

DMXControl-Projects.org

Unreal Engine für DMX Visualisierung

Unreal Engine Actor mit DMX Control 3 kontrollieren



Ergebnisse AWE Lichtsteuerung in DMX an der HTW Berlin

Basierend auf Belegarbeiten von Roman Wiegand, Ellen Seeger, Robin Zitt, Philipp Nitsche
Stand Dez. 2023

editiert von Frank.Burghardt@dmxcontrol-projects.org

Inhaltsverzeichnis

1	Warum Unreal als Visualizer für DMX?.....	- 2 -
1.1	Vor & Nachteile von Unreal	- 2 -
1.2	Systemanforderungen	- 3 -
2	Installation und Grundlagen	- 4 -
2.1	Neues Projekt anlegen.....	- 5 -
2.2	Das grafische Interface.....	- 6 -
2.3	3D Steuerung	- 8 -
2.4	DMX-Visualisierung.....	- 9 -
2.5	Integration von DMXControl in Unreal	- 10 -
3	Setup & Workflow in Unreal	- 13 -
3.1	Kommunikation testen	- 14 -
3.2	Visualisieren	- 16 -
3.3	Das Starterset	- 21 -
4	Für Fortgeschrittene: Gerätemanagement	- 25 -
4.1	Allgemeines	- 25 -
4.2	Neue Geräte erstellen	- 25 -
4.3	Fixture Patch	- 28 -
4.3.1	Blueprint Classes.....	- 30 -
4.3.2	DMX Component.....	- 30 -
4.3.3	Beispiel.....	- 31 -
4.3.4	Actor.....	- 32 -
5	Schlusswort	- 35 -
5.1	Zusammenfassung.....	- 35 -
5.2	Die finale Show.....	- 36 -
6	Links.....	- 38 -

1 Warum Unreal als Visualizer für DMX?

Unreal (Engine) ist eines der fortgeschrittenen Echtzeit 3D Programme die derzeit auf dem freien Markt verfügbar sind. Es wird nicht nur für die Produktion von Indie und AAA Videospiele verwendet, sondern ist seit einigen Jahren für viele Film und Serien Produktionen ein fester Bestandteil geworden.

Ebenso gibt es viel Support und Entwicklung im Live-Event Bereich. Unreal bietet sogar ein vollständiges Template für DMX, welches einfach genutzt werden kann.

Im Rahmen einer DMX-Veranstaltung an der HTW Berlin haben Kursteilnehmer eine Show mit DMXControl3 entwickelt, bei der nicht wie üblich das in die Jahre gekommene EasyView-Tool als Visualizer verwendet wurde - sondern es wurde prototypisch die Visualisierung mit der Games Engine "Unreal Engine" (UE) von Epic Games (<https://www.unrealengine.com/de/>) realisiert.

Bei der Nutzung des Visualizers gibt es kein entweder-oder, sondern durch die Anbindung des Tools über ArtNet ist es eine zusätzliche Option, die auch genaugenommen gar nichts mit DMXControl zu tun hat.

Nun lernen wir, DMXControl3 mit Unreal zu verknüpfen und die Visualisierung kann beginnen.

1.1 Vor & Nachteile von Unreal

Vorteile:	Nachteile:
• Realistische und akkurate Darstellung einer DMX Show und einer Bühne • Reale Bühnen können ebenfalls visualisiert bzw. nachgebaut werden um eine 1 zu 1 Simulation zu erhalten • Unreal besitzt einen Langzeit Support und bietet Sicherheit für die Zukunft • Unreal ist Stand 2023 kostenlos für einfache Virtualisierungen im nicht kostenpflichtigen Rahmen • Möglichkeit größere Shows vorher zu visualisieren / Prototypen / designen ohne jegliche Art von Technik zu kaufen	• Ein gewisses Einarbeiten in Unreal ist nötig • Unreal hat hohe Mindestanforderungen der Hardware, aufgrund der Echtzeit Visualisierung • Unreal ist nicht unbedingt Anfängerfreundlich / Es gibt jedoch tausende von Tutorials und Foren zur Hilfe) • Im kommerziellen Rahmen fallen Kosten für die Nutzung von Unreal an • Ggf. müssen zusätzliche neue Gerätedefinitionen erstellt werden

Da die Unreal Engine (UE) für unsere privaten Zwecke kostenlos verwendet werden darf, kam die Idee auf, dieses Konzept der Visualisierung zu beschreiben und schrittweise die Kopplung zwischen DMXControl 3 und Unreal Engine zu vereinfachen. Die technische Kopplung ist über das ArtNet-Protokoll kein Problem, aber es muss natürlich eine Konsistenz zwischen den Geräten in der DMXControl Show und der Unreal Engine Darstellung ermöglicht werden, was zur Zeit einigen manuellen Aufwand gene-

riert. So wie sich bei DMXControl die Suche nach einem passenden DDF für ein Gerätschwierig gestalten kann, haben wir hier das Problem erneut für das UE-Format. Dazu dann im Kapitel 4 mehr.

Mit der Games Engine UE gibt es eine Vielzahl von Vorteilen bei diesem Konzept:

- bedeutend bessere Darstellungsqualität (Grafik) als bei EasyView
- Eintritt in die Virtuelle Realität, man kann die Show aus jeder Position betrachten, also Rundgang in einer 3D-Welt
- Darstellung spezieller Effekte wie LED-Matrizen, Feuerwerk, Nebelmaschinen, Laser, Konfetti-Maschinen etc.

Aufgrund der Nutzung von ArtNet könnten DMXControl3 und der Unreal-Visualizer auch auf separaten PCs laufen. Es scheint also schon extrem attraktiv, weitere Versuche mit DMXControl und der Unreal Engine vorzunehmen. Wir würden uns hierbei über aktive Mitarbeit aus der Community sehr freuen.

1.2 Systemanforderungen

Recommended Hardware

Operating System	Windows 10 64-bit version 1909 revision .1350 or higher, or versions 2004 and 20H2 revision .789 or higher.
Processor	Quad-core Intel or AMD, 2.5 GHz or faster
Memory	8 GB RAM
Graphics Card	DirectX 11 or 12 compatible graphics card
RHI Version	▪ DirectX 11: Latest drivers ▪ DirectX 12: Latest drivers ▪ Vulkan: AMD (21.11.3+) and NVIDIA (496.76+)

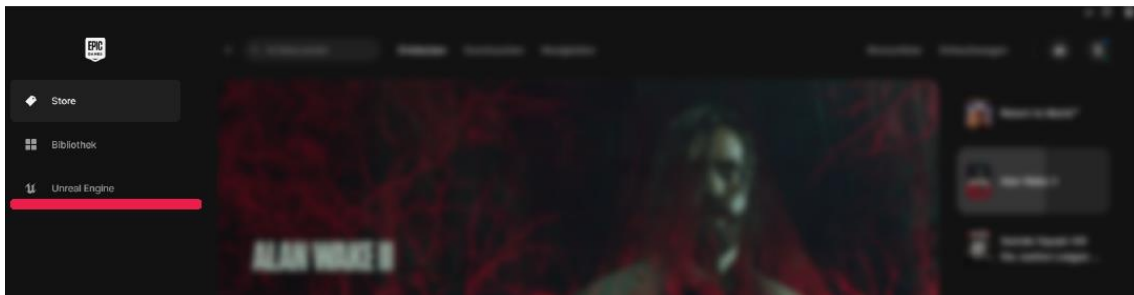
Zu beachten ist, dass während der Installation bzw. Update über 50GB auf der Festplatte erforderlich sind.

2 Installation und Grundlagen

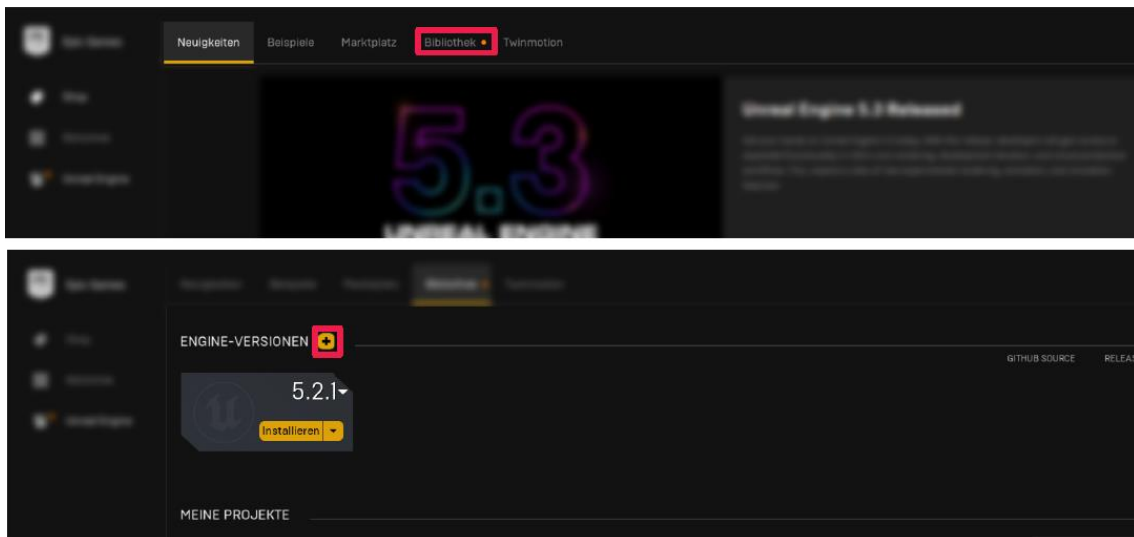
Um Unreal Engine nutzen zu können, muss das Programm über den Epic Games Launcher heruntergeladen werden. Hierfür muss man sich zunächst einen kostenlosen Epic Games Account erstellen und den Launcher installieren.

Im Launcher geht man nach dem Einloggen zunächst auf den Unreal Engine Tab, da der Shop irrelevant ist.

Link: <https://docs.unrealengine.com/5.3/en-US/installing-unreal-engine/>



Man gelangt nun auf die Startseite für die Unreal Engine. Von dort gehen wir nun in die Bibliothek um eine Engine Version hinzuzufügen. Eine neue Version kann über das + Symbol hinzugefügt werden.



Nun wählen wir bei der Version die neuste verfügbare aus. Das wäre zum aktuellen Zeitpunkt die 5.3.2. Der Installationsprozess ist von hier aus nun selbsterklärend.

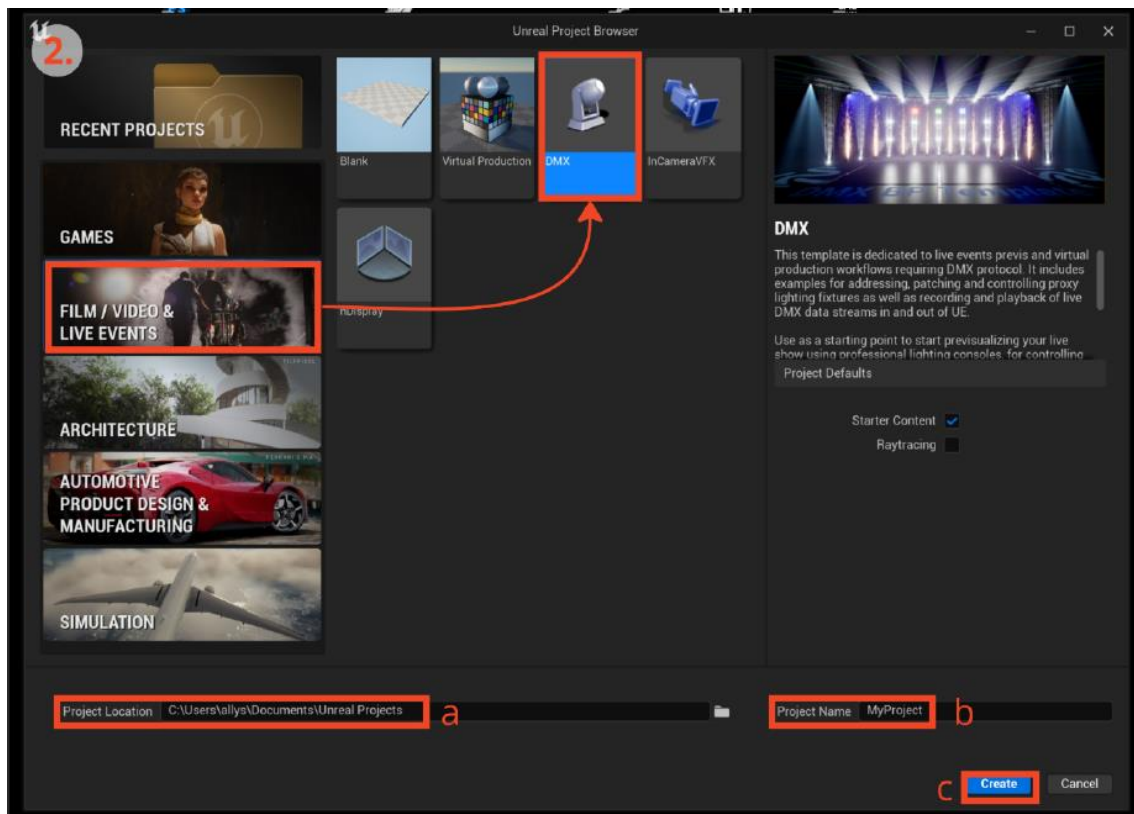
Die kostenlosen Plugins „DMX Engine“ und „DMX Protocol“ müssen installiert sein (siehe unten). „DMX Fixtures“ ist nicht zwingend notwendig, bietet aber eine gute Auswahl voreingestellter Assets.

Nach der Installation, können wir nun unser erstes neues Projekt aufsetzen.

2.1 Neues Projekt anlegen

Um ein neues Projekt zu erstellen, klicken wir zunächst auf „Starten“ bei der Engine-Version, welche wir soeben installiert haben. Es wird sich nach einem kurzen Moment der Project Browser öffnen. Dort kann man seine vorherigen Projekte zum Öffnen auswählen oder neue Projekte basierend auf vorgefertigten Templates erstellen.

Damit wir Unreal mit DMX(Control3) nutzen können, gehen wir auf den Tab für Film / Video & Live Events, worunter sich ein Template für DMX befindet. Dieses wählen wir nun aus.



Anschließend bestimmen wir noch einen Speicherpfad für das Projekt sowie einen Projektnamen und klicken danach auf „Create“.

Das Projekt wird nun erstellt und sobald es fertig geladen hat, haben wir ein DMX-Template Projekt geöffnet, welches wir nun mit DMXControl3 verbinden können.

Natürlich kann auch ein existierendes Projekt geladen werden.

Beim ersten Start kann das Laden je nach Prozessorleistung ziemlich lange dauern, da im Hintergrund sehr viele Shader compiliert werden.



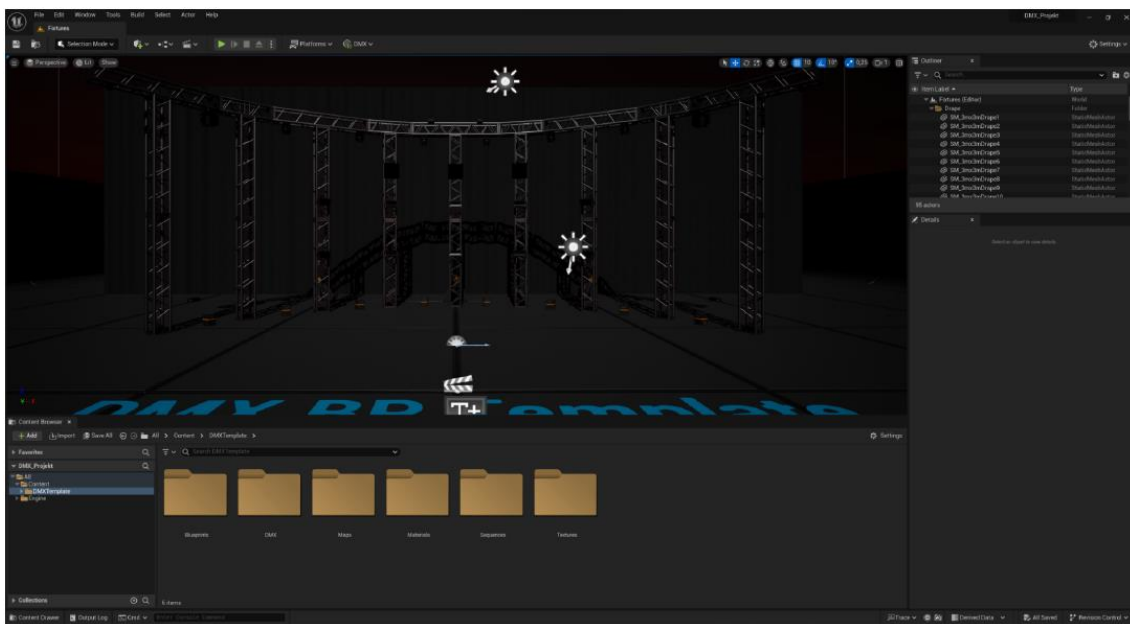
Hier noch einmal der Ablauf entsprechend der obigen Bilder (a-d):

DMX Template wählen (damit benötigte Plugins installiert, und Einstellungen konfiguriert sind).

- a. Speicherort für das Projekt wählen
- b. Projekt benennen
- c. Projekt erstellen
- d. **Tipp:** falls die im oberen Bild sichtbare orangene Mitteilung („d“) angezeigt wird:
Suche im Outliner (rechts) nach dem Begriff "directional light".
 - i. Teste durch Klicken der Augen-Symbole, welches der Lichter keinen Einfluss auf die Beleuchtung der Szene hat. (Wenn das Auge geschlossen ist, wird das Objekt im Anzeigen der Szene nicht berücksichtigt).
 - ii. Lösche das Directional Light das keinen Einfluss hat. (Jetzt sollte die orangene Mitteilung verschwunden sein).

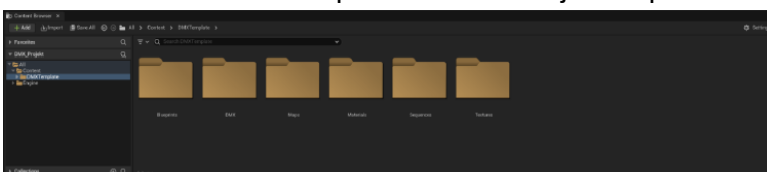
2.2 Das grafische Interface

Wenn man zuvor noch nie mit Echtzeit 3D Programmen oder Game Engines gearbeitet hat, kann das grafische Interface beim ersten Mal sehr überwältigend sein. Zum Glück muss man für die reine Visualisierung von DMX nicht sehr viel darüber wissen. Wir benötigen bloß das Wissen über die grundlegenden Bedienungsfenster und Funktionen, da wir ja kein Spiel generieren wollen.



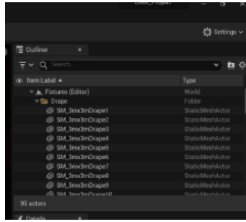
Was wir aber kennen sollten, um DMXControl3 mit Unreal nutzen zu können:

1. **Content Browser:** Äquivalent zum Projekt Explorer in DMXControl3



Wir können hier verschiedenste Dateien importieren, Unreal spezifische Assets wie zum Beispiel Blueprints erstellen oder vorhandene Dateien auswählen um diese in die Szene zu ziehen oder zu modifizieren.

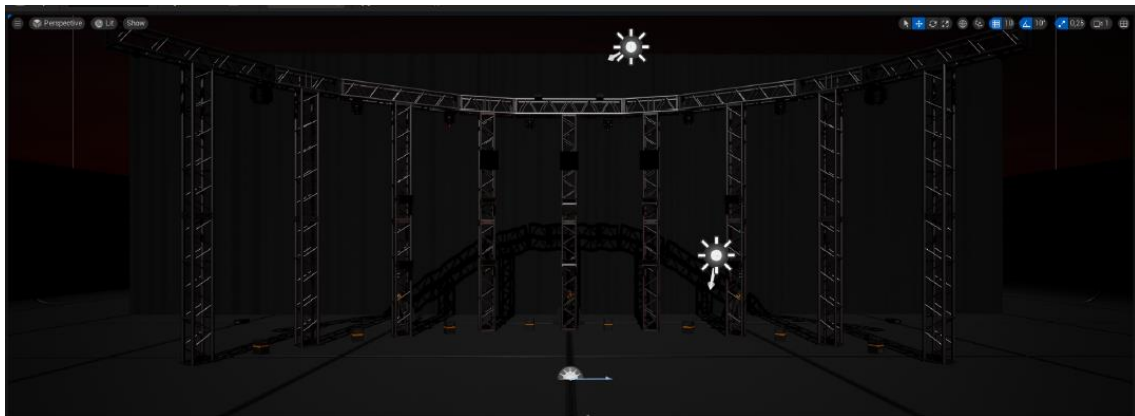
2. Outliner:



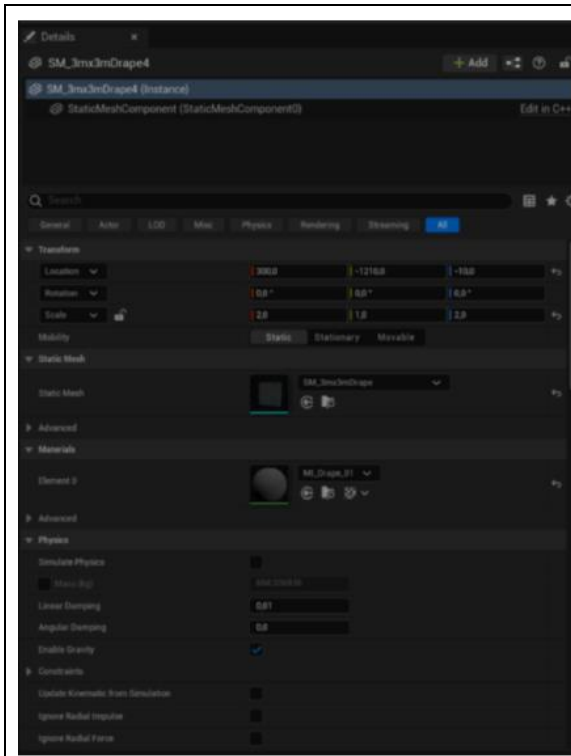
Eine Auflistung aller Objekte, die in der Szene platziert sind. Hier können wir nach bestimmten Objekten in der Szene suchen und entweder öffnen je nach Art des Objektes oder unter Details modifizieren.

3. Viewport:

Ein interaktives Fenster für die visuelle Darstellung der gebauten Szene. Hier findet die Visualisierung, der Aufbau und auch die Echtzeit Simulation der Show statt.



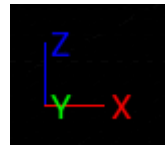
4. Details:



Das Details Panel bietet eine simplifizierte Auflistung an modifizierbaren Komponenten an, basierend auf dem, um was für eine Art Objekt es sich handelt.

Die für uns am relevanteste Komponente, die für alle Objekte gleich ist, ist die Transform Component, die für die Position, Rotation und Skalierung eines Objektes in der Szene zuständig ist.

Die Farben Rot, Grün und Blau repräsentieren die 3 Achsen X,Y und Z in dieser Reihenfolge.



Eine erwähnenswerte Sache ist ebenfalls, wenn ein Objekt sich in der Szene bewegen soll, muss bei Mobility entweder „Stationary“ oder „Movable“ ausgewählt sein, da sonst keine Bewegung simuliert wird. Das ist beispielsweise später relevant für Moving Heads.



Objekte die rein statisch in der Szene sind, sollten „Static“ ausgewählt haben, um Performance zu sparen.

2.3 3D Steuerung

Die Steuerung einer 3D Software wie zum Beispiel Unreal kann immer etwas variieren im Vergleich zu anderen Programmen. Was sie jedoch alle gemeinsam haben ist, dass sie einen Viewport nutzen in welchem man mit der Mausbewegung und Tastenkombination sich navigieren kann und Dinge in der Szene platzieren, verschieben und sonst noch alles kann. Um sich also mit Unreals Viewport vertraut zu machen, sollte man folgende Dinge wissen:

Rechtsklick halten + Maus Bewegung:

-> Rotieren der Viewport Kamera um die Y und X Achse, um sich in der Szene umzuschauen.

Rechtsklick halten + W , A, S oder D:

W -> Vorwärts
A -> Links
S -> Rückwärts
D -> Rechts

Ebenfalls kann die Bewegung der Kamera durch das Halten des Mausekzes + Bewegung der Maus beeinflusst werden. Dies ist allerdings weniger präzise und ist nur nützlich, wenn man sich nur anhand der X und Z Achse bewegen und die Höhe Y unverändert lassen möchte.

Rechtsklick + Mausekz hoch und runter:

-> Stellt die Kamera Zoom und Bewegungsgeschwindigkeit ein

Mausekz hoch und runter:

-> Kamera Zoom

Ebenso können durch die Viewport Symbole am Rand weitere Einstellung zur Darstellung des Viewports sowie für die Kamera getroffen werden.

Linke Maustaste:

-> Interaktionen mit einem Objekt in der Szene

Strg + D:

-> Duplizieren eines Objektes egal ob in der Szene, Content Browser oder in dem Blueprint Editor

Alt + Jegliche Art von Interaktion mit dem Transform Tool:

-> Erstellt eine Kopie des selektierten Objektes an Ort und Stelle und modifiziert dieses mit der bereits vollzogenen Bewegung.

2.4 DMX-Visualisierung

Welche Schritte sind für die Visualisierung nötig?

1. Ein neues Unreal Projekt mit dem DMX Template erstellen
2. Das DMXControl Projekt mit dem Unreal Projekt verbinden
3. Alle Geräte-Typen, die man im DMX-Projekt nutzt in einer DMX Library in Unreal eintragen/erstellen
4. Alle tatsächlichen Geräte, die in der Stage View vorhanden sind, in der DMX Library eintragen
5. Eine Szene in Unreal aufbauen
6. Den platzierten Geräten in Unreal das jeweilige Gerät aus DMX zuweisen
7. Fertig.

Gut zu wissen:

Wenn man einmal alle Lampen vorbereitet hat in Unreal und auch alles Nötige zugewiesen hat, braucht man nichts mehr zu ändern!

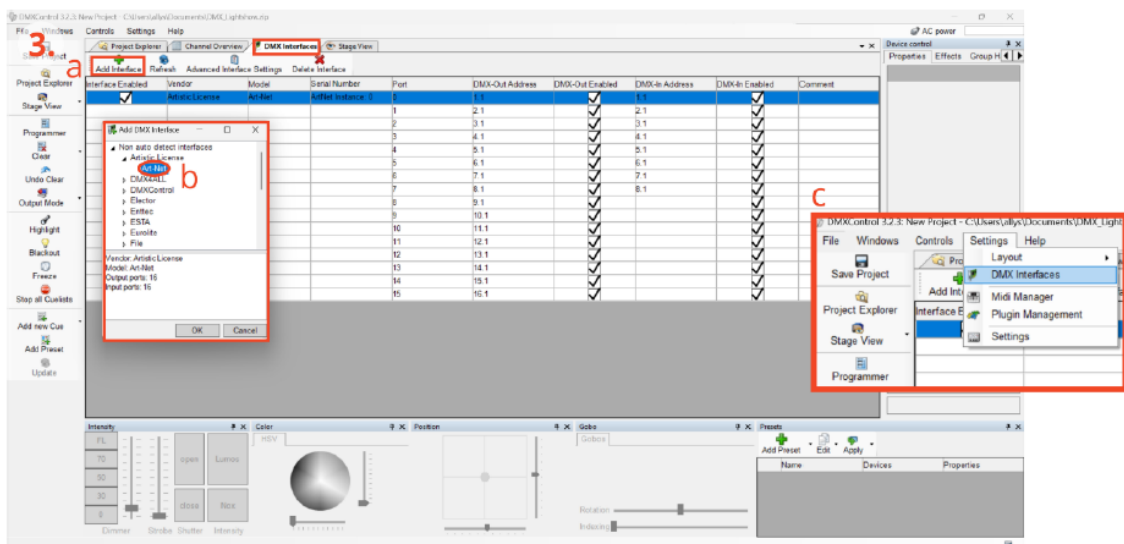
Alle Änderungen an der Lichtshow, werden am Ende ja nur von Unreal visuell dargestellt. Man muss nur dann in Unreal etwas ändern, wenn sich Geräte im DMX-Universe verschoben haben, neue Geräte hinzugefügt oder alte entfernt wurden.

Unreal ist einfach nur eine Spiegelung an Geräten aus dem DMX Projekt, mit dem Vorteil eine klarere Vorstellung der möglichen Show zu bekommen.



2.5 Integration von DMXControl in Unreal

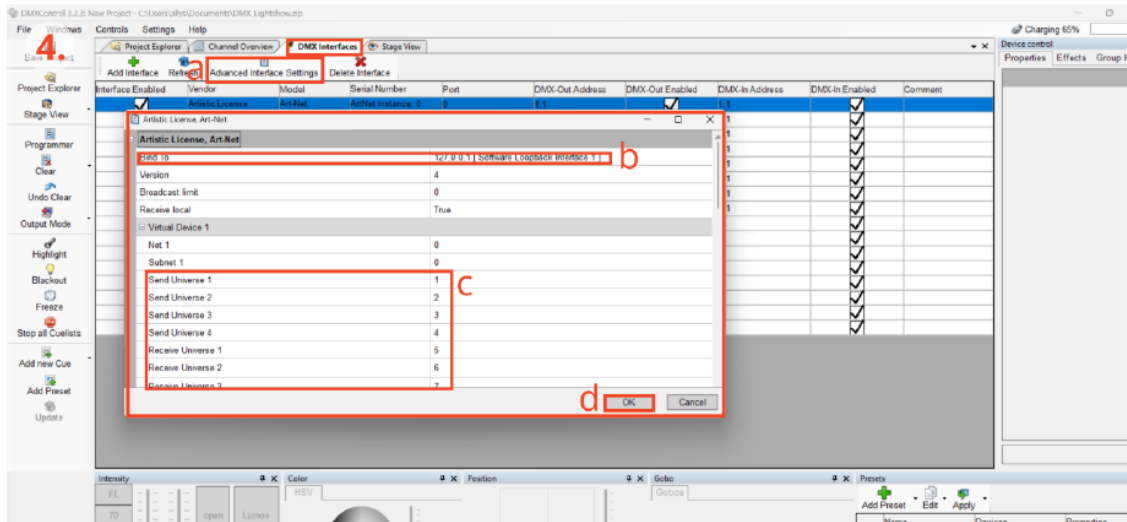
Zu aller erst müssen wir ein Interface in DMXControl mit der ArtNet Schnittstelle einrichten.



In DMX Control unter dem Fenster "DMX Interfaces" die ArtNet Schnittstelle hinzufügen:

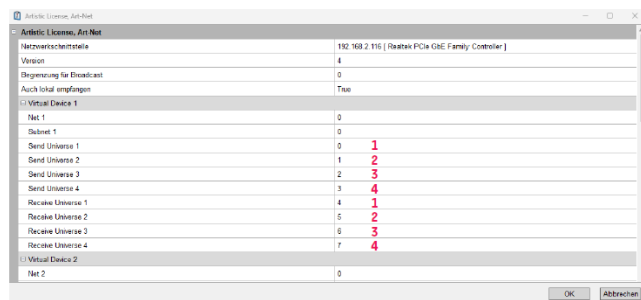
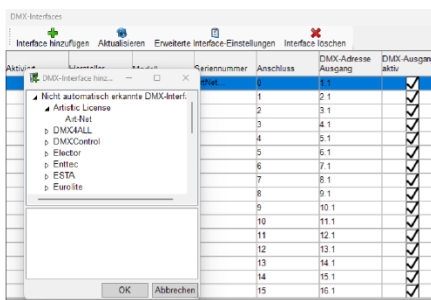
- Auf "Add Interface" klicken.
- In dem Fenster dass sich geöffnet hat, Art-Net unter dem Punkt "Artistic License" auswählen und mit "OK" bestätigen.

- d. **Tipp:** wenn das "DMX Interfaces" Fenster noch nicht geöffnet ist, einfach unter "Settings" > "DMX Interfaces" auswählen. Es sollte sich daraufhin öffnen.



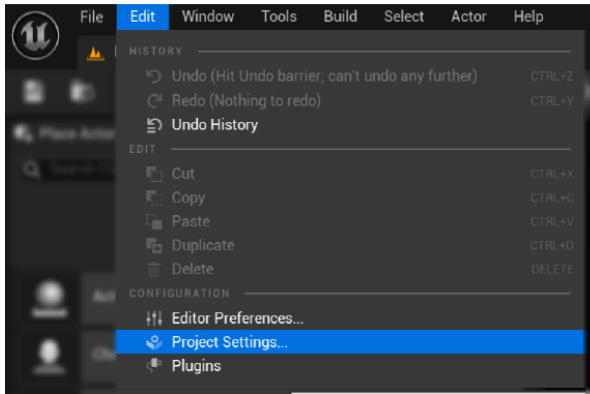
Advanced Interface Settings anpassen:

- Im "DMX Interfaces" Fenster auf "Advanced Interface Settings" klicken.
- Im neu geöffneten Fenster "Bind to" (Netzwerkinterface) mit Hilfe des drop-down Menüs die Adresse auf "127.0.0.1" einstellen (sofern UE auf dem gleichen Rechner läuft)
- "Send Universe 1" auf "1" stellen, "Send Universe 2" auf "2", "Send Universe 3" auf "3" usw. bis "Receive Universe 4". Im Interface, müssen Send Universe 1-4 und Receive Universe 1-4 von den Werten gleichgestellt werden. Sprich bei beiden 1-4 anstatt hochzählend bis 7.
- „OK“ drücken.

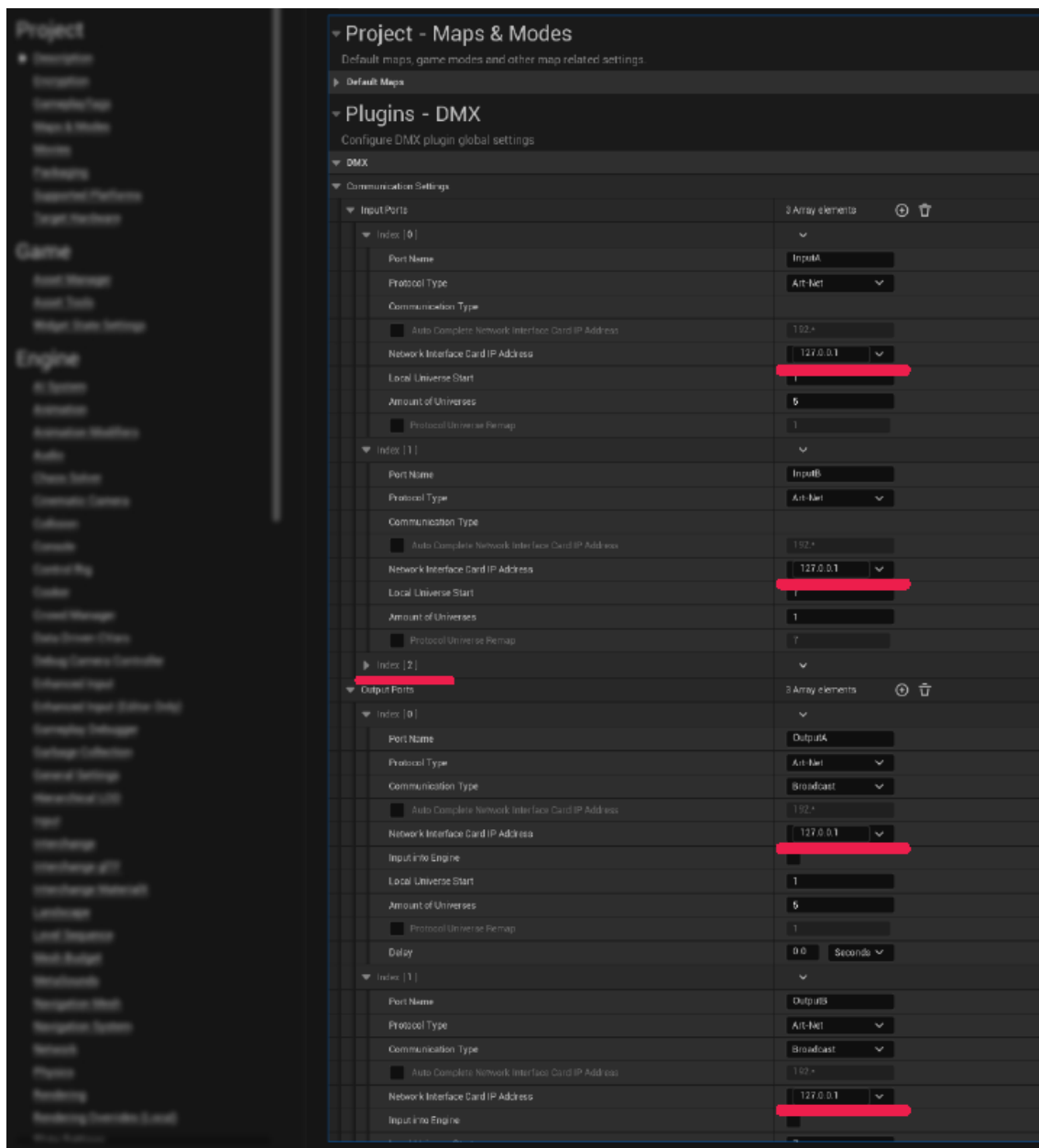


Bei der Verbindung von DMXControl3 und Unreal Engine über Art Net ist zu beachten, dass in den Geräteeinstellungen in DMXControl dieselbe IP Adresse angegeben ist wie in den Unreal Engine Projekt Settings unter Plugins - DMX. Außerdem ist es hilfreich zu wissen, dass in Unreal die DMX Universen bei 1 beginnen. Da das nicht geändert werden kann, muss es dementsprechend in DMXControl3 angeglichen werden.

Jetzt müssen wir nur noch Unreal einstellen. Dazu gehen wir unter dem „Edit“-Menüpunkt in die Project Settings in Unreal, geben in der Suchleiste unter Plugins „DMX“ ein und klappen die Input & OutputPorts auf.



In allen zugeklappten Ports, müssen wir nur unter „Network Interface Card IP Address“ die IP 127.0.0.1 auswählen.



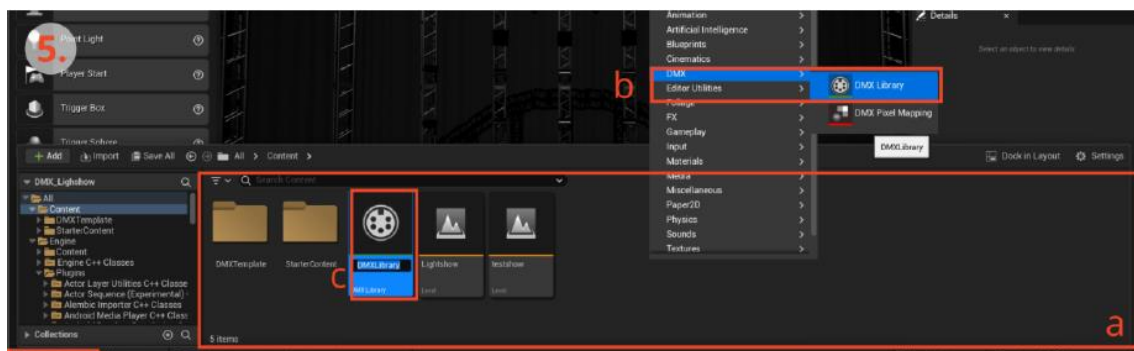
Sobald das bei Unreal eingestellt ist, ist die Verbindung zwischen DMXControl3 und Unreal über ArtNet hergestellt. Nun benötigen wir nur noch unsere Show Anforderungen und müssen alle unsere Geräte in Unreal noch hinzufügen.

3 Setup & Workflow in Unreal

Um unsere Geräte hinzufügen zu können, erstellen wir durch Rechtsklick im Content Browser unter DMX -> eine DMX Library. Diese können wir beliebig benennen. Anschließend öffnen wir diese und wir bekommen nun unseren Fixture Editor.

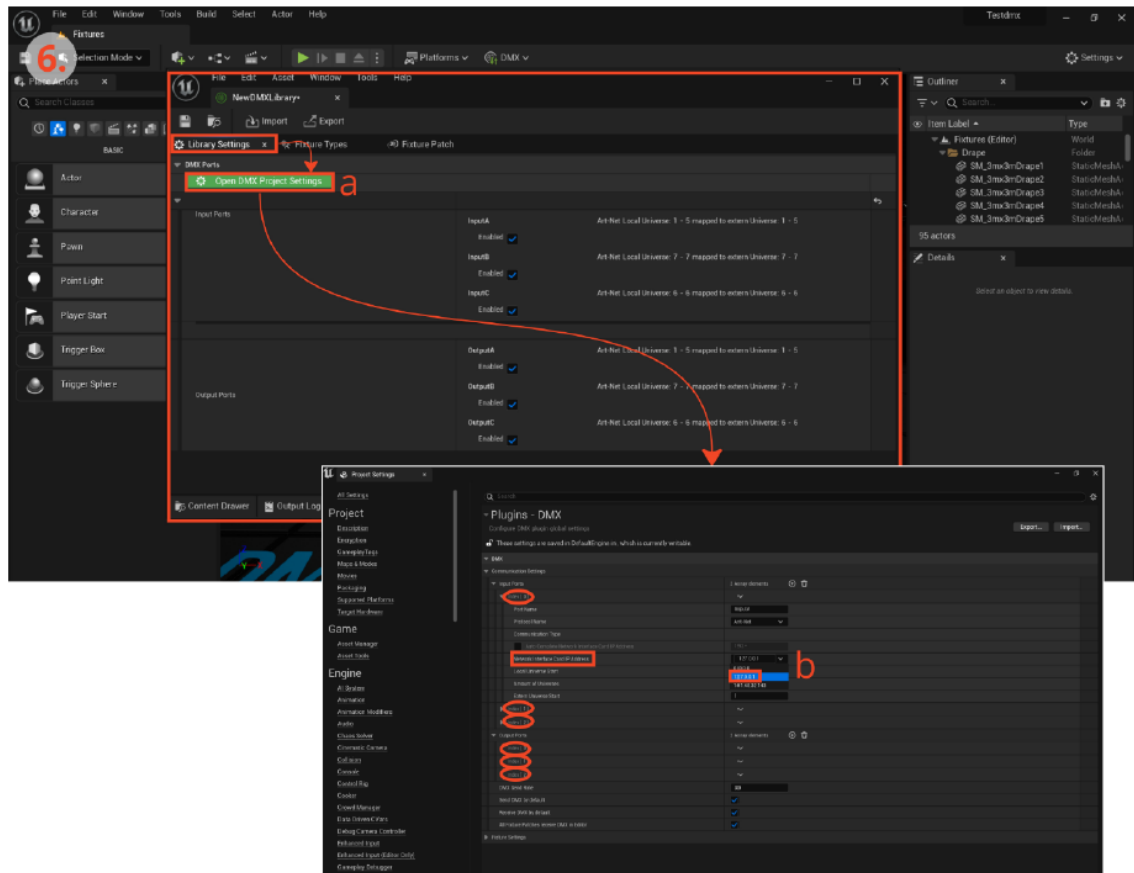
DMX Library

Um die DMX-Signale, die über Art Net zu Unreal gesendet werden, verwenden zu können, wird ein DMX Library Asset als Zwischeninstanz benötigt. Die DMX Library kann im Content Browser mit Rechtsklick > DMX > DMX Library erstellt werden.



In UE 5.3 den „Content Drawer“ öffnen (z.B. mit Strg. + Leertaste)

- Im neu geöffneten Fenster auf die leere Fläche rechts-klicken
- Im auftauchenden Menü unter "DMX" "DMX Library" auswählen
- Neu erstellte Library nach Belieben benennen



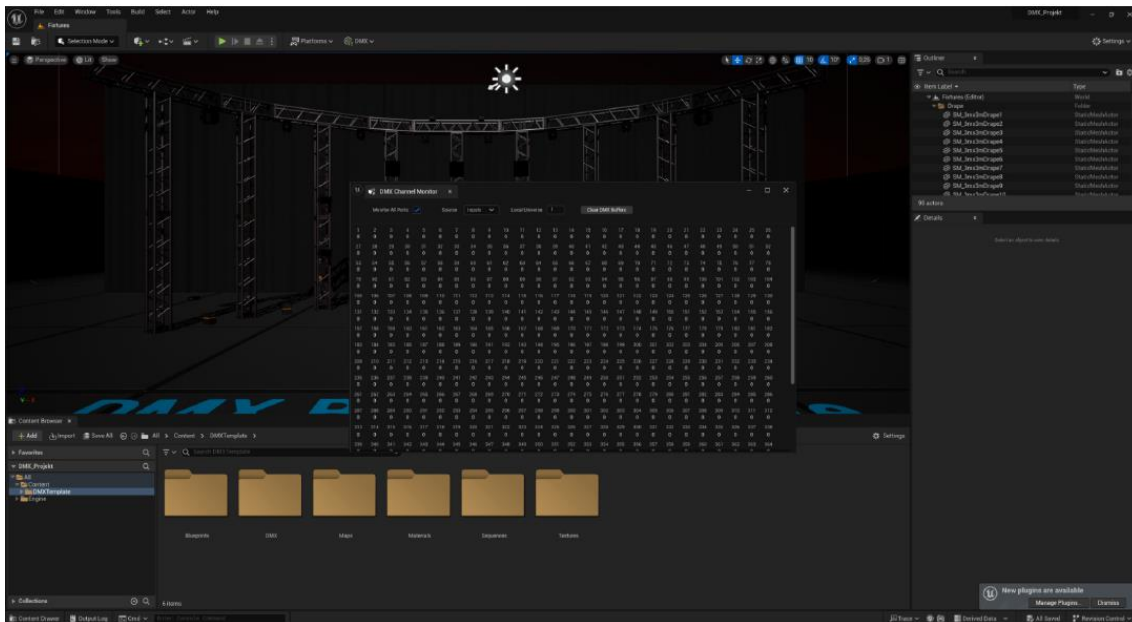
DMX Library mit Doppelklick öffnen:

- Im neu geöffneten Fenster unter dem Reiter "Library Settings" "Open DMX Project
- Settings" anklicken.
- Zur Überprüfung nochmals im dadurch erschienenen Fenster wie oben beschrieben bei **allen** input- und output-indexes, "127.0.0.1" als "Network Interface Card IP Adress" angeben.

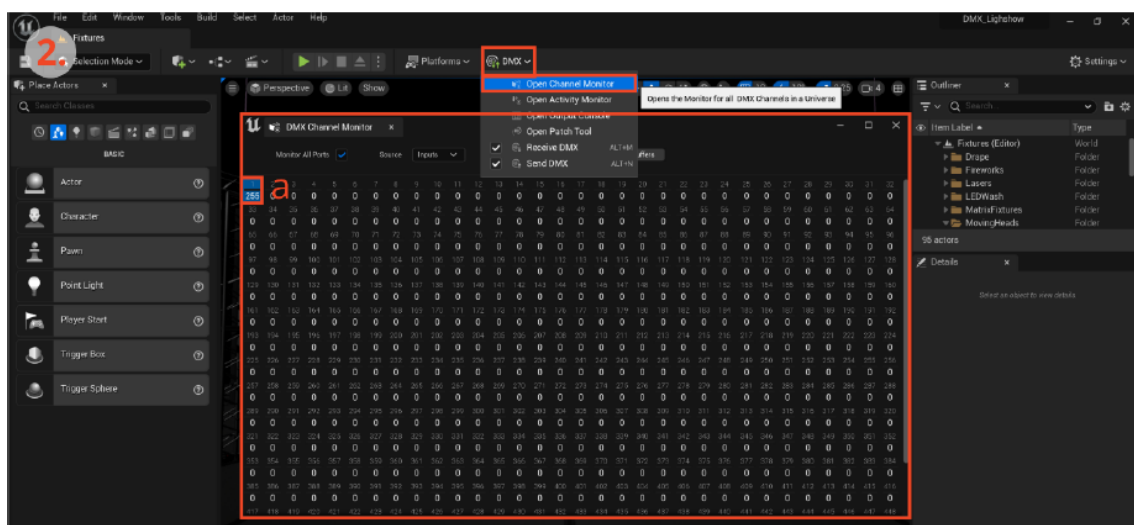
3.1 Kommunikation testen

In DMXControl die „Stage View“ öffnen.

- Mit Rechtsklick auf den leeren Space im "Stage View" Fenster das Menü aufrufen und "+ Add Device" auswählen.
- Im erschienenen Fenster eine Lampe der Wahl aussuchen (z.B. generic dimmer).
- Einmal auf „+ Add“ drücken
- Fenster schließen
- Doppelklick auf den dimmer, damit er an geht
- Optional im "channel overview" Reiter checken, ob auf Kanal 1 der "generic dimmer" mit dem Wert "255" gesendet wird.



Unter dem Menüpunkt „DMX“ kann der Channel Monitor ausgewählt werden, der der Kanalübersicht in DMXControl3 entspricht.



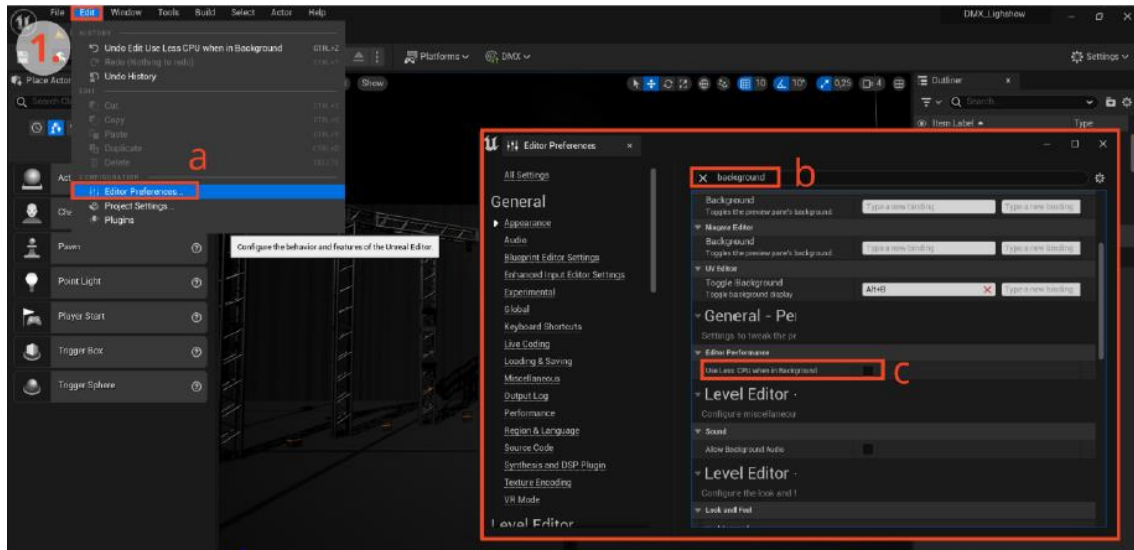
In UE 5.3 den "Channel Monitor" über das drop-down Menü „DMX“ am oberen Rand öffnen

- a. Checken ob auf Kanal 1 ebenfalls der Wert 255 gesendet wird. (Wenn ja, dann ist die Kommunikation zwischen DMX und UE erfolgreich. Wenn nicht, prüfe nochmal, ob die Schritte unter 2.5 richtig und vollständig durchgeführt wurden).

Wenn die Performance der Grafik nicht ausreichend ist, hilft manchmal auch ein Klick auf den Button „Clear DMX Buffer“ im UE Channel Monitor oder siehe Tipp in Kap. 3.2.

3.2 Visualisieren

Am Beispiel eines Clay Paky Sharpy Movingheads soll die Visualisierung getestet werden.



TIPP: damit Unreal in Echtzeit/direkt anzeigt, was man in DMXControl3 verändert, muss man noch ein Häkchen in den Einstellungen raus nehmen.

- In UE über die Menüleiste ganz oben "Edit", und dann "Editor Preferences" auswählen.
- Im neu geöffneten Fenster in der Suchleiste "background" eingeben und runter scrollen.
- Den Haken bei "Use Less CPU when in Background" rausnehmen.

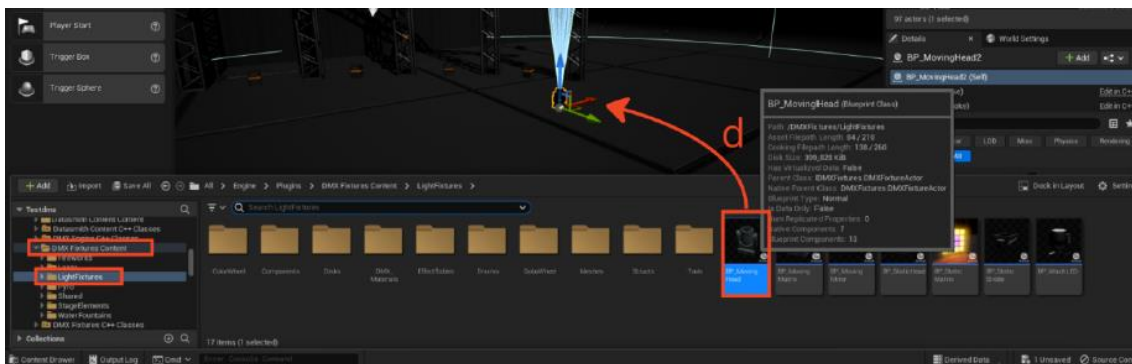
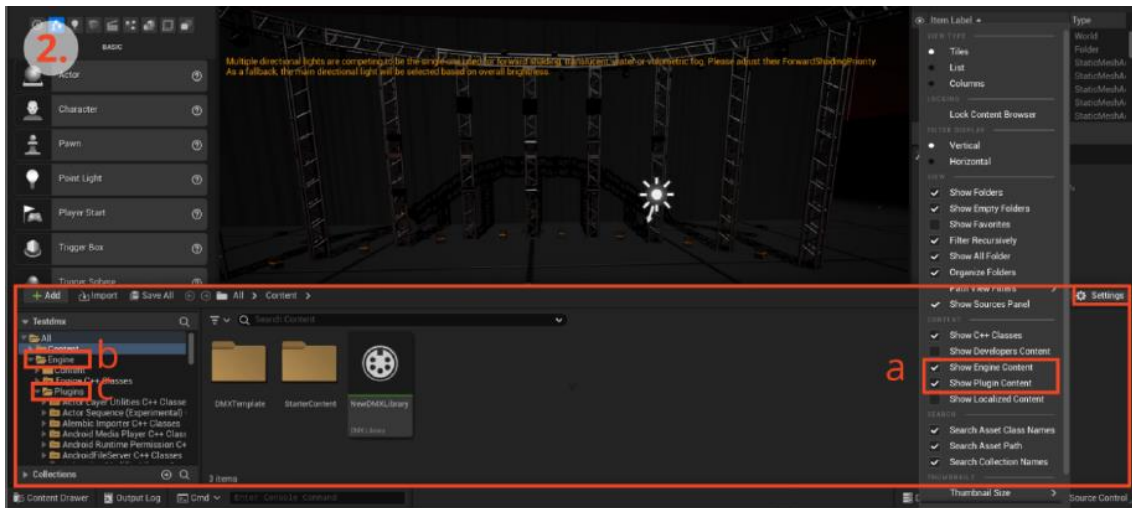
Vorsicht: das kostet Performance und kann ggf. die Prozesse verlangsamen. Wenn ihr merkt, dass das Bild zu sehr ruckelt/alles zu lange dauert, dann setzt den Haken (zwischenzeitlich) wieder.

Ohne Echtzeit Rendering wird euch der aktuelle Stand eurer Lichtkonfiguration (wie er im DMX Stage View angezeigt wird) nämlich auch angezeigt. Ihr müsst nur, nachdem ihr etwas im DMX Fenster verändert habt, zurück auf das UE Fenster klicken, damit es wieder im Vordergrund läuft und aktualisiert. Auch ein Klick auf den Channel Monitor führt nach unseren Erfahrungen zu einem „refresh“ der Anzeige.

Zusatz Tipp: Bevor ihr ausprobieren wollt, ob euer Computer Echtzeit Rendering packt, nicht vergessen davor einmal das Projekt zu speichern. Just in case :)

Lampe in UE zur Szene hinzufügen (Bild unten):

- Mit Strg. & Leertaste den Content Drawer öffnen und rechts bei "Settings" sicher stellen, dass bei "Show Engine Content" und „Show Plugin Content“ ein Häkchen gesetzt ist.
- Links, in der Ordner-Übersicht des Projekts, wird jetzt ein Ordner namens "Engine" angezeigt.
- Im Unterordner "Plugins" den Ordner "DMX Fixtures Content" suchen. Hier befinden sich alle relevanten Objekte (z.B. Lampen 3D-Modelle).
- Unter "DMX Fixtures Content" den Ordner "Light Fixtures" wählen, und den gewünschten Lampentypen in die Szenen Preview ziehen.



Möchte man nur einen statischen Scheinwerfer haben (der sich nicht bewegen kann), reicht ein „BP_StaticHead“. Möchte man auch Bewegung einstellen können, sollte man das Modell „BP_MovingHead“ nehmen (wie in diesem Beispiel). Also immer mitdenken, was die zu erstellende Lampe alles kann, und entsprechend das 3D Modell wählen.

3.

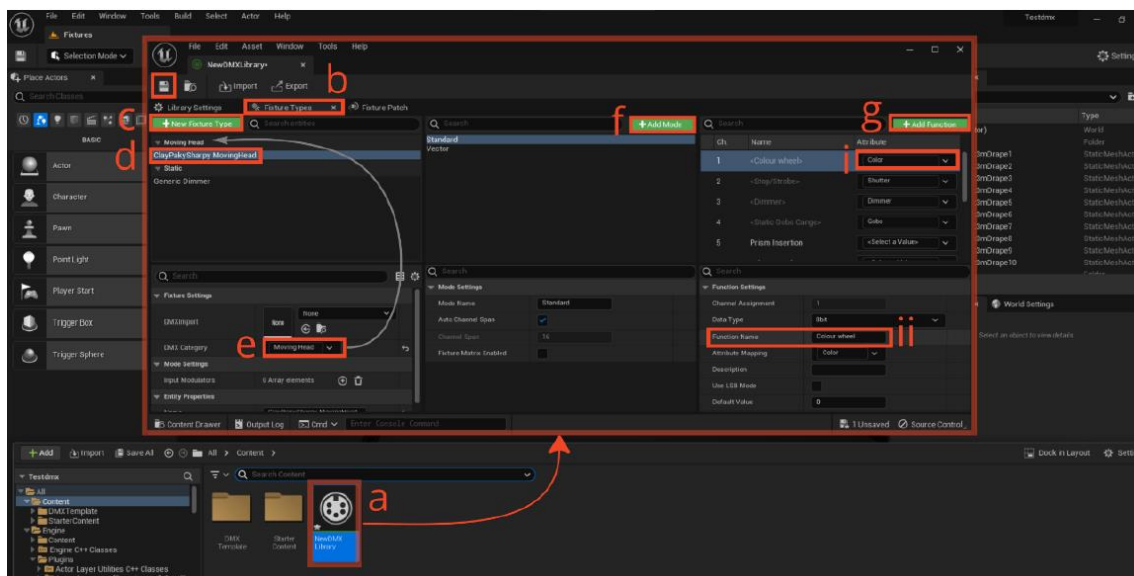
CHANNEL FUNCTION

SHARPY

NB: To prevent accidental breakage of the effects, which could collide with each other during transport, before switching the projector OFF check that all the projector Channels have been excluded (DMX level = 0 bit).

0

CHANNEL	CHANNEL MODE	
	STANDARD	VECTOR
1	COLOUR WHEEL	COLOUR WHEEL
2	STOP / STROBE	STOP / STROBE
3	DIMMER	DIMMER
4	STATIC GOBO CHANGE	STATIC GOBO CHANGE
5	PRISM INSERTION	PRISM INSERTION
6	PRISM ROTATION	PRISM ROTATION
7	EFFECTS MOVEMENT	EFFECTS MOVEMENT
8	FROST	FROST
9	FOCUS	FOCUS
10	PAN	PAN
11	PAN FINE	PAN FINE
12	TILT	TILT
13	TILT FINE	TILT FINE
14	FUNCTION	FUNCTION
15	RESET	RESET
16	LAMP CONTROL (with Option "Lamp Dmx" ON)	LAMP CONTROL (with Option "Lamp Dmx" ON)
17		PAN - TILT TIME
18		COLOUR TIME
19		BEAM TIME
20		GOBO TIME

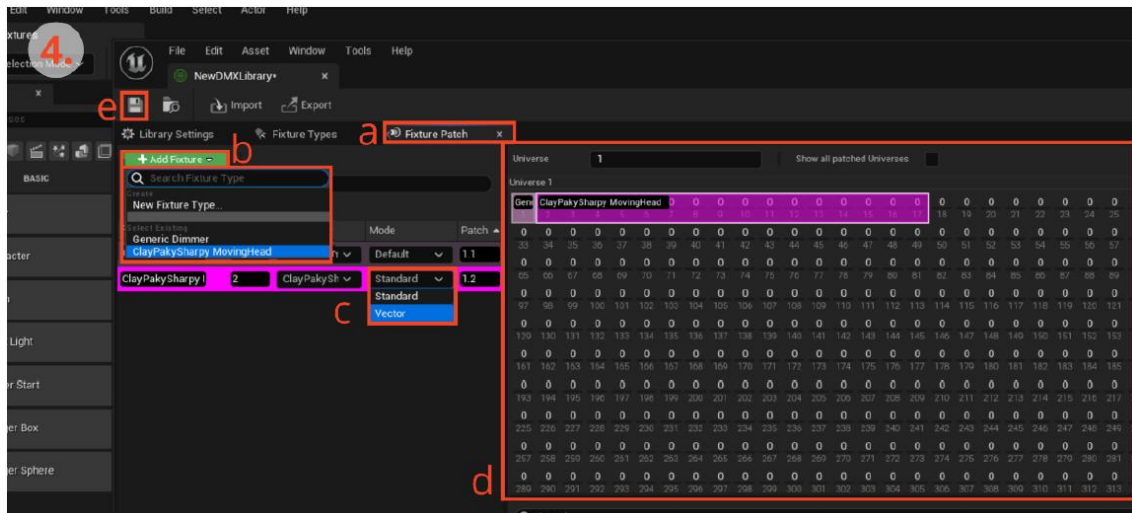


Lampe konfigurieren:

0. Damit Lampeninstanzen derselben Art in DMX Control und Unreal Engine gleich funktionieren, ist es wichtig zu wissen, welche Lampe man erstellen möchte, und deren Funktionen zu kennen. Hat man eine physische Lampe, kann man die Funktionsübersicht in der Bedienungsanleitung finden. Für manche Lampen findet man eine digitalisierte Version davon im Internet. Im Bild oben ist die Funktionsübersicht für den Clay

Paky Sharpy Movinghead zu sehen, den wir in diesem Beispiel verwenden werden. Dazu später mehr.

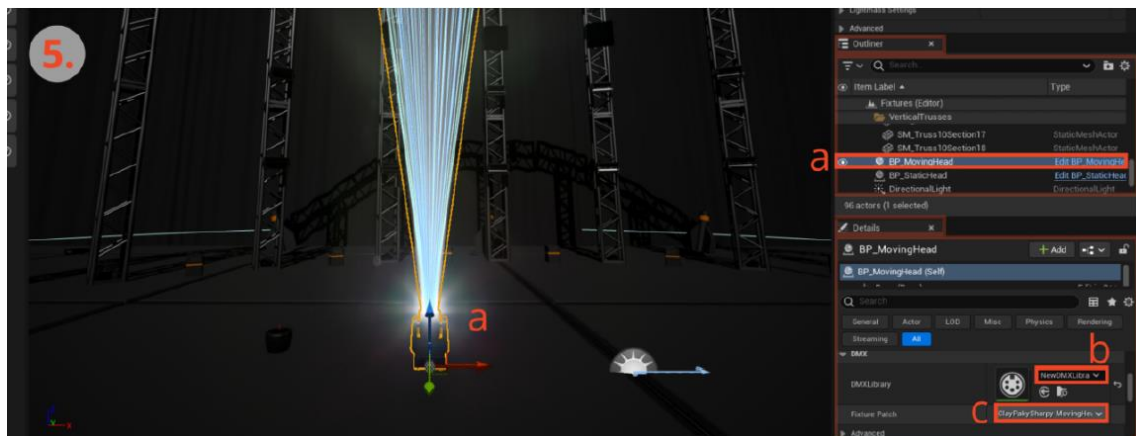
- a. In UE die am Anfang erstellte DMX Library mit Doppelklick öffnen.
- b. Auf den Reiter "Fixture Types" gehen. Hier kann man neue Arten von Lampen erstellen. Jeder "Fixture Type" entspricht dabei einer Lampen Art (z.B. Sharpy) mit all seinen Modi und Funktionen.
- c. Den Button "+ New Fixture Type" drücken.
- d. Den neu erstellten Typ benennen (z.B. „ClayPakySharpy MovingHead“).
- e. In den „Fixture Settings“ darunter, bei „DMX Category“ auswählen, welche Art von Lampe der eben erstellte Fixture Type sein soll. In unserem Beispiel ein „Moving-head“. Das dient dazu dass die erstellten Fixtures oben in der Liste automatisch in Kategorien einsortiert werden, was sehr hilfreich ist, wenn man vor hat mit vielen verschiedenen Lampen zu arbeiten.
- f. Mit "+Add Mode" so viele Modi hinzufügen wie man braucht und diese benennen. Wie in der tabellarischen Übersicht (Bild 3) zu sehen ist, hat der Sharpy Moving-head zwei Modi: einen mit 16 und einen mit 20 Channels.
Man kann jetzt beide einzeln erstellen, und jeweils konfigurieren, falls man beide Modi nutzen möchte, zwischen den Modi wechseln können möchte, oder zwei Lampen in der Szene haben möchte, die einen Sharpy darstellen sollen, aber der eine die Funktionen der 20 Channel Liste haben soll. Sieht man dafür keinen Nutzen, reicht es aus, nur einen Modus zu erstellen und konfigurieren.
- g. Mit "+Add Function" dem Modus so viele Funktionen zufügen, wie er haben soll (in diesem Beispiel einmal 16 für den Standard Modus, und einmal 20 für dem Vector Modus). Jede Funktion wird für einen Modus eines Fixture Types erstellt. Also immer aufpassen, dass der richtige Fixture Type ausgewählt ist und darunter der Modus den man gerade Konfigurieren möchte.
 - i. Mit dem drop-down Menü kann man auswählen, welche Funktion diese neu hinzugefügten Funktionen tatsächlich haben sollen. Die Liste (Bild 3.) der Reihe nach durchgehen und die Funktionen zuweisen.
Anmerkung: Aktuell wird von UE leider noch nicht jede Funktion unterstützt. Die entsprechenden Funktionen trotzdem erstellen (damit die Kanal-Reihenfolge mit der der Lampeninstanz in DMX Control übereinstimmt), aber nichts im drop-down Menü auswählen.
 - ii. Alle Funktionen nach der Liste benennen, um spätere Verwirrung zu vermeiden.



3.4. Fixture Types (Lampen Arten) patchen.

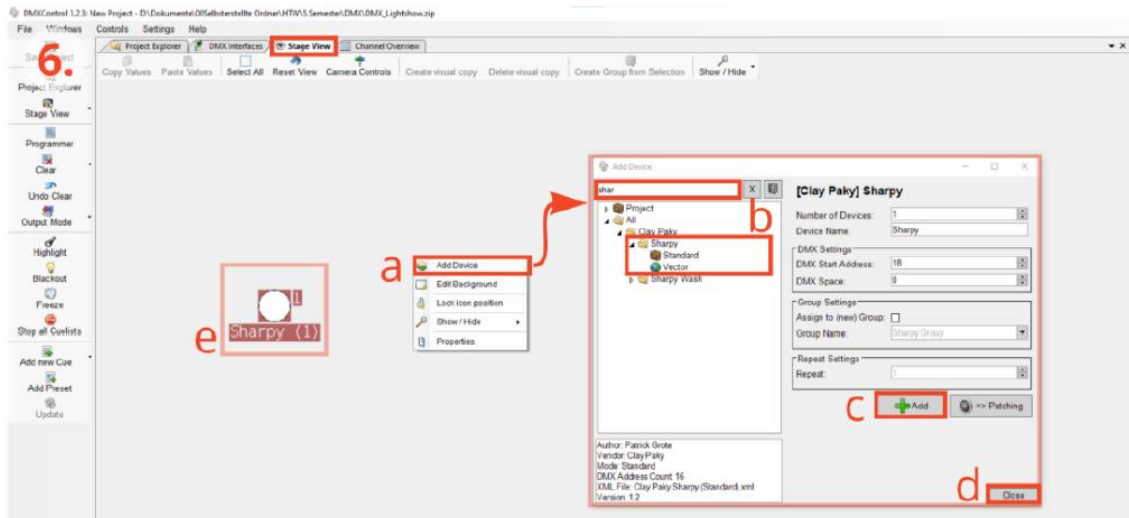
Dabei belegt jede Funktion eines Fixture Types einen Kanal.

- In der DMX Library (in UE) den Reiter "Fixture Patch" auswählen.
- Mit "+ Add Fixture" aus dem drop-down Menü den gewünschten Lampentyp auswählen.
- Ggf. mit dem drop-down Menü auswählen welcher Fixture Patch welchen Modus haben soll.
- Ggf. im rechten Universe overview händisch verschieben, welche Kanal-Spanne welche Lampe abdecken soll.
- Speichern nicht vergessen :)



Der Lampe den DMX Kanal zuweisen, auf den sie hören soll

- Lampen Modell anklicken, oder alternativ rechts im "outliner" „BP_Movinghead“ auswählen.
- Unten rechts in den "Details" runterscrollen und bei "DMX Library" die Library, in dem sich der Fixture Type den man benutzen möchte befindet, über das drop-down Menü auswählen.
- Direkt darunter bei "Fixture Patch" den gewünschten Fixture Type auswählen.
- Achtung: hierbei handelt es sich um den Modus der Lampe, der im Patch (Schritt 3.4.c.) ausgewählt wurde.



In DMX Control eine Instanz des Movingheads erstellen

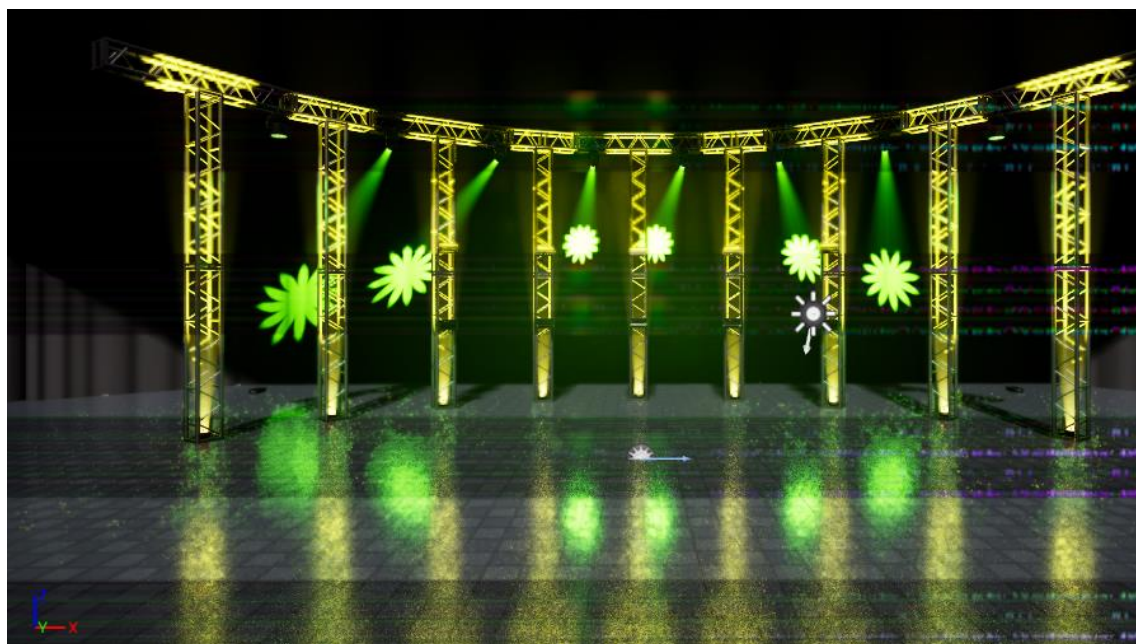
- Rechts-klick in das Stage View Fenster machen und "+ Add Device" klicken.
- Im neu geöffneten Fenster in der Suchleiste "sharp" suchen, und den gewünschten Modus (Standard oder Vector) anklicken.
Merke: In DMX Control und UE5 sollte der gleiche Modus gewählt sein.
- Rechts auf "+ Add" drücken.
- Fenster schließen.
- Mit Doppelklick auf die neu erstellte Instanz (in der Stage View), das Licht einschalten. Hat alles geklappt sollte die Lampe in UE5 jetzt auch angehen (falls Echtzeit Rendering eingeschaltet ist. Sonst muss man einmal kurz in den Play Modus in UE5 gehen).
- Sollte das nicht klappen, nochmal in DMX Control auf dem Channel Overview nachschauen, auf welchen Kanälen die Lampe liegt, und in UE5 in der DMX Library (unter dem Reiter Fixture Patch) sicher stellen, dass die Lampe dieselben Kanäle belegt.

Nach demselben Prinzip können jetzt beliebig viele Lampen erstellt und platziert werden, und nach der im AWE gelernten Weise eine Lichtshow in DMX gemacht werden, die dann in Unreal visualisiert wird.

3.3 Das Starterset

Um das Lernen und Ausprobieren mit Lichtkonfigurationen zugänglicher zu machen, könnt ihr ein Starterset von Ellen herunterladen, das ein **Unreal Engine 5.3 Projekt** und ein **DMXControl3 Projekt** enthält, die aufeinander abgestimmt sind.

So könnt ihr nach der Verknüpfung über ArtNet sofort loslegen, eure eigene Lichtshow zu entwerfen.



Im oberen Bild seht ihr die Stage View des DMXControl Projekts. Im unteren die Ansicht im Unreal Projekt mit der gleichen Lampenauswahl:

- **2 Washlights** (an der oberen Trass, außen. Können über die Gruppe G_Wash, oder einzeln eingestellt werden).
- **6 Movingheads** (an der oberen Trass, mittig. Können über die Gruppe G_Movingheads, oder einzeln eingestellt werden).
- **12 Strobes**. Können über die Gruppe G_Strobe angesteuert werden, oder in 2 Einheiten:
 - 10 Strobes (an den vertikalen Trassen, mittig. Verhalten sich alle identisch zueinander).
 - 2 Strobes (am Boden, hinter den Trassen. Verhalten sich beide identisch zueinander).
- **9 Static Lights** (am unteren Ende der vertikalen Trassen. Können über die Gruppe G_StaticHeads, oder einzeln eingestellt werden).

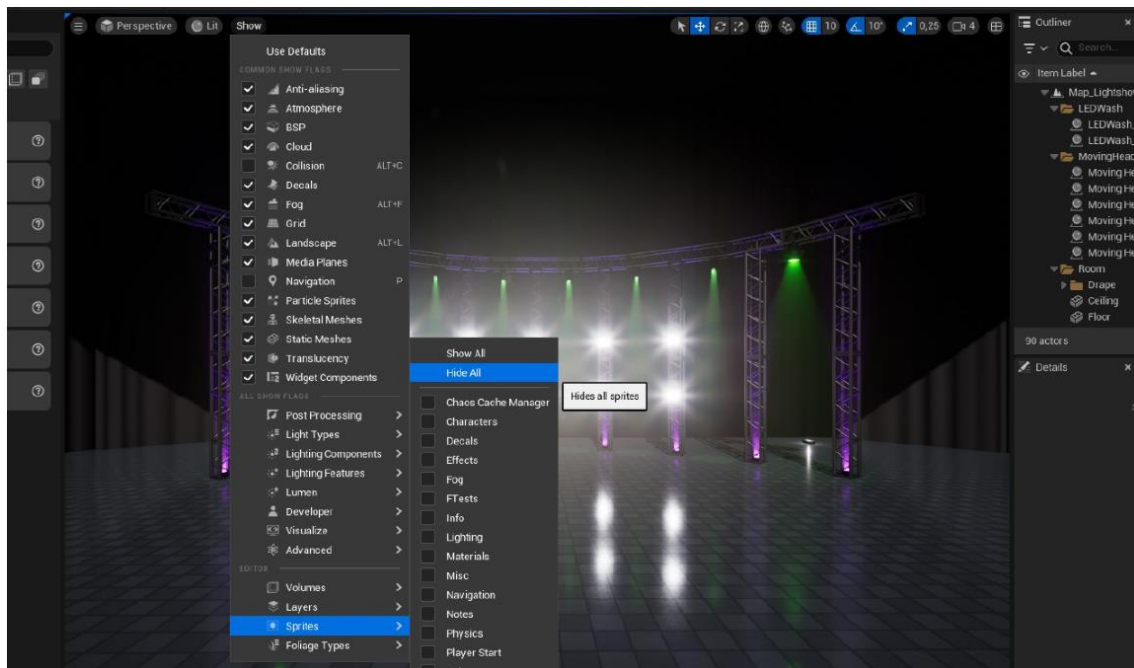
Alle Lampen sind gepatched (d.h. die Funktionen liegen bei DMX Control und bei UE schon auf den gleichen Kanälen). Ihr müsst also lediglich folgende Schritte befolgen, um loslegen zu können:

1. UE 5.3 herunterladen (siehe Schritt 1. dieser Dokumentation).
2. DMXControl3 herunterladen. (Easy View müsst ihr dabei nicht zusätzlich herunterladen, da ihr sowieso mit UE visualisiert).
3. Den Starterset Zip Ordner herunterladen und in den eigenen Dateien entpacken.
4. UE 5.3 öffnen und auf „Browse“ klicken. Den Starterset Ordner finden, darunter „DMX Lichtshow_UE“ wählen und „DMX_Lichtshow“ öffnen.
5. DMXControl3 öffnen, und oben über „File“ „Load Project“ drücken. Im neu geöffneten Fenster den Starterset Ordner finden und „DMX Lightshow_DMX C3“ öffnen.
6. Daraufhin wird sich vermutlich noch ein Fenster öffnen das fragt, ob ihr das zuletzt verwendete Fenster-Layout verwenden möchtet („Do you want to load the last used project window layout?“).
7. Auf „Ja“ drücken. Das Layout könnt ihr euch aber sowieso jederzeit nach Belieben anpassen.
8. Schritt 2.5 dieser Dokumentation folgen, um die ArtNet Schnittstelle aufzubauen.
9. Fertig :)

Testweise könnt ihr mal eine der Lampeninstanzen in DMXControl anklicken, und irgendwas an der Lampe verändern (z.B. Lichtfarbe ändern (Achtung: Die Strobes haben keine RGB Funktion)).

Dann auf das UE Fenster gehen und schauen ob sich die entsprechende Lampe verändert hat. Wenn ja dann hat alles geklappt, und ihr könnt anfangen Cuelists zu erstellen und eure Lichtshow zu entwerfen.

9. Bonustipp: Wenn ihr Screenshots oder Videos von eurer Lichtshow machen wollt, könnt ihr in UE oben links auf den „Show“ Button klicken. Unter „Sprites“ könnt ihr dann „Hide All“ auswählen, damit keine Icons die Sicht verschandeln. Viel Spaß! :)



4 Für Fortgeschrittene: Gerätemanagement

4.1 Allgemeines

Seit der Version 4.25 liefert Epic in Unreal Engine Unterstützung für die Darstellung diverser DMX-Geräte, siehe

<https://docs.unrealengine.com/en-US/WorkingWithMedia/DMX/index.html>

Aktuell gibt es keine fertigen kompletten Geräte-Bibliotheken für die Geräte der verschiedenen Hersteller, die mit der DMXControl DDF-Lib vergleichbar wären. Es gibt allerdings DMX-Templates für verschiedene Geräteklassen aus denen man seine konkreten Geräte ableiten kann.

Epic gibt an, dass Fixtures mit dem „General Device Type Format“ (GDTF) unterstützt werden. Dazu gibt es eine Import GDTF Library, siehe

<https://docs.unrealengine.com/5.0/en-US/dmx-gdtf-support-in-unreal-engine/>

Fertige Gerätedefinitionen über GDTF kann man z.B. hier finden:

- <https://gdtf-share.com/>
- <https://www.gdtf.info/docs/prologue/introduction/>

Epic hat selbst ein sehr umfangreiches Tutorial auf Youtube „Control Lighting With DMX In 4.26 | Inside Unreal“ bereitgestellt:

<https://www.youtube.com/watch?v=-olePxpasXk>

Es gibt auch erste Versuche, eine Unreal-DMX-Fixture Library aufzubauen, siehe hier <https://github.com/LexyVFX/Unreal-DMX-Fixture-Library>

“This repo is intended to be populated with PBR assets & Blueprint implementations of DMX Lights for use with Unreal Engine.”

Ob sich genügend Leute finden, die das Projekt unterstützen kann man im Moment noch nicht beurteilen. Aktuell wird das Repository nicht gepflegt.

4.2 Neue Geräte erstellen

Wenn man sich Gerätedefinitionen mit den Bordmitteln von UE erstellen möchte, die über die Basisfunktionen hinausgehen, wird es leider etwas komplizierter.

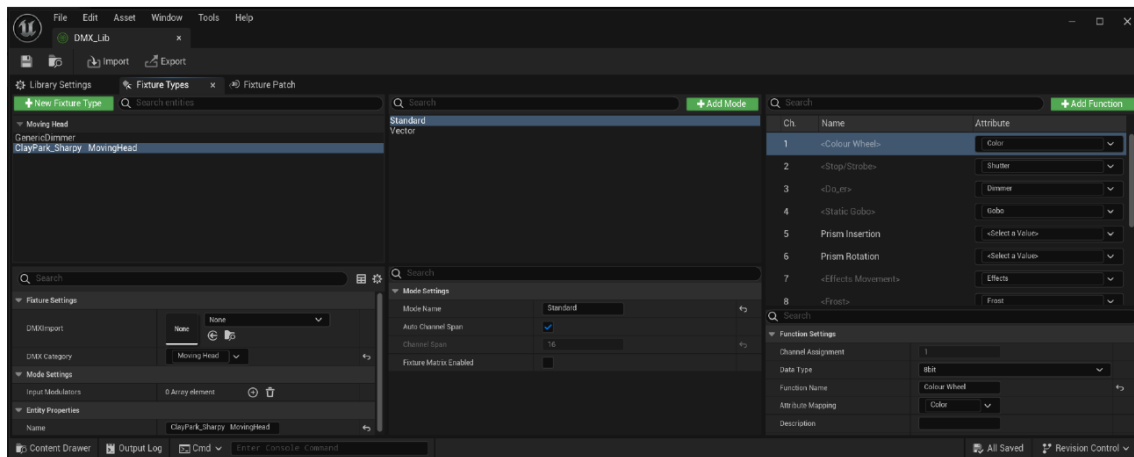
Achtung: Dieses Kapitel ist noch im draft-Zustand und soll mit Eurer Hilfe in Zukunft verbessert werden.

Fixture Types

Bevor einzelne Kanäle belegt werden können, müssen erst Geräte Typen angelegt werden. Das geht in der DMX Library im Tab „Fixture Types“. Mit dem Button „+ New Fixture Type“ kann ein neuer Gerätetyp angelegt werden.

Im mittleren Bereich des Fensters können mit dem Button „+ Add Mode“ verschiedene Modi für den jeweiligen Gerätetyp definiert werden. Diese verhalten sich wie verschiedene Versionen des Gerätetypen. Ob man verschiedene Geräte als einzelne Fixture Types oder als Modes organisiert, ist jedem freigestellt, mindestens ein Modus muss

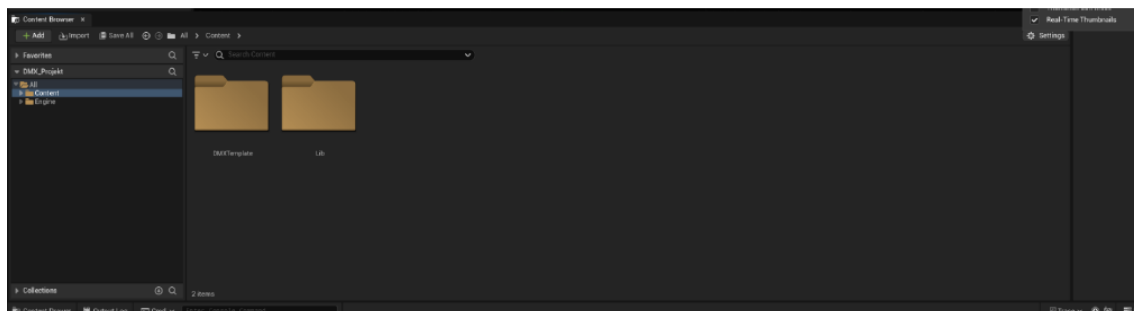
jedoch immer erstellt werden. 1 In jedem Modus können dem Gerätetyp unter „+ Add Function“ neue Funktionen verliehen werden. Diese müssen sich nicht zwangsläufig mit der Kanalbelegung in DMX Control decken, es ist aus Übersichtsgründen jedoch empfehlenswert. Die Funktionen unterscheiden sich abgesehen von ihrem Namen nur in zwei Punkten: Der Anzahl an Kanälen die sie belegen (8 oder 16bit) und dem Output des Kanalwerts (0-1 oder 0-255). Die Auswahl an Funktionen ist namentlich auf klassische Lichttechnik beschränkt, was aber nicht bedeutet, dass sie nur dafür verwendet werden können. Um die X- und Y-Position eines Objekts zu steuern bieten sich beispielsweise „Pan“ und „Tilt“ an.

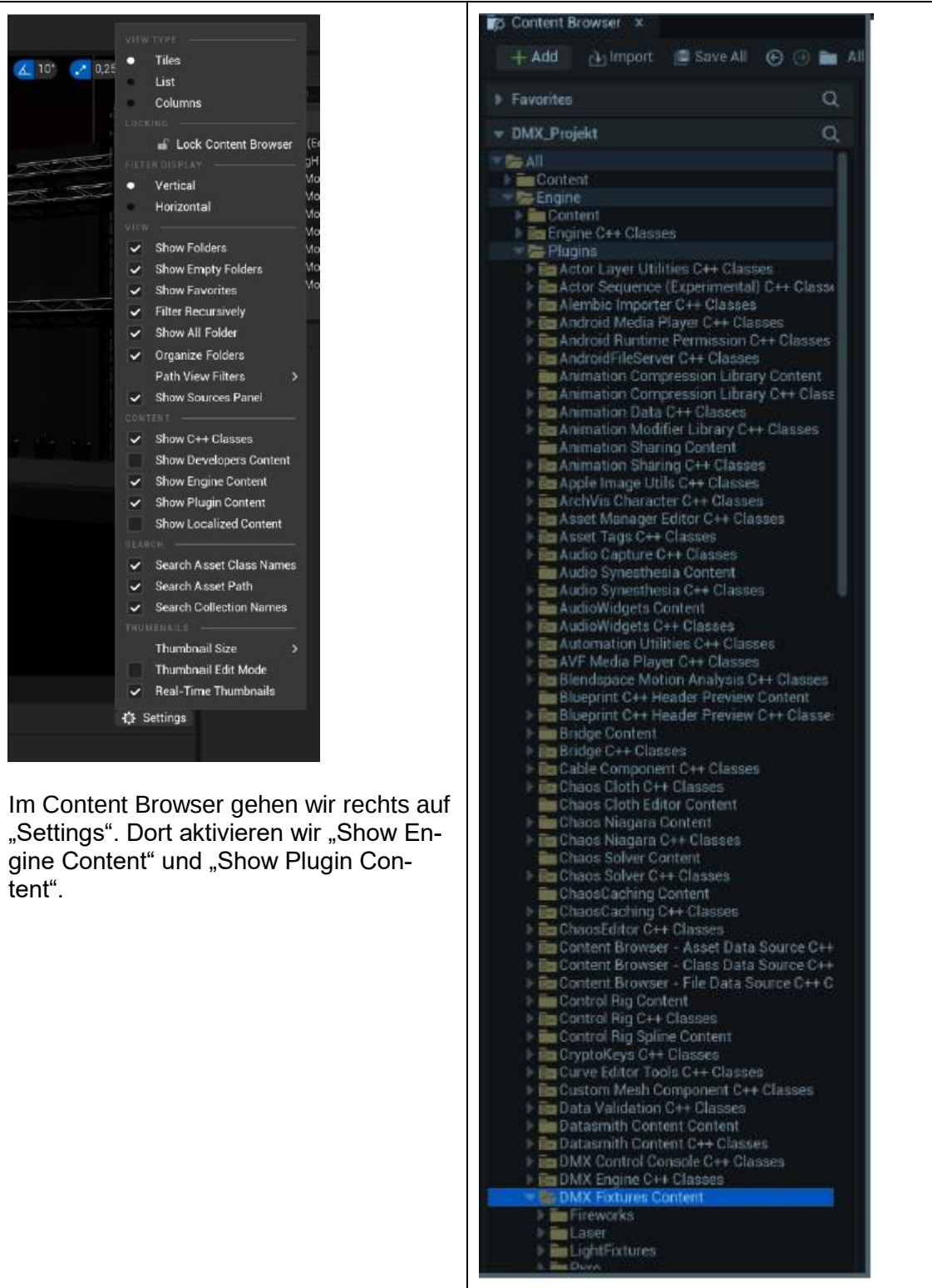


Unter Fixture Types, können wir alle Geräte Typen definieren die wir für unsere Show benötigen. Das genaue Modell, der Geräte Typ wie Zum Beispiel von Clay Paky -> Sharpy müssen wir eigenständig in Unreal definieren. Es sind keine vordefinierten Geräte in Unreal vorhanden.

Je nachdem, welche Geräte wir benutzen wollen, müssen wir auch die unterschiedlichen Modi und die dazugehörigen Kanäle des Gerätes erstellen. Zuerst fügen wir links unter „New Fixture Type“ unser neues leeres Gerät hinzu. In der Mitte unter „Add Mode“ fügen wir die Modi ein, die wir benutzen wollen. Rechts unter „Add Function“ fügen wir alle Kanäle und Funktionen des Gerätes ein. Dort wählen wir dann die jeweilige Eigenschaft der Funktion aus. Wichtig ist, die Reihenfolge der Funktionen entspricht auch der Adressreihenfolge im realen Gerät. Selbst wenn wir nicht jede Eigenschaft in den Funktionen belegen, wir müssen die Funktion drin haben damit auch im Interface die richtigen Adress-Abstände vorhanden sind.

Wenn es darum geht, die Geräte physisch in die Szene einzubinden, müssen wir zuerst einmal im Content Browser uns die Templates sichtbar schalten.





Im Content Browser gehen wir rechts auf „Settings“. Dort aktivieren wir „Show Engine Content“ und „Show Plugin Content“.

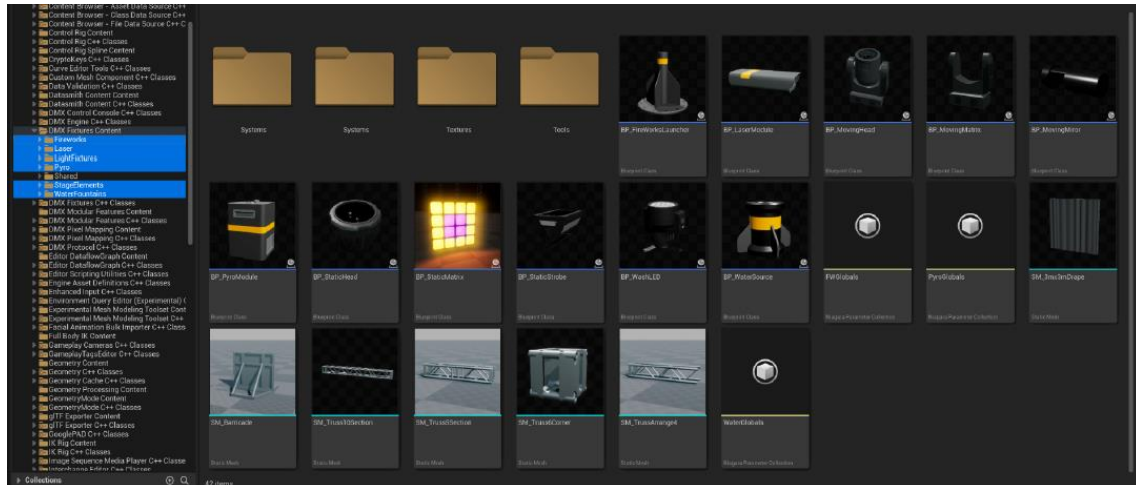
Jetzt können wir im Browser auf Engine -> Plugins und den Ordner „DMX Fixtures Content“ navigieren.

In diesem Ordner finden wir diverse unterordner für Pyrotechnik, Feuerwerk, Laser, Lichttechnik und sogar Bühnen Elemente für die Visualisierung.

Wir können diese Objekte einfach per Drag & Drop in die Szene von hier aus ziehen. Die DMX Library ist dafür da, von unterschiedlichsten Geräten die Eigenschaften zu sammeln und diese auf genau diese Blueprints im Content Browser zu übersetzen. Sie sind also eine generalisierte Lösung für Unreal.

Wir müssen nur wissen, welche Art von Gerät wir benötigen. Sei es ein Moving Head, StaticMatrix oder ein Static Head. Die Fixture Types die wir in der DMX Library erstellen, definieren ob es sich um einen Generic Dimmer oder ein Clay Paky Sharpy handelt oder etwas anderes.

In der Szene selbst definieren wir dann das Gerät mit dem Patch, wozu wir gleich kommen.



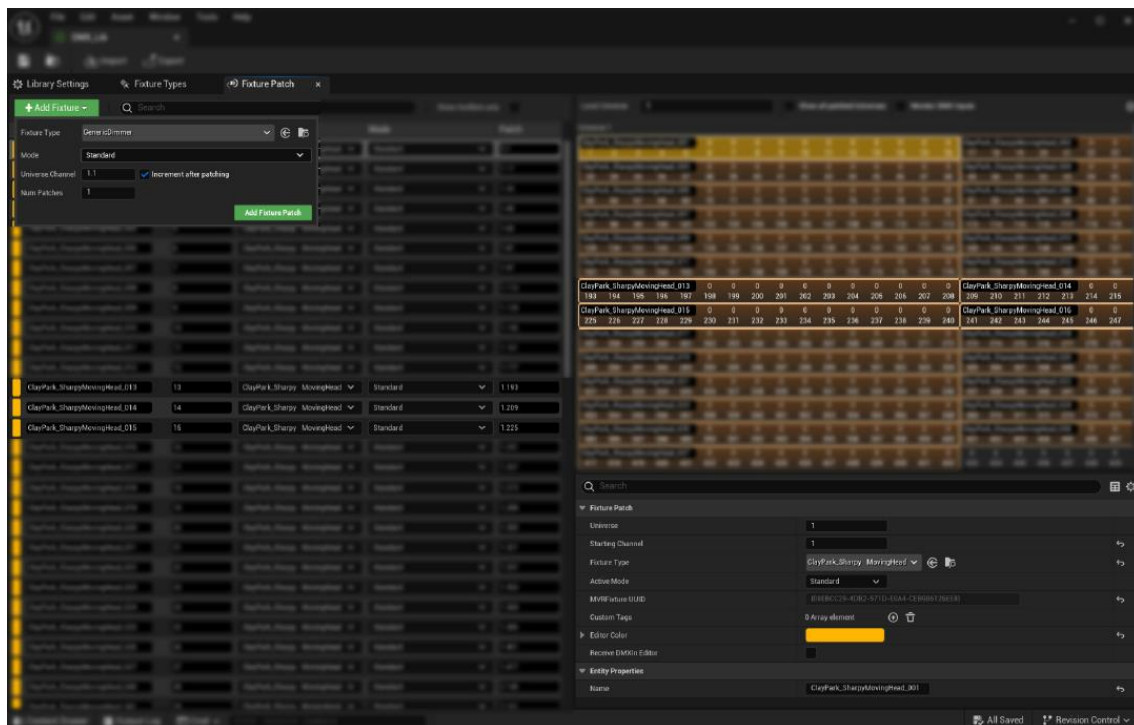
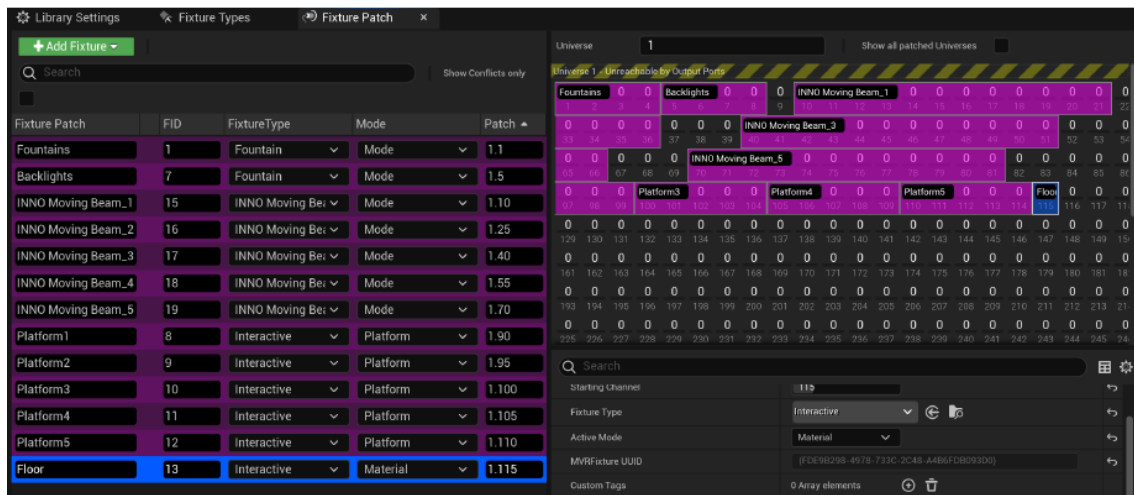
Nachdem wir den Fixtures Types die wir brauchen also erstellt haben, werden wir nun als nächsten Schritt alle einzelnen Geräte, die wir in der Show benutzen einbauen.

4.3 Fixture Patch

Nachdem alle benötigten Funktionen hinzugefügt wurden, können unter dem Reiter „Fixture Patch“ virtuelle Geräte der erstellten Gerätetypen (und Modi) den Kanälen des DMX Universums zugewiesen werden. Die belegten Kanäle sollten sich mit den in DMX Control definierten decken. Über den Namen der im Fixture Patch angelegten Geräte werden diese Später von den Objekten in der Szene gefunden, weshalb sich eine übersichtliche Namensgebung lohnt.

Unter dem Fixture Patch Tab, klicken wir auf „Add Fixture“ und stellen dort den Type, Mode, den Universe Channel von dem wir aus starten wollen und die Anzahl an Geräte dieses Types ein.

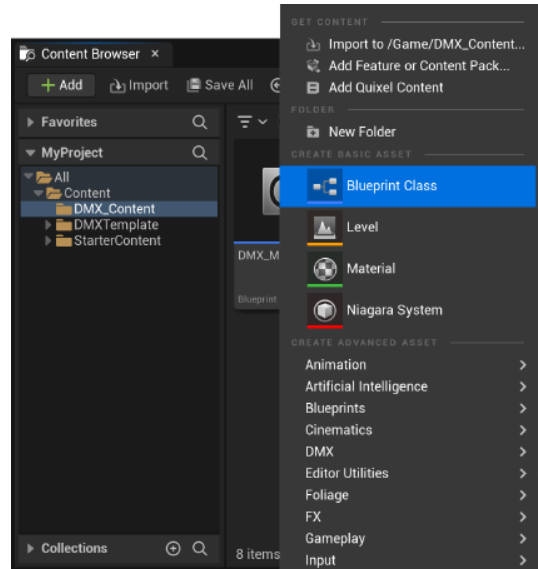
Dies wiederholen wir mit allen Geräten. Wir können rechts ebenfalls sehen, welche Adressen die Geräte haben.



Damit auch alles korrekt ist, vergleichen wir das Interface in Unreal mit dem von DMXControl3 und schauen, ob alle Universes und Adressen richtig sind. Ansonsten passen wir diese bei dem jeweiligen Gerät in Unreal an.

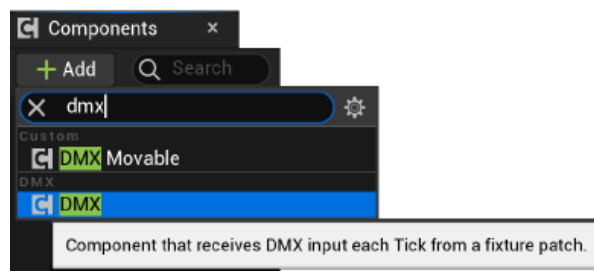
4.3.1 Blueprint Classes

Classes sind Arten von Objekten, die in einer Szene platziert werden können. Blueprint Classes benutzen die Visual Scripting Architecture von Unreal Engine. Sie können Code ausführen, der nicht in C++ geschrieben wird, sondern aus verknüpften Logic Nodes besteht. Eine neue Blueprint Class kann im Content Browser mit Rechtsklick > Blueprint Class erstellt werden. In der Blueprint werden neue Nodes mit Rechtsklick oder durch ziehen eines Pfads aus einer bestehenden Node hinzugefügt.



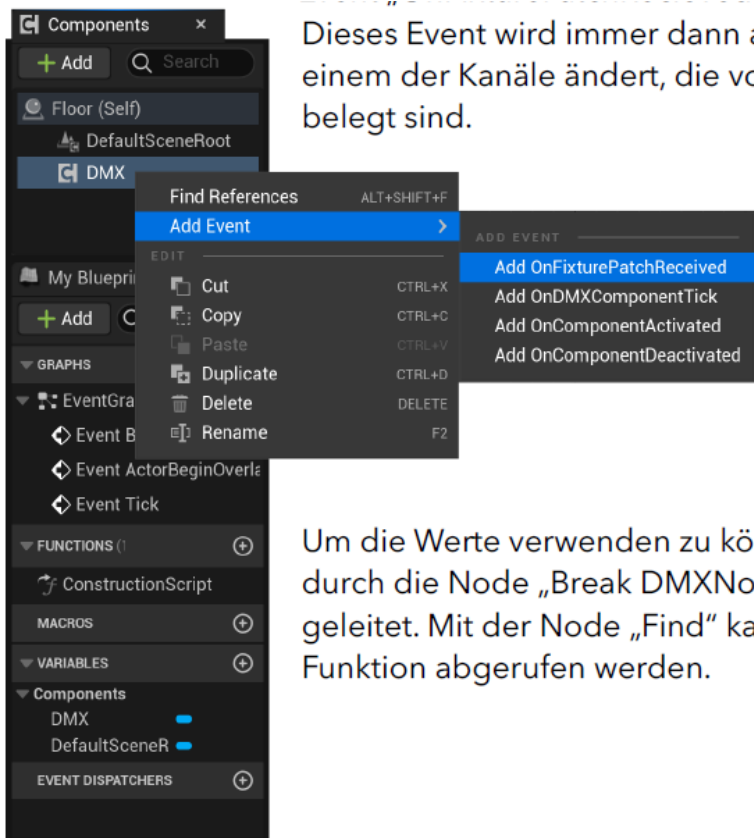
4.3.2 DMX Component

Im Blueprint Editor Fenster können der Class zusätzlich Components, wie z.B. ein Mesh (ein 3D Model) hinzugefügt werden. Um mit der DMX Library zu kommunizieren, benötigt die Class eine DMX Component.



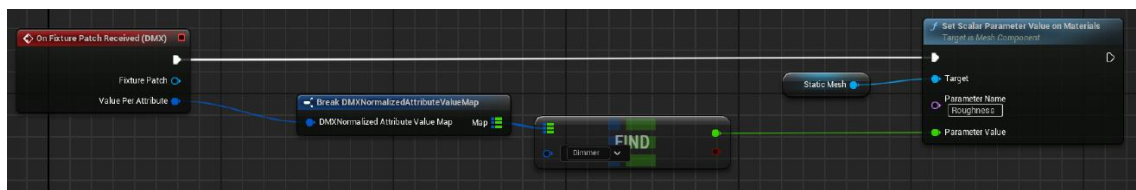
Mit Rechtsklick auf die DMX Component > Add Event kann das Event „OnFixturePatchReceived“ in die Blueprint eingefügt werden. Dieses Event wird immer dann aufgerufen, wenn sich ein Wert auf einem der Kanäle ändert, die von dem entsprechenden Gerät belegt sind.

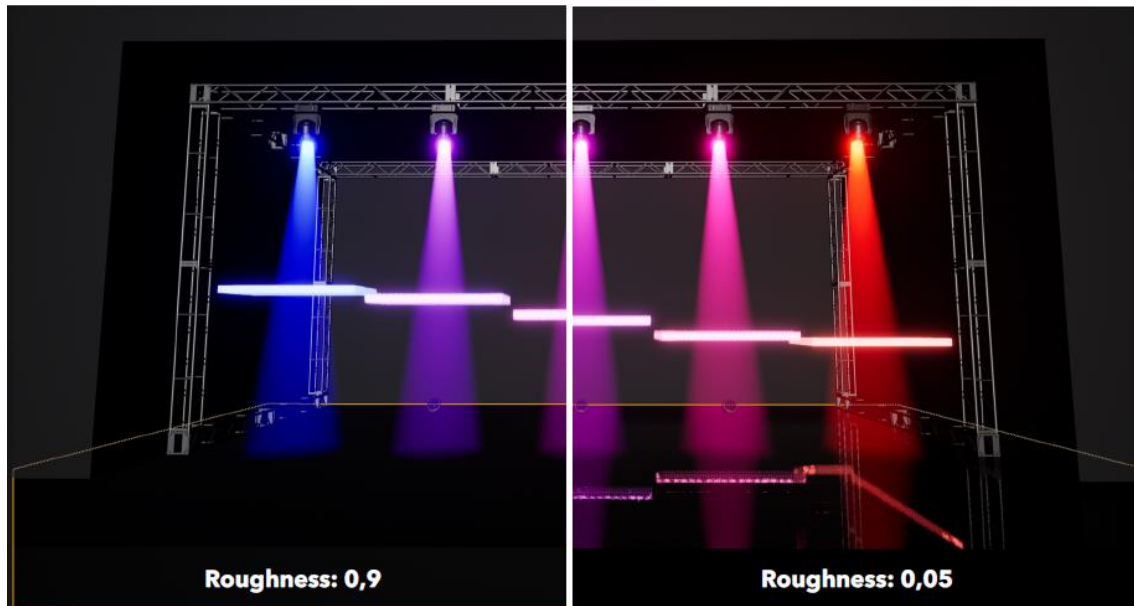
Um die Werte verwenden zu können, wird „Value per Attribute“ durch die Node „Break DMXNormalizedAttributeValueMap“ geleitet. Mit der Node „Find“ kann dann der Wert der gewünschten Funktion abgerufen werden.



4.3.3 Beispiel

Der so erhaltene Wert kann als Variable frei verwendet werden. Im folgenden Beispiel wird der Wert aus der Funktion „Dimmer“ dazu genutzt, die Roughness eines Materials einzustellen. So kann während der Show kontrolliert werden, wie stark eine Fläche reflektiert um beispielsweise Nässe zu simulieren.

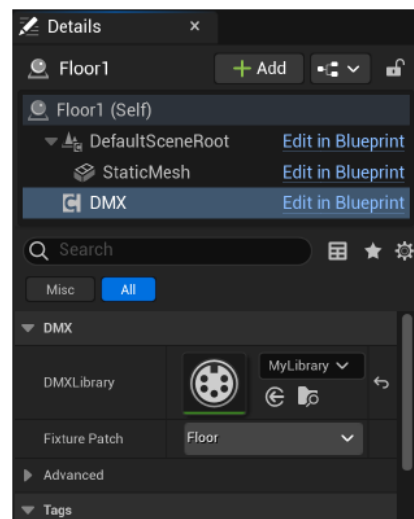




Tip: Die Blueprint sollte mit dem „Compile“ Button getestet und regelmäßig gespeichert werden. Strg+S speichert NICHT die Änderungen in der Blueprint.

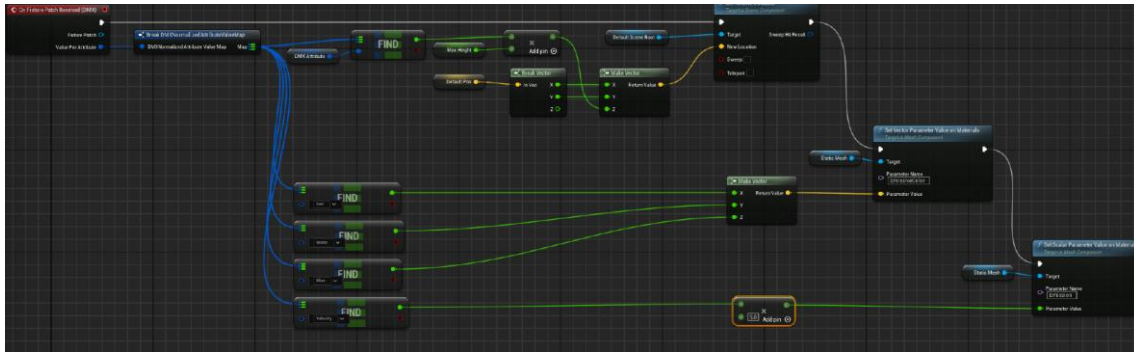
4.3.4 Actor

Die Blueprint Class kann aus dem Content Browser in die Szene gezogen werden um ein Object (Actor) dieser Klasse zu erstellen. Wählt man im Outliner das entsprechende Objekt, findet man im Details Tab die Kategorie DMX. Hier muss zum einen die genutzte DMX Library und zum anderen der Name des gewünschten Geräts im Fixture Patch angegeben werden. Tip: Ja nach Attribute wird die DMX Component einen wert zwischen 0 und 255 oder zwischen 0 und 1 ausgeben. Mit der Node „Print String“ können werte im Editor Fenster angezeigt werden.

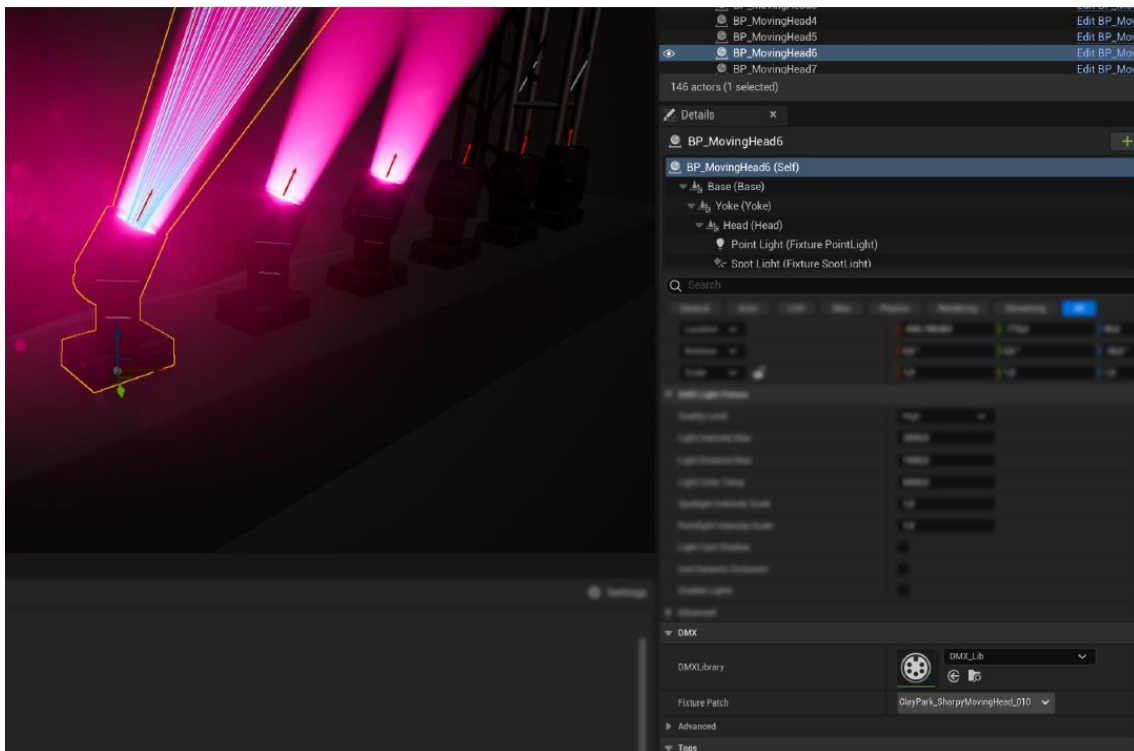


Verwendung

Im folgenden Beispiel wird ein Fixture Type mit 5 Kanälen genutzt um die Position, die Farbe und die Emission einer Plattform über DMX zu steuern. Die Dimmer Function verändert die Y-Position der Plattform. In DMX Control 3 ist die Position als generic Dimmer angelegt. Über die RGB Kanäle eines separaten Geräts wird auch in Unreal Engine die Farbe des Objekts geregelt. Auf dessen Dimmer Wert ist in Unreal Engine die Velocity Funktion gemapt, welche durch die Blueprint wiederum die Emission beeinflusst.



Da diese Blueprint Class für 5 Actor in der Szene verwendet wird, wird ein Fanning in DMXControl 3 automatisch auf die Objekte in Unreal angewendet, z.B. auf die Farbe oder die Position der Plattformen. Auch Effekte funktionieren wie gewohnt.



Nun können wir beispielsweise einen Moving Head, welcher mit dem DMX Template von Unreal kommt, in Details die DMX Library einstellen und den Fixture Patch.

Hier ist es hilfreich, wenn alle Geräte übersichtlich benannt wurden.

Wir fügen jede Lampe nun in die Szene ein, stellen den richtigen Patch ein und damit ist das vollständige Setup fertig!

Wir können jetzt noch die Szene nach Belieben gestalten. Von hier aus, werden die Geräte vollständig von DMX Control übernommen und Visualisieren automatisch die Cues und Eingaben.



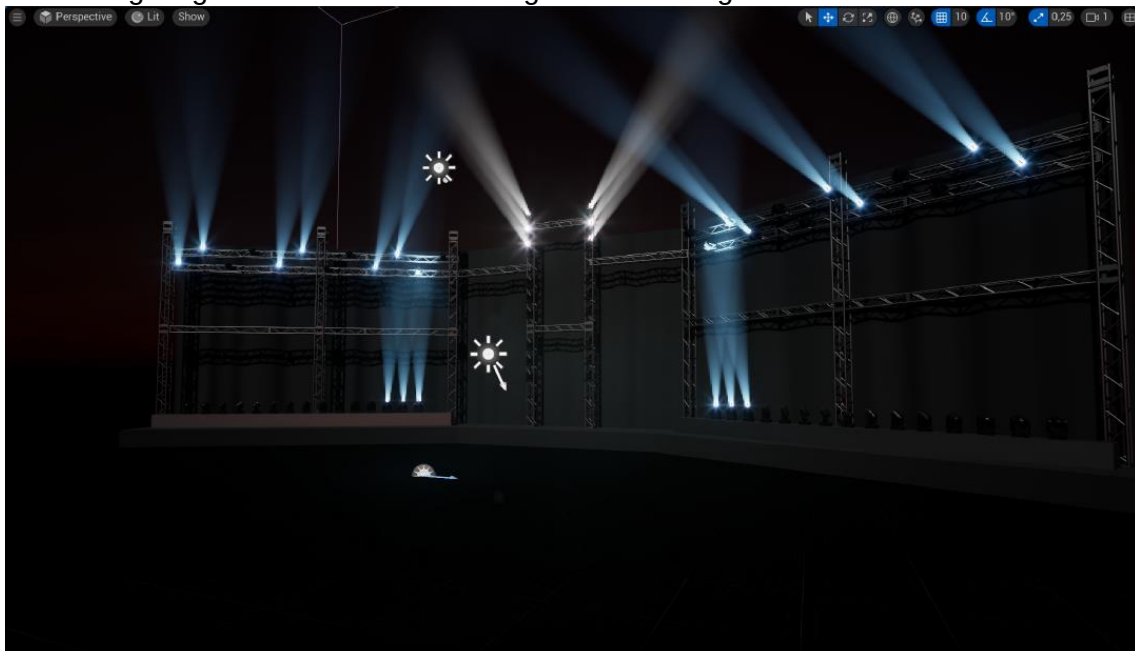
5 Schlusswort

5.1 Zusammenfassung

Einmal das Setup fertig und alles funktioniert, muss man in Unreal nichts mehr machen. Alles wird über DMX nun gesteuert. Wenn andere Lampen eingebaut werden sollen oder Pyrotechnik oder sonstige Effekte, müssen diese selbstverständlich dann in Unreal auch eingebaut und angepasst werden.

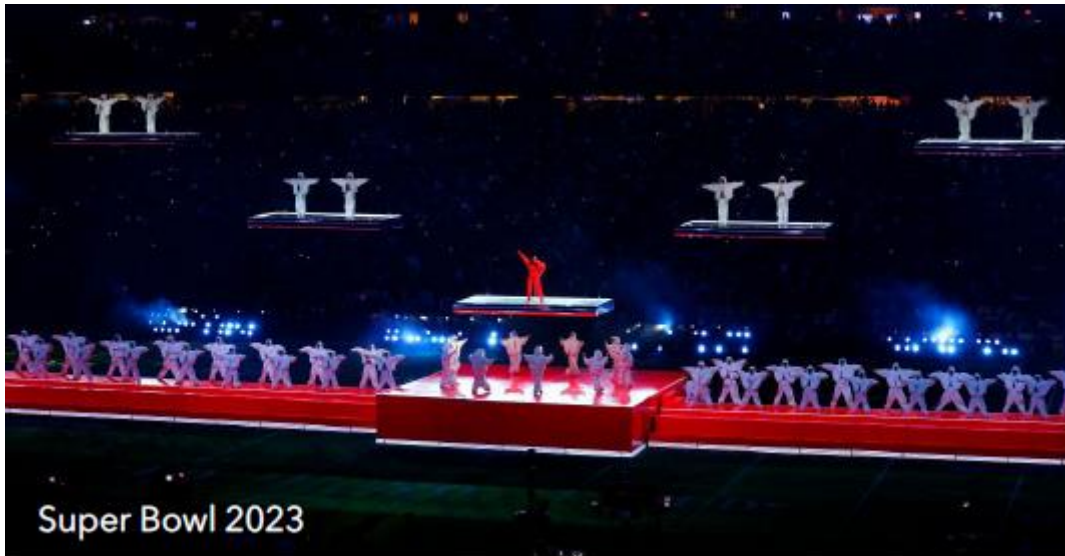
Unreal kann daher vor allem dann sinnvoll sein, wenn man beispielsweise größere Shows plant, neue Geräte ausprobieren und eine grobe Vorstellung von dem Resultat vor dem Kauf bekommen möchte oder einfach nur eine ausgereifte und gute Visualisierung eines kleinen oder auch sehr komplexen DMX Projektes sehen will.

Wenn man sich einmal in Unreal reingefuchst hat, ist es nahezu ein Kinderspiel und größere Projekte lassen sich damit bereits komplett in unter einer Stunde einstellen. Aber wie gesagt – da ist die Einarbeitungszeit nicht eingerechnet....



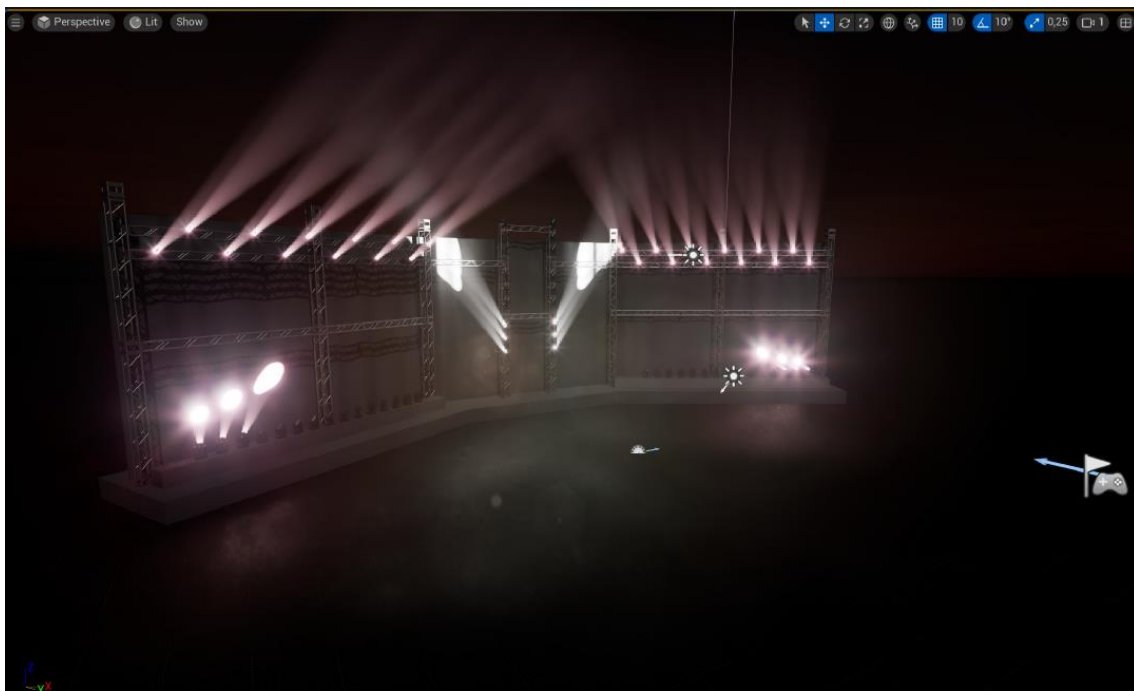
Diese Art von Virtual Produktion bietet sich sehr zum Prototyping für real life Shows an. Besonders bei sehr aufwendigen Produktionen ist es oft günstiger und schneller die Installation in Unreal Engine umzusetzen.

Auch für Mixed Media Live Shows ist Unreal Engine eine sehr gute Wahl. So können z.B. die Lichtgeräte einer Installation 1:1 digital nachgebildet werden um eine virtuelle Erweiterung des Sets zu erleuchten. Dank Real Time Raytracing muss nicht Alles im Vorhinein gerendert worden sein und die virtuelle Welt kann auf spontane Abweichungen von einer Setlist reagieren oder sogar zusammen mit dem Licht vor Ort live beeinflusst werden.



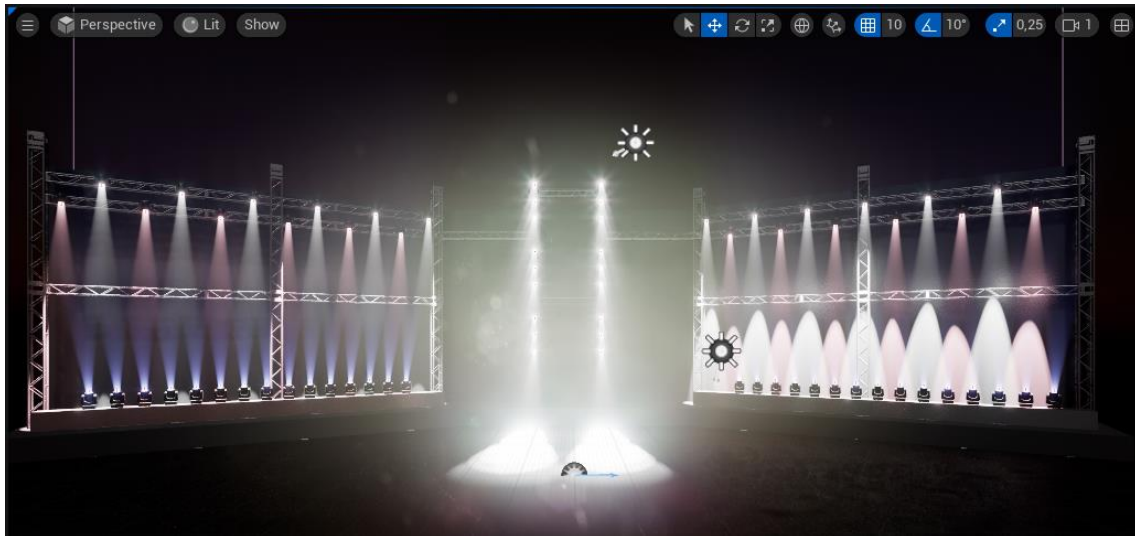
5.2 Die finale Show

Roman hat Im Lichtkurs eine finale Show programmiert, die wir euch hier nicht vorenthalten wollen. Wer Lust hat kann sicherlich die Programmierung der Cuelist in DMXControl anpassen und zur Verfügung stellen, aber das Ergebnis ist auch in diesem Zustand sehr beeindruckend.



In dieser Show wird „aus dem Vollen“ geschöpft – es werden 84 Clay Paky Moving Heads eingesetzt. Das erfordert dann allerdings auch eine entsprechende Grafikleistung um eine ruckelfreie Darstellung zu erhalten.

(Leider sind mir die Timing Daten etwas durcheinander geraten, ich würde mich hier über weitere Unterstützung aus der Community freuen.)



So wie das Starterset, stehen auch die Daten aus dieser Show zum Download zur Verfügung.

6 Links

Software:

<https://www.unrealengine.com/de/>

<https://www.dmxcontrol.de/>

Kopplung von DMXControl3 mit UE (Philipp hat ein eigenes Tutorial veröffentlicht):

<https://www.youtube.com/watch?v=6zLNjrMEgsM&t>

Hier ist der YouTube link zu Philipps Abschlussprojekt:

<https://www.youtube.com/watch?v=1udTsp8l9WI>

The setup contains 190 fixtures all in all (with matrices) all controlled by DMXControl 3 in real time.

User Tutorial UE DMX Plugin

Getting Started: <https://www.youtube.com/watch?v=T-p-evovogE>

Natürlich hat auch Unreal Engine einige Demo-Projekte veröffentlicht, siehe hier:

<https://www.youtube.com/watch?v=UXodNAcyFj8>

Sonstiges:

Virtual Festival Space:

<https://demodern.de/projekte/kollektiv-turmstrasse-virtual-festival-space>

Music: Mr. Comedian by Serendipity

<https://www.youtube.com/channel/UCcwX...>