

## Unity ちゃんに弾を撃たせる。

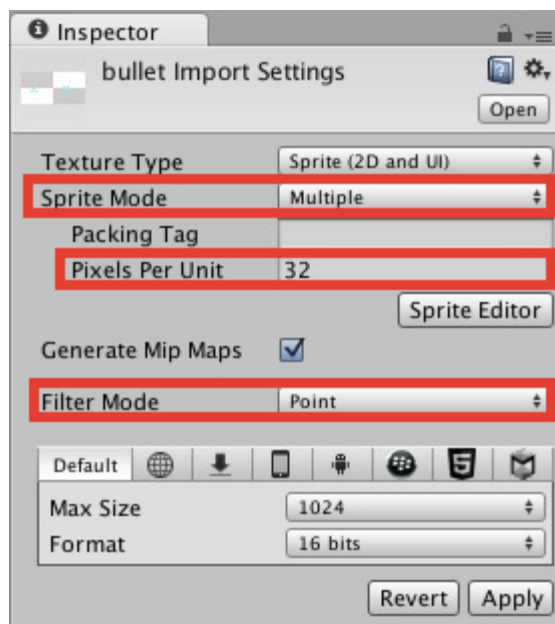
### 1. 弾の作成

¥¥mmnas01¥student¥GC2015¥02\_授業¥2017¥Unity¥20170525\_Unity ちゃんが弾を打つ¥素材に弾の画像“bullet.png”がありますので、Sprite フォルダにインポートしておきます。

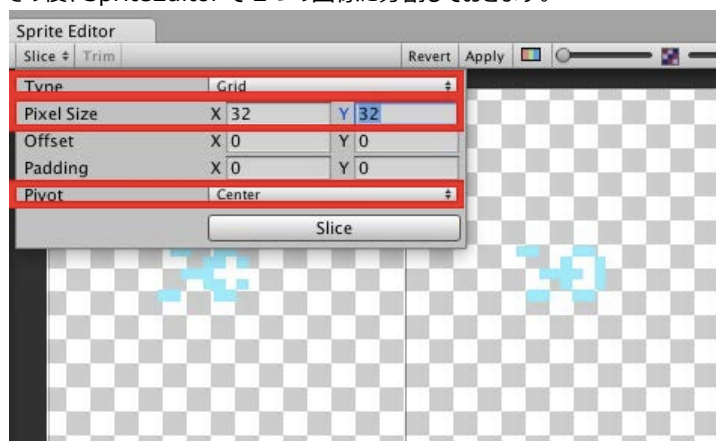


Bullet.png

インポート後、下記のように設定しておきます。

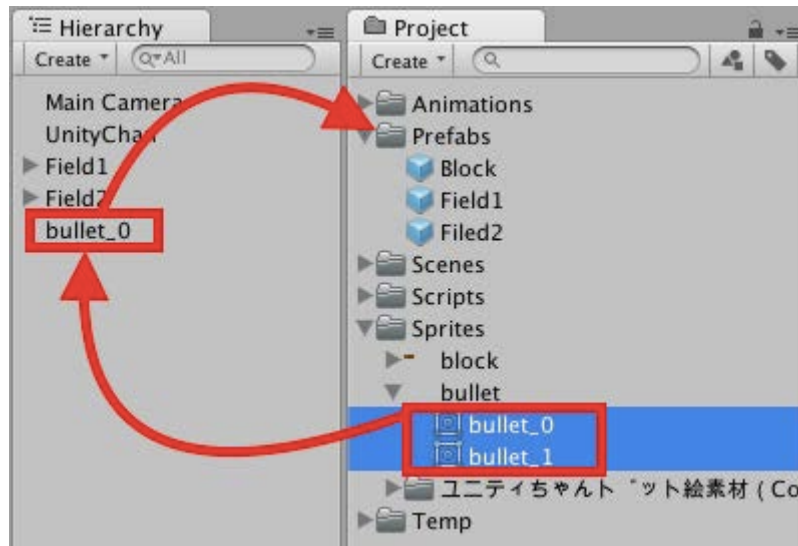


その後、SpriteEditor で 2 つの画像に分割しておきます。



背景のブロック、地面と同じ要領で prefab 化します。手順は下記の通り 2 枚の Sprite 画像に分割出来た後、

1. Hierarchy ビューヘドドラッグ&ドロップ
2. Prefabs フォルダへ移して Prefab 化する。
3. Prefab 化が完了したら、Hierarchy ビューの bullet は削除するとなります。



Prefab 化し、元のオブジェクトは削除 Prefab の名前は「Bullet」に変更し、Rigidbody2D コンポーネントを付け、Prefab 化できたら、下記の BulletScript を、先ほど作成した Bullet に取り付けます

Bullet.cs

```

1  using UnityEngine;
2  using System.Collections;
3
4  public class Bullet: MonoBehaviour {
5
6      private GameObject player;
7      private int speed = 10;
8
9      void Start () {
10         //ユニティちゃんオブジェクトを取得
11         player = GameObject.FindWithTag("UnityChan");
12         //rigidbody2D コンポーネントを取得
13         Rigidbody2D rigidbody2D = GetComponent<Rigidbody2D>();
14         //ユニティちゃんの向いている向きに弾を飛ばす
15         rigidbody2D.velocity =
16             new Vector2 (speed * player.transform.localScale.x, rigidbody2D.velocity.y);
17         //画像の向きをユニティちゃんに合わせる
18         Vector2 temp = transform.localScale;
19         temp.x = player.transform.localScale.x;
20         transform.localScale = temp;
21         //5 秒後に消滅
22         Destroy(gameObject, 5);
23     }
24 }
```

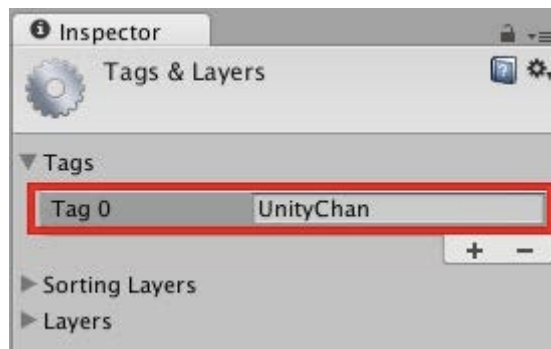
スクリプトの流れとしては、

- ① ユニティちゃんのオブジェクト情報を取得
- ② ユニティちゃんの向いている方向に弾を飛ばす  
ユニティちゃんが右を向いている時は `localScale.x` が 1 に、左を向いている時は `localScale.x` が -1 になっているのを利用します。
- ③ ユニティちゃんが向いている方向に弾の向きを変える
- ④ 5 秒後に弾を消滅(Destroy)させる

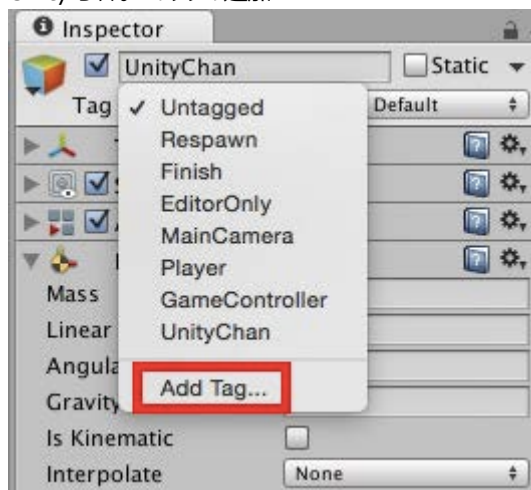
となります。このままだと、bullet が重力で墜ちてしまうため、Bullet についている Rigidbody2D の Is Kinematic にチェックを入れておきます。

その後、File ► Project Settings ► Tags & Layers に「UnityChan」という名前のタグを作成し、UnityChan にタグを追加しておきます。

タグの作成



Unity ちゃんへのタグの追加

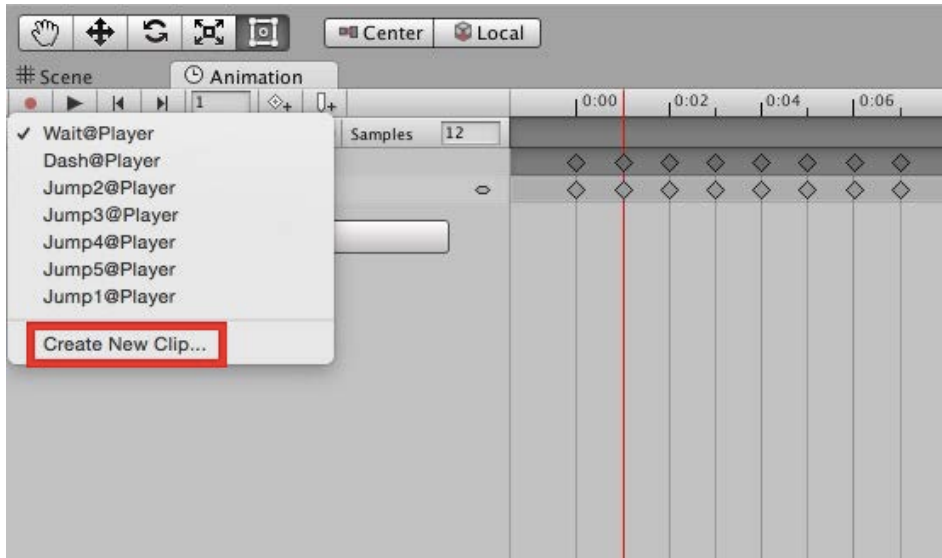


## 2. 弾を撃つアニメーションを設定する

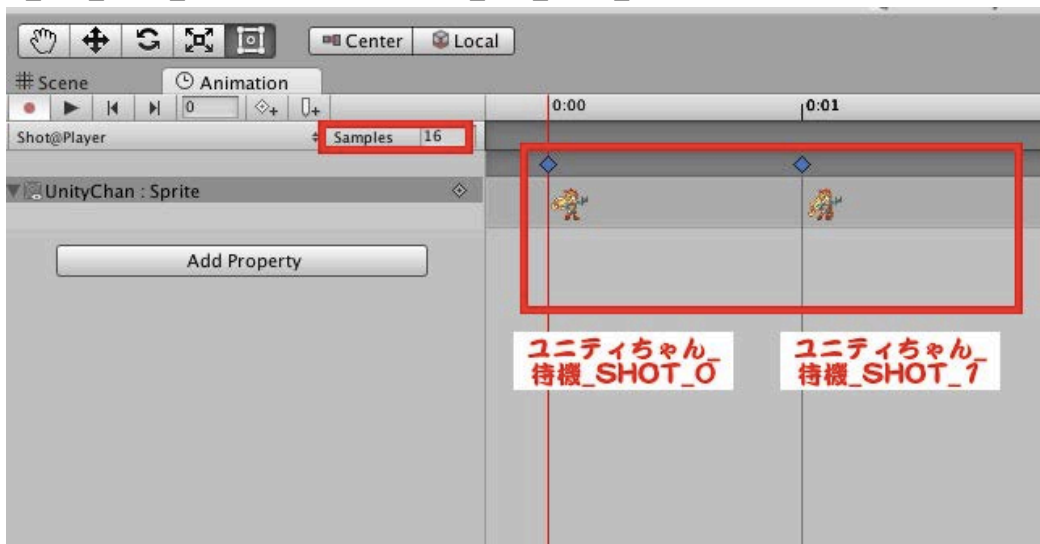
Unity ちゃんに弾を撃つアニメーションを設定します。

Hierarchy ビューで UnityChan を選択した状態で Animation ビューを開き、Create New Clip を実行して下さい。アニメーションの名前は「Shot@Player」です。

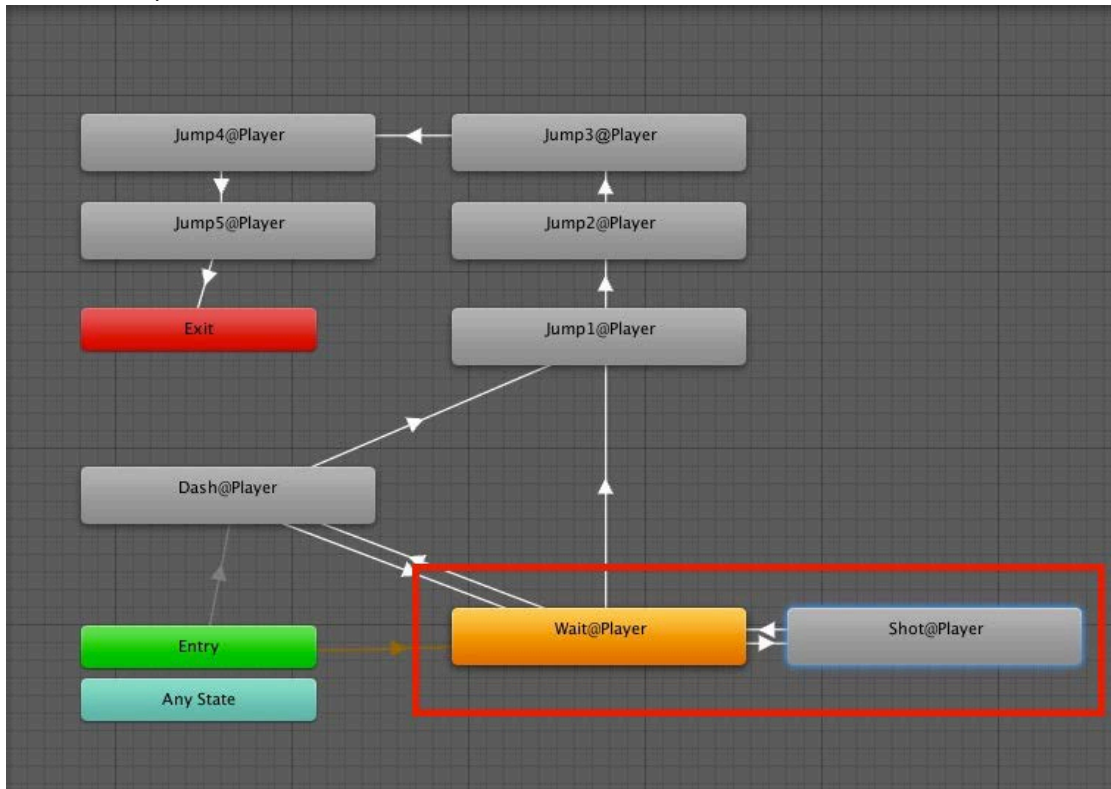
Create New Clip で Shot@Player アニメーションを作成する



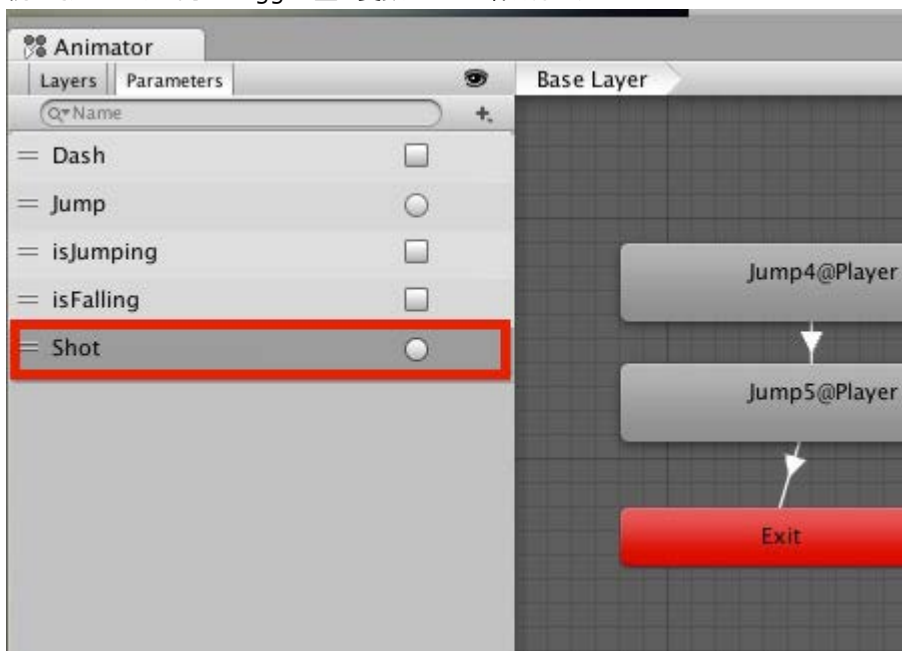
Add Property ► Sprite Renderer ► Sprite を選択し、Samples を 16、アニメーション位置 0.0 にユニティちゃん\_待機\_SHOT\_0、0.1 の位置にユニティちゃん\_待機\_SHOT\_1 を配置します。



続いてアニメーターの設定をしておきます。Animator ビューを開くと Shot@Player のアニメーションが追加されているので、Wait@Player の右に配置し、Make Transition にて相互に矢印で結びます

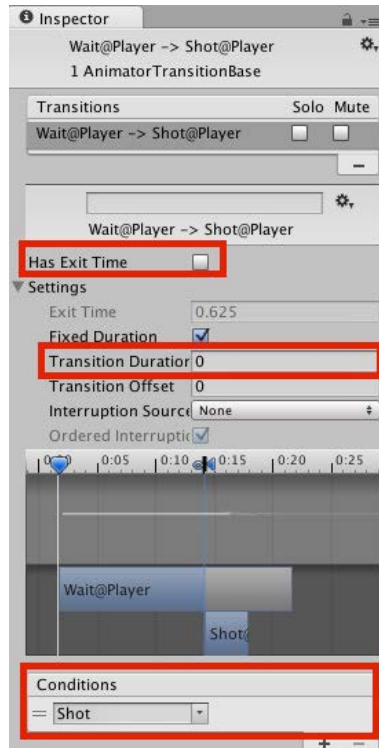


続いてアニメーター内に Trigger 型の変数 Shot を作成します～



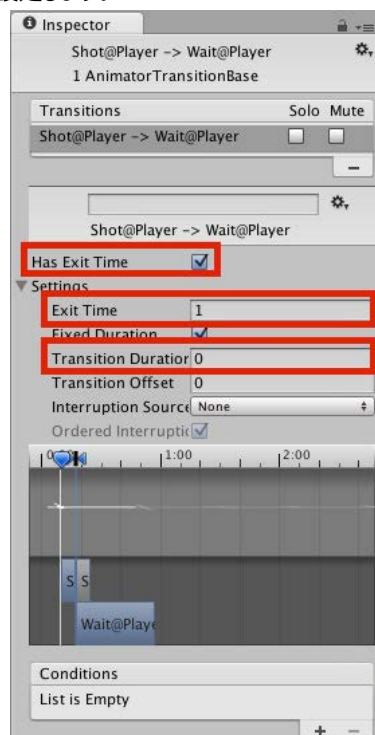
次に Wait@Player から Shot@Player へ向いている矢印をクリックし、Inspector ビューの設定をします。

Trigger 変数 Shot が true になった時に Shot@Player を再生したいので、「Has Exit Time」は false、Transition Duration は 0、Conditions に Trigger 変数 Shot を設定します



Shot@Player から Wait@Player へ向いている矢印の Inspector も設定します。

こちらは、1 回実行して戻ればいいので Has Exit Time は true、Exit Time は 1、Transition Duration は 0 に設定します。



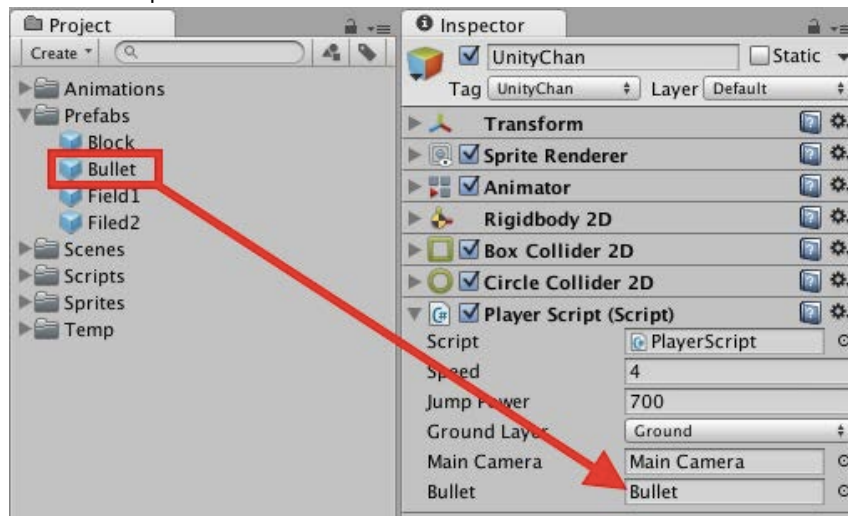
### 3. 実際に弾を撃たせてる

これで弾を撃つ準備ができました。実際に弾を撃たせるようソースに追加します。

Player.cs

```
1      using UnityEngine;
2      using System.Collections;
3
4      public class Player : MonoBehaviour {
5
6          public float speed = 4f;
7          public float jumpPower = 700;
8          public LayerMask groundLayer;
9          public GameObject mainCamera;
10         public GameObject bullet;
11
12         private Rigidbody2D rigidbody2D;
13         private Animator anim;
14         private bool isGrounded;
15
16         void Start () {
17             anim = GetComponent<Animator>();
18             rigidbody2D = GetComponent<Rigidbody2D>();
19         }
20
21         void Update ()
22         {
23             isGrounded = Physics2D.Linecast (
24                 transform.position + transform.up * 1,
25                 transform.position - transform.up * 0.05f,
26                 groundLayer);
27             if (Input.GetKeyDown ("space")) {
28                 if (isGrounded) {
29                     anim.SetTrigger("Jump");
30                     isGrounded = false;
31                     rigidbody2D.AddForce (Vector2.up * jumpPower);
32                 }
33             }
34             float velY = rigidbody2D.velocity.y;
35             bool isJumping = velY > 0.1f ? true:false;
36             bool isFalling = velY < -0.1f ? true:false;
37             anim.SetBool("isJumping",isJumping);
38             anim.SetBool("isFalling",isFalling);
39             if (Input.GetKeyDown ("left ctrl")){
40                 anim.SetTrigger("Shot");
41                 Instantiate(bullet, transform.position +
42                     new Vector3(0f,1.2f,0f), transform.rotation);
43             }
44         } // 以下省略
```

ソース入力後、public 変数の bullet に Prefab の Bullet を設定します。



左の control キーを押した時にアニメーターの Trigger 変数 Shot を true にし、Instantiate にて Bullet を生成しています～生成位置はユニティちゃんの足元から Y 軸方向に 1.2 上に調整しています。

これで弾が撃てるはずです。

