



Unityはじめるよ

～SimpleAnimationを使ってみた～

統合開発環境を内蔵したゲームエンジン
<http://japan.unity3d.com/>

※いろんな職業の方が見る資料なので説明を簡単にしてある部分があります。正確には本来の意味と違いますが上記理由のためです。ご了承ください。
この資料内の一部の画像、一部の文章はUnity公式サイトから引用しています。

資料の内容

- SimpleAnimationって？
- 使い方
- まとめ

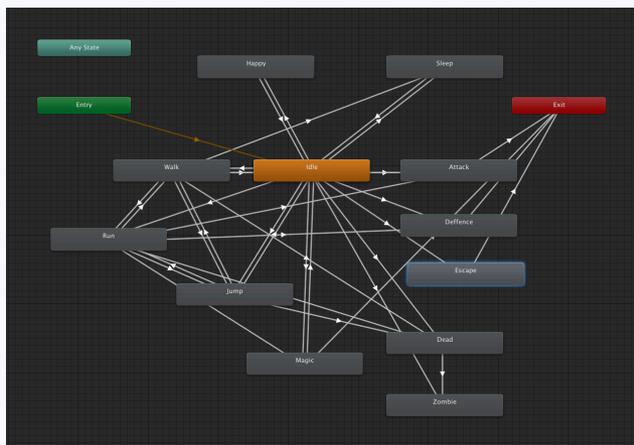
SimpleAnimationって？

スクリプトから簡単にアニメーションを切り替えるための
コンポーネント。

AnimatorControllerがあるのになぜ必要？

1、ステート(状態の)管理がスクリプト側の1箇所で済む。

AnimatorControllerを使う場合は、スクリプトとAnimatorControllerの2重管理となってしまう。



+

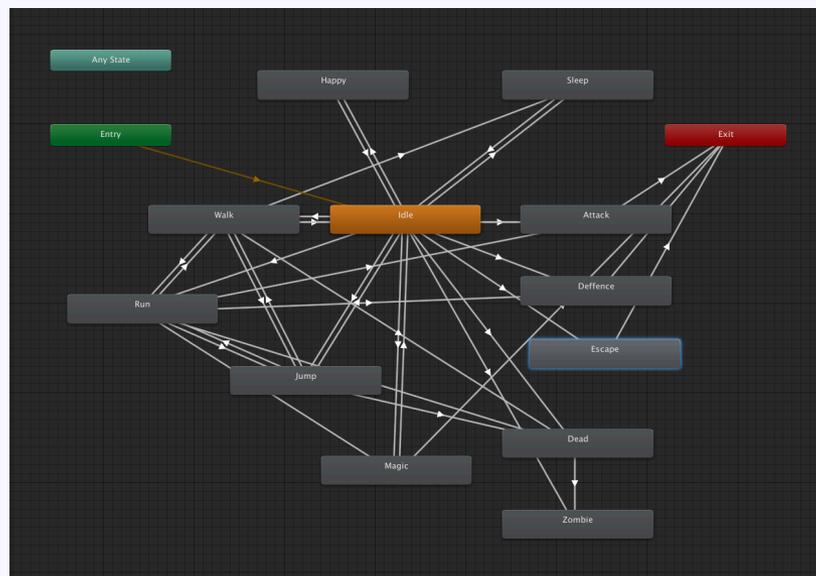
```
enum AnimationState
{
    Idle,
    Move,
    Jump,
    Sleep,
    Attack,
    Deffence,
    Escape,
    Magic,
    Happy,
    Zombie,
    Dead
}

const string kAnimationParamaterMoveSpeed = "MoveSpeed";
const string kAnimationParamaterJump     = "Jump";
const string kAnimationParamaterSleep    = "Sleep";
const string kAnimationParamaterAttack   = "Attack";
const string kAnimationParamaterDeffence = "Deffence";
const string kAnimationParamaterEscape   = "Escape";
const string kAnimationParamaterMagic    = "Magic";
const string kAnimationParamaterHappy    = "Happy";
const string kAnimationParamaterZombie   = "Zombie";
const string kAnimationParamaterDead     = "Dead";
```

2重管理苦 . . .

2、プログラマーならシンプルでわかりやすい

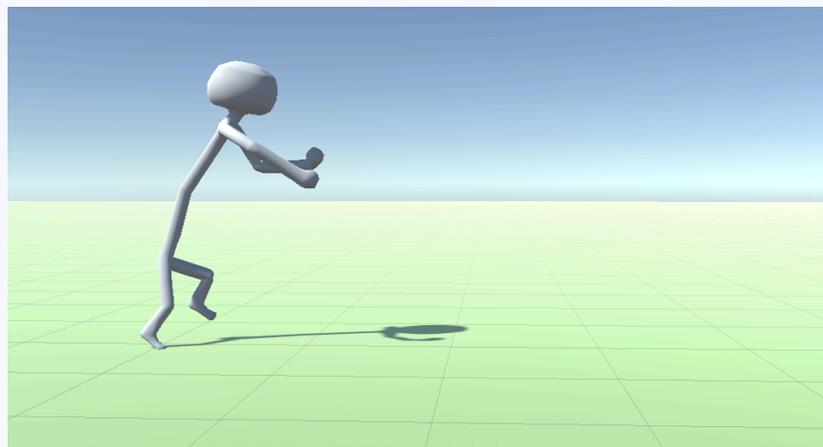
AnimatorControllerを使うステートマシンは、状態が見える化されていてわかりやすいが、ステートが増えてくると、線がぐちゃぐちゃして逆にわかりづらいものとなる。



見える化が仇に・・・

3、狙った通りに遷移する

AnimatorControllerのTrigger型は、一見シンプルで使いやすそうだが、狙った通りに遷移せず？？？となったことがあるのではないだろうか。



あれ？ジャンプしない・・・

こういった問題を解決するために生まれたのが

SimpleAnimation

である。

使い方

1、Githubで公開されているのでダウンロード。

■ SimpleAnimationダウンロード先

<https://github.com/Unity-Technologies/SimpleAnimation>

Unity-Technologies / SimpleAnimation

Watch 39 Star 214 Fork 34

Code Issues 14 Pull requests 4 Projects 0 Wiki Insights

Join GitHub today

GitHub is home to over 28 million developers working together to host and review code, manage projects, and build software together.

Sign up

A simple Animation Component that leverages PlayableGraphs

27 commits 4 branches 0 releases 2 contributors MIT

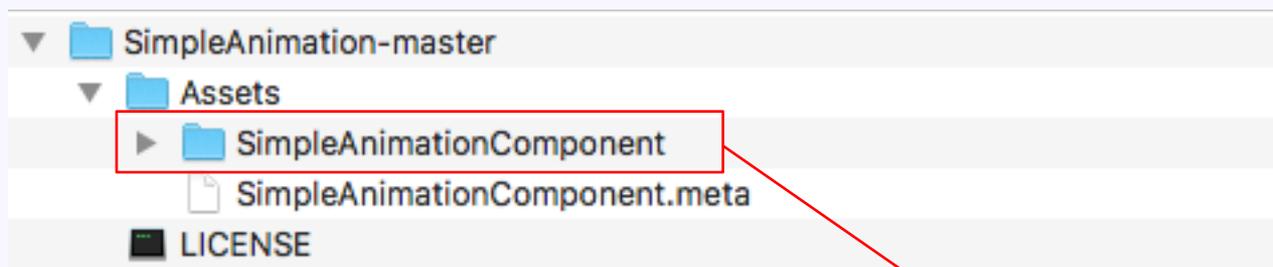
Branch: master New pull request Find file Clone or download

Assets	[Issue #24] Long due refactor of StateInfo to fix hard-to-debug state...	5 months ago
.gitignore	Fixed an issue with the automatic naming of the Simple Animation Comp...	10 months ago
LICENSE	Initial commit	11 months ago

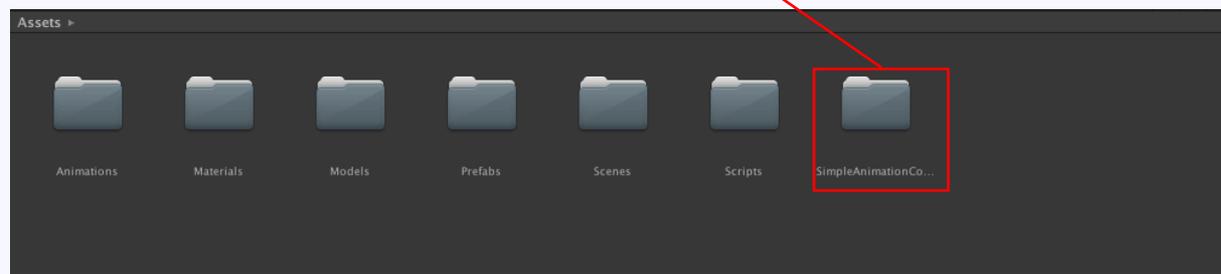
© 2018 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

ここから

2、ダウンロードしたZipを解凍し、その中にある SimpleAnimationComponentフォルダごとを Unityプロジェクトにインポート。

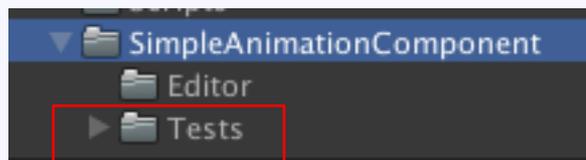


プロジェクトウィンドウに
ドラッグ&ドロップで
インポート



3、そのままだとエラーが出ているので対処する。

2018の場合Testsフォルダを削除する方法で対処。



削除しちゃえ



2017の場合、

Window > Test Runner > PlayMode > Enable playmode tests

でエラーが消えます。

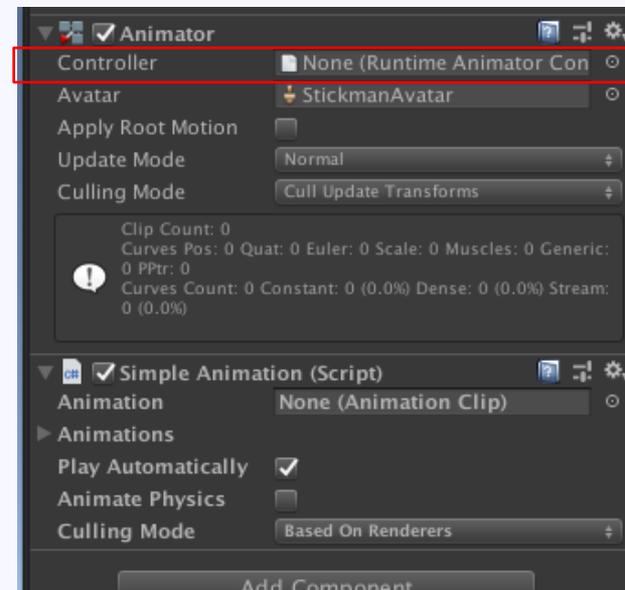
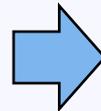
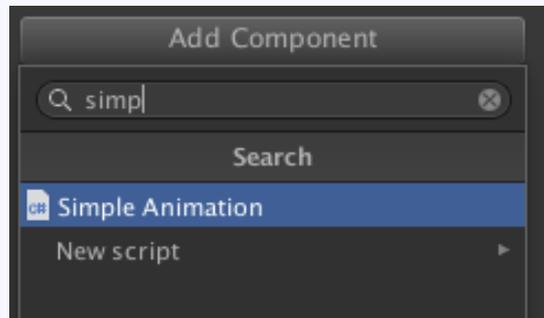
Testsフォルダを削除する方法でもOK。

4、SimpleAnimationコンポーネントをアタッチ

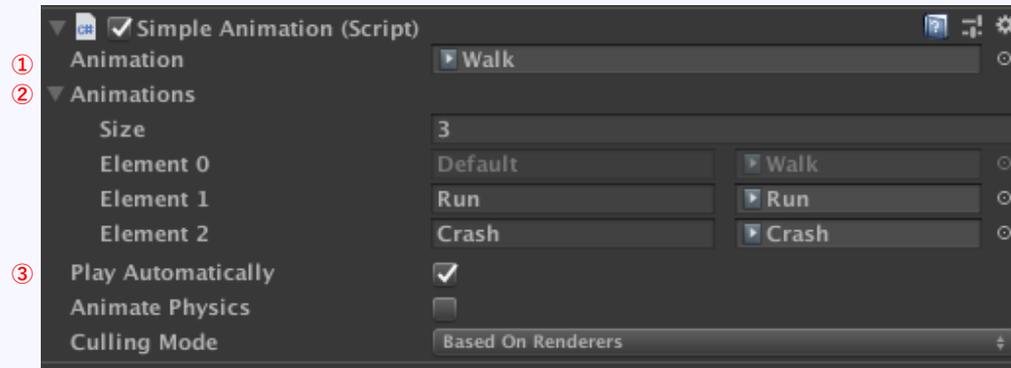
インスペクタのAddComponentボタンからSimpleAnimationを選択。

Animatorコンポーネンも自動追加。

Animatorコンポーネントにはコントローラーは不要。



SimpleAnimationコンポーネントの詳細



- ① デフォルトのアニメーション
- ② 複数のアニメーションを扱う場合は、**アニメーション名**と**アニメーションクリップ**を登録しておく
 - 左の欄にアニメーション名を指定
 - 右の欄にアニメーションクリップを指定
- ③ 自動再生を行うか

5、スクリプトで操作

まずはシンプルに、アニメーションを指定して再生する方法。

Play関数を使う。引数に**アニメーション名**を指定。

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class StickmanController : MonoBehaviour {

    // SimpleAnimationコンポーネント
    SimpleAnimation m_simpleAnimation;

    // Use this for initialization
    void Start () {
        m_simpleAnimation = GetComponent<SimpleAnimation> ();
        m_simpleAnimation.Play ("Run");
    }
}
```

SimpleAnimationコンポーネントを取得して
走るアニメーションを指定

滑らかに別のアニメーションに切り替える方法。

CrossFade関数を使う。

1つ目の引数にアニメーション名

2つ目の引数にフェード時間

を指定

```
// Update is called once per frame
void Update () {

    // 1キーが押されたらデフォルト 歩き
    if (Input.GetKeyDown(KeyCode.Alpha1))
    {
        m_simpleAnimation.CrossFade("Default", 0.5f);
    }

    // 2キーが押されたら走る
    if (Input.GetKeyDown(KeyCode.Alpha2))
    {
        m_simpleAnimation.CrossFade("Run", 0.5f);
    }

    // 3キーが押されたらクラッシュ
    if (Input.GetKeyDown(KeyCode.Alpha3))
    {
        m_simpleAnimation.CrossFade("Crash", 0.5f);
    }
}
```

まとめ

めちゃくちゃシンプル！

再生したいクリップを指定して、Play関数、CrossFade関数を呼ぶだけとは。

しかも、呼べば必ずアニメーションが切り替わるという点が何よりも素敵。

Animatorコントローラーには戻りたくないかも。

ただ、まだ調査不足なので、上半身と下半身を別々の動き位にするとか、IKとの連携とか、気になるところは多い。

参考サイト

Unity公式

<https://github.com/Unity-Technologies/SimpleAnimation>

(:3[kanのメモ帳]さん： AnimatorControllerを使わずに簡単にアニメーションを再生出来るSimpleAnimation 【Unity】
<http://kan-kikuchi.hatenablog.com/entry/SimpleAnimation>

@yos316さん： 【Unity】 SimpleAnimationについて

<https://qiita.com/yos316/items/3c0df9c296d7cf4aa605>

ご清聴ありがとうございました