

# 世界を守るサウンド制作

~EARTH DEFENSE FORECE IRON RAINにおけるサウンド実装について~



Sound ams inc.

# 田村 幸多郎

2007年サウンドエイムスに入社

ものづくりを愛する、手作り大好きな開発者です。

最近は音声認識に興味があり、色々実験しています。

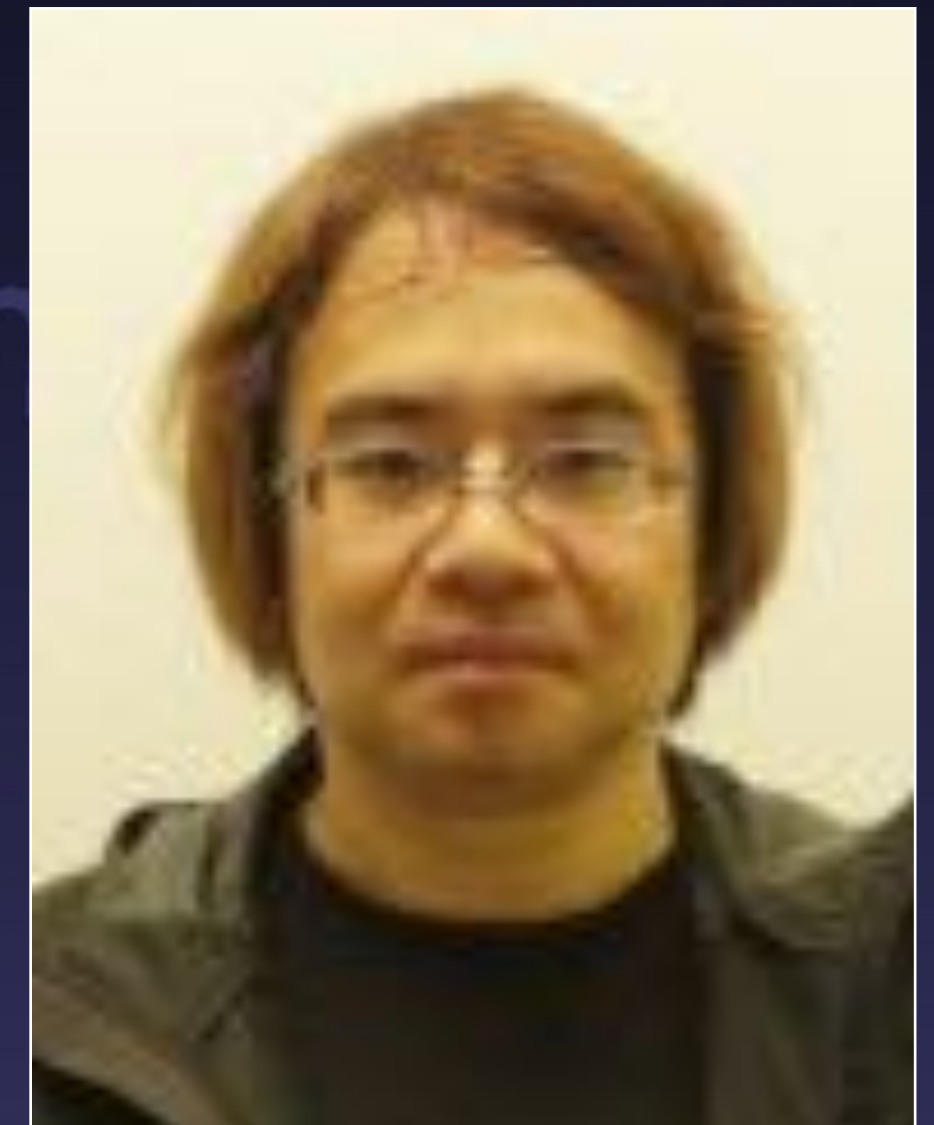


# 齊藤 孝之

コンシューマーゲーム開発会社、音楽制作会社を経て

2005年 有限会社サウンドエイムスに入社。主に効果音制作、サウンドデータ実装を担当。

根っからのゲーム好きでゲームデザインに合わせたサウンド構築に燃えてしまう性格。



1. EARTH DEFENSE FORCE IRON RAINとは
2. BGMの実装
3. SFXの実装
4. Voiceの実装
5. 処理負荷対策



Sound AMS inc.

1. EARTH DEFENSE FORCE



IRON RAIN

とは

地球を侵略者から守る3Dアクションゲーム。

本家、地球防衛軍シリーズと名前が似ておりますが、

全くの別コンセプトで1から、開発をしている案件です。



開発 Yuke's様

SoundAms はサウンド全般の作業を行いました。

- ・ 音の作成 BGM, SFX, Voice(キャストイング含む)
- ・ 武器やプロップへの音の設定
- ・ アニメーションへの音の通知設定
- ・ レベルへの音配置、作業

CRI様とは長年の付き合いがあり、プロジェクト毎に条件も環境も違う中で、オーディオの実装を同じように使用することができる為。

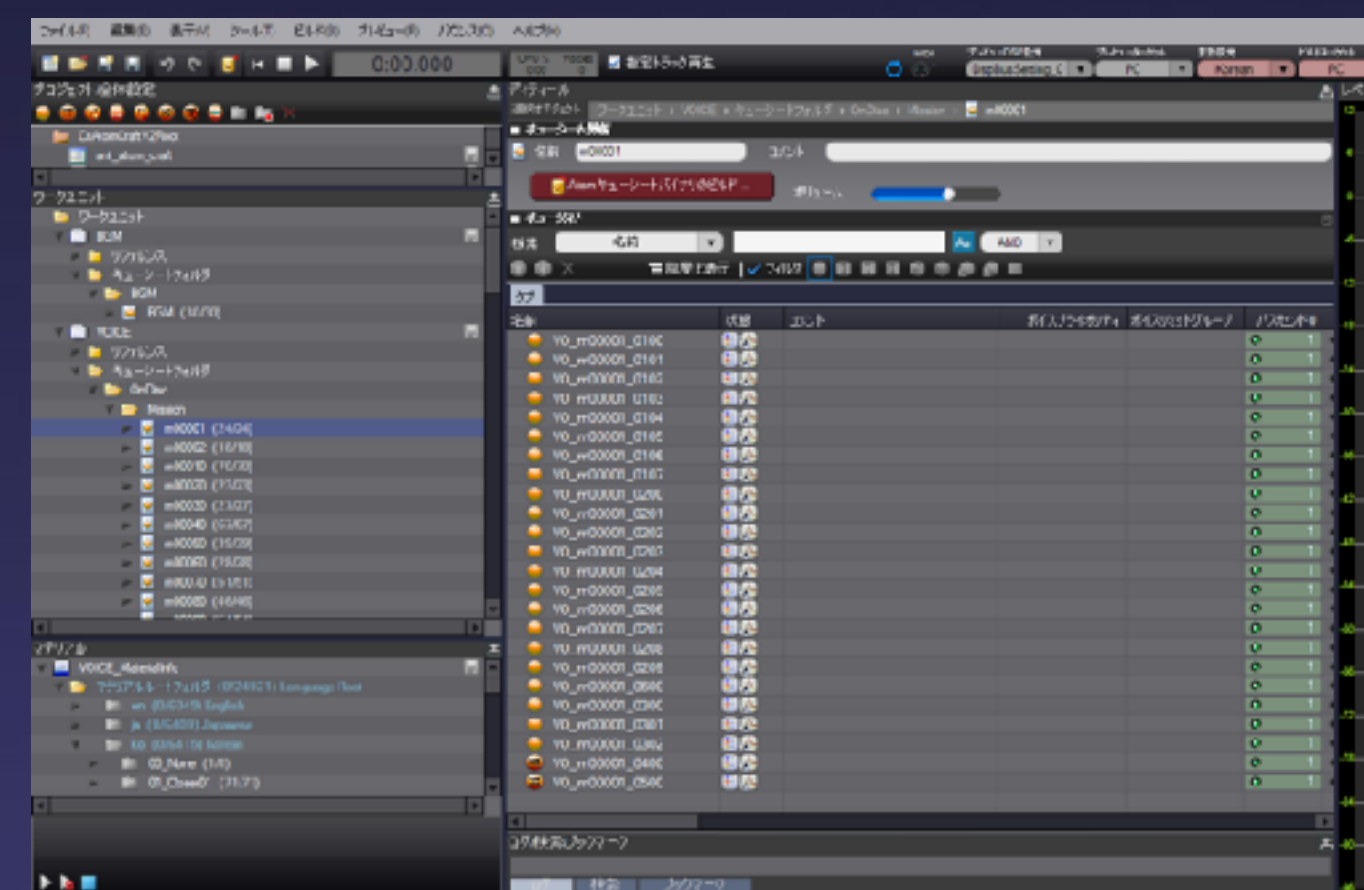
= ノウハウを生かすことが出来るので、作業効率が上がる！

※要望を、プロジェクト限定対応ではなくミドルウェアの機能としてサポートしてくれるので、ユーザー全てが恩恵を受けることができる。

プロジェクトA  
(ゲームエンジンA)

プロジェクトB  
(自社エンジン)

同じツールが使える！

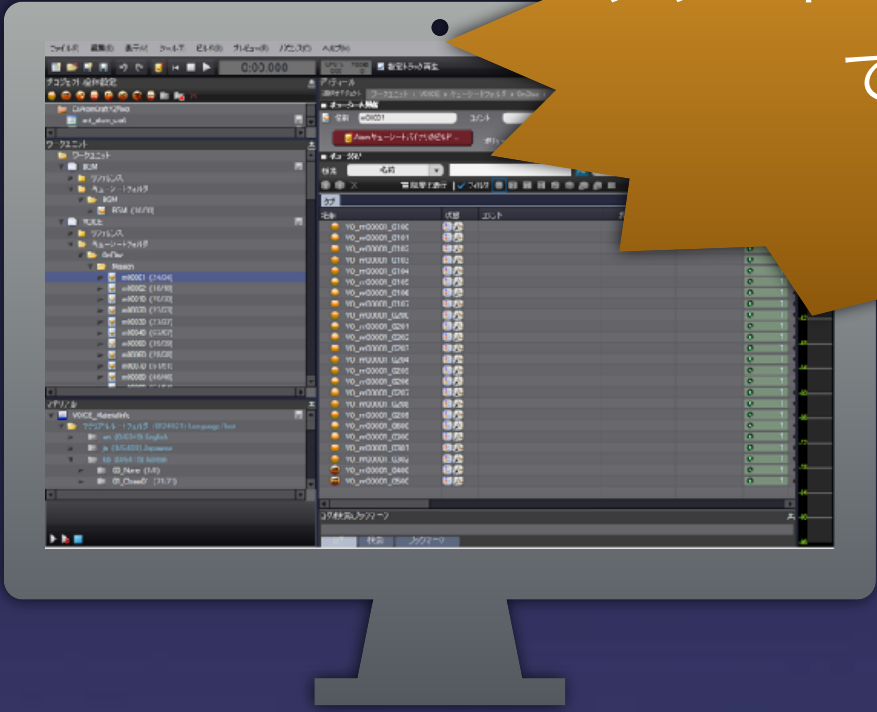




ゲームエンジンとサウンドエンジンを分離できる。

= エンジン更新への影響が分けられるので、デザイナーの作業が止まりにくい！

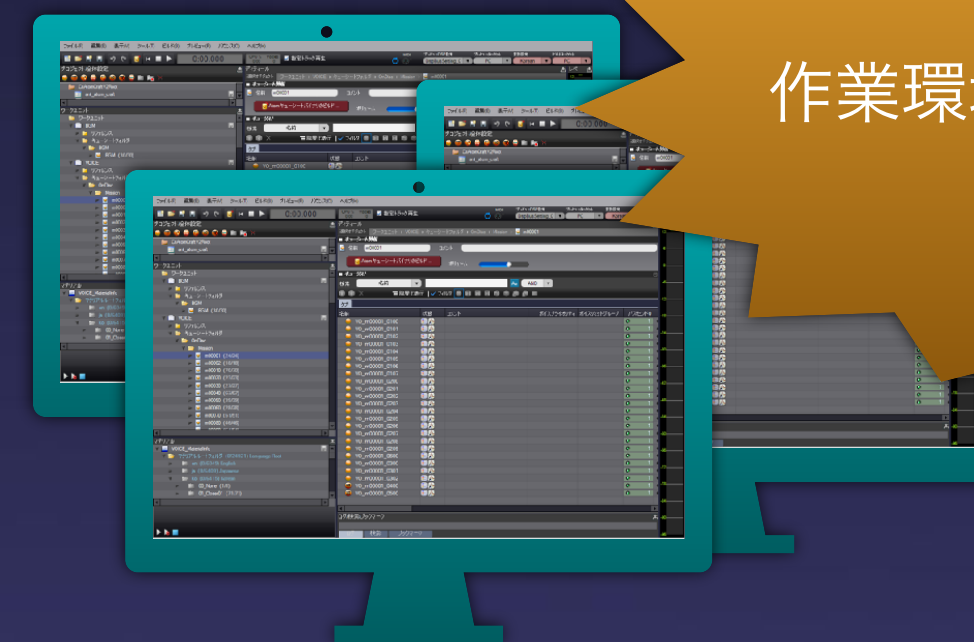
※Atom Craft(オーサリングツール)は軽量なツールなので、作業環境のコストも下げられる。



サウンドの作業はAtom Craft  
で進められる



ハイスペックPC



ミドルスペックPC

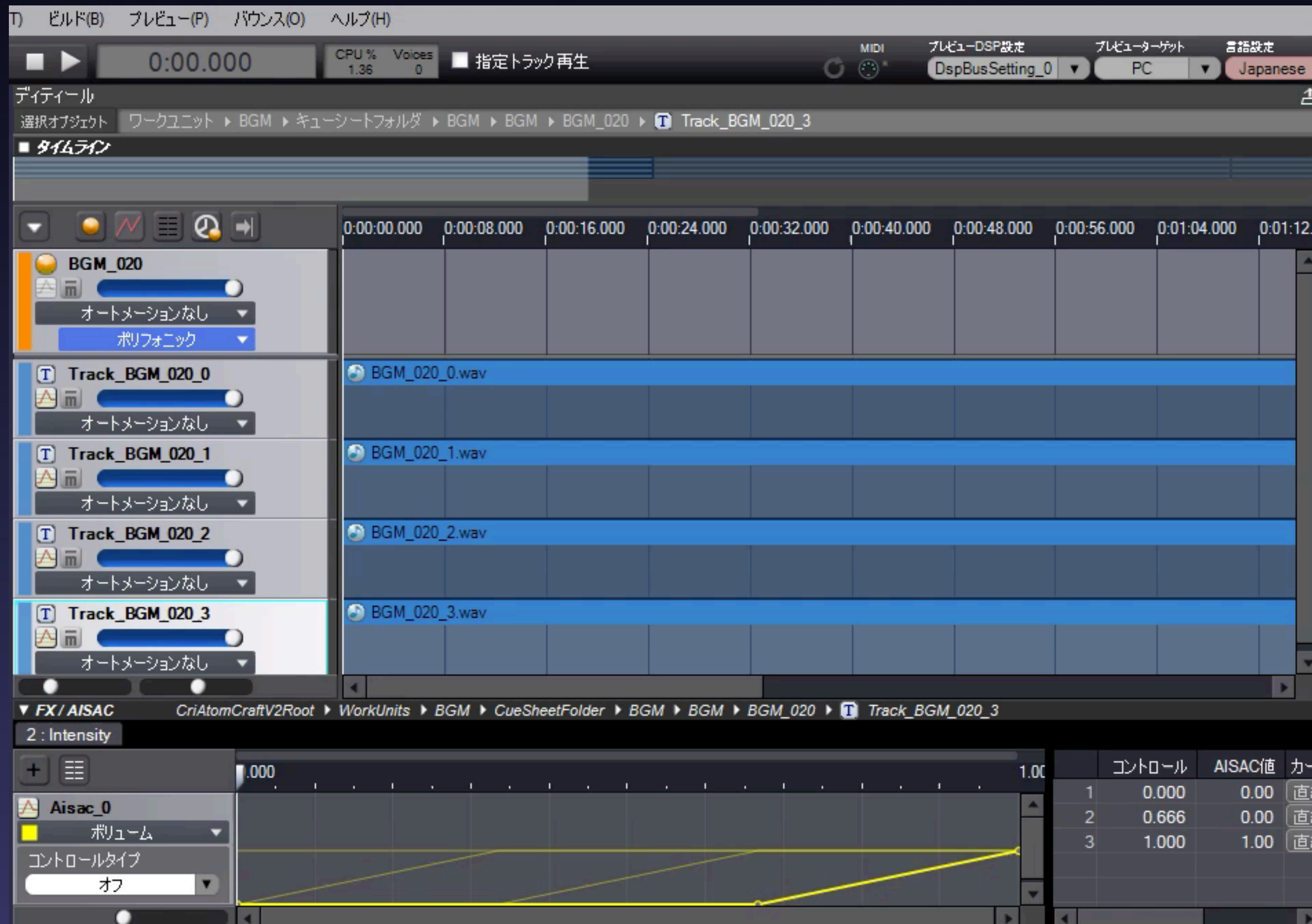
作業環境の用意がしやすい





## 2. BGMの実装

# トラックの音量を操作してBGMを変化させる



どのトラックが再生されているのか？フェードIOしているのか？などプログラムからの管理が面倒です。

AISACを使うことにより、プログラムからは条件に合わせて定義した値を変更するだけで良くなります。今回は Intensity という値を用意してBGMの動的な変化を表現しました。

※AISAC:プログラムからコントロールできる値を定義して、それを元にデザイナー側で音量や音程など様々なパラメータを設定できる。

※ADX2にはブロック再生というもっと細かい制御が出来る機能もあります。

↑  
音量

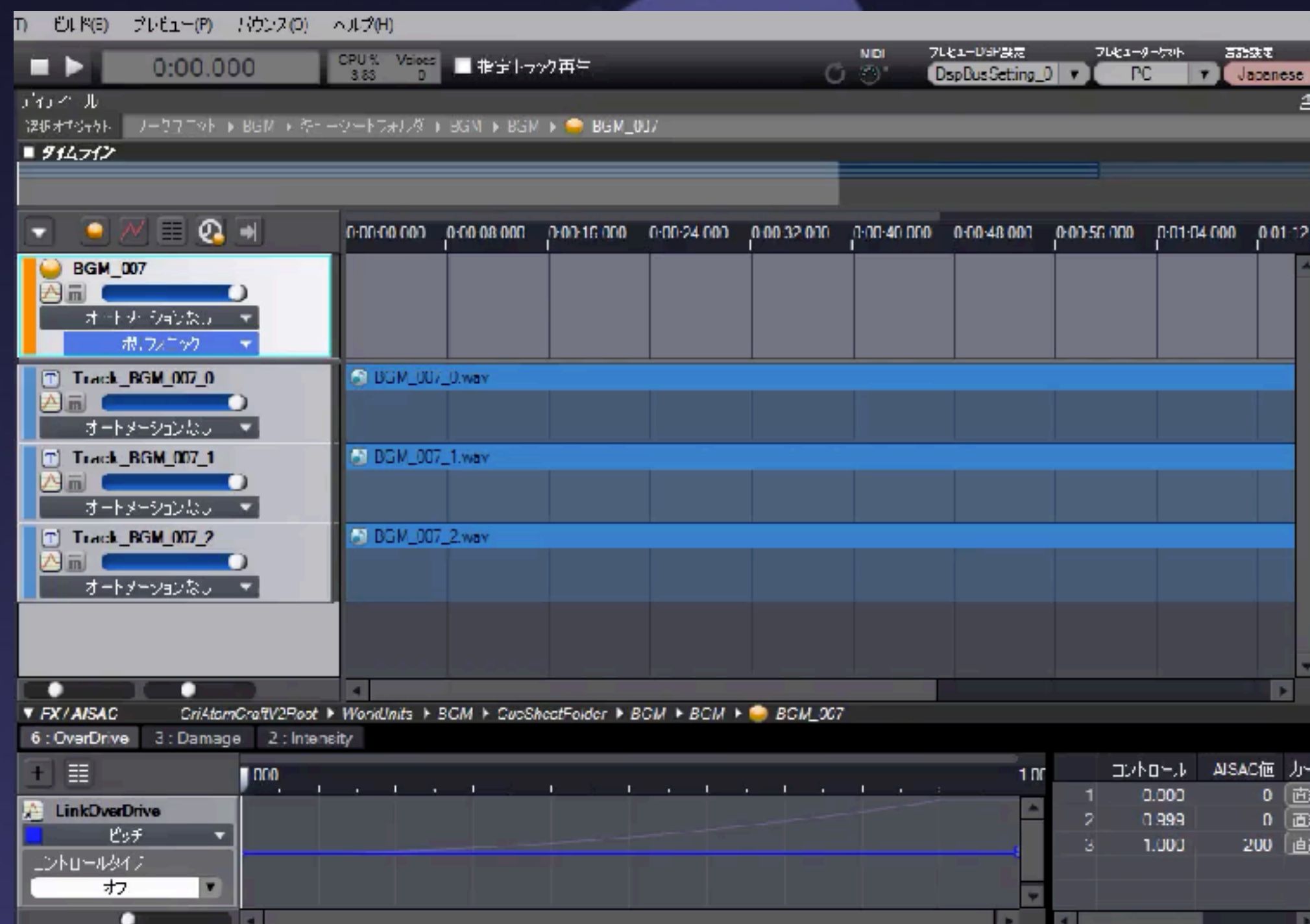
Intensity

# フィルターやピッチを変化させる

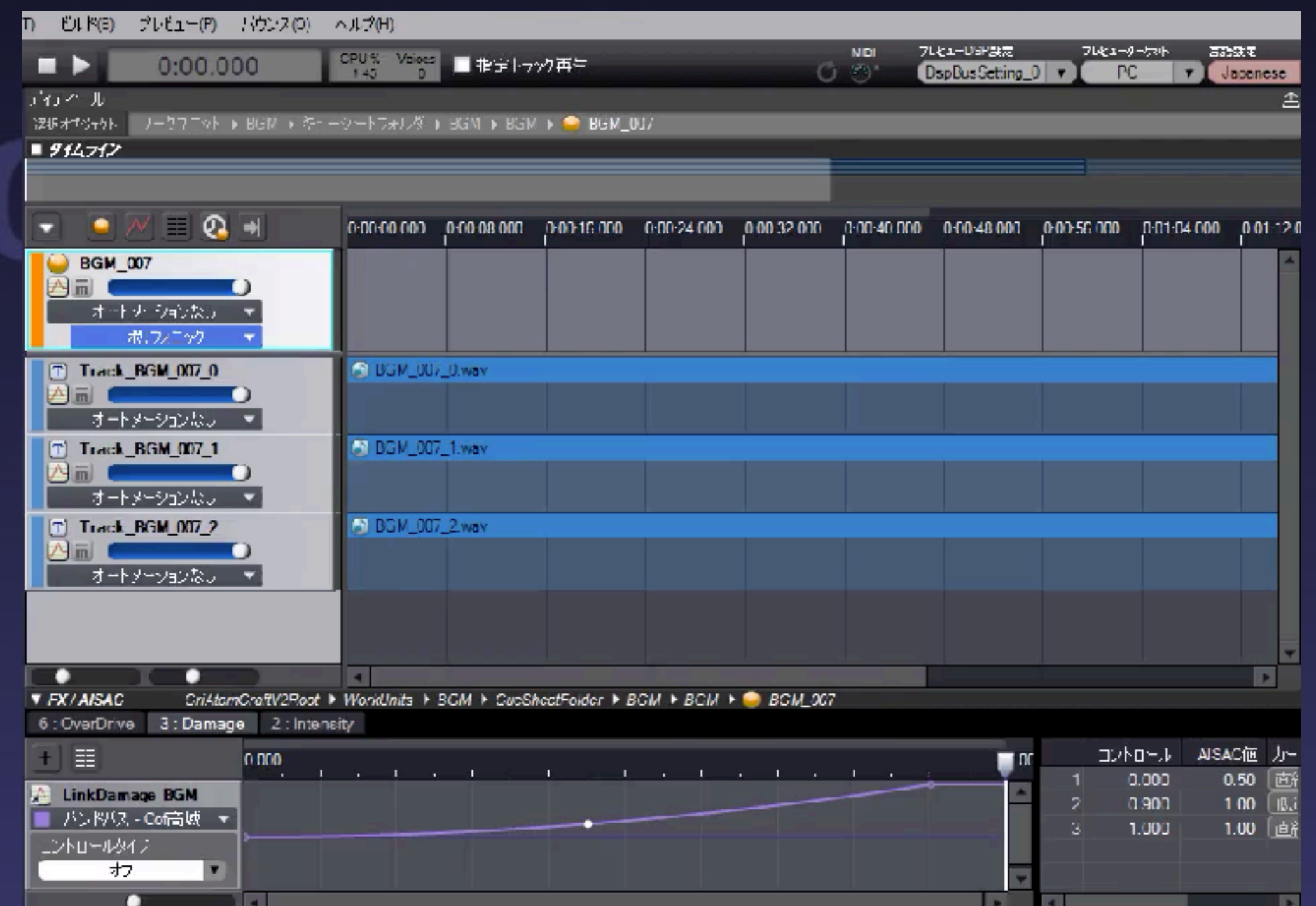
ゲームプレイ中「能力が上がる、戦闘エリア外に出る、大きなダメージを受ける」など変化がある際にAISACを使って、BGMを変化させゲームと合わせた変化をさせました。

OverDrive : ピッチが上がる

Damage : フィルターがかかる

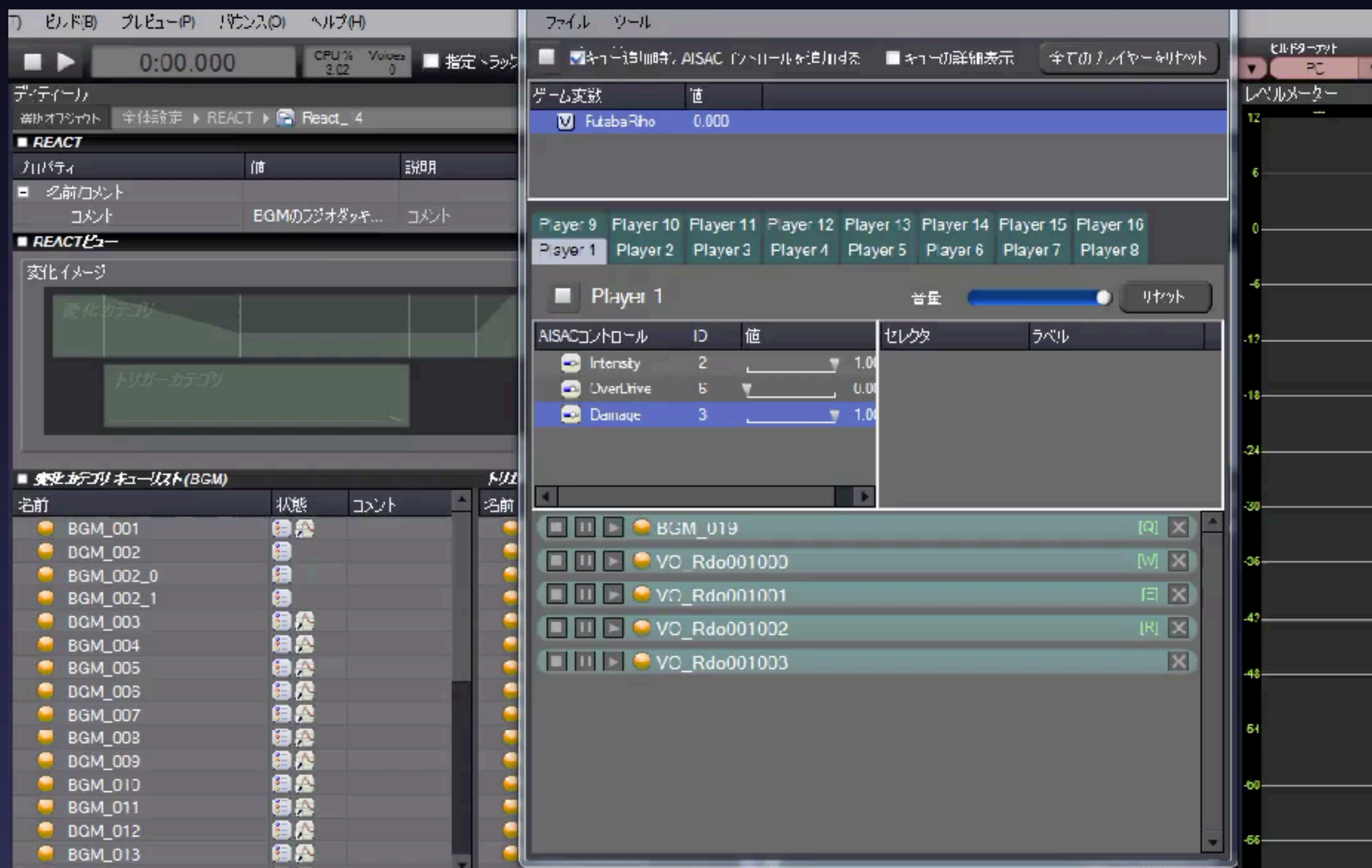


※どの音がどのように変化するかはデザイナー側で設定できます。



※AISACは他にも色々とコントロールできます。

# ボイス再生時にBGMを下げ、ボイスを際立たせる



プレイ中にストーリーが進行しますので（ボイスが再生される）大量のSFXがなる中ボイスを聞き取りやすくする為、BGMにダッキングを設定しました。

REACT機能を使うことにより、デザイナー側で簡単に実現できます。

※REACT:何かのキューをトリガーとして他のキューへ何らかのアクションを起こす機能

※ダッキング：ある音を聞かせる為に、他の音量を一時的に下げる手法



## 2-4 ゲーム画面



※画面のエフェクトに合わせエフェクトが掛かりする。



# 3. SFXの実装





# 3-1. 素材の収録



銃や戦車などの火器が多く登場することもあり総合火力演習でロケを行いました。

各種武器や乗り物の他、環境音など素材として使用しました。

### 3-1-2 素材の収録2 プロップ等



工場に協力していただき、鉄、木、ガラス、コンクリなどの破壊音、衝突音、工作機械や工具など収録を行いました。

主にマップに配置される小物、建築物の倒壊音、敵味方キャラクターの動きなどに使用しています。

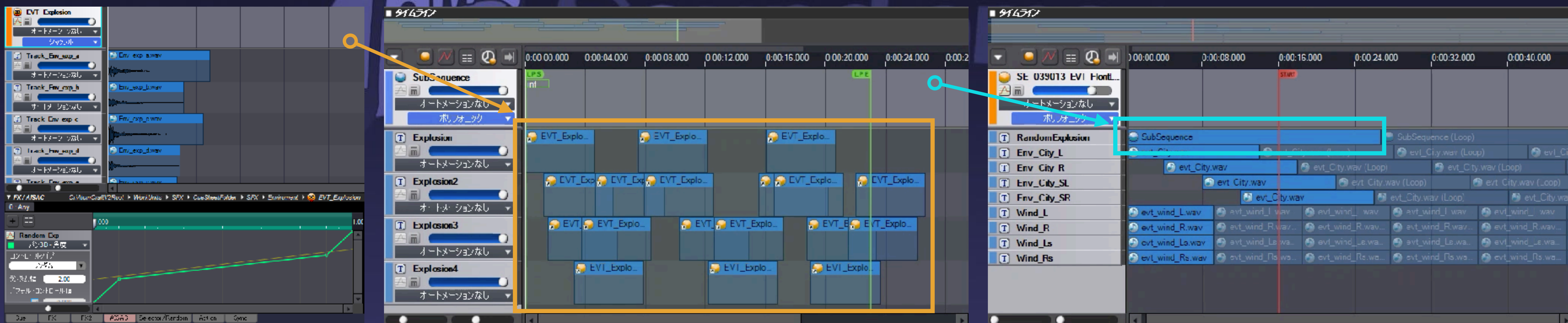


## 3-2. 環境音

## シーケンス機能とランダム再生を組み合わせてベースとなる環境音を構築

WAVループでの環境音では、繰り返し感が気になったり、かといって長くするとその分データサイズが大きくなります。

ADXのシーケンス機能とランダム機能を組み合わせることで繰り返し感の少なくしながら、データサイズを抑えることができます。



ランダムに素材、定位、遅延等を変化

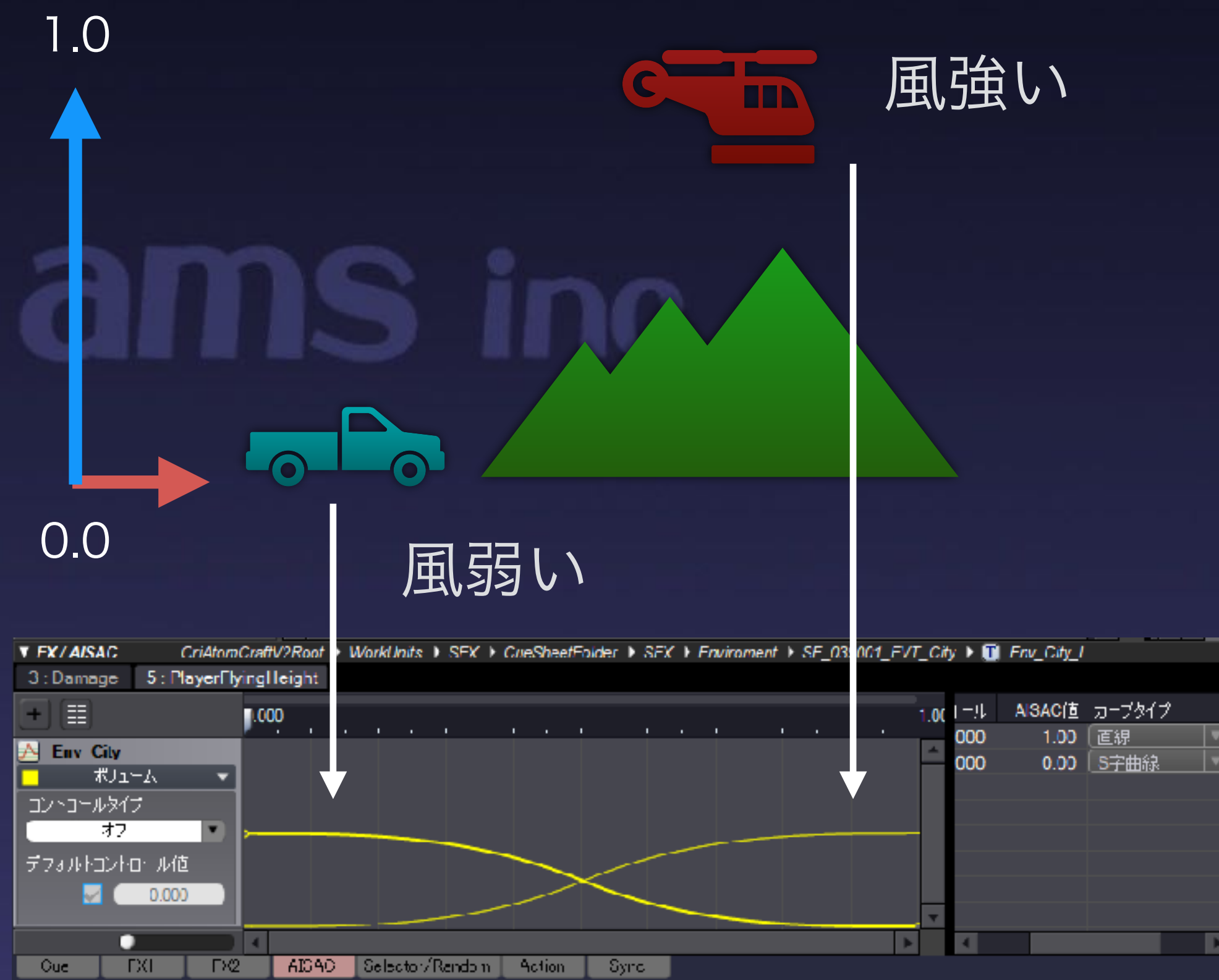
シーケンスに素材を配置

ベース環境音から呼び出す

※ビルや大型の敵の破壊音などにも応用しています。

# 高度変化に応じて、動的に環境音が変化

地表からの高さ(Z軸)をAISACに設定して環境音素材のミックスマバランスを変えることで実現しています。



※各レベルに最小最大の高さを設定し、毎フレームプレイヤーの高さ元に更新

※風以外にも、海面などに利用

# 固定のオブジェクトはAtom Componentをレベルに配置

ベース環境音と合わせて、再生することにより位置による環境音の変化を感じさせます。



注意したい点として同じ音が重なることにより、フランジング（音やせしている様な）など違和感がある聞こえがする場所が出てくる可能性があります。

素材のバリエーションを用意して同じ音が隣りあわせにならないようにしています。

※こだわろうと思えば幾らでも置けてしまうので、ゲームへの負荷、発音数なども考慮する。

# 3Dサウンドに自由な形状を持たせる機能

2019年内にリリース予定



例えば川のような、地形に沿って聞こえるサウンドを実現する機能。

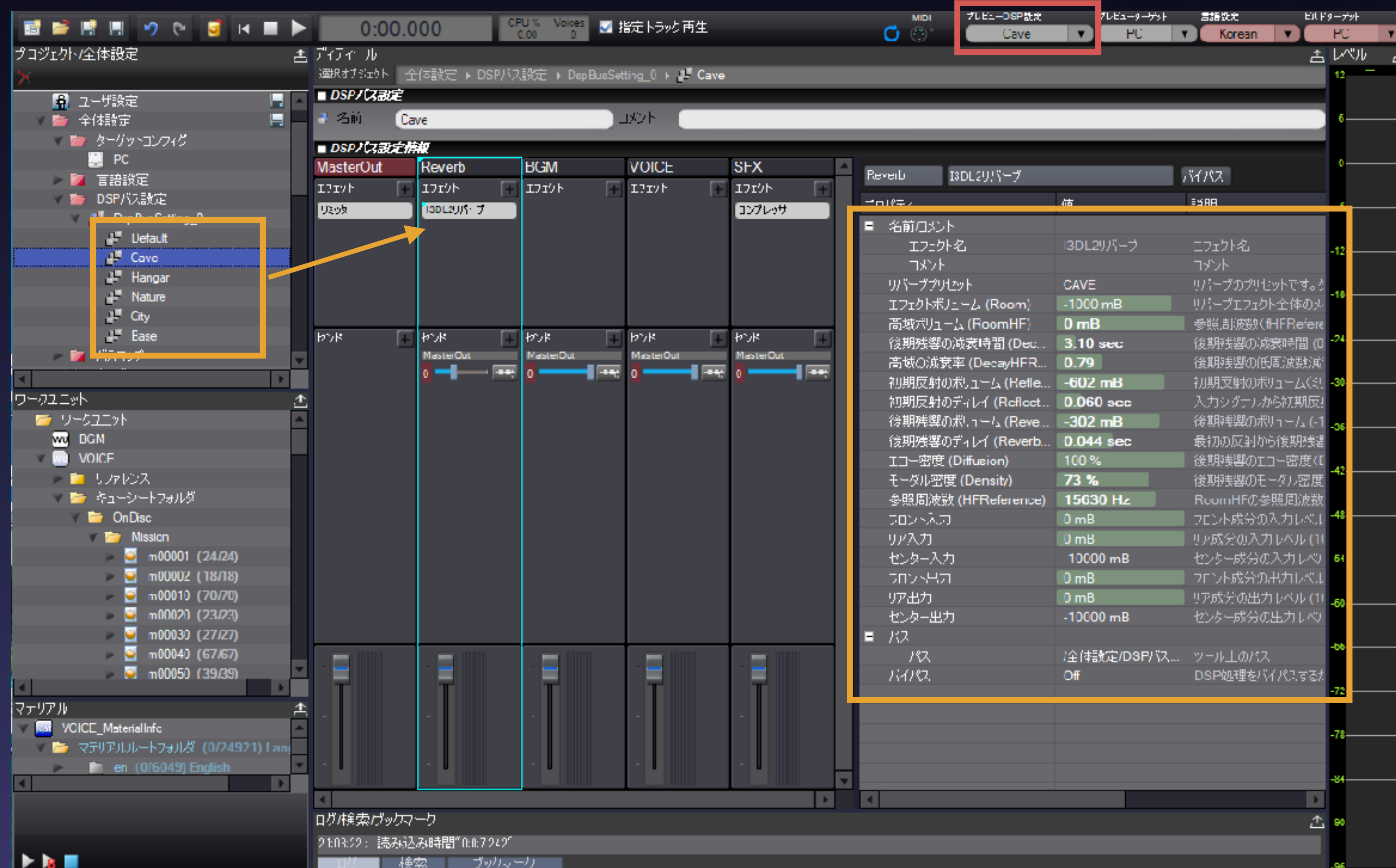
動画の中では、川を覆うように SoundShapeVolume を配置している。1つのVolumeにつき1つの川音源を配置する。

Volumeとリスナーの位置関係に応じて音源(動画中にはスピーカアイコンで表示)が適切な位置へ移動する。

地形に基づく環境音を直感的にデザインできる。プログラム不要。



# スナップショット切り替えによるシンプルなリバーブ実装



本作ではレベル内で大きく空間が変わるシチュエーション（地上から地下に入るなど）が無い為、各レベルに対してスナップショットを設定しました。

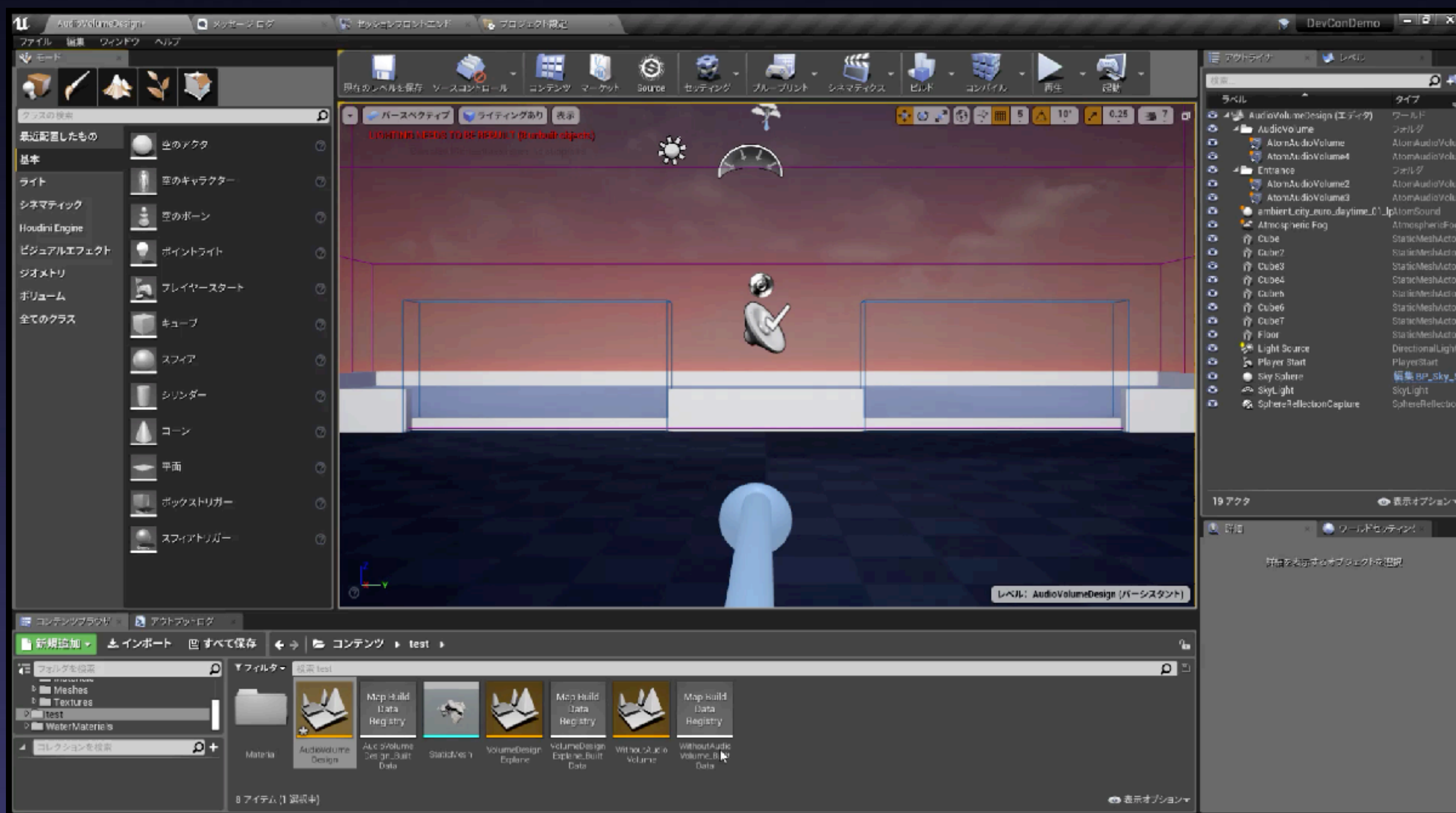
スナップショットはバスのエフェクト、センド設定を保存、切り替えが出来ます。

※個々のCueからリバーブへの送り量は設定。

※Atom Craft上でもプレビューDSP設定の切り替えで、簡単に各リバーブが確認できる。

# 3D空間にサウンド環境の情報を持たせる機能

2019年内にリリース予定



地形に応じて音の聞こえ方は変わる。例えば洞窟内の音は反響する。

動画中では、白い塀で区切られた領域を紫のAtomAudioVolumeで囲っている。Volumeの外で環境音が再生されている。

リスナーがVolumeに入ると環境音の聞こえ方が変わる。どのようなエフェクトを効かせるかはVolume側に設定可能。

青いVolumeは紫Volumeの「入口」を表す。入口から入ると、侵入距離に応じてエフェクトが強まったり弱まったりする。



# 3-3. イングゲーム

# TPSのカメラに合わせてフォーカスポイントを設定

距離減衰はADXの基本機能である3Dポジショニングを使用することで簡単に設定できます。



注意点として、リスナーがカメラかプレイヤーなのかによって定位感が変わってきますのでゲームデザインに応じて考慮する必要があります。

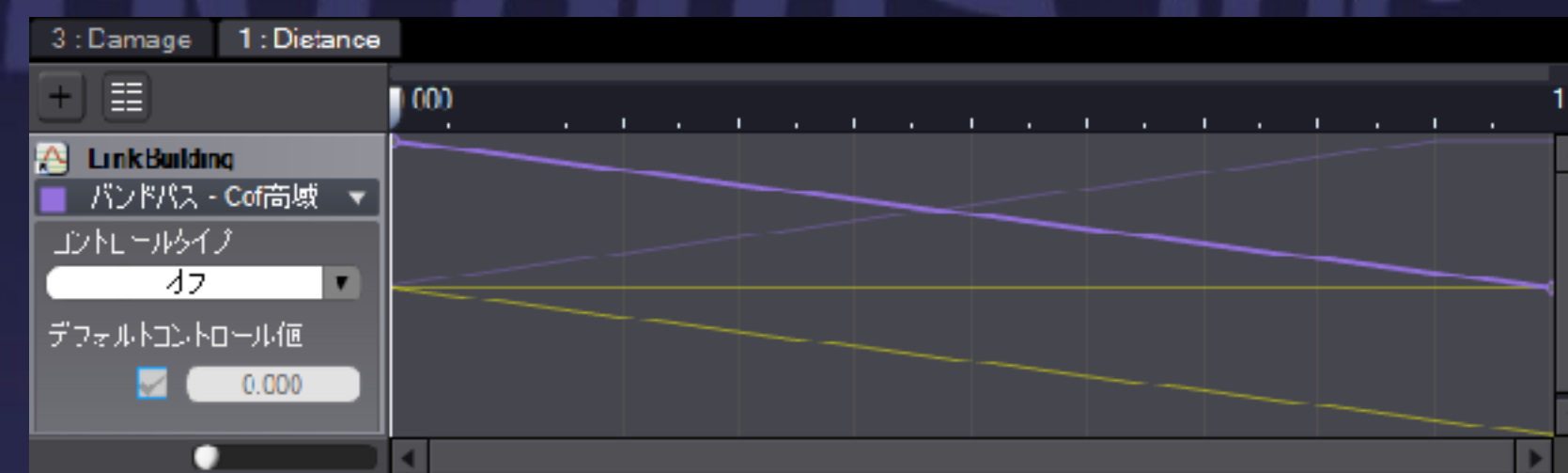
本作はカメラに対してプレイヤーが左寄りに立ちますので、リスナーを完全にカメラ基準するとプレイヤーの攻撃、動作音が左寄りに再生されてしまう為、遊びやすさを重視してプレイヤー側（定位としては中央に近い位置）に寄せた設定にしています。

ユーザー視点で考えると、**オレンジ色のライン**を基準とした方が気持ちよくプレイできる。  
※カットシーンではカメラ位置に戻している

# 距離によるフィルター変化で遠鳴りを表現



”距離が遠くなると音が小さくなる”では”小さな音が近くで鳴っている”ようにも聞こえてしまうので、フィルター効果も合わせてAISACでコントロールすることで距離感を表現しています。



空気を振動して伝わる音は、距離が遠くなるほど高音から減少していく為、離れるとくぐもる様に設定。

# 距離によって音色を変化させる

一つのCUEの中に異なる距離減衰の素材を重ねることでミックスバランスを変化させることができます。



- ・小銃の着弾音などで、プレイヤーの近くで発生したときのみ、跳弾音が目立つ
- ・ボスの足音など、近づいたときに細かな瓦礫の落下音が聞こえる

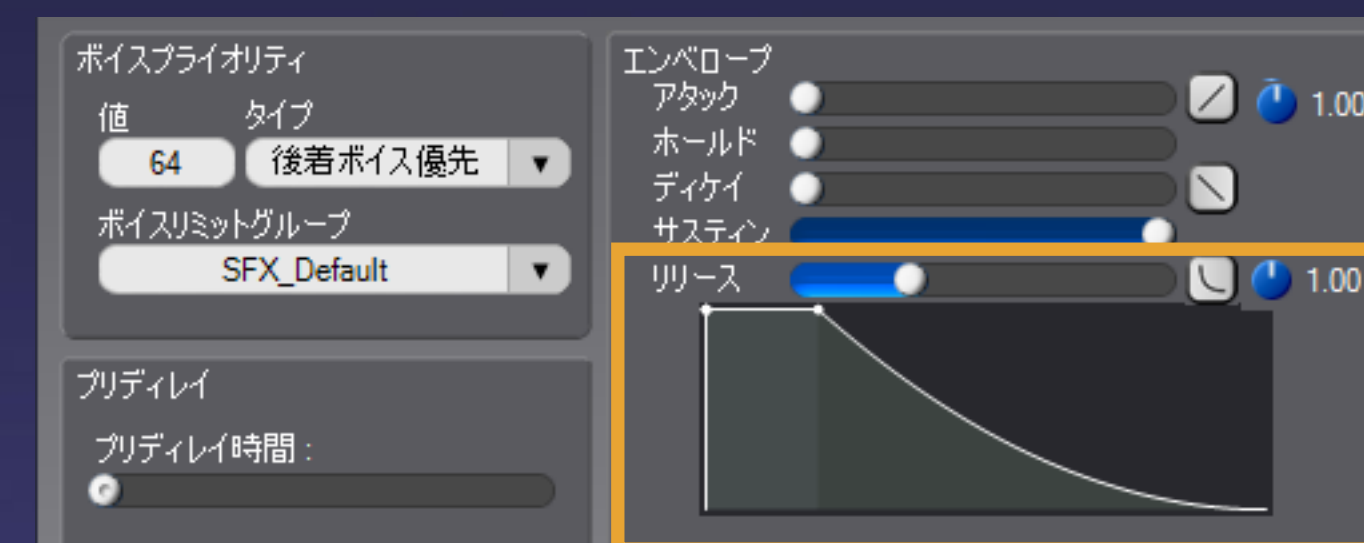
※プログラムからの呼び出しは着弾Cueのみだが、音が変わるのでCueが増えているように感じる。

# リリースを使って、尺違いに対応



すべてオブジェクトやエフェクトに尺を合わせたSFXを用意して制御するのは効率的ではありません。

発音停止時のリリースタイムを設定できますので、消滅に合わせて発音停止する仕組みにすることで、どのような尺でも綺麗にフェードアウトする様な対応が可能です。



※建物の大きさや高さは大量に種類がある為、倒壊するときの尺がちまち。

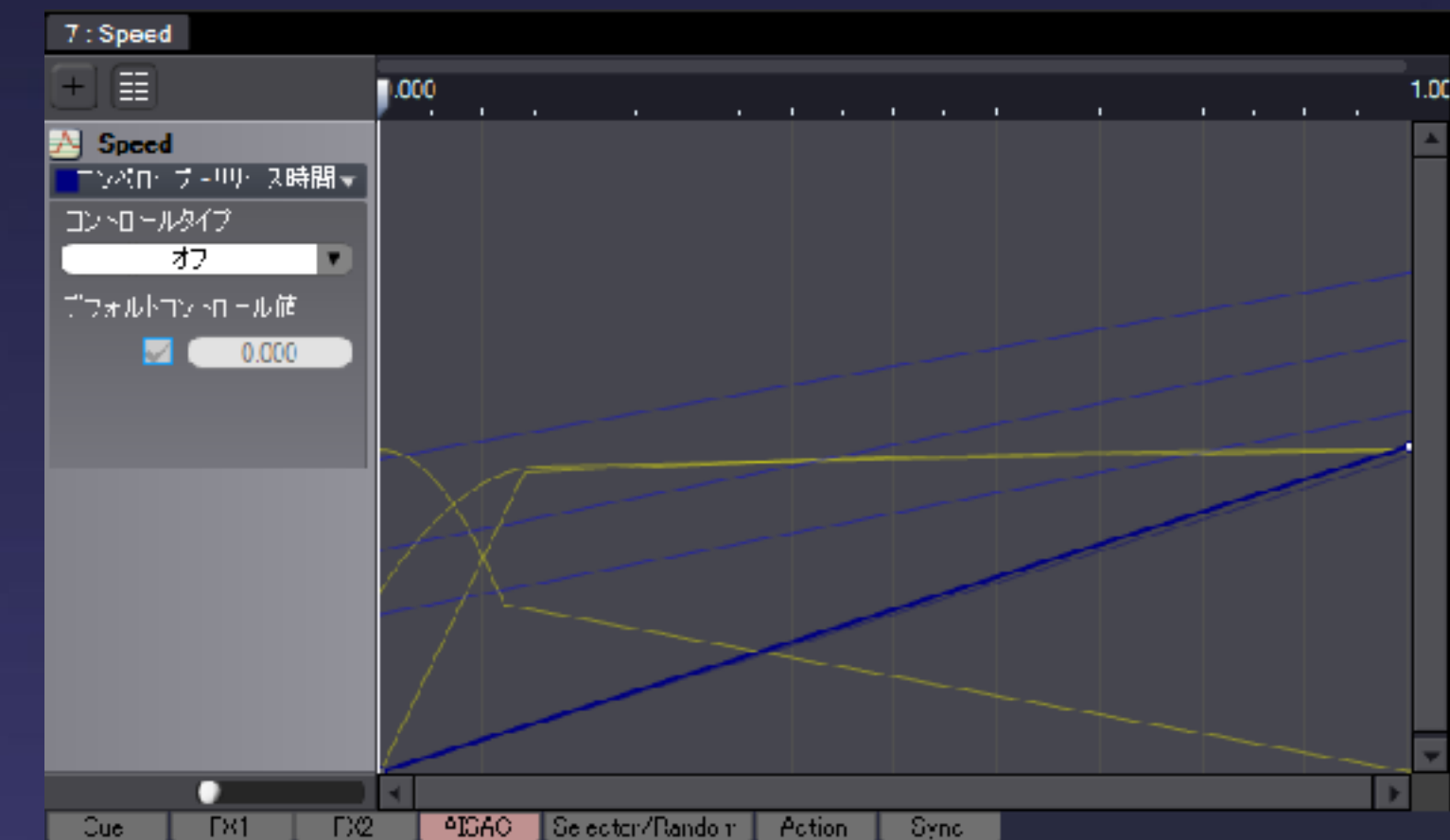
# リリースを使って、PG制御後の変化に対応

ヴィークル（戦車や戦闘機など乗り物）の走行音について、プレイヤー操作中は走行速度をAISACコントロールに割り当てして音色変化をするようにしています。

走行中にプレイヤーが乗り捨てすることも可能なのですが、この時に再生されている走行音のリリースタイムに速度を紐づけすることで、惰性で走るヴィークルの音量が自然にさがるようにしています。



※これは搭乗していないヴィークルについてSFXの制御をしないで済むため、処理負荷の面でもメリットがありました。





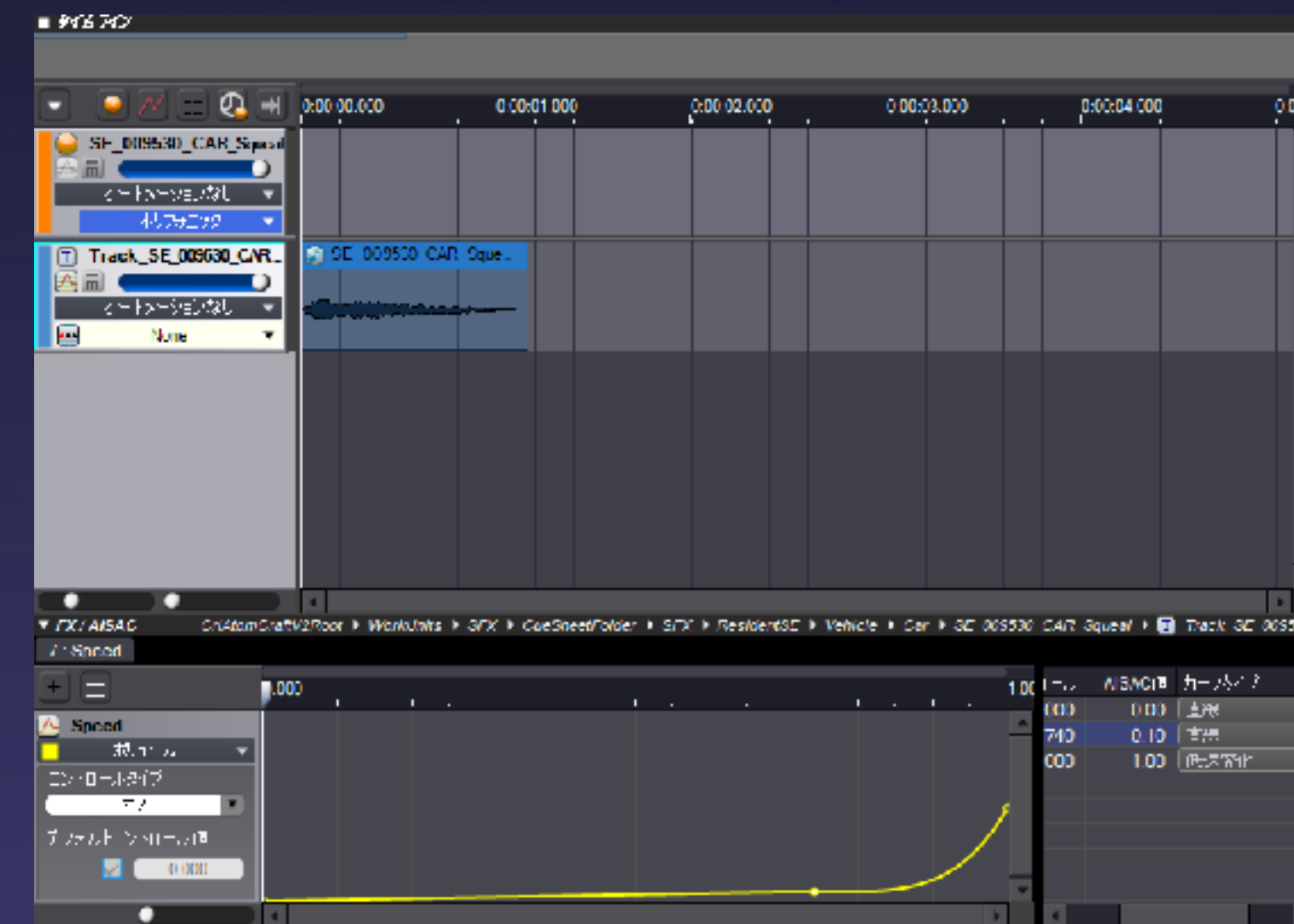
# 速度を利用した、音のコントロール

ブレーキを行った際にスキル音をAISACでコントロールしました。

規定速度を超えた状態で、プレイヤーが後進方向にスティックを入れるとSFXが鳴る処理入れてもらい、サウンド側では速度のAISACを利用してスピードの減速に応じてフェードするようなAISACカーブを設定しています。



※速度と音の関係は密接なので、一度PG対応していただければ、様々なところで利用できます。



# クリティカルヒットでメロディーを奏でる武器

連続してクリティカルヒットが発生すると、ジングルベルのメロディーが流れる。

コンボシーケンシャル機能から思いついたアイデア。



※同じCUEが連続すると再生トラックが変わるという機能。

※ヒット音が再生される仕組みは同じでも、ADX2内のCueの組み方次第で色々遊ぶことができます。

# 衝突時の速度をCue再生時に適応して音を変化させる



吹き飛びやられなどラグドール状態で「地面、壁」などに衝突した際の速さをAISACに渡して

音素材のミックスバランスをコントロールして音色変化するようにしました。



※物理挙動中は様々な速さで衝突します。

遅い → 早い  
速度

The logo for AMS (Sound ams inc.) is located on the left side of the slide. It features the letters 'AMS' in a stylized, bold font, with a microphone icon integrated into the design. The logo is semi-transparent and overlaps with the main title text.

# 3-4. アニメーション

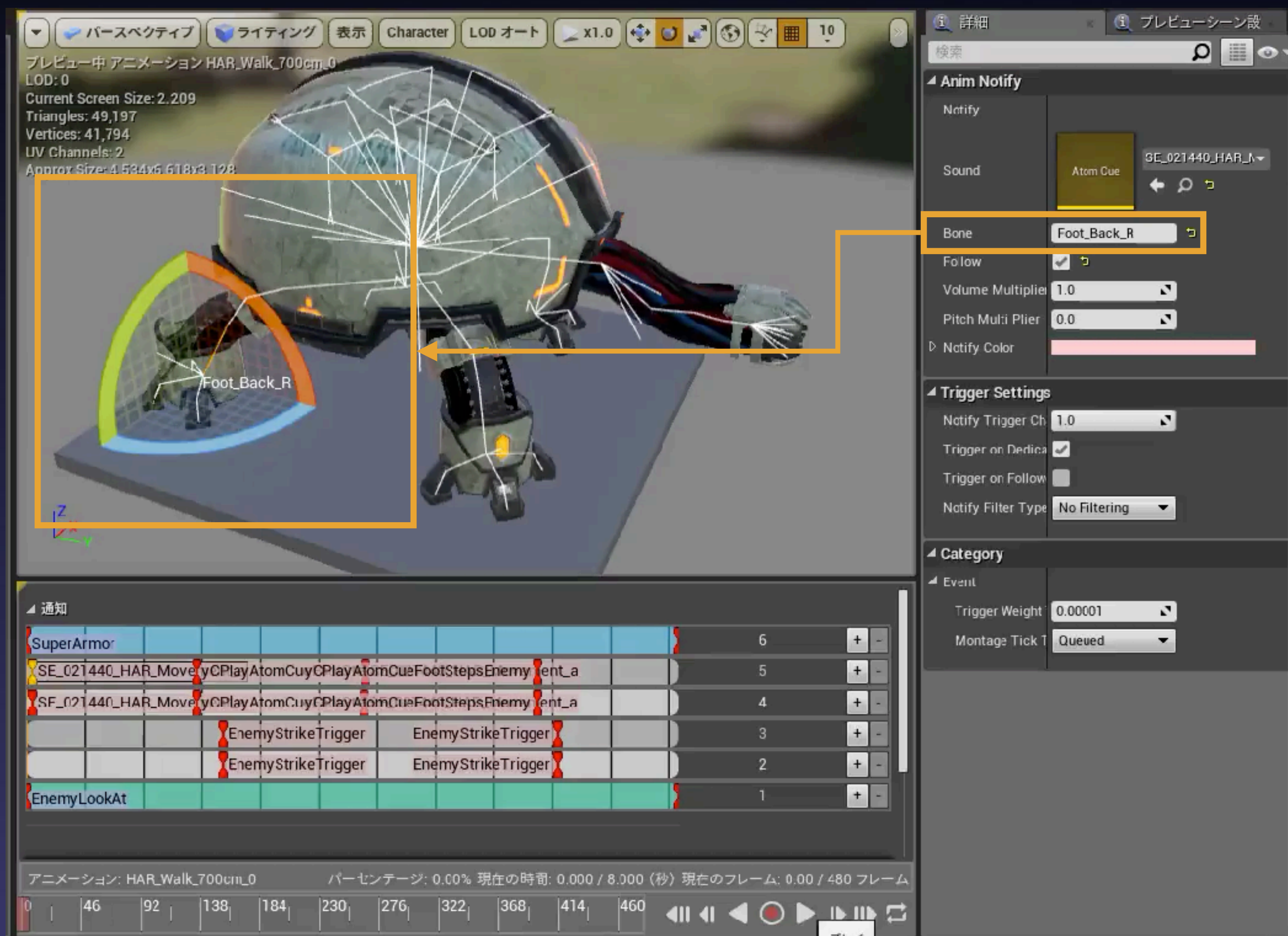
### 3-4-1 アニメーション1： ボーン、フォローの設定



※自分より大きい敵が多いので、ボーンへのアタッチやフォローは重要になってきます。

©D3 PUBLISHER ©Yuke's ©Sound ams inc.

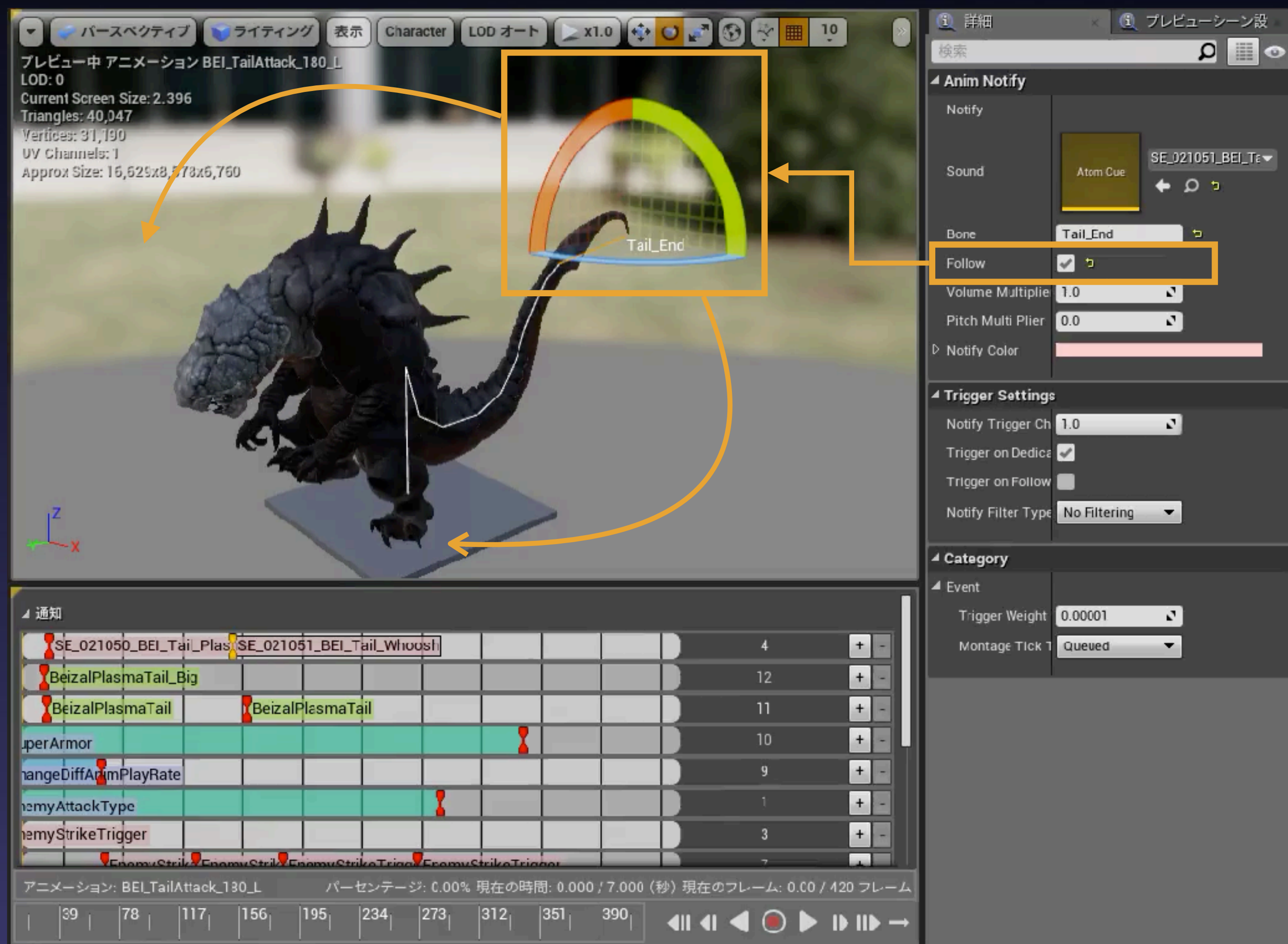
# ボーン位置で音源を再生



各足毎に音源が発生するように設定しています。

こうすることで股下に入り込んだときに周囲から足音がするようになります。

# フォローで音源位置を追従



音源位置が発生源から大きく動くものは、フォローを入れることにより、ボーンに追従されます。

尻尾の先でエフェクトが発生するものなど、体より大きく動く場合に有効です。

この怪獣の場合はプレイヤーより数倍も大きいので、尻尾の動きでも画面を大きく横切るほどに動きます。

※フォローは通常再生より負荷が上がるので、使いすぎない様に注意。

エフェクトなど範囲で発生するものについては距離減衰の最小値をキャラの大きさに合わせて中心に設定するなどしてます。

# アニメーション遷移時にCueを停止させる

スタート

ループ

エンド



AtomPlayGateを使うと、アニメーションが切り替わる際にCueが停止してくれますので「スタート、ループ、エンド」の様に分割やループするアニメーションへ有効に利用できます。

滑り出しの音を再生する。

ループアニメに遷移する際に停止される。

ループアニメーションはインゲーム中尺が不定になる。

ループするCueを用意して再生。

エンドアニメに遷移する際に停止される。





The logo for Sound AMS, featuring the letters 'AMS' in a stylized font inside a blue speech bubble shape.

# 3-5. スプリット画面

# 複数カメラの近い方を「距離、定位」に反映



ADX2が良い感じに自動で対応してるので、設定は不要でした。



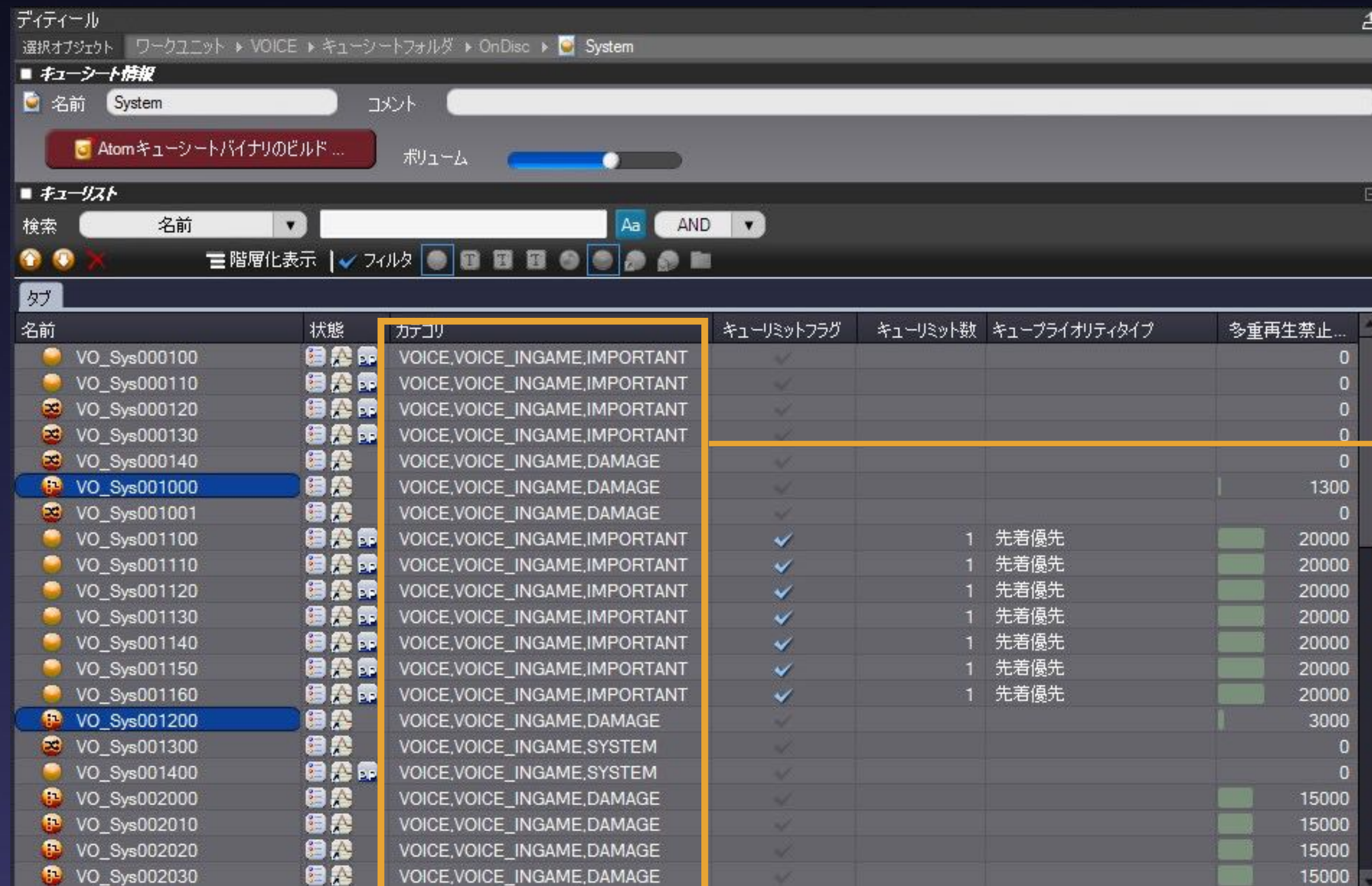
# 4. Voiceの実装

# 音声の優先度に合わせて、音量をコントロール

プレイ中様々な種類の音声再生されます。

情報、進行、雰囲気出しなどです。

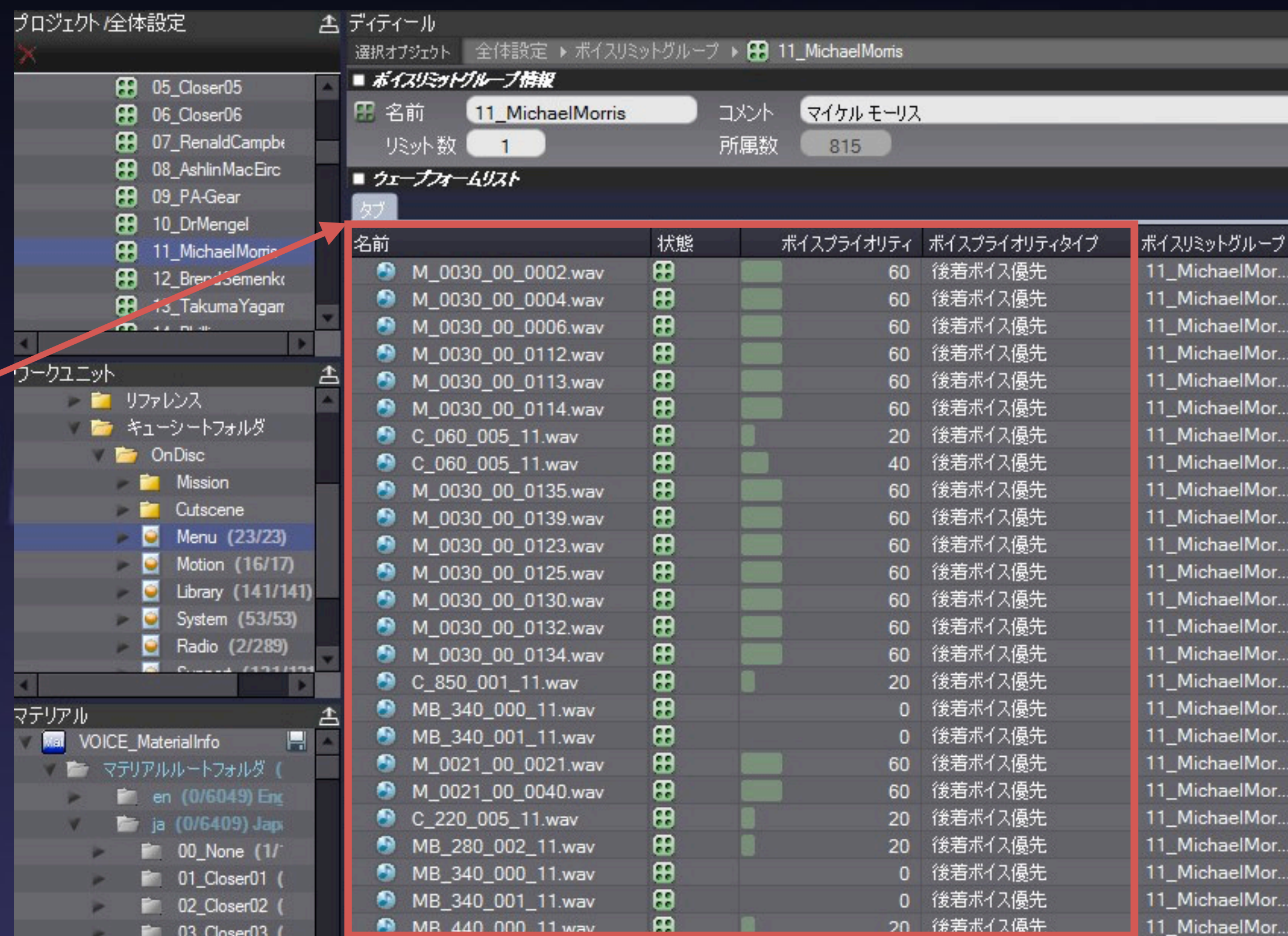
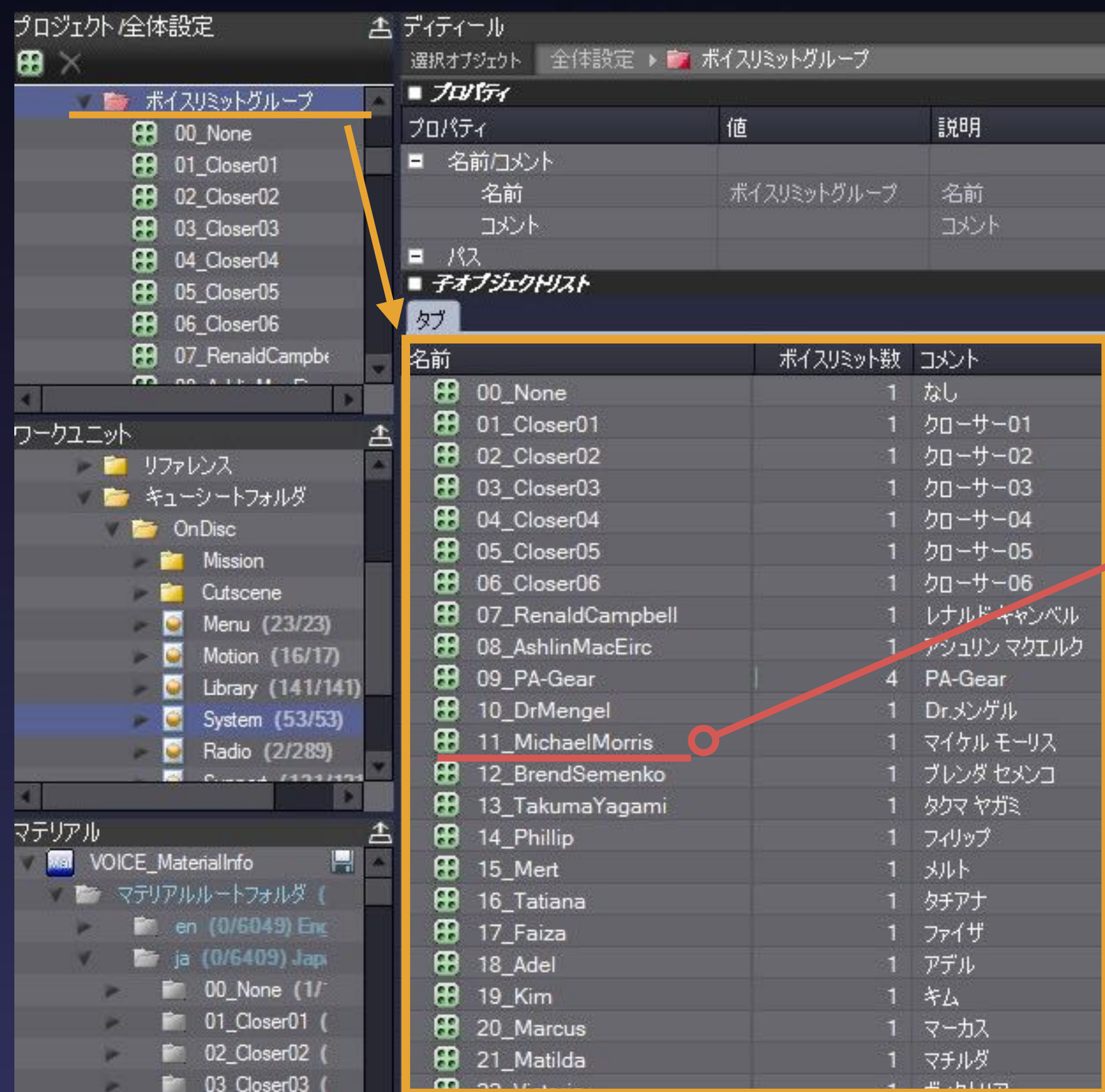
複数の音声再生されると、認識しづらくなりますので、先に紹介したREACTのダッキング機能を使い重要な音声以外を下げる事により実現しました。



カテゴリ	プライオリティ	影響先	例
System	80	Story, Important, Crowd, Damage	「残弾数ゼロ、リロードしてください」
Story	60	Important, Crowd, Damage	「本作戦の目的は、巢の破壊です」
Important	40	Crowd, Damage	「親を倒さないと、増えるぞ」
Crowd	20	Damage	「敵を蹴散らせっ」
Damage	0	-	「うわあー、助けてくれーっ」

※計10個用意しました。

# ボイスリミットグループを使い同一人物の再生を抑制



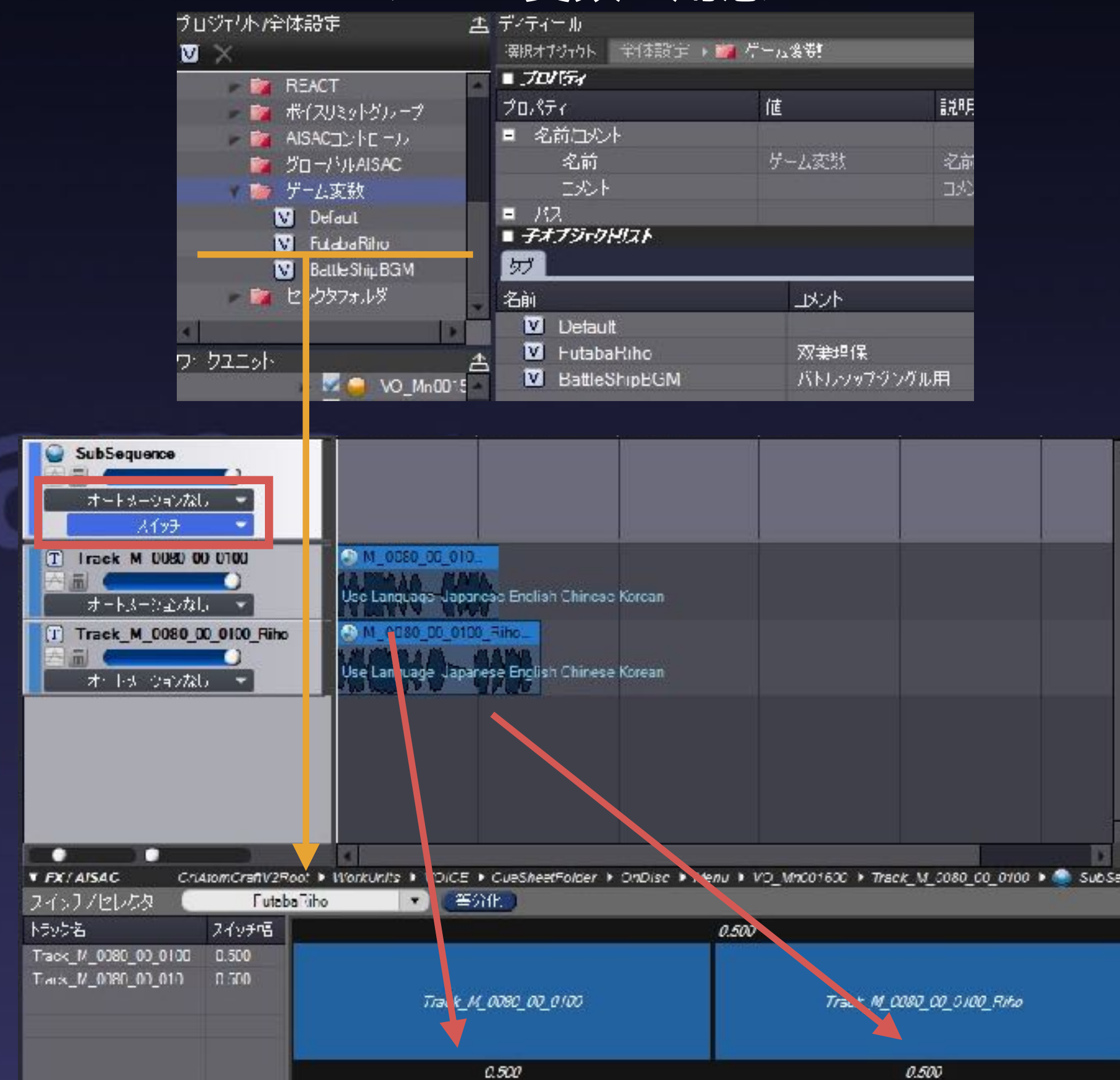
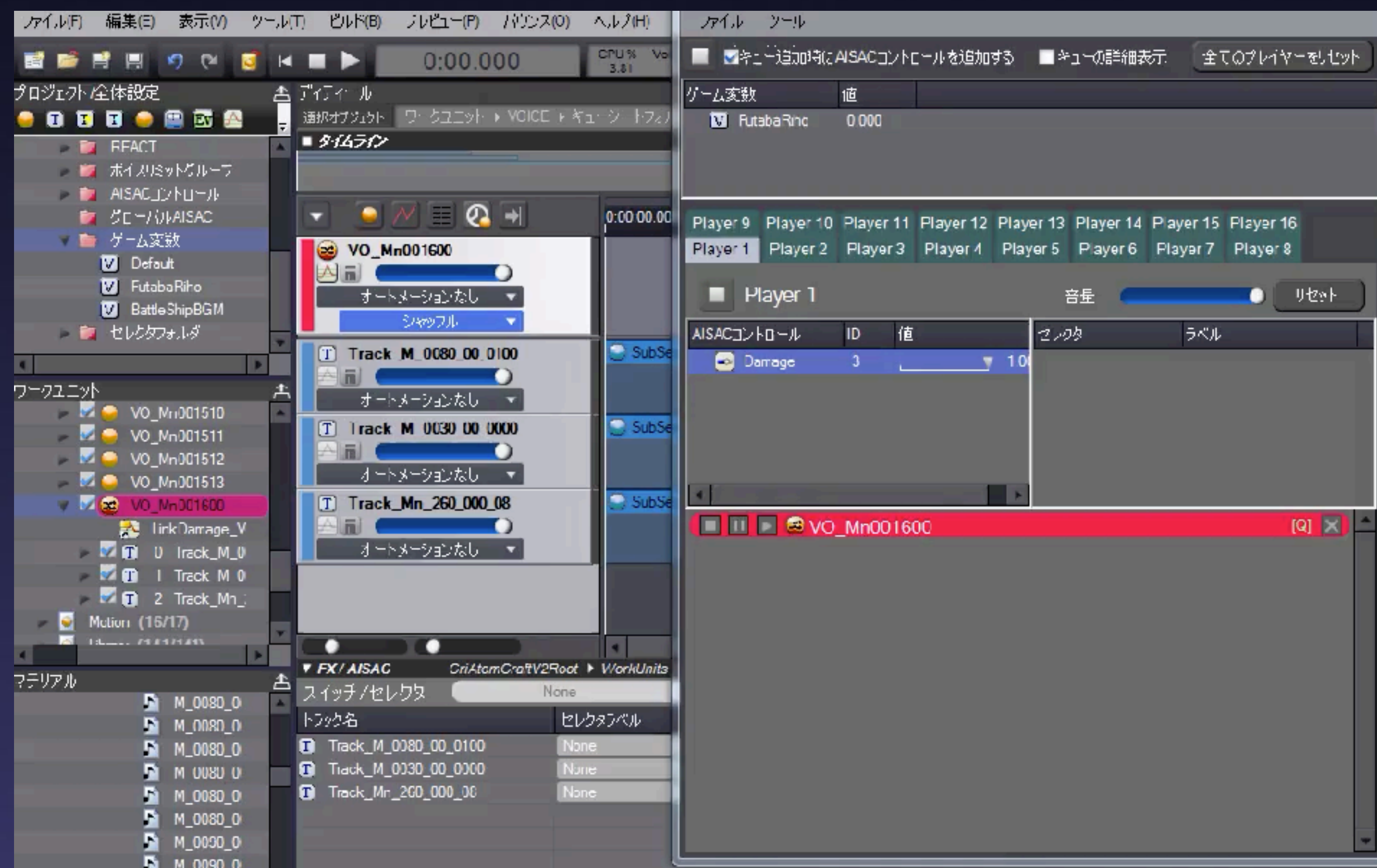
各登場人物のボイスリミットグループを用意してリミット数を1に設定。

同一人物の音声ウェーブフォームに対してボイスリミットグループを設定し、適宜プライオリティ、プライオリティタイプを設定することにより、多重再生されなくなる。

# ゲーム変数を使用してオペレーター音声の切り替える

プログラムからはゲーム変数を 0 or 1 に切り替えるだけで各Cueで使われるオペレーター音声を切り替える事が出来る。

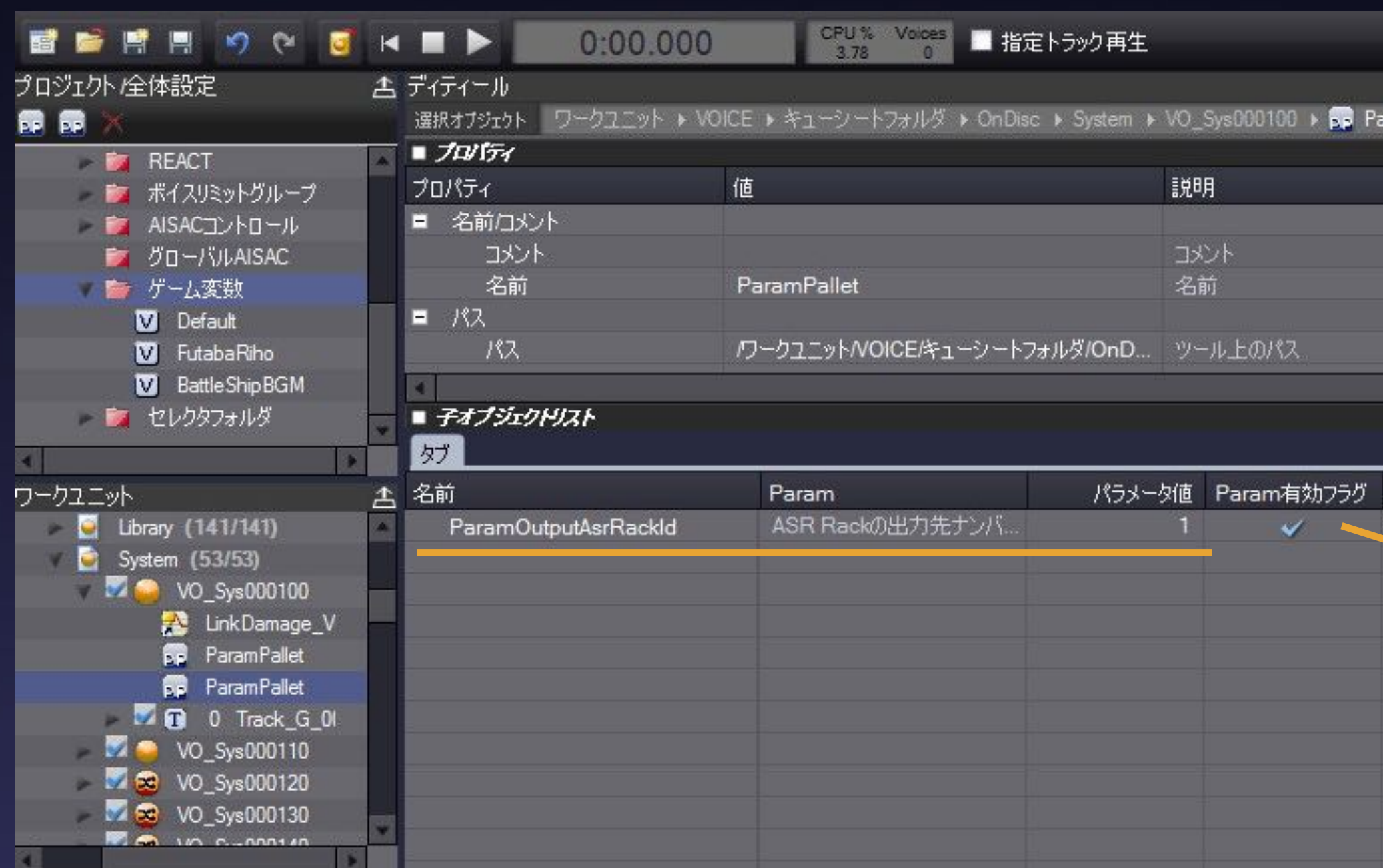
ゲーム変数を用意



※この例だと、1段階構造を増やすことによりシャッフル再生されるCueが切り替わる様になっている

Cue(SubSequence)のタイプをスイッチにしてゲーム変数の値によって再生されるトラックが切り替わる様に設定

# パッドスピーカーから出力したいCueにASR Rackを設定

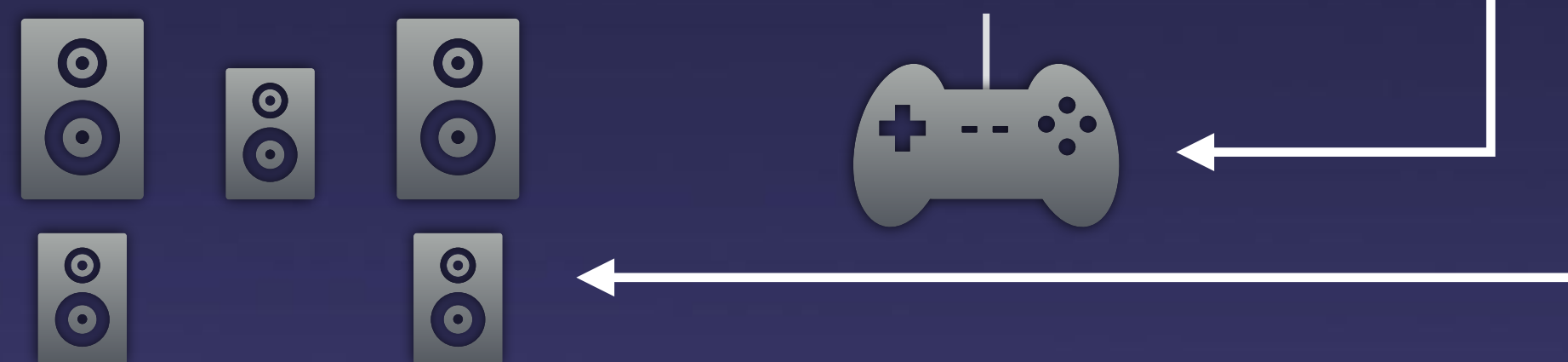


ADX2の機能でCueにパラメータを別途持たせることができます。

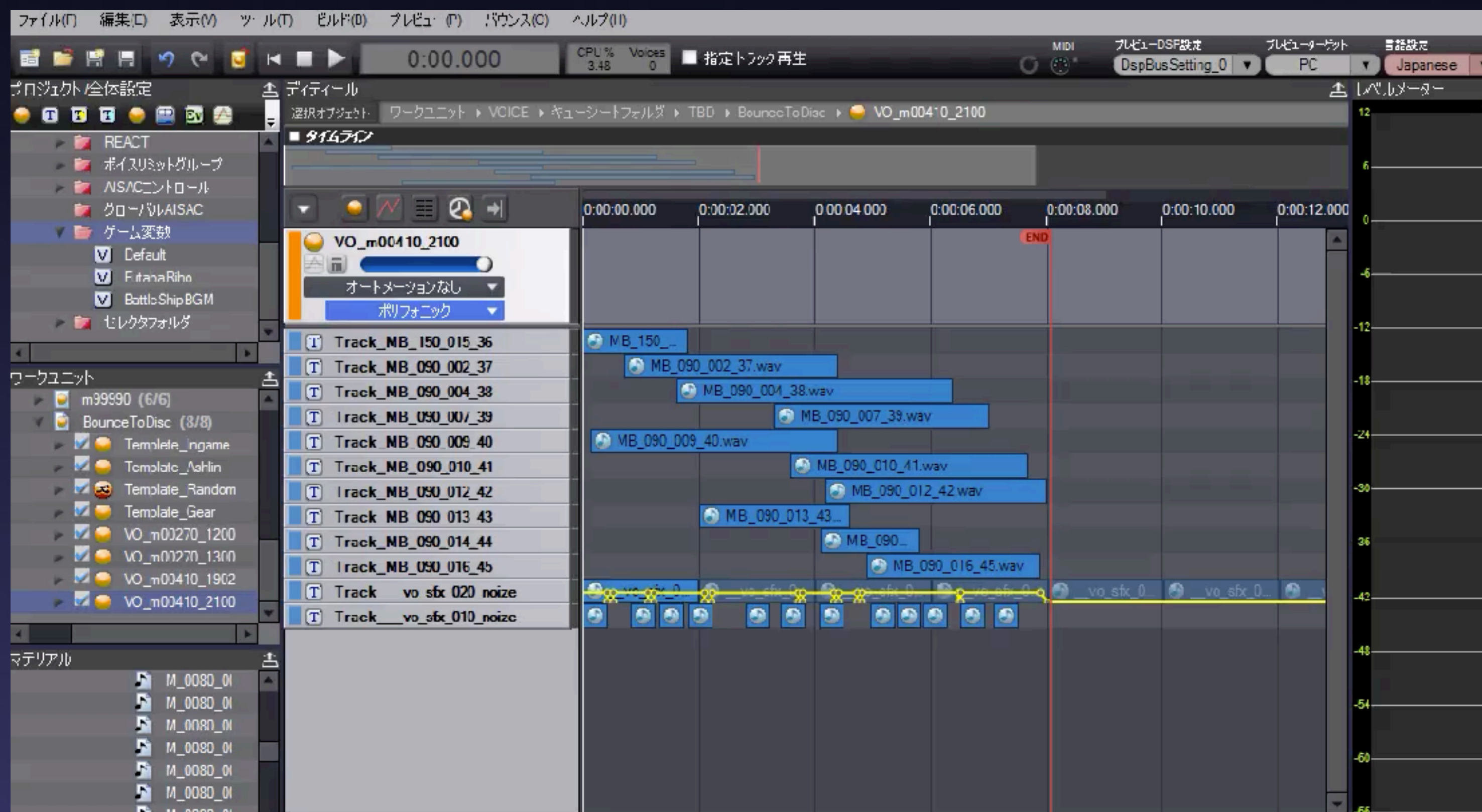
ParamPallet オブジェクトを適宜追加し出力先のIDを指定します。

出力先	パラメータ
モニタースピーカー	0
パッドスピーカー	1

本作の場合は、モニタースピーカー (テレビ等) とパッドスピーカー両方から出力するデザインでしたので、パラメーターパレットを2つ追加し、設定しました。



# Atom Craft を DAWの様に使ってWAV素材を作る



Atom CraftにはCueのバウンス (WAV書き出し)機能と言語の切り替え再生機能があります。

多言語をミックスした素材を用意する際、DAWを使うと個々に並べる必要がある為面倒です。

Atom CraftをDAWの様に使うことにより、多言語の素材制作にかかる時間を短縮しました。

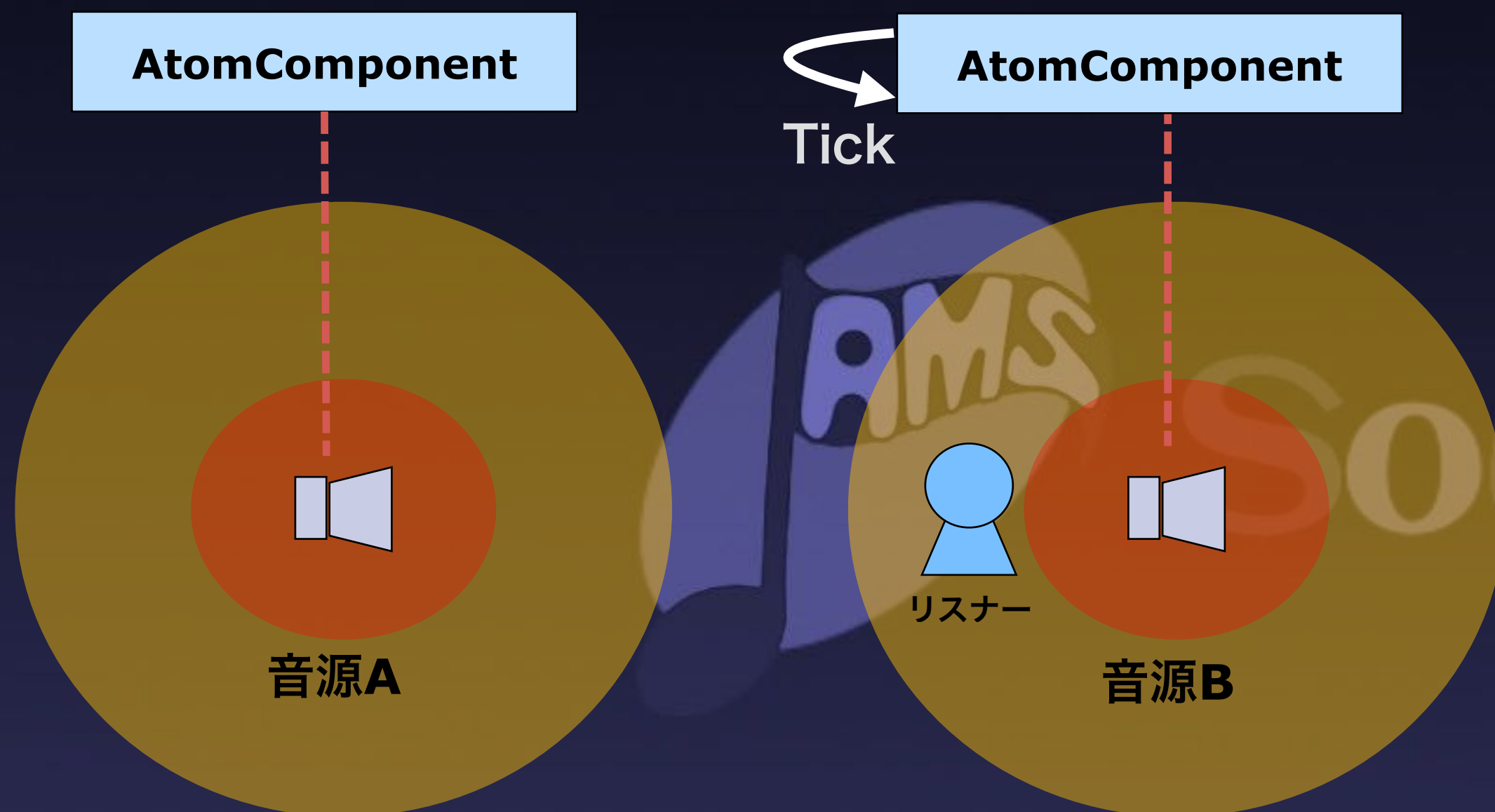




# 5. 処理負荷対策

# 可聴距離よりも遠くで発音された音はTick処理しない

→ この負荷削減機能を「距離カリング機能」という



3D音源Aの可聴範囲の外にリスナーがいる。よってAは再生されても音量ゼロ。

ユーザに聞こえないサウンドなので、余計な処理負荷は省きたい。

距離カリングを使えば発音トリガーが発火しても Tick 処理は走らない。

可聴範囲内の音源Bだけが Tick 処理される。

ゲーム状況に応じてプログラムから発音をトリガーする際の注意  
主にUAtomStatics::SpawnSound 系APIを使う。これ  
を呼ぶと、AtomComponent が生成される。  
AtomComponent はADX音声の再生制御やライフサイクル  
を管理する、CRI定義のUE4オブジェクト。大量に使うと毎  
フレームの Tick 処理で GameThread がブロックされる。

## 距離カリング機能による Tick 処理の負荷削減

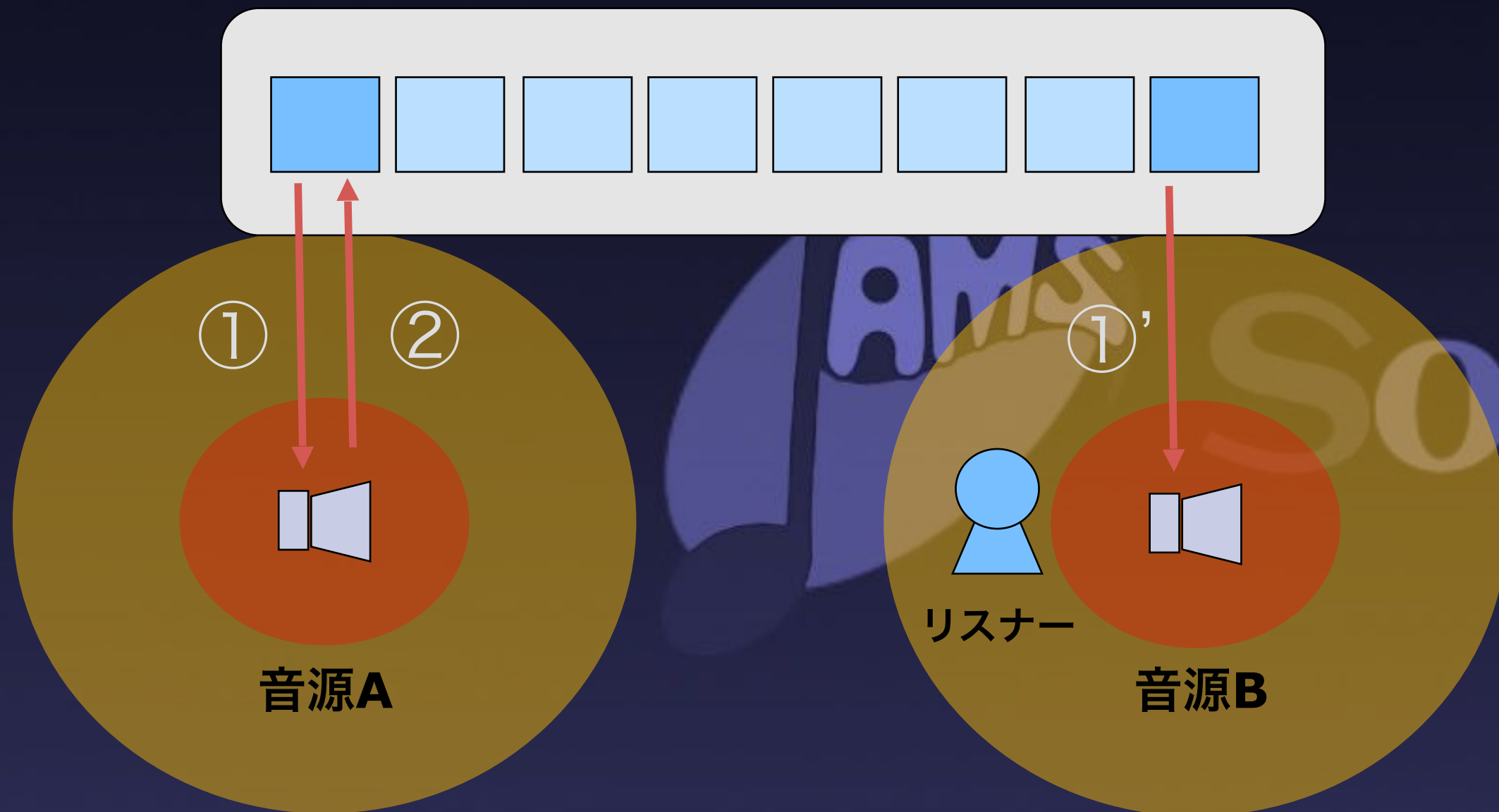
距離カリング機能を有効にすると、リスナーの可聴範囲外で生まれ  
た AtomComponent はTick 処理されない。マップ全域で  
SpawnSoundによる発音が発生しても、リスナー周囲の音だけが処理  
されるので負荷が減る。

※ 距離カリングされたサウンドであっても、AtomComponent 自体  
は生成されるので注意。あくまで Tick 処理を省くだけ

# AtomComponent を予め生成して Pool しておく

→ デフォルトでは無効化されている機能なので要注意

AtomComponentPool



Pool 内の AtomComponent は未参照のとき Tick 処理をしない。

また、Pool 内の AtomComponent は GC されない。GC 負荷を減らす

## 距離カリングだけでは防げないGC(ガベコレ)の負荷

UAtomStatics::SpawnSound系APIを呼ぶとAtomComponentが生成される。再生終了後、AtomComponentはUE4のGC機構の下で破棄される。大量のAtomComponentをGCするとGameThreadの長時間ブロック(ヒッチ)が起きる。

## AtomComponent の使い回しによるヒッチを回避

左図のように AtomComponent の Pool を設けることで負荷削減。

音源A：距離カリング対象である3Dサウンド

① Pool からAtomComponent を取得

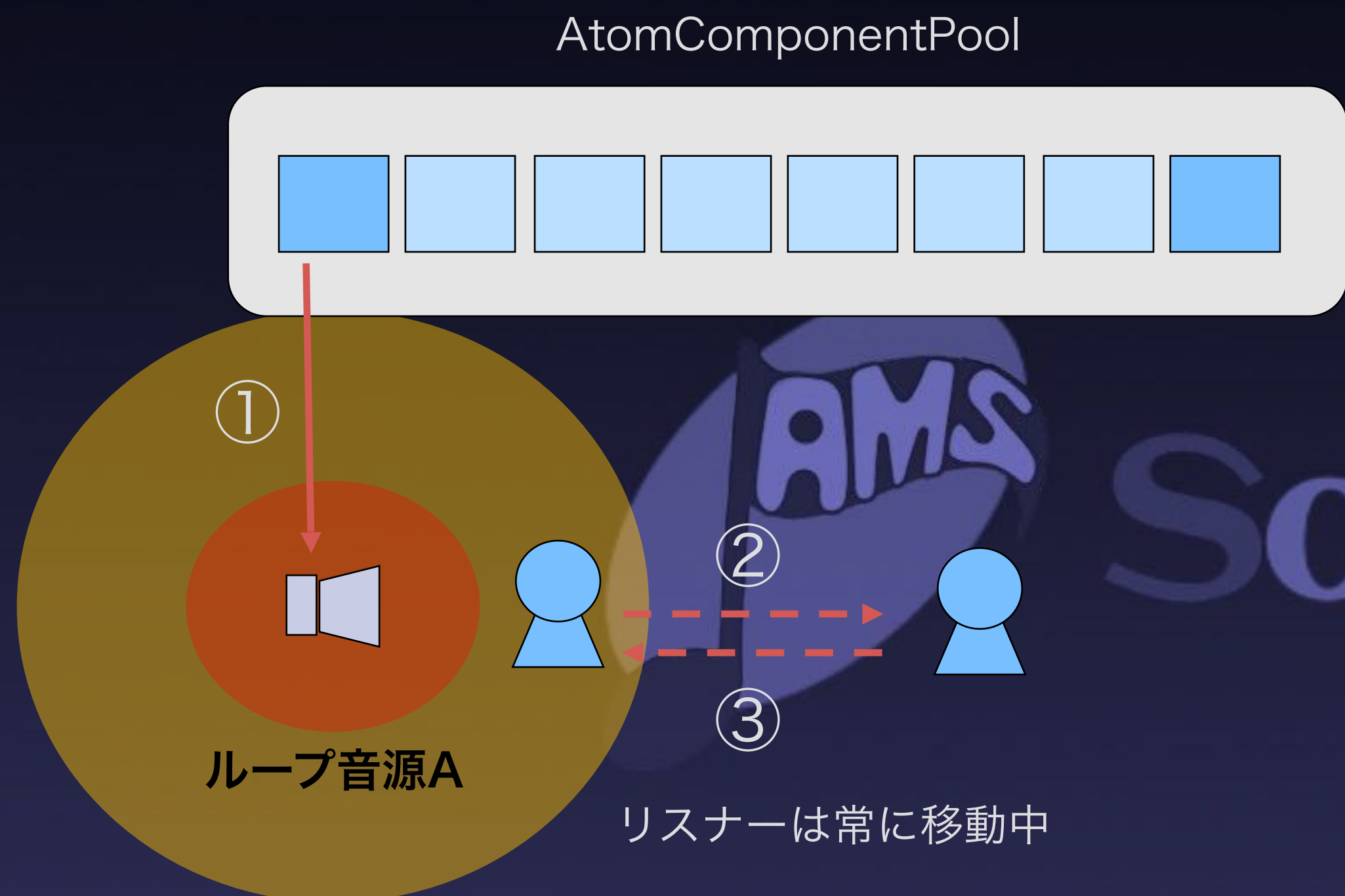
② 可聴範囲外なので Component の処理は不要。すぐ返却する

音源B：距離カリング対象でない3Dサウンド

①' サウンド再生時に Pool から AtomComponent を取得

※ サウンドの再生が終了したら Pool に返却する

# 「再生中だが聞こえない音」を仮想ボイス化する



尺の短い単発再生サウンドならば距離カリング可能だが、ループ再生されるサウンドには使えない。

代わりに仮想ボイス化で少しでも負荷を減らす。

## UE4外部で生じるサウンド処理負荷

CRI ADX2 は独自のオーディオスレッドでサウンド処理を行う。音声データのデコードやストリーミング、ミキシング、エフェクト処理など。

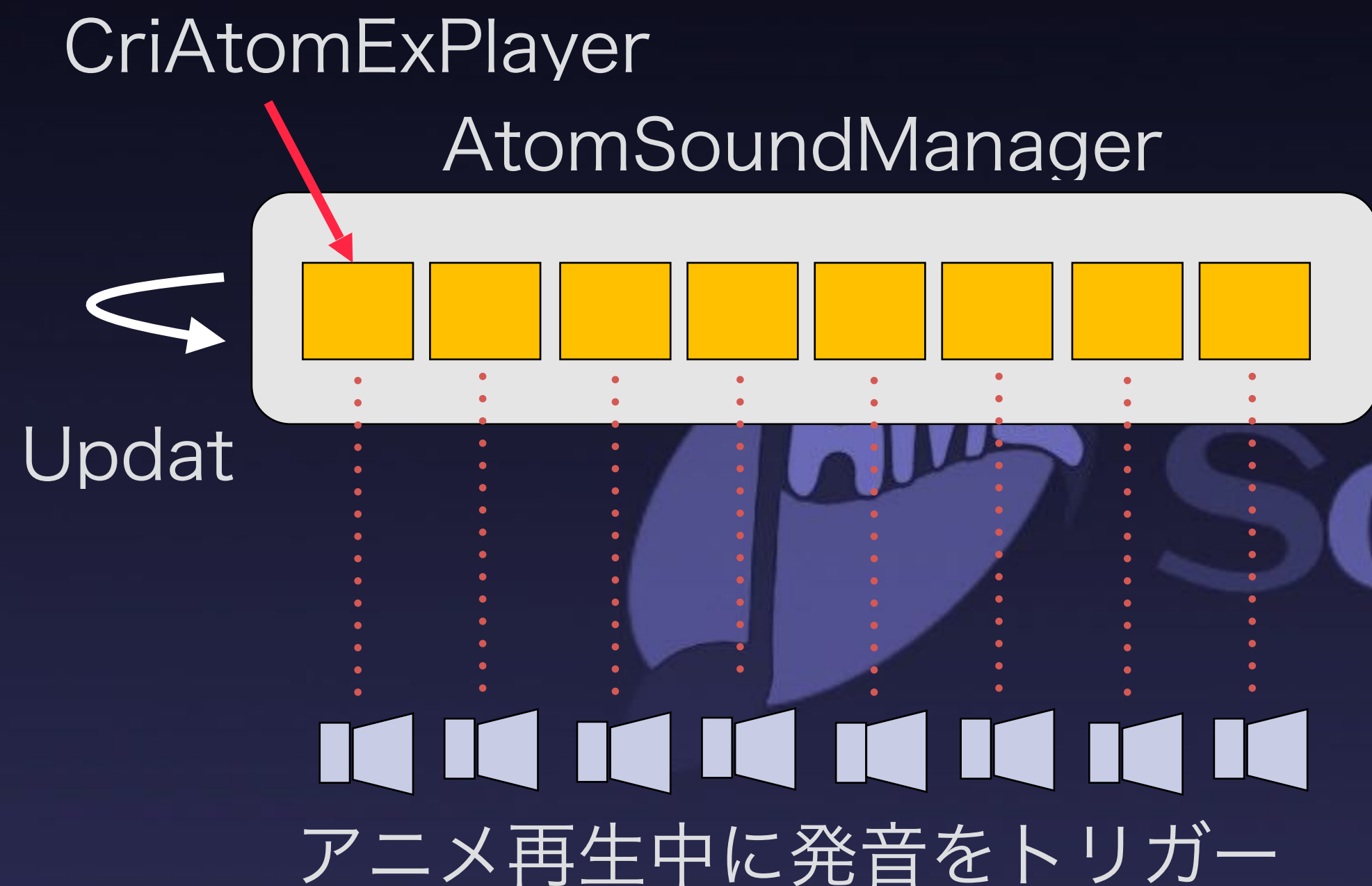
CRI ADX2 における音声再生用の発音リソースを「ボイス」という。ボイスを持たず、仮想の再生処理をおこなうための「仮想ボイス」もある。

仮想ボイスは再生パラメータと時刻のみを管理する、実体の無いボイスのこと。通常ボイスに比べて内部処理が一部省略されるため、負荷が低い。「音量がゼロになったサウンドは仮想ボイスに切り替える」という設定が可能。

### 左図：ループ音源 A の内部リソース (ボイス) 状態の切り替え例

- ① 音源A再生開始。 Pool からAtomComponent を取得
- ② リスナーが可聴範囲外に出たので、音源A の音量はゼロになる  
→ 再生中のまま、自動的に**仮想ボイス化**される
- ③ リスナーが可聴範囲内に入ったので、音源A の音量がプラスになる  
→ 再生中のまま、自動的に**仮想ボイス化**が解除される

# アニメ上の音はAtomComponentを使わない



UAtomStatics::SpawnSound 系API で発音したサウンドと異なり、プログラム側でコントロールできない。

その代わりに、あくまでアニメ上の音声として割り切った管理を行うことで無駄を省いている。

## 大量のキャラクターがマップ上で動く場合の注意

キャラクターのアニメーション内でADXサウンドの発音をトリガー可能。大量のキャラがマップ上で動く場合は処理負荷が気になる。

過去のCRIWARE UE4 Plugin バージョンではアニメ上の音も AtomComponent を使っていたが、代わりに CriAtomExPlayerHn を直に保持することで GC 等の負荷を回避できるようにした。

CriAtomExPlayerHn は CRI ADX2 内部の独自オブジェクト。音声の再生制御インタフェースや状態を備えている。UE4 の外側で生成・管理される。

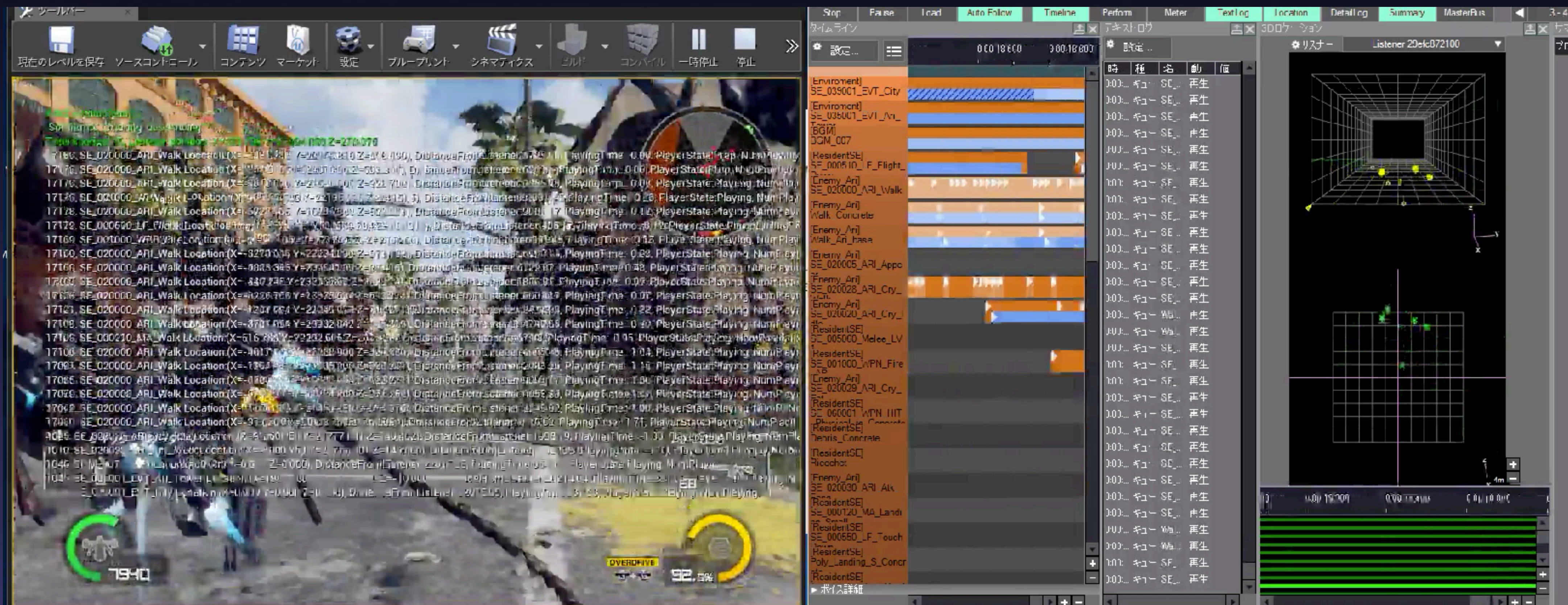
## 左図：アニメ上の音の実体を管理する機構“AtomSoundManager”

CriAtomExPlayerHn も定期更新が必要。普通は AtomComponent の Tick で行うが、ここでは AtomSoundManager という管理オブジェクトがすべてのアニメ音声を監視。各フレーム処理の先頭で Update 処理を各 CriAtomExPlayerHn に対して実行する。再生終了状態になったら削除する。

# 用途に合わせてプロファイリング

何が発音されているなど、簡単に確認したい

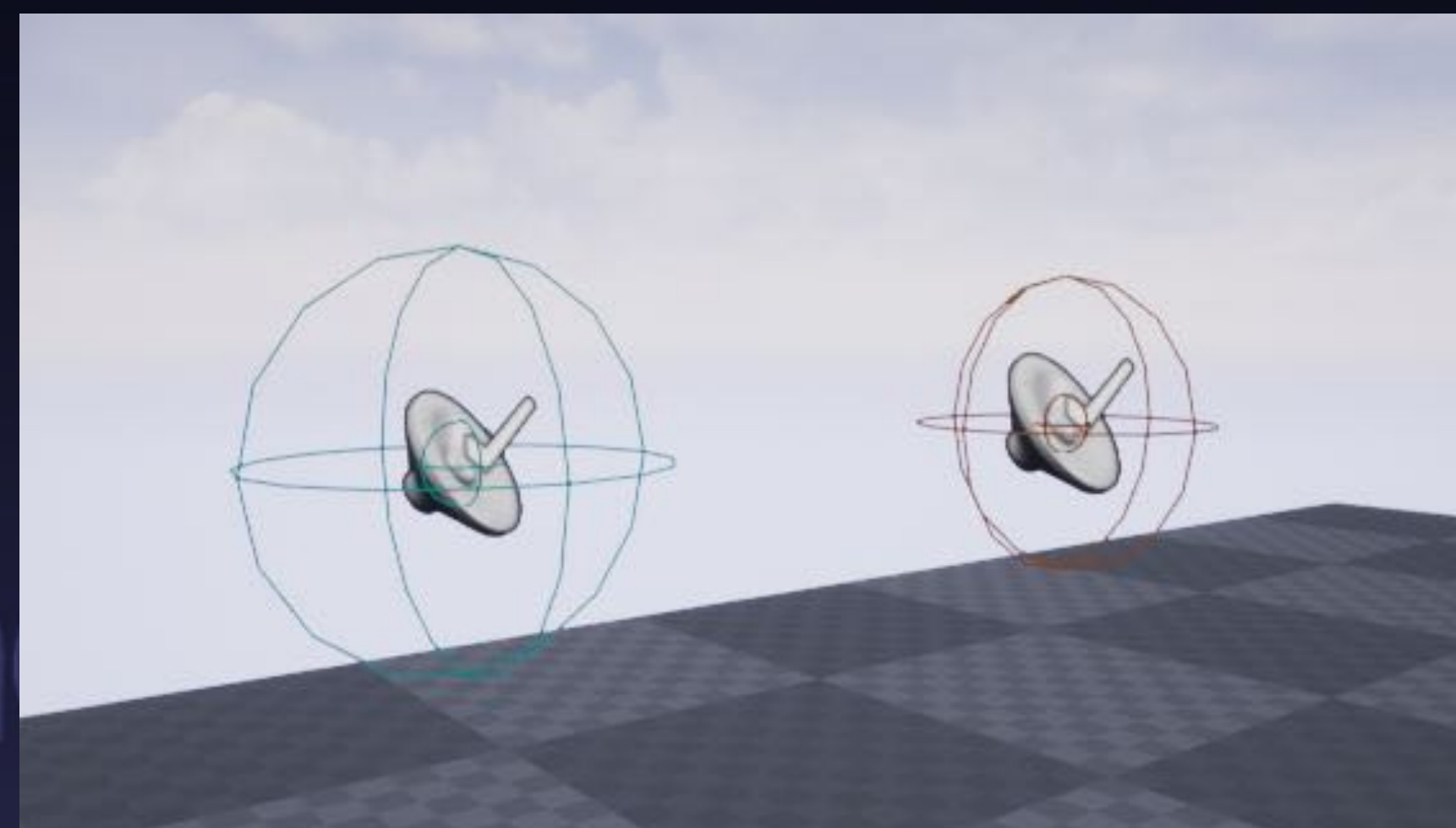
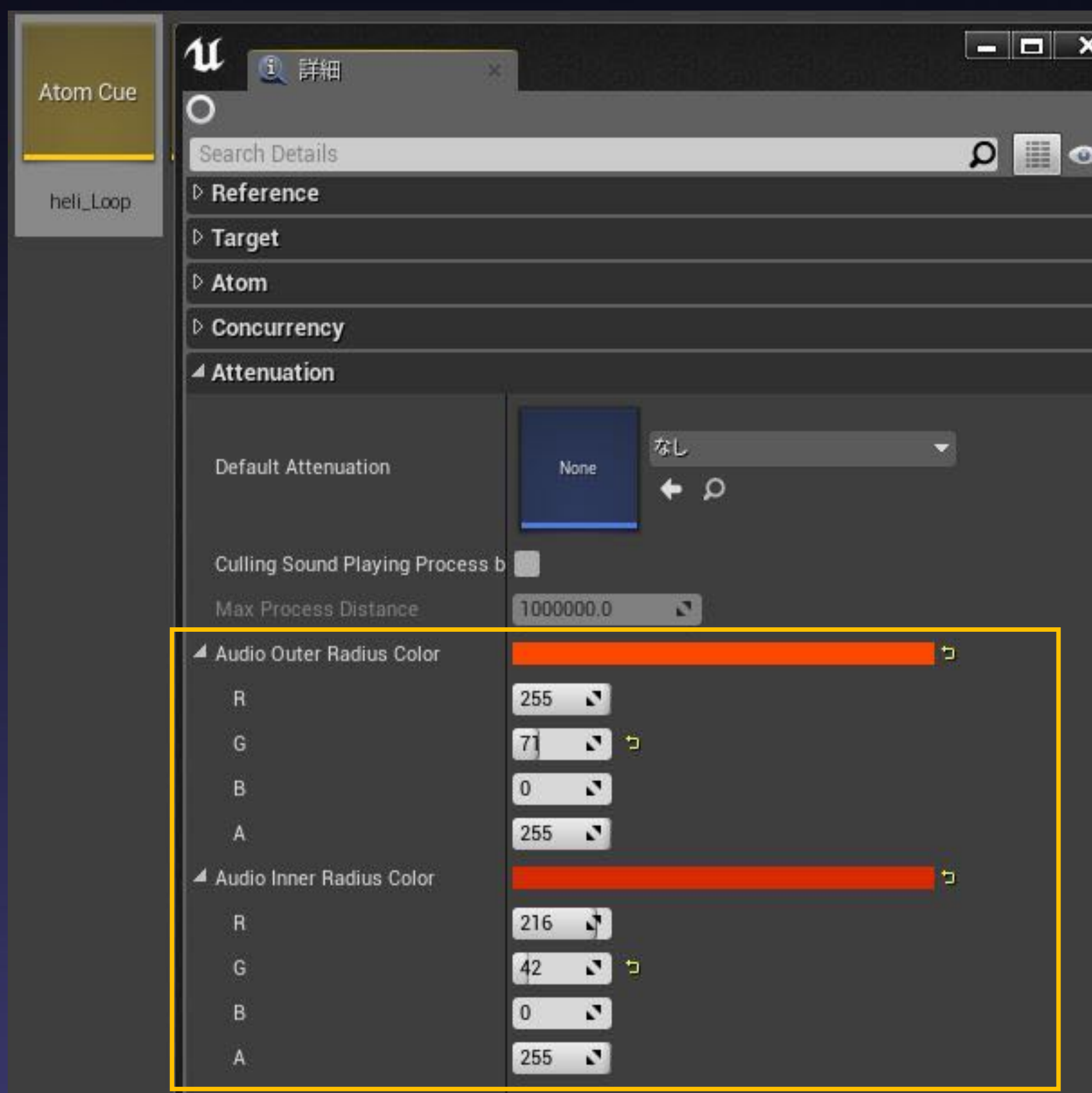
音声の制御状態など、細かく確認したい



※UE4上でデバッグコマンド “cri.ShowProfileInfo 1”

※Atom Craftのプロファイラを接続

# 大量の3Dサウンドを可視化して見分ける



3Dサウンドの距離減衰範囲を可視化

内部範囲、外部範囲について、それぞれ可視化ワイヤー表示のカラーリングを設定できる。

ご静聴ありがとうございました



Sound AMS inc.



# 補足

AISAC, REACT等ツールの専門用語がございましたが、CRI様のホームページに超分かりやすい説明があるので、URL記載いたします。

<https://cirware.info/adx2-tutorial/>

<https://game.criware.jp/learn/tutorial/ue4/>