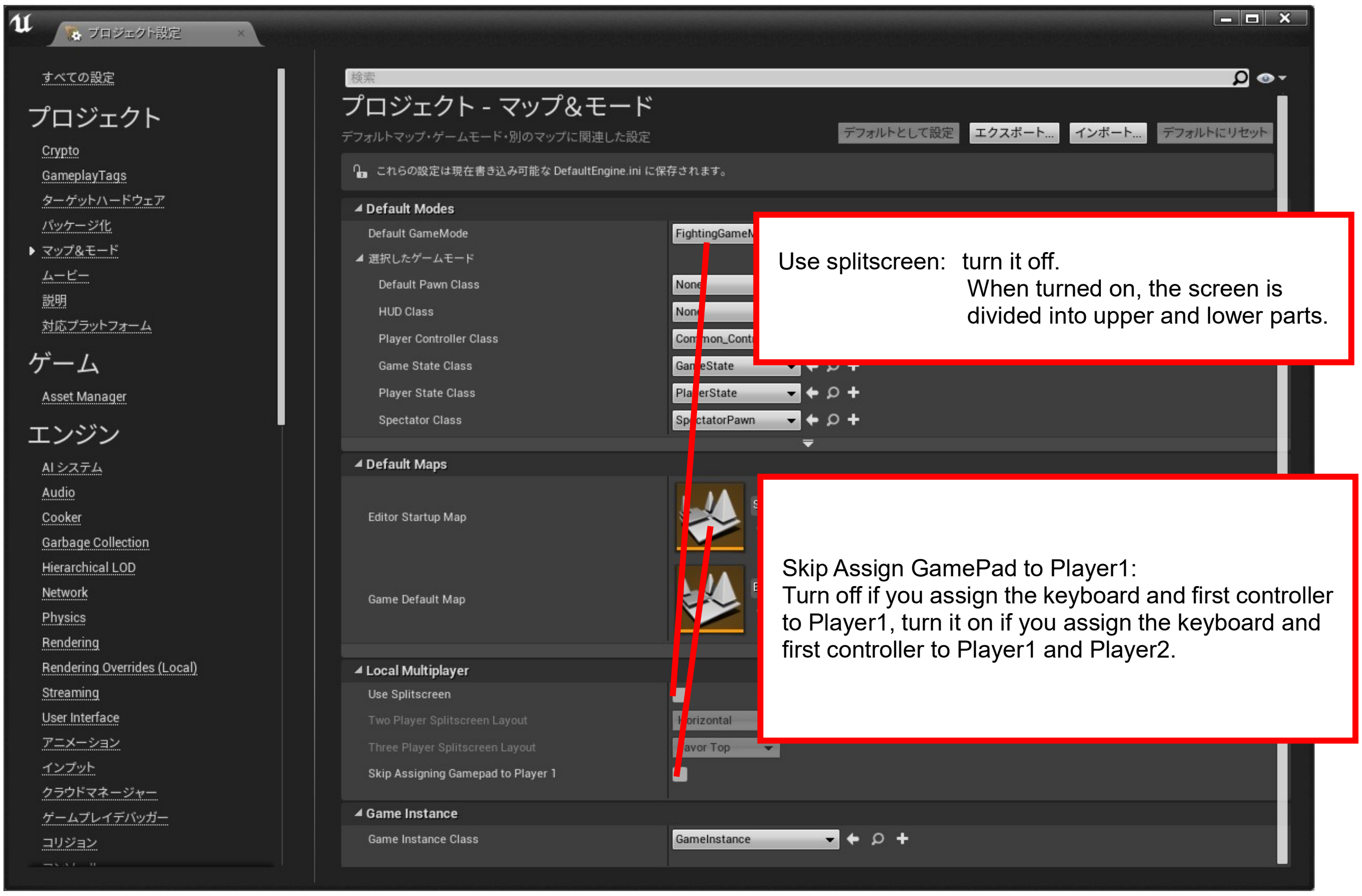
Unreal Engine Fighting Game Template project Manual

1. About project settings
Check the settings of the local multiplayer in the map & mode in the project settings by looking at the image below.



プロジェクト - マップ&モード
デフォルトマップ・ゲームモード・別のマップに関連した設定
これらの設定は現在書き込み可能な DefaultEngine.ini に保存されます。

Default Modes
- Default GameMode: FightingGameM
- 選択したゲームモード
  - Default Pawn Class: None
  - HUD Class: None
  - Player Controller Class: Common_Cont
  - Game State Class: GameState
  - Player State Class: PlayerState
  - Spectator Class: SpectatorPawn

Default Maps
- Editor Startup Map
- Game Default Map

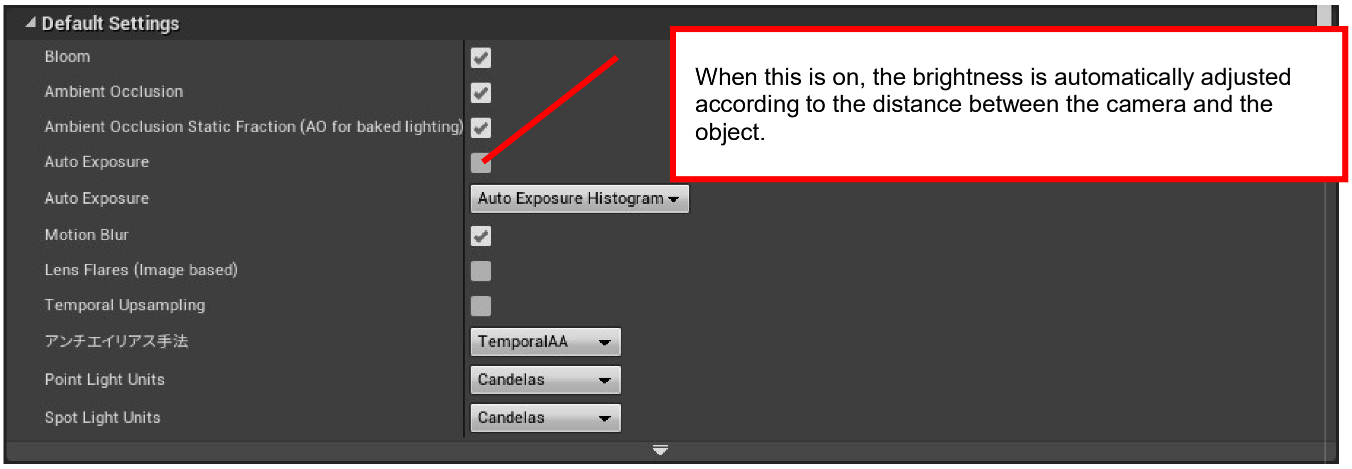
Local Multiplayer
- Use Splitscreen
- Two Player Splitscreen Layout: Horizontal
- Three Player Splitscreen Layout: Favor Top
- Skip Assigning Gamepad to Player 1

Game Instance
- Game Instance Class: GameInstance

Use splitscreen: turn it off.
When turned on, the screen is divided into upper and lower parts.

Skip Assign GamePad to Player1:
Turn off if you assign the keyboard and first controller to Player1, turn it on if you assign the keyboard and first controller to Player1 and Player2.

Next, turn off the automatic exposure in Engine → Rendering.



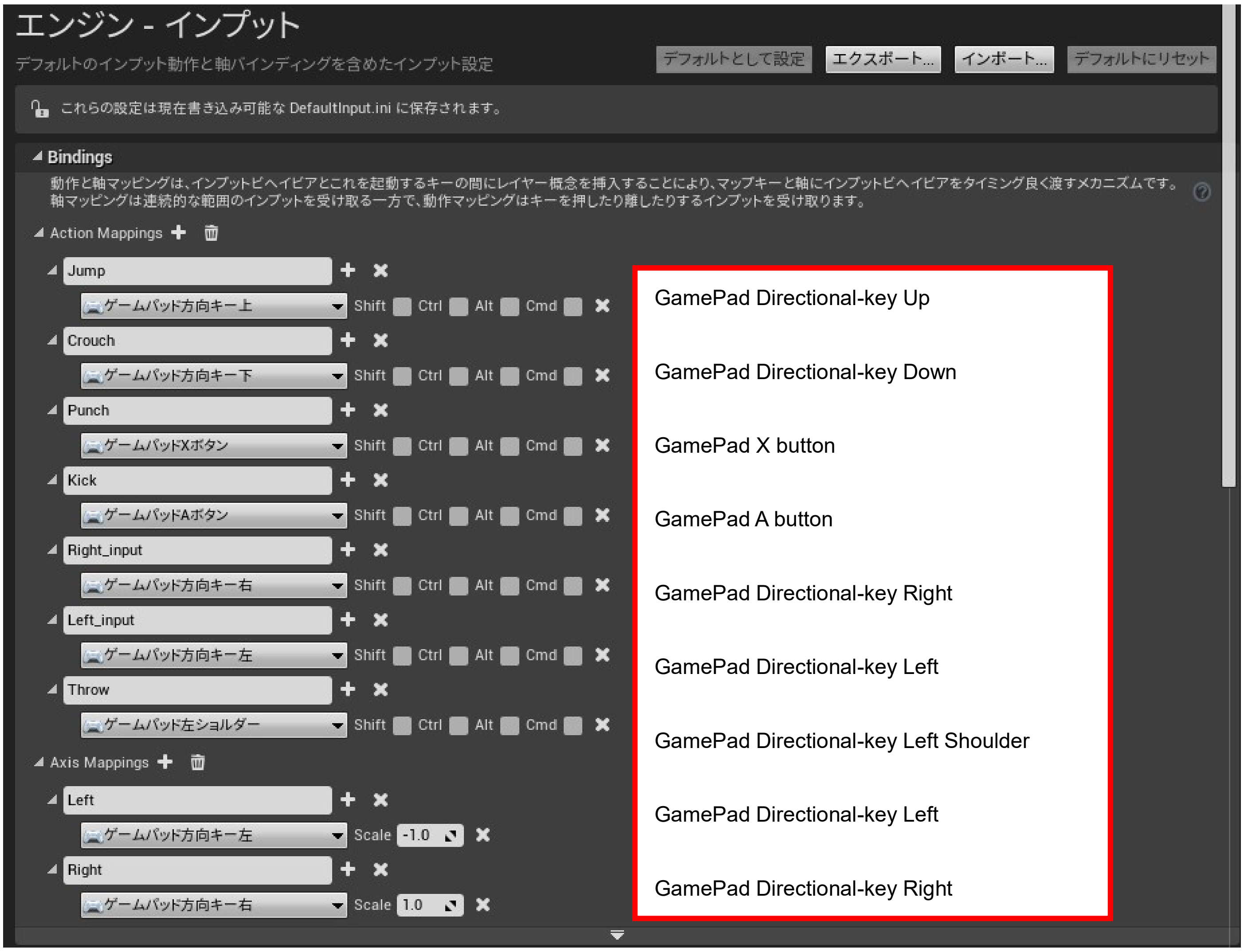
Default Settings

Bloom	✓
Ambient Occlusion	✓
Ambient Occlusion Static Fraction (AO for baked lighting)	✓
Auto Exposure	☐
Auto Exposure	Auto Exposure Histogram
Motion Blur	✓
Lens Flares (Image based)	☐
Temporal Upsampling	☐
アンチエイリアス手法	TemporalAA
Point Light Units	Candelas
Spot Light Units	Candelas

When this is on, the brightness is automatically adjusted according to the distance between the camera and the object.

Next, set the input settings of the controller in Engine → Input as shown in the following screen.
Note that if the name of the mapping is wrong, it will not work correctly.

Unreal Engine Fighting Game Template project Manual



エンジン - インプット

デフォルトのインプット動作と軸バインディングを含めたインプット設定

これらの設定は現在書き込み可能な DefaultInput.ini に保存されます。

Bindings

動作と軸マッピングは、インプットビヘイビアとこれを起動するキーの間にレイヤー概念を挿入することにより、マップキーと軸にインプットビヘイビアをタイミング良く渡すメカニズムです。
軸マッピングは連続的な範囲のインプットを受け取る一方で、動作マッピングはキーを押したり離したりするインプットを受け取ります。

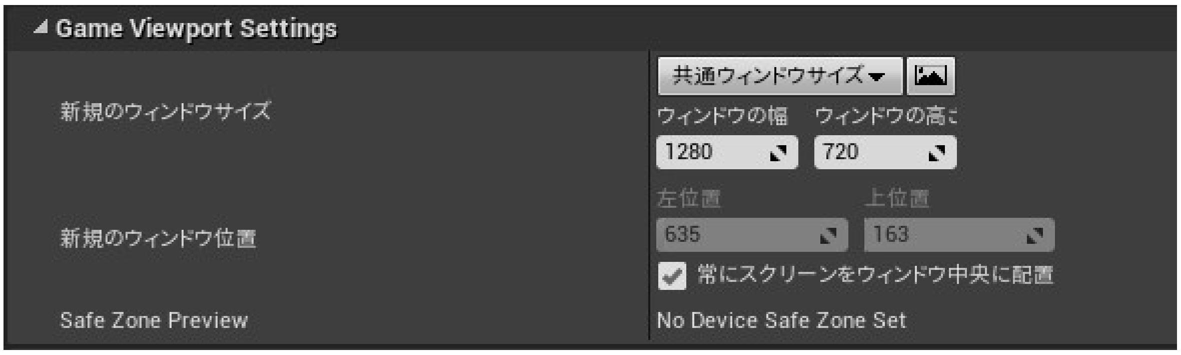
Action Mappings

Jump	ゲームパッド方向キー上	GamePad Directional-key Up
Crouch	ゲームパッド方向キー下	GamePad Directional-key Down
Punch	ゲームパッドXボタン	GamePad X button
Kick	ゲームパッドAボタン	GamePad A button
Right_input	ゲームパッド方向キー右	GamePad Directional-key Right
Left_input	ゲームパッド方向キー左	GamePad Directional-key Left
Throw	ゲームパッド左ショルダー	GamePad Directional-key Left Shoulder

Axis Mappings

Left	ゲームパッド方向キー左	Scale -1.0	GamePad Directional-key Left
Right	ゲームパッド方向キー右	Scale 1.0	GamePad Directional-key Right

Finally, specify the window size at the time of execution by level editor → play.
Enter the size at which the aspect ratio is 16: 9.
(Example: 1280 * 720/1600 * 900/1920 * 1080)

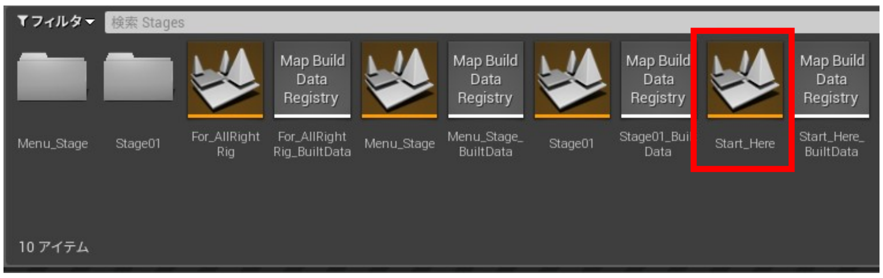


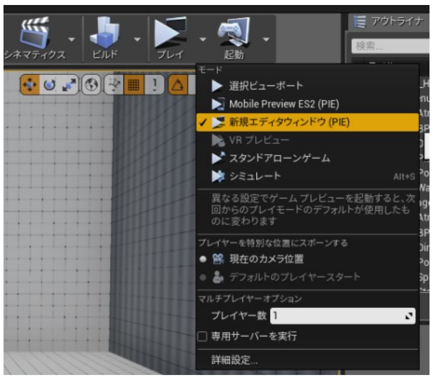
Game Viewport Settings

新規のウィンドウサイズ	共通ウィンドウサイズ; ウィンドウの幅 1280; ウィンドウの高さ 720
新規のウィンドウ位置	左位置 635; 上位置 163; ✓ 常にスクリーンをウィンドウ中央に配置
Safe Zone Preview	No Device Safe Zone Set

Unreal Engine Fighting Game Template project Manual

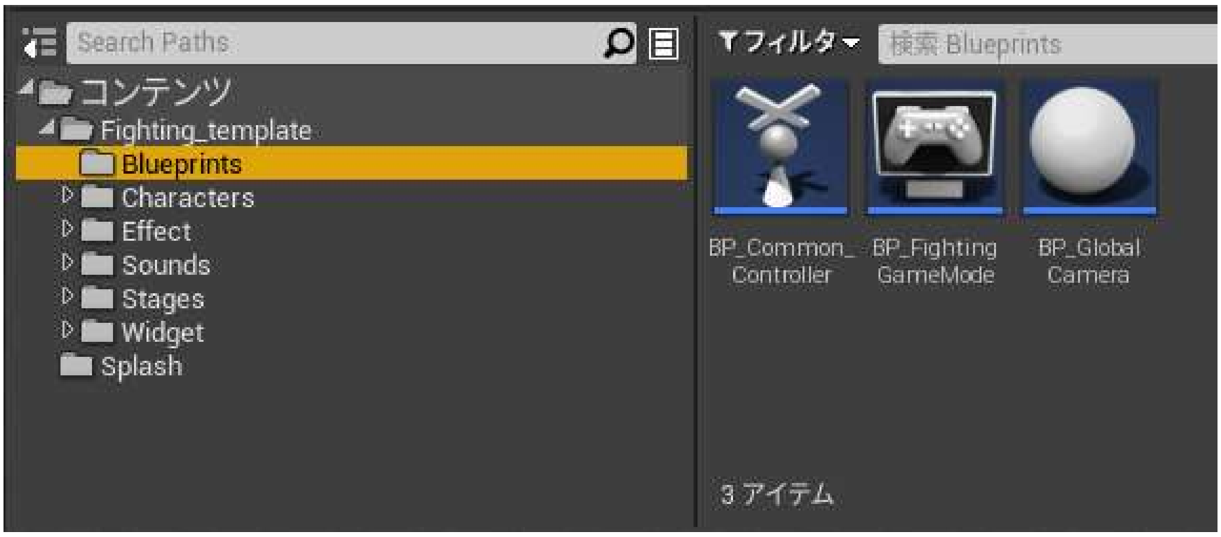
This completes the project settings.
After confirming that you have opened the Start_Here file in the Stage folder in the contents of the project, try running it.
（Check if the size is reflected in the new editor window.）




シネマティクス
ビルド
プレイ
起動
アウトライナ
モード
選択ビューポート
Mobile Preview ES2 (PIE)
✓ 新規エディタウィンドウ (PIE)
VR プレビュー
スタンドアローンゲーム
シミュレート Alt+S
異なる設定でゲーム プレビューを起動すると、次回からのプレイモードのデフォルトが使用したものに変わります
プレイヤーを特別な位置にスポーンする
● 現在のカメラ位置
デフォルトのプレイヤースタート
マルチプレイヤーオプション
プレイヤー数 1
専用サーバーを実行
詳細設定...

Check “https://youtu.be/BXCtcZlAi-s” for the operation method in the game.

2．About folder structure and files
Below is an overview of folders and files.
■Blueprints folder



BP_CommonController：This is the main program file for this project.
Receives player input and controls characters and various gauges.
Player2 control generates a file in the form of CommonController-1,
and Player1 and 2 are identified by the value of the variable Controller_ID.

BP_FightingGameMode：This file is the initial setting for this project
Basically there is no need to change the settings.

BP_GlobalCamera　　：This is the program file that controls the camera.

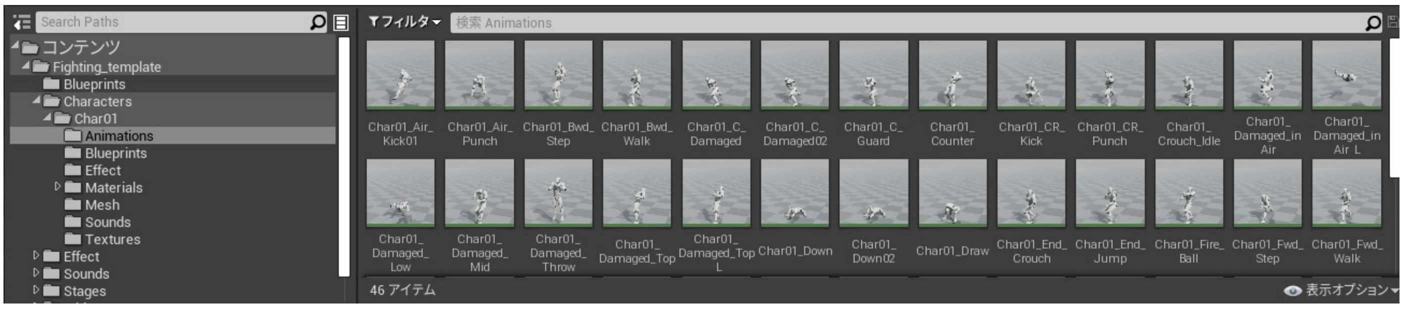
Unreal Engine Fighting Game Template project　Manual

■Character Folder

・ Char01 folder

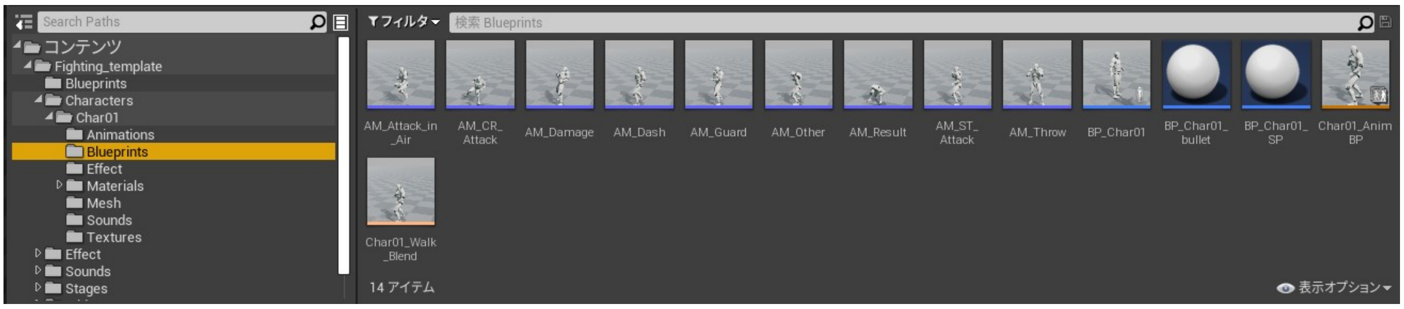
A dedicated file for each character is saved.
When adding a character, please increase Char02, Char03 and folders.

・ Animations



Character animation is saved.
There is also a folder for AllRightRigAnimation (sold separately),
but this is used when you want to edit the animation.

・ Blueprints



・ BP_Char01

Describes the processing when the character's attack / damage judgment box is set and the opponent's Capsule component overlaps.

・ Char01_AnimBP

Controls animation play.

・ BP_Char01_Bullet

Controls the actor (FireBall) that occurs when using special moves.

・ BP_Char01_SP

Controls the actor (IceBall) that occurs when using Special Moves that can be used when the SP gauge is full.

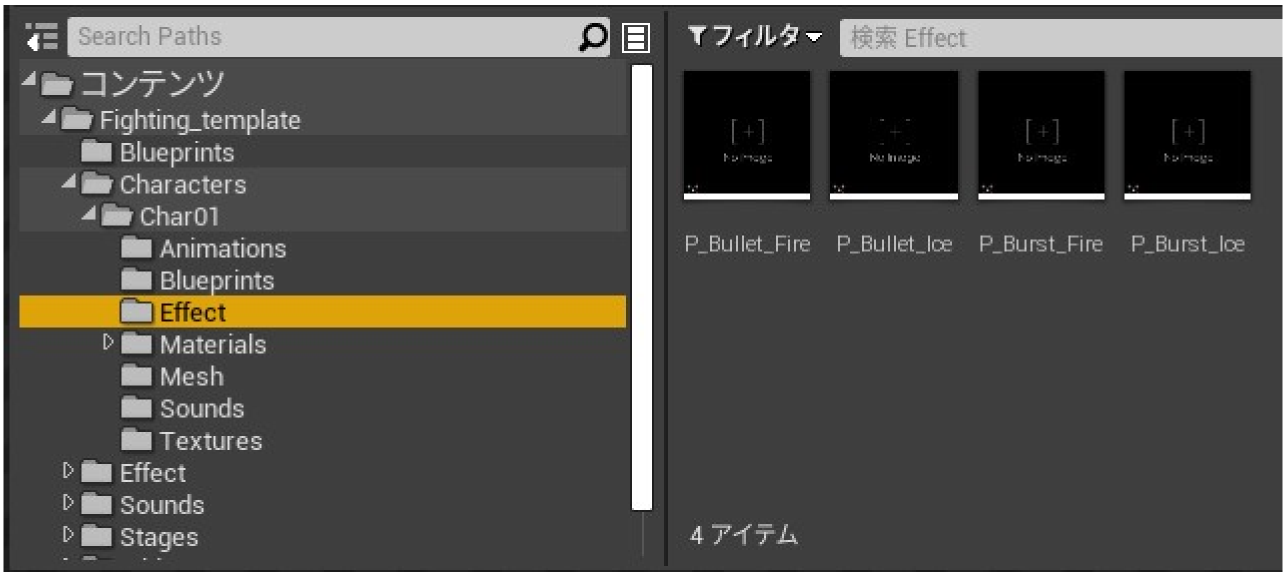
・ Other animation montages

An animation montage is set for each character state such as standing or crouching.
In this file, the timing of occurrence / disappearance of attack judgment is specified using notification.

・ Char01_Walk_Blend（Blend space 1D）

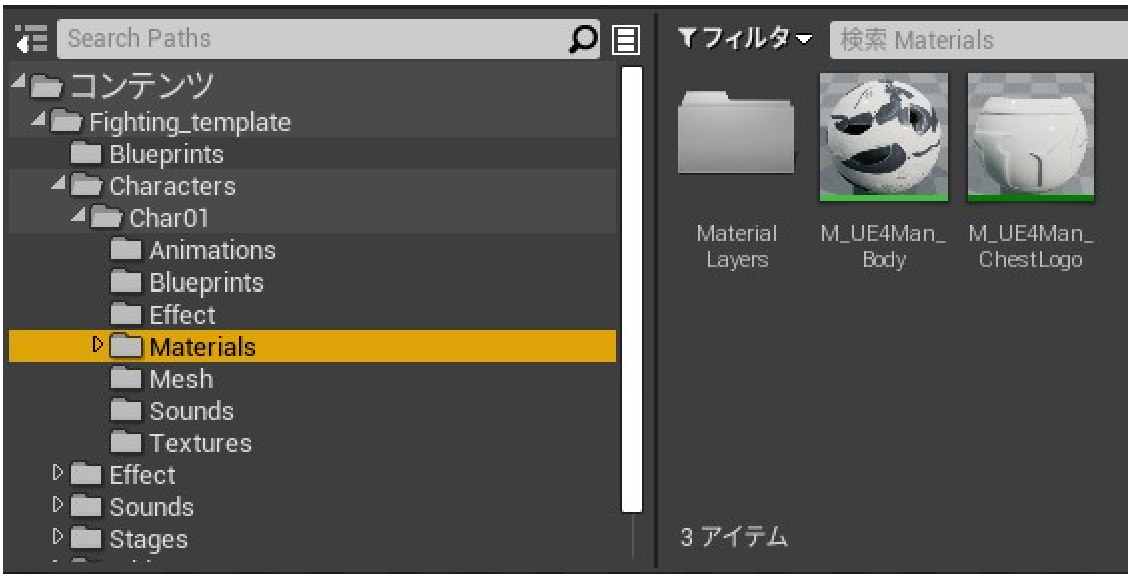
Controls the animation of standby and forward / backward movement.

・ Effect/Sounds Folder



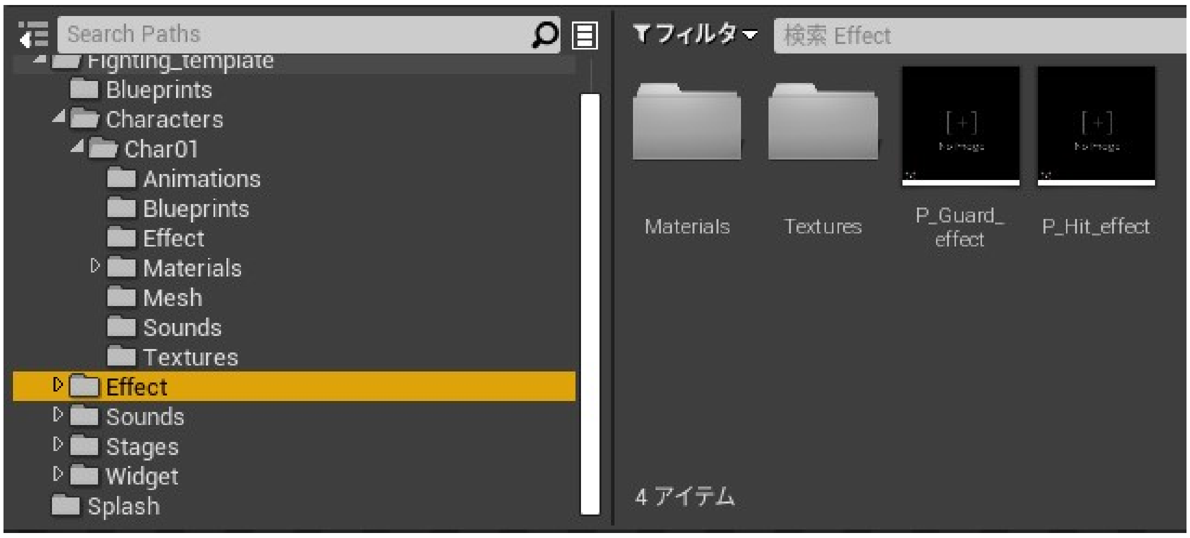
It saved particles and sounds for the special moves of characters.

・ Materials/Mesh/Textures Folder



The material of Gray Man, a standard UE4 character, is saved.

■Effect Folder



Contains the particles and their materials when hit or guarded.

■Sounds Folder

Contains sounds used for hits and widgets.

■Stages Folder

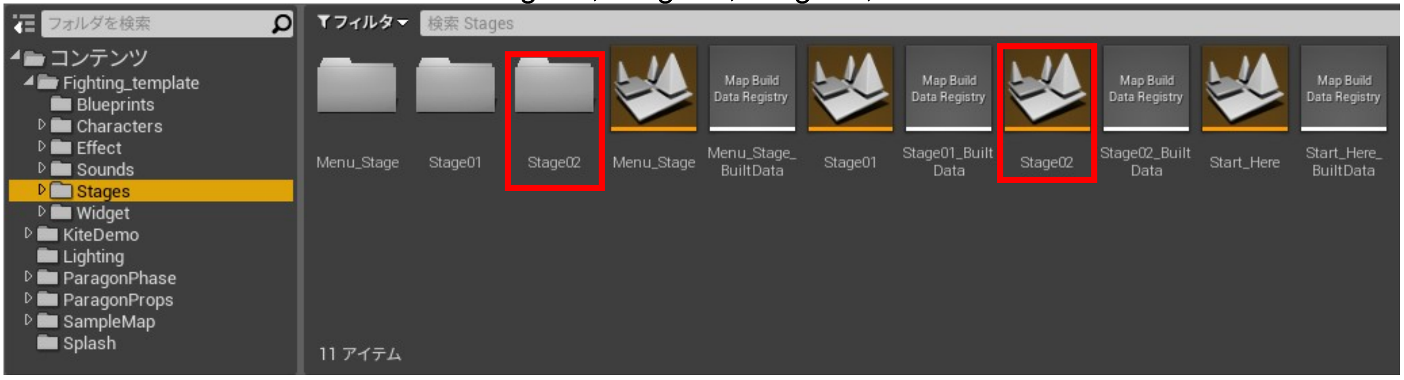
Contains stage material and level files.

■Widget

The Blueprint that controls the texture and animation required for the menu screen is saved.

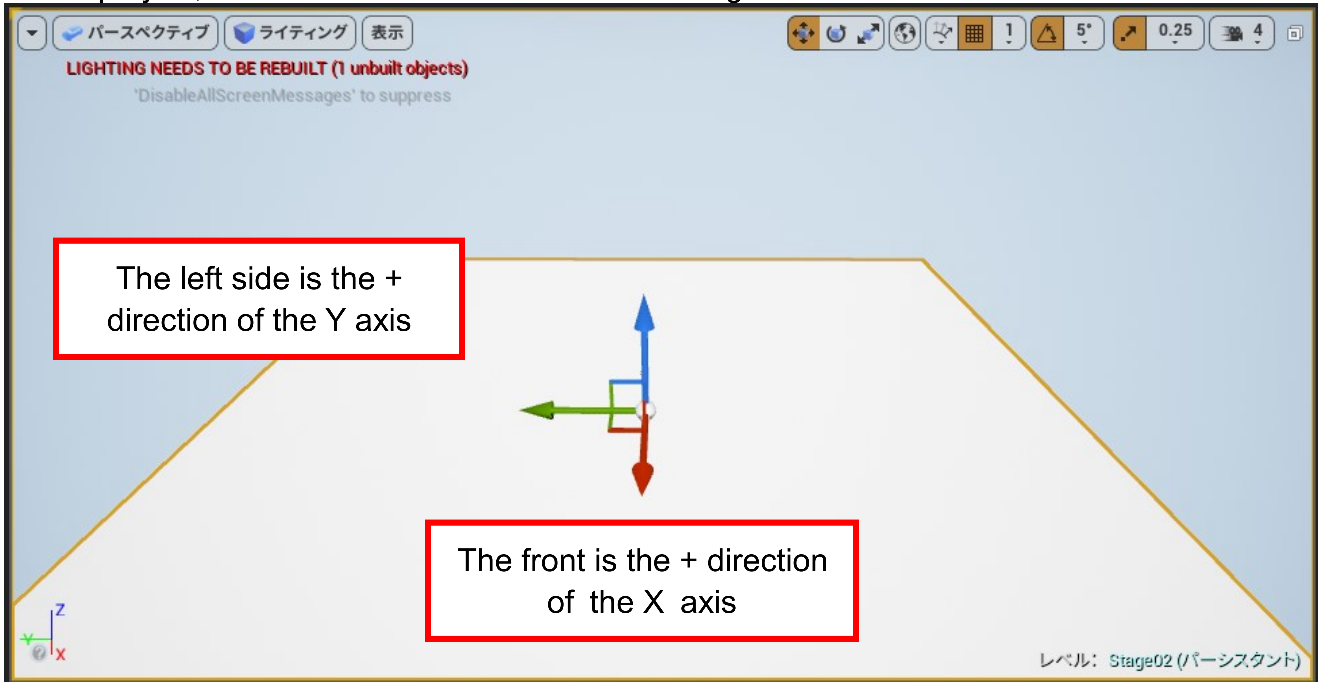
3． How to add a stage

Add a folder and level file to store the material in the Stages folder.
Add the file and folder names as Stage02, Stage03, Stage04, etc.



Open the stage02 level file and create a level.
Here, BP_Sky_Sphere and DirectionalLight are added as an example.
Next, add the plane that will be the ground, but if you want to make the wall a level,
it is recommended that the area of the plane be 20 m * 20 m, and if you want a level without a wall, it is recommended that the area be 60 m * 60 m To do.
（Here we add a 60m * 60m Plane.）

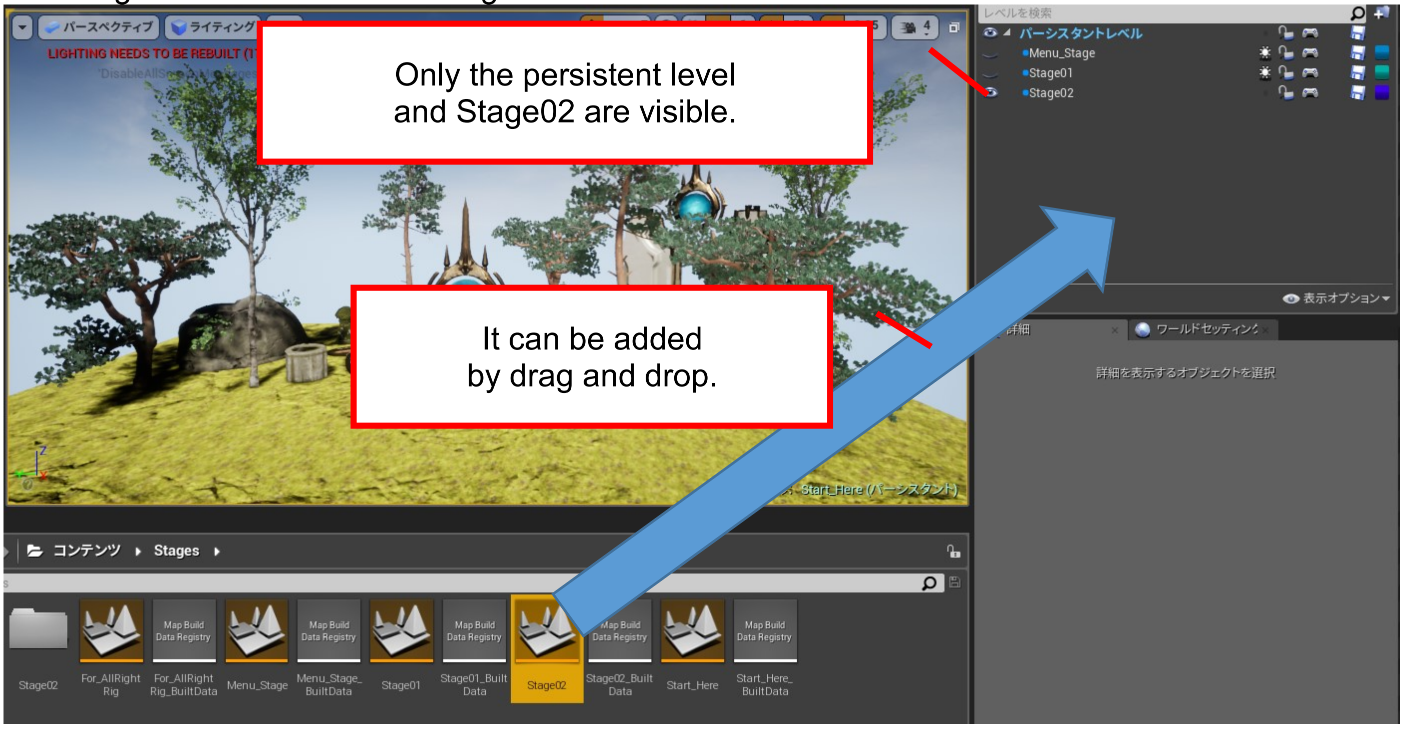
Next, check the coordinate axes.
In this project, the coordinate axes with the following orientations face the camera.


パースペクティブ
ライティング
表示
LIGHTING NEEDS TO BE REBUILT (1 unbuilt objects)
'DisableAllScreenMessages' to suppress
The left side is the + direction of the Y axis
The front is the + direction of the X axis
レベル: Stage02 (パーシスタント)

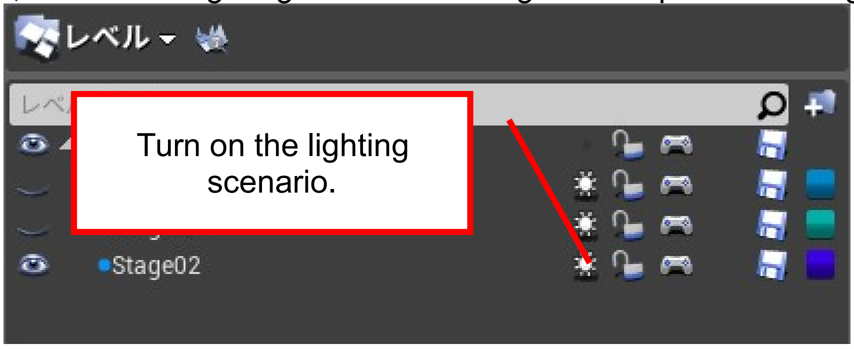
Once you have confirmed so far, create your favorite level.

Unreal Engine Fighting Game Template project　Manual

Here, we will create Paragon free assets as an example.
After writing and saving Stage02, open the StartHere level and add Stage02 as a sublevel.
Set the ground coordinates of Stage02 to X: 0 Y: 2000 Z: -70.



In the above state, turn on the lighting scenario of Stage02 and perform the lighting build.



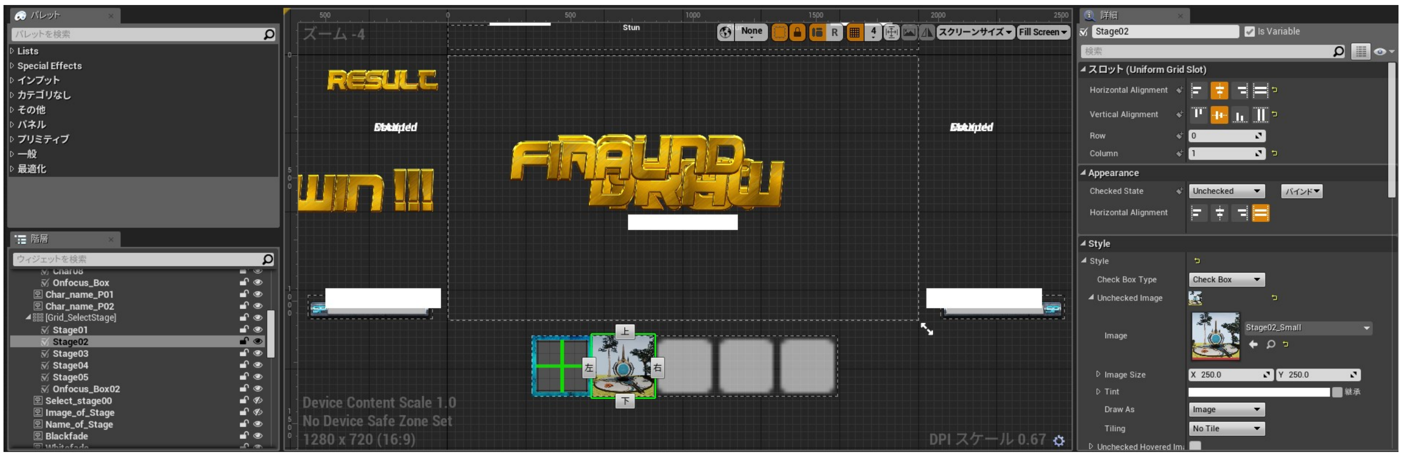
From now on, when editing sub-levels at the StartHere level, only one should be visualized.
For example, if Stage01 and Stage02 are visualized at the same time, a warning is displayed.
Next, prepare an image for the stage to be displayed on the stage select screen in the game play screen.

Stage02_Small.png(250*250pixel)

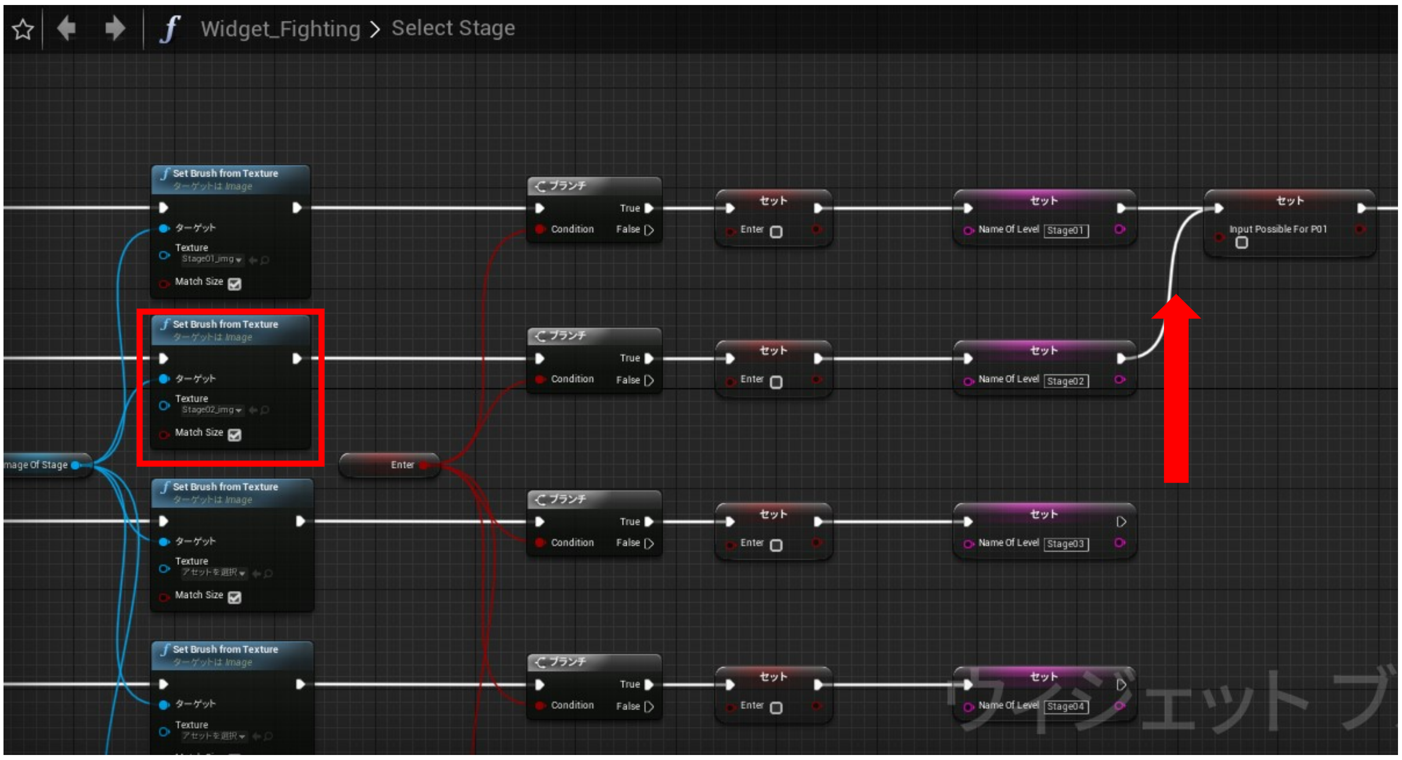
Stage02_img.png（640*480Pixel）

Import the created image into the Widget → images folder.
Next, open the Widget Blueprint and specify Stage02_Small.png on the designer screen as shown below.

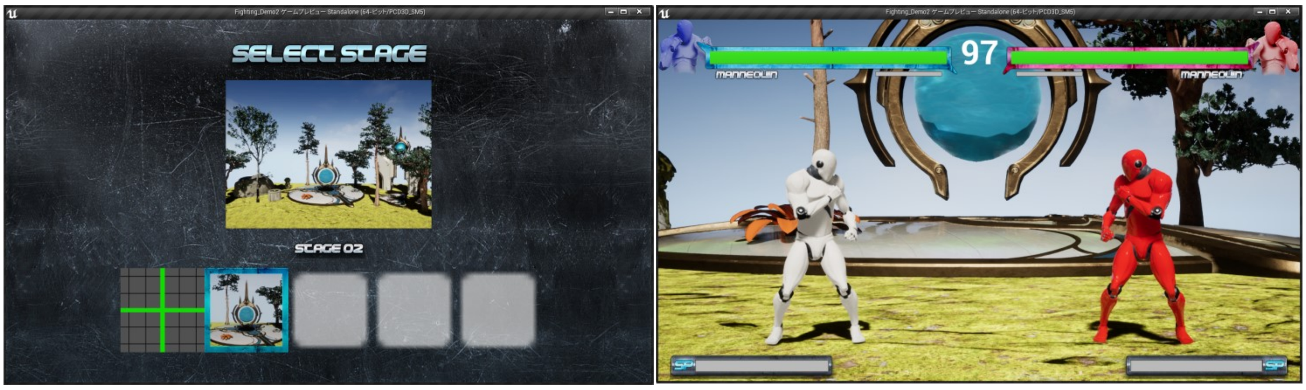
Unreal Engine Fighting Game Template project Manual



Select Stage02 in [Grid_SelectStage] and select Stage02_Small.png for Uncheckedimage. Next, open the function of Select Stage on the Widget Blueprint graph screen, specify Stage02_img.png for Set Brush from Texture, and connect the node to continue processing after selection.

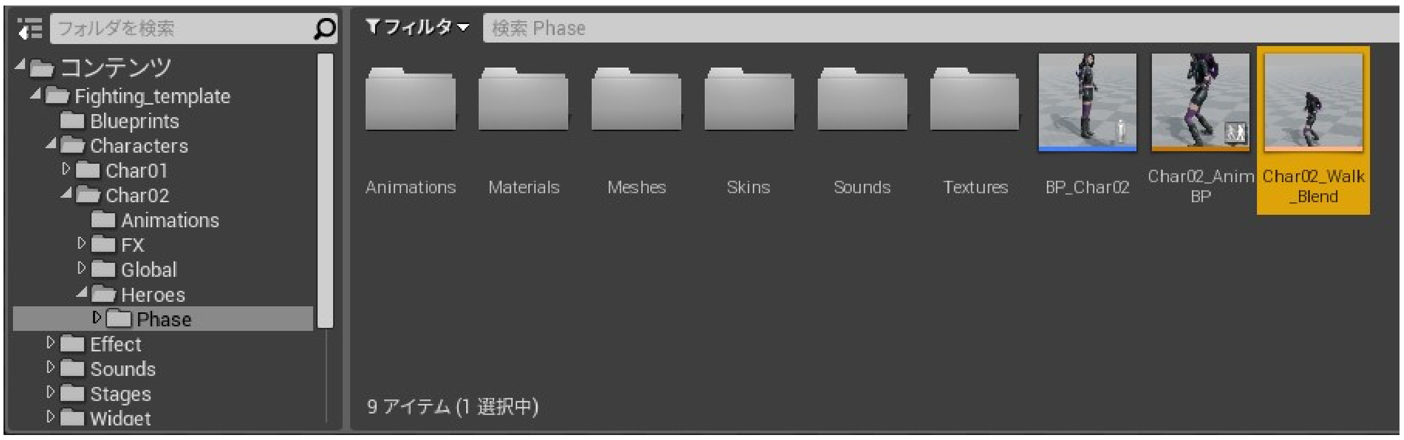


It is OK if the build is done and it is reflected as follows on the select stage screen and the battle screen.

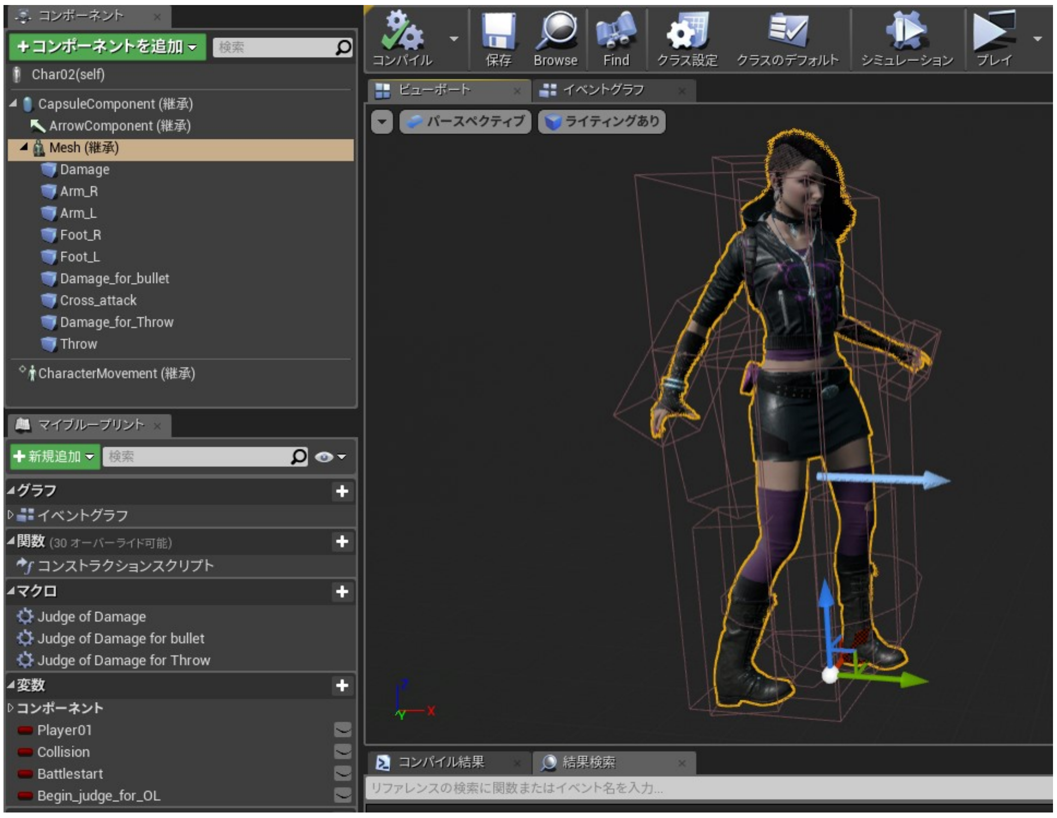


Unreal Engine Fighting Game Template project Manual

４． How to add characters
To add a character, add the Char02 folder to the Characters folder.
Here, we will explain the phase of Paragon assets as an example.

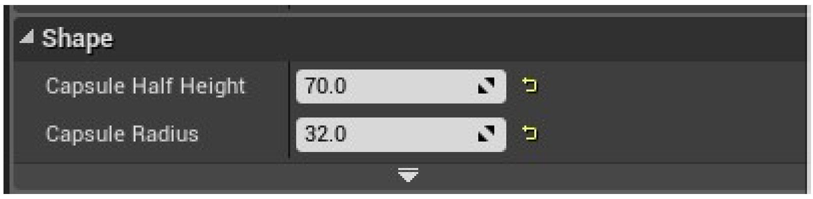


Character BP file name is BP_Char02 and character animation BP is Char02_AnimBP.
First, open the BP_Char02 and set the mesh, capsule component, Box Collision to judge attack and damage.



・ Capsule component

Set the value of Shape as follows.

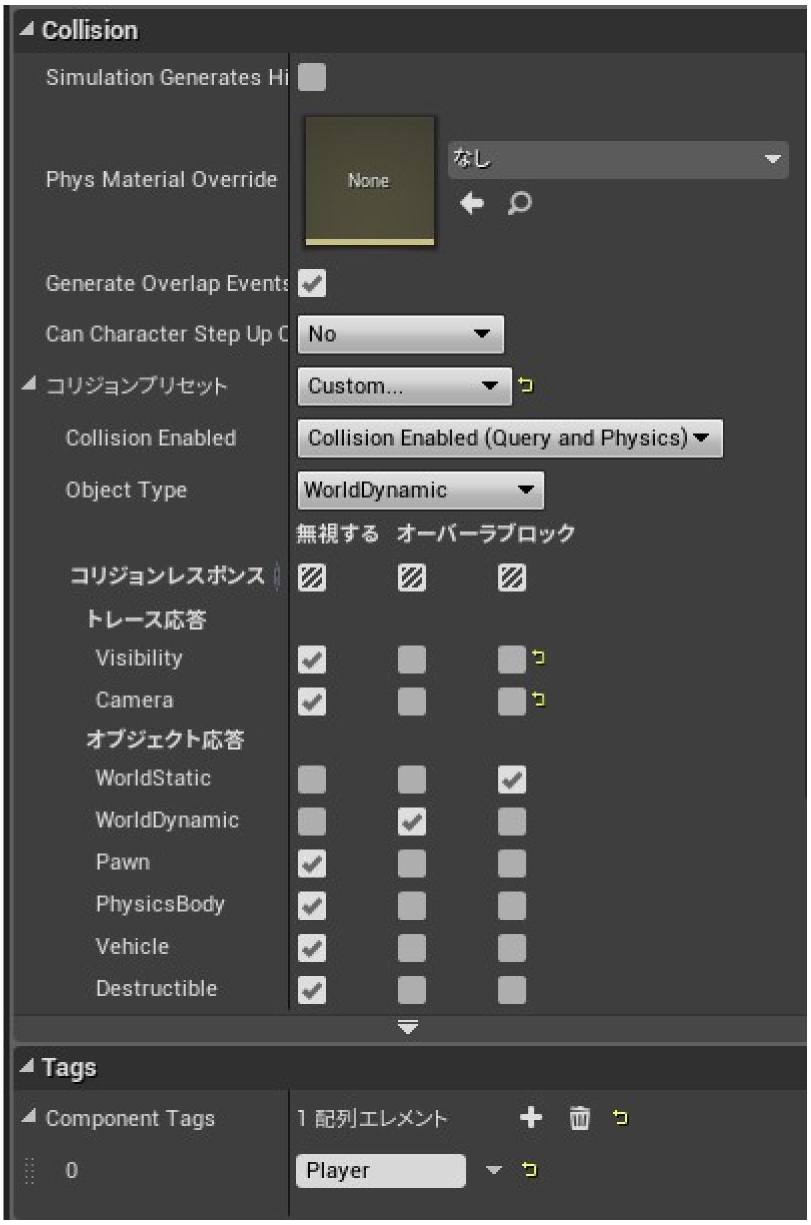


Shape

Capsule Half Height	70.0
Capsule Radius	32.0

Set Collision and Tag as follows:

Unreal Engine Fighting Game Template project　Manual



Collision

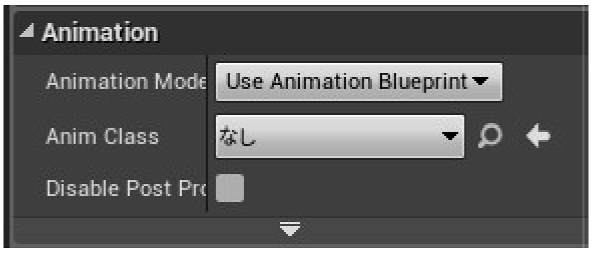
Simulation Generates Hi	☐		
Phys Material Override	None / なし		
Generate Overlap Events	✓		
Can Character Step Up C	No		
コリジョンプリセット	Custom...		
Collision Enabled	Collision Enabled (Query and Physics)		
Object Type	WorldDynamic		
	無視する	オーバーラップ	ブロック
コリジョンレスポンス	▨	▨	▨
トレース応答			
Visibility	✓	☐	☐
Camera	✓	☐	☐
オブジェクト応答			
WorldStatic	☐	☐	✓
WorldDynamic	☐	✓	☐
Pawn	✓	☐	☐
PhysicsBody	✓	☐	☐
Vehicle	✓	☐	☐
Destructible	✓	☐	☐

Tags

Component Tags	1 配列エレメント
0	Player

・ Mesh

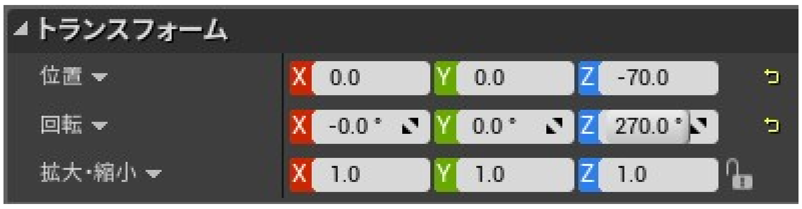
Set the animation Anim Class to “None”.
Animation Blueprint is not required when selecting a character, so call it manually before the battle.



Animation

Animation Mode	Use Animation Blueprint
Anim Class	なし
Disable Post Pro	☐

Set the mesh position and rotation as follows.



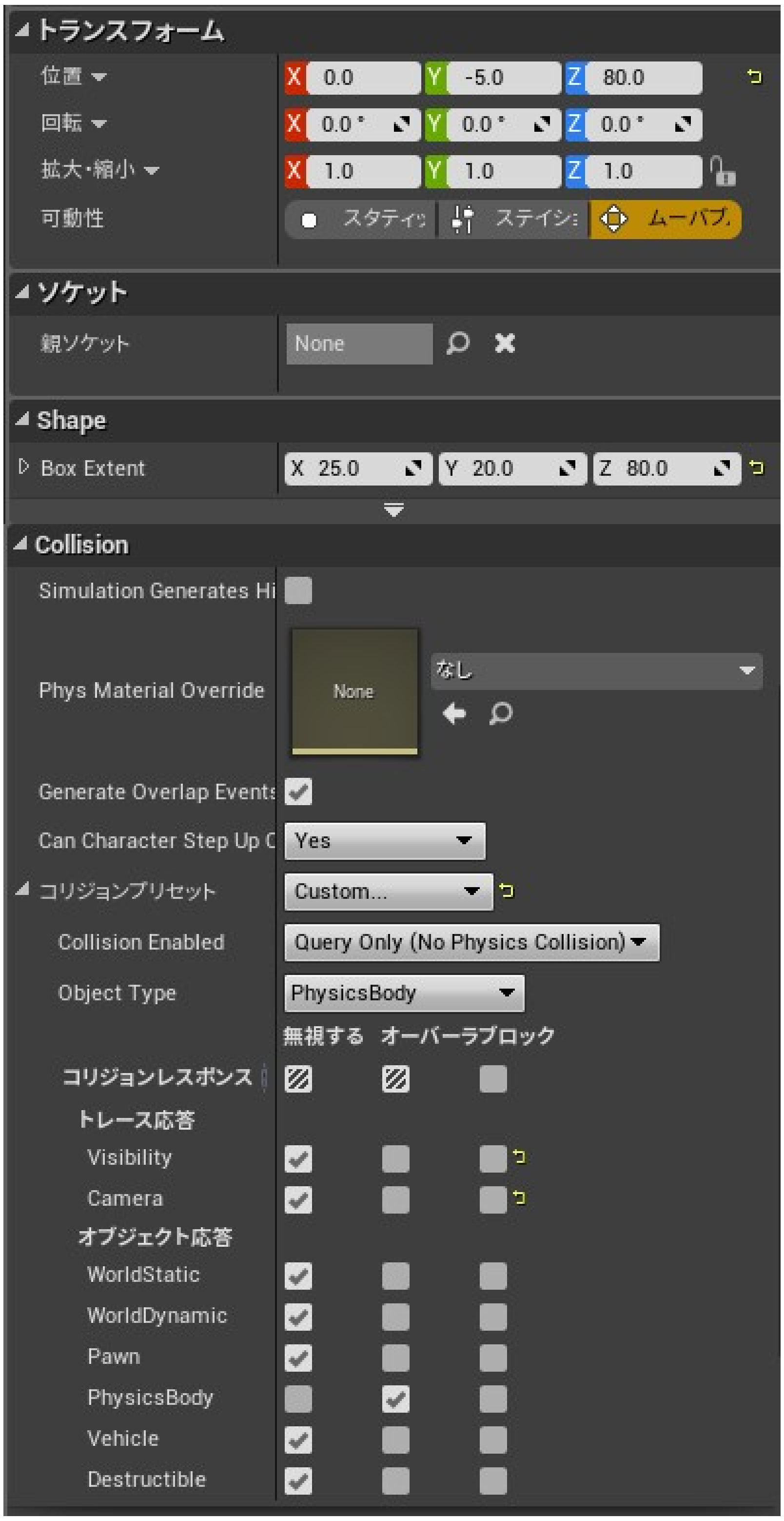
トランスフォーム

	X	Y	Z
位置	0.0	0.0	-70.0
回転	-0.0°	0.0°	270.0°
拡大・縮小	1.0	1.0	1.0

Unreal Engine Fighting Game Template project　Manual

・ Damge Box Collision　（Add as a subcomponent of Mesh.）

Set the position, rotation, shape, and collision as follows.
The position and shape are set according to the character's physique,
so you can change them. However, if you make the height too low, you will not be able to hit other characters' upper attacks.



トランスフォーム

位置	X 0.0	Y -5.0	Z 80.0
回転	X 0.0°	Y 0.0°	Z 0.0°
拡大･縮小	X 1.0	Y 1.0	Z 1.0
可動性	スタティッ	ステイシ:	[✓ ムーバブ.]

ソケット

親ソケット: None

Shape

Box Extent: X 25.0 Y 20.0 Z 80.0

Collision

Simulation Generates Hi: ☐

Phys Material Override: None なし

Generate Overlap Events: ✓

Can Character Step Up C: Yes

コリジョンプリセット: Custom...

Collision Enabled: Query Only (No Physics Collision)

Object Type: PhysicsBody

	無視する	オーバーラップ	ブロック
コリジョンレスポンス	▨	▨	☐
トレース応答			
Visibility	✓	☐	☐
Camera	✓	☐	☐
オブジェクト応答			
WorldStatic	✓	☐	☐
WorldDynamic	✓	☐	☐
Pawn	✓	☐	☐
PhysicsBody	☐	✓	☐
Vehicle	✓	☐	☐
Destructible	✓	☐	☐

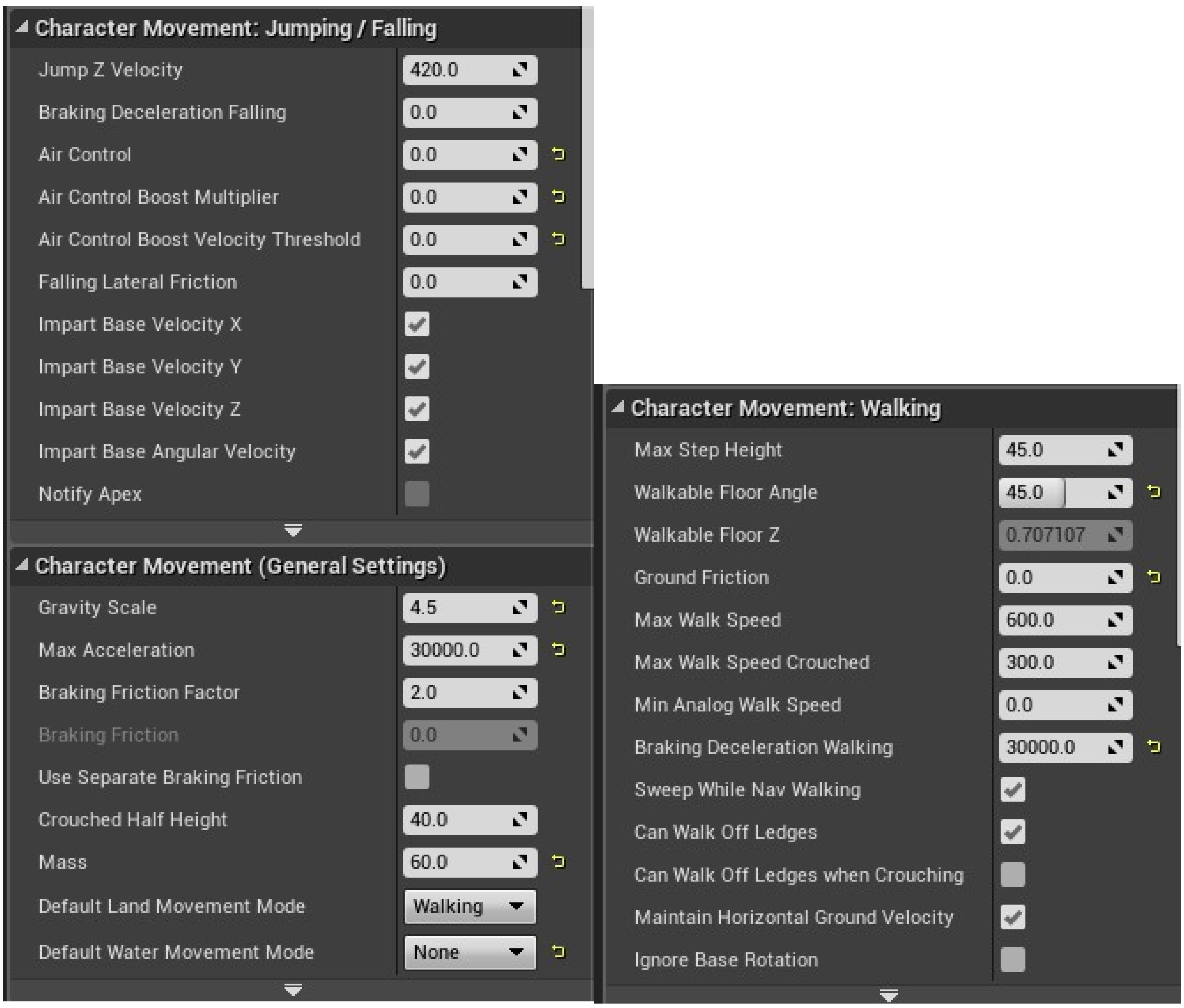
Unreal Engine Fighting Game Template project Manual

・ About other Box Collision

For Arm_R / Arm_L / Foot_R / Foot_L / Damage_for_bullet / Cross_attack / Damage_for_Throw / Throw, set the position and rotation, Shape, parent socket and collision with reference to the setting value of Char01.
Note the name of each Box Collision.
I will copy the Blueprint of Char01 to the event graph from now on, so it will not work correctly if I make a mistake.

・ About CharacterMovement

Set the values as follows:



Character Movement: Jumping / Falling

Jump Z Velocity	420.0
Braking Deceleration Falling	0.0
Air Control	0.0
Air Control Boost Multiplier	0.0
Air Control Boost Velocity Threshold	0.0
Falling Lateral Friction	0.0
Impart Base Velocity X	✓
Impart Base Velocity Y	✓
Impart Base Velocity Z	✓
Impart Base Angular Velocity	✓
Notify Apex	☐

Character Movement (General Settings)

Gravity Scale	4.5
Max Acceleration	30000.0
Braking Friction Factor	2.0
Braking Friction	0.0
Use Separate Braking Friction	☐
Crouched Half Height	40.0
Mass	60.0
Default Land Movement Mode	Walking
Default Water Movement Mode	None

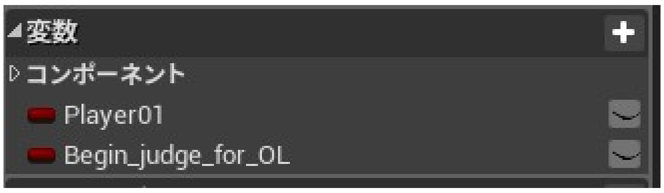
Character Movement: Walking

Max Step Height	45.0
Walkable Floor Angle	45.0
Walkable Floor Z	0.707107
Ground Friction	0.0
Max Walk Speed	600.0
Max Walk Speed Crouched	300.0
Min Analog Walk Speed	0.0
Braking Deceleration Walking	30000.0
Sweep While Nav Walking	✓
Can Walk Off Ledges	✓
Can Walk Off Ledges when Crouching	☐
Maintain Horizontal Ground Velocity	✓
Ignore Base Rotation	☐

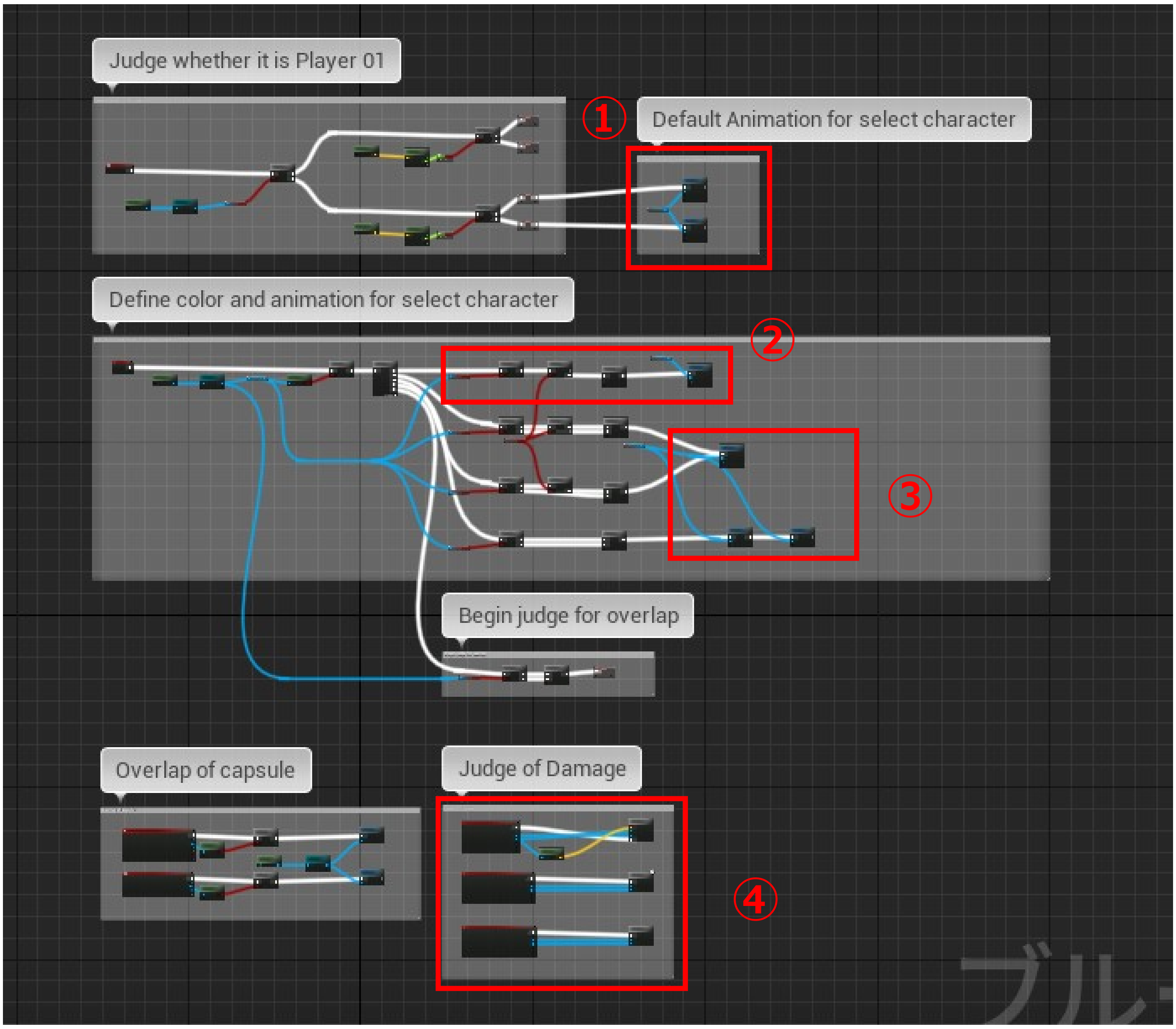
Unreal Engine Fighting Game Template project Manual

・ About event graph

First, create a variable as follows:



Next, select and copy and paste the entire Blueprint of Char01's event graph.
However, since there is a macro inside the ④red frame, the macro is not copied, so copy and paste the node where the macro was expanded in the event graph of Char01,
and fold it or make it into a macro.



Judge whether it is Player 01
①
Default Animation for select character
Define color and animation for select character
②
③
Begin judge for overlap
Overlap of capsule
Judge of Damage
④
ブル

Next, the points that need to be set individually for each character are explained.
Red frame ① in the figure above: Select the animation sequence before character selection on the character selection screen.
The top and bottom two are for Player 1 and 2.
Red frame ② in the above figure: Processing when the same character as Player1 is selected when Player2 character is selected.

Change the color of the material in the character mesh or change the mesh itself.
Note that if you change the mesh itself, you will need to prepare the animation sequence and animation montage separately for that mesh.
Red frame ③ in the figure above: Set the animation sequence when selecting a character and the animation blueprint to be used.
This completes the Char02 Blueprint settings.

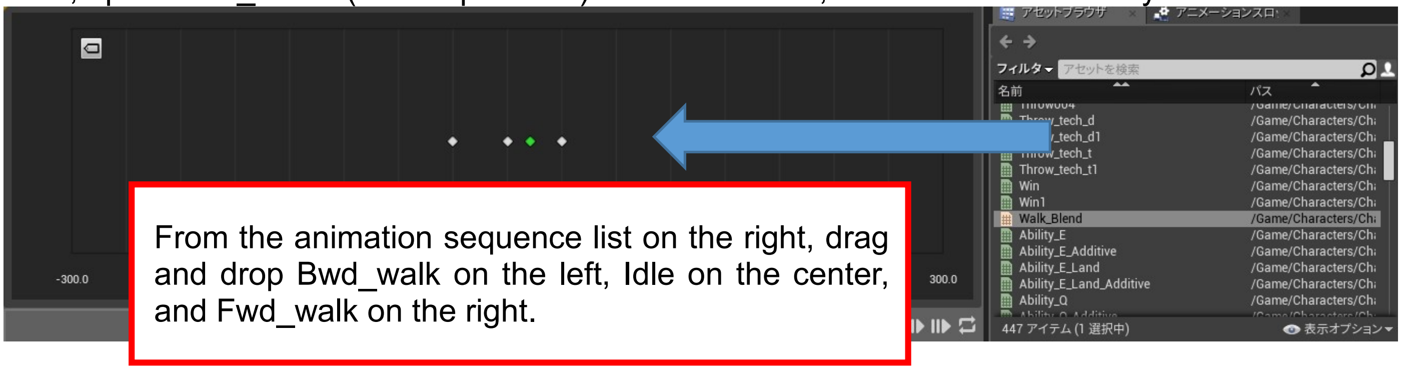
Next, set up an animated blueprint for Char02_AnimBP.
An animation sequence and an animation montage are required to set up an animation blueprint, so the list is shown below.

List of animation sequences that need to be prepared.	
Standing standby state	Animation when nothing is input while standing. Applicable to Char01 Idle.
Walk forward	Animation when moving forward. Applicable to Char01's Fwd_Walk.
Walking backwards	Animation when retreating. Applies to Char01's Bwd_Walk.
Crouch standby	Animation of crouching. Applicable to Char01's Crouch_Idle.
Crouching → standing	Animation when standing up from crouching. Applies to Char01 End_Crouch.
Jump start	Animation when jumping from standing. Applicable to Start_Jump of Char01.
Air standby state	Animation when the jump is complete. Applicable to Char01 Loop_Jump.
Landing	Animation when landing from a jump. Applicable to Char_End_Jump.
Step forward	Animation when pre-stepping. Applicable to Char01's Fwd_Dash.
Backstep	Animation when backstepping. Applies to Bwd_Step of Char01.
Standing punch	Animation when standing punch. This applies to ST_Punch of Char01.
Standing kick	Animation when standing and kicking. Applicable to Char01's ST_Kick.
Jumping punch	Animation when jump punching. Applicable to Air_Punch of Char01.
Jumping kick	Animation when jump kicking. Applicable to Air_Kick of Char01.
Crouching punch	Animation when crouching punch. Applies to CR_Punch of Char01.
Crouching kick	Animation when crouching and kicking. Applicable to CR_Kick of Char01.
Standing guard	Animation when standing guard. Applies to the Guard of Char01.
Crouching guard	Animation when crouching guard. Applicable to Char01's C_Guard.
Small damage at the top	Animation of small damage on the upper row. Applies to Damaged_Top of Char01.
Large damage at the top	Animation of heavy damage in the upper row. Applies to Damaged_Top_L of Char01.
Upper middle damage	Animation of middle damage in the upper row. Applies to Damaged_Mid of Char01.
Crouching damage	Animation of crouching damage. Applicable to C_Damaged of Char01.
Small damage in the air	Animation of small damage in the air. Applies to Damaged_inAir of Char01.
Heavy damage in the air	Animation of heavy damage in the air. Applies to Damaged_inAir_L of Char01.
Counter damage	Counter damage animation. Applicable to Counter of Char01.
KO motion	Animation when KO on the ground. Applicable to KO of Char01.
down	Animation when under attack of down attribute. Applies to Char01 Down.
Usually get up	Animation when getting up. Applies to Get_up01 of Char01.
Get up stun	Animation when getting up from stun. Applicable to Get_up02 of Char01.
Stun	Stunned animation. Applies to Char01's Stun.
Throw	Animation that grabs the opponent. Applies to Throw001 of Char01.
Throwing success	Animation to grab and throw an opponent. Applies to Throw002 of Char01.
Thrown	A thrown animation. Applies to Damaged_Throw of Char01.
Throw-tech attack side	Attacker animation when thrown away. Applies to Throw_tech_A of Char01.
Throw-tech defense side	Animation when thrown through. Applies to Char01's Throw_tech_D.
Special Move (FireBall)	FireBal animation. Applies to Char01's FireBall.
Special Move (IceBall)	IceBall animation. Applicable to SP of Char01.
Round win pose	Animation of round victory. Applicable to Char01 Win.
Round loser pose	Animation of a round loser on timeout. Applies to Draw for Char01.
Game victory pose	Animation of the game victory. Applicable to Char01 GameSet.
Change side	Animation when the character changes direction. Applies to Side_Change of Char01.

List of animation montages that need to be prepared	
ST_Attack	Register a standing attack animation. In AnimNotify, set the time zone to receive counter-attack, the time zone when the attack judgment occurs, the time zone to accept the cancellation to the Special Move, the disappearance of the attack judgment, and the off attack flag.※
CR_Attack	Register a crouching attack animation. AnimNotify is the same as ST_Attack.
Attack_in_Air	Register an attack animation in the jump state. AnimNotify sets the time period when the attack judgment occurs, the attack judgment disappearance, and the off-attack flag is off.
Dash	Register animation of previous step and back step. AnimNotify sets the timing to cancel the invincible time of step off and back step.
Guard	Register the standing guard and crouching guard animations. There is no setting for AnimNotify.
Damage	Register various animations when receiving damage. AnimNotify sets the timing for turning off the damage flag and turning off the counter flag.
Throw	Register throw and throw-tech animations. In AnimNotify, set the time zone to receive a counter attack, the time zone in which an attack judgment occurs, the time zone to allow a throw-tech, the loss of attack judgment, and the off-attack flag off.
Other	Register animations of down, getting up, stun, KO, and direction change. In AnimNotify, set the down or getting up flag off.
Result	Register victory and defeat animations. There is no setting for AnimNotify.

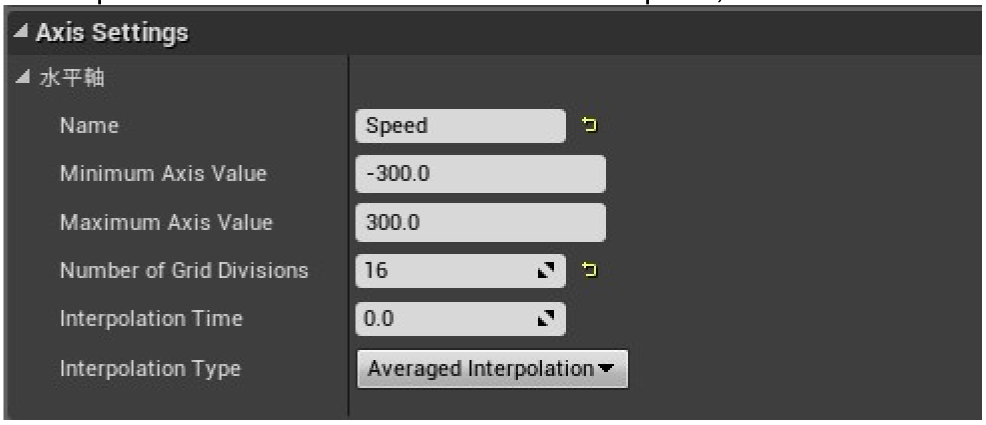
※By retargeting the Char01 animation montage with the Char02 mesh and swapping the animation sequences in the animation montage, you can reuse AnimNotify and just adjust the notification timing.

Next, open Walk_Blend (blend space 1D) and set forward, backward and standby states.



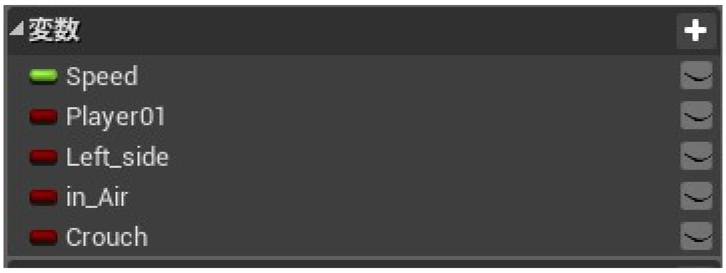
From the animation sequence list on the right, drag and drop Bwd_walk on the left, Idle on the center, and Fwd_walk on the right.

Then set the values as follows:
It is important to set the horizontal Name to Speed, and other values are reference values.

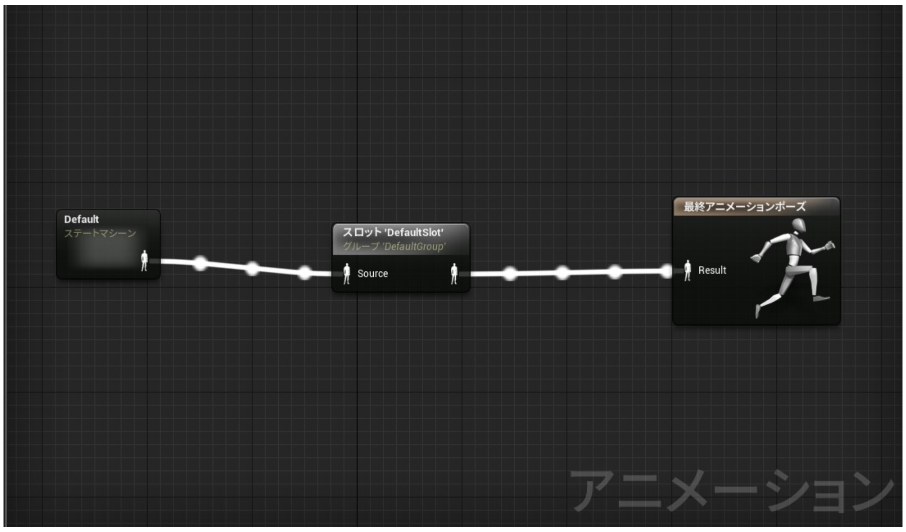

Axis Settings
水平軸

Name	Speed
Minimum Axis Value	-300.0
Maximum Axis Value	300.0
Number of Grid Divisions	16
Interpolation Time	0.0
Interpolation Type	Averaged Interpolation

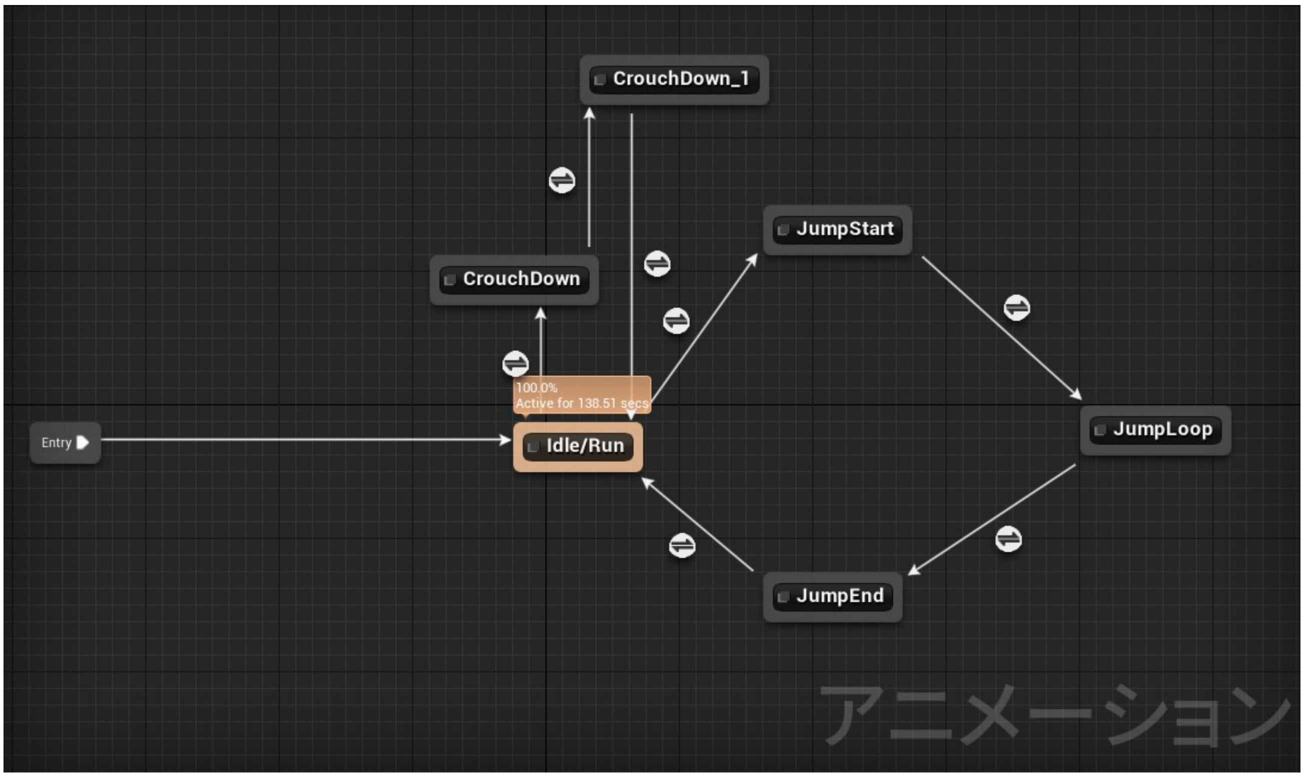
Now that the animation sequence and montage are ready, open Char02_AnimBP and edit the state machine.
First, set the variables as follows.


変数
Speed
Player01
Left_side
in_Air
Crouch

Next, set the anim graph as follows.

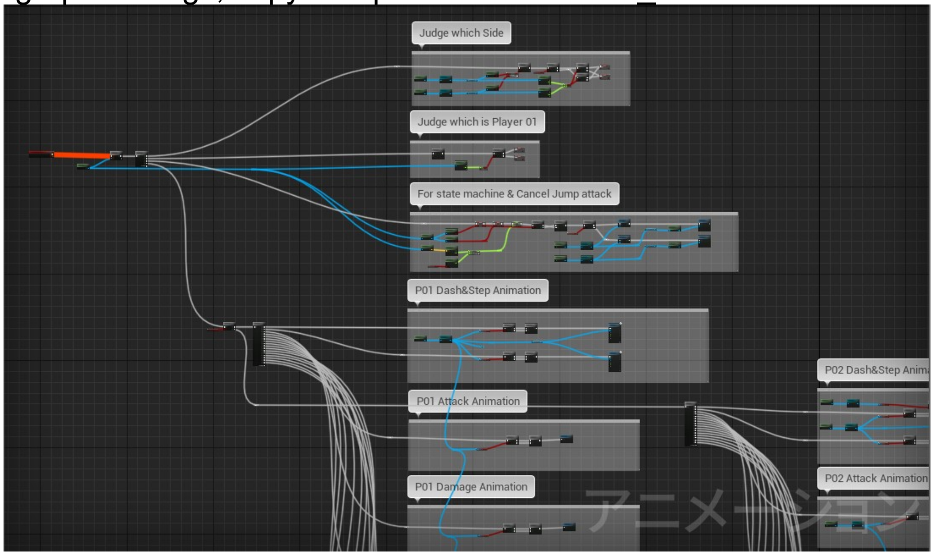


Next, set the state machine as follows.

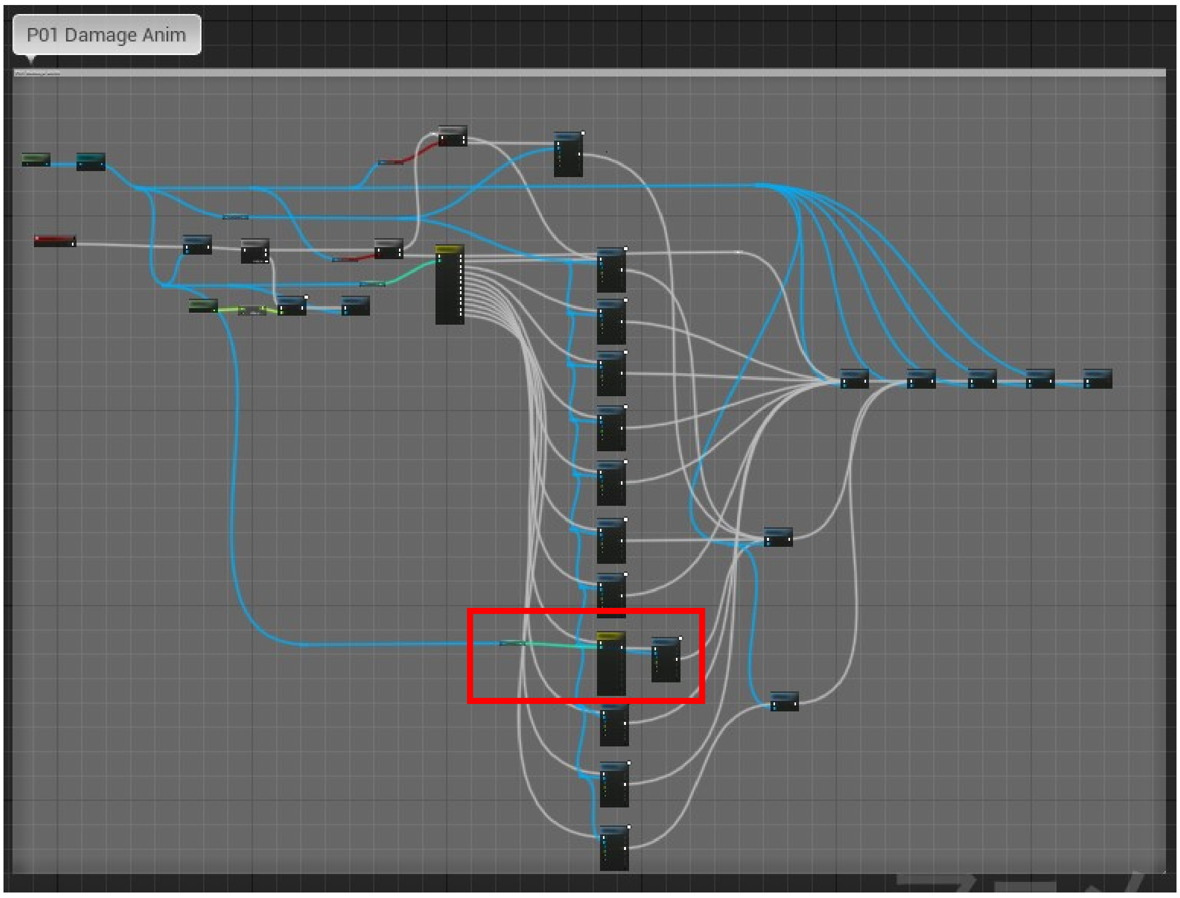


For details on each item, refer to the Char01 state machine.

Next, for event graph settings, copy and paste from Char01_AnimBP.

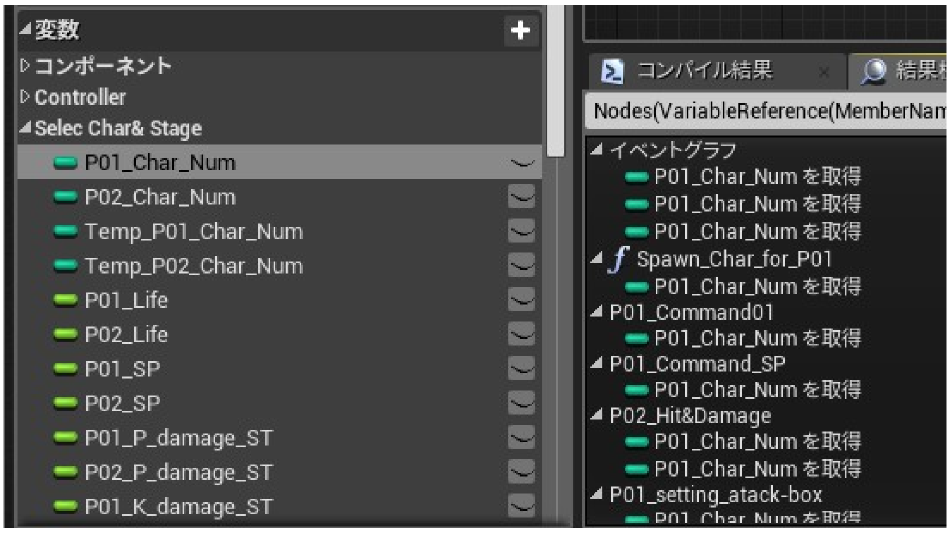


Build after copying and pasting everything.
Probably you will get errors in some custom events.
If you just copy and paste, it may not be valid. In that case, recreate the same node and build it.
Once the build is complete, set the animation montage referenced by Play Montage and Montage Stop to the Char02 montage.
Just copying it refers to the montage of Char01 and does not work even if the build passes.
Also, in the node that is processing the animation when taking damage, the animation when thrown may be unique to each character, so make individual settings in the red frame part of the image below.



When all references to Play Montage and Montage Stop are changed to Char02, Char02_Anim settings are complete.

Next, set BP_Common_Controller.
Perform a reference search by right-clicking each of the variables P01_Char_Num and P02_Char_Num, and set Char02 while referring to the Char01 settings at each search destination.

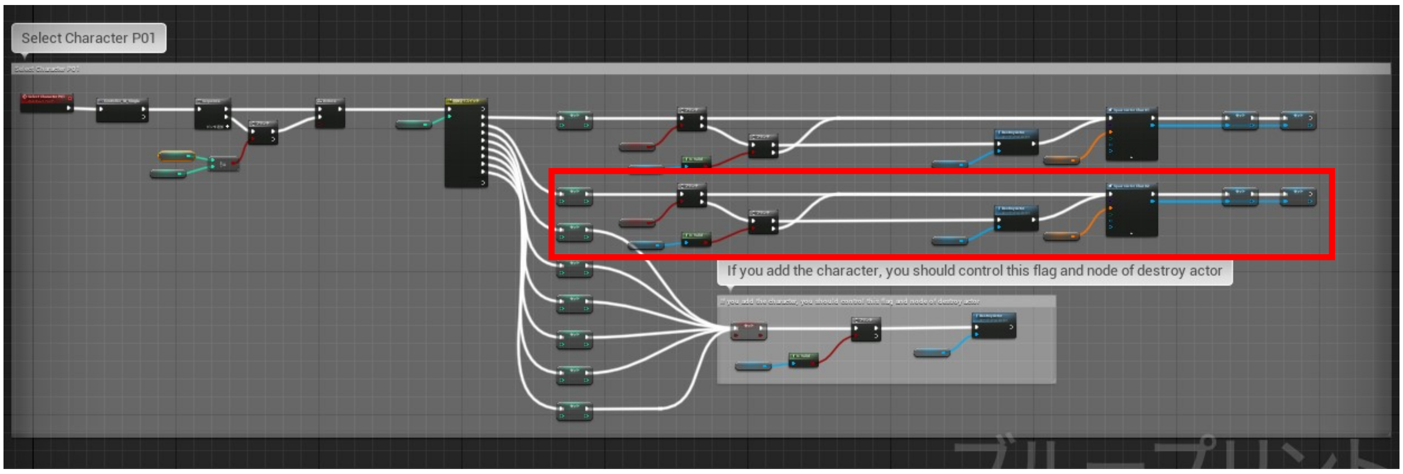


◢変数
▷コンポーネント
▷Controller
◢Selec Char& Stage
P01_Char_Num
P02_Char_Num
Temp_P01_Char_Num
Temp_P02_Char_Num
P01_Life
P02_Life
P01_SP
P02_SP
P01_P_damage_ST
P02_P_damage_ST
P01_K_damage_ST

コンパイル結果　結果
Nodes(VariableReference(MemberNam

◢イベントグラフ
P01_Char_Num を取得
P01_Char_Num を取得
P01_Char_Num を取得
◢ƒ Spawn_Char_for_P01
P01_Char_Num を取得
◢P01_Command01
P01_Char_Num を取得
◢P01_Command_SP
P01_Char_Num を取得
◢P02_Hit&Damage
P01_Char_Num を取得
P01_Char_Num を取得
◢P01_setting_atack-box
P01_Char_Num を取得

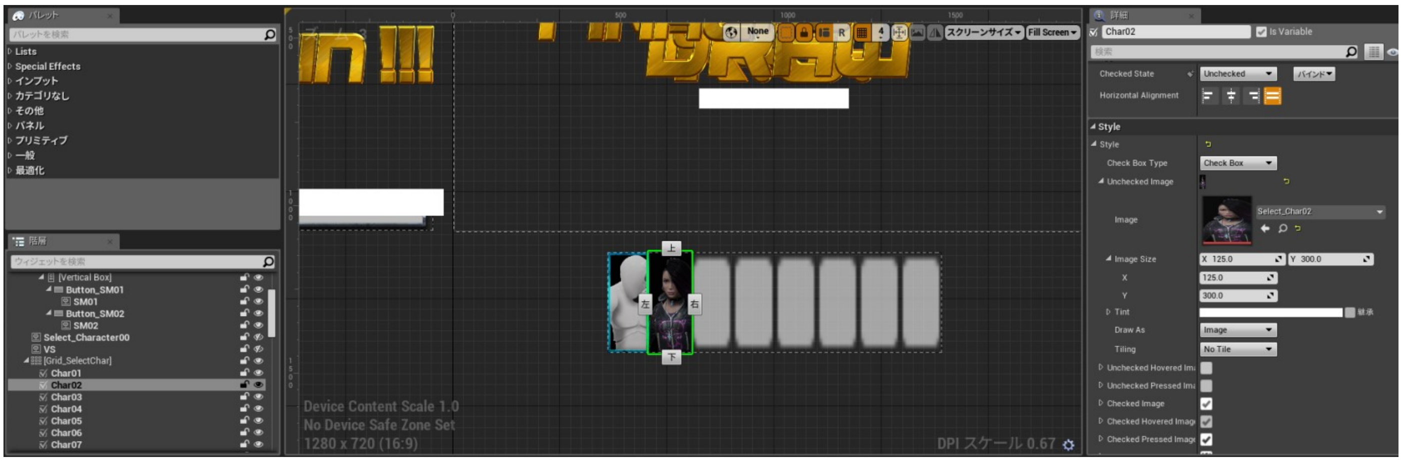
For example, the following image will be the processing node for character selection in Player01, but processing for Char02 in the red frame is added.


Select Character P01
If you add the character, you should control this flag and node of destroy actor
ブループリント

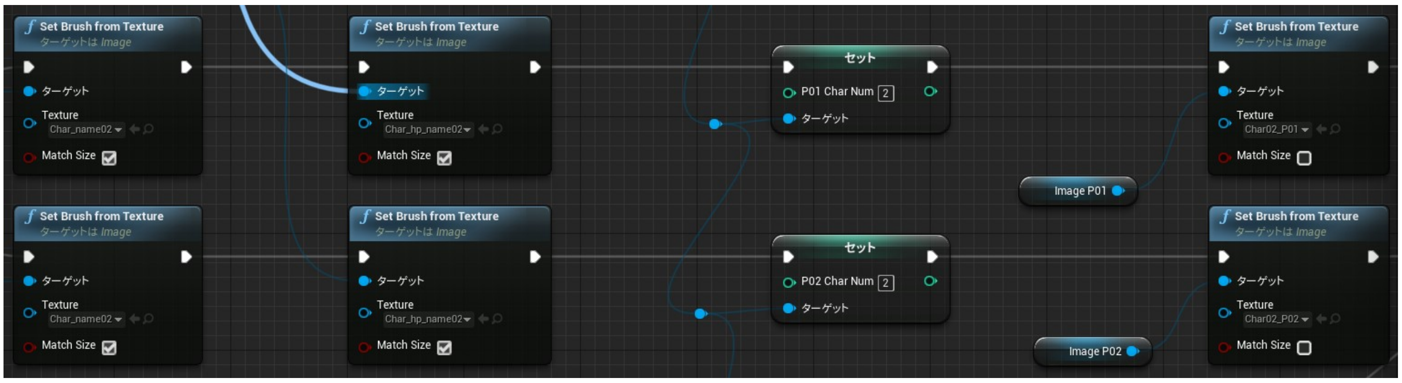
After adding all the processing, Common_Controller settings are complete.

Unreal Engine Fighting Game Template project Manual

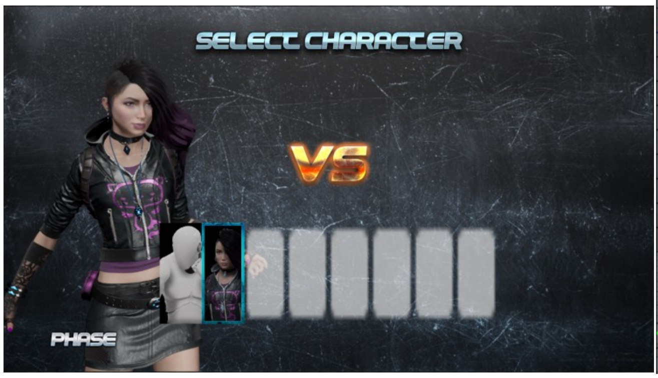
Finally, add images for character selection and battle.
First, add an image to the character select grid background on the designer screen as shown below.(125 * 300pixel)

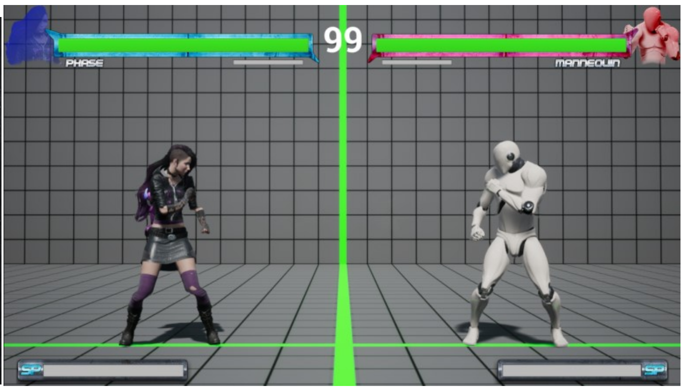

Char02 ✓ Is Variable
Checked State: Unchecked
Style
Check Box Type: Check Box
Unchecked Image
Image: Select_Char02
Image Size: X 125.0 Y 300.0
Draw As: Image
Tiling: No Tile

Next, set the Reference Texture as shown in the image below in the Select Character function in the event graph.
Char_name02 (300 * 50pixel): Name displayed below the character when the character is selected.
Char_hp_name02 (250 * 30pixel): Name displayed below the HP bar during battle.
Char02_P01 / Char02_P02 (304 * 335pixel) : Character image displayed next to HP bar during battle.


Set Brush from Texture (ターゲットは Image) — ターゲット; Texture: Char_name02; Match Size ✓
Set Brush from Texture (ターゲットは Image) — ターゲット; Texture: Char_hp_name02; Match Size ✓
セット — P01 Char Num 2; ターゲット
Image P01
Set Brush from Texture (ターゲットは Image) — ターゲット; Texture: Char02_P01; Match Size ☐
Set Brush from Texture (ターゲットは Image) — ターゲット; Texture: Char_name02; Match Size ✓
Set Brush from Texture (ターゲットは Image) — ターゲット; Texture: Char_hp_name02; Match Size ✓
セット — P02 Char Num 2; ターゲット
Image P02
Set Brush from Texture (ターゲットは Image) — ターゲット; Texture: Char02_P02; Match Size ☐

After building and running, it is OK if it is reflected as shown below.





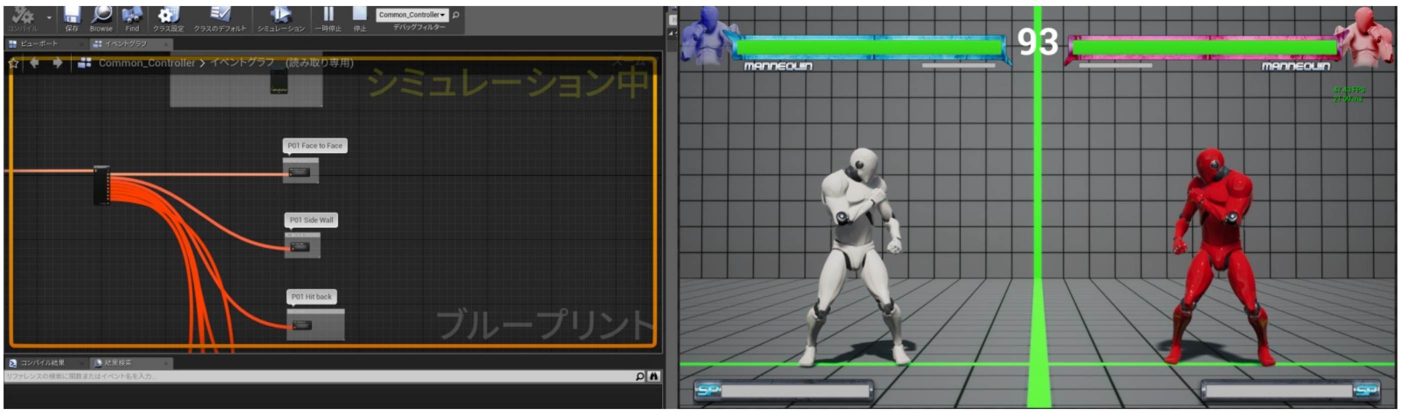
Unreal Engine Fighting Game Template project　Manual

5．Notes on debugging

When checking the operation, be sure to check the frame rate.

（Enter the Stat fps command in the output log or set the value obtained by dividing 1 by WorldDeltaSeconds to PrintString connected to Tick.）

If you operate with a reduced frame rate, you may experience a bug that occurs only when the frame rate is reduced.
As shown in the image below, it is confirmed that the frame rate decreases when playing while checking the execution of Blueprint.
In order to solve this problem, you can check the execution of Blueprint while keeping 60fps by using the multiple display and running Blueprint execution window and play window on different monitors.
However, if you zoom out the blueprint execution confirmation range to a wide range, the frame rate may decrease even if you use multiple displays.



End.

RareEncounter Ver.1.0