

SYN-G-GEN Tutorial #02

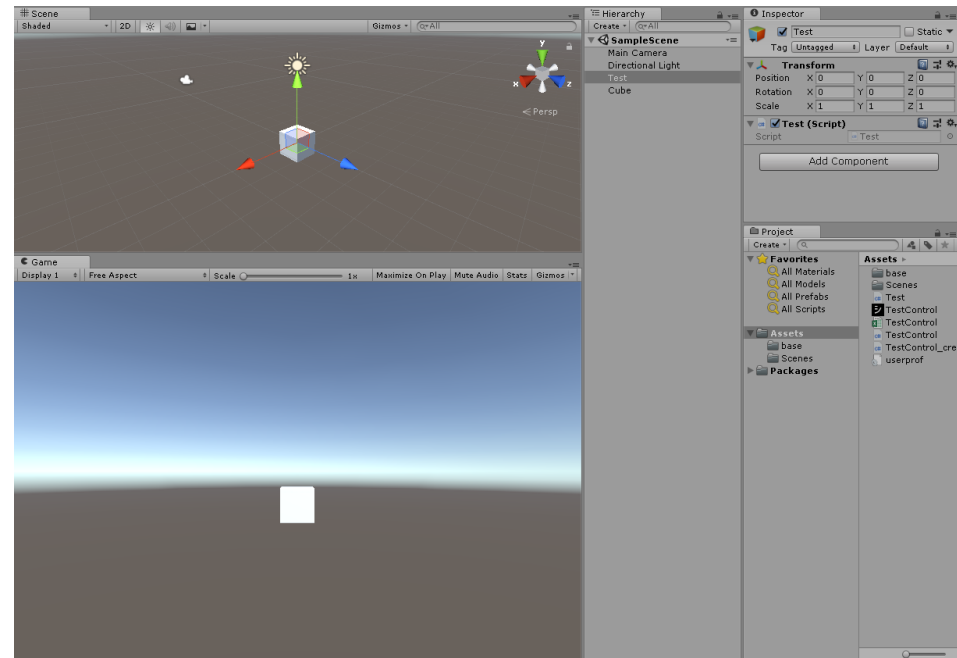
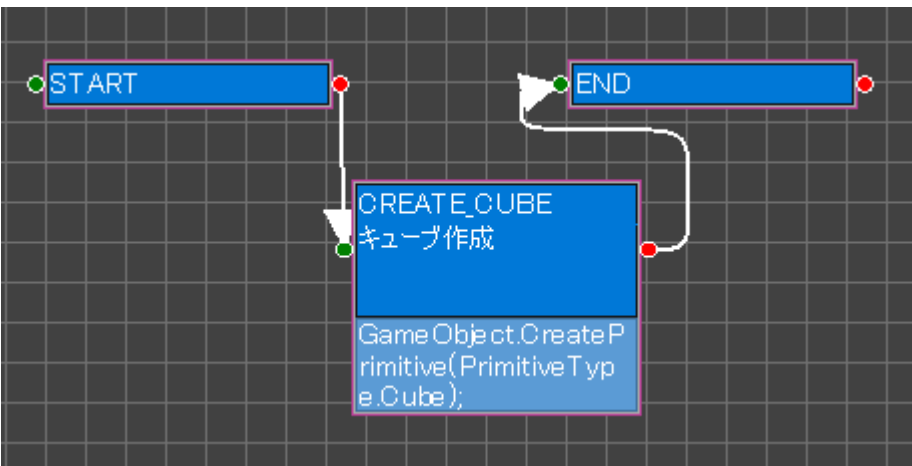
Target Unity

Programanic

2018/9/30

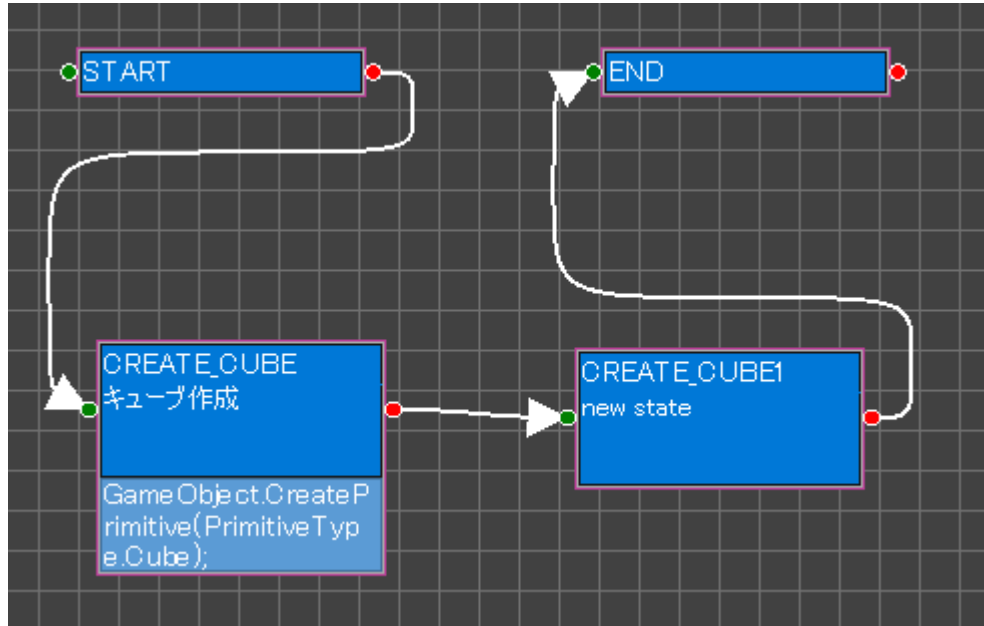
Step 1

Tutorial #01の完成



Step 2

分岐ステート作成



新規ステートを作成し、左図のように矢印をつなげる。

	Row	NAME	STATE
	1	thumbnail	(bitmap)
▶	2	state	S_BRANCH
	3	state-cmt	分岐する
	4	state-ref	
	5	nextstate	S_END
	6		
	7	embed	

OK
CANCEL

Full edit を選択して、stateに “S_BRANCH”、そのコメントに”
分岐する”と記入。

Step 3

例) 乱数で0か1を決定

10		
11	vars	int x = 0;
12	init	x = UnityEngine.Random.Range(0,2);
13	init-cmt	
14		

分岐例として、0または1の乱数を発生させて変数xに格納する。

vars項目に int x = 0;

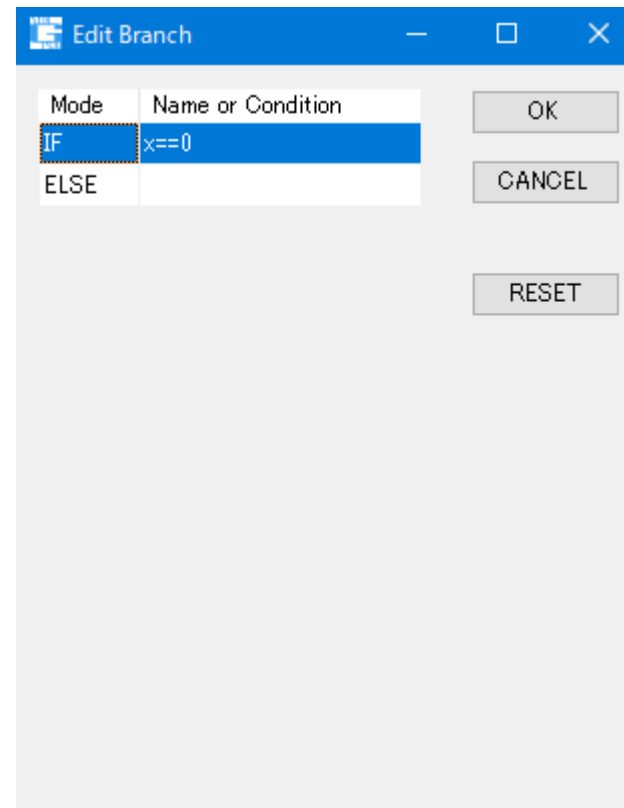
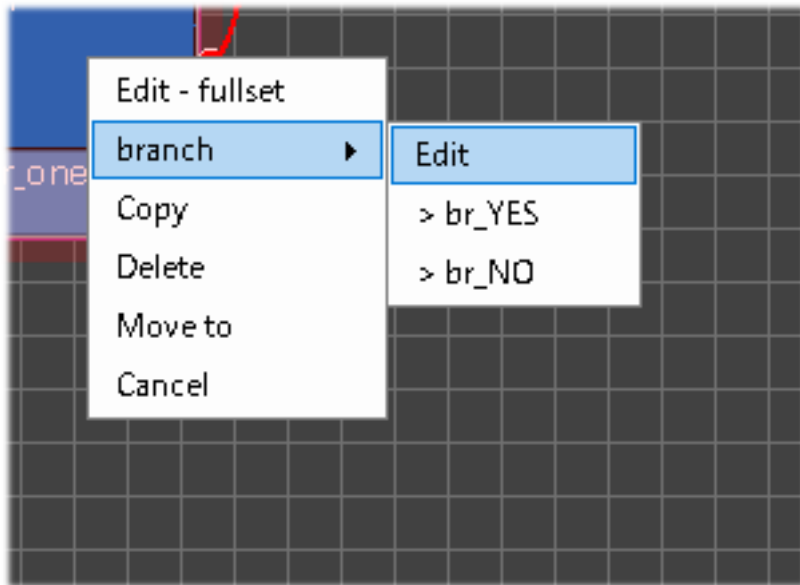
init項目に x = UnityEngine.Random.Range(0,2);

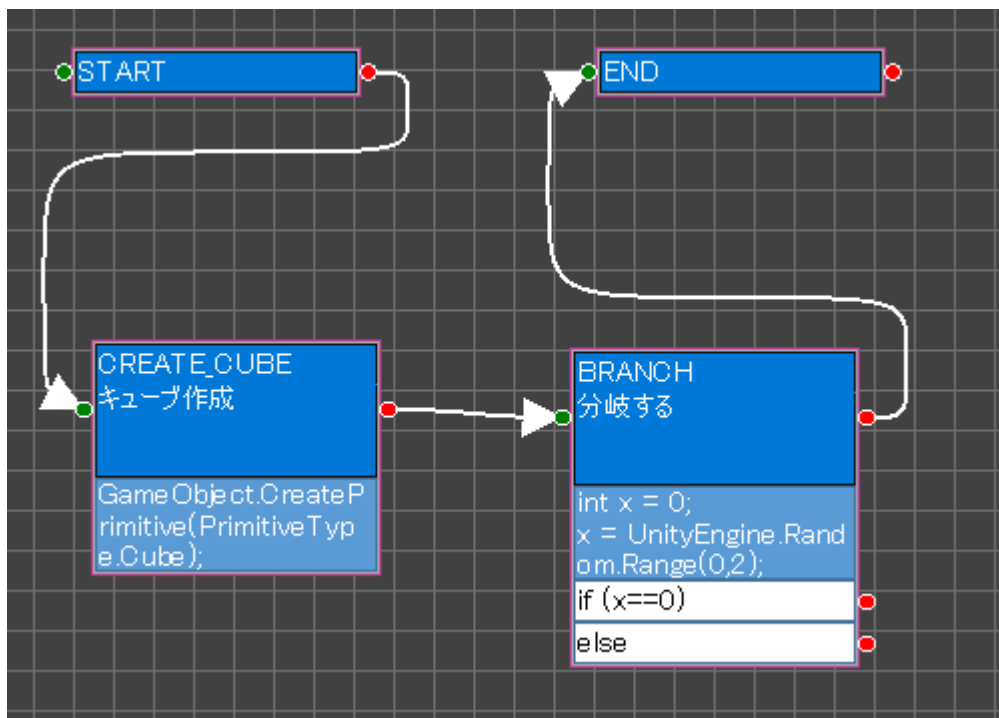
を設定する。

Step 4

branch欄の記入

Branch欄に、下図のように分岐のコードを記入。



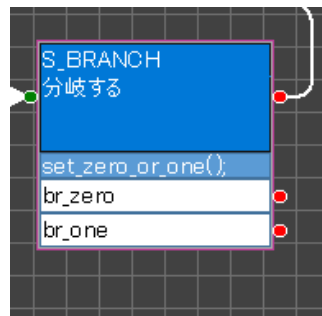


状態図に戻ると左図のようになる。

Step 5

矢印の削除

分岐用の矢印のみにしたいので、現状の矢印を削除する。



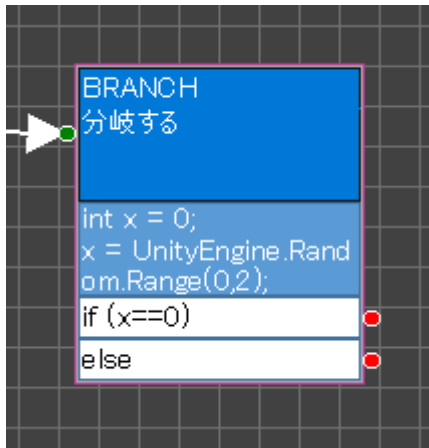
この赤点をダブルクリック。

The screenshot shows a dialog box titled 'Select Next State'. It has a text input field at the top containing 'S_END'. Below the input field is a table with three columns: 'Path', 'State', and 'Comment'. The table contains four rows of data. To the right of the table are three buttons: 'OK', 'CANCEL', and 'CLEAR'. The 'CLEAR' button is highlighted with a red rectangle.

Path	State	Comment
/	S_START	
/	S_END	
/	S_CREATE_CUBE	キューブ作成
/	S_BRANCH	分岐する

行先選択用のダイアログ表示される。

ClearしてOK

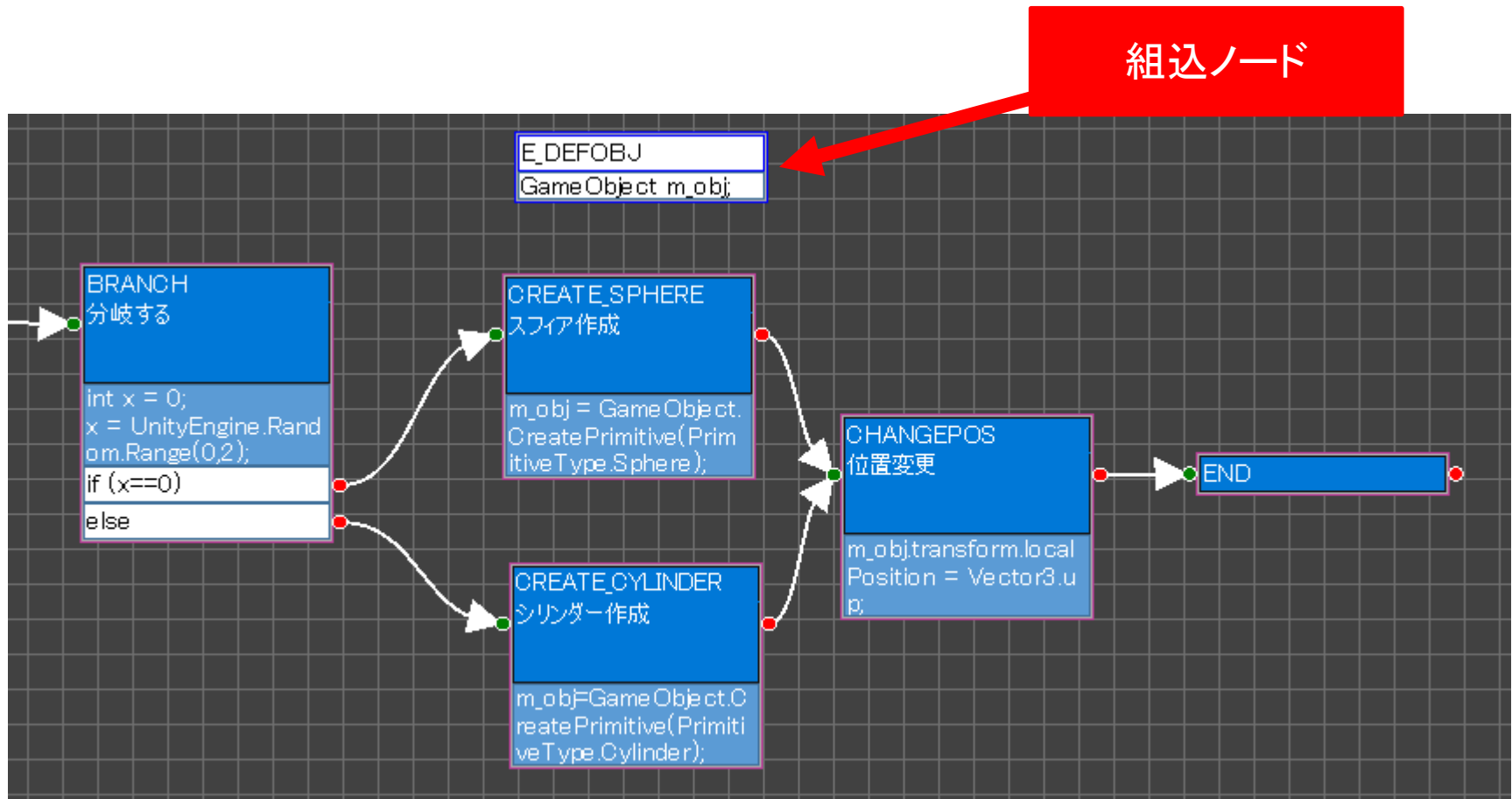


ここにあった赤点が矢印と一緒に消える。

Step 6

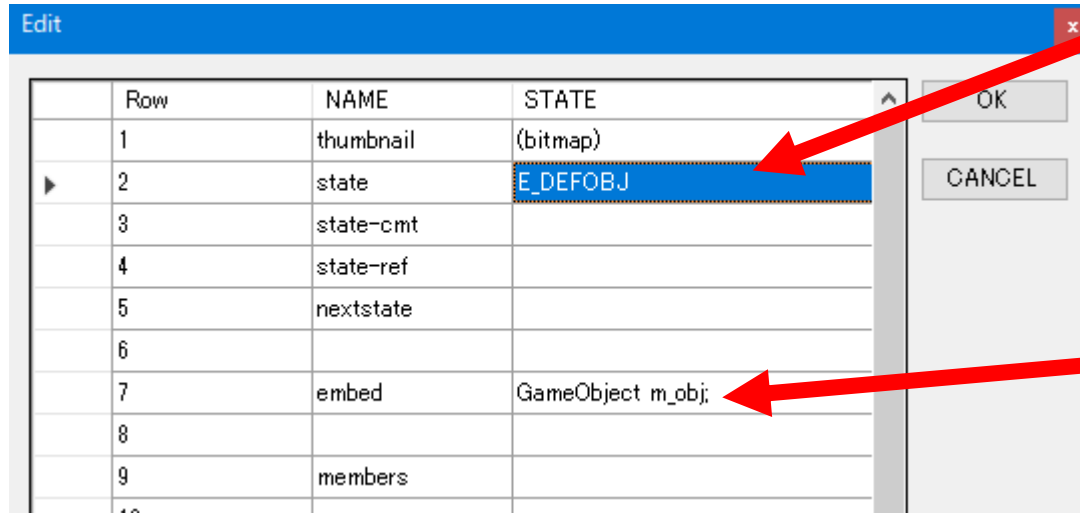
スフィア（球）とシリンダー生成ステート作成

スフィアまたはシリンダーを生成し、m_objに格納する。
m_objは 組込ノード(E_DEFOBJ)で定義。



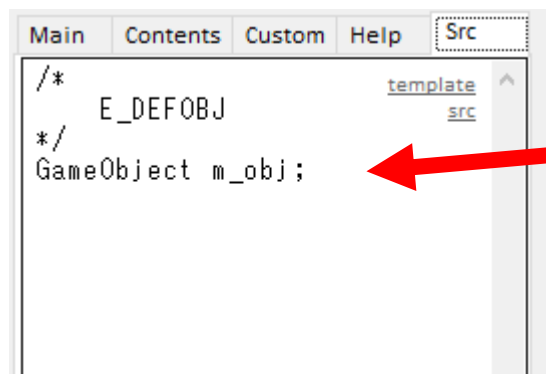
組込ノード作成

ステート名の先頭をE_とする。



Row	NAME	STATE
1	thumbnail	(bitmap)
2	state	E_DEF OBJ
3	state-cmt	
4	state-ref	
5	nextstate	
6		
7	embed	GameObject m_obj;
8		
9	members	

embedにコードを書く。



```
/*  
    E_DEF OBJ  
*/  
GameObject m_obj;
```

変換結果確認

スフィアとシリンダー

スフィア

```
m_obj = GameObject.CreatePrimitive(PrimitiveType.Sphere);
```

それぞれの
init項に入力

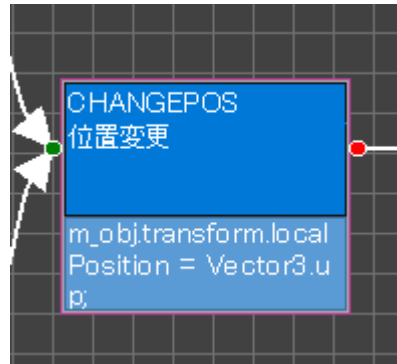
シリンダー

```
m_obj=GameObject.CreatePrimitive(PrimitiveType.Cylinder);
```

本ツールでは、入力補助がないため長い命令の場合にタイプミスが発生しやすい。
その場合・・

- 方法1) 入力補助機能付きエディタで同クラス内に潔な名前のAPIを作成して、呼び出す。
- 方法2) 一度、入力補助機能付きエディタで確認後、コピーする。

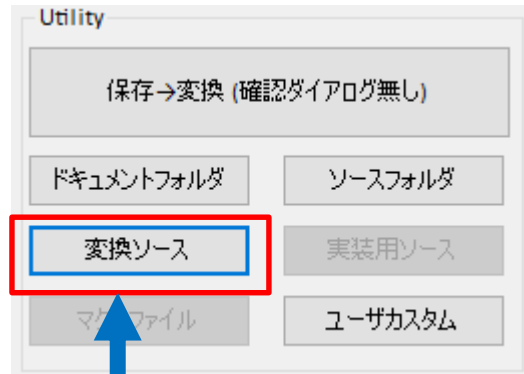
位置変更



```
m_obj.transform.localPosition = Vector3.up;
```

init項に入力

補足：ソースの編集について



ソース
表示

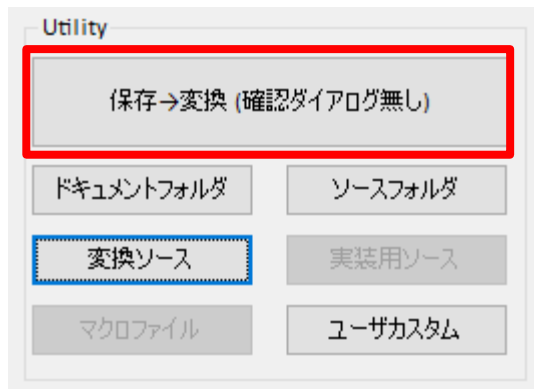
変換ソースへメソッド、メンバ変数等の追加が可能。

但し、変換結果挿入部分は除く。

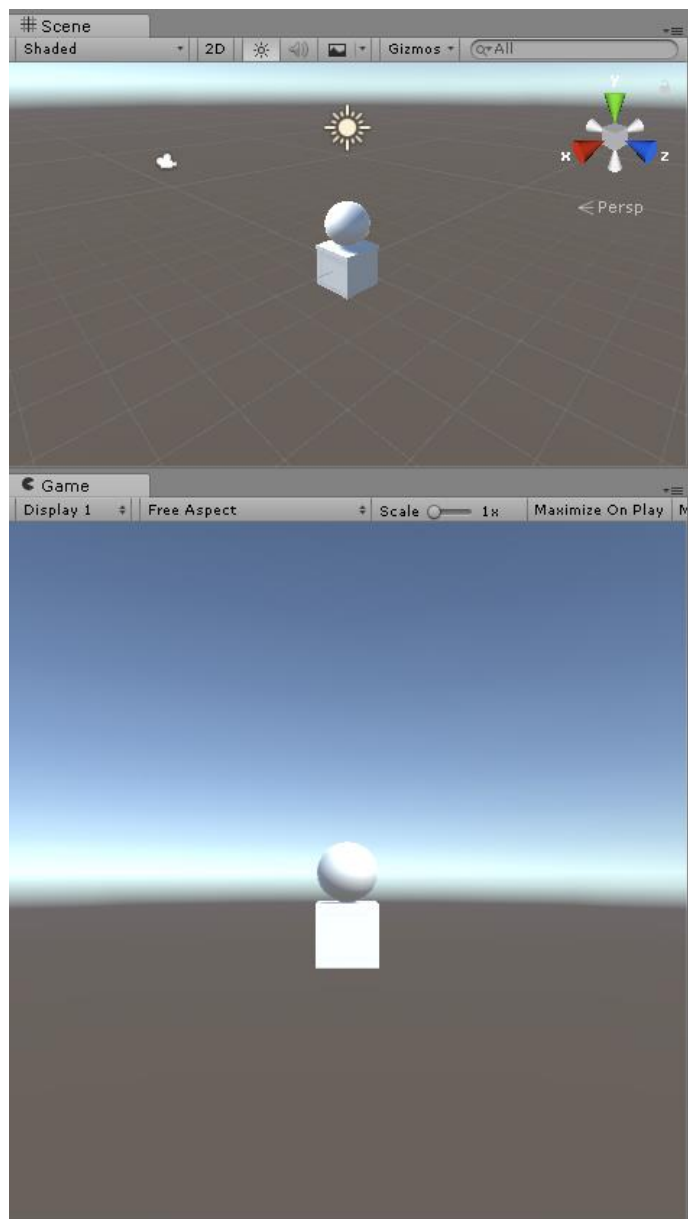
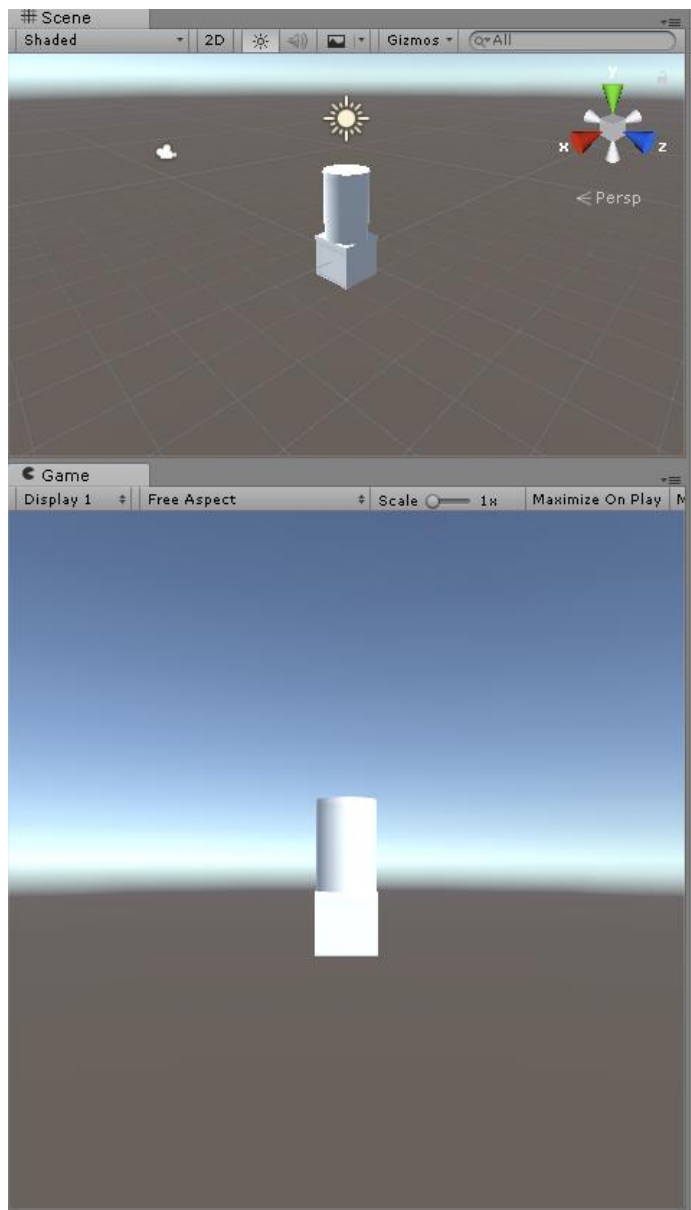
```
// [SYN-G-GEN OUTPUT START]
:
:
// [SYN-G-GEN OUTPUT END]
```

Step 7

変換と実行



ユーティリティパネルの【保存→変換】ボタンを押す。



実行の度、
どちらかになる。

まとめ

1. branch欄にif、および、else を記入するとステートボックスに分岐点が追加される
2. ステート出口の赤点をダブルクリックするとステート選択画面となり、ステート名を削除すると矢印と赤点が消える
3. 追加した関数を実装する
4. 実行する