

Inhaltsverzeichnis:

0.0.0 Einleitung (1)

1.0.0 Erste Schritte (1-3)

1.1.0 Platziere GlobalMusicSystem

1.1.1 GlobalMusicSystem

1.2.0 Play Music

1.2.1 Play Music via Nodes

1.2.2 Play Music via Trigger

2.0.0 Music Control (4-6)

2.1.0 Play & Stop Music

2.2.0 Music Settings

2.3.0 Get Music Data

3.0.0 Music Listener (6-7)

3.1.0 Interface: I\_MusicSystem\_Listener

3.2.0 Interface: Events

3.2.1 OnCurrentMusicTimeSignature

3.2.2 OnMusicFinished

3.2.3 OnMusicStart

4.0.0 Trigger (7-9)

4.1.0 Trigger Overview

4.2.0 Details Tab

4.2.1 Trigger: Music

4.2.2 Trigger: Collision

4.3.0 Override Trigger

5.0.0 DataTable DT\_MusicSequences (9-12)

5.1.0 DataTable Overview

5.2.0 Sequence

5.2.1 Music Clip

5.3.0 Data Asset PDA\_Music

## ===== 0.0.0 Einleitung =====

Dieses Music System befindet sich genau wie diese Anleitung im ALPHA Status! Feedback und Verbesserungsvorschläge sind daher herzlich willkommen. Wende dich gerne direkt an mich unter: [alex@audiolarks.com](mailto:alex@audiolarks.com) oder via Discord: Audioaxel#2231

Um eine größere Diversität, Musikalität und Anpassungsfähigkeit zu erreichen, sind die im Bundle enthaltenen Tracks in diverse Loops und Transitions, in sog. Clips unterteilt.

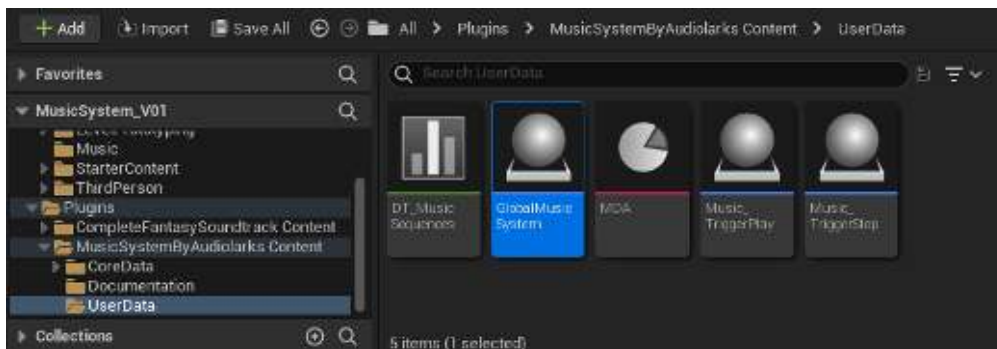
Diese Clips werden in einem DataTable mit verschiedenen Parametern versehen und zu Sequences o.a. Tracks wieder zusammengefügt. Jede Sequence bekommt ein oder mehrere DataAssets zugewiesen, über welche man anschließend via Nodes die Musik kontrolliert.

Das Music System dient aller erstes dem schnellen und einfachen Handling der im Bundle enthaltenen Musik. Es ist jedoch möglich, dieses um eigene Musik zu erweitern oder voreingestellte Eigenschaften zu ändern. Siehe dazu vor allem Punkt 5 – DataTable DT\_MusicSequences.

## ===== 1.0.0 Erste Schritte =====

### --- 1.1.0 Platziere GlobalMusicSystem ---

Navigiere zu Plugins/MusicSystemByAudiolarks/UserData und platziere GlobalMusicSystem im Level. Achte darauf, dass dieser Actor stets im Level präsent bleibt.



### --- 1.1.1 GlobalMusicSystem ---

Im Details Tab hast du die Möglichkeit, dass Printen von Debug- und System Informationen zu aktivieren. Des Weiteren kannst du dort einen anderen DataTable auswählen (Für die Verwendung von verschiedenen Bundles).

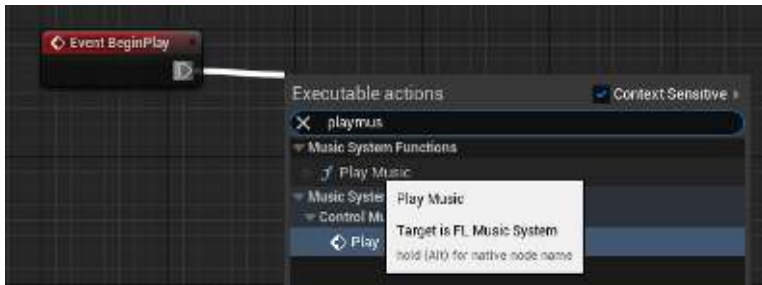
### --- 1.2.0 Play Music ---

Um Musik abzuspielen und zu kontrollieren hast du die Möglichkeit in den Blueprints auf diverse Nodes zuzugreifen. Eine Übersicht findest du unter Plugins/MusicSystemByAudiolarks/CoreData/FL\_MusicSystem

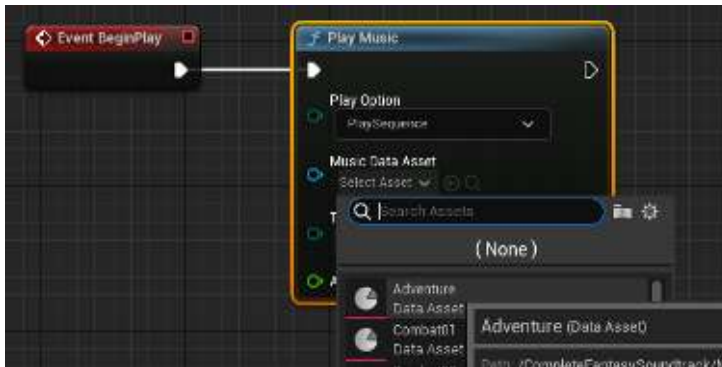
Du kannst aber auch direkt mit den mitgelieferten Triggern unter: Plugins/MusicSystemByAudiolarks/UserData/Music\_TriggerPlay direkt im Level starten.

### --- 1.2.1 Play Music via Nodes ---

Öffne das Levelblueprint (oder ein BP deiner Wahl) und suche nach PlayMusic.

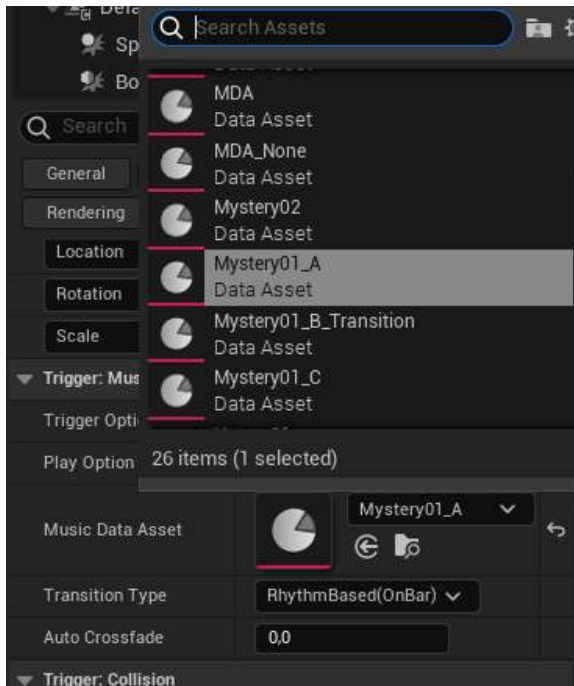


Wähle nun unter „MusicDataAsset“ einen Track(DataAsset) aus. Speichere und kompiliere das Blueprint und starte das Level. Du solltest nun deinen ausgewählten Track hören.



### --- 1.2.2 Play Music via Trigger ---

Ziehe Music\_TriggerPlay in dein Level und wähle im Details Tab, Abschnitt „Trigger: Music“ unter „MusicDataAsset“ einen Track(DataAsset) aus.



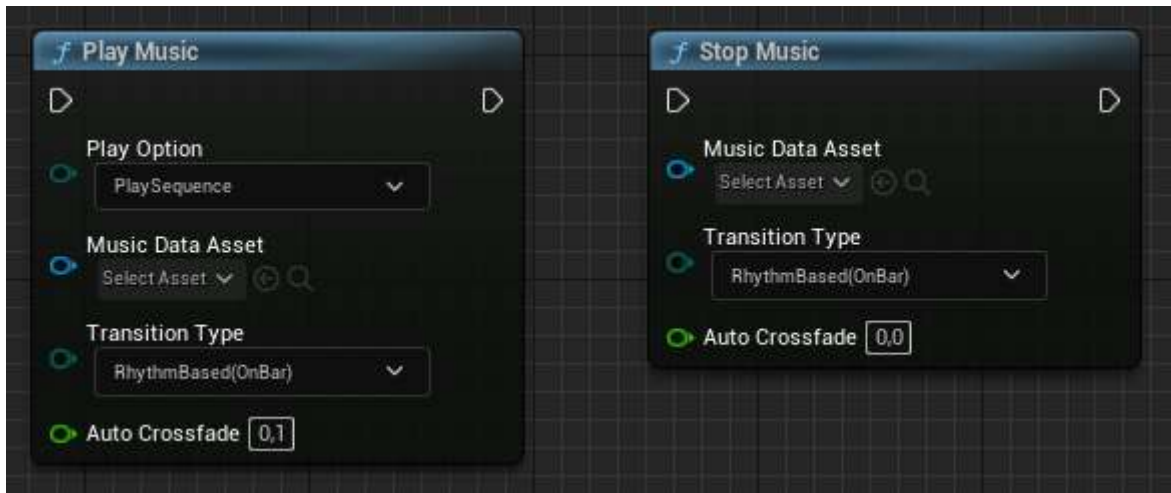
Im Abschnitt „Trigger: Collision“ kannst du nun die Form und die Größe des Colliders bestimmen.



Starte das Level. Falls du nicht innerhalb der Collision spawnst, laufe mit deinem Charakter hinein. Du solltest nun den ausgewählten Track hören.

## ===== 2.0.0 Music Control =====

### --- 2.1.0 Play & Stop Music ---



#### Play Option:

PlaySequence: Startet oder wechselt zur ausgewählten Sequence oder Clip (DataAsset).

SwitchToClip: Wechselt zum ausgewählten Clip innerhalb einer Tracks.

HINWEIS: Diese Option dient nur dem Wechsel innerhalb eines momentan abgespielten Tracks. Falls das System sich momentan nicht in diesem Track befindet, bleibt das System unberührt!

#### Music Data Asset:

Wähle das für die Clips und Sequences entsprechende DataAsset um diese abzuspielen.

HINWEIS: Bei „Stop Music“ hast du die Möglichkeit MDA\_None auszuwählen. Mit diesem „DataAsset“ kannst du den Player komplett beenden. Selbst Tracks mit „Priority“ werden dadurch gestoppt.

#### Transition Type:

RhythmBased(OnBar): Verzögert eine Ausführung bis zum nächsten Takt des aktuellen Tracks

Crossfade(Immediately): Die Ausführung beginnt sofort. Ein Wechsel der Musik findet lediglich durch Crossfades statt.

### Auto Crossfade:

Der Auto Crossfade dient der einfachen Justierung des Crossfades zwischen zwei Tracks, kann aber auch nur als FadeIn oder FadeOut genutzt werden.

=0 deaktiviert den Crossfade. Für die Tracks gelten nun die voreingestellten Werte im DT\_Music\_Sequences\_“BundleName“

<0 der FadeIn vom nächsten Track wird für den FadeOut des aktuellen Tracks genutzt. Der eingestellte Wert wird zu dem aktuellen Wert addiert.

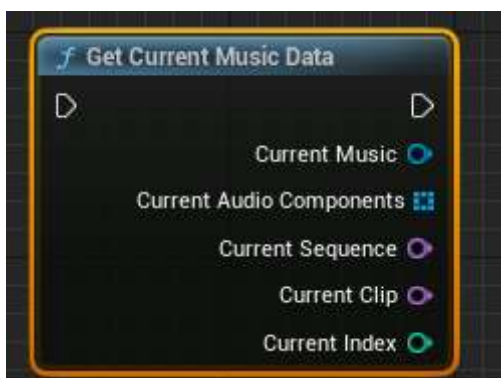
>0 der FadeIn wird durch den absoluten Wert des negativen Auto Crossfades ersetzt.

### --- 2.2.0 Music Settings ---



Ermöglicht die Abfrage und Kontrolle der aktuellen Lautstärke definiert zwischen 0 und 1.

### --- 2.3.0 Get Music Data ---



Ermöglicht das Abfragen von aktuellen Werten des Music Systems.

#### Current Music:

Gibt das aktuell abgespielte DataAsset zurück

#### Current Audio Components:

Gibt alle momentan spielenden Audio Components zurück

#### Current Sequence:

Gibt den aktuellen Namen der Sequence/Track zurück

#### Current Clip:

Gibt den aktuellen Clip Namen zurück. Dieser setzt sich wie folgt zusammen: „sequenceName|clipIndex“

#### Current Index:

Gibt den aktuellen Index des aktuellen Clips zurück. Der Index wird durch die Platzierung des Clips innerhalb einer Sequence im DT\_MusicSequences definiert.

## ==== 3.0.0 Music Listener ====

### --- 3.1.0 Interface: I\_MusicSystem\_Listener ---

Du hast die Möglichkeit Funktionen auf dem Rhythmus des Tracks auszuführen. Um Zugang dazu zu bekommen implementiere das Interface: „I\_MusicSystem\_Listener“ im entsprechenden Blueprint.

Um Informationen vom Interface nur dann zu erhalten, wenn es notwendig ist, musst du dich mit der Node: „SubscribeToMusicSystem“ in das Music System einschreiben. Nutze „UnsubscribeToMusicSystem“ um die Verbindung zu lösen.



--- 3.2.0 Interface: Events ---

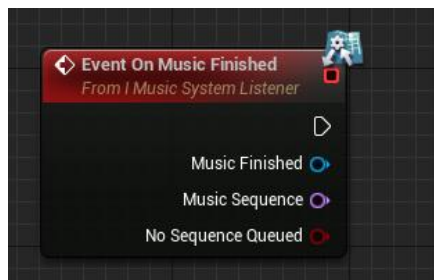
3.2.1 OnCurrentMusicTimeSignature

Bietet die Möglichkeit eine Execution auf verschiedenen Time Signatures des aktuellen Tracks auszuführen. Des Weiteren gibt das Event den aktuellen Takt und die Zählzeit mit.



3.2.2 OnMusicFinished

Bietet die Möglichkeit eine Execution am Ende eines Tracks auszuführen. Des Weiteren gibt das Event das DataAsset, den SequenceName und die Information, ob ein weiterer Track gespielt werden soll aus.



3.2.3 OnMusicStart

Bietet die Möglichkeit eine Execution am Anfang eines Tracks auszuführen. Des Weiteren gibt das Event das DataAsset und den SequenceName aus.





## ==== 4.0.0 Music Listener ====

### --- 4.1.0 Trigger Overview ---

Das System liefert 2 fertige Trigger Blueprints mit.

„Music\_TriggerPlay“ & „Music\_TriggerStop“.

Um diese zu nutzen, ziehst du den gewünschten Trigger einfach in dein Level und justierst diesen im Detail Tab.

Wenn du während des Entwicklungsprozesses diese Trigger im Spiel sehen möchtest, kannst du im Construction Script des jeweiligen Triggers den Bool „ShowTriggerInGame“ auf „true“ setzen.



### --- 4.2.0 Details Tab ---

#### --- 4.2.1 Trigger: Music ---

In diesem Abschnitt findest du exakt die gleichen Einstellungsmöglichkeiten wie bei den Nodes „PlayMusic“ & „StopMusic“ (siehe Punkt 2.1.0 Play & Stop Music)

#### --- 4.2.2 Trigger: Collision ---

Trigger Priority:

Dieser Wert entscheidet, welcher Trigger bei Überlappung verschiedener Collisions genutzt werden soll.

Shape:

Wähle eine der implementierten Collider. Für komplexe Räume hast du die Möglichkeit die Box oder Sphere durch ein Trigger Volume zu ersetzen. Bei dieser Auswahl musst du das entsprechende Volume noch unter „Trigger Volume“ zuweisen.

### Sphere Radius:

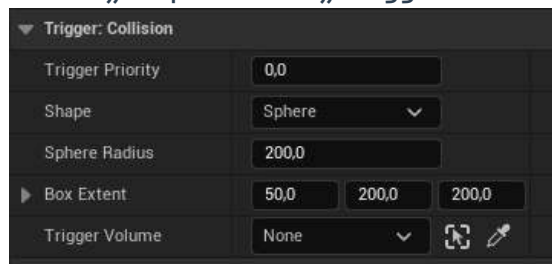
Vergrößert oder verkleinert den Radius der „Sphere Collision“

### Box Extet:

Vergrößert oder verkleinert das Volumen der „Box Collision“ entsprechend der x, y, z Achse.

### Trigger Volume:

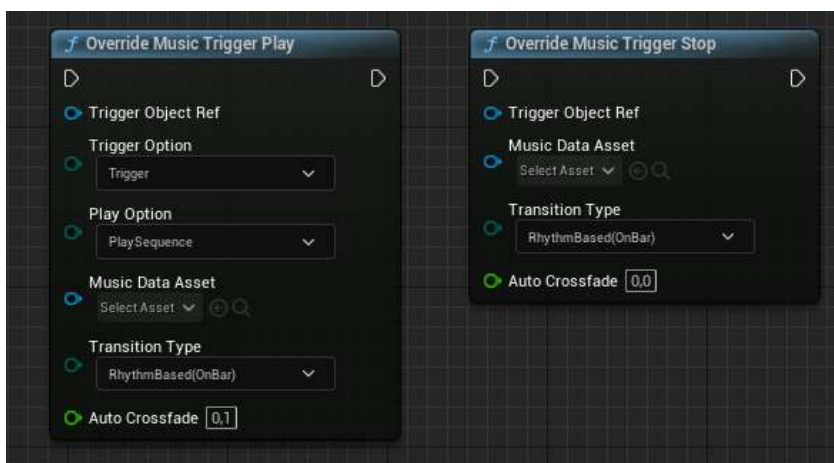
Wähle mit Hilfe des Dropdowns oder der Pipette das entsprechende „Trigger Volume“. Achte darauf, dass bei „Shape“ auch „TriggerVolume“ ausgewählt ist.



### --- 4.3.0 Override Trigger ---

Das System bietet die Möglichkeit die gewählten Werte der Kategorie „Trigger: Music“ in Runtime zu überschreiben.

Wähle hierzu eine der entsprechenden Nodes und referenziere den zu überschreibenden Actor.

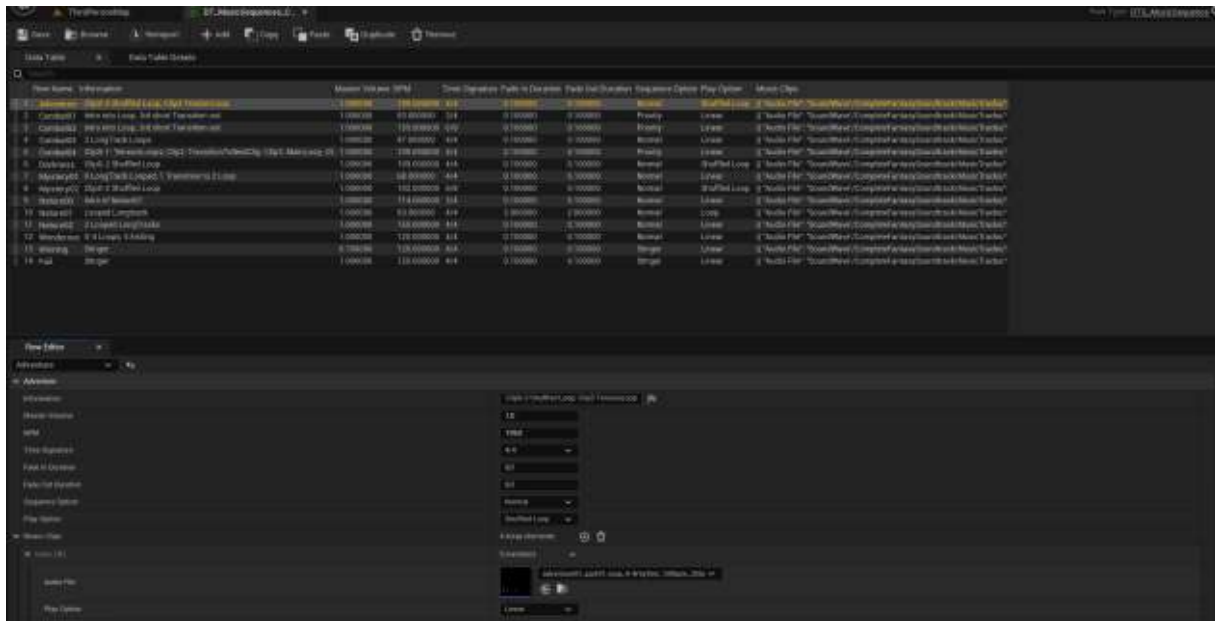


## ==== 5.0.0 Data Table DT\_MusicSequences =====

### --- 5.1.0 DataTable Overview ---

Wie in der Einleitung schon beschrieben, werden in diesem „DataTable“ die einzelnen Clips wieder zu ganzen Tracks zusammengesetzt und zudem mit Parametern versehen.

Eine „DataTableRow“ entspricht einer Sequence oder auch einem ganzen Track.



### --- 5.2.0 Sequence ---

#### Information:

Ein Textfeld ohne Einfluss auf das System. Es dient lediglich zum Speichern von Notizen.

#### Master Volume:

Gibt die Ausgabelautstärke der gesamten Sequence mit einem Wert zwischen 0 und 1 an.

#### BPM:

Ein Integer stellvertretend für die Geschwindigkeit des Liedes.

#### Time Signature:

Im Dropdown kannst du die Taktart des Liedes auswählen.

Fade In Duration:

Die Zeit als Float, die der Track benötigt, um seine volle Lautstärke zu erreichen (sofern der Auto Crossfade ausgeschaltet ist!)

Fade Out Duration:

Die Zeit als Float, die der Track benötigt, um vollständig zu verstummen. (sofern der Auto Crossfade ausgeschaltet ist!)

Sequence Option:

Normal:

Standardeinstellung einer Sequence.

Priority:

Die Sequence muss explizit beendet werden, bevor ein Wechsel in eine andere Sequence möglich ist.

Stinger:

Ein musikalischer Einwurf, der unabhängig von der aktuellen Sequence abgespielt wird. Eignet sich gut für kurze „Winning-Jingles“ o.Ä.

Play Option:

Linear:

Die Sequence spielt die Music Clips in vorgegebener Reihenfolge und wiederholt sich nicht.

Loop:

Die Sequence spielt die Music Clips in vorgegebener Reihenfolge und beginnt von vorn, sobald der letzte Clip zu Ende ist.

ShuffledLoop:

Die Sequence spielt sämtliche enthaltene Tracks die NICHT die Option „ExcludeFromShuffle“ oder „TransitionOut“ haben in zufälliger Reihenfolge.

--- 5.2.1 Music Clip ---

Der Music Clip ist der kleinste musikalische Baustein in einer Sequence. Innerhalb einer Sequence befinden sich 1 oder mehrere Clips in einem entsprechenden Array.

Audio File:

Wave oder Sound Cue.

Play Option:

Linear:

Der Clip spielt einmal durch.

Loop:

Der Clip loopt so lang, bis du ihn beendest.

**ExcludeFromShuffle:**

Der Clip verhält sich wie „Linear“ nur dass dieser von der Shuffle Funktion ignoriert wird

**TransitionOut:**

Dieser Clip wird beim Beenden (OnBar) als Abschluss Clip gespielt. Wechselt man „Immediately“, wird dieser Clip ignoriert.

**Pre Roll Length (Bars):**

Die Länge Intros o. Auftakts in Takten als Integer.

**Body Length (Bars):**

Die Länge des Clips (ohne Intro) in Takten als Integer.

**HINWEIS:**

Ich würde eine Mindestlänge von 2 empfehlen, da das System aktuell Fehler aufweisen kann.

**PDA\_Music:**

Das entsprechende DataAsset, über welches die Sequence oder auch der Clip angesprochen werden soll.

--- 5.3.0 Data Asset PDA\_Music ---

Das Primary Data Asset \_Music ist die Base Class für alle DataAssets, die stellvertretend für die Sequences und Clips genutzt werden, um diese zu steuern. Wenn du einen eigenen Song einpflegen willst, musst du mindestens ein entsprechendes DataAsset erstellen und dem Clip zuweisen.

**Music Sequence:**

Muss dem entsprechenden DataTableRowName entsprechen.

**Music Clip Index:**

Muss dem entsprechenden Clip Index entsprechen. Hier ist es auch möglich, das Array zu erweitern. Dies dient der Funktion „Shuffle“, um einen zufallsbasierten Einstieg in einen der Clips zu erhalten. Das DataAsset muss dann jedem der Clips zugeordnet werden.