixture type, if available; a black-background “character” display more reminiscent of older consoles [Figure 10]; or in a layout view that lets you draw a mini-plot with simple symbols that you can arrange however you like. This view even lets you place mini-representations of the LED matrices that LF supports, and put a background image behind the whole thing, letting you to create a very

powerful overview of the rig [Figure 11].

In some views, the quite-fine text can be a little hard to read, but the software adds graphical indicators for some functions: “active” channels—lights that are on—get a green background, level shows up as a bar chart, color appears

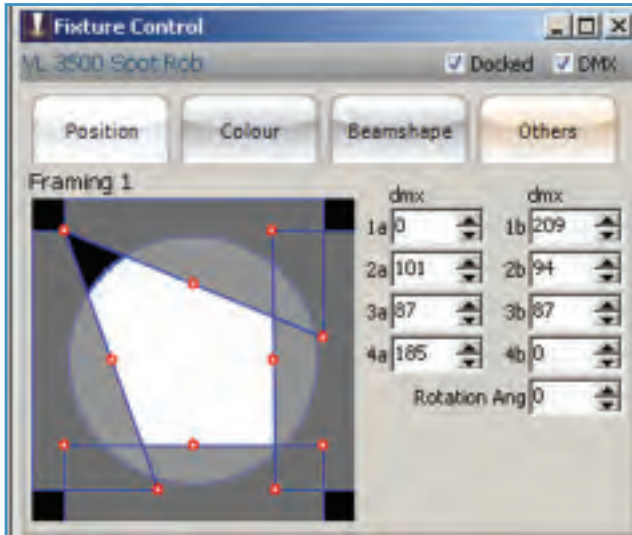


Figure 13: Framing

as a colored band, position as an “x” attempting to indicate roughly where a light is pointing [Figure 12]. The ease of switching views means you can pick the best one for any given activity.



Figure 14: Fixture control

There are several options for selecting lights (whether a new selection adds to or replaces the previous), then controlling a light is done with your mouse or trackpad. LightFactory offers some elegant control interfaces to overcome what might be a limited, one-parameter-at-a-time control system, most clearly demonstrated with a framing spotlight: You get a picture of the shutter assembly, then drag each blade to the position and angle you need—simple [Figure 13]. Click a simple light, and your mouse controls level. (A command line is available for traditionalists, with useful hints; lots of keyboard shortcuts are also offered). Click a moving light, and a “fixture control” box appears, again using a picture of the fixture—though this can distort in rather ugly ways if you make this box an odd size—divided into tabs for separate function types; you get to define these tabs in the fixture library, and they also form the beam parameter filter categories, so, unlike many other consoles, you can arrange these to suit your taste [Figure 14]. Select the position tab, and your mouse is controlling pan/tilt. Select color, and, as in graphics programs, you can pick from a variety of selection tools, whether you prefer dragging across a color palette, picking from a gel library or your own defined show palette, or moving RGB or CMY sliders; LF also has an “advanced color” option that will try to make use of extra colors a light might offer—CTO in a moving light, or colors



Figure 15: Color controls

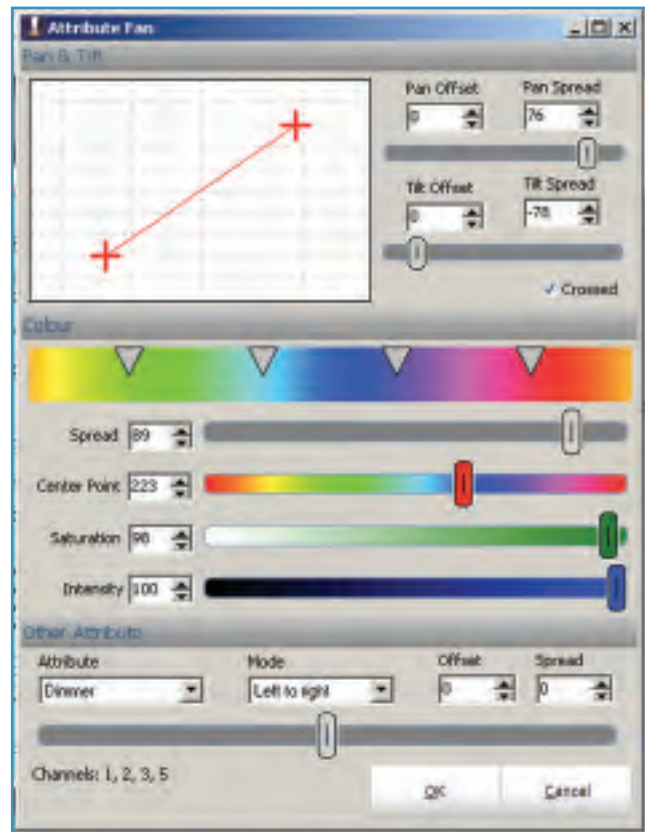


Figure 16: Fanning

beyond RGB in an LED fixture [Figure 15]. It also has the most effective controls I’ve seen for fanning positions or colors across lights [Figure 16].

LightFactory is very polite about offering all kinds of advanced control options for attributes while still letting you talk directly to the light’s functions, if that’s how you prefer to work. For example, it lets you pick a gobo, its mode (“rotate,” say), its direction (“clockwise”) and speed, but, at the same time, shows you, and lets you directly set, the DMX values it is sending to the light and storing in the cues [Figure 17]. What happens if you change light types is a little unpredictable, because the fixture library is a mash-up from two sources; some fixtures contain real-world data (so you’ll

see “15” under zoom), and some don’t (so you’ll see 20% zoom). Switch from one abstract fixture to another, and the beam will be the same size in the new light; switch to a non-abstract fixture, and it won’t. Sadly, there’s nothing obvious to tell you which fixtures have all the data and which don’t while patching.



Figure 17: Gobos

Like all consoles now, LightFactory lets you store referenced information inside cues—preset positions, colors, and the like. LF splits these into two categories: palettes, which contain the same information for every light of each type (so great for colors and gobos; palettes for these can be auto-created as lights are patched), and groups, which can contain different information for every light (useful for positions). Part of the occasional lack of clarity in the interface is that LF sometimes interchanges the words palettes and groups a little too freely. On the other hand, there are more search boxes here, making it easy to find particular groups or palettes in big shows; you can also filter the palette list to just show particular content types [Figure 18]. Palettes can also be assigned, manually or automatically, to shortcut buttons, making them easy to busk with, particularly if you’re using a touch screen, which LF supports [Figure 19].

LightFactory also offers a really interesting new trick when it comes to using groups in cues. As well as being a referenced palette (a position, say), or an easy way of setting a combination of channels quickly (“group 1@50” on other consoles), LF can also store groups into cues and will remember this as a reference [Figure 20]. So, if you make Group 1 contain Channels 1>5@FL, then put that group into Cues 1@50, 2@50, 3@60, 4@FL, you get those channels at those levels. But if you later add Channel 6 to that group, it will automatically appear in those same cues at those same levels. Though the interface for this is rather inelegant, the functionality is incredibly powerful—imagine plotting your front-of-house wash into the show, then later touring to a venue where it need six areas across instead of five.

Plus here’s another one of those real-world things: LightFactory has a power monitor, a window with three meters, one per phase [Figure 21]. As you bring up lights, LF will calculate the current draw based on level and the power and phase information in the patch then display

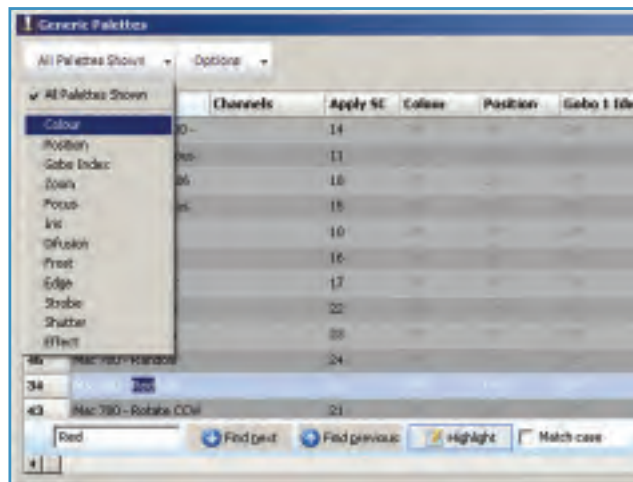


Figure 18: Palette find: filtering



Figure 19: Shortcuts



Figure 20: Groups in cue

that on-screen. Cleverer still: You can set a power limit; as you reach that limit, LF will gently throttle back the grandmaster to keep you from tripping the breakers, figuring that a little dimmer is better than sudden darkness. Sure, it’s not perfect—in particular, it doesn’t yet account for the constant background load of discharge moving lights—but it’s a brilliant use of the information the console already has. Any console manufacturer could have done it, but this one actually has.

There are so many useful touches and features like this that it’s going to be hard to fit them into this review, so perhaps let’s look at some highlights. Some are simple things, just bits of the interface that make it clear what’s going on—



Figure 21: Power meter



Figure 25: Submasters



Figure 22: Record dialogues



Figure 26: Effects

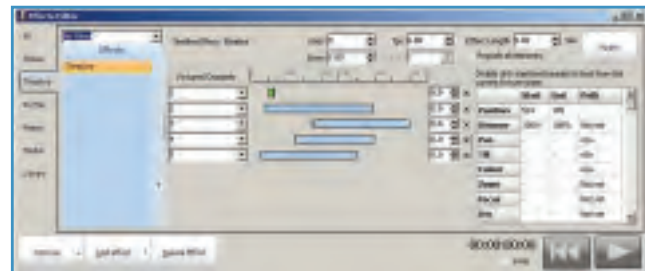


Figure 27: Timeline

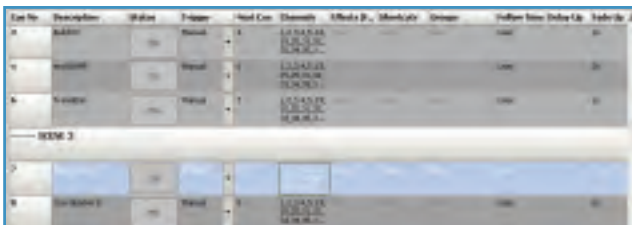


Figure 23: Scene break

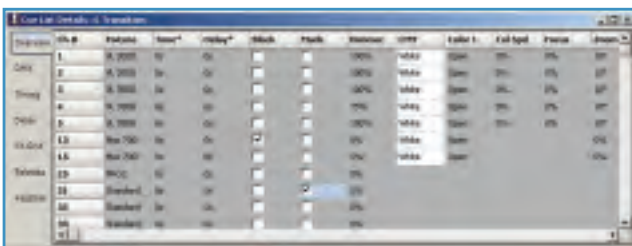


Figure 24: Channel tabs

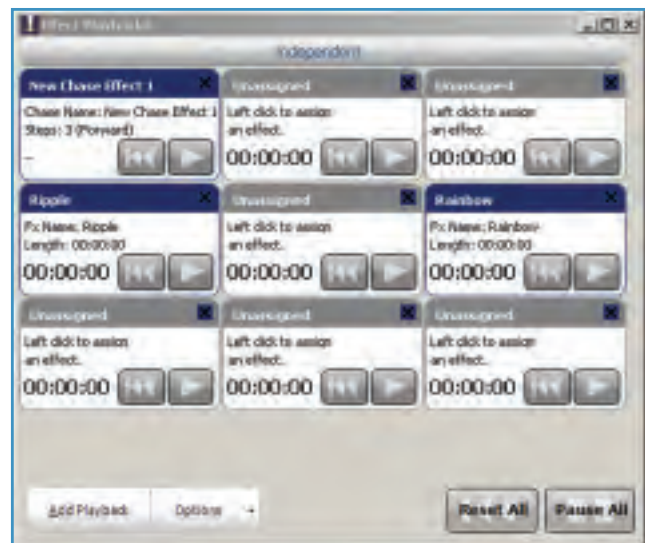


Figure 28: Effects playback

such as a “park” menu that lets you decide whether or not to record parked channels. A record dialogue box that adapts its heading from “record new cue” to “update existing cue,” depending on whether or not the cue number you type already exists or not [Figure 22]. A cue list display that lets you freely re-arrange the order of columns by just dragging, and which has fields for both a short cue description and much longer cue notes—you can write an essay about each cue if you need to. The ability to add and label very visible scene breaks between cues (as suggested in a column in this magazine a while ago, it’s good to know some-

one’s reading!) [Figure 23] A channel-editing screen that has clear flags for lights that are blocked or set to mark, rather than relying on the user to remember particular color codes, and makes switching between levels and times easy and intuitive [Figure 24].

Some are well-thought-out interface touches to make up for the limitations of running a lighting console on screen—the level wheels that abound, perfect if using a tablet computer (oh, for an iPad version; there is software available to

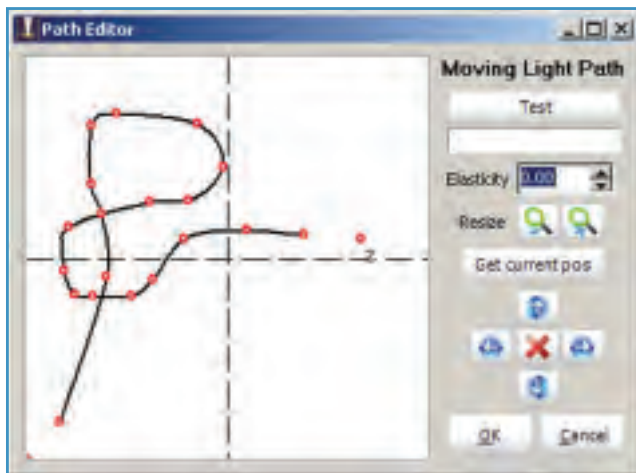
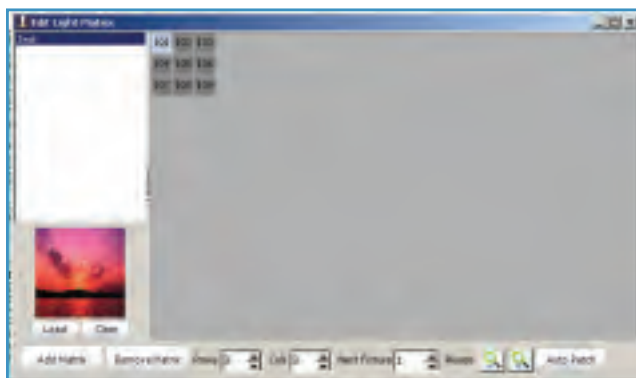


Figure 29: Fade path



30: Matrix

turn a Windows handheld into a remote, but that's not quite the same), or the ability to lock virtual submasters to a virtual wheel so you can control more than one slider at the same time [Figure 25].

Some are not even related to lighting per se—for example, the ability to bring in and display, live, a feed from a video camera: You could see the musical director right in the console, as sound operators do, or point the camera at the stage, and LF can store an image of the state into the cue notes field each time you record a cue. LF can also directly run audio and video files, with defined start points and lengths—perhaps the software is trying to replace the soundies?

And some are just very powerful lighting tools. LightFactory can have lots of cue lists, all running at the same time if necessary, and, within each, you can set timing per cue or per parameter. It also has a comprehensive effects section, able to create “shape”-type effects [Figure 26] step effects, or effects structured using a simple timeline [Figure 27] that lets you easily assign how events happen relative to one another, then run them manually for busked shows [Figure 28] or triggered by cues—all things you'd expect a modern lighting controller to do. But it also has tools that aren't standard on every modern controller: track-

ing backup/multi-user functionality, macros and real-time events, and DMX in/merge. The ability to create and edit profiles (whether simple fade curves or complex paths to be followed by moving light position or color changes) [Figure 29]. Or the ability to define matrices of (usually LED) fixtures, which you can then control by mixing colors, by defining paths through sets of colors, or by choosing image files as the background [Figure 30].

Plus, LightFactory give you unprecedented access to mess around with the program's front end: The optional Interface Builder package lets you construct your own interface, as simple or complex as you like, down to just “start” and “stop” buttons for permanent installations.

Conclusions

I realize I sound like I'm gushing more than I have in many of the other console reviews that have appeared in this magazine. To counterbalance that a little, let's be clear about some of LightFactory's limitations. Window data is sometimes not refreshed properly. The fixture library is comprehensive, but the data it contains is sometimes frustratingly incomplete, and it's a shame not all of the fixture data is real-world data. Some good ideas are not yet fully fleshed out (the “virtual intensity” control for LEDs doesn't yet work as you'd hope). And LightFactory's tracking model is currently undergoing some revisions to more closely match the behavior users of standard theatre tracking consoles expect while retaining the ability to turn tracking off and safely record absolute states—something very few consoles now offer. Also, there were occasional crashes, though always when performing some utility task rather than in active playback, and never with data loss.

One could say that the low price compensates for these, but that's needlessly demeaning. By any measure, from a bang-for-the-buck point of view, there's a remarkable amount of functionality here—both the big-ticket items (huge output count if you need it, versatile moving light control, LED matrices) and the many details that have clearly come from the developer actually using his software on real shows. Plus, it's remarkably easy to get going, almost as easy to dig deeper: The interface does a great job of explaining itself, then getting out of your way once you get the hang of it. And you get a developer who reacts quickly to fix problems as they appear and add new ideas as they're suggested.

It's not a revolution in lighting control, but it is a very useful real-world-led evolution, and one that would be well worth investigating whether you think you're in the market for a software-only lighting control system or not. Even if you prefer the hardware of your traditional console, you'll quickly miss many of LightFactory's real-world tricks, and wonder why your manufacturer didn't build them into your console. 🐼