



BLUE PROTOCOLにおける キャラクタークリエイション

～誰でも簡単！異世界転生のお手伝い～

株式会社バンダイナムコスタジオ

第3スタジオ 第11プロダクション

松尾行恵 / 金田直隆

はじめに

- ・ 撮影・SNS投稿はOKです
- ・ 資料はCEDiLに公開予定です
- ・ 質疑は講演の最後と「Ask the speaker」で受け付けます



自己紹介



松尾 行恵

株式会社バンダイナムコスタジオ 第3スタジオ

1998年株式会社ナムコ（当時）入社

「リッジレーサーV」「ゆめりあ」「ファミリースキー」

など、家庭用ゲームのキャラモデル制作を担当。

「BLUE PROTOCOL」ではキャラクリエイションやフェイシャルの仕組みなどを
含めて担当しています。





アジェンダ(前半)

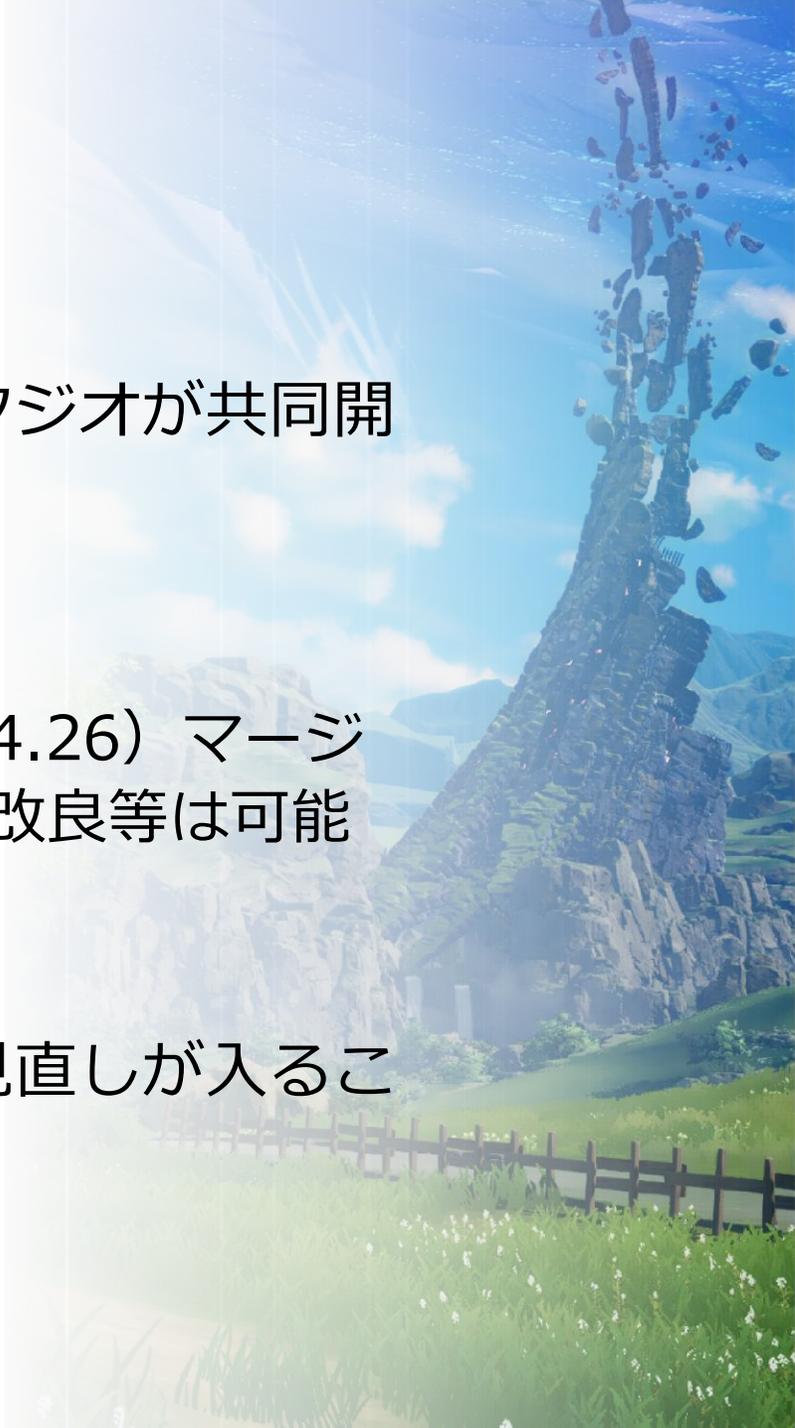
- キャラクタークリエイションの紹介
- コンセプト
- チームの体制
- 仕組み
- アクシデント
- リアル系のキャラクタークリエイションとの違い
- 前半のまとめ



ブループロトコルとは

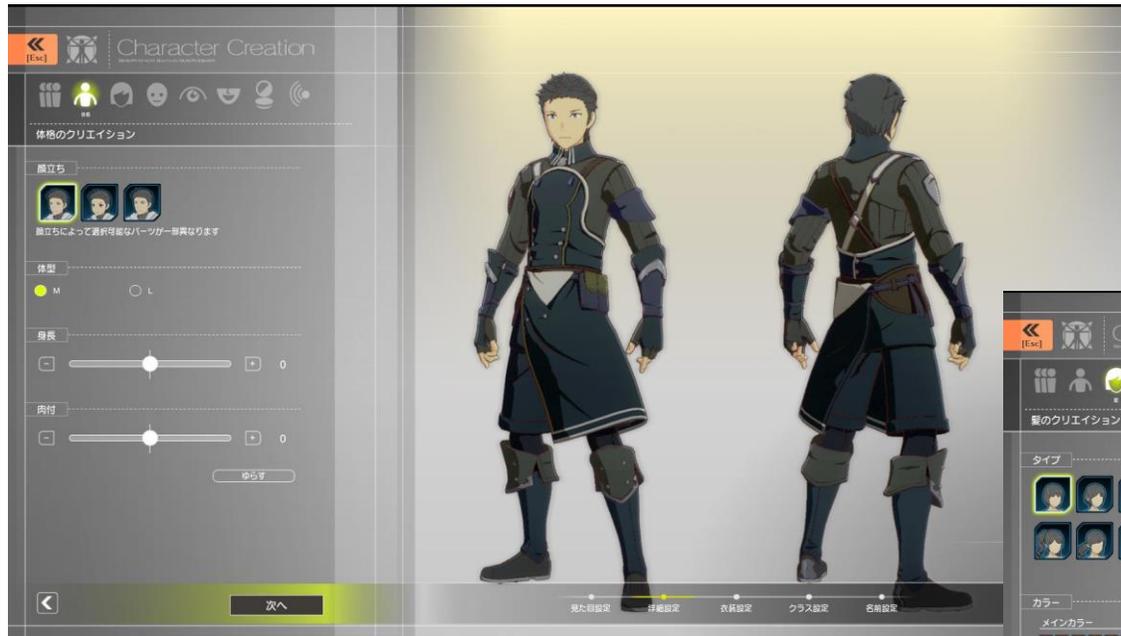
- ・バンダイナムコオンライン、バンダイナムコスタジオが共同開発中のPC向けオンラインアクションRPG
- ・ UnrealEngin4 (UE4) 使用
 - ・ バージョンは、ほぼ最新バージョンに（現在4.26）マージ
 - ・ プロジェクトの方針としては、不具合修正、改良等は可能

※開発中のプロジェクトであり、今後内部実装の見直しが入ることが予想されます



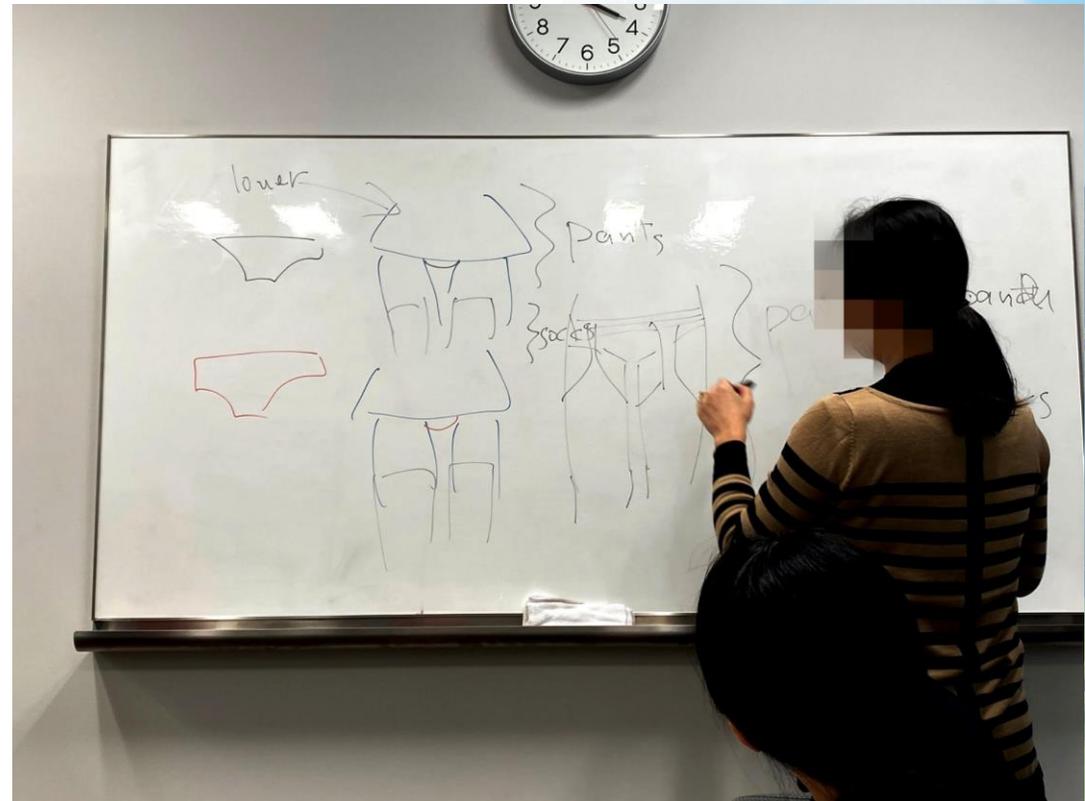
コンセプト

- アニメの設定画のような見た目
- 操作は簡単に！
- 作画崩壊 絶対ダメ！



キャラクリ制作の体制

- 企画
- エンジニア
- アートディレクター
- キャラクター班
- モーション班
- UI班

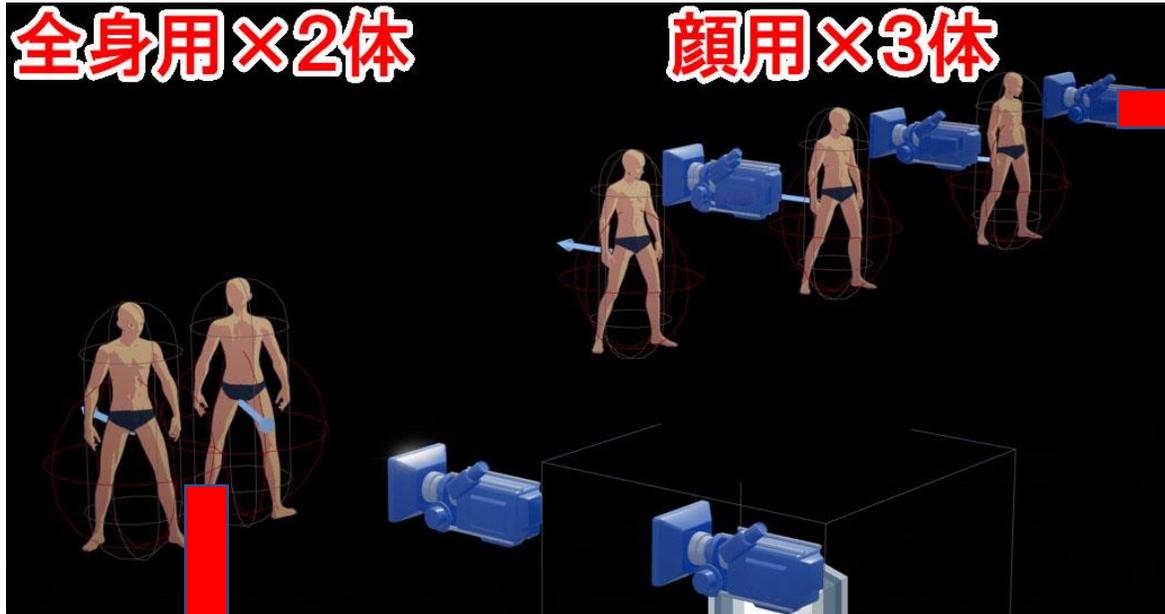


図解しながら説明することもありました



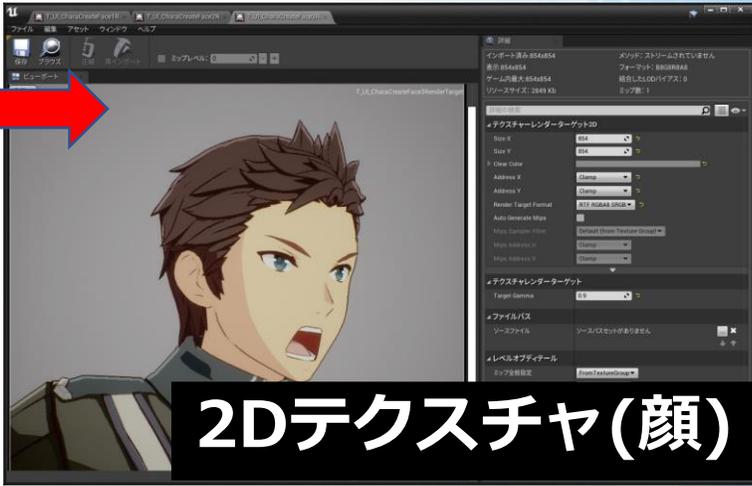
仕組み

- ・キャラクリの画面について



全身用×2体

顔用×3体



2Dテクスチャ(顔)



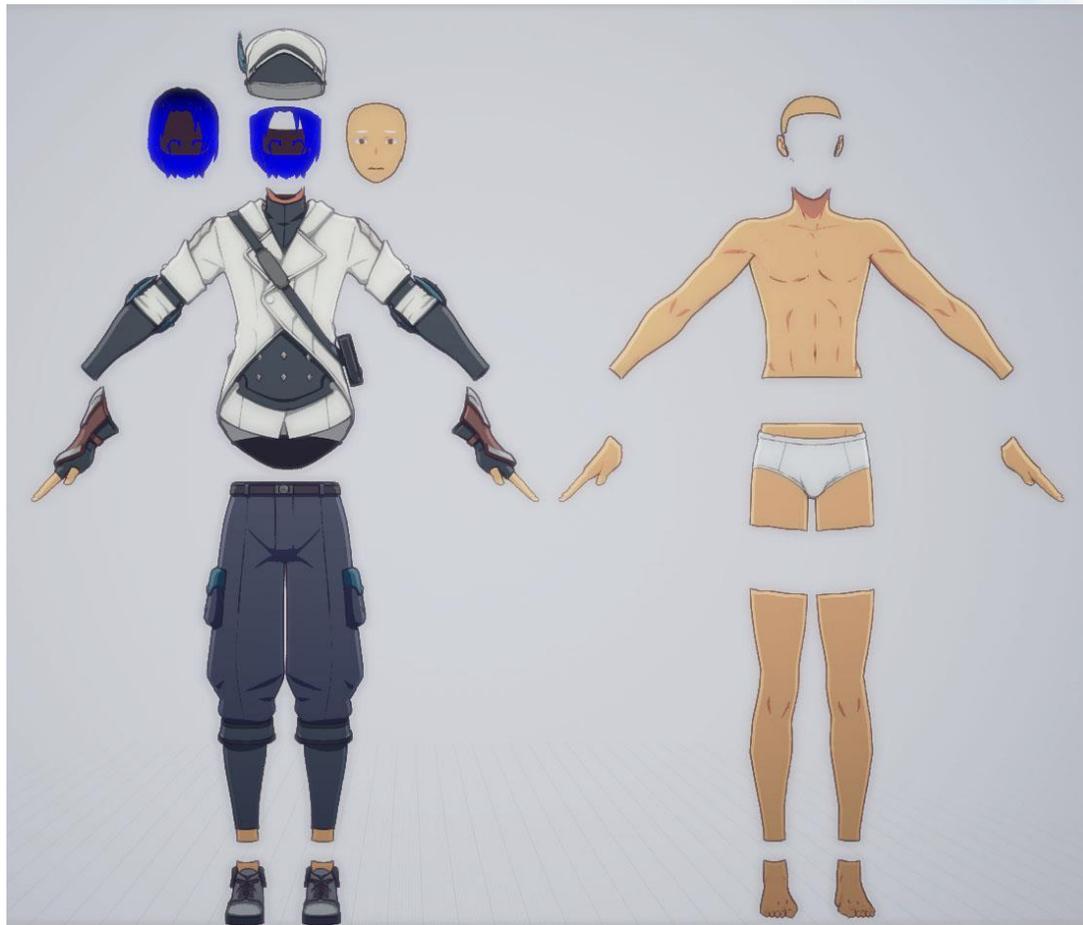
2Dテクスチャ(全身)



仕組み

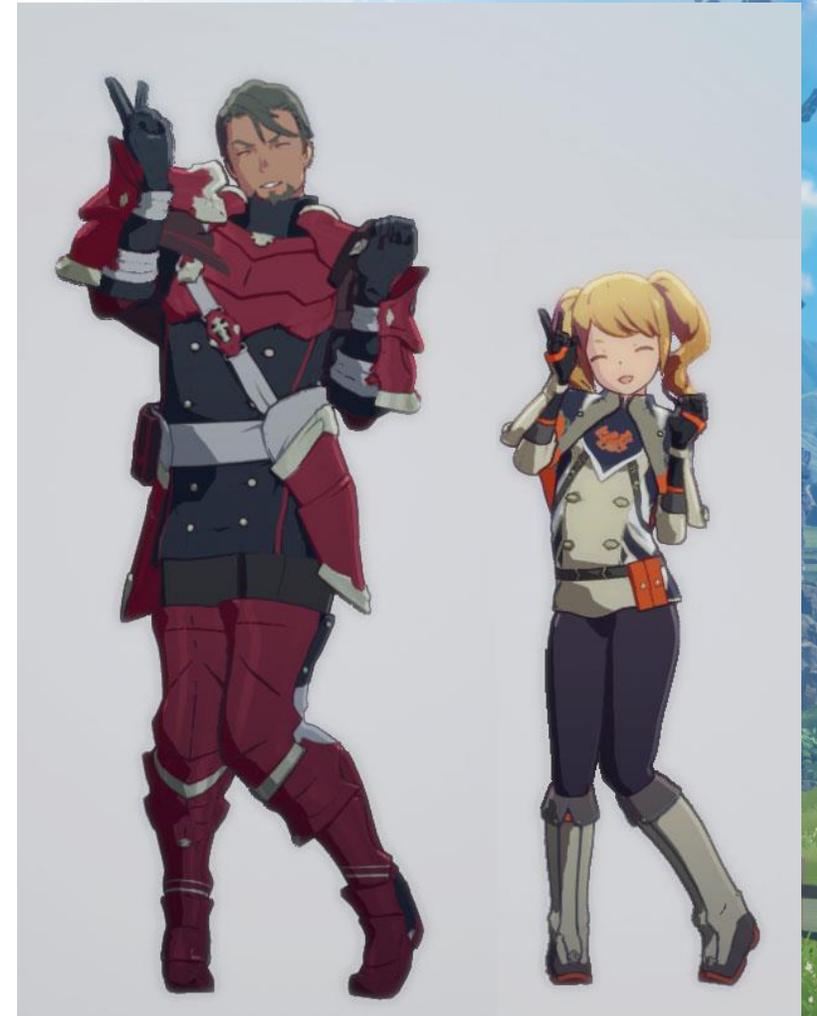
- ・パーツ構成

髪の毛、上半身、下半身、手、足、帽子、後頭部、顔パーツ群、
アンダーウェア (全9部位) + アクセサリー



仕組み

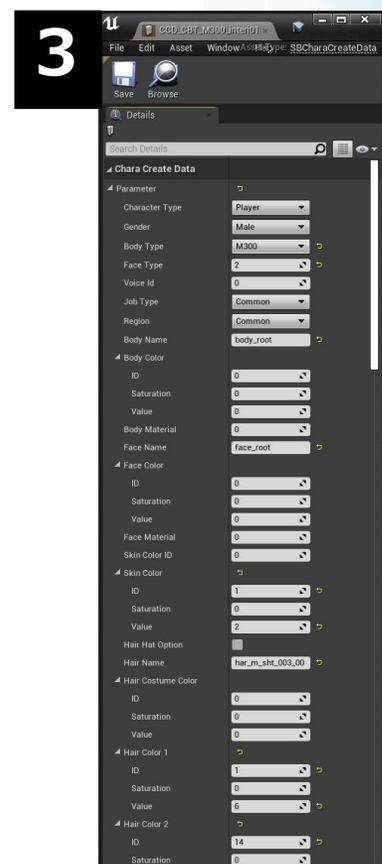
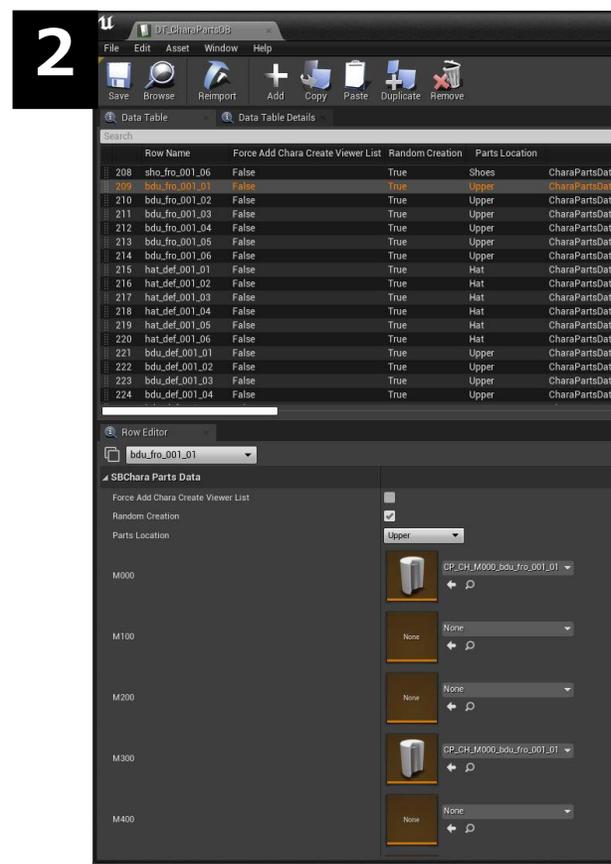
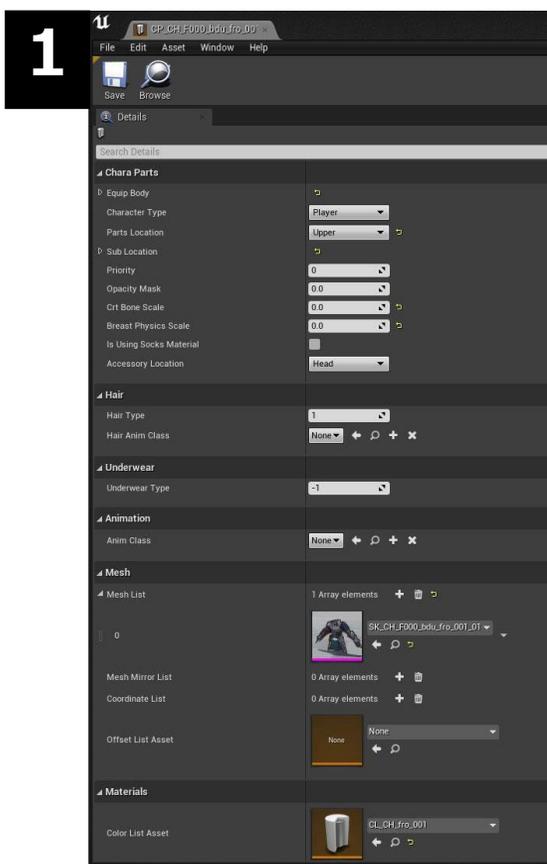
- ・ 起点のアセットについて





仕組み

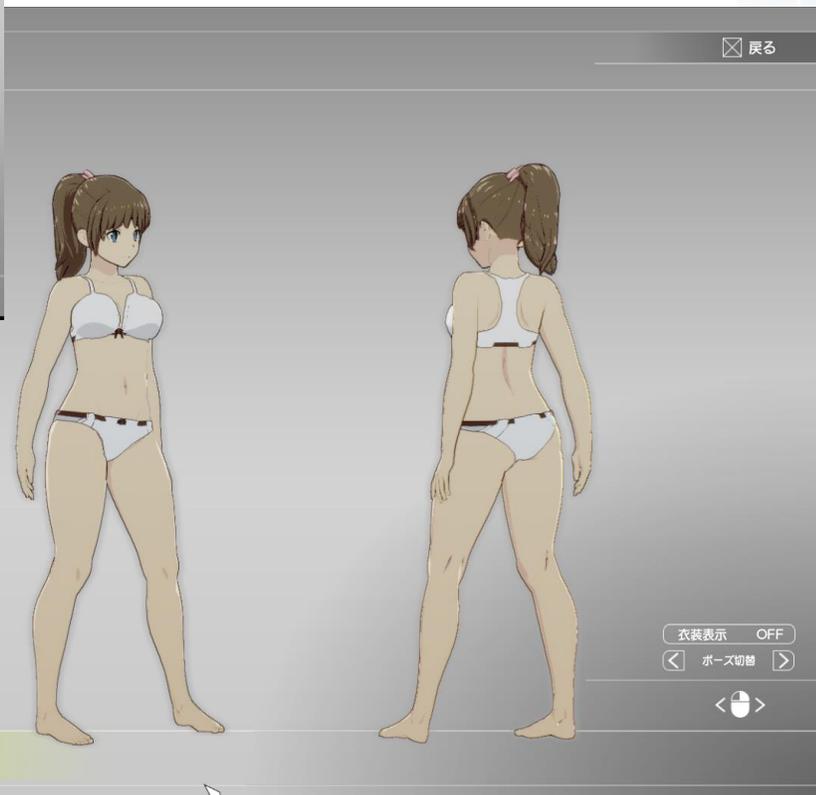
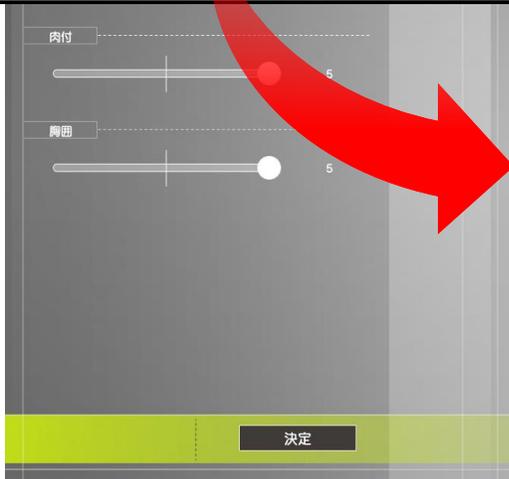
- キャラクリに必要なデータ
 1. キャラパーツデータ…個々のパーツの色々な情報
 2. データベース…IDとキャラパーツを結びつける表データ
 3. キャラクリエイトデータ…キャラクリで作られるキャラ情報





仕組み

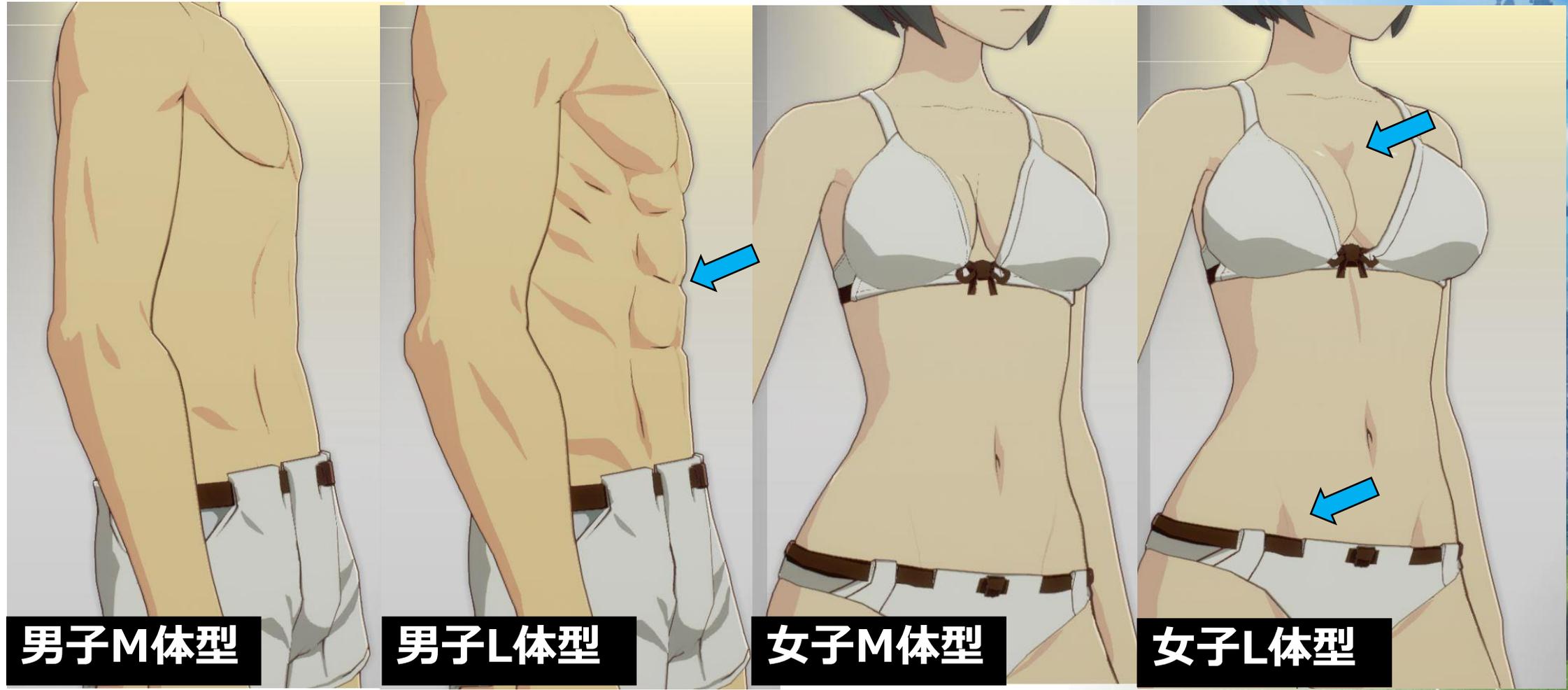
・体型変更



▲ 1	3 members
Group Name	肉付
Default Slider Value	0.5
▲ Parts List	38 Array elements
▲ 0	5 members
Scale Type	YZ Scale
Bone ID	Hip J Offset
Default Value	1.0
Min Value	0.92
Max Value	1.075
▲ 1	5 members
Scale Type	YZ Scale
Bone ID	L Up Leg J Offset
Default Value	0.95
Min Value	0.8
Max Value	1.1
▲ 2	5 members
Scale Type	YZ Scale
Bone ID	L Leg J Offset
Default Value	0.95
Min Value	0.8
Max Value	1.0
▲ 3	5 members
Scale Type	XYZ Scale
Bone ID	L Foot J
Default Value	0.95
Min Value	0.949
Max Value	0.951
▲ 4	5 members

仕組み

・Lボディの違い



男子M体型

男子L体型

女子M体型

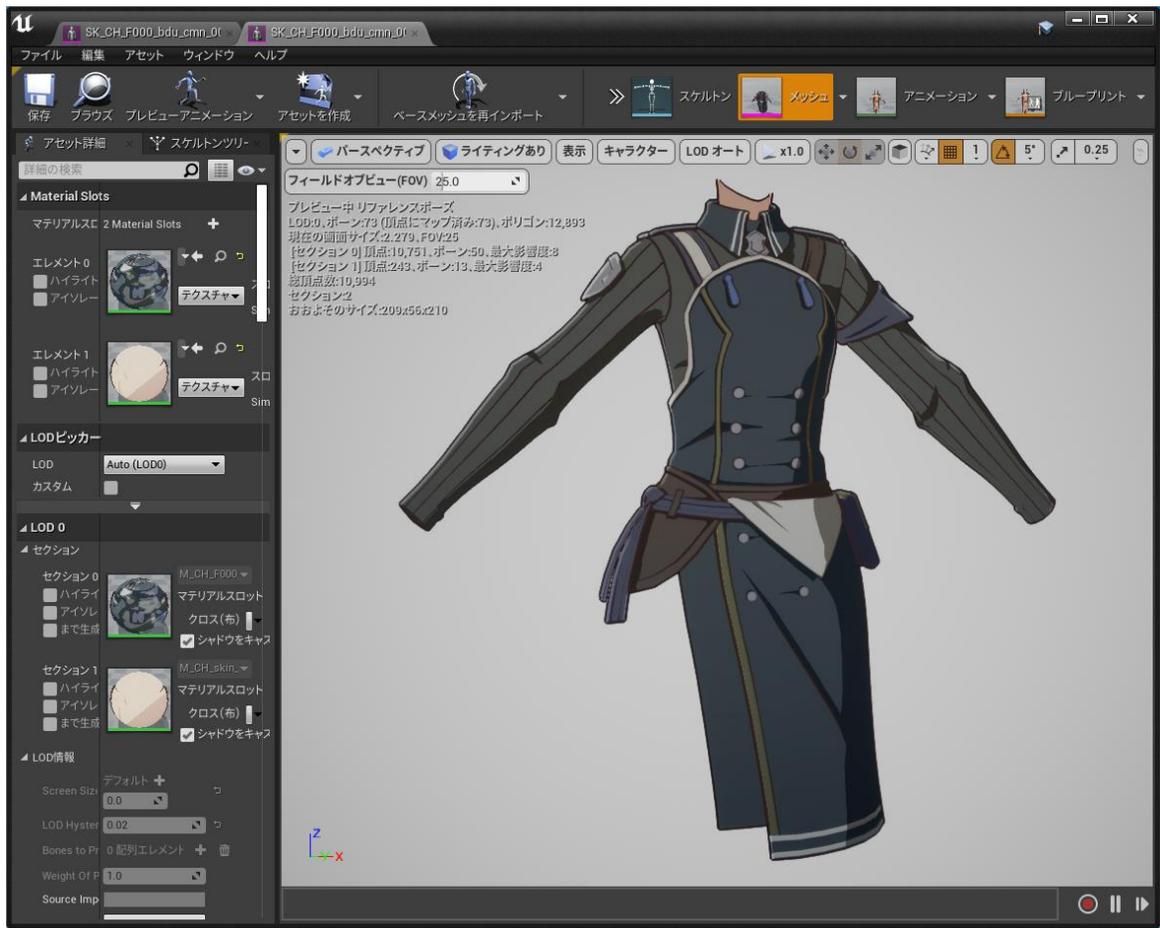
女子L体型

衣装

・ 着合わせ

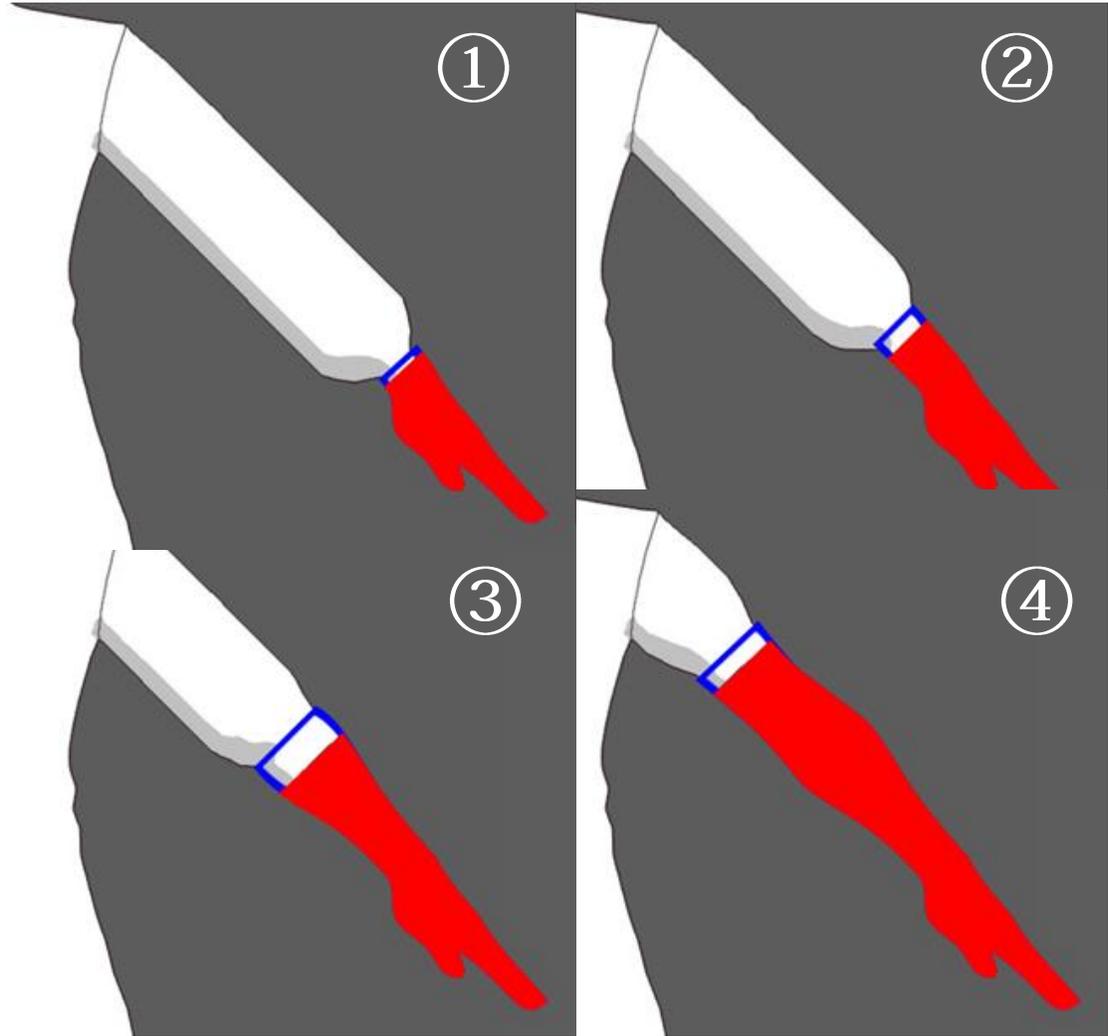
To save the world that is going to destroy, fight beyond the spacetime. Cooperate with friends, beat the mighty enemies, change the history. That is your mission. Now, let's sun out on a vast land, heading for a hopeful future!

blue-protocol.com
©GANDAI NAMCO Online Inc.
©GANDAI NAMCO Studios Inc.



衣装

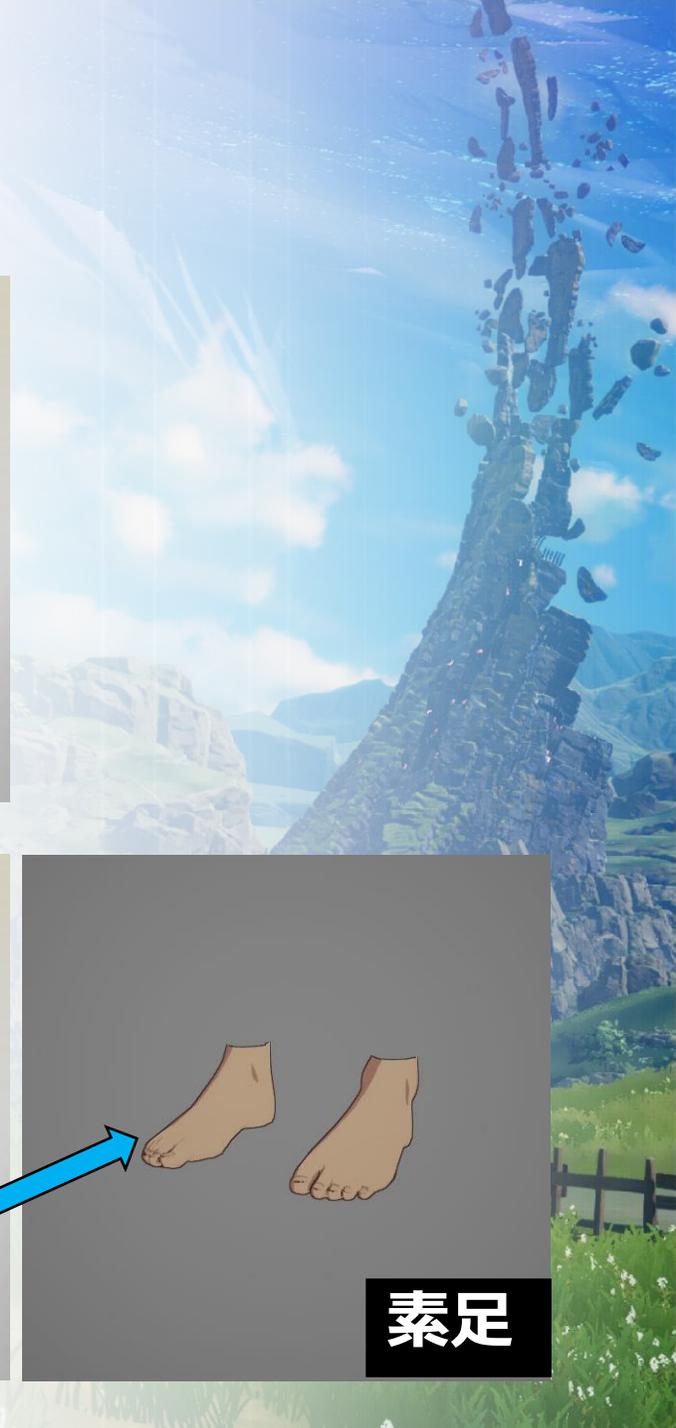
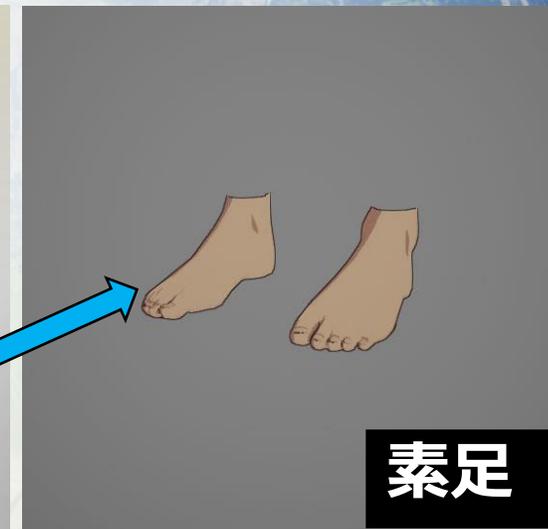
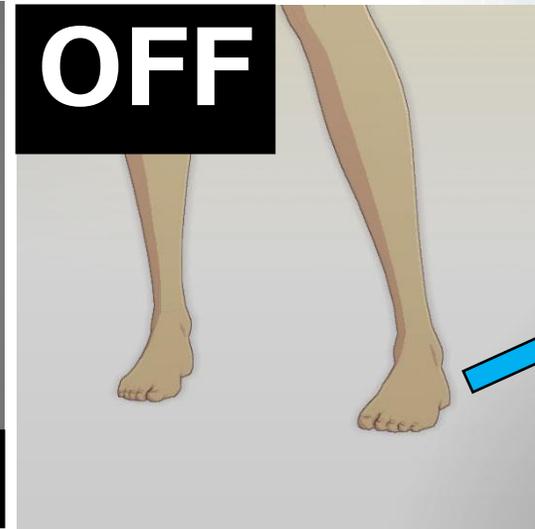
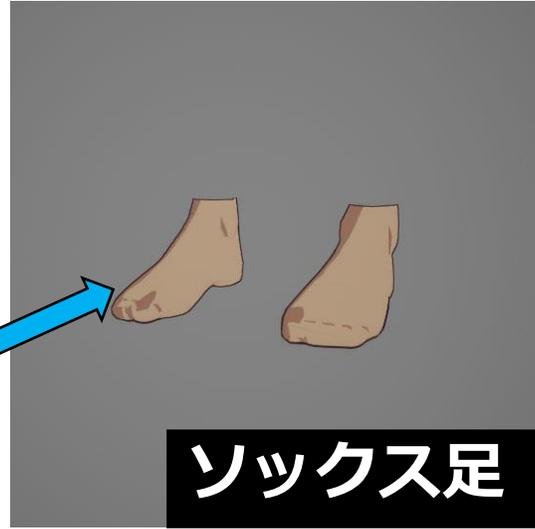
・ 着合わせ





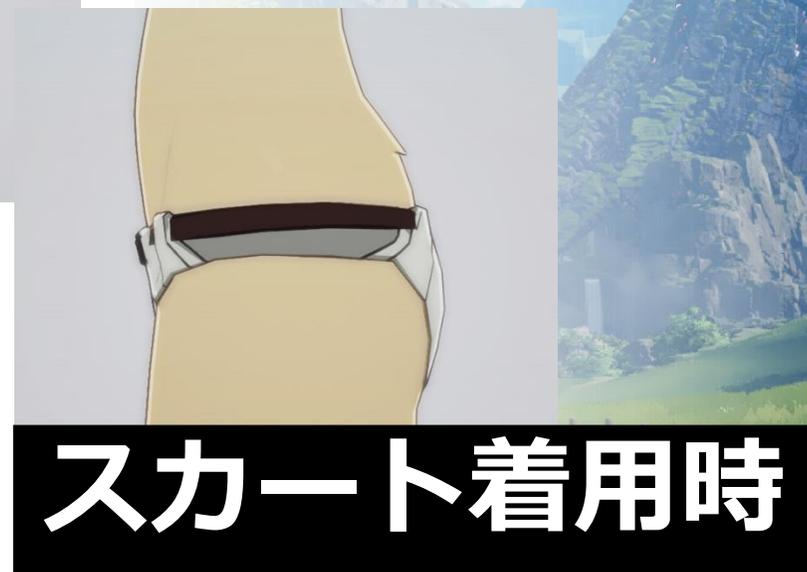
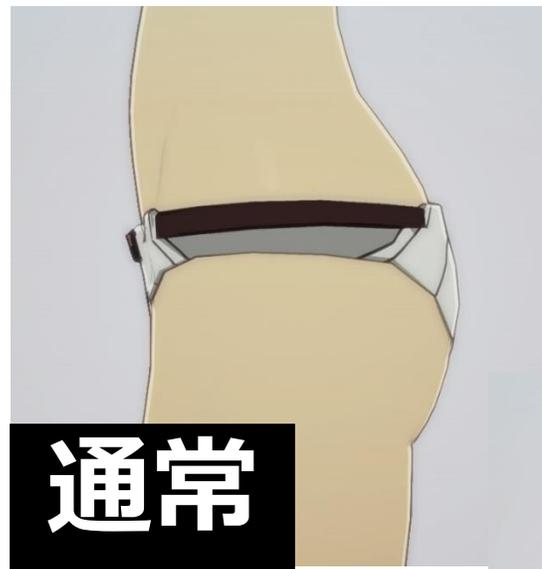
衣装

・ソックス足



衣装

- ・アンダーウェア



髪

- ・ 髪の毛はキャラの命！

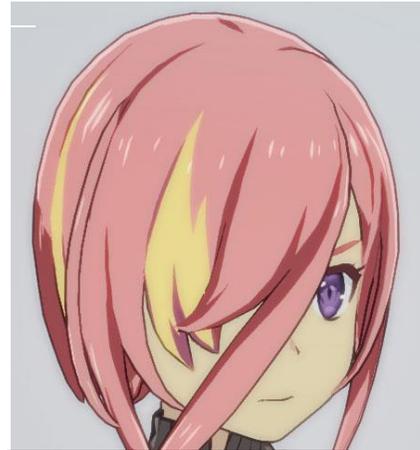


髪の色

・グラデとメッシュ



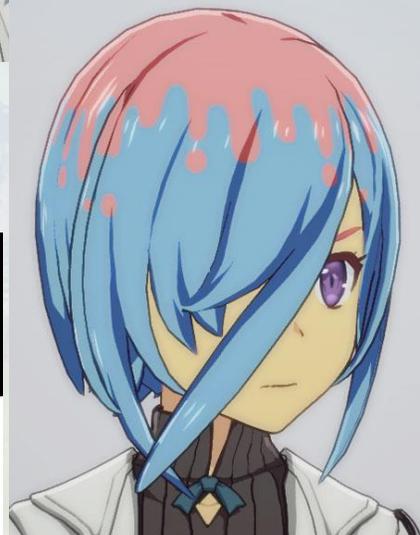
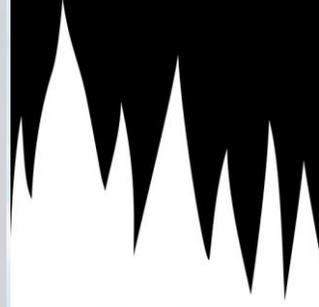
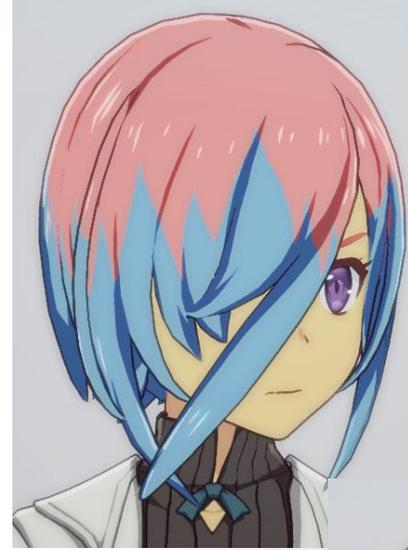
グラデーション



メッシュ



グラデ+メッシュ



グラデのバリエーション



髪揺れ骨

- 少ない骨でKawaiiPhysics

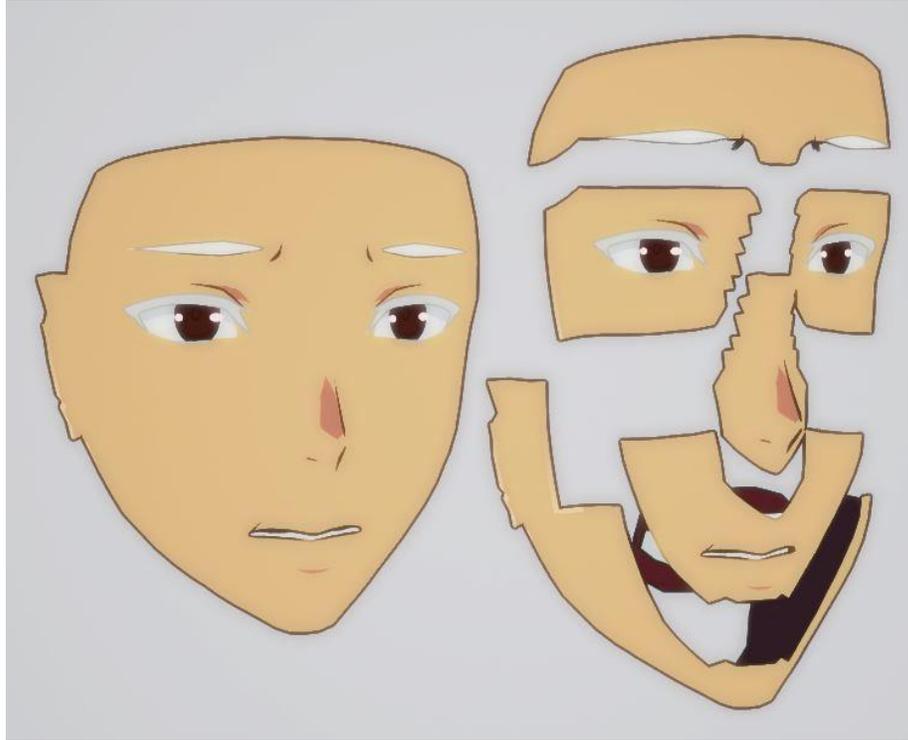
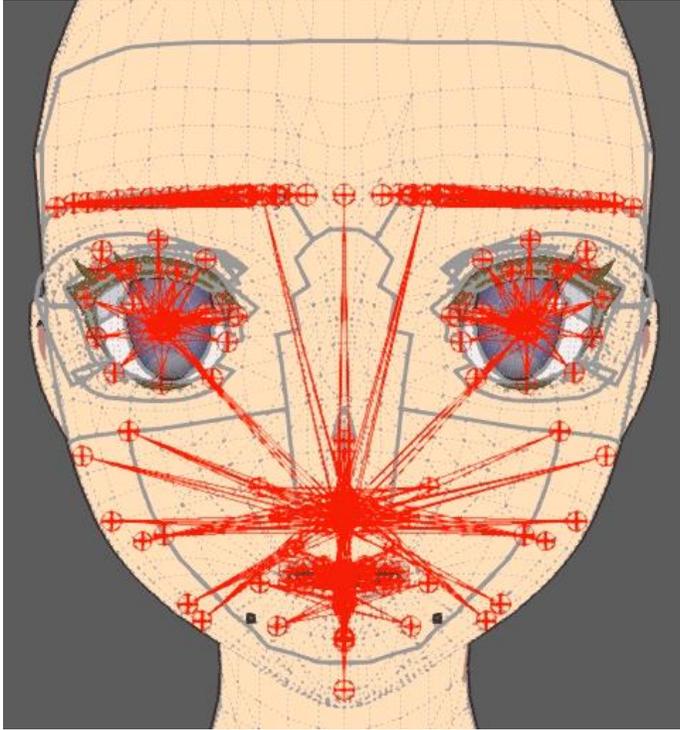
The screenshot displays the Unreal Engine 4 interface for implementing hair physics. On the left, a 3D perspective view shows a character's head with a skeletal rig for hair. The rig consists of a central root bone and several branching bones, each ending in a KawaiiPhysics component. The right panel shows the details of a KawaiiPhysics component, with the following settings:

- Debug:** Enable Debug Draw Bone, Enable Debug Bone Length, Enable Debug Draw Angle, Enable Debug Draw Sphere, Enable Debug Draw Capsule, Enable Debug Draw Plane (all checked).
- Target Framerate:** Target Framerate: 30.
- Physics Settings:** Damping: 0.6, World Damping Location: 0.8, World Damping Rotation: 0.8, Stiffness: 0.4, Radius: 1.0, Limit Angle: 30.0.
- Damping Curve Data:** A graph showing a linear relationship between position and damping.
- World Damping Location:** A graph showing a linear relationship between position and world damping.

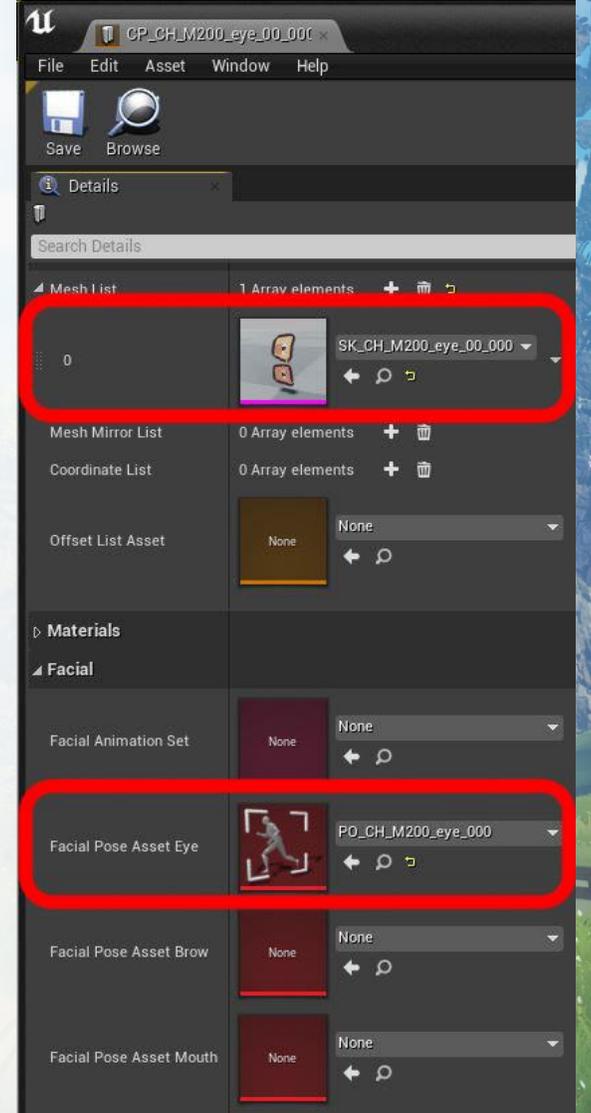
The center panel shows an animation graph with multiple KawaiiPhysics components connected to a hair layer. The graph includes a 'HairLayer' component and several 'KawaiiPhysics' components, each with its own set of settings. The 'KawaiiPhysics' components are connected to the 'HairLayer' component via 'Component Pose' and 'Component Space Selection' nodes.

顔

- ・ 顔パーツは差し替え方式、骨で動かしています



- ・ 眉 : 37本
- ・ 目 : 42本
- ・ 口 : 53本
- ・ 輪郭/鼻 : 18本



表情

- ・アバターの基本表情25種類
 1表情につき、眉×1パターン、目×4パターン（開き目・中目1・閉じ目・中目2）、口×6パターン（基本・閉じ口・中口・開き口+a）

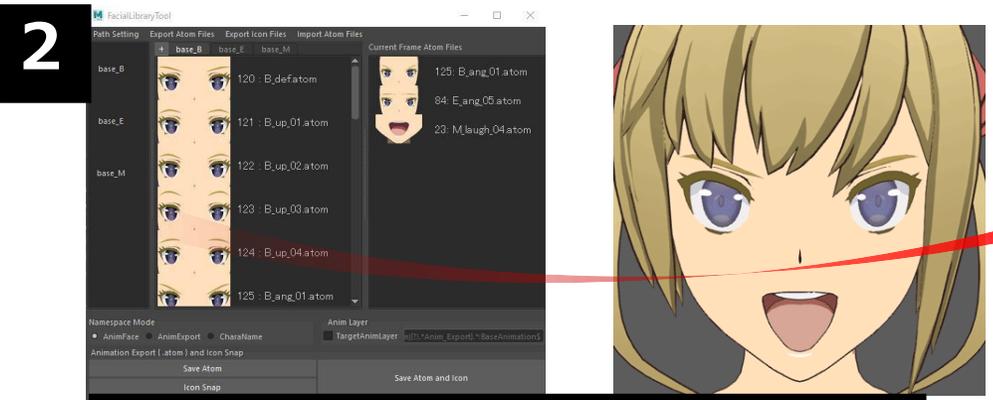
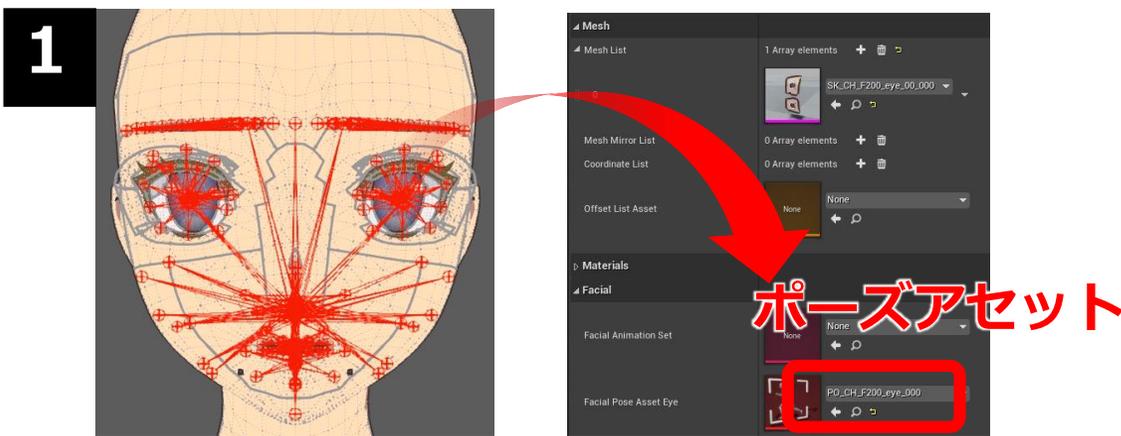




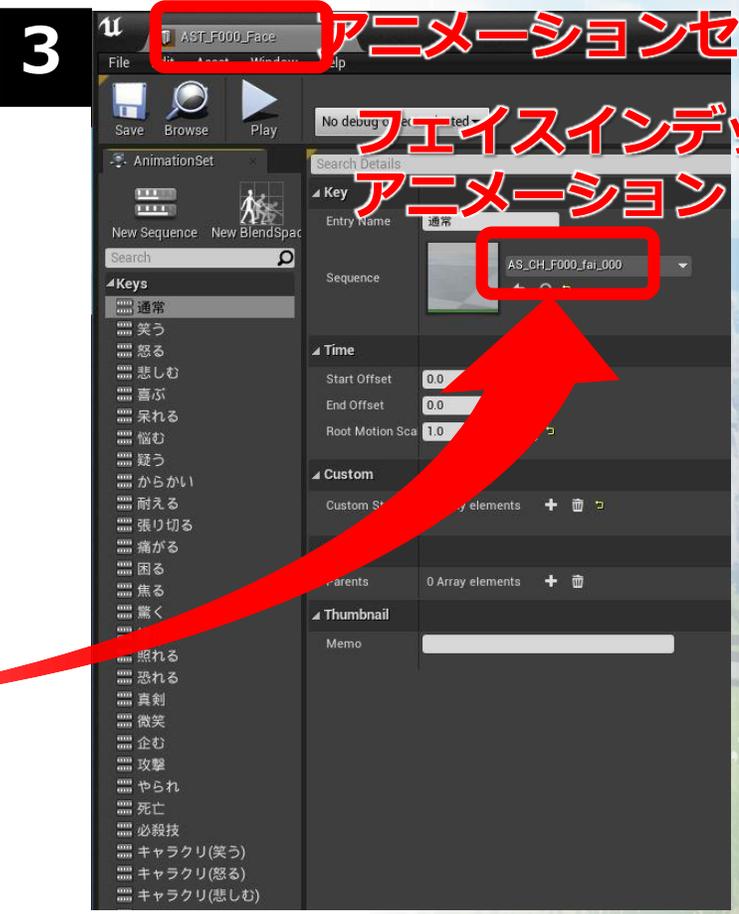
表情

・フェイシャルの仕組み

1. ポーズアセット…眉・目・口のアニメーション
2. フェイスインデックスアニメーション…各表情のフレーム数（社内ツールで指定）
3. アニメーションセット…各表情毎にフェイスインデックスアニメーションを指定



フェイシャルライブラリツール

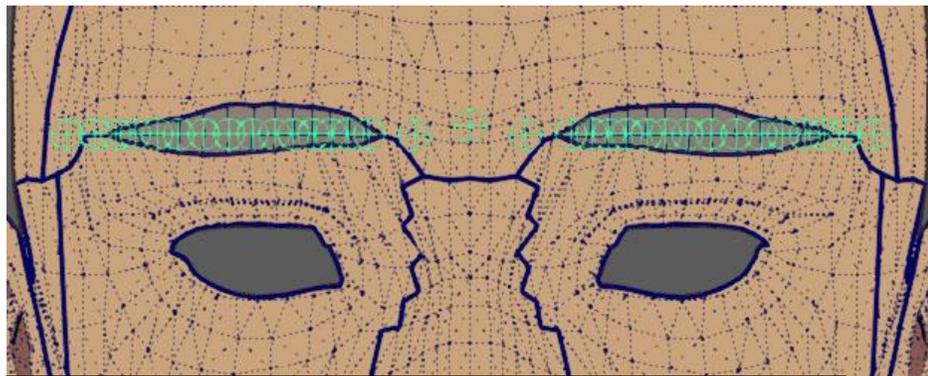


アニメーションセット
フェイスインデックス
アニメーション

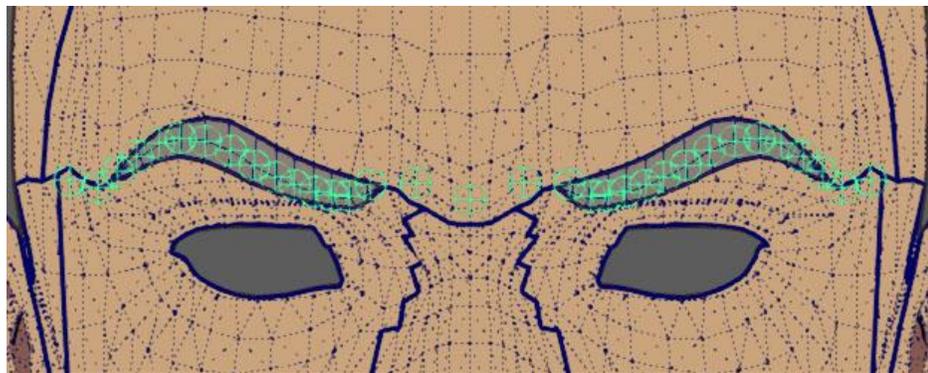


眉

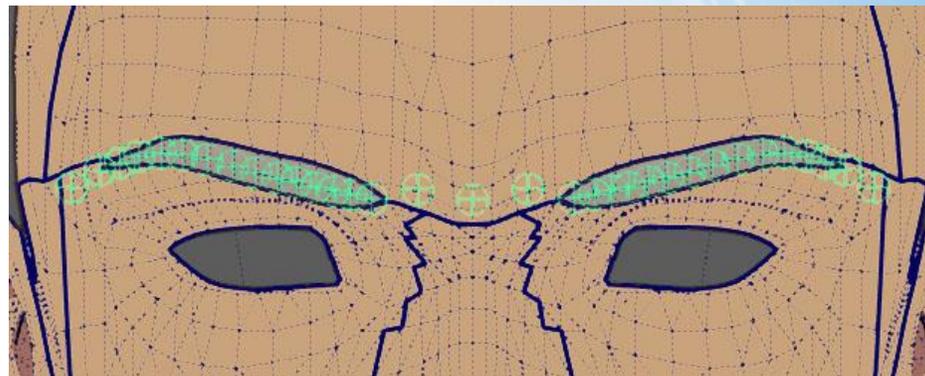
- ・パーツごとに専用アニメーションを持たせ、デフォルト形状を作成



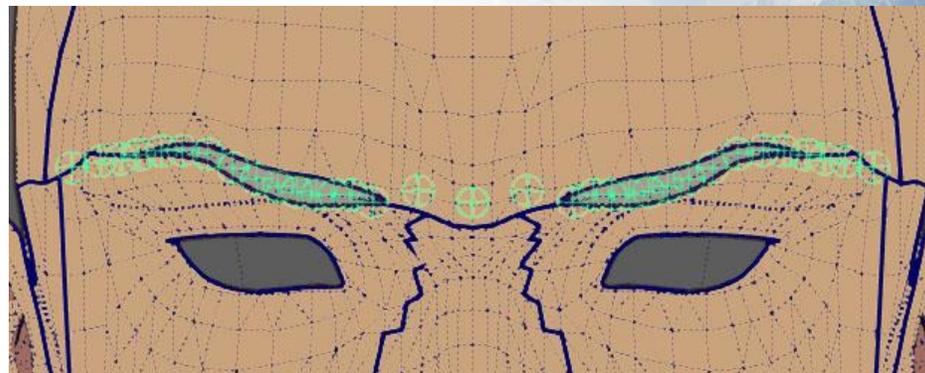
バインド時の眉形状はストレート



デフォルト形状はデザインにより様々



眉バリエーション1



眉バリエーション2

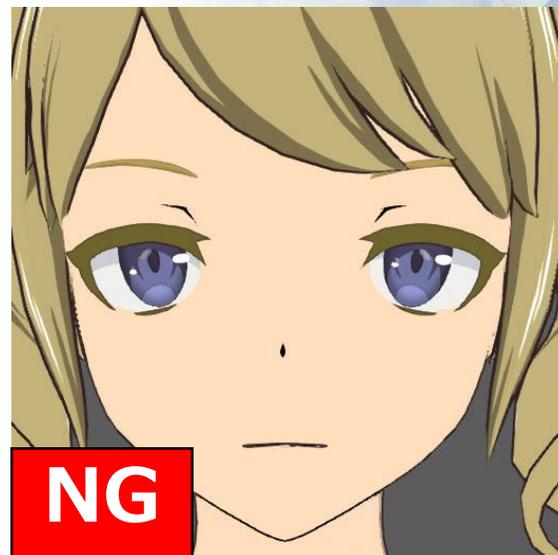
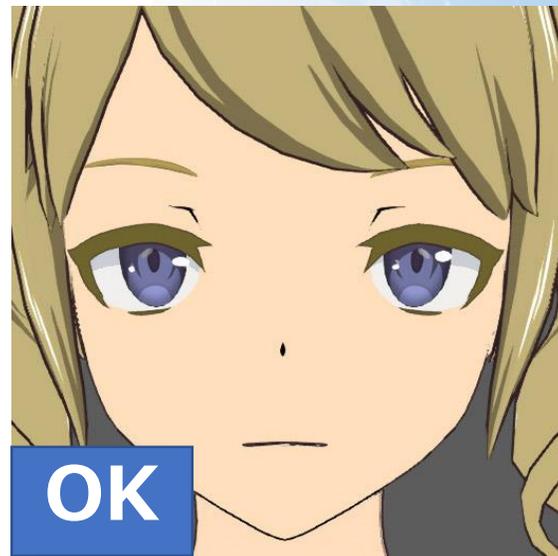
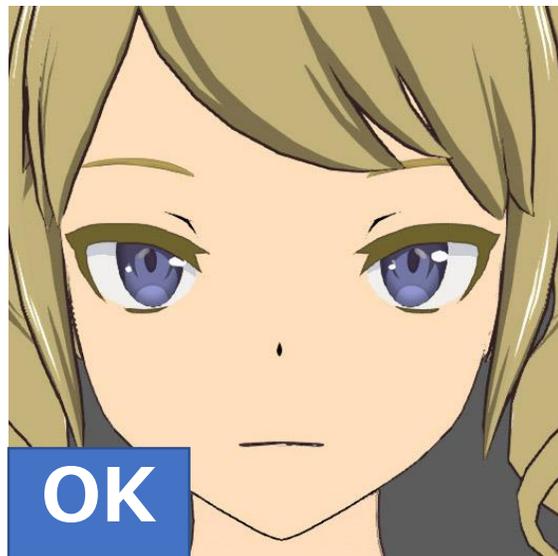
角度調整

- ・目と眉は好みの角度に調整できます



目の角度調整

- ・ 自然な見た目になるようにアニメーション調整



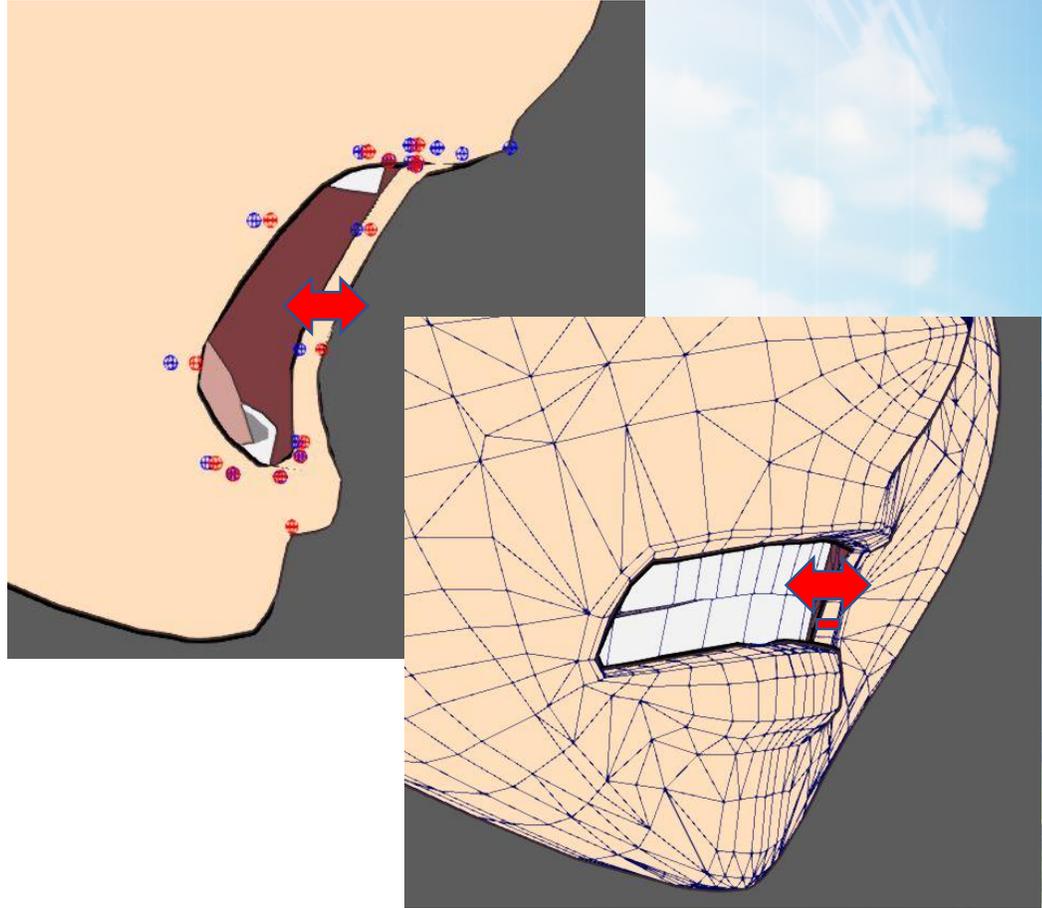
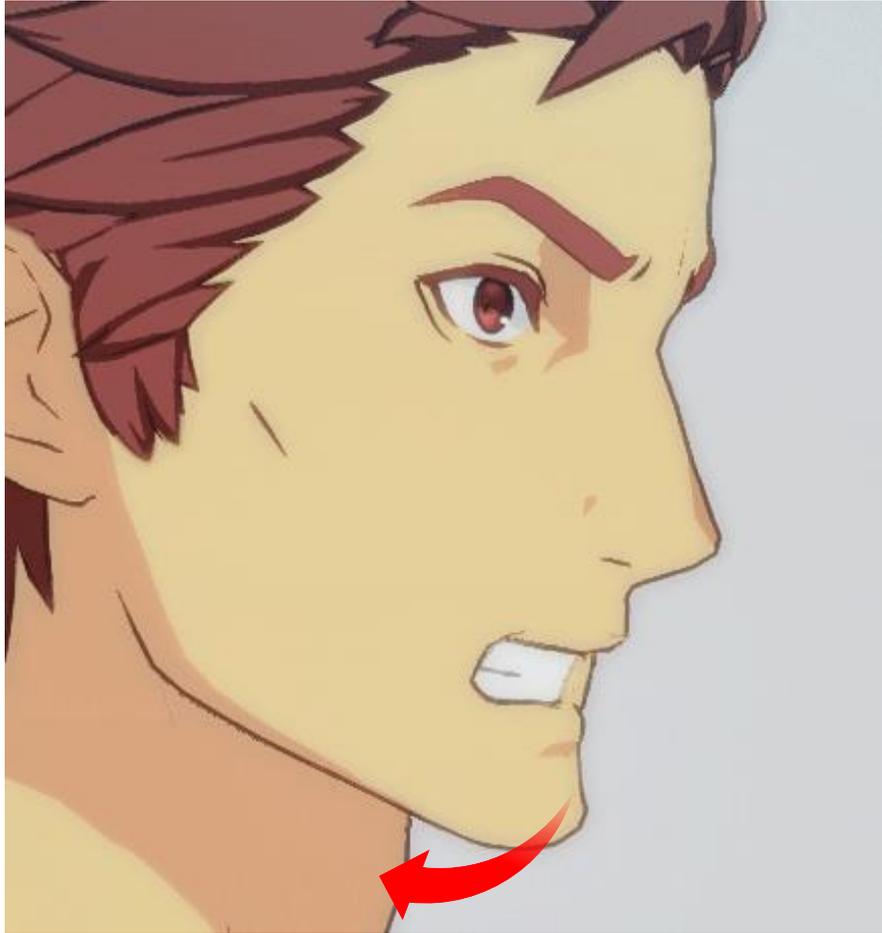
目

- まつげはテクスチャに変更、メイクのための隠しエッジ



輪郭、口、鼻

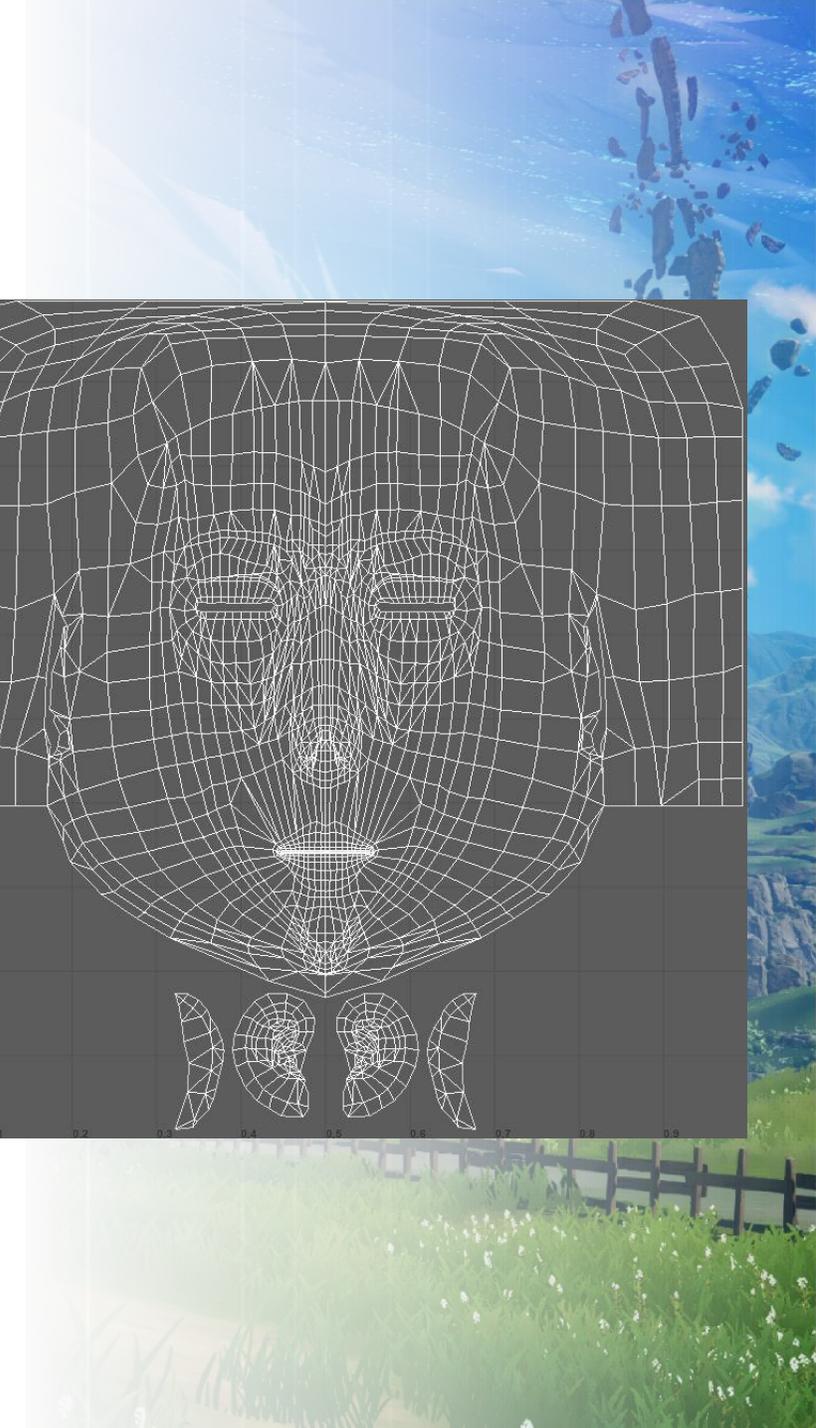
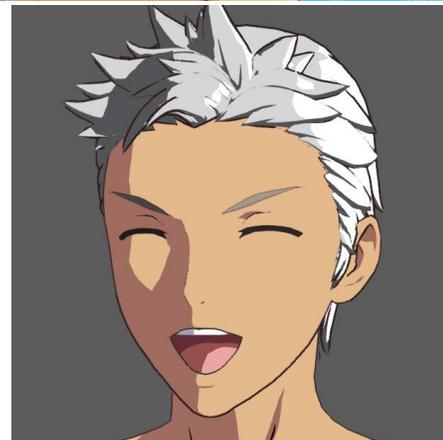
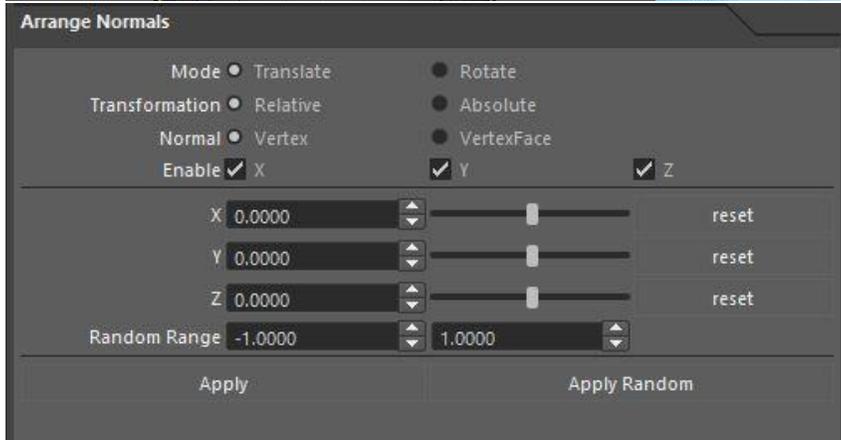
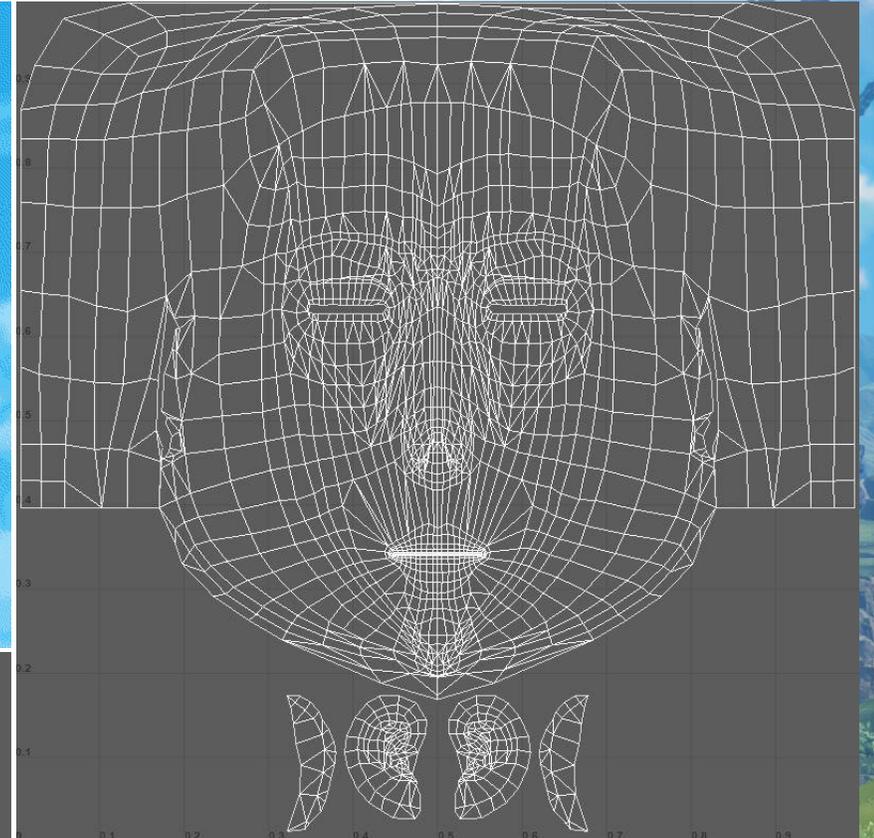
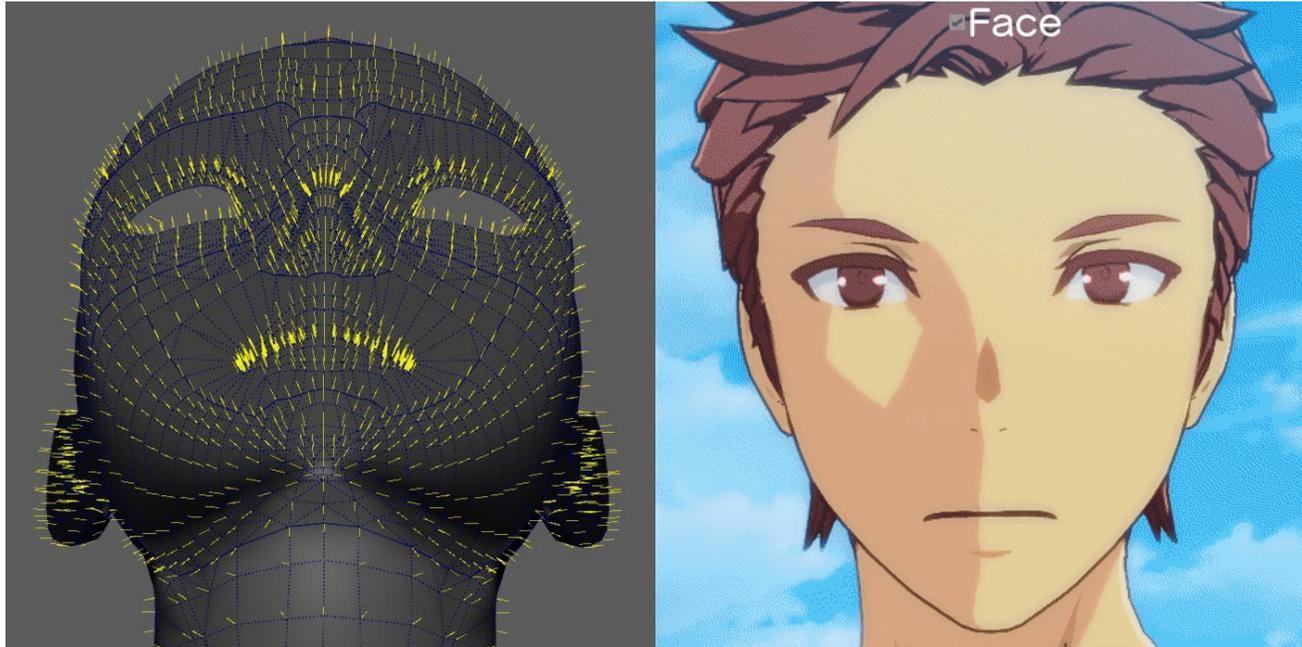
- 骨の移動で動かしている、口のフチは2重になっています





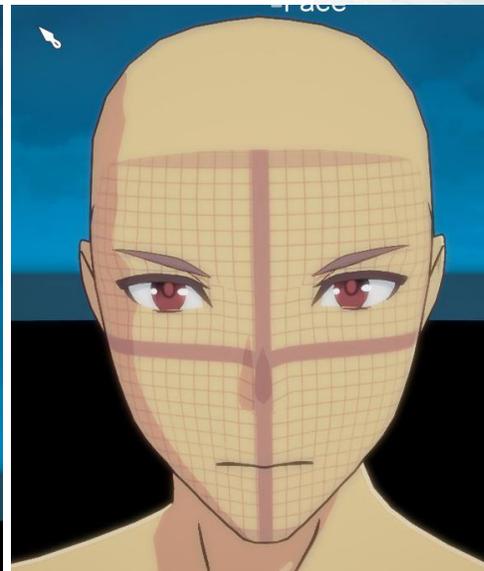
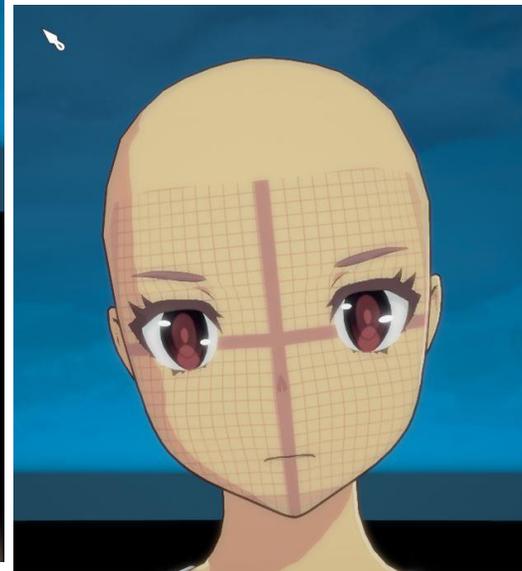
法線とUV

- ・ アニメでよく見る影の形に近づける



メイク

- ・4枚のマスキで表現。位置調整は性別年代ごとにオフセット



DT MakeupDB

Opens the Marketplace Window Help Row Type: SBMak

Save Browse Reimport Add Copy Paste Duplicate

Data Table Data Table Details

Search

	Row Name	Type	Index	Gender	Can Reverse	
1	m_cnd_def	Condition	0	Male	False	{ "ID": 0, "Se
4	m_cnd_002	Condition	3	Male	False	{ "ID": 0, "Se
5	m_cnd_003	Condition	4	Male	False	{ "ID": 0, "Se

Row Editor

m_cnd_003

SBMakeup Data

Type: Condition

Index: 4

Gender: Male

Can Reverse:

Default Color

Mask Texture: C000_fm_k_cnd_003_MASK

Projection:

Projection Scale: 0.07

Projection Offset

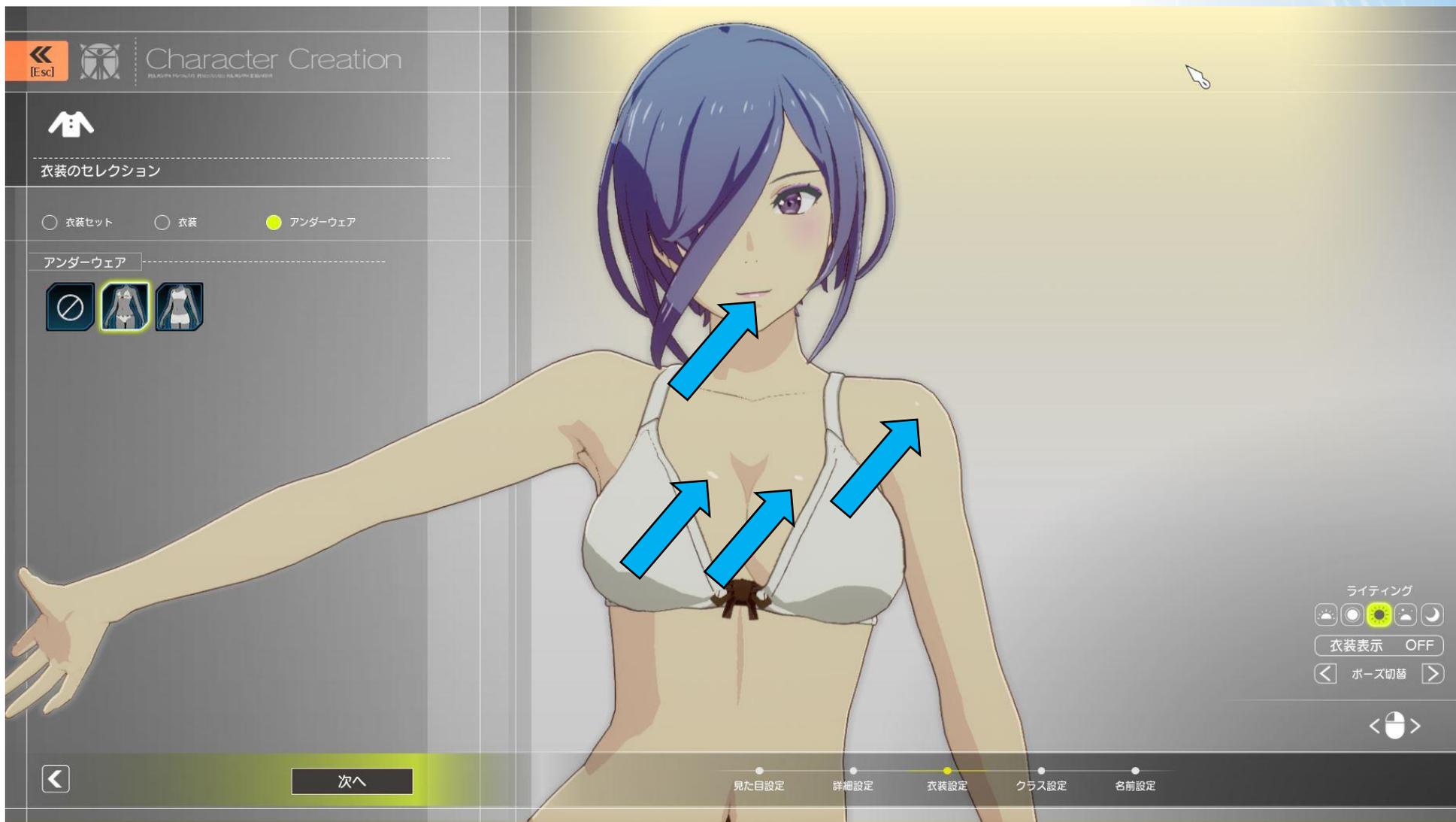
X: 0.0

Y: -1.5



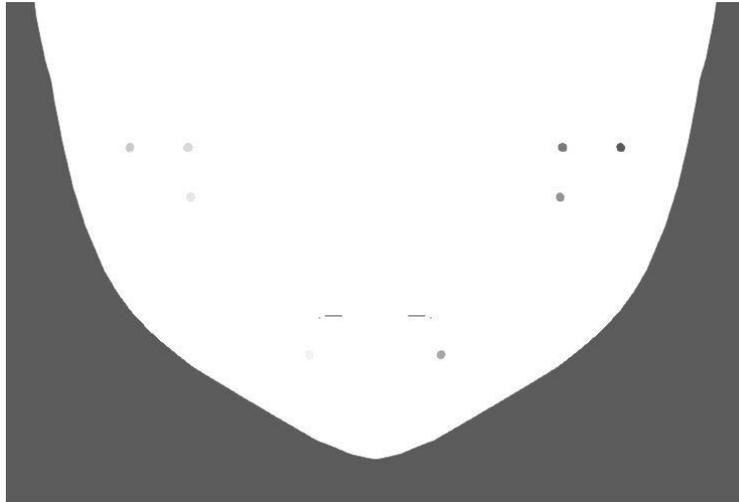
ハイライト

- ・ 大事なところにはハイライト



ホクロ

- ・小さくなくても見えるように & 位置もこだわりました



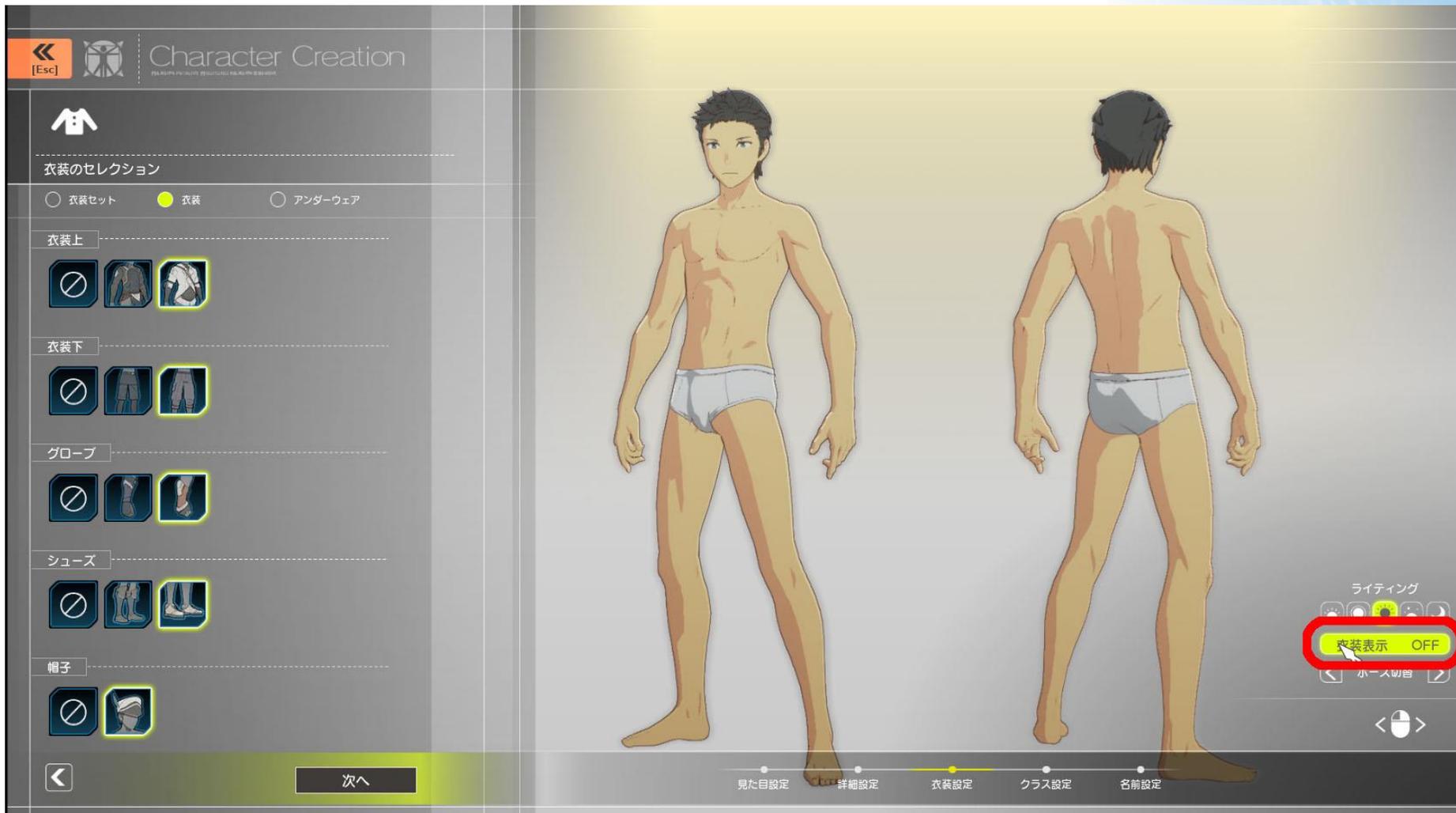
頬染め、青ざめ

- ・メイクと同様に投影で感情表現



アクシデント

- ・ 衣装非表示について



リアル系キャラクリとの違い

- ・年齢層の広い男女を同じシステムで作るために



女子(年下)
目以外のデザイン幅が狭い



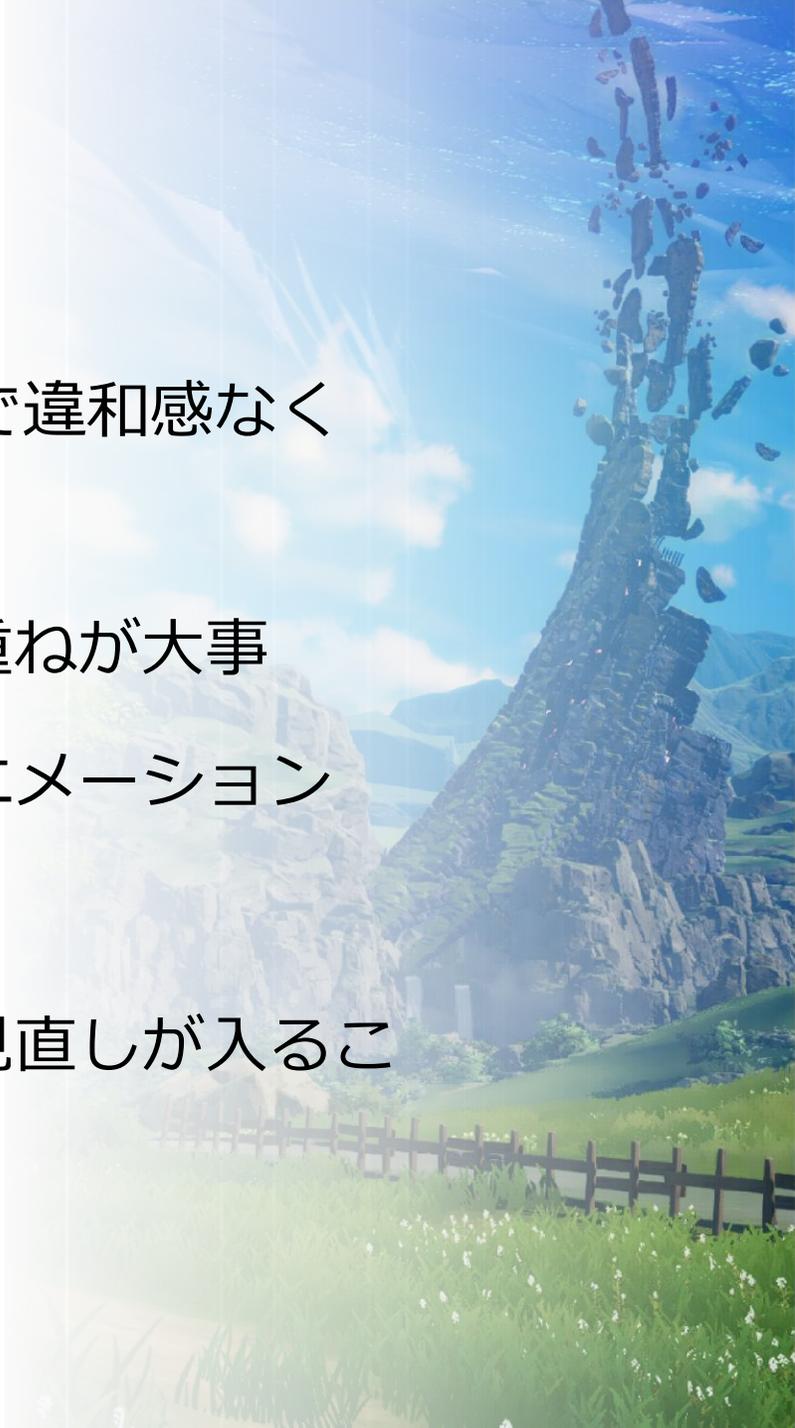
男子(おっさん)
輪郭・眉・鼻のデザイン幅が広い



まとめ(前半)

- 作画崩壊せずに、幅広い年齢層の男女を同じUIで違和感なく簡単に作れるよう、機能を取捨選択
- アニメ的な表現のため、細かいこだわりの積み重ねが大事
- 表情豊かなアバターにするため、骨ベースでアニメーション

※開発中のプロジェクトであり、今後内部実装の見直しが入ることが予想されます



自己紹介



金田 直隆

株式会社バンダイナムコスタジオ 第3スタジオ

2004年ナムコ（当時）入社

「ソウルキャリバーIV」「AceCombatアサルトホライゾン」「大乱闘スマッシュブラザーズ for 3DS WiiU」

など、アーティスト向け開発環境周りの構築等を担当。

「BLUE PROTOCOL」ではインゲームのキャラクリエイションと、アニメーション関係のプログラムの担当をしています。





アジェンダ(後半)

- 処理負荷対策
- アセット管理&作業環境
 - アセットデータ管理
 - スクリプト環境
 - キャラクリビューワー
- 体型変更
- 物理揺れ
- 汎用化



前置き

UnrealEngin4 (UE4) 使用

描画部分以外を紹介します。

描画部分は、CEDEC2021

[BLUE PROTOCOLにおけるアニメ表現技法について](#) ～実装編～





処理負荷対策

BLUE PROTOCOL™

To save the world that is going to destroy, fight beyond the spacetime. Cooperate with friends, beat the mighty enemies, change the history. That is your mission. Now, let's run out! On a vast land, heading for a hopeful future!

blue-protocol.com
©GANDAI NAMCO Online Inc.
©GANDAI NAMCO Studios Inc.



処理負荷対策

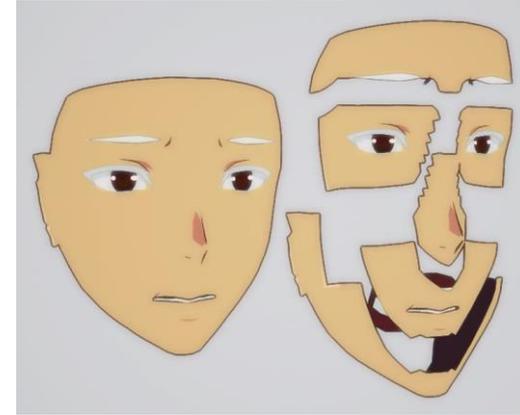
Closed βテスト時の、最終日の町の様子



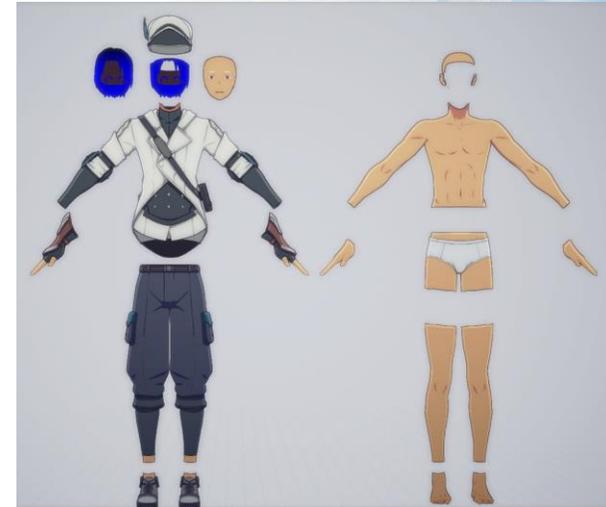
パーツの構成

- Face
 - Eye
 - Mouth
 - Nose
 - Line
 - Eyebrow
- Body
 - Upper
 - Lower
 - Glove
 - Shoes
 - Hat
 - UpperUnderwear
 - Underwear
 - BackOfHead
- AccessoryTop
- AccessoryMid
-
-

顔



体



処理負荷対策

ゲームプレイ中

- ・メッシュのマージ (非同期)



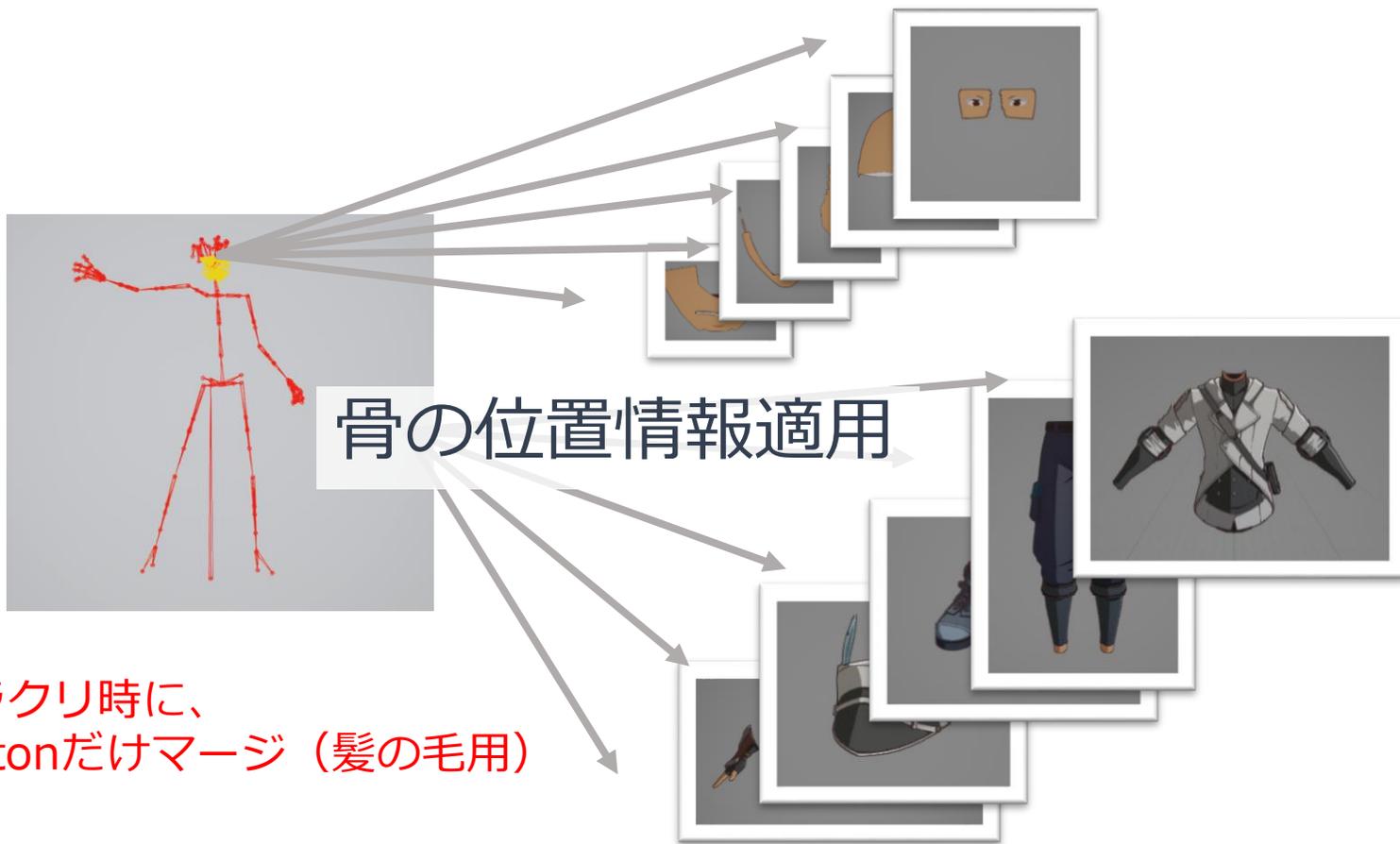
アニメーション

キャラクリ時に、
メッシュのマージ

処理負荷対策

キャラクリ中

- skeletonだけマージ（同期）（UE4 MasterPoseComponentの機能）



キャラクリ時に、
Skeletonだけマージ（髪の毛用）



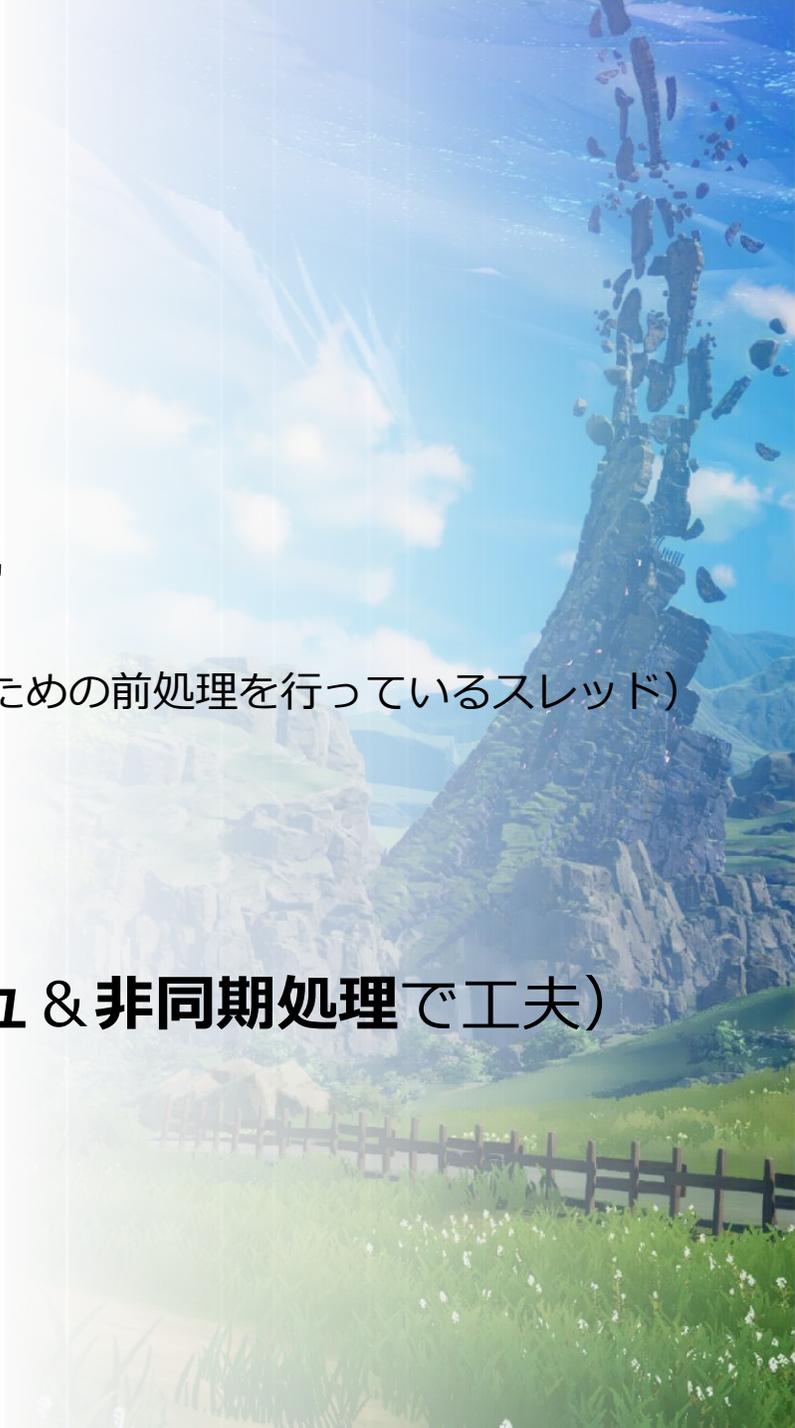
非同期メッシュマージ

メリット

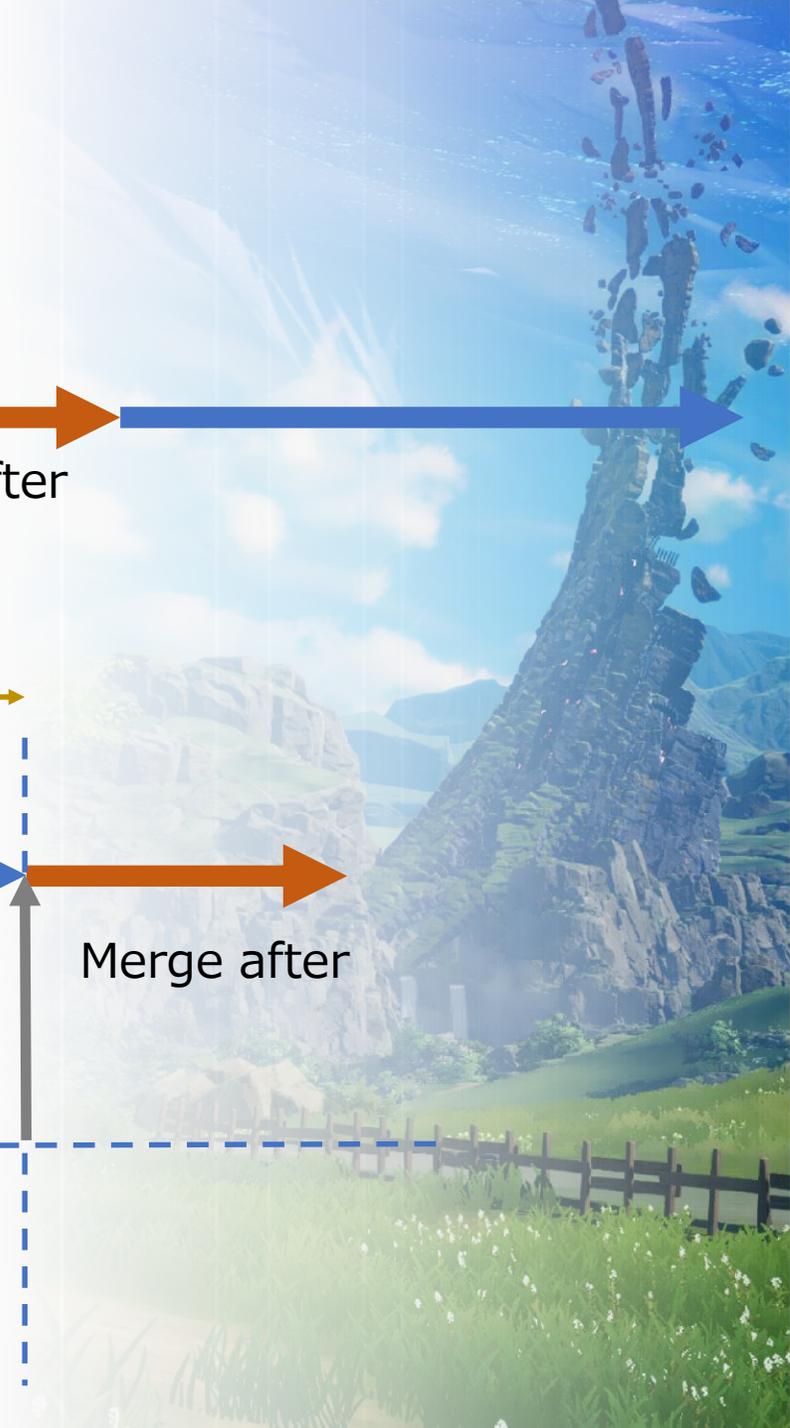
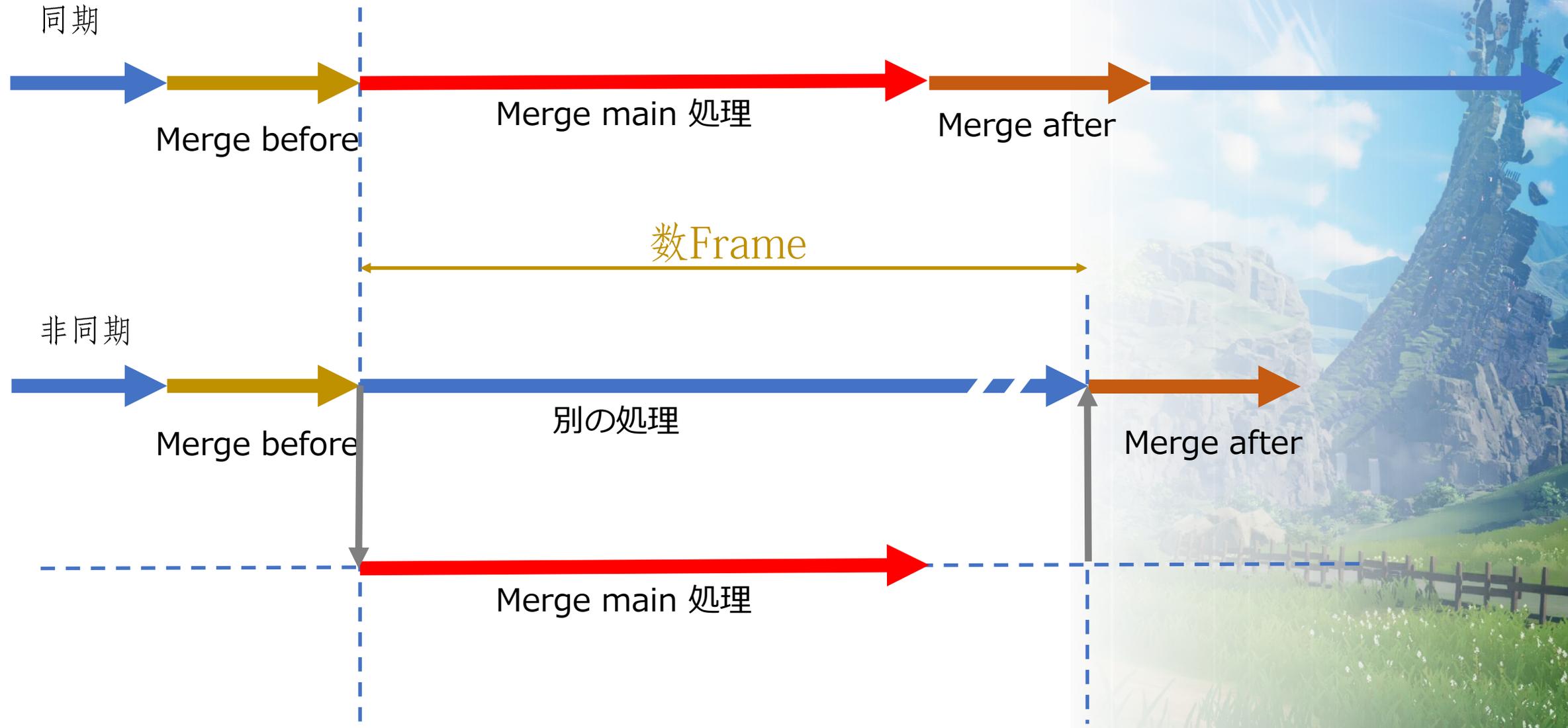
- オブジェクト数減
 - マテリアルセクション統合によるドローコールの低減
 - UE4のレンダースレッドの処理負荷低減
(レンダースレッドとは、GPUに描画やスキン処理をもっていくための前処理を行っているスレッド)
 - オブジェクト（コンポーネント）の移動処理の低減

デメリット

- マージ処理自体の負荷（15～25ms）（**キャッシュ & 非同期処理**で工夫）
- 使用メモリの増加
- キャラクリ処理完了が遅れる



同期 非同期 補足説明

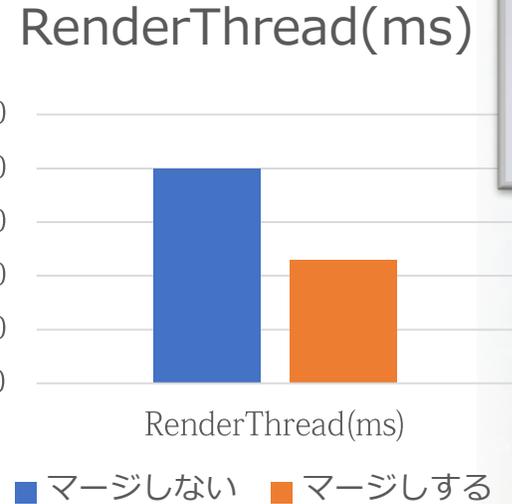
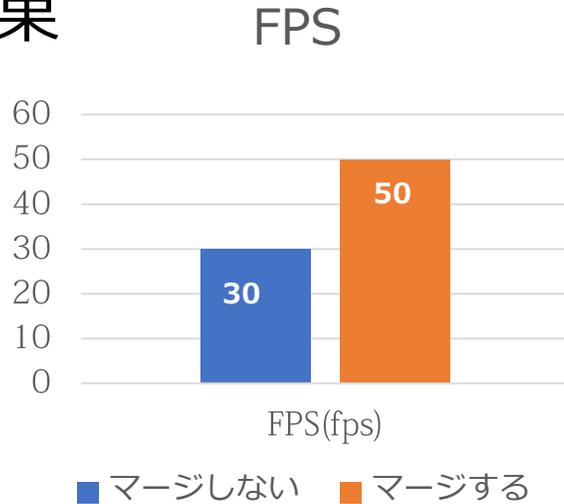


マージしたことによる処理負荷測定

測定環境

- DevelopBuild & パッケージ
- 何も無いシーン & 100体のプレイヤー見た目のNPCを登場
 - オブジェクト数 マージしない 約1350個
 - オブジェクト数 マージする 約 200個

測定結果

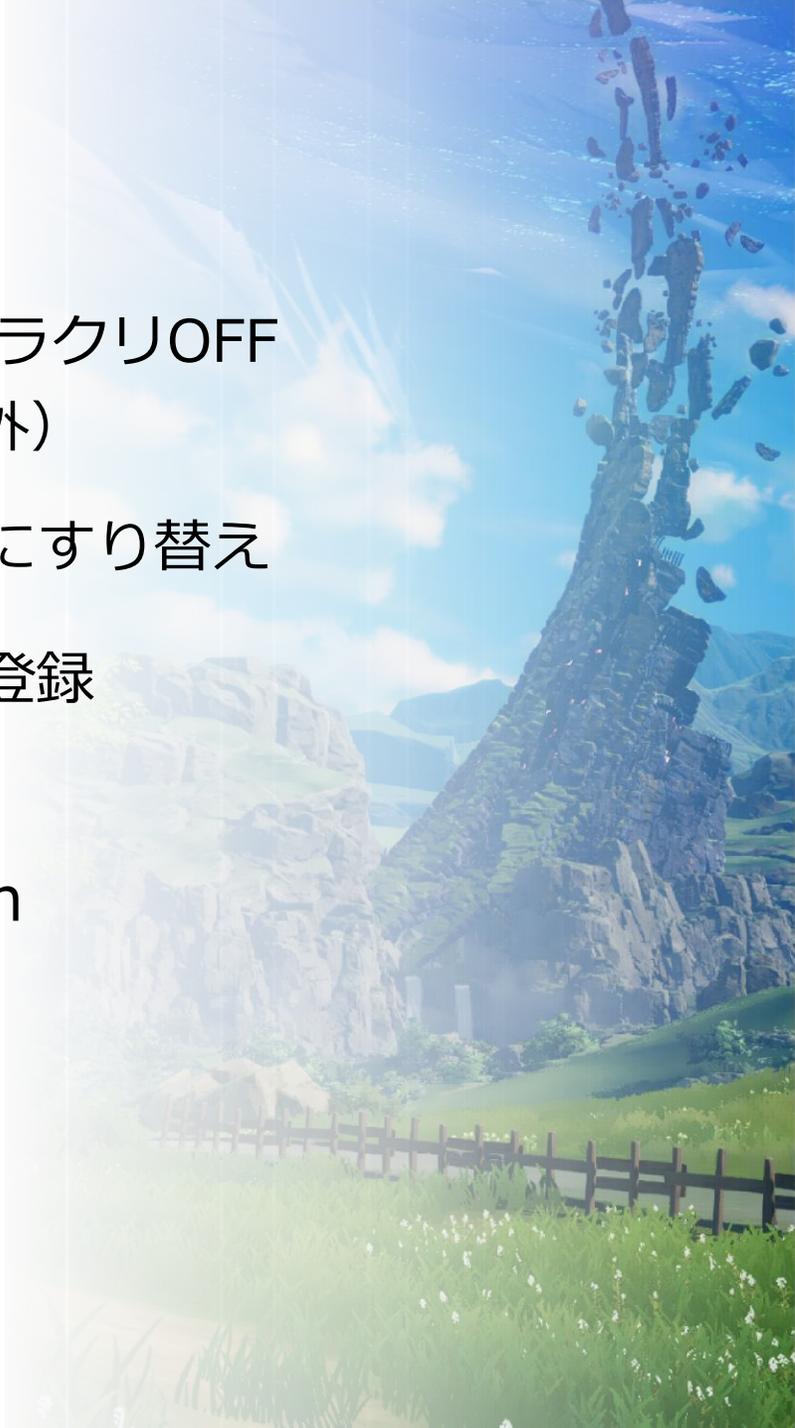


RenderThreadでは、オブジェクト数が負荷として直接影響を及ぼす処理が多い



その他の最適化

- オンラインゲームなのでサーバーでは大本からキャラクリOFF
(あらかじめ、ゲームに関わる部分はキャラクリ除外)
- 顔メッシュLOD、男女ごと共通のStaticMeshの顔にすり替え
- アニメーションしないアクセサリはStaticMeshを登録
- UE4設定項目
 - アニメーション更新頻度のLOD (URO) をOn
 - アニメーションのMultiThread更新処理をOn



アセット管理 & 作業環境

BLUE PROTOCOL™

To save the world that is going to destroy, fight beyond the spacetime. Cooperate with friends, beat the mighty enemies. On a vast land, heading for a hopeful future!

blue-protocol.com
©BANDAI NAMCO Online Inc.
©BANDAI NAMCO Studios Inc.



アセットデータ管理

キャラクリエイトデータ

- 各パーツ情報（名前、カラー、マテリアル）
- 体型スケール情報
- 男女

キャラクリエイトデータA



キャラクリエイトデータB



キャラクリエイトデータC



キャラクリエイトデータD



NPCも

アセットデータ管理

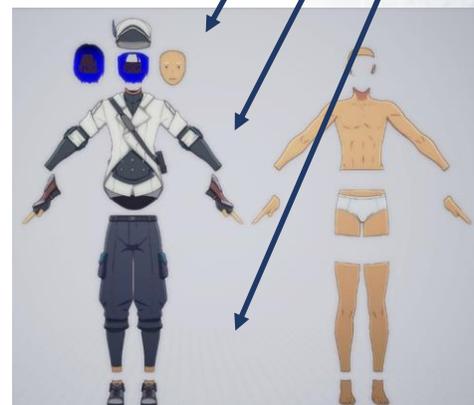
ユーザー情報を管理する
サーバー情報から上書き



キャラクリエイトデータ

- 各パーツ情報（名前、カラー、マテリアル）
- 体型スケール情報
- 男女

名前	部位	男	女
upper_a_001	Upper	<input type="text"/>	<input type="text"/>
upper_a_002	Upper	<input type="text"/>	<input type="text"/>
lower_a_001	Lower	<input type="text"/>	<input type="text"/>
lower_a_001	Lower	<input type="text"/>	<input type="text"/>
lower_b_001	Lower	<input type="text"/>	<input type="text"/>
glove_a_001	Glove	<input type="text"/>	<input type="text"/>





パーツデータテーブル管理

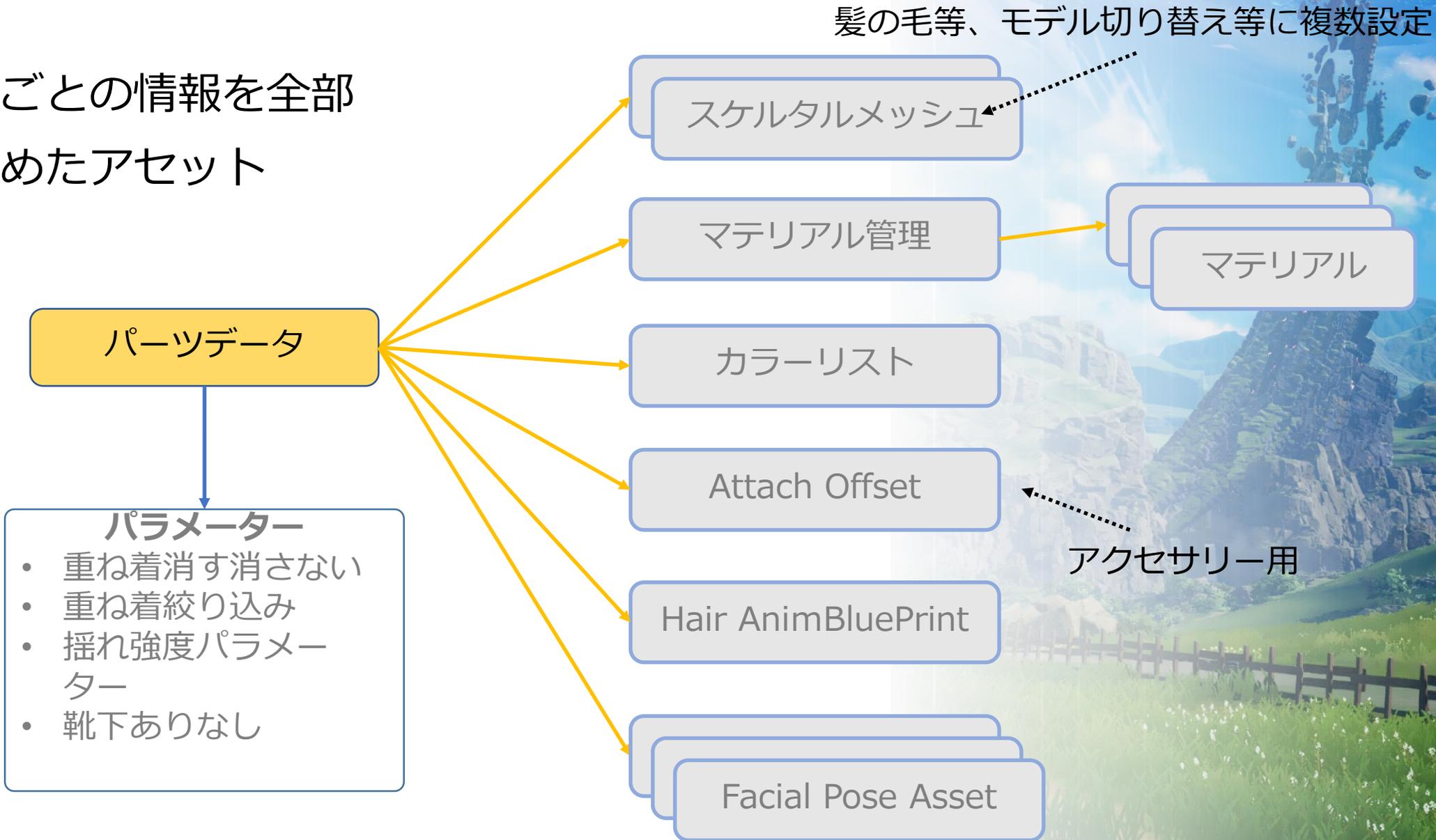
- テーブル管理
- 名前で管理

名前	部位	男	女
upper_a_001	Upper	パーツデータ	パーツデータ
upper_a_002	Upper	パーツデータ	パーツデータ
lower_a_001	Lower	パーツデータ	パーツデータ
lower_a_001	Lower	パーツデータ	パーツデータ
lower_b_001	Lower	パーツデータ	パーツデータ
glove_a_001	Glove	パーツデータ	パーツデータ



パーツデータ

- 部位ごとの情報を全部まとめたアセット



パーツデータ

- パーツアセットに紐づけて管理 サウンドは例外
 - 別テーブルに分散させる場合に比べて・・・煩雑になりにくい
 - アセットのロードの制御がシンプル
- 一括編集、一覧で確認したい場合
 - プロパティ マトリックス (UE4標準機能)
 - スクリプト
 - csv出力一覧表示
 - スクリプト一括置換

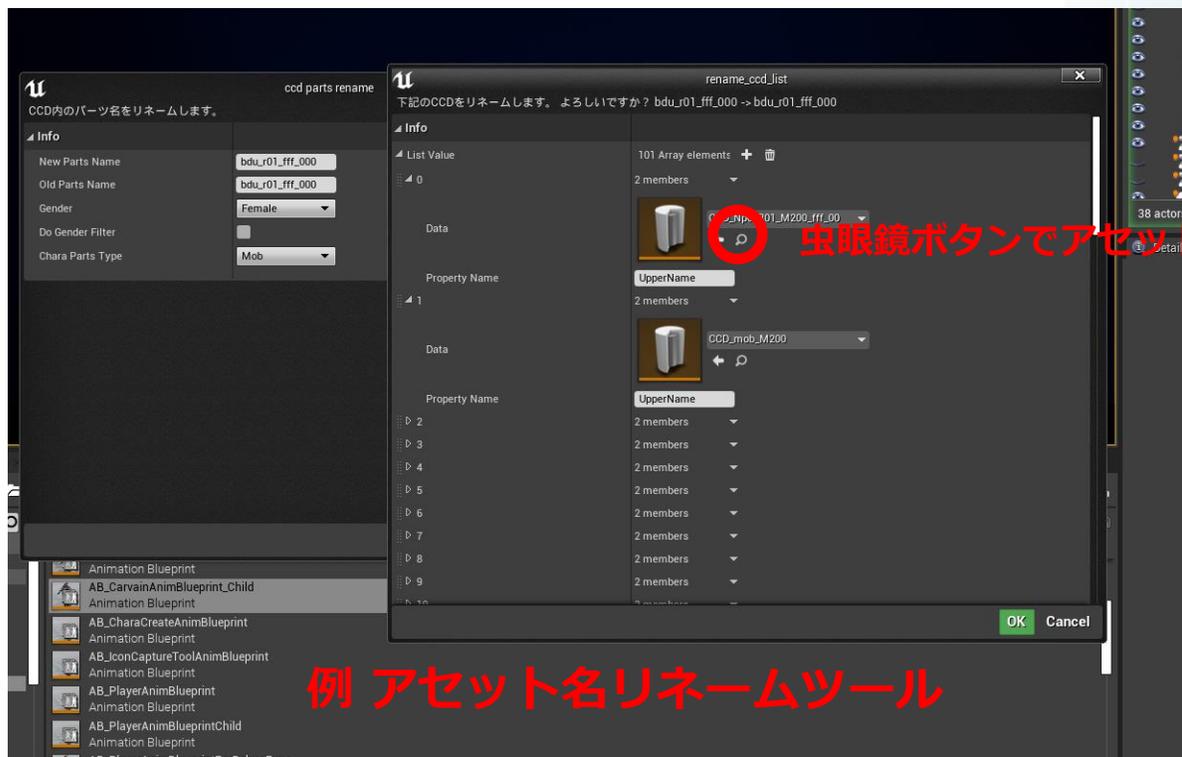


編集用Pythonスクリプト環境

- アセットを管理作業を楽に・・・

To save the world that is going to destroy, fight beyond the spacetime. Cooperate with friends, beat the mighty enemies, change the history. That is your mission. Now, let's sun out! On a vast land, heading for a hopeful future!

blue-protocol.com
©ANDAI NAMCO Online Inc.
©ANDAI NAMCO Studios Inc.



例 アセット名リネームツール





編集用Pythonスクリプト環境

メリット

- ビルド環境がなくてもコーディングができる
- C++、Blueprint比べて、見通しの良いコード書ける
- **様々なPythonモジュールの恩恵が簡単に受けれる**
- UI作成を楽に（独自改良、この後説明）

デメリット

- UE4で新しい事をする場合、かなり調べる必要がある
 - Intelisenseが使えない
 - C++のソースコードをいったり来たり





スクリプト環境 (UI)

独自改良の、UE4 Python環境のUI

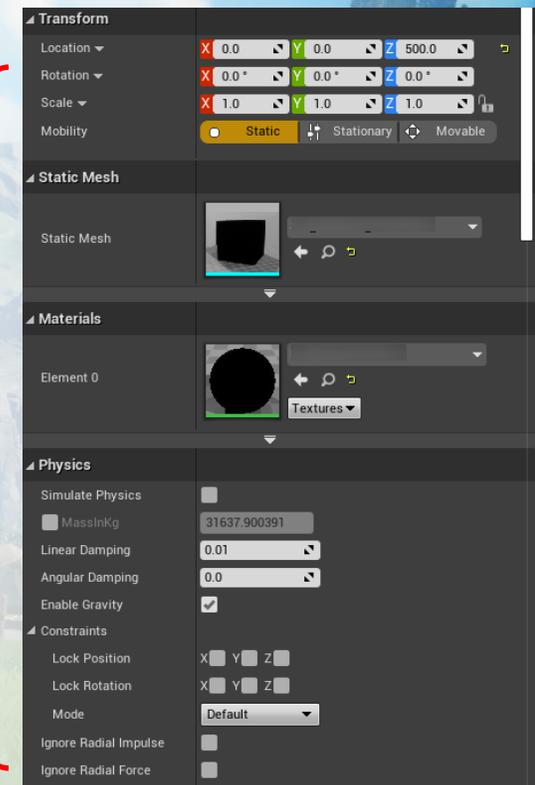
- UE4 Python
 - FStruct を動的に生成
- UE4 Editor Widget (Slate)
 - FStruct をWidgetとして、表示 & 編集できる機能



- [OK] [Cancel]を押した時のCallBack関数の追加

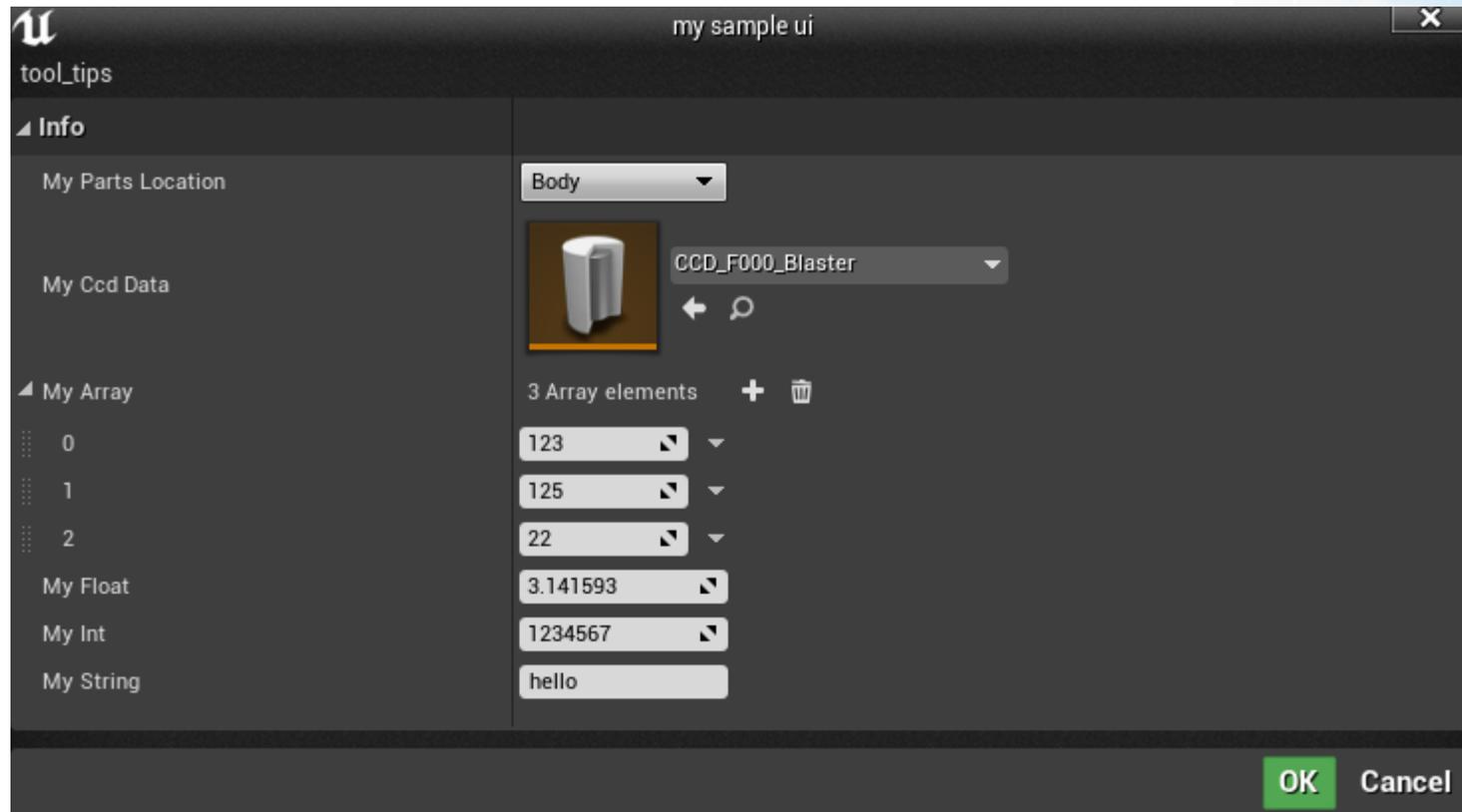
UPROPERTY

改良箇所は主に2つのみ



スクリプト環境

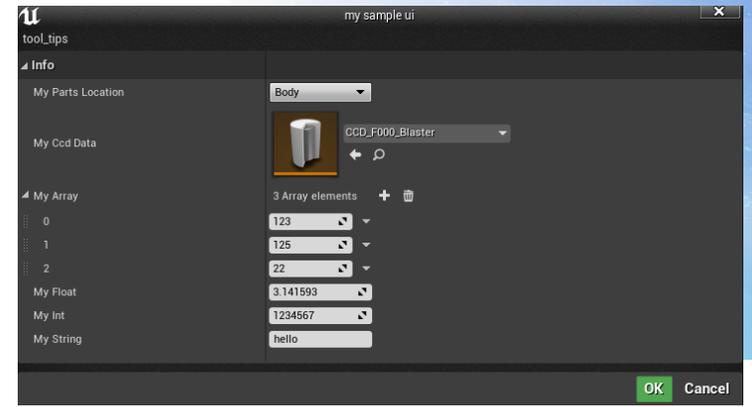
- UE4 Python





スクリプト環境

- UE4 Python



```
def sample_create_ui():
    test_array = unreal.Array(int)
    test_array.extend([123, 125, 22])

    property_dict = {
        "my_string": (str, "hello"),
        "my_int": (int, 1234567),
        "my_float": (float, 3.1415926535),
        "my_array": (unreal.Array(int), test_array),
        "my_ccd_data": (unreal.SBCharaCreateData, utility.get_asset_object("SBCharaCreateData"/Game/Character/CharaParts/CCD_F000_Bla
        "my_parts_location": (unreal.CharaPartsLocation, None),
    }
```

UI作成部分

```
def print_func(obj):
    sbgui = unreal.SBGUWrapper

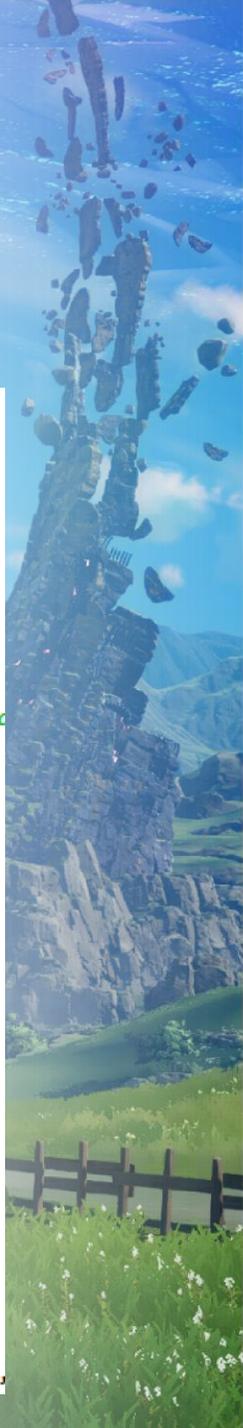
    ccd_data_value = obj.get_editor_property("my_ccd_data")
    if ccd_data_value:
        parameter = ccd_data_value.get_editor_property("Parameter")
        character_type = parameter.get_editor_property("CharacterType")
    else:
        character_type = None

    parts_location_value = obj.get_editor_property("my_parts_location")

    result_message = "character_type -> {}\nlocation -> {}".format(character_type, parts_location_value)
    print(result_message)
    result = sbgui.open_message_dialog(unreal.SBPyMessageDialogType.YES_NO, unreal.Text(result_message), unreal.Text("message"))
    print("result -> {}".format(result))
```

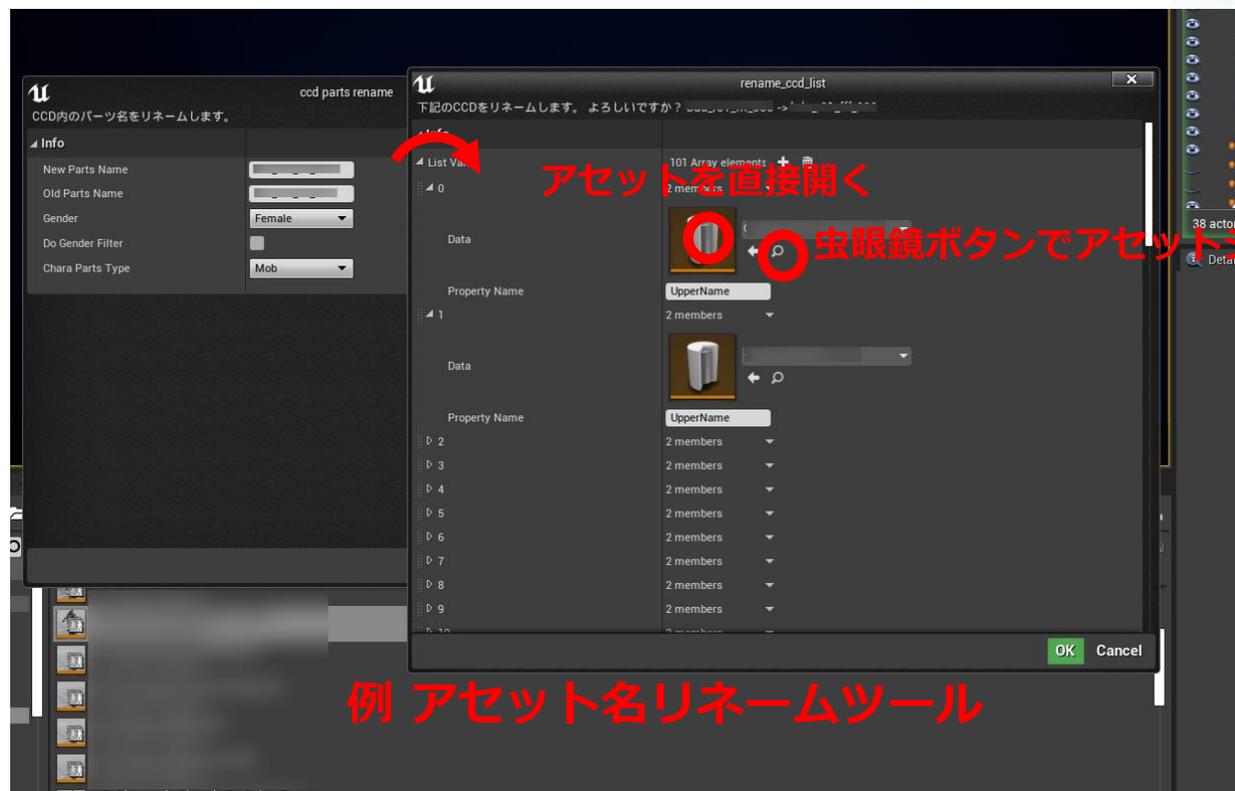
パラメーター取得
Callback関数部分

```
slate.create_custom_window(unreal.Text("my sample ui"), unreal.Text("tool_tips"), unreal.Text("tool_button_tips"), property_dict,
```



スクリプト環境 (例)

- UE4 Python



例 アセット名リネームツール



その他、Pythonスクリプトおすすめ情報

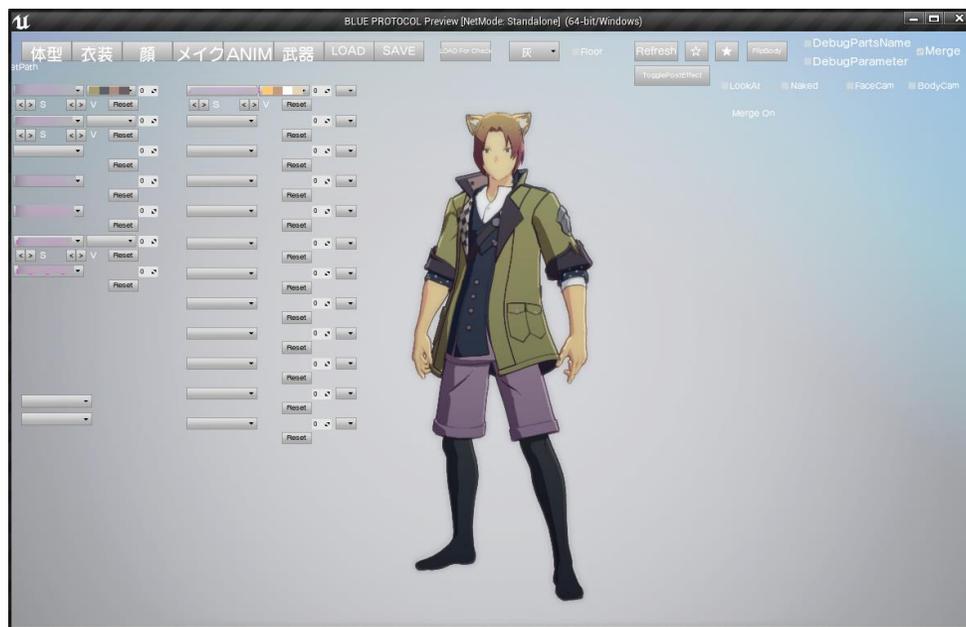
- pydevdモジュールで、IDEでステップ実行等のデバッグが可能
- Python3.7にUE4.26から対応
 - VFXPlatformがPython3へ

快適なPythonによる編集環境



キャラクリビューワー (デバッグ用)

- パーツ着せ替え等、キャラクリ全機能を触れる (色変え、メイク等も)
- アニメーション再生など



キャラクリビューワー (デバッグ用)

- 用途

- パーツデータ作成時の確認用途
- 開発時のキャラクリスクリプト資料用途
- NPC向けキャラクリエイトデータ作成ツール
- 動作確認用
 - インゲームと、キャラクリのその他 (UI、アウトゲーム) とで不具合の発生時の問題切り分け
 - マージ&非マージ



不具合小&メンテナンスコスト低になるように心掛けながら作成



体型变更

BLUE PROTOCOL™

To save the world that is going to destroy, fight beyond the spacetime. Cooperate with friends, beat the mighty enemies, change the history. That is your mission. Now, let's run out! On a vast land, heading for a hopeful future!

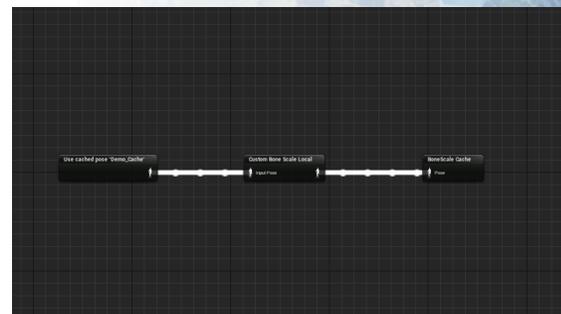
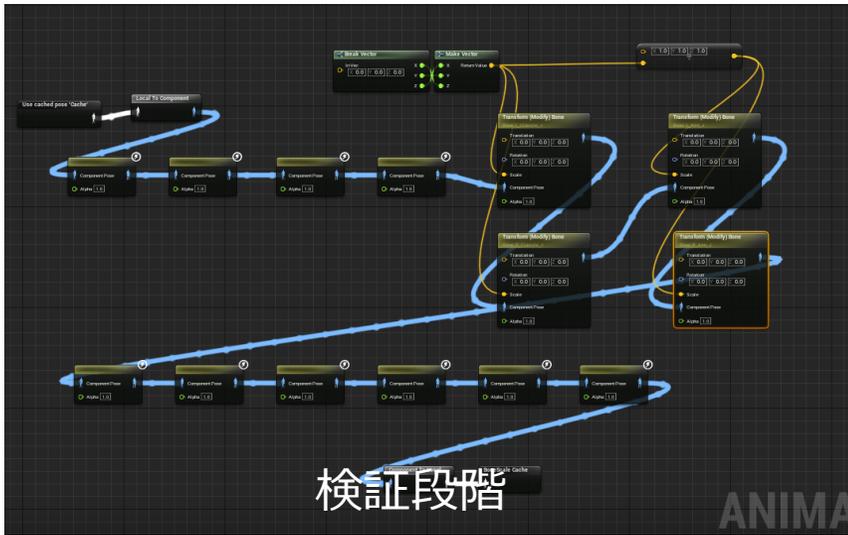
blue-protocol.com

©BANDAI NAMCO Online Inc.
©BANDAI NAMCO Studios Inc.



体型変更、最適化

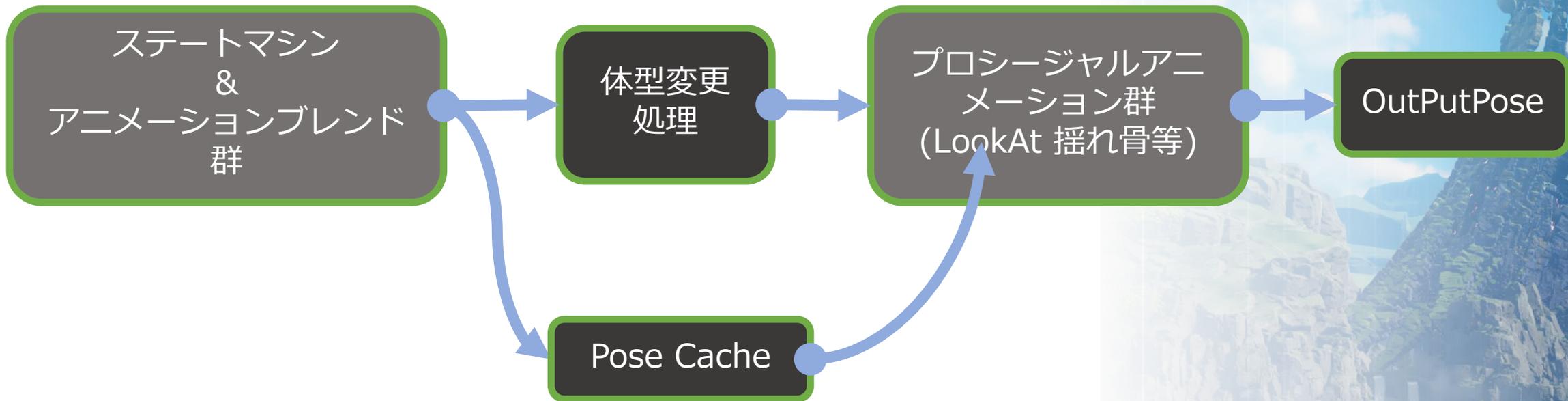
- 最終的には1ノード化
- 変更した時のみ計算
 - キャッシュしたローカルスケールを、毎回更新



1ノード化



体型変更、計算処理順番

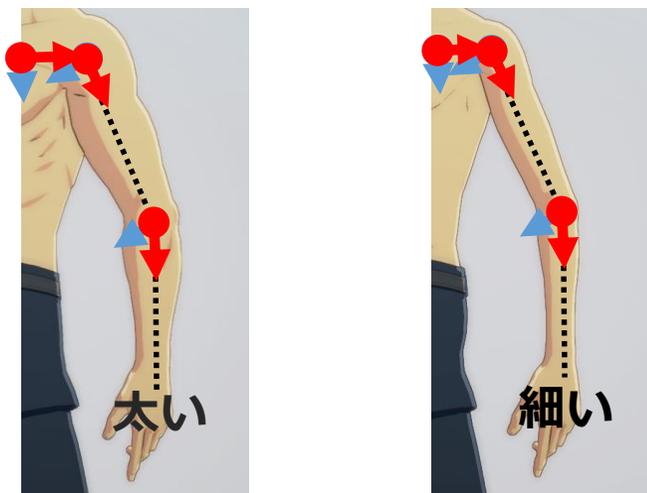


3 軸違う値の骨スケーリング時の注意

プロシージャルアニメーション等の骨計算を行う場合の注意点

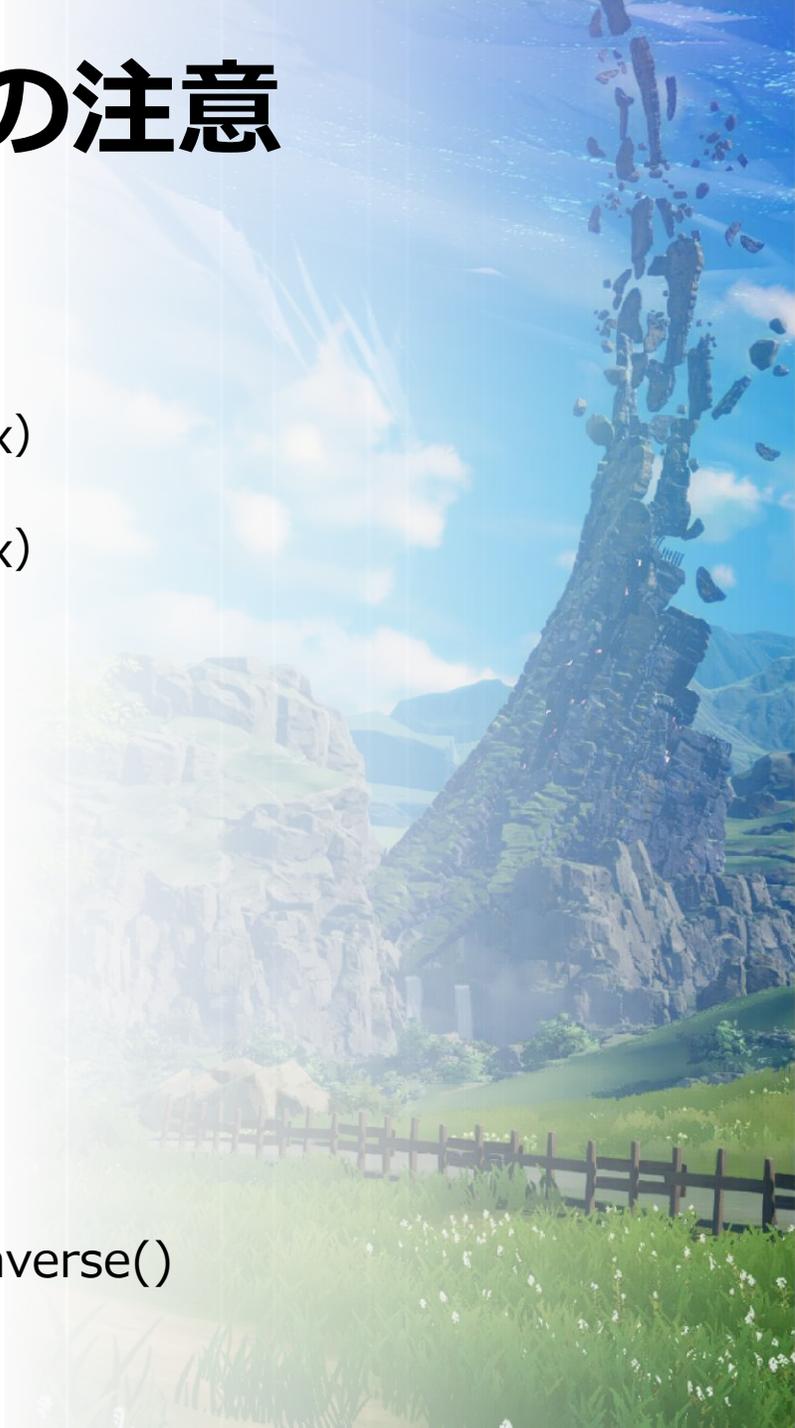
3軸のスケーリング値が、

- 同じ場合
 $UE4の階層計算 (FTransform) = Matrixの階層計算 (FMatrix)$
- 違う場合、Matrixの階層計算に誤差
 $UE4の階層計算 (FTransform) \neq Matrixの階層計算 (FMatrix)$



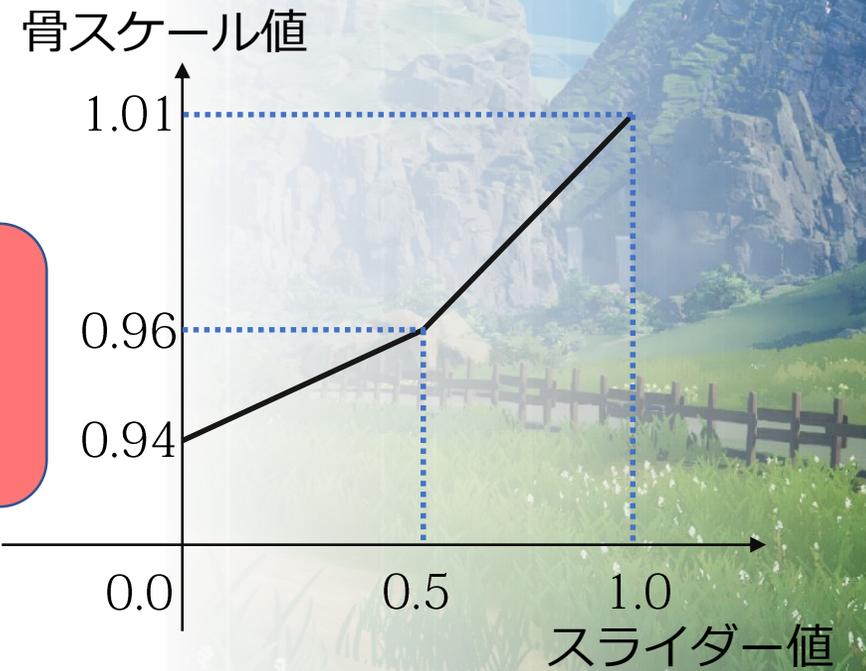
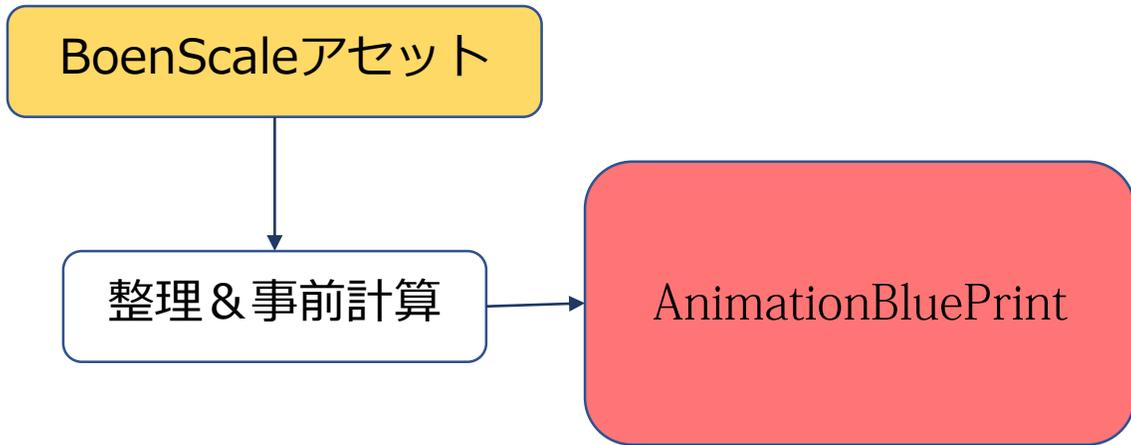
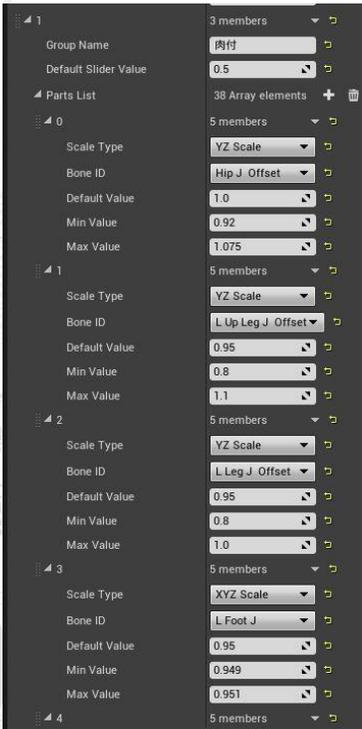
対処方法

- FMatrixを使わないで、FTransformで階層計算
- FTransformでも、Inverseが使えない、かわりに専用の関数
 $LocalTransform = WorldTransform * ParentTransform.Inverse()$
 $\Rightarrow FTransform::GetRelativeTransform$
- 一部、誤差が目立たない場合は許容



体型変更

- アーティストが調節しやすいようにスケール情報は、アセットにまとめて管理しています
 - スケールIDを介して設定
 - どの軸をスケールさせるか指定 (+X軸を子Jointの方向としてます)
 - 最小値0.0、中間値0.5、最大値1.0を設定し線形補間





物理揺れ

BLUE PROTOCOL™

To save the world that is going to destroy, fight beyond the spacetime. Cooperate with friends, beat the mighty enemies, change the history. That is your mission. Now, let's run out! On a vast land, heading for a hopeful future!

blue-protocol.com

©BANDAI NAMCO Online Inc.
©BANDAI NAMCO Studios Inc.



物理揺れ

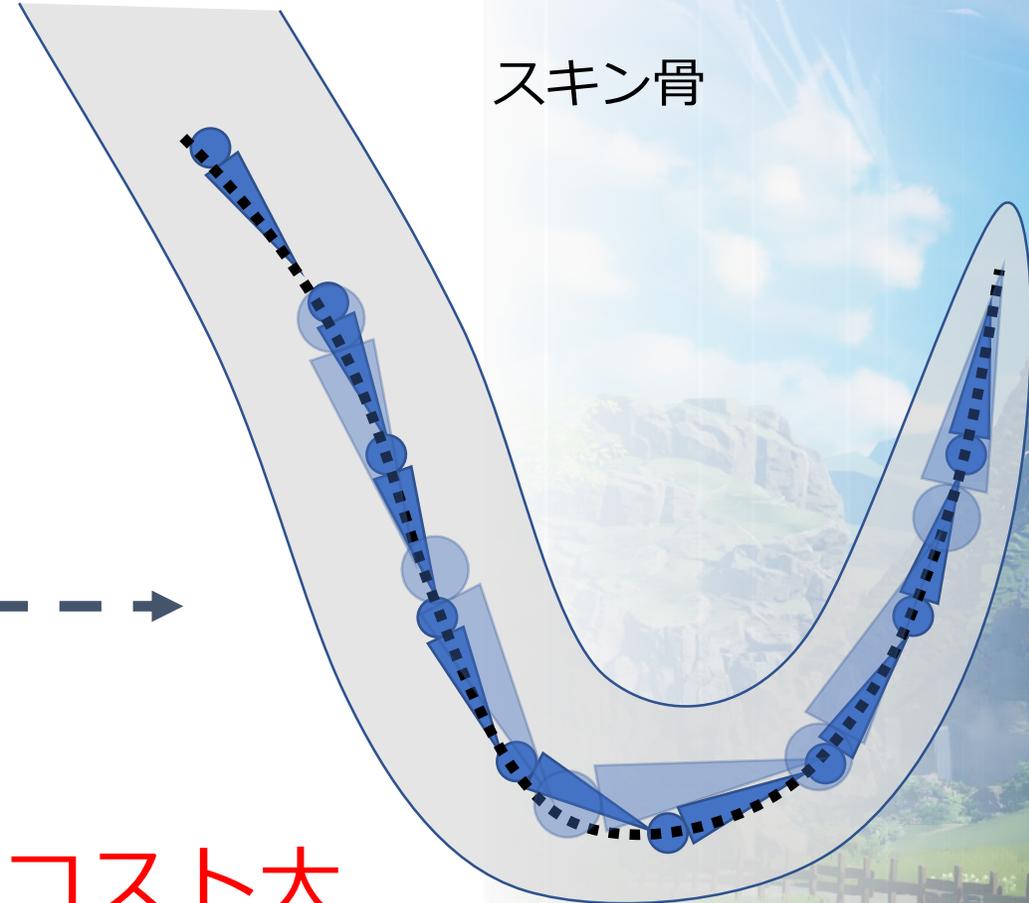
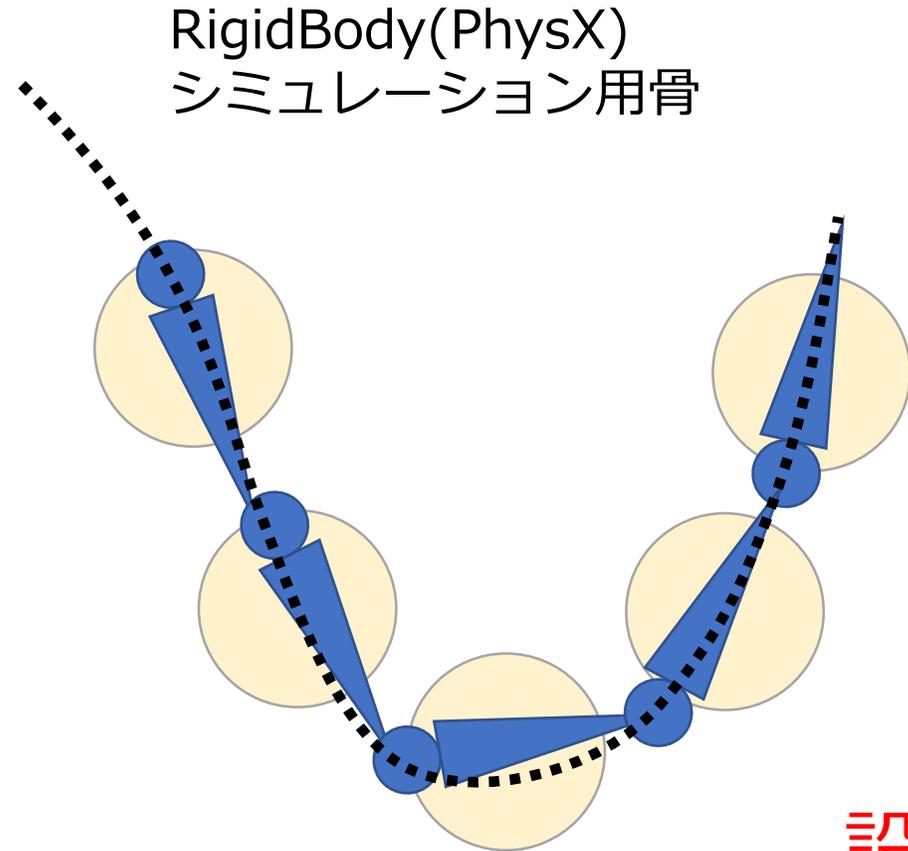
- 当初の予定
 - RigidbodyNode(UE4標準) PhysX (⇒ chaosに移行予定)
 - Cloth(UE4標準) ApexCloth (⇒ chaosに移行予定)



- 現在
 - RigidbodyNode(UE4標準)
 - **KawaiiPhysics(EPIC おかずさん個人作成)**
 - Cloth(UE4標準)



物理揺れ(RigidBodyNodeでの設定コスト)

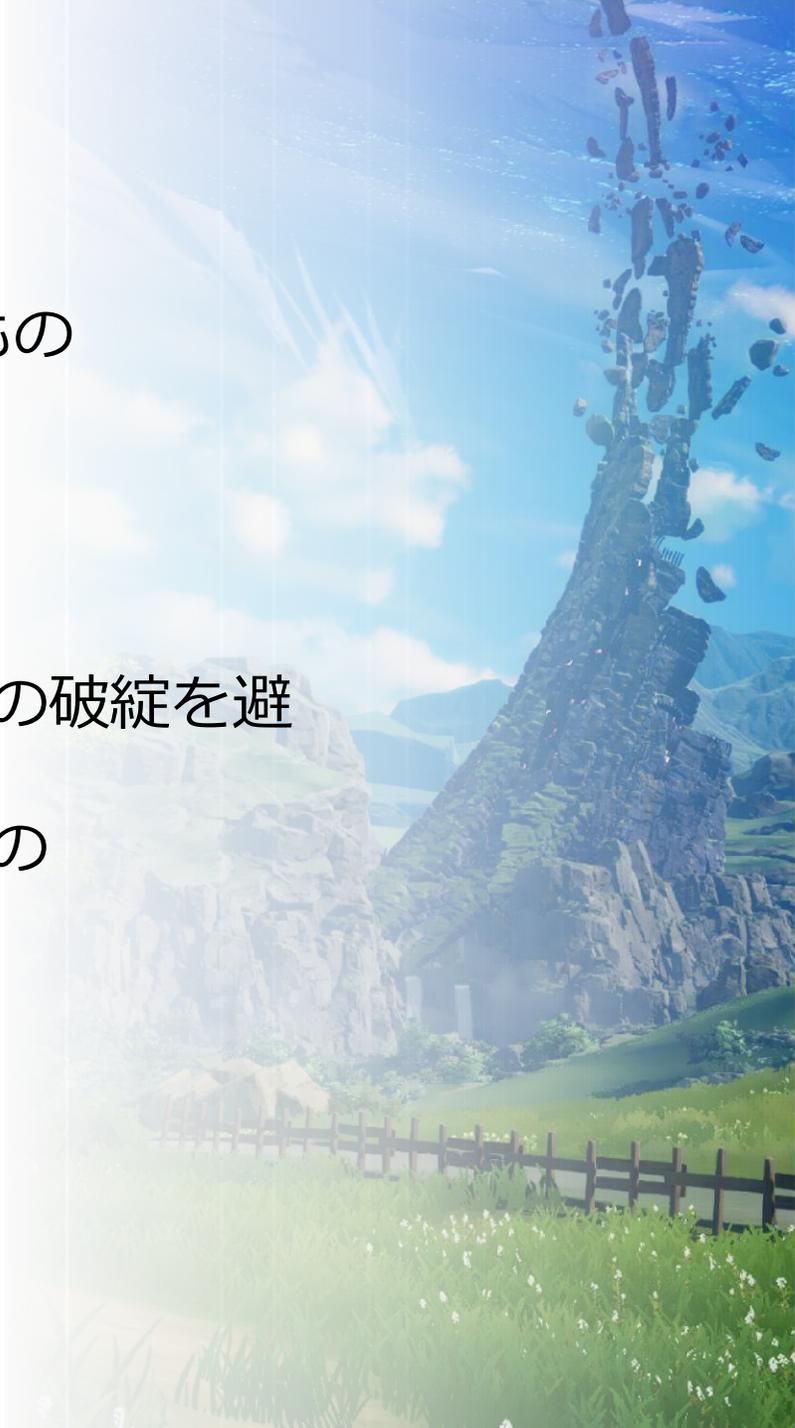


設定コスト大

この方法はオミット、直接KawaiiPhysics設定

物理揺れの使い分け

- Rigidbody
 - 回転運動など、Rigid物理的な挙動を求めるもの
 - 骨の階層が少ないもの（3以下目安）
 - 胸揺れ、アクセサリー
- KawaiiPhysics
 - チェーン状で骨の階層が深いもの、その場合の破綻を避けたいもの
 - 設定が楽なので、多くの個別設定が必要なもの
 - 髪の毛、アクセサリー一部
- Cloth
 - 縦だけでなく、横のつながりが必須なもの
 - 布



物理揺れ独自カスタマイズ（苦労話）

- 共通カスタマイズ
 - スケール対応、コリジョン&物理アセット自体のスケール
 - 揺れもののリセット&空回し処理（カットシーン向け）
- RigidbodyNode
 - 物理パラメーターをスケールするかしないかパラメータ
 - PhysXアセット汎用化（階層が同じなら設定データを流用できる対応）
- KawaiiPhysics
 - Limitのローカル対応等
- Cloth
 - メッシュマージ（Clothは骨ベースのシミュレーションではないので）
 - シミュレーション用メッシュの、LOD化対応



汎用化

BLUE PROTOCOL™

To save the world that is going to destroy, fight beyond the spacetime. Cooperate with friends, beat the mighty enemies, change the history. That is your mission. Now, let's run out! On a vast land, heading for a hopeful future!

blue-protocol.com

©BANDAI NAMCO Online Inc.
©BANDAI NAMCO Studios Inc.



汎用化

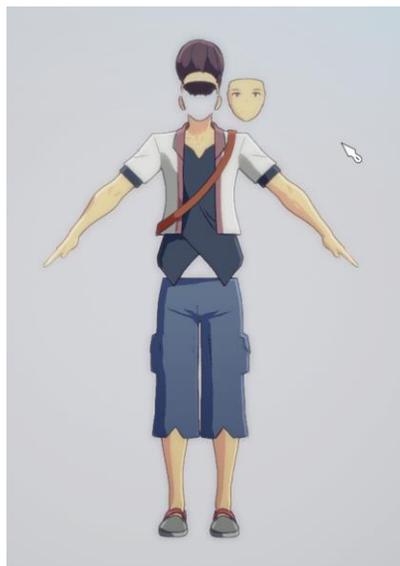
- キャラクリ適用範囲
 - プレイヤー（アバター）
 - NPC(MOB)
 - エネミー
- キャラクリの仕組みを汎用化するメリット
 - ゲーム中身と、見た目の切り分け、の活用
 - アセットの作成学習コストを下げる



NPC 汎用化

NPCのバリエーションはキャラクリの仕組みを使用

- パーツ数 4 (髪の毛、顔、上半身、下半身)
- 顔は2パターン
 - 体とマージ (話さないMOB)
 - 体と別パーツ 骨ありバージョン (話すMOB)



NPC 汎用化

NpcCharacterをPoolしておき、
キャラクリエイトデータを書き換えて見た目を変えて、使いまわし、
Characterの生成、破棄の処理負荷を低減



Enemy汎用化

- Enemyの見た目変更にも採用
 - パーツ1つ
 - メッシュ、マテリアル、カラー変更のみ



エネミーの今後の見た目の拡張性大





まとめ(後半)

- 非同期メッシュマージ紹介
 - ゲーム中は、メッシュのマージを行うことで最適化
- アセット作成の効率化
 - キャラクリデータを使った見た目管理
 - パーツ単位でのデータ管理方法
 - PythonスクリプトでUIの作成
 - キャラクリビューワーの活用
- 骨ベースのスケーリングの紹介
- 物理揺れの使い分けの紹介
- NPCやエネミーもキャラクリ化による、メリットについて紹介





キャラクリエーションを実現しようとする場合の、
開発のヒントになるものが得られれば幸いです。

ご清聴ありがとうございました。

