

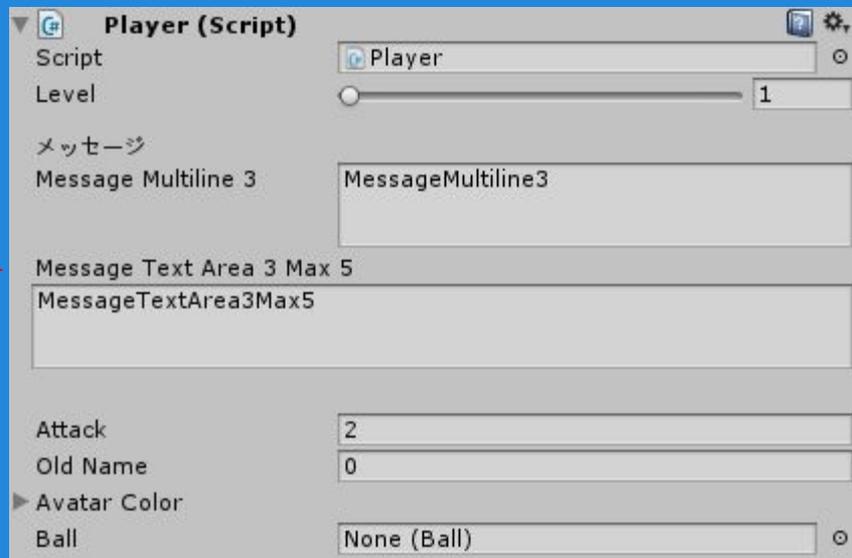
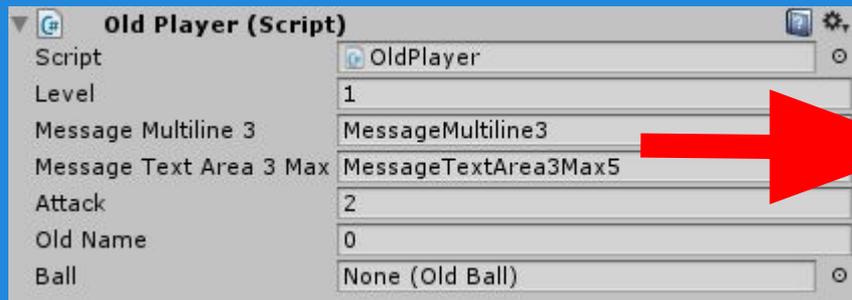
Unity

エディタ拡張入門

属性(追加情報)を与えるだけで便利になる！
すぐ使える、かんたんエディタ拡張！

例えば、どんな感じ？

これが…



こうなる…！

属性(Attribute)とは？

C#の機能の一つ (正確には.NET Frameworkの機能)

クラスやメンバに対して、追加情報を与えられる。自作も可能。

- 記述例 →

```
[Range(0, 100)]  
public int Level;
```


[RangeAttribute(0, 100)] と書いても一緒。
- 属性の定義方法 (もっと拡張したい人向け)
System.Attribute を継承して定義する(C#の話)。
Range は UnityEngine.Attribute を継承している(Unityの話)。
Unityではこちらを使用する。

見た目から変えてみよう (Player.cs編)

属性	内容
Range(0, 100)	0~100のスライダー入力。数値型のみ。
Header(“メッセージ”)	ヘッダー文を表示。
Multiline(3)	複数行入力。文字列型のみ。基本 TextAreaを使う。
TextArea(3, 5)	Multilineの自動改行、スクロールバー表示付き。
Space(20)	縦に余白を入れる。
ToolTip(“固定ダメージを与える攻撃力”)	カーソルを当てた時にツールチップが表示。
ContextMenu(“Reset”, “ResetAttack”)	メンバを右クリックして、ResetAttackを実行できる。
FormerlySerializedAs(“oldName”)	メンバ名を変更するときに、変更した値を引き継ぐ。

構造体とカラーピッカー(AvatarColor.cs編)

属性	内容
System.Serializable	シリアライズしてインスペクタに表示させる。
ColorUsage(false)	カラーピッカーのアルファ値を非表示。
ColorUsage(true, true, 0, 3, 0.5f, 2)	HDRに対応したカラーピッカー。 BrightnessとExposureの範囲を指定。

非表示とメニュー追加 (Ball.cs編)

属性	内容
SerializeField	privateメンバをインスペクタに表示。
NonSerialized	publicメンバをインスペクタから非表示。値も保持しない。
HideInInspector	publicメンバをインスペクタから非表示。値は保持する。
ContextMenu("Random")	コンポーネントを右クリックして、メソッドを実行できる。
MenuItem("Example/Create Ball 10")	メニューバーからメソッドを実行できる。静的メソッドのみ。

クラスに指定する属性 (Ball.cs編)

属性	内容
<code>RequireComponent(typeof(Rigidbody))</code>	指定したコンポーネントを強制アタッチ。
<code>DisallowMultipleComponent</code>	同一オブジェクトに複数アタッチ禁止。
<code>AddComponentMenu("Example/Ball")</code>	Add Componentのメニューでの場所を指定。
<code>SelectionBase</code>	シーンビューで選択時に子より優先で選択される。
<code>ExecuteInEditMode</code>	プレイせずエディタ上で AwakeやUpdateが実行される。

おまけ (Ball.cs編)

GetComponentの結果をキャッシュする

- GetComponent を頻繁に使うのは無駄な処理。
- gameObject.transform は内部でキャッシュされているが、変数にキャッシュした方が若干高速。

LINQ(統合言語クエリ)は超便利

- foreachが使えるところはLINQが使える。
(IEnumerable を実装するコレクションが対象)
- iOSではLINQでエラーになる場合がある。(AOTの問題)

おしまい

他にもファイル追加時やビルド直後にメソッドを実行するなども属性を与えるだけで簡単にできるよ！自作をすればインスペクタの表示まるごと変更することも！可能性無限大！