

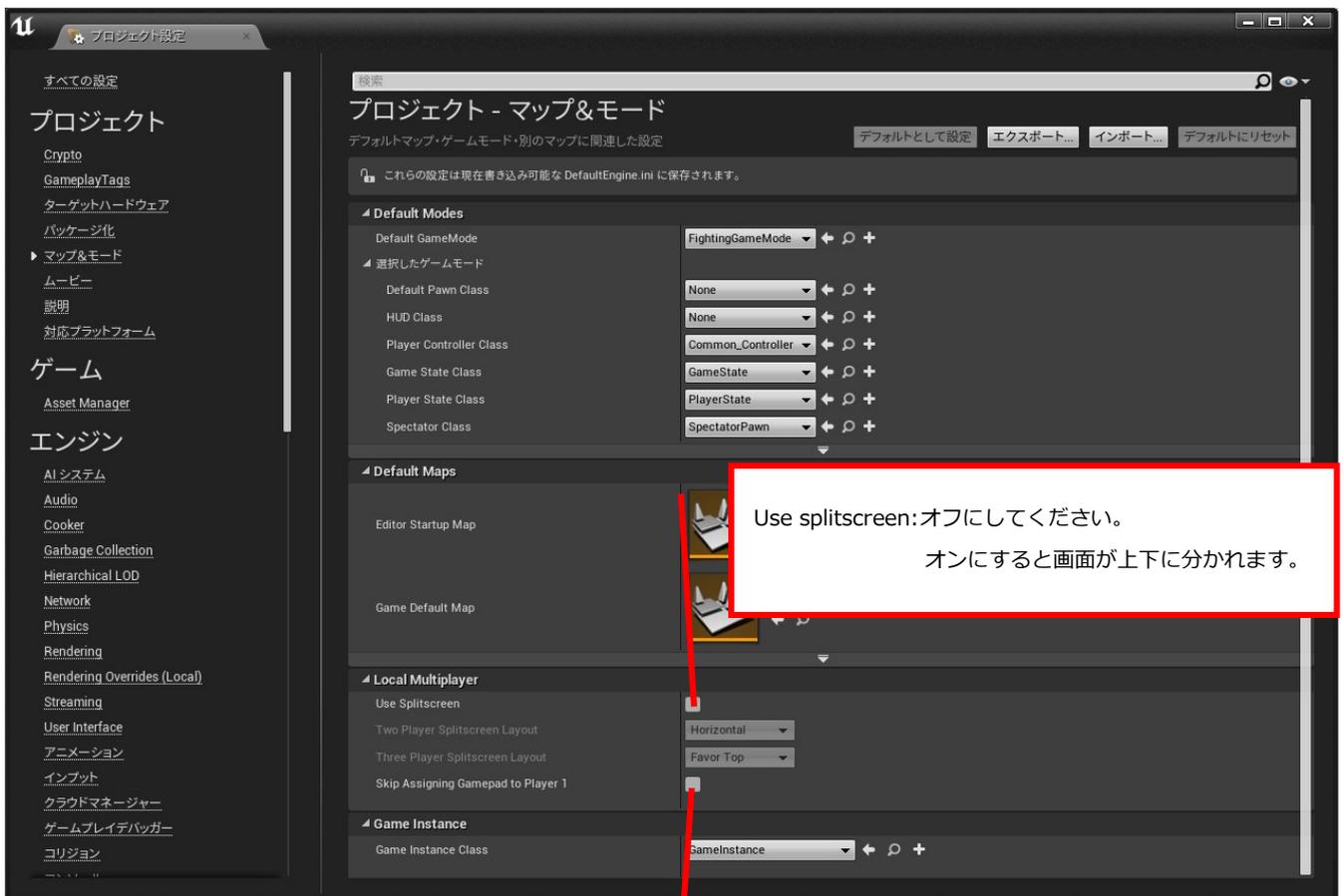
Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

目次

1. プロジェクトの設定について
2. ゲーム中の操作方法について
3. フォルダとファイルの構成について
4. キャラクターの追加と編集方法について
5. ステージの追加方法について
6. エフェクト・サウンドの編集方法について
7. デバッグ時の注意点について

1. プロジェクト設定について (Ver.3.5 の変更点は青字で記載)

プロジェクト設定にあるマップ&モード内のローカルマルチプレイヤーの設定を下の画像を見て確認してください。



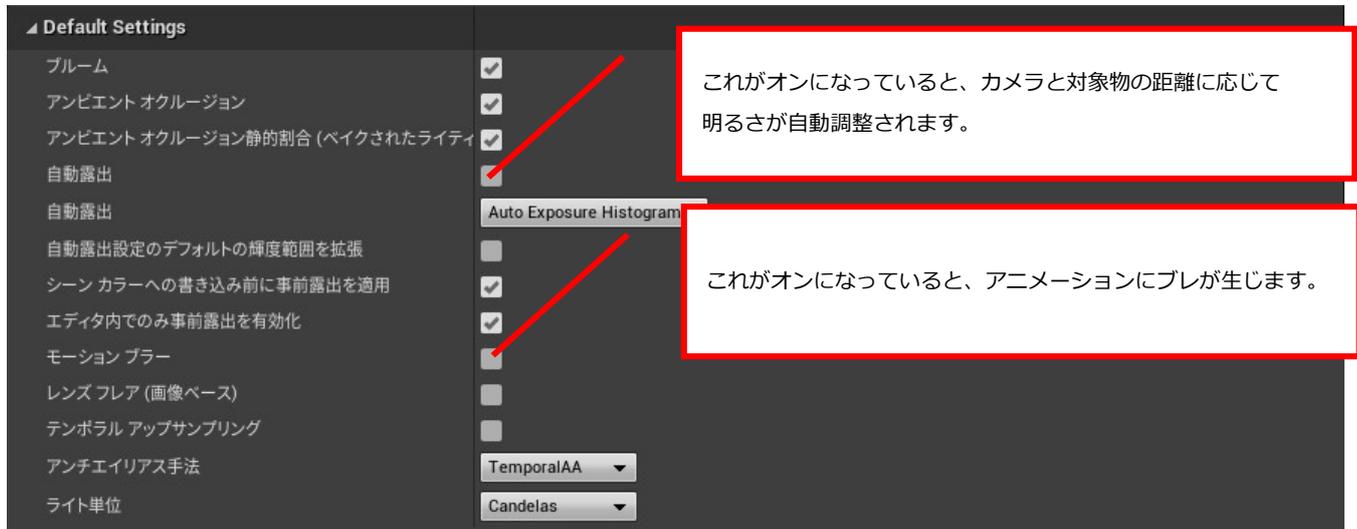
Skip Assign GamePad to Player1:

キーボードと1つ目のコントローラを Player1 に割り当てる場合はオフ、キーボードと1つ目のコントローラを Player1 と Player2 に割り当てある場合は、オンにします。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

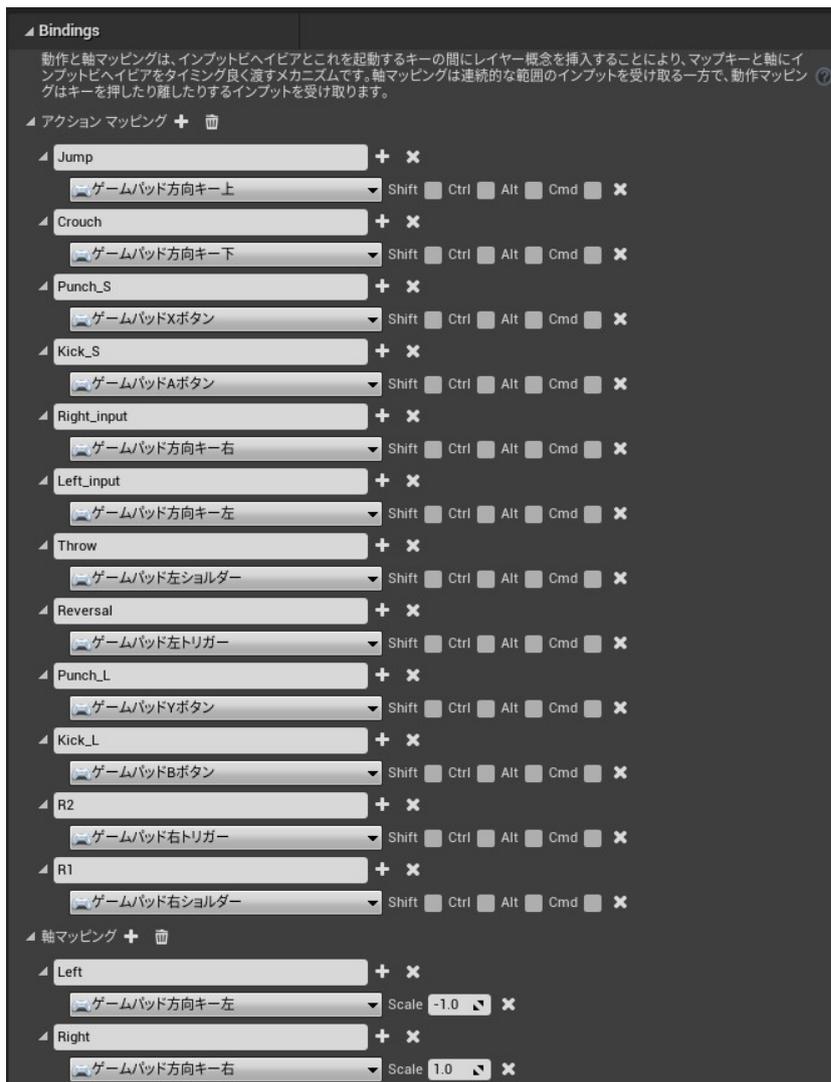
次にエンジン→レンダリングの中にある自動露光をオフにします。

更にモーショnbrラーをオフにします。



次に、エンジン→インプットの中にあるコントローラの入力設定を以下の画面通りに設定します。

マッピングの名前を間違えると正しく動作しませんので注意してください。



Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

最後にレベルエディタ→プレイにて実行時のウィンドウサイズを指定します。

縦横の比が 16:9 になるサイズを入力します。

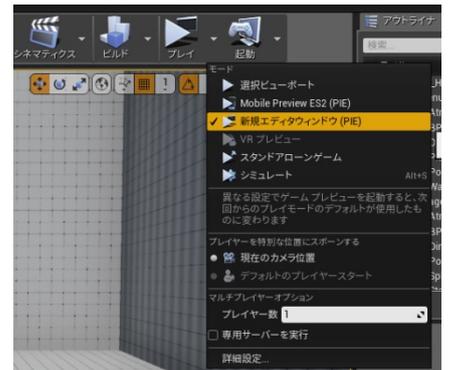
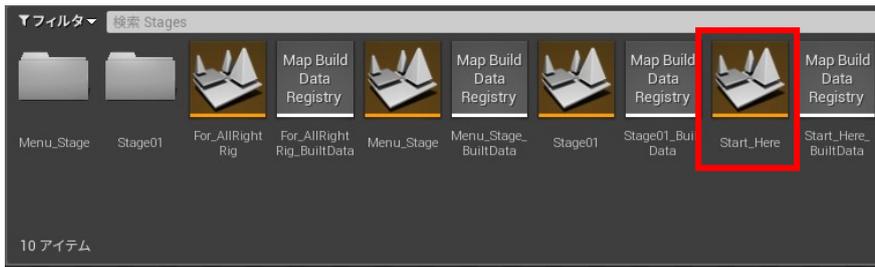
(例:1280*720 / 1600*900 / 1920*1080)



これでプロジェクト設定は完了です。

プロジェクト内のコンテンツにある Stage フォルダの Start_Here ファイルを開いていることを確認したら、実行を試してください。

(新規エディタウィンドウでサイズが反映されたか確認してください。)



2. ゲーム中の操作方法について

操作方法の説明はプレイステーションのコントローラのボタンで説明します。

■タイトル画面

○ボタンで次へ進みます。

■ファイティングモードセレクト

1on1 か 3on3 か **オプションモード** を方向キーで選択し、○ボタンで決定です。×ボタンで戻ります。

■オプションモード

ボタンとサウンドボリュームの設定ができます。

ボタン設定は○ボタンを押し続けると設定画面となります。設定完了メニューを選択後に、オプションボタンを押すと前の画面に戻ります。

サウンドボリュームは、バトル中の音楽か効果音を選択して、左右キーで音量を設定します。

■モードセレクト

VS CPU (1on1 のみ) か VS Player を選択します。×ボタンでファイティングモードへ戻ります。

■ステージセレクト

方向キーで選択し、○ボタンで決定です。×ボタンでモードセレクトへ戻ります。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

■キャラクターセレクト

方向キーで選択し、○ボタンで決定です。

3on3の場合は、3人選択しますが、同じキャラクターを複数回選択することはできません。

×ボタンで選択をキャンセルできます。

■バトル中の操作方法

方向キー：左右で前後移動、上はジャンプ、下はしゃがみ

□ボタン：弱パンチ

△ボタン：強パンチ

×ボタン：弱キック

○ボタン：強キック

L1 ボタン：投げと投げ抜け

L2 ボタン：ガード中の反撃（SPバー1本以上必要）

R1 ボタン：2番目のサブキャラクターによるアシスト攻撃（3on3のみ）

R1 ボタン+方向キー前：2番目のキャラクターとチェンジ（3on3のみ）

R2 ボタン：スウェー（1on1のみ）、3番目のサブキャラクターによるアシスト攻撃（3on3のみ）

R2 ボタン+方向キー前：3番目のキャラクターとチェンジ（3on3のみ）

□必殺技（キャラクター右向き時）

236+パンチボタン：Fireball（飛び道具）

214+パンチボタン：Flash Straight（空中可）

623+パンチボタン：Flash Straight（対空）

236236+パンチボタン：Iceball（SPバー満タン時）

□チェーンコンボ（サンプル）

立ち状態で×、□、△ボタンを一定のタイミングで順番に押す。

□エアリアルコンボ（サンプル）

チェーンコンボの△ボタンの攻撃ヒット時に方向キーの上を入力すると、相手を上に打ち上げた後、追尾するので、×、□、△、○ボタンを一定のタイミングで順番に押す。

△や○ボタンの攻撃ヒット時に、キャンセルでFlash Straightの空中版を使用することが可能。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

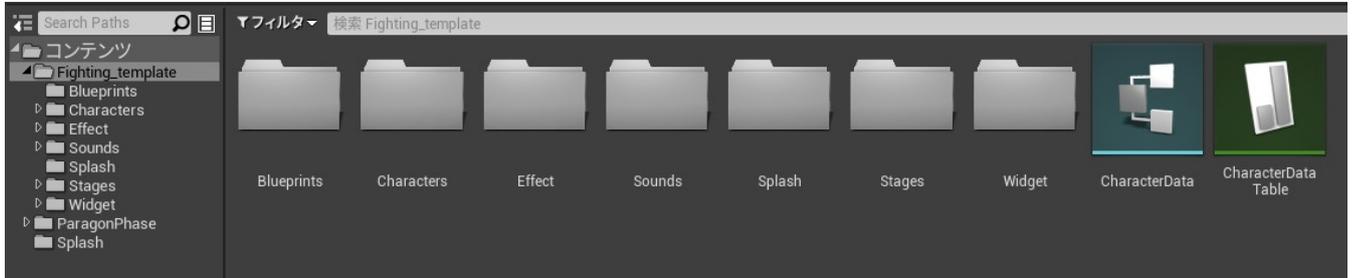
3. フォルダ構成とファイルについて

以下フォルダとファイルの概要を説明します。

■ トップフォルダ

キャラクターデータシートとキャラクターデータテーブルがあります。

キャラクターを編集するときに使用します。



■ Blueprints フォルダ



BP_CommonController : このプロジェクトのメインプログラムファイルです。

プレイヤーの入力を受け取り、キャラクタや各種ゲージの制御を行います。

Player2 の制御は CommonController-1 という形でファイルが生成され、Controller_ID という変数の値により、Player1 と 2 の識別が行われます。

BP_FightingGameMode : このプロジェクトの初期設定を行うファイルです。

基本的に設定を変更する必要はありません。

BP_GlobalCamera : カメラをコントロールしているプログラムファイルです。

BP_Save_setting : ボタンやサウンドボリューム設定の変数を定義するファイルです。

■ Character フォルダ

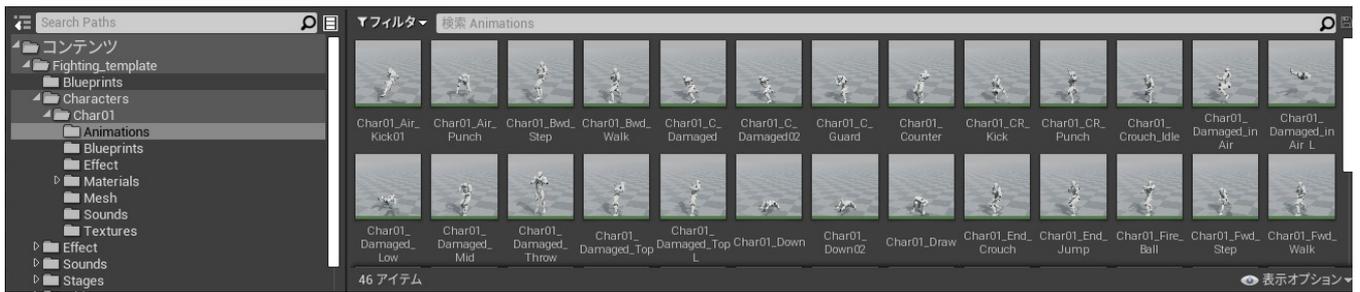
□ Char01 フォルダ

キャラクタ毎の専用ファイルが保存されています。

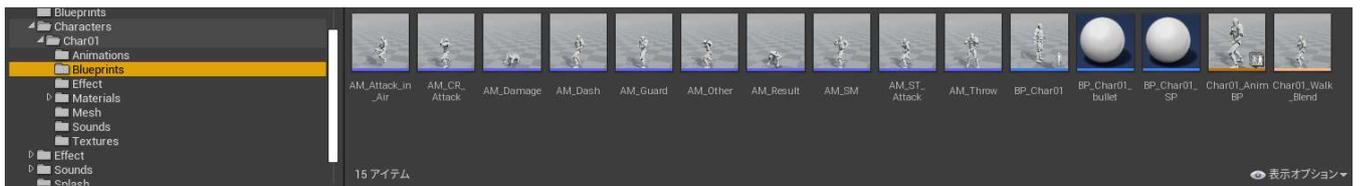
Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

□ Animations

キャラクターのアニメーションが保存されています。



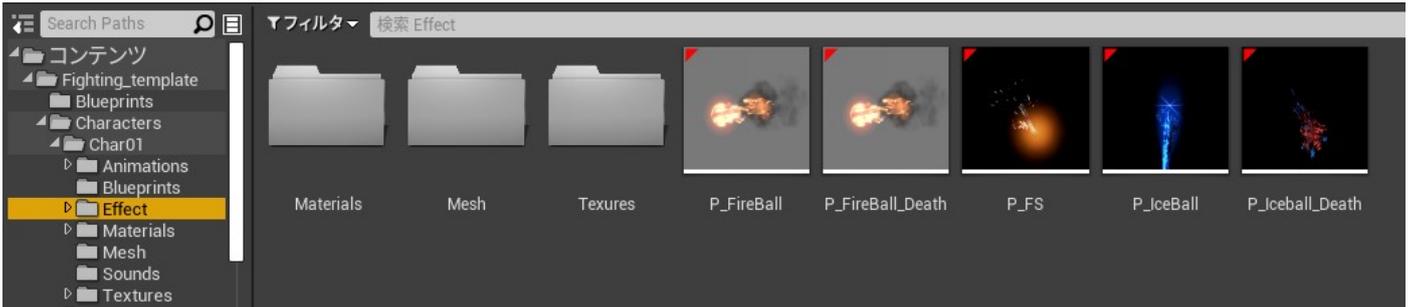
□ Blueprints



- BP_Char01
キャラクターの攻撃・ダメージの判定 BOX の設定や相手の Capsule component とオーバラップした場合の処理を記載しています。
- Char01_AnimBP
アニメーションの再生を制御しています。
- BP_Char01_projectile_SM1
必殺技 1 で飛び道具を設定したときに発生するアクターを制御します。
- BP_Char01_projectile_SM2
必殺技 2 で飛び道具を設定したときに発生するアクターを制御します。
- BP_Char01_projectile_AS
3on3 のアシストで飛び道具を設定したときに発生するアクターを制御します。
- BP_Char01_SP
SP ゲージがフルの時に使用できる必殺技使用時に発生するアクターを制御します。
- その他アニメーションモンタージュ
立ちやし্যাがみなどキャラクターの状態毎にアニメーションモンタージュを設定しています。
このファイル内で、通知を利用して攻撃判定の発生・消失のタイミングを指定しています。
- Char01_Walk_Blend (ブレンドスペース 1D)
待機状態と前進、後退のアニメーションを制御しています。

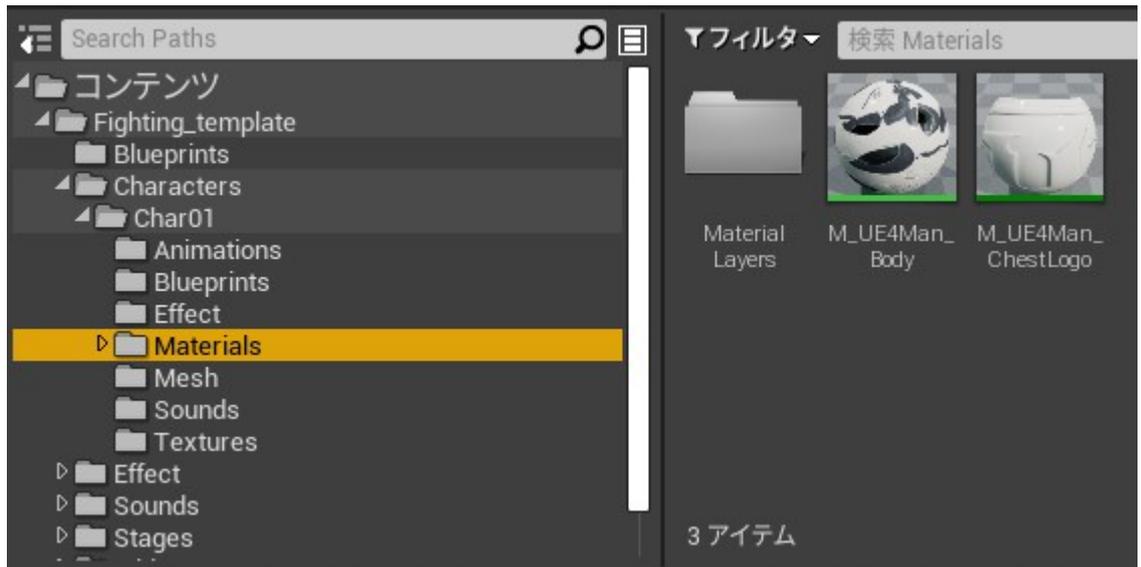
Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

- Effect/Sounds フォルダ



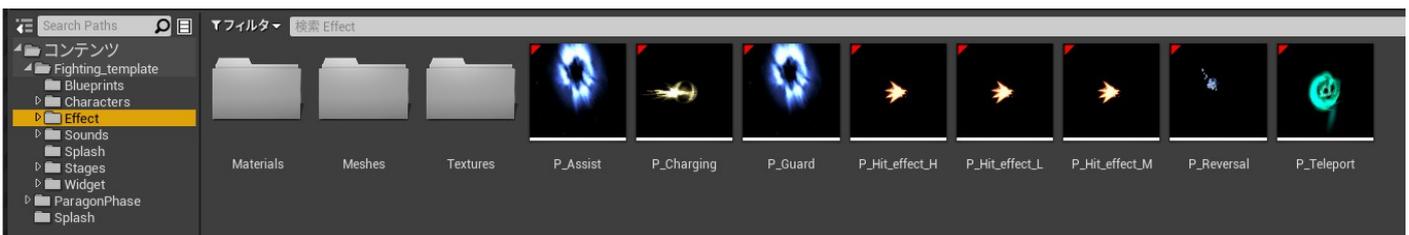
キャラクタ専用必殺技用のパーティクルやサウンドが保存されています。

- Materials/Mesh/Textures フォルダ



UE4 標準キャラクターであるグレーマンの素材が保存されています。

■ Effect フォルダ



ヒットやガードしたときのパーティクルとその素材が保存されています。

■ Sounds フォルダ

ヒット時やウィジェットで使用するサウンドが保存されています。

■ Stages フォルダ

ステージの素材やレベルのファイルが保存されています。

■ Widget

メニュー画面等で必要なテクスチャを制御する Blueprint が保存されています。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

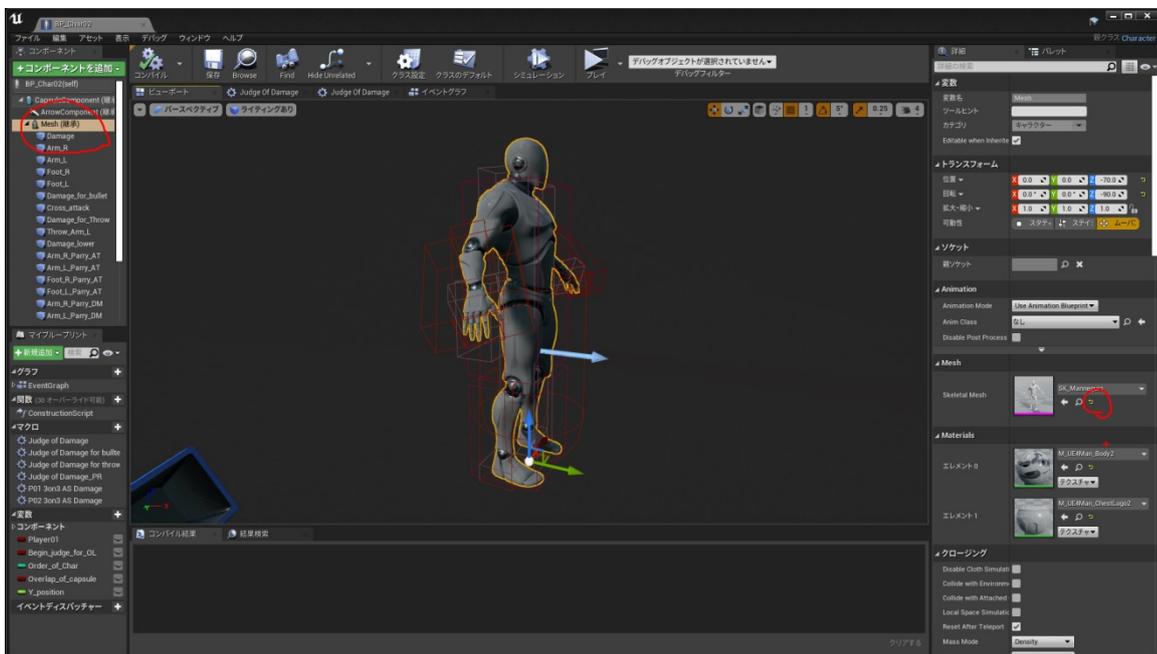
4. キャラクターの追加と編集方法について

ここでは2番目のキャラクターとして Paragon の Phase を追加する例で説明します。

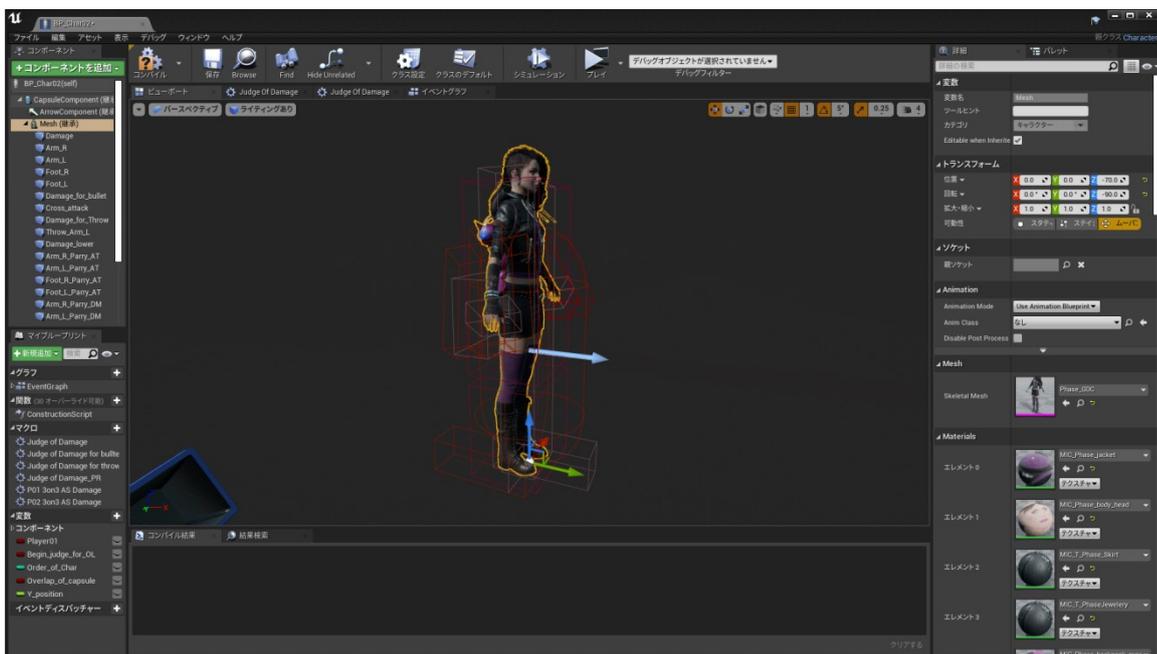
まずは Characters の Char02 フォルダにある BP_Char02 ブループリントを開きます。



左側のコンポーネントタグで Mesh を選択し、右側のメッシュのタブでメッシュをリセットします。



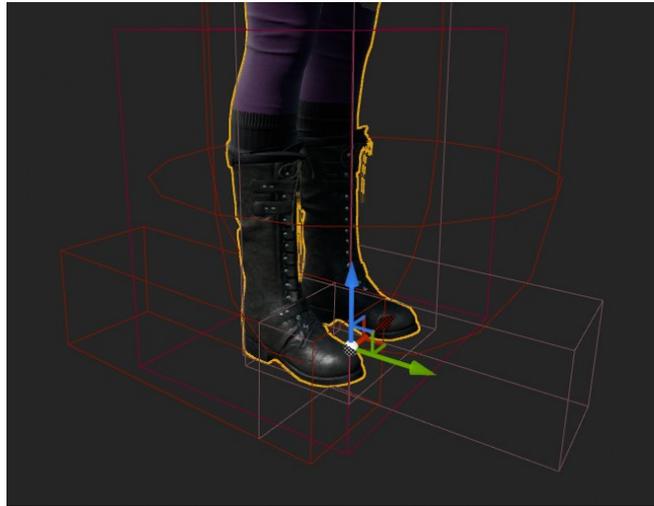
変更する Mesh を選択して Mesh を入れ替えます。



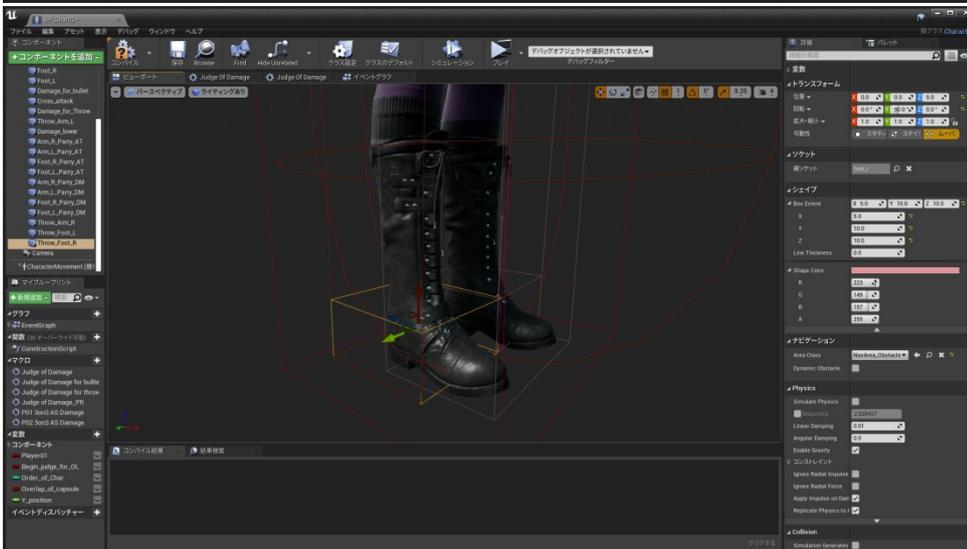
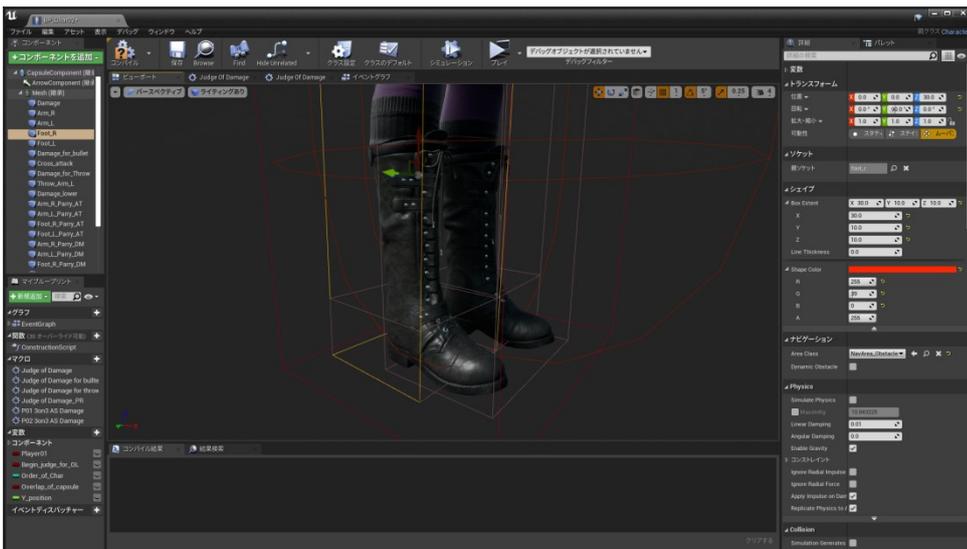
Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

Mesh を入れ替えたなら、各コリジョンボックスの位置と向きを確認します。

変更する Mesh により、マネキンで設定したコリジョンボックスの位置や向きが変更されてしまう場合があります。



今回は、両足に設定した打撃やダメージ、投げのコリジョンボックスの位置や向きが変わったので、修正します。(Foot_R&L / Foot_R&L_Parry_AT / Foot_R&L_Parry_DM / Throw_Foot_R&L)

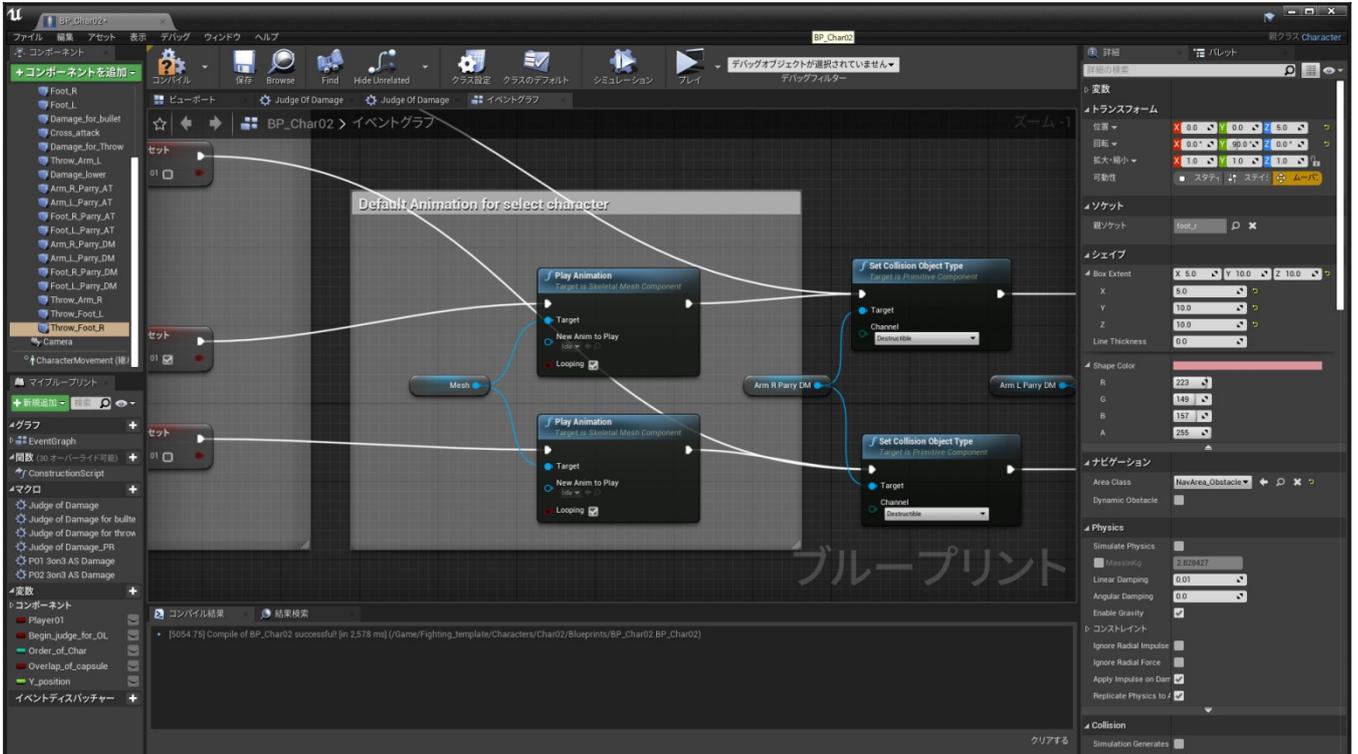


修正したら、ここで一旦コンパイルと保存をします。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

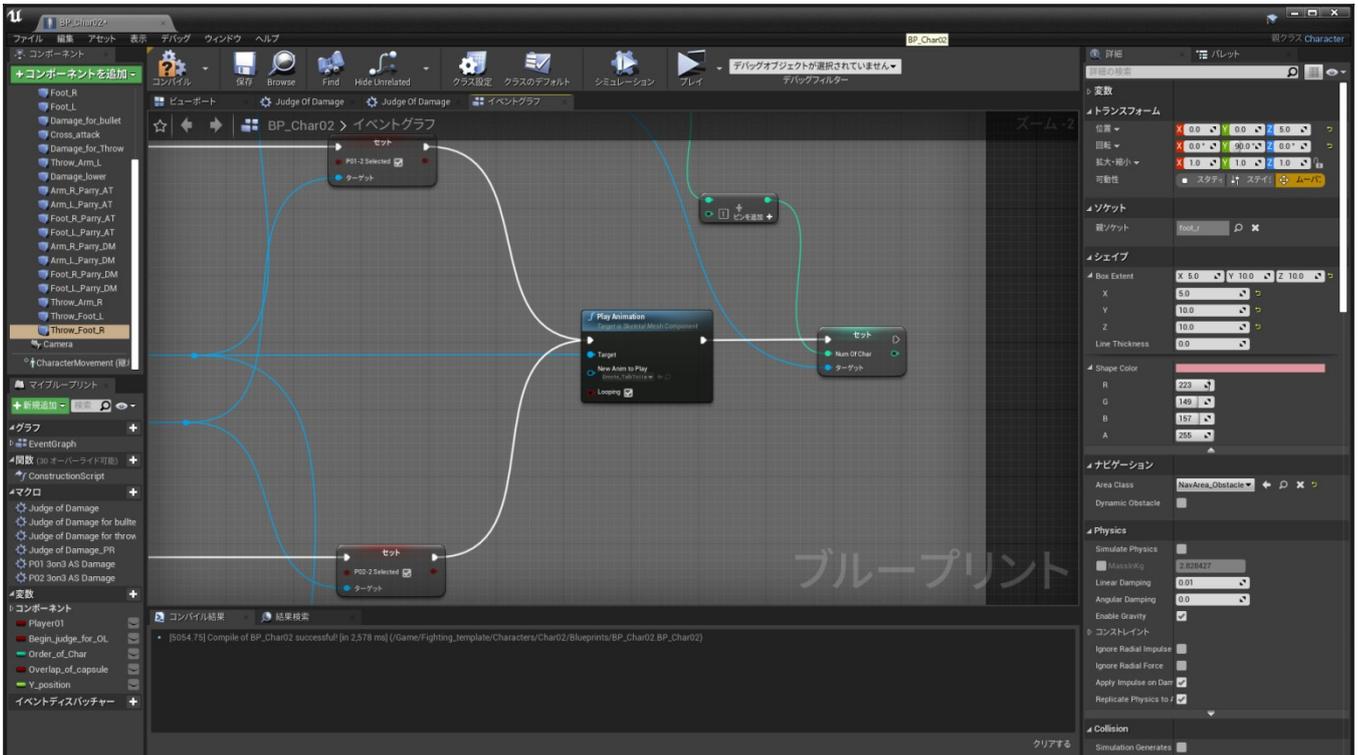
次にイベントグラフのタブで、キャラクター選択画面のアニメーションを設定します。

まずは、「Default Animation for select character」とコメントがあるノードで、キャラクターが選択される前のアニメーションを設定します。



そして、「Define color and animation for select character」とコメントがあるノードで、選択された時のアニメーションを設定します。

アニメーションをループさせたくない場合はチェックボックスをOFFにします。

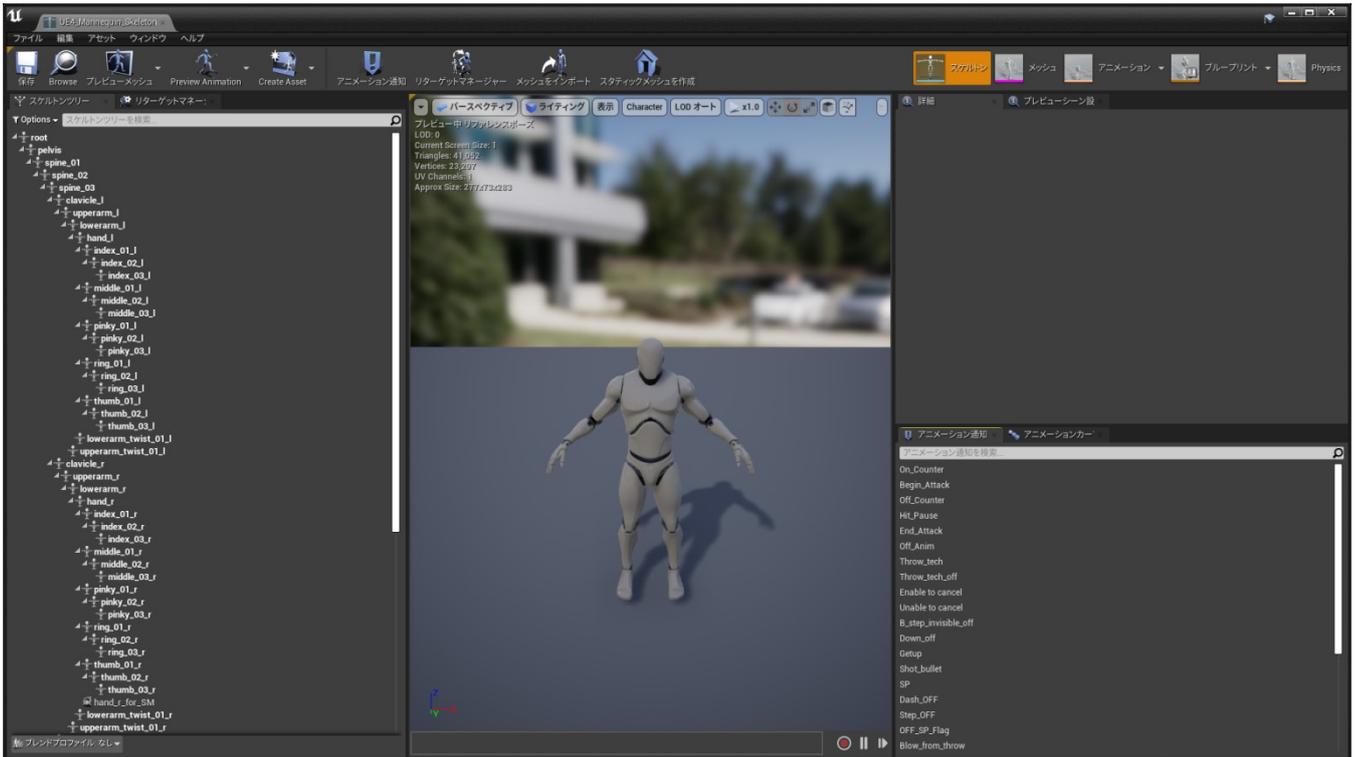


コンパイルと保存をしたら、キャラクターブループリントの設定は一旦完了です。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

次にアニメーションブループリントのリターゲットをします。

まずはスケルトンの設定が必要です。マネキンのスケルトンファイルを開きます。

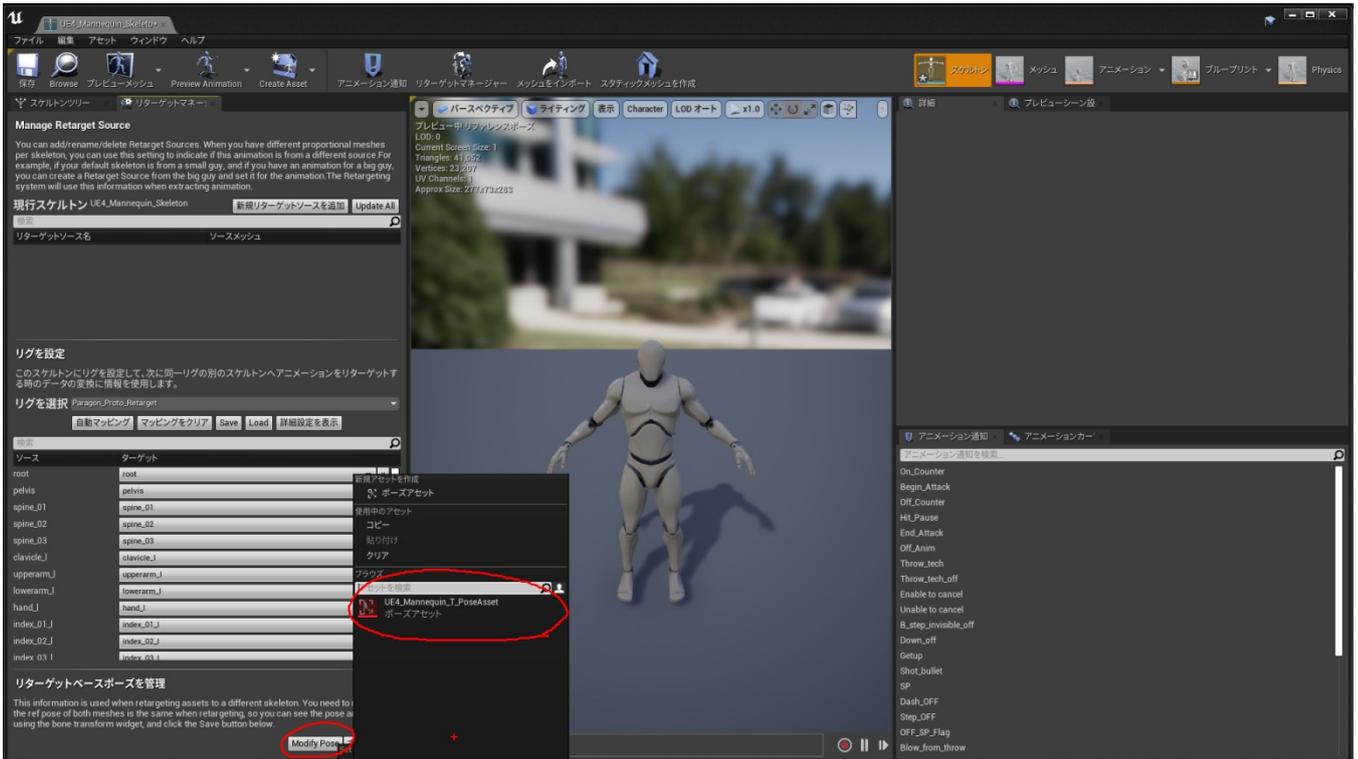


左上のリターゲットマネージャーのタブをクリックして、追加するキャラクターに適応したリグを設定します。今回は Paragon 用のリターゲットリグがあるのでそれを選択しています。



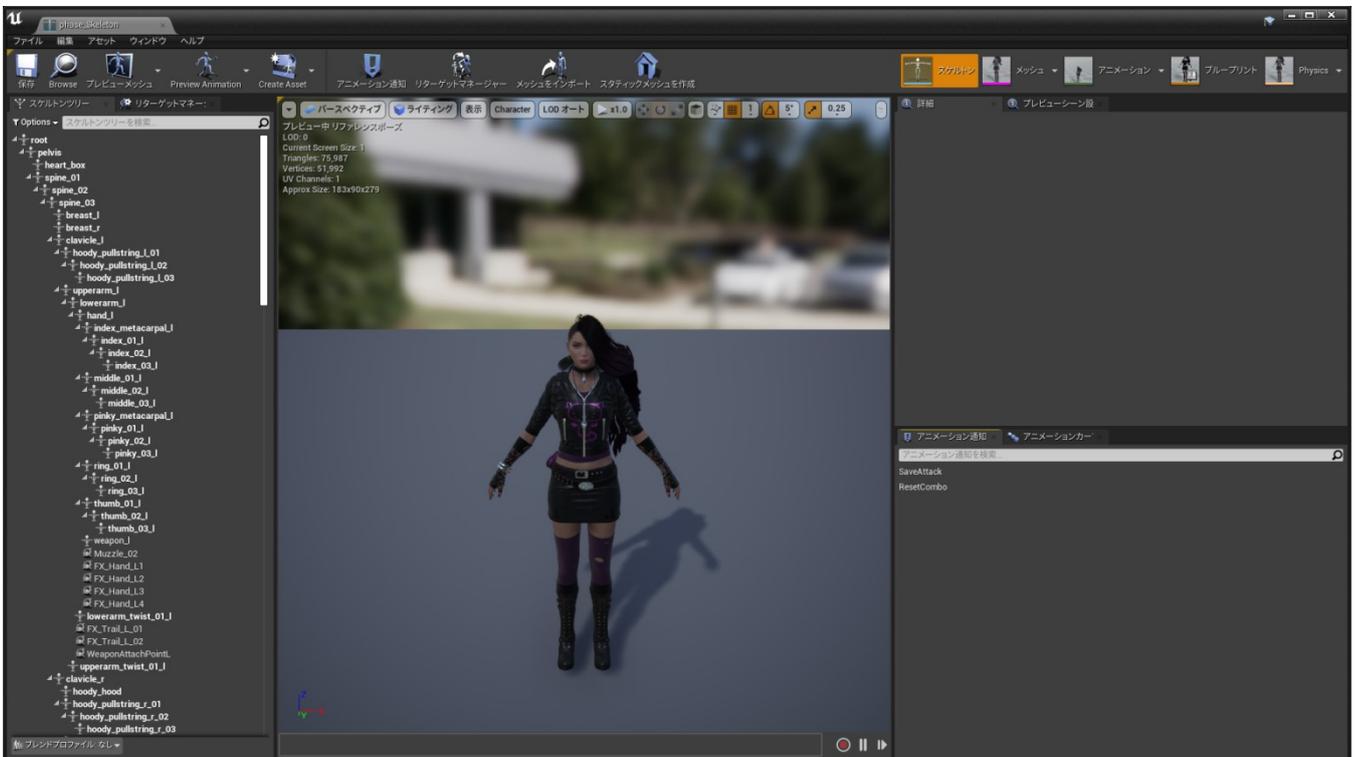
Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

もし追加するキャラクターのベースポーズがTポーズの場合は、マネキンのベースポーズをTポーズに変更してください。



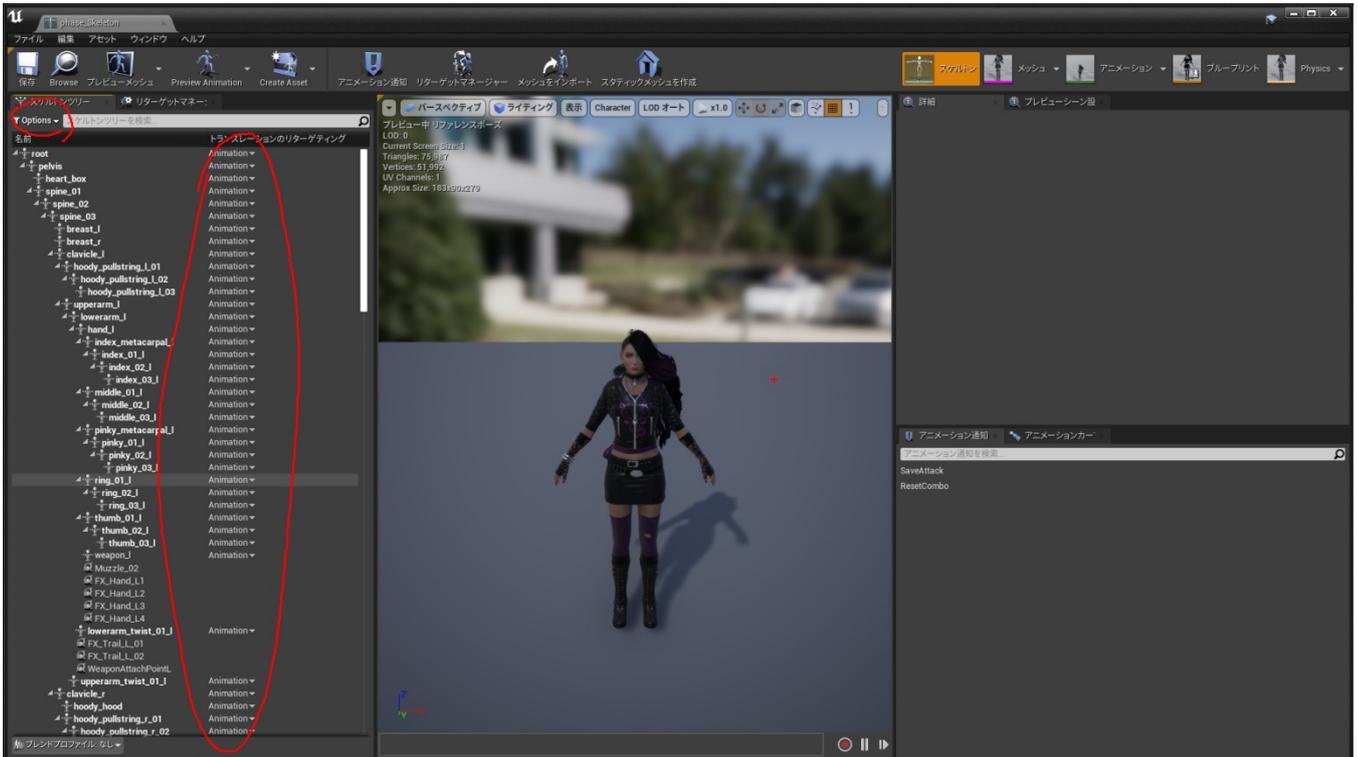
ベースポーズを確認したらスケルトンファイルを保存します。

それでは次に、追加するキャラクターのスケルトンファイルを開きます。

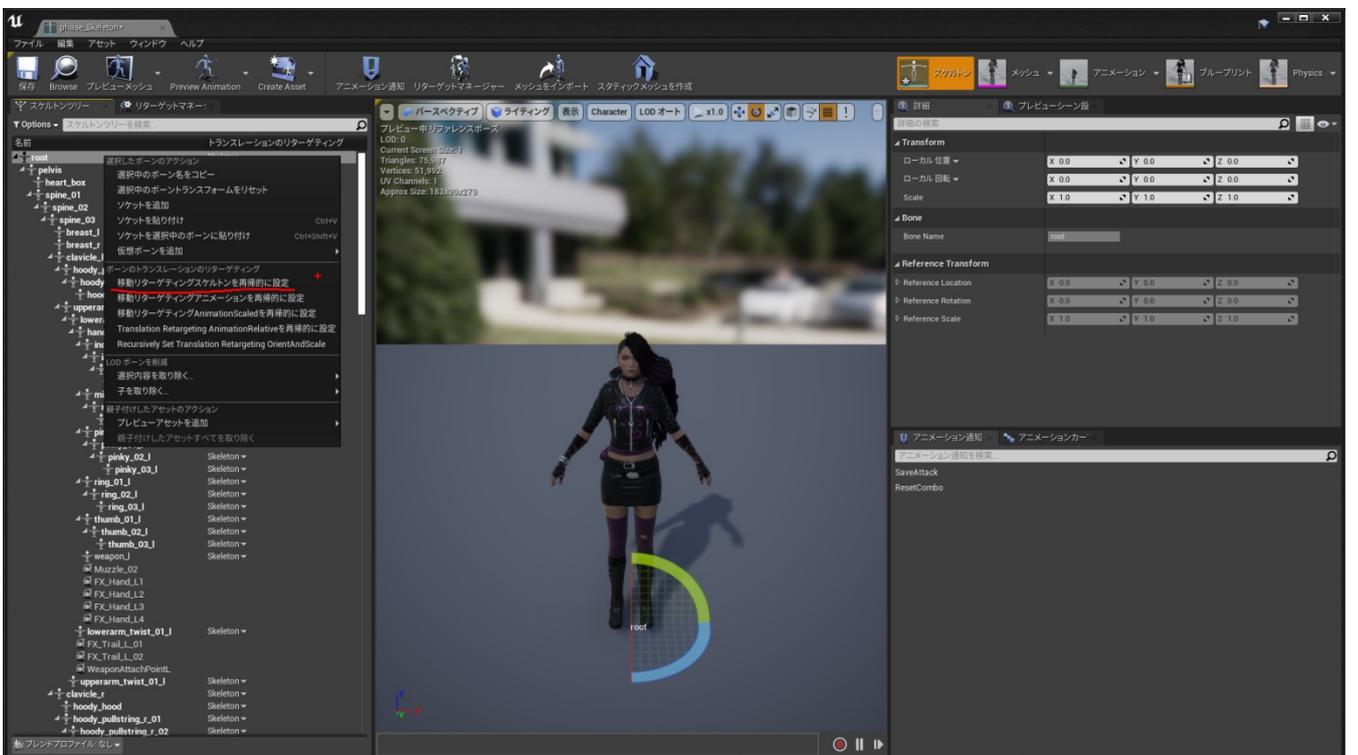


Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

次に左上のスケルトンツリーのタブの下にあるオプションを選択してリターゲットオプションを表示します。



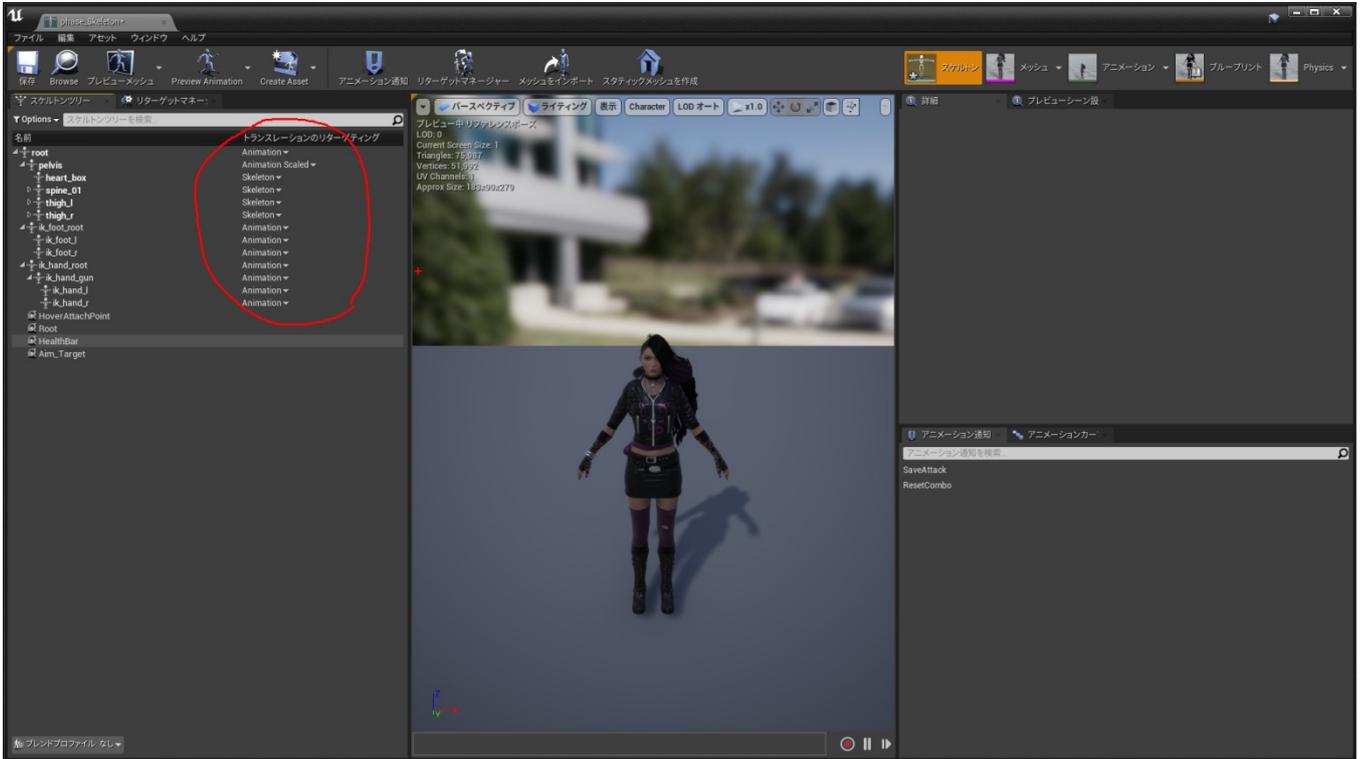
次に、root ボーンを右クリックして、全てのボーンのリターゲットオプションをスケルトンに変更します。



Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

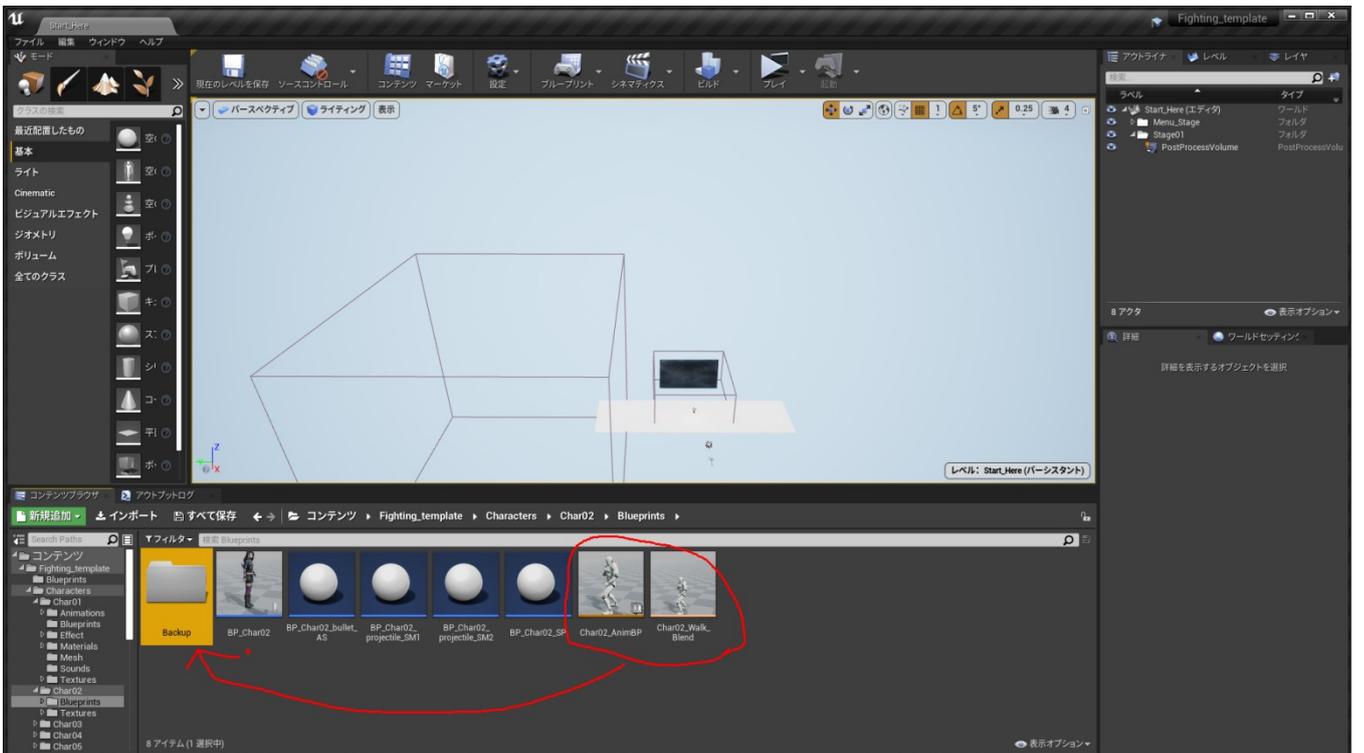
そして、root と ik のボーンのリターゲットオプションを Animation に設定し、Pelvis のリターゲットオプションを Animation Scaled に設定します。

これでボーンの長さが違ったスケルトン同士のリターゲットを問題なく実施できます。



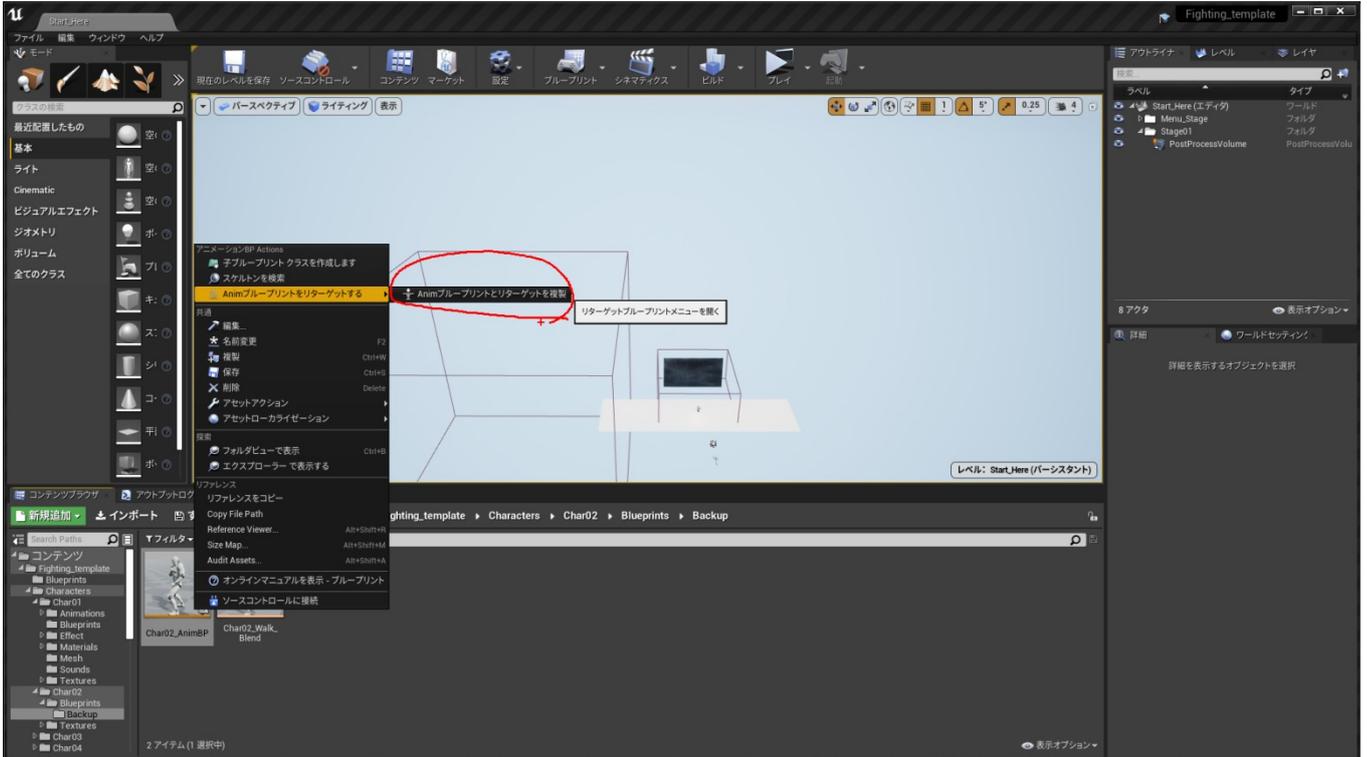
変更したらスケルトンを保存します。

それではこれからアニメブループリントのリターゲットをしますが、その前に Char02 フォルダにバックアップフォルダを作成し、変更前のアニメブループリントとアニメブレンドファイルを移動します。

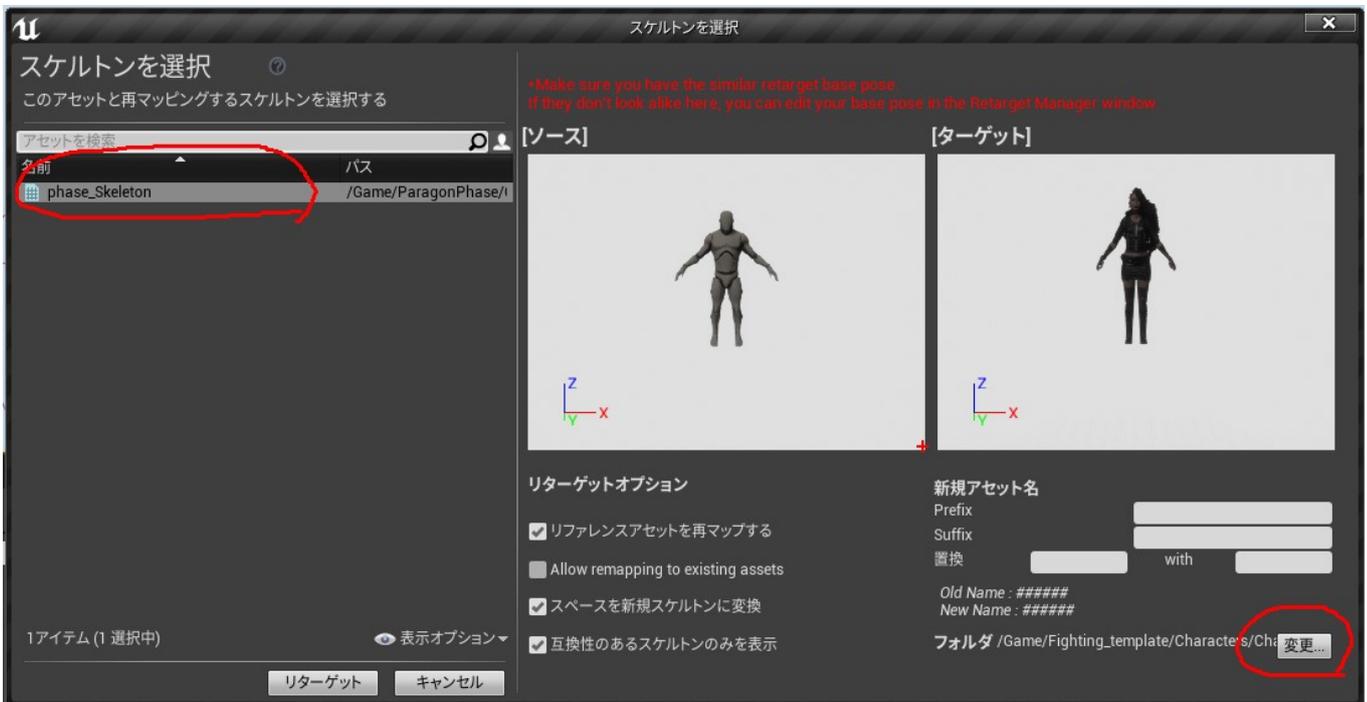


Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

次に、バックアップフォルダに移動したアニメブループリントを右クリックしてリターゲットを選択します。

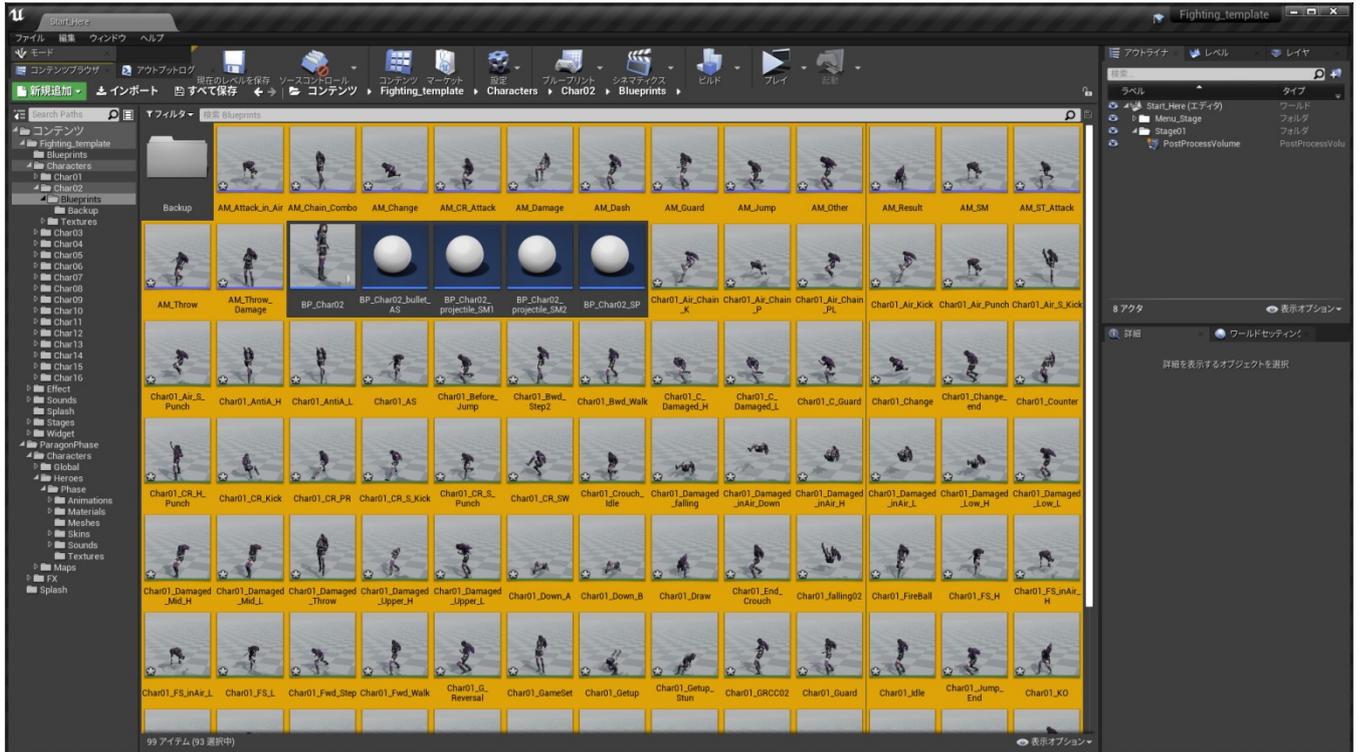


Phaseのスケルトンが選択できるようになっていますので、それを選択後にリターゲットファイルの保存先をChar02のBlueprintフォルダに変更してリターゲットをします。

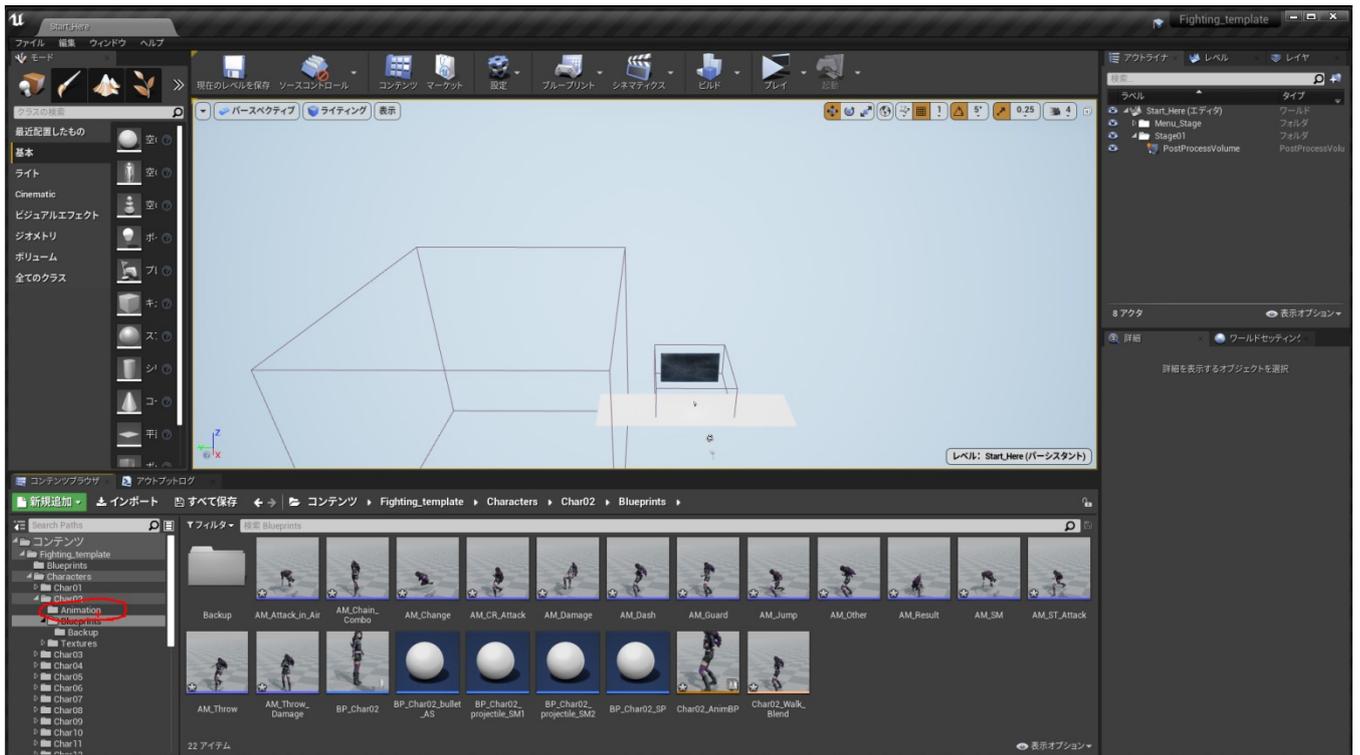


Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

リターゲットすると指定したフォルダに必要なアニメーションファイルがすべて作成されます。

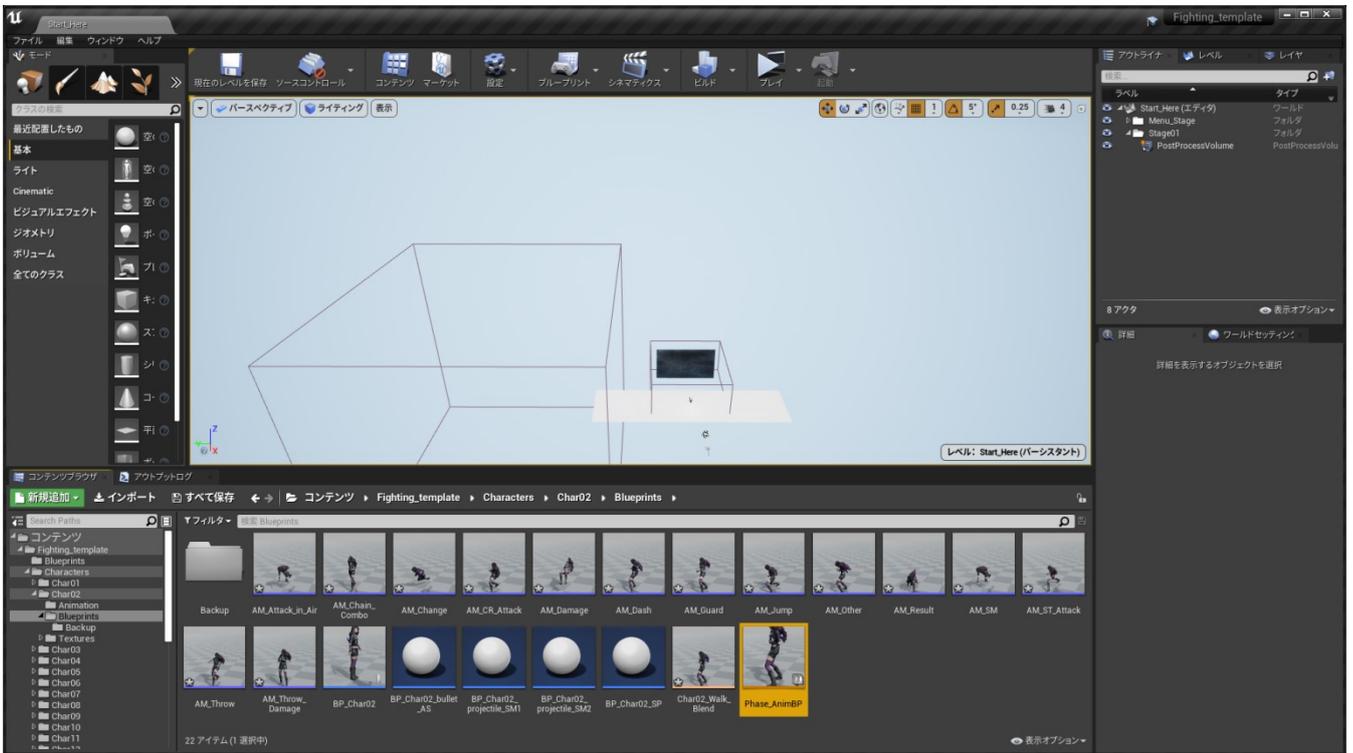


Char02 フォルダの下にアニメーションフォルダを作成してアニメーションファイルを移動しましょう。

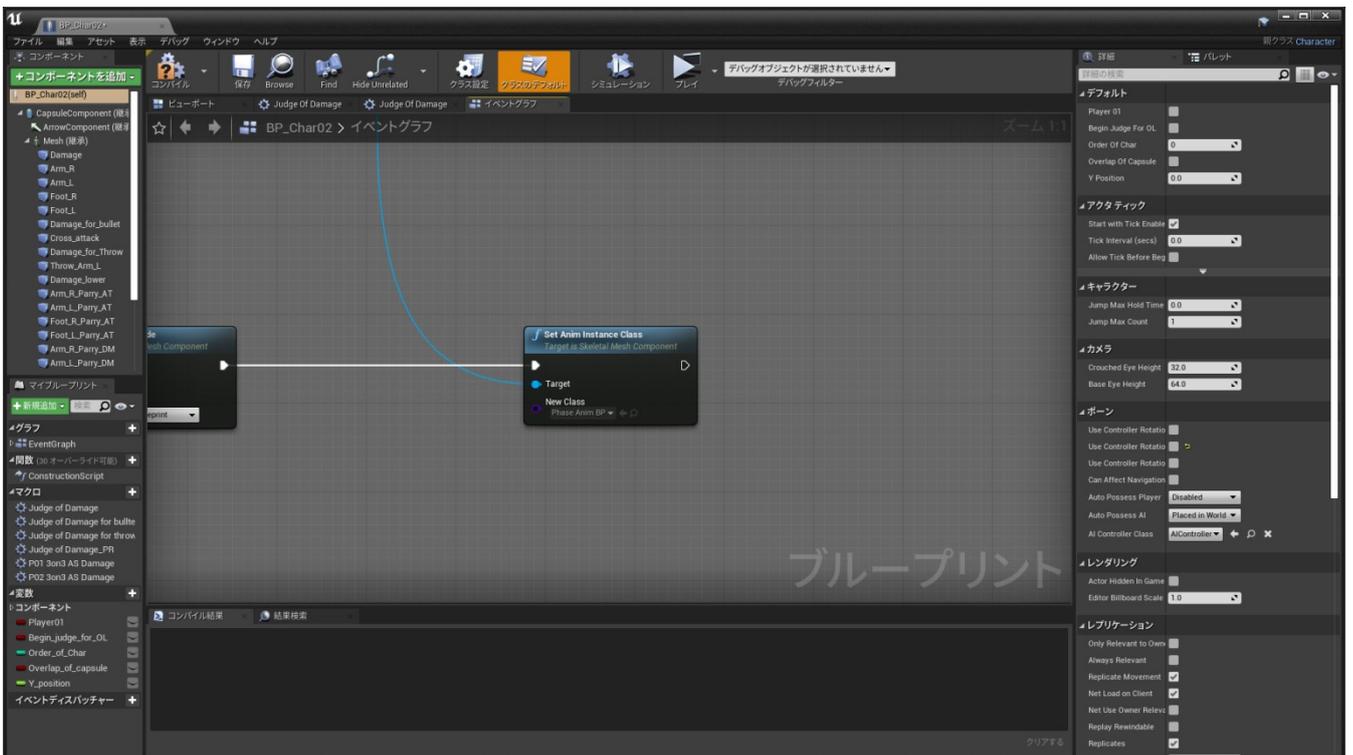


Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

次にリターゲットされた Phase の Char02_AnimBP のファイル名を Phase_AnimBP に変更します。
これはキャラクターブループリントでアニメブループリントを指定するときに、以前と同じファイル名
だとわかりにくいからです。



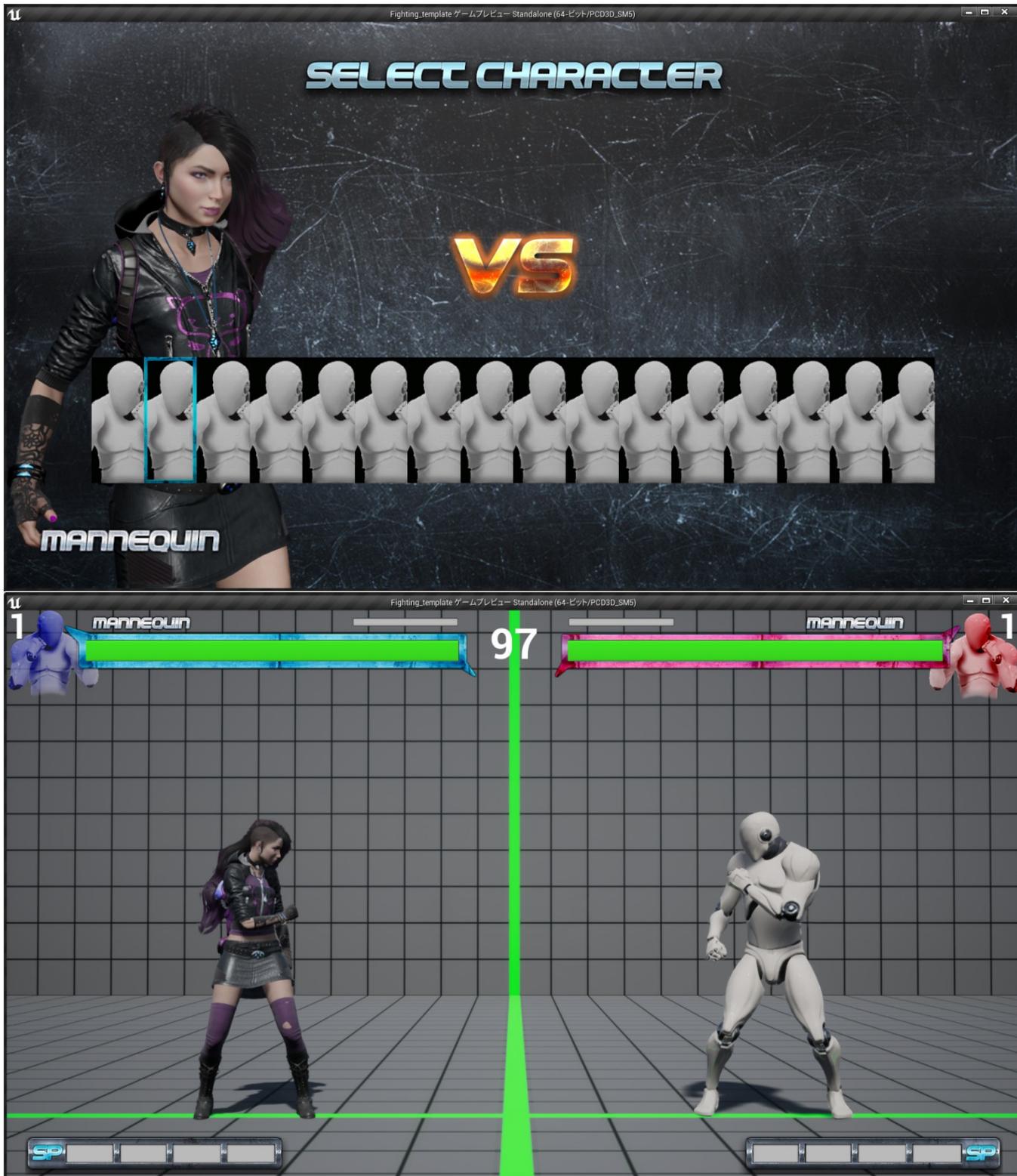
次に、BP_char02 のファイルを開いて、「Define color and animation for select character」とコメントがあるブロックの右下にあるノードで、変更したアニメブループリントを指定してコンパイルし、保存します。



Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

これで、ゲームを実行すると、キャラクター選択画面で、Phaseが登場し、バトルでもマネキンと同じアニメーションを適用したPhaseが使えるようになります。

ただし、これは仮のアニメーションで、エフェクトの表示場所や動作が一部正しく機能しない場合がありますので、後ほどアニメーションの入れ替えを説明します。

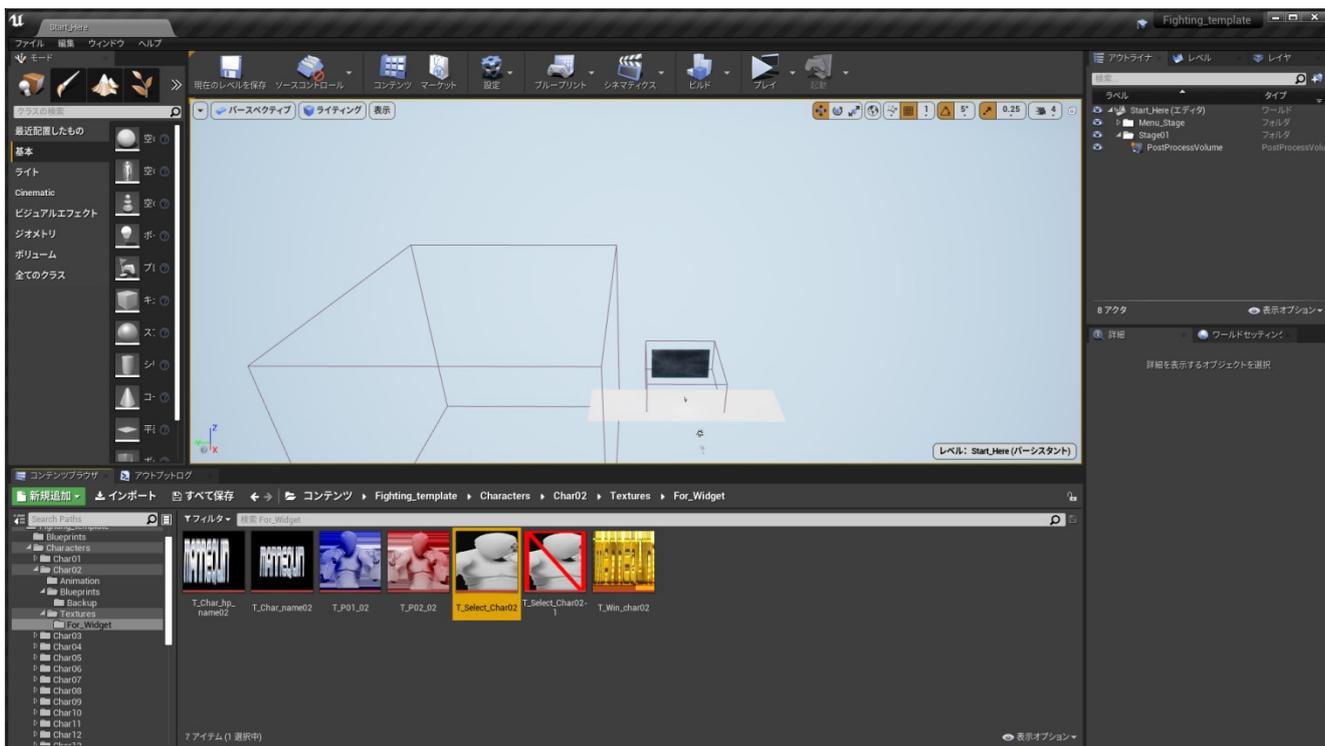


Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

次に Widget_Blueprint で使用する画像を説明します。

Char02 フォルダの下に Textures フォルダがあり、以下の画像が保存されています。

- T_Char_hp_name02(250*30pixel 72dpi バトル時 HP 上に表示するキャラクター名の画像)
- T_Char_name02(300*50pixel 72dpi キャラクターセレクト時に表示するキャラクター名の画像)
- T_P01_02(304*335pixel 72dpi バトル時 HP 横に表示するキャラクター画像 Player01 側)
- T_P02_02(304*335pixel 72dpi バトル時 HP 横に表示するキャラクター画像 Player02 側)
- T_Select_Char02(100*240pixel 72dpi キャラクターセレクト時に表示するキャラクターの画像)
- T_Select_Char02-1(100*240pixel 72dpi キャラクターセレクト後に表示するキャラクターの画像)
- T_Win_char02(650*100pixel 72dpi バトル勝利時に表示するキャラクター名の画像)

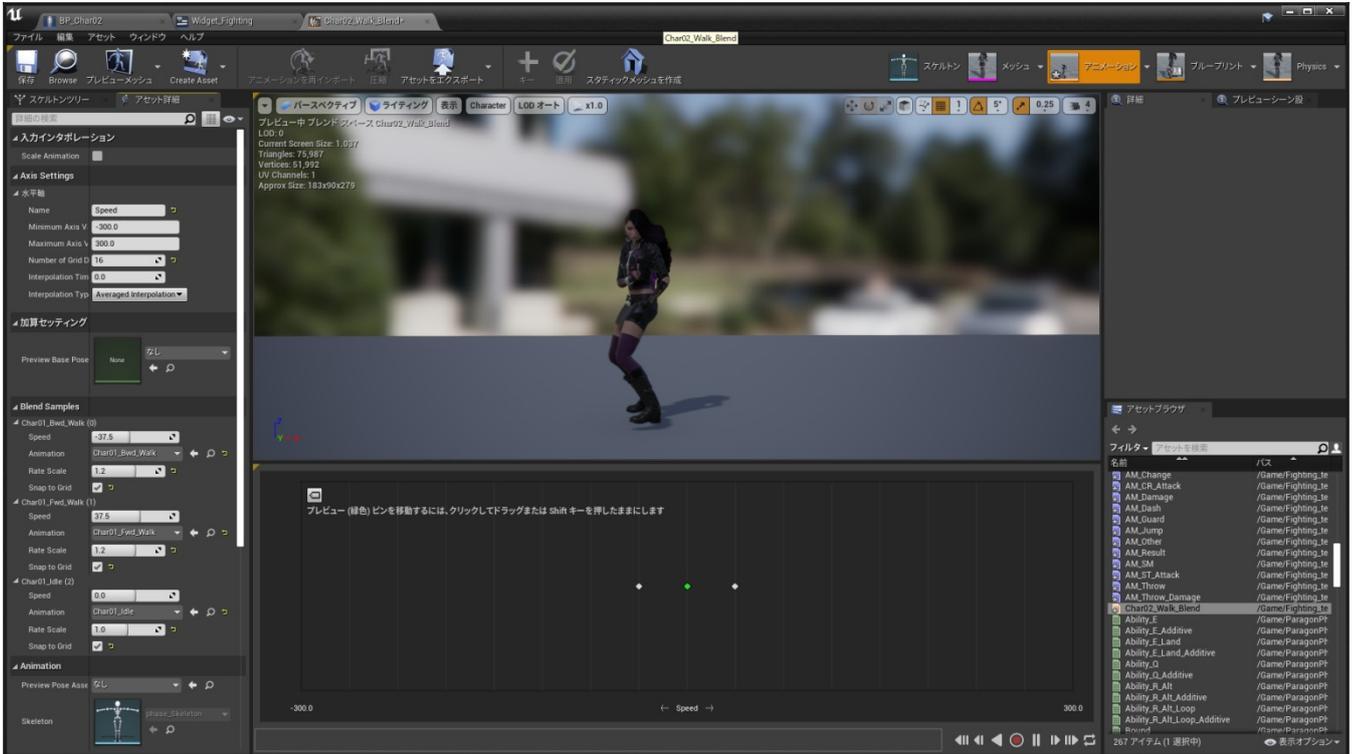


追加したキャラクターで同じファイル名と解像度のファイルを用意して上書きすることで、自動的に Widget_Blueprint へ適用されます。

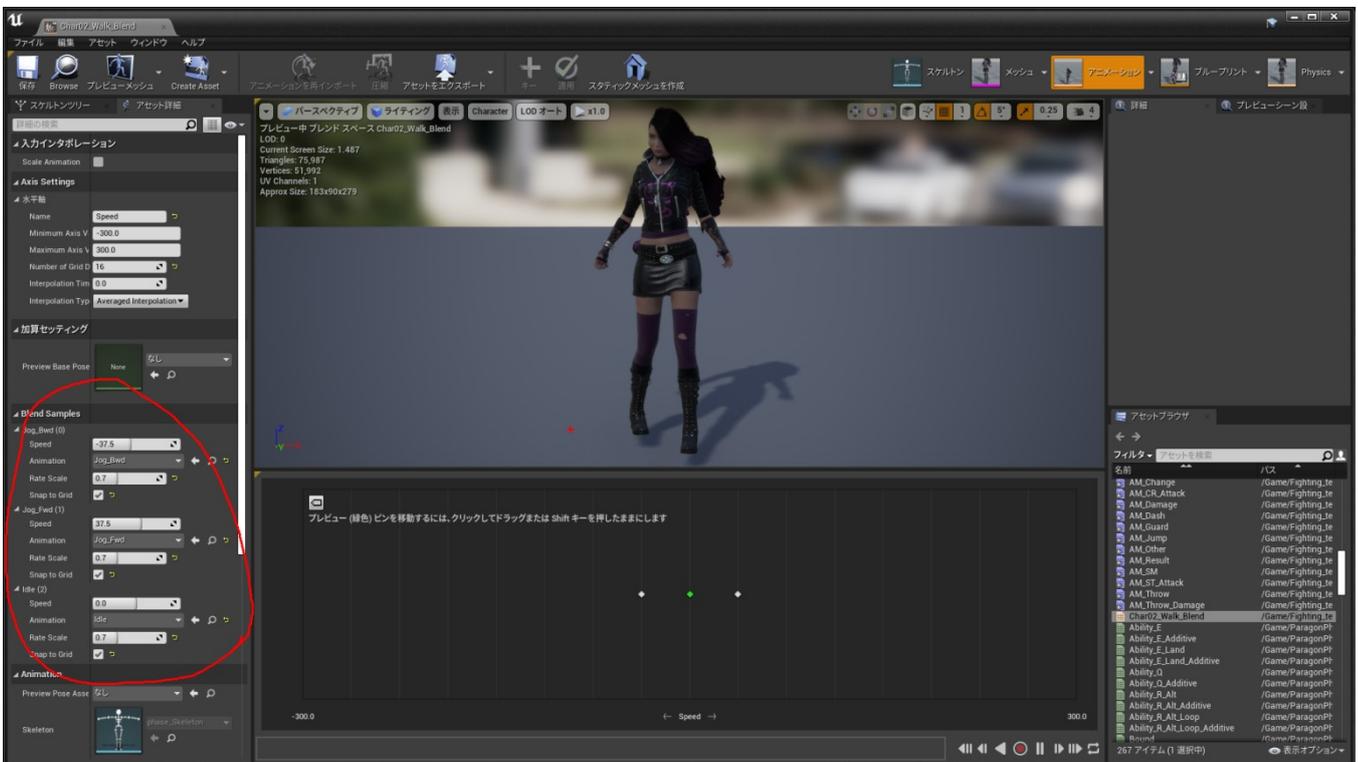
Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

それでは、ここからキャラクターのアニメーション入れ替え方法について説明します。

まずは、アイドル状態と前後の歩くアニメーションを入れ替えますので、Char02_Walk_Blendのファイルを開きます。



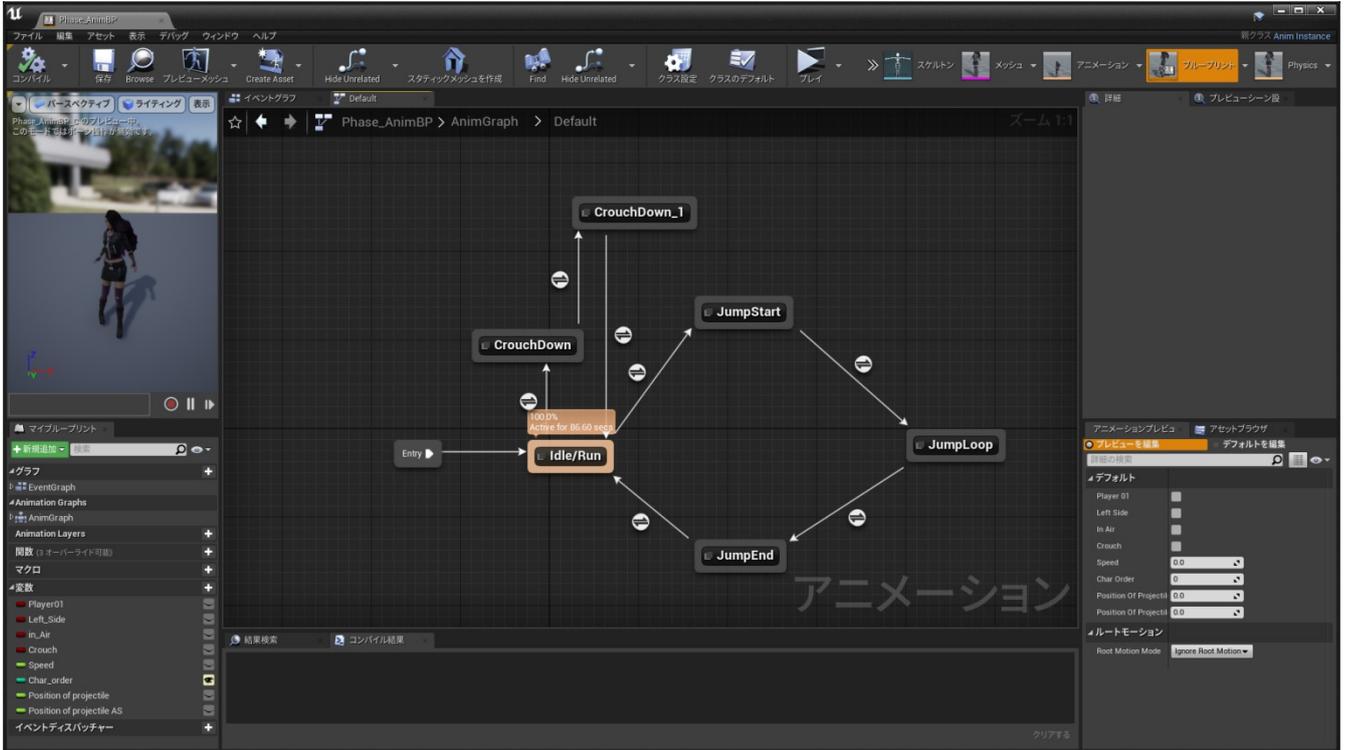
左側に Blend Samples という項目がありますので、上から後ろ歩き、前歩き、アイドルの順番でアニメーションを入れ替えます。必要に応じてアニメーションの再生レートを調整してください。



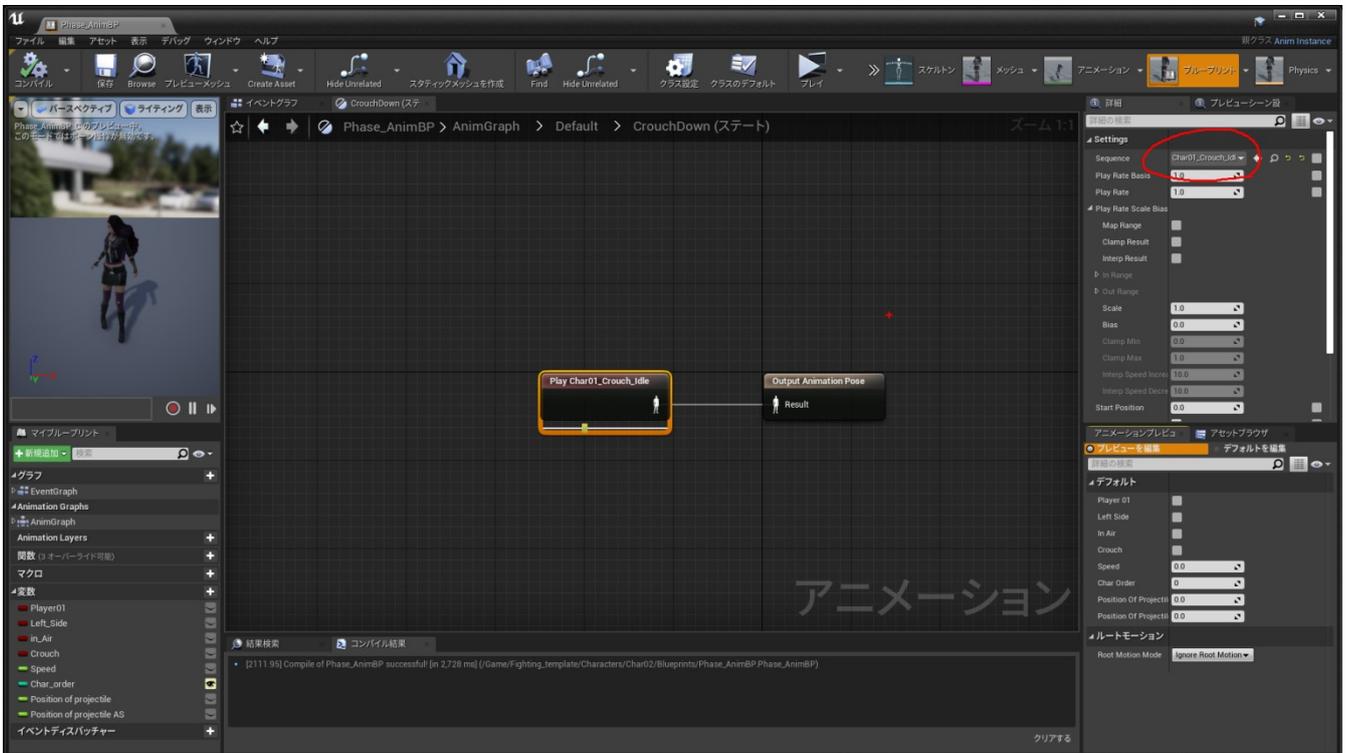
これで設定は完了です。保存してファイルを閉じます。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

次は、アニメブループリントのステートマシンに登録されたアニメーションを入れ替えますので、Phase_AnimBP のアニメグラフからステートマシンを開きます。

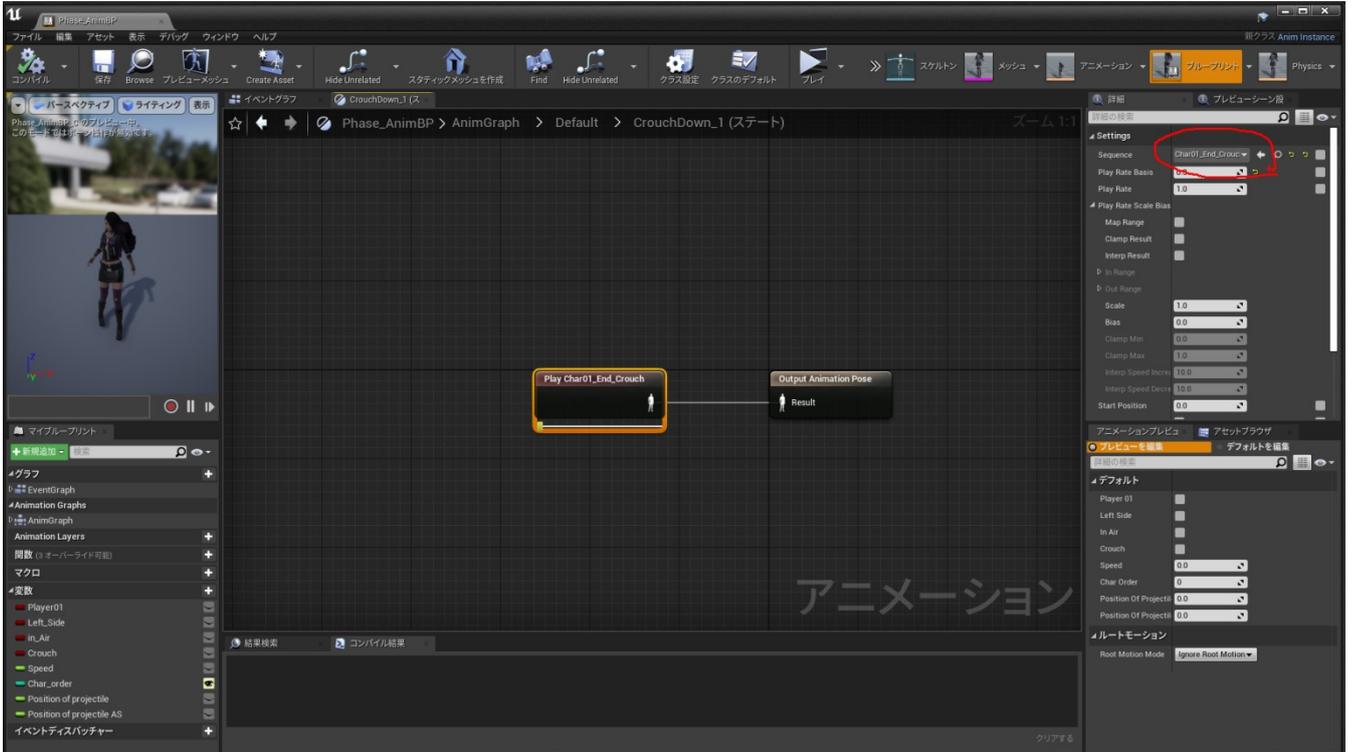


次に CrouchDown のステートを開き、右側にある詳細のタブでアニメーションを入れ替えます。ここで入れ替えるのはしゃがみ状態のアイドルアニメーションです。

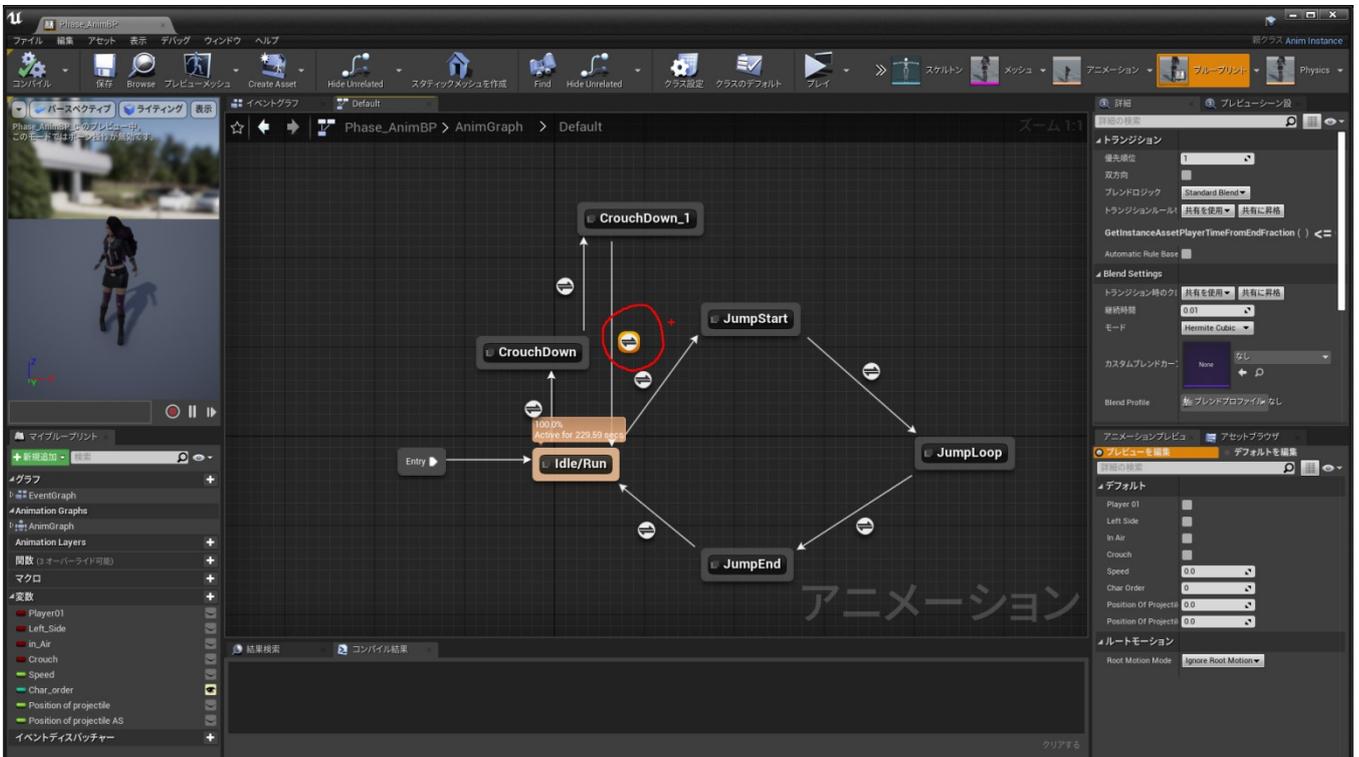


Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

次に CrouchDown_1 の状態を開き、右側にある詳細のタブでアニメーションを入れ替えます。
ここで入れ替えるのはしゃがみ状態から立ち上がる時のアニメーションです。

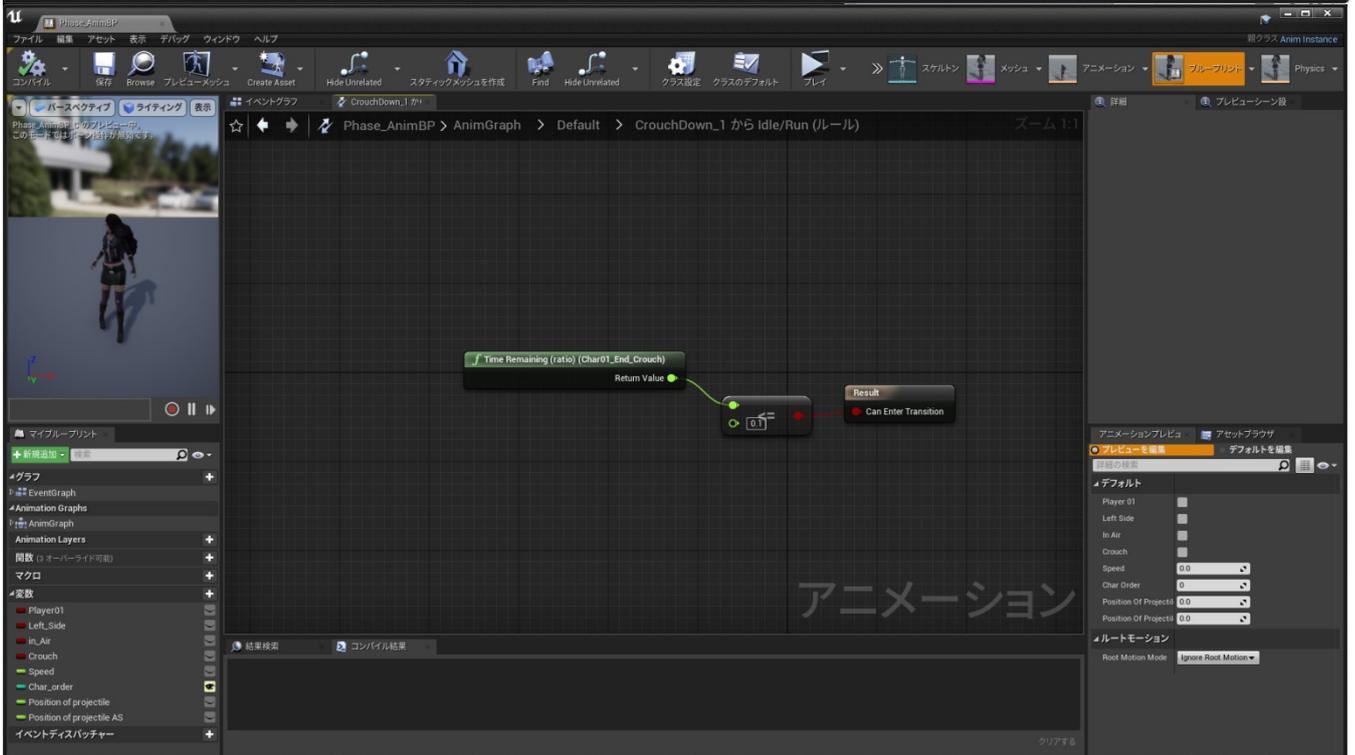
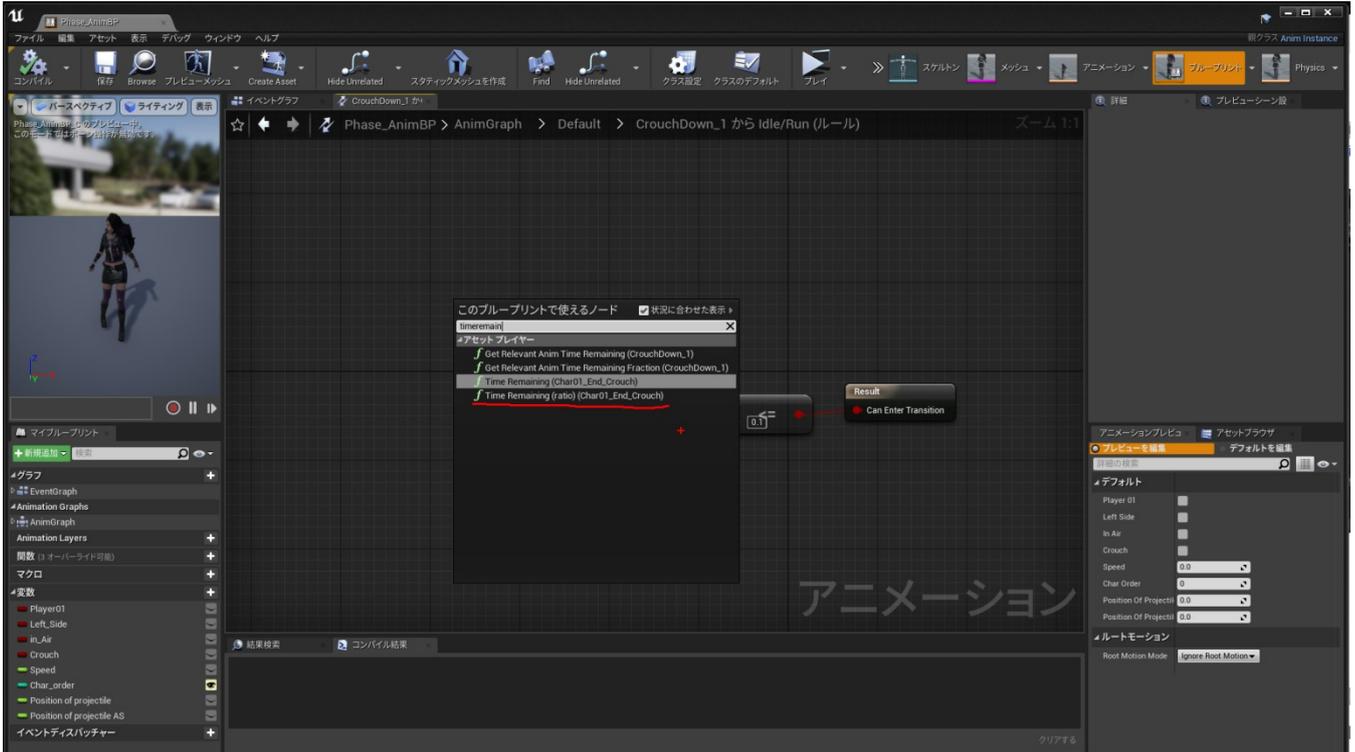


次に立ち上がる時の条件を再設定しますので、CrouchDown_1 から Idle/Run に向かっている矢印をダブルクリックします。



Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

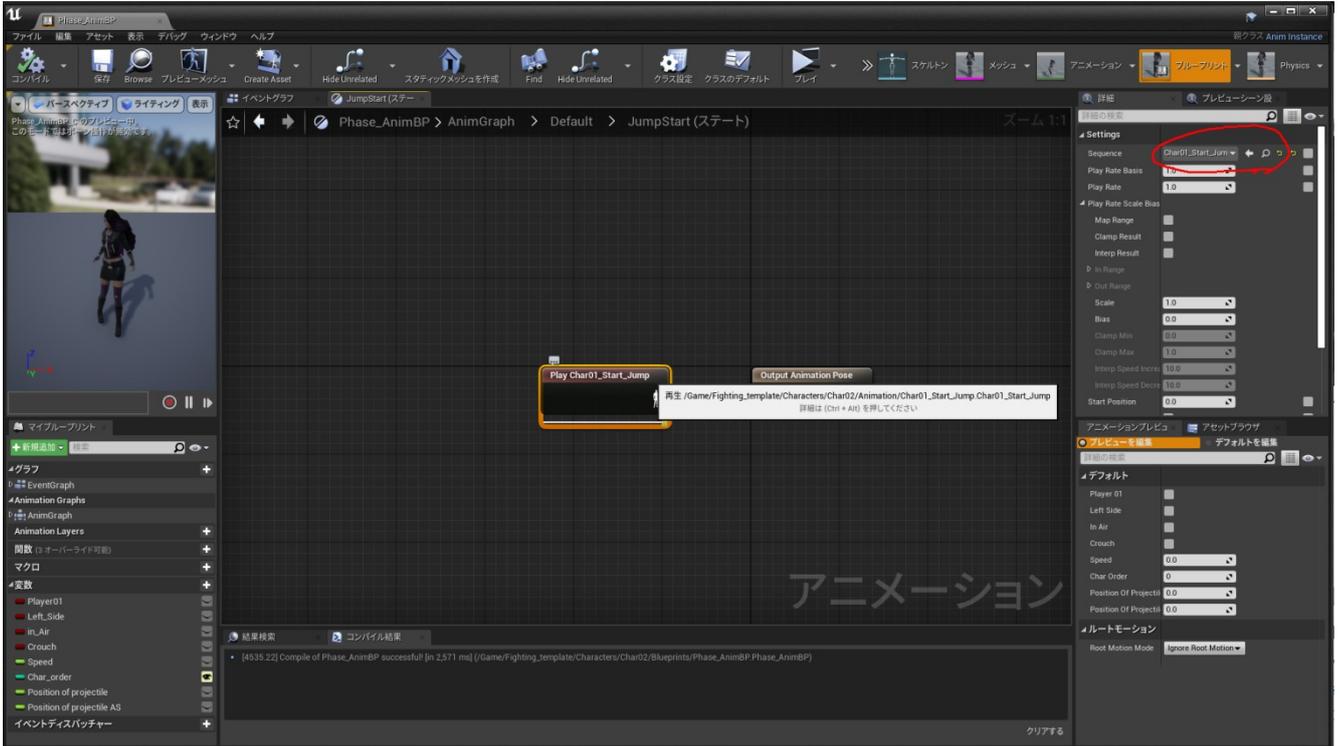
右クリックして TimeRemainning と入力し、TimeRemainning(ratio)(入れ替えたアニメーションファイル名) の関数を選択してノードをつなぎ直します。



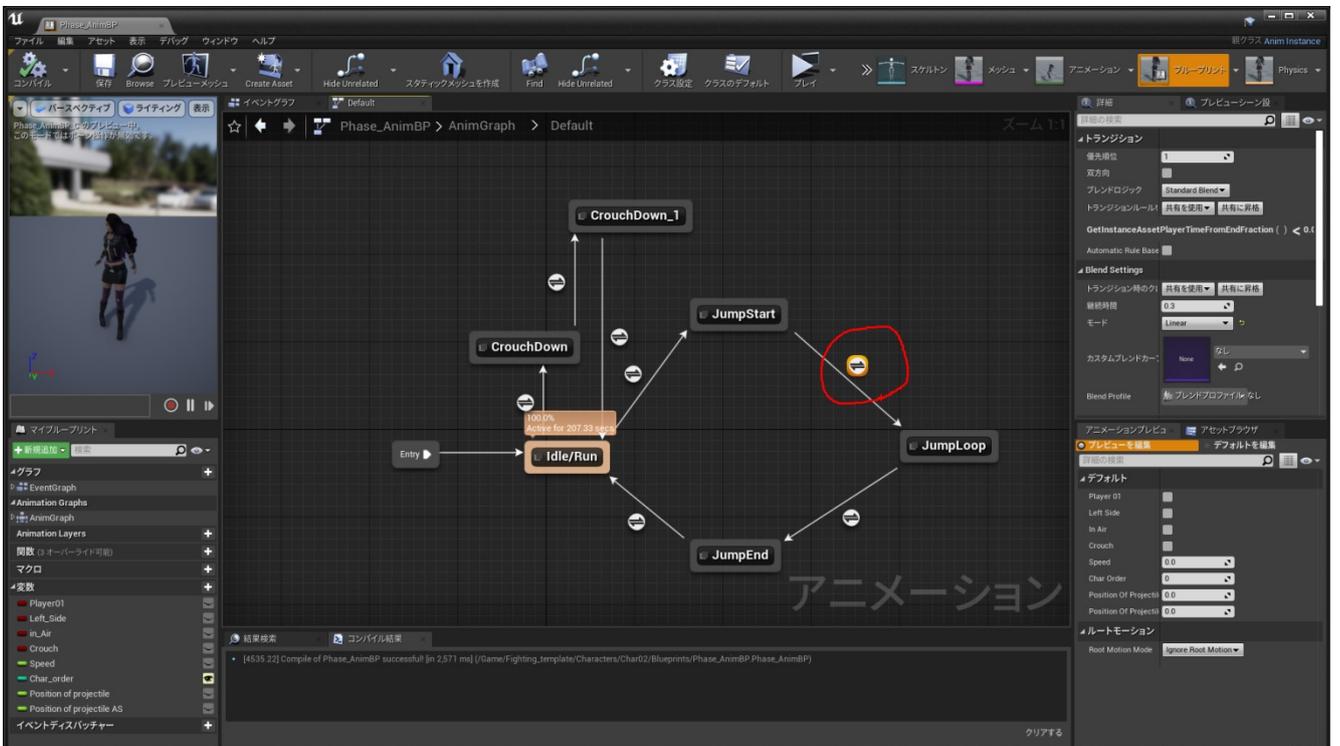
ノードをつなぎ直したらコンパイルして保存しましょう。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

次に JumpStart の状態を開き、右側にある詳細のタブでアニメーションを入れ替えます。
ここで入れ替えるのはジャンプしてから空中でのアイドル状態までのアニメーションです。

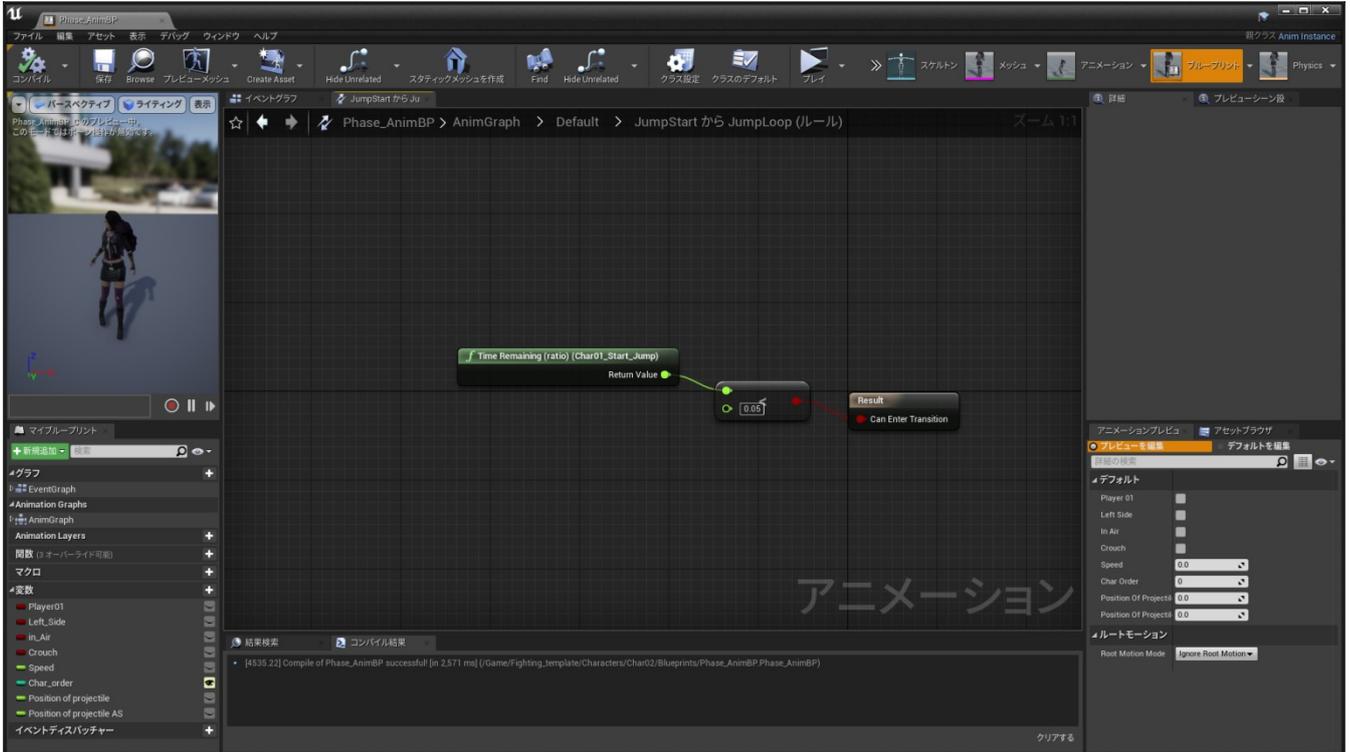


次にジャンプループまでの条件を再設定しますので、JumpStart から JumpLoop に向かっている矢印をダブルクリックします。

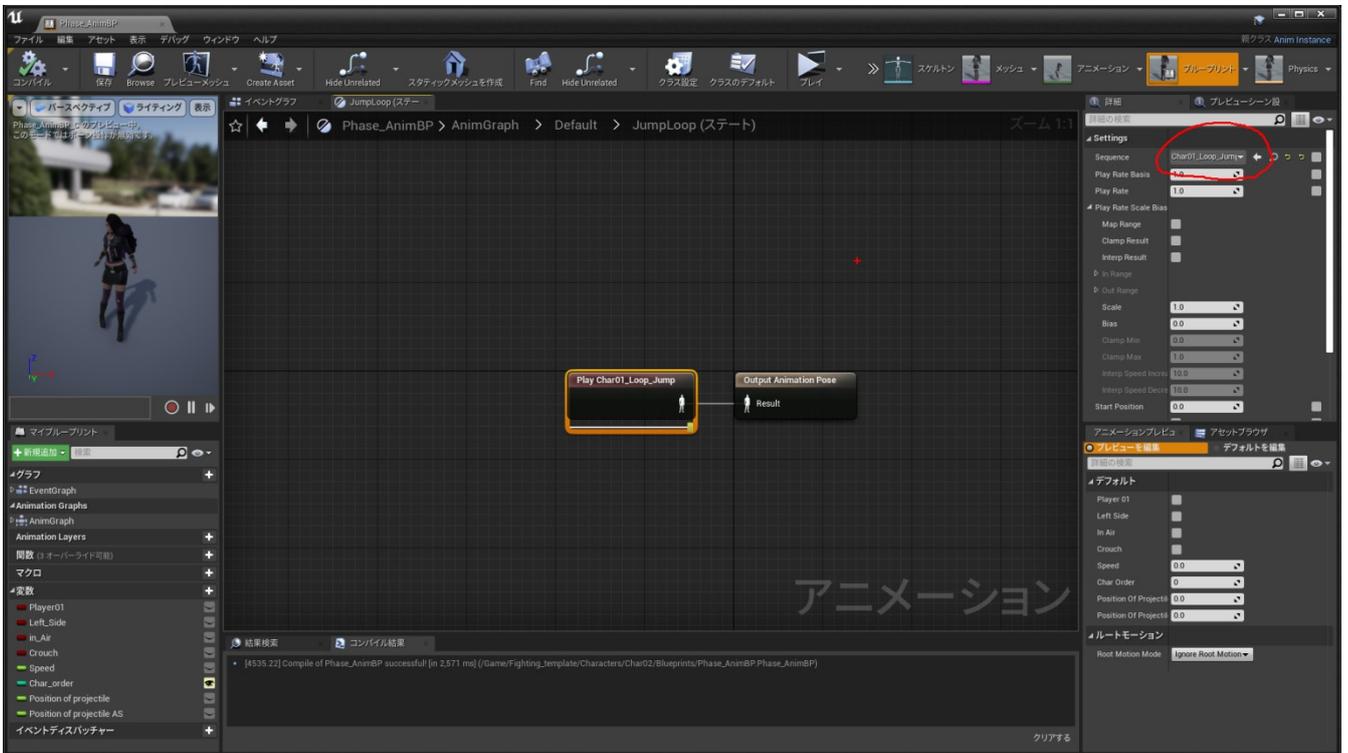


Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

右クリックして TimeRemaininging と入力し、TimeRemaininging(ratio)(入れ替えたアニメーションファイル名) の関数を選択してノードをつなぎ直します。

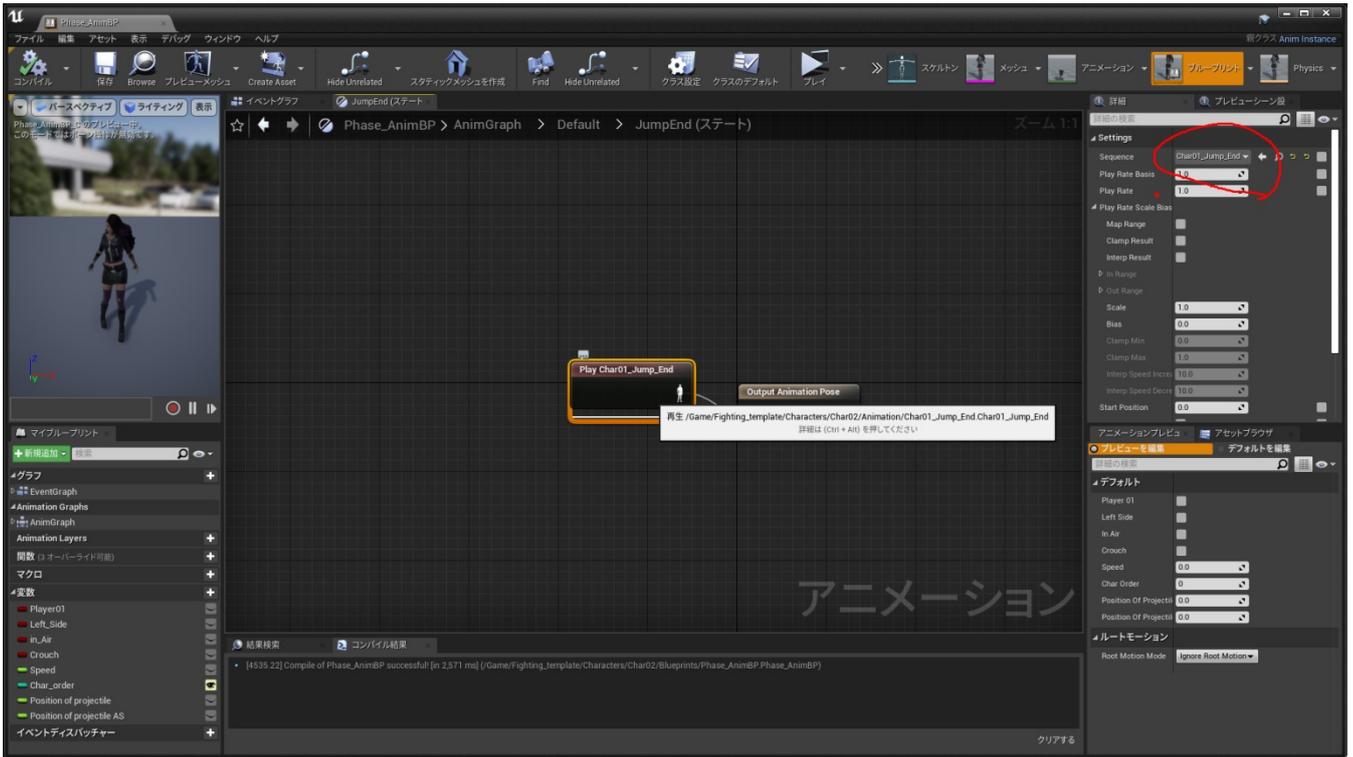


次に JumpLoop の状態を開き、右側にある詳細のタブでアニメーションを入れ替えます。
ここで入れ替えるのは空中でのアイドル状態のアニメーションです。

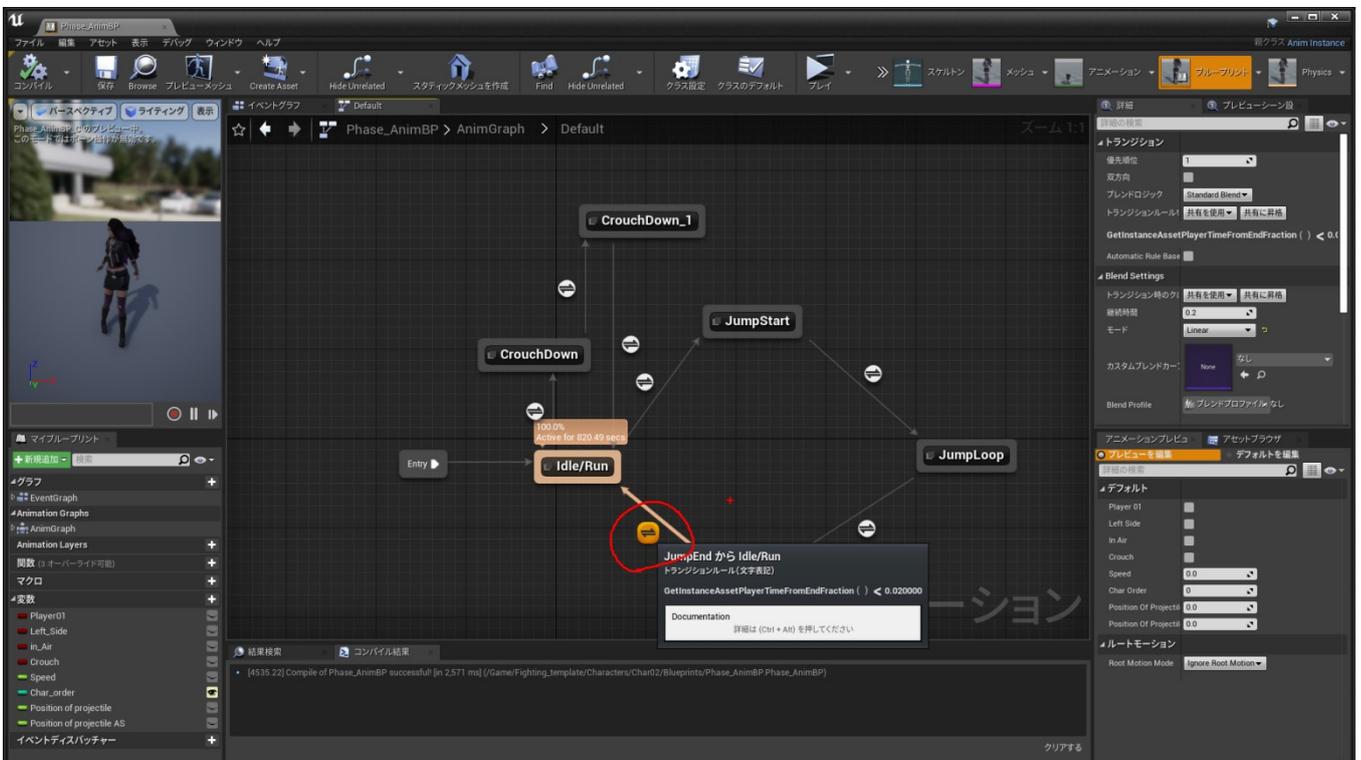


Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

次に JumpEnd の状態を開き、右側にある詳細のタブでアニメーションを入れ替えます。
ここで入れ替えるのは着地したときのアニメーションです。

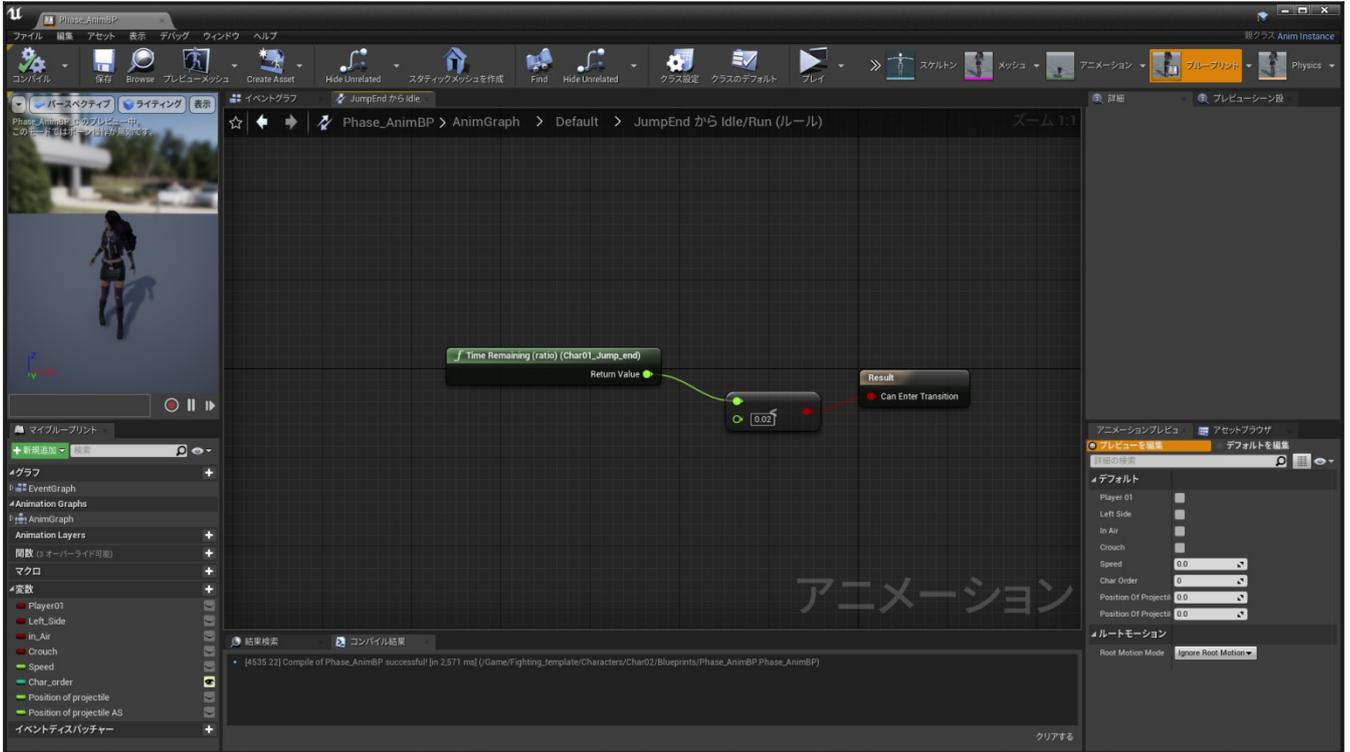


次に着地から Idle までの条件を再設定しますので、JumpEnd から JumpLoop に向かっている矢印をダブルクリックします。

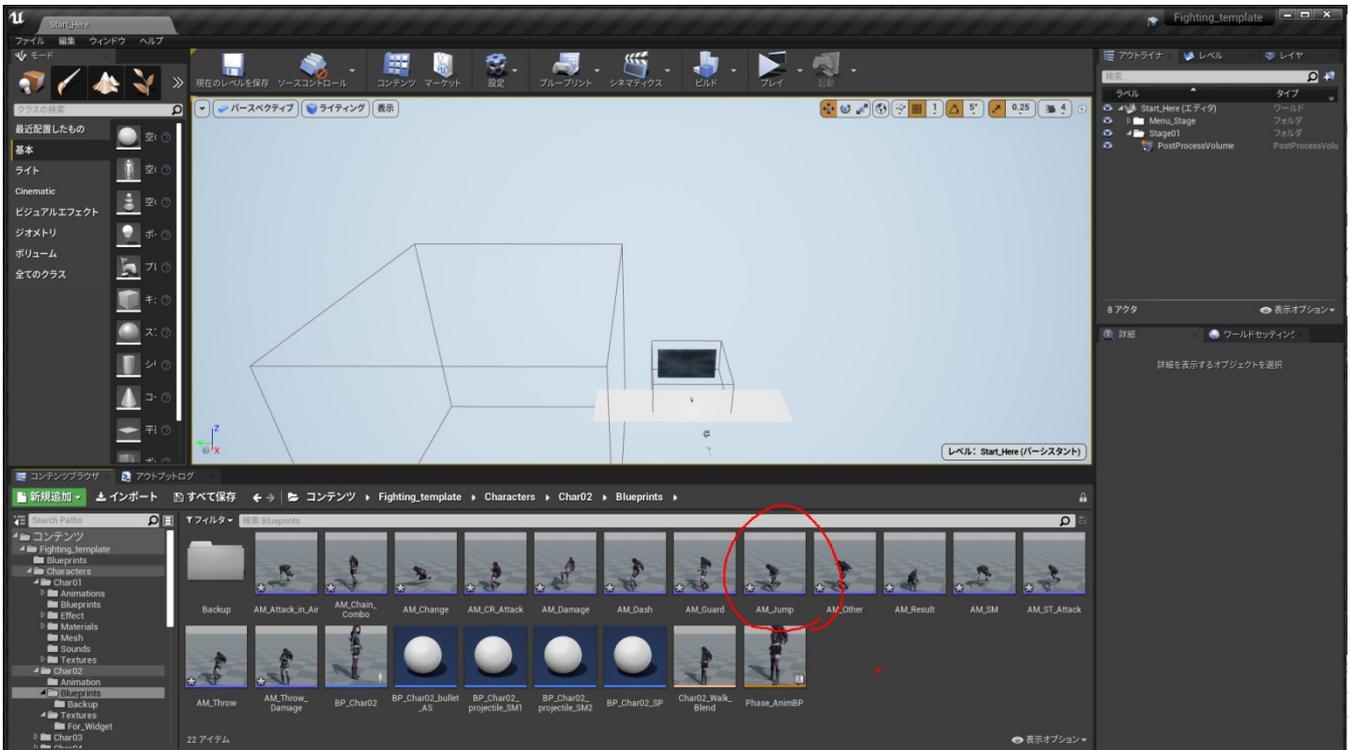


Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

右クリックして TimeRemaining と入力し、TimeRemaining(ratio)(入れ替えたアニメーションファイル名) の関数を選択してノードをつなぎ直します。

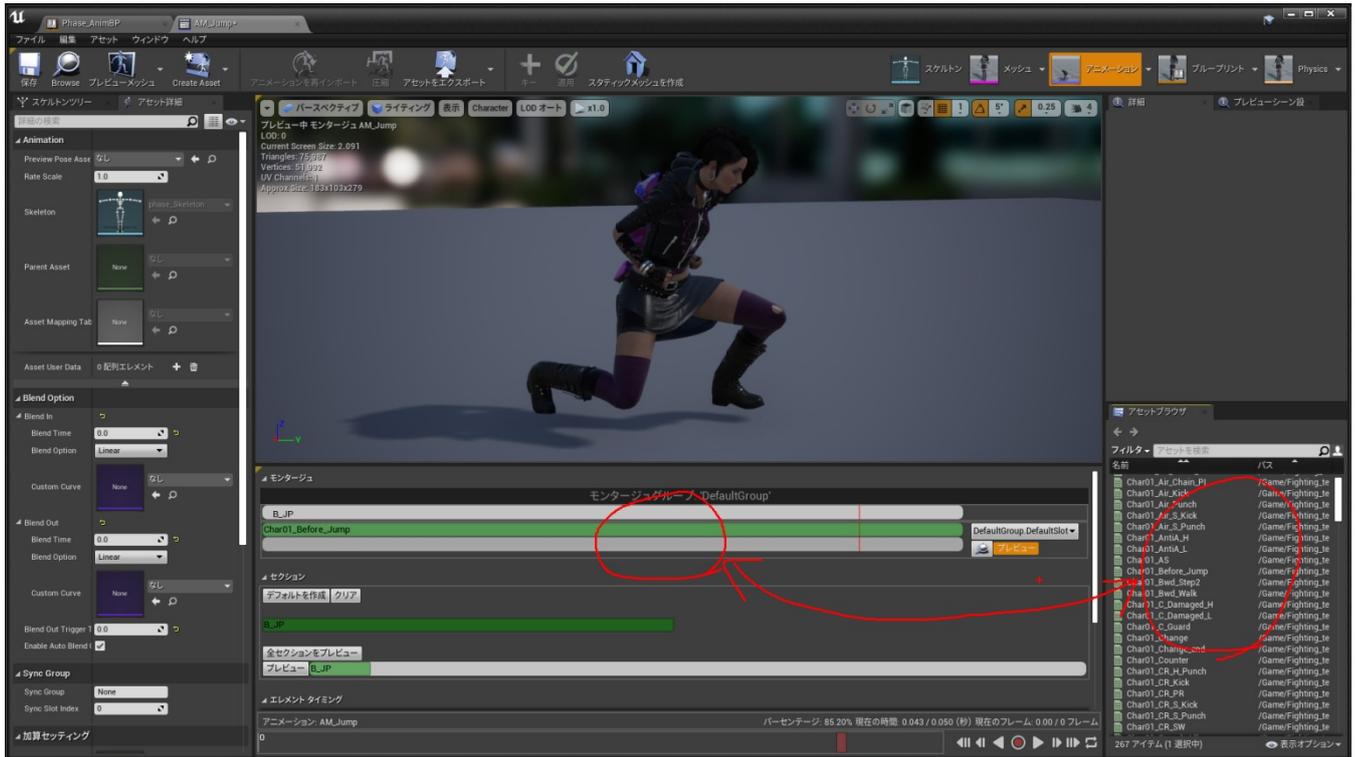


これで、ステートマシンの設定は完了です。コンパイルして保存しましょう。
それではここから、移動に関連する残りのアニメーションの設定を行います。
まずはアニメーションモーターズの AM_Jump を開きます。

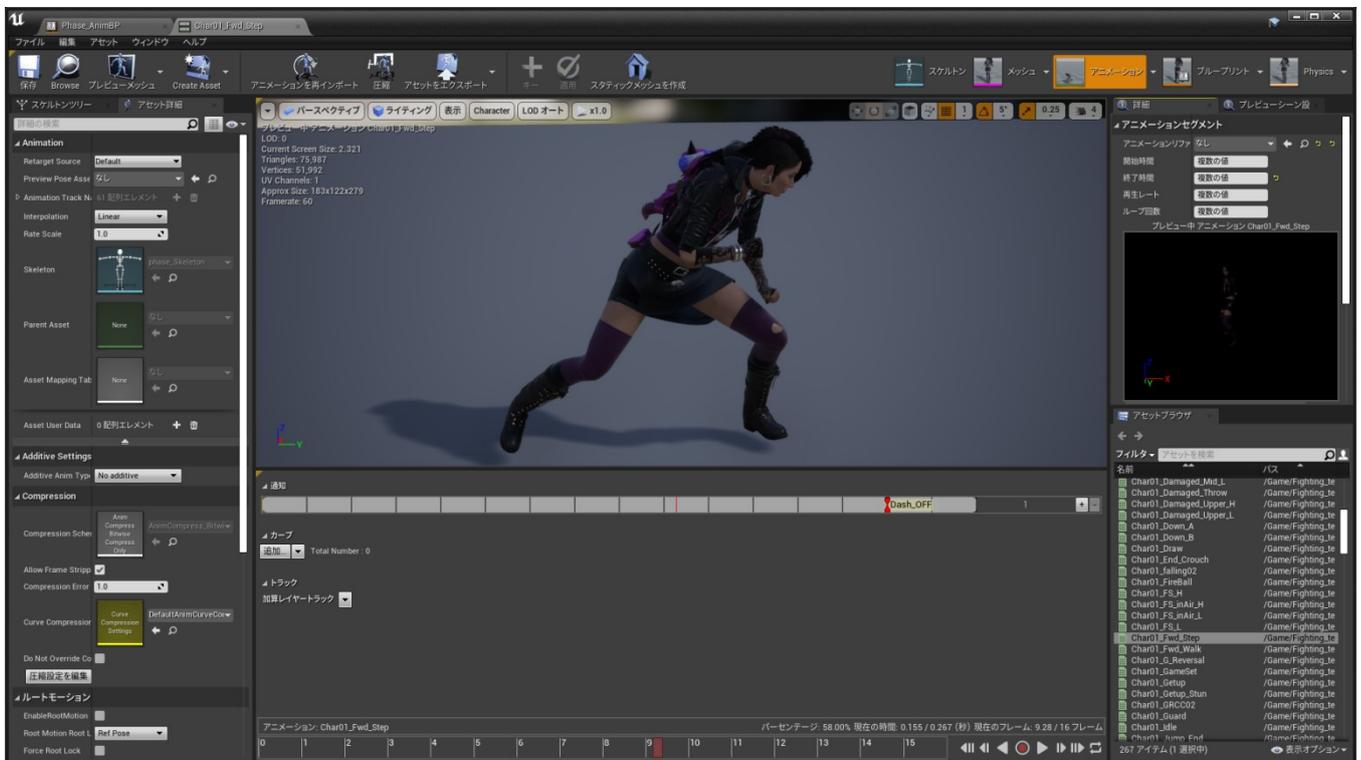


Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

ここではジャンプする直前にしゃがむ3フレームのアニメーションを登録していますので、必要に応じて入れ替えてください。このままでも大丈夫です。

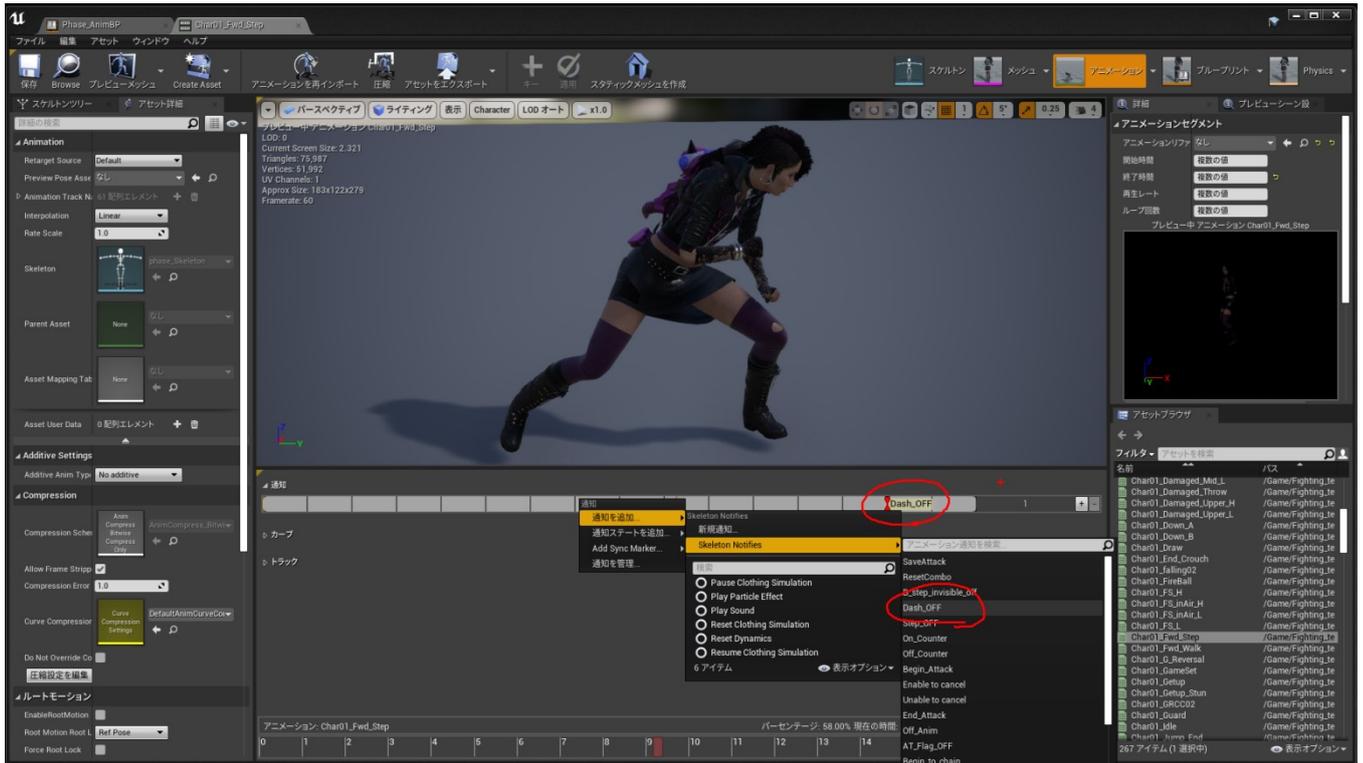


次に前ステップとバックステップのアニメーションを設定しますので、まずは前ステップに使用するアニメーションファイルを開きます。

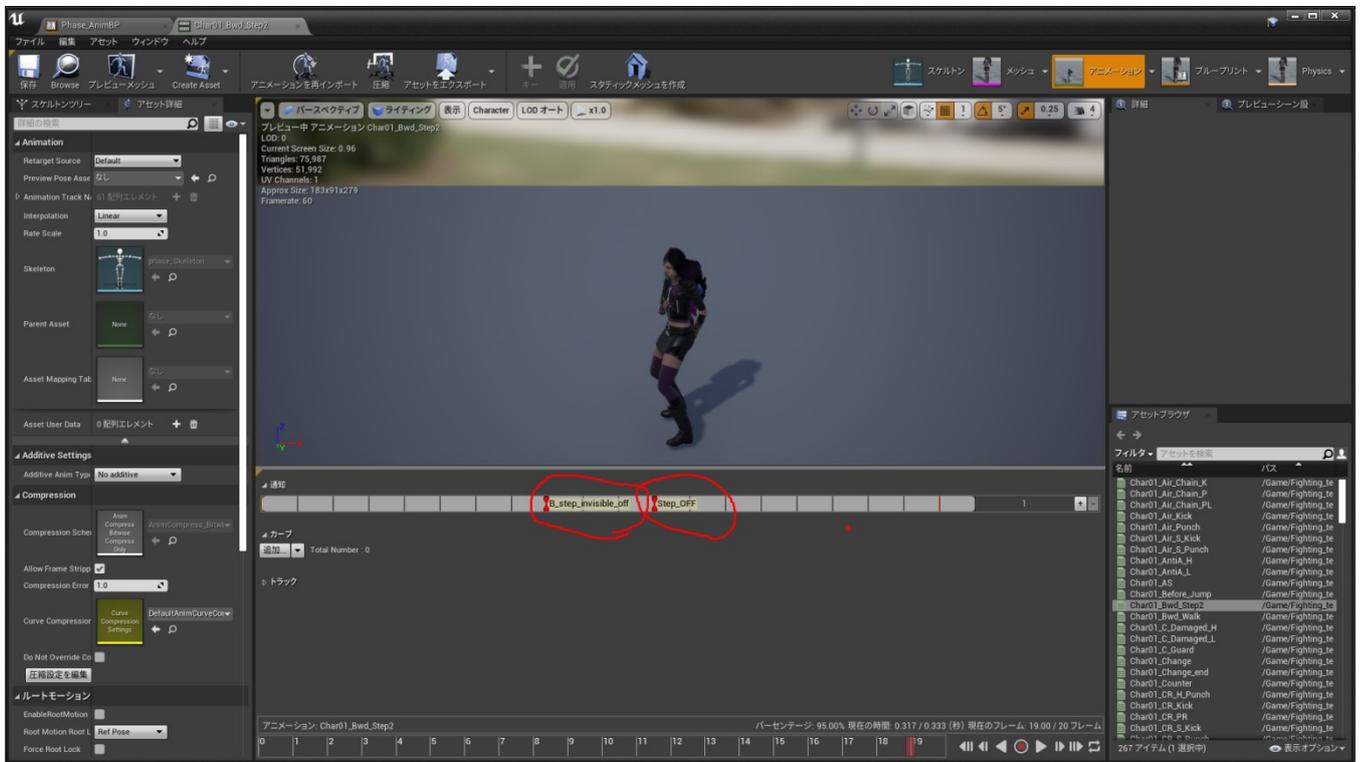


Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

開いたら通知のタイムスケールを右クリックして「通知の追加」→「Skeleton Notices」から Dash_OFF を選択してアニメーションの最後のフレーム付近に追加します。

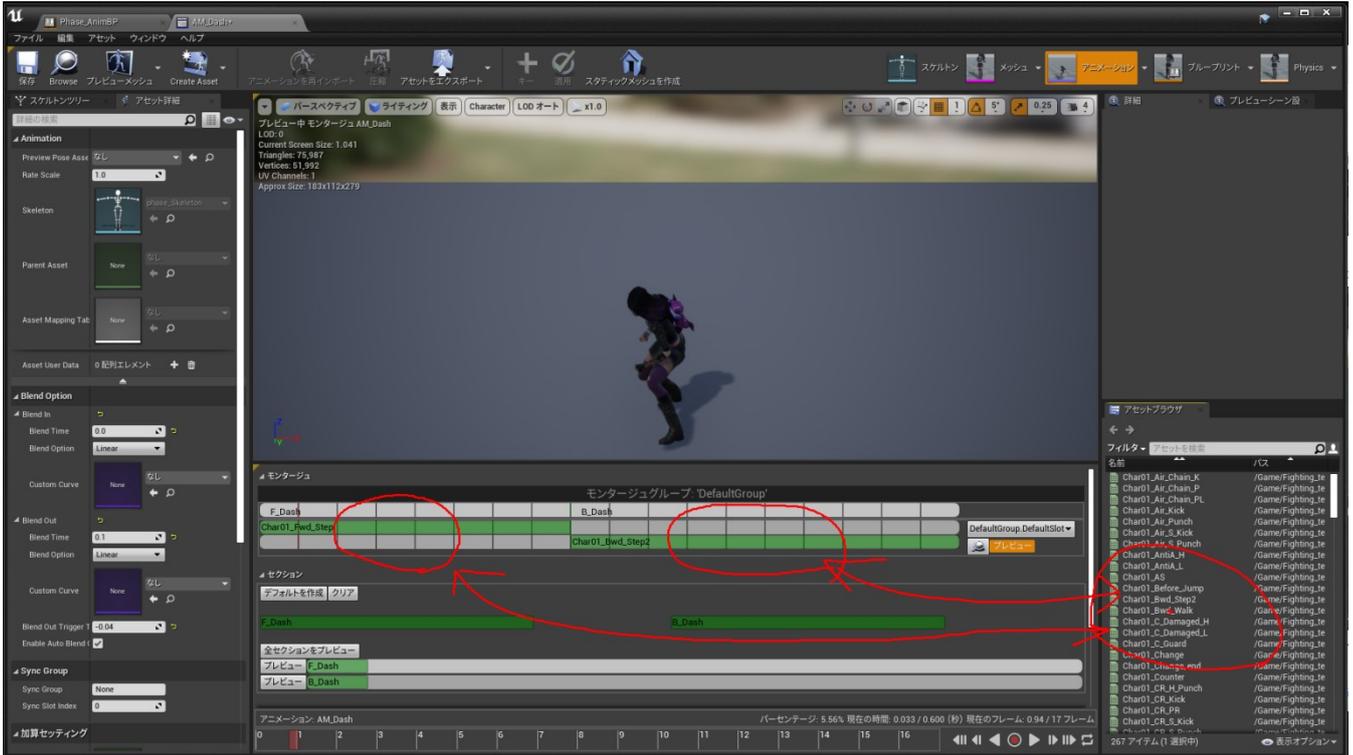


バックステップのアニメーションも同様に通知を追加しますが、バックステップはアニメーション開始から無敵になっているので、無敵を解除する通知「B_step_invisible_off」を追加して下さい。



Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

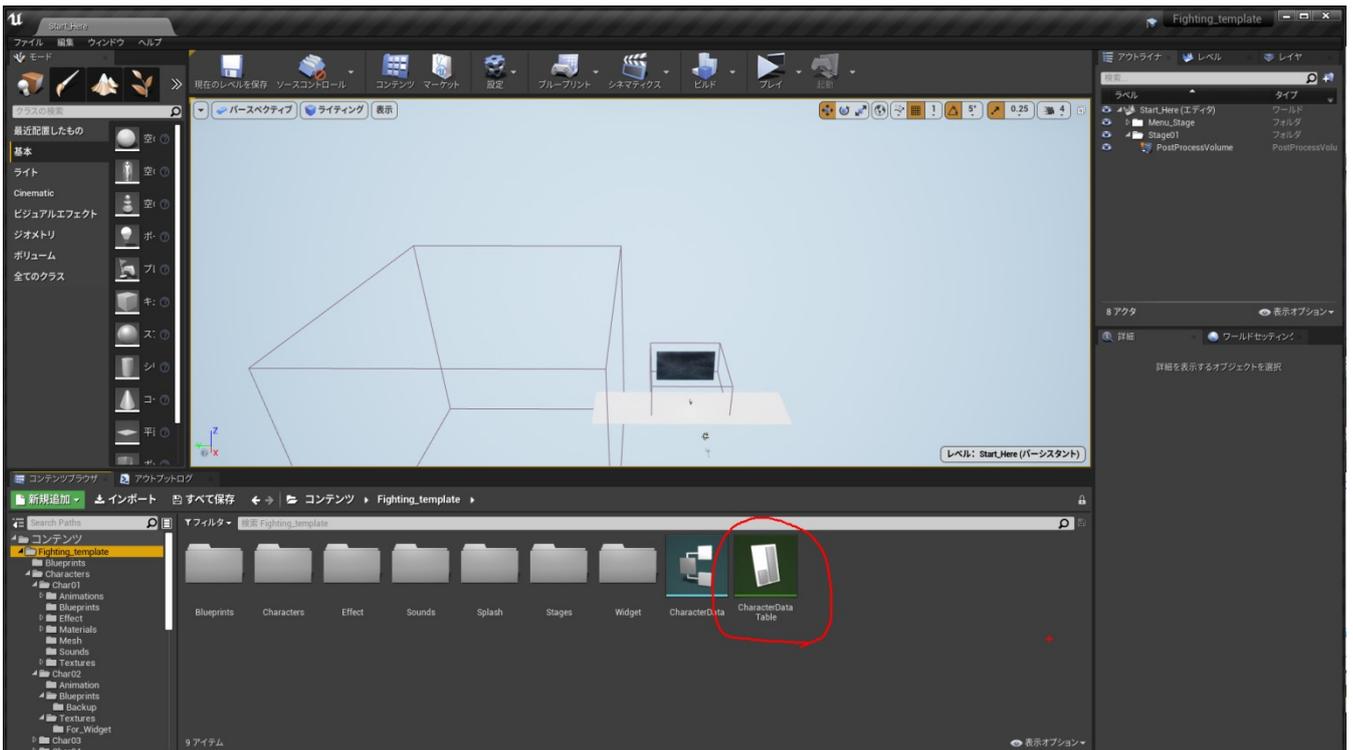
前後のステップのアニメーションに通知を追加したらアニメーションモータージュの AM_Dash を開いてアニメーションを入れ替えます。



入れ替えたときに B_Dash のセクション開始位置がずれる場合がありますので、バックステップのアニメーション開始に位置を合わせてください。

これで移動に関するアニメーションの設定が完了しましたので、キャラクターデータテーブルで HP や移動に関する数値の設定をします。

では、Fighting_template フォルダにある CharacterDataTable を開きます。

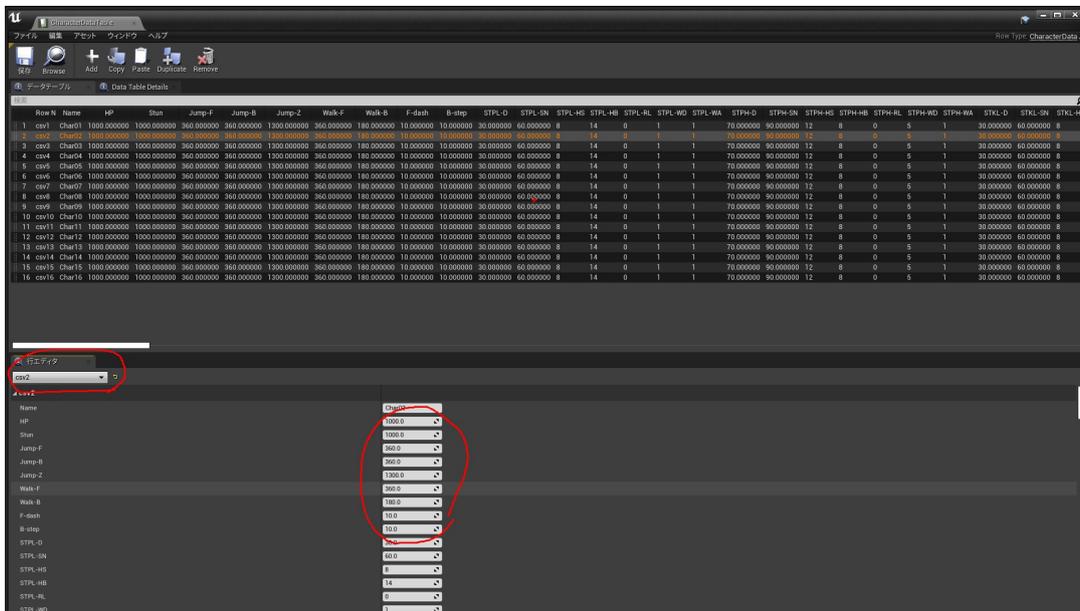


Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

行エディタのプルダウンメニューから変更するデータを選択しますが、今回は2番目のキャラクターデータを編集しますので、csv2を選択します。

それから以下の項目を設定します。

- ・ HP : ヒットポイント。標準は 1000。0 で敗北。
- ・ Stun : スタンポイント。標準は 1000。この値を超えるとスタンする。
- ・ Jump-F : 前ジャンプの横方向の移動スピード(cm/sec)。標準は 360。
- ・ Jump-B : バックジャンプの横方向の移動スピード(cm/sec)。標準は 360。
- ・ Jump-Z : ジャンプの上方向の初速スピード(cm/sec)。標準は 1300。
- ・ Walk-F : 前歩きの移動スピード(cm/sec)。標準は 360。
- ・ Walk-B : 後ろ歩き of 移動スピード(cm/sec)。標準は 180。
- ・ F-dash : 前ステップの移動スピード(cm/frame)。標準は 10。
- ・ B-step : バックステップの移動スピード(cm/frame)。標準は 10。



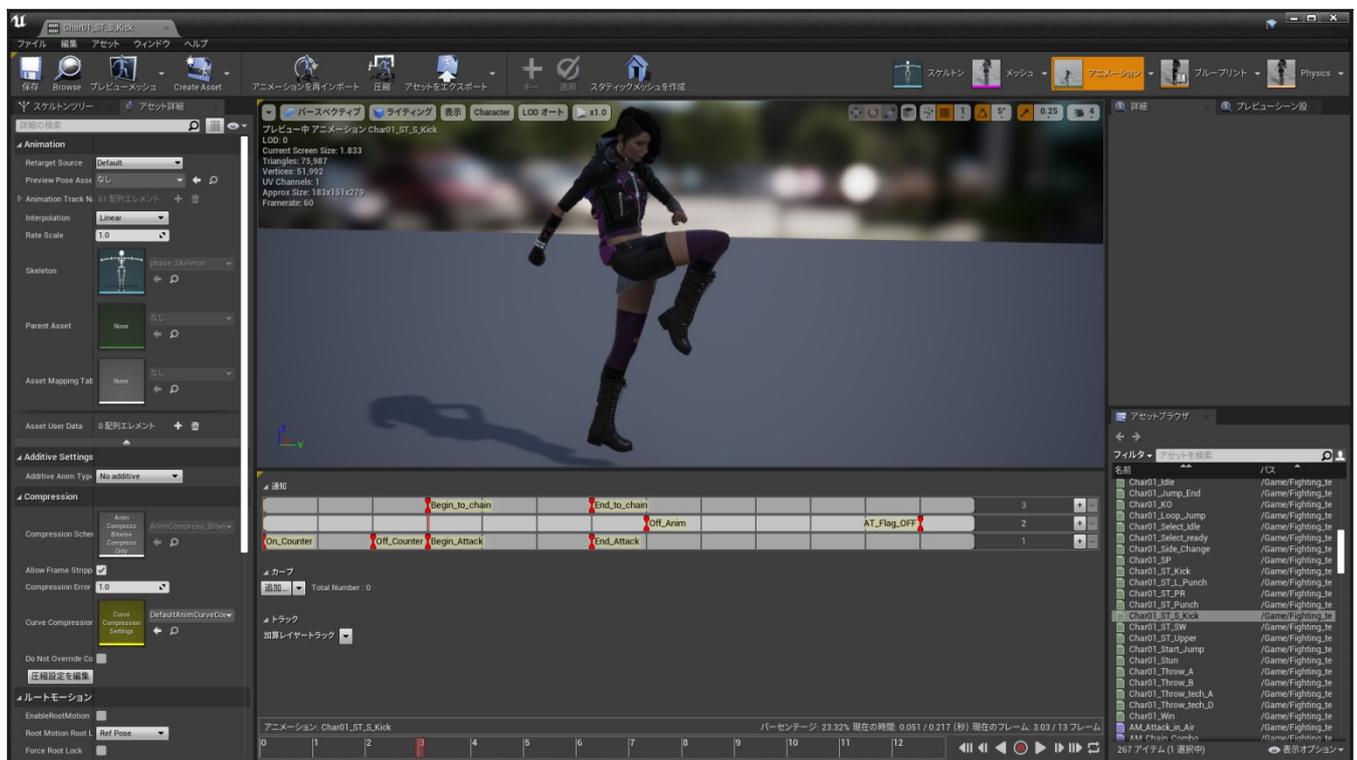
以上で移動に関する全ての設定が完了しましたのでゲームを起動して確認してみましょう。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

それでは、ここから通常攻撃のアニメーション入れ替えについて説明します。

攻撃アニメーションのファイルを開いたら、以下の通知を設定します。

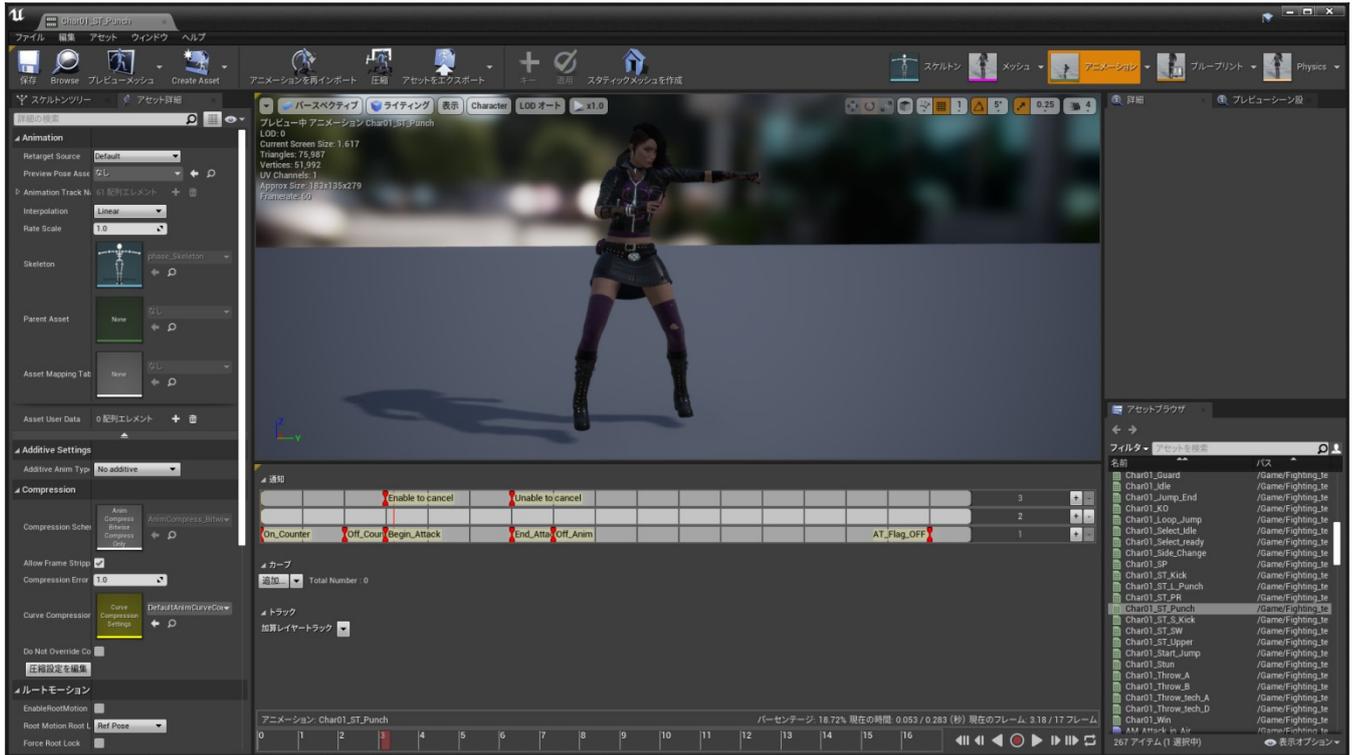
- On_Counter : 0 フレームの位置にセットします。
- Off_Counter : Counter_On から Off の間に攻撃を受けるとカウンターヒットになります。
- Begin_Attack : 攻撃判定のコリジョンボックスが On になります。
- End_Attack : 攻撃判定のコリジョンボックスが Off になります。
- Off_Anim : アニメーションフラグの処理通知です。End_Attack の 1 フレーム後に設定します。
- AT_Flag_OFF : 攻撃フラグの終了処理です。全体フレームの最後から 1 フレーム前に設定します。
- Begin_to_chain/End_to_chain : チェーンコンボとして使用するとき設定します。
Begin_Attack/End_Attack と同期して設定します。
- Enable_to_cancel/Unable_to_cancel : 必殺技をキャンセルで使用できるようにする場合は設定します。
Begin_Attack/End_Attack と同期して設定します。



今回の立ち状態の小キックの場合は、チェーンコンボの始動技として使用するので、Begin_to_chain/End_to_chain の通知が追加されています。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

もう一つ立ち状態の小パンチを見てみましょう。



こちらは、必殺技のキャンセルを可能にするために、Enable_to_cancel/Unable_to_cancelの通知を追加しています。

この設定を以下のアニメーションモニターに設定されているアニメーションの入れ替えで行います。ガードリバーサルとスウェー時のパリング攻撃も以下のモニターに含まれていますが、これは別途説明します。

- AM_ST_Attack : 立ち状態の攻撃アニメーションを登録しています。
必要に応じてチェーンコンボの始動技のアニメーションには Begin_to_chain/End_to_chain の通知を追加します。
- AM_CR_Attack : しゃがみ状態の攻撃アニメーションを登録しています。
必要に応じてチェーンコンボの始動技のアニメーションには Begin_to_chain/End_to_chain の通知を追加します。
- AM_Attack_in_Air : ジャンプ状態の攻撃アニメーションを登録しています。
チェーンコンボの始動技のアニメーションには Begin_to_chain/End_to_chain の通知を追加します。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

それではここから通常攻撃のキャラクターデータテーブルの設定方法を説明します。

まずは、通常攻撃の値を設定するデータテーブルの項目名称について説明します。

通常攻撃のデータ名称はこのようなアルファベットで表示されます。

STPL-D

ST : Standing
CR : Crouching
JP : Jumping

最初の2文字については3種類あり、STが立ち状態、CRがしゃがみ状態、JPがジャンプ状態を表します。

次の2文字はパンチやキックの種類と強度を表します。

ST**PL**-D

PL : Light Punch
PH : Heavy Punch
KL : Light Kick
KH : Heavy Kick

最後の1文字は7種類あり、ダメージ量やヒットストップ、ヒットバックのフレーム数、左右の攻撃種類、ダメージアニメーション、攻撃場所の指定を表します。

STPL-D

D: Damage	RL: Right or Left
SN : Stun damage	WD : Where damage
HS : Hit stop	WA : Where attack
HB : Hit Back	

ダメージ量やヒットストップ、ヒットバックの数値についてはデフォルトの値を参考に設定してください。
残りの項目については、詳細を説明します。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

まず、RL ですが、これは右と左のどちらで攻撃するのかを指定しますので、0が左、1が右となります。

STPL-RL

RL: Right or Left

0 : Left

1 : Right

次に WD はこの攻撃がヒットしたときのダメージアニメーションを決定します。

STPL-WD

WD : Where damage(Determining damage animation)

1: Light Upper

6: Heavy Lower

2: Light Middle

7: Light in Air

3: Light Lower

8: Heavy in Air

4: Heavy Upper

5: Heavy Middle

例としてこのようになります。

For Example

1: Light Upper

2: Light Middle

3: Light Lower



6番の下段の強攻撃のダメージアニメーションはダウンするアニメーションになります。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

次に WA はどこに攻撃するかを表します。

STPL-WA

WA : Where attack

- 1: Upper (Standing and crouching guard possible)
- 2: Middle (Crouching guard impossible)
- 3: Lower (Standing guard impossible)

1 番は上段攻撃で立ちとしゃがみのガードが可能、2 番は中段攻撃でしゃがみガードが不可能、3 番は下段攻撃で立ちガードが不可能となります。

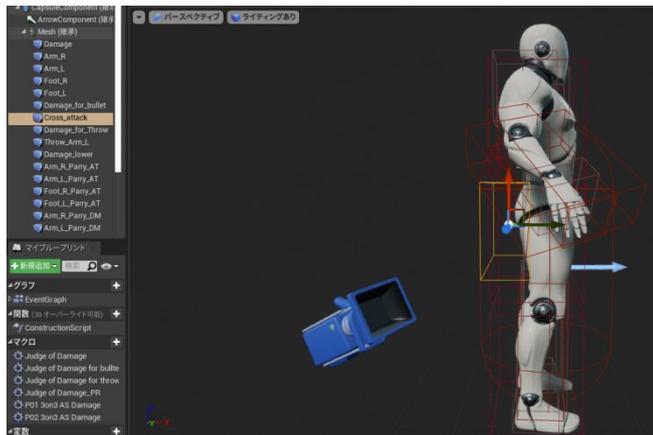
例えば立ち状態の小パンチで全ての項目を設定するとこのようになります。



STPL-D	STPL-SN	STPL-HS	STPL-HB	STPL-RL	STPL-WD	STPL-WA
30 000000	60 000000	8	14	0	1	1

ダメージ量が 30、スタンダメージが 60、ヒットストップが 8 フレーム、ヒットバックが 14 フレーム、RL は左手なので 0、WD は上段の小ダメージなので 1、WA は上段攻撃なので 1 となります。

また、ジャンプ攻撃については、Cross-up という項目があり、これにチェックを入れるとめくり攻撃用の攻撃判定が ON になります。



Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

では次にチェーンコンボについて設定方法を説明しますが、チェーンコンボはデータテーブルから説明します。

まずは、地上のチェーンコンボについて種類を決定します。

CCGR-TYPE

The type of the Chain-combo on the ground

1: KL→PL→PH	5: KL→PL→PH→KH
2: PL→KL→PH	6: KL→PL→KH→PH
3: KL→PL→KH	7: PL→KL→PH→KH
4: PL→KL→KH	8: PL→KL→KH→PH

3連か4連コンボにするのか決定すると、ボタンを押す順番を決定します。

そして青枠の1撃目については、通常攻撃に登録しているアニメーションになりますので、AM_ST_AttackかAM_CR_Attackに登録しているアニメーションにBegin_to_chain/End_to_chainの通知を追加してください。

デフォルトキャラクターはAM_ST_Attackの小キックのアニメーションに追加しています。

さらに赤枠の攻撃については方向キー上を押した場合は、エアリアルコンボの始動技となります。

それでは次に、2撃目以降のデータテーブルについて説明します。

2撃目以降は、通常攻撃と同じ項目を設定します。

CCGR02-D

D: Damage

RL: Right or Left

SN : Stun damage

WD : Where damage

HS : Hit stop

WA : Where attack

HB : Hit Back

デフォルトキャラクターは地上チェーンコンボのタイプを1番で設定しているので4撃目のCCGR04の値は設定されていません。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

RL では、両手足どれを使って攻撃するのか指定します。

CCGR02-RL

RL: Right or Left

- 0: Left hand
 - 1: Right hand
 - 2: Left foot
 - 3: Right foot
-

WD では、どのダメージアニメーションを適用するのか指定します。

CCGR02-WD

WD : Where damage(Determining damage animation)

- | | |
|-----------------|-----------------|
| 1: Light Upper | 6: Heavy Lower |
| 2: Light Middle | 7: Light in Air |
| 3: Light Lower | 8: Heavy in Air |
| 4: Heavy Upper | |
| 5: Heavy Middle | |
-

WA ではどこに攻撃するのか指定します。

CCGR02-WA

WA : Where attack

- 1: Upper (Standing and crouching guard possible)
 - 2: Middle (Crouching guard impossible)
 - 3: Lower (Standing guard impossible)
-

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

さらに3撃目は方向キー上を押した場合に、エアリアルコンボの始動技となりますが、データテーブルでは別項目として設定します。

CCGRtoAir-D

D: Damage RL: Right or Left
SN : Stun damage WD : Where damage
HS : Hit stop WA : Where attack

3撃目とエアリアルコンボ始動技のアニメーションが同じであれば、同じ値を設定しても大丈夫ですが、ヒットバックは固定値となっているのでデータテーブルでの設定は必要ありません。

WDについてはエアリアルコンボへつながるダメージアニメーションを使用するので11番を選択してください。

CCGRtoAir-WD

WD : Where damage(Determining damage animation)

- | | |
|-----------------|---------------------|
| 1: Light Upper | 6: Heavy Lower |
| 2: Light Middle | 7: Light in Air |
| 3: Light Lower | 8: Heavy in Air |
| 4: Heavy Upper | 11: to Aerial Combo |
| 5: Heavy Middle | |

次に、エアリアルコンボの種類を決定します。

CCAir-TYPE

The type of the Chain-combo in the Air

1: KL→PL→PH→KH
2: KL→PL→KH→PH
3: PL→KL→PH→KH
4: PL→KL→KH→PH

青枠の1 撃目については、通常攻撃に登録しているアニメーションになりますので、

AM_Attack_in_Air に登録しているアニメーションに Begin_to_chain/End_to_chain の通知を追加してください。

デフォルトキャラクターは AM_Attack_in_Air の小キックのアニメーションに通知を追加しています。

それでは次に、2 撃目以降のデータテーブルについては、ヒットバックや WA や WD が固定値となっているので、以下の項目だけ設定してください。

CCAir02-D

D: Damage

RL: Right or Left

SN : Stun damage

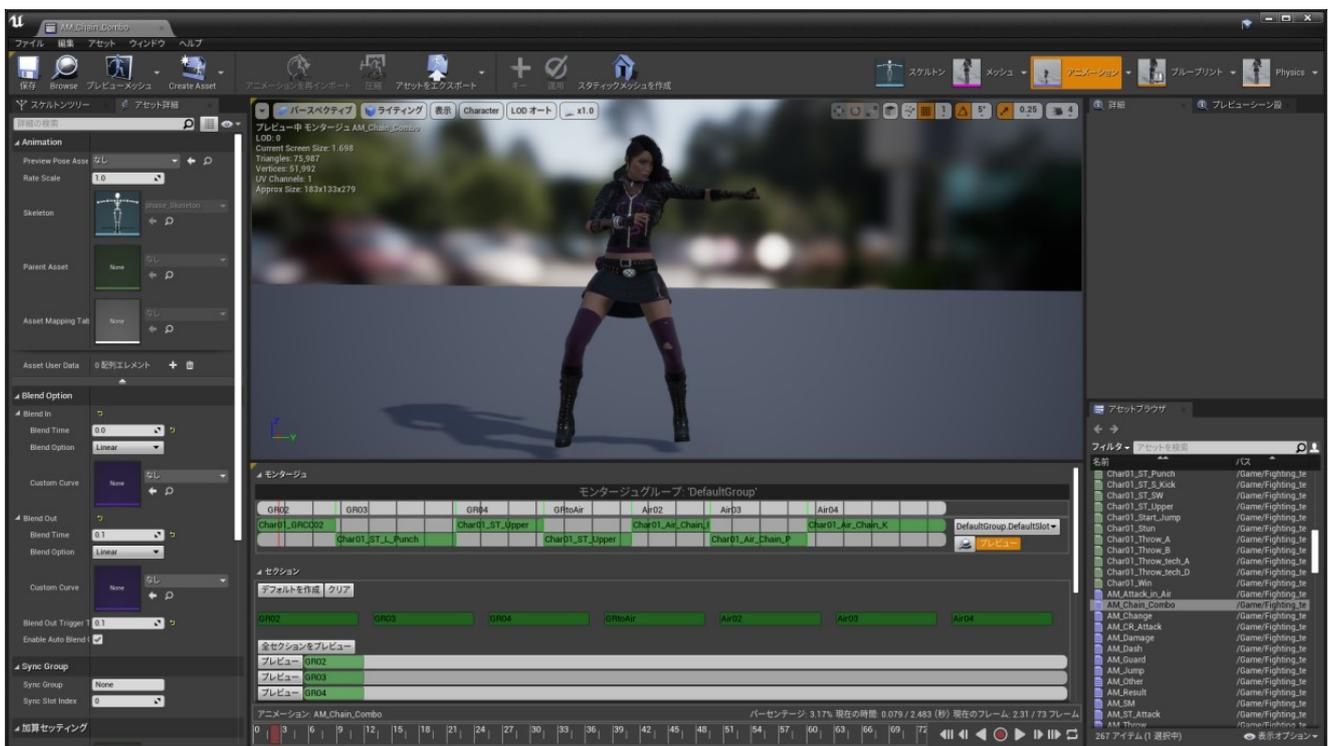
HS : Hit stop

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

それでは、アニメーション登録について説明します。

チェーンコンボのアニメーションは、AM_ChainCombo に登録しており、以下のセクションが設定されています。

- ・ GR02 : 地上のチェーンコンボ 2 撃目
- ・ GR03 : 地上のチェーンコンボ 3 撃目
- ・ GR04 : 地上のチェーンコンボ 4 撃目
- ・ GRtoAir : 地上から空中へ打ち上げる技
- ・ Air02 : 空中のチェーンコンボ 2 撃目
- ・ Air03 : 空中のチェーンコンボ 3 撃目
- ・ Air04 : 空中のチェーンコンボ 4 撃目



これらのセクションに登録するアニメーションはチェーンコンボ専用になりますが、既に別のアニメーションモニターグループに登録しているアニメーションをチェーンコンボとして使用する場合は、通知が異なる場合があるのでアニメーションファイルをコピーして、別のファイル名にしてからこのセクションに登録します。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

デフォルトキャラクターは2撃目に立ち状態の小パンチアニメーションをコピーして、ファイル名を変更して登録しています。



このように Begin_to_chain/End_to_chain の通知が追加されています。

3撃目はこのようなアニメーション通知となっています。



デフォルトキャラクターの場合は、通知に変更の必要がなかったので、既存のアニメーションを登録しています。

チェーンコンボの最後の攻撃アニメーションには、Enable&Unable to cancel の通知を追加しておくことでキャンセルで必殺技が繋がります。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

次にエアリアルコンボの始動技の通知はこの様になっていますが、空中で相手と密着できるようにステップインの通知の追加が必要になります。



次にエアリアルコンボの2 撃目のアニメーション通知です。



このように Begin_to_chain/End_to_chain の通知が追加されています。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

次にエアリアルコンボの3撃目のアニメーション通知です。



ここからは、必殺技のキャンセルができるように、Enable&Unable to cancel の通知も追加しています。

4撃目も同様となり、チェーンコンボのアニメーション設定はこれで完了です。

さらに、その他通知の追加やデータテーブル編集が必要なアニメーションについて説明します。

・投げのデータテーブル設定

TH-D

D: Damage

SN : Stun damage

RL: Right or Left

0: Left hand

1: Right hand

2: Left foot

3: Right foot

投げのダメージ量や両手足のどれで投げをするのか指定します。

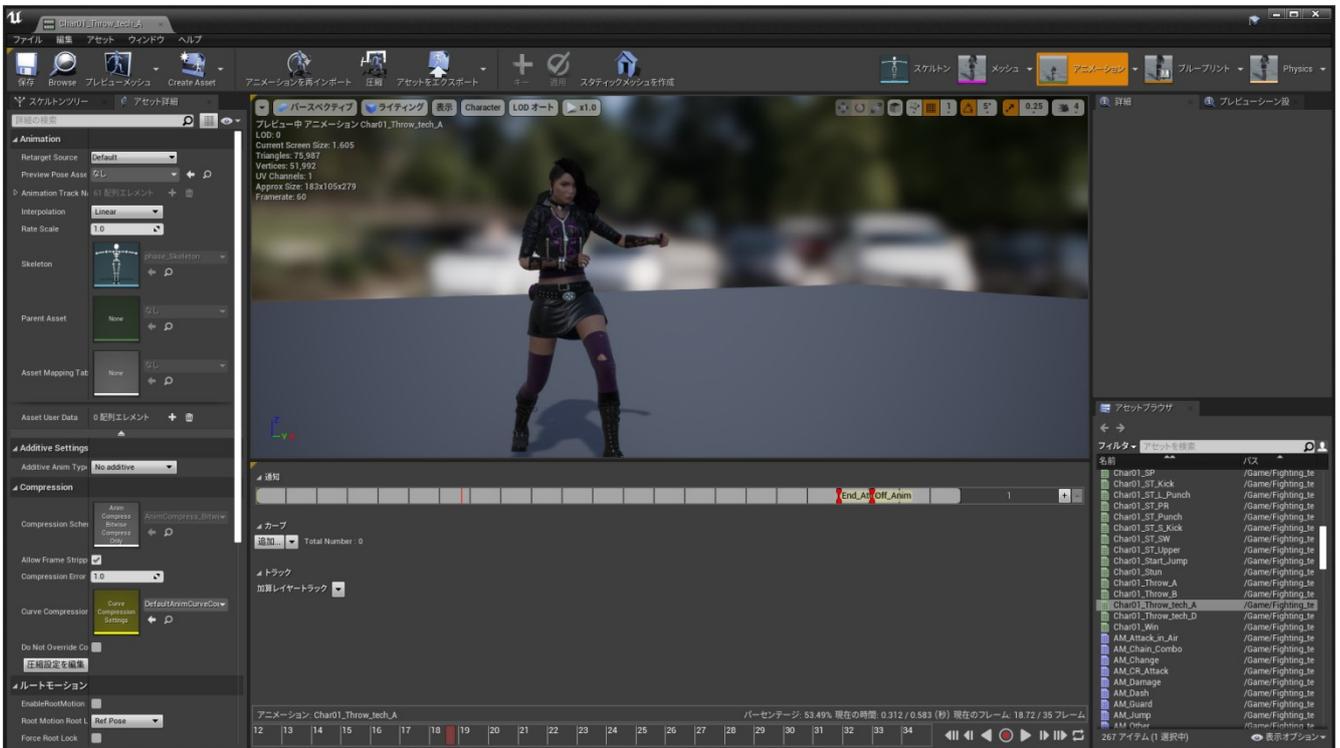
Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

・投げの通知設定



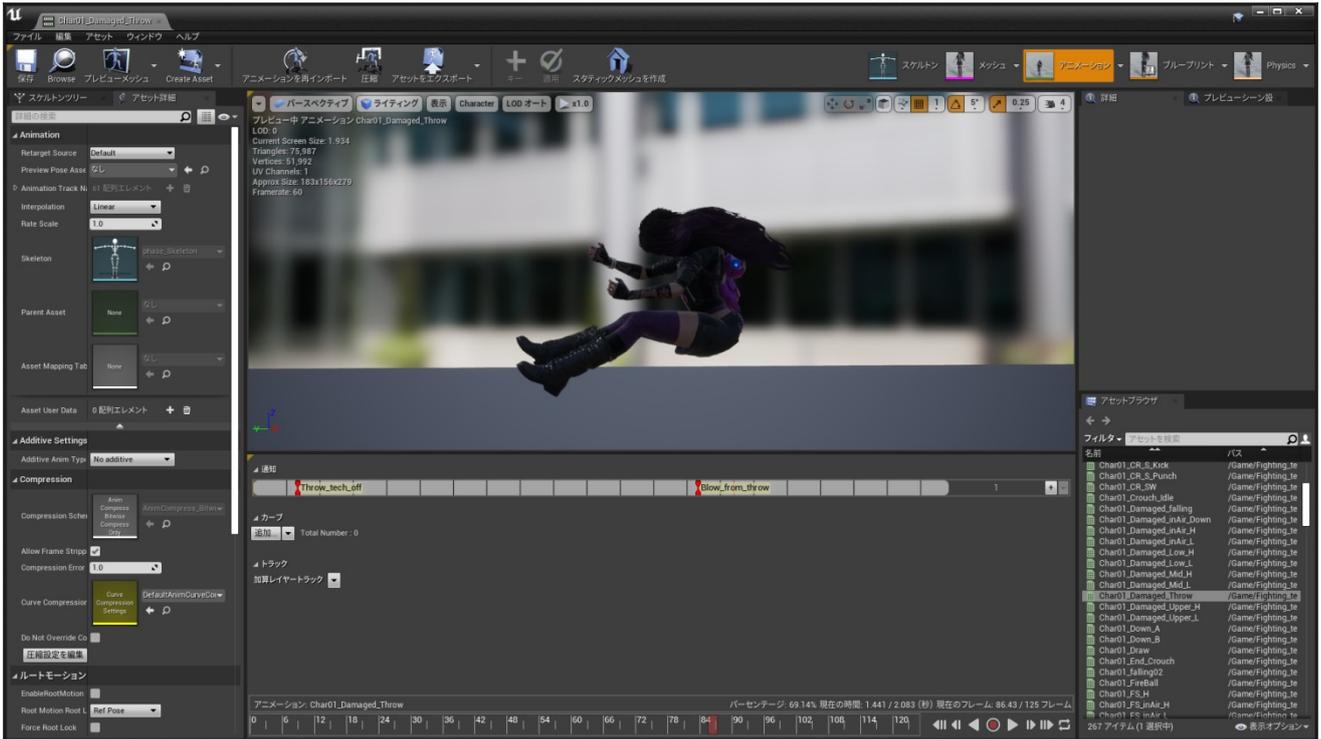
On&Off_Counter/Begin&End_Attack/Off_Anim/AT_Flag_OFF の通知を設定します。

・投げ抜きの通知設定



攻撃側、守備側ともに End_Attack/Off_Anim の通知を追加しています。

・投げダメージの通知設定



投げ抜けを off にする Throw_tech_off の通知を 8 フレーム目に追加します。

必要に応じて投げのアニメーションで打撃がある場合は、Blow_from_throw の通知を追加すると後方へ飛ぶ処理が発生します。

・パリングやガードリバーサル củaデータテーブル設定

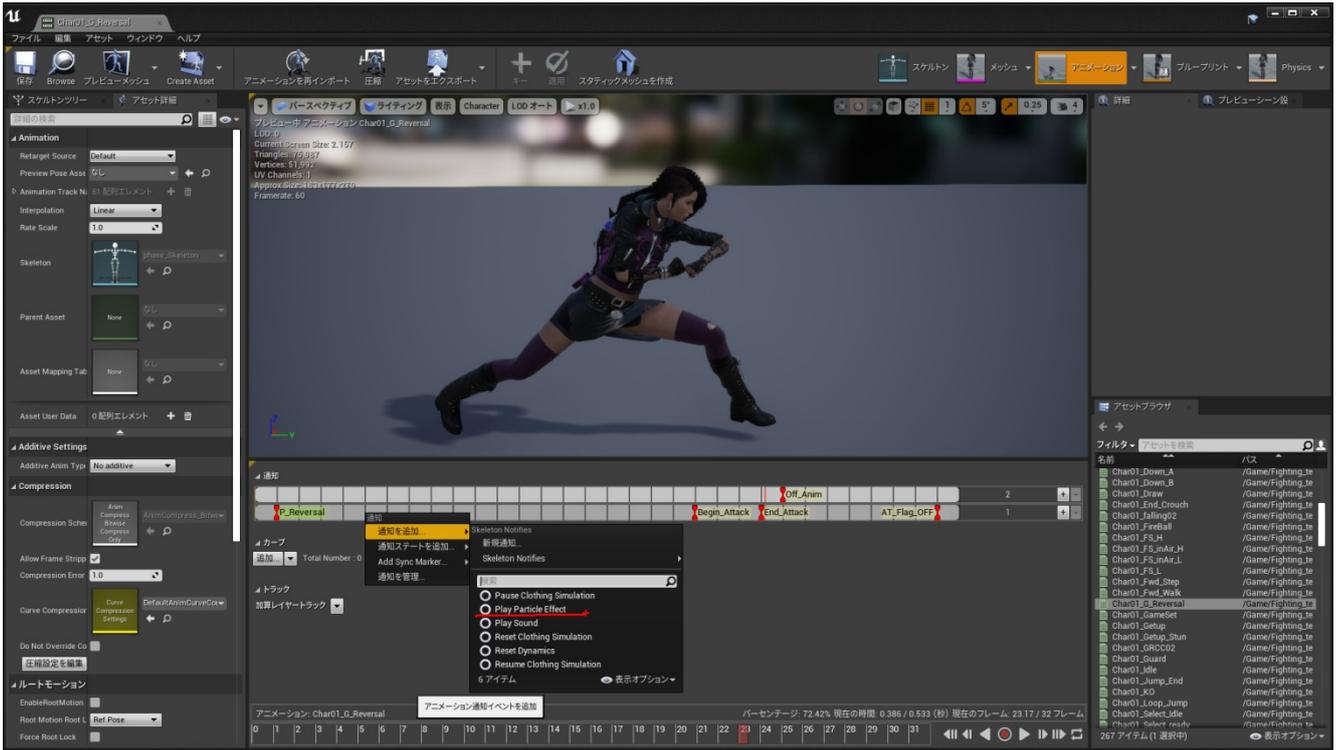
UPR-RL

- | | |
|----------------------------|-------------------|
| UPR: Parrying of the upper | RL: Right or Left |
| LPR: Parrying of the lower | 0: Left hand |
| GR: Guard Reversal | 1: Right hand |
| | 2: Left foot |
| | 3: Right foot |

上下段のパリングやガードリバーサルについて両手足のどれか指定します。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

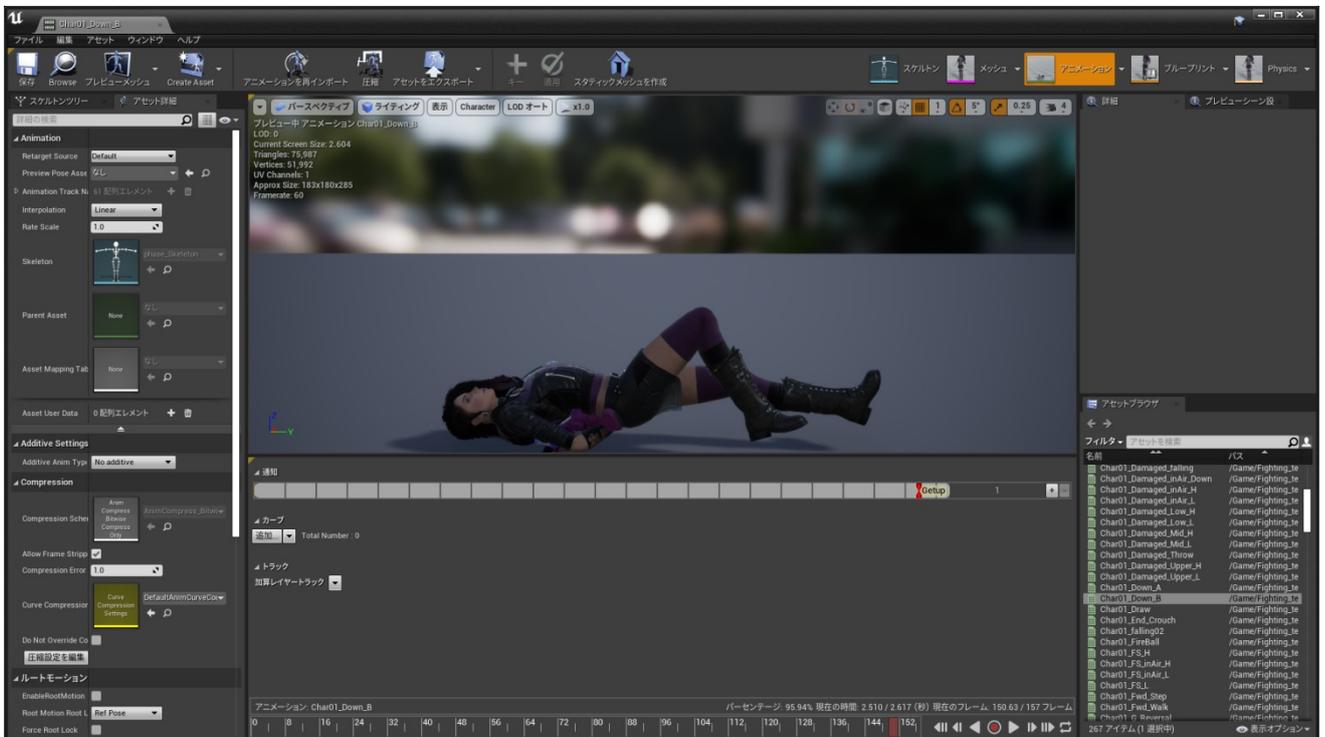
・ガードリバーサルのお知らせ設定



Play Particleの通知を使用してエフェクトを表示させています。

他にも通常の攻撃に必要な通知を追加しています。

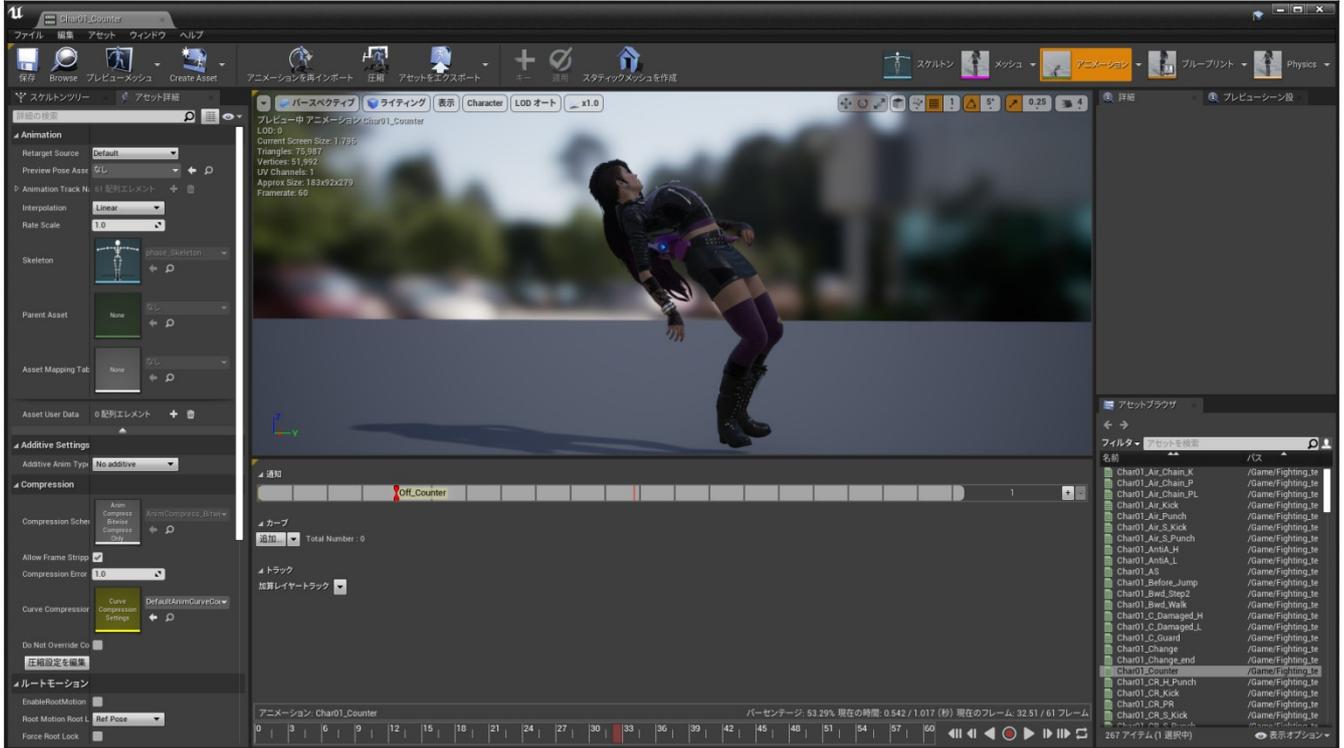
・ダウンの通知設定



150 フレーム目に Getup の通知を追加します。これで自動的に起き上がる処理が発生します。

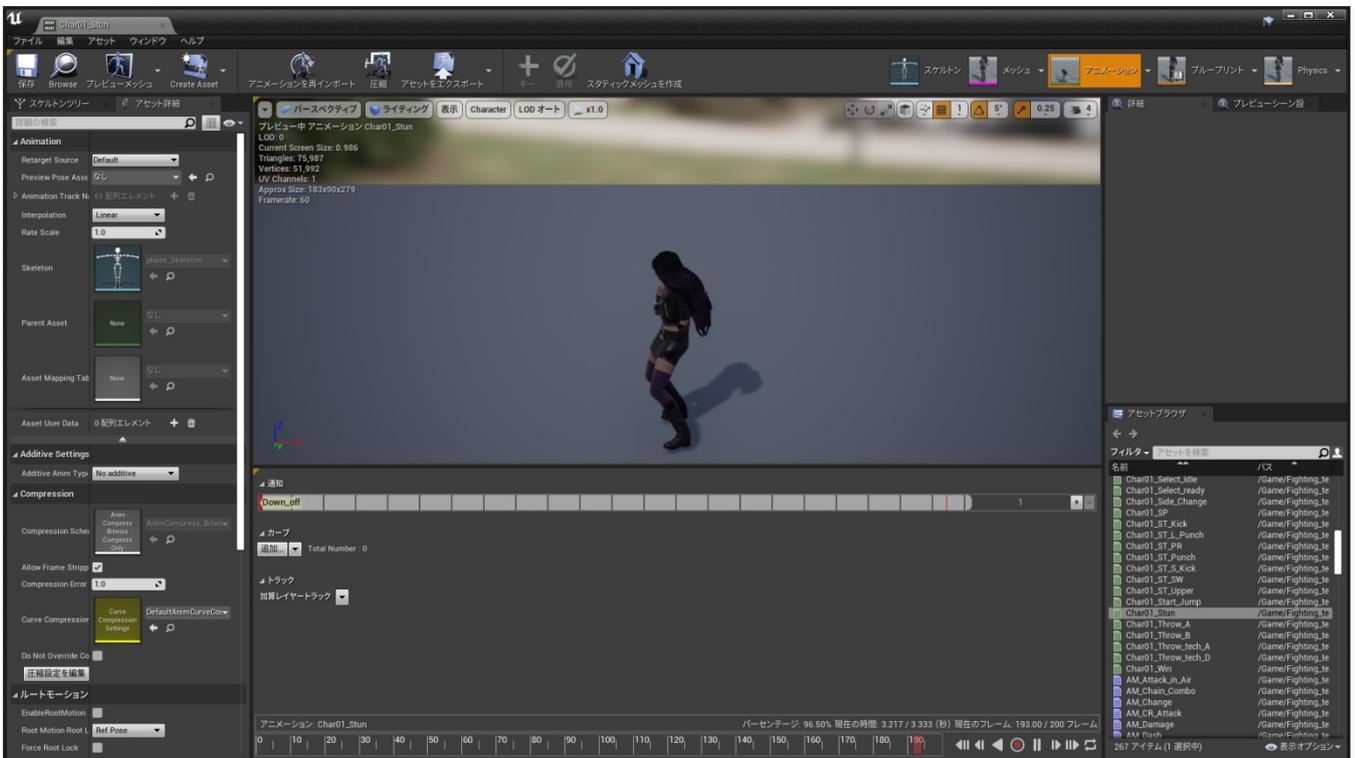
Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

・カウンターダメージの通知設定



カウンターフラグを Off にする通知を追加します。

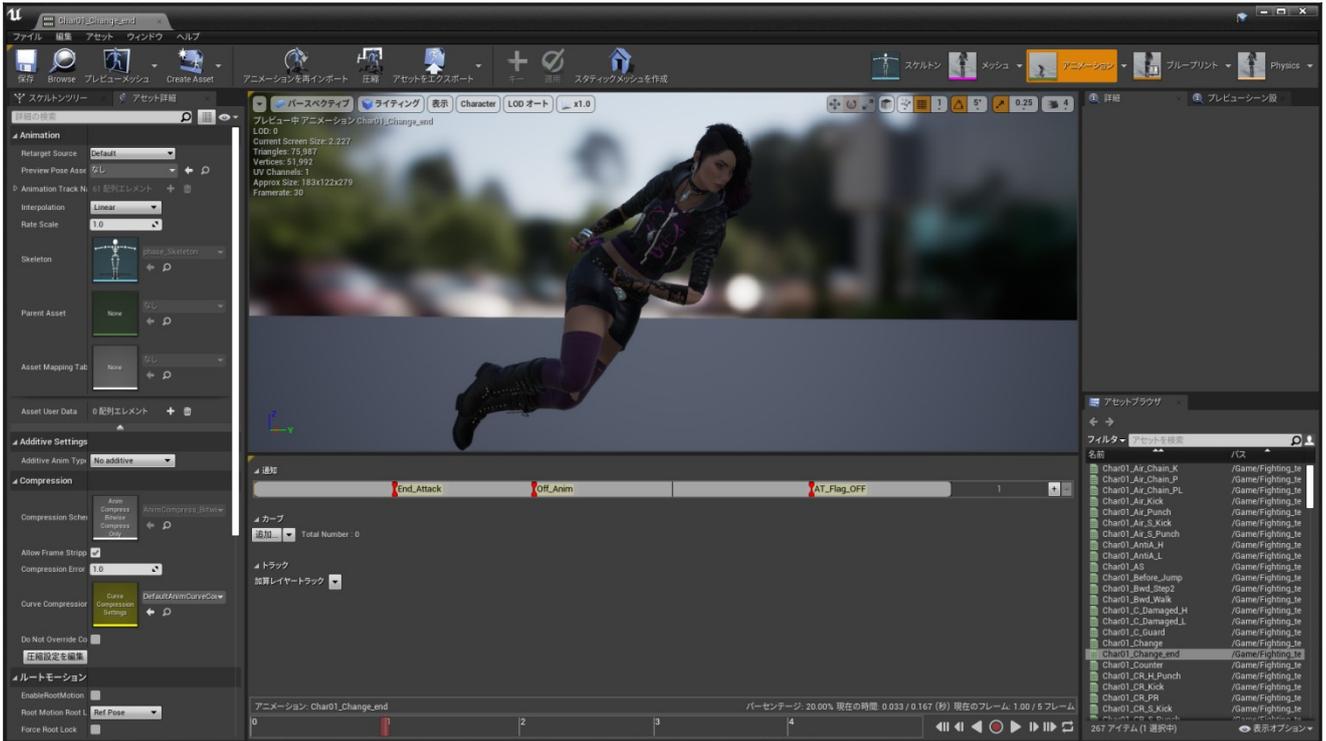
・スタンの通知設定



最初のフレームにダウン状態を解除する Down_off の通知を追加しています。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

- ・ 3on3 の交代時に使用するショルダーチャージの通知設定



End_Attack/Off_Anim/AT_Flag_OFF の通知を設定します。

それでは次に必殺技のアニメーション設定方法を説明します。

まずはデータテーブルについて、以下の項目を設定します。

SM1-ON

ON: Use or not WD: Where damage
Air : Use in Air WA: Where attack
Projectile : Use or not CM: Kind of command
RL: Right or Left or Projectile
Multiple-hit-times:Multiple-hit of projectile

必殺技は SM1 から SM5 まで 5 種類登録できますので、使用しない場合は、ON のチェックボックスを OFF にします。

必殺技を飛び道具にする場合は、Projectile のチェックボックスを ON にします。

ただし、飛び道具は SM1 と SM2 のみ登録可能です。

次に必殺技を空中で使用する場合は、Air のチェックボックスを ON にします。

ただし、ON すると空中でのみ使用可能になるので、例えば飛び道具を地上と空中で使用するには SM1 を地上用で設定して、SM2 で空中用を設定する必要があります。

Multiple-hit-times は飛び道具の強攻撃を多段ヒットにしたい場合、数値を入力します。

デフォルトは 1 になります。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

次に RL については、両手足どれで攻撃するのか指定するのと、飛び道具であれば 4 を設定します。

SM1-RL

RL: Right or Left or projectile

- 0: Left hand
- 1: Right hand
- 2: Left foot
- 3: Right foot
- 4: Projectile

次に WD については、飛び道具であれば 9 番、その他必殺技であれば 10 番を選択します。

ただし、10 番はダウンアニメーションになるので、ダウンさせたくない場合は、4 や 5 を設定します。

SM1-WD

WD : Where damage(Determining damage animation)

- | | |
|-----------------|------------------|
| 1: Light Upper | 6: Heavy Lower |
| 2: Light Middle | 7: Light in Air |
| 3: Light Lower | 8: Heavy in Air |
| 4: Heavy Upper | 9: Projectile |
| 5: Heavy Middle | 10: Special Move |

次に WA については、通常攻撃と同様です。

SM1-WA

WA : Where attack

- 1: Upper (Standing and crouching guard possible)
- 2: Middle (Crouching guard impossible)
- 3: Lower (Standing guard impossible)

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

次にコマンドについては、以下のコマンドリストから選択します。

Hは1秒以上ホールドが必要であることを意味します。

SPバー満タンで使用できる超必殺技のコマンドも含まれています。

SM1-CM

CM: Kind of command

1: 236+P	7: 4H6+P	13: 214214+P
2: 214+P	8: 4H6+K	14: 214214+K
3: 623+P	9: 2H8+P	15: 4H646+P
4: 236+K	10: 2H8+K	16: 4H646+K
5: 214+K	11: 236236+P	
6: 623+K	12: 236236+K	

H: Hold for 1 second or more

次に SM1 から SM5 の必殺技には弱と強の攻撃があるので、以下の項目を設定します。

飛び道具の速度を設定する以外は通常攻撃と同様です。

SM1L-D

L or H: Light or Heavy attack

D: Damage

SN : Stun Damage

HS: Hit stop

HB: Hit back

Projectile-speed : Setting speed

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

必殺技は AM_SM のアニメーションモンタージュに登録されており、ここからアニメーションの通知の設定について説明します。

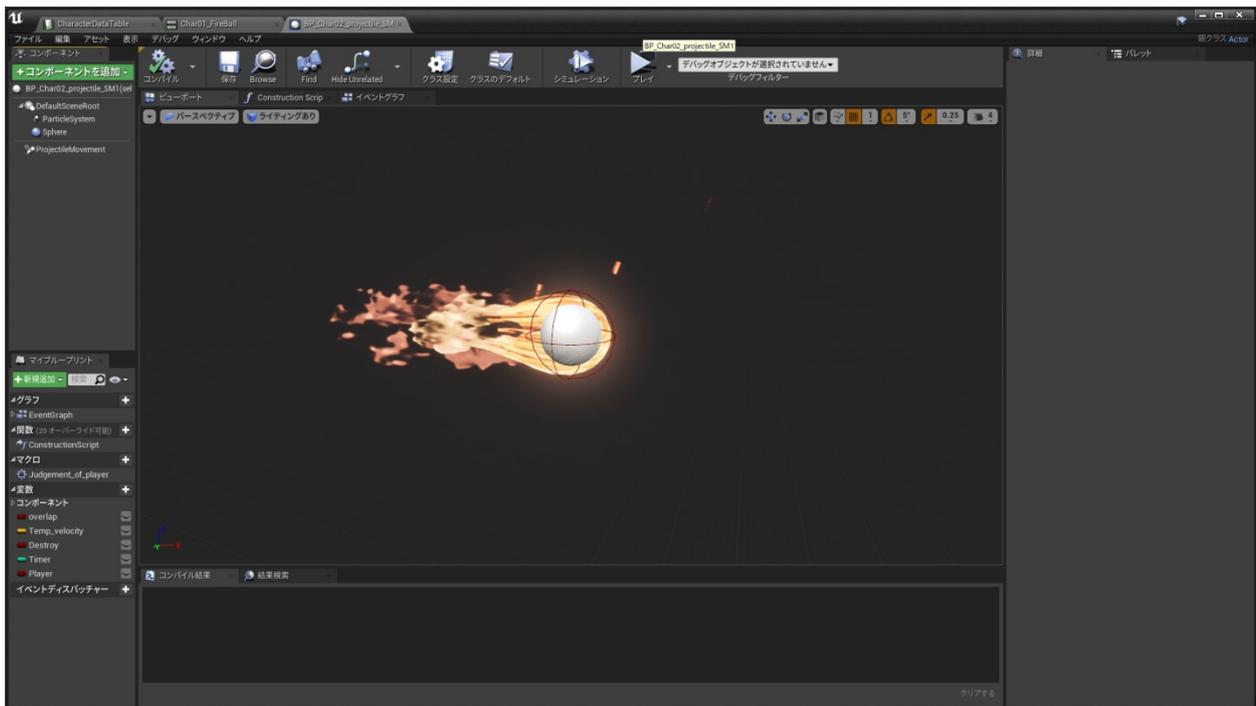
まずは、飛び道具の必殺技の通知になります。



On&OFF Counter でカウンターヒットが発生する通知と Shot bullet で飛び道具を出す通知を追加します。後は全体フレームの最後付近に Off_Anim と AT_Flag_off の通知を設定します。

通知の Shot bullet が SM1 で実行されると、BP_Char0x_SM1 がスポンされ、SM2 で実行されると BP_Char0x_SM2 をスポンします。

飛び道具の見た目を変更したい場合は BP_Char0x_projectile_SM1 の左側のコンポーネントタグにある ParticleSystem を変更してください。



Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

では次に突進の必殺技の通知設定を説明します。



通常技の設定でも使用した、On&Off Counter / Begin&End Attack / Off_Anim / AT_Flag_Off の他にエフェクトを表示させる通知や突進させるための SM_Step-in&out の通知を追加しています。この SM_Step-in&out の通知は設定されたフレームの間、前ステップ同じ速度で突進します。

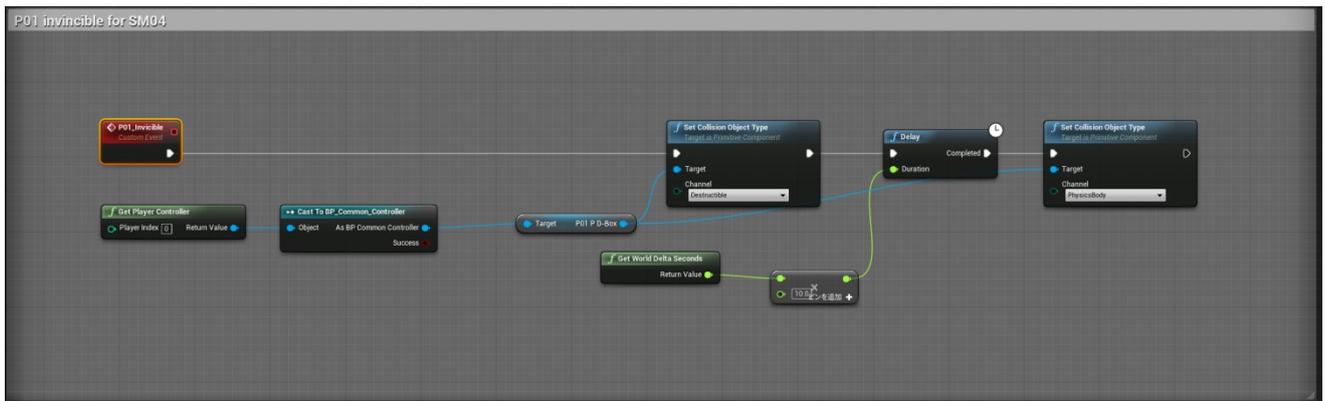
次に対空の必殺技の通知設定を説明します。



通常技の設定でも使用した、On&Off Counter / Begin&End Attack / Off_Anim / AT_Flag_Off の他にエフェクトを表示させる通知と Invincible の通知が追加され、この通知が実行されると 10 フレームの間上半身無敵になります。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

無敵の設定は AnimBlueprint で設定していますので、ここを編集すると無敵にする部分や時間を設定できます。



下半身も無敵にする場合は P01 P D-Box-low をターゲットにしてコリジョンオブジェクトタイプを Destructible から PhysicsBody に変更します。

投げ無敵は P01 P D-Box for B をターゲットにしてコリジョンオブジェクトタイプを Destructible から Vehicle に変更します。

飛び道具無敵は P01 P D-Box for B をターゲットにしてコリジョンオブジェクトタイプを Destructible から Pawn に変更します。

では次に SP バーが満タンのときに使用できる超必殺技のデータテーブルについて説明します。

SP-D

D : Damage	Projectile-speed
SN : Stun Damage	WD: Where damage
Air : Use in Air	WA: Where attack
Projectile : Use or not	HS&HB: Hit stop&back
RL: Right or Left or Projectile	CM: Kind of command
Multiple-hit-times:Multiple-hit of projectile	

設定する項目は多いですが、これまで説明してきた項目のみです。

アニメーションの通知は次のように設定します。



Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

超必殺技が発動するとカメラが切り替わります。

そのカメラを元に戻すのは OFF_SP_Camera の通知になります。

飛び道具の必殺技の場合は、SP という通知を設定すると BP_Char0x_SP がスポンされます。

後は、Off_Anim / AT_Flag_Off/ OFF_SP_Flag を設定してください。

多段の打撃技の場合は、Begin&End Attack の通知を複数箇所に設定して下さい。

それでは最後に 3on3 のアシストで使用する必殺技の設定を説明します。

データテーブルは以下の項目を設定します。

3on3AS-D

D : Damage Projectile-speed
SN : Stun Damage WD: Where damage
HS&HB: Hit stop&back WA: Where attack
Projectile : Use or not
RL: Right or Left or Projectile
Multiple-hit-times:Multiple-hit of projectile

次にアニメーションの通知を説明します。

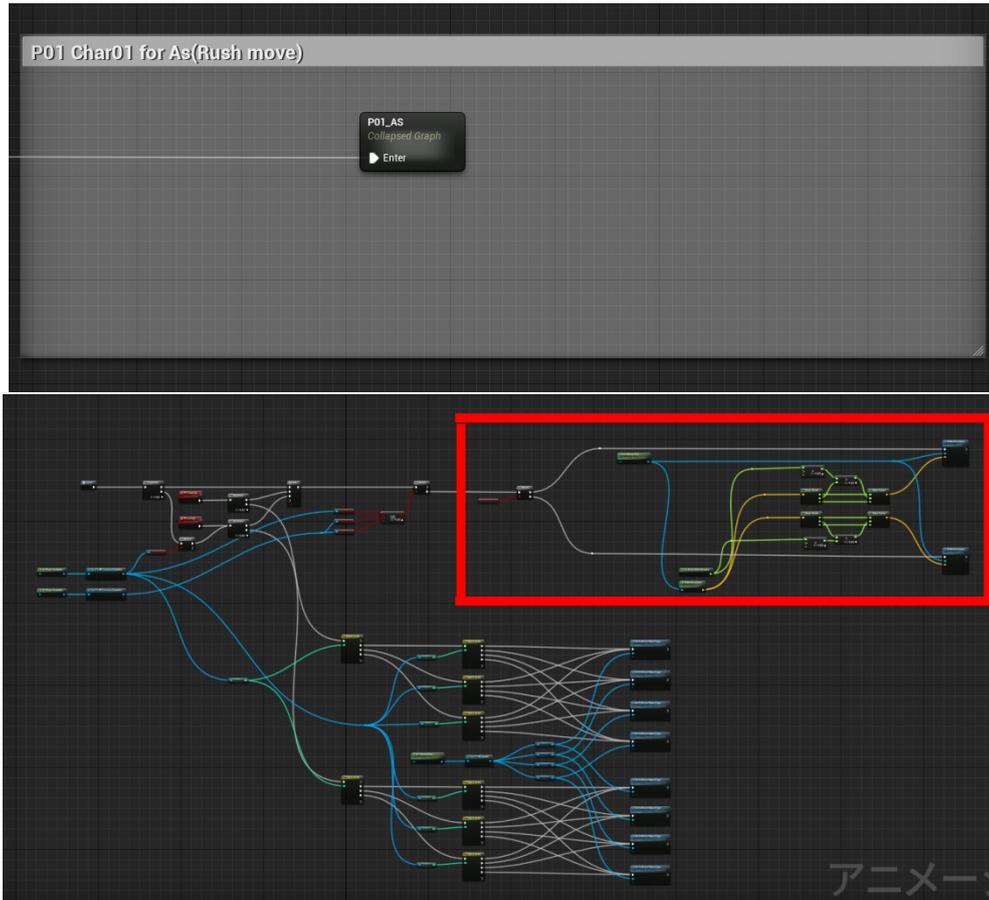
飛び道具以外のアシストの場合は以下の通知を設定します。



Begin_to_AT_As と End_to_AT_As を追加するのと必要に応じてエフェクトの通知を追加します。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

デフォルトキャラクターのアシストは突進技のアシストを設定しており、AnimBlueprint で移動処理をしているので、不要の場合はこのノードを切断して下さい。



飛び道具の場合は以下の通知を設定します。

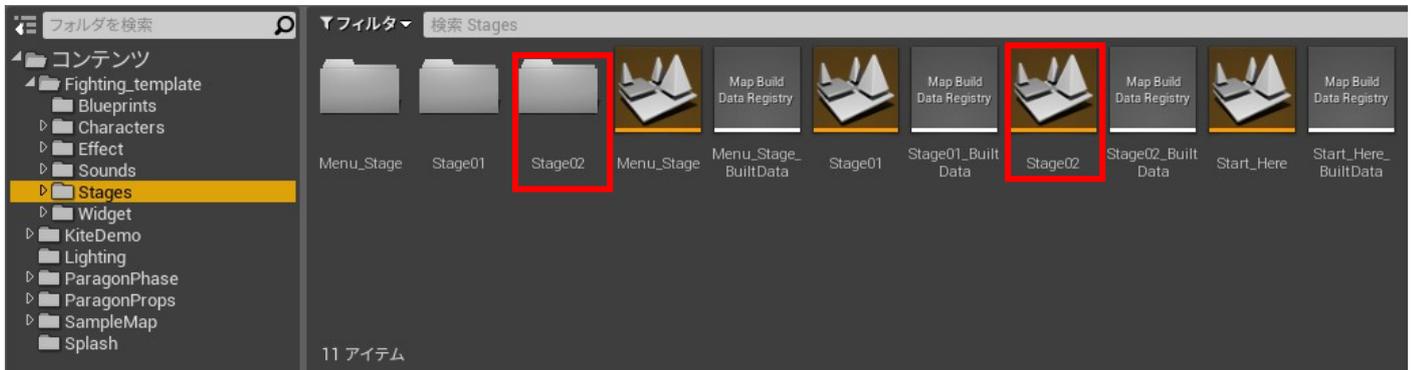


Shot_bullet_AS の通知を追加すると BP_Char0x_projectile_AS をスポーンします。
キャラクターアニメーションの設定は以上です。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

5. ステージの追加方法

Stages フォルダに素材を格納するフォルダとレベルファイルに適切な名称を付けて追加します。
フォルダの名称は、Stage03、Stage04・・・と追加してください。



追加したレベルファイルを開き、レベルを作成します。

ここでは、事例として、BP_Sky_Sphere と DirectionalLight を追加しておきます。

次に地面となる Plane を追加しますが、壁があるレベルにする場合は、Plane の面積を 20m*20m に、
壁がないレベルにする場合は、面積を 60m*60m にすることをお勧めします。

(ここでは 60m*60m の Plane を追加します。)

次に、座標軸の確認をします。

このプロジェクトで、カメラに向かって正面となるのは、以下の向きの座標軸です。

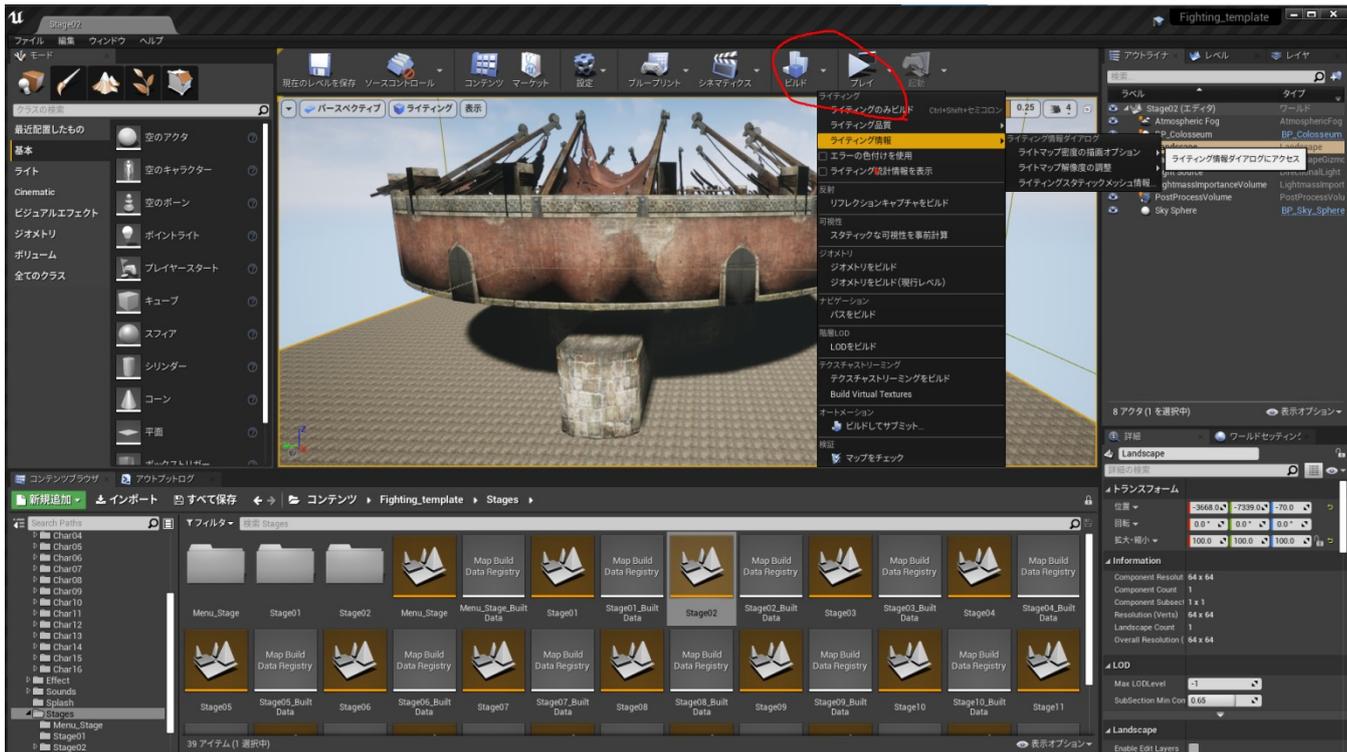


ここまで確認できたら、好みのレベルを作成してください。

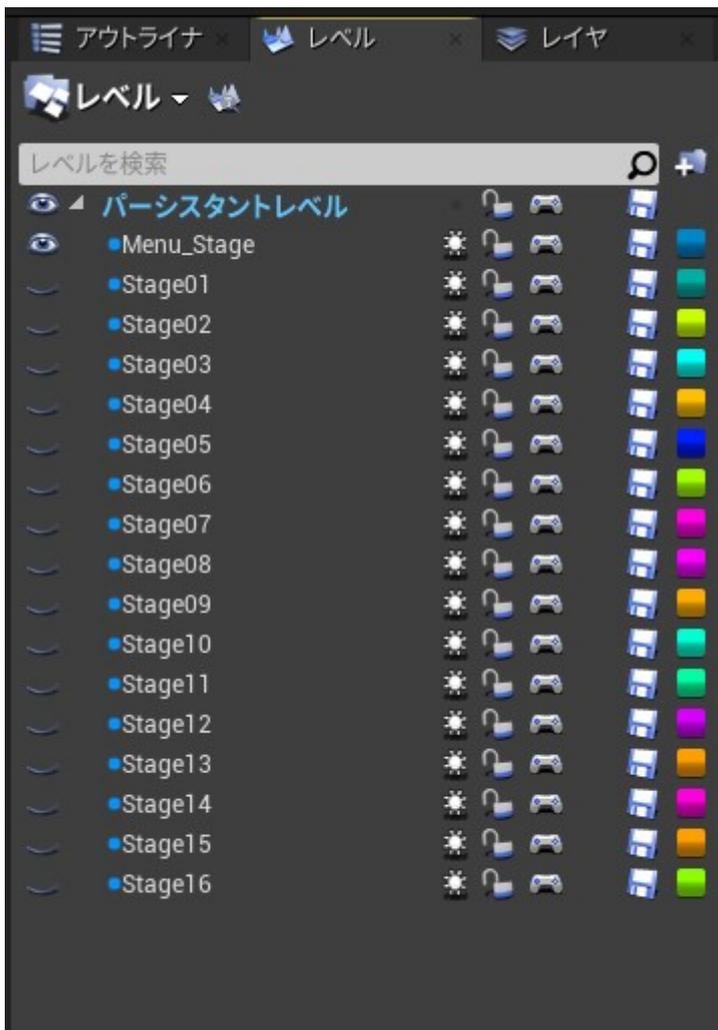
ここでは、InfinityBlade の無料アセットを事例として作成しますが、地面の Z 座標を -70 にしてください。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

完成したらライティングビルドを行います。



次に StartHere のレベルファイルを開き、右上のレベルのタブを開きます。

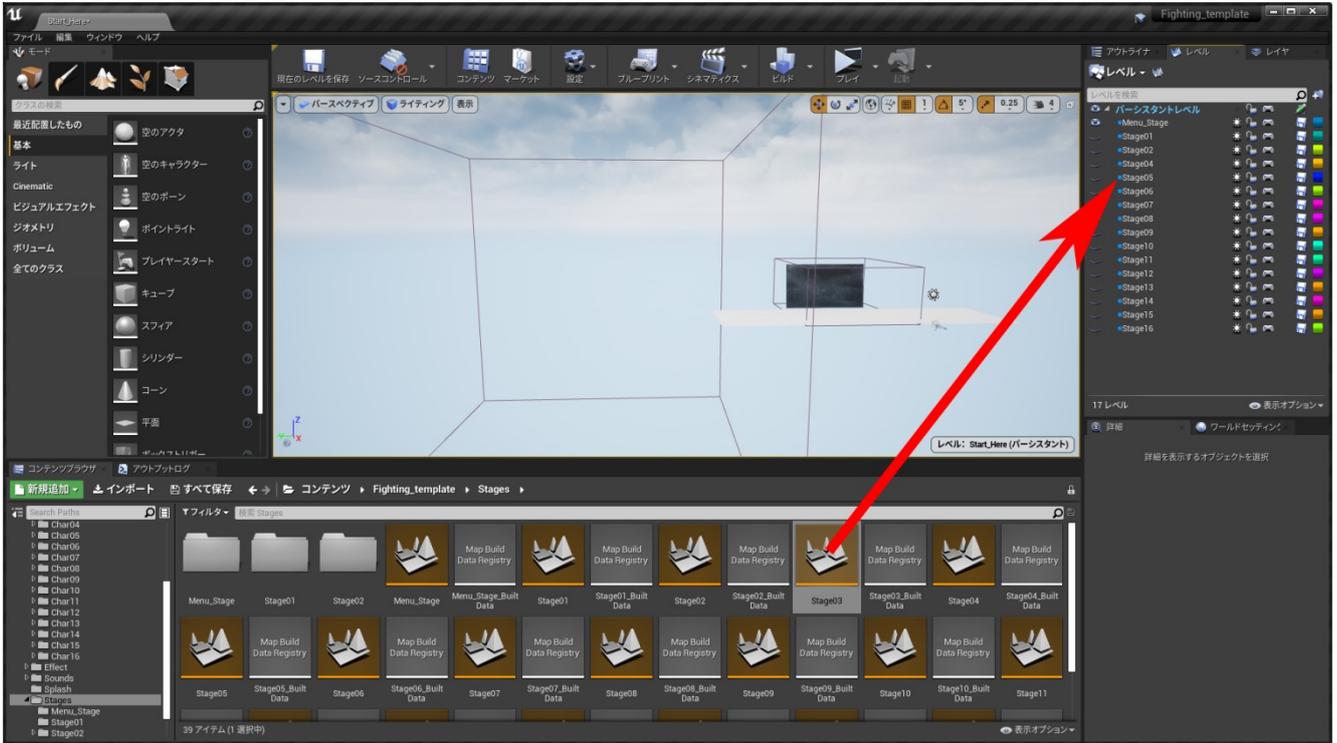


Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

59

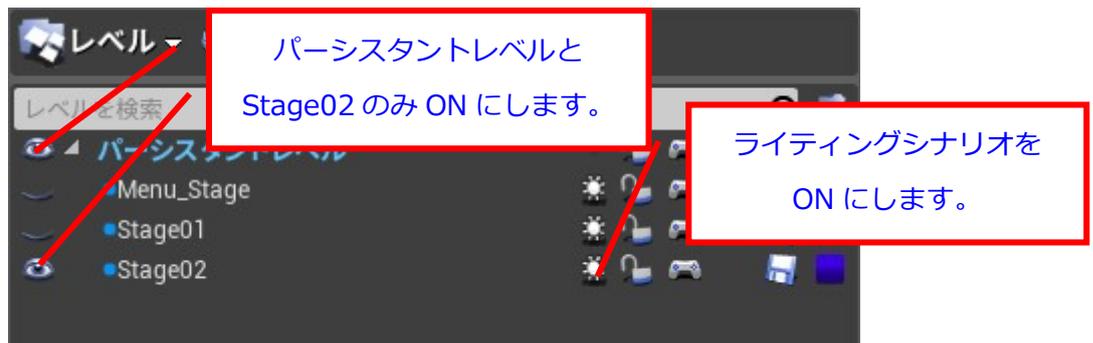
Stage02 として追加する場合は、レベルタブにある既存の Stage02 を削除して、既にある Stage02 とビルドデータも削除します。

追加するレベルファイルの名称を Stage02 に変更し、レベルタブヘドドラッグします。



Stage02 を可視化した状態で、地面の Z 座標が -70 で、ステージ中心の X 座標が 0、Y 座標が 2000 になるように位置を調整します。

上記の状態では、Stage02 のライティングシナリオを ON にすると、かつパーシスタントレベルと Stage02 を可視化した状態でライティングビルドを行います。



Sky light のエラーが発生する場合がありますが、無視して大丈夫です。

これ以降、StartHere レベルでサブレベルを編集する際は、可視化するのはいずれか一つにしてください。

例えば、Stage01 と Stage02 を同時に可視化すると警告が表示されます。

次にゲーム実行画面内のステージセレクト画面で表示するステージ用の画像を準備します。

T_Stage02_Small.png(100*240pixel)

T_Stage02_background.png (1280*460Pixel)

作成した画像は Widget→images フォルダへインポートしておきます。

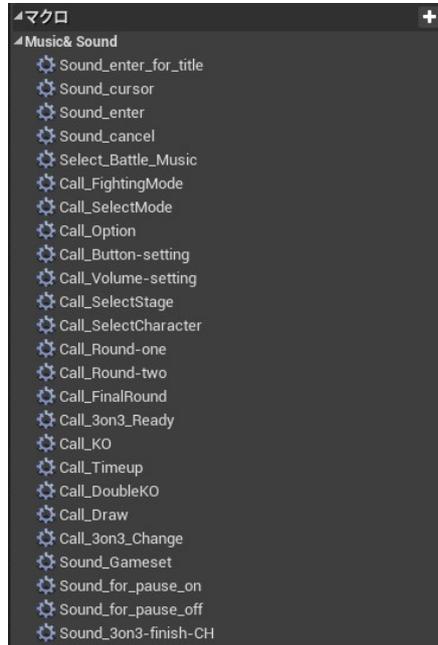
これでビルドを行い、セレクトステージ画面とバトル画面で反映されていれば OK です。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

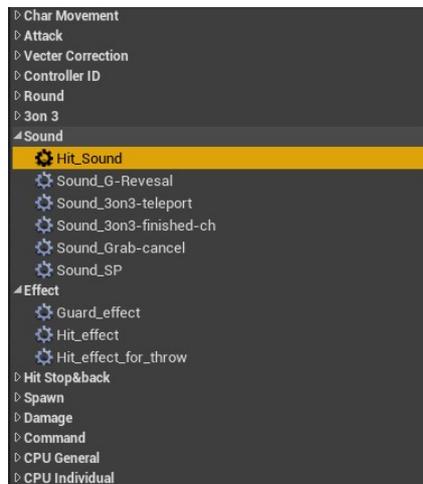
6. サウンドとエフェクトの編集方法

まずはメニュー画面で使用しているサウンドについて説明します。

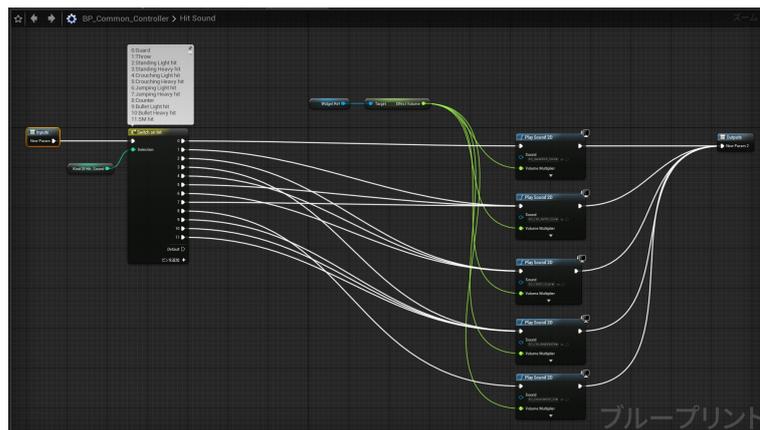
Widget_fightingのマクロ Music&Sound に使用している全てのサウンドがありますので、入れ替えることができます。



また、バトル中のサウンドとエフェクトはBP_Common_Controllerにあるマクロで設定しています。

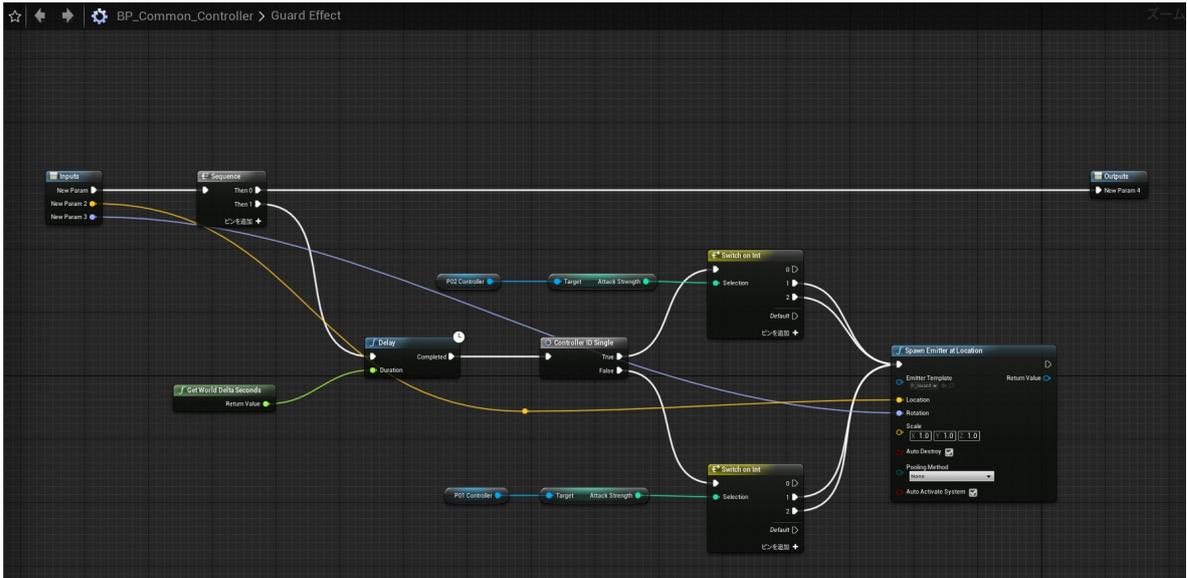


まず、Hit_Soundのマクロを開くと、ガードやヒット時のサウンドが設定されていますので、Switchノードのコメントを見ながら、サウンドファイルを入れ替えてください。

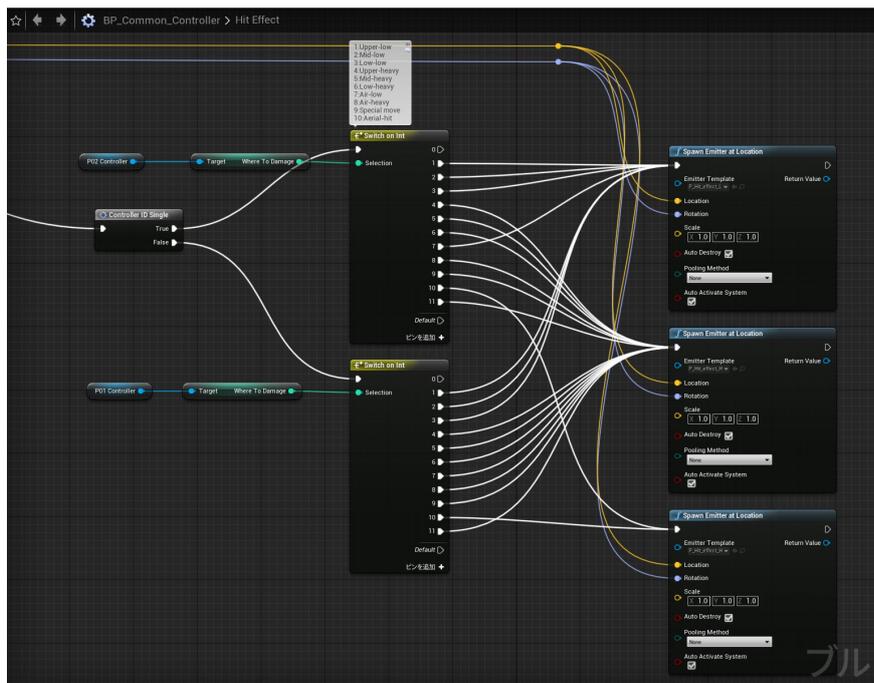


Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

次にガードエフェクトのマクロ開くとガード時に表示されるエフェクトが設定されています。



次にヒットエフェクトのマクロを開くとヒット時のエフェクトが設定されていますので、Switch ノードのコメントを見ながら、エフェクト入れ替えてください。



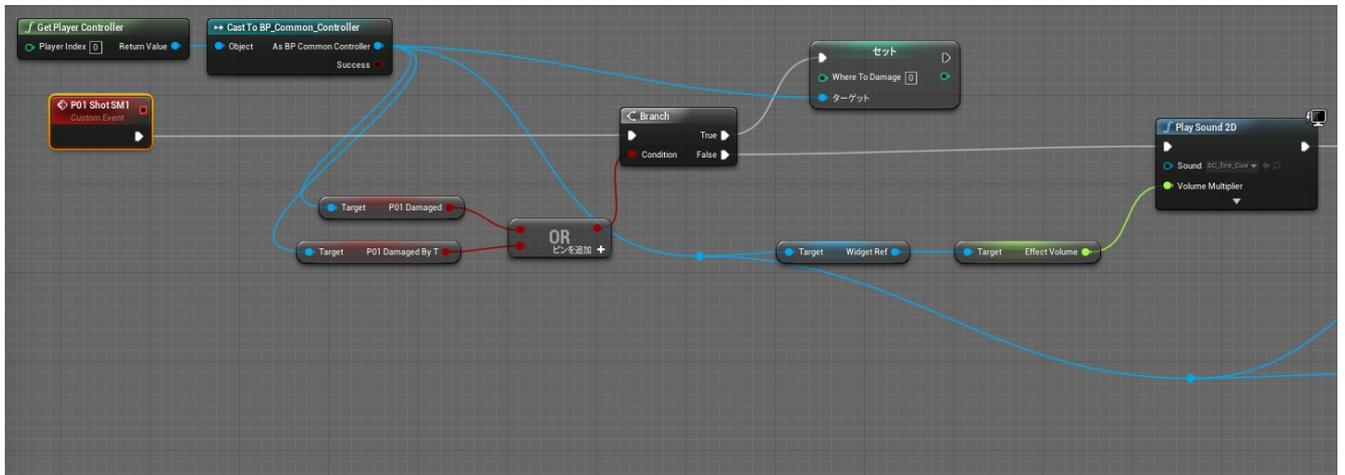
もし投げのアニメーションに打撃がある場合は、Hit_effect_for_throw のマクロでエフェクトを設定していますので、必要に応じて入れ替えてください。

Unreal Engine 格闘ゲーム テンプレートプロジェクト マニュアル Ver.3.5

62

更に必殺技を使用したときのサウンドについては、アニメーションの通知に設定することもできますが、OPTION メニューで設定した音量設定を有効にすることができないので、AnimNotify を使用して、変数の Effect Volume を Volume Multiple に繋げて再生してください。

以下の画像は、SM1 の Projectile の例になります。



7. デバッグ時の注意点

動作確認をする際は、必ずフレームレートを確認して行ってください。

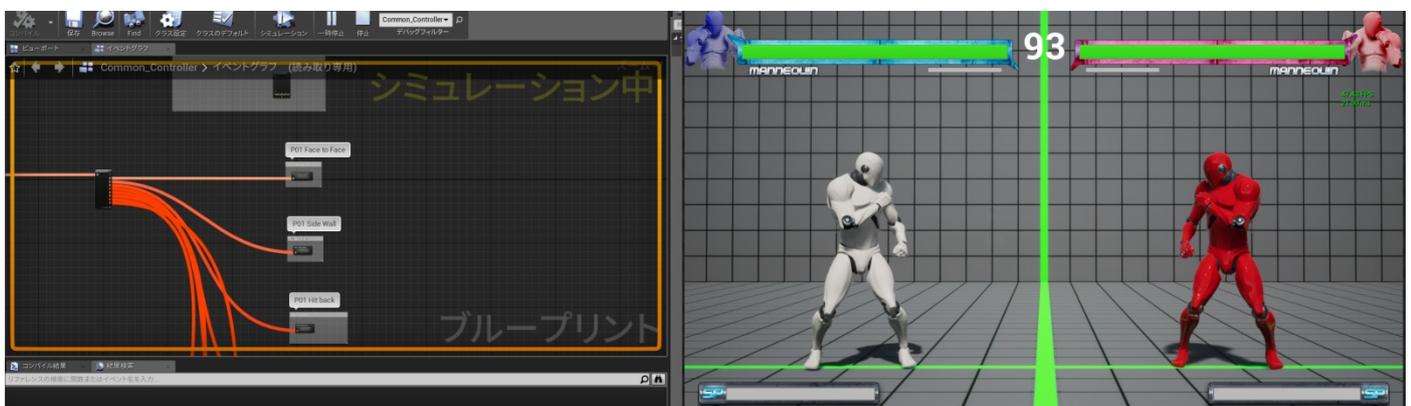
(アウトプットログで Stat fps コマンドを入力するか Tick に繋いだ PrintString に 1 を WorldDeltaSeconds で割った値をセットして表示させてください。)

フレームレートが下がっている状態で動作すると、フレームレートが下がっている状態でしか発生しない不具合が発生することがあります。

以下画像のように Blueprint の実行を確認しながらプレイを行うとフレームレートが下がることを確認しています。

これを解消するためにはマルチディスプレイを利用して、Blueprint の実行ウィンドウとプレイウィンドウを別々のモニターで実行すると 60fps をキープしながら Blueprint の実行を確認できます。

ただし、Blueprint の実行確認範囲をズームアウトして広範囲にするとマルチディスプレイにしても、フレームレートが下がる場合があります。



以上