

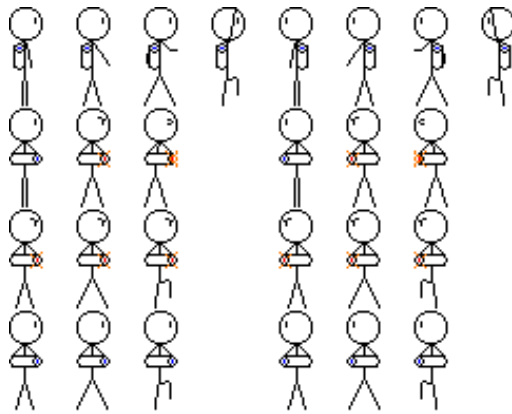
# Unity 2 D 講習第 4 回

## 1. アニメーション

ゲームはだいたい形になりました (バグもありますしゲームクリアも作ってませんが、自分でどうにかできるはずです)。ここからはプレイヤーにモーションをつけていきたいと思います。

① 作業をする前にモーション＝アニメーションの仕組みを知っておきましょう。アニメーションは、パラパラ漫画のように、画面に映る絵を前の絵とすこしずらした絵にすることで動いたように見せる技術です。Unity には Animation という機能があり、これでスプライトを変更することでキャラが動いているように見せます (ボーンアニメーションというものもありますが高度なので今回は割愛)。

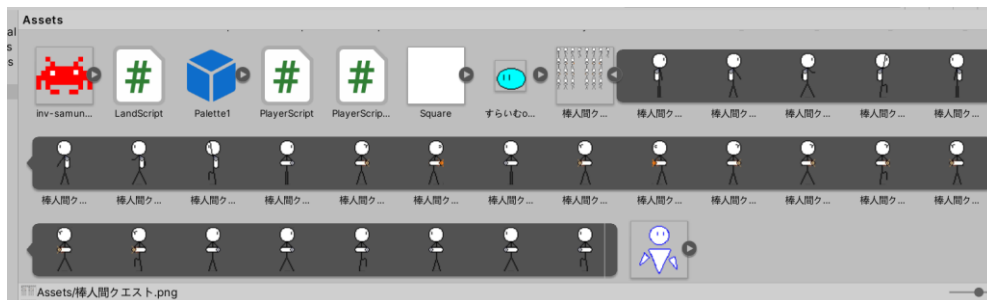
② ではモーションをつくっていきましょう。まずは絵を用意します。



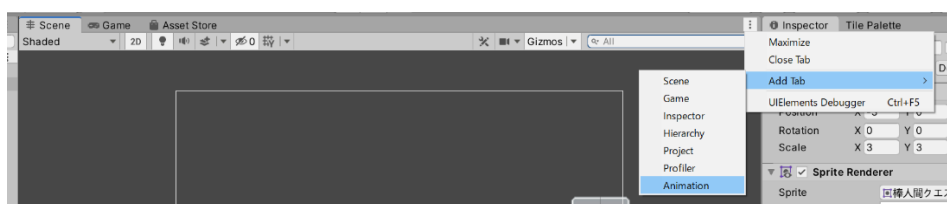
(絵が下手なのはごめんね)

立っている絵、歩いている絵、ジャンプしている絵、攻撃している絵など、欲しい絵を用意してください。絵のサイズは同じだとよいです。また、歩いている絵は2種類以上あるとよいです (足を出している絵とその途中の絵など)。上の画像はそれぞれに左右反転の絵も用意していますが、これは用意しなくても大丈夫です。

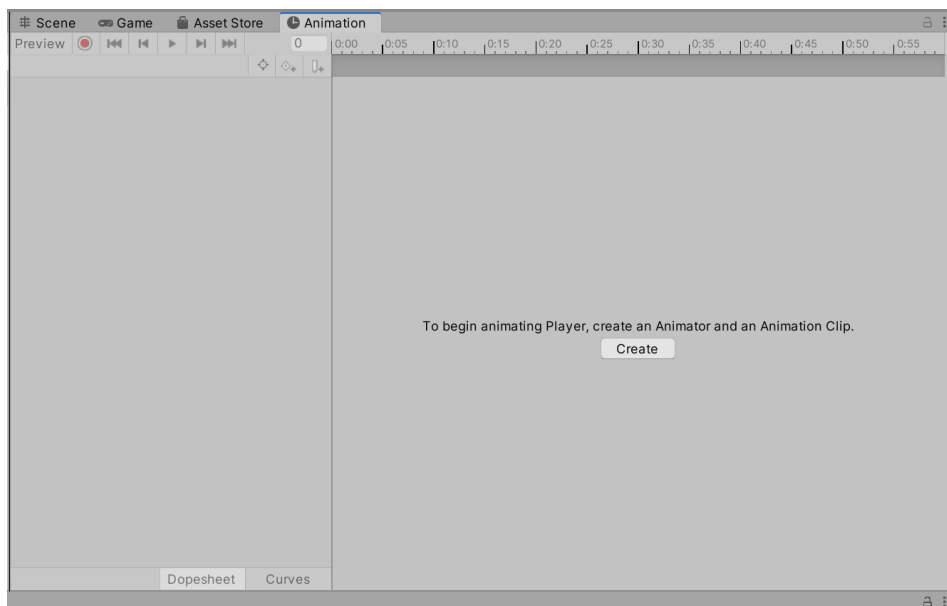
③ この絵をプロジェクトに放り込み、分割します。



- ④ シーンビュー右上の：ボタン（点は本当は3つ）を押して、Add Tab→Animationを選択します。

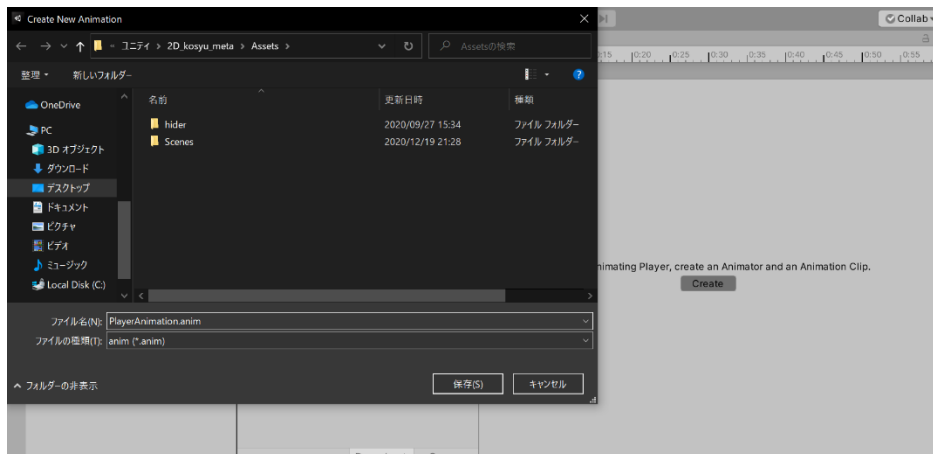


すると Animation タブができます。これはインスペクターやシーンビューなどと同系列のものです。

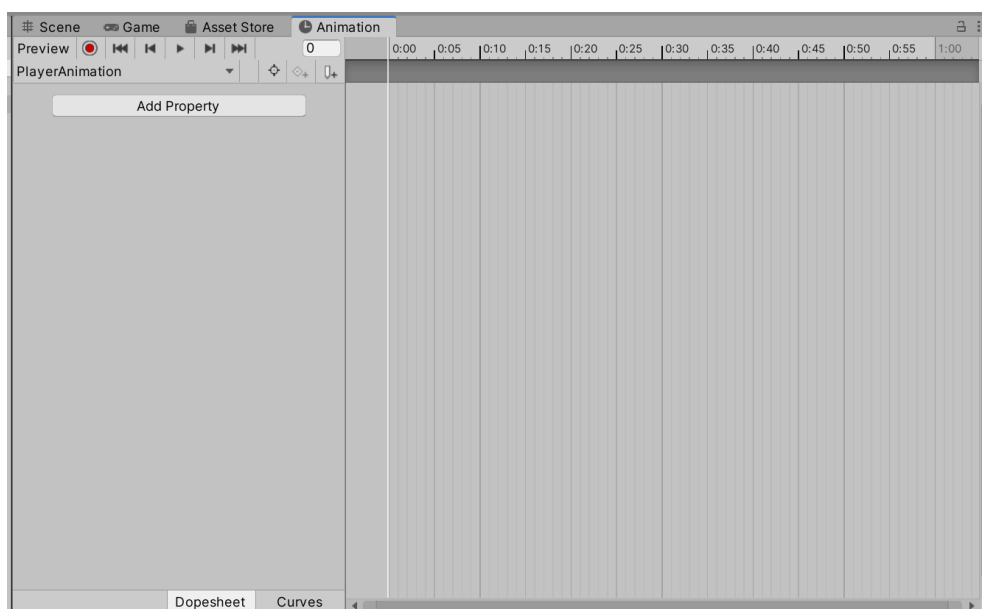


ゲームビューにもどりたときは、左上の Scene ボタンを押すともどれます。

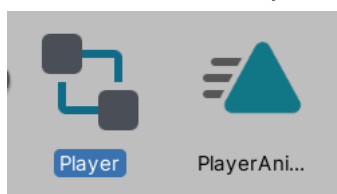
- ⑤ Player を選択した状態で、Animation タブの真ん中にある create ボタンを押してください。すると保存画面にうつるので、PlayerAnimation.anim という名前で保存してください。



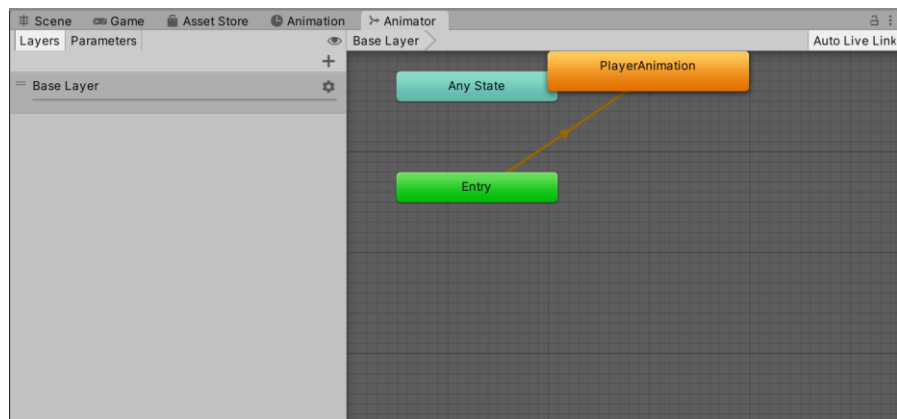
すると Animation タブの中身が変化します。



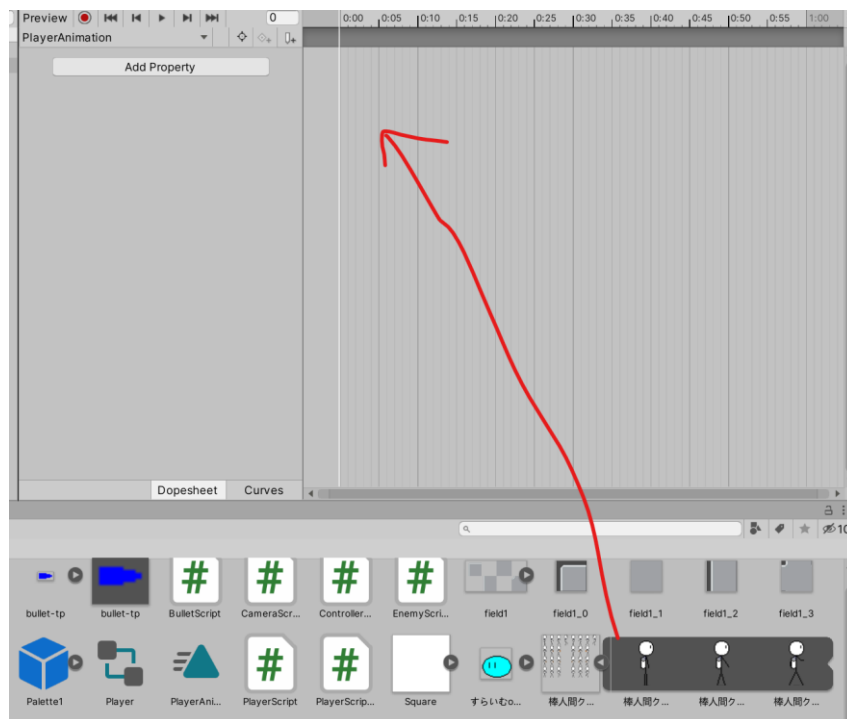
- ⑥ また、プロジェクトに Player と PlayerAnimation という名前のファイルができて  
いると思うので、Player の方をクリックしてください。



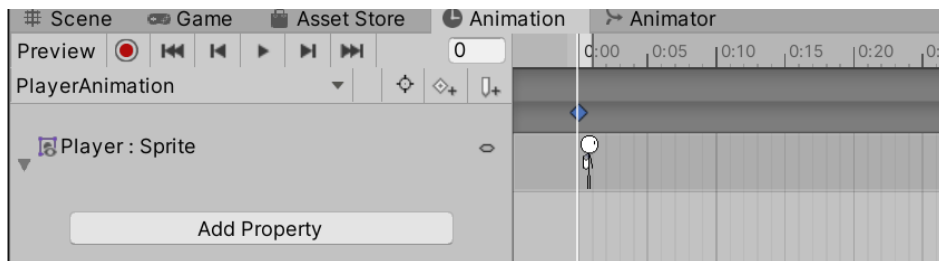
すると Animator という名前のタブが開かれます。



- ⑦ ここまできたところで Animation 機能について説明します。まず、Animation というのは、スプライトを何秒かごとに切り替えて一つのまとまった動きをあらわすものです。Animation は一つのオブジェクトに対して複数つくることができ、役割ごとに複数つくるのが基本です（歩いているときのモーション、ダメージを受けたときのモーションなど）。そして、Animation を管理するものがあり、それが Animator です。主に Animation の切り替えをします。では実際にやってみましょう。
- ⑧ まずは止まっているモーションを作ります。Animation タブに移動し、止まっているときに表示したいスプライトをタブの右側にドラッグアンドドロップしてください。

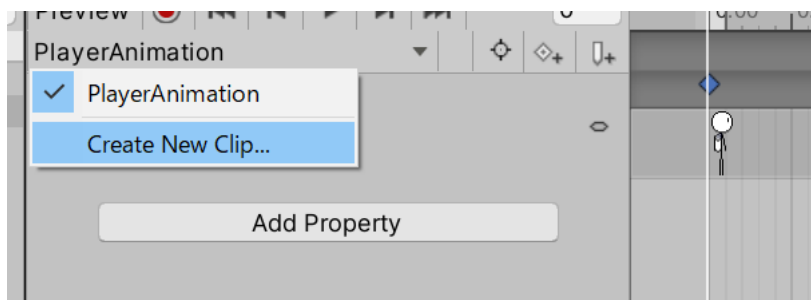


すると以下ようになります。



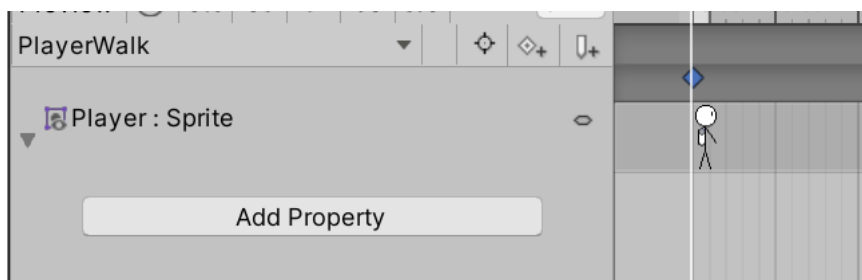
これで、PlayerAnimation の一番最初の絵が登録されました。この画面は時間経過とその時のスプライトを表示するタイムラインなので、0秒目に止まった絵が表示されることになります。止まっているときにキャラは動かない（待機モーションとかもありますが今回はそういう仕様にしました）ので、止まっているモーションはこれで完成です。

- ⑨ 次に歩いているときのモーションを作ります。タブ左側の PlayerAnimation と書かれた部分の右側の▼をクリックし、Create new clip を選択→PlayerWalk という名前で保存してください。

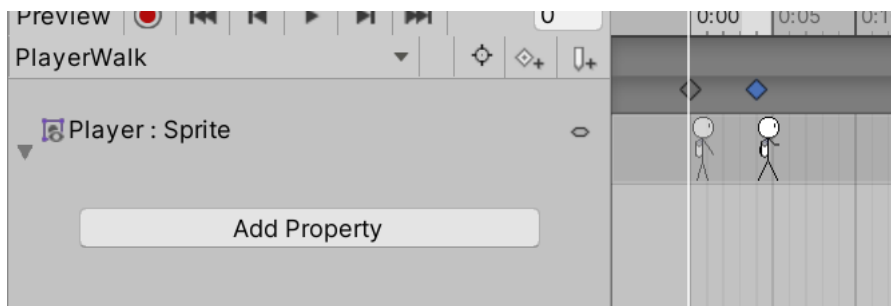


すると編集している Animation が PlayerAnimation から PlayerWalk になります。先ほどのモーションをもう一度編集したいときは、今の▼を押して編集したい Animation へ移動してください。

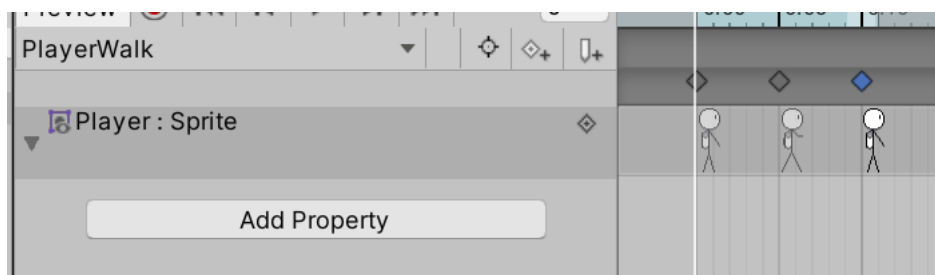
- ⑩ 歩いているときのスプライトを一つドラッグアンドドロップしてください。



- ⑪ ⑩とは別の歩いているときのスプライトをその後ろ側にドラッグアンドドロップしてください。



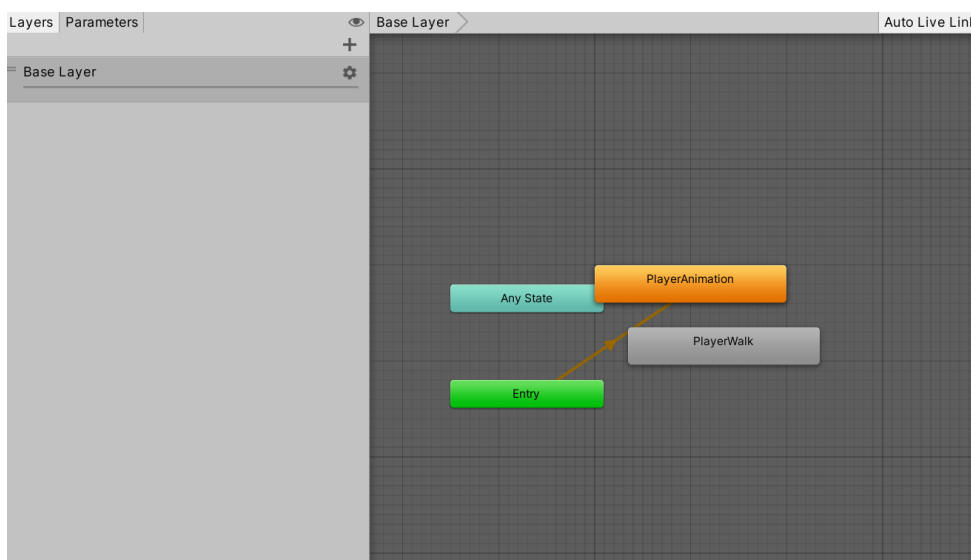
- ⑫ ⑩と同じスプライトをそのさらに後ろにドラッグアンドドロップしてください。



これは、この Animation を再生したときに最後に来たら一瞬で最初に戻ってしまい、最後に配置したスプライトが見えなくなってしまうからです。

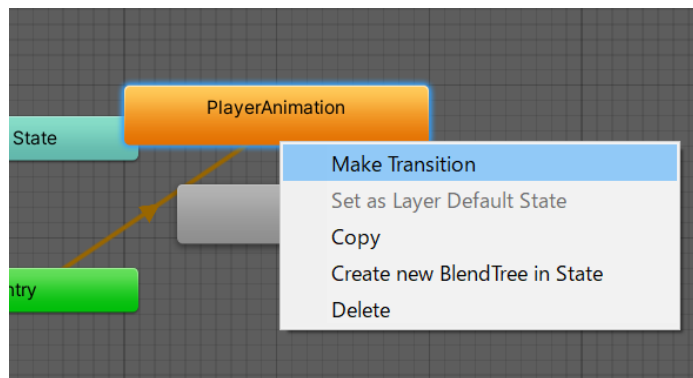
これで歩きモーションは完成とします。

- ⑬ ではこれを Player に適応してみましょう。Animator タブに移動します。

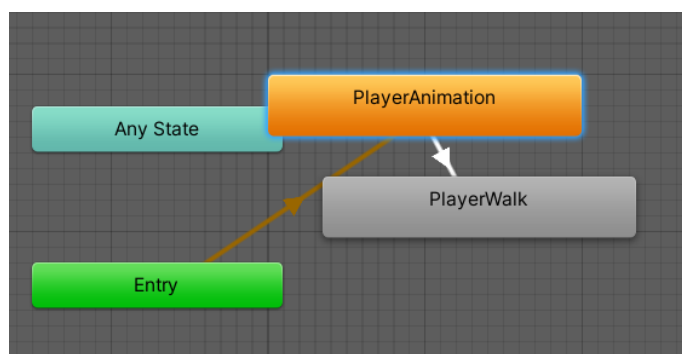


この画面では先ほど作った Animation をどのようなときに切り替えるのかの設定をします。Entry から PlayerAnimation に矢印が出ていますが、これは始まった時は PlayerAnimation の状態にする、という意味です。

- ⑭ この画面ではある Animation から別の Animation への遷移を矢印で表します。PlayerAnimation を右クリックし、Make Transition をクリックした後、PlayerWalk をクリックしてください。



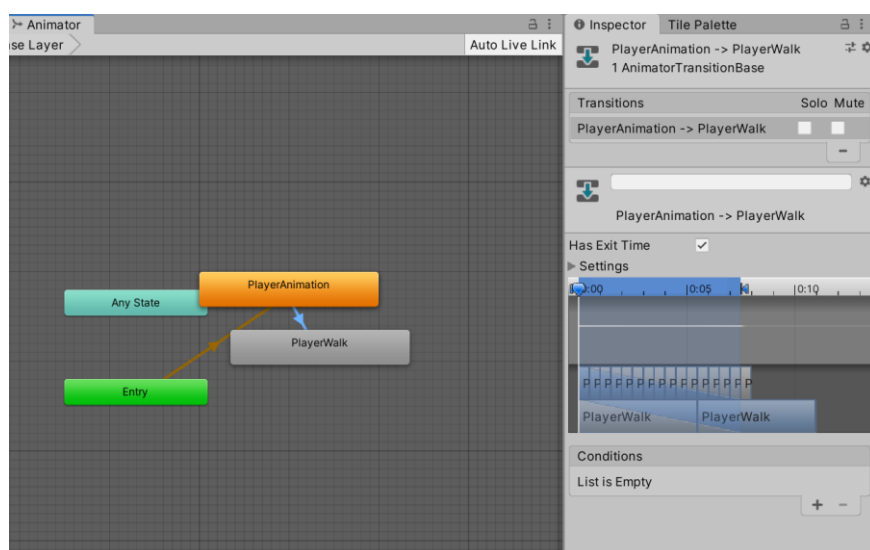
すると矢印が作られました。



- ⑮ ここでゲームを実行してみましょう。Player が PlayerWalk で設定したように動いているのがわかるはずです。これは、ゲームが始まった後 PlayerAnimation を再生し、その再生が終わったら PlayerWalk を再生したことになります。

## 1. アニメーション 2

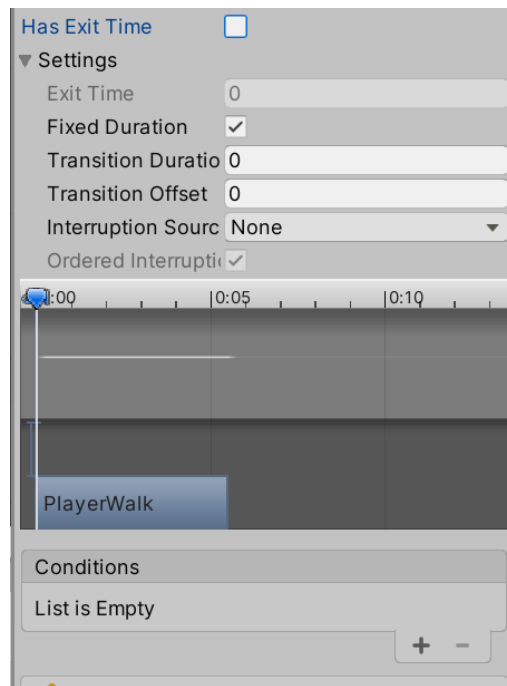
- ① 歩いているときだけ PlayerWalk を表示してもらいたいのので、そのように設定をします。まずは Animator タブの先ほど作った矢印をクリックしてください。



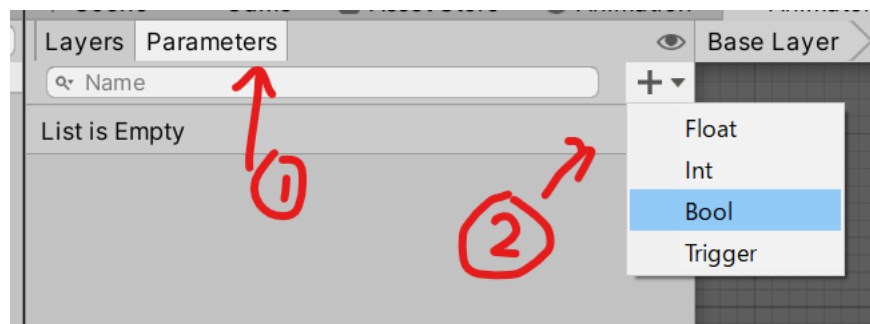
インスペクターに矢印の設定が出ました。これをいじることでアニメーション遷

移の設定が行えます。

- ② まずは Has Exit Time のチェックをオフにしてください。これは「アニメーションが終わったら次のアニメーションに遷移する」という意味です。
- また、Settings を開いたところにある Transition Duration を 0 にしてください。  
この項目は 3D の時に使うものです。



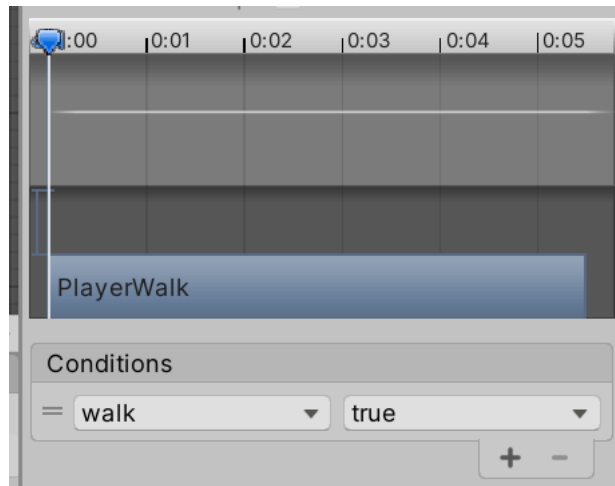
- ③ いまのままで遷移する条件がないので、それをつくります。右上の Parameter をクリックして、左下に出てきた + ボタンをクリックして、Bool を選択してください。すると名前を付けるように言われるので、walk としてください。



これは、Bool 型の変数 walk を作ったということになります。この walk が true の時に PlayerWalk のアニメーションを再生し、false の時に PlayerAnimation を再生することにします。

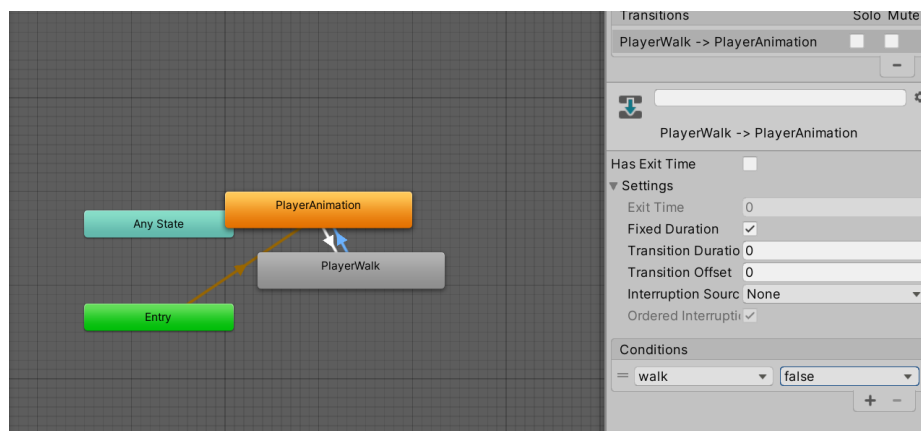
- ④ 矢印の設定の Conditions の下にある + ボタンを押してください。すると以下のようになります。





Conditions は何がどうなったらアニメーションを遷移させるかの条件の設定です。ここでは変数 walk が true になったらアニメーションが遷移する、という意味です。

- ⑤ このままだと PlayerAnimation から PlayerWalk に遷移すると、PlayerAnimation にもどることができません。そのため PlayerWalk から PlayerAnimation に新しい矢印を作ってもどれるようにします。



先ほどと同じように矢印を作り、設定を上画像のように変更してください。先ほどの違いは Conditions が false になっていることです。

- ⑥ これで変数 walk の値でアニメーションを管理することができるようになりました。しかし、変数 walk は勝手には変わらないので、スクリプトから変えてやる必要があります。PlayerScript を編集します。

```

public class PlayerScript2 : MonoBehaviour
{
    private Rigidbody2D rigid;
    private LandScript landSc;
    private Animator anime; //変更
    private float XForce;
    private float YForce;
    public float gravity;
    public float jumpForce;
    public float Speed;
    public int HP = 10;
    public GameObject bullet;
    // Start is called before the first frame update
    void Start()
    {
        anime = GetComponent<Animator>();
        rigid = GetComponent<Rigidbody2D>();
        landSc = GetComponentInChildren<LandScript>();
    }

    // Update is called once per frame
    void Update()
    {
        XForce = Speed * Input.GetAxis("Horizontal");
        if(Input.GetAxis("Horizontal") != 0){ //変更
            anime.SetBool("walk",true); //
        }else{ //
            anime.SetBool("walk",false); //
        } //ここまで
        if(landSc.IsLand){
            if(Input.GetKey(KeyCode.UpArrow)){

```

変数 anime にアニメーターの情報を取得し、SetBool という命令で walk の値を変えています。

anime.SetBool(“パラメータの名前”,値)でパラメータの値を変えられます。

- ⑦ 実行すると、プレイヤーが移動しているときだけ歩くアニメーションをしています。これでアニメーションの基本は完成です。

これで Unity 2 D 講習は終わりです。

## ここまで来たあなたに

講習お疲れ様でした！この資料を最後まで読み切ったあなたはある程度の Unity の実力を得たはずです。そこで、最後に実力を確かめる課題を出します。この課題の提出をもって講習を卒業したと認めます。いままでの資料を見ながらも、ネットや本を見ながらも

よいので、やって提出してみてください。

#### 課題

この Unity 2D 講習で作ったゲームは多くの改善点があります。自分が気になった点を改善し、私に見せてください。

(改善点の例：敵の上に乗ると行動不可能になる、ジャンプのアニメーションがない、など)