

# 「アクションゲーム開発をしてみて」 プログラマ編

新大阪UE4勉強会

# 「自己紹介」

- ・名前: 小屋 聖揮 21才
- ・Twitter: Ya-kon @Ya\_konpota
- ・バンタンゲームアカデミー大阪校 ゲームプログラム総合2年
- ・最近はアナログゲームを作ろうと考えてます。UE4歴は1年半
- ・現在就活中です。誰か拾ってください。



# 「はじめに」

- ・制作事例を簡単に紹介するスライドです。(何をしたか・何を使ったか)チュートリアルではないので注意してください。
- ・スライドで紹介するものはUE4の最適解ではありません。
- ・このスライドの対象者
  - ・Unreal Engine4でゲームを作った経験がある人

# 「SkySeekerの紹介」

- ・開発メンバー13人(初期メンバー:7人)
- ・開発期間:約4ヶ月(プロジェクト着手してからTGS出展まで)
- ・UE4のバージョンは4.22
- ・2020年内にSteamでリリース予定
- ・プログラムのサポートをしていました。

\* オブジェクトの挙動

\* プレイヤーシステム

\* UIプログラム

\* エネミーシステム



# 「SkySeekerの紹介」

## プレイ動画



A large, faint, stylized letter "S" is centered in the background. It is composed of two overlapping, flowing shapes that create a sense of motion. The "S" is a light blue color, matching the background's gradient.

どうでしたか？

A large, stylized, light blue 'S' logo is centered in the background. The 'S' is composed of two thick, curved strokes that meet at the top and bottom, creating a sense of motion. The logo is set against a dark blue background with a subtle, circular, textured pattern that resembles a globe or a planet's surface.

**Steamでの配信をお楽しみに！！**

A large, faint, light blue stylized letter "S" is centered in the background. It is enclosed within a circular frame that has a textured, metallic appearance. The "S" itself is composed of thick, flowing strokes.

ここから本題



# 「略語」

- Blueprint → BP
- Animation Blueprint → ABP



# 「全体の流れ」

## -1.敵キャラクター制作事例(ざっくり説明)

- \* アニメーション制御(P.13)
- \* 行動制御(P.26)
- \* 設計(P.37)

## -2.小ネタ

- \* 動画挿入(P.46)

## -3.おまけ

- \* 開発で少しだけ困ったこと(P.58)
- \* 個人的に好きなプラグイン1つを紹介(P.67)

## 「注意」

- ・SkySeekerでは敵キャラクターが3体いますが、これからスライドでお見せする画像は敵3体の内の1体についての画像ですので注意。

A large, faint, stylized letter "S" is centered in the background of the slide. The "S" is composed of two main curved segments, with a smaller, more intricate swirl at the top right. It is rendered in a light blue or white color, making it subtle against the dark blue background.

# 「1.敵キャラクター制作事例」

A large, faint, stylized letter "S" is centered in the background. It is composed of two main curved segments, one above and one below, with a small gap in the middle. The "S" is rendered in a light blue or white color, making it stand out against the dark blue background.

# 「敵アニメーション制御」

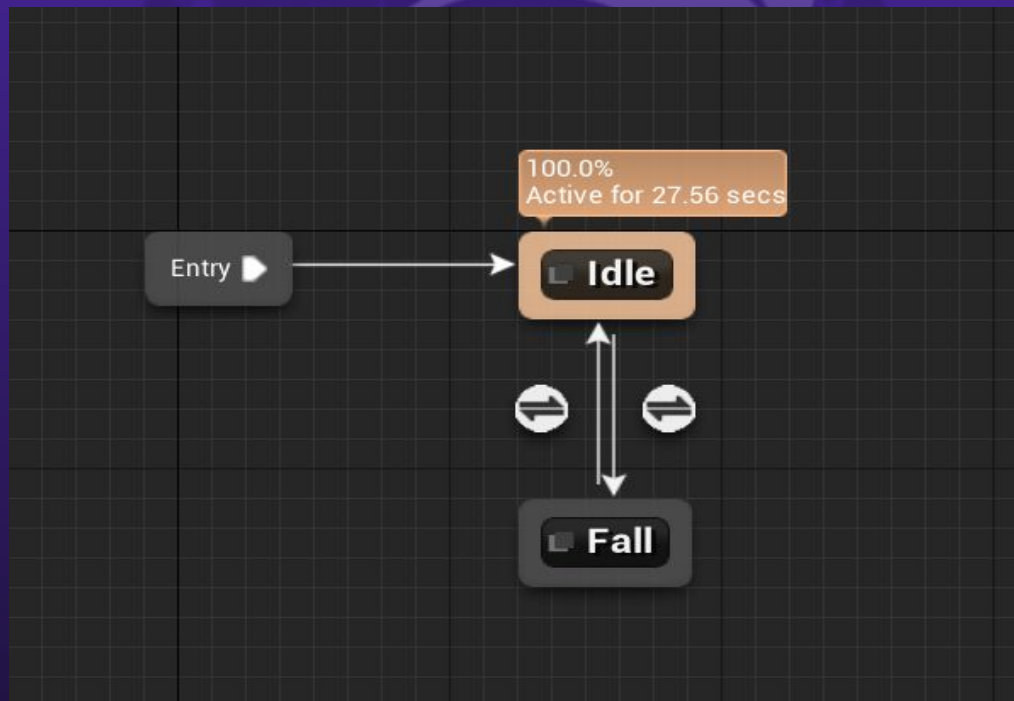
# 「アニメーション制御要件」

- ・アニメーションの追加がしやすいようにする。
- ・BPとABPの連携を複雑になりすぎないようにする。
- ・自分達が全く理解出来ていない機能を使わない。

A large, stylized, light blue letter 'S' is centered in the background. The 'S' has a thick, brush-stroke-like texture and is enclosed within a circular frame that also has a textured, metallic appearance. The entire background is a solid dark blue.

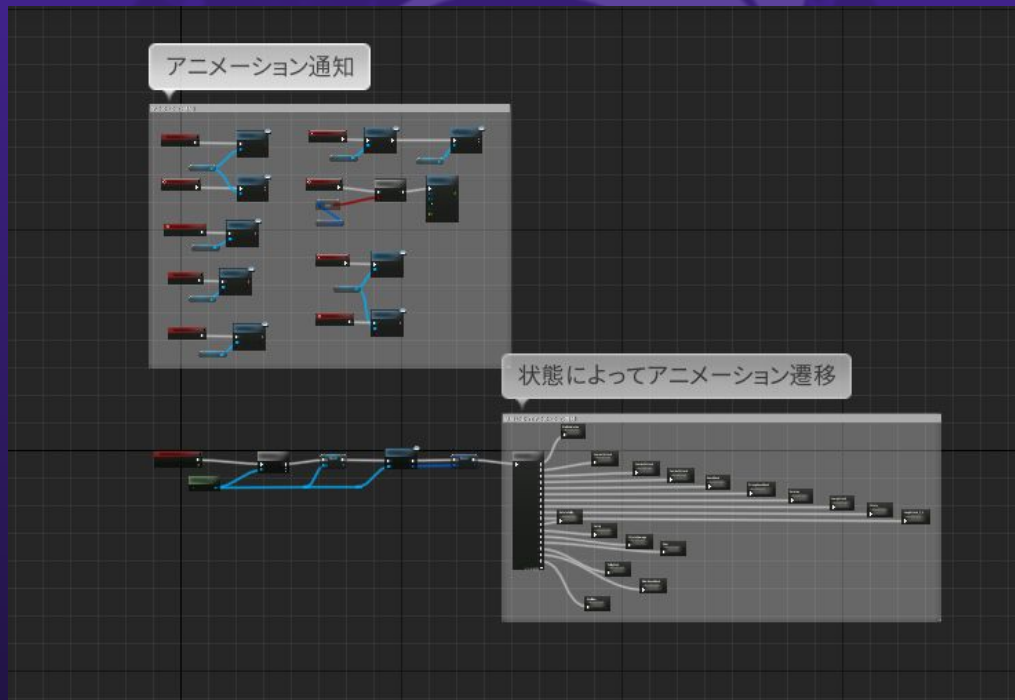
「最終的に出来たもの」

# 「最終的に出来たABP1/2」

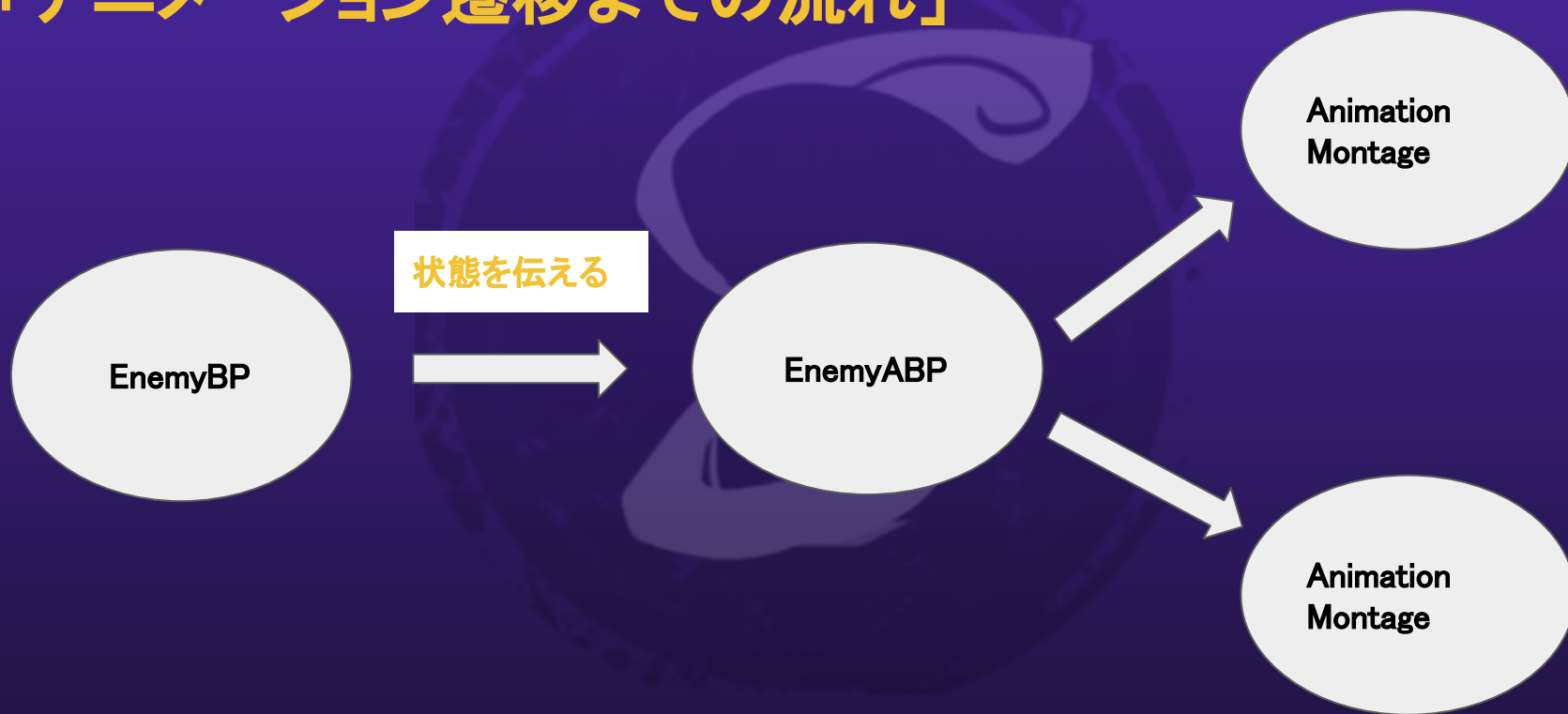




# 「最終的に出来たABP2/2」



# 「アニメーション遷移までの流れ」



## 「概要」

- ・ステートマシンは複雑になると後々大変なので基本的には Animation Montageを使用するようにした。

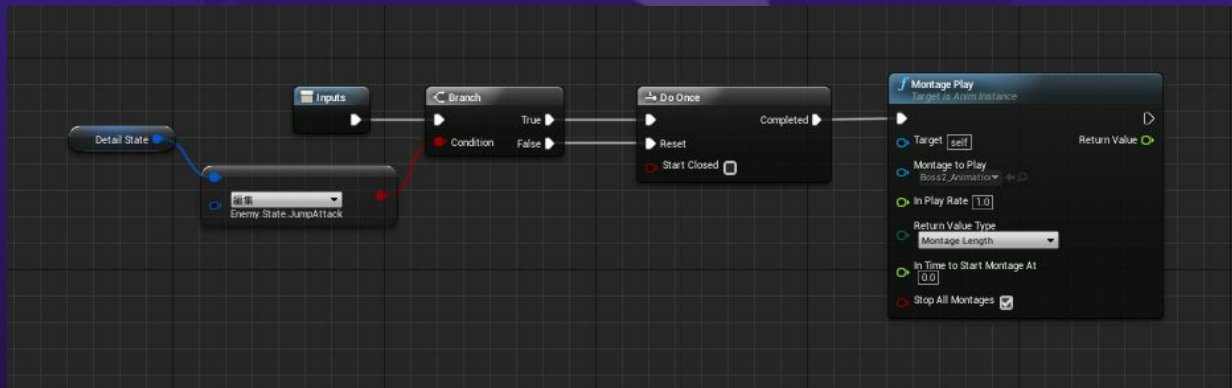
Animation Montageについては公式ドキュメントに詳しく書いています。

<https://docs.unrealengine.com/ja/Engine/Animation/AnimMontage/Editor/index.html>

- ・EnmeyBPの状態を受け取りEnemyABP内で状態に沿ったアニメーション遷移をする。
- ・アニメーションと連動したい処理はアニメーション通知を使用した。

# 「アニメーションの遷移1/2」

- ・アニメーション遷移にはGamePlayTagを利用して行う。
- ・EnemyBP内でGamePlayTagの情報を持った変数を作り、その変数を受信をする。目的のGamePlayTagならアニメーションを遷移する。



## 「アニメーションの遷移2/2」

- ・※GamePlayTagの変数の値の変更はABP内ではしないようにルール化。  
変更箇所が多くなると遷移の管理が難しくなるため。
- ・ABP内ではアニメーションの振る舞いをさせることだけ(基本的に受信)  
させるやり方が個人的におすすめ。



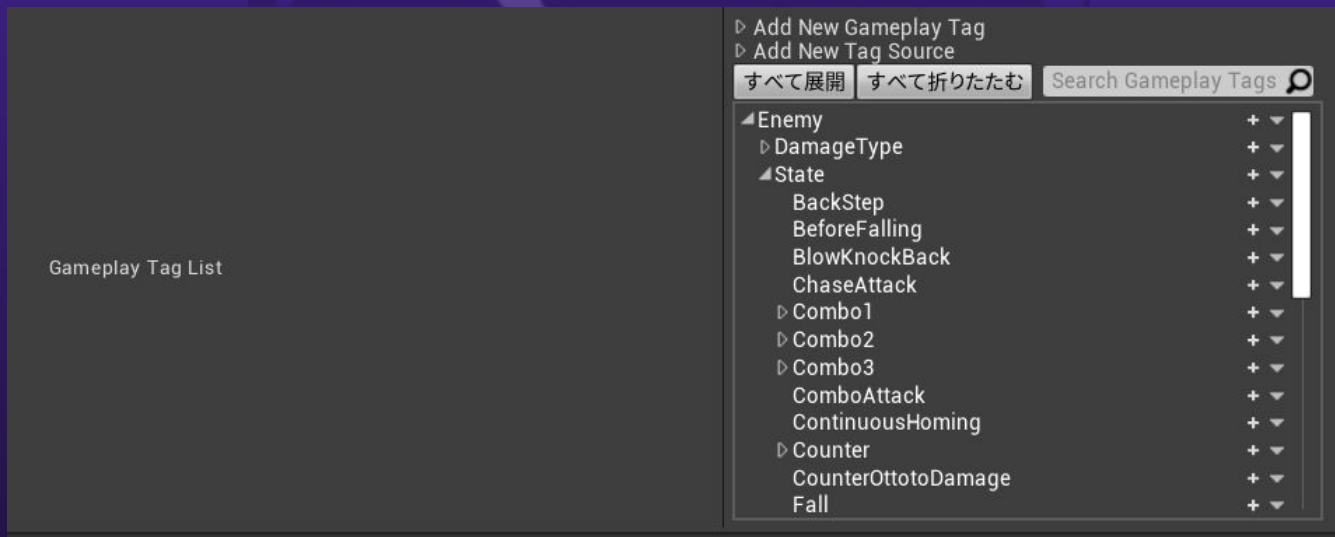
# 「GamePlayTag1/3」

- ・オブジェクトを特定、分類、マッチング、フィルタリングすることできるもの。
- ・階層持つことができる。
- ・公式ドキュメントに詳しいことは書いてあるので見てください！

<https://docs.unrealengine.com/ja/Gameplay/Tags/index.html>

# 「GamePlayTag2/3」

- ・GamePlayTagは管理がしやすい。(UE4のプロジェクト内のGamePlay Tag List からタグ全体を見ることが出来る。検索・追加・削除が簡単にできる。



# 「GamePlayTag3/3」

- ・今回のゲームの場合、キャラクターの状態の管理とダメージの種類を分けるのにGamePlayTag利用してます。



# 「アニメーション通知」

- ・Animation Montage内で通知を作成し、主にアニメーションの終了通知・攻撃時の当たり判定ON・OF通知などで活用。



A large, stylized, light blue letter 'S' is centered in the background. The 'S' has a thick, brush-stroke-like texture and is enclosed within a circular frame that also has a textured, metallic appearance. The entire background is a solid dark blue.

# 「キャラクターの行動制御」

# 「キャラクター行動制御要件」

- ・デバッグがしやすい。(ここが一番重要)
- ・敵の行動ごとにプログラマが作業することができる。

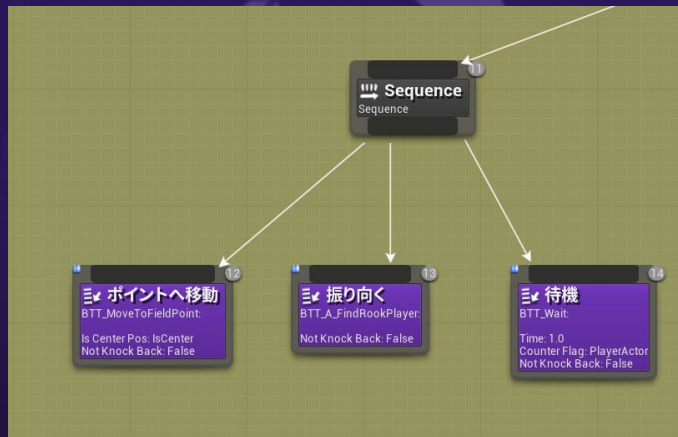
A large, stylized, light blue letter "S" is centered on the page. It is set against a dark blue circular background that has a textured, slightly distressed appearance. The "S" itself has a smooth, flowing design with a small hook at the top and bottom.

何を使ったか

# 「Behavior Tree」

- ・簡単にいうとキャラクターに意思決定と判断をさせるもの。
- ・条件式によって対応しているタスク(アクション)を実行する
- ・詳しくは公式ドキュメントに書いてます！

<https://docs.unrealengine.com/ja/Engine/ArtificialIntelligence/BehaviorTrees/index.html>



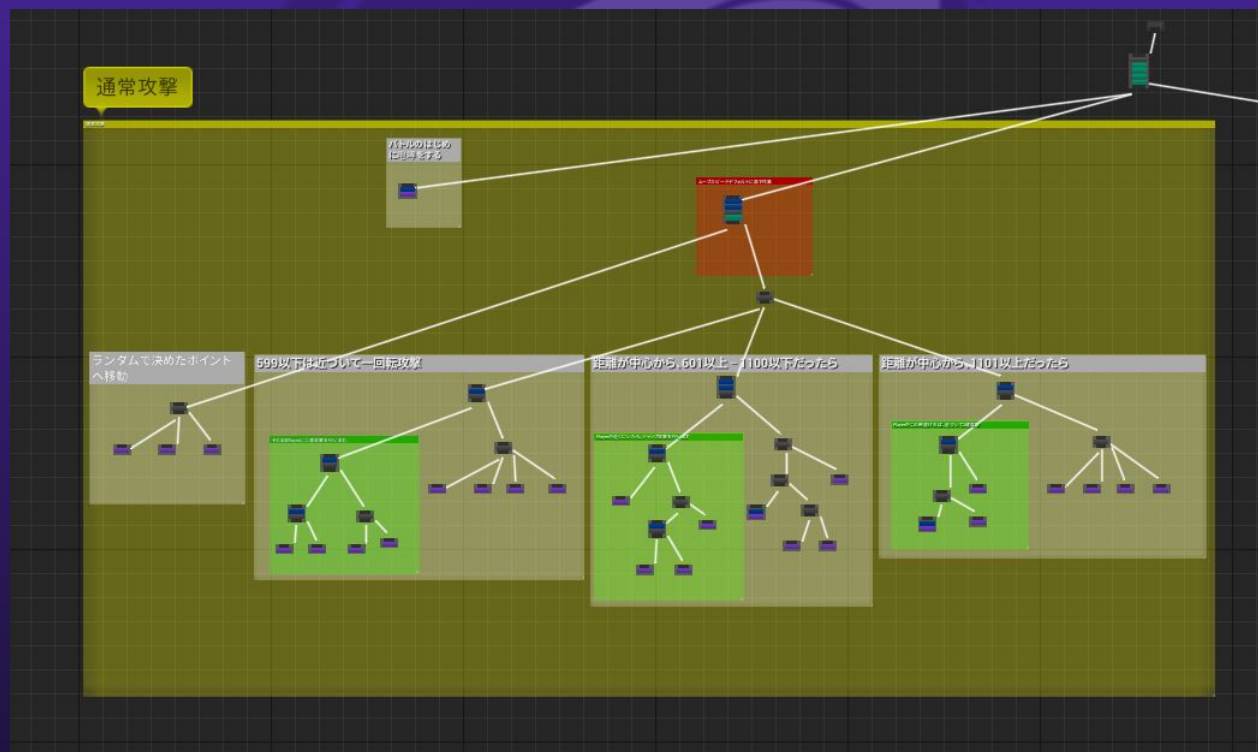
# 「なぜBehavior Treeなのか」

- ・タスクごとにプログラマが処理を組める。
- ・ゲームプレイ中にフローの流れがはっきり見えてデバッグが容易。

A large, faint, stylized letter "S" is centered in the background. It is a light blue color and has a thick, brush-stroke-like texture. The "S" is enclosed within a circular frame that also has a textured, stone-like appearance.

「最終的に出来たもの」

# 「実際にできたBehavior Tree1/2」





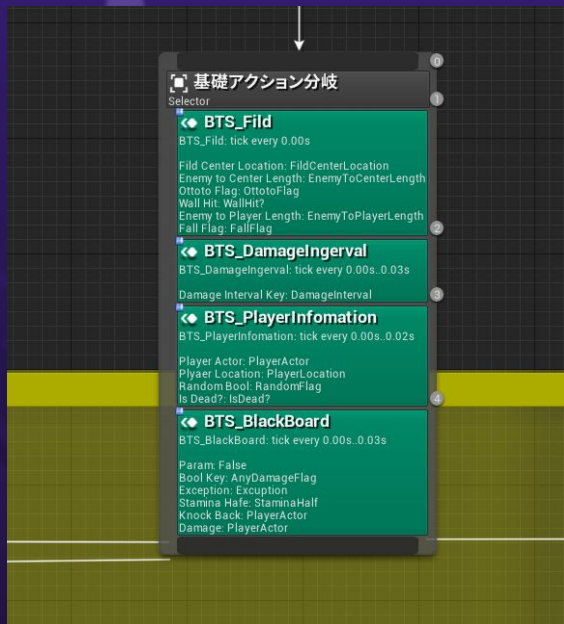


# 「概要」

- Service、Decoratorを新しく作りBehavior Treeの挙動を制御。
- Task内では敵のアニメーション以外の挙動を制御。(移動距離・回転量など)
- Behavior TreeとABPとBPの関係は後ほどの設計のスライドで説明します。

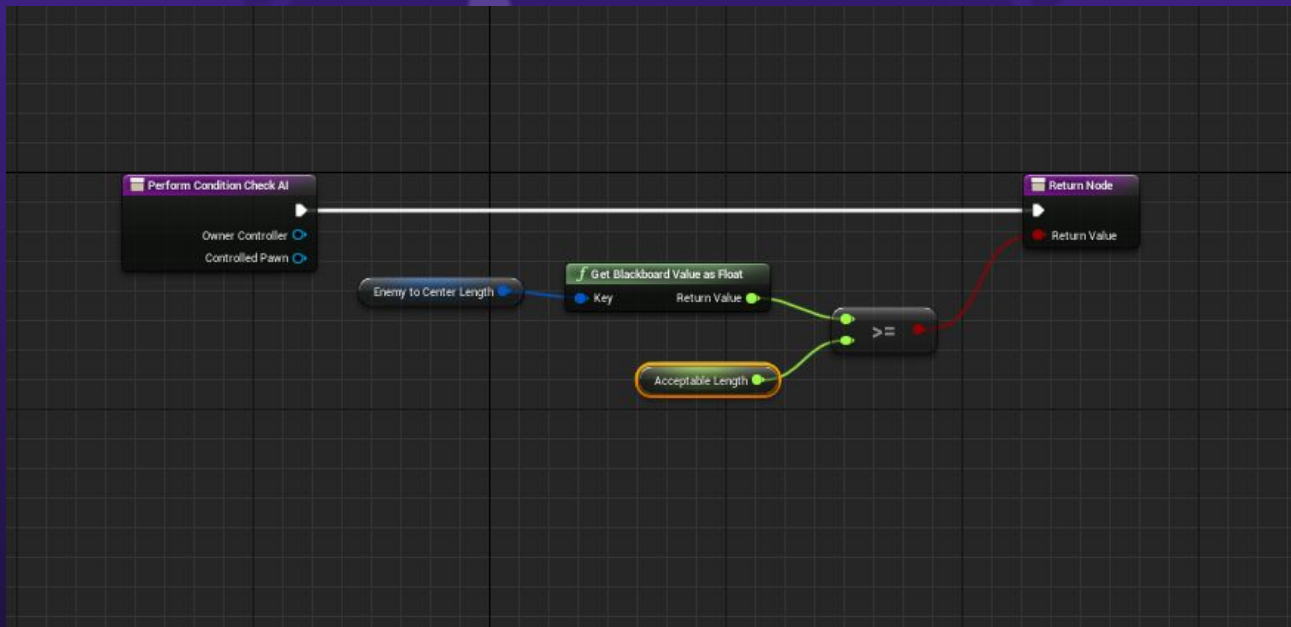
# 「Serviceの主な使い方」

- ・Blackbordで使用する変数の更新。(プレイヤーの情報・ダメージが入ったかどうかなど)



# 「Decoratorの主な使い方」

- ・値を比較して実行判定を出しています。



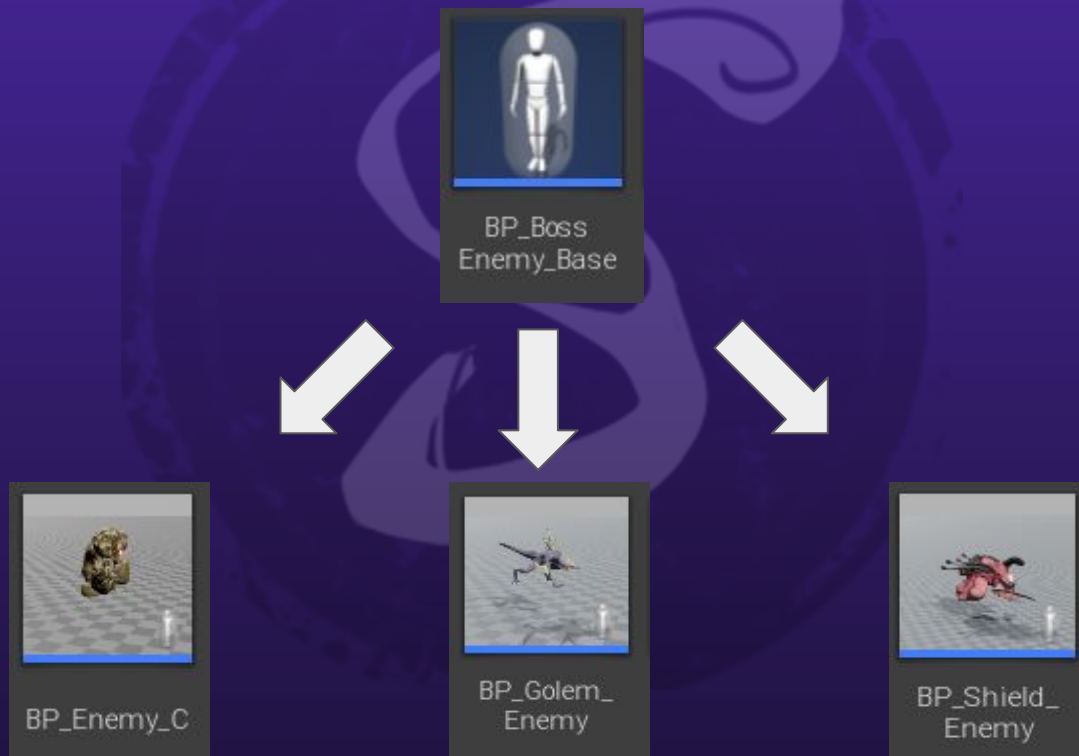
A large, faint, stylized letter "S" is centered in the background. It is a light blue color and has a thick, brush-stroke-like texture. The "S" is enclosed within a circular frame that also has a textured, slightly irregular appearance.

# 「敵キャラクター設計」

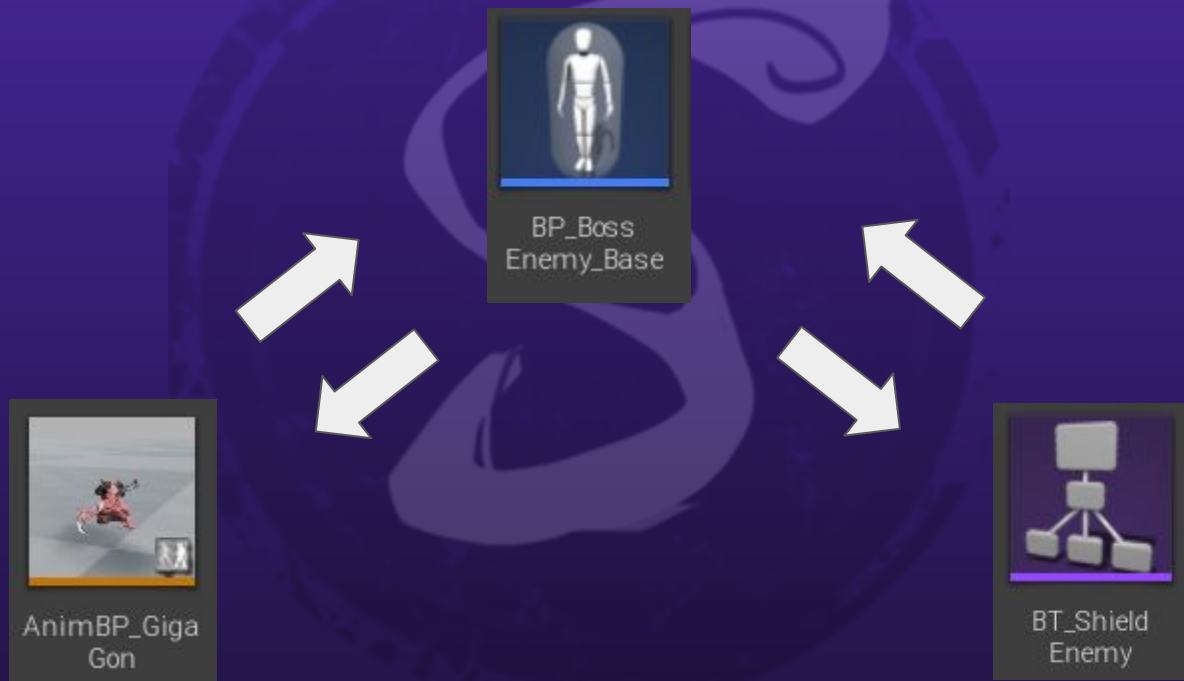
# 「敵キャラクター設計要件」

- ・デバッグがしやすい設計。(処理を分担、ルール決め)
- ・Behavior TreeとABPとキャラクターBPの通信が複雑にならないようにする

# 「最終的に出来た敵キャラクター設計1/2」



# 「最終的に出来た敵キャラクター設計2/2」





# 「敵キャラクターのベースクラス」

- ・敵キャラクターがプレイヤーにダメージを与えられた時の処理  
(サウンド・エフェクト・)
- ・GamePlayTagを管理。ABPとBehavior Treeの橋渡しの役割。
- ・アニメーションの終了フラグやダメージを受けたことをタスクに伝える役割。



# 「ベースクラスを継承した敵クラス」

- ・攻撃の当たり判定を出すかどうかを判断する役割。
- ・パラメーターの更新。



BP\_Enemy\_C



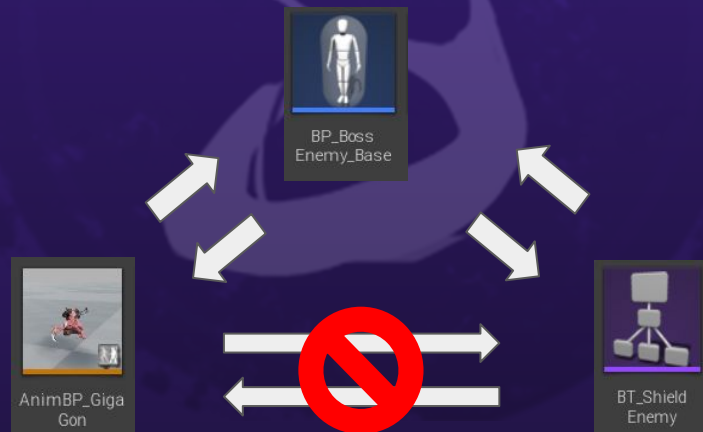
BP\_Golem\_  
Enemy



BP\_Shield\_  
Enemy

# 「Behavior TreeからABPに情報を渡す」

- ・Taskが実行されていると実行されているTaskに応じてGamePlayTagを更新。  
敵キャラクターのベースクラスに情報を送る。
- ・敵キャラクターのベースクラスからABPに情報を伝える。



# 「良かったこと」

- ・設計をシンプルにしたことでバグが出ても発見場所がある程度特定される。
- ・ルールが決まったことでバグが少なくなった。

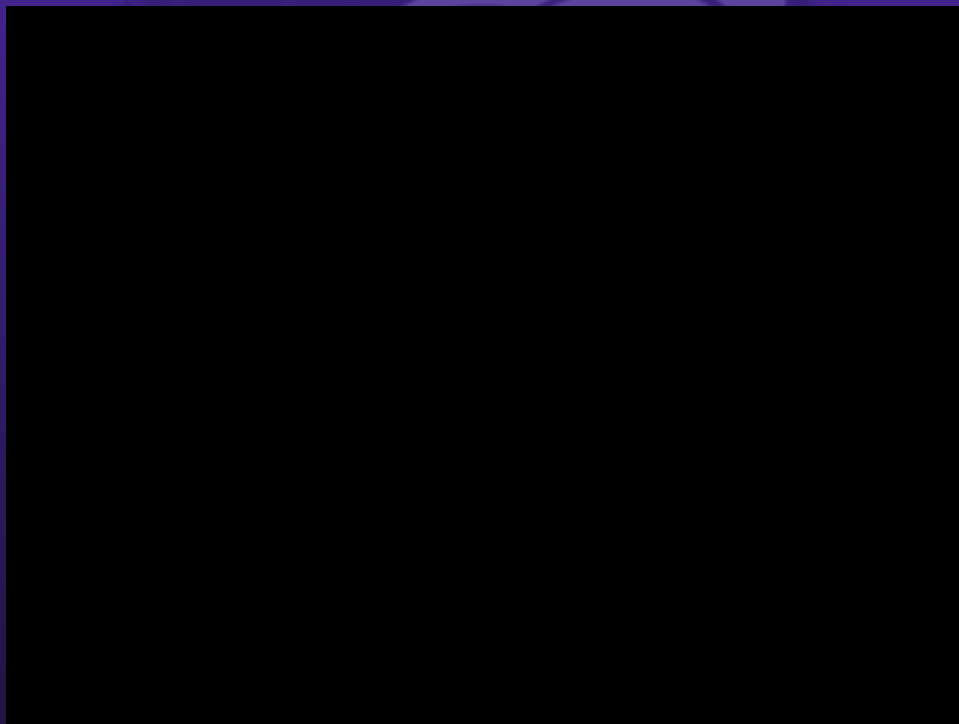
A large, stylized, light blue 'S' logo is centered on the page. The 'S' is composed of two main curved segments, with a smaller, darker blue segment in the middle. The logo is set against a dark blue background that features a faint, circular, textured pattern, possibly representing a globe or a celestial body.

## 「2.小ネタ」

A large, faint, light blue stylized letter "S" is centered in the background. The "S" has a thick, brush-stroke-like texture. It is enclosed within a circular frame that has a textured, stone-like appearance.

「動画挿入」

「こんな感じで背景に動画を流す」



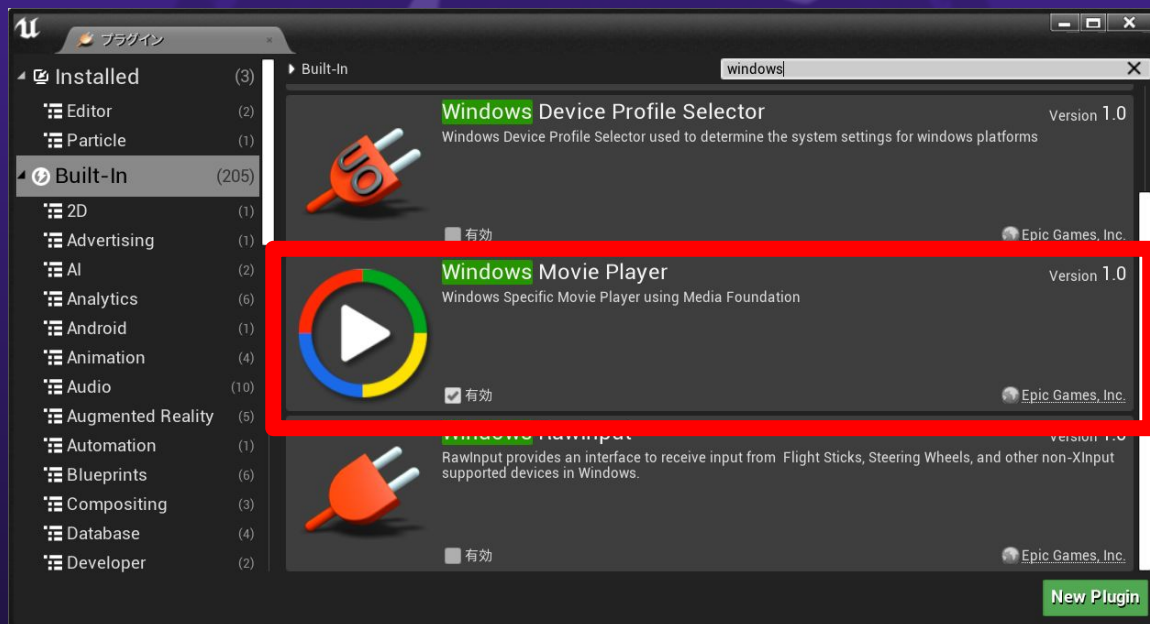
# 「はじめに」

- ・動画ファイルの形式はmp4です。
- ・動作を正常に確認しているUE4のバージョンは4.22です。



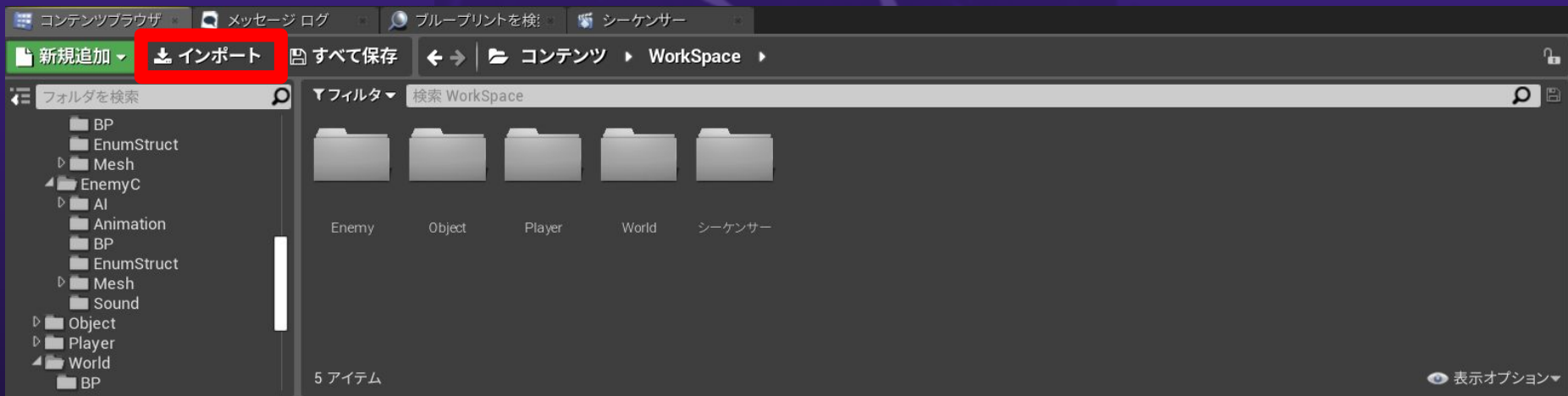
# 「必要なものを準備1/6」

・Windows Movie Player プラグインを有効化



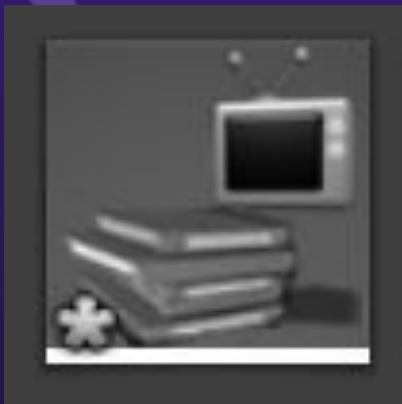
# 「必要なものを準備2/6」

- ・動画ファイルをUE4プロジェクトのContent/Movies直下に置く
- ・その次に動画ファイルをUE4にインポートする



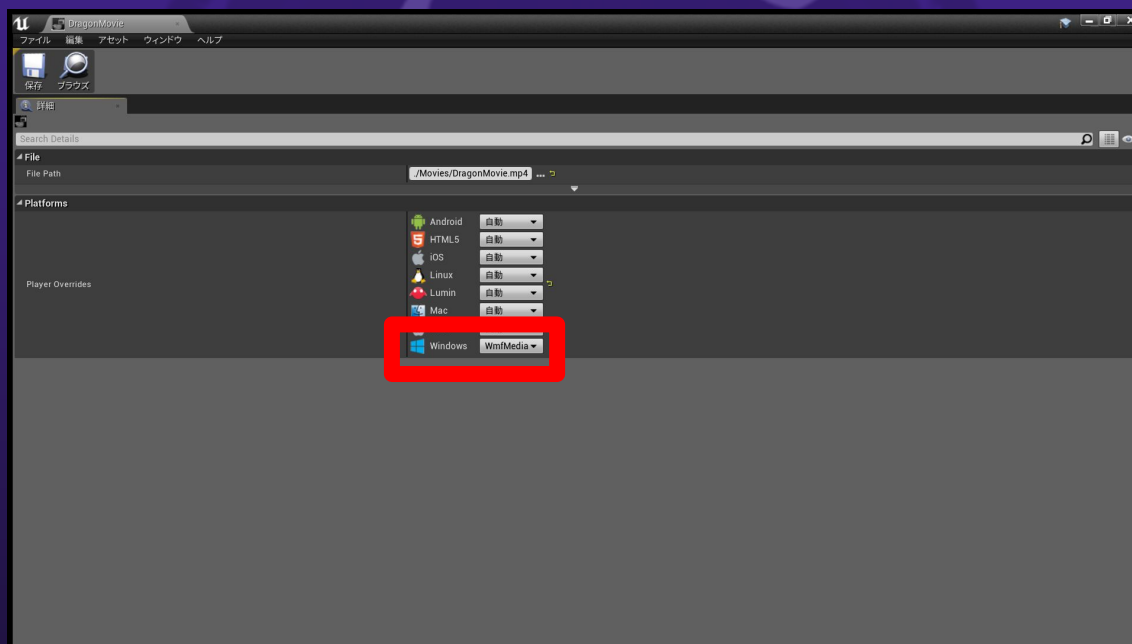
## 「必要なものを準備3/6」

- ・インポートが完了するとFile Media Sourceが生成されます。



# 「必要なものを準備4/6」

- File Media Sourceの設定を変更(Windowsの項目をWmFMediaに)



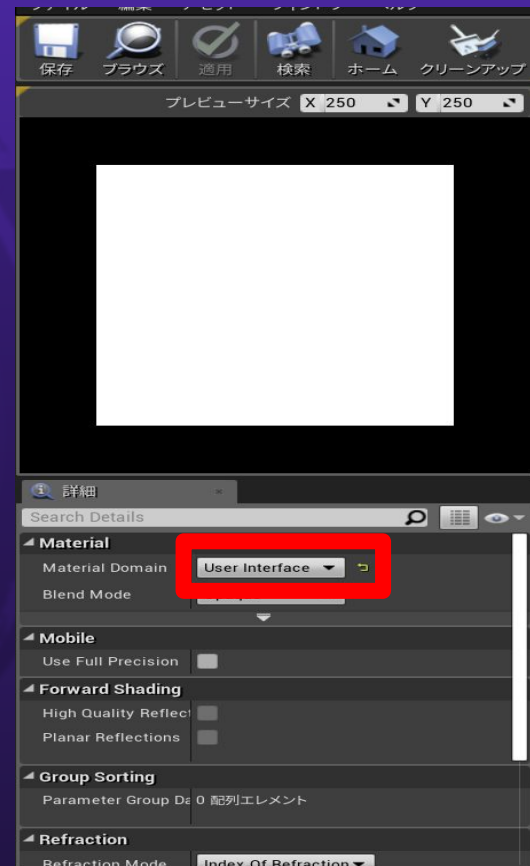
## 「必要なものを準備5/6」

- ・新規追加→Media→Media Playerを選択し追加。
- ・このウインドウ↓が出たらVideo output Media Texture assetの項目をチェック



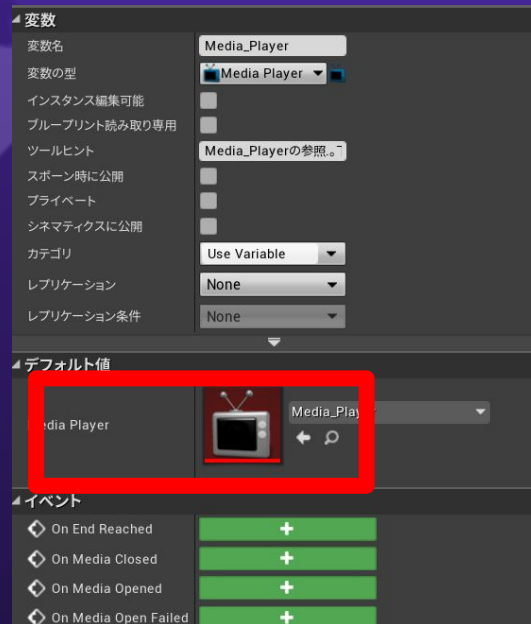
# 「必要なものを準備6/6」

- ・Media Player と Textureが生成されているのを確認する。
- ・Textureファイルを右クリックしてマテリアル作成を選択
- ・今回はUMGで動画を再生させたいのでマテリアルの設定をUserInterfaceに変更



# 「再生1/2」

- ・UMG内でMedia Playerの変数を追加
- ・デフォルト値を作成したMedia Playerに変更
- ・UMG内のImageなどにMedia Playerのマテリアルを適応



## 「再生2/2」

- ・あとはBP内で好きなタイミングで動画を再生させよう
- ・フェードなどを入れるといい感じになるのでおすすめ





A large, stylized, light blue letter 'S' is centered on the page. It is set against a dark blue circular background that has a textured, slightly grainy appearance. The 'S' is thick and has a slight shadow, giving it a three-dimensional feel.

# 「3.おまけ」

A large, faint, stylized letter "S" is centered in the background. It is a light blue or white color, matching the text, and has a thick, brush-like texture. The "S" is enclosed within a circular frame that has a rough, torn-edge appearance, similar to the one in the logo above.

「開発で少し困ったこと」

# 「開発で少し困ったこと.1」

プログラムにコメントがない

# 「課題」

- ・特定のプログラムを探すのに時間がかかる。
- ・コードの流れを理解するのに時間がかかる



# 「解決」

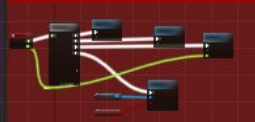
- ・何のプログラムなのかをコメントを書いて説明
- ・色をつけて区別する

基本的には概念だけ記入、処理は各イベントグラフへ

Begin



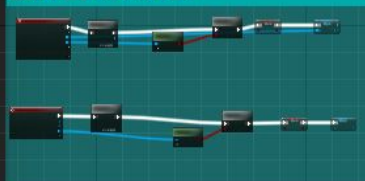
Tick



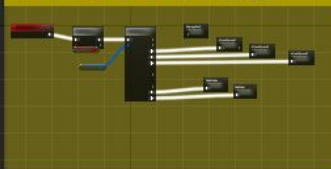
パラメータのアップデートをする



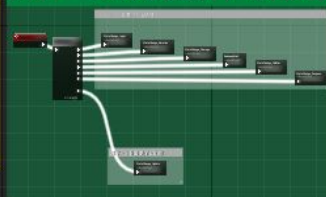
壁に当たったら



ダメージを与える処理



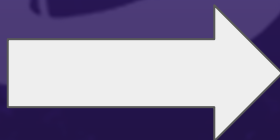
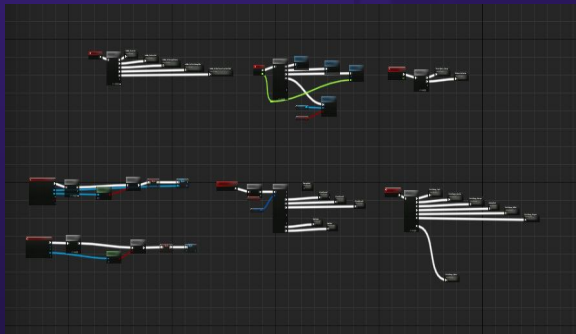
ステータスを更新する処理



# 「効果」

- ・瞬時に何のコードかを把握させることで、比較的すぐにプログラムを特定できた
- ・コードでは伝えられない部分を書くことで理解するスピードが上がった

コメントなし



コメントあり



## 「開発で少し困ったこと.2」

命名が不規則

# 「課題」

- ・名前で区別がつけられない(ブループリント、関数、変数、マテリアル、アニメーション)
- ・ファイル名検索で見つけれない





# 「解決」

- ・チームで話して細かく命名規則を決めた
- ・Epic Gamesさんの資料を参考にしました。

URL: [https://wiki.unrealengine.com/Assets\\_Naming\\_Convention\\_JP](https://wiki.unrealengine.com/Assets_Naming_Convention_JP)

(一部抜粋)

BP_	ブループリント
SK_	スケタルメッシュ
SM_	スタティックメッシュ
広告_	Apexディストラクティブルアセット (破壊可能)
交流_	Apexクロスアセット (布)
MT_	モーターゲット (変形先)
ST_	スピードツリー
PS_	パーティクルシステム
LF_	レンズフレア
VF_	データベースフィールド
S_	サウンド
SC_	サウンドキュー
M_	マテリアル
MI_	マテリアルエイリアス
MITV_	マテリアルプロキシタイムバッキング (時間変化)
MF_	マテリアルファンクション
MPC_	マテリアルポリシーコレクション
T_	デクスチャー

# 「効果」

- ・目的のファイルに早くたどり着ける
- ・数が多いもの(Particle,Material)でも名前で判別できるようになって良い

## 実際に決めたもの(一部)

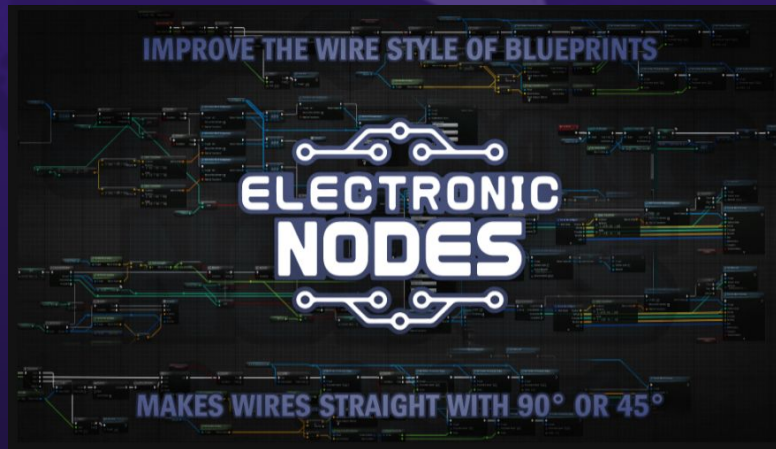
Name	NameBP	BP_		Struct	STR_
	Material	M_/MI_		enum	E_
	Texture	T_		Component	COM_
	Sound	S_		Function	F_
	Particle	PS_		animation	AnimBP
	Mesh	SM_		Widget	UI_

A large, faint, stylized letter 'S' is centered in the background. It is a light blue or white color, matching the background, and has a thick, brush-like texture. The 'S' is slightly tilted and has a decorative, swirling tail at the bottom right.

「個人的に好きなプラグインを一つ紹介」

# 「Electronic Nodes」

- ・ノードとノードをつなぐ線をより視覚化してくれるプラグイン
- ・自分好みにカスタマイズできる
- ・Rerouteノードを出さなくてもある程度視覚化されるのでおすすめ



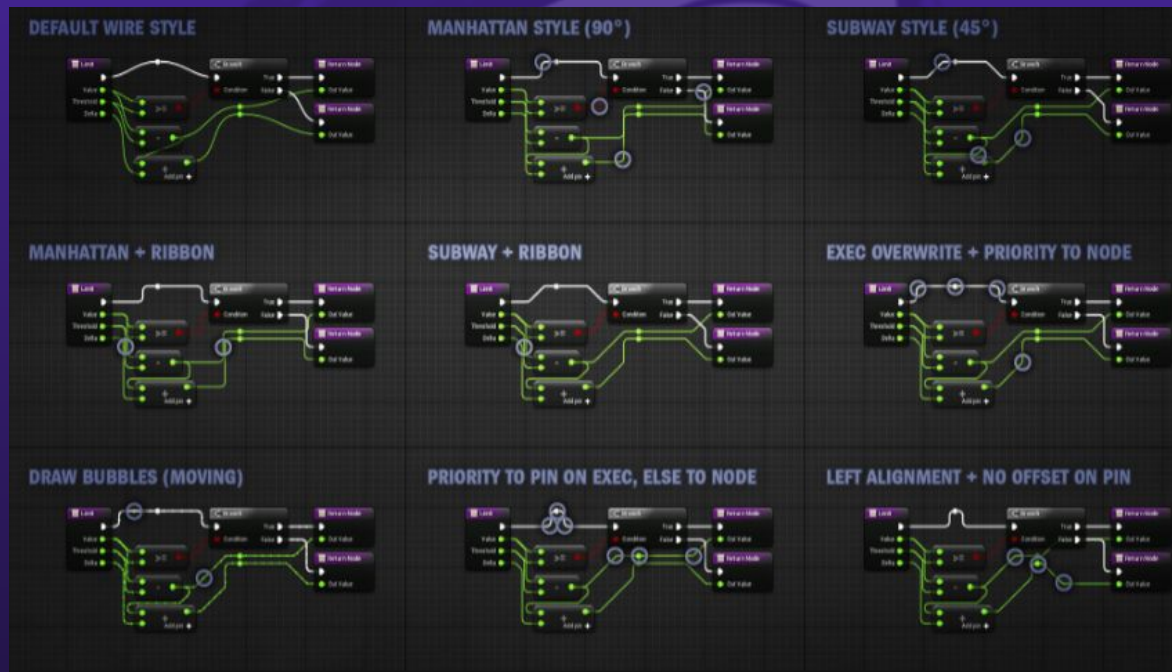
# 「Electronic Nodesを使ったほうがいい人」

- ・個人で開発をすることが多くて、ノードの可視性をめっちゃ気にする人
- ・ノード見やすくするための修正の時間を無駄だなと考えている人
- ・開発をしていて気が付いたらスパゲッティノードになっている人。

# 「Electronic Nodesを使う際に覚えておくこと」

- ・1人でElectronic Nodesをプロジェクトに有効にするのは全然OK。
- ・複数に開発でElectronic Nodesを適用する場合だと全員が同じ環境(共有するプロジェクトにElectronic Nodesが有効にされている)じゃないとプラグインを入れていないプロジェクトとの差異があり、少しグチャっとなるので注意。
- ・もし複数にでElectronic Nodesを使用したい場合は、全員がElectronic Nodesを購入してください。
- ・上記の理由があるため、個人での開発や小規模開発での開発におすすめです。

# 「UE4 Marketplaceで検索！」



A large, stylized, light blue letter 'S' is centered on the page. The 'S' has a thick, brush-stroke-like texture and is set against a dark blue circular background that has a slightly distressed or metallic appearance. The overall background of the slide is a solid dark blue.

おまけ終わり！



A large, stylized, light blue letter "S" is centered on the page. It is set against a dark blue circular background that has a textured, slightly distressed appearance. The "S" itself has a thick, brush-stroke-like quality.

最後に！

A large, faint, stylized letter "S" is centered in the background. It is a light blue color and has a thick, brush-stroke-like texture. The "S" is set against a dark blue circular backdrop that has a slightly distressed or metallic appearance.

「個人的にチーム開発で大事だと思うこと」

# 「チーム開発で大事なこと」

- ・1.チームプレイを意識する
- ・2.1人1人が自分の役割を意識する
- ・3.困ったときはみんなで対応

# 「1.チームプレイを意識する」

- ・1人で開発をしていると勘違いをしないで欲しい。
- ・基本的に報・連・相をしっかりする。
- ・自分が何の作業をしているかをチームのみんなと共有する。  
→共有出来ていないとプロジェクト進捗の管理不具合・急なバグが出て困ったり  
想定外のことがよく起こります。

## 「2.1人1人が自分の役割を意識する」

- ・何でも良いはだめ。(作業分担の時・ゲームの企画を考える時)
- ・チームのみんなに頼るのではなくて、自分が今何をすればこのチームにとって良い結果を生むのかを考えて行動する。

## 「3.困ったときはみんなで対応」

- ・何か作業をしている時に困ったらチームの誰かに相談しよう。
- ・作業に困って誰にも相談せずに過ごす1時間と誰かに相談して過ごす1時間とどちらがチームにとって良いか天秤にかけよう。
- ・自分がもし誰かに教育している立場だと考えて、学習している人が困っていて何も進捗がないまま1時間が過ぎるのと、困ったことがあってすぐに相談してきてくれるのでは全然違う。
  - 何が違うか？
  - その時点からその学習を続けるかどうかの判断ができる。
  - アドバイス(違った視点で助言)ができる。
  - 制限時間が決まっているなら、制限時間から逆算して判断できる。

A large, faint, stylized letter 'S' is centered in the background. It is a light blue or white color, matching the text. The 'S' is thick and has a slightly irregular, hand-drawn appearance. It is set against a dark blue background with a subtle, darker blue circular pattern that looks like a stylized globe or a planet with a ring.

良きゲームライフを！！！！

A large, faint, light blue stylized letter 'S' is centered in the background. The 'S' is composed of thick, flowing strokes and is enclosed within a circular border that has a textured, stone-like appearance.

ご清聴ありがとうございました