

DMXControl 3 Stories

dmxcontrol-projects.org

**Ein Community-Skript
zur Nutzung der freien Lichtsteuerungs-Software DMXControl 3**

+

[Vereinsseite](#)

[Software Download](#)

+

+



.1. Einleitung – warum dieses Skript?

Dieses Dokument entstand als Skript im Rahmen der Vorbereitung eines DMX-Lichttechnik-Kurses als „Allgemeine Wissenschaftliche Erweiterung (AWE)“ an der HTW Berlin. In diesem Tutorial geht es hauptsächlich um die Anwendung des freien Tools DMXControl3 für die DMX-basierte Lichtsteuerung, da die Teilnehmer am Ende des Kurses in der Lage sein sollen, eine Show mit dem Tonkollektiv der HTW zu beleuchten.

Natürlich gibt es schon sehr viele Youtube-Beiträge zu DMXControl3 und noch viel mehr Wiki-Artikel dazu. Diese Quellen werden auch im Skript stets zur Vertiefung des Verständnisses des Tools empfohlen und verlinkt. Viele Grundlagen zur DMX-Technik werden auch im schon betagten Buch „Lichttechnik für Einsteiger“ beschrieben, welches von unserem Verein DMXControl-projects.org herausgegeben wurde.

In diesem Dokument soll es also nicht um die Beschreibung der Menüpunkte und einzelnen Funktionen wie im Handbuch oder in den Wiki-Seiten gehen, sondern um grundsätzliche Herangehensweisen (Stories) bei der Nutzung von DMXControl3. Aufgrund des riesigen Umfangs von DMXControl3 kann und will dieses Skript niemals die Wiki-Seiten ersetzen. Mit diesem Skript soll aber der Zugang zum Tool DMXControl3 erleichtert werden, wobei sich das Skript konkret auf die Version 3.3 bezieht.

Daher werden verschiedene Anwendungsfälle in den diversen Showtypen herausgestellt und viele Tricks verraten. Eine nach Zeitmarken gesteuerte Show ist etwas anderes als ein Livekonzert, wo der Lichtoperator auf aktuelle Situationen reagieren muss. Ganz anders ist es wiederum bei Partys, wo er sogar zwischendurch mal ein Bier trinken gehen kann und das Programm allein laufen lassen kann.

Aus systematischen und didaktischen Gründen wurden die Anwendungsfälle (Stories) immer konkreten Eventtypen (z.B. Disco, Musical) zugeordnet. Dennoch kann man sie auch in anderen Szenarien anwenden. Es empfiehlt sich also, das Skript komplett zu lesen, auch wenn man z.B. nur konkret eine Party plant. Hinzu kommt, dass die Anwendungsfälle auch mit jedem Kapitel immer komplexer gestaltet werden. Letztendlich erlangt man das komplette Wissen nur durch Üben, Üben und nochmals Üben.

.1.1. Lizenz

Dieses Skript ist lizenziert mit der Creative Commons Attribution-ShareAlike 4.0 International License. Die Lizenz kann hier eingesehen werden: [creativcommons.org](https://creativecommons.org)

Wir danken allen Beteiligten, insbesondere dem DMXControl-Marketing Team für die Unterstützung zu diesem Skript. Es wurden viele schriftliche und mündliche Aussagen von deren Mitgliedern einfach „übernommen“. Ziel ist es, eventuell dieses Skript langfristig durch die Community zu einem open book auszubauen.

Statt einem konkreten Wiki-Link wird manchmal nur der erforderliche Suchbegriff als Stichwort angegeben, so dass man selbst im DMXControl-Wiki suchen kann. Natürlich finden sich auf der Webseite des Vereins [DMXControl-Projects.org](https://dmxcontrol-projects.org) auch Spendenmöglichkeiten für dieses Projekt.

.1.2. Die Story

In diesem Skript treten drei Personen auf, die an mehreren Lichtshows mitwirken - Jana, Jannik und Stefan. Jana ist vom Beruf Bühnenbildnerin und übernimmt das Lichtdesign. Sie hat mehr Ideen als ein Hund Flöhe hat und sieht sich als die unangefochtene Chefin der Show. Sie weiß aus ihrem Theater- und Musicalhintergrund, dass die Programmierer einer Show mit Cues und Cuelisten arbeiten, sie selbst arbeitet aber nur mit Skizzen zur Show. Wenn sie vor einer Vorstellung nicht mindestens einmal ihre Ideen und Meinungen ändert, dann ist etwas faul.

Jannik ist der Programmierer der Show mit DMXControl3. Da er schon mit einigen Lichtsteuerungspulten gearbeitet hat, kennt er viele allgemeine Konzepte der Showprogrammierung, aber er ist noch nicht so richtig firm mit einer Software-Steuerung DMXControl3. Daher hat er vielleicht eine große Gemeinsamkeit mit vielen Lesern. Jannik ist sehr begeistert, was diese Freeware im Vergleich zu vielen Pulten alles beherrscht und wie schnell er sich in die Software hineinarbeiten kann.

Der Experte im Hintergrund ist Stefan. Er ist intensiv mit der Entwicklung von DMXControl3 verbunden, kennt fast jedes Feature und hat schon hunderte Wiki-Seiten zu diesem Tool erstellt. Außerdem erstellt er regelmäßig mit einem Kumpel aus dem Projekt die „Schon gewusst...“-Beiträge im DMXControl-Forum, die den Nutzer an sehr hilfreiche Nutzungstricks erinnern. So kann man ihn zu jedem Problem fragen und meistens hat er schnell eine Antwort parat. In der Tat sind einige Dinge implementiert, wo selbst die Entwickler schon vergessen haben, dass es sie gibt – Stefan kennt diese und unterstützt Jannik stets mit Tipps.

Diese drei Personen – obwohl sie manchmal unterschiedliche Ansichten haben - werden im Team also mit Euch gemeinsam dieses Buch und die unterschiedlichen Shows gestalten.

1. Kapitel 1: Alles fängt bei den Grundlagen an

Auch wenn wir im Zeitalter der KI sind – die Beleuchtung einer Show (egal wie groß oder klein) macht sich immer noch nicht von allein und man muss schon einige Grundlagen beherrschen. Hier gibt Stefan einen ersten Überblick zum Verständnis der Basics, vertiefende Details findet man wie üblich im Wiki und in späteren Kapiteln.

1.1. Einige Begriffe – kennst Du alle?

Jannik kennt sich ja bereits mit der DMX-Steuerung mit einem Pult aus. Viele der dort verwendeten Begriffe findet er auch in der Software DMXControl3 wieder. Auch wenn man normalerweise in jedem Buch das Abkürzungsverzeichnis und die Begriffserklärung überspringt, solltest Du hier bei unbekannten Begriffen eine Ausnahme machen. Hier wird sozusagen die Grundlage des Verständnisses der weiteren Kapitel gelegt, aber später wird alles noch mal ausführlich erklärt. Ein Experte würde vielleicht manchmal an unseren Erklärungen rummäkeln, aber Stefan versucht es mit sehr einfachen und verständlichen Worten zum Einstieg.

Table 1. Grundbegriffe

Begriff	Erklärung
HAL Hardware Abstraction Layer	<p>DMXControl3 denkt nur in Eigenschaften/Attributen der Lampe wie Farbe, Helligkeit, Position (bestehend aus Pan, Tilt) usw.. Wir werden nun immer von Eigenschaften der Geräte sprechen. Konkrete DMX-Werte zur Ansteuerung sollen den Nutzer nicht interessieren, d.h. DMXControl3 übersetzt automatisch z.B. die Farbe „rot“ in einen DMX-Wert, der an die Lampe gesendet wird. Um dieses tun zu können wird für jedes Gerät ein DDF benötigt.</p> <p>Wiki: Hardware Abstraction Layer (HAL)</p>
DDF Device Definition File	<p>Damit DMXControl3 die HAL implementieren kann, benötigt es die Information, welcher DMX-Wert auf welchem Kanal eine entsprechende Reaktion erzeugt. Im Prinzip wird hierbei die DMX-Belegungstabelle aus der Bedienungsanleitung des Gerätes in einem XML-Beschreibungsformat abgebildet. Damit nicht jeder Nutzer für seine Geräte die DDFs selbst erstellen muss, werden viele DDFs für Hunderte Lampentypen über die Community von DMXControl bereitgestellt. Man kann Geräte in DMXControl3 nur komfortabel verwenden, wenn ein entsprechendes DDF verfügbar ist.</p> <p>Wiki: DDF-Einführung DMXC3</p>
Cue und Cueliste	<p>Deutsch Cue: Szene/Lichtstimmung. Der Begriff kommt ursprünglich aus dem Theater und bedeutet, dass auf ein „Schlagwort“ eine neue Szene beginnt. Das wurde auf das Licht übertragen, zumal die neue Szene meistens wirklich mit einer neuen Lichtstimmung verbunden ist. Cues müssen nicht immer statisch sein, sondern können auch dynamische Effekte enthalten, wie Du sie vielleicht bei schwenkenden Movingheads schon gesehen hast. Die einzelnen Positionen der Scheinwerfer müssen dabei nicht einzelne Cues sein (könnten es aber auch).</p> <p>Eine Cueliste enthält eine (Ab-)Folge von Cues, aber wird öfter auch nur eine einzelne Cue enthalten.</p> <p>Wiki: Cuelist DMXC3</p>
Effekt	<p>Einen Effekt kannst Du wie eine vordefinierte Funktion betrachten, die Du in eine Cue einfügen kannst, z.B. ein Lauflicht innerhalb einer Gerätegruppe oder ein sinusartiger Verlauf der Helligkeit oder Position der Lampen in einer Gruppe. Effekte können extrem viel Zeit bei der Programmierung der Show einsparen.</p> <p>Wiki: Effects and Filters DMXC3</p>

Begriff	Erklärung
Master	<p>Einen Master kann man sich wie einen Slider/Fader auf einem Pult vorstellen, mit dem man etwas (z.B. Helligkeit) zwischen 0 und 100% regelt, aber es gibt auch andere Regelgrößen wie Speed oder Color. Am bekanntesten ist der Grand Master, mit dem man die Helligkeit der Show insgesamt steuern kann. Es gibt aber auch Master für Geschwindigkeit, Farben und vieles andere.</p> <p>Wiki: Master DMXC3</p>
Executor	<p>Auch den Begriff Executor kann man wörtlich nehmen: Er „vollsteckt“ bestimmte Aktivitäten. So kann er z.B. Cuelisten starten, pausieren und stoppen, wenn ihm diese vorher zugeordnet wurden. Es kann in einer Show mehrere aktive Executoren geben. Ein Executor besteht immer ähnlich zu einem Master aus einem Slider, aber hat standardmäßig noch einige Buttons für spezielle Funktionen (Flash, Start, Go, Stop).</p> <p>Wiki: Executor DMXC3</p>
Softdesk	<p>DMXControl3 hat als Software-Produkt eher eine flexible Window-basierte Oberfläche und unterscheidet sich damit erstmal von der Bedienung mit einem DMX-Lichtpult. Um aber ein ähnliches Bediengefühl wie an einem Pult zu erlauben, gibt es die Möglichkeit sich virtuelle Pulte zu basteln. Dort kann man Slider/Fader, Buttons und andere Elemente nach dem eigenen Geschmack platzieren. Solche Softdesks werden auch manchmal erstellt, um einem nicht-affinen bzw. „ungelernten“ Bediener zu ermöglichen, die Show abzufahren. So könnte Jana mit einem speziellen Softdesk ohne Vorkenntnisse das Licht steuern, wobei sie nur einzelne Buttons drücken muss.</p> <p>Wiki: Softdesk DMXC3</p>
HTP/LTP	<p><i>(wird ergänzt, Wo wird das eingestellt?)</i></p> <p>Wiki: HTP-, LTP- and LoTP-Mixing (Cuelist) DMXC3</p>
Tracking	<p>Eine Show ist immer eine Folge von Cues. Ganz vereinfacht gesagt, regelt man mit dem Tracking ob jede Cue komplett unabhängig ist oder ob der Ablauf der Cues im Vordergrund steht, die vorhergehenden Cues also „Spuren“ (Track) für die folgenden hinterlassen. Bei aktiviertem Tracking in einer Cuelist bleiben alle Werte solange am Ausgang bis sie durch eine neue Cue überschrieben werden.</p> <p>Wiki: Tracking (Cuelist) DMXC3</p>

Begriff	Erklärung
Trigger	<p>Mit Triggern wird die Abfolge von Cues in einer Cueliste gesteuert, so kann man z.B. Zeitspannen oder absolute Zeiten als Startpunkt angeben. Es gibt weitere Formen, z.B. kann die nachfolgende Cue gestartet werden, wenn 3 Beats erkannt wurden.</p> <p>Wiki: Cuelist (Options) DMXC3</p>
RGB/HSV	<p>RGB hat jeder schon mal gehört. Das sind die rot-grün-blau Farbanteile für eine Mischfarbe. HSV steht natürlich nicht für „Hamburger Sport Verein“, sondern steht für hue-saturation-value (also Farbwert, Sättigung, Hellwert) und ist ein alternatives Farbmodell das der menschlichen Farbwahrnehmung stärker entspricht. Es gibt da komplexe Umrechnungstheorien. Wir begnügen uns damit, dass wir hiermit besser unsere Farbwahrnehmung für farbige LEDs einstellen können und DMXControl auch dieses Farbmodell unterstützt.</p> <p>Wiki: HSV-Farbraum</p>
Universe	<p>Ein Universe ist ein Strang aus Geräten mit bis zu 512 DMX-Kanälen. Für Theaterscheinwerfer war das seinerzeit eine gigantische Zahl. Mit der Zahl kamen Geräte auf dem Markt, die allein schon 30 oder 50 Kanäle in Anspruch nehmen, so dass irgendwann 1 Universe für alle Lampen nicht mehr ausreichte. DMXControl unterstützt mehrere Universes (siehe erste Ziffer bei der Adressierung der Lampen, z.B. 1.102 oder 3.84). Unser Nodle-Interface dagegen kann nur 1 Universe, so dass man ggf. mehrere DMX-Interfaces benötigt. Aber: Alle Shows, die Jannik bisher gemacht hatte, kamen immer mit einem Universe aus.</p>
<i>Input Assignment</i>	Eingangszuweisung ... wird ergänzt
Timecode-player	<p>Das ist ein ganz cooles Tool, das Du Dir wie ein Videoschnittprogramm für eine Lichtshow vorstellen kannst. Man kann auf den einzelnen Spuren Audiofiles und Lichteffekte zeitlich platzieren, so wie man in einem Videoschnittprogramm Video- und Audioschnipsel einfügen kann.</p> <p>Wiki:</p>
Visualizer	<p>Wie der Name andeutet, kannst Dir in einem Visualizer deine Show vorab visualisieren, also anschauen. Mit der Bühnenansicht liefert DMXControl3 einen internen 2D-Visualiser der für die meisten Anwendungen komplett ausreicht. Wer es komfortabler haben möchte, kann auf (externe) 3D-Visualiser zurückgreifen, die bis in die Kategorie virtuelle Welten/Metaverse (z.B. Unreal Engine) reichen können. Siehe Kapitel x.x</p>

Offene Begriffe (für spätere Kapitel): - Highlight. Siehe „schon gewusst“ SG5 - Blackout - Autoprepere

1.2. Die Schummelseite

Wie in vielen Büchern der Reihe „<Irgendwas> für Dummies“ gibt Dir Stefan hier zwei „Schummelseiten“ zu Beginn mit, die die wichtigsten Botschaften des Skriptes zusammenfassen.

Deine Showaktivitäten bestehen immer aus 3 Teilen:

- Einrichten (Vorbereiten) der Show inkl. Setup der Lampen
- Editieren (Programmieren) der Show
- Exekutieren (Abfahren) der Show

1.2.1. Vorbereitung der Show

- Du benötigst eins der von DMXControl3 unterstützten DMX-Interfaces, ansonsten kannst Du die Show nur in der 2D-Bühnenansicht oder in einem 3D-Visualizer sehen.
- Halte Dir stets eine Übersicht über deine Lampen bereit. In der Regel können sie bei jeder Show immer die gleichen DMX-Adressen behalten. Daher mache Dir einen Plan bzgl. der Position der Lampen, ihrer Adressen und in welchem Kanalmodus sie betrieben werden sollen.
- Suche die entsprechenden DDF-Dateien vorab heraus. Notfalls kannst Du diese selbst erstellen.
- Nun kannst Du die Geräte in der Bühnenansicht oder im Project Explorer erzeugen und zu Gruppen zuordnen. Dabei auf die richtige Adresszuordnung achten.

1.2.2. Programmieren der Show

- Wichtig ist eine Gruppenzuordnung der Lampen. Eine Lampe muss nicht nur der initial erzeugten Gruppe bei Einrichten zugeordnet sein, sondern kann auch zusätzlich in anderen Gruppen enthalten sein.
- Eine Gruppe können gleichartige Geräte sein, aber z.B. auch alle Lampen mit einer bestimmten Aufgabe wie Frontlicht, Blender, Seitenlicht, Oberlicht, linke Seite, rechte Seite usw.
- Denke bei der Programmierung mit DMXControl niemals in DMX-Werten, sondern nur in Eigenschaften (Attributen) wie Farbe, Position, Stroben usw.
- Nachdem Du diese Eigenschaften für Gruppen oder einzelne Lampen eingestellt hast, kannst Du die aktuelle Lichtstimmung in einer Cue abspeichern. Die eingestellten Eigenschaften kannst Du vorher im Programmer überprüfen und ggf. editieren.
- Eine Show kann aus einer oder vielen Cuelists bestehen. Eine Cuelist kann eine oder viele Cues enthalten, die über verschiedene Mechanismen getriggert (also gestartet) werden können, z.B. manuell, per Timecode oder beat getriggert.
- Möglicherweise möchtest Du die Cuelisten nicht direkt über die GO-, Pause- oder Stop-Button in der Cuelist steuern, sondern die Aktivierung über die Tastatur, ein DMX- oder MIDI-Pult oder mit einem Softdesk vorbereiten. Dann musst Du Dich mit dem Input Assignment beschäftigen.
- Cues können besonders wirkungs- und effektiv sein, wenn Du Fanning oder einen der vielen vorbereiteten Effekte integrierst.
- Während des Programmierens kannst Du die Show stets in der Bühnenansicht prüfen oder (mit etwas mehr Aufwand) auch in einem Visualizer anschauen.

1.2.3. Abfahren (Exekutieren) der Show

- Das Ausführen der Show hängt sehr stark von der Art der Veranstaltung ab (Band-Konzert, Musical, Theater, Club-/Tanzveranstaltung). Das wurde natürlich schon während der Programmierung berücksichtigt.
- Entsprechend werden unterschiedliche Mittel eingesetzt. Manchmal nur eine Cuelist, die pro Cue manuell weitergeschaltet wird, manchmal baut man sich ein Softpult um live per Button-Klick bestimmte Cuelisten zu starten.
- Es gibt auch Shows in denen man die Cues der Lichtshow präzise auf Millisekunden zu einem fertigen Song startet, d.h. während dieser Zeit läuft alles ganz automatisch ab.
- Die vielfältigen Möglichkeiten wirst Du in diesem Skript kennenlernen.

1.2.4. Übersicht über Quellen

- Forum <https://forum.dmxcontrol-projects.org/>
- Wiki <https://wiki-de.dmxcontrol-projects.org/index.php?title=Hauptseite>
- Youtube-Channel <https://www.youtube.com/dmxcontrol>
- Buch „Lichttechnik für Einsteiger“ tbd. (Elektor-Link)

Mitwirkende:

- DMXControl: Frank B., Marten J., Stefan Ki., Jens-Peter K. Christian H. ala Hoc
- HTW Berlin: Patrick F., Bruno A., Roman W., ...

2. Kapitel 2: Überblick über DMXControl3

Jannik muss sich erstmal mit DMXControl3 vertraut machen. Natürlich hat er das Forum auf der Webseite schon besucht und hat sich dort kostenlos registriert. Einige der dort gestellten „Anfängerfragen“ werden in diesem Kapitel besprochen.

2.1. Voraussetzungen

Jannik muss also erstmal seinen Laptop vorbereiten und DMXControl3 (Version 3.3) installieren. Um das Tool auszuprobieren benötigt man erstmal keinen der unterstützten DMX-Transmitter (DMX-Interfaces), da man sich seine Show auch in einem Visualizer oder der Bühnenansicht anschauen könnte. DMXControl3 läuft nur unter aktuellen Windows-Versionen (64 bit). Es müssen .NET6 und ASP.NET (jeweils sowohl 32 als auch 64 bit Version) installiert sein. In der Regel wird das mit dem Installer überprüft und installiert.

[DMXControl3 Launcher (ab Version 3.3)] | *Kap2_Launcher.JPG*

Abb.2-1: DMXControl3 Launcher (ab Version 3.3)

Ab DMXControl3 Version 3.3 gibt es den Launcher, der den Start der benötigten Komponenten übernimmt:

- Die GUI ist der sichtbare Teil zur Programmierung und Ausführung der Show.
- Der Kernel läuft standardmäßig im Hintergrund und ist ein Prozess, der besonders stabil läuft damit der PC stets Signale an das Interface sendet. Er empfängt die Signale über die Umbra-Instanz von einer oder mehreren GUI-Instanzen und sendet diese über das DMX-Interface an die Lampen. Selbst im unwahrscheinlichen Fall des Absturzes einer GUI würde der Kernel weiter die letzten DMX-Signale senden.
- Der Umbra-Prozess ist meistens verdeckt. Dieser Prozess wird nur im Hintergrund benötigt um die Kommunikation innerhalb DMXControl3 zu gewährleisten. Technisch steckt da das gRPC-Protokoll dahinter, aber das braucht Jannik wirklich nicht zu interessieren.

[Architektur] | *Kap2_DMXC_Architektur.JPG*

Abb. 2-2: Architekturbild DMXControl3 (ab Version 3.3 läuft die Kommunikation immer über Umbra)

Das Gesamtsystem startet man mit dem großen Button im Launcher. Als normaler Nutzer hat man nichts mit dem Kernel- und Umbra-Prozessen zu tun, außer dass euch wegen der erforderlichen Client-Server Kommunikation eventuell der Firewall auf dem Rechner dazu um Erlaubnis fragen wird. Die beiden internen Prozesse haben „nur“ ein sog. Command Line Interface, d.h. man kann bestimmte Steuerungsbefehle eintippen, siehe Wiki „Kernel DMXC3“. Aber das benötigt Jannik in der Beginner-Phase wirklich nicht.

2.1.1. Toolübersicht

Beim Starten des Tools erscheint erstmal ein großes Fenster mit vielen Unterfenstern. Dazu hier ein kurzer Überblick über eine übliche GUI-Konfiguration von DMXControl3. Je nach Anwendung kann man sich völlig unterschiedliche Fenster- bzw. Tool-Konfigurationen erstellen, aber hier erklärt Stefan einige der wichtigsten Funktionen im üblichen Hauptfenster.

[Toolübersicht] | *Kap2_ToolUeberblick.JPG*

Abb. 2-1: Toolüberblick

Table 2. Toolübersicht

Nr	Erklärung	Wiki-Begriff
1	Programm-Menü. Hierüber können die verschiedenen Tools aufgerufen werden und Konfigurationseinstellungen vorgenommen werden. Die Menüpunkte werden weiter unten erklärt.	
2	Hauptmenü. Hier sind die wichtigsten Funktionen im Schnellzugriff erreichbar, so z.B. der Programmer- und der Cue-Bereich. Während der Aufführung sind die Buttons „Hervorheben, Blackout, etc.“ sehr nützlich.	
3	Projektexplorer. Ähnlich wie beim Windows-Explorer kann man mit dem Projektexplorer alle Objekte der Show managen. Das können Geräte, Bedienelemente, Cues und Dateien sein. Neben der Bühnenansicht (4) das wichtigste Administrationstool!	

Nr	Erklärung	Wiki-Begriff
4	Bühnenansicht. Diese liefert eine 2D-Ansicht und eine ungefähre Visualisierung der Show. Hier können Geräte (-Gruppen) ausgewählt werden um für diese dann Lichtstimmungen (Cues) zu einzustellen. Sie zeigt aber unbedingt nicht das an, was am Ende rauskommt, sondern das, was DMXC3 ausgeben möchte. Sie hört im Gegensatz zu einem Visualizer also nicht auf DMX-Werte.	
5	Eigenschaften-Controls. Hier kann man über grafische Controls die wichtigsten Eigenschaften der Lampen beeinflussen: Helligkeit, Farbe, Gobo, Pan/Tilt. Die Auswirkungen sieht man sofort in der Bühnenansicht (4) und in der Gerätesteuerung (6).	
6	Gerätesteuerung. Wie bei den Controls (5) können hier die Eigenschaften der Geräte gesetzt werden. Durch eine Zahlen-/Kommandoeingabe kann das aber viel granularer und genauer erfolgen. In anderen Programmen würde dieses Tool vielleicht „Inspector“ genannt werden, da man sich wirklich alle Eigenschaften genau anschauen kann. Weiterhin können tolle Effekte wie Fanning oder Sinuseffekte hier programmiert bzw. eingefügt werden.	
7	Effekte und Filter. Diese vorprogrammierten Effekte können im Zusammenhang mit der Gerätesteuerung (6) verwendet werden. Ein beliebtes Beispiel ist der Chaser. Die Effekte kann man in der Regel per Drag&Drop in die Gerätesteuerung (6) ziehen.	
8	Netz. DMXControl ist als verteiltes System programmiert. Hier hat man einen Monitor für den Netzwerktraffic. <Wann wird es kritisch??>	
9	Projektadministrations-Menü. Hier können allgemeine Projektmanagement-Aufgaben aufgerufen werden	
10	Diverse Reiter. DMXControl erlaubt es, beliebige Fensteranordnungen zu gestalten. Diese Reiter-Anordnung ist eine platzsparende Möglichkeit, viele Fenster/Tools gleichzeitig zugreifbar zu machen.	
11	Preset-Panel (in Abb. nicht zu sehen): Presets sind wie Templates für bestimmte Eigenschaften, die man in mehreren Cues nachnutzen kann um immer konsistente Farben oder andere Eigenschaften zu generieren.	

2.1.2. Das Fenstersystem

Aufgrund der großen Anzahl von Tools haben nicht alle gleichzeitig Platz auf dem Bildschirm. Außerdem benötigt jeder Nutzer nach Anwendungsfall sowieso unterschiedliche Fenster/Panels. Je nachdem, ob Jannik nur mit dem Laptop arbeitet oder 2 oder 3 zusätzliche Monitore zur Verfügung hat, wird er seine eigene GUI-Konfiguration anders wählen. Eine individuell konfigurierte Benutzeroberfläche kann auch im DMXControl-Showprojekt gespeichert werden, so dass beim nächsten Projektaufwurf die gleiche Fensteranordnung wieder erscheint.

Prinzipiell werden die Tools über das Programmmenü aufgerufen. Sie können als freischwebende Fenster agieren, einen eigenen Bereich bekommen oder an existierende Fenster angedockt werden, wo sie dann als neuer Reiter erscheinen. Prinzipiell zeigt der hellblaue Bereich beim Verschieben des Fensters, wo das neue Panel landen würde. Nach kurzer Übung schafft es Jannik jedes Panel

wie gewünscht anzuordnen. Er muss dazu den Mauszeiger über den erscheinenden transparenten Pfeil ziehen.



Diese Panel-Anordnung klappt allerdings manchmal nur auf dem Hauptmonitor. (Sollte aber nicht so sein, das hängt scheinbar von der Grafikkarte ab.)

Ein nicht mehr benötigtes Panel kann Jannik mit dem X-Symbol (jeweils rechts oben im Panel) entfernen. In der Regel füllen dann die restlichen Panels den freigewordenen Platz automatisch aus.

Wiki: Panel-Konzept DMXC3

2.1.3. Projektextplorer

Stefan erklärt Jannik den eigentlichen Startpunkt des Projektes: Der Projektextplorer übernimmt in DMXControl3 die zentrale Verwaltung des gesamten Projekts. Man kann hier alle im Projekt verwendeten Objekte wie Geräte, Cuelisten, Bühnenansichten, Timecodeshows, Dateien etc. organisieren, neue hinzufügen oder nicht mehr benötigte löschen. Für eine bessere Übersicht wurden, ähnlich wie im Windows Explorer auch, Ordner angelegt und einzelne Objekte können darin einsortiert werden. Die Menüleiste des Projektextplorers oder die rechte Maustaste bieten einfachen Zugriff auf verschiedene Funktionen der Objekte.

Die Struktur des Projektbaums (siehe Ziffer 3 in Abb. 2.1) mit seinen einzelnen Zweigen ist in DMXControl3 fest vorgegeben und kann nicht geändert werden. Jeder Zweig ist ein Ordner und strukturiert auf diese Weise das Projekt, so dass zum Beispiel Cuelisten immer im Ordner Cuelisten zu finden sind. Einige Objekte kann man nicht über die Bühnenansicht erzeugen, sondern man muss sie immer im Projektextplorer erstellen (z.B. Executoren, Softdesks). In dieser Verzeichnisstruktur werden sich später auch andere Objekte wie mp3-Dateien, Hintergrundfotos usw. finden.

Jannik kann sich also hier stets einen Überblick verschaffen, welche Objekte in der Show existieren. Es gibt vielleicht eine Ausnahme: Die einzelnen Cues sind hier nicht zu finden, diese kann man auch niemals direkt aufrufen, sondern immer nur in ihrem „Container“, der Cuelist.

<Bild mit gefülltem Projektextplorer ergänzen>

Jannik ist begeistert, dass hier viele Operationen über Drag&Drop realisiert werden können, z.B. das Hinzufügen von Geräten zu Gerätegruppen, Cuelisten zu Cuelistgruppen, Audiofiles zu Timecode-Tracks, aber auch vieles mehr. Was Gerätegruppen sind, erklärt Stefan im nächsten Abschnitt.

Wiki: Project Explorer DMXC3

2.2. Grundlegende Konzepte zum Verständnis

2.2.1. Gruppen – Sind Deine Lampen schon in Teams organisiert?

Stefan sagt: **Axiom 1:** Gewöhne Dir von Anfang an, immer im Gruppenkonzept zu denken!

Gruppen-Unterstützung ist neben HAL und dem Fanning (Kap.x.x) eines der stärksten Features von DMXControl3. Wenn Du Lampen in Gruppen anordnest, kann Du zukünftig Effekte über eine solche Gruppe programmieren. Du kannst existierende Gruppen um neue Lampen erweitern und die bereits programmierten Effekte werden automatisch auf die neuen Mitglieder erweitert. Das ist irgendwie Magic... Sollte mal ein Gerät kaputtgehen oder ausgetauscht werden müssen, ersetzt Du es einfach in dieser Gruppe. Das tollste ist, das neue Gerät kann sogar einen anderen Typ haben und DMXControl3 versucht trotzdem das Beste herauszuholen, so dass das neue Gerät sich wie die anderen Gruppenmitglieder verhält. Man kann hier also von echtem Teamwork reden und DMXControl3 unterstützt Vielfalt in den Geräten! Wenn man in Vornherein die Show über Gruppen programmiert, ist man also viel flexibler als wenn man eine Cue über einzelne Geräte definiert! Wie im täglichen Leben: Gruppenarbeit ist toll...

Aus diesem Grund kann es sogar sinnvoll sein, auch für eine einzelne Lampe eine Gruppe anzulegen. Wir werden auch sehen, dass eine Lampe auch zu mehreren Gruppen gehören kann.

Die Organisation der Gruppen kann über mehrere Szenarien erfolgen:

- beim Hinzufügen der Geräte im Setup-Menü
- in der Bühnenansicht (Markieren der Geräte und Kontextmenü)
- im Projektextplorer (Drag&Drop in Gerätegruppen)

Wiki: Device_Group_DMXC3

2.2.2. Cues als atomarer Baustein jeder Show

Stefan erklärt, dass eine Show irgendwie immer eine Abfolge von Cues ist. Mit Cue bezeichnet man klassisch eine Lichtstimmung, also im Theater z.B. die konkrete Beleuchtung der Schauspieler oder des Bühnenhintergrundes. Im einfachsten Fall wird die Lichtstimmung (Cue) in der Bühnenansicht eingestellt, dazu kann man verschiedene Controls oder das Fenster der Geräteeigenschaften verwenden. Die aktuellen Werte gehen normalerweise sofort an den Ausgang. Sie werden aber auch im Programmer zwischengespeichert. Im Programmer kannst Du sehen, welche Änderungen Du an welchen Geräten und welchen Eigenschaften gemacht hast und diese Änderungen ggf. auch wieder abwählen. Wenn alles OK ist, speicherst Du es in einer neuen Cue. Bevor die Cue einer Cuelist hinzugefügt wird öffnet sich der Programmer Filter, wo man letzte Überprüfungen und Korrekturen machen kann. Es ist aber auch vorher im Programmer selbst möglich, bestimmte Werte für bestimmte Funktionen wieder zu entfernen. Erst dann werden die Werte tatsächlich in einer Cue gespeichert. Damit hast Du die aktuelle Lichtstimmung eingefangen. Das machst Du nun genauso mit der gewünschten nachfolgenden Cue: neue Cue einstellen, im Programmer (Filter) kontrollieren, speichern, fertig.

Eine Cue kann niemals selbständig existieren. Eine Cue benötigt immer einen Container – die Cueliste - in dem sie aufbewahrt wird. Ebenso kann man eine Cue nie direkt aufrufen, sondern nur die Cueliste, in der sie enthalten ist.

Damit später beim Abspielen der Cuelist alles genauso aussieht, musst Du das **Tracking** deaktivieren. Warum es standardmäßig angestellt ist, erklären wir später im Abschnitt x.x...

2.2.3. DMX-Ausgabe – Was passiert dabei wirklich?

Alles schön und gut, denkt Jannik. Aber wie hängt das nun alles mit dem Ziel zusammen, meine Lampen in der Show zu steuern? Was passiert eigentlich grundsätzlich bei der DMX-Ausgabe und welche Tools spielen da rein?

Ach, ich hab‘ da etwas, sagt Stefan. Hier ist ein Bild, wo ich mal für Studenten des Lichtsteuerungskurses aufgemalt habe, wie das alles zusammenhängt:

[DMX-Ausgabe] | *Kap2_DMXAusgabe_Gesamtsicht.JPG*

Abb. 2-x: Signalfluss in DMXControl3

Jannik ist erstmal erschlagen von der Komplexität, aber die meisten Begriffe wurden ja schon auf den früheren Seiten eingeführt. Da musst Du notfalls nochmal nachlesen, sagt Stefan! Beachte, dass die Änderungen in der Kanalübersicht nur direkt in die Kernel-Ausgabe gehen, aber nicht für die Cue-Programmierung benutzt werden können, was Absicht vom Programmiererteam ist. Ansonsten würde man das HAL-Konzept unterlaufen. Die Kanalübersicht ist nur für Prüf- und Testoperationen gedacht.



Warnung von Stefan: Stelle niemals bei der Cuelist-Erstellung irgendwelche Werte über die Kanalübersicht ein. Außer beim Testen von Geräten oder bei der Fehlersuche gilt: Hände weg von der Kanalübersicht! Überprüfe vor der Show, dass in der Kanalübersicht alle Werte auf Null gesetzt wurden (ab DMXControl 3.3 werden dazu entsprechende Warnungen gesendet).

Die wirklich neue Information für Jannik ist nur, dass der Programmer so zentral im Fokus steht. Zwei Punkte sind hier hervorzuheben:

Axiom 2: Alles was im Programmer steht, geht direkt an die Ausgabe (blauer Pfeil) und an die Bühnensicht. Du kannst den Inhalt des Programmers aber über den Button „Programmer leeren“ bereinigen, dann bleibt für die Ausgabe nur das übrig, was gerade durch die aktuellen Cuelists ausgegeben wird.

Axiom 3: Eine neue Cue erzeugt man ausschließlich dadurch, dass das, was im Programmer steht, gespeichert wird. Es gibt keinen anderen Weg, eine „normale“ Cue zu erzeugen. (Ausnahmen wie Spezialcues für Audio usw. erklärt Stefan später.)

Jannik erkennt aus der Abbildung, dass sich der Inhalt des Programmers mit den Werten aus den gerade aktiven Cuelisten überlagern können. So kann es nach „manuellen“ Eingriffen in die laufende Show nützlich sein, zwischendurch den Programmer zu leeren damit nicht die folgenden Cues betroffen sind.

Jannik ist nun vieles klarer geworden, aber eine wichtige Frage hat er noch im nächsten Abschnitt, bevor es an die erste Show geht..

2.2.4. Programmer - Was programmiert dieser eigentlich?

Stefan sagt doch: In DMXControl3 führen alle Wege über den Programmer. Hier werden alle Werte zentral zusammengeführt und zwischengespeichert, die durch das Einstellen einer Lichtszene mit

Hilfe der Gerätesteuerung (Device Control) oder über die Steuerungsfenster (Control Panel) gesetzt werden. Nur Werte, die im Programmer aufgeführt sind, kann man in einer Szene (Cue) speichern.

Im Programmer kann Jannik in Prozentzahlen, Farbdarstellungen usw. nochmal prüfen, was eben passiert ist. Im Gegensatz zur Bühnensicht, wo alle aktuellen Werte bzw. Ausgaben angezeigt werden, sieht er im Programmer nur die letzten Änderungen.

[Programmer] | *Kap2_Programmer.JPG*

Abb. 2-x: Der Programmer, Farbe und Dimmerwerte sind für 6 Lampen verändert

Diese Änderungen kann er nun ggf. korrigieren - mittels Gerätesteuerung oder -Controls oder direkt im Programmer Filter bestimmte Attribute deaktivieren. Wenn alles OK ist, speicherst er es halt in einer neuen Cue.

2.2.5. Programmerfilter

Den aktuellen Inhalt des Programmers kann man sich über den entsprechenden Menüpunkt anschauen, man kann auch alle gesetzten Werte löschen oder nur einzelne Attribute oder Geräte bei der Auswahl herausnehmen. Der Button „Programmer leeren“ macht erwartungsgemäß was sein Name ausdrückt: Alle notierten Änderungen werden gelöscht.

Stefan erläutert Jannik ein Beispiel, bei dem sich Anfänger manchmal wundern: Wenn man beim Abfahren einer Cuelist nebenbei Änderungen in Lichtstimmungen macht, gehen diese sofort in den Programmer und damit zusätzlich zu den DMX-Werten aus der aktuellen Cue an den Kernel. Wenn man nun zur nächsten Cue schaltet, bleiben die Werte aus dem Programmer trotzdem noch am Ausgang. Wenn das nicht gewünscht ist, musst Du eben den Programmer leeren.

Stefan ist noch nicht fertig, er sagt zu Jannik: Der Programmer spielt auch eine große Rolle beim Ändern von bereits erstellten Cues. Wenn Du eine bereits vorhandene Szene in den Programmer lädst (Menüpunkt: „Im Programmer editieren“), werden alle zuvor enthaltenden Werte im Programmer durch die Werte aus der Cue überschrieben. Nun kann man wie gewohnt die Cue-Werte anpassen und das Ergebnis abspeichern.

Nun klickt es endgültig bei Jannik: Der Programmer programmiert im eigentlichen Sinne gar nichts, sondern es ist eher ein Sammler aller der seit dem letzten Löschen oder Speichern einer Cue geänderten Werte. Programmieren muss er dagegen mit der Gerätesteuerung, den Gerätecontrols und den Effekten.



Das Speichern einer Cue löscht die Werte nicht aus dem Programmer. Sie werden nur im Programmer Filter nicht mehr vorausgewählt.

Der Name Programmer ist nur ein historisches Relikt aus Zeiten der Pulte, wo es keine anderen Möglichkeiten gab, die Werte für die Ausgabe zu sammeln. Wir müssen später aber Stefan noch fragen, wofür man die Schnappschuß-Funktion im Programmer verwendet...

2.2.6. Executoren vs. Master

Wenn wir schon bei historischen Relikten sind, dann will Stefan noch schnell die Rolle von

Executoren und Mastern als weitere Bedienelemente zur Showsteuerung klären (siehe Begriffe Kap. 1).

Jannik hat gesehen, dass man damit den Ablauf von Cuelisten steuern kann (siehe Abb. 2-2: Signalfuss), aber nicht muss, weil das auch über andere Wege geht. Sowohl Master und Executoren werden traditionell mit Reglern (Slider, Fader) dargestellt und bedient. Beide Typen kommen eigentlich nur beim Ausführen der Show zum Einsatz, sollten aber während der Showprogrammierung geeignet konfiguriert werden.

[Master] | *Kap2_Master.JPG*

Abb. 2-x: Master in DMXControl3

Die Master können global die Helligkeit von Effekten beeinflussen. Das gilt unabhängig davon, welche Cuelists gerade ausgeführt werden. Wenn im Effekt ein Speedmaster referenziert wird kann auch die Effektgeschwindigkeit geregelt werden.

Executoren dagegen sind in der Regel mit einer Cuelist verknüpft, deren Ablauf man steuern will (Stefan: Mit DMXC3.3 ist diese Aussage nicht mehr Stand der Dinge. ??). So kann man sich nach Bedarf diverse Executoren ggf. auch in verschiedenen Bänken (Pages) erstellen. Die Zuordnung der Cues passiert wieder über Drag&Drop im Projektexplorer, nachdem dort entsprechende Executoren erstellt wurden. Reicht die Standard-Bank nicht aus, kann man im Projektexplorer weitere Bänke anlegen.

Warum man Executoren auch mit Mastern verknüpfen kann, ist eine gute Frage, die Stefan vielleicht im nächsten Live-Meeting beantworten kann

Zu ergänzen: Jannik hat im Forum gelesen, dass es nun auch Color Master und Parameter Master gibt. Was zum Teufel ist das nun schon wieder? Gerade bei Parameter Mastern hat er nicht mal eine geringste Idee...

2.2.7. Fit for Executoren

Hier sollen „Expertentipps“ zur Verwendung von Executoren gesammelt werden

2.3. Das erste Beispiel-Projekt

Nach dem ganzen theoretischen, nervigen und möglicherweise unsinnigen Vorgeplänkel will Jannik nun endlich ein kleines Beispielprojekt erstellen um die Tools zum ersten Mal auszuprobieren (eine Nachahmung für den Leser ist dringend empfohlen, das Projekt wird aber auch im github nachgeliefert!).

In einem Partykeller sollen 6 LED-PARs installiert werden. Jannik entscheidet sich für dieses Gerät namens „LED PARTY Spot“ vom Hersteller Eurolite:

[LEDParty] | *Kap2_LEDParty_Thomann2023.JPG*

Abb. 2-x: Eurolite LED PARTY TCL Spot (Quelle: Thomann.de)

Der Gerätenamen ist übrigens ein nettes Wortspiel, weil darin PAR auftaucht, was in der

Lichttechnik der allgemeine Name für Scheinwerfer ist (<https://de.wikipedia.org/wiki/PAR-Scheinwerfer>). In der Anleitung findet er die Information, dass das Gerät im 6-Kanal-Modus arbeitet:

[Kap2_LEDParty_6Kanal] | *Kap2_LEDParty_6Kanal.JPG*

Abb. 2-x: DMX-Betrieb LED PARTY (Quelle Handbuch: steinigke.de)

Jannik startet mit einem neuen Projekt in der Projektadministration:

[Kap2_Projektadministration] | *Kap2_Projektadministration.JPG*

Abb 2-x: Anlegen des ersten Testprojektes

Im nächsten Schritt fügt er 6 Geräte der gewählten LED PARTY Lampe hinzu, z.B. in der Bühnensicht. Alle Lampen organisiert er gleichzeitig in einer Gruppe namens „Group LED PARTY“. Alle Namen können natürlich beliebig geändert werden. Glücklicherweise hat der hilfsbereite Nutzer Kleena schon ein passendes DDF in der DDF-Library erzeugt. Über die Suche mit dem Buch-Button ist es schnell unter dem Hersteller Eurolite gefunden und kann ausgewählt werden.

[Kap2_GeraetHinzufuegen_LEDParty] | *Kap2_GeraetHinzufuegen_LEDParty.JPG*

Abb 2-x: Anlegen der Geräte

Nun macht Jannik die erste Kontrolle in der „Kanalübersicht“. Alle Lampen (in der Darstellung unten sind wegen fehlender Scrolling-Möglichkeit im Screenshot nur die ersten vier sichtbar) sind dort ordentlich angelegt und er kann mit den linken Slider testen, ob alle Kanäle richtig funktionieren, indem er damit die DMX-Werte in der Ausgabe direkt beeinflusst. Wie bereits erwähnt: Diesen Slider in der Kanalübersicht niemals zum Programmieren von Cues verwenden!

[KanalUebersicht] | *Kap2_KanalUebersicht.JPG*.

Abb 2-x: Kanalübersicht

Schließlich schaut sich Jannik die Darstellung in DMXControl3 an. Mit den Farb- und Intensitäts-Control-Fenstern werden die Lampen manuell an- und ausgeschaltet und die Farbe vorgegeben. Lumos im Intensitäts-Control bedeutet, dass die höchste Helligkeit gewählt wird. Diese Änderungen werden sofort in der Bühnensicht angezeigt (und sind natürlich im Programmer verfügbar). Die in der Abbildung zu sehenden Navigations- und Zoom-Icons können mit der Kamerasteuerung ein- und ausgeblendet werden. Die Lampen können in der Bühnensicht einzeln oder mittels ihrem Gruppen-Icon ausgewählt werden

[Testprojekt LEDParty] | *Kap2_TestProjekt01_6grueneLEDParas.JPG*

Abb. 2-x: Die ersten 6 LED-PARs werden bedient.

Die standardmäßig eingeblendeten Controls für Position und Gobo machen bei den verwendeten Lampen keinen Sinn und sind daher ausgegraut, da DMXControl3 ja mitdenkt. In dieser Abbildung ist der Projektextplorer als eigenständiges Fenster eingeblendet. Dort sieht man, dass es 6 Geräte und eine Gruppe für die 6 Geräte gibt. Das Gerätesteuerungsfenster auf der rechten Seite wurde so

aufgeklappt, dass alle Eigenschaften und aktuellen Attribut-Werte zu sehen sind, insbesondere die Farbkanäle. Auch in diesem Fenster können die Werte durch direktes Eintippen geändert werden.

Klicke wie Jannik dort einfach mal etwas rum. Es kann ja nichts kaputtgehen!

Öffne das Fenster mit den Mastern und schaue, wo schon mal standardmäßig etwas passiert. Nicht zuletzt, einfach mal einen Effekt auf den Dimmer oder die Farben in der Gerätesteuerung per Drag&Drop ziehen. Zu den Effekten kommen wir aber später im Kapitel 4. Sie können im Gerätesteuerungspanel über das X-Zeichen auf der rechten Seite oder die Entf-Taste auch wieder gelöscht werden.

2.4. Generelle Entscheidungen

Dieses Beispiel war nun erstmal für den absoluten Einstieg gedacht. Welche weiteren Features DMXControl3 liefert werden Jana und Jannik nun anhand diverser Szenarien erforschen:

- Welche Art der Veranstaltungen (z.B. Theater oder Disco) werden unterstützt?
- Wieviel sollte vorab programmiert werden? Oder wird alles live gesteuert?
- Welche Programmier- und Steuerungsmöglichkeiten hat man als Anfänger oder als Experte oder etwas dazwischen?
- Was will ich eigentlich? Kann DMXControl3 alles tun, was letztens in der Show beim Konzert im Olympiastadion gesehen zu sehen war?
- Wie gestaltet man das Spiel mit Farben, wie aufwändig sind bestimmte Lichteffekte?
- Was ist dieser Timecode, wann braucht Jannik diesen?
- usw., usw.

Stefan und das DMXControl-Team wünschen nun den beiden und euch viel Spaß bei der Reise durch das DMXControl-Universum!

3. Kapitel 3: Nun geht es endlich los – die erste richtige Show

3.1. Showvorbereitung

In der Aula steht ein Konzert des HTW-Tonkollektivs, einem Chor mit Musikern an. Auf der Songliste der Rhapsody-Show stehen mehrere Lieder der Band Queen, die aber im Klassikstil mit einem Orchester interpretiert werden.

Daher hat sich Jana entschieden, das Lichtdesign dezent zu gestalten. Im Prinzip gibt es zu jedem Song ein bis drei statische Lichtstimmungen (Cues), die eine farbige Raumatmosphäre definieren.

Wir beginnen also mit einem relativ übersichtlichen, aber nicht mit einem allereinfachsten Szenario einer Show, um verschiedene grundlegende weitere Features von DMXControl3 vorstellen zu können.

Jana hat ihre Design-Vorstellungen aufgezeichnet und dabei zwischen der Bühne (inkl. vordere Wandflächen), den Seitenwänden und der Leinwand im Hintergrund unterschieden:

[Beispiel Design Jana Cue 12]

Abb. 3-1: Beispiel Design Jana Cue 12

Hier noch ein 2. Beispiel aus ihrem Regiebuch, im Finale soll etwas mehr Action sein.

[Beispiel Design Jana Cue 20]

Abb. 3-2: Beispiel Design Jana Cue 20

Es gibt ca. 30 solcher Bilder, die sie an Jannik übergibt. Jana hat sich keine Gedanken um die verfügbaren Lampen gemacht, geschweige denn um die Installation. Sie sagt: „Jannik, nun mach mal!“

Natürlich kann es nicht sofort mit der Show-Programmierung losgehen. Jannik schaut erstmal nach den verfügbaren Lampen vom Tonkollektiv und installiert diese mit anderen Studenten am Truss und auf dem Boden. Die dabei einzuhaltenden Sicherheitsmaßnahmen, elektrischen Randbedingungen der Stromkreise usw. werden wie in jeder seiner Shows umgesetzt, sollen aber hier nicht betrachtet werden. Er stellt fest, dass für die erste Show keine für die Aula passenden leistungsstarken Movingheads zur Verfügung stehen.

Wichtig ist erstmal, dass vor oder während des Aufbaus ein DMX-Verkabelungs- und Adressierungs-Plan erstellt wird.

- Jede Lampe hat einen Typ für den das passende DDF bereitgestellt werden muss.
- Jede Lampe hat eine DMX-Adresse.
- An den Lampen muss eingestellt werden, welcher Kanalmodus verwendet werden soll (dies entspricht dann auch dem zu wählenden DDF).



Die Verfügbarkeit der passenden DDFs ist entscheidend für die schnelle und korrekte Showprogrammierung. Ein nicht passendes DDF wird später ungewünschte Effekte hervorrufen.

Es wird also unbedingt eine Skizze und/oder eine Tabelle benötigt, wo die Typen, die DMX-Adressen, Eigenschaften und die ungefähren Positionen der Lampen aufgeführt sind.

[Beispielskizze Plan für Lampen-Setup]

Abb. 3-3: Beispielskizze Plan für Lampen-Setup (muss natürlich ausgetauscht werden)

Dieser Plan kann nun in der sog. Setup (Einrichtungs-)Phase in DMXControl3 in der Bühnensicht und mittels Patching umgesetzt werden.

3.1.1. Geräte und DDFs

Jedes Gerät unterstützt basierend auf seinem technischen Aufbau und der Firmware unterschiedliche Eigenschaften, die gesteuert werden können. Abgesehen vom GDTF-Ansatz gibt es

bisher noch keinen allgemeinen Standard zur softwaremäßigen Beschreibung von Lampeneigenschaften. DMXControl hat sein eigenes DDF-Format (device definition file) auf der Basis von XML definiert. Die Erstellung solcher Files kann mit einem normalen Text- oder XML-Editor gemäß der vorgegebenen Syntax und Semantik erfolgen. Mit dem DDF-Creator gab es auch einen Versuch, ein interaktives Tool anzubieten, aber leider wird das Tool aktuell nicht mehr gepflegt.

3.1.2. Bühnenansicht und Patching

Die Bühnenansicht ist der zentrale Teil der Show während der Einrichtung, des Programmierens und zum Teil auch während der Aufführung. Man könnte zwar alle Geräte auch im Projekt Explorer einrichten und sehen, aber in der Bühnenansicht hat man gleich eine grafische Darstellung mit 2D-Animation der dargestellten Farben, Gobos und Bewegungen. Man kann sich die Lampen ungefähr so wie im gerade vorgestellten Plan anordnen und hat somit immer einen guten Überblick, was diese gerade machen. Später wird erklärt, wie auch ein 3D-Visualizer zur Ansicht verwendet werden kann. Ein moderner Mensch würde von einer Art digitaler Zwilling sprechen, sagt Stefan.

Neue Lampen können über verschiedene Menüpunkt inkl. Kontextmenü angelegt werden.



Hier einige Tipps dazu:

- Wenn das richtige DDF nicht in der Liste auftaucht, sollte das entsprechende xml-File in das Geräte-Verzeichnis „Eigene DDFs“ kopiert werden.
- Hat man es nicht gefunden, lohnt ein Blick in die Cloud/DDF-Bibliothek. <https://ddf.dmxcontrol.de/home> . Auch im Setup-Menü gibt es neben der DDF-Suche einen entsprechenden Button („Buch-Icon“)
- Hat man mehrere Geräte vom gleichen Typ, können diese auf einen Schlag eingerichtet werden. Obwohl technisch nicht zwingend erforderlich, kann man zwischen den Geräten eine Lücke bzgl. der DMX-Kanäle lassen. Stefan sagt, dass es das Adressieren einfacher machen könnte, wenn man zum Beispiel seine Lampen in 10er-Schritten patcht (also nur Adressen wie 20, 30, 40, 70 usw. vergeben werden. Für seine eigenen Lampen hat man meistens sowieso immer feste voreingestellte Adressen.
- In den meisten Fällen ist es sinnvoll, gleich automatisch eine Gruppe für die Geräte des gleichen Typs anlegen zu lassen. Wenn man möchte, dass sich diese Geräte in verschiedenen Situationen immer „gleich verhalten“, kann man dann später die Programmierung über die Gruppe realisieren, statt immer alle Geräte einzeln auszuwählen. Mehr zu Gruppen findet man im Kap. x.x



Stefans Tipp: Platziere das Gruppen-Icon in die Nähe der Icons der einzelnen Lampen.

Table 3. DDF Ablage

[Kap3 DDF Ablage]

Für die eigenen DDFs gibt es einen vorbereiteten Ablageplatz. Diesen findet man am besten über das Windows-Menü unter dem Eintrag DMXControl3. Einfach die DDFs, die man sich selbst erstellt hat oder aus der Cloud geladen hat, in dieses Verzeichnis kopieren. Wenn man ein selbst erstelltes DDF für ein neues Gerät hat, weil dieses noch nicht in der Cloud verfügbar war, ist es natürlich sehr fair, wenn man dies selbst hoch lädt. Die Community wird es Dir danken!

3.1.3. Patching

Jannik freut sich, nun kann es ja los gehen mit der Programmierung. „Halt!“, ruft da Stefan. „Überprüfe sicherheitshalber noch mal die Kanalbelegung in der Kanalübersicht. Dort siehst Du wie die Kanäle der Lampen über das Universum verteilt sind. Du wirst zwar später nie mehr die DMX-Adressen benötigen, aber es können sich beim Setup Flüchtigkeitsfehler eingeschlichen haben, die später zu total unerwünschten Effekten führen könnten.“

Bitte prüfe daher immer, ob * sich Kanalbelegungen der Lampen ungewollt überschneiden (das kann in seltenen Ausnahmefällen gewünscht sein, aber dazu kommen wir später) * die Lampen auch wirklich die gewünschte Zahl der Kanäle belegen, also der Kanalmodus konsistent zum an der Lampe eingestellten Wert ist.



In unerwartet und selten vorkommenden Fehlerfällen kann man über die Kanalübersicht auch prüfen, ob die Lampen überhaupt reagieren. Dazu muss man einen DMX-Kanal auswählen und mit dem Slider auf der linken Seite entsprechende DMX-Werte zwischen 0 und 255 einstellen. Diese werden sofort an die DMX-Ausgabe weitergereicht (und gehen nicht über den Programmer). Dies ist die einzige Stelle in DMXControl3, wo der Nutzer mit wirklichen DMX-Werten arbeitet.

Aber nun zum Patching:

Was ist das für ein Projekt, in welchem nicht einmal neue Geräte hinzukommen und vorhandene Geräte hierfür eine andere DMX-Adresse erhalten müssen? Oder ihr ändert euren Aufbau und der bestehende Patch passt dafür nicht. Dann geht es natürlich direkt an das Ändern der DMX-Adressen.

zu ergänzen: Verschieben von Geräten in ein anderes Universe> <https://forum.dmxcontrol-projects.org/core/index.php?article/894-schon-gewusst-38-schnelleres-%C3%A4ndern-des-patches-bitte/>

Nachdem nun alles zu den DMX-Kanalbelegungen geklärt ist (zu komplexeren Geräten kommen wir später), müssen wir uns noch das Programmierkonzept überlegen, mit dem Jana's Design am besten umgesetzt werden kann. Unbewusst hat Jana ja die Geräte schon eingeteilt: * Chor- und

Orchesterbeleuchtung auf der Bühne (Frontlicht und Oberlicht) * Hintergrundbeleuchtung der Leinwand hinter dem Chor * Wandbeleuchtung neben der Bühne * Wandbeleuchtung an den Seitensäulen

Damit wird Jannik sofort klar, dass er seine Geräte in entsprechende Gruppen einteilen sollte und einen Teil der Programmierung nur noch über diese Gruppen tätigen sollte. (Für die Informatiker unter Euch: Das ist eigentlich eine klassische Dekomposition in einem Algorithmus, was zu mehr Struktur, Übersichtlichkeit und Effizienz in der Implementierung führt.)

Zum Teil entsprechen die Gruppen schon den Initialgruppen aus dem Gerätesetup (diese sollten dann ggf. nur noch umbenannt werden), zum Teil werden neue Gruppen erstellt in dem alle Gruppenmitglieder ausgewählt werden und eine Gruppe dafür angelegt wird. Eine Lampe kann problemlos parallel in mehreren Gruppen enthalten sein. Ebenso können bei DMXControl Lampen unterschiedlicher Typen in einer Gruppe sein. Das Programm versucht sein Bestmögliches, dass sich dann alle Lampen in der Gruppe gleichartig verhalten. Dennoch kann es kleine Unterschiede z.B. in der Farbdarstellung geben.

3.1.4. StageView - Bühnenansicht

Richtig Spaß macht die Showprogrammierung nur mit der Bühnenansicht. Jannik ordnet sich dort nun alle Lampen ungefähr so an, wie sie in der Aufbauphase installiert wurden:

- 6 LED PARs für Orchesterbeleuchtung (Gruppe Orchester)
- 6 LED PARs für Chorbeleuchtung (Gruppe Chor)
- 2 zusätzliche LED Washer für Chorbeleuchtung (Gruppe Washer)
- 2x4 LED PARs für die seitlichen Wände (Gruppe Säulen)
- 4 LED Bars für Hintergrundbeleuchtung der Leinwand (Gruppe Leinwand)
- 6 LED Bars für die Bühnenfront (Gruppe Bühne)
- 2 PAR Spots als Verfolger (Gruppe Verfolger)

Gerade wenn man viele Lampen hat, macht es Sinn, auf die Möglichkeiten zurückzugreifen, die Anordnung halbautomatisch zu beeinflussen. Man kann die Icons der Lampen

- gleichmäßig auseinander ziehen
- in bestimmten Formen anordnen, z.B. über Shortcuts
 - Strg+C - Anordnung im Kreis,
 - Strg+F - Anordnung entlang eines Pfades,
 - Strg+M Anordnung in einer Matrix
 - +Shift Icon-Beschreibung ignorieren
- sortieren, verschieben
- F10 - Camera-Control ein- und ausschalten,
- F7 - Reset der Ansicht -
 - ...

Bei Effekten ist es immer relevant in welcher Reihenfolge die Lampen in der Gruppe angesiedelt sind. (Das gilt insbesondere für das Fanning welches später erklärt wird.) Die Reihenfolge hat nichts mit der Positionierung in der Bühnensicht zu tun, sondern mit der Reihenfolge der Geräte im Projektextplorer.

3.2. Das Showkonzept

Wie erwähnt, denkt Jana im Theatermodus. Dort soll jede Aufführung total gleich sein, die Lichtstimmungen absolut optimiert, also die fertigen Lichtstimmungen im Prinzip nie mehr angefasst werden. Das hat seine Berechtigung und gegen dieses Konzept ist im Prinzip nichts einzuwenden.

Allerdings kommt Jannik eher aus dem Live-Bereich. Er ist es also gewohnt spontan zu reagieren, die Helligkeit oder auch Farben von Lampen mal spontan anzupassen. Das sture Abfahren einer Show aus der Konserve widerstrebt ihm (obwohl es natürlich bei bestimmten Events großen Sinn macht). Wie sollten wir diesen „Widerspruch“ auflösen? Natürlich durch einen Anruf bei Stefan.

Er macht den Vorschlag, mehrere Cuelisten in der Show zu benutzen. Eine Hauptliste, wo die Lichtstimmungen der einzelnen Songs hintereinander fest programmiert werden. Daneben gibt es je eine Cueliste für die Chor- und Orchesterbeleuchtung. Diese Cuelisten werden Executoren zugeordnet, damit man darüber auch live die Helligkeit der jeweiligen Lampengruppe anpassen kann.

[Cuelisten und zugeordnete Ausführungselemente]

Abb. 3-4: Cuelisten und zugeordnete Ausführungselemente

3.2.1. Fit for Cuelist

Jannik kennt bereits Cuelisten von den Pulten und fasst noch mal zusammen: Cuelisten sind eine Abfolge von Lichtstimmungen, also Cues (Szenen). Die Cues enthalten Eigenschaftswerte von einem oder mehreren Geräten (z.B. Dimmer 60% und rot). Was in einer Szene (Cue) gespeichert ist, siehst du, wenn du die jeweilige Szene in den Programmer lädst. Cuelisten können gestartet und gestoppt werden, die enthaltenen Cues haben oft eine (Ein-)fade-Zeit und eine Dauer.

Stefan ergänzt: Cuelisten können wiederholt und in Loops ablaufen – das stellt man im Mode-Menü ein. Mehrere Cuelisten können auch parallel ablaufen. Dabei sind Prioritäten und Gruppierungen zu beachten. Manchmal wird es wirklich knifflig, wenn verschiedene Cuelisten eine gleiche Eigenschaft einer Lampe überlagern. Diese komplexeren Aspekte werden später im Kap. x.x besprochen, nun wollen wir erstmal die Cuelisten für die Rhapsody-Show erzeugen.

Fragt man ChatPT, wie eine Show erstellt wird, kommt diese doch sehr allgemeine Antwort:

Erstellen von Abläufen: Definieren Sie den zeitlichen Ablauf der Lichtshow, indem Sie verschiedene Szenen in einer bestimmten Reihenfolge anordnen. Sie können Übergänge zwischen den Szenen festlegen, um einen sanften Übergang von einem Zustand zum nächsten zu ermöglichen.

Der Dirigent hat sich z.B. sehr fließende und keine ruckartigen Übergangszeiten zwischen den Cues gewünscht. Also hat Jannik eine sehr große Fadezeit (z.B. 20 Sekunden) in einigen Cues eingetragen.

Stefan hat den prinzipiellen Ablauf der Cue-Erzeugung in DMXControl3 aufgezeichnet (Abb. 3-x) und erklärt:

Um eine Lichtstimmung vorzubereiten, wählst du in der Bühnensicht im Idealfall immer eine Gerätegruppe aus. Dann füllt sich die Gerätesteuerung (Device Control) strukturiert mit allen Funktionen, die das Gerät mitbringt. Gleichzeitig werden die Schaltflächen in den Steuerungsfenstern (Control Panel) aktiv, immer jeweils im Kontext der verfügbaren Funktionen. Über diese Fenster kannst du nun eine einfache, statische Lichtstimmung einstellen. Aber aufgepasst: gespeichert ist an dieser Stelle noch nichts. Wird DMXControl 3 geschlossen, ist diese Lichtstimmung weg. Vielmehr ist es die Vorstufe, um die Lichtstimmung später richtig abzuspeichern.

Möchtest du wieder alles auf 0 setzen, klickst du in der linken Seitenleiste auf den Button [Programmer leeren (Clear programmer)].

image: Kap03_CueListErstellen.jpg[Grobes Szenario „Cues erstellen“]

Abb. 3-x: Grobes Szenario „Cues erstellen“ <Offen in Abb.: Update-Button, Clear programmer>

Nun gibt es für jede Cue in der Cueliste noch viele Parameter, die mit dem Triggern (also dem Auslösen) und dem Zeitverhalten zu tun haben. Eine Cue muss nicht einfach so plötzlich da sein und wieder weg sein. Im Gegenteil: Eine Cue hat gar kein Ende, sie hört nicht einfach auf. Sie wird irgendwann durch eine nachfolgende Cue überschrieben. Schauen wir uns den zeitlichen Ablauf an:



Der Lebenszyklus einer Cue kann bei der Showgestaltung ausgenutzt werden.

[Lebenszyklus einer Cue]

Abb. 3-5: Lebenszyklus einer Cue

Durch den Trigger wird die Zeit oder das Event zum Auslösen vorgegeben.

Es gibt verschiedene Trigger (manual, one shot, follow,..). Wie der Name es vermuten lässt, muss man bei „manual“ selbst z.B. mit dem GO-Botton zum richtigen Zeitpunkt weiterschalten. Dies ist der Standardfall. Hat man eine fixe zeitliche Abhängigkeit zwischen den Cues, bieten sich der follow- oder wait-Trigger an (siehe Abb. 3-x.). Gibt es feste Zeitpunkte zum Starten einer Cue verwendet man den rtc-Trigger (Absolutzeit) oder den timecode-Trigger. Letzteren kann man sich als konkreten Zeitpunkt innerhalb eines Songs vorstellen, dazu mehr im Kapitel Timecode-Player. DMXControl3 kennt auch ein (internes und externes) Beatsignal – dafür gibt es den Beattrigger, der sozusagen die Cues im Takt weiterschaltet. Der one-Shot-Trigger wird nur bei Effekten verwendet.

Table 4. trigger

Trigger-Typ	Beispiel	Beschreibung / Funktionsweise
manual	3	Szene wird bei Klick auf [Go] weiter geschaltet. Optionaler Wert: "3" (beim dritten Klick auf [Go] weiter schalten)

Trigger-Typ	Beispiel	Beschreibung / Funktionsweise
follow	5s	Wartet bis zum Ende des Einblenden (fade in) der voranstehenden Szene, lässt dann die "follow"-Zeit ablaufen.
wait	5s	Die "wait"-Zeit beginnt zu laufen, wenn die Trigger-Zeit der vorherigen Szene abgelaufen ist.
one shot	1	Gibt die Anzahl der Durchläufe eines Effekts wie zum Beispiel eines Kreis an, nach dem die vorherige Szene ausgefaded werden soll. One Shot Szenen sind grundsätzlich leer.
rtc	13:45:00	Die Szenen wird zur spezifizierten Uhrzeit eingefadet, hier um 13:45 Uhr.
timecode	00:26:05	Erwartet einen Timecode, sowohl aus einer externen Quelle als auch von der in der Cuelist enthaltenen Audioszene.
beat	1	Weiterschaltung der Szene durch ein Beatsignal, welches vom Speedmaster oder vom Manual Beat / Beat Tool kommt. Die Quelle für Beat-Signal in Cuelist Options unter "Beat Source"

In unserer ersten Show kommen wir allein mit dem manual-Trigger aus. Jannik ist genervt, dass er mit der Maus immer den kleinen GO-Button im Cuelist-Editor treffen muss. Aber Stefan sagt, hier greifen wir mal kurz mit dem Input Assingment vor. Darüber können wir mit Drag&Drop den GO-Button auf die große Leertaste ziehen. Nun ist alles bestens – so bequem hat Jannik noch nie eine Show bedient.

offen: <Screenshot>

Cuelist stoppen:

Jannik fällt auf, dass es nach dem Drücken des Stop-Buttons bei einer Cuelist immer eine gewisse Zeit zum „Rückspulen“ der Cues braucht, wie man im Fortschrittsbalken sieht. Er fragt Stefan, was da eigentlich passiert. Warum stoppt die Cuelist nicht einfach und fertig? Stefan antwortet: „Dass beim Zurückfahren beim Stoppen der Cues immer eine Zeit vergeht, hängt mit der eingestellten Release Time zusammen.“

Später ergänzen: Bei Tracking macht Random Mode keinen Sinn.



Nochmal der Hinweis: Es kann sehr sinnvoll sein, Cuelists mit nur 1 Cue zu füllen, wenn man diese z.B. mit einem Executor steuern möchte oder Prioritäten zwischen Cue(list)s ausnutzen möchte.

3.2.2. Grundlicht, Oberlicht hell genug?

Wenn irgendjemand auf der Bühne auftritt, solltest Du immer Grundlicht und Oberlicht zur Verfügung haben.

Im Beispiel besteht das Oberlicht aus 6 LED-Pars für die Orchesterbeleuchtung am Truss oben (hier geht es nicht darum, die einzelnen Gesichter der Musiker zu erkennen). Das Grundlicht oder

Frontlicht wird durch 6 weitere LED-Lampen am Truss gebildet, wovon 2 leistungsstarke Washer sind. Damit wird mit schrägem Licht von oben und vorn gesichert, dass die Chormitglieder inkl. ihrer Köpfe gut zu erkennen sind.

Die Lampen werden also zu zwei Gruppen „Chorlicht“ und „Oberlicht“ zugeordnet. Für jede Gruppe gibt es eine Cuelist, die jeweils nur eine Cue enthält. In dieser Cue wird die Helligkeit auf 100% gesetzt und die Farbe weiß-gelblich eingestellt, damit das Licht nicht so kalt wirkt.

Damit diese Cuelisten unproblematisch genutzt werden können, werden sie jeweils einem Executor zugeordnet. Damit kann man die Cuelist einmalig starten und hat den Vorteil, dass man die Helligkeit ähnlich zu einem Master stufenlos regeln kann während parallel die Haupt-Cueliste abgearbeitet wird.

zu ergänzen: <Bild mit Executoren>

3.3. Die Generalprobe und Aufführung

Die Cuelisten der Show sind fertig programmiert, alles ist schon mal im Schnelldurchlauf getestet worden, nun geht es an die Generalprobe. Jana und Jannik sitzen auf sich allein gestellt im Rang im FoH (Front of Head), also hinter dem DMXControl-Laptop.

Es geht los, die Spannung steigt, es sind über 60 Leute im Orchester und auf der Bühne im Chor. Jannik startet die Eingangsszene, die den letzten Zuspätkommern im Publikum noch erlaubt ihren Platz zu finden. Ein Zeichen des Dirigenten, jetzt geht es richtig mit der ersten Cue für den ersten Song ab.

Jana und Jannik erkennen, dass der Chor zu dunkel beleuchtet ist. Was ist zu tun? Jannik zieht den Chor-Executor hoch, Jana will aber ein Update in der Cue, damit morgen bei der Aufführung alles sofort perfekt ist. Jannik ist Informatiker und denkt: „Never change a running system“ – schon gar nicht während einer Aufführung! Es kommt zum ernsthaften Streit, so dass sie fast den Einsatz der nächsten Cue verpassen, wo das gleiche Spiel mit anderen Attributwerten weitergeht. Jannik fühlt sich als Anfänger bei DMXControl3 nicht sicher genug, solche Änderungen live zu machen. Er fürchtet Seiteneffekte, gegenseitige Beeinflussungen der Cues, Überlagerungen, aber auch Programminstabilitäten. Nichts ist peinlicher, als wenn während der Aufführung das Licht ausgeht...

Als er später Stefan davon erzählt, lacht dieser: Wir haben den blind mode, wir haben den Update-Button, wir haben verteilte GUI-Arbeitsplätze – das geht schon alles, aber man muss es halt beherrschen - as erkläre ich Dir später.

Zurück zur Show: Grundsätzliche Änderungen der Cues wurden dann nachträglich durchgeführt. Aber Jannik hat ja immer noch seine Executoren um stets einzugreifen und so lief während der Aufführung alles perfekt ab.

Nur Jannik war ja auch aus seiner Perspektive genervt, dass lichtmäßig wegen der statischen Szenen von Jana so wenig während der Show passiert ist (bis auf drei Ausnahme-Cues). Was niemand auf der Rechnung hatte: Nach der Show hat das Publikum vor Begeisterung getobt und Zugaben gefordert. Dafür gab es keine Vorbereitung und kein Konzept von der Lichttechnik!

Selbst der Dirigent musste spontan entscheiden, welche Songs noch mal angespielt werden. Jannik standen einerseits die Schweißperlen auf der Stirn, andererseits ist das doch die Gelegenheit, live und spontan sein eigenes Lichtkonzept zu realisieren und Jana zu zeigen, wie es auch gehen kann □

[Aufführung]

Abb. 3-6: Während der Aufführung

Jannik hat einfach improvisiert, hier mal einen Chaser auf eine Gerätegruppe ziehen, dort mal live in der Bühnensicht mittels Farbcontrol geändert oder einen Sinus-Effekt an ein Attribut gehängt. Dazu gibt es in den folgenden Kapiteln mehr Infos.

Über das Ergebnis waren alle happy. Jana und Jannik haben sich angelächelt.

4. Kapitel 4: Eine Party

Jana hat einen runden Geburtstag, den sie richtig feiern möchte. Dazu hat Jannik eine Partylocation gebucht. Leider gibt es dort kein „richtiges“ Disco-Licht. Also werden kurzerhand die 6 LED PARTY-Lampen sowie 4 relativ kleine Movingheads mitgebracht. Das Problem ist, dass auf der Party alle Leute Party machen wollen und niemand hinter einem Lichtpult stehen möchte. Es soll schon möglichst viel automatisch ablaufen. Jannik wird die Möglichkeiten von DMXControl3 dazu in diesem Kapitel vorstellen.

Natürlich haben etliche Disco-Lampen einen eingebauten Sound-Modus, der z.B. zufällig Farben ändert, wenn das eingebaute Mikrofon irgendwelche Lautstärkeänderungen erkennt oder Beats vom Schlagzeug identifiziert. Diesen eher „primitiven“ zufälligen Modus zu nutzen, würde aber an der Ehre von Jannik kratzen.

In diesem Kapitel geht es dagegen hauptsächlich um * Beat-Trigger * Endlos-Cuelisten

Aber dazu später mehr. Erstmal starten wir mit den sogenannten (einfachen) Effekten und der Struktur einer Show für eine Party. Bisher hat Jannik mit eher statischen Lichtstimmungen und einer Haupt-Cueliste gearbeitet.

[PartyProgrammierung]

Abb. 4-1: Party Programmierung

4.1. Baukastensystem für Cuelisten

Jannik hat absolut keine Idee, wie die Show zu Jana's Party programmiert werden sollte. Er will unterschiedliche Farben, etwas Lauflicht-artiges, die Moving Heads sollen sich scheinbar zufällig bewegen und das soll alles mit möglichst geringem Aufwand bedienbar sein.

Also wird Stefan wieder um seinen Rat gefragt. Er meint, dass man maximale Flexibilität durch ein „Baukastensystem“ aus verschiedenen Cuelisten erhält. Also keine fertige Gesamt-Cueliste, sondern viele einzelne Cuelisten, die für verschiedene einzelne Eigenschaften oder auch Gerätegruppen zuständig ist. Ebenso kann es Cuelisten geben, die verschiedene Typen von Effekten beinhalten. Diese Cuelisten können nun manuell oder automatisch zufällig aufgerufen werden.

Ein mögliches Szenario wäre:

- Alle 5 Minuten eine verschiedene Grundfarbe aufrufen
- Alle 2 Minuten wechselt der Lichteffekt
- Verschiedene Typen von Effekten: im Chaser laufen 1,2,4 oder 8 Lampen, ebenso können über verschiedene Dimmerwerte interessante Effekte erreicht werden
- falls Moving Heads vorhanden sind: Flower Effekte u.ä. über Gobowechsel
- Die Effekte werden alle 10 Minuten weitergeschaltet

Diese Baukastenelemente können z.B. manuell über verschiedene Input-Trigger gestartet werden. So könnte ein Softdesk erstellt werden, bei dem Buttons für die Cuelists angelegt wurden.

Will man es mehr automatisch haben, können diese Cuelisten in übergeordneten Cuelisten angeordnet werden. Solche Cuelisten können im Endlos-Mode betrieben werden. Die aufgerufenen Cuelisten werden z.B. alle x Minuten weitergeschaltet. Ebenso kann auch der Random-Mode genutzt werden um mehr Zufall in die Show zu bringen.

4.2. Fit for (einfache) Effekte

Statische Lichtstimmungen können auf Dauer langweilig sein. DMXControl3 bietet selbstverständlich die Möglichkeit, auch dynamische Effekte wie zum Beispiel Bewegungen für Moving-Heads oder Lauflichter auf den Dimmer, der Farbe und nahezu allen anderen Gerätefunktionen anzuwenden. Dies erfolgt allesamt mit Hilfe des Fensters Effekte und Filter (Effects and Filters). Die mitgelieferten Effekte sind dabei in verschiedene Kategorien unterteilt, die darauf basieren, welche Effekte bzw. Filter bei welchen Gerätefunktionen anwenden kannst.

- Möchtest du einen der Effekte anwenden, ziehst du diesen zum Beispiel per Drag & Drop auf die gewünschte Gerätefunktion im Fenster Geräteeigenschaften.
- Jeder neu eingefügte Effekt wird mit einer Standardkonfiguration angewendet. Um diesen auf deine Bedürfnisse anzupassen, musst du am Ende durch Ausprobieren herausfinden, welche Werte für die verschiedenen Parameter eines Effekts gut aussehen. Bitte beachte, dass sich bei allen numerischen Parametern ebenfalls das Fanning (Kap. 5) verwenden lässt.
- Das Hinzufügen von dynamischen Effekten zu einer Szenenliste funktioniert auf die gleiche Weise wie bei statischen Lichtstimmungen. Hier gibt es seitens DMXControl3 keine weitere Differenzierung. Dem entsprechend kannst du auch statische und dynamische Elemente einer Lichtstimmung in einer Szene mischen, sofern es dem grundlegenden Konzept deines Projekts hilfreich ist. Konkret bedeutet dies, dass du sowohl eine Bewegung als auch statische Werte für den Dimmer, zum Beispiel in Form eines Fanning in nur einer Cue speichern kannst.

[JanniksParty]

Abb. 4-2: Janniks Party

4.2.1. Effekte über Effekte

*Lauflicht, Chaser *

Die Lampen auf denen das Lauflicht laufen soll, in eine Device Gruppe zusammen packen. Diese Gruppe in der Stage View auswählen. Im Device Control Fenster eine Farbe auswählen und dann aus den „Effekt und Filtern“ einen Chaser auf den Dimmer der Gruppe im Device Control ziehen. Dann sollte das Lauflicht schon laufen. Nun kannst Du das dann als Szene speichern.

Stroben

Hier funktioniert es im Prinzip genauso. Du wählst die Gruppe aus und stellst einen Strobe-Faktor beim Dimmer im Device Control ein.

Livestream "Effekte über Effekte über Effekte". In diesem Livestream haben wir ab 1:25:00 mal gezeigt, wie flexibel der Circle-Effekt ist. Link: https://www.youtube.com/watch?v=cl6z3rV7_jI&t=7693s

zu ergänzen: 1D-Effekte und 2D-Effekte

4.2.2. Effektvisualisierung

Um besser zu verstehen, was die verschiedenen Parameter eines Effekts bewirken, kannst du dir die Effektvisualisierung (Effect Visualizer) öffnen. Sobald du den Effekt in der Gerätesteuerung auswählst, siehst du die aktuelle Vorschau. https://wiki-de.dmxcontrol-projects.org/index.php?title=Effect_Visualizer_DMXC3

4.3. Sound to Light – auch mit DMX?

zu ergänzen

4.4. Farben und Colorlisten – Was ist das denn nun schon wieder?

Unter den Effekten gibt es auch den Colorchaser, der standardmäßig mit bunten Farben aus dem Spektrum agiert. Jannik ist das aber zu bunt und zu grell – er möchte mit dezenteren Farben arbeiten. Dazu kann man den Effekt mit einer Colorlist (Farbliste) parametrisieren. Aber wie kommt man zu dieser Farbliste?

Diese gehört zur Kategorie „Ordnungsliste“ in den Verzeichnissen im Projektextplorer. Man erstellt in diesem Verzeichnis ein neues Objekt, gibt ihm den Typ Farbliste und kann dann sein eigenes Set an Farben definieren. Super simple, nur anfangs nicht leicht zu finden, denkt sich Jannik.

Und vergessen Sie nicht die Möglichkeit, mit Cues zu arbeiten, die in einer Schleife ausgeführt werden. Innerhalb der Cues geben Sie dem ersten Cue die Werte 255;0;0 # 0;0;255 für Farbe und dem zweiten 0;0;255 # 255;0;0, wenn der Chaser für jedes Gerät alternativ sein soll in der ausgewählten Gerätegruppe. Übrigens funktioniert das sinngemäß auch gleichermaßen mit Gobo-Listen.

4.5. Softdesk - der Start

<to do: Einfache Konzepte Softdesk beschreiben, wo noch keine tiefe Kenntnis über das Input Assignment erforderlich ist. z.B. (Toggle)Button ansteuern

5. Kapitel 5: Karneval

Dieses Kapitel betrifft Veranstaltungen wie Karneval, Musical usw. wo man eine hohe Kombination von vorbereiteten Cuelisten, Timecodelisten und Live-Eingriff hat. In Ergänzung zu den vielen Möglichkeiten aus den vorigen Kapiteln werden hier super Dinge vorgestellt, die wirklich nur mit einer computerbasierten Lichtsteuerung möglich sind. So werden Effekte nach mathematischen Algorithmen auf diverse Lampen automatisch umgerechnet. Da der Computer die Berechnungen für die einzelnen Lampen-Eigenschaften macht, minimiert sich der Programmieraufwand dadurch dramatisch! Wo man ansonsten zig-Cues benötigen würde, kommt man nun mit wenigen Cues aus um eindrucksvolle Lichtstimmungen und -Effekte zu zaubern.

Als eindrucksvolles Beispiel erklärt Stefan zuerst verschiedene Anwendungen für das Fanning. Denn Jannik möchte wissen, wie man es programmiert, dass die Beams von Moving Heads fächerförmig aufgestellt werden, ... und am besten noch mit einem gleichförmigen Farbverlauf – das wäre doch cool!

5.1. Fanning

Die in der folgenden Abbildung symbolisch dargestellten 9 Moving Heads sollen also in der Pan-Position gleichmäßig „verteilt“ werden. Außerdem ist ein fließender Farbverlauf von blau zu rot gefordert. Der Begriff „Fanning“ steht nicht nur für Ventilator, sondern auch für Fächer. Damit kommen wir zur Definition: Die Werte einer Geräte-Eigenschaft werden aufgefächert.

[Fanning]

Abb. 5-1: Fanning von Farben und Pan-Position

Noch genauer ausgedrückt: Die ausgegeben Werte für die Eigenschaft werden mathematisch „gleichverteilt“ und passen sich automatisch an die ausgewählten Geräte an (z.B. innerhalb einer Gruppe). Das Fanning ist also vorteilhaft mit Gerätegruppen zu verwenden. Je nach Anzahl der enthaltenen Geräte werden die Werte intervallmäßig aufgeteilt. Dies funktioniert für fast alle Eigenschaften, also z.B. Helligkeit, Farbe, Position. Die Zuweisung des Fanning-Features erfolgt in der Gerätesteuerung, dort wird in diesem Fall nicht mehr ein statischer Helligkeitswert eingetragen, sondern ein „arithmetischer Term“ mit Fanning-Operatoren in textueller Form, z.B. 10 > 90

Damit drückt man aus, dass sich die Helligkeitswerte in der Gruppe zwischen den Werte 10 und 90 gleichmäßig aufteilen. Dabei wird die Reihenfolge der Geräte aus der Gruppendifinition abgeleitet, also ggf. sollte man im Projektextplorer unter Gerätegruppen die Geräte neu sortieren, ansonsten sind Reihenfolgeprobleme möglich. Andererseits passt sich das Fanning automatisch an, wenn man Geräte zur Gruppe hinzufügt oder löscht.

Die verschiedenen Fanning-Operatoren sind in der nachfolgenden Tabelle dargestellt.

Table 5. Fanning-Operatoren (todo: harte Leerzeichen einfügen)

Bereich	Kurzbeschreibung	Operatoren
linear	Gleichmäßige zu- oder abnehmende Werte	< > <> ><
alternierend	Abwechselnde Werte	# ##
symmetrisch	Spiegelbildliche Aufteilung der Werte	(senkrechte Balken)
zufällig	Zufällige Ausgabe von Werten innerhalb des angegebenen Wertebereichs	? ??
gekreuzt	<zu ergänzen>	~< ~> ~<> ~><

Wenn der Fanning-Operator doppelt angegeben wird (also z.B. ## statt #), wird die Gruppe in zwei Hälften aufgeteilt in denen jeweils das Fanning Muster angewendet wird. Damit erreicht man eine gleichmäßige Aufteilung von der Mitte nach außen.

In der folgenden Abbildung erkennt man Fanning

- beim Dimmer/Helligkeit (MSP1...MSP8)
- in der Farbe (MSP9...MSP16)
- in der Tilt-Position (DMH1...DMH8)

[Fanning-Beispiele]

Abb. 5-1: Fanning-Beispiele

Falls die gleichmäßige Aufteilung zwischen zwei Werten nicht ausreicht, kann man die Fanning Operatoren auch verketteten, siehe nachfolgendes Symbolbild. Allerdings darf immer nur der gleiche Operator in der Kette verwendet werden.

[Fanning-verkettet]

Abb. 5-2: Fanning-verkettet

[Fanning-verkettet Bsp]

Abb. 5-3: Fanning-verkettet

Jannik hat da noch eine Frage an Stefan: Was ist eigentlich der Unterschied zwischen Fanning und Fading?

Stefan: Naja, beim Fading wird die Änderung einer Eigenschaft über den Zeitverlauf gestreckt. Beim Fanning werden die Werte einer Eigenschaft auf eine Gruppe gleichmäßig aufgeteilt. Wenn Du beide Dinge kombinierst, kannst Du Dir selbst super Effekte zusammenbauen.

5.2. Audio-Cues

Bei verschiedenen Veranstaltungen wie Weihnachtsshows, Musicals aber auch Karneval wird oft

Musik aus der Konserve eingespielt. Das bedeutet eigentlich, dass man die Lichtshow für solche Stücke schon Milli-Sekunden genau vorbereiten kann. Natürlich könnte ein Profi das auch live hinbekommen, aber dazu braucht man musikalisches Talent und Taktgefühl wie z.B. der Schlagzeuger – aber der sitzt ja nun mal während der Show nicht hinter dem Licht-PC.

Jannik weiss, dass man als Cuelist-Trigger für solche Situationen den Timecode einsetzen kann. Man kann sich mittels externer Tools wie z.B. Audacity die Waveform und die exakte Zeit (also den Timecode) für besondere Events anschauen, bei denen man z.B. sekundengenau eine Cueliste starten oder stoppen möchte.

[Specialcue]

Abb.5.x: Einfügen einer Audiocue

Für dieses Konzept gibt es die Specialcues „play“ and „stop“ Audiocue die Jannik zu Beginn und am Ende der Cueliste hinzufügt. Mittels der Timecodes (die man sich vielleicht auf einem Zettel notiert hat), kann man dann über entsprechende Timecode-Trigger in der Cuelist die vorher erstellen Cues zum richtigen Zeitpunkt starten.

[AudioCue]

Abb.: 5.x: Specialcue „Audio“ in DMXControl

Auf diese Weise erreicht man, dass die Lichtstimmungen zum entsprechenden Song bei jeder Aufführung absolut gleich sind.

5.3. Timecodeplayer

An dieser Stelle hakt Stefan ein und zeigt Jannik den seit DMXControl 3.3 existierenden Timecode-Player. Im Projektextplorer gibt es jetzt den neuen Objekttyp Timecodeshows. Er sagt: Dieses neue Objekt ist kein reiner Audioplayer mehr (wie man ihn aus DMXControl2 kennt), sondern das Teil kann viel mehr.

Das beginnt damit, dass Jannik im Audiotrack nicht nur die normale Waveform sehen kann, sondern es gibt auch die Darstellung als sog. Wasserfallmodell – dort sieht man die Beats besser, mit den Reglern links im Audiotrack kann das Highligthing besser angepasst werden, es werden also unterschiedliche Filter herausgehoben.

[Waveform]

Abb. 5.x: Waveform

[Wasserfall]

Abb. 5.x: sog. Wasserfallmodell des gleichen Songs

Zuerst muss Jannik einen Audiotrack anlegen. Das geht im Projektextplorer, aber auch in der geöffneten Timecodeshow. So kann Jannik z.B. eine mp3 Datei aus dem Ordner „Files“ im Projektextplorer per drag& drop auf die Timecodeshow raufschieben. Prinzipiell könnte man auch weitere Audiofiles hintereinander auf den Audiotrack schieben. Die Audioschnipsel können in der

Länge beschnitten und beliebig positioniert werden – so wie man es auch von (Midi-)Sequenzern kennt.

[Audiotrack einfügen]

Abb. 5.x: Steuerungsfenster Audiotrack

Im Audiotrack kann Jannik mit dem Hand-Icon (rechts oben in obiger Abb. 5.x) die Lautstärke einstellen, die anderen Icons werden im Bild erklärt (tbd.).

[Kap5 TCP Wasserfall mehrere]

Abb. 5.x: mehrere Audioschnipsel in einem Audiotrack

Nun will Jannik wissen, wie man dies zur Lichtsteuerung nutzt. In DMXControl2 hatte man ja an einzelnen Stellen in der Audiospur grafische „Markierungen“ gesetzt, denen dann jeweils eine Cue zugeordnet wurde. Jetzt ist das Konzept enorm erweitert worden, erklärt Stefan. Es gibt nun 3 Track-Typen in der Timecodeshow:

- Audio Track – wie gerade besprochen zur Sichtbarmachung der Audiospur
- Cue list Track(s) – es kann eine oder mehrere davon geben, in denen die verwendeten Cuelisten wie „Videoschnipsel“ angeordnet werden können
- Executor Track(s) – diese Tracks ermöglichen, Signale und Befehle die man sonst mit Executoren ausführen würde, in die Timecodeshow zu integrieren

[Kap5 5 TCP Cuelist einfügen] Abb. 5.x: Cuelist-Track

Kommen wir erstmal zum Cuelist-Track. Hier sieht Jannik einen großen grauen Block, das ist die Cue Liste. Die Länge des Kastens ist Gesamtlänge der Cuelist, also die Summe aller Cue-Dauern inkl. ihrer Fadezeiten, Releasezeiten usw.. Die Länge eines Audioschnipsels kann grafisch angepasst werden.

[Synchronisation]

Abb. 5.x: Synchronisation zwischen Cuelist-Editor und Cuelist-Track

Jannik findet beeindruckend, dass die Synchronisation zwischen Cuelist (in Cuelist Editor) und Cuelist-Track im Timecodeplayer in beide Richtungen zwischen Text und Grafik problemlos funktioniert. Ändert er in einem Tool einen Wert, wird das sofort im anderen Tool aktualisiert. Jannik will wissen, ob das mit jeder Cueliste funktioniert?

Nein, wirft Stefan ein. Das Konzept der Timecodeshow ist explizit auf Timecodes bzw. Timecode-Trigger zugeschnitten, daher ja auch der Name. Jede Ausführungszeit ist immer eine Timecode-Marke! Das Konzept würde bei wait, follow, manual-Trigger in der Cueliste gar nicht funktionieren, da diese die Zeitsteuerung durcheinanderbringen würden. Ebenso macht nur der Einmalig-Mode in der Cueliste Sinn – ganz klar, dass der random-Mode in eine Timecode-Show in 99,99% der Fälle nur Chaos anrichten würde. Vielleicht gibt es ja doch einen sinnvollen Anwendungsfall für die 0,01%? Das prüft Jannik später...

Der Timecodeplayer arbeitet jetzt zur Vereinfachung mit dem Timecode auf Millisekundenbasis.

(Bei anderen Timecode-Triggern außerhalb der Timecodeshow kann natürlich nach Belieben auch weiterhin die Frame-Zählung angewendet werden.)

[Kap5 7 TCP Cuelist mehrere]

Abb. 5.x: Mehrere Cuelist-Tracks

Es kann auch mehrere Cuelist Tracks geben, die dann parallel arbeiten. Das entspricht dem Konzept, dass man mehrere Cuelisten im DMXControl3 parallel ausführen kann und kann sinnvoll genutzt werden, wenn man seine Cuelisten nach dem Bausteinprinzip aufgebaut hat.

Wie erwähnt, stellt sich Jannik die Cuelist-Tracks wie Videoschnipsel vor, die synchron zum Audio eingeblendet werden. Nun fragt er aber Stefan, was die Executor-Tracks sind?

[ExecutorTrack]

Abb. 5.x: Executor Track

Stefan schwärmt – das ist wirklich ganz neu und kann bei der Showprogrammierung eine große Zeitersparnis bringen – wenn man es beherrscht.

Stell Dir vor, sagt Stefan, Du hast eine Live-Show die Du mittels Executors steuerst. Also wie im Kapitel „Executoren“ erklärt, ist jedem eine Cueliste zugeordnet und Du steuerst live die Intensität, Geschwindigkeit oder auch das Starten, Stoppen und Pausieren von Cuelisten. Also normale Aktionen einer normalen Liveshow.

Das Tolle am Executor-Track ist nun, dass Du die Aktionen eines Executors aufnehmen („recorden“) kannst. Wenn der rote Aufnahmebutton gedrückt ist, bedienst Du die Show zum Audiofile wie in der Liveaufführung und hast es hinterher zum Erneuten Abspielen im Kasten. Es werden also die Aktionen eines Executors inkl. der Faderbewegungen, Start/Stop-Buttons usw. aufgezeichnet. Du kannst natürlich hinterher ein Finetuning machen. Also in diesem Konzept müssen Cuelisten nicht explizit über einen Cuelist-Track eingefügt werden, sondern Du kannst sie mit einem Executor Track an beliebigen Stellen mehrfach starten!

[Kap5 9 TCP ExecutorTrack2]

Abb. 5.x: Nutzung mehrerer Executor-Tracks

Wow! Jannik ist erstmal erschlagen von den vielen Features und wird es nun erstmal fleißig mit dem Timecodeplayer trainieren.

Aber nun hat Jannik doch noch eine Frage an Stefan: Kann ich den Timecodeplayer aus einer Cueliste starten, also ähnlich der Audio-Cue? Stefan überlegt: Nein, das ist so nicht vorgesehen, denn der Timecodeplayer soll ja selbst der „Master“ sein und Cuelisten starten. Aber mir fallen ein paar Dinge aus der Trickkiste ein, wenn Du die Timecodeshow nicht manuell starten möchtest: Du kannst das Input Assignment nutzen - dort gibt es ein entsprechendes Node (Wrapper → Timecodes → Timecode-Show), um eine Timecodeshow zu starten. Alternativ kann man eine TimecodeShow auf einen Executor legen mit diesem und kannst Du die Show starten.

Link zum Youtube-Tutorial: https://www.youtube.com/watch?v=zOHG_S43qoI

5.4. Softdesk – der Start

6. Kapitel 6: Licht im Club

Zufällig ist Stefan in einer Disco für das Licht zuständig. Er hat dort natürlich schon sehr viele Effekte mit DMXControl3 programmiert. Für eine Urlaubsvertretung bittet er Jannik das Geschäft für zwei Wochen zu übernehmen. Jannik bekommt natürlich eine umfangreiche Einführung in das existierende Konzept. Aber einige neue Dinge darf er auch neu einführen...

6.1. Fit for Cuelist als Profi

Den grundsätzlichen Umgang mit Cuelisten hat Jannik bereits gelernt. Cuelisten bieten verschiedene selbsterklärende Ausführungsmodi (einmalig, wiederholend, ...) die man im Cuelist-Editor einstellen kann. Für eine richtige professionelle Show gibt es aber viel mehr Möglichkeiten. Wusstest Du, dass Cuelisten Prioritäten haben können?

Ein Beispiel: Jannik hat die Show komplett programmiert. Nun kommt der Dirigent und möchte einige Lampen doch stärker gedimmt oder sogar ausgeschaltet haben. Um alle Cues zu ändern bleibt keine Zeit mehr. Was tun in der Not?

Stefan empfiehlt eine neue Gruppe der betroffenen Lampen zu bilden. Dann soll Jannik eine Cueliste für diese Gruppe erstellen, wo nur der Dimmer auf 100% gesetzt wird. Diese kann mit einem Executor gestartet werden. Erstmal passiert aber gar nichts, wenn man den Regler im Executor bewegt, weil ja die Originalcue schon (hohe) Helligkeitswerte gesetzt hat. Wenn nun die neue Cue eine höhere Priorität bekommt (Projektexplorer → Cuelist suchen → Eigenschaften), dann tritt der gewünschte Effekt ein. Je größer die Zahl, desto höher die Priorität, es sind auch negative Zahlen erlaubt. Standardmäßig hat jede erzeugte Cuelist die Priorität Null.

[CueList Konzepte]

Abb. 6.1: Erweiterte Steuerungsmöglichkeiten von Cuelists

6.1.1. Gruppen von Cuelisten

Jannik hat folgenden Anwendungsfall: Wenn ich dort im Softdesk den Toggle-Button "Saal warmes Weiß" drücke, dann starte ich eine Szenenliste, welche meine PAR anspricht und auf weiß stellt. Wenn ich dann den Toggle-Button "Rot" aktiviere, möchte ich, dass der Button "Saal warmes Weiß" und die verknüpfte Cuelist beendet wird.

Aber wie verknüpfe ich das Ganze?

Stefan antwortet: Hast Du die Szenenlisten die nicht zusammen laufen sollen in eine Szenenliste-Gruppe (Cuelist- Group) gepackt ? Die Eigenschaften dieser Szenenliste-Gruppe musst Du dann noch auf Single list running stellen. Dann beendet die neu gestartete Szenenliste die vorherige usw.

Über das Group Handling kann ich dann halt einstellen, wie die Gruppen sich zueinander verhalten sollen. Hierdurch gibt es sehr viele Kombinationsmöglichkeiten, um Effekte und Filter anzuwenden.

Also, das Handling von Cuelist-Gruppen ist Teil des Gesamtdesigns der Show. Genauso, wie man sich vorher überlegen muss, welche Geräte und Lichtstimmungen man haben möchte, sollte man sich vorher einen Plan machen, welche Cuelisten wie miteinander zusammenhängen. Das wird insbesondere durch die verfügbaren Optionen bestimmt:

- Nur eine aus der Gruppe ausführen (alle anderen werden damit gestoppt)
- Alle aus der Gruppe gemeinsam starten/stoppen
- Master-Cueliste startet/beendet alle

Jannik möchte gern dafür plausible Beispiele genannt haben. *Stefan: ???*

6.1.2. Spezielle Cues

Im Menü „Einfügen“ im Cuelist-Editor gibt es den Menüpunkt „Special-Cue“. Hier kann man u.a. sogenannte Audio-Cues einfügen, wie schon im Kapitel 4.x erwähnt.

Eine andere sehr interessante Form ist das Einfügen von existierenden Cuelisten in die gerade bearbeitete. Aus Informatikersicht kann man sich das wie den Aufruf einer Funktion/Prozedur vorstellen. Man kann also eine existierende Cueliste an beliebigen Stellen immer wieder aufrufen.

Noch zu erklären: HTP, LTP, LoTP

6.2. Matrizen

Bei Matrizen denkt man ja zuerst an große LED-Matrizen wie z.B. eine Video-LED-Wand.

[LED-Matrix (Quelle: Steff)]

Abb. 6.x: LED-Matrix (Quelle: Steff)

Auch solche „Geräte“ können mittels Adapter letztendlich über DMXControl3 gesteuert werden, so ist es denkbar dass jedes Pixel wie ein RGB-Gerät mit 3 Kanälen gesteuert wird. Für die abgebildete Matrix mit 10 Reihen und 15 Spalten werden also 450 DMX-Kanäle benötigt. DMXControl3 stellt etliche Matrix-Effekte zur Verfügung, die dann direkt für das Matrix-Gerät aktiviert werden können (z.B. Plasma, Text, Random, Boom, usw.). So etwas ist sicherlich auch eine nette Anwendung in einem Club, aber hier soll es um folgenden Anwendungsfall gehen, wenn keine eigentliche LED-Matrix verfügbar ist:

Jana wünscht sich, dass eine Art Kaminfeuer-Atmosphäre im Club herrschen soll. Das könnte man natürlich mit zufälligen Farben im orange-rot-gelb Spektrum irgendwie erreichen, aber würde viel Arbeit machen. Stefan gibt wieder einen wertvollen Tipp: Nutzt einfach den Fire-Matrix-Effekt!

Alle Lampen, die irgendwie RGB können werden zu einer „künstlichen“ Matrix definiert. Hat man 24 Lampen, ist es egal, ob diese in 6x4 oder 2x12 oder 1x24 angeordnet werden. Wenn ein solches virtuelles Matrix-Gerät definiert wurde, kann dafür jeden verfügbaren Matrix-Effekt starten.

Man benötigt dafür das virtuelle Matrix-DDF (unter generic devices), welches entsprechend der gewünschten Dimension konfiguriert wird. Danach wird jeder einzelne Punkt der Matrix mit einem der verfügbaren Geräte gepatcht. Natürlich bleiben die Geräte parallel in ihren sonstigen

Gerätegruppen und Cuelisten erhalten. Man hat nur eine zusätzliche Möglichkeit, tolle Effekte zu starten!

Auf jeden Fall ist Jana überrascht, wie schnell der Club in einer Kaminfeuer-Lichtstimmung ist!

Für weitere Details hilft auf jeden Fall dieser Wiki-Beitrag weiter: https://wiki-de.dmxcontrol-projects.org/index.php?title=Lektion_23_Tut3

6.3. Input Assignment

< noch im frühen Entwurfsstatus >

Die Eingangszuweisung (Input Assignment) hatte Jannik schon kennengelernt, als es ihm zu nervig war mit dem Mausklick die Go-Taste der Cuelist zu treffen. Stefan hat erklärt, dass man diese Aktivität z.B. auch mit der Leertaste des Laptops auslösen kann und hat dazu eine Verbindung (connection) im Input Assignment per Drag&Drop hergestellt.

Genaugenommen ist damit das Konzept des Input Assignments schon charakterisiert. Man definiert sich beliebige Eingangsgrößen (events) um irgendwelche Aktionen zu triggern.

Was gibt es da noch so an Events außer den Tastaturevents vom Laptop? will Jannik wissen. Die Vielfalt der Antwort überrascht ihn: Das können Signale von externen DMX-Pulten sein oder auch MIDI-Signale und OSL-Signale, z.B. aus dem DJ-Bereich. Ebenso können Taktsignale (Beattrigger, Timecodes, Frequenzsignale) zum Starten von Aktivitäten verwendet werden. Es gibt auch „interne“ Signale von DMXControl wie vom Speedmaster, Audioanalyser oder anderen Plugins die man vorteilhaft zum Triggern von Cuelisten verwenden kann. Die verschiedenen Kategorien sind in der folgenden Abbildung beispielhaft sichtbar:

[Kap6 Eingangszuweisung01]	[Kap6 InputAssignment]
----------------------------	------------------------

Abb. 6.x: Input Assingment mit Input- und Output-Kategorien

Stefan erklärt, dass selbst exotische Events wie Elektrizitäts-Overload oder Drücken des Blackout-Taste zu Events führe können. In den Abbildungen fehlen Softdesks oder Timecodeshows, die dann sichtbar werden wenn es entsprechende Objekte davon im Projektextplorer geben sollte.

Auf der anderen Seite gibt es auf der „Ausgangsseite“, also das was beeinflusst werden soll nicht nur die Cuelisten, sondern auch DMX-Out-Signale oder Master-/Executor-Operationen.

Stefan erklärt, dass damit eine wahnsinnige Flexibilität innerhalb von DMXControl zur Showsteuerung erreicht werden kann. Besonders interessant ist das Input Assignment für die Programmierung der Operation des Softdesks, dazu mehr im folgenden Kapitel.

Nicht immer ist das Input Assignment darauf beschränkt, eine Eingangsgröße einer Ausgangsgröße zuzuweisen. Manchmal möchte man eine Rückmeldung bekommen (z.B. am Softdesk), manchmal muss man die Werte umwandeln bevor sie verarbeitet werden können oder manchmal möchte man eine bestimmte Logik implementieren, wo es Abhängigkeiten oder Zusammenhänge zwischen verschiedenen Signalen geben kann. Stefan sagt: Damit dies nicht in einer Art Programmiersprache ausartet, hat sich das Entwicklerteam für eine grafische „Programmierung“ entschlossen, so wie man sie vielleicht aus anderen Tools wie Node-Red oder vvvv kennt. Die Logik besteht dabei immer

aus einer Vielzahl miteinander verbundener Knoten. Jeder Knoten hat auf der rechten Seite Pins für Ausgangssignale und auf der linken Seite Pins für Eingangssignale. Ein Ausgangspin kann potentiell mit einem Eingangspin über eine Linie verbunden werden, so wie wir das bereits bei der Go-Connection gesehen hatten. Natürlich kann ein Ausgangssignal auch Input für verschiedene Eingänge sein, nur man muss immer auf den Typ der Signale achten.

Die Menge der Eingangspins auf der linken Seite wird auch Eingangs-Hub (Input Hub, auch port) genannt, entsprechend gibt es auch den Ausgangs-Hub (Output Hub) auf der rechten Seite. Es können auch jeweils nur Ausgangs-Hubs mit Input-Hubs verbunden werden.

[Kap6 Eingangszuweisung02 GO]	<p>Für diese einfache Connection wird der Space-Eintrag aus der Spalte Eingänge per Drag&Drop auf den GO-Eingangspin der gewünschten Cueliste in der Spalte Ausgänge geschoben.</p> <p>Es entsteht ein Tabelleneintrag dieser Form. Dazu muss man darauf achten, dass das Fenster genügend Platz hat um den rechten Teil wie unten anzuzeigen.</p> <p>[Kap6 Eingangszuweisung03 Tabelle]</p> <p>Wie man erkennt können existierende Connections auch deaktiviert werden, falls man sie aus irgendeinem Grunde gerade nicht benötigt.</p>
-------------------------------	--

Abb. 6.x: Herstellung einer Connection

Auf diese Weise kann auch ein Netzwerk von verbundenen Knoten entstehen, dessen Komplexität höchstens durch die Sehschärfe des Programmierers beschränkt ist. Die verfügbaren Knoten sind natürlich im Wiki beschrieben. Jeder Knoten kann für verschiedene Eigenschaften parametrisiert werden (Eigenschaften-Menü). Dort kann man z.B. Werteintervalle oder Rechenvorschriften vorgeben.

Jedes Input Assignment Netzwerk wird Connection (Verbindungsset) genannt. Alle verfügbaren Connections können tabellarisch oder in ihrer Grafikanzeige angezeigt werden.

[Kap6 Eingangszuweisung04 Grafik]

Abb. 6.x: Grafische Knotenansicht im Input Assignment

Wenn man eine sehr große Anzahl von Connections hat, können diese ähnlich zu Executoren in Bänken organisiert werden. So kann man logisch zusammenhängende Connections in einer Bank zusammenfassen um eine bessere Übersicht zu behalten. Auch ganze Bänke können gezielt ein- und ausgeschaltet werden.

Es gibt verschiedene Kategorien von Nodes: * Converter (Typwandlung) Farben, Bitmaps, Color to RGB, HSV to Color, Beat to Bool, Framat * Logic (Math. und Logikfunktionen, Gatter) FlipFlop, UND,

OR, Math, Expression, Timer, Delay, Counter, Compare, BinarySwitcher * Wrapper (Informationen über Objekte wie Cues, Devices, Steuern der Objekte) * Value Object (erstellt Matrix) * Logging (erstellt csv-Datei von Werten)

Detaillierte Informationen gibt es hier: [https://wiki-de.dmxcontrol-projects.org/index.php?title=Cuelist_\(Input_Assignment\)_DMXC3](https://wiki-de.dmxcontrol-projects.org/index.php?title=Cuelist_(Input_Assignment)_DMXC3)

6.4. Fit for Softdesk

.. zu ergänzen...

Hier werden „sophisticated“ Features vorgestellt im Zusammenhang mit dem Input Assignment, inkl. Rückmeldungen usw. * Neben dem Ansprechen von programminternen Funktionen Szenenlisten, Executoren, Master etc. bietet das Eingänge zuweisen auch die Möglichkeit, die verschiedensten Informationen zum Beispiel im Softdesk zu visualisieren * Das schließt nicht nur das Ändern des Status eines Buttons ein, ob dieser gedrückt oder nicht gedrückt ist, sondern man kann sich auch Informationen auf Textbasis wie den Namen einer aktuell aktiven Szenenliste oder die BPM-Werte aller im Projekt genutzten Speedmaster anzeigen lassen

7. Kapitel 7: Beleuchtung eines Band-Konzertes

7.1. Presets

Bei einer richtig großen ansprechenden Show kommt man um das Konzept der Presets nicht herum. Stefan sagt zwar, für den ersten Einstieg in DMXControl 3 sollten Presets erst einmal außer Acht gelassen werden. Hier empfiehlt sich, mit Cuelists zu arbeiten, welche dann eine oder mehrere Cues enthalten.

Was sind nun Presets? Ähnlich wie in Cues werden dort Geräteeigenschaften gespeichert. Es gab schon Nutzer, die aufgrund Verständnisproblemen von Cues ihre Lichtshow nur mit Presets gesteuert haben - aber das ist definitiv nicht im Sinne des Erfinders.



Presets sind nicht dazu gedacht, sie während der Show direkt aus dem Preset-Fenster aufzurufen (anstelle Cuelisten starten).

Vielmehr sind Presets der zentrale, programmweite Speicher für in mehreren Cues verwendete Geräteeigenschaften. Wer ein wenig Informatik-affin wie Jannik ist, kann sich darunter „Makros“, „Templates“, oder halt Voreinstellungen vorstellen, die in vielen Cues wiederverwendet werden sollen.

Wofür man das benötigt erklärt Stefan folgendermaßen: Mehrere Cues können sich auf die gleichen Presets beziehen. Wenn nun Jana wieder mal kurzfristig Änderungswünsche hat, muss man die Änderungen nur an einer Stelle vornehmen (nämlich im Preset), aber das hat Auswirkungen auf alle Stellen (Cues), wo das Preset verwendet wird. Damit werden die Cues praktisch aktualisiert. Er gibt ein Beispiel: Man kann zuhause eine Show vorbereiten und dann vor

Ort mit den echten Lampen nur noch die Presets an die realen Lampen-Werte anzupassen ohne dass man nochmal seine Cues editieren muss.

Wiki: Preset_DMXC3

Übrigens kam Janik der Begriff schon von den Preset-Pulten bekannt vor, aber der wird dort in einem anderen Sinne verwendet:

Eine Voreinstellung (Preset, auch Palette) ist die Möglichkeit, während der Ausgabe einer Lichtszene bereits eine zweite auf dem Pult einzustellen, um nach Ende der Szene sofort auf die andere überblenden zu können.

[Kap7 Faderbank]

Abb. 7-1: "Historische" Faderbank für Presets

Um dies zu ermöglichen, müssen bei Pulten für jeden Kanal mindestens 2 Regler vorhanden sein. Meistens sind diese in 2 Reihen bzw. Bänken untereinander angeordnet. Die erste Reihe ist dann für das erste, die zweite für das zweite Preset. Mit ein oder zwei Crossfadern wird zwischen den beiden Presets überblendet.

Was sind nun typische Anwendungsbeispiele für Presets bei DMXControl? Es handelt sich dabei um vorher eingegebene Helligkeiten, Positionen oder andere Parameter die auf beliebige Geräte angewendet werden können, so kann man z.B. Standardfarben an Presets vereinbaren. Ebenso können Grundpositionen von MovingHeads („Nullstellung“ ausrichten, "Himmelsposition", etc.) als Presets definiert werden oder man fokussiert die Lampen auf den Schlagzeuger oder die typische Sängerposition. Selbst bestimmte Lichteffekte (Kegel, Kreis, etc.) können als Preset definiert werden. Nun wird es Jannik klar, dass man damit wahnsinnig viel Arbeit in größeren Shows sparen kann. Man muss nicht für jede Cue, die z.B. den Sänger beleuchten soll diese Einstellungen neu machen!

[Kap7 PresetsDMXC3]

Abb.7-1: Preset-Fenster in DMXControl 3

Häufig benutzte Positionen der Scanner/Movingheads müssen so nicht jedes Mal neu eingegeben werden. Bei der Erstellung einer Cue verwendet man die Presets, als ob man eine Geräteeigenschaft anwenden will, statt Farbe auswählen z.B. Preset auswählen. Einfach Gerät(e) oder Gruppe selektieren, Preset anwenden indem es ausgewählt und auf die Geräte gezogen wird, Cue abspeichern und fertig! DMXControl 3 merkt sich, dass in dieser Cue ein Preset steckt.

Das ist das Basiswissen, aber kann es noch ein wenig komplizierter haben, denn Stefan erklärt die 3 Arten von Presets:

- **Reference Preset** Das Preset wird einfach in eine Cuelist gezogen, und wird dadurch zur "globalen Szene", ergo an dieser Stelle wird das Preset 1:1 wie eine Szene verwendet.
- **Device Preset** Das Preset wird auf ein bzw. mehrere Geräte bzw. Gruppen gezogen und wird nur auf den Geräten angewendet, welche auch im Preset gespeichert sind.
- **Property Preset** Die Eigenschaftswerte des Presets werden verwendet. Dabei spielen die Geräte keine Rolle. Jedes selektierte Gerät mit den passenden Eigenschaften wendet den Wert des

Presets an.

Stefan erwähnt noch, dass Werte über den Programmer gefiltert werden können und in einer Cue können auch mehrere Presets gleichzeitig verwendet werden.

7.2. Fit for Effects als Profi

tbd. Jana möchte für die finale Show besonders eindrucksvolle Effekte haben, denn inzwischen hat sie ihren klassschen Theaterhintergrund verlassen...

[MH Schwenken Effekt Trapezoid]

7.3. Das finale Projekt