

【目標】タイトル、ゲームオーバー、リザルトのシーン切り替えⅡ。

タイトル画面からステージ紹介画面、ステージ画面、ゲームオーバー/次ステージへのシーン切り替えの実装を行います。

ここでの重要キーワード： **Invoke, OnCollisionEnter2D, OnTriggerEnter2D, SetBool**

1. ゲームオーバー機能の実装：穴に落ちる。

続いて、前回の宿題だった「穴に落ちたときの処理」を実装します。ゲームオーバーの条件として

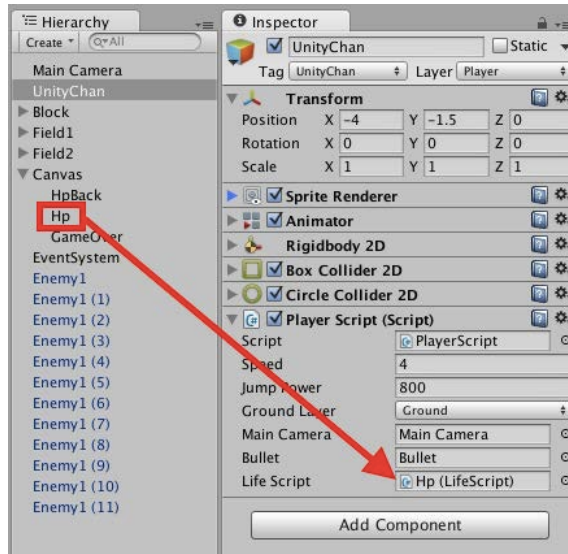
“穴に落ちる” = “プレイヤーの位置が規定の位置よりも下になる”

ことで、判断できます。そこで、メインカメラの位置よりも “8” 低くなった時点で、ゲームオーバーとします。Player スクリプトで、HP オブジェクトを取得し、Update にて現在のカメラの高さ(y)を比較し、HP オブジェクトに設定されているライフ管理クラス(Life)の GameOver()メソッドを呼び出せばよいのです。

Player.cs

```
1      using UnityEngine;
2      using UnityEngine.SceneManagement;
3      using System.Collections;
4
5      public class PlayerScript : MonoBehaviour {
6
7          public float speed = 4f;
8          public float jumpPower = 700;
9          public LayerMask groundLayer;
10         public GameObject mainCamera;
11         public GameObject bullet;
12         public Life life;
13         : // 途中省略
14
15
16
17
18
19
20
21         void Update ()
22         {
23             : // 途中省略
24
25             if (Input.GetKeyDown ("left ctrl")) {
26                 anim.SetTrigger ("Shot");
27                 Instantiate (bullet, transform.position + new Vector3 (0f, 1.2f, 0f)
28                     , Quaternion.identity);
29             }
30
31             //現在のカメラの位置から 8 低くした位置を下回った時
32             if (gameObject.transform.position.y < Camera.main.transform.position.y - 8) {
33                 //Life の GameOver メソッドを実行する
34                 life.GameOver();
35             }
36         }
37
38         :
39         : // 途中省略
40
41     }
```

public 変数 life を作成したので、いつも通り指定しおきます。



2. 画面外に敵が出たときの処理。

敵が画面外に出た時に消去する処理を実装します。Enemy1 スクリプトに加筆していきます。

Player.cs

```

1  using UnityEngine;
2  using UnityEngine.SceneManagement;
3  using System.Collections;
4
5  public class Enemy1 : MonoBehaviour {
6
7      Rigidbody2D rigidbody2D;
8      public int speed = -3;
9
10     :
11     : // 途中省略
12     :
13
14     void Update ()
15     {
16         if (_isRendered) {
17             rigidbody2D.velocity = new Vector2 (speed, rigidbody2D.velocity.y);
18         }
19         if (gameObject.transform.position.y < Camera.main.transform.position.y - 8 ||
20             gameObject.transform.position.x < Camera.main.transform.position.x - 10) {
21             Destroy(gameObject);
22         }
23     }
24
25     : // 途中省略
26 }

```

先ほど Player スクリプトで実装した処理の応用です。ユニティちゃんは画面より左に行くことはないですが、敵キャラは左に移動してしまうので、そうなった時に Destroy します。

3. ゲーム画面にゲームクリア機能を付ける。

それでは最後にゲームクリアの処理を追加します。

まずはゲームクリアテキストを “GameClear” という名前で作成しておきます。ゲームオーバーテキストと同様に、Text コンポーネントは非アクティブ化しておきます。

作成例

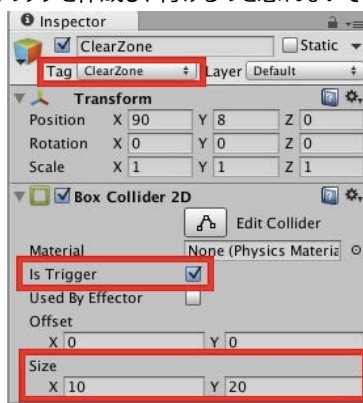


続いて、ゲームクリア・オブジェクト、ClearZone オブジェクトを作成します。

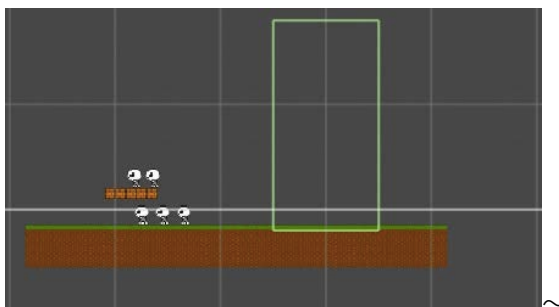
Hierarchy ビューから Create > Create Empty を選択し、名前を ClearZone としておきます。



プレイヤーとのあたり判定を行いますので、BoxCollider2D を付けて、以下のように設定します。ClearZone という名前のタグを作成し、付けるのを忘れないで下さい。



位置に関しては、図のようにステージの最後、クリアになる場所に配置します。



Player スクリプトを変更します。

ゲームクリアを管理する gameClear フラグを作成し、クリアした際に gameClear フラグを true に切り替えることとします。

ゲームクリア後は、ジャンプや弾の発射、カメラの Unity ちゃん追尾など禁止して、Unity ちゃんが画面右方向に進み画面から消える演出を入れることにします。

そのため、gameClear フラグが true になった時、ジャンプ・ショット・移動と全ての操作を無効にし、MainCamera の Unity ちゃん追尾動作を停止し。ユニティちゃんが自動で右に進み画面から出るように、各処理にフラグのチェックを追加します。

また Unity ちゃんが画面外に出た後、5 秒後にタイトルに遷移させます。Invoke メソッドは、指定した秒数後に指定したメソッドを実行できますので、今回 5 秒後にタイトル画面を呼び出す CallTitle メソッドを呼び出しています。

Player.cs

```
1 using UnityEngine;
2 using System.Collections;
3 using UnityEngine.SceneManagement;
4 using UnityEngine.UI;
5
6 public class Player : MonoBehaviour {
7
8     public float speed = 4f;
9     public float jumpPower = 700;
10    public LayerMask groundLayer;
11    public GameObject mainCamera;
12    public GameObject bullet;
13    public Life life;
14
15    private Rigidbody2D rigidbody2D;
16    private Animator anim;
17    private bool isGrounded;
18    private Renderer renderer;
19
20    private bool gameClear = false; //ゲームクリアしたら操作を無効にする
21    public Text clearText; //ゲームクリア時に表示するテキスト
22
23    void Start () {
24        anim = GetComponent<Animator>();
25        rigidbody2D = GetComponent<Rigidbody2D>();
26        renderer = GetComponent<Renderer>();
27    }
28
29    void Update ()
30    {
31        isGrounded = Physics2D.Linecast (
32            transform.position + transform.up * 1,
33            transform.position - transform.up * 0.05f,
34            groundLayer);
35    }
```

```

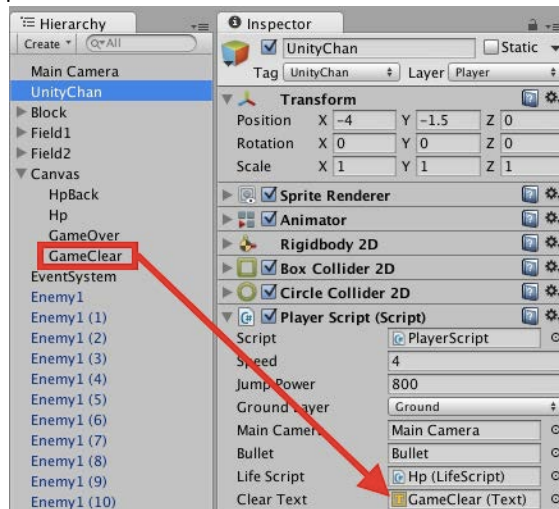
36         if ( ) { // ←追加部分 ゲームクリア以降ジャンプ禁止の if : 分岐条件を
37             if (Input.GetKeyDown ("space")) {
38                 if (isGrounded) {
39                     anim.SetTrigger ("Jump");
40                     isGrounded = false;
41                     rigidbody2D.AddForce (Vector2.up * jumpPower);
42                 }
43             }
44         }
45
46         float vely = rigidbody2D.velocity.y;
47         bool isJumping = vely > 0.1f ? true : false;
48         bool isFalling = vely < -0.1f ? true : false;
49         anim.SetBool ("isJumping", isJumping);
50         anim.SetBool ("isFalling", isFalling);
51
52         //ゲームクリア以降弾を打たせない、カメラの位置から “8” 下がるとゲームオーバー
53         if ( ) { // 追加 if : 分岐条件を考えてください。
54             if (Input.GetKeyDown ("left ctrl")) {
55                 anim.SetTrigger ("Shot");
56                 Instantiate (bullet, transform.position + new Vector3 (0f, 1.2f, 0f)
57                     , Quaternion.identity);
58             }
59
60             if (gameObject.transform.position.y < Camera.main.transform.position.y - 8)
61             {
62                 life.GameOver ();
63             }
64         }
65     }
66
67     void FixedUpdate ()
68     {
69         //左右に移動させない MainCamera を動かさない
70         if ( ) { // ← 追加 if : 分岐条件を考えて下さい。
71             // ゲームクリア前処理
72             float x = Input.GetAxisRaw ("Horizontal");
73             if (x != 0) {
74                 rigidbody2D.velocity = new Vector2 (x * speed, rigidbody2D.velocity.y);
75                 Vector2 temp = transform.localScale;
76                 temp.x = x;
77                 transform.localScale = temp;
78                 anim.SetBool ("Dash", true);
79                 if (transform.position.x > mainCamera.transform.position.x - 4) {
80                     Vector3 cameraPos = mainCamera.transform.position;
81                     cameraPos.x = transform.position.x + 4;
82                     mainCamera.transform.position = cameraPos;

```

```
83     }
84     Vector2 min = Camera.main.ViewportToWorldPoint (new Vector2 (0, 0));
85     Vector2 max = Camera.main.ViewportToWorldPoint (new Vector2 (1, 1));
86     Vector2 pos = transform.position;
87     pos.x = Mathf.Clamp (pos.x, min.x + 0.5f, max.x);
88     transform.position = pos;
89     } else {
90         rigidbody2D.velocity = new Vector2 (0, rigidbody2D.velocity.y);
91         anim.SetBool ("Dash", false);
92     }
93 } else {
94     //ゲームクリア以降の処理
95     //クリアテキストを表示
96     clearText.enabled = true;
97     //アニメーションは走り
98     anim.SetBool ("Dash", true);
99     //右に進み続ける
100    rigidbody2D.velocity = new Vector2 (speed, rigidbody2D.velocity.y);
101    //5 秒後にタイトル画面へ戻る CallTitle メソッドを呼び出す
102    Invoke("CallTitle", 5);
103 }
104 }
105
106 void OnCollisionEnter2D (Collision2D col)
107 {
108     if (!gameClear) {
109         if (col.gameObject.tag == "Enemy") {
110             StartCoroutine ("Damage");
111         }
112     }
113 }
114
115 IEnumerator Damage ()
116 {
117     gameObject.layer = LayerMask.NameToLayer("PlayerDamage");
118     int count = 10;
119     while (count > 0){
120         renderer.material.color = new Color (1,1,1,0);
121         yield return new WaitForSeconds(0.05f);
122         renderer.material.color = new Color (1,1,1,1);
123         yield return new WaitForSeconds(0.05f);
124         count--;
125     }
126     gameObject.layer = LayerMask.NameToLayer("Player");
127 }
128
129
```

```
130
131
132 // クリアオブジェクトとの衝突処理
133 void OnTriggerEnter2D (Collider2D col)
134 {
135     // タグが ClearZone である Trigger にぶつかったら
136     if (col.tag == "ClearZone") {
137         // ゲームクリアフラグ処理
138         gameClear = true;
139     }
140 }
141
142
143 void CallTitle ()
144 {
145     // タイトル画面へ
146     SceneManager.LoadScene("Title");
147 }
148 }
```

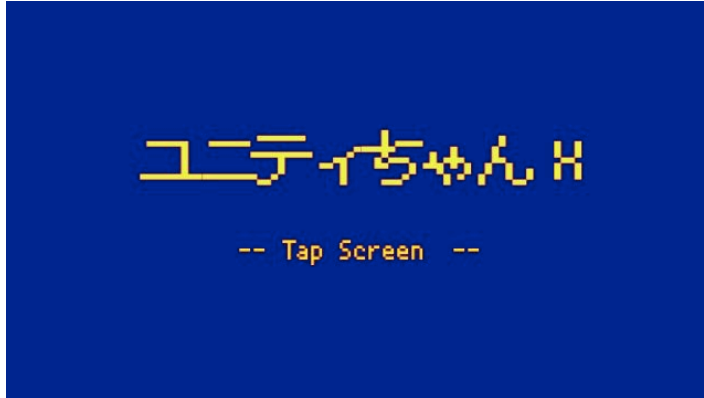
public 変数である ClearText を指定しておきます。



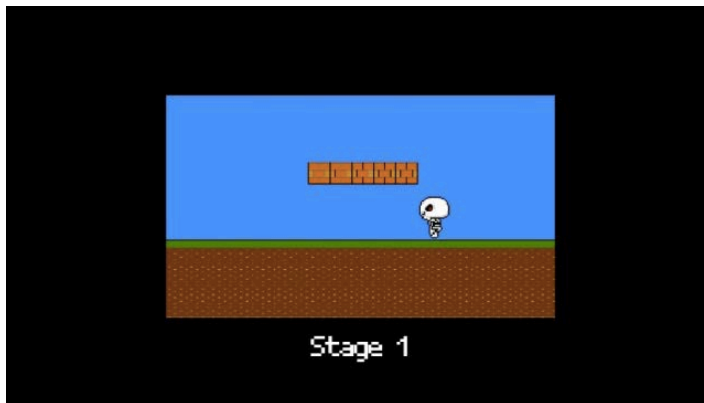
4. 結果

これで、ゲーム全体が完成しました。最終的に下記の通りの動作になるはずです。

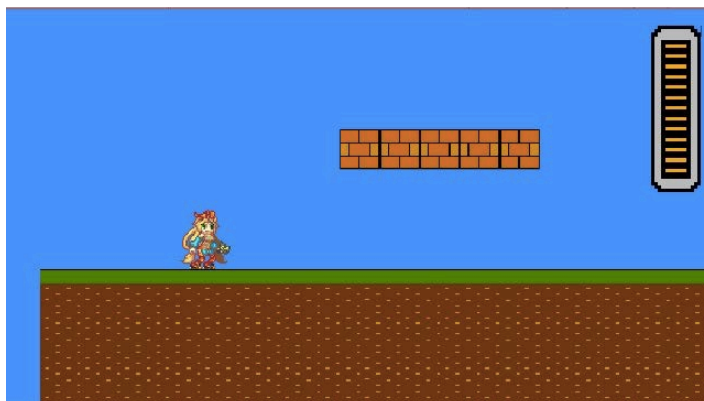
1) Title 画面



2) ステージ紹介表示



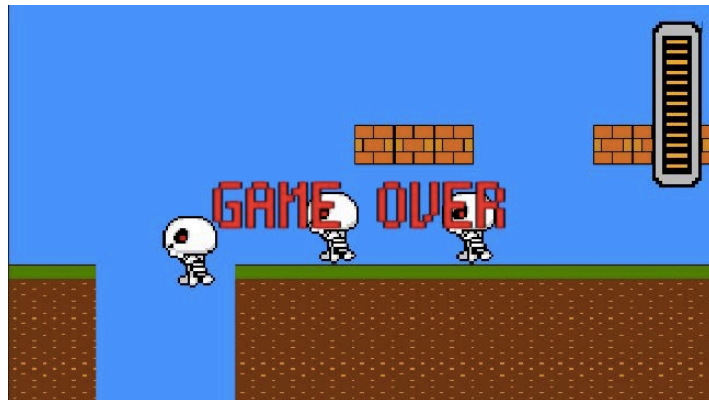
3) ゲーム開始



(ア) ライフが“0”でゲームオーバー



(イ) 穴に落ちてゲームオーバー



4) クリア地点まで進むとゲームクリア、Unity ちゃんが画面右へ



5) ゲームオーバー後、クリックか、クリア 5 秒でタイトルへ

