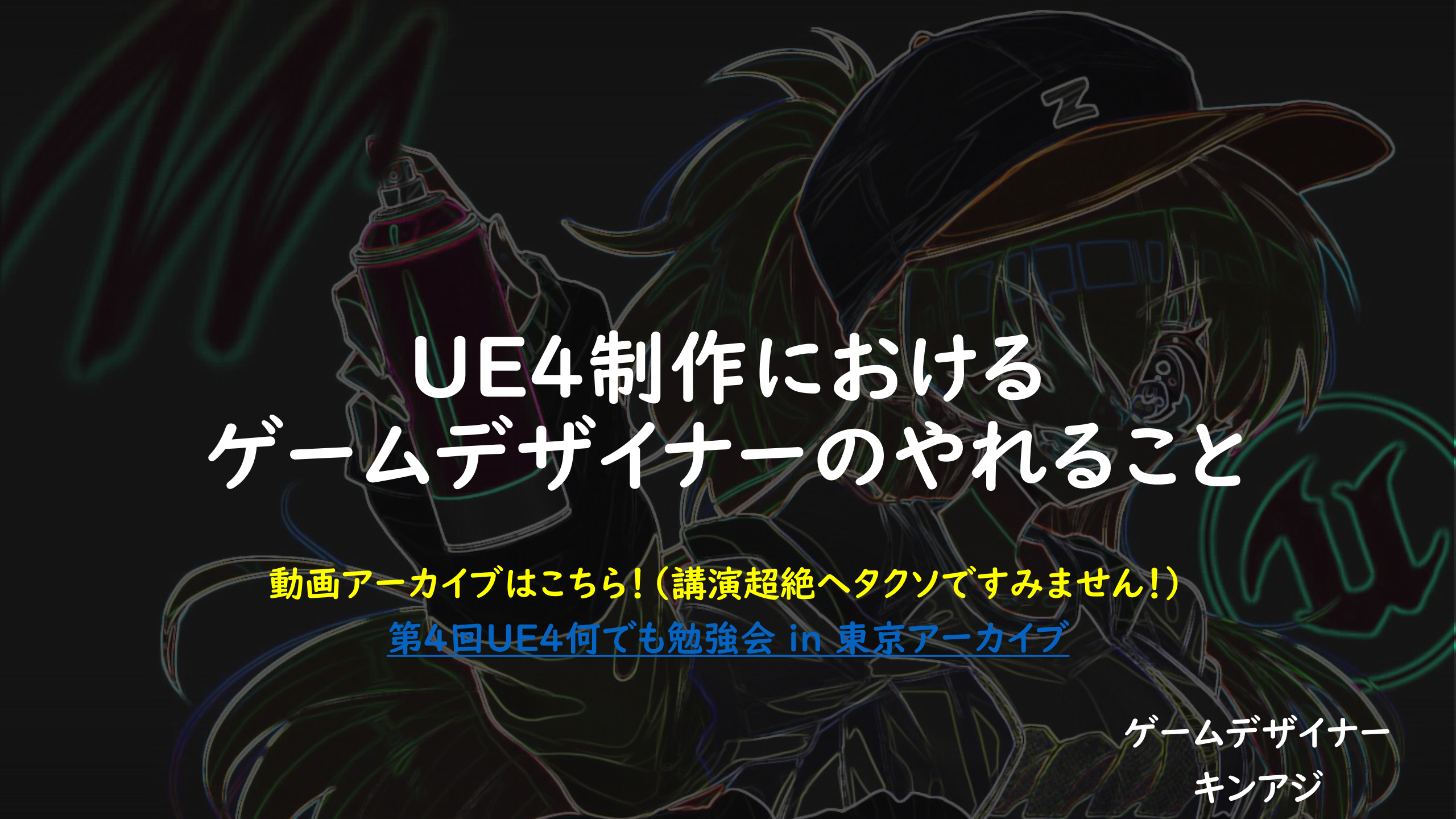


UE4制作における
ゲームデザイナーのやれること

動画アーカイブはこちら！（講演超絶ヘタクソですみません！）

第4回UE4何でも勉強会 in 東京アーカイブ

ゲームデザイナー
キンアジ

注意

- スライド内のUE4Editorのキャプチャー画像は
  4.24.2を使用しています（一部4.23.2）
- 業界や会社によって、多少文言の解釈が異なる部分があるかと思いますが、
  ここに書かれているのはあくまで個人的な解釈での意味合いとなります。
  （例えばゲームプランナー≒ゲームデザイナー）
- こちらの資料は後日配布いたします。
- SNSへの投稿はジャンジャンやっちゃってください！
  タグ：#UE4AllStudy

自己紹介

キンアジ

・Twitter：@kinnaji_blog

・UE4ブログ：https://kinnaji.com/

・UE4歴：丁度５年ぐらい

・UE4での開発経験：PS4、Nintendo switch、アーケード、オンライン、ノンゲームなど

・趣味：釣り

会社紹介

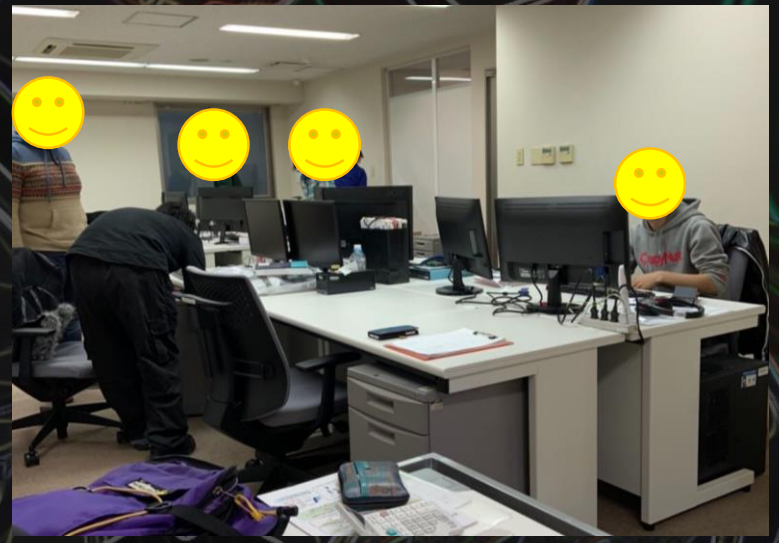
株式会社アンナプルナ

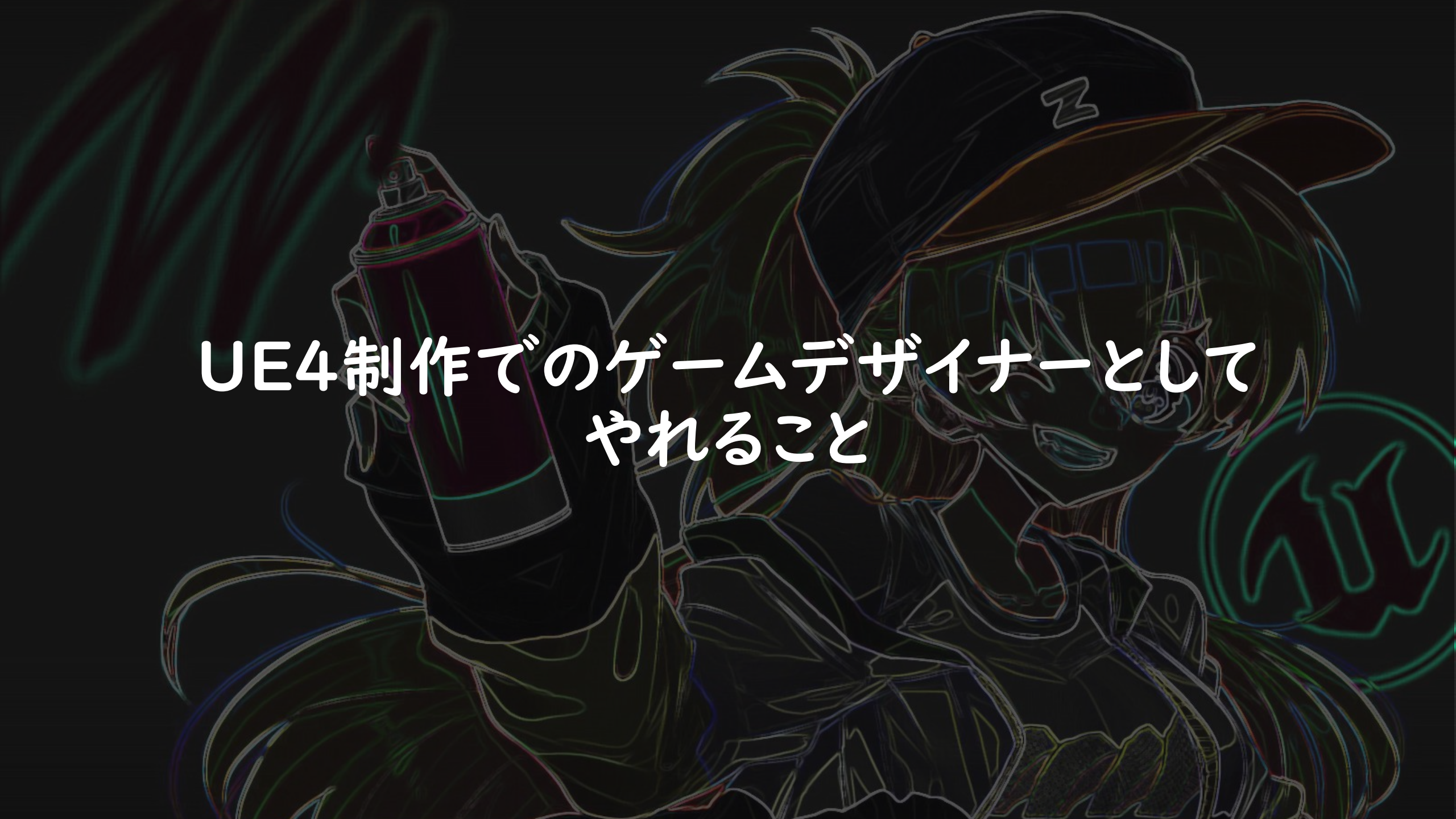
→HP:http://anapurna.co.jp/
→Twitter:@Anna045GameDev

ドラゴンクエストXI　S（開発協力）
ナレルンダー！仮面ライダージオウ

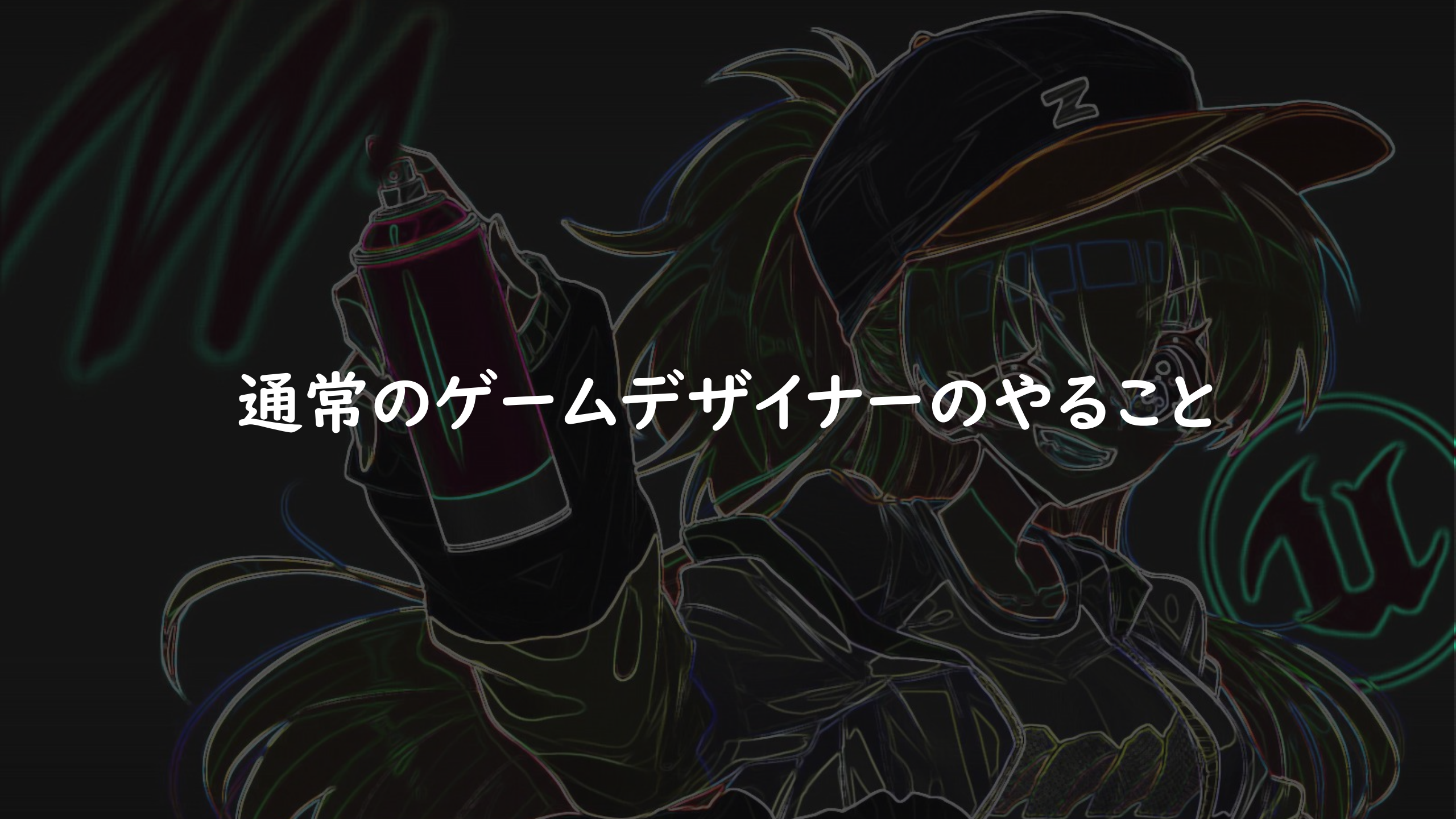
※この２月に横浜にオフィスを構えました。
（仕事ください！）

Epic Games Launcher
に会社のブログ乗ってたり



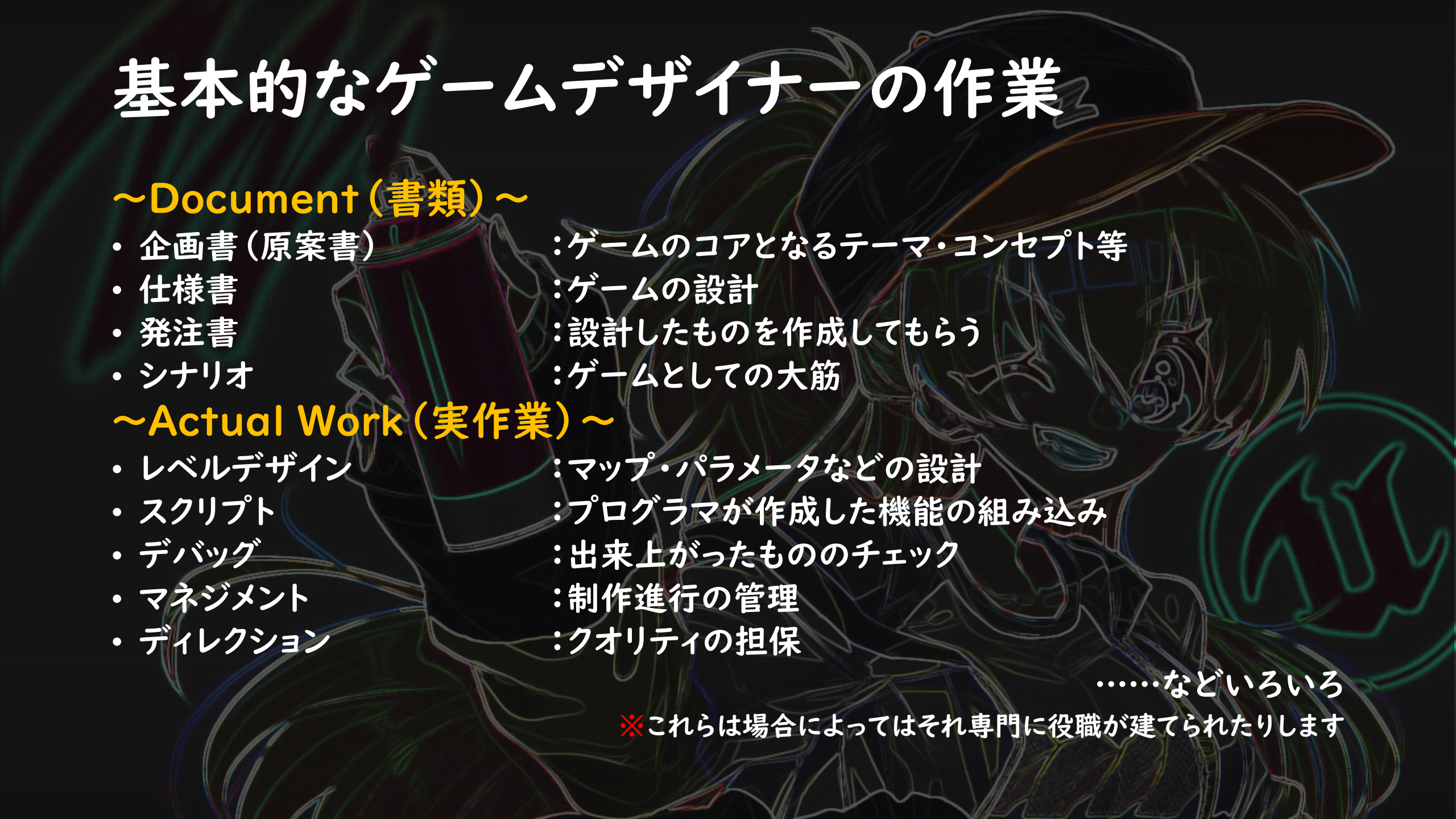


UE4制作でのゲームデザイナーとしてやれること



通常のゲームデザイナーのやること

基本的なゲームデザイナーの作業



～Document（書類）～

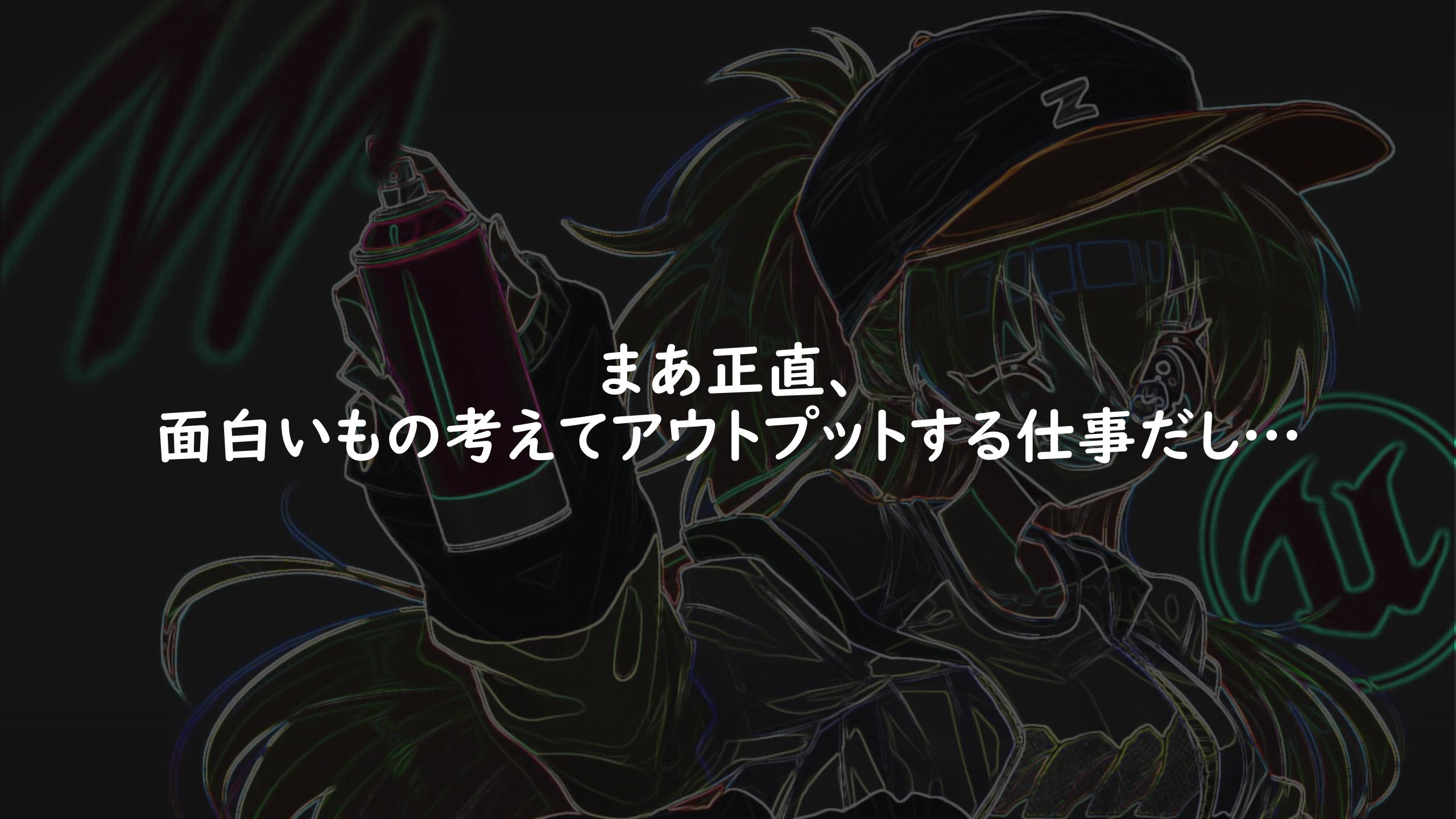
- 企画書（原案書）　：ゲームのコアとなるテーマ・コンセプト等
- 仕様書　：ゲームの設計
- 発注書　：設計したものを作成してもらう
- シナリオ　：ゲームとしての大筋

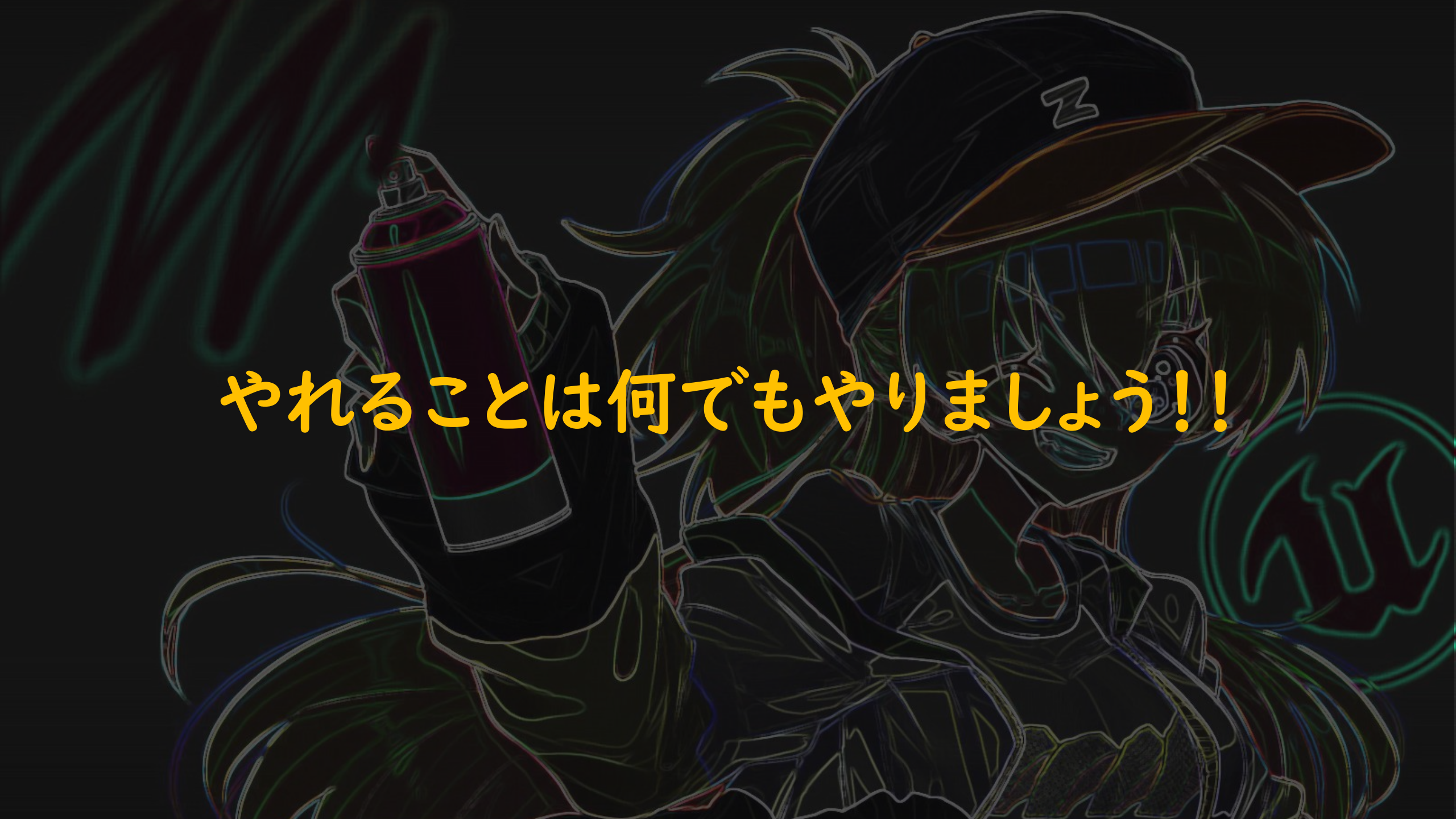
～Actual Work（実作業）～

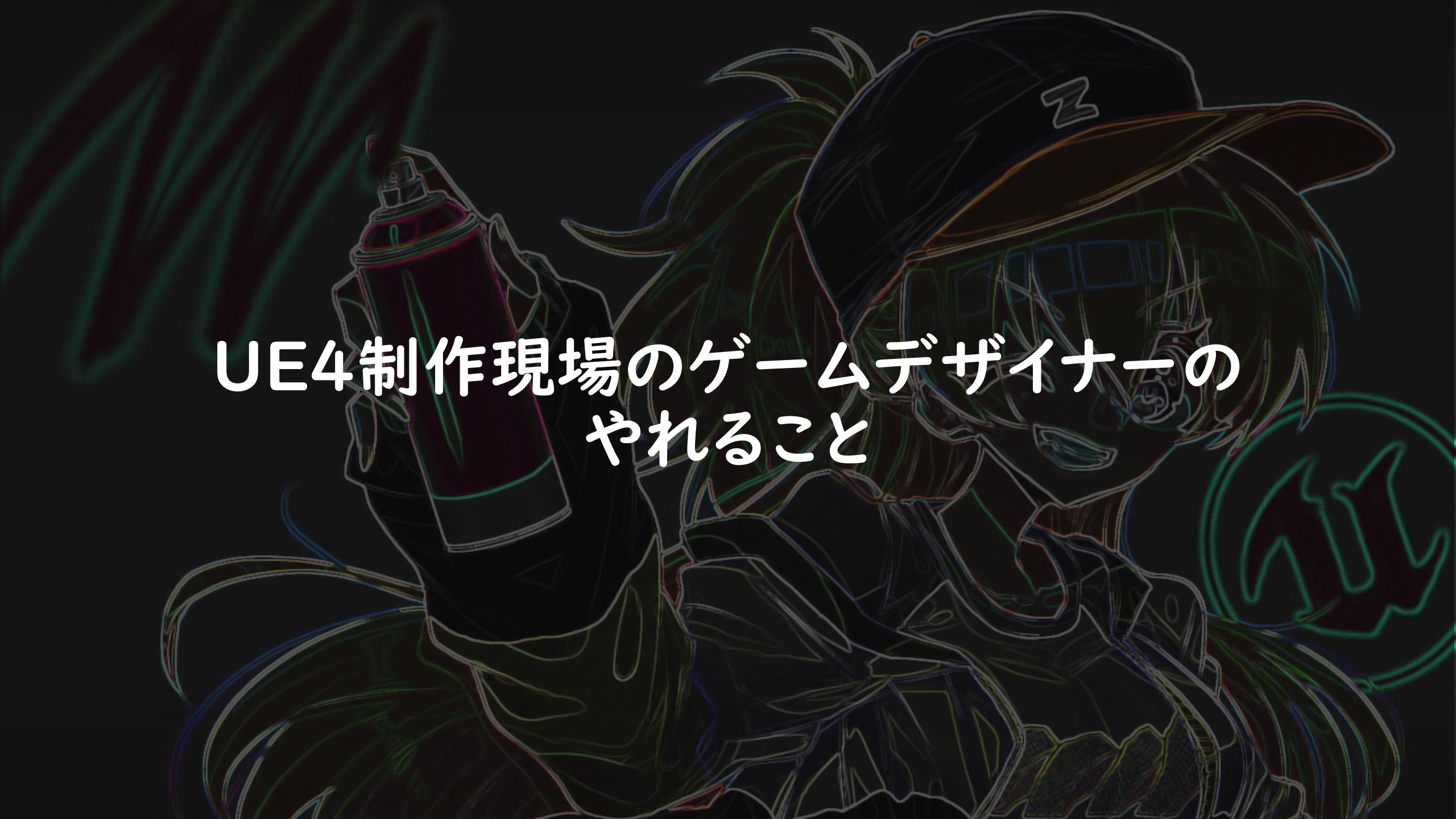
- レベルデザイン　：マップ・パラメータなどの設計
- スクリプト　：プログラマが作成した機能の組み込み
- デバッグ　：出来上がったもののチェック
- マネジメント　：制作進行の管理
- ディレクション　：クオリティの担保

……などいろいろ

※これらは場合によってはそれ専門に役職が建てられたりします


まあ正直、
面白いもの考えてアウトプットする仕事だし…


やれることは何でもやりましょう!!



UE4制作現場のゲームデザイナーのやれること

UE4制作の現場では特にここで活躍！

～Document（書類）～

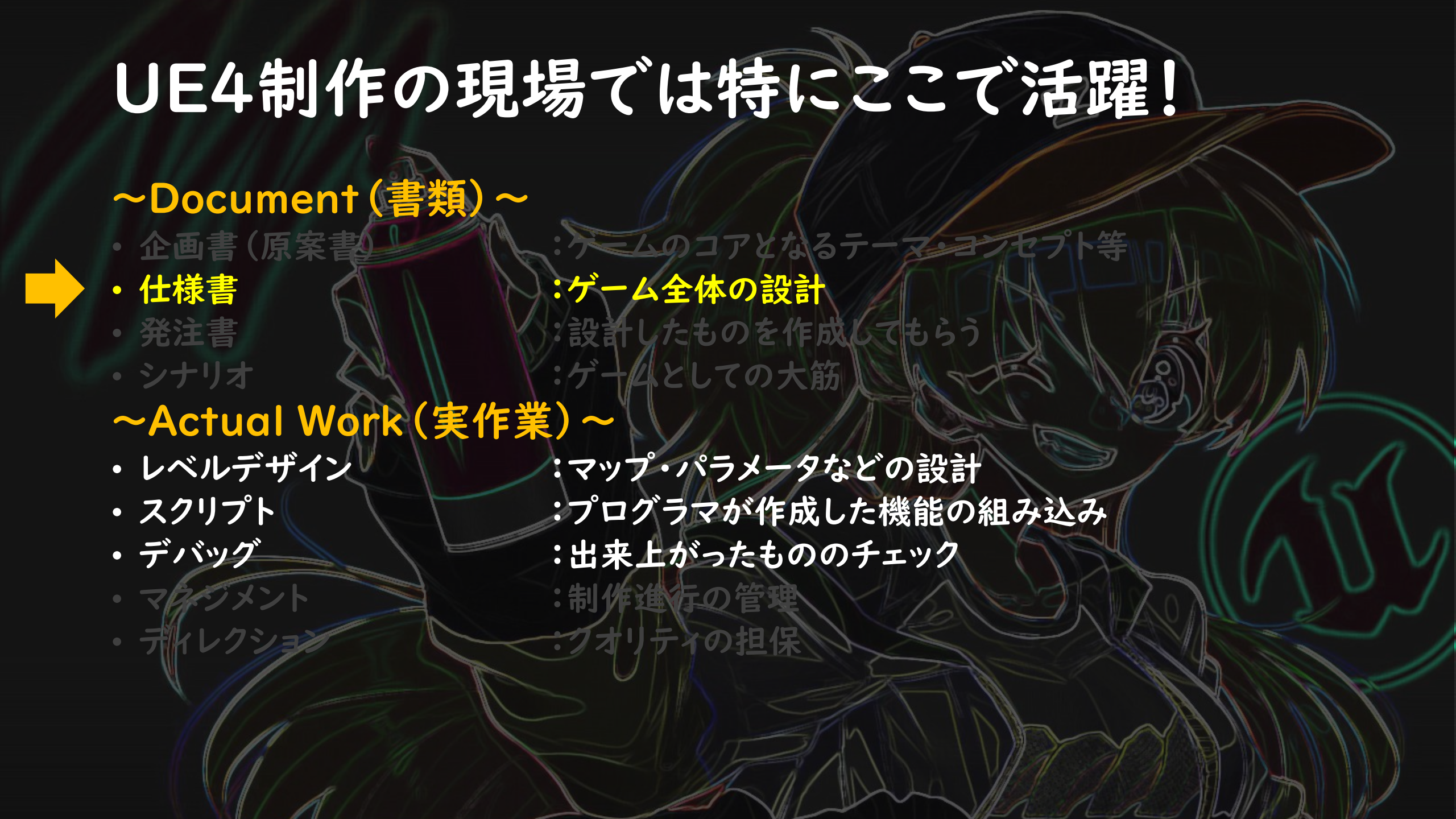
- 企画書（原案書）　：ゲームのコアとなるテーマ・コンセプト等
- 仕様書　：ゲーム全体の設計
- 発注書　：設計したものを作成してもらう
- シナリオ　：ゲームとしての大筋

～Actual Work（実作業）～

- レベルデザイン　：マップ・パラメータなどの設計
- スクリプト　：プログラマが作成した機能の組み込み
- デバッグ　：出来上がったもののチェック
- マネジメント　：制作進行の管理
- ディレクション　：クオリティの担保

誰でもエンジンを使えば実際に動くものを素早く作成できる！

UE4制作の現場では特にここで活躍！

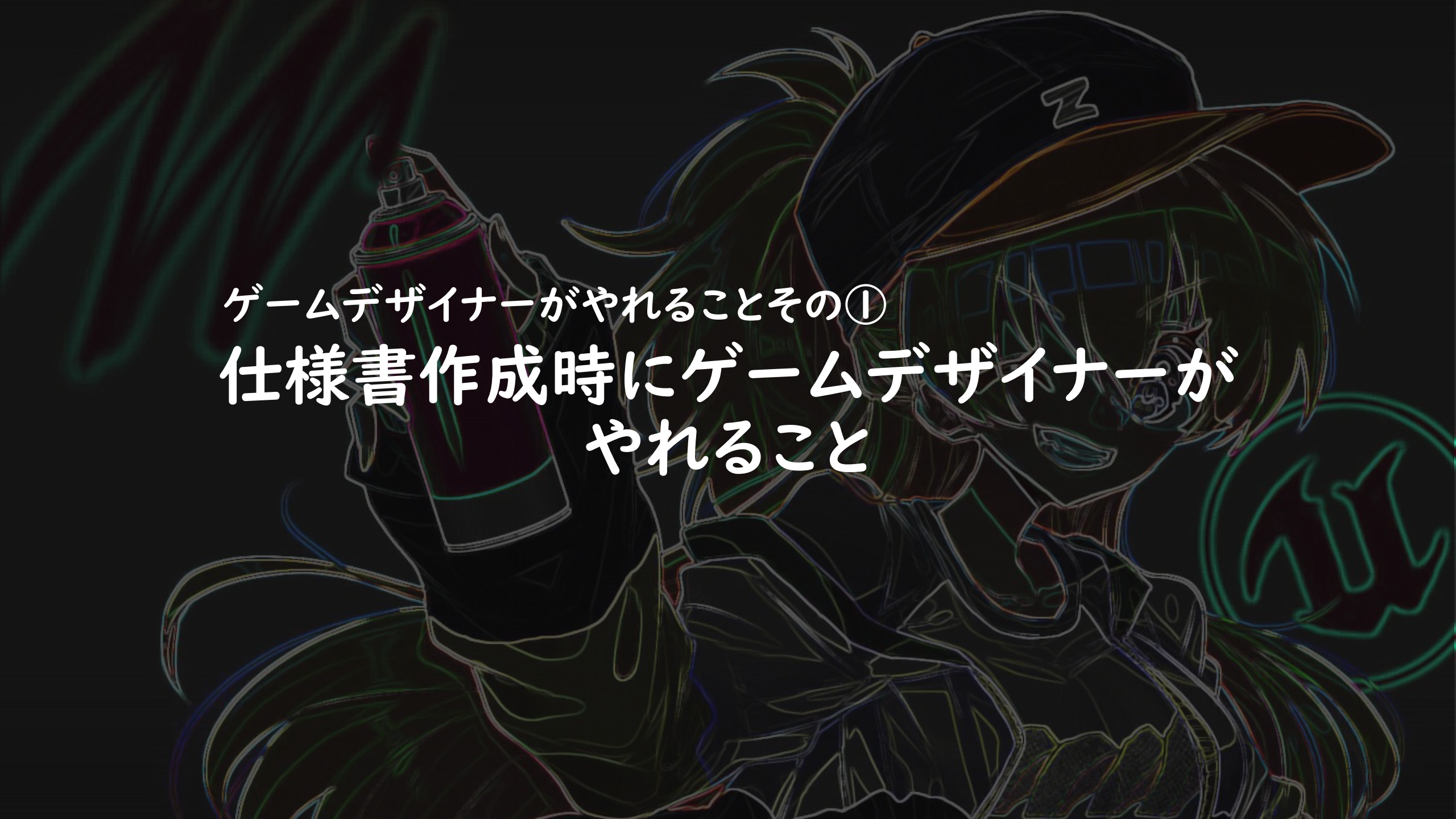


～Document（書類）～

- 企画書（原案書）　：ゲームのコアとなるテーマ・コンセプト等
- **仕様書**　：**ゲーム全体の設計**
- 発注書　：設計したものを作成してもらう
- シナリオ　：ゲームとしての大筋

～Actual Work（実作業）～

- レベルデザイン　：マップ・パラメータなどの設計
- スクリプト　：プログラマが作成した機能の組み込み
- デバッグ　：出来上がったもののチェック
- マネジメント　：制作進行の管理
- ディレクション　：クオリティの担保



ゲームデザイナーがやれることその①

仕様書作成時にゲームデザイナーがやれること

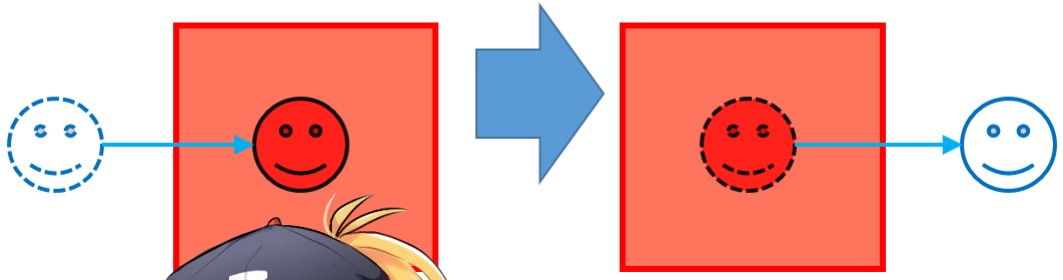
そろそろ…
Excelアートやめません？

エクセルアートの仕様

ギミック説明

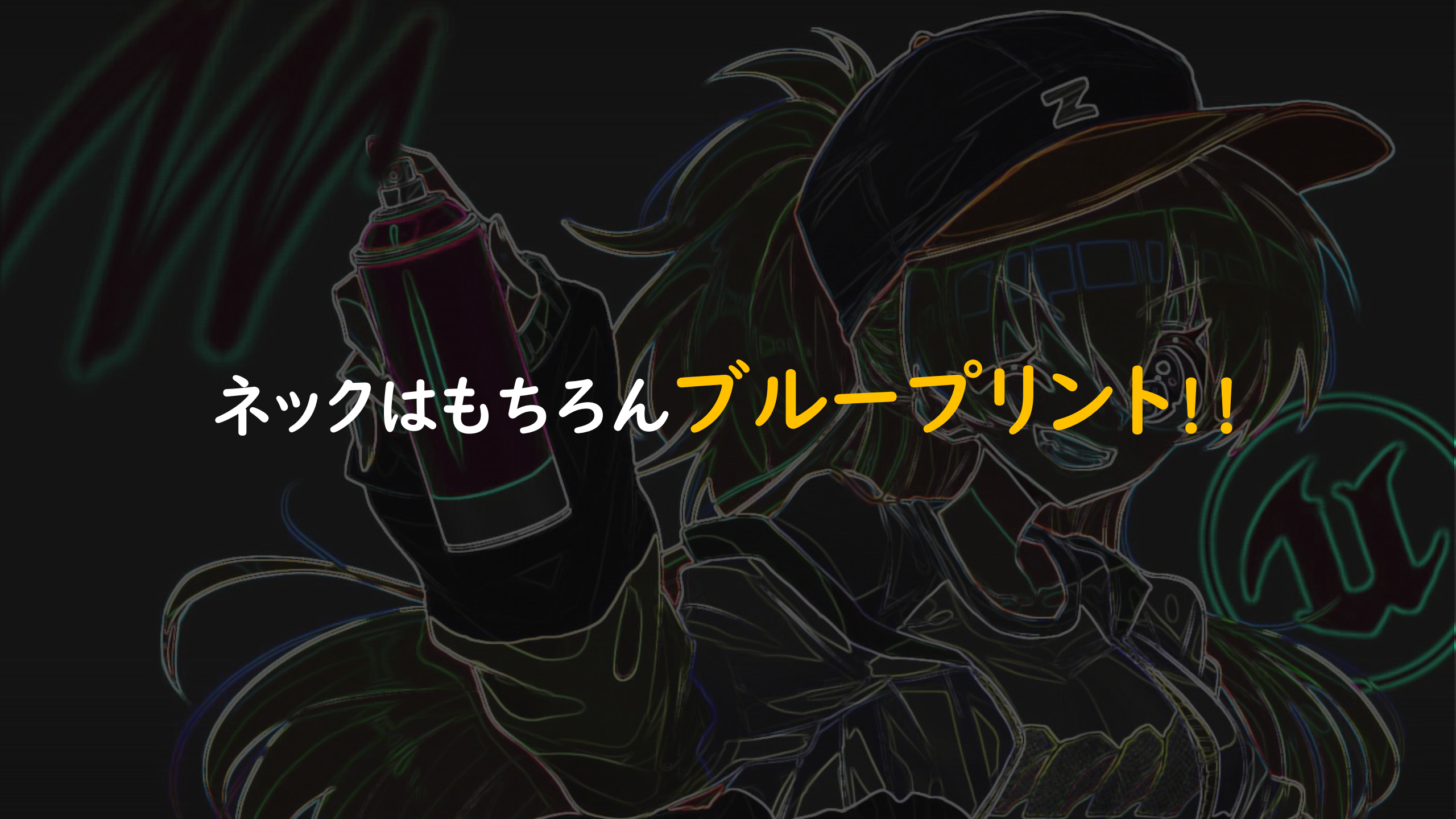
地面が赤いエリアにプレイヤーが入ると
プレイヤーが赤くなる

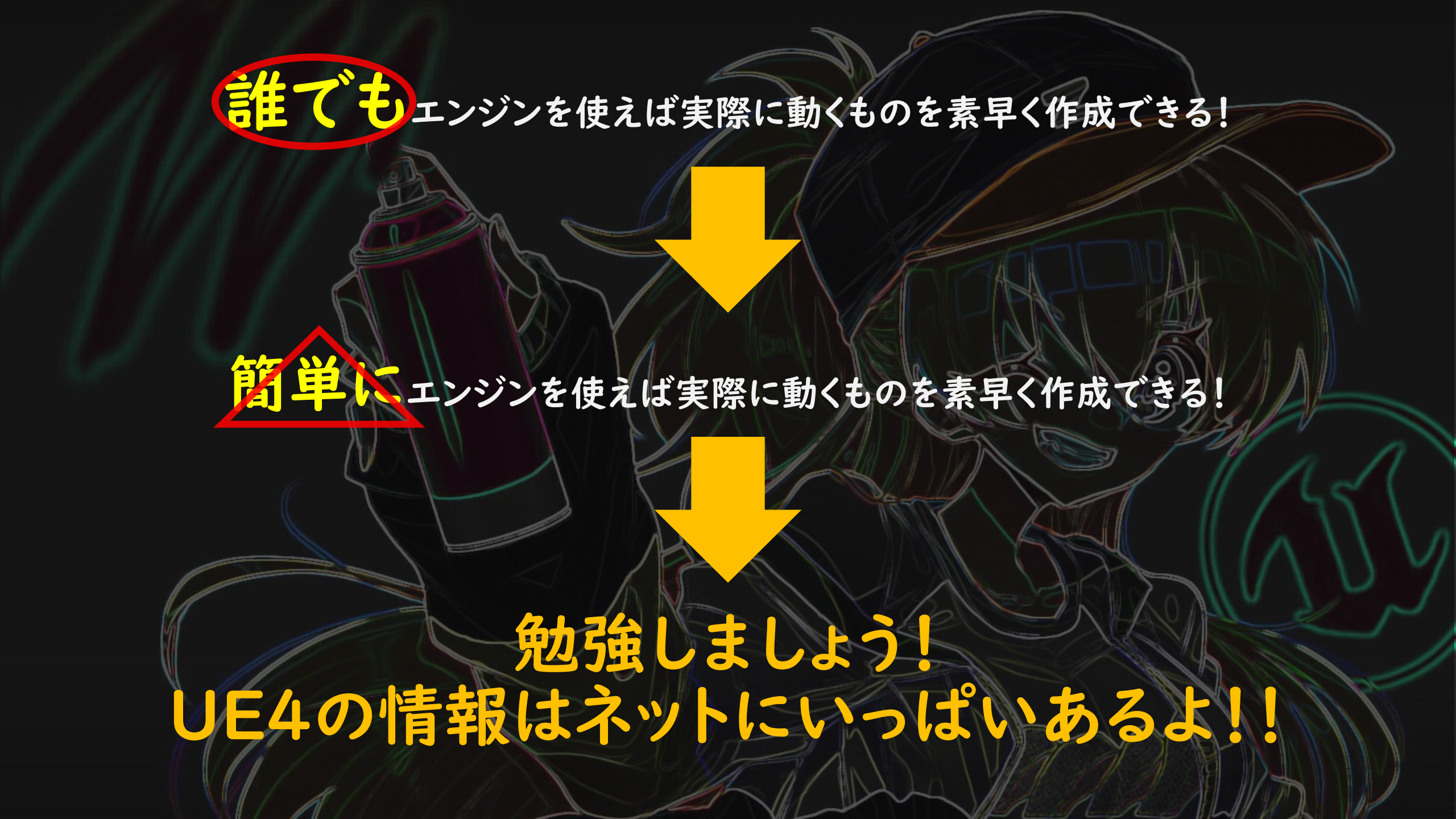
地面が赤いエリアからプレイヤーが出ると
プレイヤーが元に戻る



実際に動く仕様

UE4あるんだから、
動くものをぱっと作ろう！

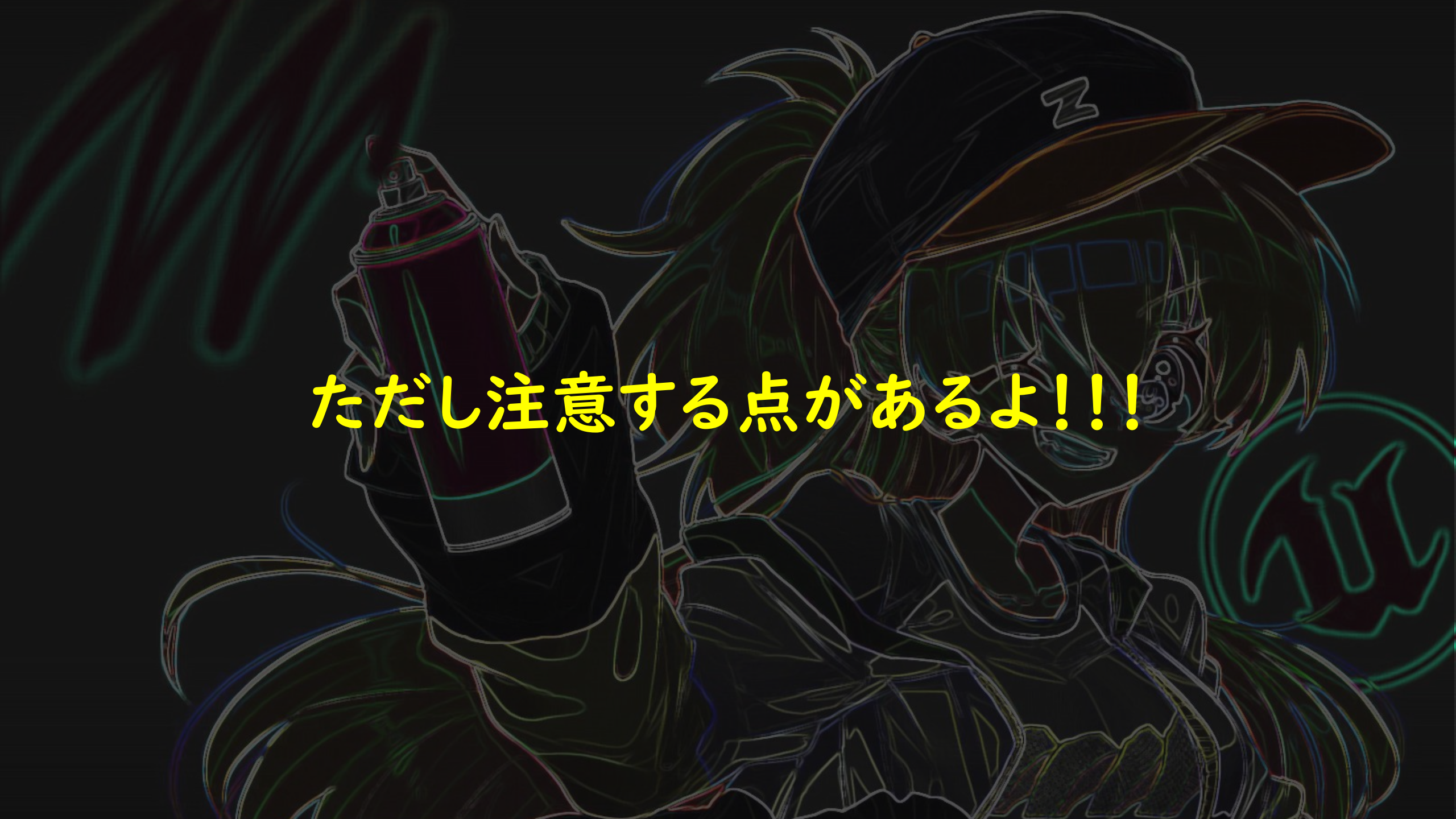

ネックはもちろんブループリント!!


誰でも
エンジンを使えば実際に動くものを素早く作成できる！
簡単に
エンジンを使えば実際に動くものを素早く作成できる！
勉強しましょう！
UE4の情報はネットにいっぱいあるよ！！

ブループリントを使いこなせ！

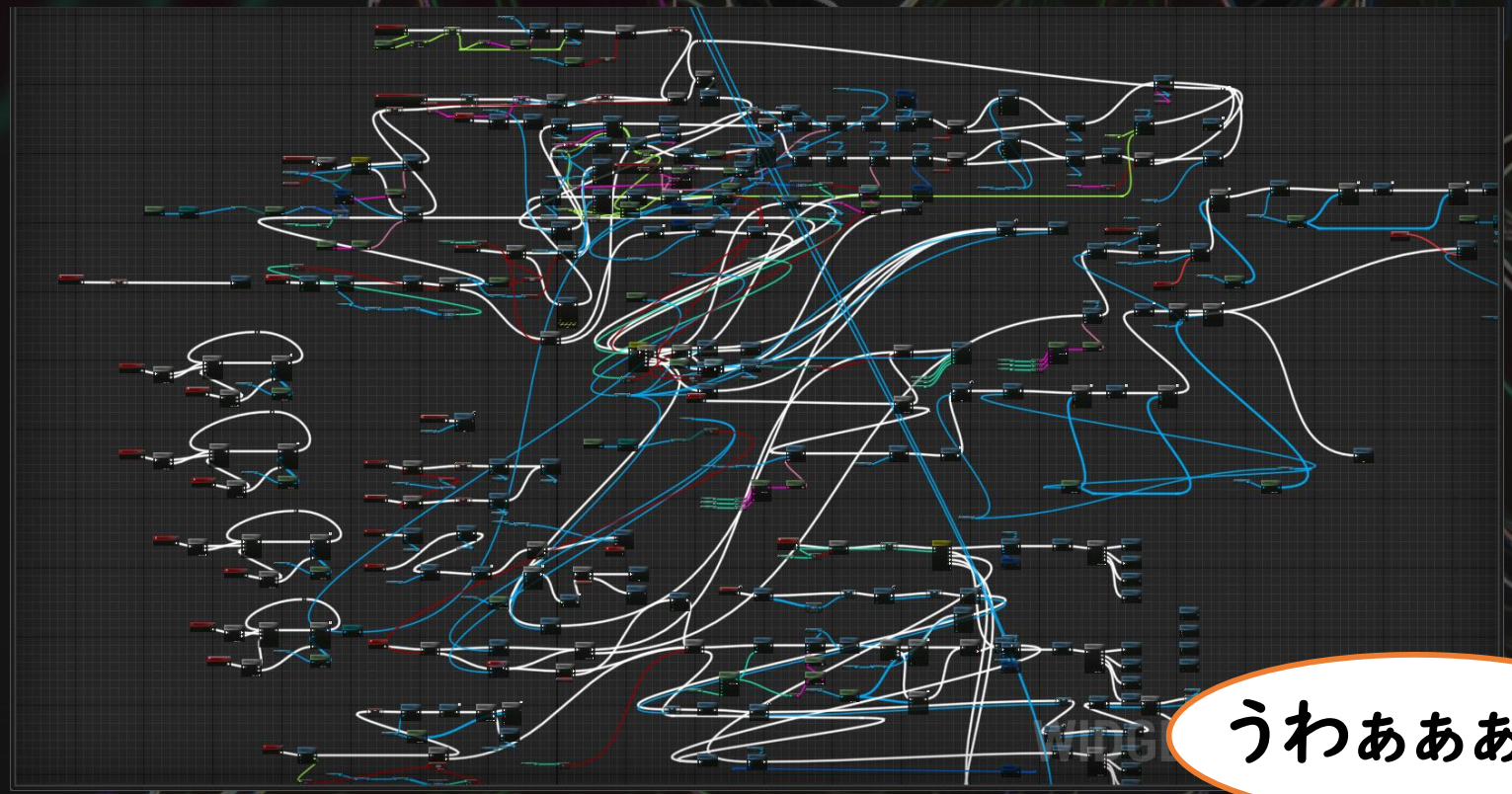
ある程度の設計ができているなら
実際に仮組みしてみたほうが、仕様的な破綻や面白さなどの部分が
プログラマやアーティストの手を使わなくても見ることができる！

※もちろん書類ベースの仕様は必要！
→アクセスのしやすさだったり、フローチャートなんかの一覧性は、
あった方がいいとは思う。
→仮組みが出来上がったら書類にしてやったり、
　パラメータなどのリソースは作成。


ただし注意する点があるよ!!!

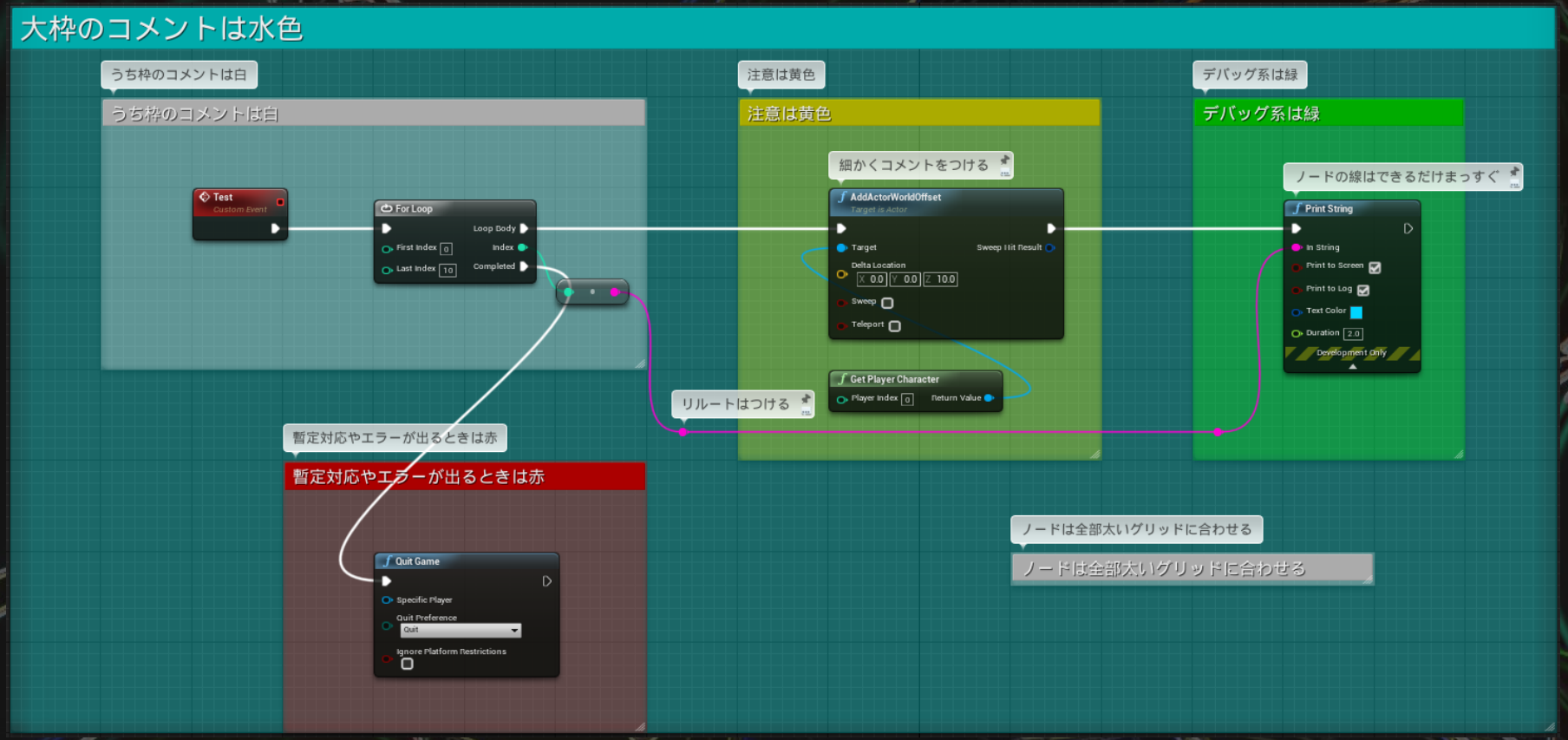
①ブループリントはきれいにを心がける！

・書類でもそうですが、ぱっと見で何がどこにあるかわからない
のは後々に本実装するプログラマだけでなく、自分をも苦しめます



①ブループリントはきれいにを心がける！

自分がブループリントを作成するときには、いくつかルールを設けています。



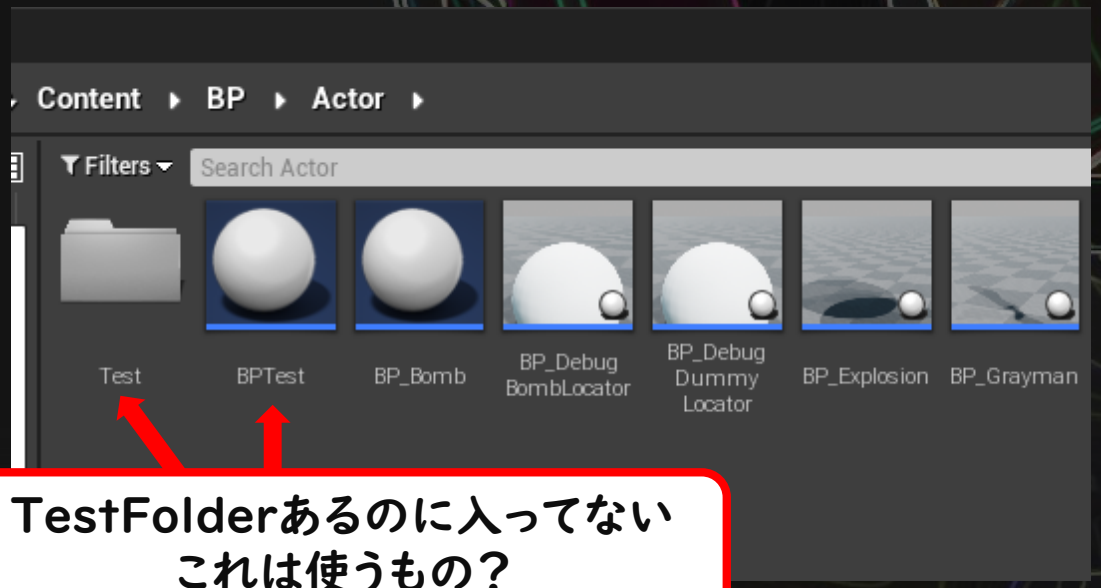
できるだけ誰が見てもわかるようなものになっているのがいいと思います。

※自分が公開しているブログ内にあるサンプルプロジェクトも、大体同じルールで書いていますので参考程度に見てみてください！

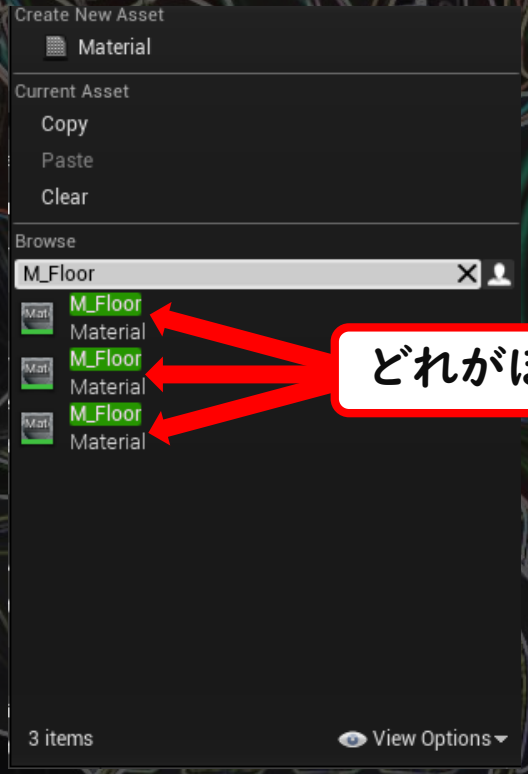
②仮組みのアセットなどの命名・保存場所は仮であることがわかるように！

気づかずに本実装に仮作成したものが組み込まれてしまいます。
命名・階層には十分注意してください！

特に、全く同じ名前があったりするとどちらが使っていいものなのかわからなくなります。



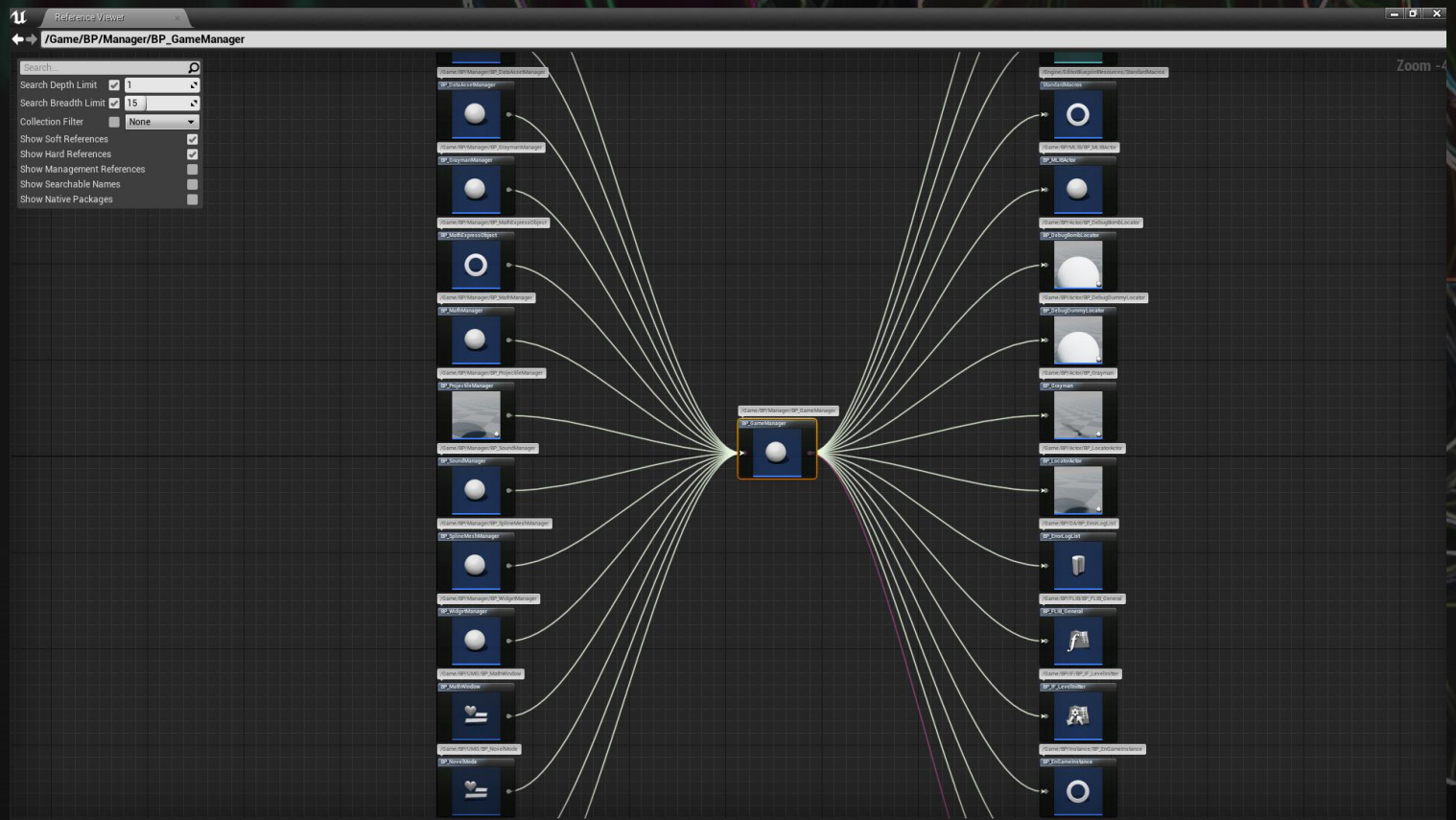
TestFolderあるのに入ってない
これは使うもの？



どれがほんとに使うやつ？

②仮組みのアセットなどの命名・保存場所は仮であることがわかるように！

自分が作ったアセットが本実装されていないか不安な場合は、
アセットを右クリック→「Show Reference Viewer」を選択すると
そのアセットがどこで使われているかがわかります。



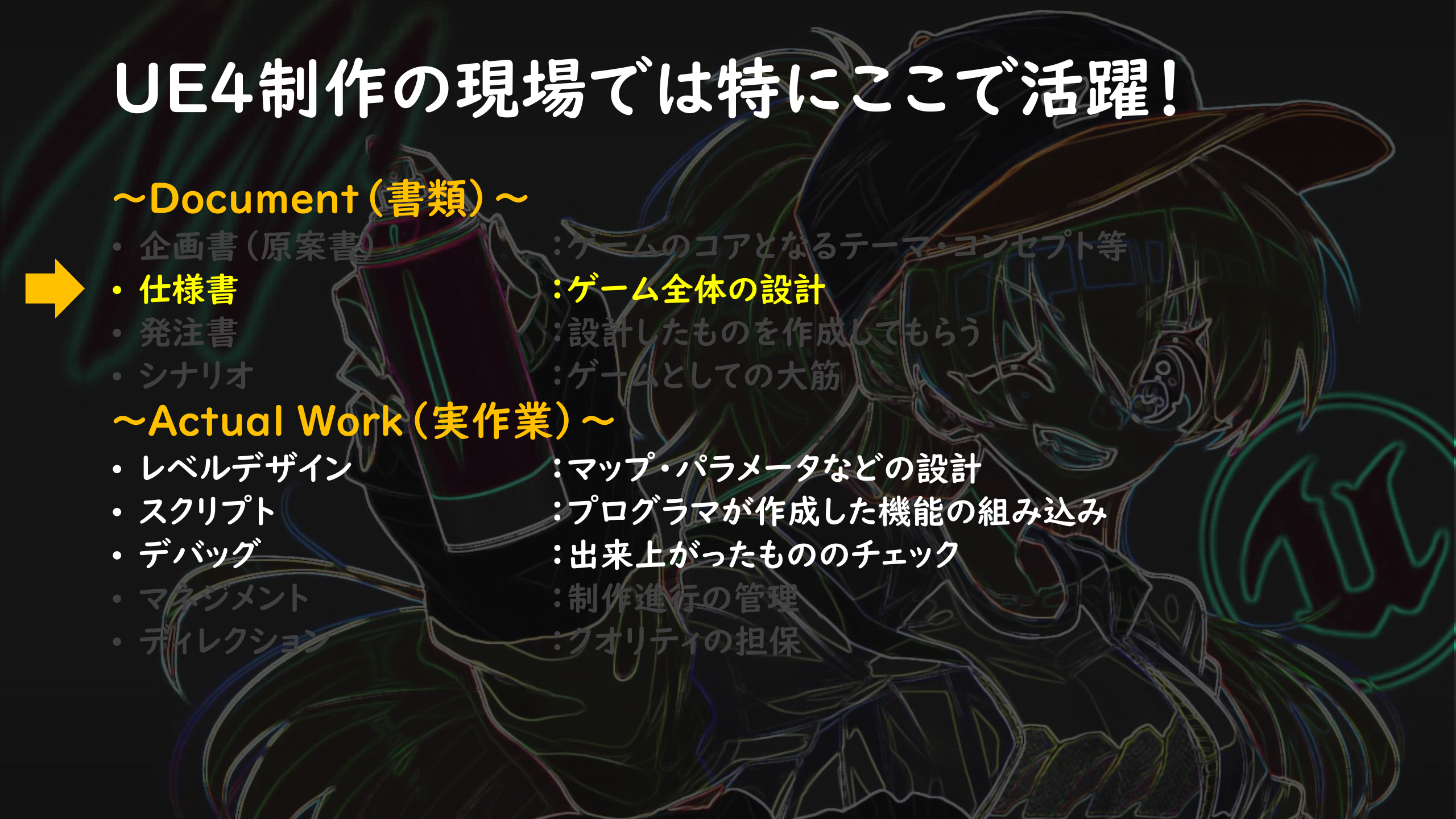
③何でもやっていいわけではない

各プロジェクトやプラットフォームごとに実現できることが決まっています。

安易に作成したりして、後でプログラマに発注する際に実現するのに大幅なコストが必要だったりして現実的ではない仕様になっていたりする場合があります。

ざっくり作ろうと思っているものをプログラマやテクニカルアーティストに相談して、方向性が良さそうなら実際に作ってみましょう。

UE4制作の現場では特にここで活躍！



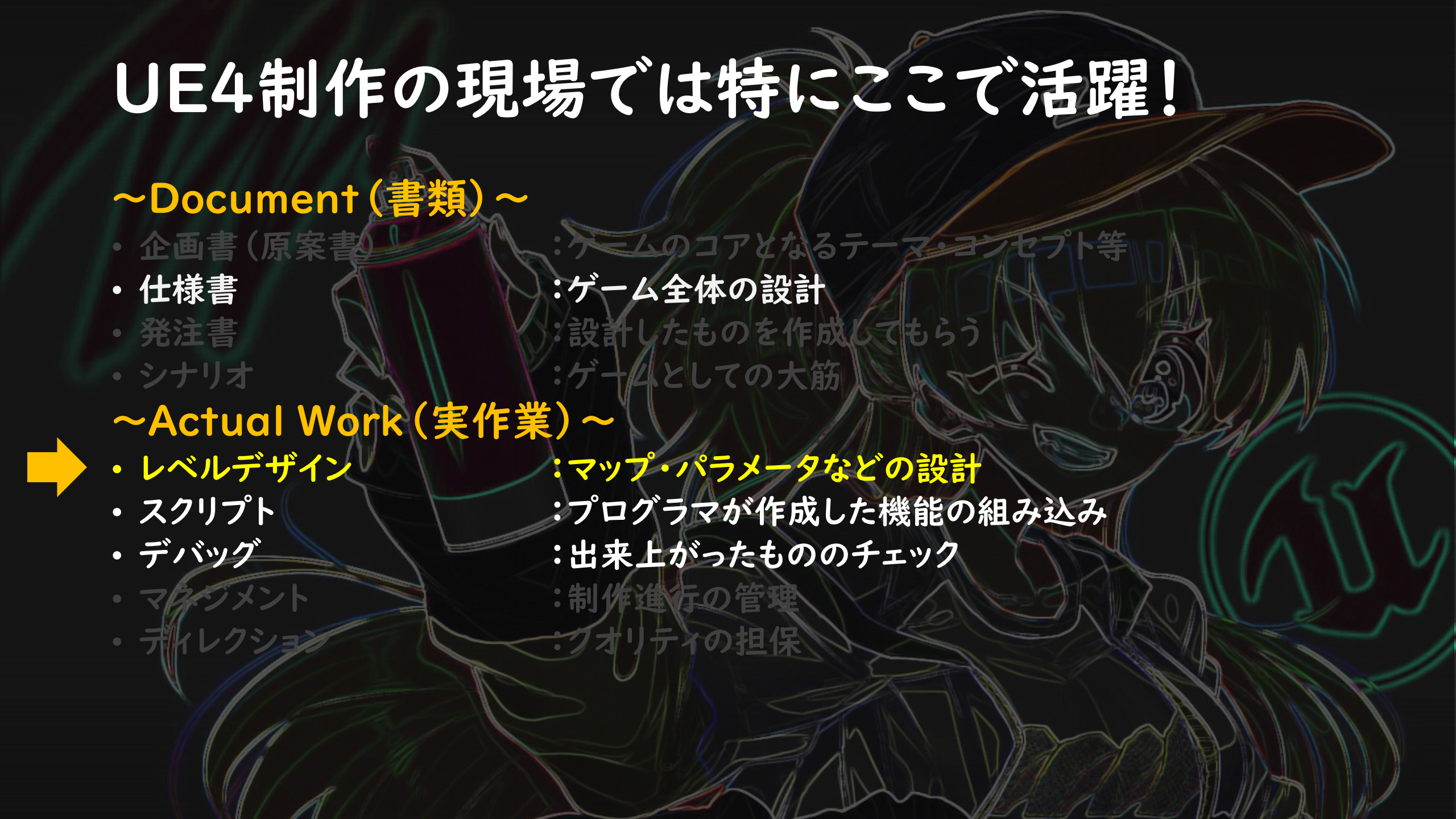
～Document（書類）～

- 企画書（原案書）　：ゲームのコアとなるテーマ・コンセプト等
- **仕様書**　：**ゲーム全体の設計**
- 発注書　：設計したものを作成してもらう
- シナリオ　：ゲームとしての大筋

～Actual Work（実作業）～

- レベルデザイン　：マップ・パラメータなどの設計
- スクリプト　：プログラマが作成した機能の組み込み
- デバッグ　：出来上がったもののチェック
- マネジメント　：制作進行の管理
- ディレクション　：クオリティの担保

UE4制作の現場では特にここで活躍！

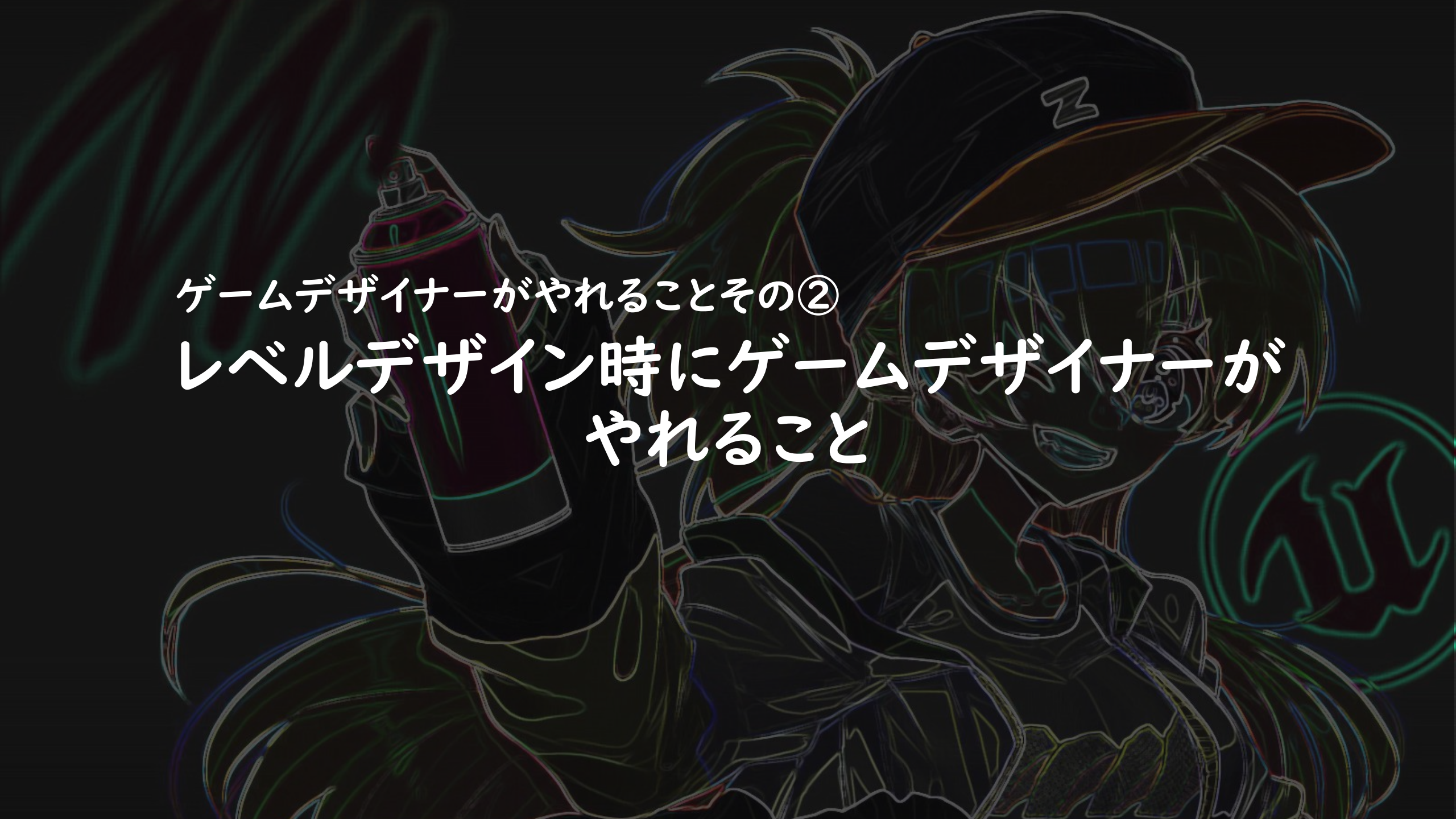


～Document（書類）～

- 企画書（原案書）　：ゲームのコアとなるテーマ・コンセプト等
- 仕様書　：ゲーム全体の設計
- 発注書　：設計したものを作成してもらう
- シナリオ　：ゲームとしての大筋

～Actual Work（実作業）～

- レベルデザイン　：マップ・パラメータなどの設計
- スクリプト　：プログラマが作成した機能の組み込み
- デバッグ　：出来上がったもののチェック
- マネジメント　：制作進行の管理
- ディレクション　：クオリティの担保



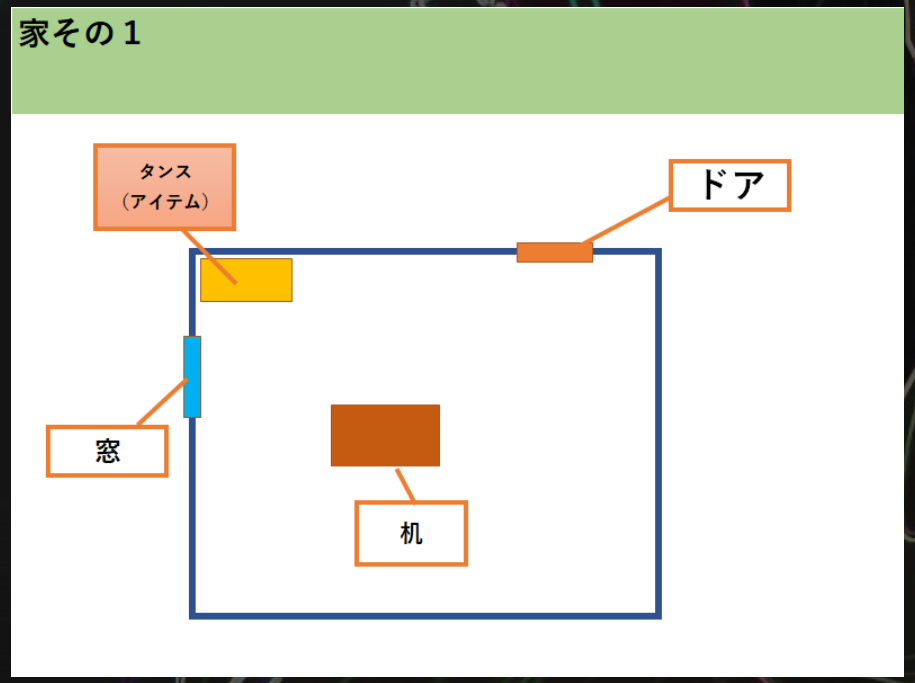
ゲームデザイナーがやれることその②

レベルデザイン時にゲームデザイナーがやれること

やっぱり…
Excelアートやめません？

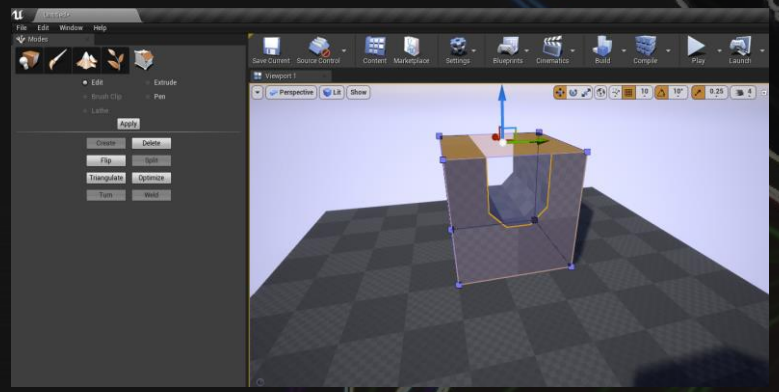
マップ制作

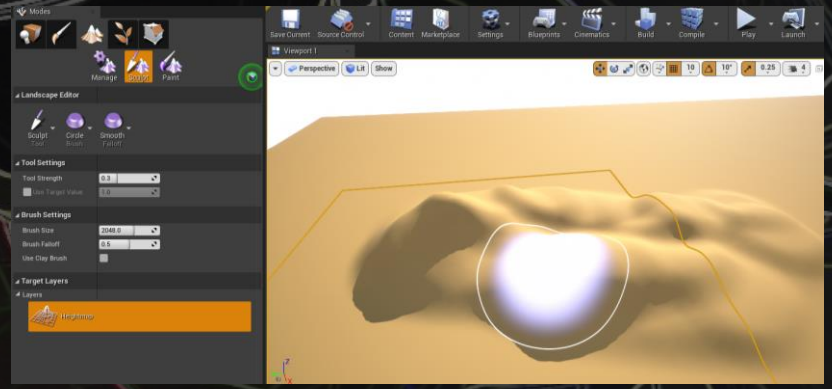
実際のマップを、アーティストが用意してくれたアセットや
グレーボックスを使って組んでみて、サイズ感や導線などを
素早く作成・確認が可能。

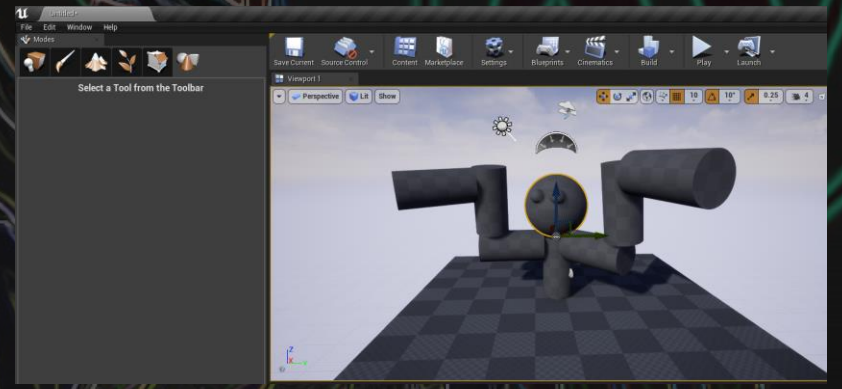


マップ制作

UE4では、プリミティブな3Dオブジェクト以外にも、
「LandScape」や「Geometry Brush」、
4.24でベータ版として入った「Dynamic Mesh Sculpt Tools」
など、DCCツールがなくてもある程度様々なものを
作ることができるでしょう。

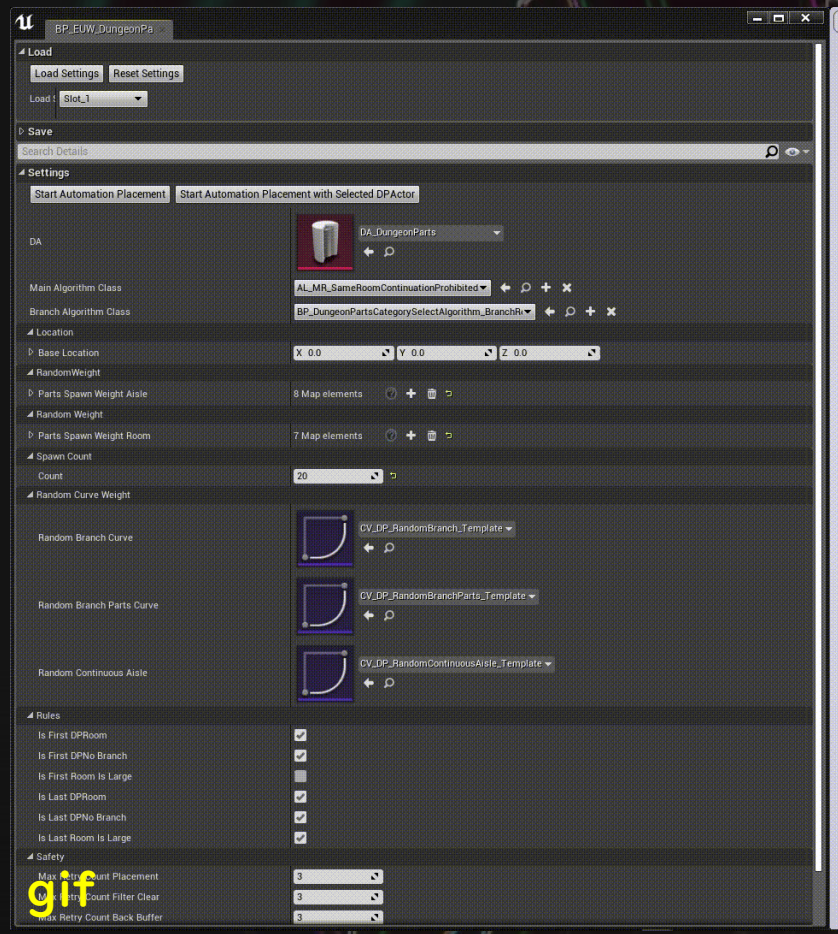


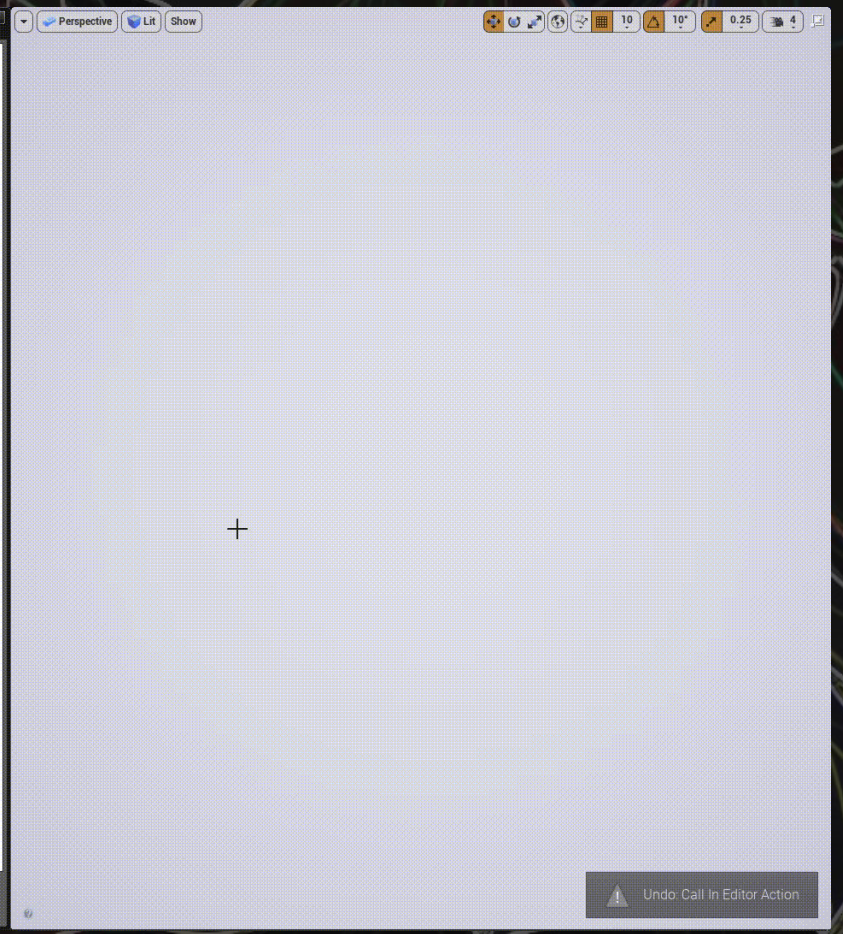




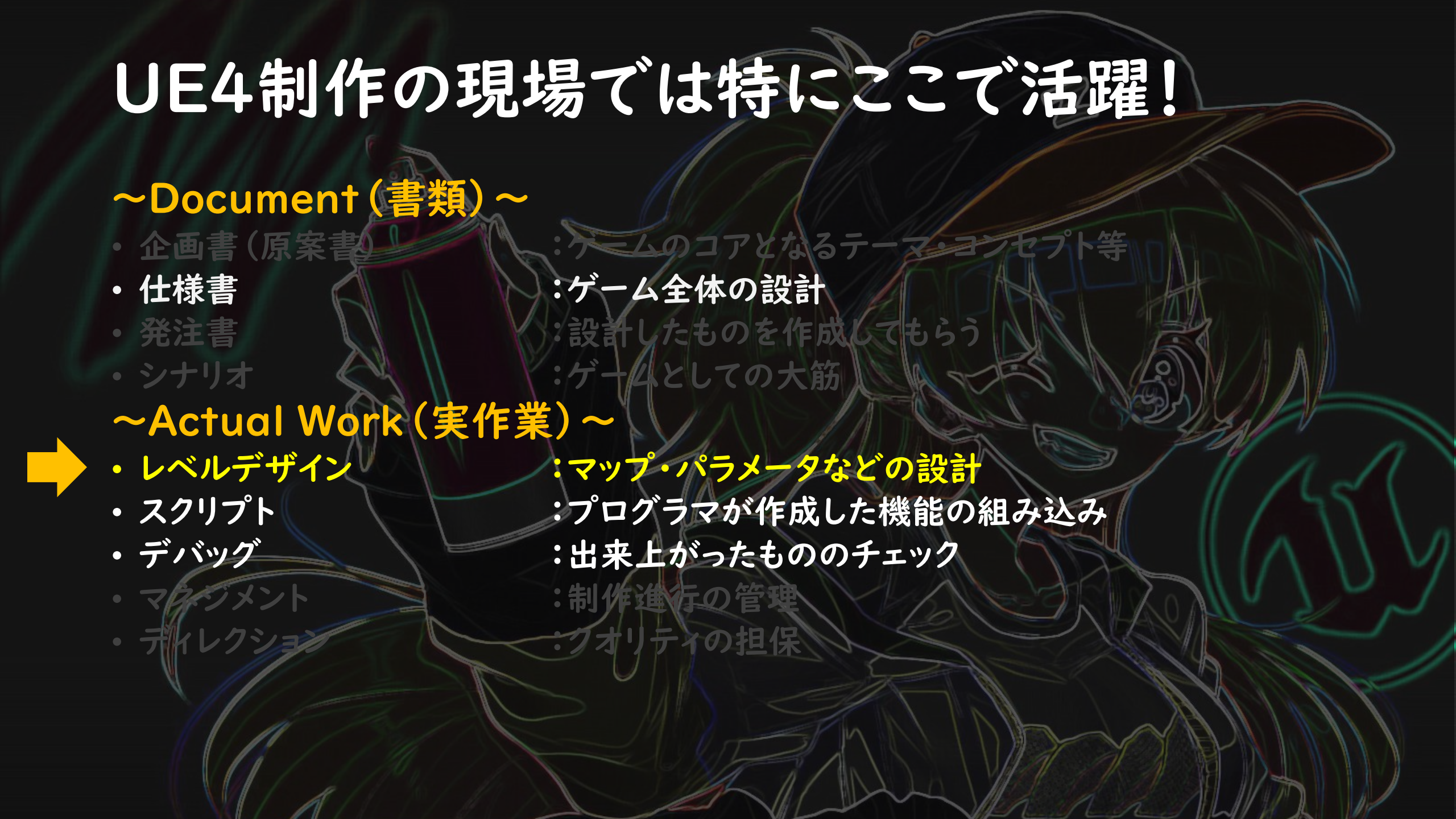
マップ制作

なんなら、レギュレーションなどがはっきりしており、決まった形のアセットを使ってマップを作ったりするなら、自動で生成してくれるような機能を作ってしまうのもありだと思います。（↓ブループリントのみで作成）





UE4制作の現場では特にここで活躍！



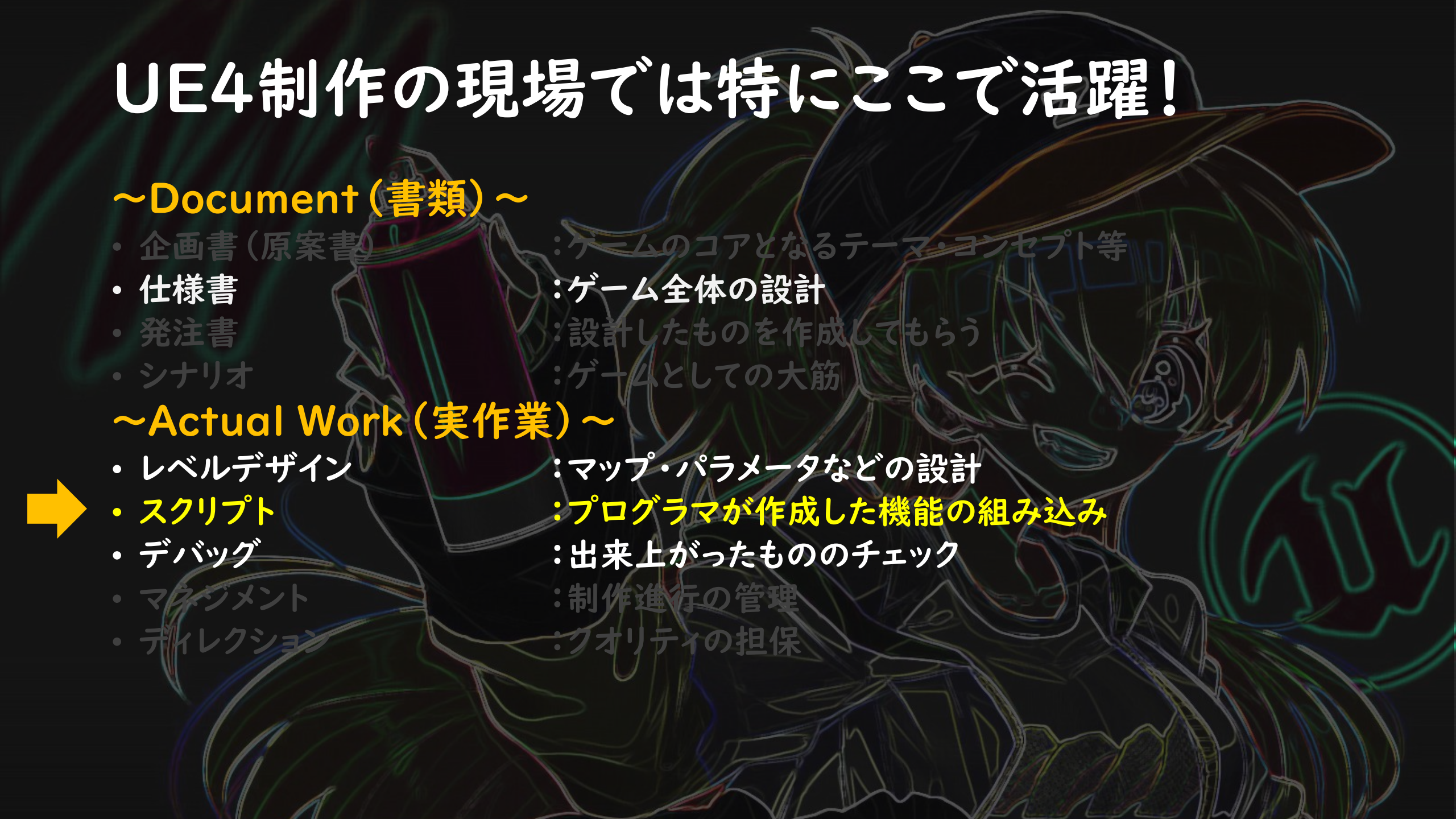
～Document（書類）～

- 企画書（原案書）　：ゲームのコアとなるテーマ・コンセプト等
- 仕様書　：ゲーム全体の設計
- 発注書　：設計したものを作成してもらう
- シナリオ　：ゲームとしての大筋

～Actual Work（実作業）～

- レベルデザイン　：マップ・パラメータなどの設計
- スクリプト　：プログラマが作成した機能の組み込み
- デバッグ　：出来上がったもののチェック
- マネジメント　：制作進行の管理
- ディレクション　：クオリティの担保

UE4制作の現場では特にここで活躍！

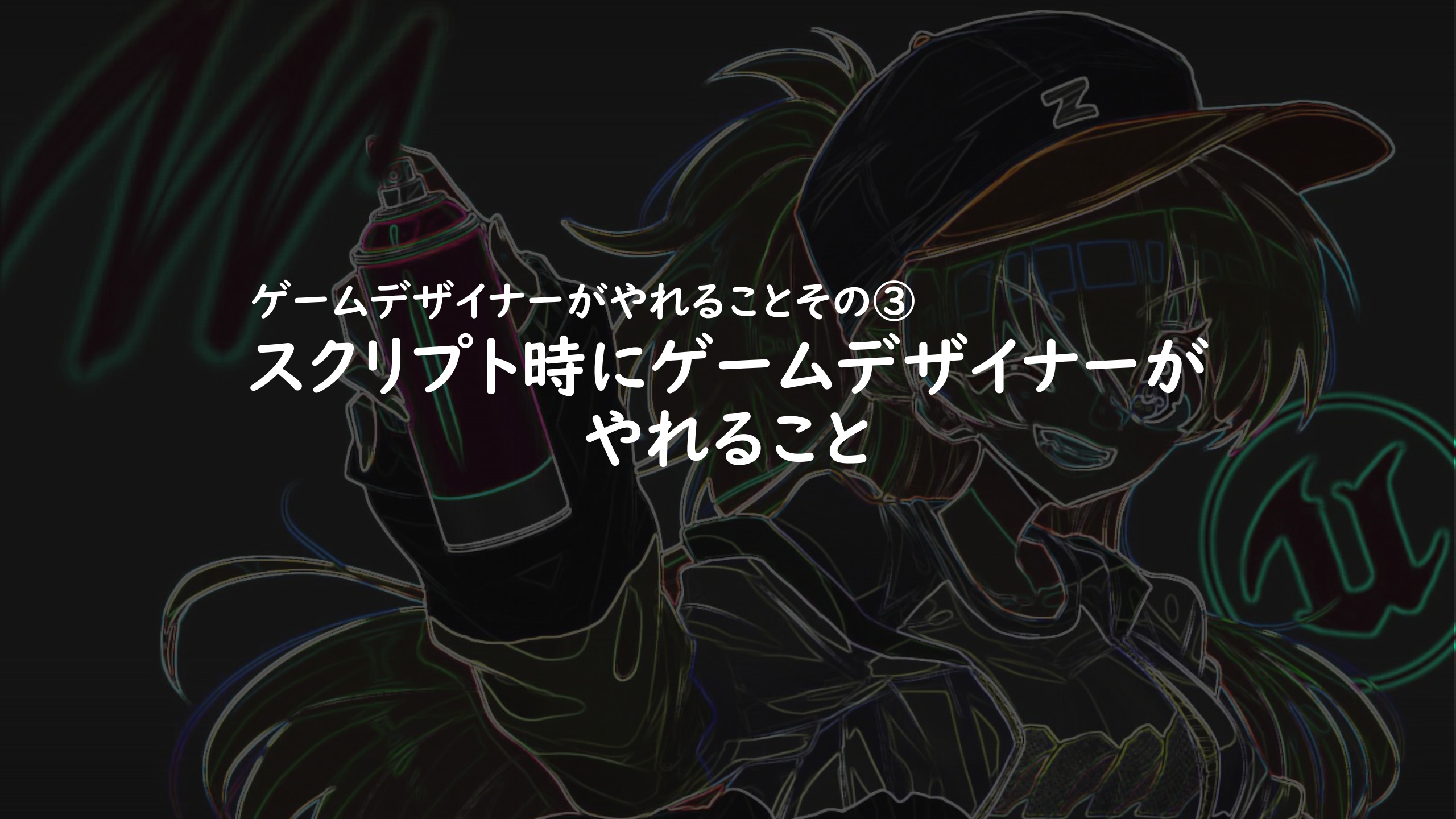


～Document（書類）～

- 企画書（原案書）　：ゲームのコアとなるテーマ・コンセプト等
- 仕様書　：ゲーム全体の設計
- 発注書　：設計したものを作成してもらう
- シナリオ　：ゲームとしての大筋

～Actual Work（実作業）～

- レベルデザイン　：マップ・パラメータなどの設計
- スクリプト　：プログラマが作成した機能の組み込み
- デバッグ　：出来上がったもののチェック
- マネジメント　：制作進行の管理
- ディレクション　：クオリティの担保



ゲームデザイナーがやれることその③

スクリプト時にゲームデザイナーがやれること

UE4のプログラム

プリミティブ（基本的）な機能

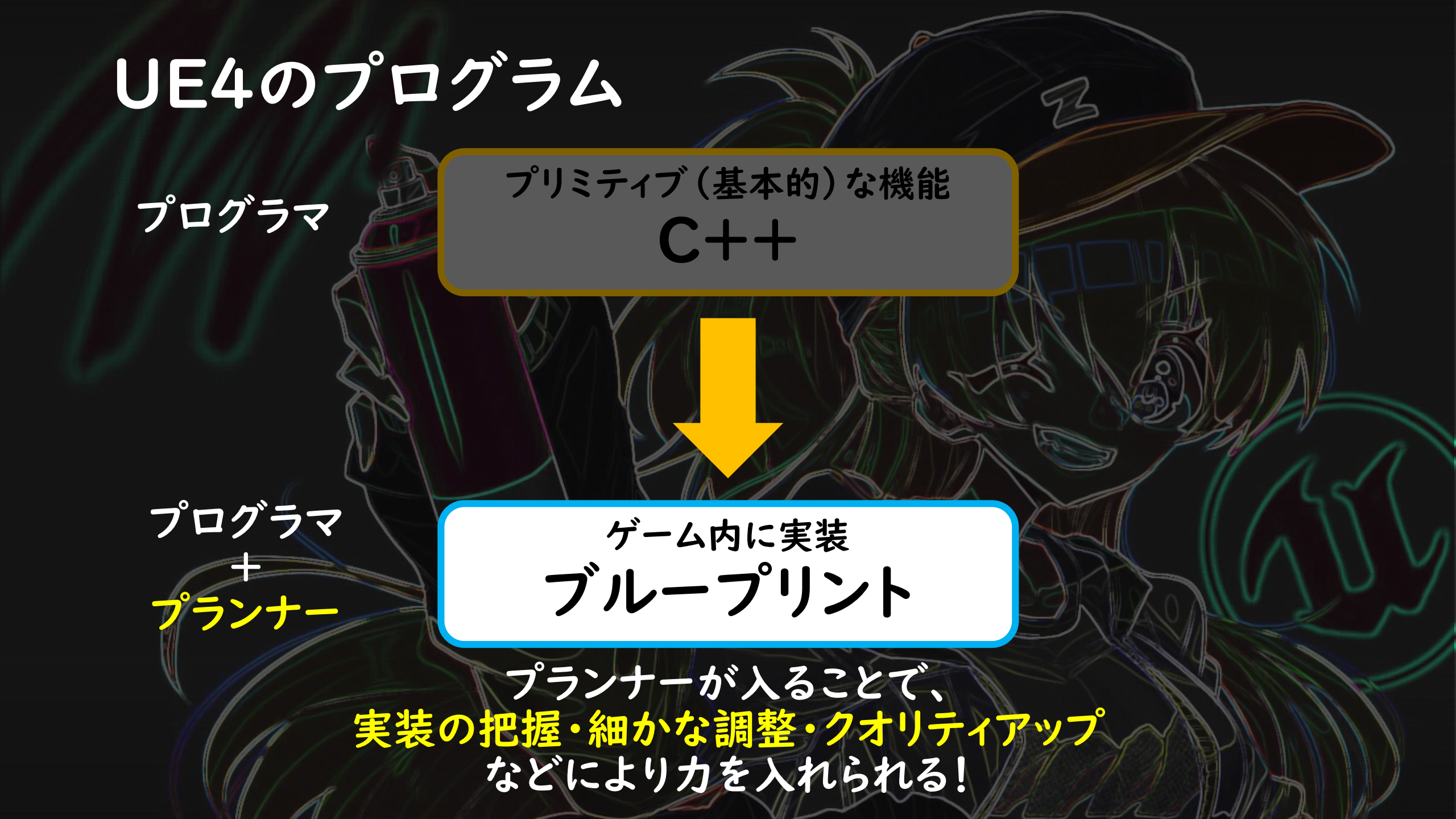
C++

C++をもとにして作成
（継承）

ゲーム内に実装

ブループリント

UE4のプログラム



プログラマ

プリミティブ（基本的）な機能
C++

プログラマ
＋
プランナー

ゲーム内に実装
ブループリント

プランナーが入ることで、
実装の把握・細かな調整・クオリティアップ
などにより力を入れられる！

UE4のプログラム（簡単なもの）

プログラマ

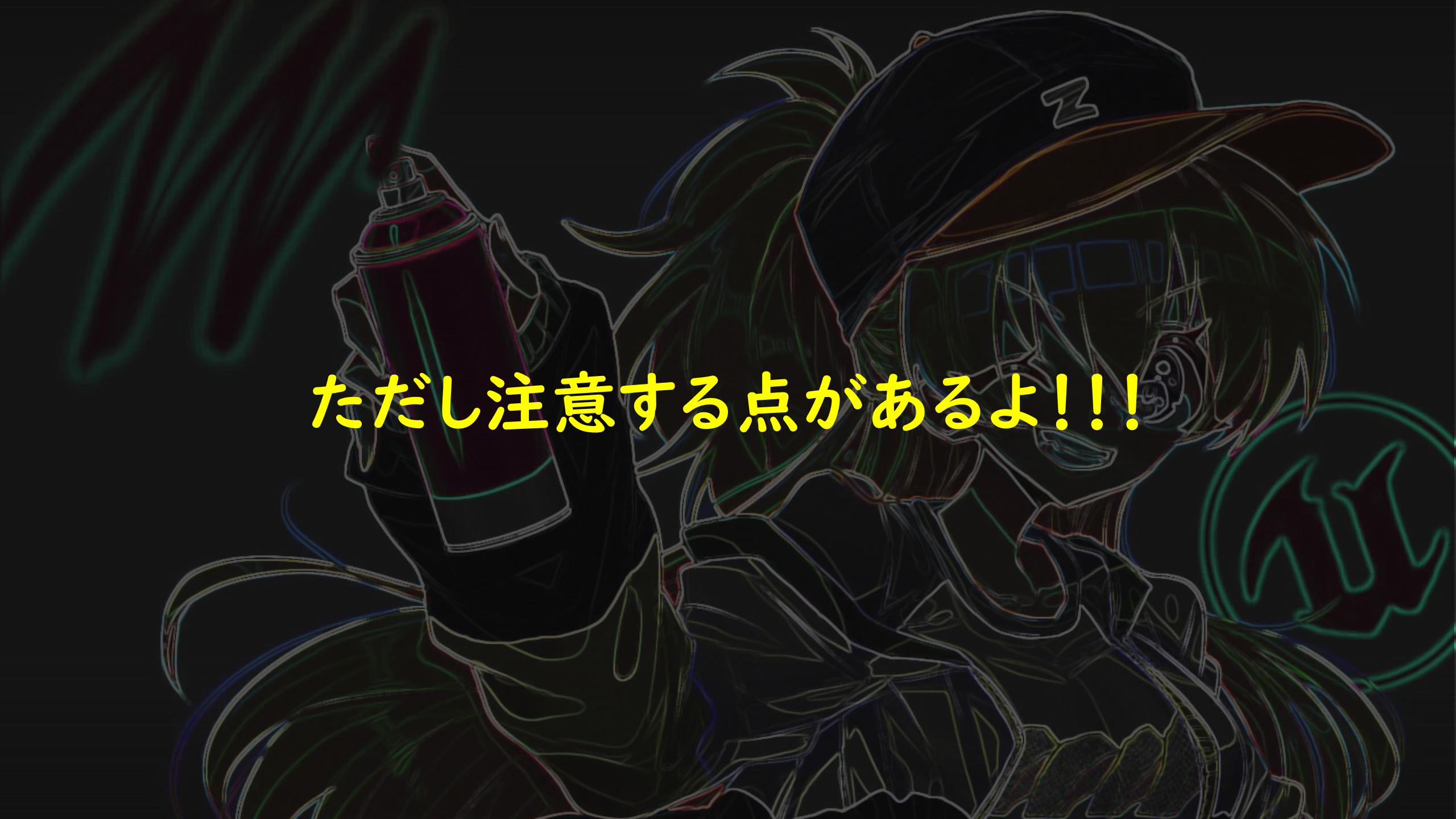
~~プリミティブ（基本的）な機能~~
~~C++~~

単純な機能で処理負荷などを
気にしなかったり
他に依存しないならBPだけで作ることも！

プランナー

ゲーム内に実装
ブループリント

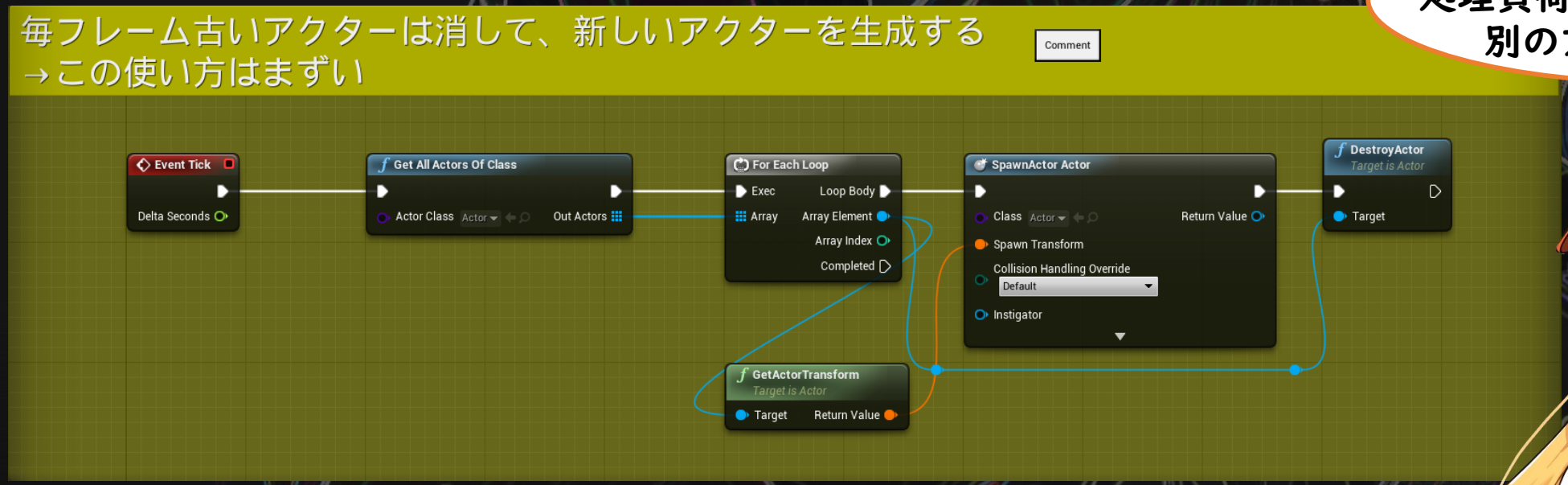
簡単なものなら、プランナーだけでも組める！
※仕様作成でも言いましたが、デタラメに書くのはダメ!!
作成したら、プログラマさんにレビューしてもらうなどしよう！


ただし注意する点があるよ!!!

①重い処理は多様しない

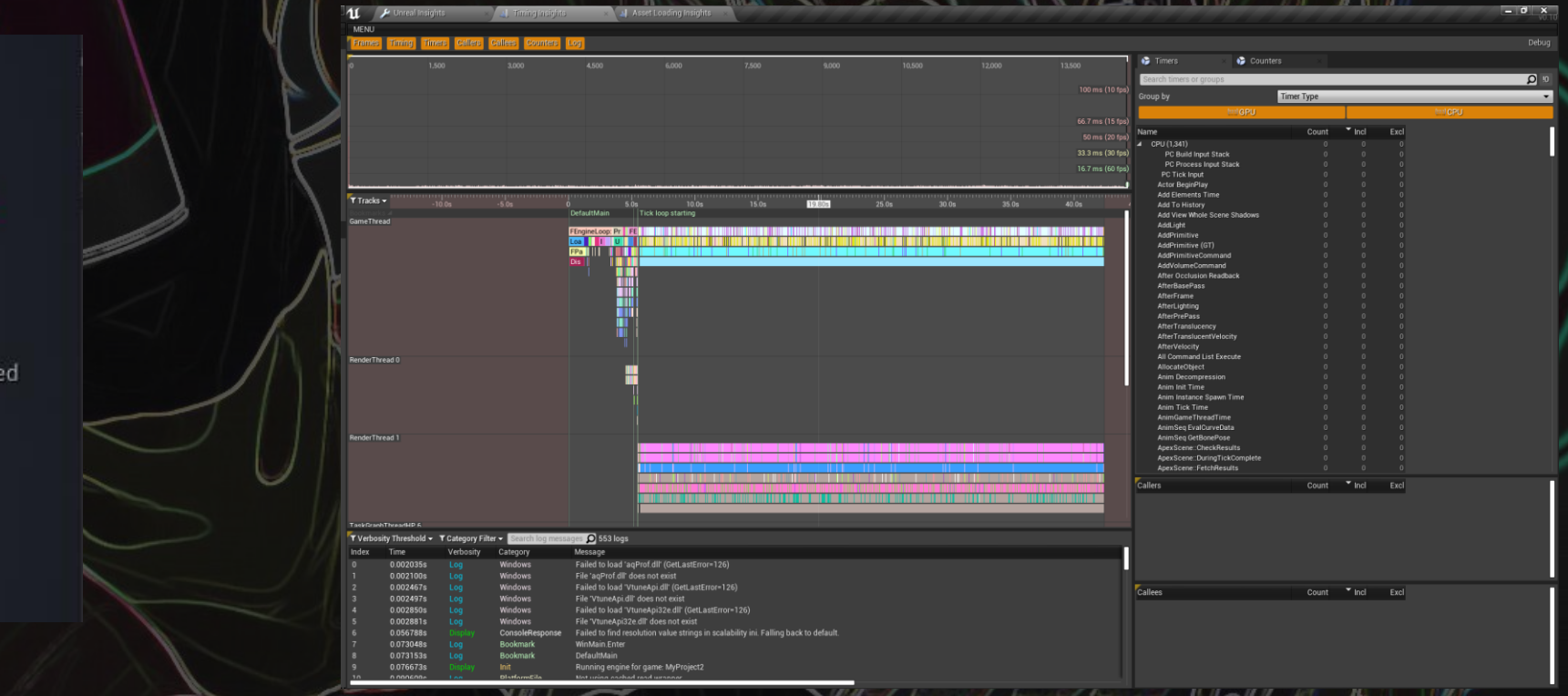
- 最初に言いますが、**使ってはいけないわけではありません。**
  毎フレーム処理をおこなうTickやGet All Actorなどのシークが多い関数、
  またはSpawnやDestroyを繰り返すことでおこるGCスパイクをさけるため、
  これらを**高頻度で使う**のは避けましょう。
  ゲームの処理が重くなり、画面がカクカクするようになったりします。

  使う場合はプログラマに相談し、問題がなければ使いましょう。



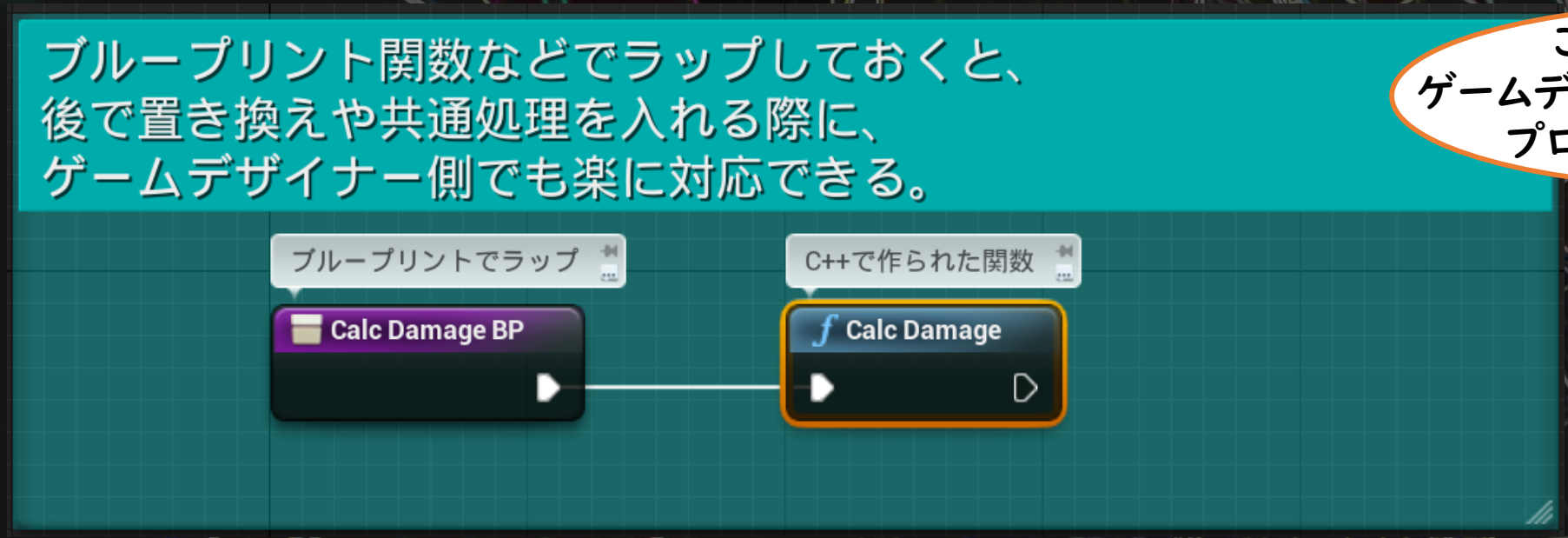
①重い処理は多様しない

- もし処理が気になるなら、コンソールコマンドで「Stat FPS」などのStatコマンドや、最近使えるようになった「Unreal Insight」などのプロファイル機能を使って、処理を入れる前と入れた後でどれぐらいの差が出るか、確認してみよう！明らかに数値が落ちたりするようなら、処理を改善したり、C++に処理を書いてもらうようにプログラマに相談しよう！（パッケージの種類やプラットフォームでも処理負荷は変わっていきます）

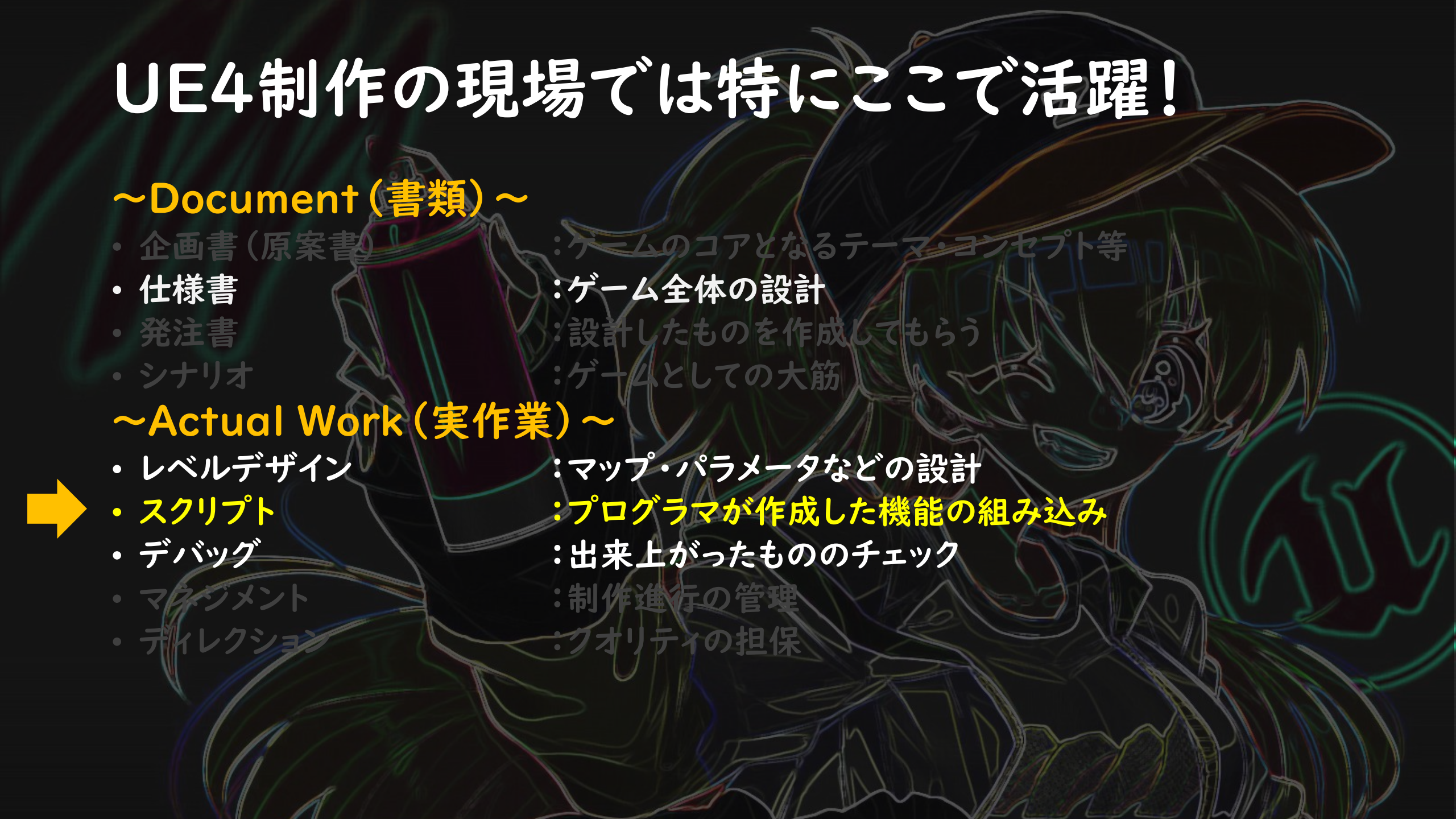


②C++で作った関数はBPでラップ

- C++で作られた関数をそのまま使うより、ブループリントでラップした関数を使う方が、後々変更対応などが楽になります。
  ゲームデザイナーが使うのは「～～～BP」と名前の付いた関数だけと言う風な規則にすることで、よりブループリントの管理をしやすくなり、バグなどの対応も容易になるかと思います。



UE4制作の現場では特にここで活躍！



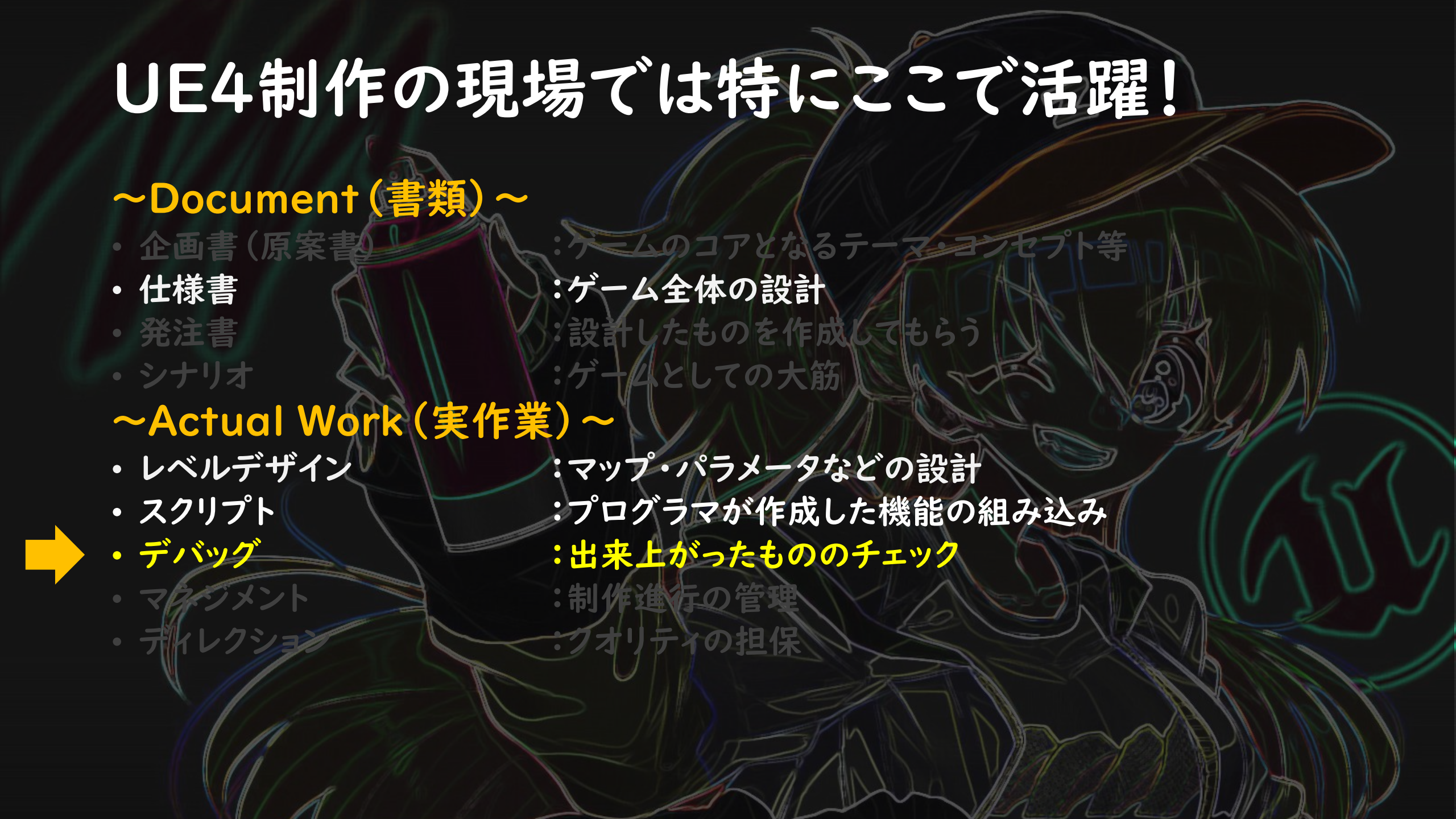
～Document（書類）～

- 企画書（原案書）　：ゲームのコアとなるテーマ・コンセプト等
- 仕様書　：ゲーム全体の設計
- 発注書　：設計したものを作成してもらう
- シナリオ　：ゲームとしての大筋

～Actual Work（実作業）～

- レベルデザイン　：マップ・パラメータなどの設計
- スクリプト　：プログラマが作成した機能の組み込み
- デバッグ　：出来上がったもののチェック
- マネジメント　：制作進行の管理
- ディレクション　：クオリティの担保

UE4制作の現場では特にここで活躍！

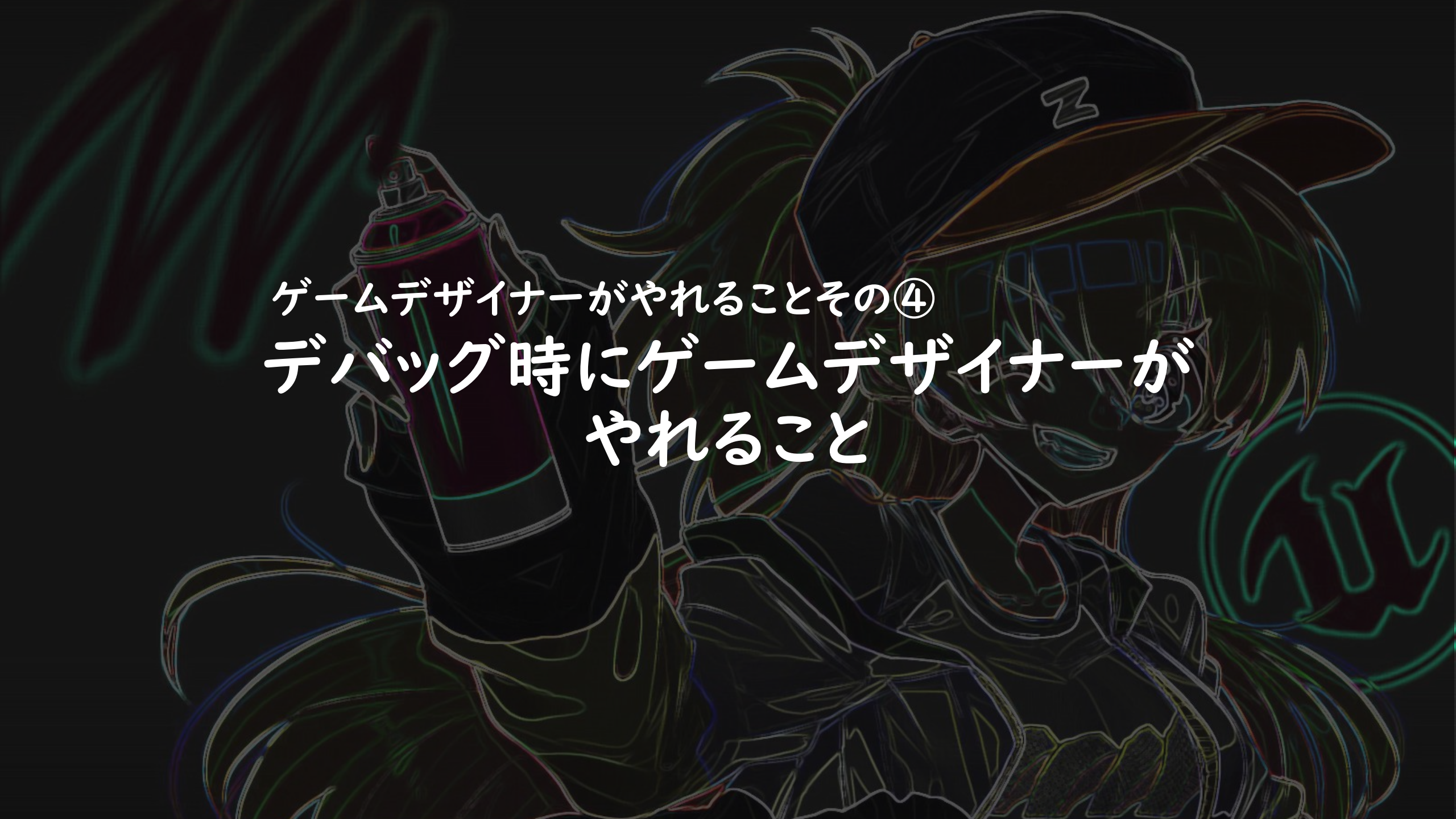


～Document（書類）～

- 企画書（原案書）　：ゲームのコアとなるテーマ・コンセプト等
- 仕様書　：ゲーム全体の設計
- 発注書　：設計したものを作成してもらう
- シナリオ　：ゲームとしての大筋

～Actual Work（実作業）～

- レベルデザイン　：マップ・パラメータなどの設計
- スクリプト　：プログラマが作成した機能の組み込み
- デバッグ　：出来上がったもののチェック
- マネジメント　：制作進行の管理
- ディレクション　：クオリティの担保



ゲームデザイナーがやれることその④

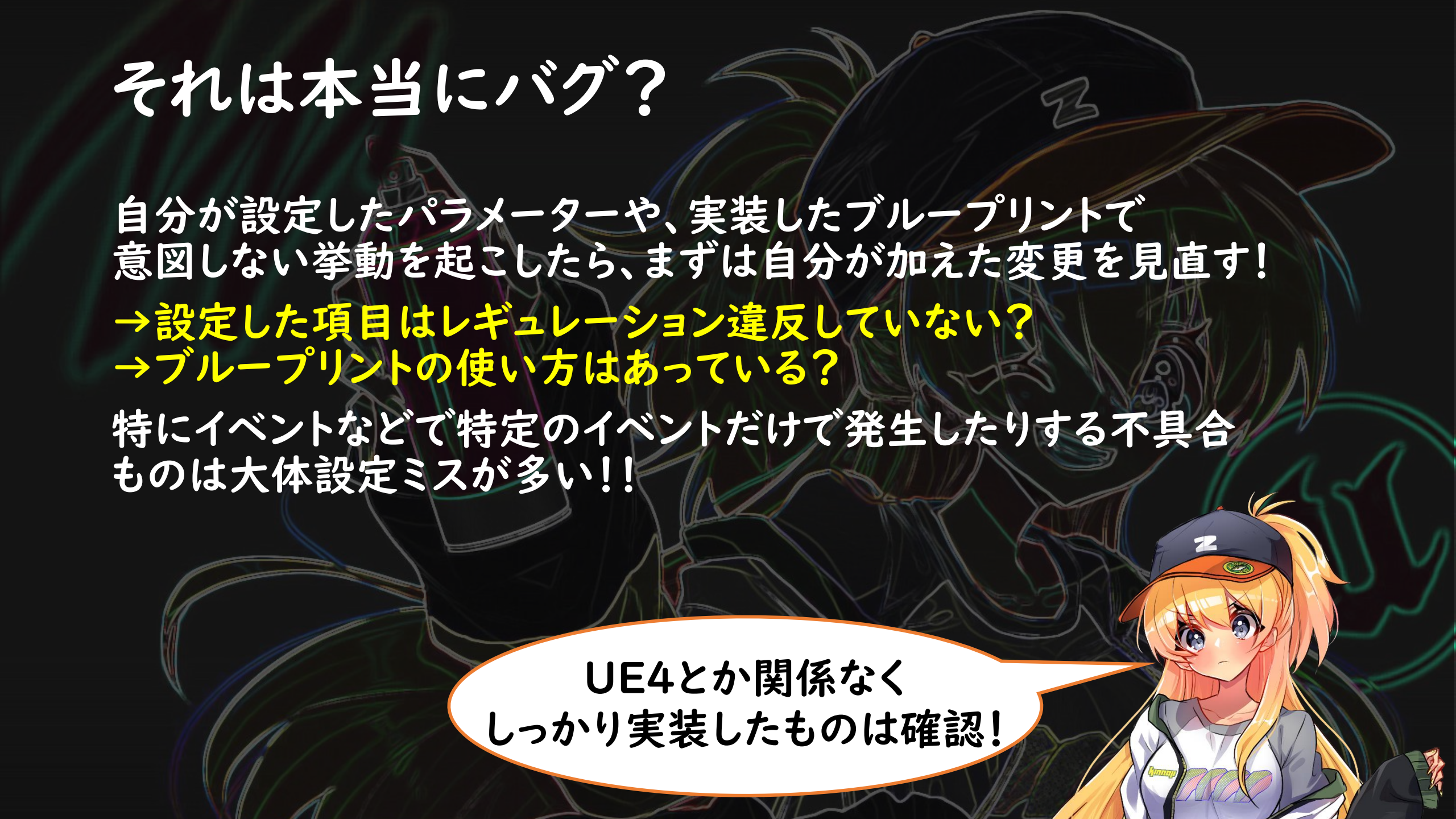
デバッグ時にゲームデザイナーがやれること

それは本当にバグ？

自分が設定したパラメーターや、実装したブループリントで
意図しない挙動を起こしたら、まずは自分が加えた変更を見直す！

→設定した項目はレギュレーション違反していない？
→ブループリントの使い方はあっている？

特にイベントなどで特定のイベントだけで発生したりする不具合
ものは大体設定ミスが多い！！

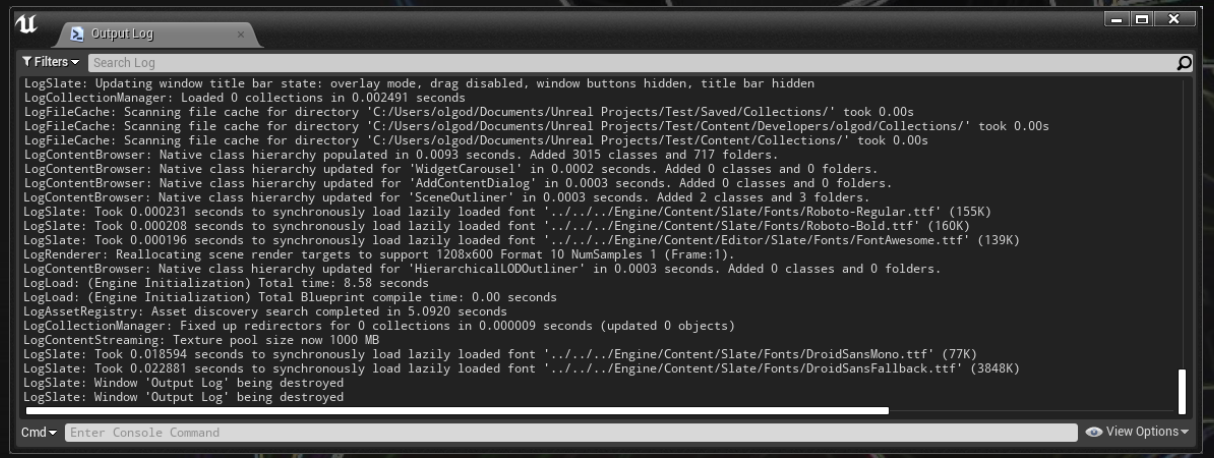


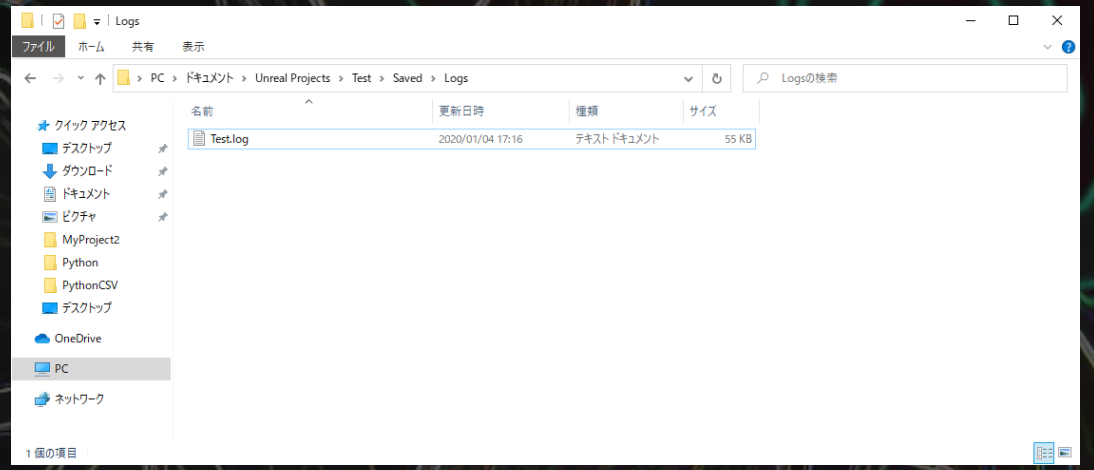
それは本当にバグ？

Editorなら「Output Log」で出力されるログを確認できます。

大抵はプログラマさんがエラーチェック用にログを出力したりしてくれているので、不具合が起きても一度確認して自分なりに調べて見ましょう。

また、LogはProject階層の「/Saved/Logs/」以下に保存されるので、テキストでも確認できます。





それは本当にバグ？

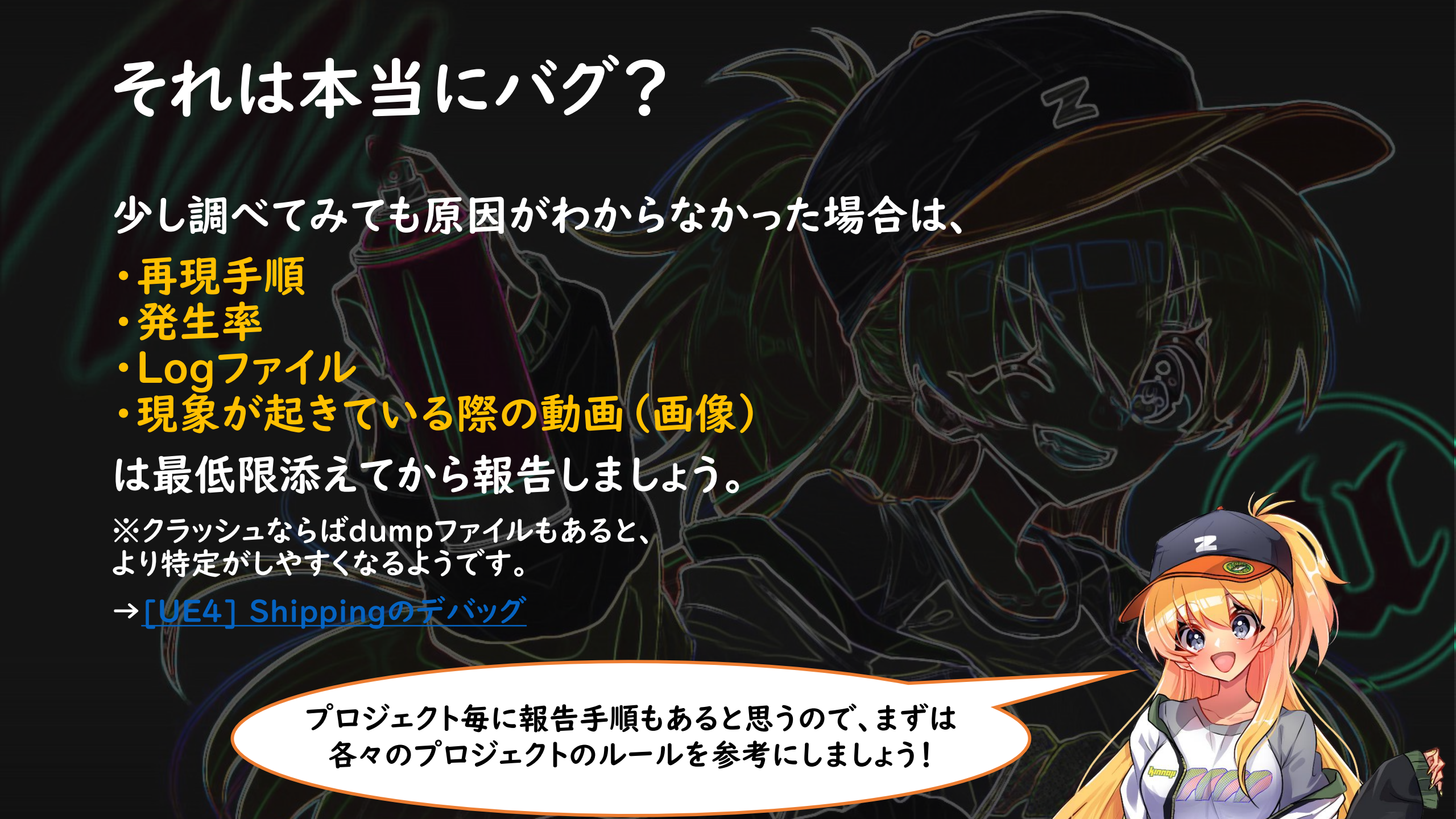
少し調べてみても原因がわからなかった場合は、

- 再現手順
- 発生率
- Logファイル
- 現象が起きている際の動画（画像）

は最低限添えてから報告しましょう。

※クラッシュならばdumpファイルもあると、
より特定がしやすくなるようです。

→[UE4] Shippingのデバッグ



コンソールコマンド

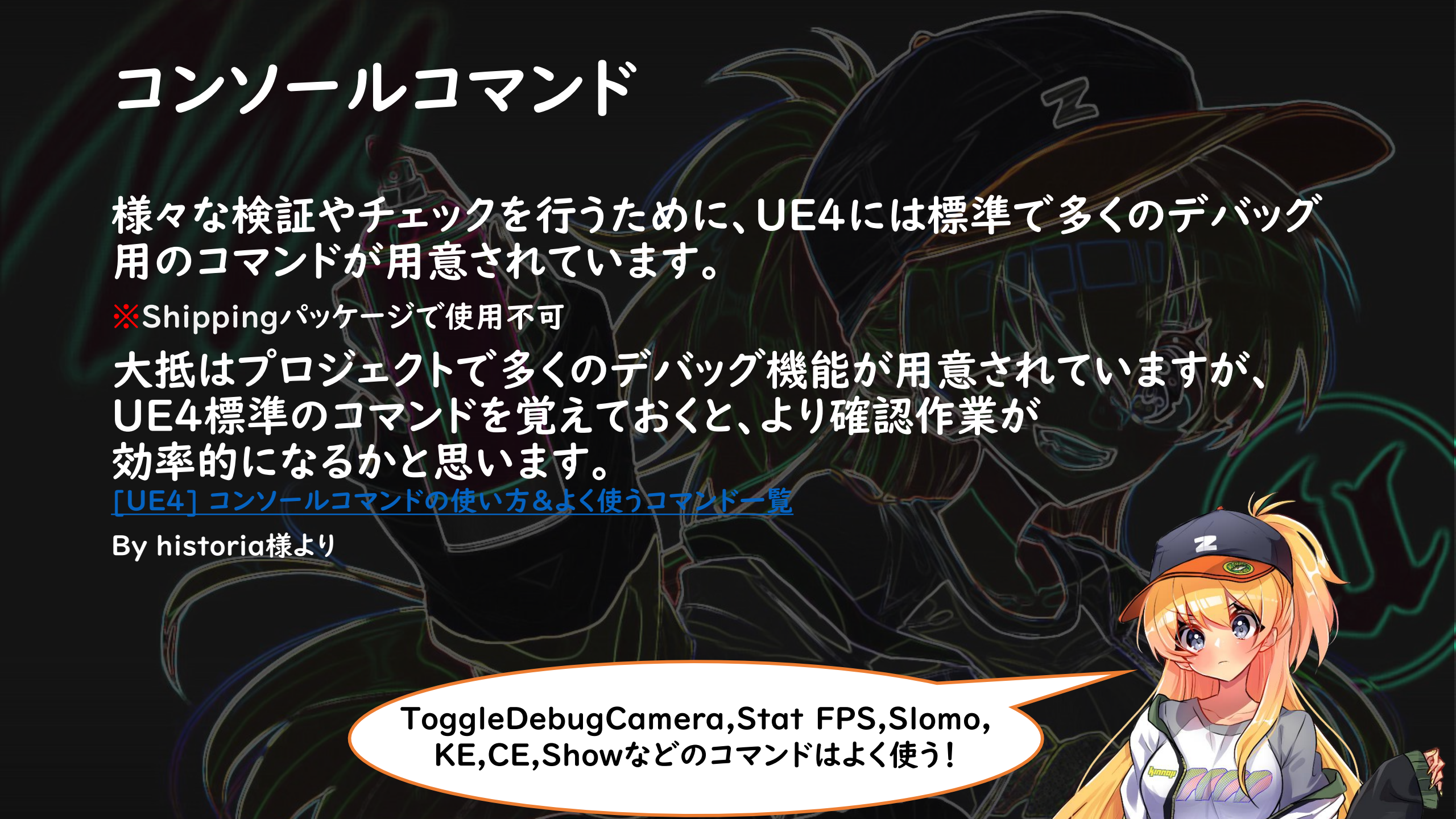
様々な検証やチェックを行うために、UE4には標準で多くのデバッグ用のコマンドが用意されています。

※Shippingパッケージで使用不可

大抵はプロジェクトで多くのデバッグ機能が用意されていますが、
UE4標準のコマンドを覚えておくと、より確認作業が
効率的になるかと思います。

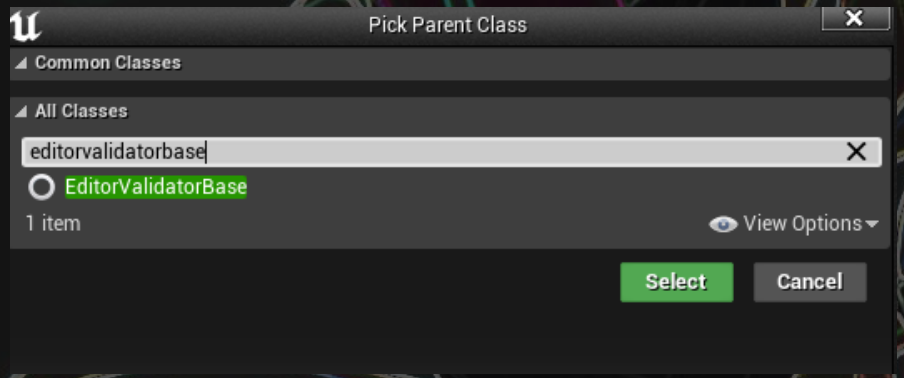
[UE4] コンソールコマンドの使い方＆よく使うコマンド一覧

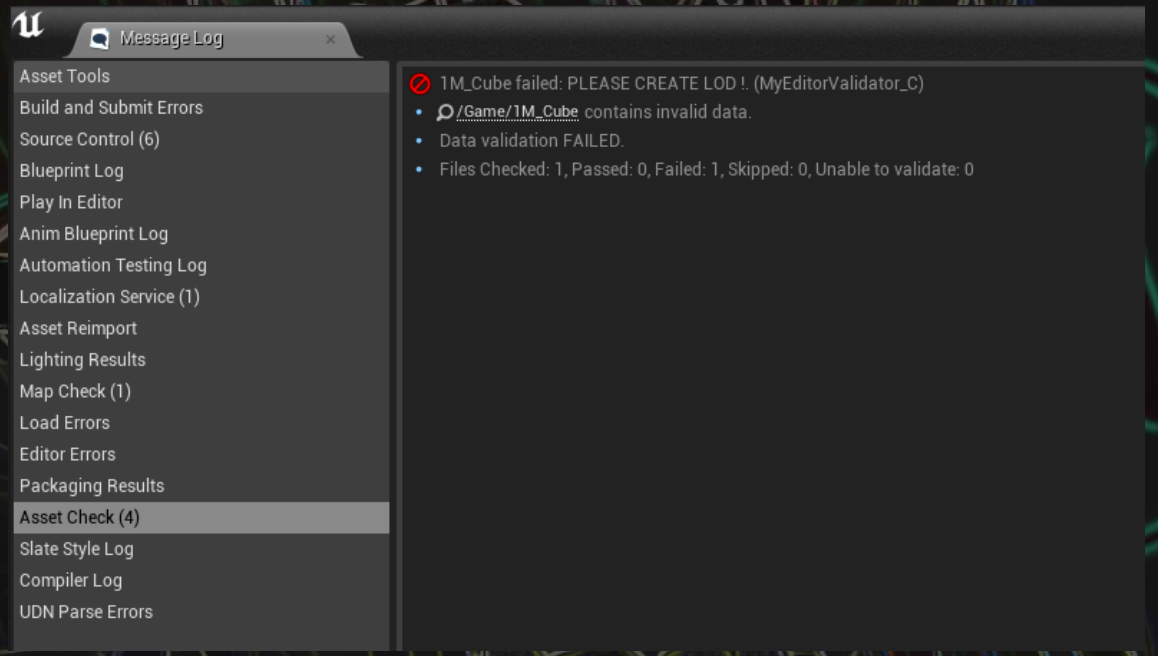
By historia様より

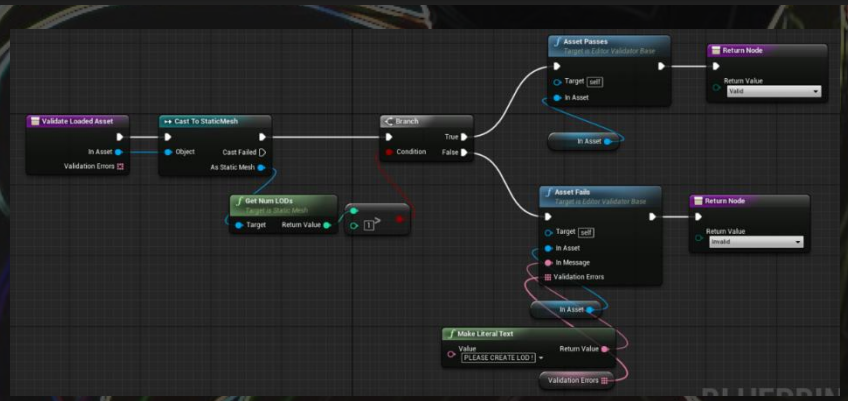


それは本当にバグ？

ちなみにUE4.23からお手軽に使えるようになった、アセットを保存時に独自で作成したエラーチェックをしてくれる「Editor Validator Subsystem」を使えば、ある程度事前に設定の抜けなどを確認することも出来る！
ブループリントオンリーで作れるよ！

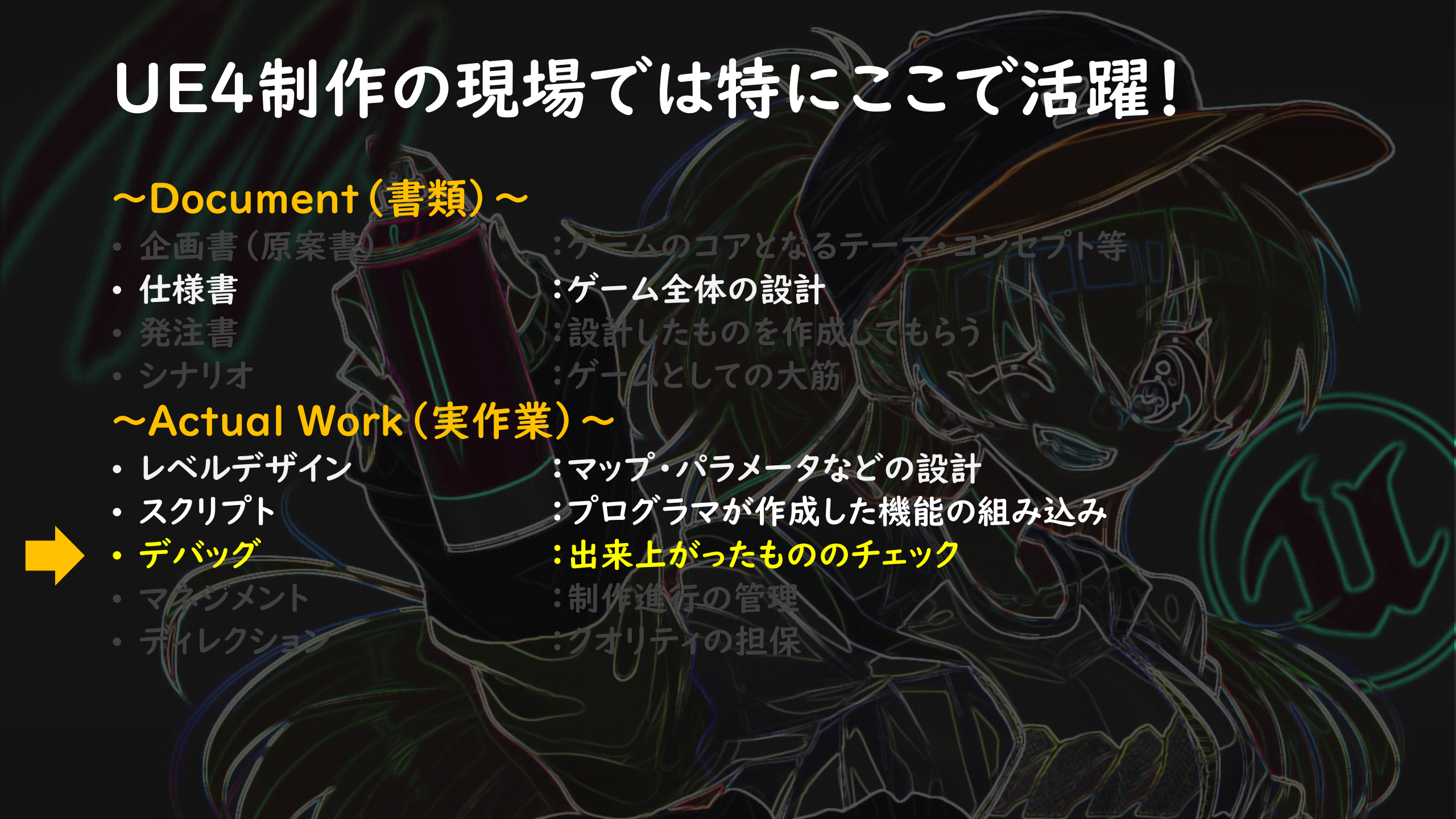






UE4.23から入った「Editor Validator Subsystem」を使って、アセット保存時などで走るチェック処理（Validate Assets）を拡張しよう！ By おかず様より

UE4制作の現場では特にここで活躍！



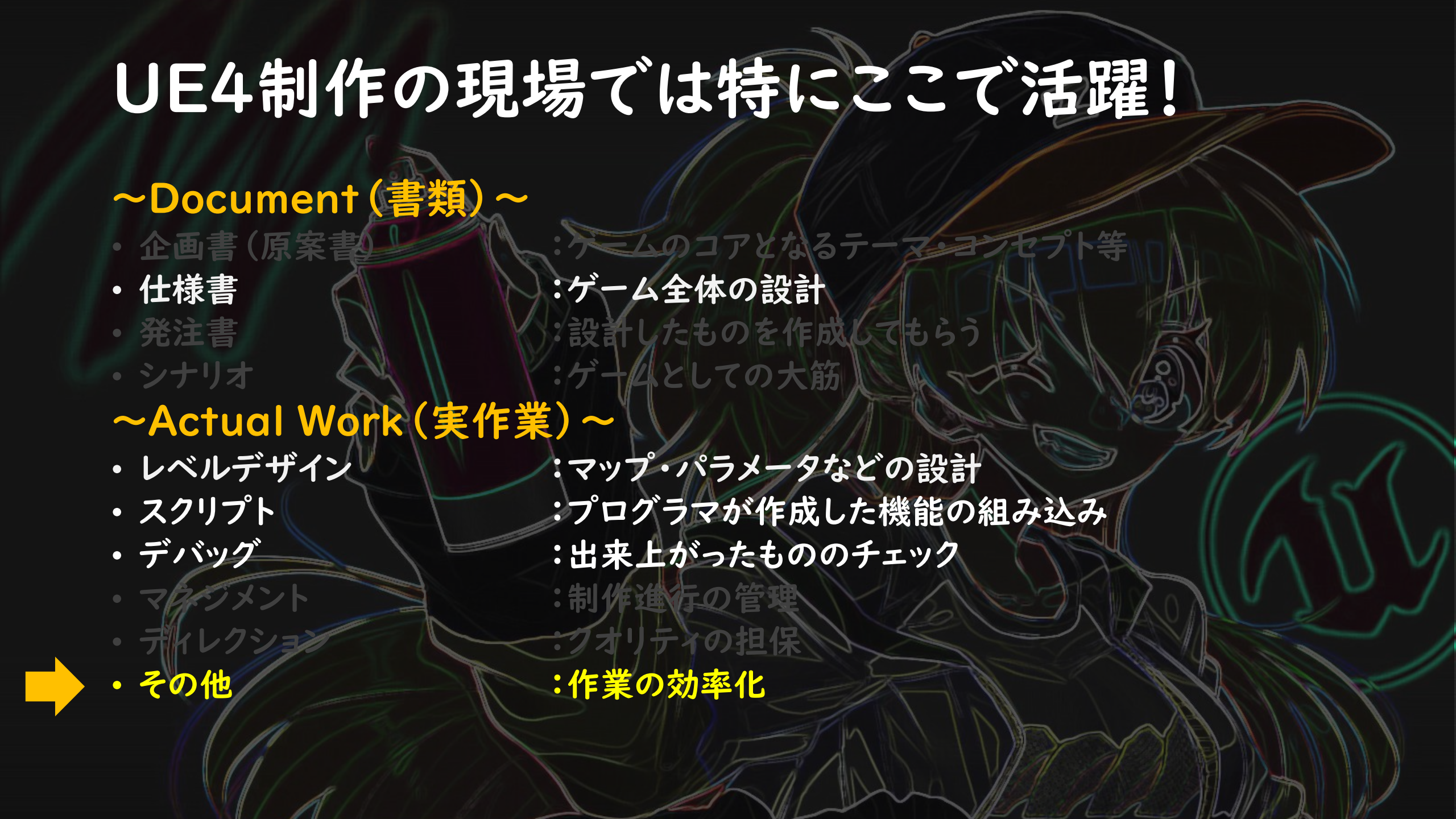
～Document（書類）～

- 企画書（原案書）：ゲームのコアとなるテーマ・コンセプト等
- 仕様書：ゲーム全体の設計
- 発注書：設計したものを作成してもらう
- シナリオ：ゲームとしての大筋

～Actual Work（実作業）～

- レベルデザイン：マップ・パラメータなどの設計
- スクリプト：プログラマが作成した機能の組み込み
- **デバッグ：出来上がったもののチェック**
- マネジメント：制作進行の管理
- ディレクション：クオリティの担保

UE4制作の現場では特にここで活躍！

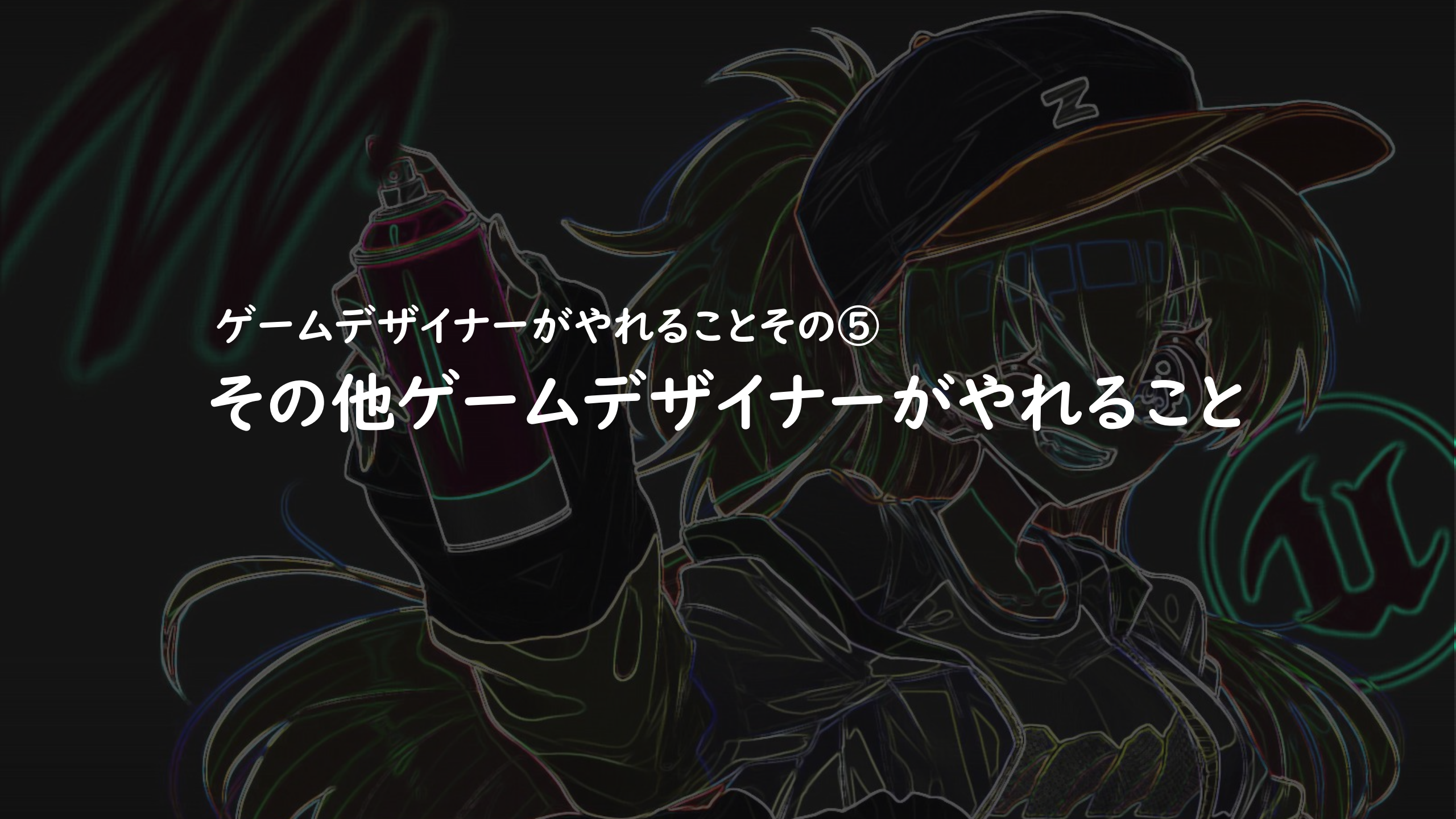


～Document（書類）～

- 企画書（原案書）　：ゲームのコアとなるテーマ・コンセプト等
- 仕様書　：ゲーム全体の設計
- 発注書　：設計したものを作成してもらう
- シナリオ　：ゲームとしての大筋

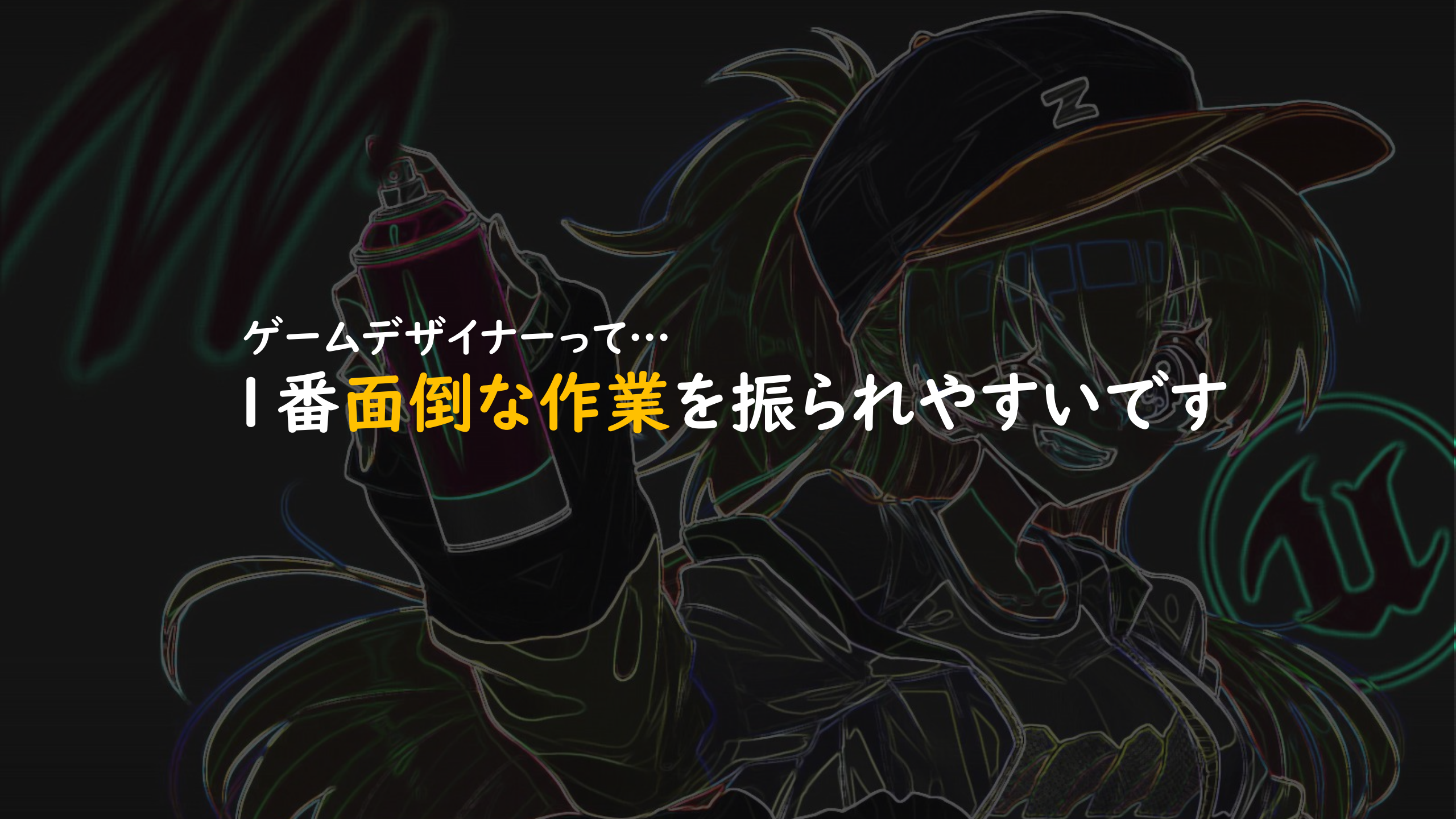
～Actual Work（実作業）～

- レベルデザイン　：マップ・パラメータなどの設計
- スクリプト　：プログラマが作成した機能の組み込み
- デバッグ　：出来上がったもののチェック
- マネジメント　：制作進行の管理
- ディレクション　：クオリティの担保
- その他　：作業の効率化



ゲームデザイナーがやれることその⑤

その他ゲームデザイナーがやれること

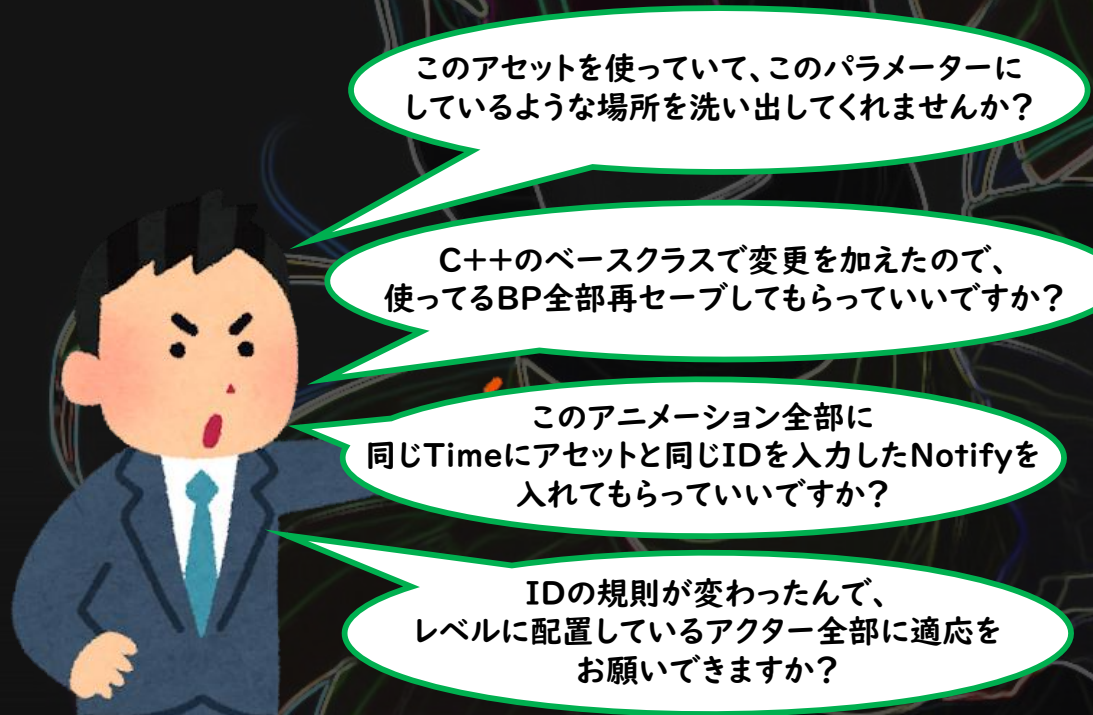


ゲームデザイナーって…

1番面倒な作業を振られやすいです

作業はなるべく自動化しよう

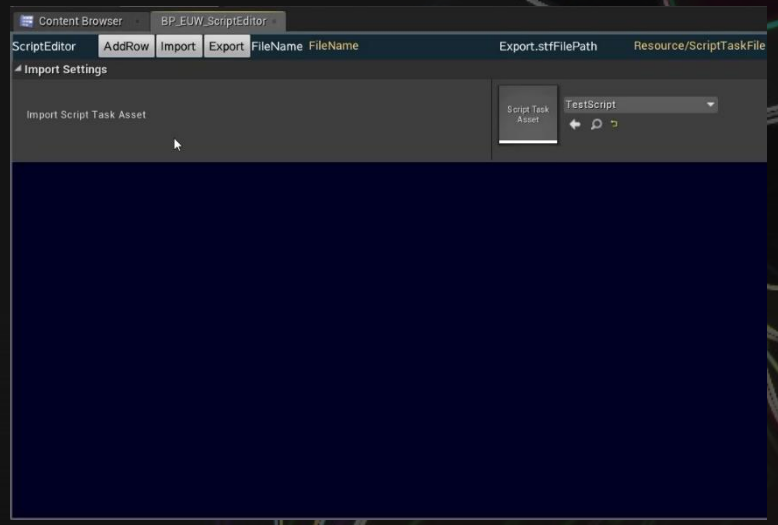
- 例えば、「50個のアクターのタグに、それぞれ連番で数字を入れてほしい」と言われたときに、愚直に手作業でやるのと、ツールを作るのでは、最初の作業こそ手作業の方が早いかもしれませんが、後々「あ、やっぱり名前+数字で！」とか「あーほかのレベルも全部やってほしい」というようなことがあるかもしれません。（というか絶対に1回きりの作業なんてないと思った方がいいです）



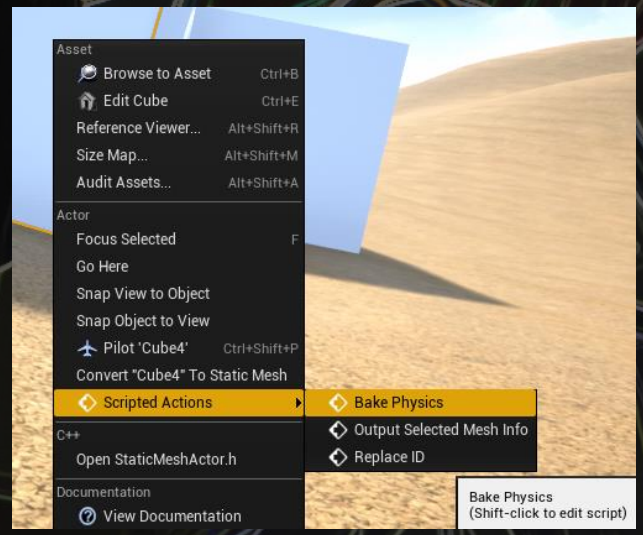
作業はなるべく自動化しよう

- 実作業時間をどれだけ効率化できるかで、その分「クリエイティブに費やす時間」が変わってきます。
- 現在のUE4では、「Editor Utility」や「Python」プラグインなどで、幅広いエディター拡張が行えます。みなさんも単純作業を振られた際は、是非活用してください。（次の講演で少しお話します）

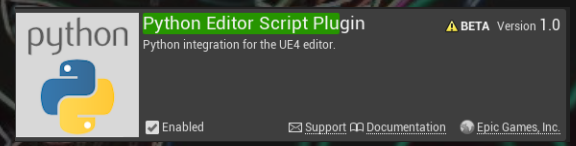
Editor Utility Widget

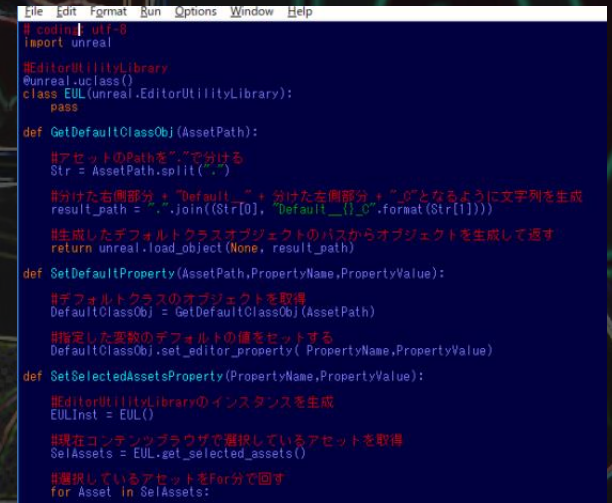


Editor Utility Blueprint

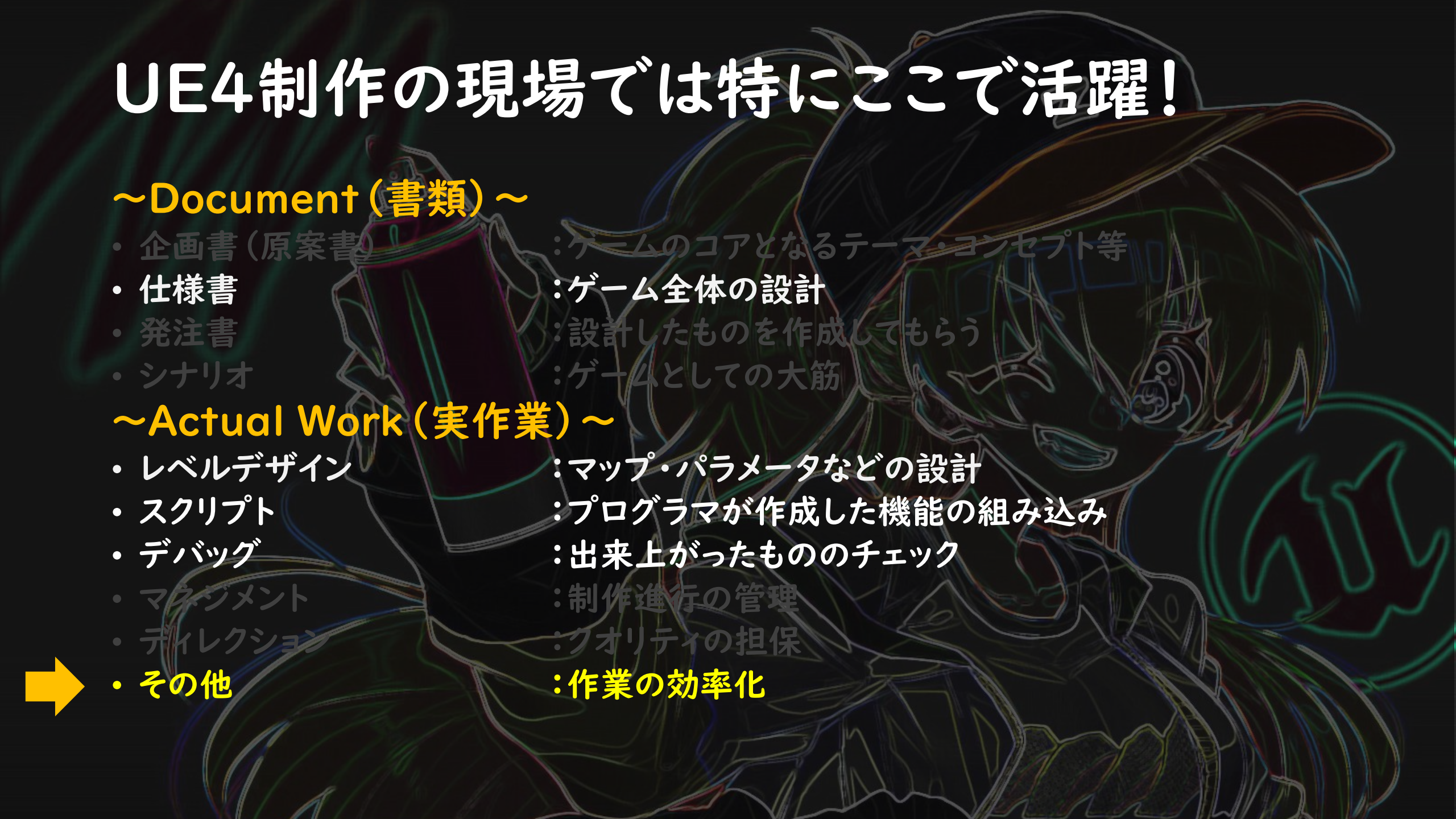


Python Editor Script Plugin





UE4制作の現場では特にここで活躍！

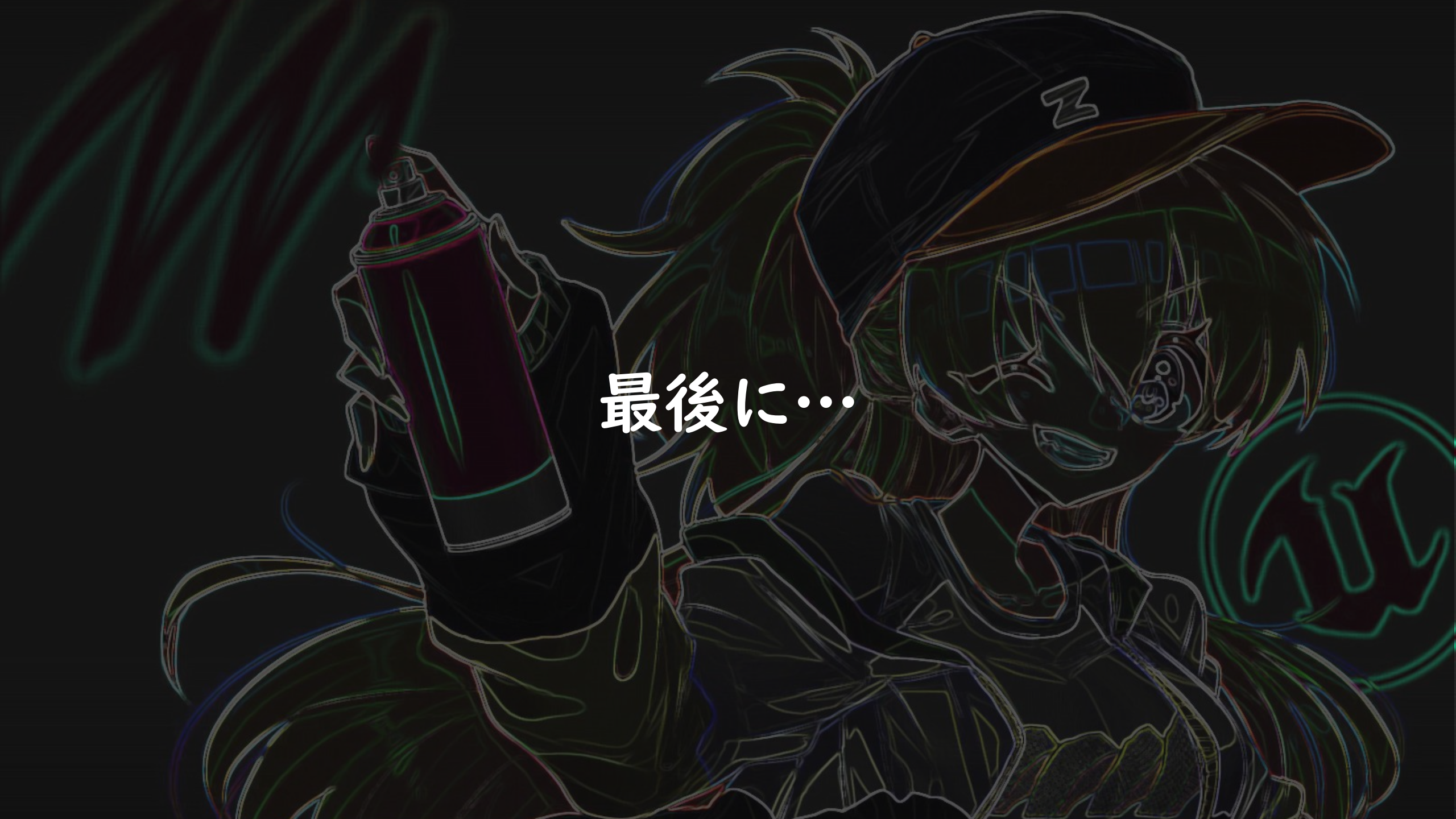


～Document（書類）～

- 企画書（原案書）　：ゲームのコアとなるテーマ・コンセプト等
- 仕様書　：ゲーム全体の設計
- 発注書　：設計したものを作成してもらう
- シナリオ　：ゲームとしての大筋

～Actual Work（実作業）～

- レベルデザイン　：マップ・パラメータなどの設計
- スクリプト　：プログラマが作成した機能の組み込み
- デバッグ　：出来上がったもののチェック
- マネジメント　：制作進行の管理
- ディレクション　：クオリティの担保
- その他　：作業の効率化

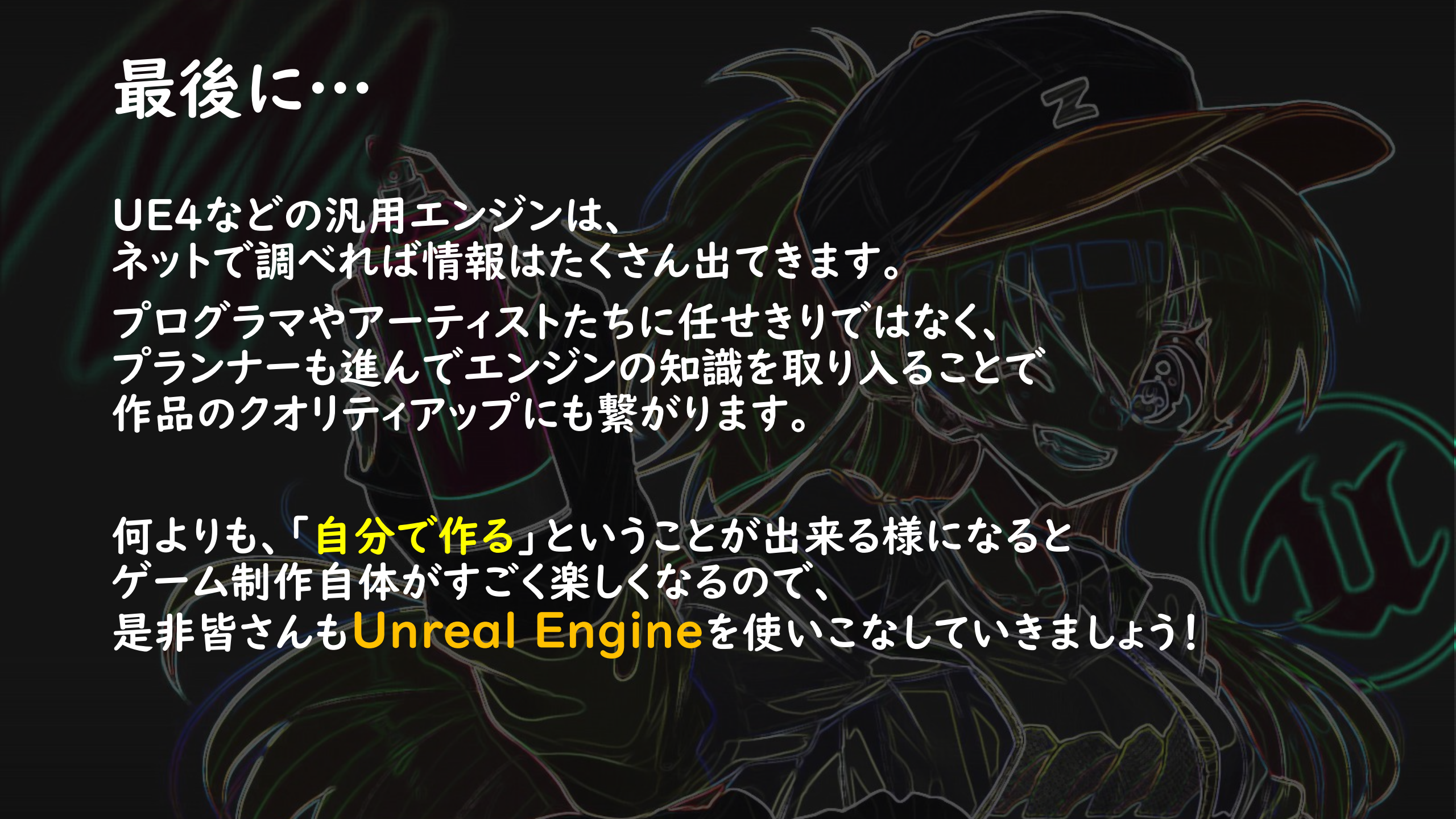

最後に…

最後に…

個人的には、ゲームは
「面白いゲームを作る」ことを考えるだけでなく、
「面白くゲームを作る」ことも考えて制作に取り組むことで
よりクオリティの高いものができると思います。

「楽しく」「効率的」に制作すれば、より細部にまでこだわって
作り込むことができると思います。

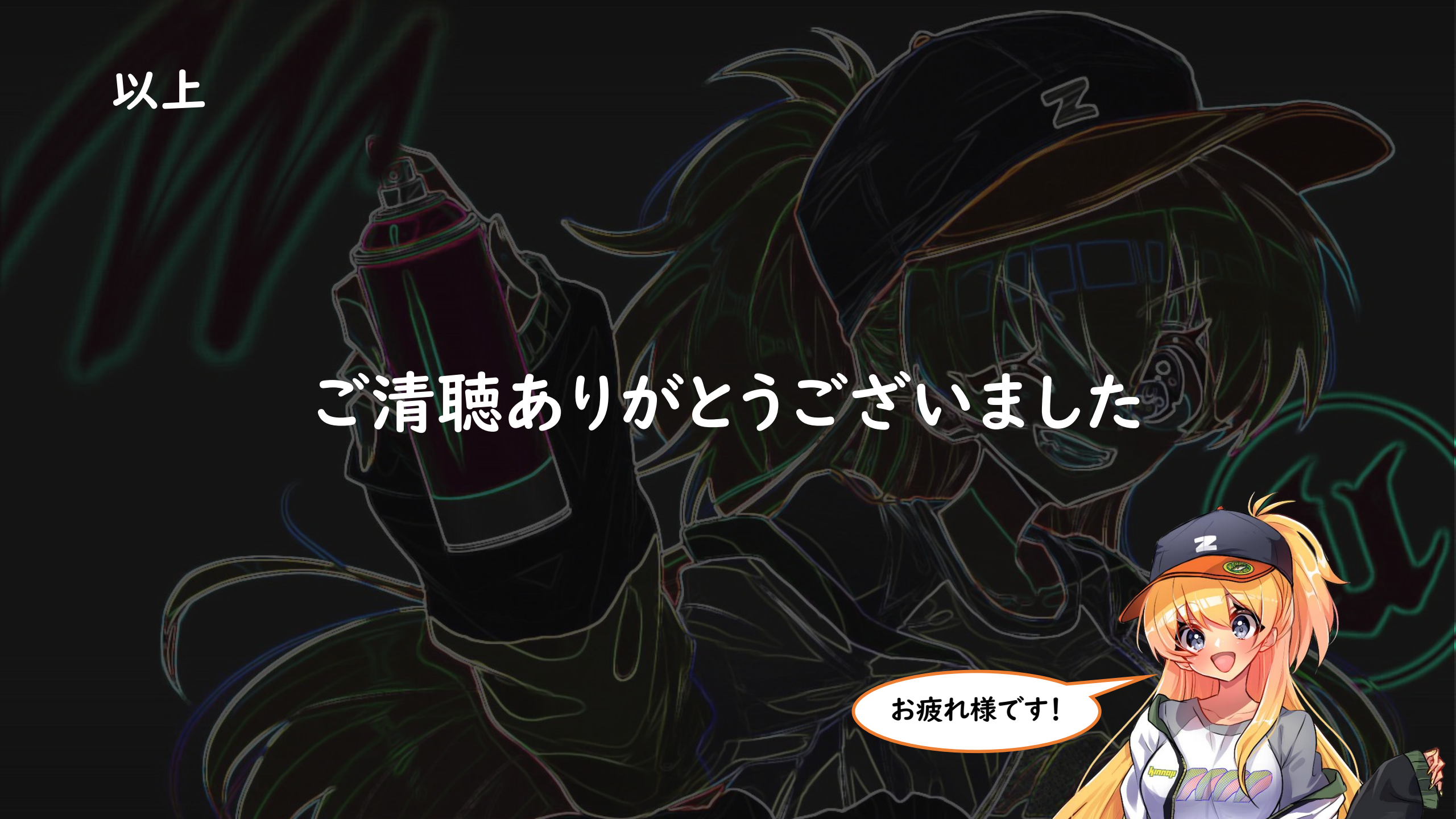
最後に…



UE4などの汎用エンジンは、
ネットで調べれば情報はたくさん出てきます。

プログラマやアーティストたちに任せきりではなく、
プランナーも進んでエンジンの知識を取り入れることで
作品のクオリティアップにも繋がります。

何よりも、「**自分で作る**」ということが出来る様になると
ゲーム制作自体がすごく楽しくなるので、
是非皆さんも**Unreal Engine**を使いこなしていきましょう！

以上

ご清聴ありがとうございました