



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) DE 600 23 730 T2 2006.07.06

(12)

Übersetzung der europäischen Patentschrift

(97) EP 1 224 845 B1

(51) Int Cl.⁸: H05B 37/02 (2006.01)

(21) Deutsches Aktenzeichen: 600 23 730.3

(86) PCT-Aktenzeichen: PCT/US00/19274

(96) Europäisches Aktenzeichen: 00 950 360.8

(87) PCT-Veröffentlichungs-Nr.: WO 2001/005195

(86) PCT-Anmeldetag: 14.07.2000

(87) Veröffentlichungstag

der PCT-Anmeldung: 18.01.2001

(97) Erstveröffentlichung durch das EPA: 24.07.2002

(97) Veröffentlichungstag

der Patenterteilung beim EPA: 02.11.2005

(47) Veröffentlichungstag im Patentblatt: 06.07.2006

(30) Unionspriorität:

143790 P 14.07.1999 US

(84) Benannte Vertragsstaaten:

AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT,
LI, LU, MC, NL, PT, SE

(73) Patentinhaber:

Color Kinetics Inc., Boston, Mass., US

(72) Erfinder:

Lys, Ihor A., Milton, US; Morgan, Frederick M.,
Quincy, US; Dowling, Kevin J., Westford, US;
Blackwell, Michael K., Milton, US

(74) Vertreter:

Samson & Partner, Patentanwälte, 80538 München

(54) Bezeichnung: SYSTEME UND METHODEN UM LICHTSEQUENZEN ZU ERSTELLEN

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelebt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

Gebiet der Erfindung

[0001] Die vorliegende Erfindung bezieht sich allgemein auf Systeme und Verfahren zum Steuern von Lichtsystemen und insbesondere auf computerbasierte Systeme und Verfahren zum Gestalten von Lichtsequenzen und Ausführen solcher Sequenzen auf Beleuchtungssystemen.

Hintergrund der Erfindung

[0002] Die meisten modernen Lichtsteuerungsgeräte sind entworfen, um weißes Licht (oder monochromatisches Licht) in einem Theater oder gehobenem Geschäftsumfeld zu steuern. Ein Beleuchtungskörper, der monochromatisches Licht, wie weiß, blau oder rot erzeugt, kann in erster Linie entlang einer einzigen Dimension – Helligkeit – vom ausgeschalteten Zustand zu einer maximalen Helligkeit verändert werden. Aktuelle Steuerungsgeräte erlauben es einem Benutzer, eine Helligkeit für jeden Beleuchtungskörper über eine Zeit hinweg zu bestimmen.

[0003] Dieses Verfahren wird zunehmend komplizierter für Beleuchtungskörper, die imstande sind, die Farbe des emittierten Lichts zu verändern, weil die resultierende Farbe und Intensität eine Kombination aus der Intensität von drei zusammengesetzten Primärfarben ist, von denen jede unabhängig von den anderen für ein bestimmtes Licht gesetzt werden kann. Deshalb ist die Ausgabe eine Funktion von drei Dimensionen, anstatt einer, die für jeden Zeitpunkt spezifiziert werden muß, wobei die Anstrengung und die Zeit, die zum Erzeugen eines Effekts benötigt wird, stark ansteigt. US-Patent Nr. 5,307,295 von Taylor et al. beschreibt ein System zum Erzeugen von Lichtsequenzen, das einige Aspekte des Erzeugens einer Lichtsequenz vereinfacht, aber viele der Parameter müssen immer noch für jeden Lichtkörper spezifiziert werden, so als wären sie auf einer Standardlichtkonsole. Ein intuitiveres Verfahren zum Gestalten von Lichtsequenzen würde nicht nur den Entwurfsprozeß vereinfachen und beschleunigen, sondern würde es Benutzern erlauben, Lichtsequenzen mit weniger Training und Erfahrung, als es heute oft notwendig ist, zu gestalten.

[0004] Auch wenn Sequenzen durch traditionelle Verfahren gestaltet und abgespielt werden können, schreitet der Inhalt der Sequenzen ferner typischerweise mit der Zeit voran und kann während des Abspielens nicht verändert werden. Falls zum Beispiel eine dramatische Szene einen Lichtblitz benötigt, der zu einer bestimmten Zeit simuliert werden soll, wird dieser Effekt typischerweise entweder durch akribisches zeitliches Festlegen der Inszenierung, um den programmierten Blitz und den kritischen Moment zusammenfallen zu lassen, oder durch manuelles Ausführen des Blitzes zum kritischen Moment erreicht. Solche Techniken benötigen entweder einen beträchtlichen Verlaß auf den Zufall oder schließen einen Verlaß auf die Automatisierung aus.

[0005] Eine Technik, die einen intuitiven Zugang zum Gestalten von Lichtsequenzen erlaubt, würde die Zeit und das Training, das benötigt wird, um einen gewünschten Effekt zu erzielen, reduzieren und würde es erlauben, daß farbige Lichter mit minimaler Auswirkung auf die Effizienz betätigt würden. Zusätzlich wird ein Verfahren zum Ausführen solcher Lichtsequenzen, das eine Flexibilität bei der Wiedergabe der Sequenz unterstützt, eine erhöhte Freiheit in einer dazugehörigen Aufführung erlauben, oder die Verwendung programmierter Lichtsequenzen in Situationen erlauben, die von Natur aus unvorhersagbar sind.

[0006] EP-A-495305 beschreibt ein Modellierungs- und Steuerungssystem zum Offline-Erzeugen von Beleuchtungsentwürfen und zum Online-Steuern der Tätigkeit der eigentlichen Lichtsysteme, die diese Entwürfe erzeugen.

[0007] US 3898643 beschreibt ein Lichtsystem, das über ein elektronisches Display gesteuert wird, und ein Verfahren zum Steuern einer großen Anzahl an Theaterbühnenlichtköpfen, einschließlich Datenspeichervorrichtungen zum Speichern von Information, die Sequenzen von Bühnenlichthinweisen repräsentiert.

[0008] Entsprechend der Erfindung gemäß einem Aspekt wird ein System zum Vorbereiten einer Lichtsequenz entsprechend Anspruch 1 bereitgestellt.

[0009] Entsprechend der Erfindung gemäß einem anderen Aspekt wird ein Verfahren zum Vorbereiten einer Lichtsequenz entsprechend Anspruch 17 bereitgestellt.

[0010] Die hierin beschriebenen Systeme und Verfahren beziehen sich auf eine intuitive Schnittstelle zum Gestalten von Lichtsequenzen, wie durch Bereitstellen einer visuellen Repräsentation einer Sequenz während ih-

res Entwurfes. Zusätzlich beziehen sich die hierin beschriebenen Systeme und Verfahren auf die Wiedergabe programmierter Lichtsequenzen, so daß die Sequenz während des Abspielens modifiziert werden kann, z.B. auf der Grundlage externer Impulse oder Hinweise.

[0011] Ein System zum Steuern einer Vielzahl von Lichteinheiten kann eine Datenschnittstelle zum Empfangen von Instruktionen, zum Steuern einer Vielzahl von Lichteinheiten, eine Signalschnittstelle zum Empfangen externer Signale, einen Prozessor zum Konvertieren der Instruktionen in einen Datenstrom und zum Verändern der Konversion dieser Instruktionen auf der Grundlage der empfangenen externen Signale und eine Datenausgabe zum Übertragen des Datenstroms an eine Vielzahl von Lichteinheiten umfassen.

[0012] Ein Verfahren zum Steuern einer Vielzahl von Lichteinheiten entsprechend den Prinzipien der Erfindung kann ein Empfangen von Instruktionen zum Steuern einer Vielzahl von Lichteinheiten, ein Empfangen externer Signale, ein Konvertieren dieser Instruktion in einen Datenstrom auf der Grundlage der empfangenen externen Signale und ein Übertragen des Datenstroms an eine Vielzahl von Lichteinheiten umfassen.

[0013] Ein anderes Verfahren zum Steuern einer Vielzahl von Lichteinheiten entsprechend der Prinzipien der Erfindung kann ein Empfangen von Instruktionen einschließlich eines primären Lichteffekts und eines sekundären Lichteffekts umfassen, wobei der sekundäre Lichteffekt dazu bestimmt ist, in Bezug auf eine vorbestimmte Bedingung anstelle des primären Lichteffekts ausgeführt zu werden, ein Senden von Instruktionen an eine Lichteinheit, um den primären Lichteffekt auszuführen, ein Empfangen eines Signals, das die vorbestimmte Bedingung angibt, und ein Senden von Instruktionen an die Lichteinheit, um den sekundären Lichteffekt auszuführen.

[0014] Noch ein weiteres Verfahren zum Steuern einer Vielzahl von Lichteinheiten umfaßt ein Empfangen von Instruktionen zum Ausführen einer zeitlich festgelegten Sequenz von Lichteffekten, ein Ausführen der Sequenz von Lichteffekten unter Verwendung einer Vielzahl von Lichteinheiten, ein Empfangen eines externen Signals und ein Verändern der Ausführung der Sequenz der Lichteffekte.

Kurze Beschreibung der Figuren

[0015] Die folgenden Figuren stellen bestimmte veranschaulichende Ausführungsformen der Erfindung dar, in der sich gleiche Bezugszeichen auf gleiche Elemente beziehen. Diese dargestellten Ausführungsformen sollen als veranschaulichend für bevorzugte Umsetzungen der Erfindung umzusetzen, verstanden werden.

[0016] [Fig. 1](#) veranschaulicht ein System zum Erstellen einer Lichtsequenz und zum Ausführen der Lichtsequenz auf einer Vielzahl von Lichteinheiten, wie hierin beschrieben wird.

[0017] [Fig. 2](#) stellt ein beispielhaftes Verfahren zum Erstellen eines Lichteffekts dar, wie hierin beschrieben wird.

[0018] [Fig. 3](#) stellt eine typische Schnittstelle zum Beschreiben einer Anordnung von Lichteinheiten dar.

[0019] [Fig. 4](#) stellt eine alternative Schnittstelle zum graphischen Wiedergeben einer Lichtsequenz dar.

[0020] [Fig. 5](#) stellt eine entsprechende Schnittstelle zum Erstellen einer Lichtsequenz dar, wie hierin beschrieben wird.

[0021] [Fig. 6](#) zeigt eine Ausführungsform eines Steuerungsgerätes zum Ausführen einer Lichtsequenz, wie hierin beschrieben wird.

Detaillierte Beschreibung der dargestellten Ausführungsformen

[0022] Die nachfolgende Beschreibung bezieht sich auf mehrere veranschaulichende Ausführungsformen der Erfindung. Auch wenn viele Variationen der Erfindung von einem Fachmann ausgedacht werden können, sollen solche Variationen und Verbesserungen in den Bereich dieser Offenbarung fallen, wie er durch die geänderten Ansprüche definiert ist. Folglich ist der Umfang dieser Erfindung, wie er durch die geänderten Ansprüche definiert ist, nicht in irgendeiner Weise durch nachstehende Offenbarungen begrenzt. Die Begriffe "Sequenz" oder "Lichtsequenz", wie sie hierin verwendet werden, sind gedacht, sich auf sequentielle Anzeigen zu beziehen, sowie auf nicht-sequentielle Anzeigen, flußgesteuerte Anzeigen, unterbrechungsgesteuerte oder ereignisgesteuerte Anzeigen oder beliebige andere gesteuerte Überlappungen oder sequentielle Anzeigen mit

einem oder mehreren Lichtkörpern. Die hierin beschriebenen Systeme und Verfahren beziehen sich auf ein System, wie einen Prozessor **10**, der eine Softwareanwendung unterstützt, die eine Schnittstelle **15** hat, wie in [Fig. 1](#) dargestellt wird, mit der ein Benutzer ein Lichtprogramm **20** erstellen kann, das eine oder mehrere Lichtsequenzen umfassen kann, das von einem Lichtsteuerungsgerät **30** ausgeführt werden kann, das eine oder mehrere Lichteinheiten **40** steuert. Der Begriff "Sequenz" im Kontext dieser Offenbarung wird verwendet, um sich auf irgendein Muster, irgendeine Show, Sequenz, Anordnung oder Sammlung von Befehlen zu beziehen, die verwendet werden, um Lichteinheiten oder andere Geräte durch das System zu betreiben. Ein Fachmann würde erkennen, daß eine Sequenz auch keine geordnete Sequenz sein muß oder einen linearen Entwurf haben muß. Sequenzen, die nichtlineare, prioritätenbasierte und/oder überlappende Befehle umfassen, können auch noch eine Sequenz umfassen. Die Softwareanwendung kann eine allein operierende unabhängige Anwendung sein, wie ein ausführbares Speicherabbild eines C++ oder Fortran-Programms oder ein anderer ausführbarer Code und/oder Bibliotheken, oder kann in Verbindung mit oder zugänglich durch einen Webbrowser, wie ein Java Applet oder eine oder mehrere HTML-Websites etc. umgesetzt sein. Der Prozessor **10** kann irgendein System zum Verarbeiten in Reaktion auf ein Signal oder Daten sein und sollte so verstanden werden, daß er Mikroprozessoren, Mikrocontroller oder andere integrierte Schaltkreise, Computersoftware, Computerhardware, elektrische Schaltkreise, anwendungsspezifische integrierte Schaltkreise, PCs, Chips und andere Geräte alleine oder in Kombination umfaßt, die Verarbeitungsfunktionen zur Verfügung stellen können. Zum Beispiel kann der Prozessor **10** eine beliebige geeignete datenverarbeitende Plattform sein, wie eine konventionelle IBM PC-Workstation, auf der das Windows-Betriebssystem läuft, oder eine SUN Workstation, auf der eine Version des Unix Betriebssystems, wie Solaris läuft, oder irgendeine andere geeignete Workstation sein. Eine Steuerung **30** kann mit Lichteinheiten **40** über Radiofrequenz (RF), Ultraschall, Hörschall, Infrarot (IR), optische Wellen, Mikrowellen, Laser, elektromagnetische Wellen oder irgendein anderes Übertragungs- oder Verbindungsverfahren oder -system kommunizieren. Es kann ein beliebiges geeignetes Protokoll zur Übertragung verwendet werden, einschließlich pulsbreitenmodulierte Signale, wie DMX, RS-485, RS-232, oder irgendein anderes geeignetes Protokoll. Die Lichteinheiten **40** können eine Glühbirne, LED, Leuchtröhre, Halogenlampe, Laser oder irgendeine andere Art von Lichtquelle sein, z.B. so eingerichtet, daß jede Lichtquelle mit einer vorbestimmten zugewiesenen Adresse assoziiert ist, entweder eindeutig mit dieser Lichteinheit oder mit der Adresse anderer Lichteinheiten überlappend. In bestimmten Ausführungsformen kann eine einzelne Komponente imstande sein, sowohl einem Benutzer zu erlauben, ein Lichtprogramm zu erstellen, als auch die Lichteinheiten zu steuern, und die vorliegende Erfindung soll diese und andere Variationen des Systems umfassen, das in [Fig. 1](#) dargestellt ist, das verwendet werden kann, die unten beschriebenen Verfahren umzusetzen. In bestimmten Ausführungsformen können die Funktionen der Softwareanwendung von einem Hardwaregerät bereitgestellt werden, wie einem Chip oder einer Karte oder irgendeinem anderen beliebigen System, das irgendeine der hierin beschriebenen Funktionen bereitstellen kann.

[0023] Entsprechend eines Verfahrens **200** zum Erzeugen einer Lichtsequenz, das in [Fig. 2](#) dargelegt ist, kann ein Benutzer aus einer Menge vorbestimmter "Bestands"-Effekte **210** wählen. Die Bestandseffekte funktionieren als diskrete Elemente oder Baublöcke, die für ein Zusammenstellen einer Sequenz nützlich sind. Zusätzlich kann ein Benutzer eine bestimmte Sequenz zusammensetzen und diese Sequenz in die Bestandseffekte einfügen, um das Erfordernis zu beseitigen, wiederholte Elemente immer wieder neu zu erstellen, jedes Mal, wenn der Effekt benötigt wird. Zum Beispiel kann die Menge von Bestandseffekten einen Abdunklungseffekt und einen Aufhellungseffekt umfassen. Ein Benutzer kann einen Pulseffekt durch Spezifizieren der Abfolge der Abdunklungs- und Aufhellungseffekte zusammenstellen und den Pulseffekt in die Menge der Bestandseffekte einfügen. Folglich kann jedes Mal, wenn danach ein Pulseffekt benötigt wird, der Bestandseffekt verwendet werden, ohne daß wiederholt Abdunklungs- und Aufhellungseffekte zum Erreichen desselben Ziels ausgewählt werden müssen. In bestimmten Ausführungsformen können Bestandseffekte auch von einem Computer durch eine Programmiersprache, wie Java, C++ oder irgendeine andere geeignete Sprache erzeugt werden. Effekte können zur Menge an Bestandseffekten hinzugefügt werden, indem man die Effekte als Plug-ins zur Verfügung stellt, indem man die Effekte in eine Effektedatei eingliedert oder durch irgendeine andere Technik, die zum Organisieren von Effekten auf eine Art und Weise geeignet ist, die es erlaubt, die Menge an Effekten zu erweitern, zu vermindern und zu verändern.

[0024] Zusätzlich kann ein Benutzer einen Effekt auswählen und einen Zeitpunkt angeben, zu dem der Effekt beginnen soll **220**. Zum Beispiel kann der Benutzer angeben, daß ein Aufhellungseffekt drei Minuten nach Beginn einer Sequenz starten sollte. Zusätzlich kann der Benutzer einen Endzeitpunkt oder eine Dauer für den Effekt **230** auswählen. Deshalb kann ein Benutzer durch die Angabe, daß der Effekt fünf Minuten nach Beginn einer Sequenz enden sollte, oder äquivalent durch Angabe, daß der Effekt zwei Minuten dauern sollte, die Zeitparameter für den Aufhellungseffekt setzen. Zusätzliche Parameter können vom Benutzer spezifiziert werden, soweit es für den speziellen Effekt **240** erforderlich sein sollte. Zum Beispiel kann ein Aufhellungs- oder Abdunklungseffekt ferner durch eine Anfangshelligkeit und eine Endhelligkeit definiert werden. Die Änderungsge-

schwindigkeit kann vorbestimmt sein, d.h. der Abdunklungseffekt kann eine lineare Abdunklungsgeschwindigkeit über die zugewiesene Zeitspanne anwenden oder sie kann durch den Benutzer veränderbar sein, kann z.B. ein langsames Abdunkeln zu Beginn erlauben, gefolgt von einem schnellen Abfall, oder irgendein anderes Schema, das der Benutzer spezifiziert. Auf ähnliche Weise kann ein Pulseffekt, wie oben beschrieben, stattdessen durch eine maximale Helligkeit, eine minimale Helligkeit und eine Periodizität oder Wechselrate charakterisiert werden. Zusätzlich kann die Art und Weise des Wechsels durch den Benutzer veränderbar sein, z.B. die Veränderungen der Helligkeit können eine Sinusfunktion oder alternierende lineare Veränderungen wiedergeben. In Ausführungsformen, bei welchen farbverändernde Lichtkörper eingesetzt werden, können Parameter wie Anfangsfarbe, Endfarbe, Änderungsrate, etc. vom Benutzer spezifiziert werden. Viele zusätzliche Effekte und geeignete Parameter dafür sind bekannt oder werden den Fachleuten offensichtlich sein und fallen innerhalb des Umfangs dieser Offenbarung.

[0025] In bestimmten Ausführungsformen kann ein Benutzer einen Übergang zwischen zwei nacheinander auftretenden Effekten festlegen. Wenn zum Beispiel ein Pulseffekt von einem Abdunklungseffekt gefolgt wird, kann sich der Pulseffekt zum Ende des Effekts hin weniger schnell verändern, allmählich dunkler werden oder weniger zwischen maximaler und minimaler Helligkeit variieren. Techniken für den Übergang zwischen diesen oder anderen Effekten können vom Benutzer für jeden Übergang bestimmt werden, z.B. durch Auswählen eines Übergangseffekts aus einer Menge vorbestimmter Übergangseffekte, oder durch Setzen von Übergangsparametern für den Beginn und/oder das Ende eines oder beider Effekte.

[0026] In einer weiteren Ausführungsform können Benutzer mehrere Lichteffekte für dieselbe Lichteinheit spezifizieren, die Effekte überlappend in Zeit oder Ort plazieren. Diese überlappenden Effekte können auf eine additive oder subtraktive Art und Weise verwendet werden, so daß die Vielzahl an Effekten miteinander interagieren. Zum Beispiel könnte ein Benutzer einen Aufhellungseffekt über einen pulsierenden Effekt legen, wobei der Aufhellungseffekt dem Parameter minimale Helligkeit des Pulses auferlegt wird, um den Effekt des Pulssens langsam ansteigend in ein stetiges Licht zu wandeln.

[0027] In einer anderen Ausführungsform könnten die überlappenden Lichteffekte Prioritäten oder Hinweise haben, die ihnen zugewiesen wurden, die einer speziellen Lichteinheit erlauben könnten, einen Effekt nach Erhalt eines Hinweises hin zu verändern. Dieser Hinweis könnte irgendeine Art von Hinweis sein, der extern oder intern im System empfangen wird und umfaßt, ist aber nicht darauf beschränkt, einen benutzergesteuerten Hinweis, wie einen manuellen Schalter oder Drucktaster; einen benutzerdefinierten Hinweis, wie eine bestimmte Tastenkombination oder eine Zeittaste, die es einem Benutzer erlauben, einen bestimmten Effekt anzusteuern oder abzuschreiten; einen Hinweis, der vom System erzeugt wird, wie ein interner Taktgebermechanismus, ein interner Speichermechanismus oder ein softwarebasierter Mechanismus; einen mechanischen Hinweis, der von einem analogen oder digitalen Gerät erzeugt wurde, der dem System angefügt wurde, wie eine Uhr, ein externer Lichtsensor, ein Musiksynchrosynchronisationsgerät, ein Geräuschpegelerfassungsgerät, oder ein manuelles Gerät, wie ein Schalter; einen Hinweis, der über ein Übertragungsmedium, wie ein elektrischer Draht oder Kabel, ein Funksignal oder ein IR-Signal empfangen wurde; oder einen Hinweis, der von einer an das System angeschlossenen Lichteinheit empfangen wurde. Die Priorität könnte dem System erlauben, einen voreingestellten Prioritätseffekt zu wählen, der derjenige Effekt ist, der von der Lichteinheit verwendet wird, wenn nicht ein spezieller Hinweis zu dem Zeitpunkt empfangen wurde, an dem das System die Verwendung eines anderen Effekts anweist. Dieser Wechsel des Effekts könnte temporär auftreten, nur während der Dauer des Hinweises, oder für eine spezifizierte Zeitspanne definiert sein, könnte permanent sein, und nicht den weiteren Empfang anderer Effekte oder Hinweise erlauben, oder könnte prioritätsbasiert sein, und auf einen neuen Hinweis warten, um zu dem ursprünglichen Effekt zurückzukehren oder um einen neuen Effekt auszuwählen. Alternativ könnte das System Effekte auswählen, die auf dem Zustand eines Hinweises und der Wichtigkeit eines gewünschten Effekts basieren. Falls ein Geräuschsensor ein plötzliches Geräusch wahrnehmen würde, könnte er beispielsweise einen Alarmaffekt mit hoher Priorität auslösen, der alle anderen Effekte überschreibt, die sonst vorhanden sind oder auf ihre Ausführung warten. Die Priorität könnte auch zustandsabhängig sein, wobei ein Hinweis einen alternativen Effekt auswählt oder, abhängig vom aktuellen Zustand des Systems, ignoriert wird.

[0028] In bestimmten Ausführungsformen kann das Ergebnis eines Effekts programmiert sein, von einem zweiten Effekt abzuhängen. Zum Beispiel kann ein Effekt, der einer Lichteinheit zugeordnet ist, ein zufälliger Farbeffekt sein, und ein Effekt, der einer zweiten Lichteinheit zugeordnet ist, kann dazu bestimmt sein, die Farbe des zufälligen Farbeffekts abzugleichen. Alternativ kann eine Lichteinheit programmiert sein, einen Effekt auszuführen, wie einen Blinkeffekt, wann immer eine zweite Lichteinheit eine bestimmte Bedingung erfüllt, wie ihr Ausschalten. Es können sogar komplexere Anordnungen durch dieses Schema erzeugt werden, wie ein Effekt, der auf eine bestimmte Bedingung eines Effekts hin initiiert wird und sich auf die Farbe eines anderen

Effekts und die Rate eines dritten Effekts abgleicht. Andere Kombinationen von Effekten, bei denen wenigstens ein Parameter oder das Auftreten eines Effekts von einem Parameter oder dem Auftreten eines zweiten Effekts abhängt, sind für die Fachleute offensichtlich und sollen in den Bereich dieser Offenbarung fallen.

[0029] In noch anderen Ausführungsformen erlauben die hierin beschriebenen Systeme und Verfahren, daß eine Lichtsequenz von externen Eingaben während einer Aufführung beeinflußt wird. Zum Beispiel kann eine Lichtsequenz oder ein Effekt programmiert werden, um auf Empfang eines Auslösesignals hin zu beginnen, eine Sequenz oder ein Effekt kann Vorrang haben, falls ein Signal empfangen wird, eine Sequenz oder ein Effekt kann dazu bestimmt sein, sich zu wiederholen oder bis zum Empfang eines Signals fortzufahren, etc. Folglich kann ein Benutzer, anstatt einem Effekt oder einer Sequenz eine diskrete Startzeit zuzuordnen, stattdessen diesen Effekt oder diese Sequenz dazu bestimmen, nach Empfang eines bestimmten Auslöseimpulses zu starten. Bei der Erzeugung kann ein Benutzer außerdem zwei oder mehr Effekte dazu bestimmen, sich zu überlappen oder gleichzeitige Zeitspannen, zu haben und kann den Effekten unterschiedliche Prioritäten oder Bedingungen zuweisen, um zu bestimmen, welcher Effekt beim Abspielen ausgeführt wird. In noch einer anderen Ausführungsform kann ein Benutzer einen Parameter für einen Effekt mit einem externen Eingabesignal verbinden, einschließlich analoger, digitaler und manueller Eingaben, so daß die Farbe, Geschwindigkeit oder ein anderes Attribut eines Effekts von einem Signal von einem externen Gerät abhängen kann, welches z.B. die Lautstärke, Helligkeit, Temperatur, Tonhöhe, Steigung, Wellenlänge oder irgendeine andere geeignete Bedingung mißt. Folglich kann die Auswahl einer Lichtsequenz, die Auswahl eines Effekts oder die Auswahl eines Parameters durch eine Eingabe von einer externen Quelle, wie einem Benutzer, Zeitmesser, Gerät oder Sensor bestimmt oder beeinflußt werden.

[0030] In ereignisorientierten Ausführungsformen, wie jenen, die externe Eingaben verwenden und jenen, die Ausgaben anderer Effekte als Eingaben verwenden, kann ein Menü zur Verfügung gestellt werden, um Eingaben und die Folgen daraus zu definieren. Zum Beispiel kann einem Benutzer eine Palette vorbestimmter Eingaben zur Verfügung gestellt werden. Jede Eingabe, wie ein spezifizierter Signalumwandler oder die Ausgabe eines anderen Effekts, kann innerhalb einer verfaßten Lichtsequenz als ein Auslöser für einen neuen Effekt ausgewählt und plaziert werden, oder als ein Auslöser zu einer Variation in einem existierenden Effekt. Bekannte Eingaben können z.B. umfassen Thermistoren, Uhren, Tastaturen, numerische Tastenfelder, MIDI-Eingaben ("Midi") Musical Instrument Digital Interface ("Midi"), Eingänge, DMX-Steuersignale, logische TTL- oder CMOS-Signale, andere visuelle oder Audiosignale oder irgendein anderes Protokoll, andere Standard oder andere Signal- oder Steuertechnik, die eine vorbestimmte Form hat, ob nun analog, digital, manuell oder irgendeine andere Form. Die Palette kann auch eine kundenspezifische Eingabe umfassen, die z.B. als ein Icon in einer Palette oder eine Option in einem Dropdown-Menü repräsentiert ist. Die kundenspezifische Eingabe kann es einem Benutzer erlauben, die Spannung, Stromstärke, Dauer und/oder Form (d.h. Sinuskurve, Puls, Schritt, Modulation) für ein Eingabesignal zu definieren, das als eine Steuerung oder ein Auslöser in einer Sequenz agieren wird.

[0031] Zum Beispiel kann eine Theaterlichtsequenz programmierte Lichtsequenzen und Spezialeffekte in der Reihenfolge umfassen, in der sie auftreten, die aber Eingabe an spezifierten Punkten benötigen, bevor die nächste Sequenz oder ein Teil davon ausgeführt wird. Auf diese Weise können Szenenänderungen nicht automatisch als eine Funktion des Zeitablaufs allein, sondern auf den Hinweis eines Regisseurs, Produzenten, Bühnenarbeiters oder eines anderen Teilnehmers stattfinden. Auf ähnliche Weise können Effekte, die zeitgleich mit einer Handlung auf der Bühne abgestimmt werden müssen, wie ein Aufhellen, wenn ein Schauspieler eine Kerze anzündet oder einen Schalter umlegt, ein dramatisches Blinken oder Blitzen etc., von einem Regisseur, einem Produzenten, Bühnenarbeiter oder anderem Teilnehmer – sogar einem Schauspieler – exakt angegeben werden, wobei man dadurch die Schwierigkeit und das Risiko, sich allein auf einen vorprogrammierten Zeitablauf zu verlassen, reduziert.

[0032] Eine Eingabe aus Sensoren kann auch verwendet werden, um Lichtsequenzen zu modifizieren. Zum Beispiel kann ein Lichtsensor benutzt werden, um die Helligkeit der Lichtkörper zu modifizieren, z.B. um ein konstantes Beleuchtungsniveau aufrechtzuerhalten, ungeachtet der Menge des Sonnenlichts, das in den Raum gelangt oder um sicherzustellen, daß ein Lichteffekt trotz der Anwesenheit anderer Lichtquellen hervortritt. Ein Bewegungssensor oder anderer Detektor kann als Auslöser verwendet werden, um eine Lichtsequenz zu beginnen oder zu verändern. Zum Beispiel kann ein Benutzer eine Lichtsequenz zu Reklame- oder Anzeigezwecken programmieren, sich zu verändern, wenn sich eine Person einer Verkaufstheke oder Anzeige nähert. Temperatursensoren können auch verwendet werden, um eine Eingabe bereitzustellen. Zum Beispiel kann die Farbe des Lichts in einer Gefriertruhe temperaturabhängig programmiert werden, z.B. Vorsehen eines blauen Lichts zur Anzeige einer kalten Temperatur und ein allmähliches Ändern nach Rot mit steigender Temperatur, bis eine kritische Temperatur erreicht ist, worauf ein Blinken oder anderer Warneffekt beginnen kann.

In ähnlicher Weise kann ein Alarmsystem verwendet werden, um ein Signal bereitzustellen, das eine Lichtsequenz oder einen Effekt auslöst, um ein Warnsignal, Notsignal oder eine andere Angabe bereitzustellen. Eine interaktive Lichtsequenz kann erzeugt werden, wobei z.B. der ausgeführte Effekt entsprechend der Position, den Bewegungen oder anderen Aktionen einer Person variiert.

[0033] In bestimmten Ausführungsformen kann ein Benutzer Information zur Verfügung stellen, die repräsentativ für die Anzahl und Typen der Lichteinheiten und die räumlichen Beziehungen zwischen ihnen ist. Zum Beispiel kann eine Schnittstelle **300** zur Verfügung gestellt werden, wie in [Fig. 3](#) dargestellt ist, wie ein Gitter oder ein anderes, zweidimensionales Feld, das es dem Benutzer erlaubt, Icons oder andere repräsentative Elemente anzugeben, um die Anordnung der verwendeten Lichteinheiten zu repräsentieren. In einer Ausführungsform, die in [Fig. 3](#) dargestellt ist, stellt die Schnittstelle **300** einem Benutzer eine Auswahl von Standardtypen von Lichteinheiten **310**, z.B. Seitenbeleuchtungen, Lampen, Scheinwerfer etc., wie durch Bereitstellen einer Auswahl von Typen von Lichteinheiten in einem Menü, auf einer Palette, auf einer Werkzeugleiste, usw. zur Verfügung. Der Benutzer kann dann die Lichteinheiten auf der Schnittstelle auswählen und anordnen, z.B. innerhalb des Gestaltungsräums **320** in einer Anordnung, die die physikalische Anordnung der eigentlichen Lichteinheiten approximiert.

[0034] In bestimmten Ausführungsformen können die Lichteinheiten in unterschiedlichen Gruppen organisiert werden, z.B. um die Manipulation einer großen Anzahl an Lichteinheiten zu erleichtern. Lichteinheiten können in Gruppen organisiert sein, basierend auf räumlichen Beziehungen, funktionellen Beziehungen, Typen von Lichteinheiten oder irgendeinem anderen Schema, das vom Benutzer gewünscht wird. Räumliche Anordnungen können hilfreich sein, um Lichteffekte einfach einzugeben und auszuführen. Falls z.B. eine Gruppe von Lichtkörpern in einer Reihe angeordnet ist und diese Information dem System zur Verfügung gestellt wird, kann das System dann Effekte umsetzen, wie einen Regenbogen oder ein sequentielles Blinken, ohne daß ein Benutzer für jede Lichteinheit ein separates und einzelnes Programm spezifizieren muß. Alle die oben genannten Implementierungstypen oder Effekte könnten auf einer Gruppe von Einheiten genauso verwendet werden wie auf einzelnen Lichteinheiten. Die Benutzung von Gruppen kann es einem Benutzer auch erlauben, einen einzelnen Befehl oder Hinweis einzugeben, um eine vorbestimmte Auswahl von Lichteinheiten zu steuern.

[0035] Eine Lichtsequenz kann auf einem Lichtsystem getestet oder ausgeführt werden, um die vom Benutzer erzeugten Effekte erfahrbar zu machen. Zusätzlich kann die Schnittstelle **300** imstande sein, eine Lichtsequenz wiederzugeben, die vom Benutzer erzeugt wurde, z.B. durch Nachbilden der programmierten Effekte, so als ob die Icons auf der Schnittstelle die zu steuernden Lichteinheiten wären. Folglich kann, falls eine Lichtsequenz spezifiziert würde, daß eine bestimmte Lichteinheit beim Abspielen allmählich bis zu einer mittleren Intensität heller wird, das Icon, das diese Lichteinheit repräsentiert, schwarz beginnen und allmählich heller bis grau werden. In ähnlicher Weise können Farbänderungen, Blinken und andere Effekte visuell auf der Schnittstelle dargestellt werden. Diese Funktion kann es einem Benutzer erlauben, eine gänzlich oder teilweise erzeugte Lichtsequenz auf einem Monitor oder anderem Bildterminal zu präsentieren, die Wiedergabe zu pausieren und die Lichtsequenz zu modifizieren, bevor man mit der Wiedergabe weitermacht, um eine hochinteraktive Methode zur Erstellung einer Show bereitzustellen. In einer weiteren Ausführungsform könnte das System ein schnelles Vorspielen, rückwärts abspielen, zurückspulen oder andere Funktionen erlauben, um ein Editieren eines beliebigen Teils der Lichtsequenz zu erlauben. In noch einer weiteren Ausführungsform könnte das System zusätzliche Schnittstellenmerkmale verwenden wie jene, die aus dem Stand der Technik bekannt sind. Dies kann ein nicht-lineares Editieren umfassen, ist aber nicht darauf beschränkt, ein nicht-lineares Editieren wie dasjenige, das in Adobe oder solchen Geräten oder Steuerelementen verwendet wird, wie Bildlaufleisten, Ziehbalken oder andere Geräte und Steuerelemente.

[0036] Eine alternative Schnittstelle **400** zur Wiedergabe einer Lichtsequenz ist in [Fig. 4](#) dargestellt. Die Schnittstelle **400** umfaßt Repräsentationen von Lichtelementen **410** und Wiedergabesteuerungen **420**. Andere Techniken zum Visualisieren einer Lichtsequenz sind den Fachleuten offensichtlich und können eingesetzt werden, ohne vom Umfang und Geist dieser Offenbarung abzuweichen.

[0037] Eine Schnittstelle, die die Lichtsequenz repräsentieren kann, kann auch während der Eingabe der Lichtsequenz verwendet werden. Z.B. kann ein Gitter, wie die Schnittstelle **15** der [Fig. 1](#), eingesetzt werden, bei dem verfügbare Lichteinheiten entlang einer Achse repräsentiert werden und die Zeit entlang einer zweiten Achse repräsentiert wird. Wenn folglich ein Benutzer spezifiziert, dass eine bestimmte Lichteinheit allmählich bis zu einer mittleren Intensität heller wird, kann der durch diese Lichteinheit, die Startzeit und die Endzeit definierte Gitterabschnitt an einem Ende des Gitterabschnitts schwarz erscheinen und allmählich heller werden bis hin zu grau am anderen Ende des Gitterabschnitts. Auf diese Weise kann der Effekt dem Benutzer auf der Schnittstelle visuell dargestellt werden als die Lichtsequenz, die gerade erzeugt wird. In bestimmten Ausfüh-

rungsformen können Effekte, die schwierig mit einer statischen Repräsentation zu repräsentieren sind, wie Blinken, zufällige Farbwechsel usw. kinetisch auf der Schnittstelle, z.B. durch Blinken oder zufälliges Verändern der Farbe des definierten Gitterabschnitts repräsentiert werden. Ein Beispiel einer Schnittstelle **500**, die eine Sequenz für eine Auswahl dreier Lichteinheiten repräsentiert, ist in [Fig. 5](#) gezeigt. Ein Zeitdiagramm **510** stellt die Ausgabe jeder der drei Lichtkörper zu jedem Zeitpunkt dar, entsprechend der Zeitachse **515**. Auf einen Blick kann der Benutzer leicht bestimmen, welcher Effekt welcher Lichteinheit zu irgendeinem Zeitpunkt zugeordnet ist, was die Koordinierung der Effekte über verschiedene Lichteinheiten hinweg vereinfacht und eine schnelle Durchsicht der Lichtsequenz erlaubt.

[0038] Zusätzlich stellt [Fig. 5](#) eine Palette **520** dar, die die Bestandseffekte umfaßt, von denen ein Benutzer Lichteffekte auswählen kann, obwohl andere Techniken zum Bereitstellen der Menge von Bestandseffekten, wie durch ein Menü, eine Werkzeugleiste, etc. in den hierin beschriebenen Systemen und Vorrichtungen, angewendet werden können. In der Palette **520** sind Icons für Bestandseffekte für das Beleuchten eines Effekts mit fester Farbe **552**, ein Effekt mit einem Überblenden zwischen zwei Farben **554**, ein Effekt mit zufälliger Farbe **558**, ein Effekt **560**, ein Effekt mit einem Farbwechsel in Regenbogenfarben **565**, ein Stroboskopeffekt **564** und ein Blitzeffekt **568**. Diese Liste ist mitnichten erschöpfend und andere Typen von Effekten könnten umfaßt sein, so wie sie für einen Fachmann offensichtlich sind. Um einen Effekt einer Lichtquelle zuzuweisen, kann der Benutzer einen Effekt aus der Palette auswählen und einen Gitterabschnitt entsprechend der geeigneten Lichteinheit oder Einheiten und dem gewünschten Zeitintervall für den Effekt auswählen. Zusätzliche Parameter können durch eine geeignete Technik gesetzt werden, wie durch Eingeben numerischer Werte, Auswählen von Optionen aus einer Palette, einem Menü oder einer Werkzeugleiste, Ziehen eines Vektors, oder durch irgendeine andere Technik, die im Stand der Technik bekannt ist, wie das Parametereingabefeld **525**. Andere Schnittstellen und Techniken zum Eingeben von Lichtsequenzen, die geeignet sind, um einige oder alle der verschiedenen hierin beschriebenen Funktionen, durchzuführen, können verwendet werden und sind beabsichtigt, und sollen vom Umfang dieser Offenbarung umfaßt zu sein.

[0039] Die oben beschriebenen Verfahren können leicht angepaßt werden, um Einheiten, die nicht Lichteinheiten sind, zu steuern. Zum Beispiel können im Theaterumfeld Nebelmaschinen, Geräuscheffekte, Windmaschinen, Vorhänge, Seifenblasenmaschinen, Projektoren, Bühnenanwendungen (stage practicals), Bühnenaufzüge, pyrotechnische Geräte, Kulissen und beliebige andere Geräte, die von einem Computer gesteuert werden können, durch eine hierin beschriebene Sequenz gesteuert werden. Auf diese Weise können mehrere Ereignisse automatisiert und zeitlich gesteuert werden. Zum Beispiel kann der Benutzer die Lichtkörper programmieren, mit einem Aufhellen zu beginnen, so wie der Vorhang nach oben geht, gefolgt von dem Geräusch eines Gewehrschusses, während der Nebel über die Bühne wälzt. In einem Hause kann z.B. ein Programm verwendet werden, um Lichter anzuschalten und einen Wecker um 7.00 Uhr läuten zu lassen und die Kaffeemaschine **15** Minuten später anzustellen. Feiertagsbeleuchtungsanordnungen, wie auf Bäumen oder Häusern, können mit der Bewegung mechanischer Figuren oder musikalischen Aufzeichnungen synchronisiert werden. Eine Ausstellungs- oder eine Vergnügungsfahrt kann Gewitterschlag, Wind, Geräusche und Lichter in einem simulierten Gewittersturm koordinieren. Ein Gewächshaus, eine Viehscheune oder andere Umgebungen zum Aufziehen lebender Wesen können Umgebungslicht mit automatisierten Fütterungs- und Bewässerungsgeräten synchronisieren. Eine beliebige Kombination elektromechanischer Geräte kann von den hierin beschriebenen Systemen und Verfahren zeitgesteuert und/oder koordiniert werden. Solche Geräte können auf einer Schnittstelle zum Erstellen der Sequenz als zusätzliche Linien auf einem Gitter dargestellt werden, zum Beispiel eine Linie für jede unterschiedliche Komponente, die gesteuert wird, oder durch irgendein anderes geeignetes Mittel. Effekte dieser anderen Geräte können dem Benutzer auch visuell dargestellt werden. Zum Beispiel könnte eine kontinuierliche Verwendung einer Rauchmaschine andere Gitter austrüben, eine Kaffeemaschine könnte durch eine kleine Darstellung einer Kaffeemaschine, die augenscheinlich Kaffee brüht, auf der Schnittstelle dargestellt werden, sobald die Handlung an dem Gerät stattfindet, oder die Schnittstelle kann einen Balken zeigen, der langsamer werdend verändernde Farben zeigt, sobald Futter in einer Viehscheune ausgebracht wird. Andere solche statischen oder dynamischen Effekte dürften für einen Fachmann leicht ersichtlich sein und sind alle in diese Offenbarung eingebunden.

[0040] In bestimmten Ausführungsformen, bei denen sich die Lichteinheiten bewegen können, zum Beispiel durch Gleiten, Schwenken, Rotieren, Kippen usw., kann der Benutzer Anweisungen für die Bewegung der Lichteinheiten eingeben. Diese Funktion kann durch beliebige Mittel erreicht werden. Falls zum Beispiel die Lichteinheit einen Motor oder ein anderes System umfaßt, das Bewegung verursachen kann, kann die gewünschte Bewegung durch Auswählen eines Bewegungseffekts aus einer Menge von Bewegungseffekten, wie für die Lichteffekte oben beschrieben wurde, durchgeführt werden. Folglich kann z.B. eine Lichteinheit, die auf ihrer Basis rotieren kann, ausgewählt werden, und ein Regenbogenverwaschungseffekt kann programmiert werden, um gleichzeitig mit einem Drehbewegungseffekt stattzufinden. In anderen Ausführungsformen

können Lichteinheiten auf bewegliche Plattformen oder Träger montiert werden, die unabhängig von den Lichtkörpern z.B. durch Bereitstellen einer zusätzlichen Linie auf einer Gitterschnittstelle gesteuert werden können, wie oben beschrieben wurde. Bewegungseffekte können auch Parameter haben, wie Geschwindigkeit und Betrag (z.B. einen Winkel, einen Abstand, etc.), die vom Benutzer spezifiziert werden können. Solche Licht-/Bewegungskombinationen können in einer breiten Vielfalt von Situationen nützlich sein, wie Lichtshows, Planetariumspräsentationen, bewegende Scheinwerfer und ein beliebiges anderes Szenario, in dem programmierbare sich bewegende Lichter wünschenswert sein können.

[0041] Auf ähnliche Weise können Anweisungen zum Steuern von Objekten, die zwischen einer Lichteinheit und einem angeleuchteten Objekt plaziert werden, wie Gobos, Schablonen, Filter, Linsen, Blenden und andere Objekte, durch die Licht dringen kann, von einem Benutzer entsprechend der hierin beschriebenen Systeme und Verfahren bereitgestellt werden. Auf diese Weise kann sogar ein weiteres Feld von Lichteffekten gestaltet werden und für eine spätere Ausführung vorprogrammiert werden.

[0042] Eine Ausführungsform der hierin beschriebenen Systeme und Verfahren ist ein Computersystem, wie der in Fig. dargestellte Prozessor **10**, das konfiguriert ist, eine Lichtsequenz entsprechend der hierin beschriebenen Systeme und Verfahren zu gestalten, z.B. durch Ausführen eines Computerprogramms in einer Computersprache, die entweder interpretiert oder kompiliert wird, z.B. Fortran, C, Java, C++, etc. In einer zusätzlichen Ausführungsform beziehen sich die hierin beschriebenen Systeme und Verfahren auf eine Diskette, CD oder ein anderes permanentes computerlesbares Speichermedium, das ein Computerprogramm kodiert, das imstande ist, einige oder alle der oben beschriebenen Funktionen auszuführen, die einem Benutzer ermöglichen, eine Lichtsequenz zu erzeugen oder zu gestalten, die benutzt werden kann, um eine Vielzahl von Lichteinheiten zu steuern.

[0043] Eine Lichtsequenz kann auf einem Speichermedium, wie einer CD, Diskette, Festplatte, einem magnetischen Band, einem flüchtigem oder nichtflüchtigem Festspeichergerät oder irgendeinem anderen permanenten computerlesbaren Speichermedium aufgezeichnet werden. Die Lichtsequenz kann auf eine Weise gespeichert werden, die die Effekte und ihre Parameter, wie sie von einem Benutzer erzeugt wurden, speichert, auf eine Art und Weise, die dieses Format in ein Format konvertiert, das den endgültigen Datenstrom repräsentiert, z.B. geeignet zum direkten Steuern von Lichteinheiten oder anderen Geräten oder in irgendeinem anderen Format, das geeignet ist, die Lichtsequenz auszuführen. In Ausführungsformen, bei welchen die Sequenz als ein Datenstrom gespeichert ist, kann das System einem Benutzer erlauben, aus einer Auswahl von Datenformaten, wie DMX, RS-485, RS-232, usw. auszuwählen. Zusätzlich können Lichtsequenzen miteinander verbunden werden, z.B. so, daß am Schluss einer Sequenz eine andere Sequenz ausgeführt wird, oder es kann eine Mastersequenz zum Koordinieren der Ausführung einer Vielzahl von Untersequenzen erzeugt werden, z.B. auf der Grundlage externer Signale, Bedingungen, Zeit, Zufall, usw. In bestimmten Ausführungsformen kann eine Lichtsequenz **20** direkt von einem Prozessor **10** ausgeführt werden, obwohl in anderen Ausführungsformen eine Lichtsequenz **20** unter Verwendung einer Steuerung **30** ausgeführt werden kann, wie nachfolgend beschrieben wird.

[0044] Eine wie in [Fig. 6](#) dargestellte Steuerung **30** kann verwendet werden, um Lichtsequenzen **20** auszuführen, die auf einem anderen Gerät programmiert, gestaltet oder erzeugt wurden. Weil die Steuerung **30** ein engeres Spektrum an Funktionen als der Prozessor bereitstellen kann, der zur Erstellung der Sequenz verwendet wurde, kann die Steuerung **30** weniger Hardware umfassen und kostengünstiger sein als ein komplexeres System, das ein Entwickeln erlaubt, einen Videomonitor umfaßt oder andere Hilfsfunktionalitäten hat. Die Steuerung **30** kann irgendeine geeignete Ladeschnittstelle **610** einsetzen, um ein Lichtprogramm **20** zu erhalten, z.B. eine Schnittstelle zum Lesen eines Lichtprogramms **20** aus einem Speichermedium, wie einer CD, einer Diskette, einem magnetischen Band, einer Smartcard oder einem anderem Gerät, oder eine Schnittstelle zum Empfangen einer Übertragung von einem anderen System, wie eine serielle Schnittstelle, ein USB-Anschluß, ein paralleler Anschluß, ein Infrarotempfänger oder andere Verbindungen zum Empfangen eines Lichtprogramms **20**. In bestimmten Ausführungsformen kann das Lichtprogramm **20** über das Internet übertragen werden. Die Steuerung **30** kann auch eine Schnittstelle zum Kommunizieren mit mehreren Lichteinheiten **40** umfassen.

[0045] Eine Steuerung **30** kann die Ausführung einer Lichtsequenz **20** auf das Laden der Lichtsequenz **20** hin starten, auf den Empfang eines Befehls oder Signals von einem Benutzer oder einem Gerät oder einem Sensor, zu einer spezifizierten Zeit oder auf irgendeine andere geeignete Bedingung hin. Die Startbedingung kann in die Lichtsequenz **20** eingefügt werden, oder kann durch die Konfiguration der Steuerung **30** bestimmt werden. Zusätzlich kann die Steuerung in bestimmten Ausführungsformen eine Ausführung einer Lichtsequenz **20** beginnen, indem sie zu einem Zeitpunkt in der Mitte der Lichtsequenz **20** startet. Zum Beispiel kann die Steu-

erung **30** bei Empfang einer Anfrage vom Benutzer eine Lichtsequenz **20** ausführen, beginnend von einem Punkt drei Minuten vom Start der Lichtsequenz, oder zu irgendeinem anderen spezifizierten Punkt, z.B. vom fünften Effekt, etc. an. Die Steuerung **30** kann bei Empfang eines Signals von einem Benutzer oder einem Gerät oder Sensor das Abspielen pausieren und bei Empfang eines geeigneten Signals die Wiedergabe vom Punkt des Pausierens an fortsetzen. Die Steuerung kann fortfahren, die Lichtsequenz **20** auszuführen, bis die Sequenz terminiert, bis ein Befehl oder ein Signal von einem Benutzer oder einem Gerät oder Sensor empfangen wird, bis zu einem spezifizierten Zeitpunkt oder bis zu irgendeiner anderen geeigneten Bedingung.

[0046] Eine Steuerung **30** kann eine Speichereinheit, Datenbank oder irgendein anderes geeignetes Modul **620** zum Speichern mehrerer vorbestimmter Bestandseffekte und Instruktionen zum Konvertieren dieser Effekte in ein Datenformat, wie DMX, RS-485 oder RS-232 umfassen, das geeignet ist mehrere Lichteinheiten zu steuern. Das Speichermodul **620** kann für eine Menge von Bestandseffekten vorkonfiguriert sein, das Speichermodul **620** kann Effekte und Instruktionen von einer Lichtsequenz **20** aus empfangen, oder das Speichermodul **620** kann eine vorkonfigurierte Menge von Bestandseffekten umfassen, die durch zusätzliche in der Lichtsequenz **20** gespeicherte Effekte ergänzt werden können. Ein Vorkonfigurieren des Speichermoduls **620** mit einer Menge von Bestandseffekten erlaubt eine Reduzierung des Speichers, der benötigt wird, um eine Lichtsequenz **20** zu speichern, weil die Lichtsequenz **20** Konversionsinstruktionen für Effekte, die in der Steuerung **30** vorkonfiguriert ist, weglassen kann. In Ausführungsformen, bei welchen die Lichtsequenz **20** vom Autor entworfene Bestandseffekte umfaßt, können geeignete Anweisungen in die Lichtsequenz **20** eingefügt und im Speichermodul **620** gespeichert werden, z.B. beim Laden oder Ausführen der Lichtsequenz **20**.

[0047] Die Steuerung **30** kann eine externe Schnittstelle **650** umfassen, wobei die Steuerung **30** externe Signale empfangen kann, die zum Modifizieren der Ausführung der Lichtsequenz **20** nützlich sind. Zum Beispiel kann die externe Schnittstelle **650** eine Benutzerschnittstelle umfassen, die wiederum Schalter, Knöpfe, Wähl scheiben, Schieber, eine Konsole, eine Tastatur oder irgendein anderes Gerät, wie einen Sensor umfassen kann, wobei ein Benutzer einen Befehl oder ein Signal der Steuerung **30** bereitstellen kann oder auf andere Weise die Ausführung oder Ausgabe der Lichtsequenz **20** beeinflussen kann. Die externe Schnittstelle **650** kann zeitliche Information von einem oder mehreren Zeitmessern empfangen, wie einem lokalen Zeitmodul **660**, das wie ein Zähler zum Messen der Zeit von einem vorbestimmten Startpunkt aus funktioniert, wie wenn die Steuerung **30** angeschaltet wird oder wenn der Zähler zurückgesetzt wird, oder ein Datumszeitmodul **665**, das das aktuelle Datum und die Zeit berechnet. Zusätzlich kann die Steuerung **30** Befehle oder Signale von einem oder mehreren externen Geräten oder Sensoren durch die externe Eingabe **668** empfangen. Solche Geräte können an die Steuerung **30** direkt angekoppelt sein, oder Signale können von der Steuerung durch einen Infrarotsensor oder eine andere geeignete Schnittstelle empfangen werden. Von der Steuerung **30** empfangene Signale können mit einer Hinweistabelle **30** verglichen oder von dieser interpretiert werden, die Information in Bezug auf die unterschiedlichen Eingaben oder Bedingungen enthalten kann, die vom Autor der Lichtsequenz **20** dazu bestimmt sind, die Ausführung oder die Ausgabe der Lichtsequenz **20** zu beeinflussen. Falls folglich die Steuerung **30** eine Eingabe mit der Hinweistabelle **630** vergleicht und bestimmt, daß eine Bedingung erfüllt oder ein bestimmtes Signal empfangen wurde, kann die Steuerung **30** dann die Ausführung oder Ausgabe der Lichtsequenz **20**, wie vom Programm angegeben, verändern.

[0048] In bestimmten Ausführungsformen kann die Steuerung auch auf externe Signale antworten, die nicht durch die Inhalte und Instruktionen der Lichtsequenz **20** bestimmt sind. Zum Beispiel kann die externe Schnittstelle **650** eine Wähl scheibe, einen Schieber oder eine andere Einrichtung umfassen, durch die ein Benutzer die Ablaufgeschwindigkeit der Lichtsequenz **20** verändern kann, z.B. durch Verändern der Geschwindigkeit des lokalen Zeitzählers **660** oder durch Verändern der Interpretation dieses Zählers durch die Steuerung **30**. Auf ähnliche Weise kann das externe Interface **650** eine Einrichtung umfassen, durch die ein Benutzer die Helligkeit, Farbe oder eine andere Charakteristik der Ausgabe anpassen kann. In bestimmten Ausführungsformen kann eine Lichtsequenz **20** Instruktionen umfassen, um einen Parameter für einen Effekt von einer Einrichtung oder einer anderen Benutzerschnittstelle auf der externen Schnittstelle **650** zu empfangen, die eine Benutzersteuerung über spezifische Effekte während der Wiedergabe anstelle über die Ausgabe oder das System der Lichteinheiten als Ganzes erlaubt.

[0049] Die Steuerung **30** kann auch einen flüchtigen Speicher **640** umfassen. Der flüchtige Speicher **640** kann temporäre Information speichern, wie den aktuellen Zustand jeder Lichteinheit unter seiner Steuerung, was als eine Referenz für die Ausführung der Lichtsequenz **20** nützlich sein kann. Zum Beispiel können, wie oben beschrieben, einige Effekte Ausgaben von anderen Effekten verwenden, um einen Parameter zu bestimmen, solche Effekte können die Ausgabe des anderen Effekts abrufen, da sie er im flüchtigen Speicher **640** gespeichert ist. Fachleute werden andere Situationen erkennen, in denen ein flüchtiger Speicher **640** nützlich sein kann, und solche Verwendungen sollen von der vorliegenden Offenbarung umfaßt zu sein.

[0050] Die Steuerung **30** kann die von der Ausführung der Lichtsequenz **20** erzeugten Daten an Lichteinheiten schicken, indem sie die Daten der Netzwerkausgabe **680** bereitstellt, optional mittelbar über einen Ausgabepuffer **670**. Signale an zusätzliche Geräte können durch die Netzwerkausgabe **680**, oder durch eine separate externe Ausgabe **662**, übermittelt werden, so wie es geeignet oder gewünscht wird. Die Daten können übertragen werden durch Datenverbindungen, wie Drähte oder Kabel, wie IR- oder Funkübertragungen, andere geeignete Verfahren zum Datentransfer oder eine beliebige Kombination von Verfahren, die Lichteinheiten und/oder andere Geräte zu steuern können.

[0051] In bestimmten Ausführungsformen kann die Steuerung **30** nicht direkt mit den Lichteinheiten kommunizieren, sondern kann stattdessen mit einer oder mehreren Untersteuerungen kommunizieren, die wiederum die Lichteinheiten oder eine andere Ebene der Untersteuerungen etc. steuern. Die Verwendung von Untersteuerungen erlaubt eine verteilende Zuweisung von Berechnungsanforderungen. Ein Beispiel eines solchen Systems, das diese Art von verteilendem Schema verwendet, ist im US-Patent Nr. 5,769,527 von Taylor offenbart, das darin als ein "Master/Slave"-Steuersystem beschrieben wird. Für die hierin beschriebenen Systeme und Verfahren kann die Kommunikation zwischen den verschiedenen Ebenen unidirektional sein, wobei die Steuerung **30** Instruktionen oder Subroutinen bereitstellt, die von den Untersteuerungen ausgeführt werden, oder bidirektional, wobei Untersteuerungen Informationen an die Steuerung **30** zurück leiten, z.B. um Information bereitzustellen, die für Effekte, die sich auf die Ausgabe anderer Effekte, wie oben beschrieben wurde, für die Synchronisierung oder für irgendeinen anderen denkbaren Zweck nützlich sind.

[0052] Obwohl die obige Beschreibung eine spezielle Konfiguration einer Steuerung **30** darstellt, werden den Fachleuten andere Konfigurationen zum Erzielen derselben oder ähnlicher Funktionen offensichtlich sein, und solche Variationen und Modifikationen sollen von der vorliegenden Erfindung umfaßt sein. Das nachfolgende Beispiel beschreibt genauer eine Ausführungsform einer Steuerung **30**, wie sie oben beschrieben wurde.

[0053] Das Folgende beschreibt eine Ausführungsform einer Steuerung entsprechend der hierin beschriebenen Systeme und Verfahren, wie in [Fig. 6](#) beispielhaft dargestellt wird, einschließlich des Entwurfs und Formats einer Showrepräsentation, des Managements externer Ein- und Ausgaben, der Interpretation und Ausführung von Shows, und der Erzeugung von DMX-kompatiblen Ausgaben. Die Steuerungarchitektur dieser Ausführungsform verwendet ein Java-basiertes objektorientiertes Design; jedoch können andere objektorientierte, strukturierte oder andere Programmiersprachen mit der Erfindung verwendet werden.

[0054] Die Steuerungarchitektur erlaubt es, dass Effekte auf externen Umweltbedingungen oder anderen Eingabe basieren. Ein Effekt ist eine vorbestimmte Ausgabe, die eine oder mehrere Lichteinheiten umfaßt. Zum Beispiel sind eine feste Farbe, eine Farbverwaschung oder eine Regenbogenverwaschung alles Effekttypen. Ein Effekt kann ferner durch einen oder mehrere Parameter definiert werden, die z.B. zu steuernde Lichter, zu verwendende Farben, Geschwindigkeit des Effekts oder andere Aspekte eines Effekts spezifizieren. Die Umgebung bezieht sich auf irgendeine externe Information, die als Eingabe verwendet werden kann, um einen Effekt zu modifizieren oder zu steuern, wie die aktuelle Zeit oder externe Eingaben, wie Schalter, Knöpfe oder andere Signalumwandler, die Steuerungssignale erzeugen können, oder Ereignisse, die von anderer Software oder Effekten erzeugt wurden. Schließlich kann ein Effekt einen oder mehrere Zustände umfassen, so daß der Effekt Information über den Zeitverlauf speichern kann. Eine Kombination des Zustands, der Umwelt und der Parameter kann verwendet werden, um die Ausgabe eines Effekts zu irgendeinem Zeitmoment und über die Zeitdauer hinweg vollständig zu definieren.

[0055] Zusätzlich kann die Steuerung Effektprioritäten umsetzen. Zum Beispiel können unterschiedliche Effekte denselben Lichtkörpern zugewiesen werden. Durch Verwenden eines Prioritätsschemas wird nur der höchste Prioritätseffekt die Lichtausgabe bestimmen. Wenn mehrere Effekte einen Lichtkörper mit der gleichen Priorität steuern, kann die endgültige Ausgabe ein Durchschnitt oder eine andere Kombination der Effektausgaben sein.

[0056] Eine wie oben beschriebene Lichtsequenz kann als ein Programmfragment eingesetzt werden. Solche Fragmente können in ein Zwischenformat kompiliert werden, wie durch Verwenden eines verfügbaren Java-Compilers, um das Programm als Bytecode zu kompilieren. In so einem Bytecodeformat kann das Fragment eine Sequenz genannt werden. Eine Sequenz kann von der Steuerung **30** interpretiert oder ausgeführt werden. Die Sequenz ist kein alleinstehendes Programm und hält an einem definierten Format fest, wie eine Instantiierung eines Objekts aus einer Klasse, das die Steuerung **30** verwenden kann, um Effekte zu generieren. Wenn es in die Steuerung **30** (über eine serielle Schnittstelle, einen Infrarotanschluß, eine Smartcard oder irgendeine andere Schnittstelle) heruntergeladen wird, interpretiert die Steuerung **30** die Sequenz, wobei sie Teile basierend auf der Zeit oder Eingabeimpulsen ausführt.

[0057] Ein Baustein zum Erstellen einer Show ist ein Effektobjekt. Das Effektobjekt umfaßt Instruktionen zum Erstellen eines spezifischen Effekts, wie einer Farbverwaschung, ein Überblenden oder eine feste Farbe auf der Grundlage anfänglicher Parameter (z.B. welche Lichtkörper zu steuern sind, Anfangsfarbe, Auswaschperiode etc.) und Eingaben (wie Zeit, Umweltbedingungen oder Ergebnisse aus anderen Effektobjekten). Die Sequenz enthält die gesamte Information, um jedes Effektobjekt für die Show zu erzeugen. Die Steuerung **30** instantiiert alle Effektobjekte einmal, wenn die Show gestartet wird, dann aktiviert sie periodisch sequentiell jedes einzelne. Auf der Grundlage des Zustands des gesamten Systems kann jeder Effekt aufgrund des Programms entscheiden, falls und wie er die Lichtkörper, die er steuert, verändert.

[0058] Die auf der Steuerung **30** laufende Laufzeitumgebungssoftware kann als ein Dirigent bezeichnet werden. Der Dirigent kann verantwortlich sein für das Herunterladen von Sequenzen, Erzeugen und Aufrechterhalten einer Liste von Effektobjektinstanzen, Verwalten der Schnittstelle zu externen Eingaben und Ausgaben (einschließlich DMX), Verwalten der Uhr, und periodische Aufrufe jedes Effektobjekts. Der Dirigent pflegt auch einen Speicher, den Objekte verwenden können, um miteinander zu kommunizieren.

[0059] Die Steuerung **30** kann zwei verschiedene, aber synchronisierte Zeitrepräsentationen pflegen. Die erste ist LocalTime, welche die Anzahl an Millisekunden ist, seitdem die Steuerung **30** angeschaltet wurde. LocalTime kann als ein ganzzahliger 32-Bit Wert repräsentiert werden, der wieder von vorne beginnt, nachdem er seinen maximalen Wert erreicht hat. Die andere Zeitrepräsentation ist DateTime, die eine definierte Struktur ist, die die Tageszeit (bis zur Sekundenauflösung sowie Tag, Monat und Jahr) pflegt.

[0060] LocalTime kann von Effekten zum Berechnen relativer Veränderungen, wie ein Farbtonwechsel seit der letzten Ausführung in einem Farbverwaschungseffekt, verwendet werden. Das Wieder-von-vorne-beginnen von Local Time sollte nicht zu einem Ausfall oder einer Fehlfunktion von Effekten führen.

[0061] Der Dirigent kann Nutzenfunktionen für übliche Operationen, wie Zeitdeltas bereitstellen.

[0062] Ein Effektobjekt kann eine Instanz einer Effekt-Klasse sein. Jedes Effektobjekt kann zwei öffentliche Methoden bereitstellen, die von Effekt unterklassifiziert werden, um den gewünschten Effekt zu erzeugen. Diese sind die constructor- und die run()-Methoden.

[0063] Die constructor- Methode kann von einer Sequenz aufgerufen werden, wenn eine Instanz des Effekts erzeugt wird. Sie kann eine beliebige Anzahl und Typen von Parametern haben, die notwendig sind, um die gewünschten Effektvariationen zu erzeugen. Die Entwicklungssoftware kann für das Erstellen der zugehörigen Constructor-Parameter beim Erzeugen der Sequenz verantwortlich sein.

[0064] Das erste Argument für den constructor kann ein ganzzahliger Identifikator (ID) sein. Der ID kann von der Showentwicklungssoftware zugewiesen werden und kann eindeutig sein.

[0065] Der constructor kann super() aufrufen, um beliebige dirigentenspezifische Initialisierungen durchzuführen.

[0066] Die Effect-Klasse kann auch next- und prev- Mitglieder umfassen, die von der Sequenz und dem Dirigent verwendet werden, um eine verbundene Liste von Effekten zu halten. Auf diese Mitglieder kann von den Effektmethoden aus nicht intern zugegriffen werden.

[0067] Bestimmte typische Effekte können immer wieder verwendet werden. Diese typischen Effekte können vom Dirigenten bereitgestellt werden, wobei sie die Speicher/Herunterladegröße von Sequenzen minimieren. Die typischen Effekte können, falls gewünscht, weiter unterklassifiziert werden.

[0068] Eine Sequenz ist ein geeignetes Mittel, um die gesamte Information zu bündeln, die notwendig ist, um eine Show zu erzeugen. Die Sequenz kann nur eine benötigte öffentliche Methode init() haben, die einmal vom Leiter vor Ablauf der Show aufgerufen wird. Die init()-Methode kann jeden Effekt instantiiieren, der von der Show verwendet wird, wobei sie den ID und beliebige Parameter als constructor Argumente weitergibt. Die init()-Methode kann dann die Effektobjekte in eine verbundene Liste zusammenbinden und die Liste an den Dirigenten zurückgeben.

[0069] Die verbundene Liste wird durch die next- und prev-Mitglieder der Effektobjekte gepflegt. Das prev-Mitglied des ersten Objekts ist nil, und das nächste Mitglied des letzten Objekts ist nil. Der erste Effekt wird als der Wert von init() zurückgegeben.

[0070] Die optionale dispose()-Methode wird aufgerufen, wenn die Sequenz deaktiviert wird. Diese Methode kann verwendet werden, um irgendwelche von der Sequenz zugewiesenen Ressourcen zu erledigen. Automatische Prozesse können unabhängig verwendet werden, um einen zugewiesenen Speicher zu bearbeiten. Die Basisklasse dispose() wird die Effektobjekte durch die verbundene Liste weiterreichen und dieselben freigeben, so dass es notwendig sein kann, super() aufzurufen, wenn dispose() unterklassifiziert ist.

[0071] Die optionale öffentliche Methode String getSequenceInfo() kann verwendet werden, um Versions- und Copyrightinformation zurückzugeben. Es kann wünschenswert sein, einige zusätzliche getSequence*()-Routinen umzusetzen, um Information zurückzugeben, die für die Steuerungs/Benutzerschnittstelle nützlich sein kann.

[0072] Eine Sequenz kann zusätzliche Unterstützungsklassen benötigen. Diese können zusammen mit dem Sequenzobjekt in eine Datei, wie eine JAR (Java Archive) Datei, eingefügt werden. Die JAR-Datei kann dann an den Dirigenten heruntergeladen werden. Werkzeuge für JAR-Dateien sind Teil der Standard-Java-Entwicklungswerzeuge.

[0073] Jede beliebige DMX-Kommunikation kann von einer DMX_Interface-Klasse bearbeitet werden. Jede Instanz eines DMX_Interface steuert ein DMX-Universum. Die DMX_Interface-Grundklasse kann unterklassifiziert werden, um über eine spezifische Hardware-Schnittstelle zu kommunizieren (seriell, parallel, USB).

[0074] Ein Kanal kann ein einzelnes Datenbyte in einer bestimmten Stelle in dem DMX-Universum sein. Ein Rahmen kann alle Kanäle im Universum sein. Die Anzahl an Kanälen im Universum wird spezifiziert, wenn die Klasse instantiiert wird.

[0075] Intern hält das DMX_Interface drei Puffer, von denen jeder die Länge der Anzahl an Kanälen hat: Der letzte Rahmen von Kanälen, der geschickt wurde, der nächste Rahmen von Kanälen, der darauf wartet, geschickt zu werden, und die neueste Priorität der Daten für jeden Kanal. Effektmodule können die Kanaldaten modifizieren, die darauf warten, über die SetChannel()-Methode gesendet zu werden, und der Dirigent kann nach dem Rahmen fragen, der über SendFrame() gesendet werden soll.

[0076] Wenn ein Effektobjekt die Daten für einen speziellen Kanal setzt, kann es diesen Daten auch eine Priorität zuweisen. Falls die Priorität größer ist als die Priorität der letzten für diesen Kanal gesetzten Daten, dann können die neuen Daten anstelle der alten Daten treten. Falls die Priorität geringer ist, dann kann der alte Wert aufrechterhalten werden. Falls die Prioritäten gleich sind, dann kann der neue Datenwert zu einem laufenden Gesamtwert addiert werden und ein Zähler kann für diesen Kanal inkrementiert werden. Wenn der Rahmen gesendet wird, kann die Summe der Datenwerte für jeden Kanal durch den Kanalzähler geteilt werden, um einen durchschnittlichen Wert für die höchsten Prioritätsdaten zu erzeugen.

[0077] Nachdem jeder Rahmen gesendet wurde, können die Kanalprioritäten alle auf Null zurückgesetzt werden. Die zu sendenden Daten können gespeichert werden, so dass, falls keine neuen Daten für einen bestimmten Kanal geschrieben werden, sie ihren letzten Wert behalten, und auch in einen Puffer kopiert werden, für den Fall, daß irgendwelche Effektobjekte daran interessiert sind. Ein beispielhaftes DMX-Interface kann die folgenden Methoden umsetzen:

Eine DMX_Interface(int num_channels)-Methode ist ein Konstruktor, der ein DMX-Universum von num_channel (24..512)- Kanälen aufstellt. Wenn es unterklassifiziert wird, kann die Methode zusätzliche Argumente aufnehmen, um eine Hardwareanschlussinformation zu spezifizieren.

Eine void SetChannel(int channel, int data, int priority)-Methode setzt die zu sendenden Daten (0..255) für den Kanal, falls die Priorität größer als die aktuelle Datenpriorität ist. Die Methode kann fehlerbehandelnde Ausnahmen werfen, wie ChannelOutOfRange- und DataOutOfRange-Ausnahmen.

Eine void SetChannels(int first_channel, int num_channels, int data[], int priority)-Methode setzt num_channels von zu sendenden Daten zum Starten mit dem first_channel aus den Arraydaten. Die Methode kann fehlerbehandelnde Ausnahmen werfen, wie ChannelOutOfRange-, DataOutOfRange- und ArrayIndexOutOfBoundsException-Ausnahmen.

Eine int GetChannelLast(int channel)-Methode gibt die letzten Daten, die für den Kanal gesendet wurden, zurück. Die Methode kann fehlerbehandelnde Ausnahmen werfen, wie ChannelOutOfRange- oder NoDataSent-Ausnahmen.

Eine void SendFrame(void) Methode veranlaßt, daß der aktuelle Rahmen gesendet wird. Das wird durch einen separaten Thread erreicht, so daß das Verarbeiten durch den Dirigenten nicht pausieren wird. Falls ein Rahmen bereits im Ablauf ist, wird er beendet und der neue Rahmen wird gestartet.

Eine int FrameInProgress(void)-Methode gibt Null zurück, falls momentan kein Rahmen gesendet wird. Falls

ein Rahmen im Ablauf ist, gibt sie die Nummer des letzten gesendeten Kanals zurück.

[0078] Der Dirigent ist die Laufzeitkomponente der Steuerung, der die verschiedenen Daten und Eingabeelemente zusammenführt. Der Dirigent kann Sequenzen herunterladen, die Benutzerschnittstelle verwalten, den Zeitgeber und andere externe Eingaben und Sequenzen durch die aktiven Effektobjekte verwalten.

[0079] Die Technik zum Herunterladen der Sequenz-JAR-Datei in den Dirigenten kann variieren, abhängig von der Hardware und dem Transportmechanismus. Verschiedene Javawerkzeuge können verwendet werden, um das Java-Format zu interpretieren. In einer Ausführungsform können das Sequenzobjekt und verschiedene benötigte Klassen, zusammen mit einer Referenz auf das Sequenzobjekt, in den Speicher geladen werden.

[0080] In einer Ausführungsform können mehr als ein Sequenzobjekt in den Dirigenten geladen werden, und nur eine Sequenz kann aktiv sein. Der Dirigent kann eine Sequenz auf der Grundlage externer Eingaben, wie der Benutzerschnittstelle oder der Tageszeit, aktivieren.

[0081] Falls eine Sequenz bereits aktiv ist, wird die dispose()-Methode aufgerufen, bevor man für die bereits aktive Sequenz eine neue Sequenz aktiviert.

[0082] Um eine Sequenz zu aktivieren, wird die init()-Methode der Sequenz aufgerufen und bis zum Ende durchlaufen.

[0083] Die Steuerungen können einige Methoden aufrufen, um die Zeit zu messen. Auf Zeitwerte kann durch die LocalTime()- und GetDateTime()-Methoden zugegriffen werden. Auf andere Eingaben kann von einer ganzzahligen Referenzzahl zugegriffen und diese aufgezählt werden. Die Werte aller Eingaben können auch auf ganzzahlige Werte abgebildet werden. Eine GetInput(intref)-Methode gibt den Wert von Eingabe ref zurück und kann Ausnahmen, wie eine NoSuchInput- Ausnahme, werfen.

[0084] Die Effektliste kann von der init()-Methode der Sequenz erzeugt und zurückgegeben werden. In festen Intervallen kann der Dirigent sequentiell die run()-Methode jedes Effektobjekts in der Liste aufrufen.

[0085] Das Intervall kann spezifisch für die spezielle Steuerungshardware sein und kann veränderbar sein, z.B. durch eine externe Schnittstelle. Falls die Effektlistenausführung nicht in einem Intervallzeitraum beendet wird, kann die nächste Iteration bis zur vollen Intervallzeit verzögert werden. Effektobjekte müssen nicht jedes Intervall durchlaufen, um Veränderungen zu berechnen, sondern können eine Differenz zwischen dem aktuellen Zeitpunkt und dem vorhergehenden Zeitpunkt verwenden.

[0086] Effekte können gestaltet werden, um die Nutzung von Rechenpower zu minimieren, so daß die gesamte Effektliste schnell durchlaufen werden kann. Falls ein Effekt einen großen Berechnungsumfang benötigt, kann er einen Thread mit geringer Priorität veranlassen, die Aufgabe durchzuführen. Während der Thread läuft, kann die run()-Methode direkt zurückkehren, so daß die Lichtkörper unverändert bleiben. Wenn die run()-Methode entdeckt, daß der Thread beendet worden ist, kann sie die Ergebnisse verwenden, um die Lichtausgaben zu erneuern.

[0087] Der Speicher erlaubt unterschiedlichen Effekten, miteinander zu kommunizieren. Wie externe Ausgaben können Speicherelemente ganzzahlige Werte sein. Speicherelemente können durch zwei Informationsteile differenziert werden: Der ID des Effekts, der die Information erzeugte, und eine ganzzahlige Referenzzahl, die eindeutig für diesen Effekt ist. Die Zugriffsmethoden sind:

```
void SetScratch(int effect_id, int ref_num, int value),  
int getScratch(int effect_id, int5 ref_num)
```

[0088] Beide Methoden können fehlerbehandelnde Ausnahmen werfen, wie NoSuchEffect- und NoSuchReference- Ausnahmen.

[0089] Effekte können in beliebiger Reihenfolge ablaufen. Effekte, die Resultate aus anderen Effekten verwenden, können den Empfang von Resultaten aus der vorherigen Iteration erwarten.

[0090] Zusätzliche Routinen können das Folgende umfassen.

[0091] Eine in Delta Time(int last)-Methode berechnet die Veränderung in der Zeit zwischen dem aktuellen

und dem letzten Zeitpunkt.

[0092] Eine DMX_Interface_GetUniverse (int num)-Methode gibt das DMX_Interface-Objekt zurück, das mit der Universumsnummer num assoziiert ist. Dieser Wert sollte sich nicht ändern, während eine Sequenz läuft, damit er in den Cachespeicher aufgenommen werden kann. Die Methode kann fehlerbehandelnde Ausnahmen werfen, wie NoSuchUniverse-Ausnahmen.

[0093] Eine int HSBtoRGB(int hue, int set, int bright)-Methode konvertiert den Farbton (0-1553), die Sättigung (0-255), und die Helligkeit (0 bis 255) in Rot/Grün/Blauwerte, die auf die ersten drei Elemente des resultierenden Feldes geschrieben werden. Die Methode kann fehlerbehandelnde Ausnahmen werfen, wie ValueOutOfRange-Ausnahmen.

[0094] Eine int LightToDMX(int light)-Methode gibt die DMX-Adresse eines Lichtkörpers mit einer logischen Nummer des Lichtkörpers zurück. Die Methode kann fehlerbehandelnde Ausnahmen, wie DMXAdressOutOfRangeExceptions-Ausnahmen, werfen.

[0095] Eine void LinkEffects(Effect a, Effect b)-Methode setzt A.next=b; b.prev=a.

[0096] Jede Steuerung kann eine Konfigurationsdatei haben, die von der Showentwicklungssoftware verwendet wird. Die Konfigurationsdatei kann Abbildungen zwischen den ganzzahligen Eingaberefenzahlen und nützlicheren Beschreibungen ihrer Funktionen und Werte umfassen, z.B. etwas wie : Input2="Schieber" Bereich=(0-99). Die Konfigurationsdatei kann auch andere nützliche Information umfassen, wie eine Anzahl an DMX Universen.

[0097] Das Folgende ist ein Codebeispiel, das eine Lichtsequenz veranschaulicht, die entsprechend der Prinzipien der Erfindung verfasst wurde. Es ist verständlich, daß das folgende Beispiel in keiner Weise einschränkend ist:

Beispiel 1:

```
// Eine Beispieldsequenz.
// Durchläuft ein Band von 12 Seitenbeleuchtungskörpern,
// die sequentiell durchnummieret sind, beginnend mit
// Adresse 1
// Eingabe #1 ist ein Binärschalter
// Die Seitenbeleuchtung durchläuft eine kontinuierliche
// Farbverwaschung
// Wenn der Schalter geöffnet wird, wird ein
// Verfolger-Stroboskopeffekt angestoßen, der eine weiße
```

```
// Stroboskop-Seitenbeleuchtung durchläuft. Der Effekt wird
// solange nicht wiederholt, bis der Schalter zurückgesetzt
// wird.

Import java.sequence.*

public Class ExampleSequence extends Sequence {
    private int CoveGroup[] = {
        LightToDMX(1),LightToDMX(2),LightToDMX(3)
        LightToDMX(4),LightToDMX(5),LightToDMX(6)
        LightToDMX(7),LightToDMX(8),LightToDMX(9)
        LightToDMX(10),LightToDMX(11),LightToDMX(12)
    };

    public String getSequenceInfo() {
        return ``Example sequence version 1.0``;
    }

    public Effect init() {
        super.init();                      // Rufe Basisklasse init auf

        // Erstelle die Effektobjekte mit den geeigneten
        // Variationsparamtern

        washEff = new WashEffect(
            1,                         // ID
            CoveGroup, 1, 1,           // Welche Lichtkörper, Universum 1,
                                // Priorität 1
            true,                      // Richtung=vorwärts
            20000);                   // Geschwindigkeit (20 Sekunden)

        strobeEff=new ChaseStrobeEffect(
            2,                         // ID
            CoveGroup, 1, 2,           // Welche Lichtkörper, Universum 1,
                                // Priorität 2
            1,                         // Auslösereingabe
            true,                      // Richtung=vorwärts
```

```

100,           // Dauer des Stroboskopeffekt
               // (100 ms)
400,           // Zeit zwischen Stroboskopeffekten
               // (400 ms)
255,255,255); // Stroboskopeffektfarbe (weiß)

// Verbinde die Effekte
LinkEffects(washEff, strobeEff); // setzt next und prev

// Gib die Effekteliste an den Dirigenten zurück
return(washEff);
}

// Deklariere alle Effekte
WashEffect washEff;
ChaseStrobeEffect strobeEff;
}

// WashEffect kann als ein Bestandseffekt umgesetzt sein,
// aber wir werden eine einfache Version hier zur
// Veranschaulichung umsetzen.

public class WashEffect extends Effect {
    private int hue, sat, bright;
    private int last_time;          // letztes Mal, das wir
                                    // durchlaufen haben
    private int lights[];
    private DMX_Interface universe;
    private int priority;
    private boolean direction;
    private int speed;

    public WashEffect(int id, int lights[], int univ, int prio,
                      boolean dir, int speed) {
        // mache Kopien von Variationsparametern und initialisiere
        // beliebige andere Variablen
        this.lights=ligths;
        this.universe=GetUniverse(univ);
    }
}

```

```

this.prio=prio;
this.direction=dir;
this.speed=speed;
hue=0;
sat=255;
bright=255;
last_time=0;
super(id);

```

Patentansprüche

1. System zum Vorbereiten einer Lichtsequenz (**20**), welches einen Prozessor (**10**) umfasst, der zum Gestalten und Erzeugen der Lichtsequenz konfiguriert ist, indem er eine Anzeigenschnittstelle (**15, 300, 400, 500**) bereitstellt, die geeignet ist, um Information, die eine Vielzahl von Lichteffekten repräsentiert, darzustellen, und eine Sequenzautorenschnittstelle (**310, 320, 420, 520, 525**), die geeignet ist, um einen Benutzer zu gestatten einen Lichteffekt, eine Lichteinheit zum Ausführen des Lichteffektes, eine Startzeit für den Lichteffekt und eine Stopzeit für den Lichteffekt auszuwählen, wobei das System **dadurch gekennzeichnet** ist, dass die Anzeigenschnittstelle geeignet ist, ein Gitter anzuzeigen, wobei die oder jede Lichteinheit entlang einer Achse des Gitters dargestellt ist und die Zeit entlang der zweiten Achse des Gitters dargestellt ist, und wobei die Anzeigenschnittstelle geeignet ist, um den ausgewählten Lichteffekt auf einem Bereich des Gitters visuell darzustellen, der durch die Lichteinheit, die Startzeit und die Stopzeit, die dem ausgewählten Lichteffekt zugeordnet sind, definiert ist.
2. System nach Anspruch 1, bei welchem die Sequenzautorenschnittstelle geeignet ist, um Information, die eine Anordnung einer Vielzahl von Lichteinheiten repräsentiert, zu empfangen, und die Anzeigenschnittstelle geeignet ist, um eine Darstellung der Anordnung der Vielzahl von Lichteinheiten basierend auf der empfangenen Information visuell darzustellen.
3. System nach Anspruch 1 oder 2, bei welchem die Anzeigenschnittstelle geeignet ist, um eine von einem Benutzer erzeugte Lichtsequenz wiederzugeben.
4. System nach Anspruch 1, 2 oder 3, bei welchem die Lichteinheit eine aus einer Vielzahl von Lichteinheiten ist und bei welchem jede Lichteinheit einer eindeutigen Adresse zugeordnet ist.
5. System nach einem der vorhergehenden Ansprüche, bei welchem die oder jede Lichteinheit eine LED Lichteinheit umfasst, welche Licht einer beliebigen Farbe aus einem Bereich verschiedener Farben emittieren kann.
6. System nach einem der vorhergehenden Ansprüche, welches ferner ein Speichermedium umfasst, welches geeignet ist, Benutzerauswahlen zu speichern.
7. System nach einem der vorhergehenden Ansprüche, bei welchem die Sequenzautorenschnittstelle (**500**) geeignet ist, um dem Benutzer zu gestatten, eine Farbe (**552**) für den gewählten Lichteffekt auszuwählen.
8. System nach einem der vorhergehenden Ansprüche, bei welchem die Sequenzautorenschnittstelle (**500**) geeignet ist, um dem Benutzer zu gestatten, eine Startfarbe und eine Endfarbe für den gewählten Lichteffekt auszuwählen.
9. System nach einem der vorhergehenden Ansprüche, bei welchem die Sequenzautorenschnittstelle geeignet ist, um dem Benutzer zu gestatten, einen Übergangseffekt für einen Übergang zwischen einem ersten Lichteffekt und einem zweiten Lichteffekt auszuwählen.
10. System nach einem der Ansprüche 1 bis 8, bei welchem die Sequenzautorenschnittstelle geeignet ist, um dem Benutzer zu gestatten, eine Priorität für einen ersten Lichteffekt zu bestimmen, der eine zeitliche Überlappung mit einem zweiten Lichteffekt teilt.

11. System nach einem der vorhergehenden Ansprüche, bei welchem die Sequenzautorenschnittstelle geeignet ist, um dem Benutzer zu gestatten, eine Helligkeit für den gewählten Lichteffekt zu bestimmen.

12. System nach einem der vorhergehenden Ansprüche, bei welchem die Sequenzautorenschnittstelle geeignet ist, um dem Benutzer zu gestatten, Anweisungen bereitzustellen, um den gewählten Lichteffekt basierend auf einem externen Auslöseimpuls zu starten.

13. System nach einem der vorhergehenden Ansprüche, bei welchem die Sequenzautorenschnittstelle geeignet ist, um dem Benutzer zu gestatten, eine Bewegung der Lichteinheit zu bestimmen.

14. System nach einem der vorhergehenden Ansprüche, bei welchem die Sequenzautorenschnittstelle geeignet ist, um einem Benutzer zu gestatten, zumindest einen vom Benutzer zusammengestellten Lichteffekt zu gestalten.

15. System nach einem der vorhergehenden Ansprüche, welches ferner eine Steuerung (**30**) umfasst, die zum Ausführen der Lichtsequenz geeignet ist, um die zumindest eine Lichteinheit zu steuern.

16. System nach Anspruch 15, bei welchem die Steuerung zumindest ein Speichermedium (**620**) umfasst, um die Lichtsequenz in einem Datenformat zu speichern, das einen Datenstrom darstellt, der die zumindest eine Lichteinheit direkt steuern kann.

17. Verfahren zur Vorbereitung einer Lichtsequenz (**20**), die von einem Controller (**30**) ausgeführt werden kann, welches die Schritte umfasst:

Anzeigen von Information, die eine Vielzahl von Lichteffekten darstellt,

Auswählen eines Lichteffekts für die Lichtsequenz auf der Grundlage der angezeigten Information,

Auswählen einer Lichteinheit, um den Lichteffekt auszuführen,

Auswählen einer Startzeit für den ausgewählten Lichteffekt,

Auswählen einer Stopzeit für den ausgewählten Lichteffekt, wobei das Verfahren dadurch gekennzeichnet ist, ferner ein Gitter anzuzeigen, wobei die oder jede Lichteinheit entlang einer Achse des Gitters dargestellt ist und die Zeit entlang einer zweiten Achse des Gitters dargestellt ist, und

ferner den ausgewählten Lichteffekt auf einem Bereich des Gitters visuell darzustellen, der durch die Lichteinheit, die Startzeit und die Stopzeit, die dem ausgewählten Lichteffekt zugeordnet sind, definiert ist.

18. Verfahren nach Anspruch 17, welches ferner umfasst:

Empfangen von Information, die eine Anordnung einer Vielzahl von Lichteinheiten darstellt, und

Anzeigen einer Darstellung der Anordnung der Vielzahl von Lichteinheiten auf der Grundlage der empfangenen Information.

19. Verfahren nach Anspruch 17 oder 18, welches ferner das visuelle Darstellen des ausgewählten Lichteffekts umfasst.

20. Verfahren nach einem der Ansprüche 17 bis 19, welches ferner umfasst:

Auswählen einer zweiten Lichteinheit, und

Auswählen eines Lichteffekts für die Ausführung durch die zweite Lichteinheit.

21. Verfahren nach einem der Ansprüche 17 bis 20, welches ferner das Speichern von Benutzerauswahlen auf einem elektronischen Speichermedium umfasst.

22. Verfahren nach einem der Ansprüche 17 bis 21, welches ferner das Auswählen wenigstens einer Farbe für den gewählten Lichteffekt umfasst.

23. Verfahren nach einem der Ansprüche 17 bis 19, welches ferner das Auswählen eines zweiten Lichteffekts der Lichtsequenz auf der Grundlage der angezeigten Information, das Auswählen einer Startzeit für den zweiten Lichteffekt, und das Auswählen einer Stopzeit für den zweiten Lichteffekt umfasst.

24. Verfahren nach Anspruch 23, welches ferner das Auswählen eines Übergangseffekts zwischen dem ausgewählten Lichteffekt und dem zweiten Lichteffekt umfasst.

25. Verfahren nach Anspruch 20, 23 oder 24, welches ferner das Bestimmen einer Priorität für mehrere ausgewählte Lichteffekte umfasst.

26. Verfahren nach einem der Ansprüche 17 bis 25, welches ferner das Auswählen einer Helligkeit für den ausgewählten Lichteffekt umfasst.
27. Verfahren nach einem der Ansprüche 17 bis 26, wobei das Auswählen einer Lichteinheit das Auswählen einer Vielzahl von Lichteinheiten zum Ausführen des ausgewählten Lichteffekts umfasst.
28. Verfahren nach einem der Ansprüche 17 bis 27, wobei das Auswählen einer Lichteinheit das Auswählen einer LED Lichteinheit umfasst, die zum Emittieren von Licht einer beliebigen Farbe aus einem Farbbereich fähig ist.
29. Verfahren nach einem der Ansprüche 17 bis 28, wobei das Auswählen einer Startzeit das Bereitstellen von Anweisungen umfasst, um den ausgewählten Lichteffekt auf einem externen Auslöseimpuls basierend zu starten.
30. Verfahren nach Anspruch 18 oder 26 oder einem der Ansprüche 19 bis 29, soweit abhängig von diesen, wobei jede Lichteinheit einer eindeutigen Adresse zugeordnet ist.
31. Verfahren nach einem der Ansprüche 17 bis 30, welches ferner das Festlegen einer Bewegung der ausgewählten Lichteinheit umfasst.
32. Verfahren nach einem der Ansprüche 17 bis 31, welches ferner das Ausführen der Lichtsequenz umfasst, um die zumindest eine Lichteinheit zu steuern.
33. Verfahren nach einem der Ansprüche 17 bis 32, welches ferner das Speichern der Lichtsequenz in einem Datenformat umfasst, welches einen Datenstrom umfasst, der die zumindest eine Lichteinheit direkt steuern kann.
34. Computerlesbares Medium, das mit zumindest einem Programm codiert ist, das bei Ausführung das Verfahren nach einem der Ansprüche 17 bis 33 durchführt.

Es folgen 5 Blatt Zeichnungen

Anhängende Zeichnungen

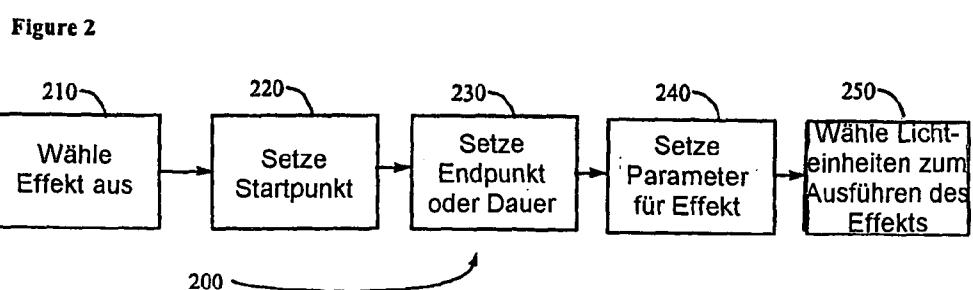
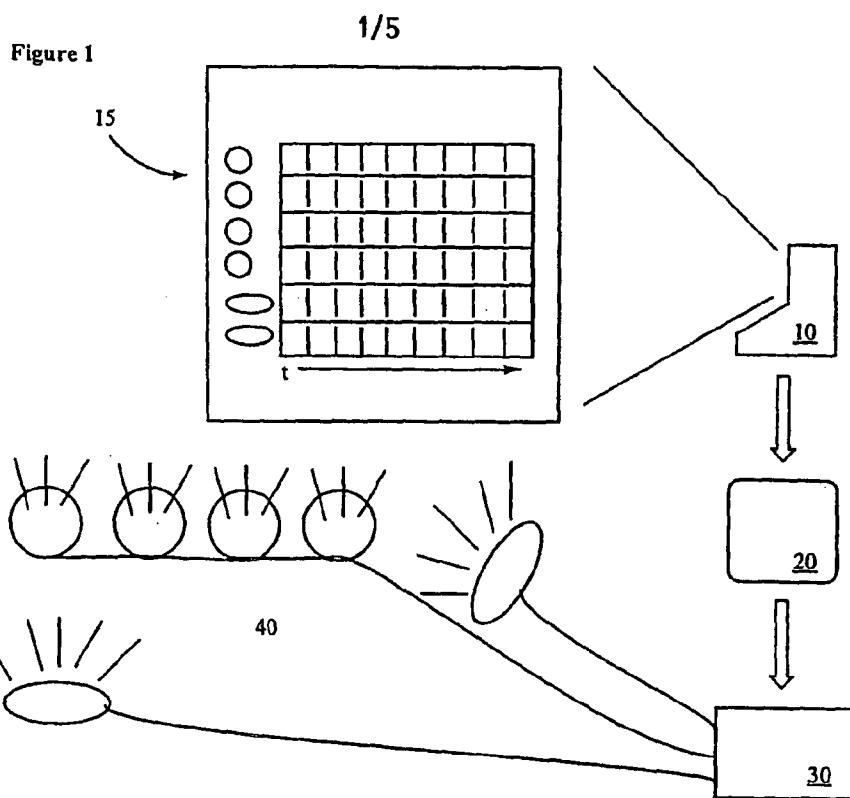


Fig. 3

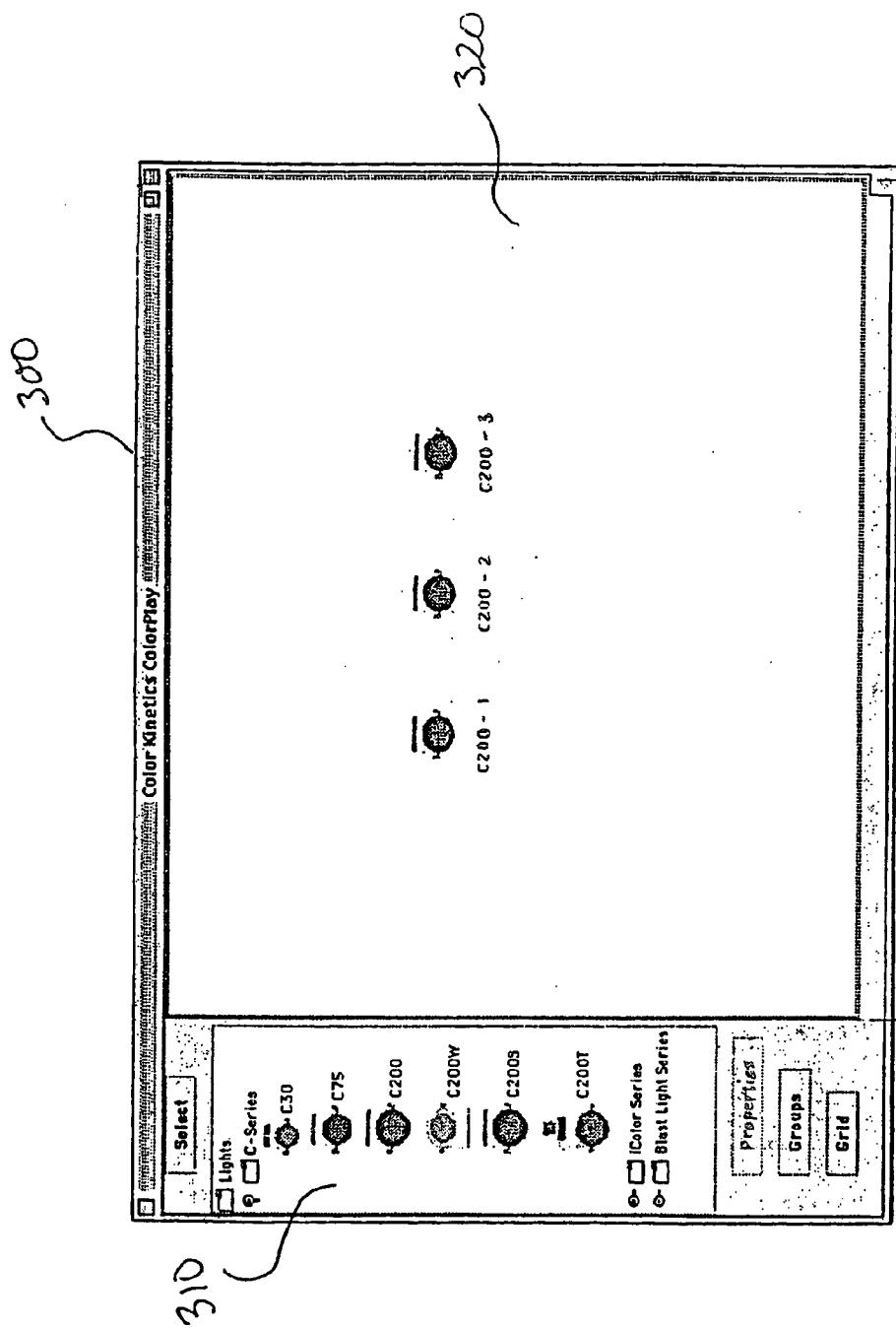
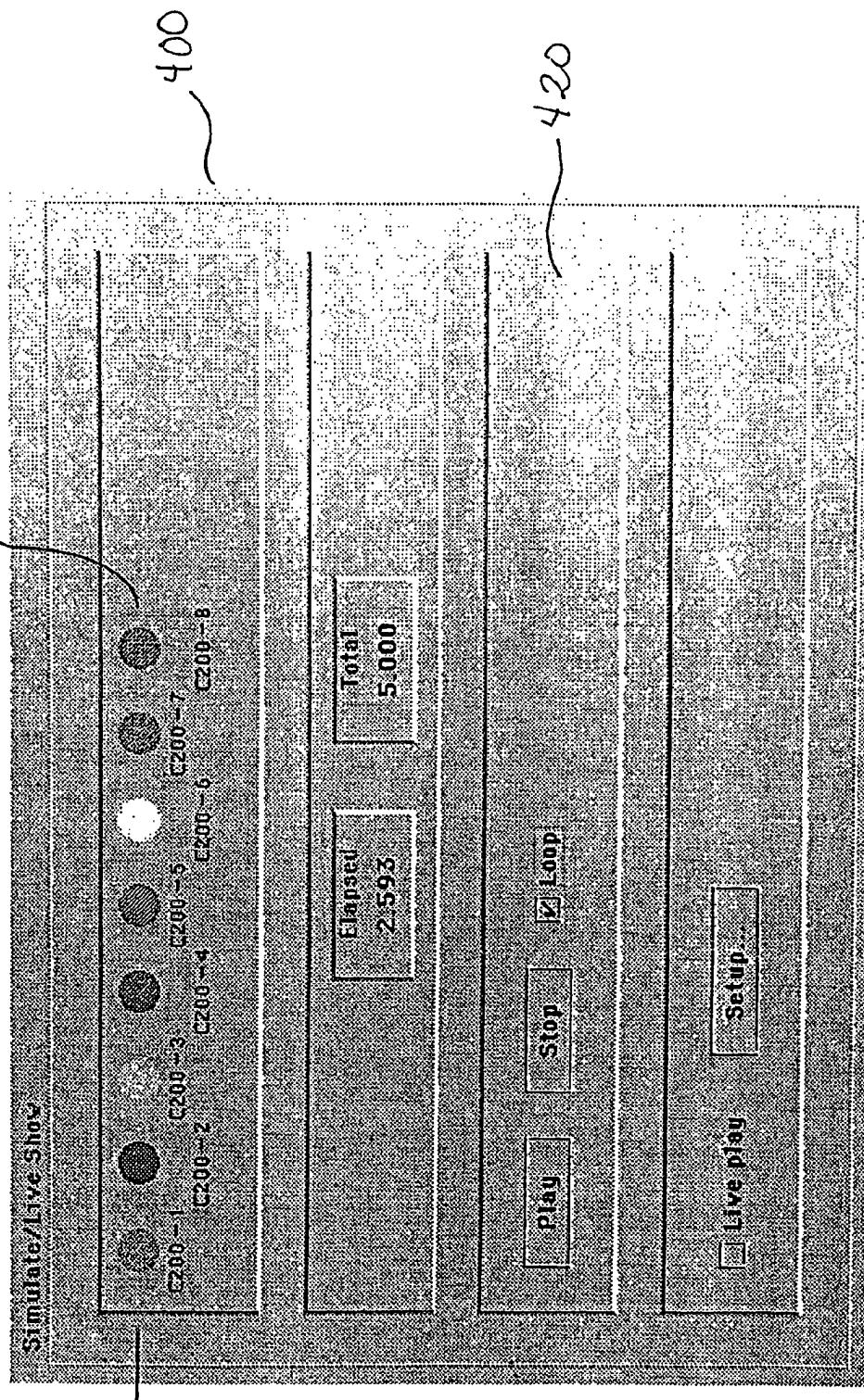


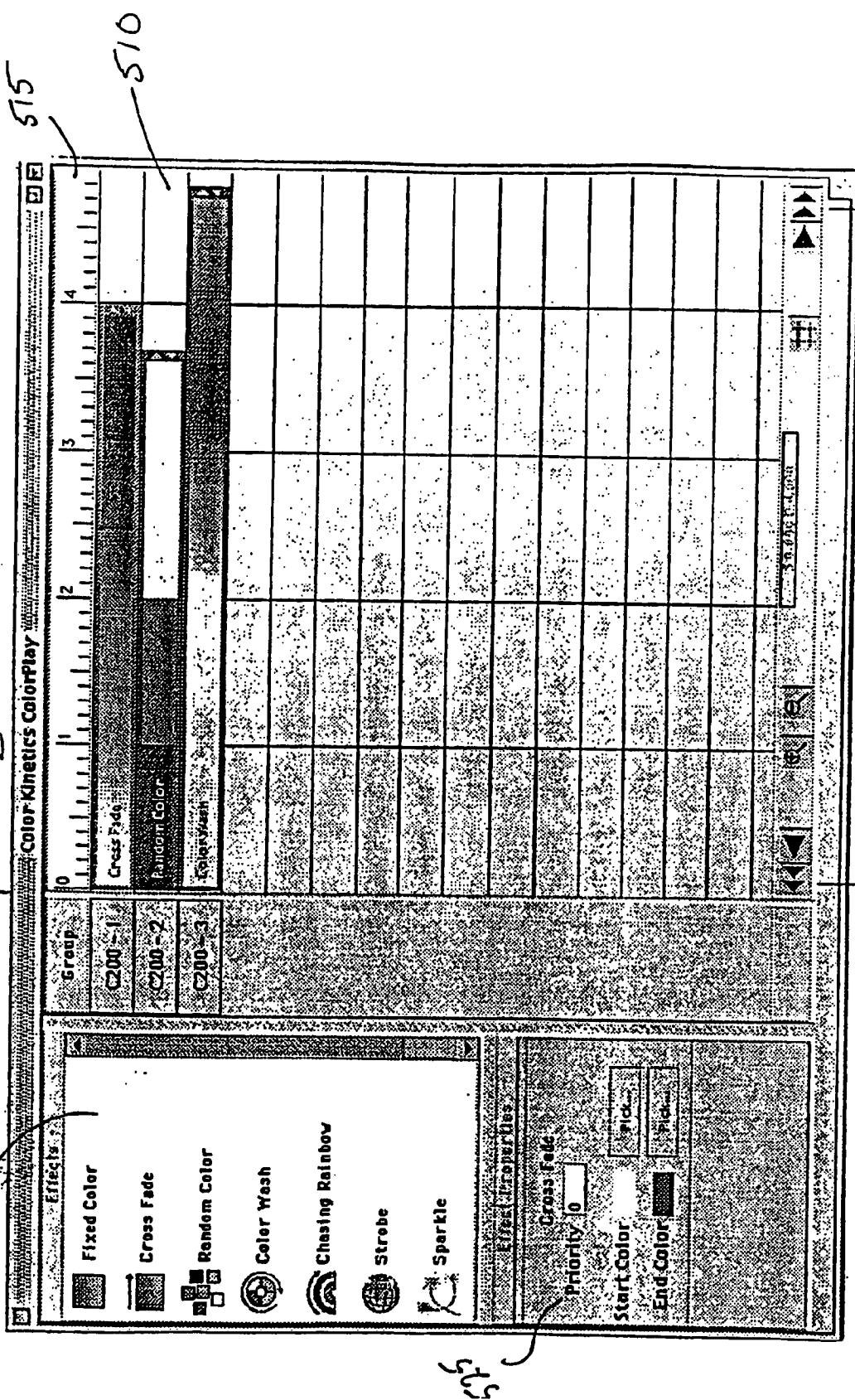
Fig. 4



500

Fig. 5

620



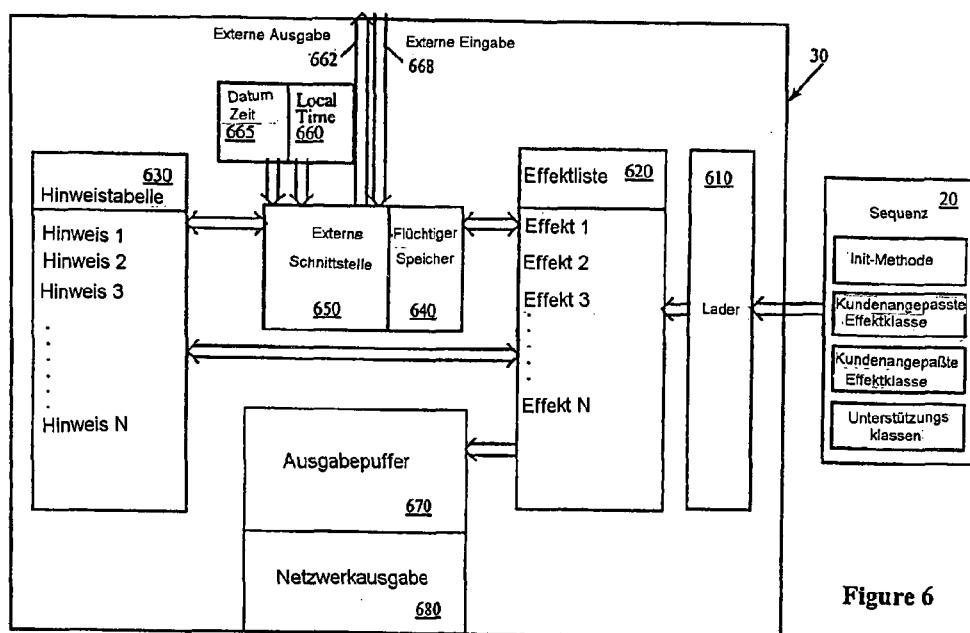


Figure 6