UNREAL
ENGINE

Unreal Engine 5.2
注目機能の紹介

- GTMF 2023 -

エピック ゲームズ ジャパン
岡田 和也, 鈴木 孝司

はじめに

本講演では UE5.2 における機能追加・改善の
すべて紹介することは **しません**

UE5.2 リリースノートから（個人的に）重要なものを紹介したり
少し分かりづらい項目の補足などを行います

https://docs.unrealengine.com/5.2/ja/unreal-engine-5.2-release-notes/

Nanite, Lumen などのレンダリング機能や PCG は この後の講演で

Unreal Engine 5.2 アップデート ～Rendering/PCG～

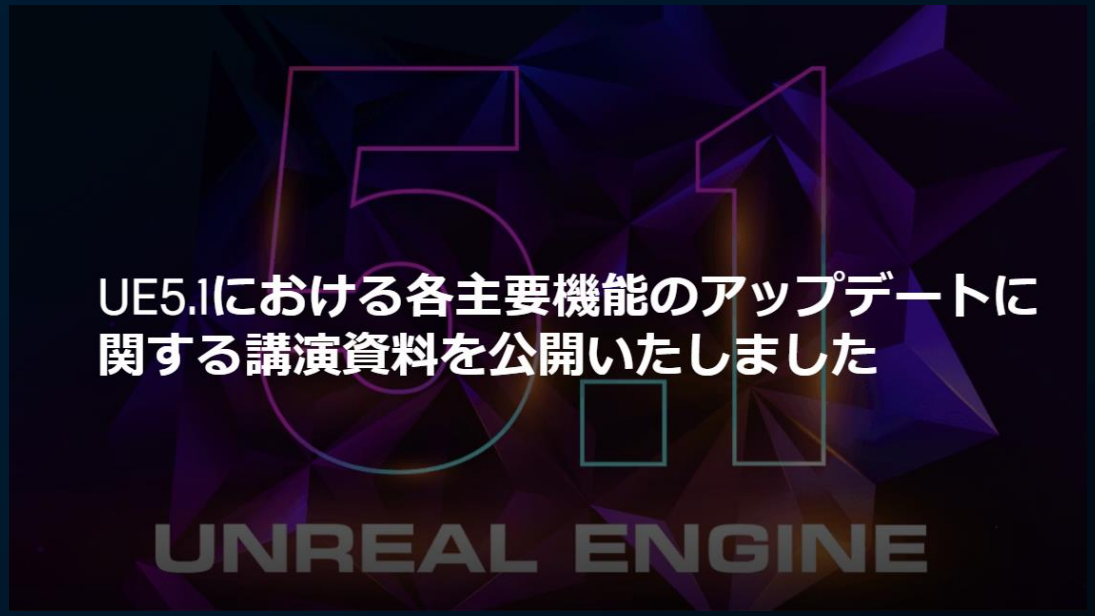
エピック ゲームズ ジャパン

プランナー　プログラマー (クライアント)　デザイナー(2D/3D)

Unreal Engine 5.2では様々な新機能・改善が入り、グラフィックス関連ではLumenやNaniteといったUE5の注目機能が更に強化され、新たにSubstrateと呼ばれるマテリアルの作成手法が導入されました。

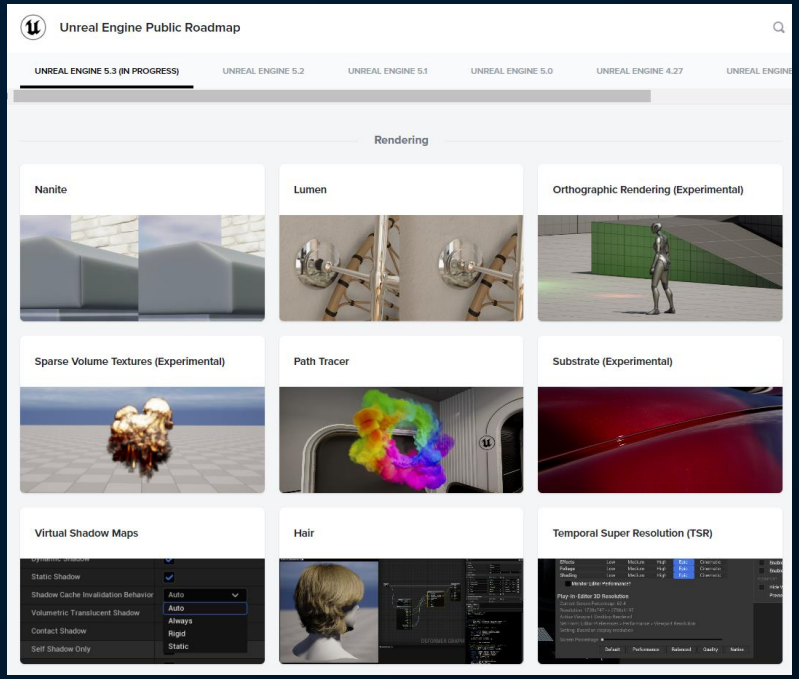
その他にも大規模なワールドを効率的に作成するためのPCG(Procedural Content Generation Framework)など様々な機能が追加されており、本講演ではそれらの中から特にオススメしたい機能についてご紹介いたします。

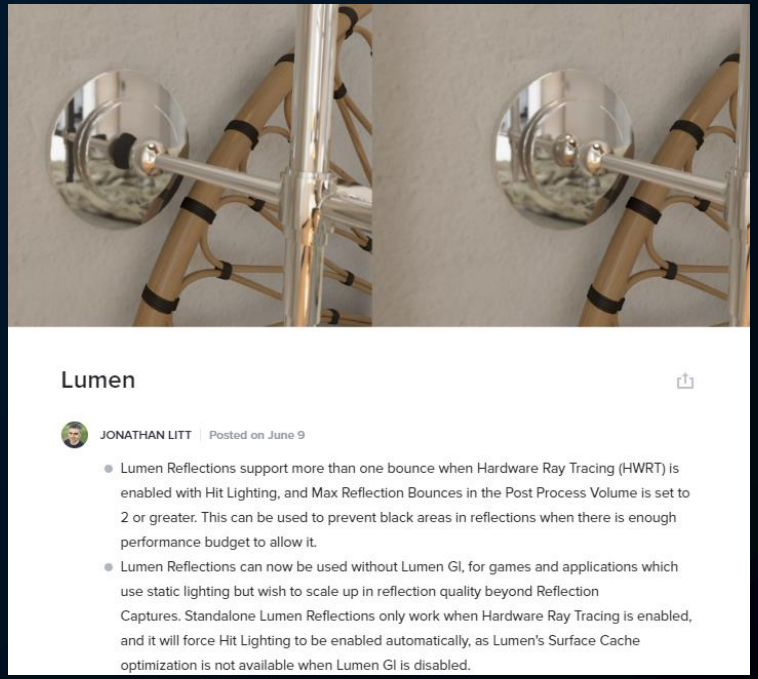
UE5.1 におけるアップデート情報はこちら



https://www.unrealengine.com/ja/events/epicgamesjapan-ue5-1-updateinfo

UE5.3 におけるアップデート情報はこちら





https://portal.productboard.com/epicgames/1-unreal-engine-public-roadmap/tabs/88-unreal-engine-5-3-in-progress

Animation

UE5.2 アップデート

Python リターゲティングのサポート

Python リターゲティングのサポート

カスタム仕様の Python スクリプトを実装して、
IK リグを使ったリターゲティング プロセスを
自動化および高速化できるようになりました

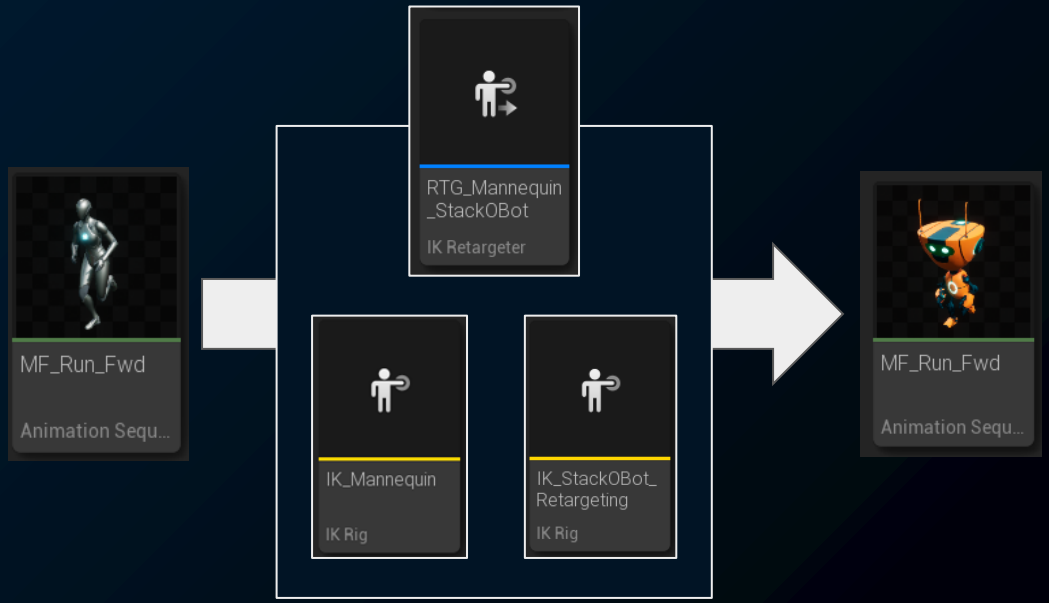
Python リターゲティングのサポート

カスタム仕様の Python スクリプトを実装して、
IK リグを使ったリターゲティング プロセスを
自動化および高速化できるようになりました

IK Rig・IK Retargeter を
Blueprint または Python から自動生成したり
リターゲット処理を実行できるようになりました

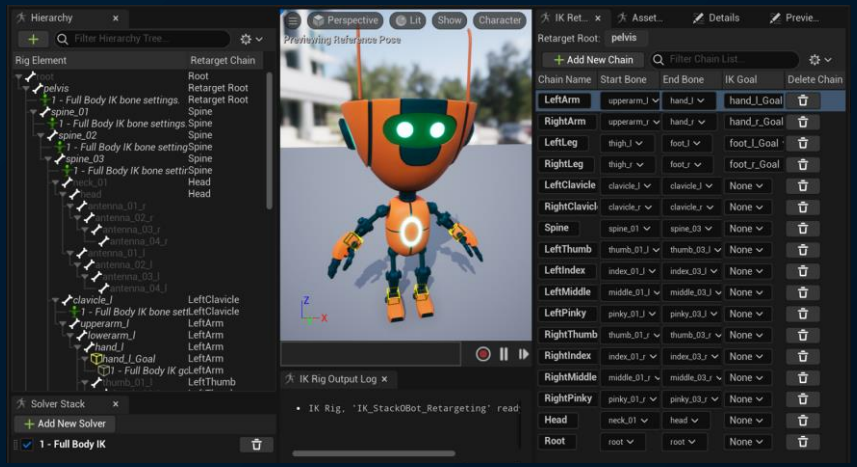
IK Rig / IK Retargeter

ボーン構造が異なっていても、アニメーションを流用するための仕組み
UE5でのアニメーションの流用方法について



IK Rig / IK Retargeter のこれまでの課題

- セットアップ項目が多い
- Skeleton毎にアセットを作成・調整する必要がある



IK Rig

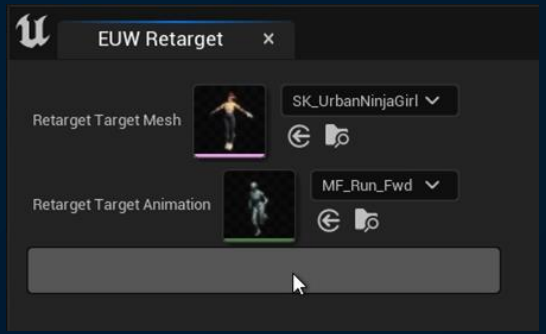
IK Retargeter

UE5.2から BP / Python で生成・設定可能に


```python
fbik_index = ikr_controller.add_solver(
unreal.IKRigFBIKSolver)
ikr_controller.set_root_bone("pelvis", 0)
```

```python
ikr_controller.add_new_goal("hand_l_goal",
"hand_l")
ikr_controller.add_new_goal("hand_r_goal",
"hand_r")
```

ボタン1つで、面倒な作業を自動化・実行！



EditorUtilityで
作成した自作ツール

設定済みのIK Rig, Retargeterを自動生成
アニメーションリターゲットを実行

https://www.youtube.com/watch?v=lt-j0OICUf4

学習リソース

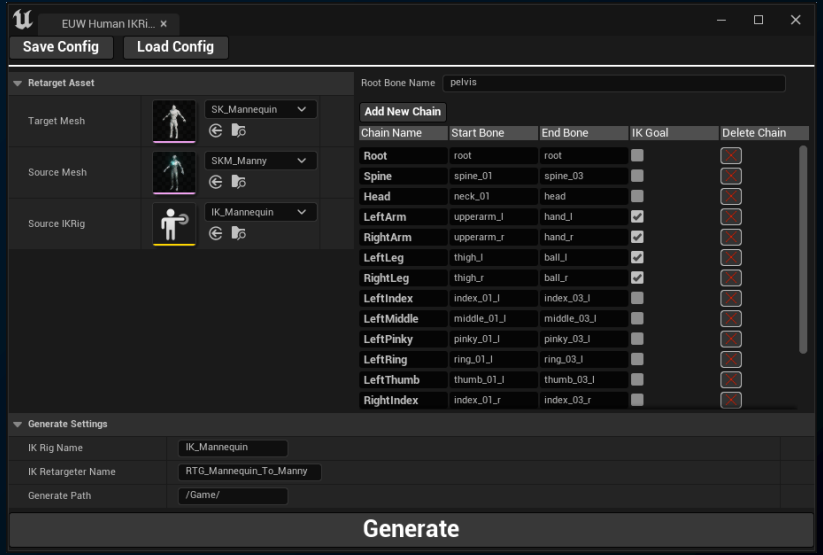
Python で IK リグを使用する
https://docs.unrealengine.com/5.2/ja/using-python-to-create-and-edit-ik-rigs-in-unreal-engine/

IKリターゲッタで Python を使用する
https://docs.unrealengine.com/5.2/ja/using-python-to-create-and-edit-ik-retargeter-assets-in-unreal-engine/

BPで自動生成・実行するサンプル
https://dev.epicgames.com/community/snippets/J55V/unreal-engine-generate-ikrig-and-ikretargeter-with-blueprint-bp-ikrig-ikretargeter

UE5.2の新機能を活かした生成ツールが！

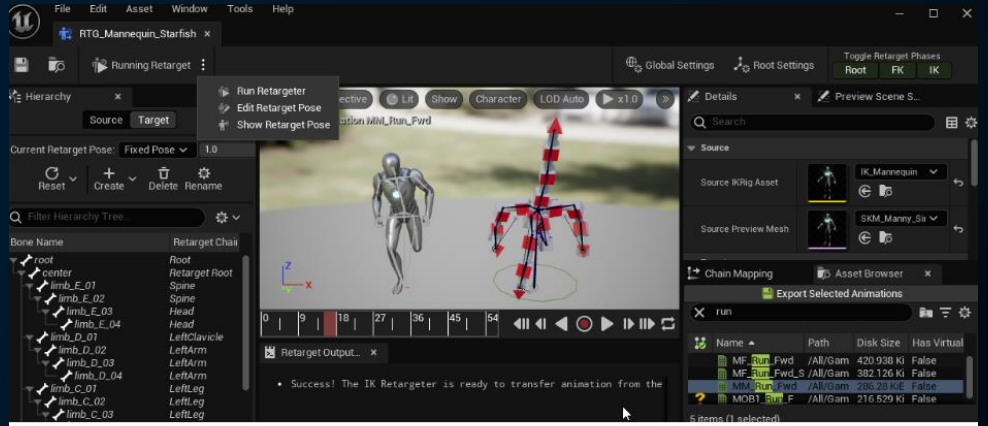
UE専門のゲームデベロッパーである株式会社Leon Gameworks様のツイート



https://github.com/LeonGameworks/Human_IKRigGenerator

UE5.3にて
IK Retargeterエディタへの機能追加など様々な改善が入る予定

Unreal Engine Public Roadmap より
https://portal.productboard.com/epicgames/1-unreal-engine-public-roadmap/c/1076-animation-retargeting



Animation Retargeting

GREG RICHARDSON | Posted on April 18

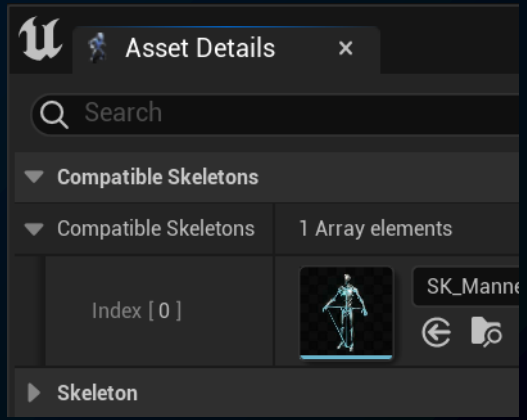
This release focuses on quality-of-life improvements for Retargeting.

- Toolbar in Retarget Editor - Easy access to Global and Root Settings along with the Retarget Phases. Mode switches between Running the Retarget, editing the Retarget Pose, and viewing the Retarget Pose.
- Retarget Poses - Edit a bone's local space to zero out bone rotations.
- Create IK Rig/IK Retargeter assets from a selected Skeletal Mesh/IK Rig in the Right Mouse Context Menu.
- Thumbnails for IK Rig and IK Retargeter assets.

スケルトンの互換性 の改善
（ Compatible Skeleton ）

スケルトンの互換性について

Skeletonアセットが異なっていても
Compatible（互換性）があるから アニメーションの共有はOK！という機能



設定画面

https://www.docswell.com/s/EpicGamesJapan/KQN3EK-UE5-ShareAnimation

スケルトンの互換性に関する改善点

互換性の設定が双方向に

- Skeleton A -> B の設定をすれば、 A <- B も互換性アリ状態に

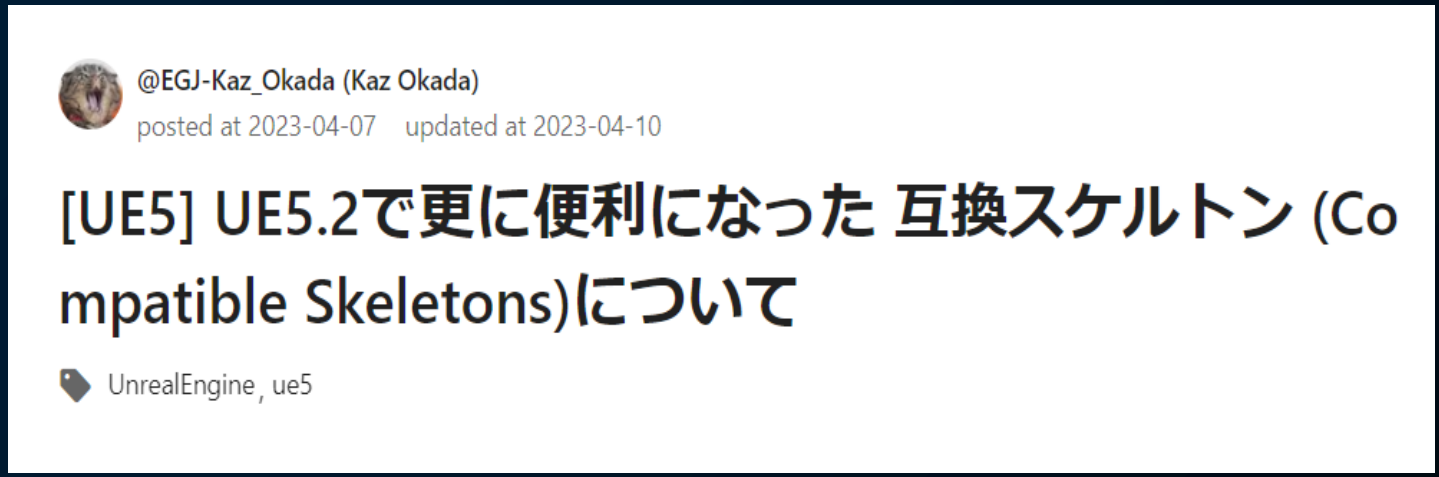
互換性の設定をしなくても、
他Skeletonのアニメーションが利用可能に

- Skeleton が異なっていても
  ボーン名・構造の比較でアニメーション再生をするように

互換性の設定がEditor Onlyに

- リマッピング処理がランタイム上で実行されるように
- 余計な参照関係を回避

学習リソース



https://qiita.com/EGJ-Kaz_Okada/items/387b73923089c4622352

Beta

機械学習 (ML) デフォーマ

MLデフォーマ

事前に機械学習したデータを用いて
高品質のメッシュ変形を**模倣**する機能

UE5.1で β 版をリリース。**5.2で改良**

MLデフォーマ サンプルを公開

https://www.unrealengine.com/marketplace/ja/product/ml-deformer-sample

学習リソース

公式ドキュメント「機械学習デフォーマの使い方」
https://docs.unrealengine.com/5.2/ja/how-to-use-the-machine-learning-deformer-in-unreal-engine/

Nearest Neighbor Model (NNM)モデルの解説
https://dev.epicgames.com/community/learning/tutorials/2lJy/unreal-engine-nearest-neighbor-model

MetaHuman Framework & Machine Learning for Next-Gen Character Deformation | GDC 2023
https://www.youtube.com/watch?v=OmMi6E0EkQw

UE5.3 で
様々な最適化・
改善を追加する予定です

Unreal Engine Public Roadmap より
https://portal.productboard.com/epicgames/1-unreal-engine-public-roadmap/c/1162-ml-deformer

ML Deformer

TB TIM BRAKENSIEK | Posted on June 9

ML Deformer enables users to approximate a complex rig, nonlinear deformers, or any arbitrary deformation by training a Machine Learning model that runs in real time in UE.

Improvements in UE 5.3 include:

- The addition of a masking system for Local Neural Morph Model, which improves the ability of users to work with structured data
- Improvements to the Nearest Neighbor model UI and workflow
- The removal of dependency on NNI framework
- The ability to create training masks, and visualize the masks per bone / bone group / curve in the viewport
- The ability to evaluate optimized trained neural morph models on all platforms that support morph targets (PC, Xbox, PS5, Android, iOS, Switch, etc.)
- Training masks operate as expected: a limited mask for a bone limits deformation to the masked region for the given bone
- The ability to create and train new models using the Nearest Neighbor and Neural Morph model on Windows
- The ability to open the ML Deformer Sample Showcase and retrain all shipped models without error on Windows
- There is no longer any need to take any manual steps (for instance, installing a Python package) for the Nearest Neighbor and Neural Morph models to train correctly on Windows
- The ability to run trained Neural Morph and Nearest Neighbor models on Linux and macOS (although training errors may be encountered)

MetaHuman Animator を公開

iPhone 1台で忠実度の高いパフォーマンス キャプチャを実現
https://www.unrealengine.com/ja/blog/delivering-high-quality-facial-animation-in-minutes-metahuman-animator-is-now-available

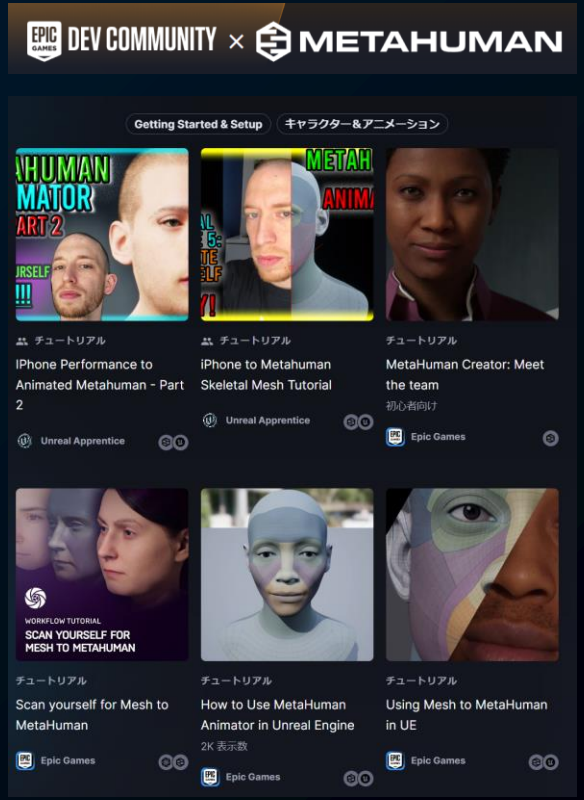
https://www.youtube.com/watch?v=WghYEFIoaM8

学習リソース

How to Use MetaHuman Animator in Unreal Engine
https://www.youtube.com/watch?v=WWLF-a68-CE

Hellblade actor demos MetaHuman Animator | State of Unreal | GDC 2023
https://www.youtube.com/watch?v=uAYI2rOtgM4

MetaHuman ハブ
https://dev.epicgames.com/community/metahuman



Modeling / Editor

UE5.2 アップデート

Beta

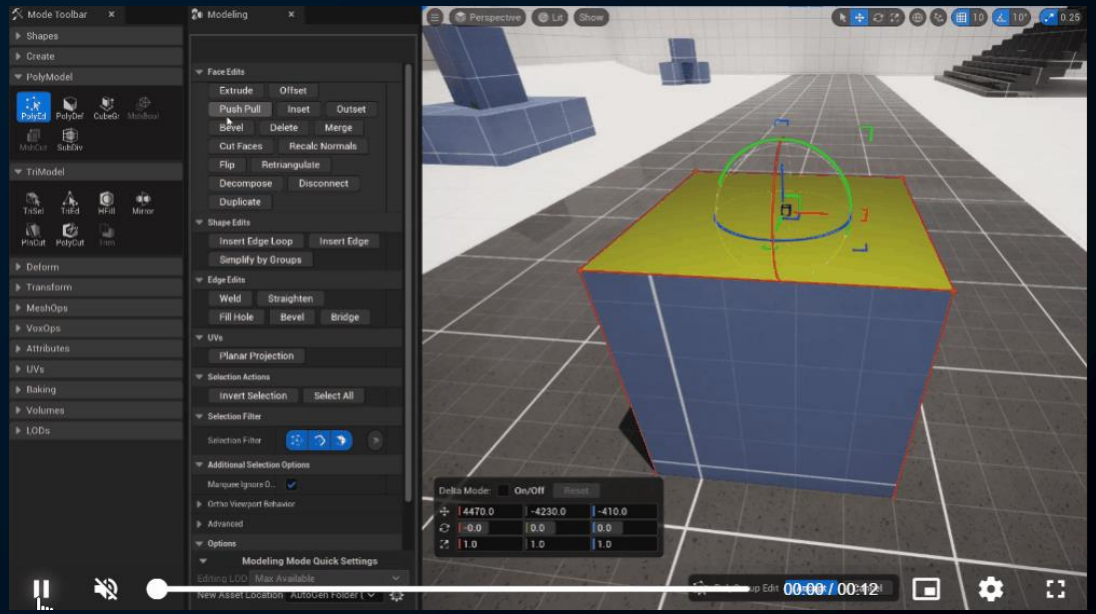
モデリングモード Update

モデリングモードとは

エディタ上で
モデル作成・編集・最適化が
できる専用のモード・機能

- メッシュ生成・編集・変形
- 自動UV展開、UV編集
- ピボット（基準位置）編集
- ベイク処理
- リダクション
- テクスチャへのベイク
- LOD生成・調整

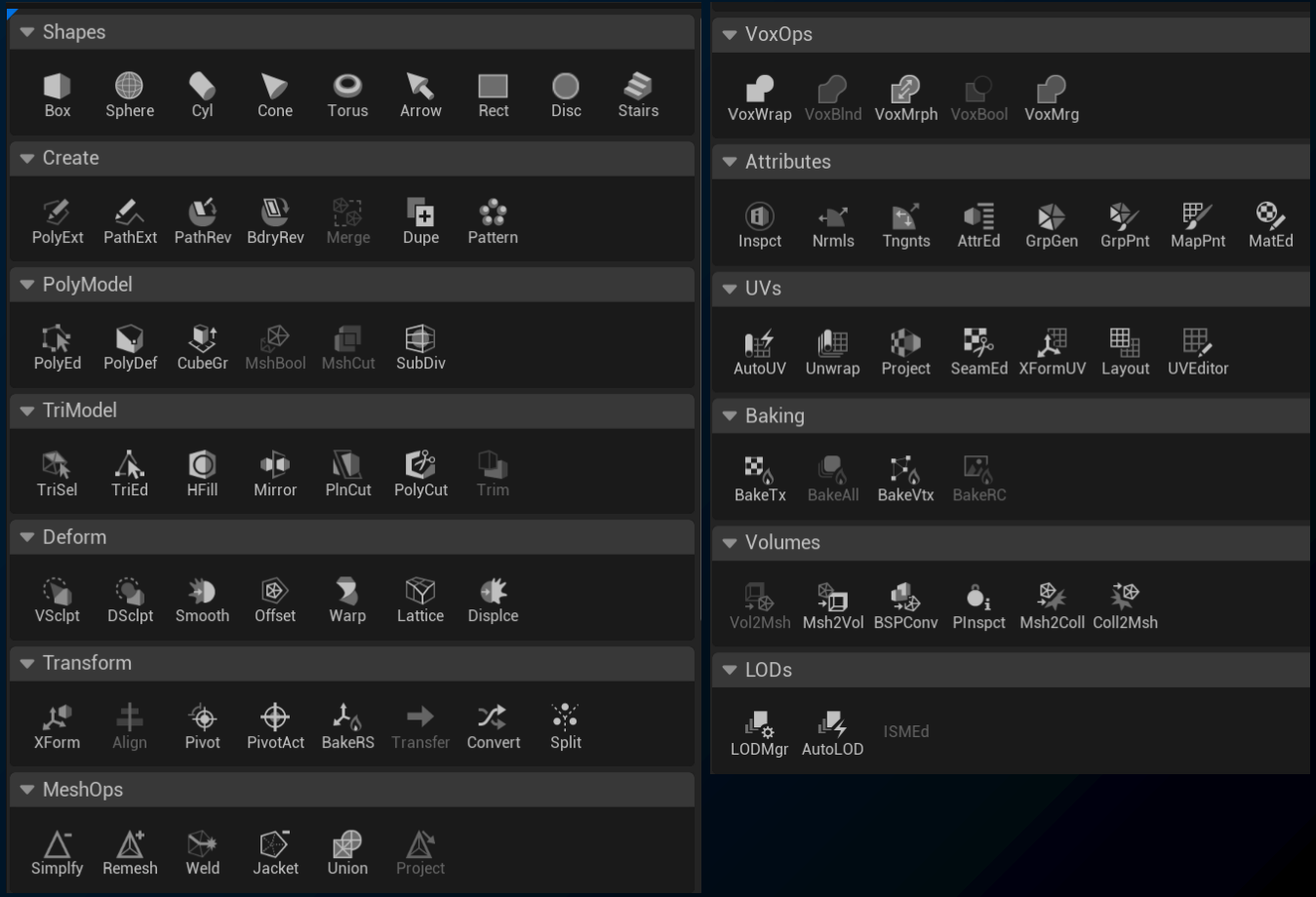
など



80種類の
ツールを提供！

https://docs.unrealengine.com/5.2/ja/modeling-tools-in-unreal-engine/

ツールパレットが
多くなりすぎたので
UE5.3で
UIが一新予定



UNREAL ENGINE

モデリングモードが目指すもの

レベルデザイン が
そのレベル内 で 完結すること

BSP から モデリングモード へ

近年のレベルデザイン・ジオメトリの複雑さに対応するには
より高性能・多機能なツールが必要に

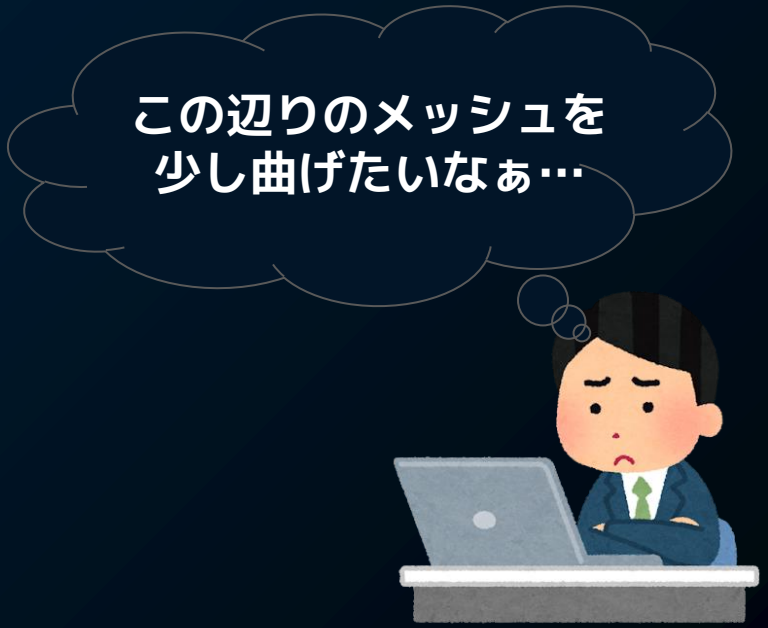
Unreal Tournament 4(2014)

Lyra Starter Game (2022)

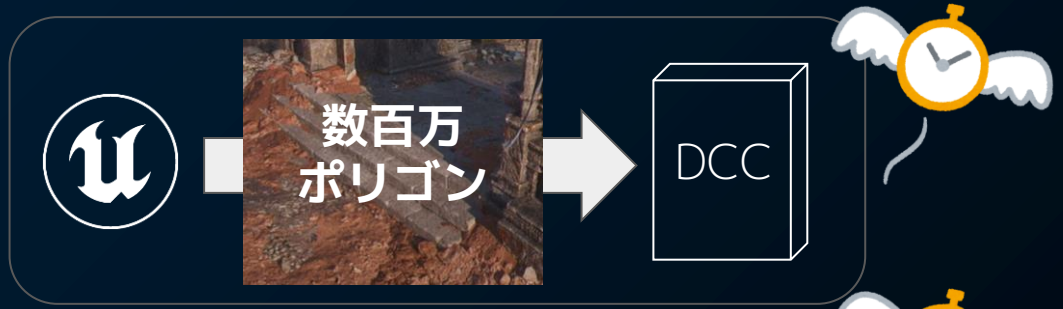

単純な斜面ではなく
階段を生成・配置
丸みを帯びた角に
壁をくり抜いて
入り口・窓を作成

ハイポリメッシュを扱う難しさ

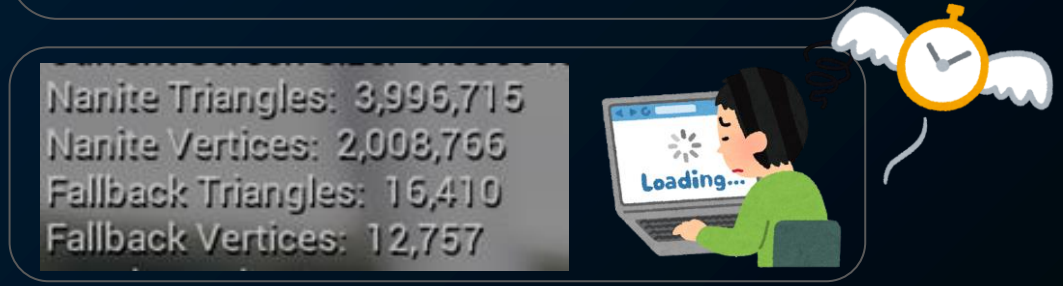
Megascansで
キットバッシングしたレベル



ハイポリメッシュの
エクスポート(UE)
インポート(DCC)



ハイポリメッシュと
相性が悪い
環境・ツールでの作業



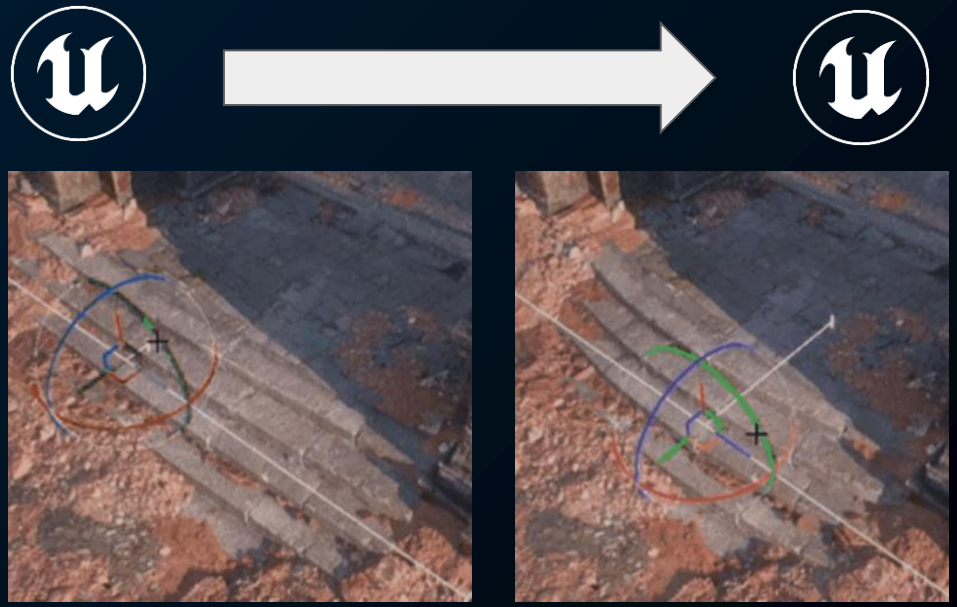
ハイポリメッシュの
エクスポート(DCC)
インポート(UE)

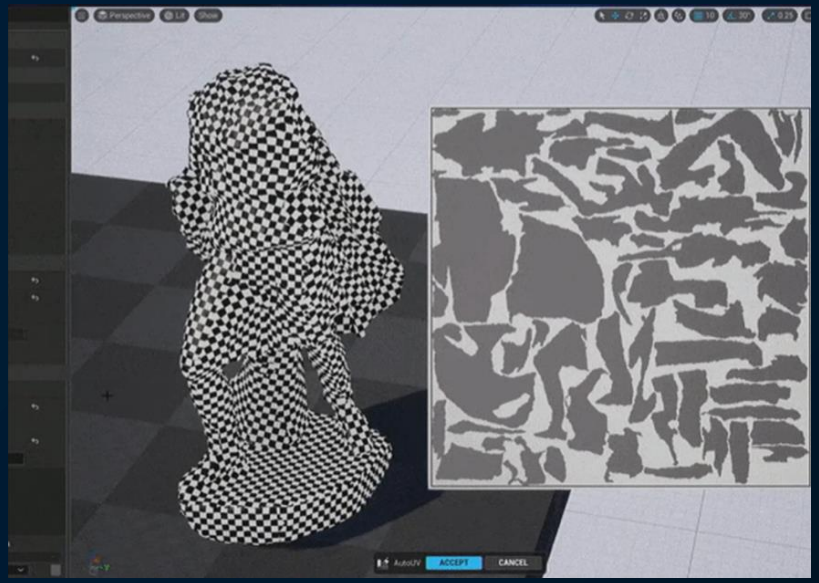
DCCツールを介さないことで、作業を高速化

UE上で作業を完結させるメリット

- ツール間を行き来する際の
  待ち時間を排除
- ハイポリメッシュを
  リアルタイムで編集可能
- 編集後の結果を
  最終クオリティですぐ確認可能

ハイポリメッシュでも
高速なイテレーションを実現！





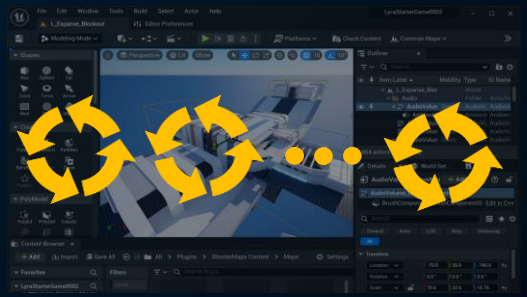
ハイポリメッシュを自動UV展開

スキャンデータをクリーンアップ
ハイポリメッシュをリダクション

モデリングモードが 実現する こと

- レベルデザイン用のモデルを生成・編集
- インポートしたアセットのカスタマイズ

- DDCツールを介さない高速な反復作業
- ハイポリメッシュの最適化



モデリングモードが 実現しない こと

従来のDCCツールを使った
モデリング・ワークフローの置き換えは「しない」

プロジェクト固有のヒーローキャラクターや
ランドマークとなるモデルを作るツールではない

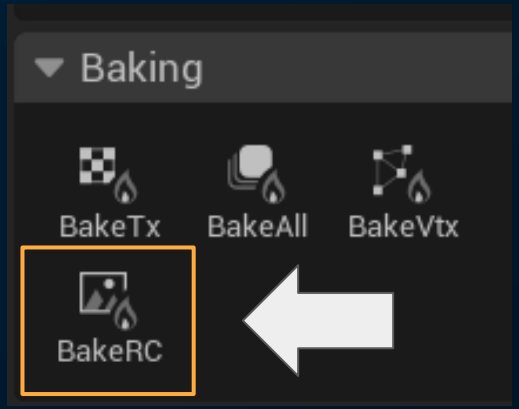
Beta

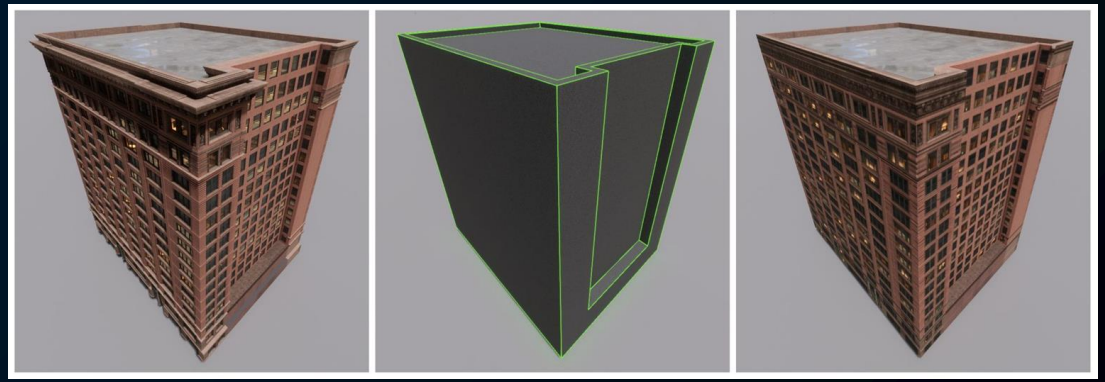
モデリングモード Update

BakeRC（レンダリング キャプチャ ベイク ツール）

UE5.2リリースノートより

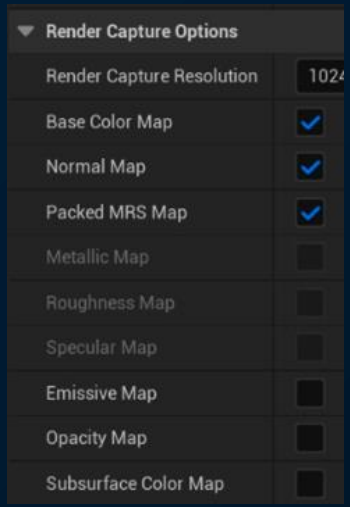
BakeRC では、バーチャル フォトやレンダリング キャプチャを介して、
複数のソース メッシュからターゲット メッシュのテクスチャをベイクします





BakeRC の 仕組み

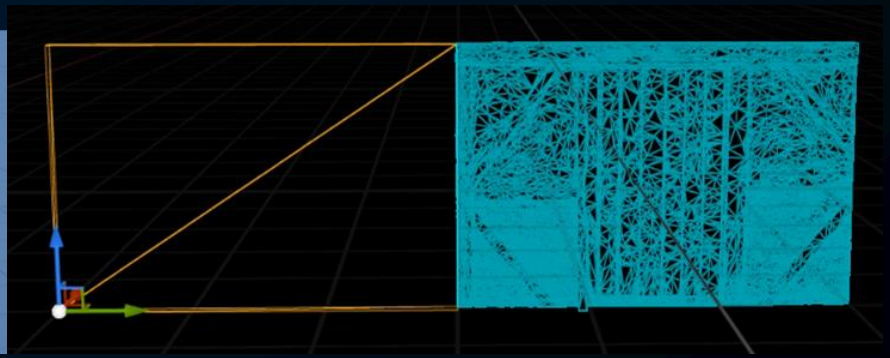
複数視点からの描画結果をテクスチャにベイク
BaseColorだけでなく、NormalやMetallicなどもベイク可能



BakeRC の メリット・使い方

ベイクしたテクスチャを使うことで
ローポリメッシュでハイポリメッシュの見た目を（ある程度）再現できる

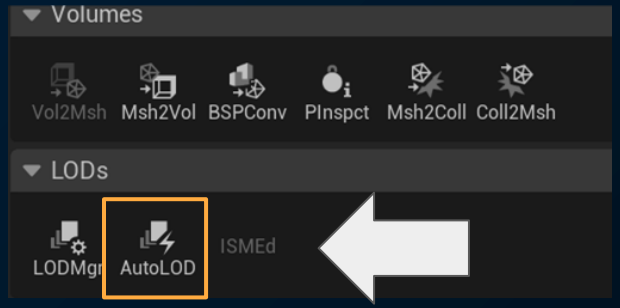
使い方：
https://twitter.com/pafuhana1213/status/1671717634503737344

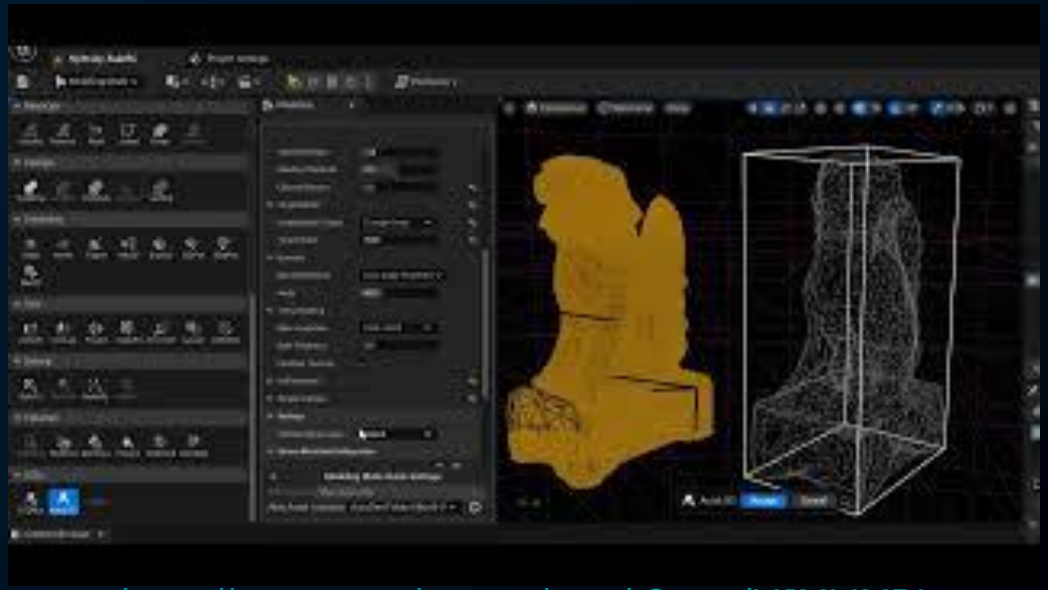


Auto LOD

ハイポリメッシュから
ローポリ版を作成する機能

- 豊富なリダクション設定
  - テクスチャも対象
- 自動UV展開
- 法線のテクスチャベイク
- リアルタイムで設定反映

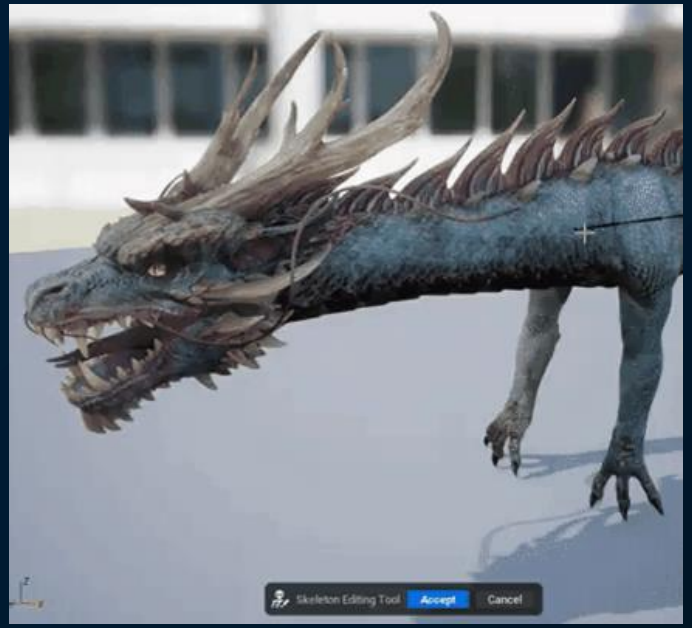




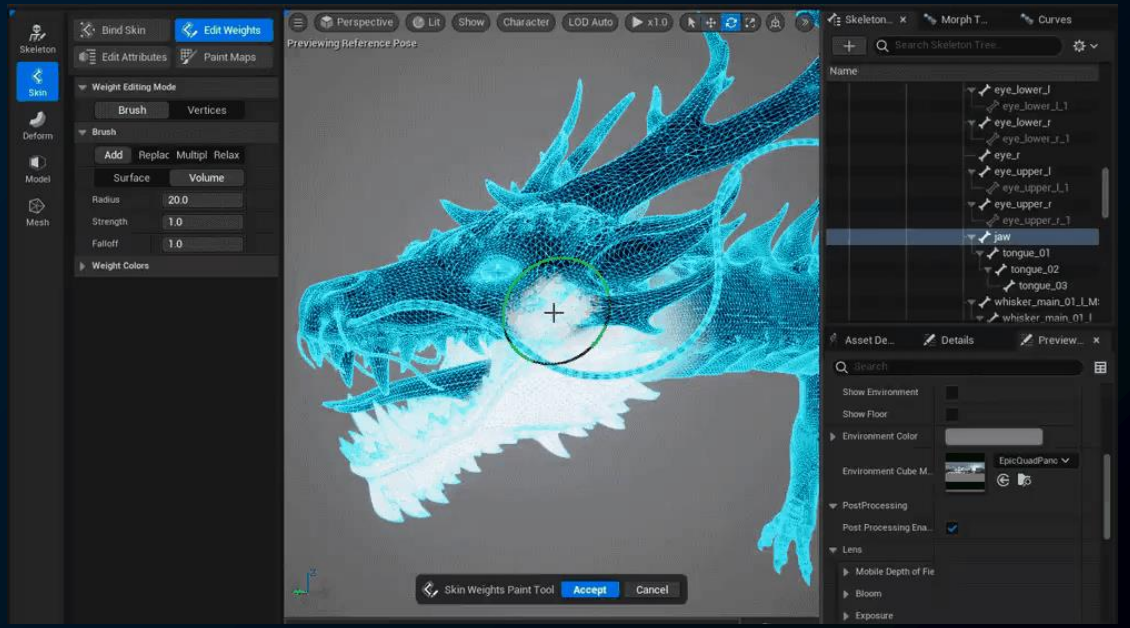
https://www.youtube.com/watch?v=_xiVJMLIMBI

https://twitter.com/pafuhana1213/status/1670840734108794888

UE5.3 では Skeletal Editor を追加



スケルトンの作成・編集



スキンウェイトの作成・編集

UNREAL ENGINE

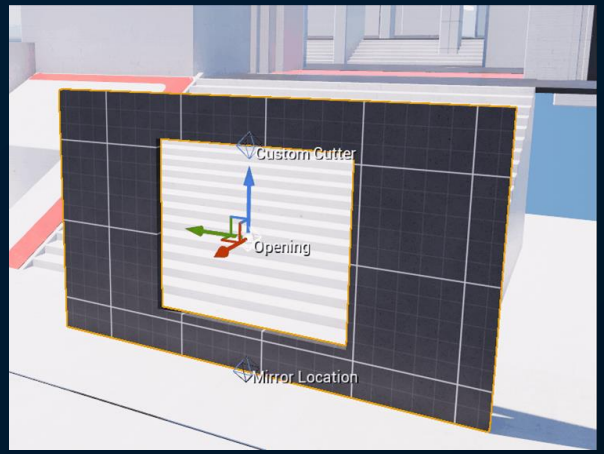
Experimental

Geometry Script Update

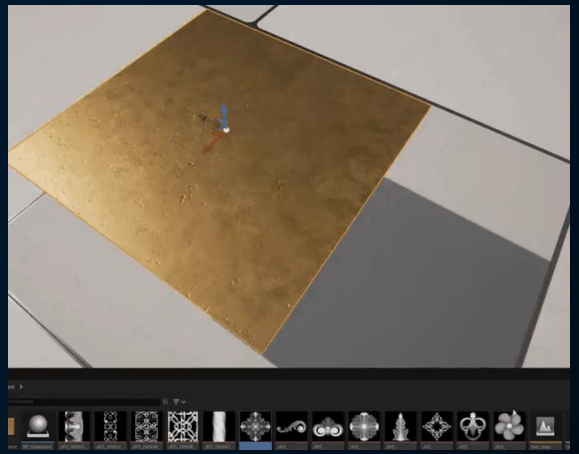
Geometry Scriptとは

Blueprint / Python で メッシュを動的に生成・編集できる機能

レベルデザインだけでなく、プロシージャルモデリング用途でも利用可能



Lyra のジオメトリ ツール



Alpha / Height mapから
メッシュ作成

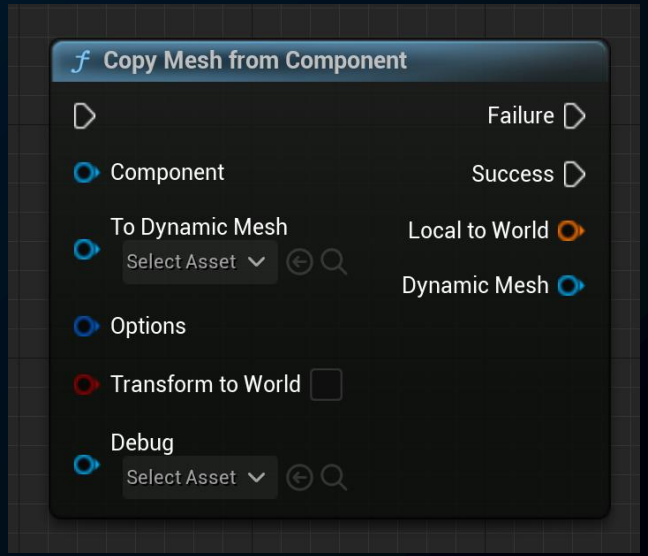
草木 ジェネレータ

Copy Mesh From Component の 改善

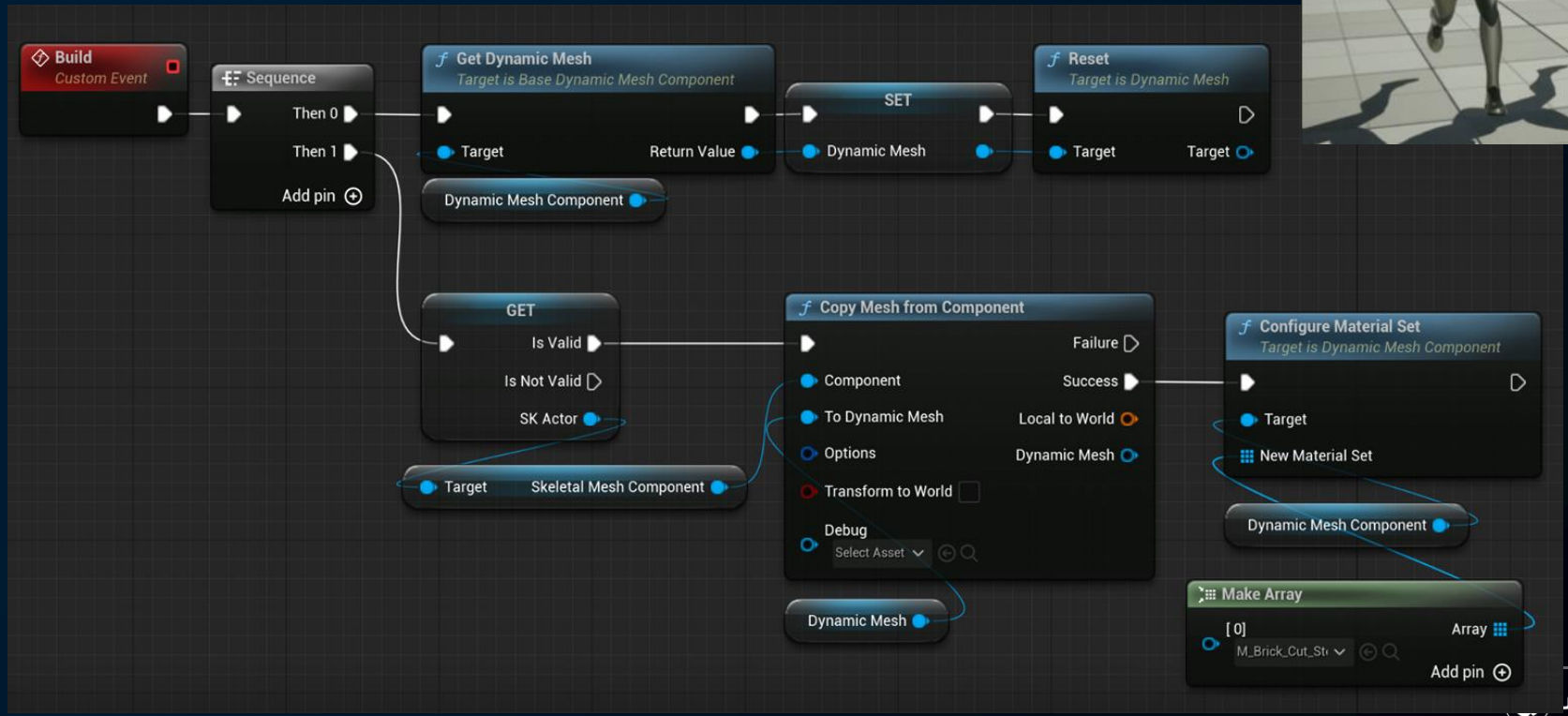
指定のComponent から メッシュ情報を取得し
Geometry Script が扱う Dynamic Mesh にコピーするノード

対応しているComponent：

- Static Mesh
- Dynamic Mesh
- Brush
- **Spline Mesh** **New !**
- **Posed Skinned Mesh** **New !**
- **Skeletal Mesh** **New !**

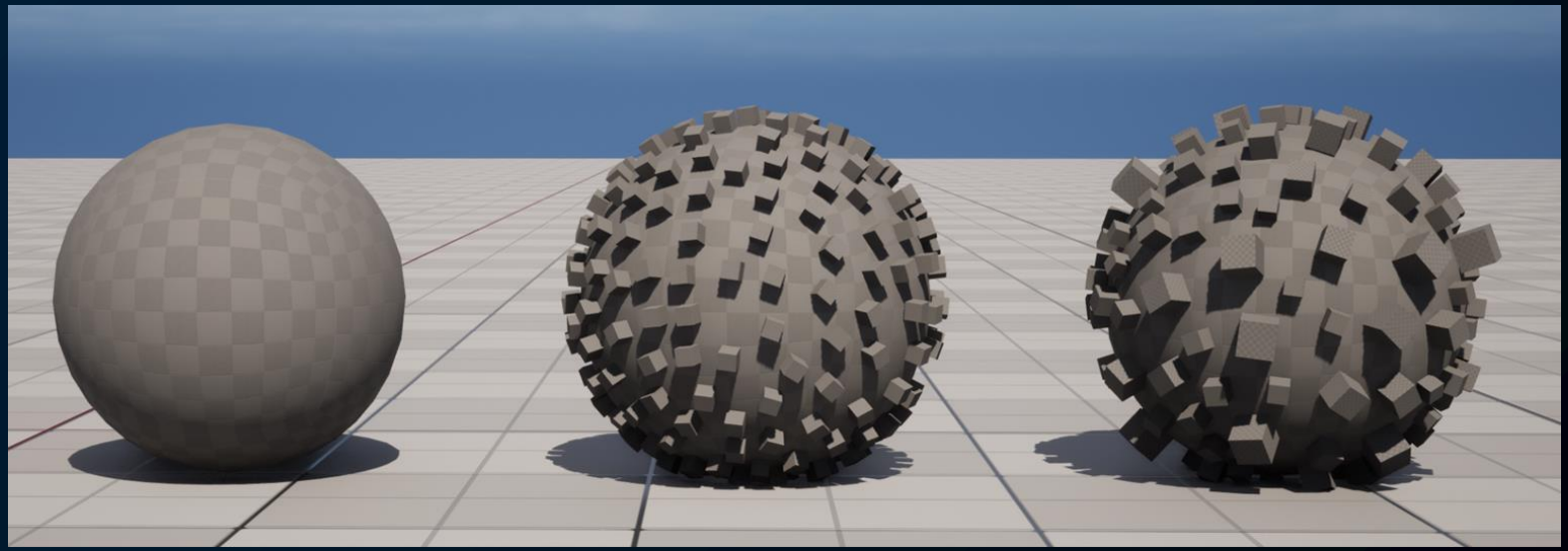


Skeletal Mesh Component から ポーズ付きメッシュ情報をコピー

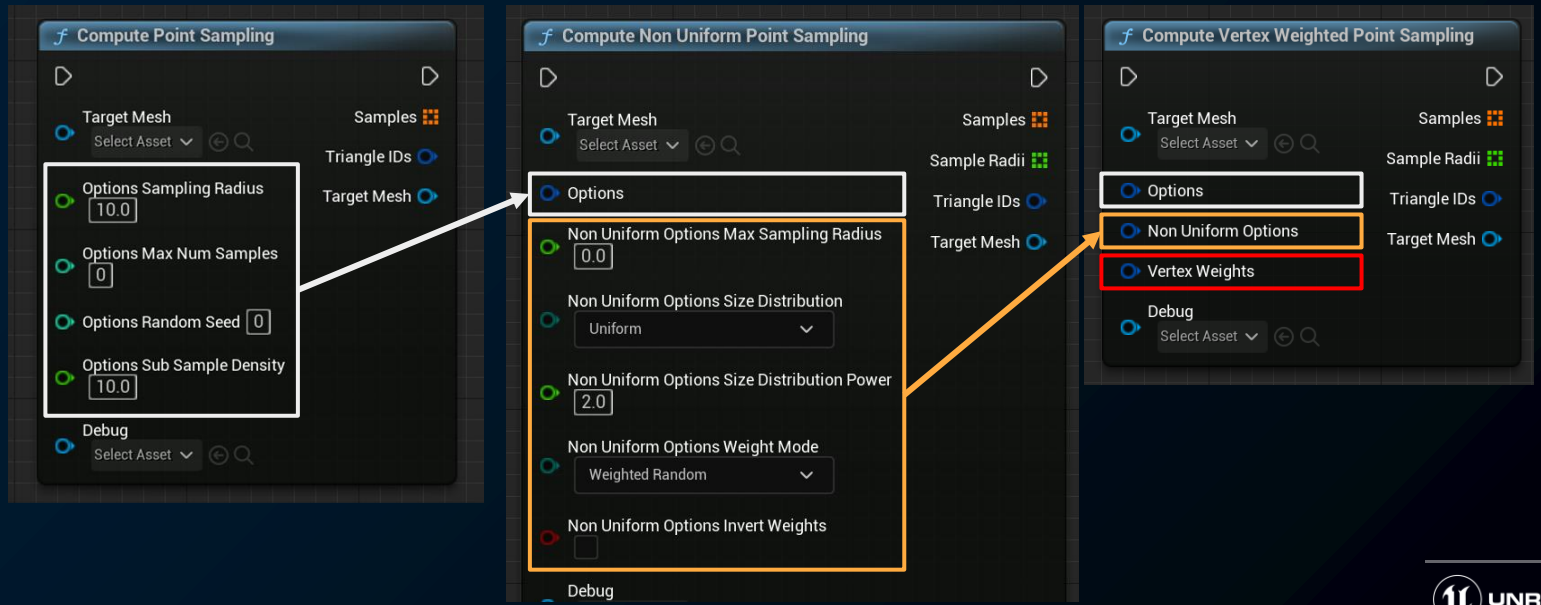


メッシュ表面のサンプリング

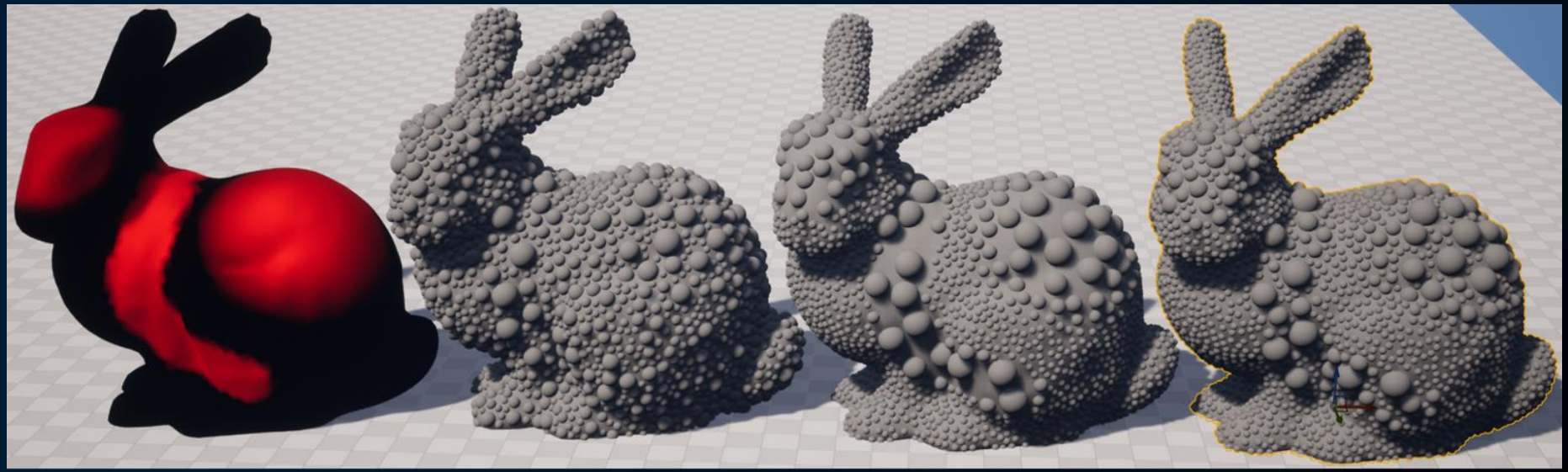
任意の 3D メッシュ上で
「均等 or 不均等」 かつ **「交差せず」** にサンプル点を生成する機能



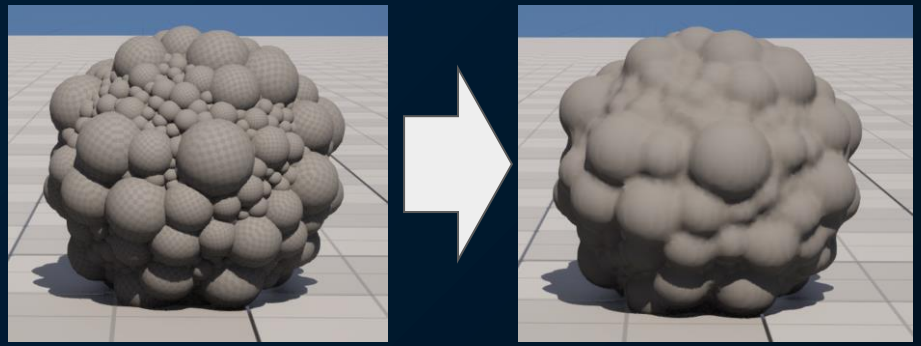
Compute Point Sampling：
　均等に分布
Compute **Non Uniform** Point Sampling：
　不均等に分布
Compute **Vertex Weighted** Point Sampling：
　不均等 かつ **頂点毎に重み付け** して分布



頂点カラーの R値 を
頂点ごとの重み付けに利用した例



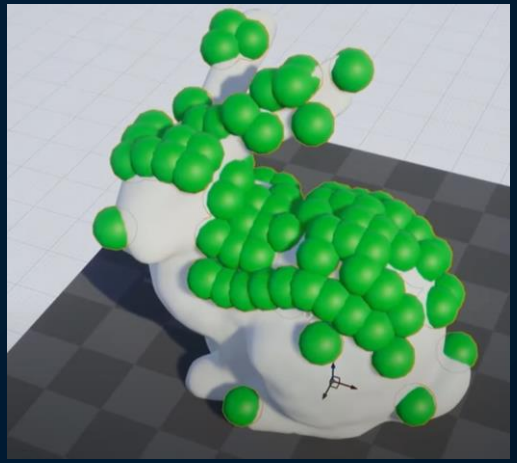
サンプリング結果を元に配置したメッシュに対して SDFベースの結合処理 や スムージングを行うと…

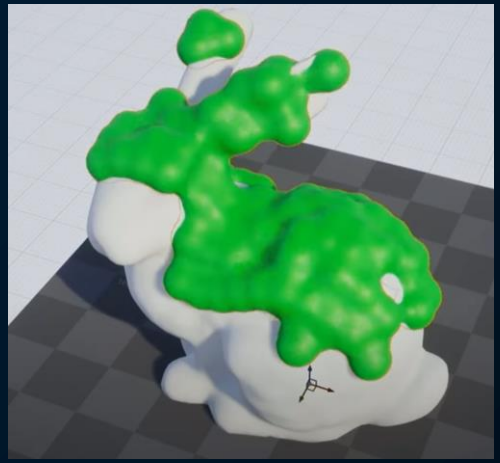


SDFベースの
結合処理

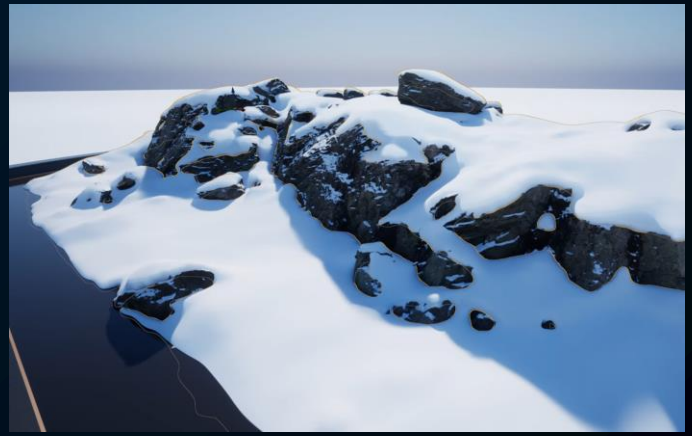
スムージング処理
を複数回実行

活用例





Modeling and Geometry Scripting in UE:
Past, Present, and Future

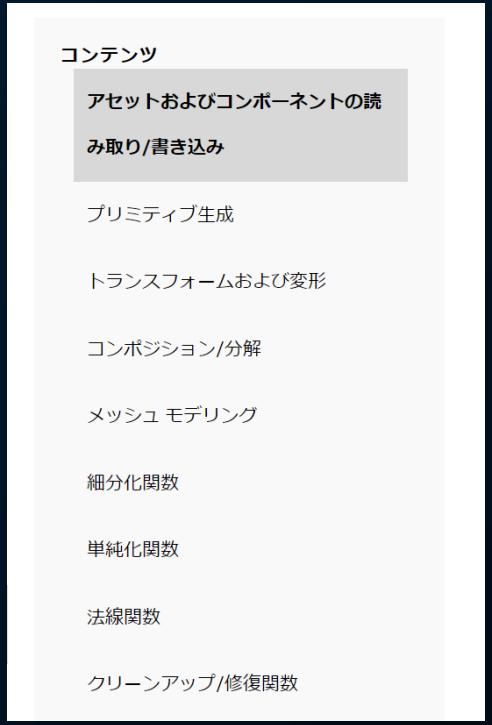


Geometry Script を使って
岩の上に雪を配置するデモ・解説

ジオメトリ スクリプトのリファレンス

Geometry Script は
現時点で **250 近くのノードを提供**

基本的なものは
公式ドキュメントに解説あり
https://docs.unrealengine.com/5.2/ja/geometry-script-reference-in-unreal-engine/



学習リソース

公式ドキュメント「モデリングとジオメトリ スクリプト」
https://docs.unrealengine.com/5.2/ja/modeling-and-geometry-scripting-in-unreal-engine/
Exploring Geometry Tools in UE5
https://www.youtube.com/watch?v=apCSgAAkDTU
Introduction to Geometry Scripting
https://www.youtube.com/watch?v=gDso9X4HgLA
Modeling and Geometry Scripting in UE: Past, Present, and Future
https://www.youtube.com/watch?v=pEC3krOwcbM
Blockout and Asset Production in UE5
https://www.youtube.com/watch?v=R5TsbnW4fk8
開発者の Ryan さんの Youtubeチャンネル
https://www.youtube.com/@RyanSchmidtEpic

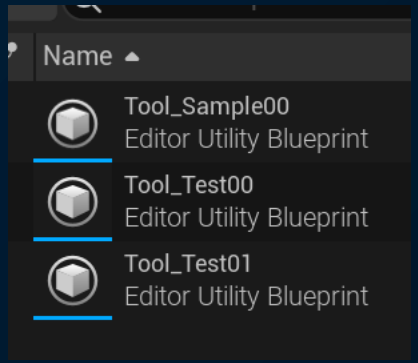
UE5攻略リンク
https://ue5study.com/unrealengine-modeling/
https://ue5study.com/unrealengine-geometry-script/

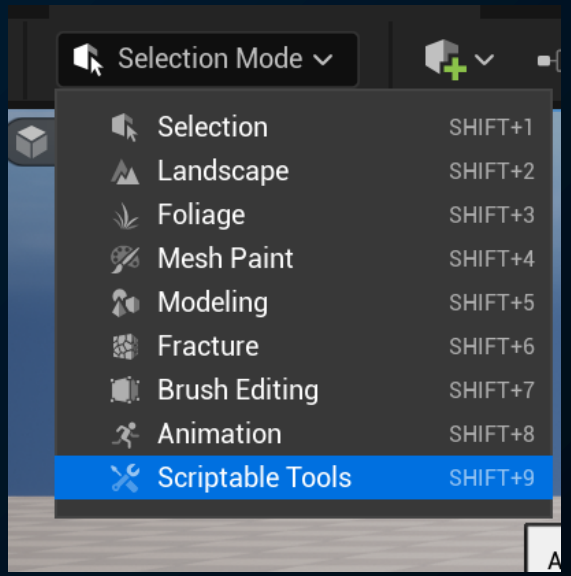
Experimental

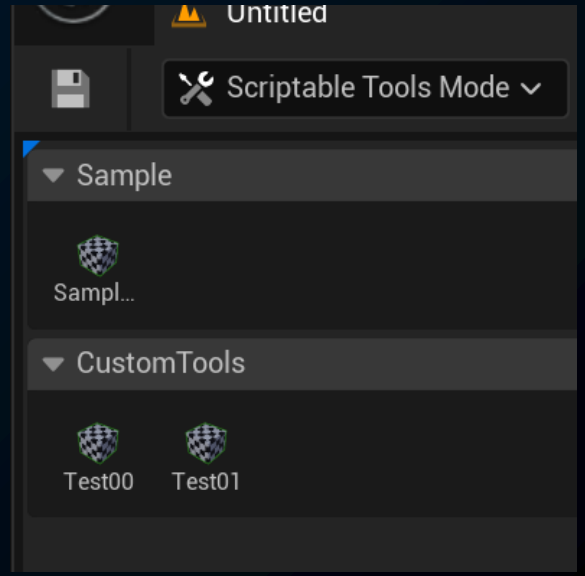
スクリプト可能ツールの
フレームワーク
（Scriptable Tools Framework）

Scriptable Tools Framework

BP / C++ を使って
プロジェクト独自 の エディタモード・ツール を作成可能に！

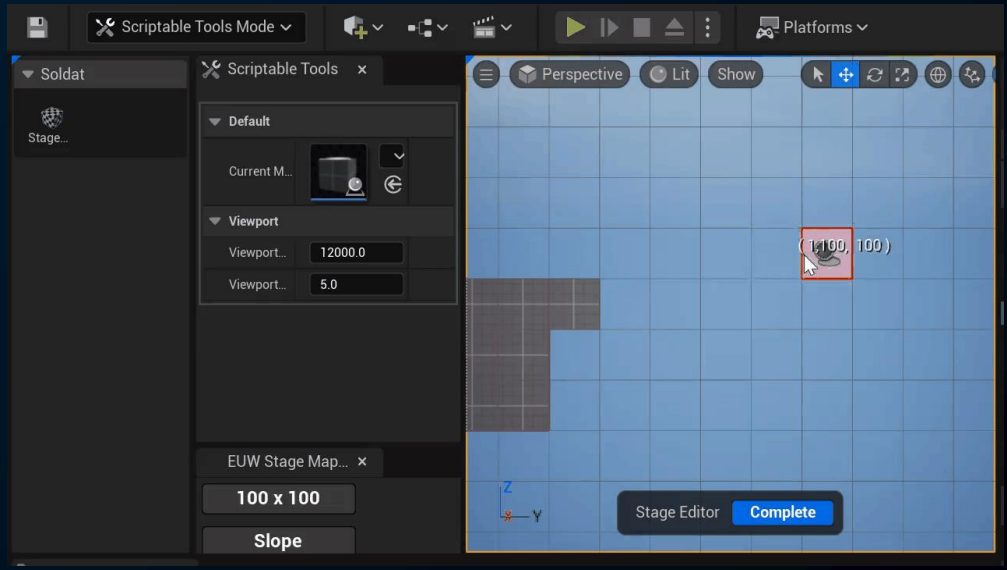






活用例

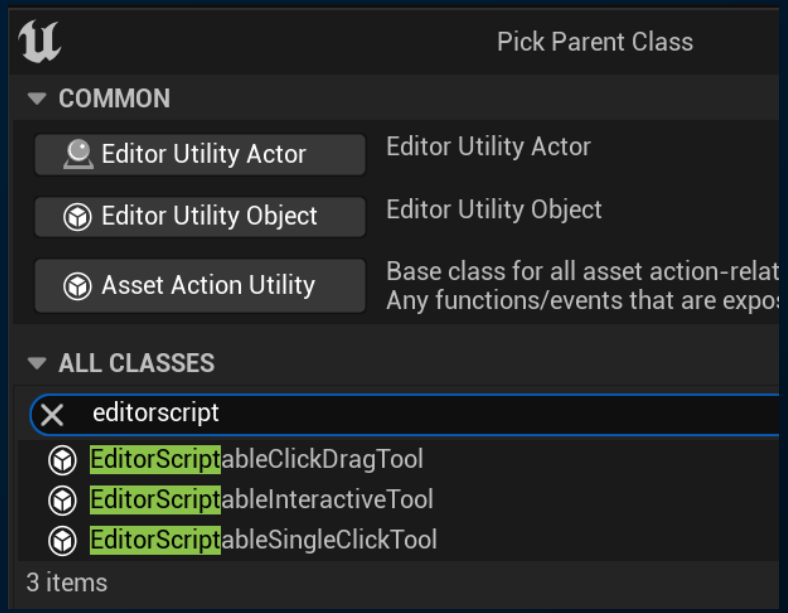
エディタ上で引いた線に沿って
草・枝を配置（Twitterより）

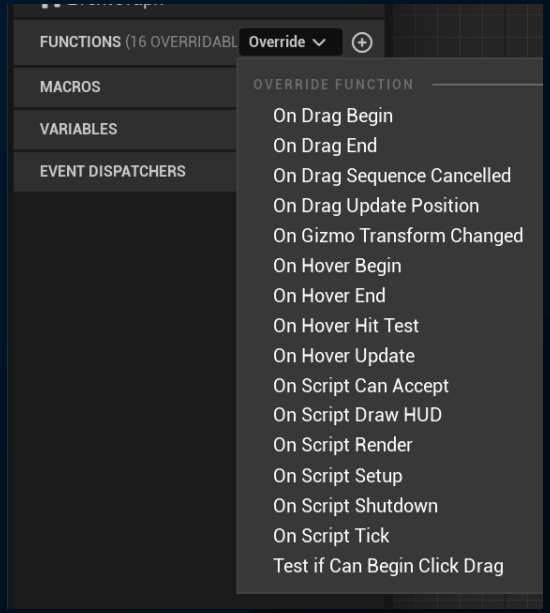


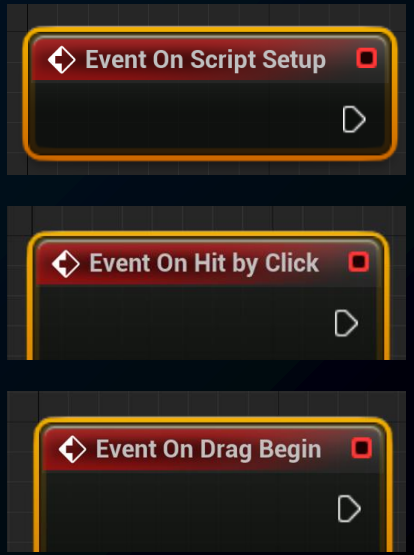
横スクロールゲーム用
ステージエディタ（Twitterより）

ツール作成用の EditorUtility Blueprint

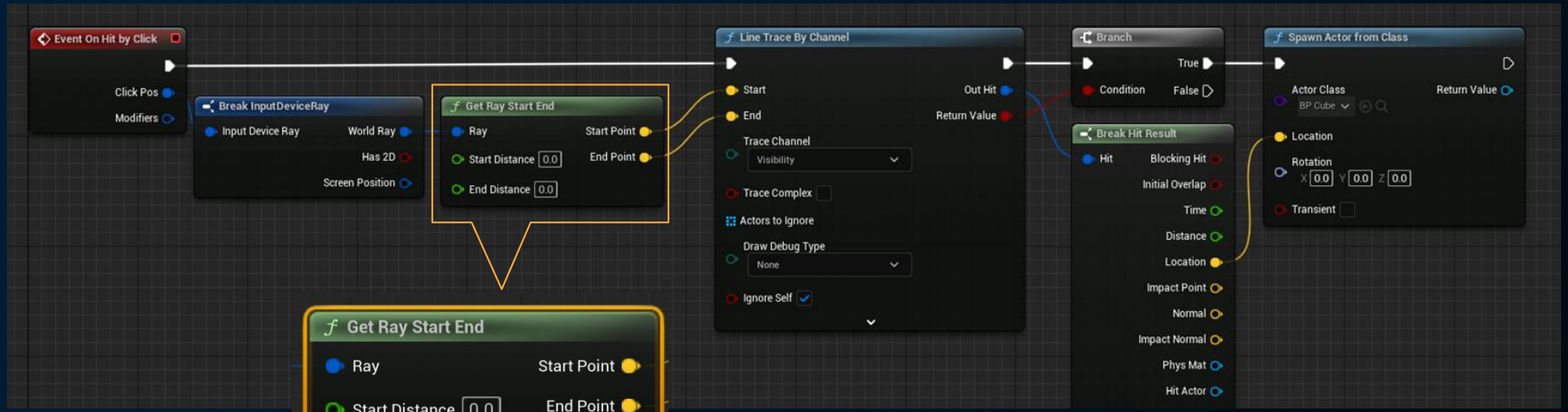
マウスの入力イベント、テキスト表示、3Dギズモの生成・制御など
基本的な機能を標準で提供



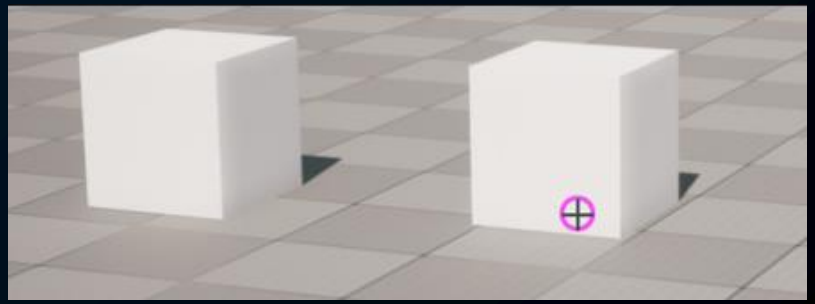




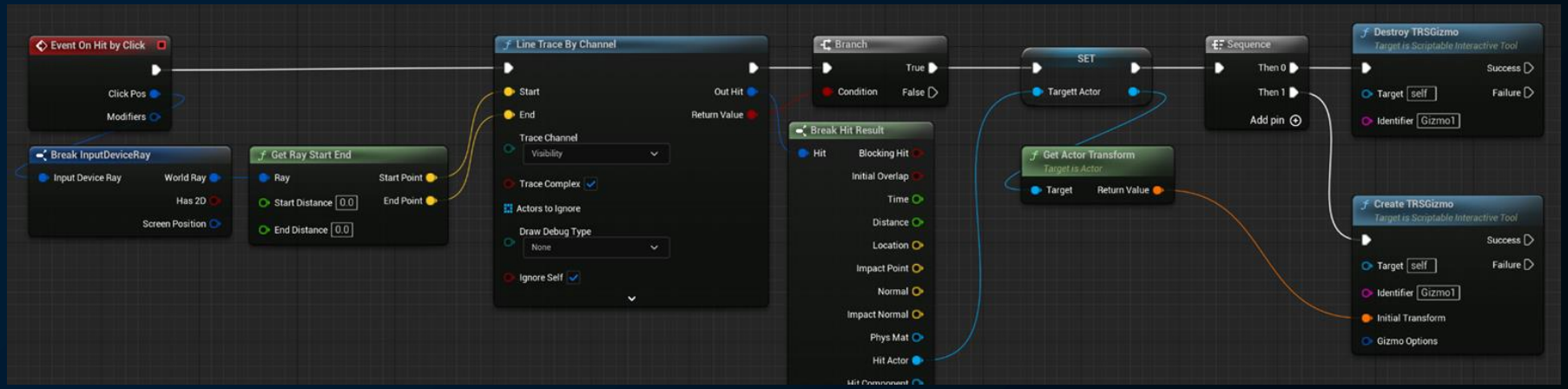
マウスクリックした位置にActorを生成

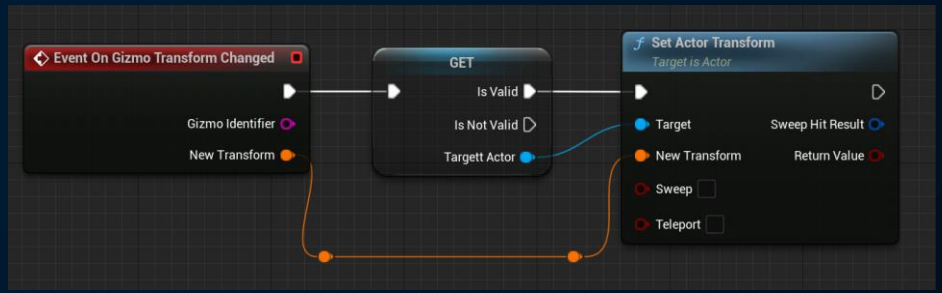


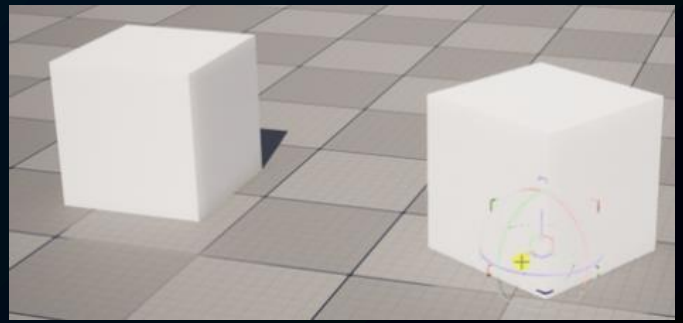
GeometryScriptプラグインを有効にする必要あり



3Dギズモによる Actor制御

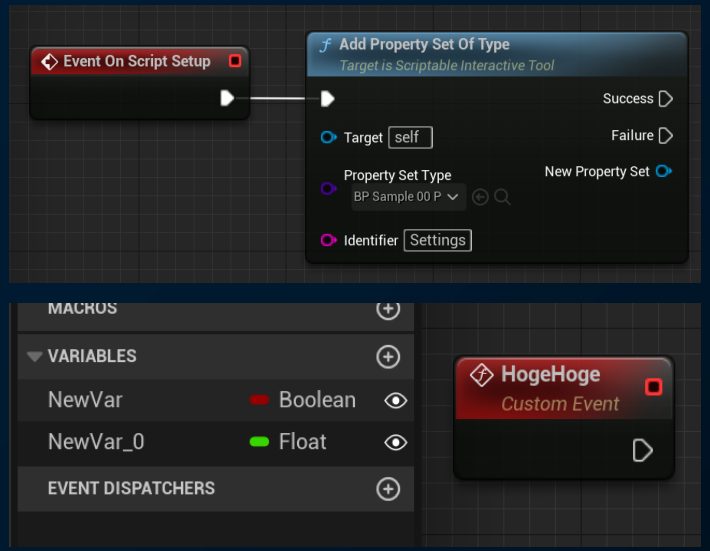


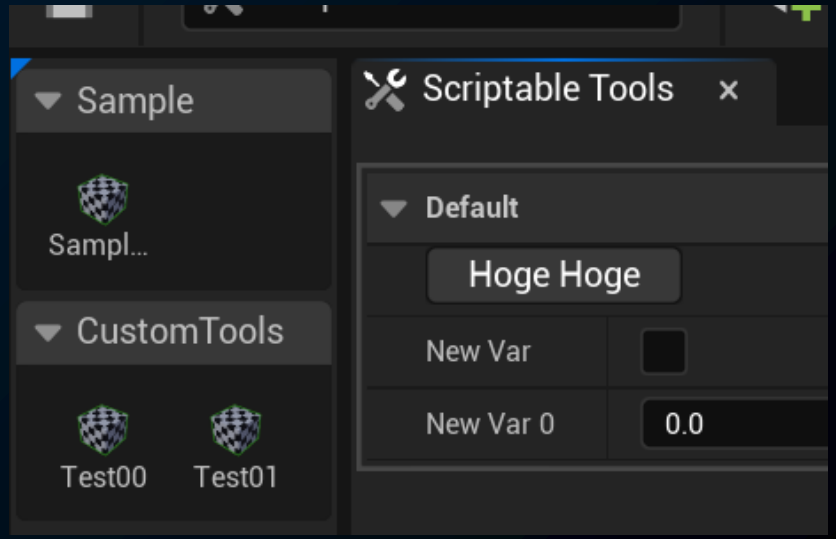




ツールのプロパティ、イベントを公開

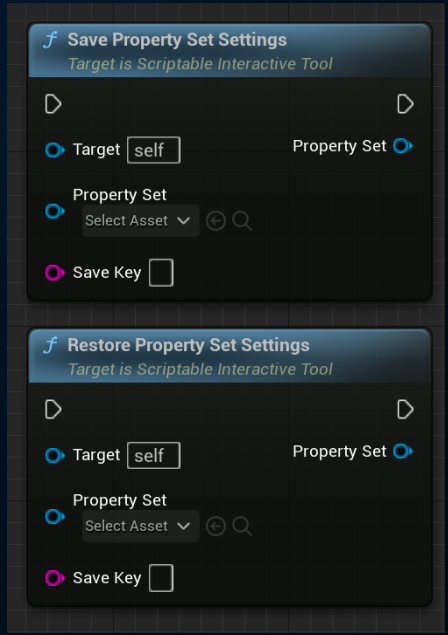
ScriptableInteractiveToolPropertySet 継承の
BPを用意・登録すればOK！





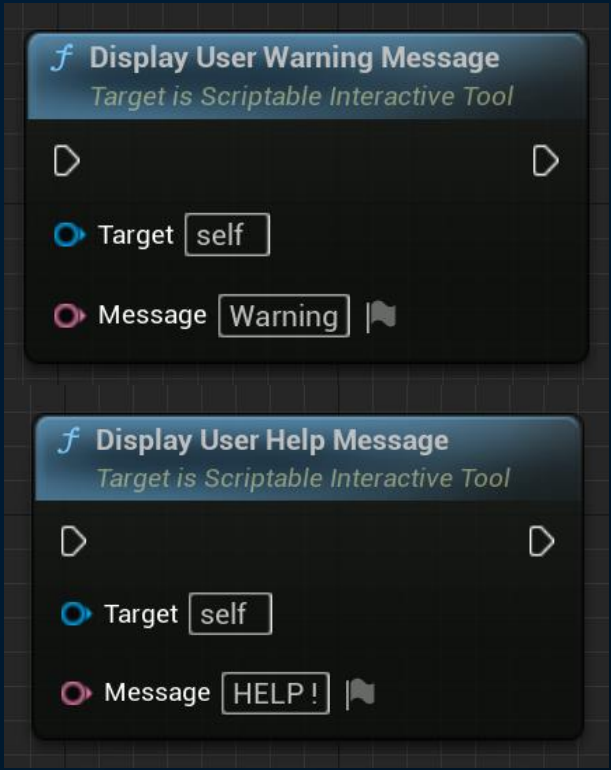
PropertySet を扱うためのヘルパー関数

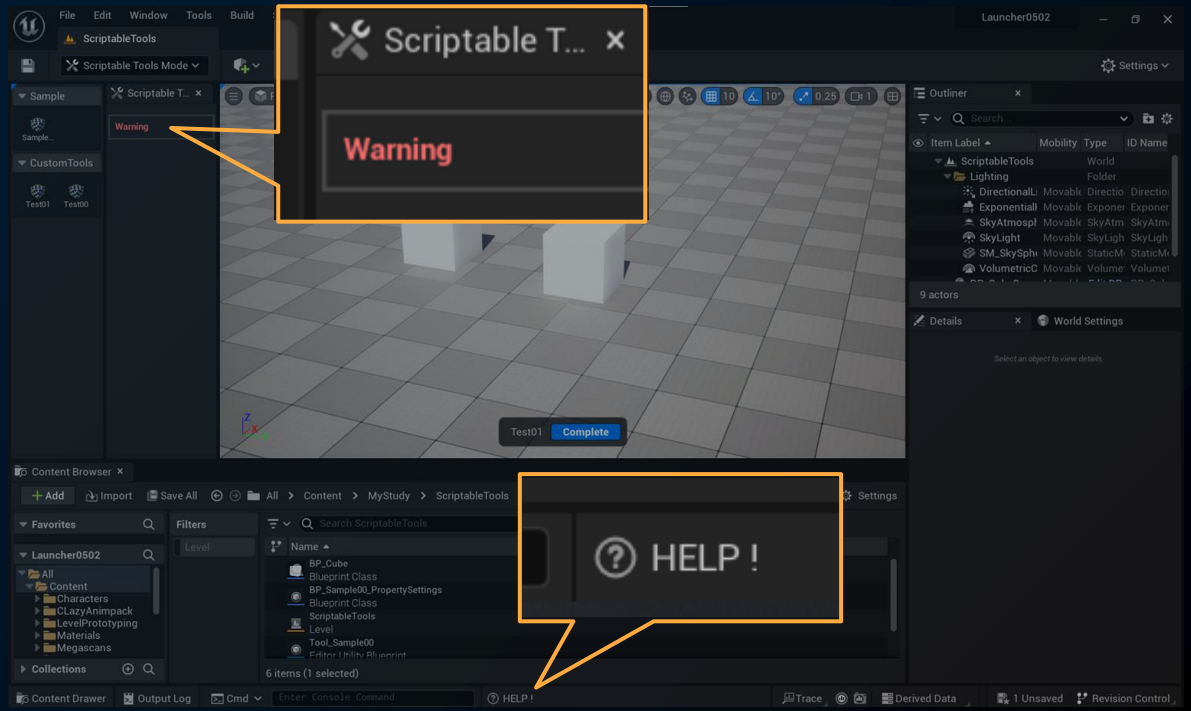
プロパティの変更イベントを検出



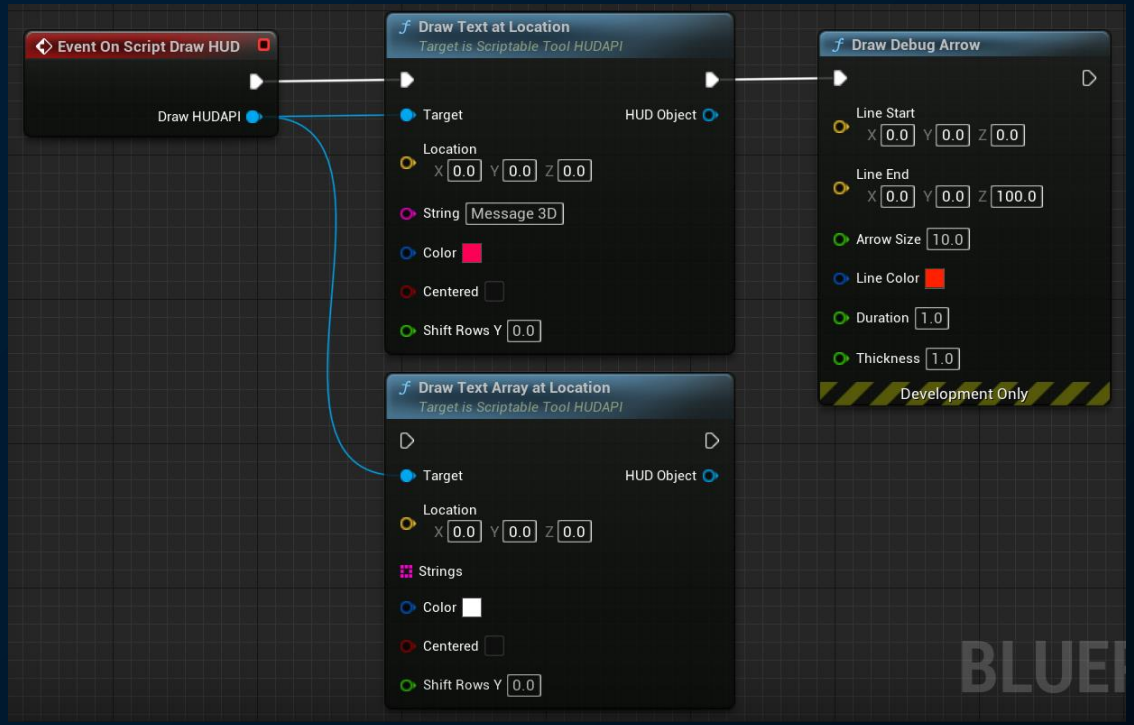
プロパティの保存・復元

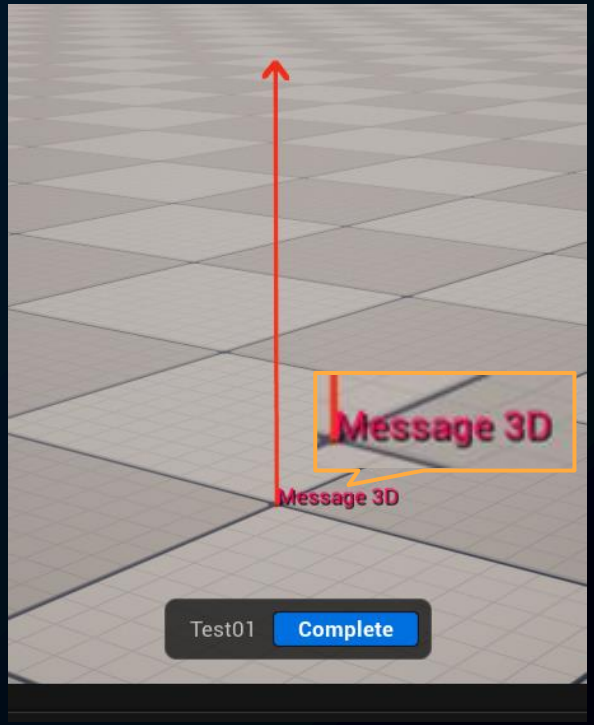
ツールでの テキストメッセージ





ビューポートへのテキスト・デバッグ表示





Scriptable Tools Framework はいいぞ！

プロジェクトに特化したレベルデザインツールを作れる！
　Geometry Script や
　Procedural Content Generation Framework などと組み合わせたり

ビューポートでの操作を限定できるという強みも！
　他ツール（選択など）が無効化される
　意図しない編集によるトラブル・ヒューマンエラーを回避できる

ご紹介した通り、**ツールで必要な機能をシンプルな形式で提供**
　その反面、カスタムUIをつくることが現時点でできない
　EditorUtilityWidgetとうまく使い分け・併用するなどの運用が必要

学習リソース

Scriptable Tools & Editor Mode
https://dev.epicgames.com/community/learning/tutorials/1loo/unreal-engine-scriptable-tools-editor-mode
Scriptable Tools & Editor Mode Reference
https://dev.epicgames.com/community/learning/tutorials/7BKd/unreal-engine-scriptable-tools-system-editor-mode-reference
Scriptable Tools導入の流れ（Twitter）
https://twitter.com/pafuhana1213/status/1654124259684519938

Editor Scriptable Tools: Basic Tools UE5.2
https://www.youtube.com/watch?v=xWxtxS2jADo
UE5.2 Scriptable Tools : Node overview
https://www.youtube.com/watch?v=w-PXmo8QKZs
Unreal Engine 5.2 - Scriptable Tools Introduction (Physics Scattering)
https://www.youtube.com/watch?v=0-DcPwR4CI0

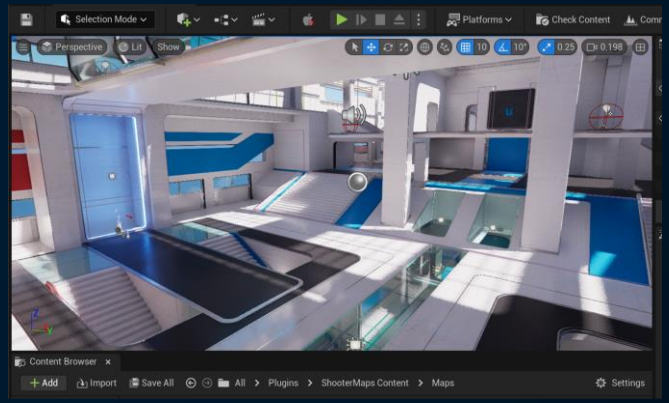
Mobile / Mac

UE5.2 アップデート

モバイル プレビューの改善点

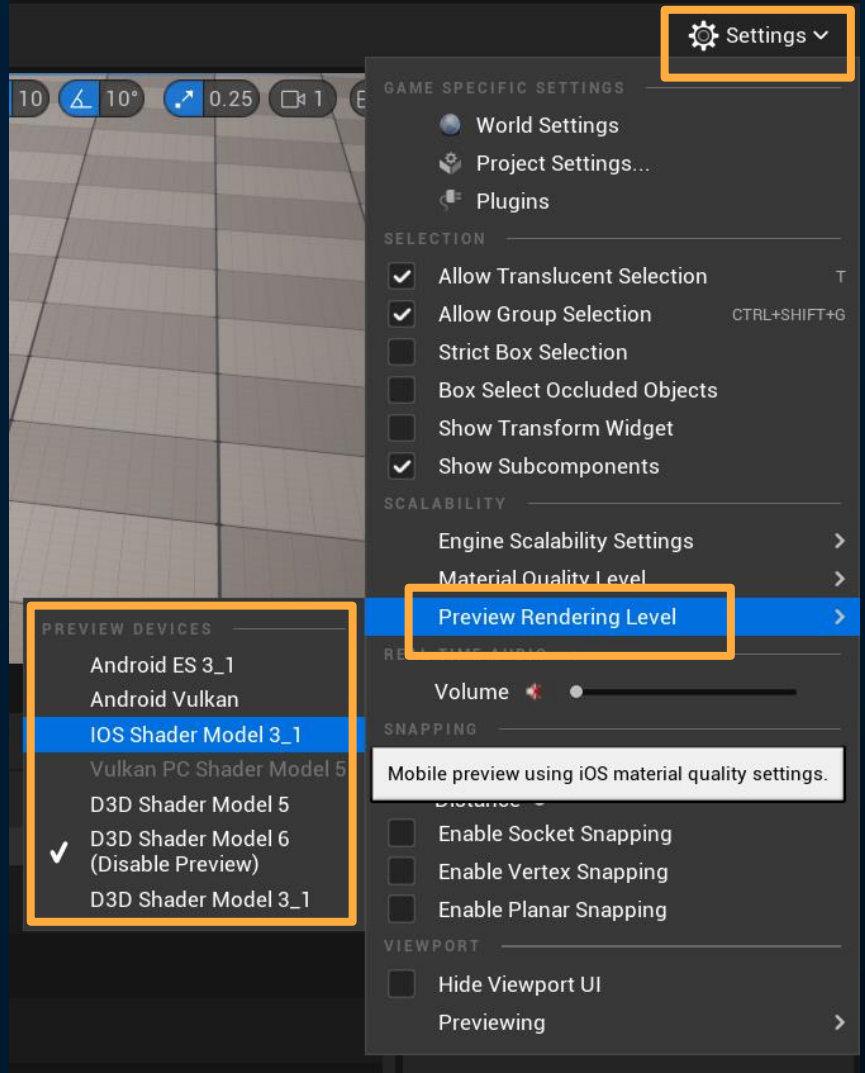
モバイルプレビュー

異なるモバイル デバイス（レンダラー）における描画結果の違いを
エディタ上で **エミュレート** する機能
https://docs.unrealengine.com/5.2/ja/using-the-mobile-previewer-in-unreal-engine/

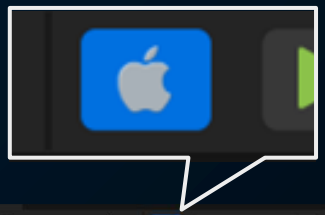


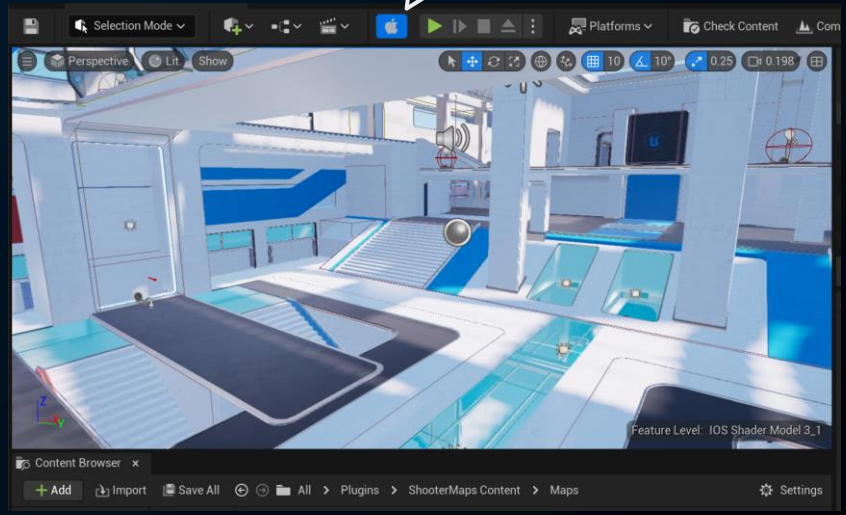
D3D Shader Model 6

IOS Shader Model 3.1



このボタンで
ON/OFFを切り替え可能





UNREAL ENGINE

UE5.2 リリースノートより

モバイル プレビューの改善点

UE 5.2 では、Unreal Editor のモバイル プレビュー モードに安定性と精度の改善をいくつか行いました。これは、[Settings (設定)] メニューから [Preview Rendering Level (プレビュー レンダリング レベル)] のデバイスを選択することで使用可能です。

- プレビュー デバイスを変更する際の安定性を改善しました。
- エディタで [Engine Scalability Settings (エンジンの拡張機能設定)] ウィンドウを使用すると、プレビュー プラットフォームのスケーラビリティ設定が適用され、プラットフォームでオーバーライドされたスケーラビリティ変数がエディタ ビューポートに適用されるようになりました。たとえば、プラットフォームのスケーラビリティ ini ファイルのオーバーライドに依存するメッシュ LOD トランジション距離のような設定が正確にプレビューされます。
- ターゲットのプレビュー プラットフォームのデバイス プロファイルによって設定されるコンソール変数がすべて適用されるようになるため、エディタ ビューポートでは、デバイス上で実行されると有効になる一連のレンダリング機能が正確にプレビューされます。

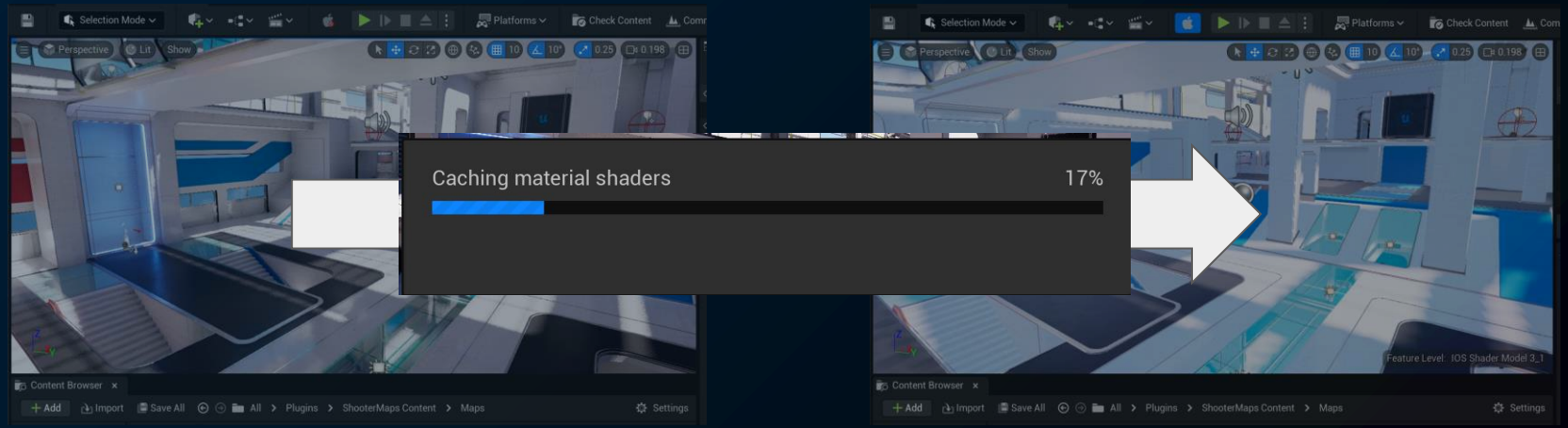
モバイルプレビューへの切り替え時間が爆速に

ボタンを押してから
モバイルプレビュー に切り替わるまで

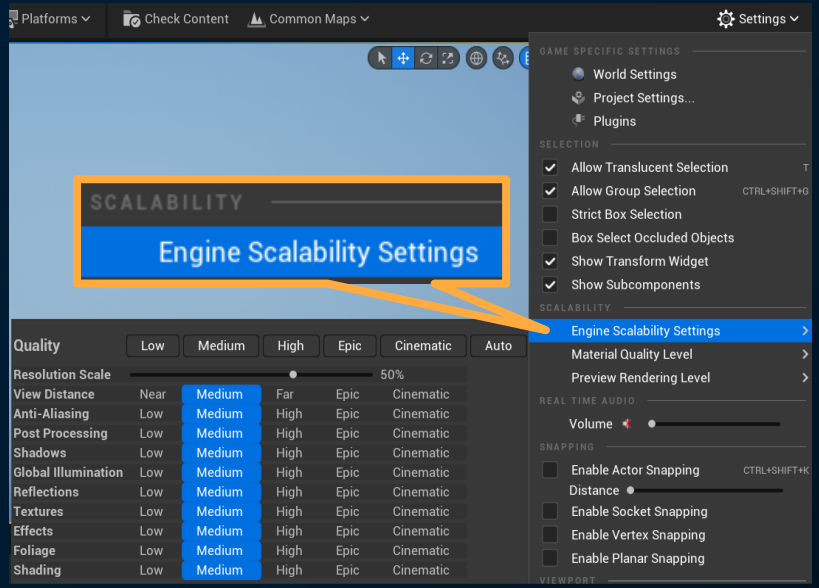
※ Lyra ＆ 2回目以降での計測
（初回はシェーダコンパイルがあるため）

UE5.1 ：17秒 **UE5.2：2秒**

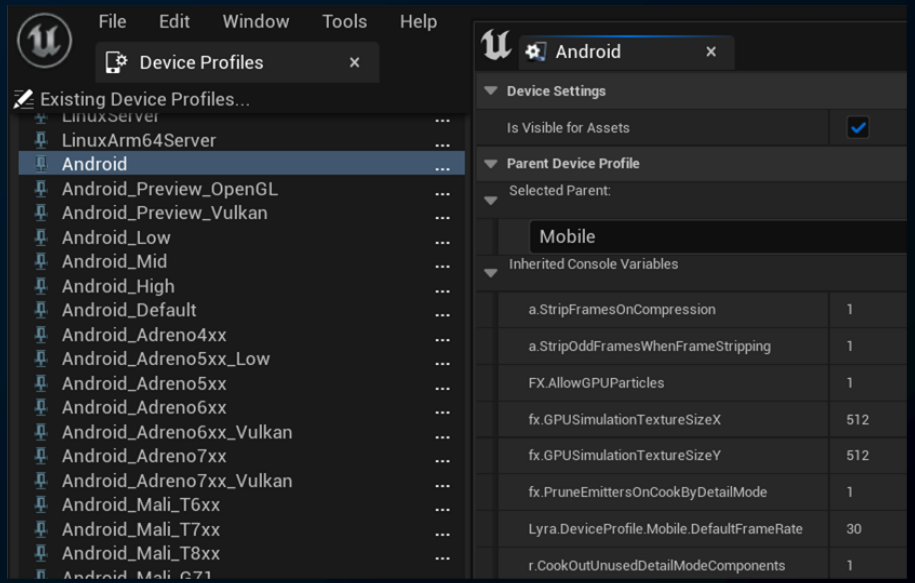
大規模なプロジェクトの場合
5~10分 から 数秒 ~ 1分 に改善することも



プラットフォーム毎の品質・機能ごとの設定を適用し
より高精度なエミュレートを実現



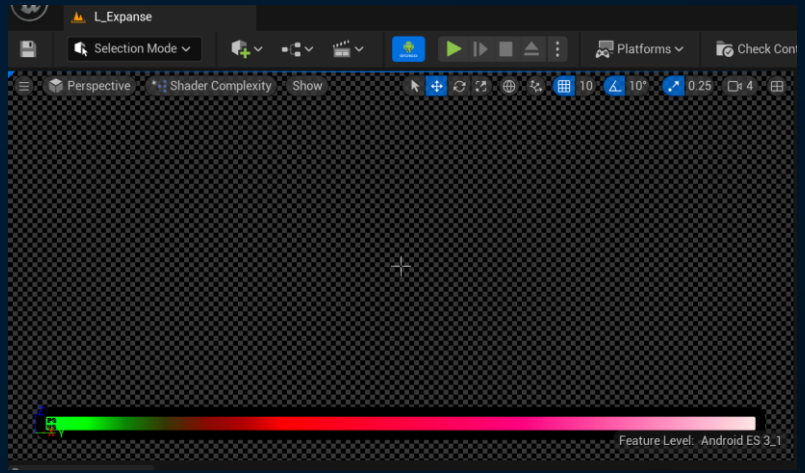
Scalability(機能ごとの品質設定)が
プラットフォーム毎の管理に



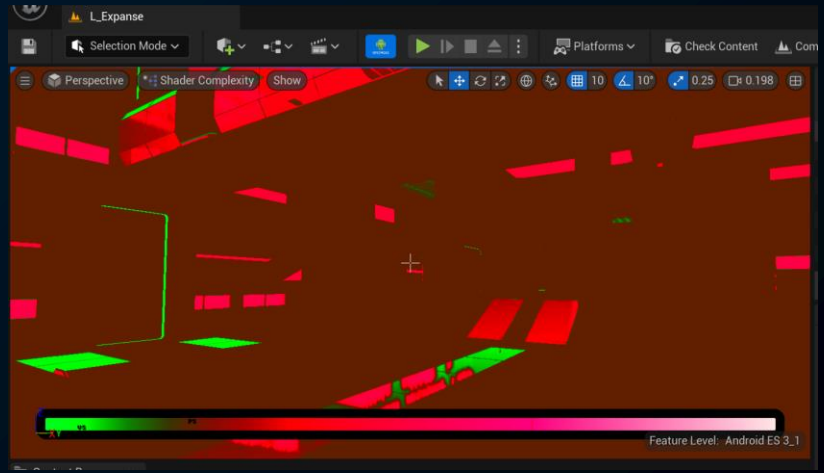
デバイスプロファイルで設定した
CVar(コンソール変数)を適応するように

様々な不具合を修正し、安定性が向上

シェーダ複雑度（Shader Compexity）が
モバイルプレビューでも正常に動作するように（5.3でさらに改善予定）



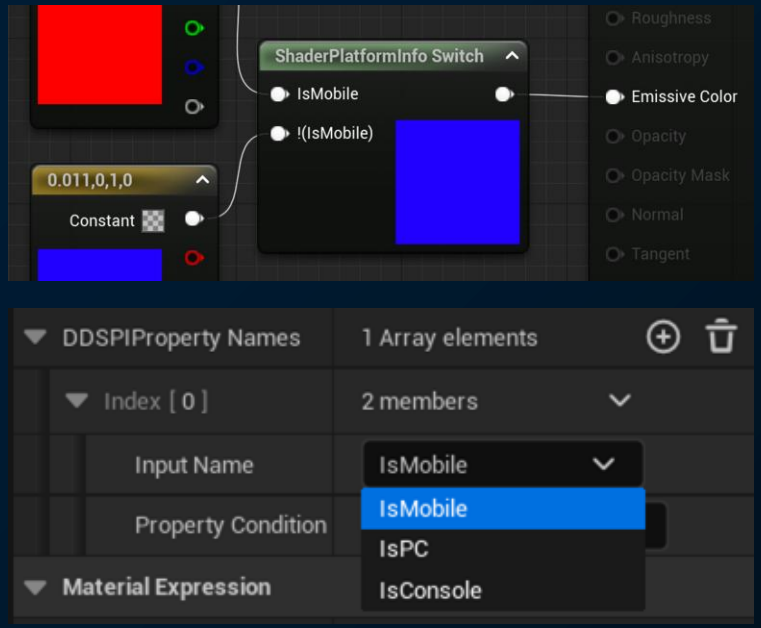
UE5.1

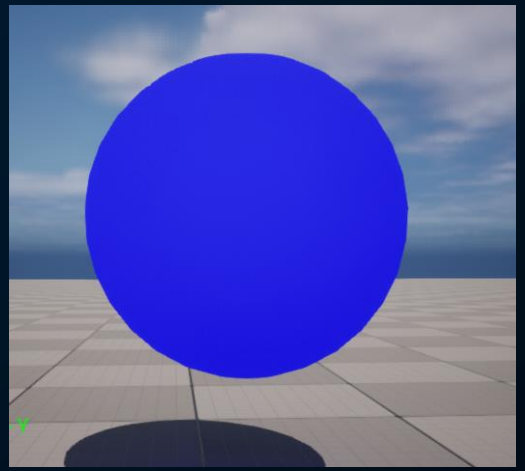


UE5.2

ShaderPlatformInfoSwitch ノードを追加

プラットフォームの種類で Switch できるマテリアルノード





PC

Mobile

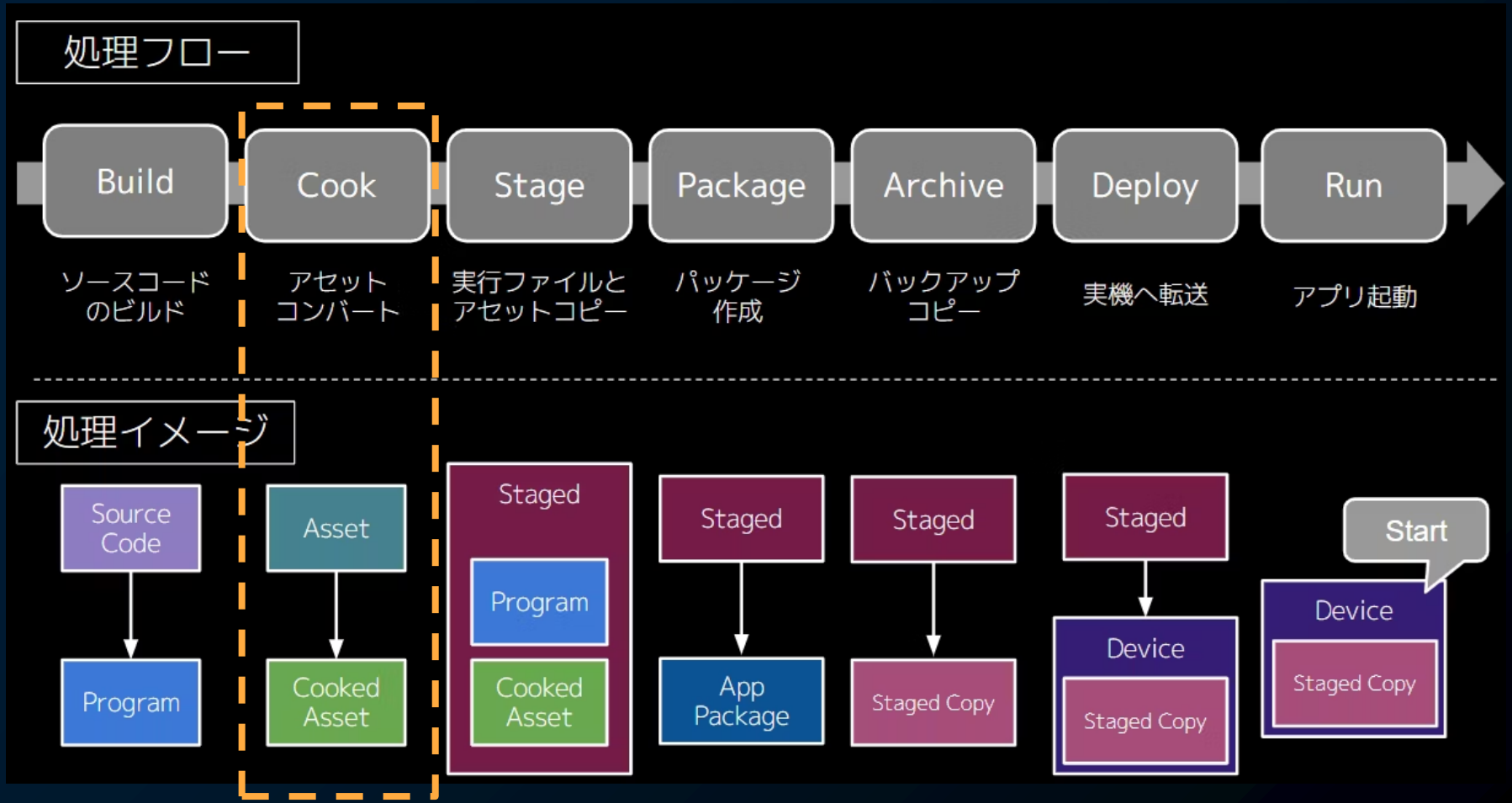
Beta

iOS、tvOS、iPadOS のデバッグの準備 (Prepare For Debugging)

Prepare For Debugging

過去のCook(アセットの変換)データを使い回すことで、
パッケージの作成・デバッグを高速に行うための仕組み

[UE4] ビルドプロセスの流れ　より



「Press Button, Drink Coffee」
UE4における ビルドパイプラインとメンテナンスの全体像より

最低限必要な作業

- Build（コードのビルド）
- Cook（アセットの変換）

最悪のケース

- エディタビルド
  （Cookするために）
- Build, Cook
- 失敗
- コード・アセットをDL
- エディタのビルド
- Build, Cook
- ・・・

Prepare For Debugging の効果

過去のCook結果を使い回すことで…

- Cook処理をスキップ
- エディタのビルドも不要に！

ソースコードのビルドのみで
パッケージ作成完了！

Prepare For Debugging の効果

過去のCook結果を使い回すことで…

- Cook処理をスキップ
- エディタのビルドも不要に！

ソースコードのビルドのみで
パッケージ作成完了！

パッケージの実行・デバッグまで
最長1日以上かかっていたのが…

最短で数秒、長くても数分に！

※ 2019年の画像

※ 横浜ファンの人が作りました

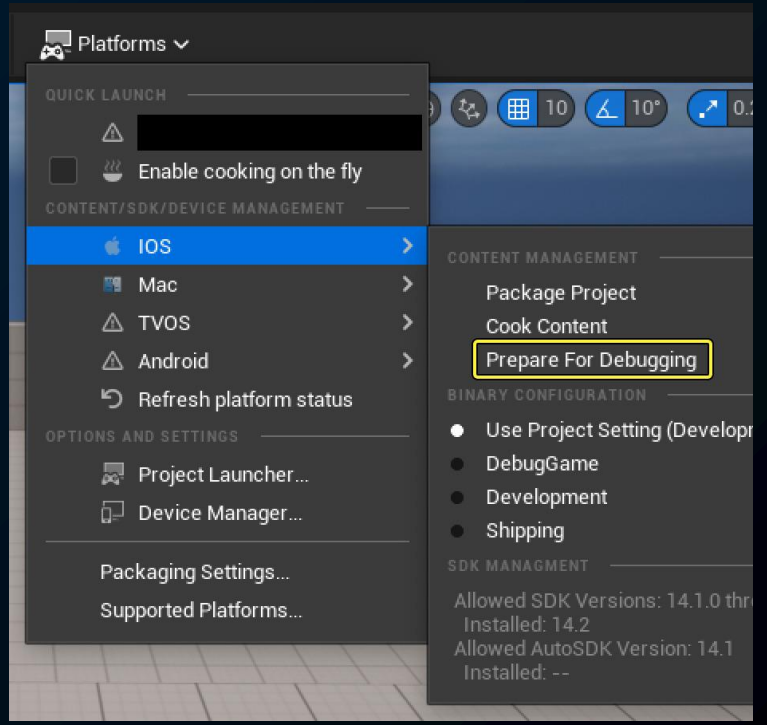
「Press Button, Drink Coffee」
UE4における ビルドパイプラインとメンテナンスの全体像より

Prepare For Debuggingの使い方・今後の予定

UATの
WrangleContentForDebugging コマンド
または、エディタから実行可能

RunUAT.command
WrangleContentForDebugging -platform=ios -
ProjectFilePath=XXX.uproject
-nocompile -nocompileuat

UE5.3 で
Windowsからのデバッグにも対応予定！

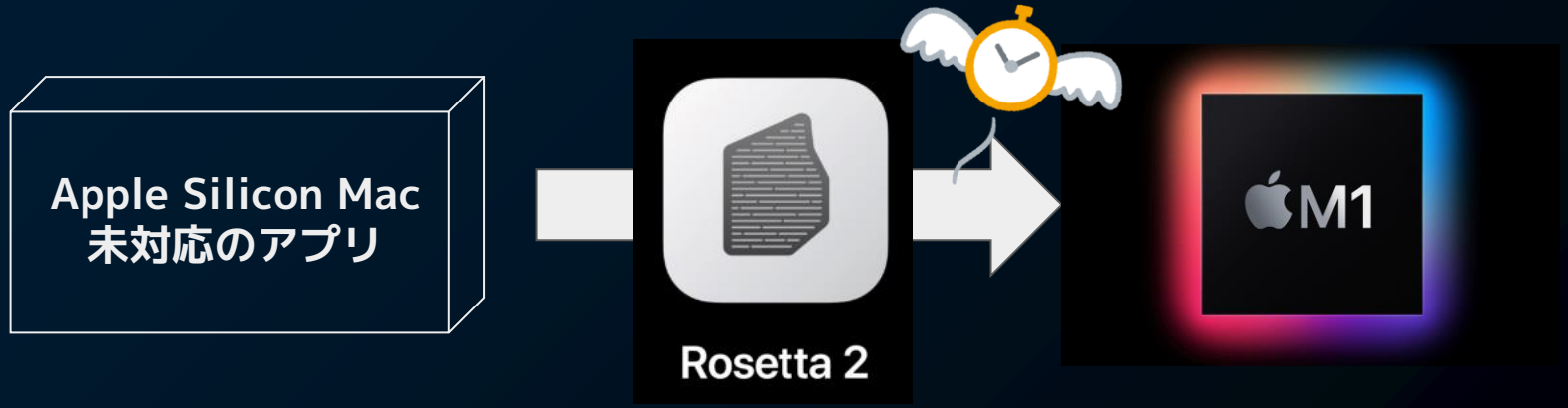


Unreal Editor での Apple Silicon のネイティブ サポート

Intel Mac と Apple Silicon Mac

2020年以降よりIntel製のSoCではなく、
Apple独自のSoC（Apple Silicon：M1, M2）を積んだMacが販売されるように

Intel Mac時代に開発されたアプリを
Apple Silicon Mac で動かすためには、**Rosetta 2 によるエミュレート**が必要



UE5 の Apple Silicon対応状況

UE4.27, UE5.0：
Apple Silicon対応のパッケージを作成可能に！
Unreal Editorは未対応

UE5.1：
Unreal Editor（ビルド版）が対応！
Unreal Editor（ランチャー版）は未対応

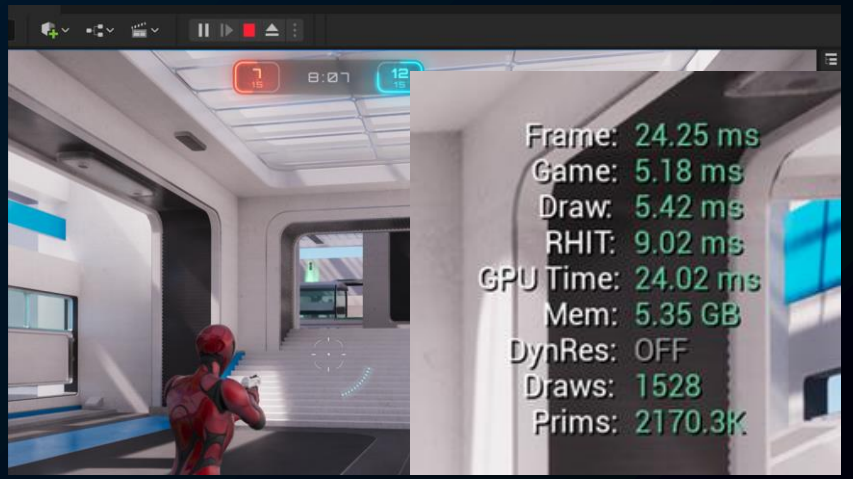
UE5.2：
Unreal Editor（ランチャー版）も対応！
マーケットプレイスも対応

いまここ！

ネイティブサポートによるCPU負荷の削減

CPU負荷 が 3~5 ms 削減！
プロジェクト：Lyra　環境：MacBookPro 2021 (M1 Max)

UE5.1 TAA　Game : 9 ~ 10 ms
SkyLightの不具合あり（5.2で修正済み）



UE5.2 TAA　Game : 5 ~ 8 ms

UE 5.2 における macOS の制限事項

- **Nanite は M2 Mac のみ実験的にサポート**
  デフォルトは無効。有効にするためには コード変更 が必要
- **ヘア (毛髪)/毛皮/グルームのストランド は未サポート**
- ハードウェア レイ トレーシングは非対応のため
  Lumen は ソフトウェアレイトレーシングのみ使用可能
- TSR の最適化が完了していないため、
  アンチエイリアシング設定を FXAAまたはTAAに変更することを強く推奨
  - TSR -> TAA でGPU負荷が 10ms 改善

M2 Nanite

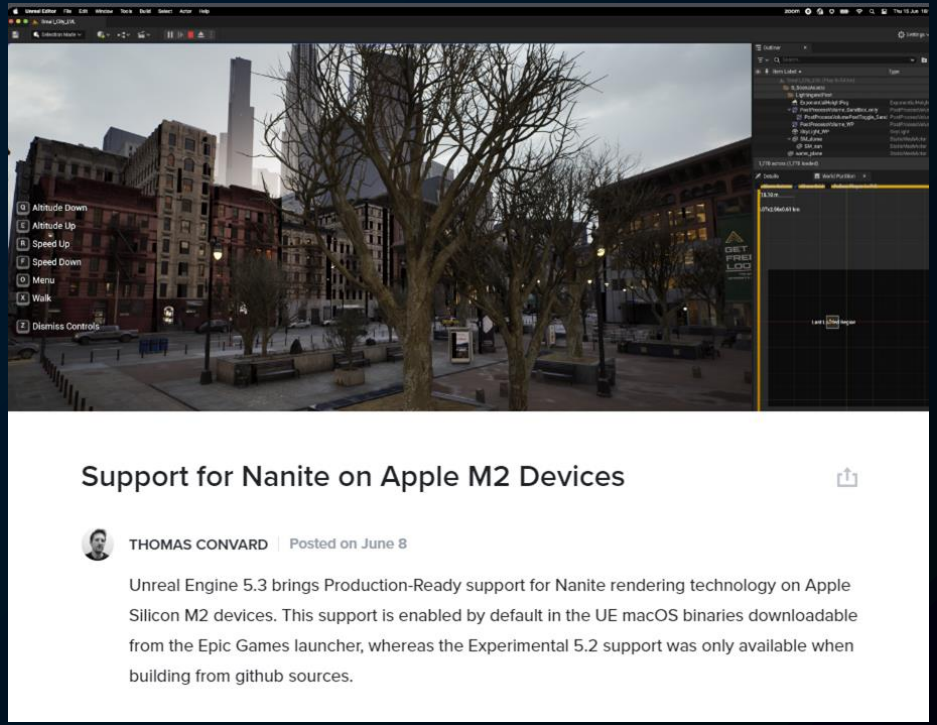
M2 デバイスで実験的な Nanite サポートを使用する方法

待望の macOS での Nanite の完全サポートに向けた最初のステップとして、ue5-main および GitHub 上の 5.2 ブランチで実験的サポートが利用できるようになりました。ただし、これはデフォルトでは無効になっています。これを有効にするには、Unreal Engine のソース コードで次の変更を行う必要があります。

1. 「UEBuildMac.cs (Engine/Source/Programs/UnrealBuildTool/Platform/Mac/UEBuildMac.cs)」で PLATFORM_MAC_ENABLE_EXPERIMENTAL_NANITE_SUPPORT=1 を設定します。
2. 「spirv_msl.hpp」で UE_EXPERIMENTAL_MAC_NANITE_SUPPORT を有効にします。
3. 「UEBuildMac.cs」の指示に従って ShaderConductor をリビルドします。
4. 「DataDrivenPlatformInfo.ini (Engine/Config/Mac/DataDrivenPlatformInfo.ini)」で bSupportsNanite=true と bSupportsUInt64ImageAtomics=true を設定します。

https://www.unrealengine.com/ja/tech-blog/unreal-engine-5-2-brings-native-support-for-apple-silicon-and-other-developments-for-macos

UE5.3 から
M2 Macにおける
Naniteサポートが
Production-Readyに



https://portal.productboard.com/epicgames/1-unreal-engine-public-roadmap/c/1151-support-for-nanite-on-apple-m2-devices

Appleプラットフォーム向けのUE開発環境 (2023/6)

UE5.1 ~ 5.2 で 多くの改善が入り
1~2年前と比べると、Macを使った開発が凄く快適に！

今から開発環境を整えるなら

- **M1 ではなく、M2 系のチップ搭載で**
- **メモリは 16 GB 以上 に**
- 負荷が低いシーンなら、MacBook Air でもOK
  高負荷なら、MacBook Pro 以上で

Thanks Axel !

モバイル開発向けの機能追加・改良は
UE5.3以降もガンガン入る予定です

ご期待下さい！

鈴木パート へ

NNE plugin

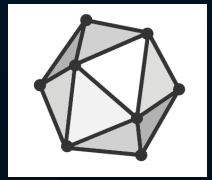
Neural Network Engine

NNE プラグイン

目的

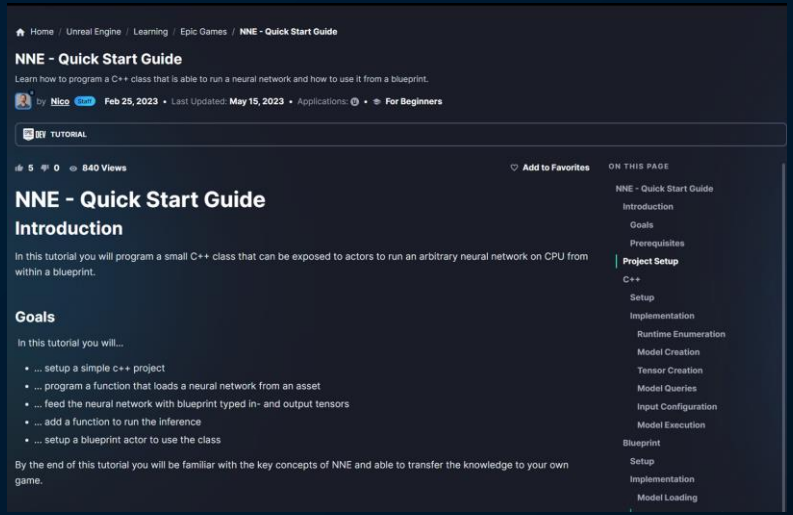
- 学習済みの**ONNXファイル**をハイパフォーマンスで実行
- クロスプラットフォームで動作 (デスクトップ + 現世代コンソール機)

- UE5.0で導入されたNNIプラグインが改名されたもの
- CPUまたはGPUで動作するが一部プラットフォームは現時点ではCPUのみ

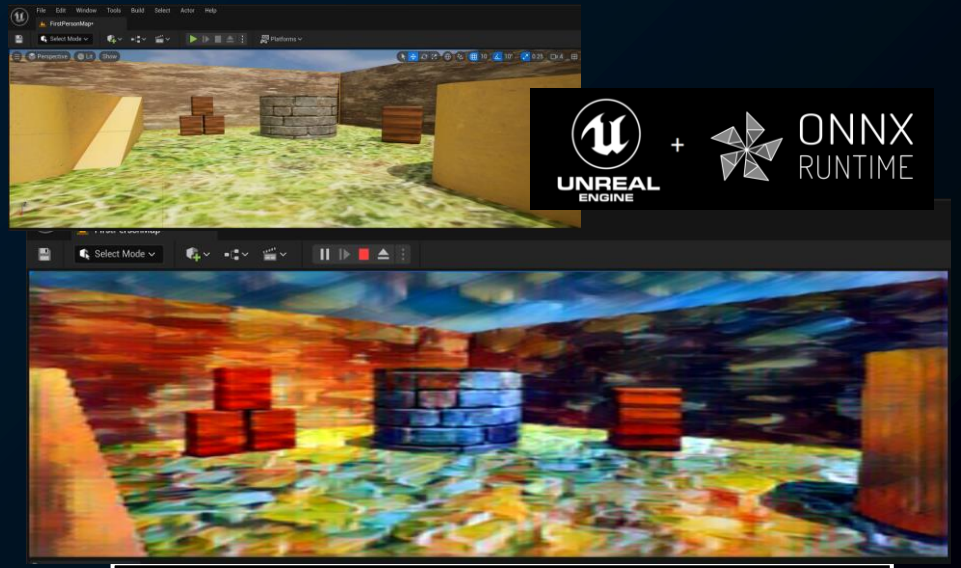


https://onnx.ai/

NNE プラグイン 学習用コンテンツ

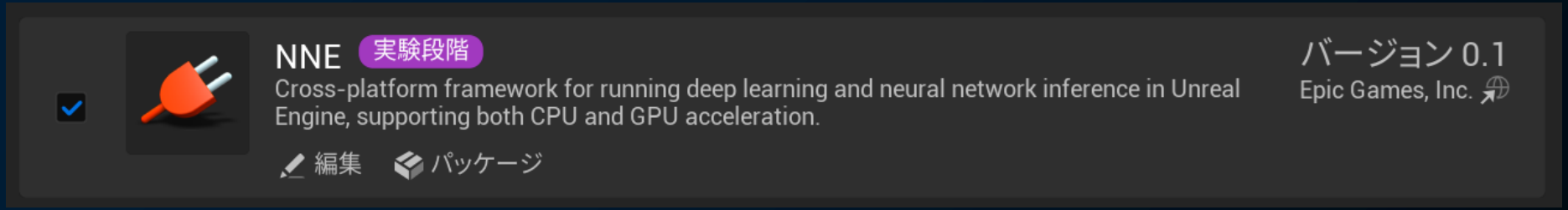


https://dev.epicgames.com/community/learning/tutorials/yXJ3/unreal-engine-nne-quick-start-guide



https://github.com/microsoft/OnnxRuntime-UnrealEngine

NNE プラグイン



Experimental

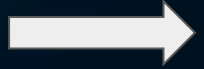
NNE プラグイン の動作の流れ

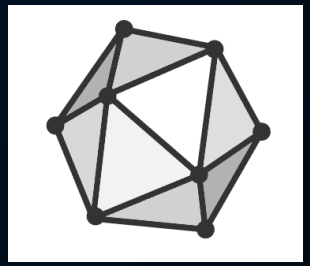
実コードサンプル

```
bool UNeuralNetworkModel::SetInputs(const TArray<FNeuralNetworkTensor>& Inputs)
{
    check(Model.IsValid())

        using namespace UE::NNECore;

    InputBindings.Reset();
    InputShapes.Reset();

    TConstArrayView<FTensorDesc> InputDescs = Model->GetInputTensorDescs();
    if (InputDescs.Num() != Inputs.Num())
    {
        UE_LOG(LogTemp, Error, TEXT("Invalid number of input tensors provided"));
        return false;
    }

    InputBindings.SetNum(Inputs.Num());
    InputShapes.SetNum(Inputs.Num());
    for (int32 i = 0; i < Inputs.Num(); i++)
    {
        InputBindings[i].Data = (void*)Inputs[i].Data.GetData();
        InputBindings[i].SizeInBytes = Inputs[i].Data.Num() * sizeof(float);
        InputShapes[i] = FTensorShape::MakeFromSymbolic(
            FSymbolicTensorShape::Make(Inputs[i].Shape)
        );
    }

    if (Model->SetInputTensorShapes(InputShapes) != 0)
    {
        UE_LOG(LogTemp, Error, TEXT("Failed to set the input shapes"));
        return false;
    }
    return true;
}
```

```
bool UNeuralNetworkModel::RunSync(UPARAM(ref) TArray<FNeuralNetworkTensor>& Outputs)
{
    check(Model.IsValid());

    using namespace UE::NNECore;

    TConstArrayView<FTensorDesc> OutputDescs = Model->GetOutputTensorDescs();
    if (OutputDescs.Num() != Outputs.Num())
    {
        UE_LOG(LogTemp, Error, TEXT("Invalid number of output tensors provided"));
        return false;
    }

    TArray<FTensorBindingCPU> OutputBindings;
    OutputBindings.SetNum(Outputs.Num());
    for (int32 i = 0; i < Outputs.Num(); i++)
    {
        OutputBindings[i].Data = (void*)Outputs[i].Data.GetData();
        OutputBindings[i].SizeInBytes = Outputs[i].Data.Num() * sizeof(float);
    }

    return Model->RunSync(InputBindings, OutputBindings) == 0;
}
```

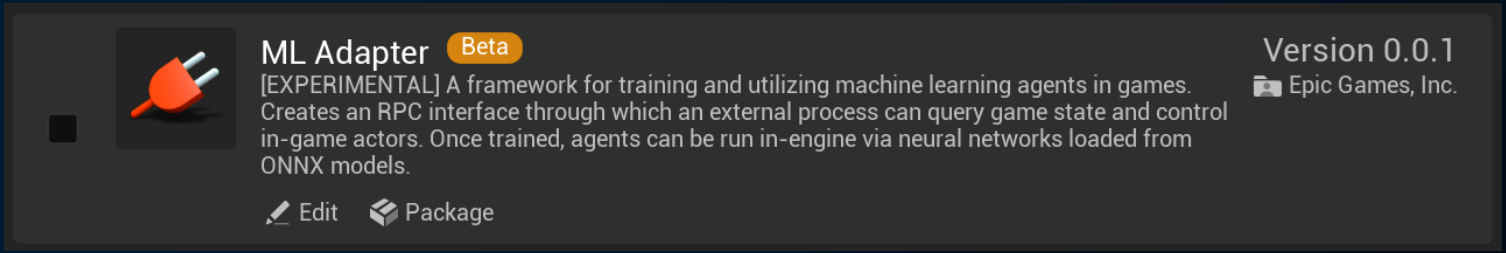
ONNXファイルは各種学習フレームワークから





https://onnx.ai/supported-tools.html#buildModel

ゲーム内容を学習させたい！



Very Experimental

続報をお待ちください。

参考リンク

- [UE5] Neural Network Inferenceプラグインについて
  https://historia.co.jp/archives/26238/

Chaos Flesh

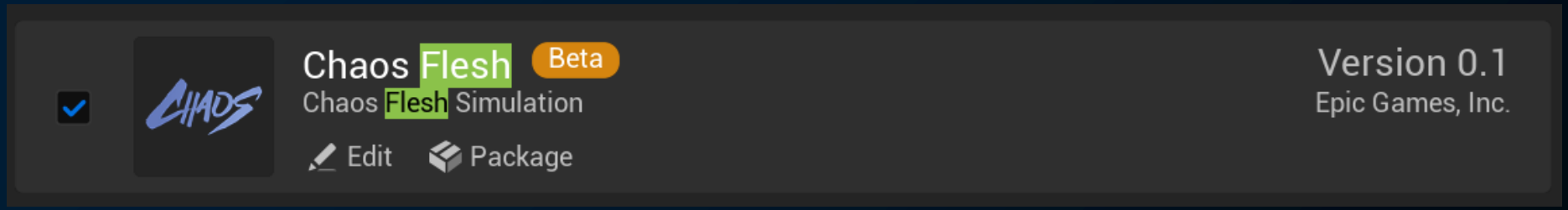
ChaosFlesh

目的

- 組み込みの**軟体シミュレーション**
- エンジン内で編集

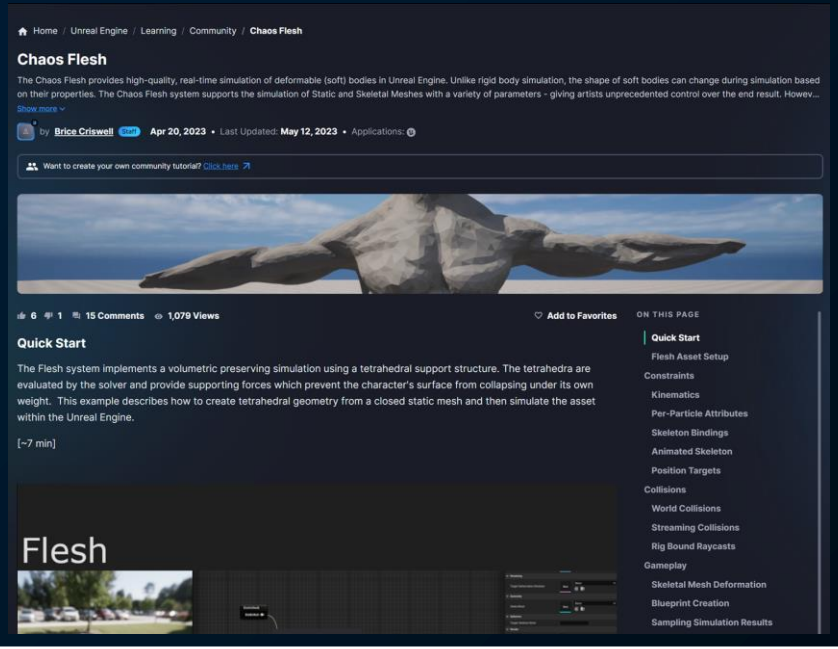
- 四面体メッシュによるシミュレーション
- シミュレーション結果を表示用メッシュに転送
- デバッグ用に四面体ボリュームを表示可能

Chaos Flesh プラグイン



Experimental

Chaos Flesh チュートリアル



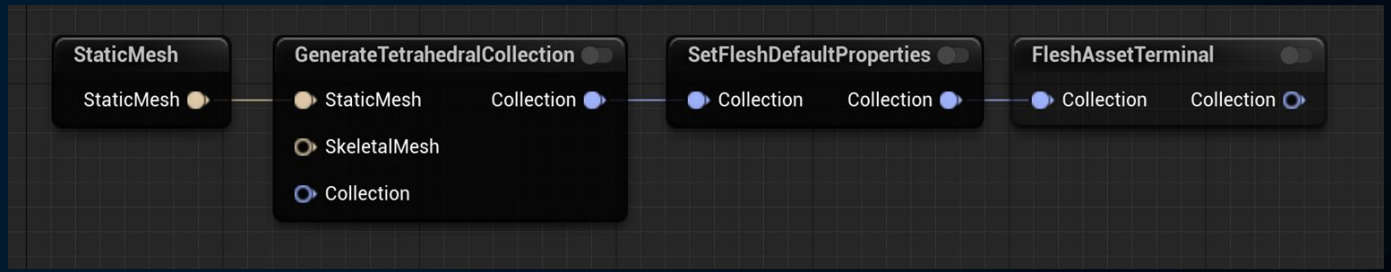
https://dev.epicgames.com/community/learning/tutorials/BEby/unreal-engine-chaos-flesh

Chaos Flesh 構成

- DataGraph
  - 四面体メッシュ作成処理を記述したノードグラフ
- FleshAsset
  - DataGraphと使用するメッシュなどのパラメータ
  - 生成された四面体メッシュを保持
- DeformableSolverActor/DeformableSolverActor
  - シミュレーションソルバー
- FleshComponent
  - FleshAssetとDeformableSolverを指定

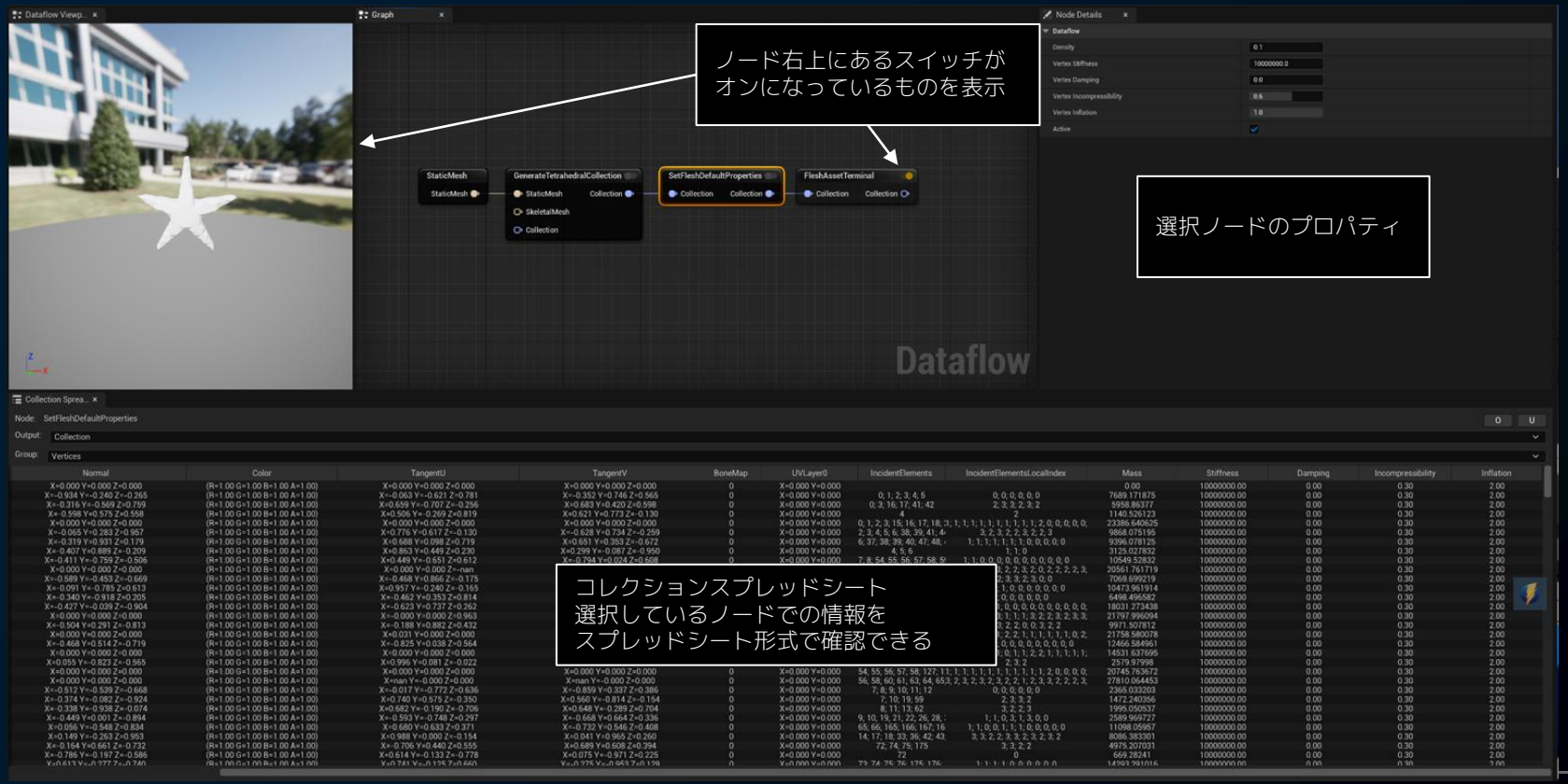
DataGraph 概要

- シミュレーションメッシュ(四面体メッシュ)の作成
- シミュレーション頂点に値を設定
- コレクションをFleshAssetTerminal入力

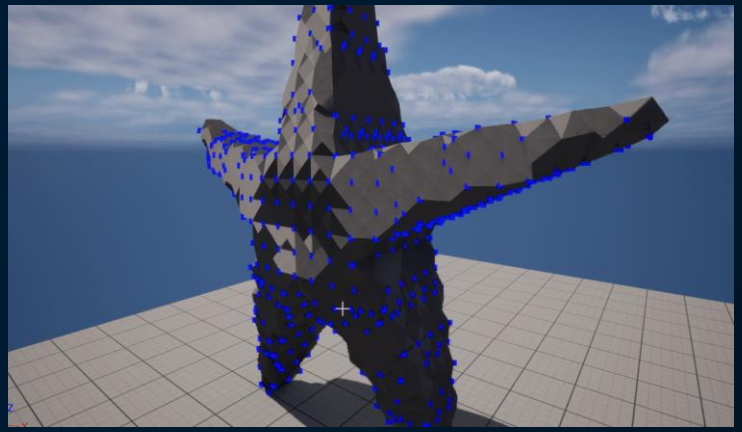


最小データグラフ

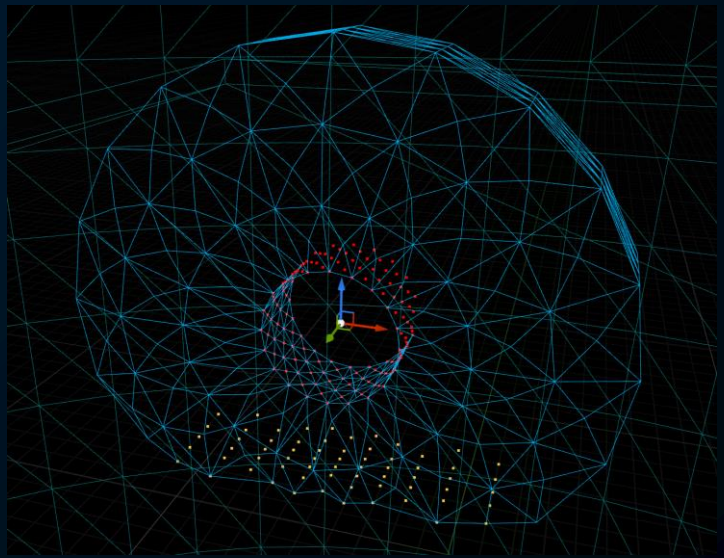
DataGraph エディタ



DataGraph デバッグ用コンソールコマンド

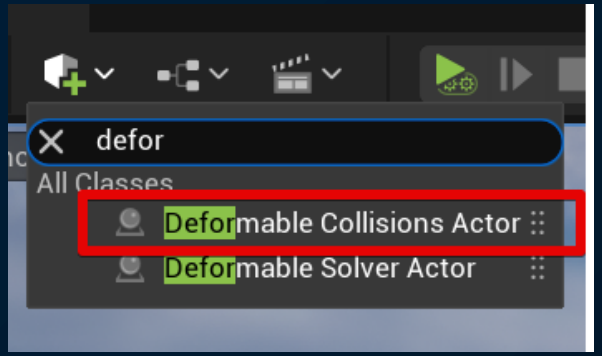


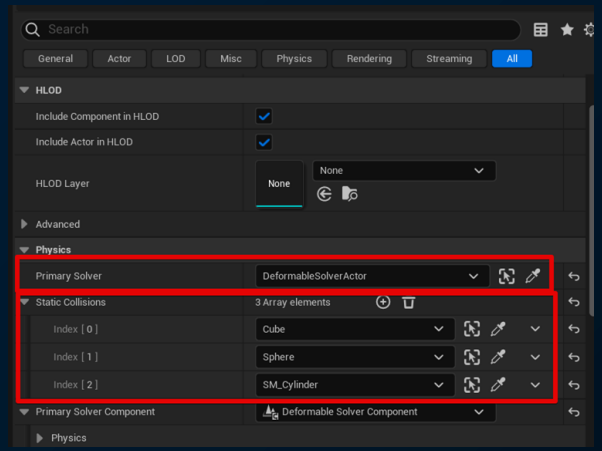
p.Chaos.DebugDraw.Enabled 1
p.Chaos.DebugDraw.Deformable.TetrahedralParticle 1

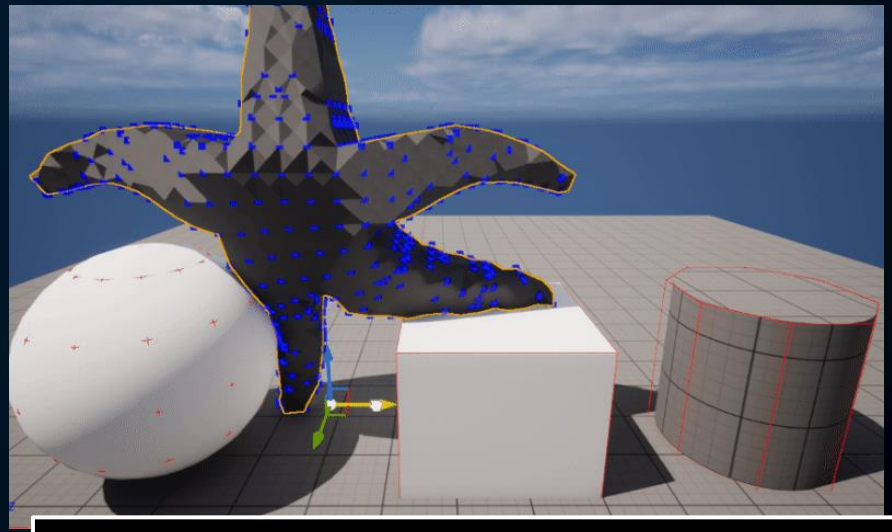


p.Chaos.DebugDraw.Deformable.KinematicParticle 1
p.Chaos.DebugDraw.Deformable.TransientKinematicParticle 1

DeformableCollision

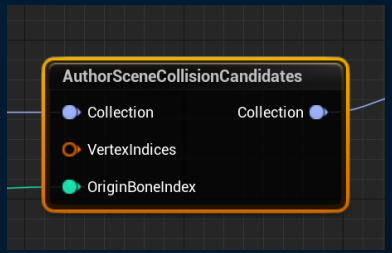


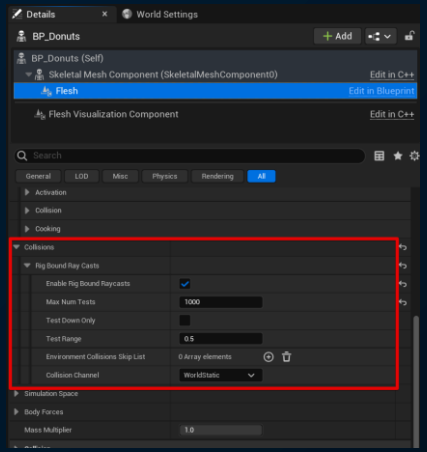


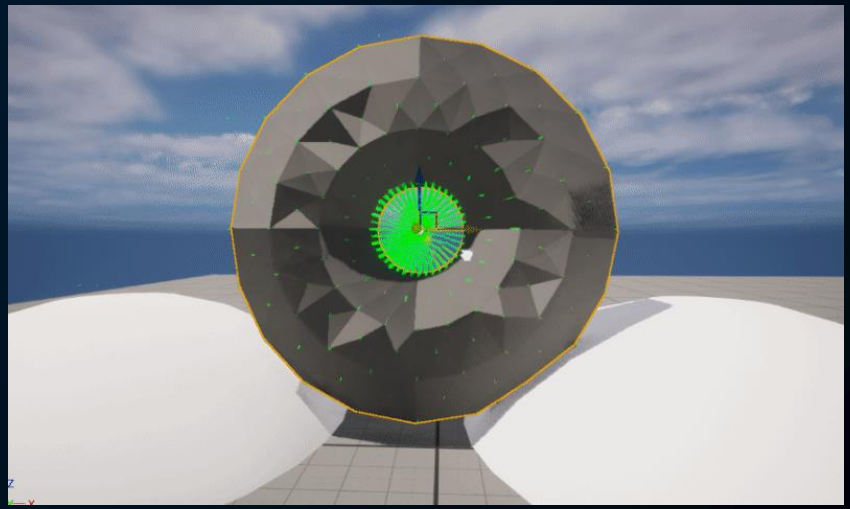


p.Chaos.DebugDraw.Deformable.KinematicParticle 1
p.Chaos.DebugDraw.Deformable.RigidCollisionGeometry 1

RaycastCollision



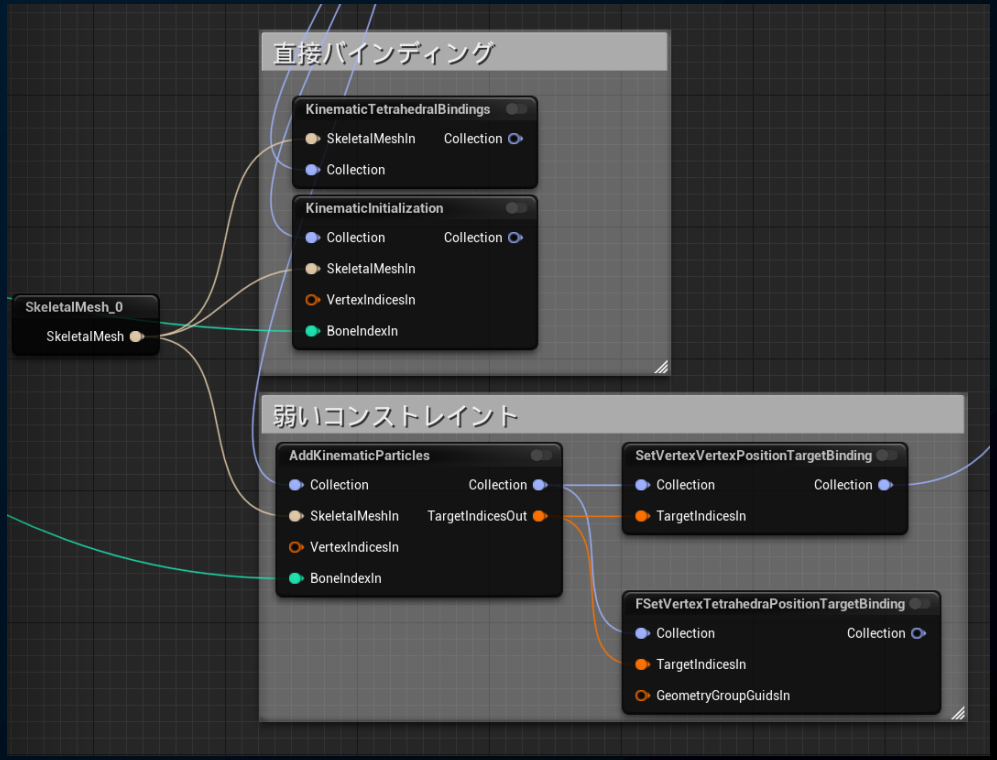




p.Chaos.DebugDraw.Enabled 1
p.Chaos.DebugDraw.Deformable.SceneRaycasts 1

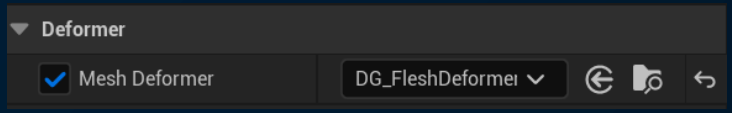
アニメーションバインディング

- 親のSkeletalMeshComponentの
  アニメーションを取り込む
- 参照方法
  - 直接バインディング
  - ポジションターゲット

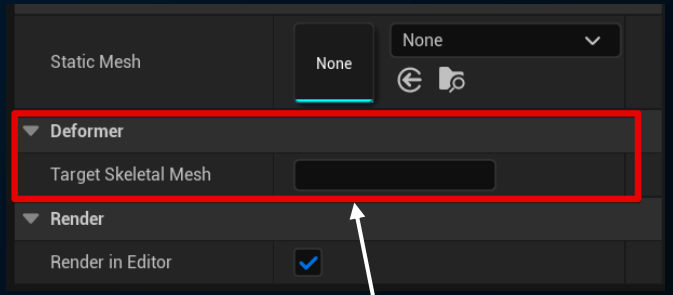


SkeletalMeshで描画

SkeletalMeshComponentのMeshDeformer設定



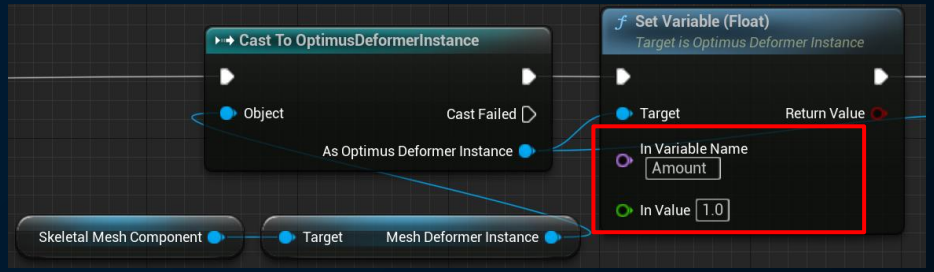
FleshAssetのTargetSkeletalMesh設定



Component名を指定

SkeletalMeshが一つしかないなら空でもOK

DeformerInstanceにAmountを設定

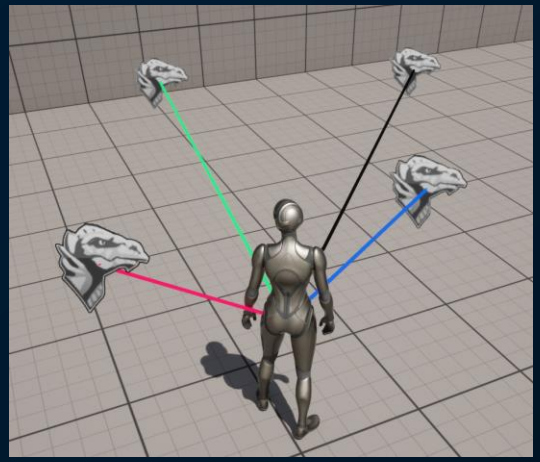


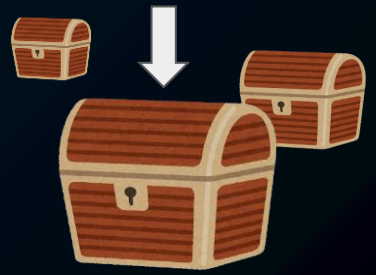
Targeting System プラグイン

Targeting System プラグイン Beta

目的

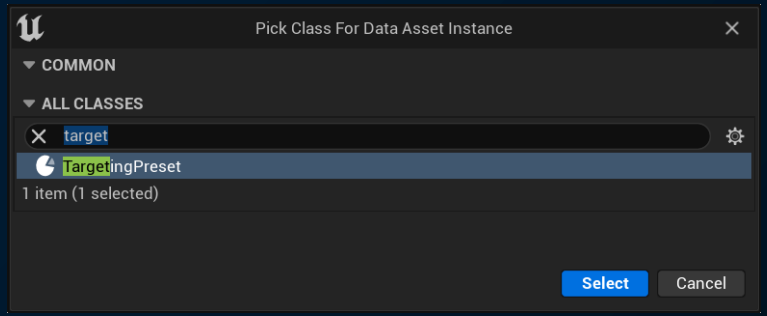
- 同期または非同期でアクターを検索する仕組み
- ゲームプレイアビリティまたはBP/C++から呼び出して利用

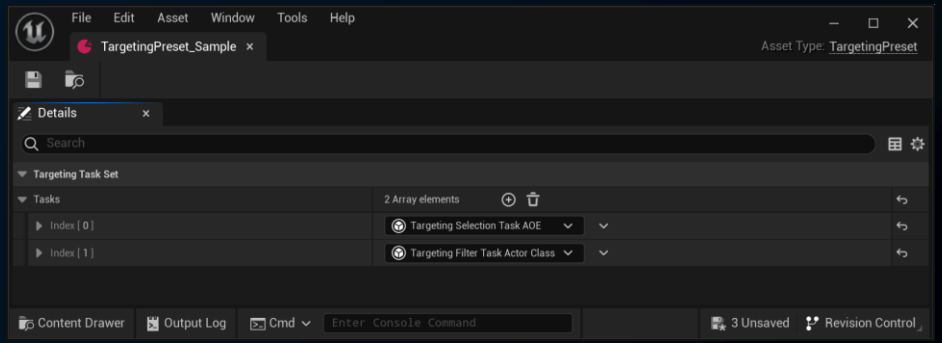




TargetingPreset

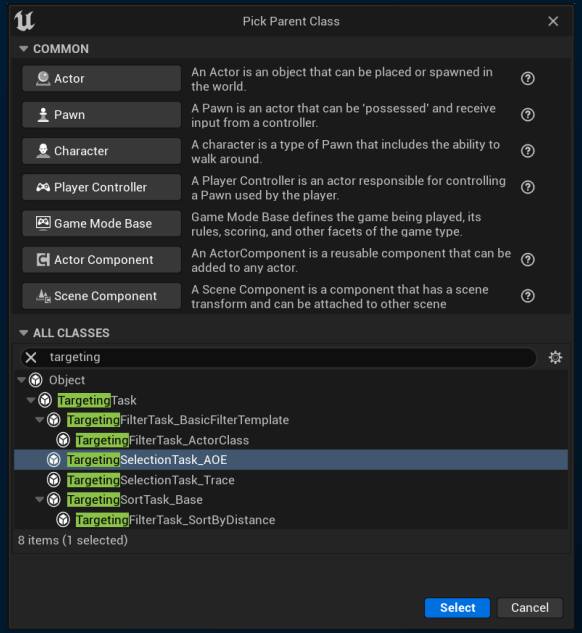
- TargetingPresetの作成
  - **DataAssetから作成**する
- TargetingPresetの設定
  - **Targeting Task**を配列に登録 (先頭から評価)
  - Targeting Selection ～ で
    まず対象のアクタをリストアップ
  - フィルタやソートなどの加工処理を並べる



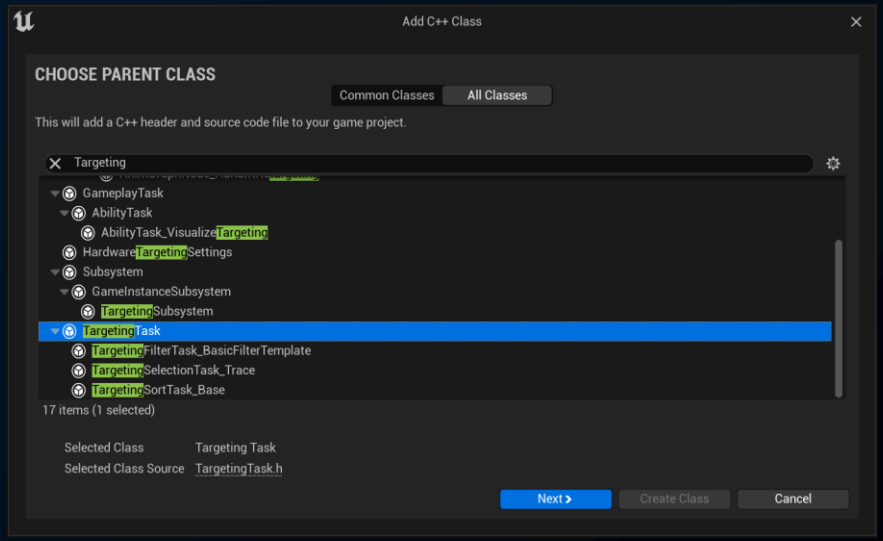


c++またはブループリントでの拡張

- Targeting*Task* を継承して
  ブループリントを作成

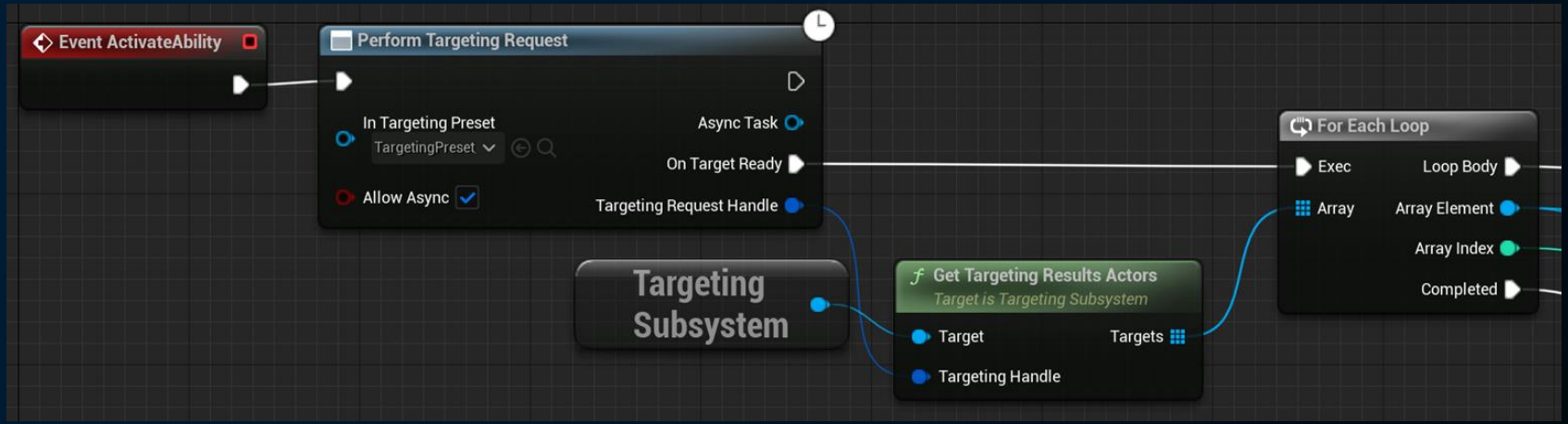


- UTargetTaskなどを継承した
  C++クラスを作成

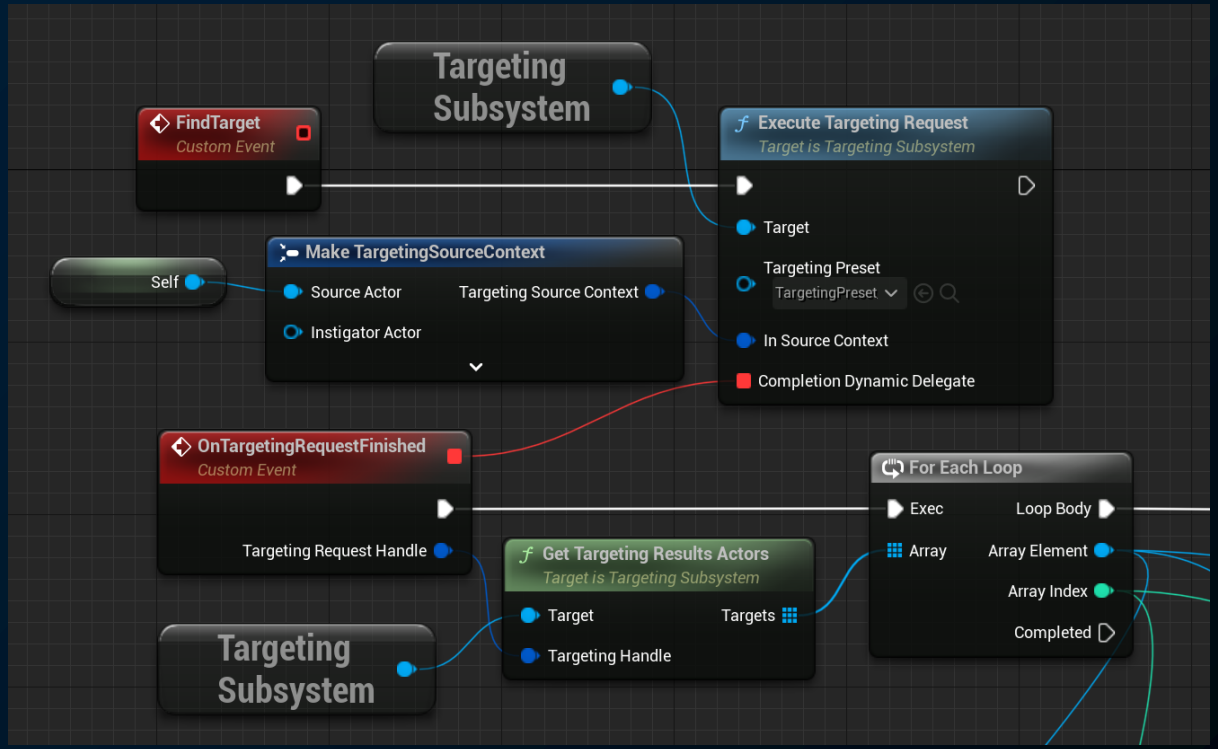


UNREAL ENGINE

Targeting Task 実行 (GameplayAbilityから)

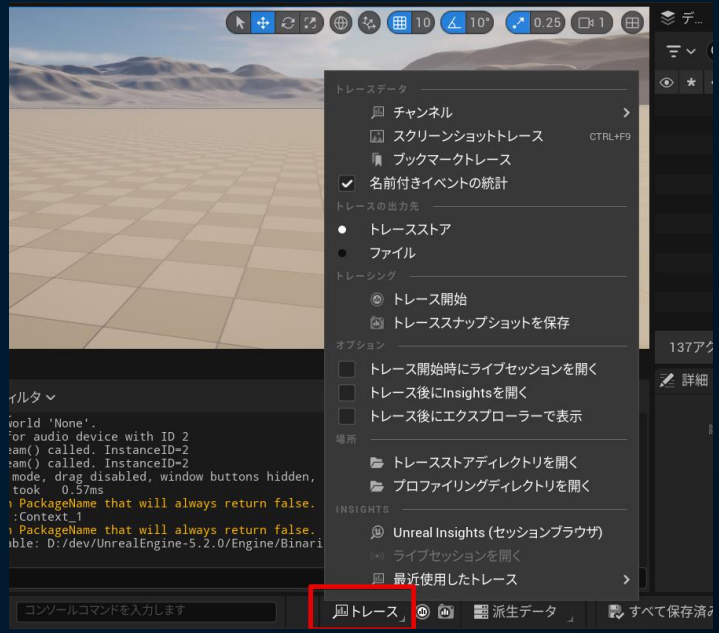


Targeting Task 実行 GA以外から

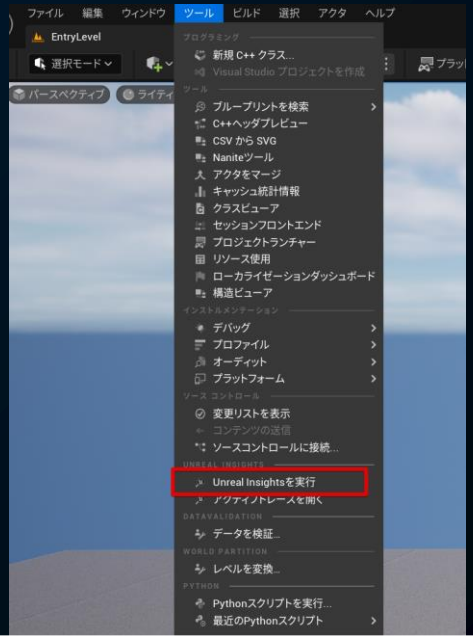


Unreal Insights 改善

エディターからの操作が大幅改善

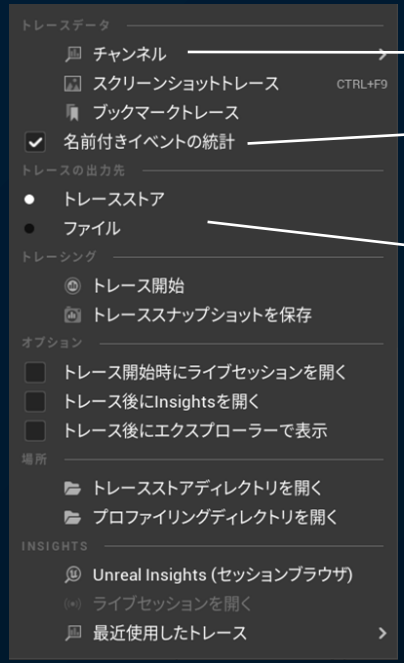


5.2 ~



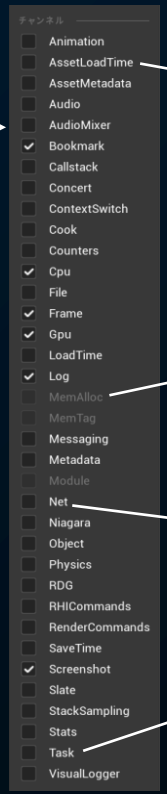
5.0 ~ 5.1

トレースのメニュー



stats.namedevents 0/1

保存位置指定



ロード時間計測
+
ブループリントの関数名表示

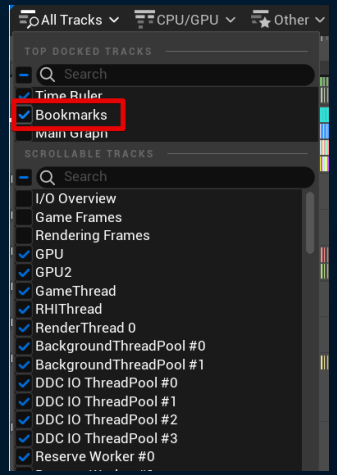
Memory Insights
メモリーリーク調査

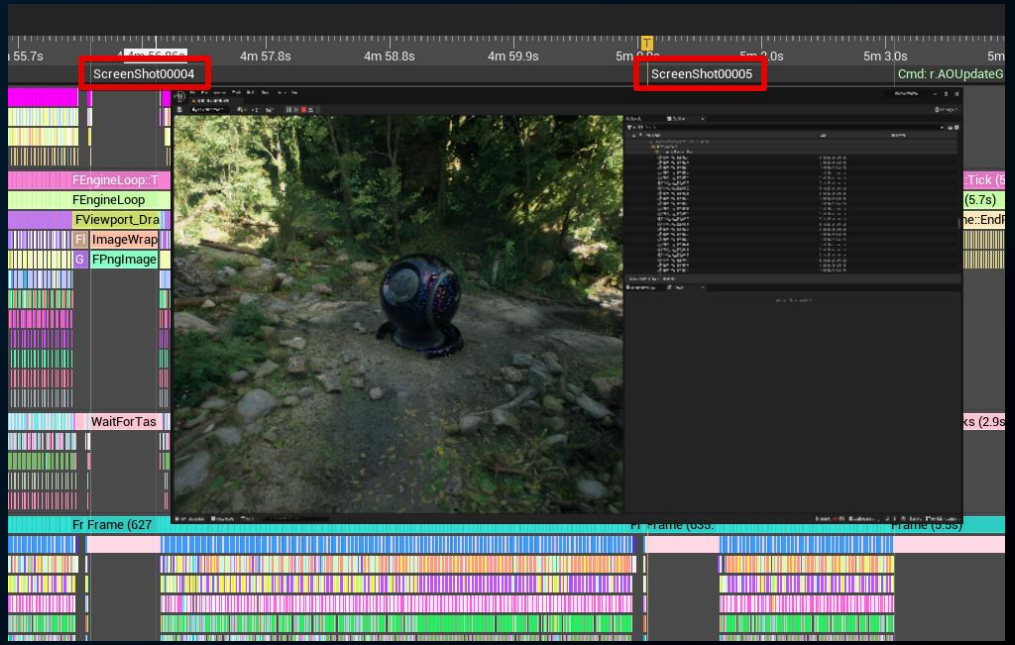
Network Insights
通信内容の調査

Task Insights
タスク間依存関係の表示

スクリーンショットトレース

- スクリーンショットを記録
- Bookmark行でポイントして表示
- Trace.SnapshotSend

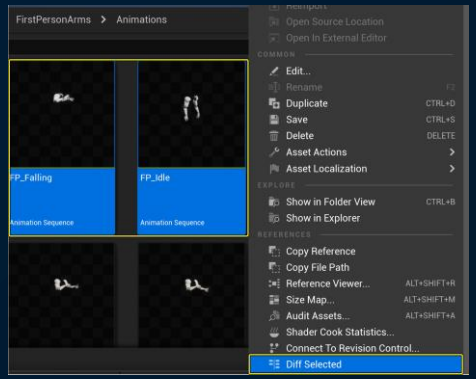




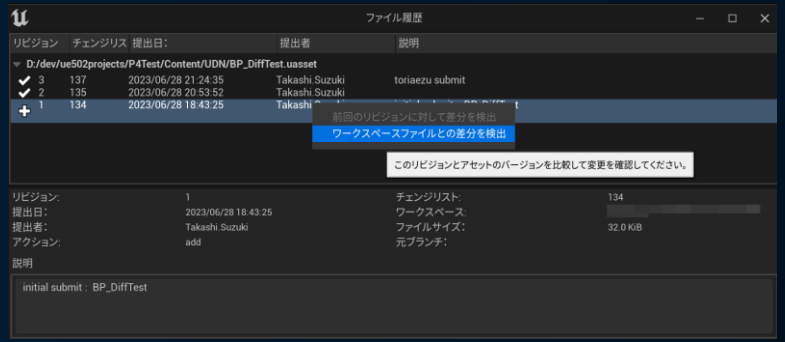
UNREAL ENGINE

Diff/Reviewツール

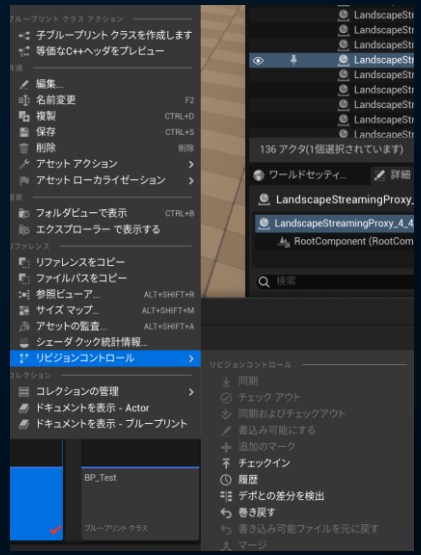
Diffツール



二つのファイルの比較



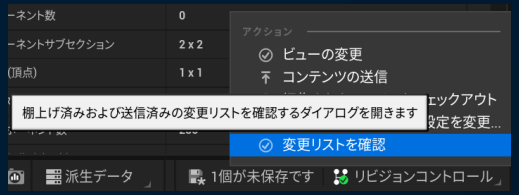
ファイル履歴からの比較



編集中のファイルとディポの比較

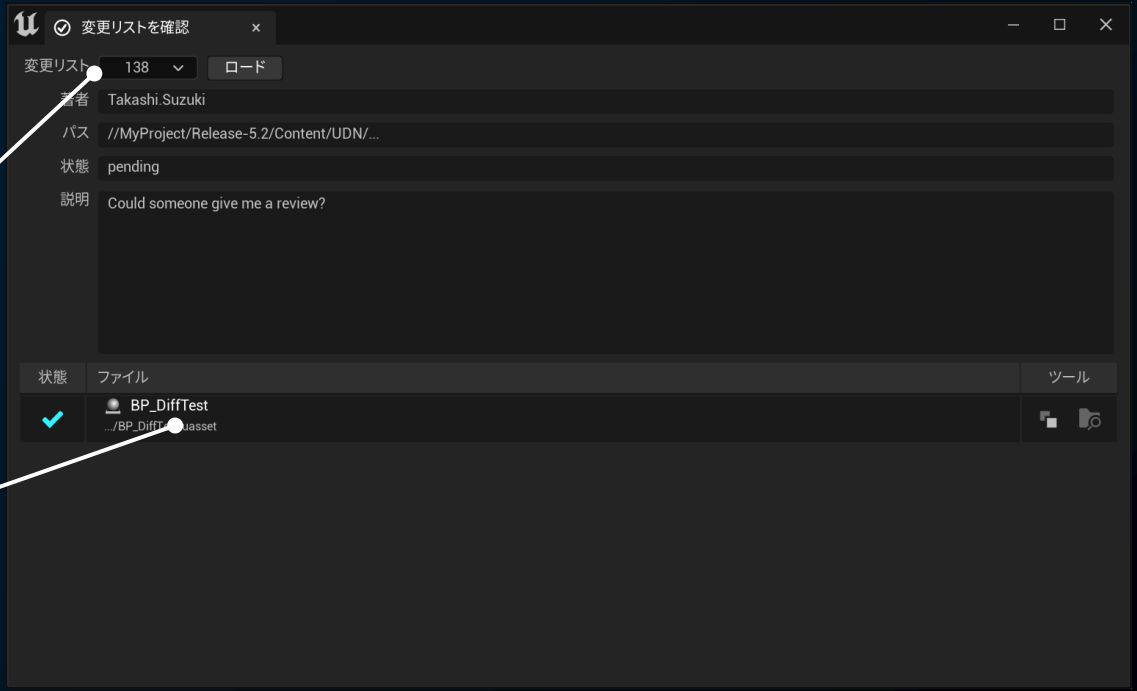
Unreal Diff ツール
https://docs.unrealengine.com/5.2/ja/ue-diff-tool-in-unreal-engine/

Reviewツール



チェンジリスト番号を
入力してロードを押す

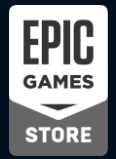
ファイルを選んでDiff



最後に

- メールでのお問い合わせ先
  - EGJ-support@epicgames.com
  - [カスタムライセンシー様向け]
    - 技術的課題や問題点に関するご相談
    - パフォーマンスプロファイリング・チューニング
  - 弊社製品に関するご契約などのご相談





- 新たにライセンスをご検討される方
  - https://www.unrealengine.com/ja/license#contact-us-form
- 過去の講演のスライド | Docswell
  - https://www.docswell.com/user/EpicGamesJapan
- Unreal Fest 2023 Tokyo プレイリスト
  - https://www.youtube.com/playlist?list=PLr_Cbd4sUDTzvdUTMe6cB5TVKDFZ8fMfC


EPIC
GAMES