

### 【Step3】 01\_アニメーションを使ってみよう 小テスト

#### 1. アニメーションを作る際の以下の説明の中から、正しいものを全て選びなさい。

- シーンビューにドラッグ&ドロップする方法でアニメーションを作った場合、アニメーションと一緒にアニメーターコントローラーも作られる
- スプライト（画像）を複数選択して、シーンビューにドラッグ&ドロップするとアニメーションを作ることが出来る
- スプライト（画像）を1つ選択して、シーンビューにドラッグ&ドロップするとアニメーションを作ることが出来る

アニメーションを作る方法は他にもありますが、このやり方でアニメーションを作成した場合、ゲームオブジェクトと一緒に作成されましたね。そのおかげで、アニメーションや、サイズの確認がしやすくなるので、レッスンではこの作り方を覚えました。

#### 2. アニメーターに関連する以下の説明の中から、正しいものを全て選びなさい。

- Animatorコンポーネントの「コントローラー」にアタッチできるのは、アニメーターコントローラーである
- Animatorコンポーネントの「コントローラー」にアタッチできるのは、アニメーションである
- アニメーターコントローラーをダブルクリックすると「アニメーターウィンドウ」を表示することが出来る
- 問1の方法でアニメーションを作った時、「Animatorコンポーネント」がアタッチされたゲームオブジェクトが作成される

アニメーションを再生するには、Animator コンポーネントの「コントローラー」に、アニメーターコントローラーがアタッチされている必要があります。「アニメーターウィンドウ」は、他のウィンドウを表示した時と同じように、Unity エディタ上部のウィンドウ>アニメーション>アニメーターからも表示することが出来ます。

## 【Step3】 02\_プレイヤーが左右に移動するときにアニメーションを付けよう 小テスト

1. 以下のようなスクリプトを作成しましたが、実行してもエラーが出てしまいます。

```
using UnityEngine;
public class PlayerController : MonoBehaviour
{
    Rigidbody2D rigidbody2D;
    Animator anim;
    public float moveSpeed;
    float vx;

    void Start()
    {
        rigidbody2D = this.GetComponent<Rigidbody2D>();
        anim = this.GetComponent<Animator>();
    }

    void Update()
    {
        if (Input.GetKey("right"))//右が押されている時
        {
            vx = moveSpeed;
            animator.Play(RunAnimation);
        }
        else if (Input.GetKey("left"))//左が押されている時
        {
            vx = -moveSpeed;
            animator.Play(RunAnimation);
        }
        else//右も左も押されていない時
        {
            vx = 0;
            animator.Play(idleAnimation);
        }

        rigidbody2D.velocity = new Vector2(vx, rigidbody2D.velocity.y);
    }
}
```

正しくアニメーションの切り替えが出来るようにしたい時、修正しなければならないものを全て選びなさい。

※アニメーターコントローラーには、「runAnimation」と「idleAnimation」という名前のアニメーションが登録されているものとする

- 「animator.Play」を「anim.Play」に変更する
- Play()の中のアニメーション名を、「」(ダブルクォーテーションマーク) で囲む
- 「RunAnimation」を「runAnimation」に変更する
- 「idleAnimation」を「IdleAnimation」に変更する

このスクリプトでは、Animator コンポーネントを「anim」という変数の中に代入しているので、anim.Play()としなければ Animator コンポーネントのメソッドを使えません。また、Play()の中のアニメーション名は、アニメーターコントローラーに、登録したアニメーションの名前と同じである必要があります。アニメーション名を、"" (ダブルクォーテーションマーク) で囲って、入力してくださいね。今回の場合は、「anim.Play("idleAnimation");」と「anim.Play("runAnimation");」が正しいコードの書き方になります。

2. ゲームオブジェクトの向きを、コライダー(トリガー)ごと左右反転させる方法として、正しいものを全て選びなさい。※ゲームオブジェクトには、SpriteRendererコンポーネントがアタッチされているものとする

- Transformコンポーネントの回転の「X」を「0」→「180」に変更する
- Transformコンポーネントの回転の「Y」を「0」→「180」に変更する
- Transformコンポーネントの拡大/縮小の「X」を「1」→「-1」に変更する
- Transformコンポーネントの拡大/縮小の「Y」を「1」→「-1」に変更する
- SpriteRendererコンポーネントの反転の「X」にチェックを入れる
- SpriteRendererコンポーネントの反転の「Y」にチェックを入れる

SpriteRenderer コンポーネントの反転をつかうと、画像の見ただけが反転し、コライダー(トリガー)はそのままです。また、Transform コンポーネントの回転の「X」と「Y」は、X 軸に沿って回転するか、Y 軸に沿って回転するかを表しています。それに対し、Transform コンポーネントの拡大/縮小や、SpriteRenderer コンポーネントの反転の「X」と「Y」は、横(X)方向、縦(Y)方向という意味合いが強いので、混同しないように注意が必要です。

## 【Step3】03\_プレイヤーがジャンプするときにアニメーションを付けよう 1 小テスト

1. 以下のようなスクリプトを作成して、ゲームオブジェクトにアタッチしました。

```
using UnityEngine;

public class Test : MonoBehaviour
{
    int num = 3;

    void Start()
    {
        Debug.Log("A");

        if (num == 3)
        {
            Debug.Log("B");
        }
        else
        {
            Debug.Log("C");
        }

        Debug.Log("D");
    }
}
```

再生した時にコンソールウィンドウに表示されるアルファベットを、以下の選択肢の中から全て選びなさい。

- A
- B
- C
- D

(num == 3)という条件式は、「num」が「3」の場合「true」になり、「それ以外」の場合は「false」になります。今回「num」は「3」なので、if (num == 3){・・・}の中が実行され、下のelse{・・・}の中は実行されません。else{・・・}の中を通らないと、コンソールにCは表示されないなので、それ以外の、A・B・Dが正解となります。

2. 以下のようなスクリプトを作成して、ゲームオブジェクトにアタッチしました。

```
using UnityEngine;

public class Test : MonoBehaviour
{
    int num1 = 5;
    int num2 = 10;

    void Start()
    {
        if (num1 == 0)
        {
            Debug.Log("A");

            if (num1 == 5)
            {
                Debug.Log("B");
            }
            else
            {
                Debug.Log("C");
            }
        }
        else
        {
            Debug.Log("D");

            if (num2 == 10)
            {
                Debug.Log("E");
            }
            else
            {
                Debug.Log("F");
            }
        }
    }
}
```

再生した時にコンソールウィンドウに表示されるアルファベットを、以下の選択肢の中から全て選びなさい。

- A
- B
- C
- D
- E
- F

問 1 と同じように考えます。 `(num1 == 0)` という条件式は、「num1」が「0」の場合「true」になり、それ以外の場合は「false」になります。今回「num1」は「5」なので、最初の `if (num1 == 0){・・・}` の中は、実行されずに、下の `else{・・・}` の中が実行されます。この時点で、A、B、C はコンソールに表示されないことが分かります。B を表示する条件式は `(num1 == 5)` ですが、そこまでたどり着くことはできないですね。下の `else{}` の中では、まず「D」がコンソールに表示され、その後 `(num2 == 10)` が「true」となるので、「E」が表示されます。よって、D・E が正解となります。

**【Step3】 04\_プレイヤーがジャンプするときにアニメーションを付けよう 2 小テ**

**スト**

1. 以下の説明が正しい場合は「真」、間違っている場合は「偽」を選びなさい。

アニメーションウィンドウ→新しいクリップを作成 としてアニメーション (クリップ) を作成した場合、アニメーターコントローラーは自動で作成される

- 真  
 偽

複数のスプライトをシーンビューにドラッグ&ドロップしてアニメーションを作成した時とは違い、アニメーションコントローラーは自動で作成されません。

2. 以下のようなスクリプトを作成して、ゲームオブジェクトにアタッチしました。

```
using UnityEngine;
public class Test : MonoBehaviour
{
    int num = 10;

    void Start()
    {
        if (num > 0)
        {
            if (num < 5)
            {
                Debug.Log("A");
            }
            else
            {
                Debug.Log("B");
            }
        }
        else
        {
            Debug.Log("C");
        }
    }
}
```

再生した時にコンソールウィンドウに表示されるアルファベットを、以下の選択肢の中から選びなさい。

- A
- B
- C

外側の if の (num > 0) という条件式は、「num」が「0」より大きい場合「true」になり、「それ以外」の場合は「false」になります。今回「num」は「10」なので、(num > 0) は「true」となります。よって、この if の中を通るので、この時点で「C」が表示されないことが分かります。内側の if に移動しますが、(num < 5) という条件式は、「num」が「5」より小さい場合「true」になり、「それ以外」の場合は「false」になります。よって、(num < 5) は「false」となり、内側の if は通らず、内側の else を通るので、コンソールには「B」が表示されます。



### 3. コンソールウィンドウに「こんにちは」と表示するメソッドを作成しました。

```
void SayHello()
{
    Debug.Log("こんにちは");
}
```

このメソッドを実行したい時（呼び出したい時）、どのようにコードを書けば良いでしょうか。  
正しいものを選びなさい。

- SayHello(){ }
- SayHello()
- SayHello();
- SayHello(){ };
- SayHello;

メソッドを実行する（呼び出す）時は、メソッド名の後に()を忘れずにつけてください。{}は、必要ありません。そしていつも通り「;」も付けましょう。