【目標】uGUI を使った HUD の追加。

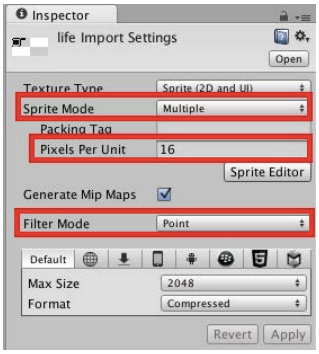
ここでは、プレイヤーのライフゲージを作成し、的からのダメージを視覚化します。

1. uGUI 用画像の準備

まずは敵キャラを作っていきます。

¥¥mmnas01¥student¥GC2015¥02_授業¥2017¥Unity¥20170602_Unity2D_HUD

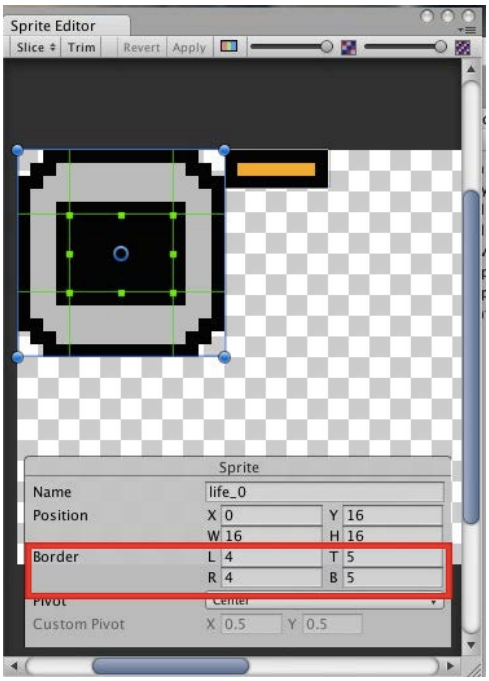
にある、「HUD.png」をインポートします。



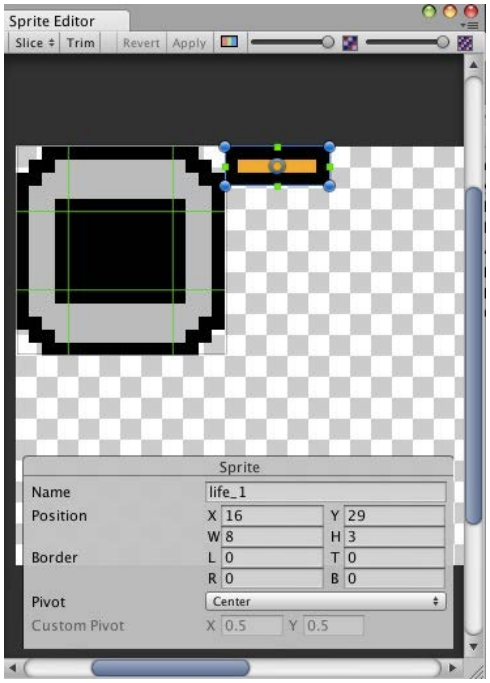
次に、Sprite Editor の設定です。今回は uGUI の授業で使った手法で、Image Type を Tiled にして、画像を引き伸ばした時に四隅は引き伸ばされないようにします。

まずは左側の背景画像から見ていきます。Position は以外に、今回は Border の設定も行います。

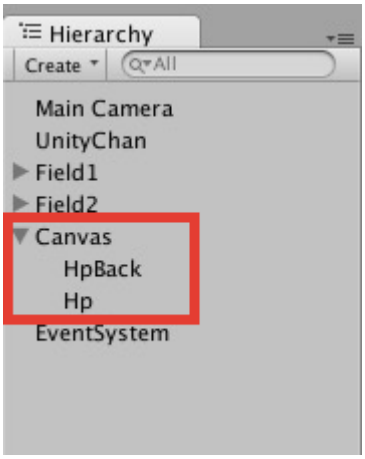
左側画像の設定



右側画像の設定

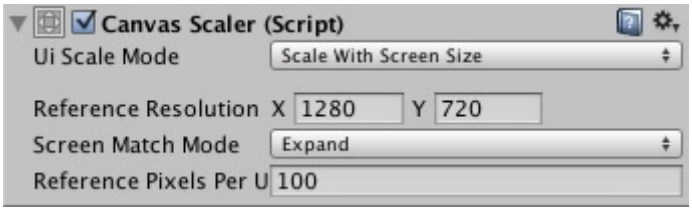


続いて uGUI の Image を作成します。Hierarchy ビューから　Create>UI>Image　を背景画像と目盛り用にふたつ選択します。それぞれ「HpBack」、「Hp」という名前にします。



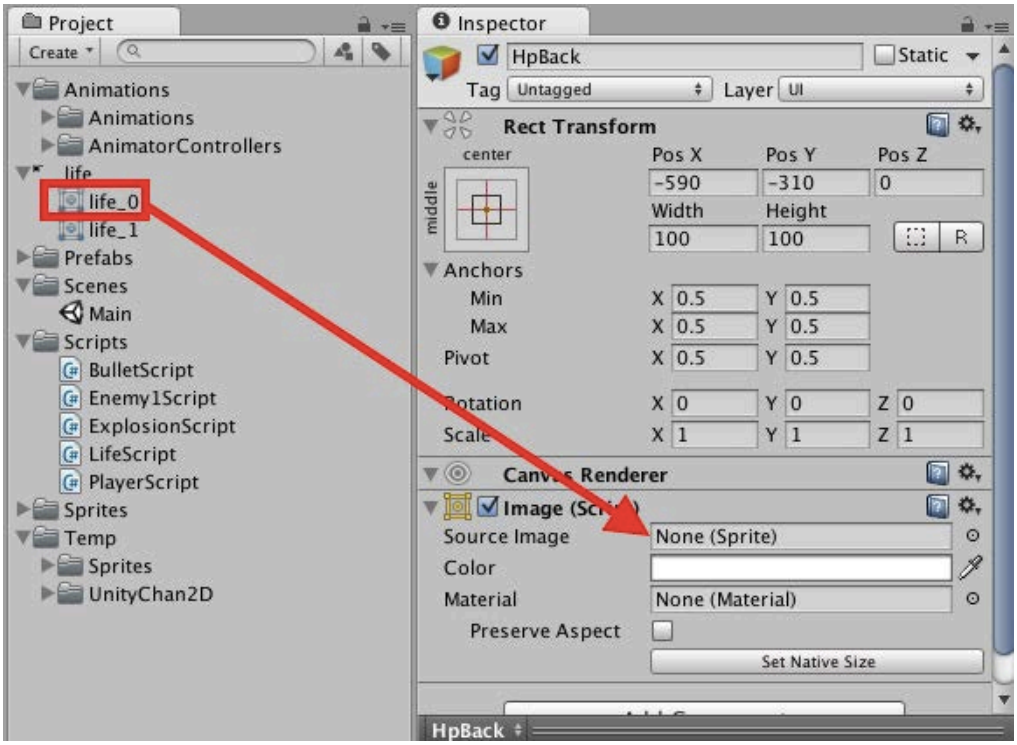
並び順が下にある程前面に表示されるので、HpBack を上、Hp を下に配置します。
続いて Canvas を設定します。Canvas の Inspector ビューにある Canvas Scaler にて、

・UI Scale Mode を Scale With Screen Size、
・Reference Resolution を X: 1280、Y:720、
・Screen Match Mode を Expand

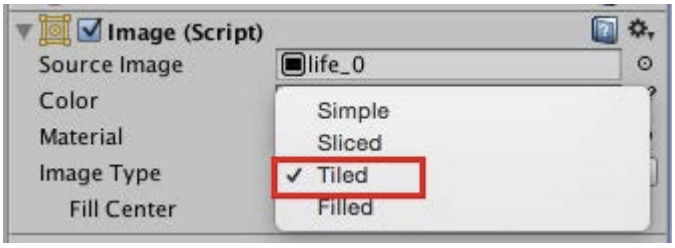


にします。

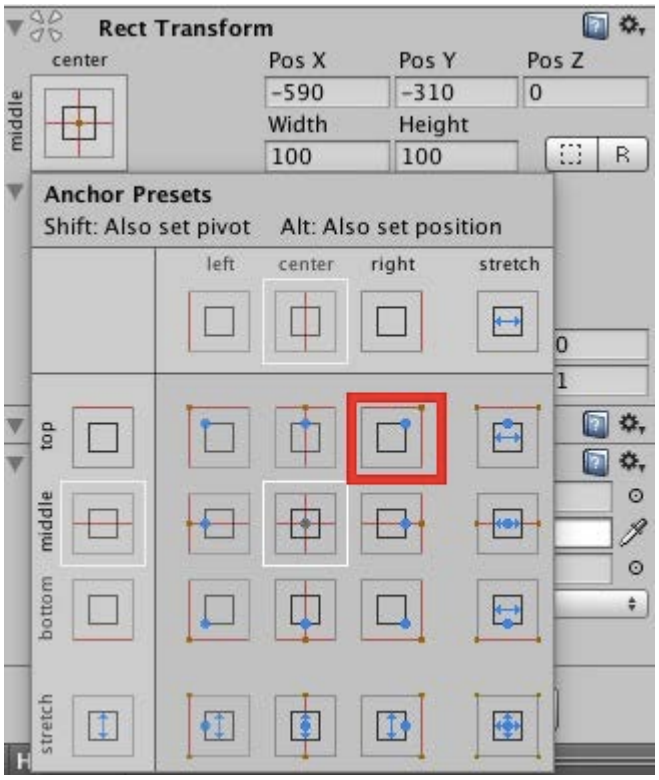
続いて HpBack の設定をしていきます。Inspector ビューの Image コンポーネントの Source Image に、先ほど作成した life_0 画像を指定します。



続いて Image Type を Tiled に変更します。先ほど言った通り Tiled にすると、四隅を固定したまま画像を引き伸ばすことができます。

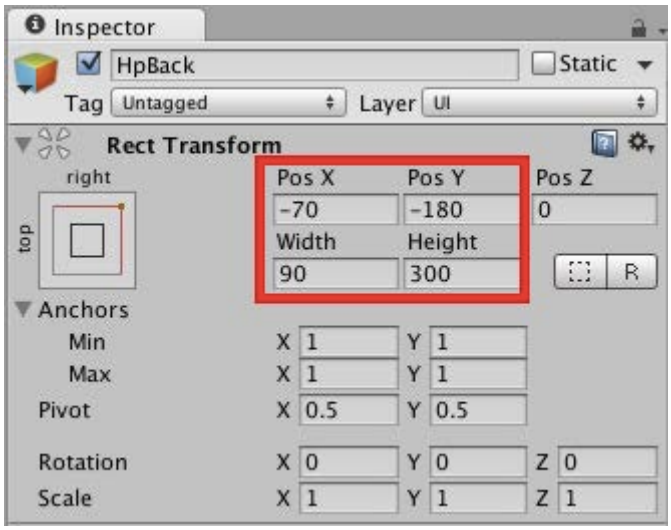


続いて Rect Transform を設定して、UI の位置や大きさを変えていきます。
まずは Anchor Preset を変更し、Anchor を右上に設定します



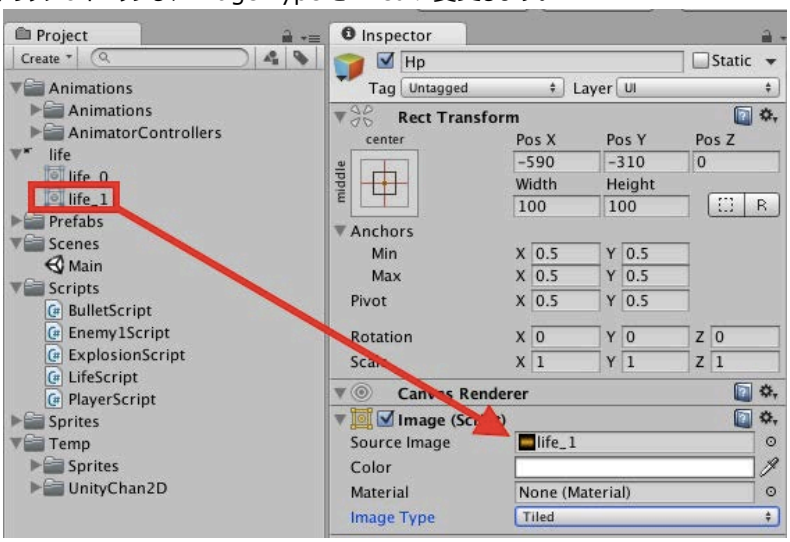
alt を押しながら赤枠をクリックして右上を Anchor に設定します。

位置と大きさは以下のように設定します

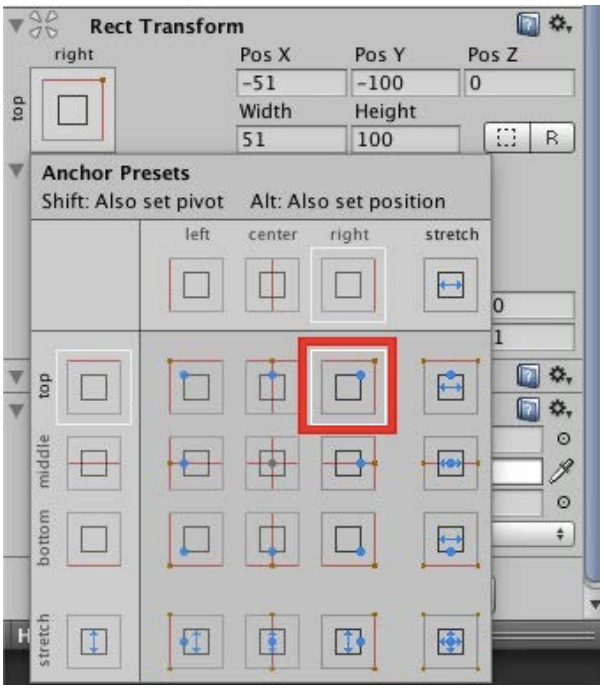


そうすると下記の通りになります。

続いて Hp の設定を行います。先ほどと同じようにカットした life_1 画像を Image コンポーネントの Source Image にドラッグ＆ドロップし、Image Type を Tiled に変更します。

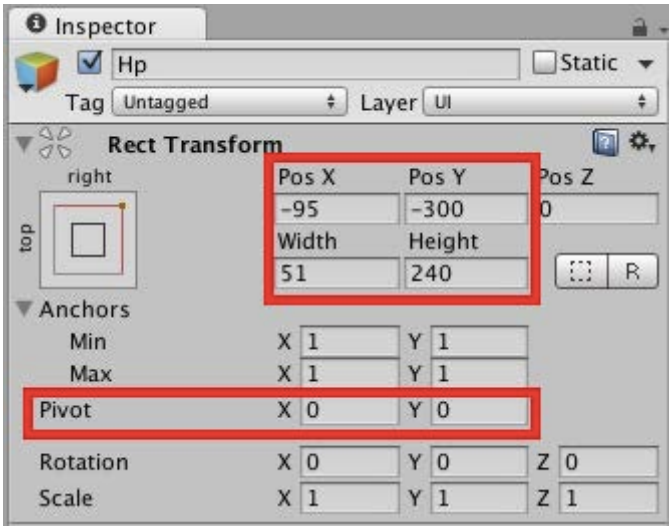


Anchor Presets を右上にします



alt を押しながら赤枠をクリック

Rect Transform は以下のように設定します。



Pivot を X:0, Y:0 に変更するのを忘れないで下さい。これは下図のように、目盛りの Height を 240 から 140 に減らした時側が基点(Achor)となるため、画像の上部から短くなってくれるためです。

2. ライフポイントを変化させる。

ここまでできると、後はライフの目盛り画像の Width を減らしていけば、ダメージを表現できるようになります。それでは LifeScript を作り、Hp オブジェクトに取り付けましょう

Life.cs

```
1   using UnityEngine;
2   using System.Collections;
3
4   public class Life : MonoBehaviour {
5
6       RectTransform rt;
7
8       void Start ()
9       {
10          rt = GetComponent<RectTransform>();
11      }
12
13      public void LifeDown (int ap){
14          //RectTransform のサイズを取得し、マイナスする
15          rt.sizeDelta -= new Vector2 (0,ap);
16      }
17  }
```

Hp オブジェクトの Rect Transform コンポーネントの Width と Height は sizeDelta で取得できるので、Height から引数の ap(AttackPoint の略)分だけマイナスします。敵とぶつかった時に、この LifeDown メソッドを呼び出せばライフゲージが下がります。

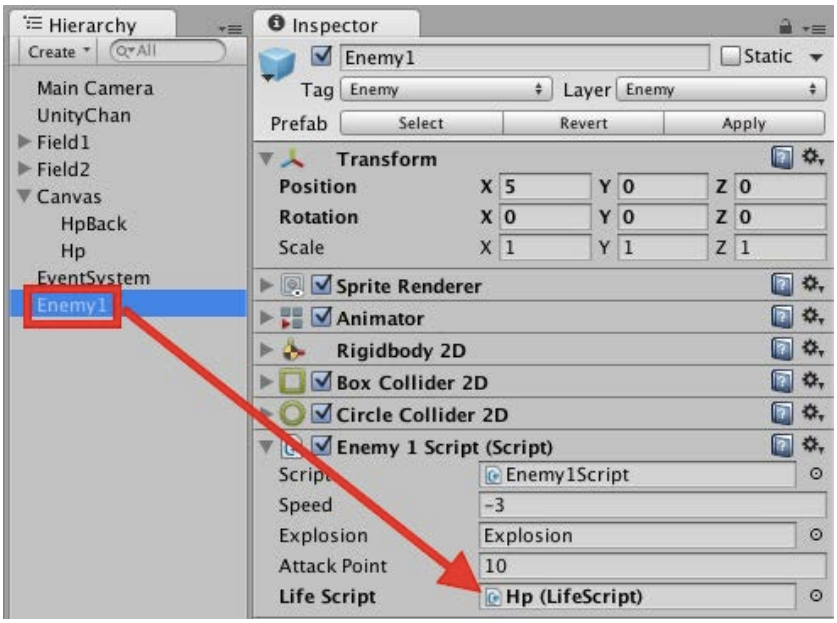
次に、Enemy1Script に加筆していきます。

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       public GameObject explosion;
9       public int attackPoint = 10;
10      public Life lifeScript;
11
12      void Start () {
13          rigidbody2D = GetComponent<Rigidbody2D>();
14      }
15
16      void Update () {
17          rigidbody2D.velocity = new Vector2 (speed, rigidbody2D.velocity.y);
18      }
```

```
19
20          void OnTriggerEnter2D (Collider2D col)
21          {
22              if (col.tag == "Bullet") {
23                  Destroy (gameObject);
24                  Instantiate (explosion, transform.position, transform.rotation);
25              }
26          }
27          void OnCollisionEnter2D (Collision2D col)
28          {
29              //UnityChan とぶつかった時
30              if (col.gameObject.tag == "UnityChan") {
31                  //Life の LifeDown メソッドを実行
32                  life.LifeDown(attackPoint);
33              }
34          }
35      }
```

public 変数である lifeScript があるので、Inspector ビューから LifeScript を持っている Hp オブジェクトをドラッグ＆ドロップしておきます。



敵がユニティちゃん(UnityChan タグ付き)とぶつかった時、LifeScript の LifeDown メソッドを呼び出しています。

引数は AttackPoint で 10 にしていますが、public 変数にしているので Inspector ビューから自由に変更できます。

結果として下記の通りになります。

ライフゲージが減ります。