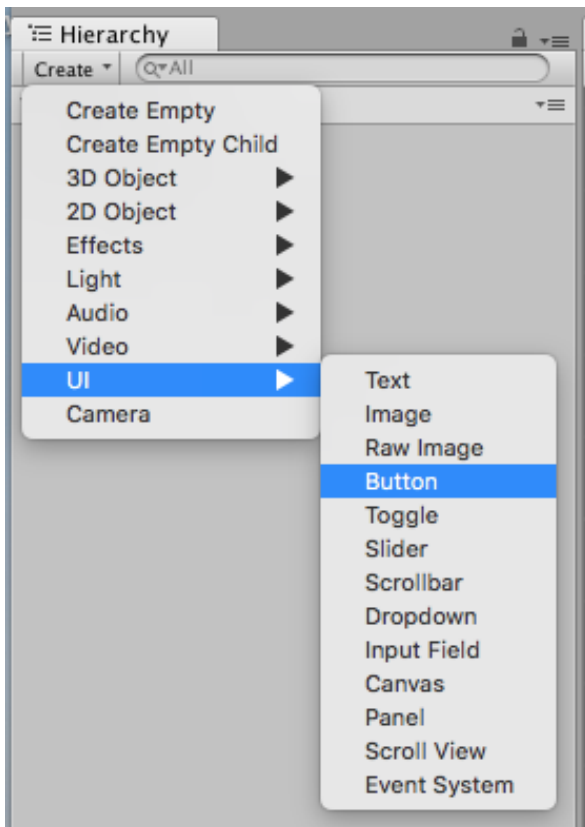


Which is more? (後編)

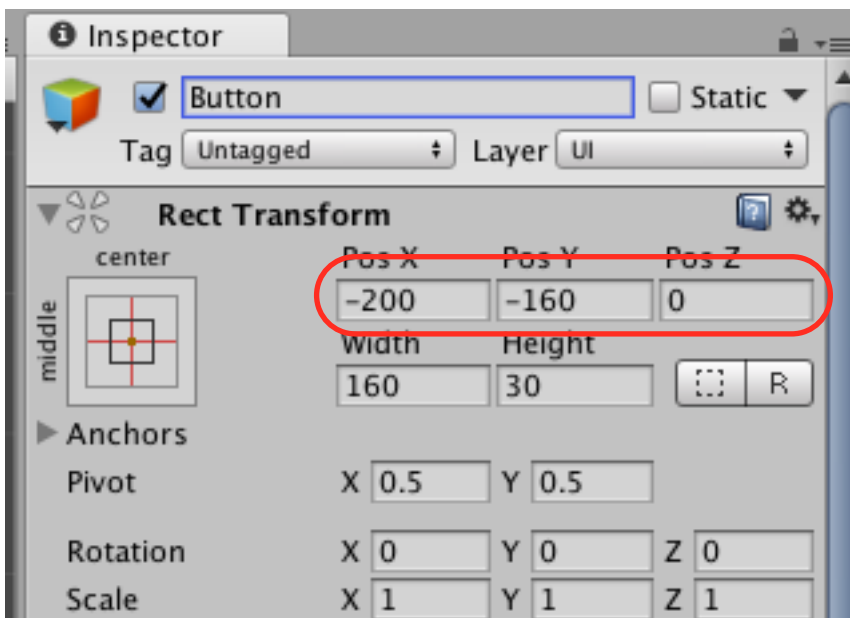


ボタンをつくる。

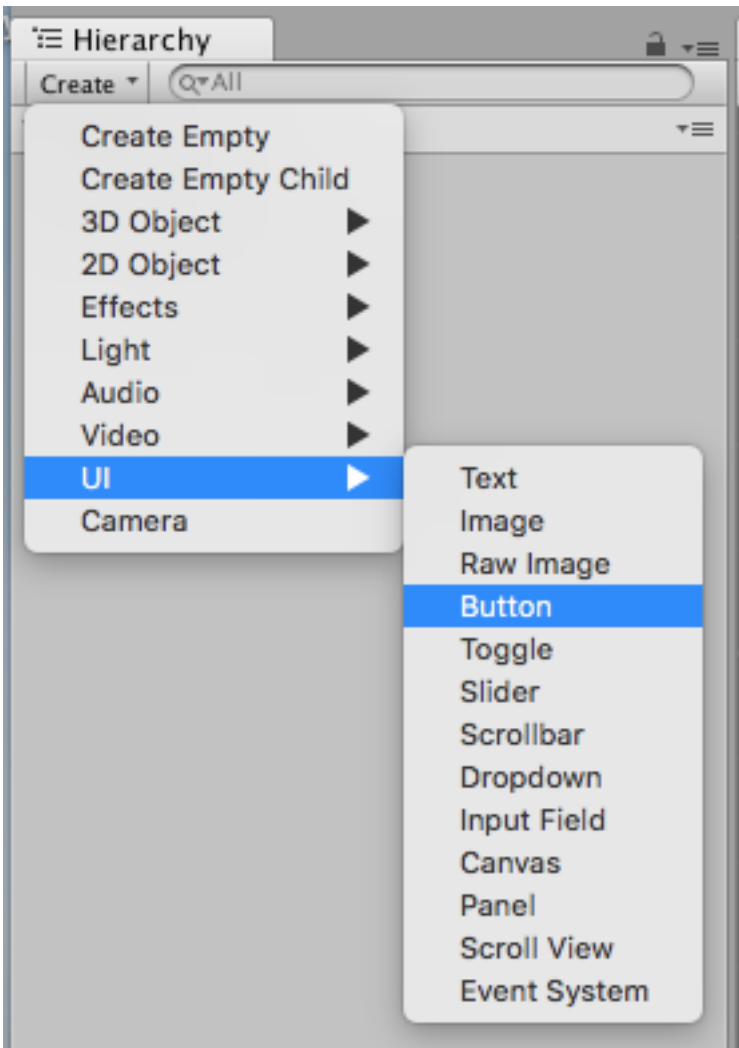
ヒエラルキーで、Create→UI→Buttonの順に選択します。



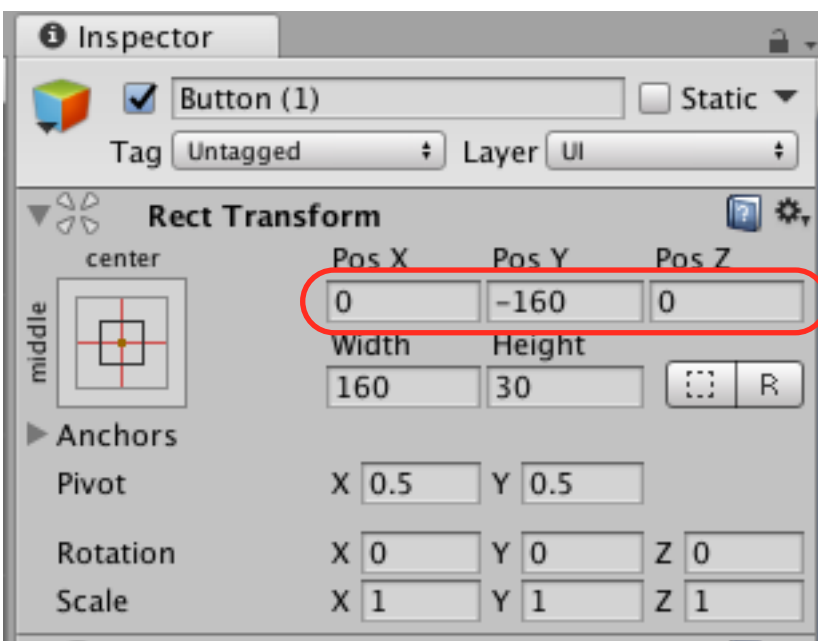
ボタンの位置を調整します。



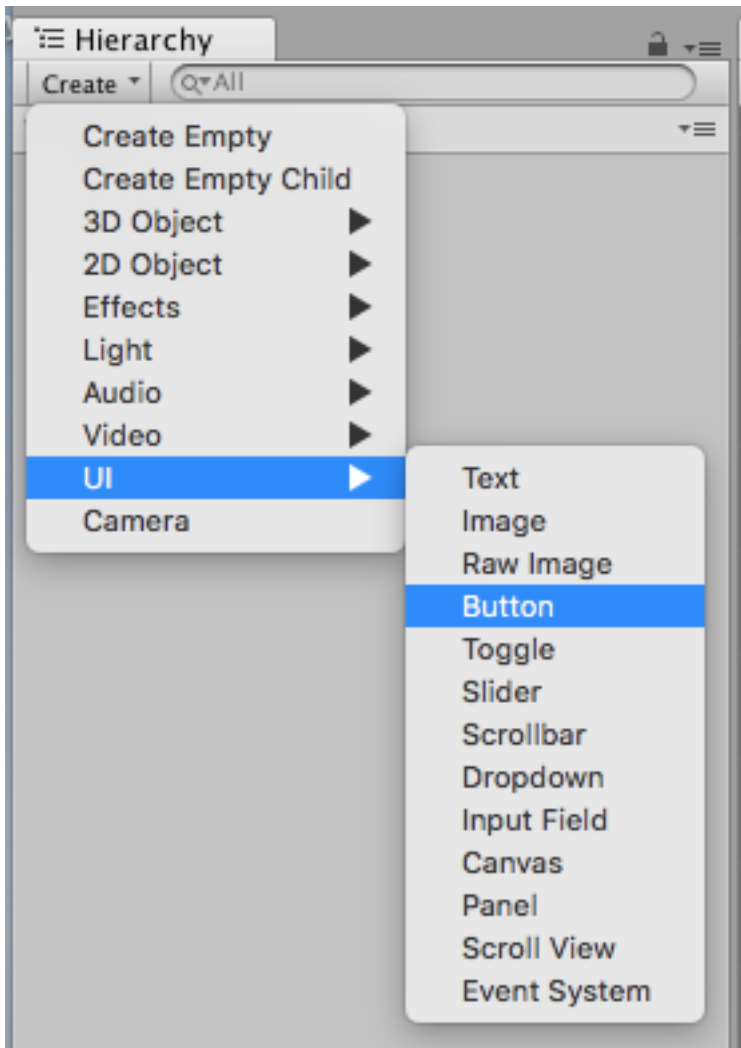
2つ目のボタンを作ります。



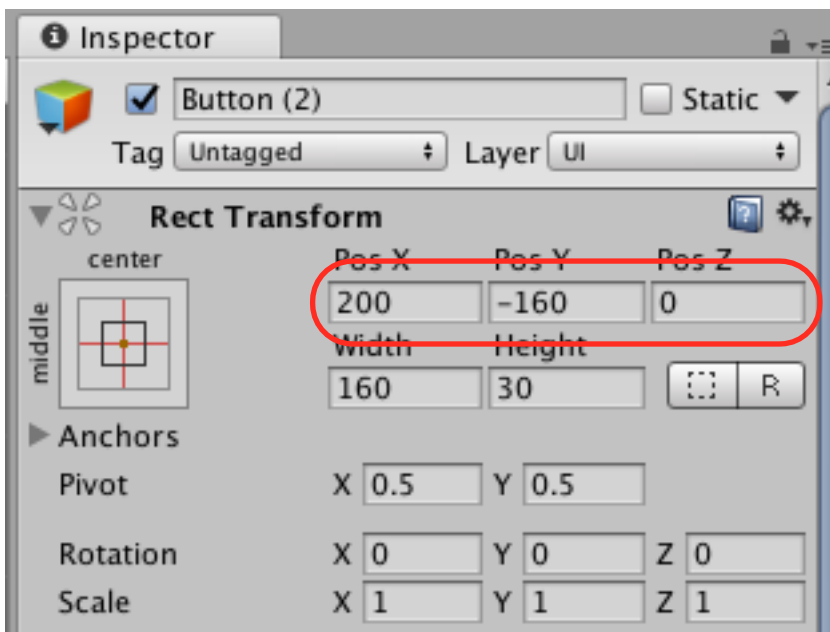
ボタンの位置を調整します。



3つ目のボタンを作ります。

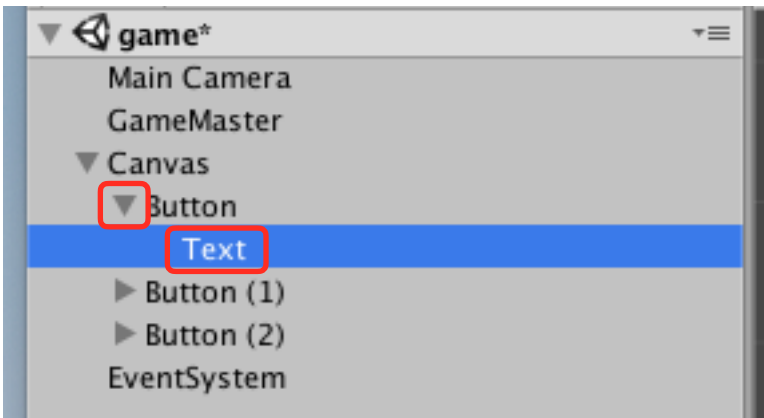


ボタンの位置を調整します。

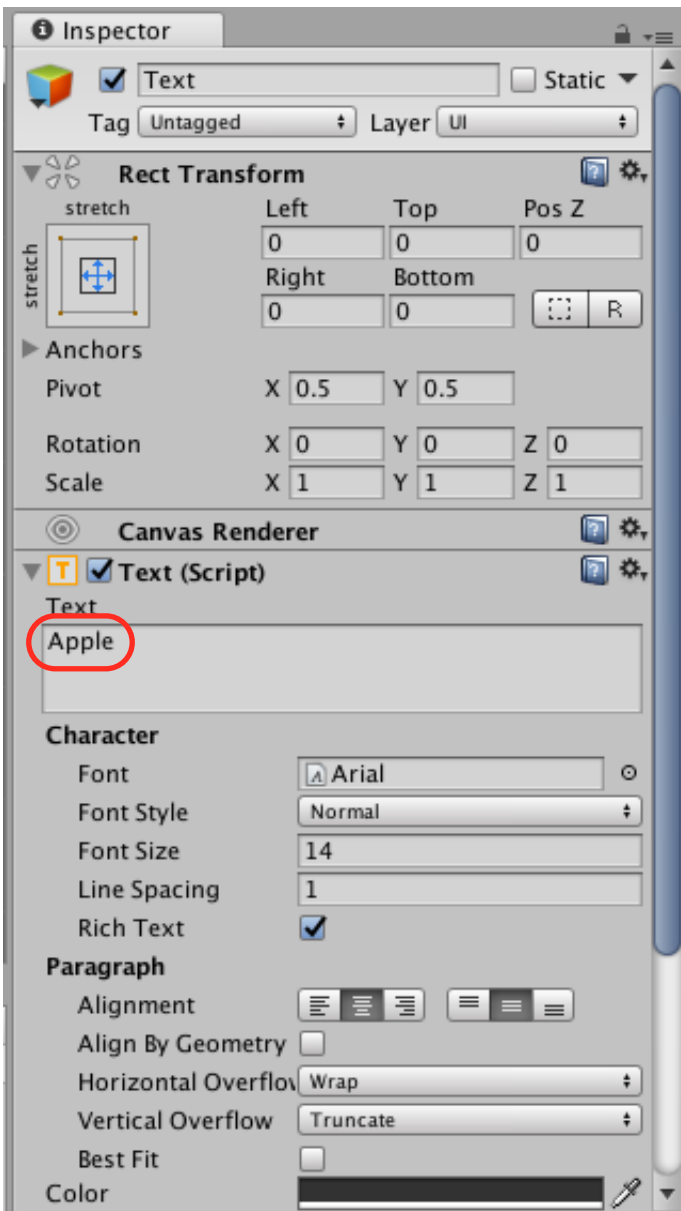


ボタンの中のテキスト (Text) を書き換える。

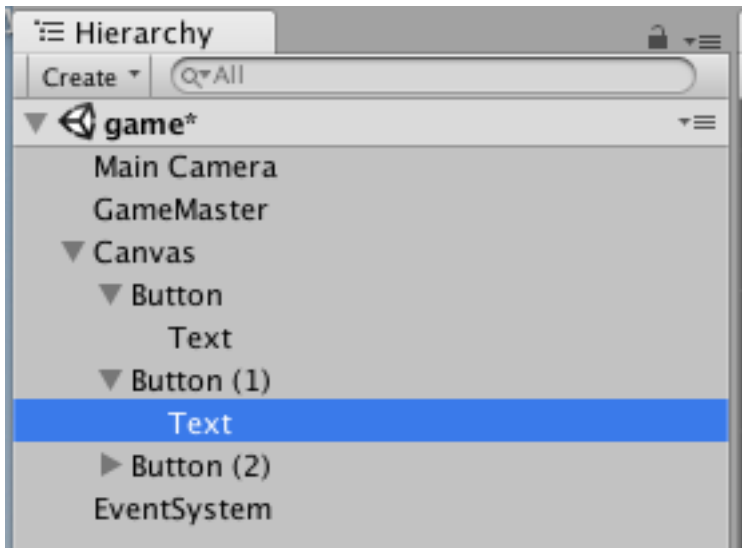
Button (ボタン) の左側の三角マークをクリックし、Text (テキスト) を選択します。



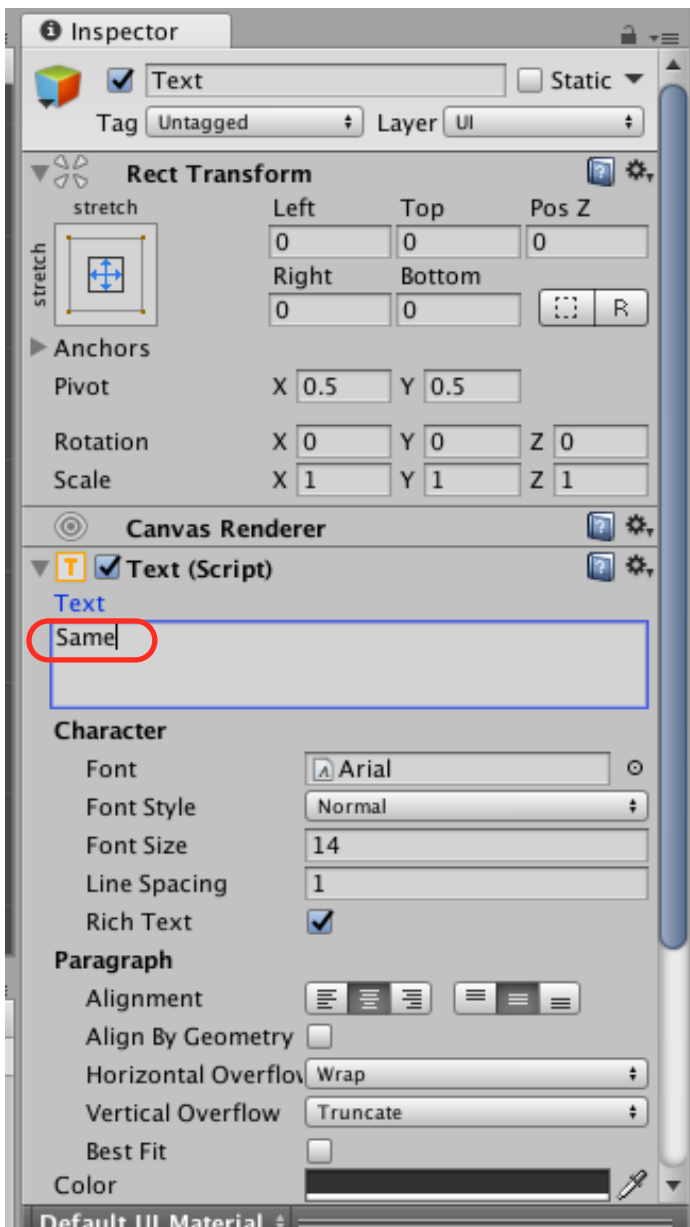
Textを「Apple」 (りんご) とします。



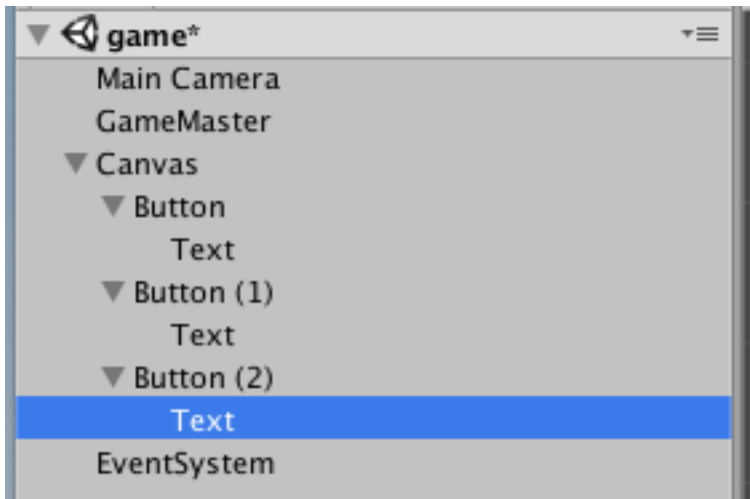
Button(1)の左側の三角マークをクリックし、Textを選択します。



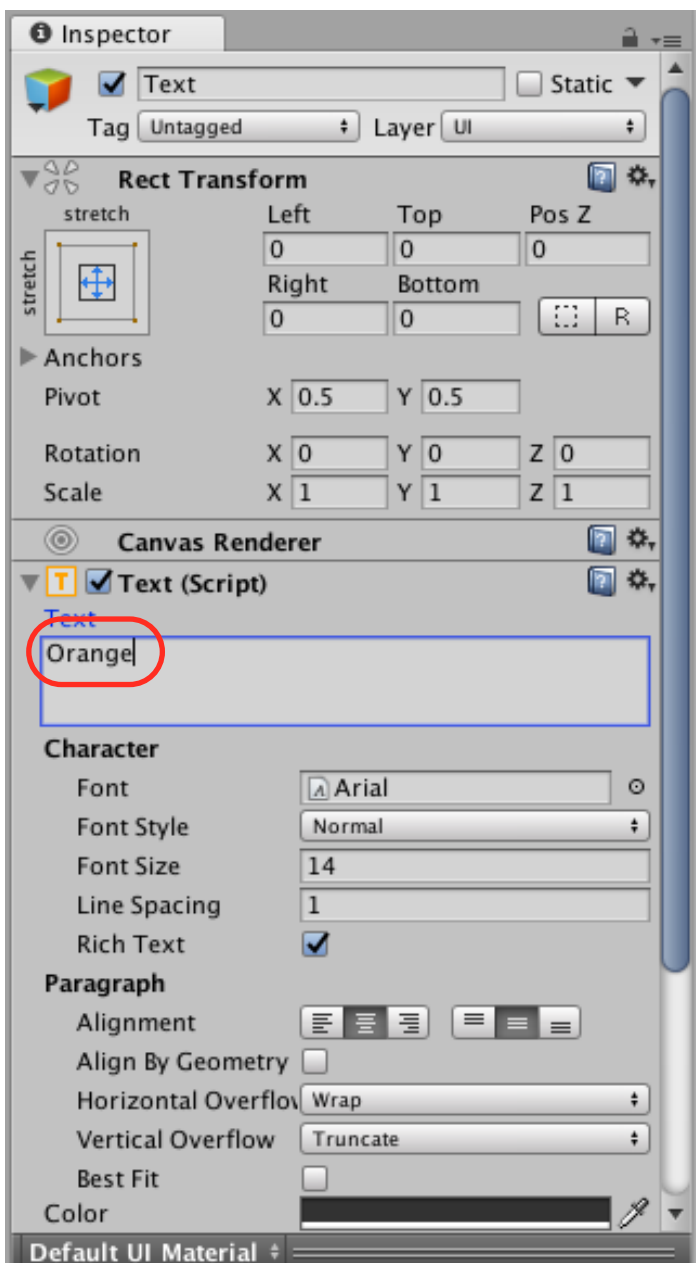
Textを、「Same」（同じ）とします。



Button(2)の左側の三角マークをクリックし、Textを選択します。

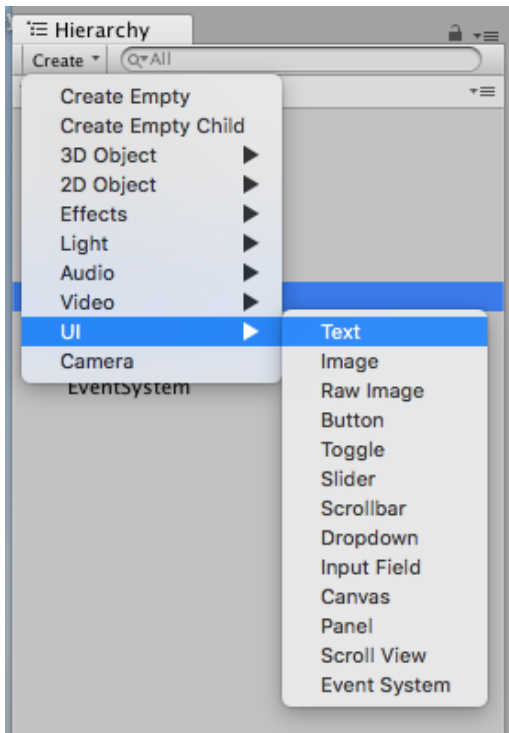


Textを「Orange」（みかん）とします。

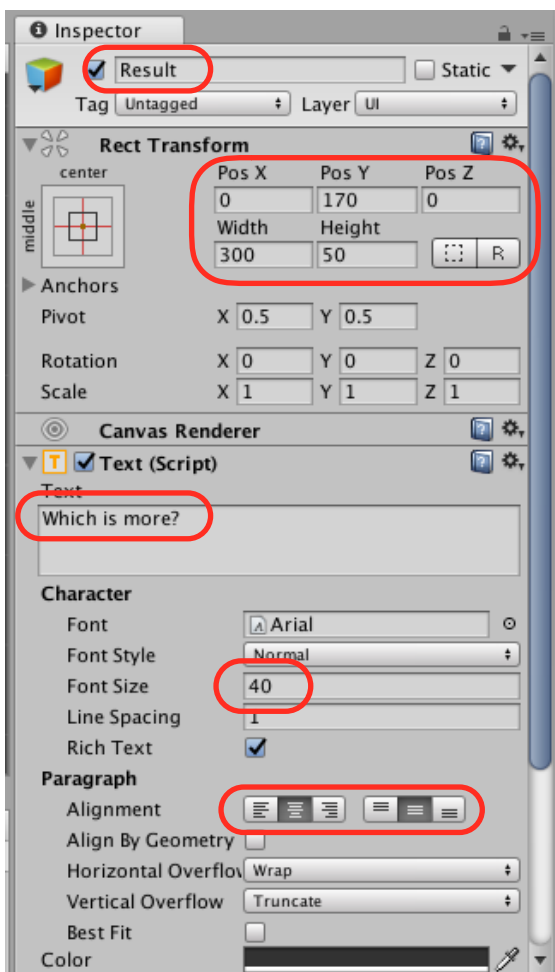


メッセージをつくる。

ヒエラルキーで、Create→UI→Textの順に選択します。



インスペクターで、赤枠部分を変更します。



スクリプトを編集する。

赤線部分を追加します。

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class Master : MonoBehaviour {
7
8     public GameObject applePrefab;
9     public GameObject orangePrefab;
10
11     int numberApple;
12     int numberOrange;
13
14     float xPos;
15     float yPos;
16
17     public GameObject result;
18
19     // Use this for initialization
20     void Start () {
21         //りんごとみかんの個数をランダムに決める。
22         numberApple = Random.Range (1, 10);
23         numberOrange = Random.Range (1, 10);
24
25         //りんごの個数・みかんの個数のログをとる。
```

ボタンを押した時の処理を作る。

Updateの下に、赤枠で示した、新しい関数を作ります。

```
45
46 // Update is called once per frame
47 void Update () {
48
49 }
50
51 //ボタンを押した時の処理
52 public void Button0() {
53     |
54 }
55
56
57
58 }
59
```

その関数の中にif文を入れます。

```
45
46 // Update is called once per frame
47 void Update () {
48
49 }
50
51 //ボタンを押した時の処理
52 public void Button0() {
53     if (numberApple > numberOrange) {
54
55     }
56 }
57
58
59
60 }
61
```

if文の中の処理を追加します。

```
45
46 // Update is called once per frame
47 void Update () {
48
49 }
50
51 //ボタンを押した時の処理
52 public void Button0() {
53     if (numberApple > numberOrange) {
54         result.GetComponent<Text> ().text = "Collect";
55     }
56 }
57
58
59
60 }
61
```

else文（でなければ）を追加します。

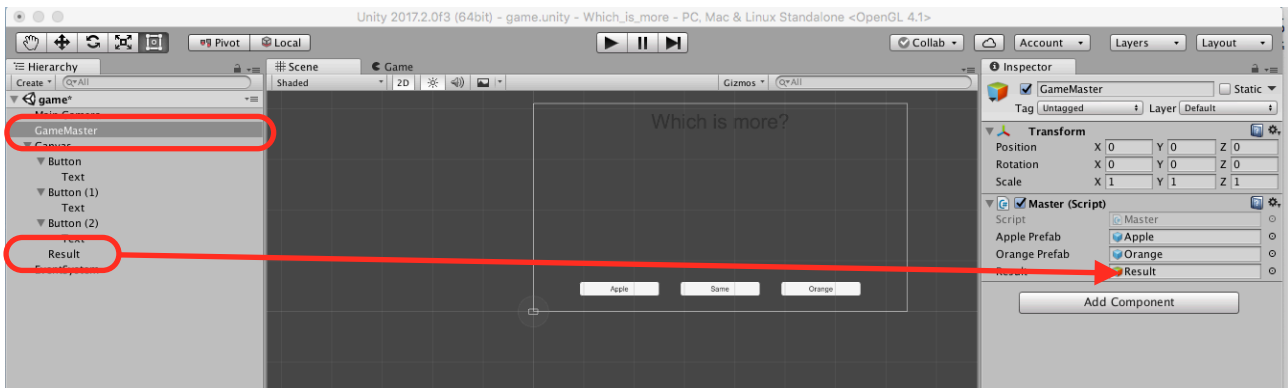
```
45
46 // Update is called once per frame
47 void Update () {
48
49 }
50
51 //ボタンを押した時の処理
52 public void Button0() {
53     if (numberApple > numberOrange) {
54         result.GetComponent<Text> ().text = "Collect";
55     } else {
56
57     }
58 }
59
60
61
62 }
63
```

else文の中の処理を追加します。

```
45
46 // Update is called once per frame
47 void Update () {
48
49 }
50
51 //ボタンを押した時の処理
52 public void Button0() {
53     if (numberApple > numberOrange) {
54         result.GetComponent<Text> ().text = "Collect";
55     } else {
56         result.GetComponent<Text> ().text = "Incorrect";
57     }
58 }
59
60
61
62 }
63
```

メッセージのオブジェクトを設定する。

- ①GameMasterを選択します。
- ②Resultを、インスペクターの枠内にDDします。



スクリプトをコピペで増やす。

- ①下のように、ドラッグして範囲選択します。
- ②command+Cでコピーします(windowsは、Ctrl+C)。

```


45
46 // Update is called once per frame
47 void Update () {
48 }
49 }
50
51 //ボタンを押した時の処理
52 public void Button0() {
53     if (numberApple > numberOrange) {
54         result.GetComponent<Text> ().text = "Collect";
55     } else {
56         result.GetComponent<Text> ().text = "Incorrect";
57     }
58 }
59
60
61 }
62 }
63

```

- ③赤矢印の部分をクリックしてカーソルを表示する。
- ④command+V(windowsは、Ctrl+V)でペーストします。
- ⑤赤線部分を修正します。

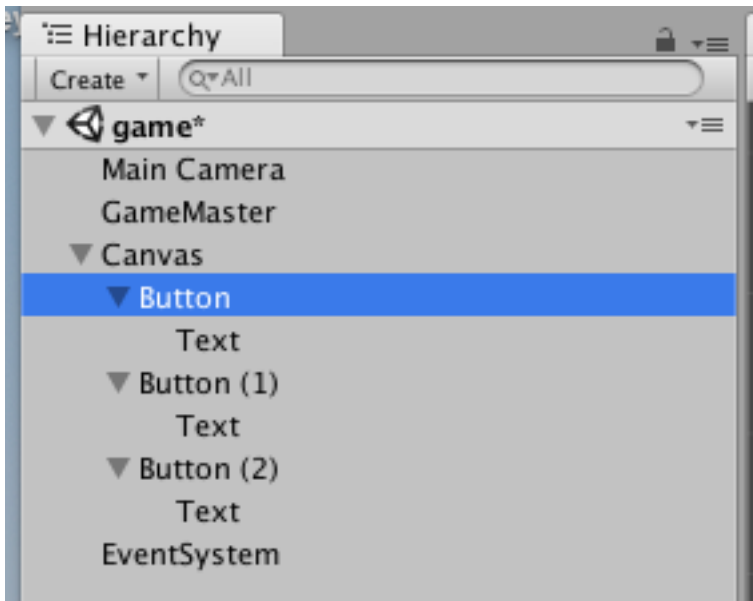
```
45
46 // Update is called once per frame
47 void Update () {
48
49 }
50
51 //ボタンを押した時の処理
52 public void Button0() {
53     if (numberApple > numberOrange) {
54         result.GetComponent<Text> ().text = "Collect";
55     } else {
56         result.GetComponent<Text> ().text = "Incorrect";
57     }
58 }
59 public void Button1() {
60     if (numberApple == numberOrange) {
61         result.GetComponent<Text> ().text = "Collect";
62     } else {
63         result.GetComponent<Text> ().text = "Incorrect";
64     }
65 }
66
67
68
69
70
71 }
72
```

- ⑥赤矢印の部分をクリックしてカーソルを表示する。
- ⑦command+V(windowsは、Ctrl+V)でペーストします。
- ⑧赤線部分を修正します。

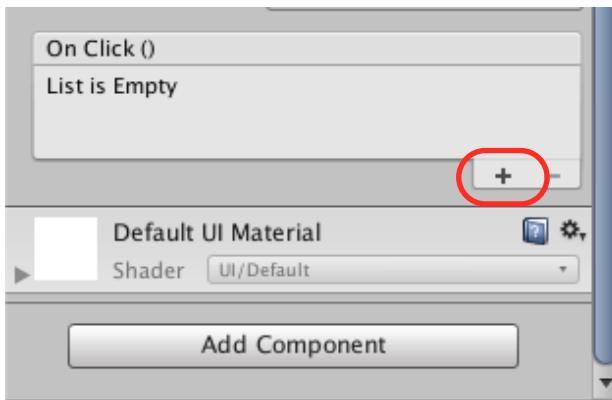
```
45
46 // Update is called once per frame
47 void Update () {
48
49 }
50
51 //ボタンを押した時の処理
52 public void Button0() {
53     if (numberApple > numberOrange) {
54         result.GetComponent<Text> ().text = "Collect";
55     } else {
56         result.GetComponent<Text> ().text = "Incorrect";
57     }
58 }
59 public void Button1() {
60     if (numberApple == numberOrange) {
61         result.GetComponent<Text> ().text = "Collect";
62     } else {
63         result.GetComponent<Text> ().text = "Incorrect";
64     }
65 }
66  public void Button2() {
67     if (numberApple < numberOrange) {
68         result.GetComponent<Text> ().text = "Collect";
69     } else {
70         result.GetComponent<Text> ().text = "Incorrect";
71     }
72 }
73 }
74
```

ボタンと関数を関連づける。

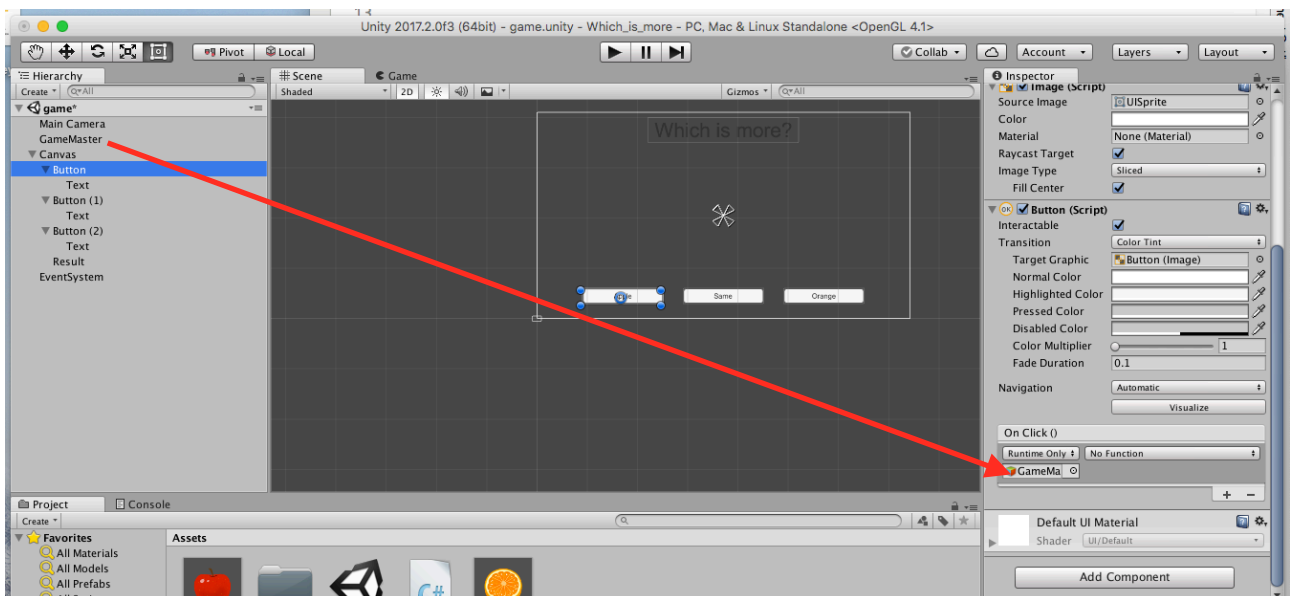
①ヒエラルキーで、Buttonを選択します。



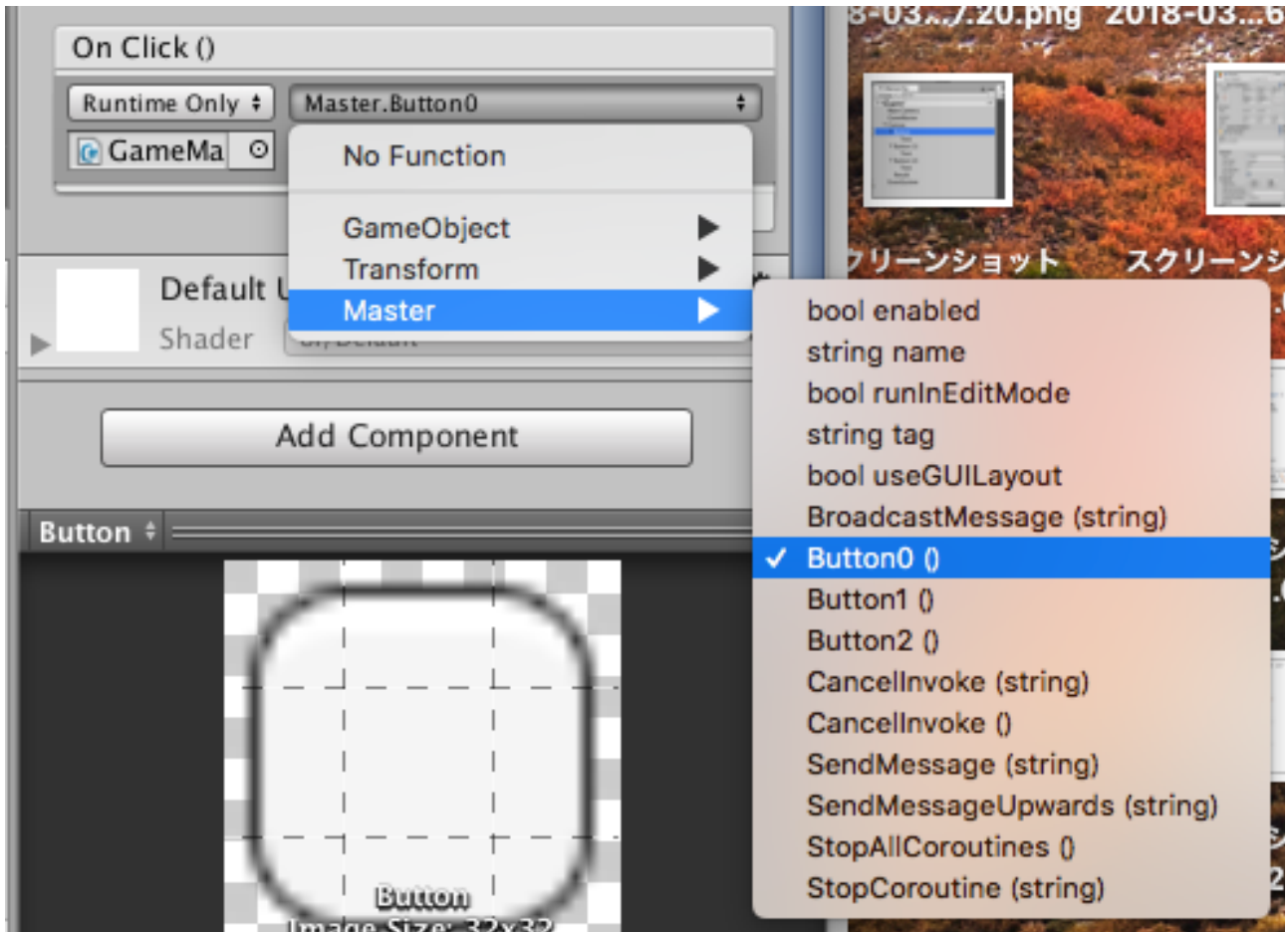
②インスペクターで、On Click()の「+」をクリックします。



③GameManagerを、On Click()の枠内にDDします。



④On Click()で、No Function→Master→Button 0 ()の順に選択します。



課題

- (1) Button(1)も同様に関数と関連づけしましょう。
- (2) Button(2)も同様に関数と関連づけしましょう。

完成です。

動かしてみましよう。

お疲れ様でした。