

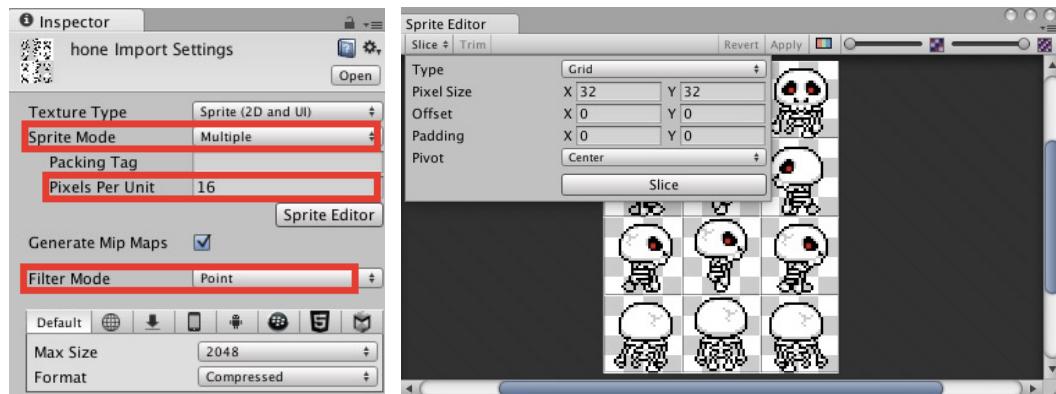
**【目標】敵キャラを作って倒す。**

ここでは、敵をステージ上に出し、倒すところまでを実装します。

**1. 敵キャラを作成する。**

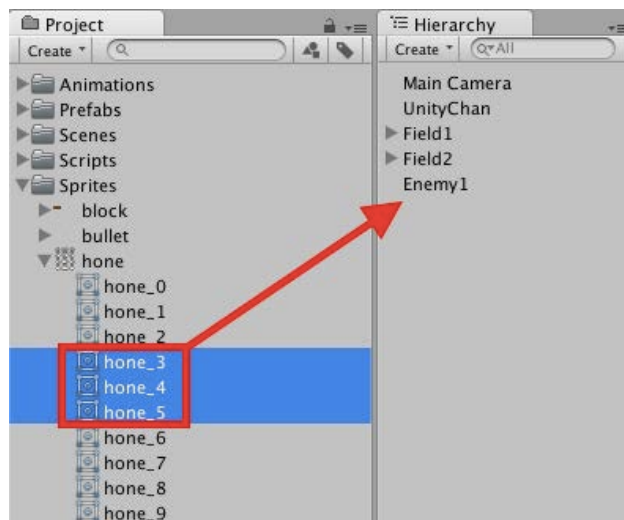
まずは敵キャラを作っていきます。

¥¥mmnas01¥student¥GC2015¥02\_授業¥2017¥Unity¥20170529\_Unity2D\_敵にある、「hone.png」をインポートし、分割しておきます。

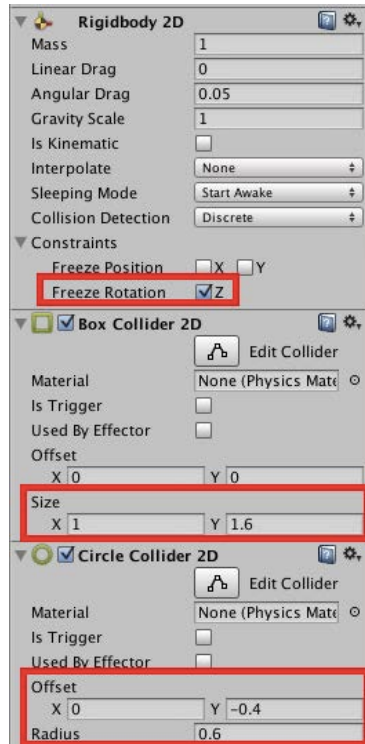


カットが完了したら、hone\_3～5の画像をまとめてドラッグして、Hierarchyビューにドロップし、名前を「Enemy1」に変更しておきます。

また、一緒に作成された Animation は名前を「Walk@Enemy1」に変更し、Animations フォルダに入れておきます。



その後 Unity ちゃんと同様に Enemy1 に Rigidbody2D、BoxCollider2D、CircleCollider2D 加えておきます。



転倒を防ぐために rotation をフリーズし、Unity ちゃん同様に各コライダーのサイズ調整を行います。

次に、敵を Unity ちゃんに向けて動かしてみます。リジッドボディを取得して、その後画面右から左に向けてのベクトルを設定し動かします。

#### Enemy1.cs

```
1  using UnityEngine;
2  using System.Collections;
3
4  public class Enemy1 : MonoBehaviour {
5
6      Rigidbody2D rigidbody2D;
7      public int speed = -3;
8
9      void Start () {
10         rigidbody2D = GetComponent<Rigidbody2D>();
11     }
12
13     void Update () {
14         rigidbody2D.velocity = new Vector2 (speed, rigidbody2D.velocity.y);
15     }
16 }
```

これで敵が、Unity ちゃんに向けて動いてくるはずです。

## 2. 爆発アニメーションを作る。

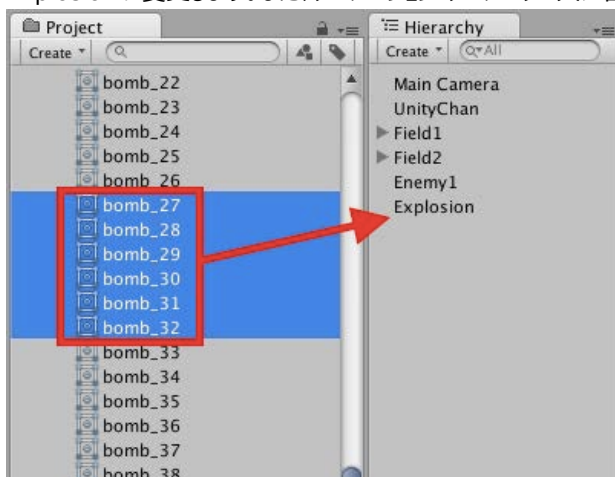
それでは敵を倒す処理を作っていきますよ～その前に、敵を倒した時に発生させる爆発を作成します。

¥¥mmnas01¥student¥GC2015¥02\_授業¥2017¥Unity¥20170529\_Unity2D\_敵

にある、「bomb.png」をインポートし、分割します。今回の画像には、様々なパターンの爆発が含まれています。



カットが完了したら、bomb\_27～bomb\_32 をまとめて Hierarchy ビューへドロップして下さい。ドロップ後は名前を Explosion に変更します。また、アニメーション・アニメーター共に名前を Explotion に変更します。



爆発は 1 回だけなので、Explosion アニメーションの Inspector ビューにある Loop Time のチェックを外しておきます。



次にアニメーションが終了した時に、そのままゲーム画面上に Explosion オブジェクトが残らないようにするために、ExplosionScript を作成します。

Explosion.cs

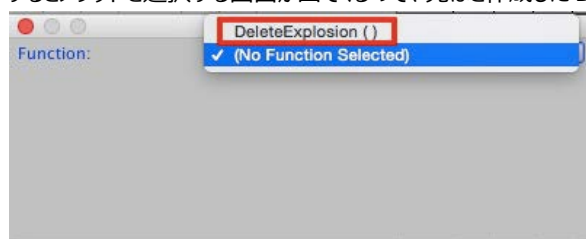
```
1 using UnityEngine;
2 using System.Collections;
3
4 public class ExplosionScript : MonoBehaviour {
5
6     public void DeleteExplosion ()
7     {
8         Destroy(gameObject);
9     }
10 }
```

DeleteExplosion メソッドを呼び出し、オブジェクト自身を削除します。このスクリプトを Explosion オブジェクトの Animation ビューで呼び出す設定をします。

Animator ビューで、ダイヤモンドが並ぶ欄の一段上、その中でもアニメーションの最後で右クリックして下さい。すると、Add Animation Event と出てくるので、クリックします。



するとメソッドを選択する画面が出てくるので、先ほど作成した DeleteExplosion メソッドを選択します



これで爆発アニメーションが終わった時に、Explosion オブジェクトが消えるようになります。  
完成した Explosion オブジェクトは Prefabs 化しておきます。

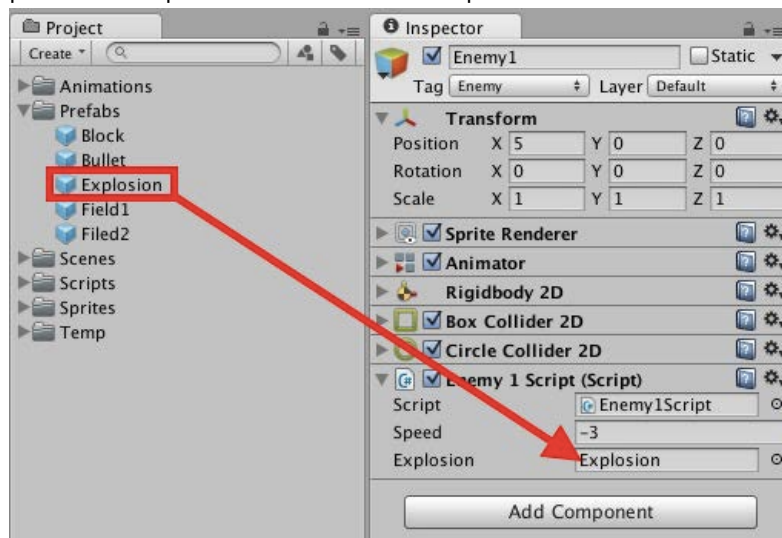
### 3. 敵を倒す。

それでは敵を倒す処理を、Enemy1 に追加します。

Enemy1.cs

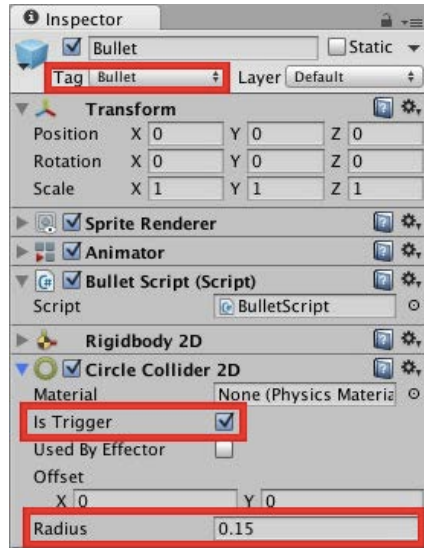
```
1  using UnityEngine;
2  using System.Collections;
3
4  public class Enemy1Script : MonoBehaviour {
5
6      Rigidbody2D rigidbody2D;
7      public int speed = -3;
8      public GameObject explosion;
9
10     void Start () {
11         rigidbody2D = GetComponent<Rigidbody2D>();
12     }
13
14     void Update () {
15         rigidbody2D.velocity = new Vector2 (speed, rigidbody2D.velocity.y);
16     }
17
18     void OnTriggerEnter2D (Collider2D col)
19     {
20         if (col.tag == "Bullet") {
21             Destroy (gameObject);
22             Instantiate (explosion, transform.position, transform.rotation);
23         }
24     }
25 }
```

public 変数 explosion に、先ほど作成した Explosion プレファブを設定します。



OnTriggerEnter2D にて、ぶつかったオブジェクトのタグが Bullet であった場合、自分自身を破壊して爆発アニメーションを呼び出しています。

OnTriggerEnter2D を実行するためには Bullet に Circle Collider2D を付け、Is Trigger を true にする必要があります。また、Bullet という名前のタグを用意して取り付けます。

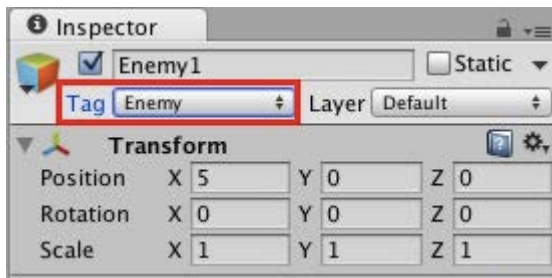


最後に、Bullet オブジェクトに付いている Bullet.cs へも処理を追加します。

Bullet.cs

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class BulletScript: MonoBehaviour {
5
6     private GameObject player;
7     private int speed = 10;
8
9     void Start () {
10         player = GameObject.FindWithTag("UnityChan");
11         Rigidbody2D rigidbody2D = GetComponent<Rigidbody2D>();
12         rigidbody2D.velocity = new Vector2 (speed * player.transform.localScale.x,
13 rigidbody2D.velocity.y);
14         Vector2 temp = transform.localScale;
15         temp.x = player.transform.localScale.x;
16         transform.localScale = temp;
17         Destroy(gameObject, 5);
18     }
19     void OnTriggerEnter2D (Collider2D col)
20     {
21         if (col.gameObject.tag == "Enemy") {
22             Destroy(gameObject);
23         }
24     }
25 }
```

先ほどと同様に、OnTriggerEnter2D を実装し、ぶつかったオブジェクトのタグが Enemy だった時、Bullet を削除しています。有効にするため、Enemy1 オブジェクトに Enemy タグを取り付けて下さい。



これで、弾に当たった敵が爆発するはずですが。