Unity 講習資料 3

1．UI

ゲームらしくするため、アイテムがあと何個あるかを表示するようにしましょう。

① GameController を以下のように加筆してください。

```
void Update()
{

}

2 references
public int Count;
0 references
void OnGUI()
{
    Count = GameObject.FindGameObjectsWithTag("Item").Length;
    GUI.TextField(new Rect(10,10,300,80),"残り" + Count.ToString() + "個");
}
```

void OnGUI(){}：内部で GUI に関する命令を扱える、という意味。GUI とは UI の一種で、まあ文字やボタンと考えてください。詳しいことは調べてください。

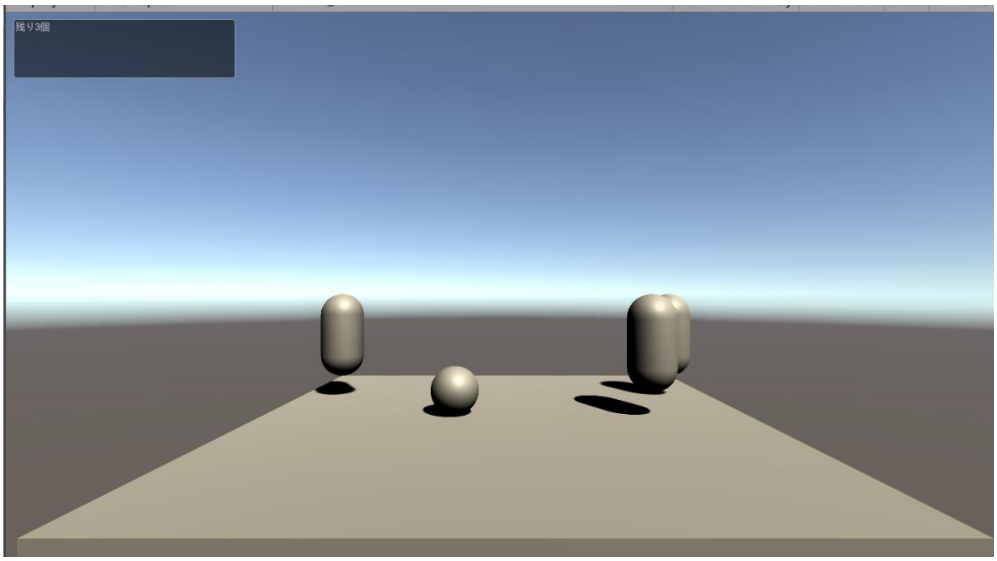
GameObject.FindGameObjectsWithTag(“Item”)：Item というタグが付いているオブジェクトを配列で取得する命令。この文は Item というタグのついたオブジェクトがすべて入った配列として扱えます。.Length は配列の長さを表すので、変数 Count には Item というタグがついたオブジェクトの数が入ります。

GUI.TextField(new Rect(x,y,width,height),”text”);：画面に文字を表示する命令。Rect は長方形という意味で、点(x,y)を起点に横 width、縦 height の長方形範囲を表します。その範囲に、text 部分に書かれたことが書かれます。今回はテキストが+でつながれていますが、これでも ok です。

Count.ToString();：int 型変数 Count を string 型に変換する、という意味。変数 Count は、このままでは文字として扱えないので、こんなことをする必要があります。

以上より、この文は「画面左上に Item の残っている数を「残り～個」と表示する」という意味です。

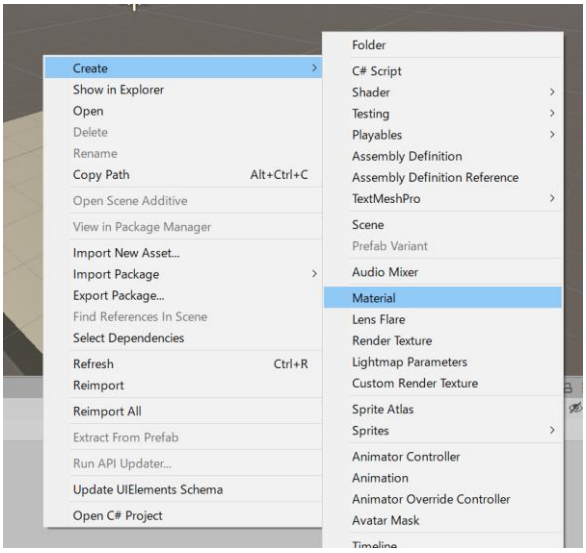
② 実行すると、左上に Item の残った数が表示されます。Item をとると数が 1 減ります。



2．マテリアル

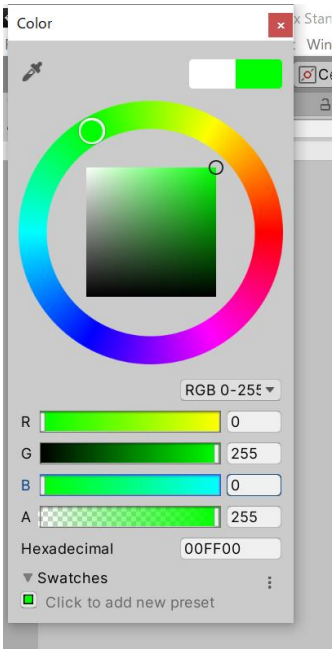
見た目がぱっとしないので、オブジェクトの色を変更します。

① プロジェクトビュー上で右クリックし、Create>Material を選択します。

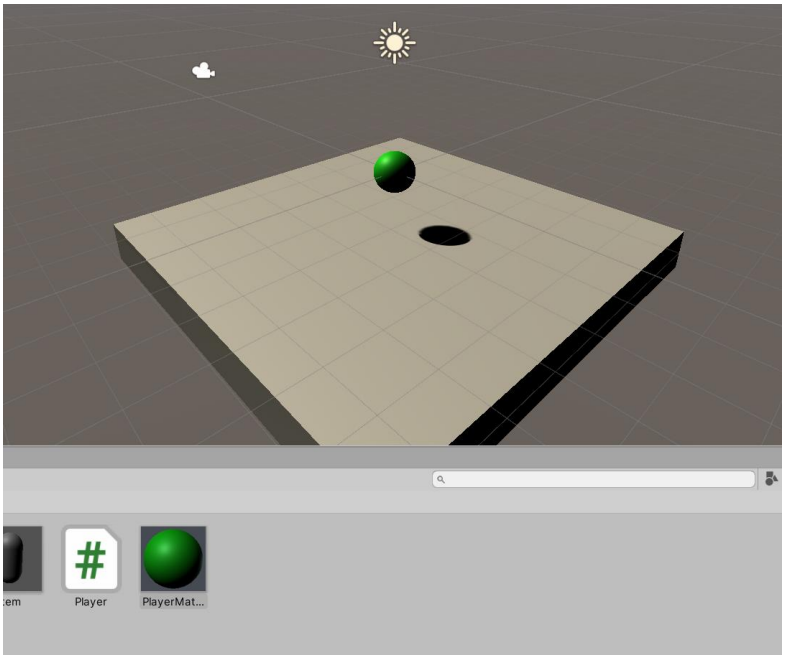


作成時に名前の入力ができるようになるので PlayerMaterial と名付けましょう。Material はオブジェクトの描画にかかわるコンポーネントで、色や質感を変更できます。

② インスペクターの Main Maps から Albedo の右の四角を選択します。すると色を選べる画面が現れるので、好きな色に変更してみましょう。

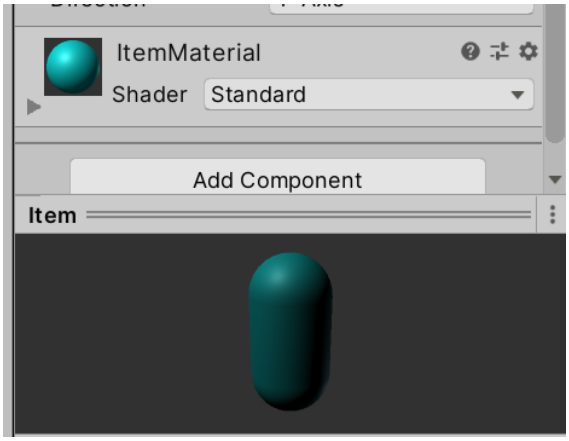


③ プロジェクトビューの PlayerMaterial をシーンの Player にドラッグアンドドロップします。すると Player の色が変わります。これは Player に PlayerMaterial が適応されたからです。



④ 同様にして ItemMaterial を作成し、色を変更しましょう。

⑤ Item のプレハブを選択し、インスペクターの一番下の Add Component の横に ItemMaterial をドラッグアンドドロップしましょう。すると DefaultMaterial が ItemMaterial に置き換わります。



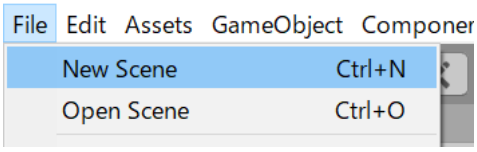
⑥ お好みで足場の色も変えましょう。

3．シーン作成

ゲームの始まりと終わりを作りましょう。最初に言ったっきりだったシーン機能を使います。

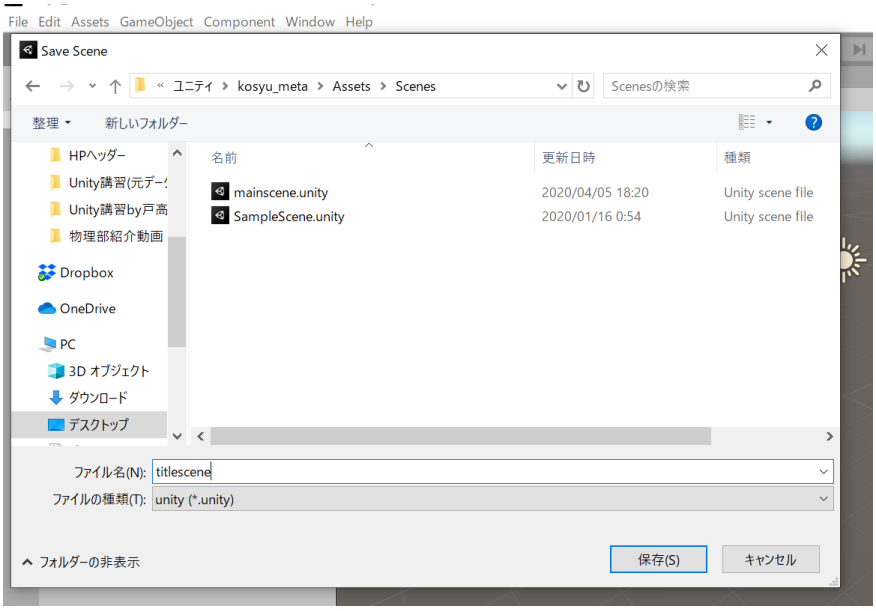
① 今のシーンを保存してください。

② 画面左上の File>New Scene を押してください。

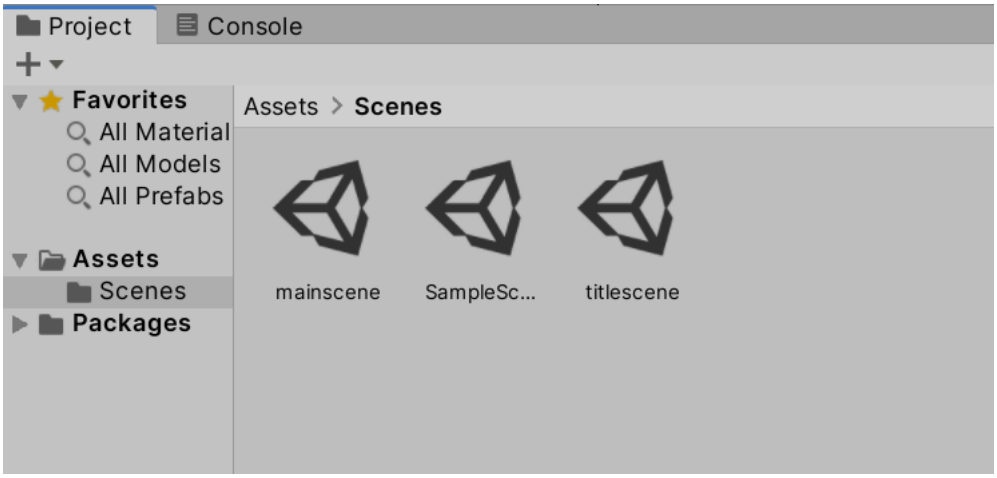


するとシーンからオブジェクトがなくなりました。これはデータが消えたわけではなく、新しいシーンになったからです。

③ File>Save As をクリックし、保存先に Scenes を選択します。ファイル名を titlescene にして保存ボタンを押します。

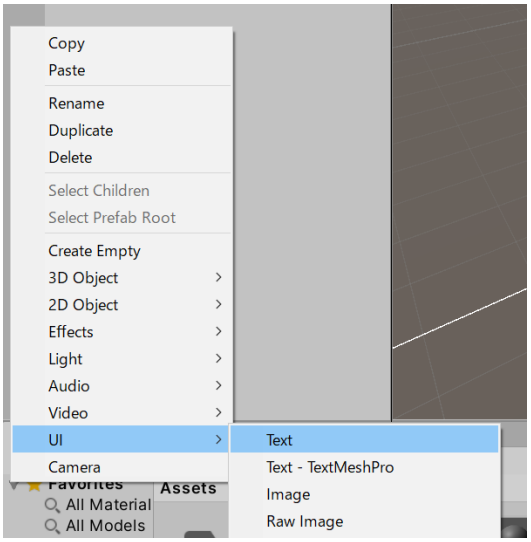


④ プロジェクトビューの Scenes を選択すると mainscene,titlescene,SampleScene の三つがあるのがわかります。mainscene を選択すると、先ほどまで編集していたシーンへ戻ることができます。

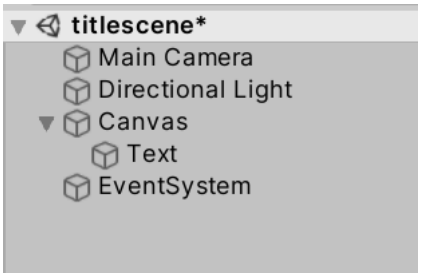


⑤ このゲームのタイトルを表示させます。OnGUI でもいいのですが、別の方法を使います。

ヒエラルキーで右クリックして、UI>Text を選択します。



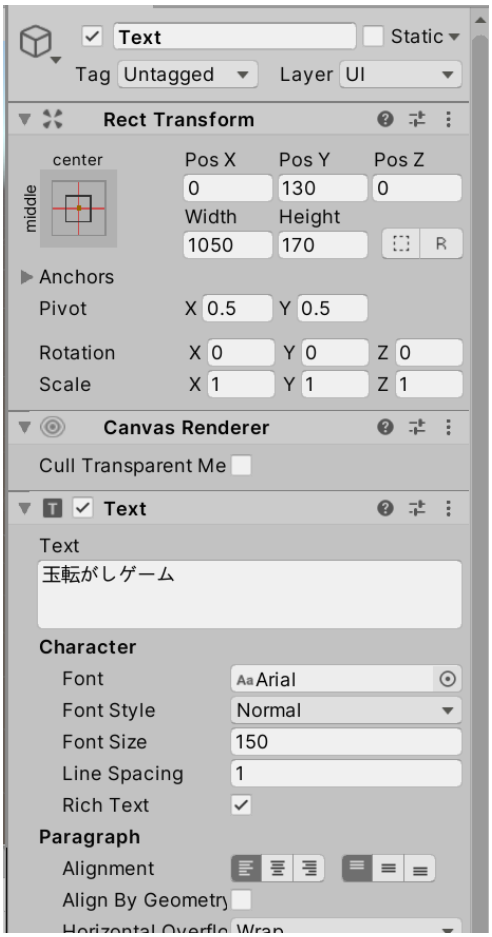
すると、Canvas、Text、EventSystem が作られます。



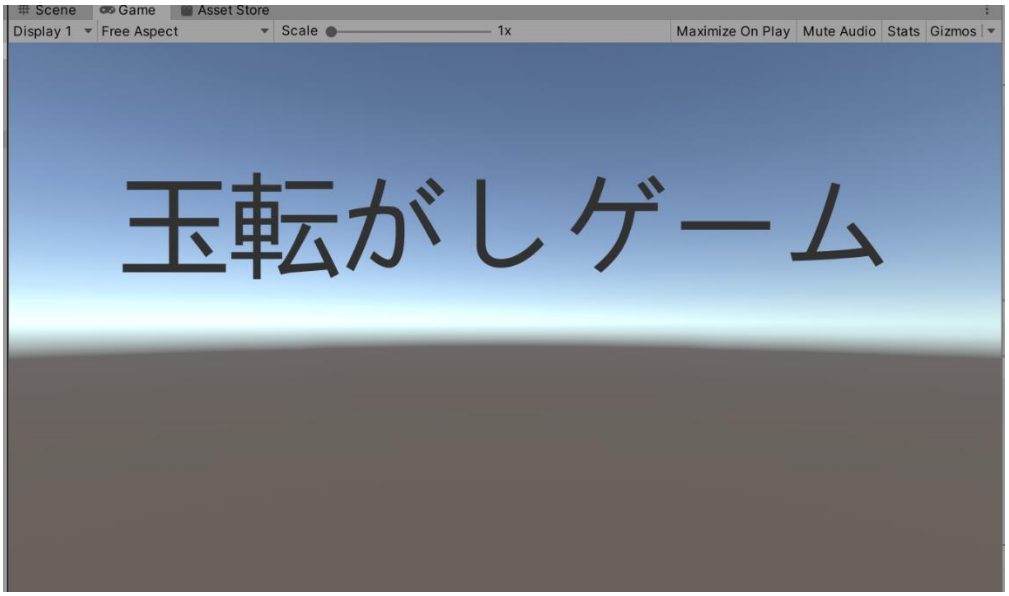
Text が文字で、Canvas はそれを映し出すスクリーンだと思ってください。また、Text は Canvas の子オブジェクトです。子オブジェクトは、親であるオブジェクトから様々な影響を受けるようになります（座標など）。詳しいことは後でやります。

EventSystem はボタンなどの入力を感知するために必要なオブジェクトです。いじる必要はありません。

⑥ Text のコンポーネントの Rect Transform,Text,Font Size を以下のように変えてください。PosX,Y,Z は中心の座標、Width,Height は大きさ、Text は書かれる内容、Font Size は字の大きさを表します。



⑦ 実行すると「玉転がしゲーム」と書かれているのが分かります。

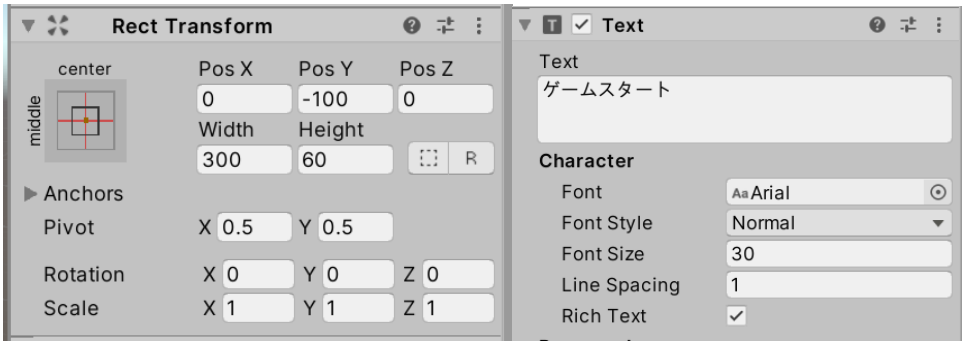


4．シーン遷移

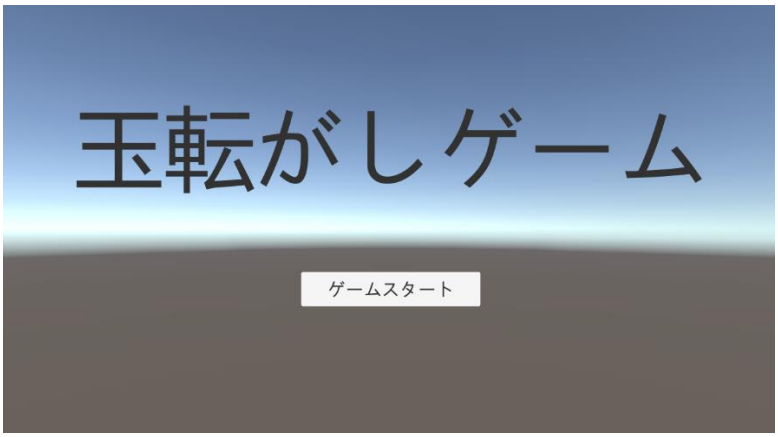
実行中にシーンを変更できるようにします。

① UI>Button からボタンオブジェクトを作ります。

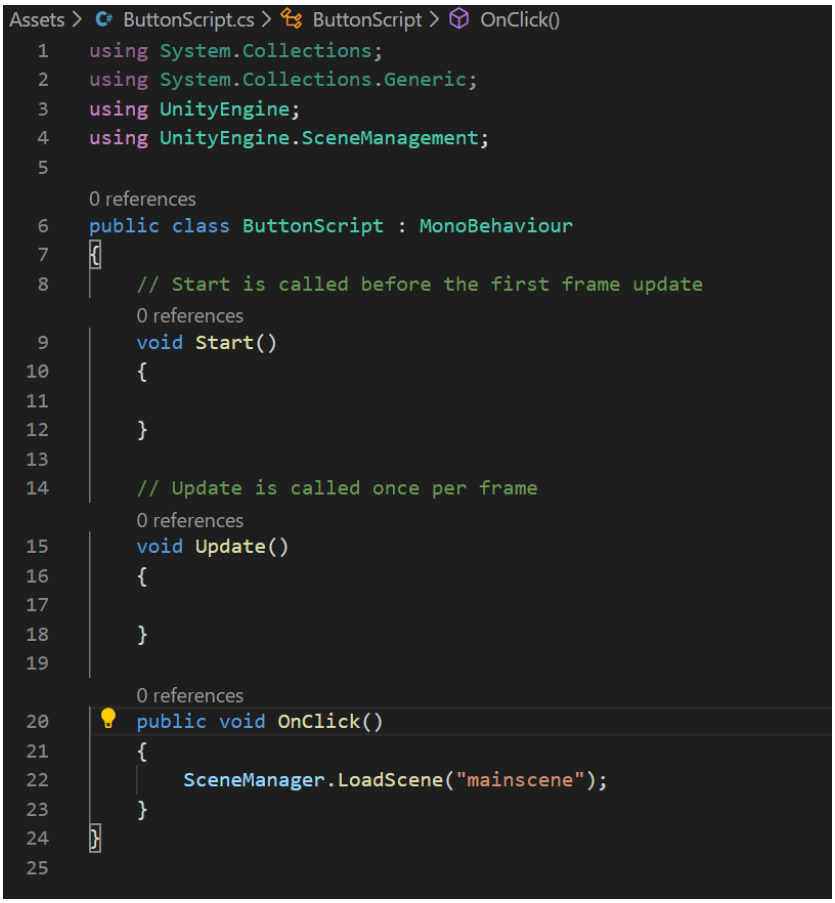
② Button の Rect Transform コンポーネントと子オブジェクトの Text の Text を編集します。



③ 実行するとボタンが表示されているのが分かります。しかし、押しても反応しません。



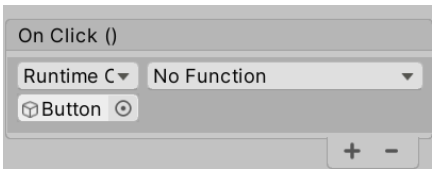
④ 押すと mainscene に移動するようにします。Button に Add Component>New Script から ButtonScript という名前でスクリプトを追加し、以下のように編集します。4 行目にも追加されている命令があるので注意してください。また、OnClick には public がついていることにも注意してください。



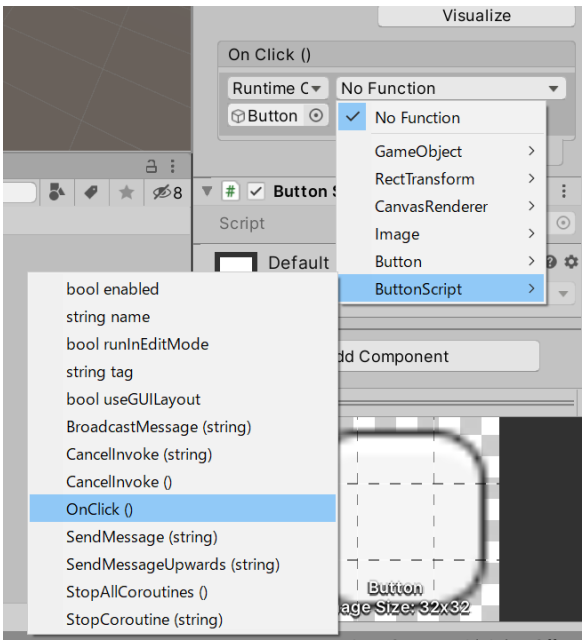
４行目：シーンに関する命令を書くときに必要な記述です。
OnClick（）のようなものを「メソッド」といいます。Update や OnTriggerEnter２D などもメソッドの一種ですが、これらは特別な意味をもち、特定の時に呼び出されます。Onclick は特別な意味を持たないので、このままだと呼び出される事無く終わってしまいますが、あとからボタンが押されたときに呼び出されるようにします。くわしいことはいつかやります。
SceneManager.LoadScene(シーン名);：カッコ内のシーンに移動できます。

⑤ Button の Button コンポーネントの一番下にある On Click の+マークをクリックします。

⑥ None となっているところにヒエラルキーの Button をドラッグアンドドロップします。



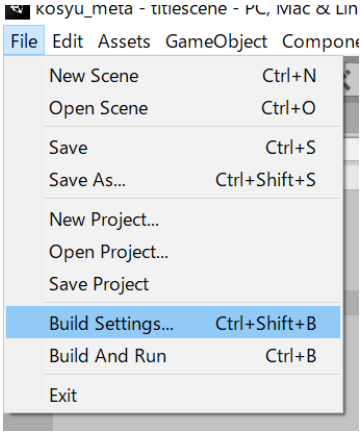
⑦ No Function となっているところを選択し、Button Script>OnClick()をクリックします。



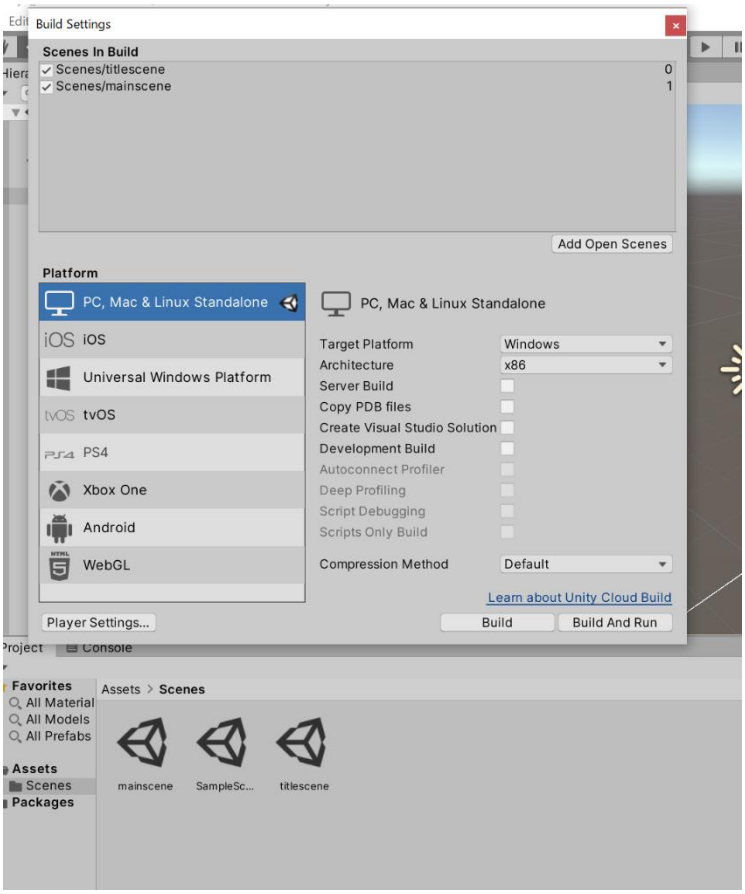
これで、ボタンが押されると ButtonScript の OnClick 内の処理が実行されるよう

になりました。

⑧ File>Build Settings を開きます。



⑨ Scenes in Build にプロジェクトビューから titlescene と mainscene をドラッグアンドドロップします。



これをしないとシーンを移動することができません。

⑩ 実行してボタンを押すと、シーンが変わりました。

5．ゲームを終了させるシーンを作る

はじめのシーンを作ったので終わりのシーンも作りましょう。

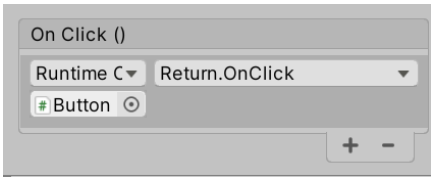
① 現在のシーンを保存し、File>New Scene でシーンを作り、Save As から endscene という名前で保存します。

② 以下のようになるように、Text と Button を作って配置してください。位置はだいたいで構いません。

③ もう一度、と書かれた方に Return という名前でスクリプトを追加して、以下のように編集します。

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

0 references
public class Return : MonoBehaviour
{
    // Start is called before the first frame update
    0 references
    void Start()
    {

    }

    // Update is called once per frame
    0 references
    void Update()
    {

    }

    0 references
    public void OnClick()
    {
        SceneManager.LoadScene("titlescene");
    }
}
```

④ ボタンが押されるとこのスクリプトが実行されるようにします。



⑤ File>Build Settings からこのシーンを Scenes in Build に入れます。

⑥ 実行すると、もう一度ボタンが押されるとタイトルに移動するのが分かります。

6．ゲームの終了

終わるボタンを押すとゲームが終了するようにしましょう。

① 終わるボタンに Quit という名前のスクリプトを追加し、以下のように編集しましょう。

```
    }

    // Update is called once per frame
    0 references
    void Update()
    {

    }

    0 references
    public void OnClick()
    {
        #if UNITY_EDITOR
            UnityEditor.EditorApplication.isPlaying = false;
        #else
            Application.Quit();
        #endif
    }
}

```

#if は if(){}とだいたい同じと考えてください。#if UNITY_EDITOR で、unity エディターで実行されているなら、という意味です。

UnityEditor.…… ＝false は、実行状態でなくす、という意味です。再生ボタンを押したときと同じです。

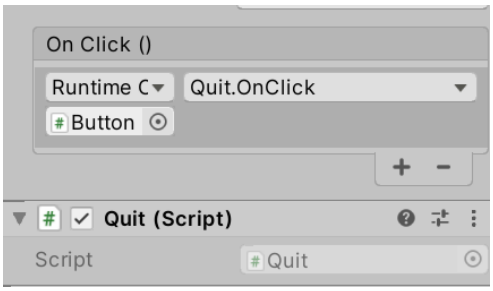
#else は、#if と同時に使われ、そうでないなら、という意味になります。この場合、unity エディター以外で実行されているなら、という意味です。

Application.Quit();はアプリを終了する、という意味です。ウィンドウ右上の×を

押したときと同じです。

以上より、このスクリプトは「Unity エディター上で実行されているなら実行状態でなくし、それ以外で実行されているならアプリを終了する」という意味になります。Unity エディター以外で実行することがあるのかよ、と思うかもしれませんが、ゲームが完成したときに実行するのは PC やスマホや switch です

② ボタンが押されたときにこのスクリプトが実行されるようにします。



③ 実行して終わるボタンを押すと、実行状態が解除されます。

7．シーン遷移

メインシーンでアイテムをすべて取ったらエンドシーンに移動するようにします。

① 現在のシーンを保存して、mainscene に移動してください。

② GameController スクリプトを以下のように編集します。

４行目の追加と、Count = ～の位置の変更を忘れないで下さい。

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

0 references
public class Gamecontroller : MonoBehaviour
{
    1 reference
    public GameObject Item;
    3 references
    public Vector3[] ItemPosition;
    // Start is called before the first frame update
    0 references
    void Start()
    {
        for(int i = 0;i < ItemPosition.Length;i++)
        {
            ItemPosition[i] = new Vector3(Random.Range(-5,5),0,Random.Range(-5,5));
            Instantiate(Item,ItemPosition[i],Quaternion.Euler(0,0,0));
        }
    }

    // Update is called once per frame
    0 references
    void Update()
    {
        Count = GameObject.FindGameObjectsWithTag("Item").Length;
        if(Count == 0)
        {
            SceneManager.LoadScene("endscene");
        }
    }

    3 references
    public int Count;
    0 references
    void OnGUI()
    {
        GUI.TextField(new Rect(10,10,300,80),"残り" + Count.ToString() + "個");
    }
}

```

24 行目：「＝」は代入を意味しましたが、「＝＝」は左辺と右辺が同じかを判断する条件式となります。if(Count ==0)で「Count が 0 なら」の意味です

③ 実行すると、アイテムをすべて取ったときに endscene に移動します。

8．クリアできないことをなくす

このままだと、Player が足場から落ちるとクリアできないままになってしまいます。そのため、足場から落ちないようにします。

① Player スクリプトを以下のように編集します。

```
if (Input.GetKey(KeyCode.DownArrow))
{
    GetComponent<Rigidbody>().AddForce(0f, 0f, -Speed);
}

if(transform.position.x >5)
{
    transform.position = new Vector3(5,transform.position.y,transform.position.z);
}
if(transform.position.x <-5)
{
    transform.position = new Vector3(-5,transform.position.y,transform.position.z);
}
if(transform.position.z >5)
{
    transform.position = new Vector3(transform.position.x,transform.position.y,5);
}
if(transform.position.z <-5)
{
    transform.position = new Vector3(transform.position.x,transform.position.y,-5);
}

}
```

transform.position：Transform コンポーネントの position の情報。

transform.position.x：Transform コンポーネントの position の情報のうち X の数値。

transform.position.x > 5：>は数学的な意味どおり、左辺のほうが大きいことを判断する条件式です。if の中で「X 座標が 5 より大きいなら」という意味になります。

以上から、X 座標が 5 以上なら X 座標を 5 にする、というような命令です。

なお、transform.position.x = 5;とは書けません。

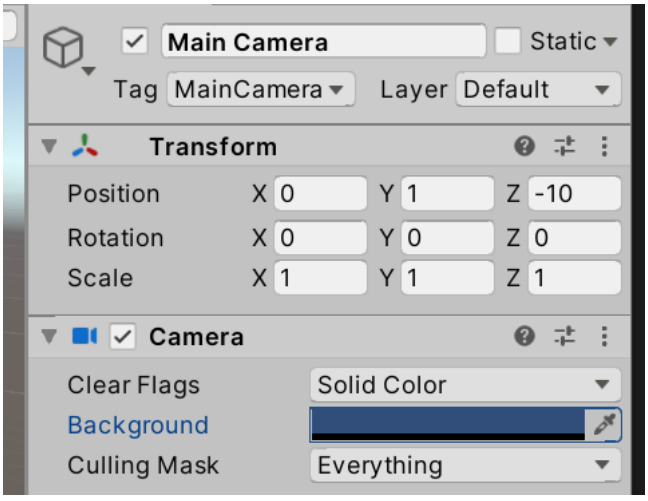
9．背景の変更

背景を変更します。

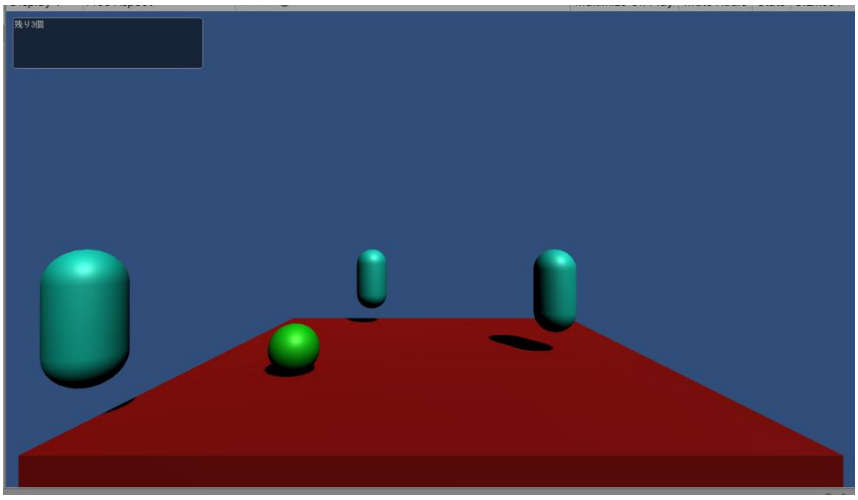
① Main Camera を選択します。

② Clear Frags を Sky Box から Solid Color に変更します。

③ Background で好きな色に変更します。



④ 実行すると、背景が変わります。

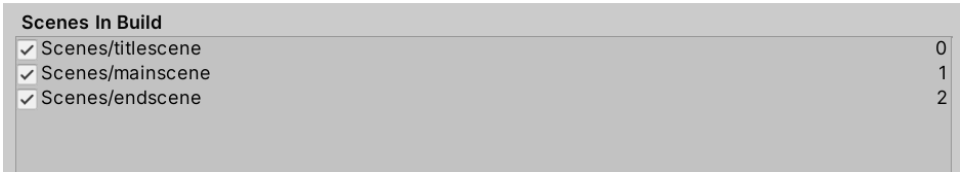


⑤ ほかの二つのシーンの背景も変更しましょう。

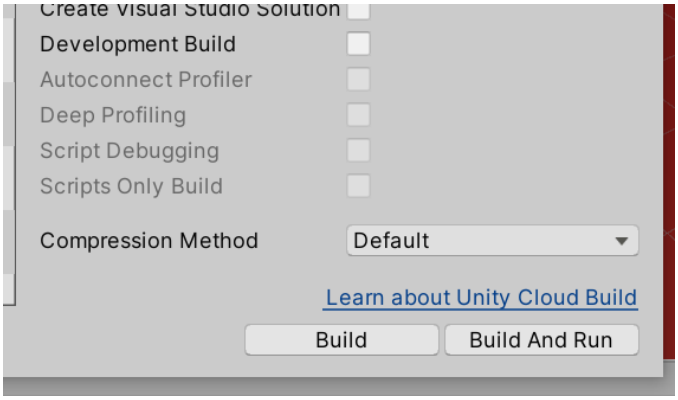
１０．　ビルド

ゲームをアプリケーションにします。

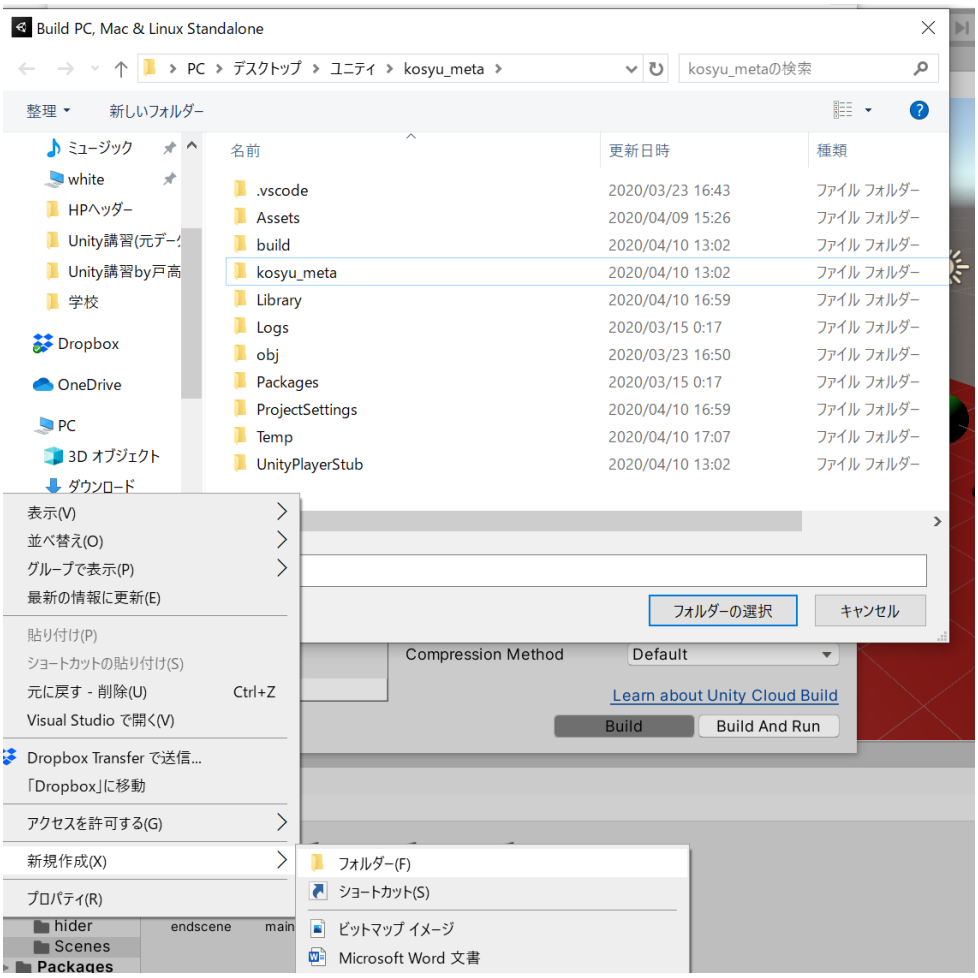
① File>Build Setting から順番が以下のようになっていることを確認します。なっていなければドラッグで順番を変更してください。



② Build を選択します。



③ 実行ファイルを作る位置を聞かれるので、右クリック>新規作成>フォルダーで EXE という名前のフォルダを作り、それを選択します。



④ しばらく待ちます。

⑤ ファイルが完成するので、その中の 「名前.exe」というものを実行します。

⑥ 先ほどまで作っていたゲームが遊べます。

⑦ これにてゲームの完成です！

これでゲームの完成です。お疲れさまでした。次回からは２Ｄゲームの作り方をやっていきます。