

## 【Step4】 01\_ゴールを作ろう

このレッスンでは、ゴールまでたどり着いた時に、コンソールに「クリア」と表示させていきます。当たり判定を制限することのできる、「レイヤー」の使い方について学んでいこう！

### 【ポイント】

- 「OnTriggerEnter2D」と「OnCollisionEnter2D」  
どちらも物体同士の当たり判定に使用する。

#### 「OnCollisionEnter2D」

オブジェクトのコライダーが別のコライダーに触れたときに呼び出される  
オブジェクト同士は衝突する（すり抜けない）

#### 《使う条件》

- 判定したいオブジェクトすべてに Collider コンポーネントがアタッチされている
- 判定したいオブジェクトすべてに Rigidbody2D コンポーネントがアタッチされている
- 判定したいオブジェクトすべての Collider の「Is Trigger」のチェックが付いていない

#### 「OnTriggerEnter2D」

コライダーとトリガーが触れたとき、または、トリガーとトリガーが触れたときに呼び出される  
オブジェクト同士はすり抜ける

#### 《使う条件》

- 判定したいオブジェクトすべてに Collider コンポーネントがアタッチされている
- 判定したいオブジェクトの少なくともどちらか一方に Rigidbody2D コンポーネントがアタッチされている
- 判定したいオブジェクトの少なくともどちらか一方の Collider の「Is Trigger」のチェックが付いている

- ・特定のオブジェクト同士のみ当たり判定を行いたい場合 → レイヤーを使う

#### ■ レイヤーとは

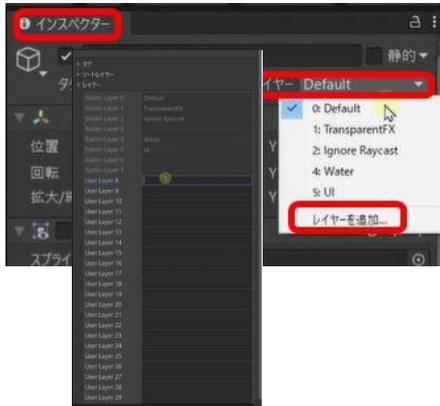
ゲームの中にあるもの（キャラクター、壁、敵など）を「グループ」に分けるためのラベルみたいなものです。

今回は、ステージのレイヤーを作ったので、ステージのグループを作ったイメージです。

■ レイヤーの使い方

① レイヤーの追加

インスペクターウィンドウ → レイヤー → レイヤーを追加 を選択



→ 名前を付ける (8 番目以降から自分で作成できる)

② レイヤーの設定

設定対象のオブジェクトを選択 → インスペクターウィンドウ → レイヤー → 設定したいレイヤーを選択

③ レイヤー同士の衝突判定設定

編集 → プロジェクト設定 → 2D 物理 → Layer Collision Matrix

▼ Matrix (マトリックス) とは

マトリックスは「表」のようなもので、「どのグループがどのグループとぶつかるか」を決めるために使う

例

ぶつかる相手	キャラクター	壁	敵
敵	○	○	×
壁	○	×	○
キャラクター	○	○	×

○…ぶつかる ×…ぶつからない

表をみると、

- ・キャラクターと敵 → ぶつかる
- ・キャラクターと壁 → ぶつかる
- ・・・

というようになります。

要するに、Layer Collision Matrix を設定することで、ぶつかる相手をコントロールします。

▼ 縦と横の項目についての補足

Layer Collision Matrix の縦方向と横方向に並んでいる項目は、どちらもレイヤー名を表している。

- ・縦方向：設定対象のレイヤー（どのグループについて設定を行うか）
- ・横方向：衝突対象のレイヤー（どのグループと衝突するか）

両方ともレイヤー名なので、「縦が基準」や「横が基準」という区別はなく、交差するポイント（セル）ごとに個別の設定が可能。

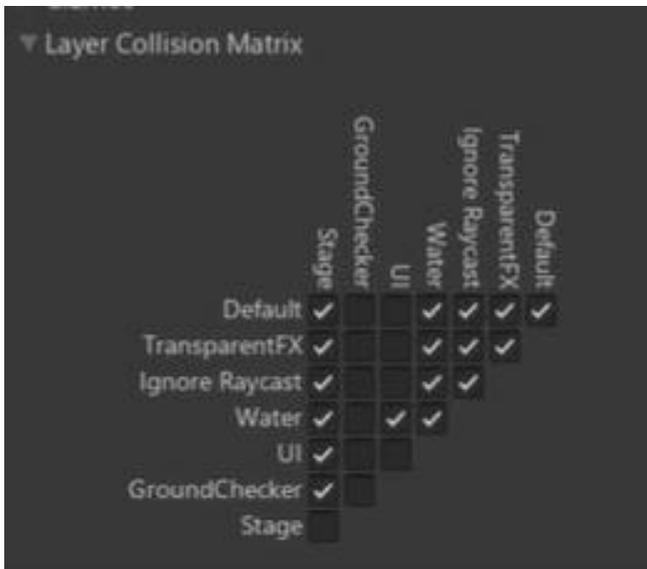
例

縦に「Player」、横に「Wall」がある場合、交差するセルにチェックを入れると「Player が Wall と衝突する」という設定になる

▼基準について

Layer Collision Matrix では、縦横どちらかに特別な基準があるわけではない  
表全体で「どのレイヤーがどのレイヤーと衝突するか」を定義し、交差ポイント（セル）にチェックを入れることで、それぞれの衝突設定が反映される

実際の Layer Collision Matrix で見てみると

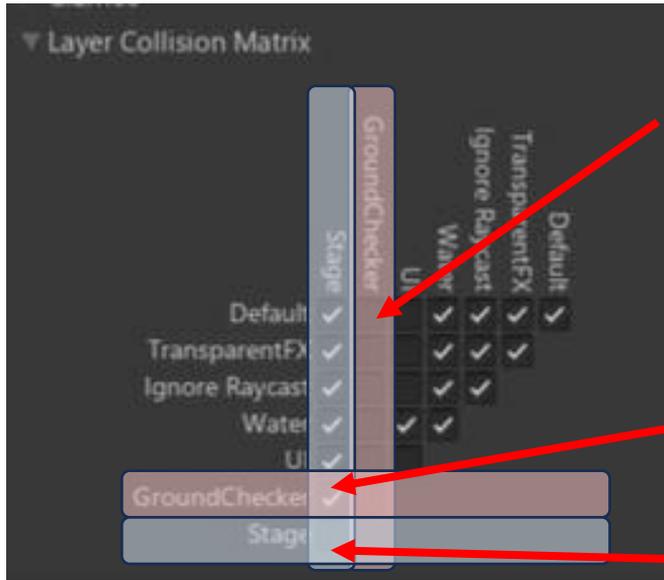


Default 、 TransparentFX、 Ignore Raycast、 Water、 UI は元々用意されているものです。

今回、 Stage と GroundChecker が追加したもののなので、ここの項目をみてください。

Stage と Stage はぶつからないので、チェックを外しています。

GroundChecker は Stage とのみぶつかるので、 Stage と交わるところだけ残して、他はチェックを外しています。



GroundChecker は Stage 以外ぶつからない

GroundChecker は Stage とぶつかる

Stage は Stage とぶつからない

▼まとめ

- レイヤー名の交わる場所にあるチェックボックスは「相互作用するか」を表している
- チェックが入っている場合に当たり判定をする
- 当たり判定をしたくない場合はチェックを外す

- 名前は""（ダブルクォーテーション）で囲む

【用語】

- レイヤー…層
- 引数（ひきすう）…関数を呼び出した時に引き渡される値
- 引数 collision（コリジョン）…当たった相手の情報が入っている
- collision.name（コリジョン.ネーム）…当たった相手の名前

## 【Step4】 02\_トラップを作ろう

このレッスンでは、トラップ（罠）を作って、トラップに当たった時に、コンソールに「ゲームオーバー」と表示させていきます。

ゲームオブジェクトを複製する時に便利な「プレファブ（プレハブ）」や、ゲームオブジェクトの種類を判別する「タグ」等の使い方も学んでいこう。

### 【ポイント】

- 空のオブジェクトを作成してトリガーにする方法（画像が見えている必要のないトリガーの作成）  
ヒエラルキーウィンドウ → +を押す → 空のオブジェクトを作成 → Collider コンポーネントを追加 → トリガーにするにチェックを付ける

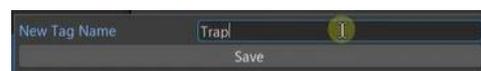
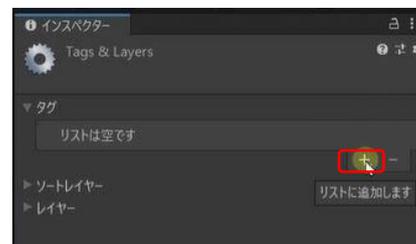
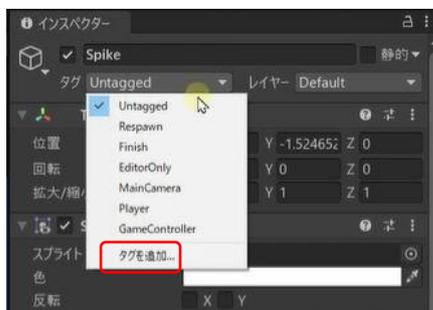
- タグ

タグはゲームオブジェクト1つに1つずつ設定できる

複数のゲームオブジェクトに同一のタグを設定すれば複数のゲームオブジェクトをまとめて処理できる

#### ➤ タグの作り方

インスペクターウィンドウ → タグ → タグを追加 → +を選択 → 名前を付ける → save



※作成しただけでは設定されていない。

#### ➤ タグの設定

設定対象のオブジェクトを選択 → インスペクターウィンドウ → タグ → 設定したいタグを選択

- Destroy()メソッド

( ) の中のものを削除する

「Destroy(this);」とすると、コンポーネントが削除される（this = Destroy を呼び出したコンポーネント）

（現在作成中のゲームでは、PlayerController スクリプトが削除されます）

ゲームオブジェクトを削除したい場合は、「Destroy(this.gameObject);」としましょう

- オブジェクトの複製の方法
  - ▼ヒエラルキーウィンドウの中の増やしたいオブジェクトの上で右クリック→複製を選択  
オブジェクトに変更を加えても、他の複製したオブジェクトは変更されていない。変更するには1つずつ変更しなければならない。
  - ▼オブジェクトをプレファブにして（プレファブ化）、プレファブからオブジェクトを複製する（インスタンス化）  
（簡単に同じ機能のオブジェクトを複製できる）
    - ① プレファブ（プレハブ）化  
ヒエラルキーウィンドウのゲームオブジェクトをプロジェクトウィンドウにドラッグ&ドロップする  
プレファブになったら、名前やアイコンの色が変わる  
元となるプレファブに変更を加えると、プレファブから作られたゲームオブジェクトにも基本的にはその変更が反映される
    - ② インスタンス化…プレファブからゲームオブジェクトを作ること  
プロジェクトウィンドウのプレファブをヒエラルキーかシーンビューにドラッグ&ドロップする
- インスタンスの複製…ヒエラルキーウィンドウのインスタンスを選択して Ctrl+D(複製)
- プレファブから作成したインスタンスを個々で変更した場合、個々での変更が優先される
- 個々のインスタンスの変更は、インスペクターウィンドウのオーバーライドから元に戻すことができる
- 空のオブジェクトを作成してフォルダのように使うこともできる  
ヒエラルキーウィンドウ → +を押す → 空のオブジェクトを作成 → Transform をリセットする  
→ 名前をつける

#### 【用語】

- タグ…分類ラベルみたいなもの
- collision.tag (コリジョン, タグ) …当たった相手のタグ
- Destroy (デストロイ) …破壊する
- Prefab (プレファブ (プレハブ)) …設計図
- インスタンス…実体 (設計図をもとに作られたオブジェクト)

## 【Step4】 03\_ゲームオーバー時のアニメーションを追加しよう

このレッスンでは、トラップに当たった時に、プレイヤーが消えていくアニメーションを追加していきます。

ただ単純にアニメーション切り替えの命令を追加しても、思い通りに動いてくれないので、思い通りの動きをさせるためにはどうすればいいかを1つずつ覚えて実践していこう。

### 【ポイント】

- アニメーターウィンドウの表示…プロジェクトウィンドウのアニメーターコントローラーをダブルクリック
- アニメーションウィンドウの表示…プロジェクトウィンドウのアニメーションをダブルクリック
- bool 型の初期値は False になっている

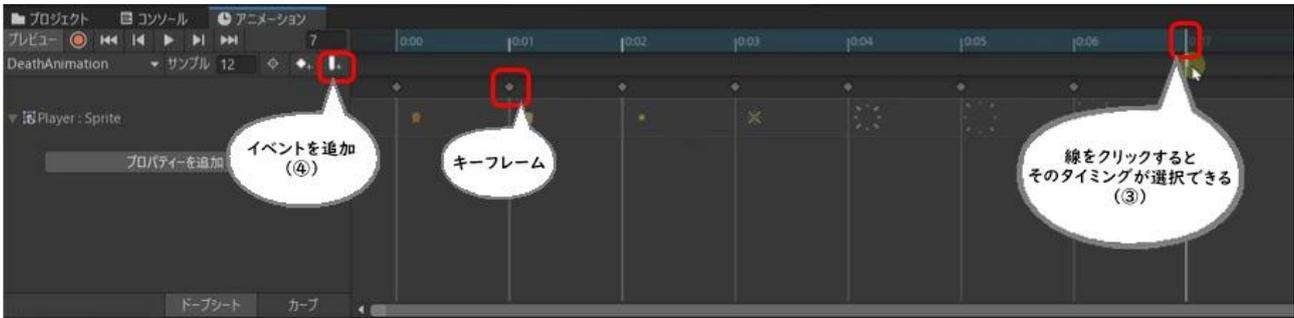
### 《今回のアニメーション切り替え時の問題点》

- 「DeathAnimation」に切り替わらない
  - 一瞬切り替わってはいるが、すぐに他のアニメーションが再生されているため、切り替わっているのがわからない
  - 対策：ゲームオーバーになっているかどうかの判断を入れたらよい
- トラップに当たったあと、プレイヤーが動く
  - 「Rigidbody 2D」コンポーネントが追加されているから  
(速度が当たった時のまま(0ではない)であるためと、プレイヤーと地面の摩擦がないため)
  - 対策：ゲームオーバーになったら、「Rigidbody 2D」を削除する
- 「DeathAnimation」が繰り返し再生されている
  - 対策：アニメーションが終わったタイミングでプレイヤーを削除する

### 《アニメーションについて》

- キーフレームがあるところで画像が切り替わる
- イベント…アニメーションの指定のタイミングでスクリプトの関数(メソッド)を呼び出すことができる
  - イベントの追加方法
    - ① ヒエラルキーウィンドウで対象のオブジェクトを選択
    - ② アニメーションウィンドウでイベントを追加したいアニメーション表示
    - ③ イベントを追加したいタイミングを選択

- ④ 棒の横に+マークのついたボタンでイベントを追加
- ⑤ Function に呼び出す関数名を入力



- private (プライベート)  
何もつけていなければ private になる  
クラスの外からはアクセスできない
- public (パブリック)  
クラスの外からアクセスできる (どこからでもアクセスできる)

#### 【用語】

- Disappearing (ディスアピア) …姿を消す、消滅する
- Function (ファンクション) …関数

#### 【よくあるエラーの対処法】

##### ▼エラー内容

再生すると

「'xxxx' AnimationEvent has no function name specified!」('xxxx'はオブジェクトの名前)  
というエラーが表示され動かない

##### ▼原因

- ・不要なアニメーションイベントを追加していた  
(・アニメーションイベントに関数が設定されていない)

※エラーメッセージの内容は「アニメーションイベントに関数名が設定されていない」という内容)

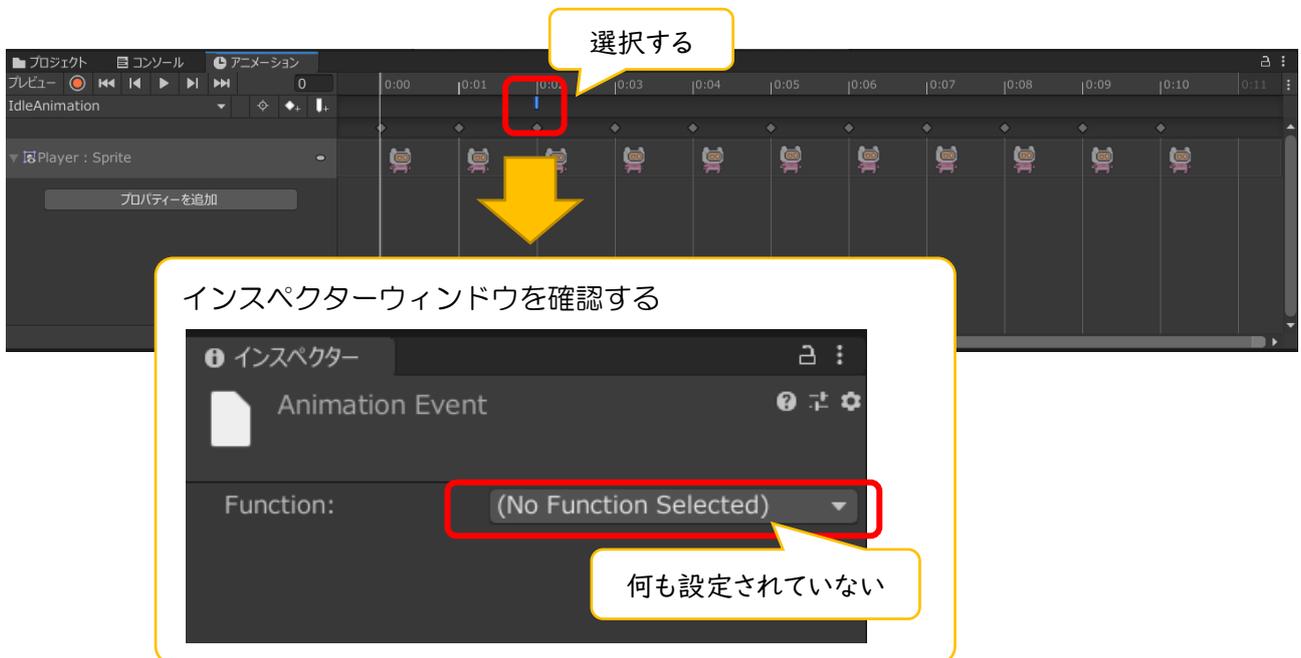
##### ▼対処法

- ・アニメーションイベントを削除  
(・アニメーションイベントに関数を設定する)

不要なイベントを追加してしまっていたということが多いため、確認して削除する方法を説明します。  
アニメーションウィンドウでアニメーションイベントがないか確認します。



アニメーションイベントを選択し、インスペクターウィンドウを確認します。  
関数名が設定されていないと思います。



Delete キーで削除します。(必要なアニメーションイベントの場合は、きちんと関数名を指定します)



## 【Step4】 04\_アイテムを作ろう

このレッスンでは、アイテムを追加していきます。アイテムに触れた時にアイテムを消して、プレイヤーがアイテムを取ったように見せていきます。Step4 もあと少し！頑張ってください。

### 【ポイント】

- アニメーションイベントからは、アクセス修飾子が Private でも呼び出せる
- 関数（メソッド）名を変更した場合、関数を呼び出している部分も変更後の名前に修正する必要がある
- 2D サークルコライダー  
当たり判定を表現するときに使われる Collider（コライダー）の種類の一つ  
コライダーの形状は円  
大きさは半径で調整  
場所はオフセットの X、Y で調整
- コライダーは追加したいゲームのオブジェクトの形に合わせて選択するとよい
- `this.gameObject`… スクリプトがアタッチされているゲームオブジェクト（動画の場合、`PlayerController` がついているプレイヤー）
- `collision.gameObject`… 衝突した相手のゲームオブジェクト
- インスタンス… プレファブ（プレハブ）から作ったゲームオブジェクトのこと

### 【用語】

- Collected（コレクティド）… 集めた
- オフセット… 基準点からの距離で位置を表現したもの