

# 3Dモデルを動かす

---



このコンテンツは、『[ユニティちゃんライセンス](#)』で提供されています。

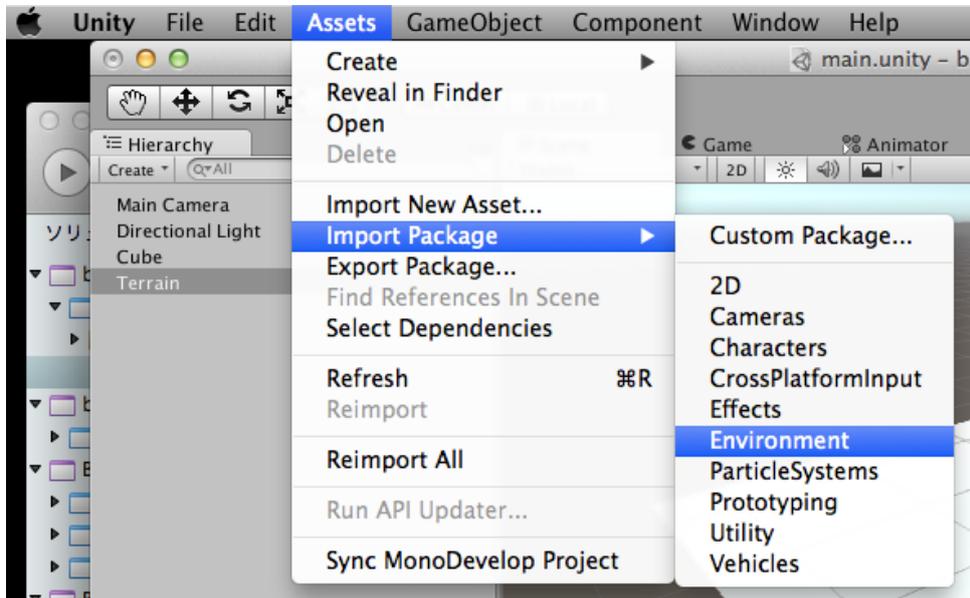
---

ZERONE studio

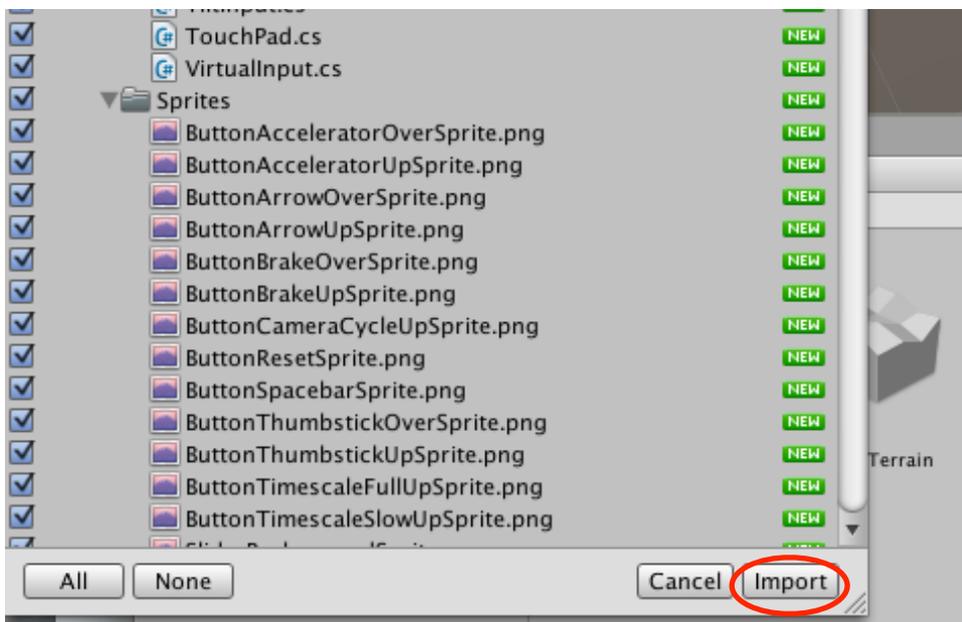
---

☆準備をします。

①一番上のAssetをクリック。Import Packageを選択して、Environmentをクリック。

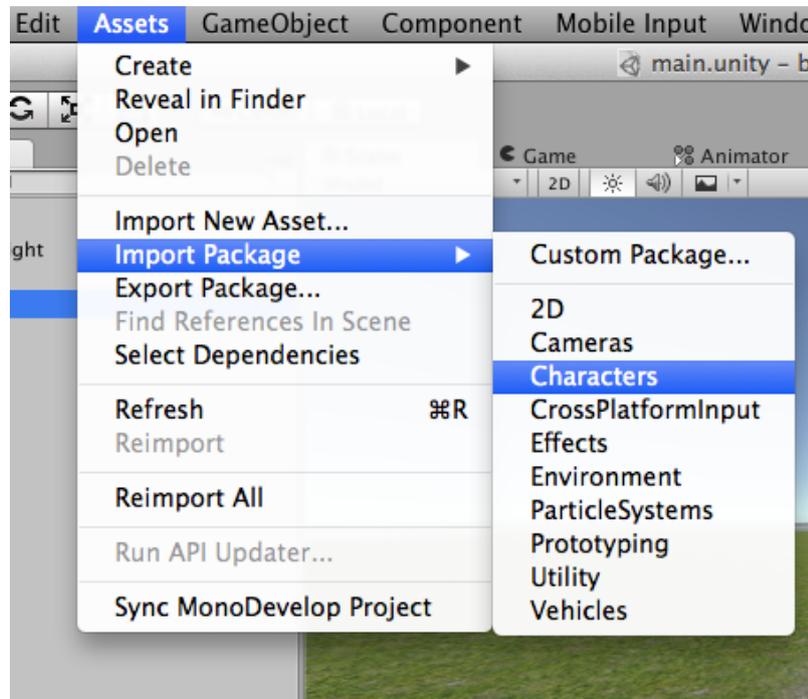


出て来たウィンドウのImportをクリック。



ProjectビューのAssetsの中に、Standard Assetsが生成される。

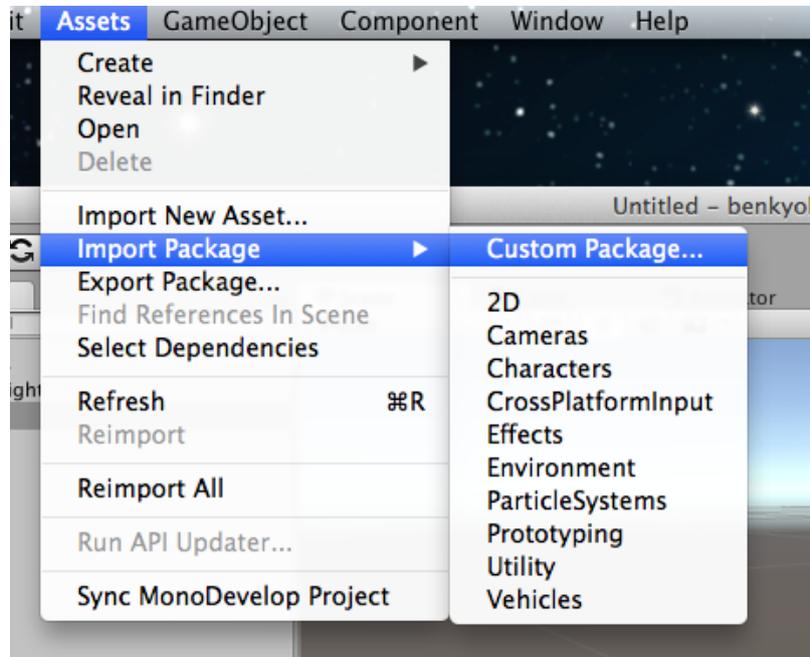
②上のメニューのAssetsをクリック。Import Packageを選択して、Charactersをクリック。



出て来たウインドウの右下のImportをクリック。

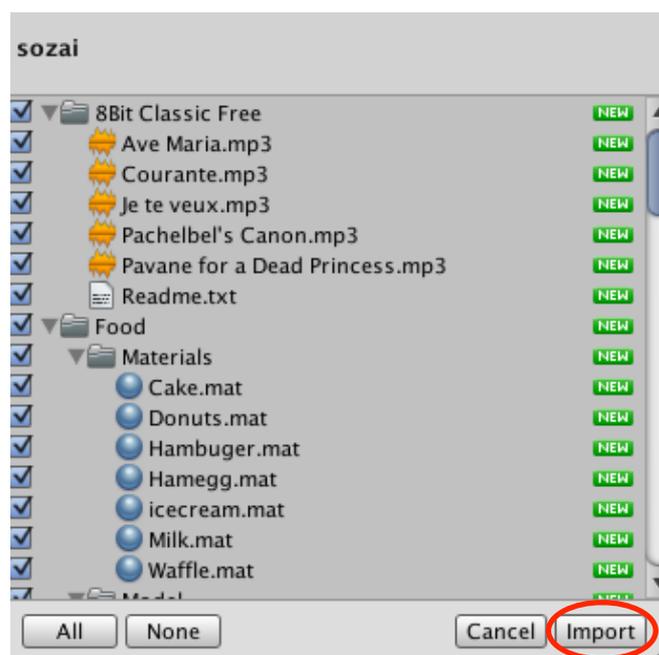
## ☆準備をします（2）

①上のメニューのAssetsをクリックします。Import Package..を選択し、Custom Package..をクリックします。



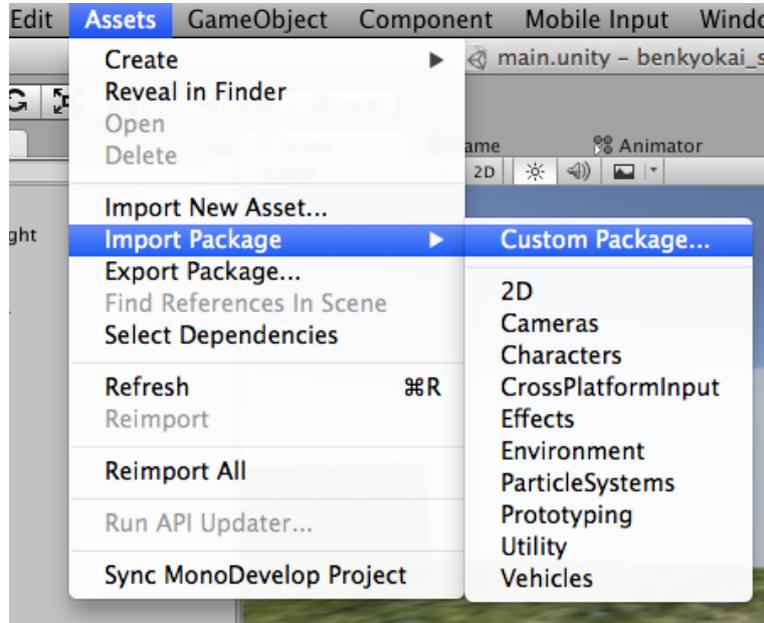
②デスクトップにあるsozai.unitypackageをopenします。

③出て来たウインドウの右下のImportをクリックします。



### ☆準備をします（3）

①上のメニューのAssetsをクリック。Import Packageを選択し、Custom Package..をクリック。



②デスクトップにあるUnityChan.unitypackageを選択し、Openをクリック。

③出て来たウインドウの右下のImportをクリックする。



このコンテンツは、『[ユニティちゃんライセンス](#)』で提供されています。

## ☆Unityの画面説明

### ①シーンビュー(Scene View)

ここにゲームの世界を作ります。

### ②ヒエラルキービュー(Hierarchy View)

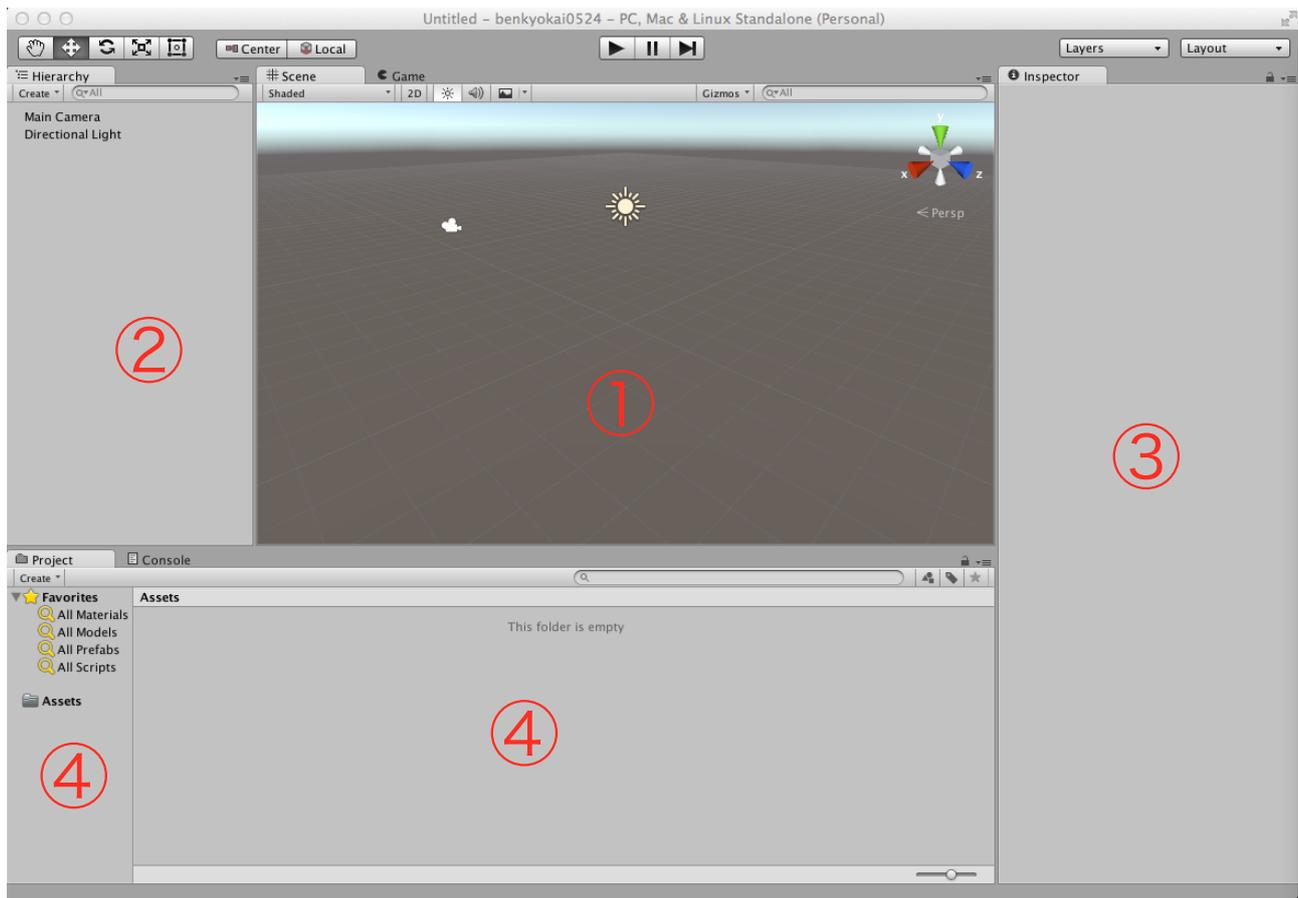
①に置いてあるもののリストが表示されます。

### ③インスペクタービュー(Inspector View)

①に置いてあるものを選択すると、詳細な情報が表示されます。

### ④プロジェクトビュー(Project View)

ゲームを作るときの材料の全部が表示されます。

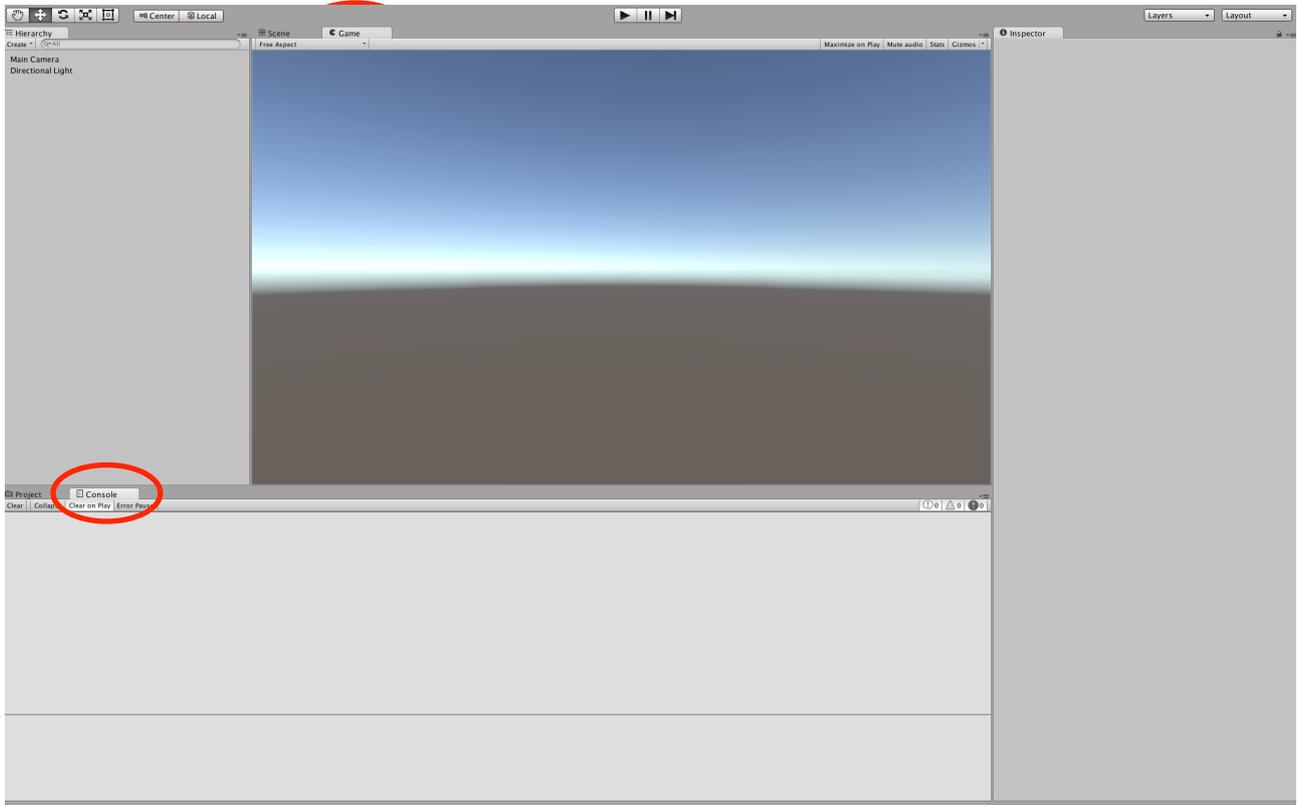


Sceneビューの上にあるタブで、Gameに切り替えます。

### ⑤ゲームビュー(Game View)

カメラに映っているものが表示されます。

ゲーム画面に出てくるのが、この画像になります。



Projectビューの上にあるタブを、Consoleに切り替えます。

### ⑥コンソールビュー(Console View)

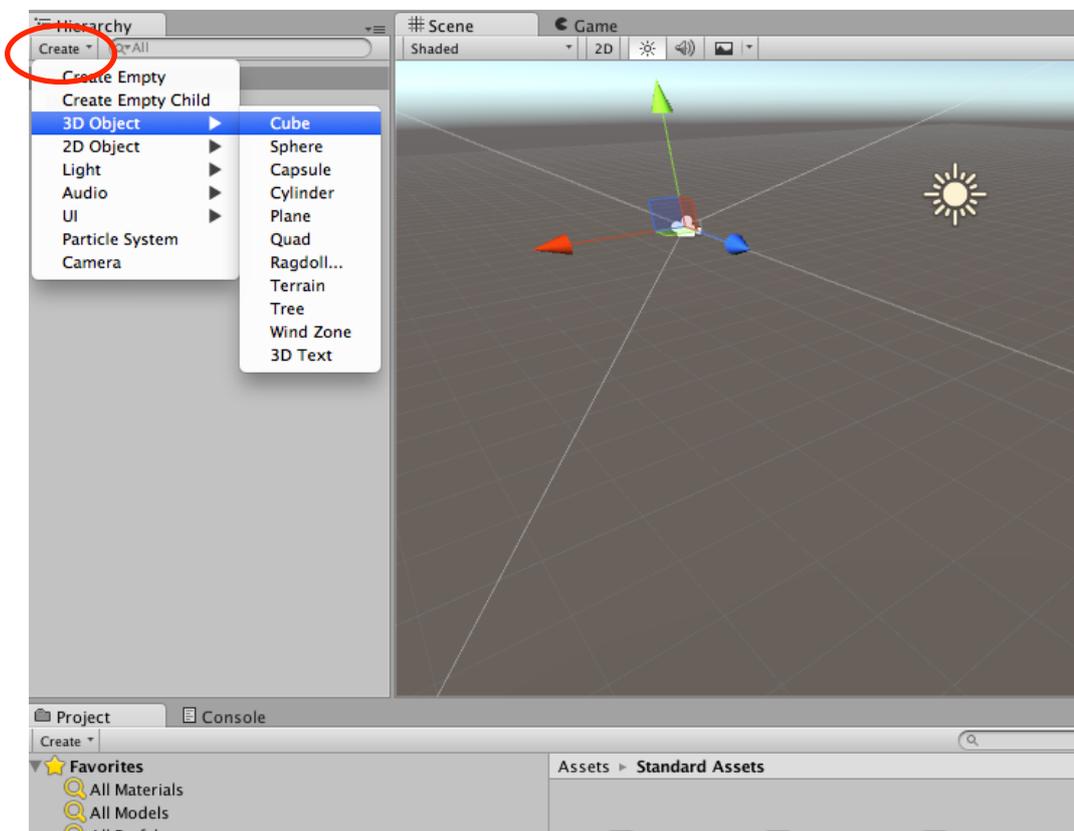
エラーなどが表示されます。

☆シーンをセーブします。

①Ctrl+Sでセーブします。最初だけdialogが出て来ますので、mainという名前でセーブしてください。unityが途中で落ちる(Hung up)ことがあるので、こまめに、Ctrl+Sでセーブしてください。

☆Cubeを作ります。

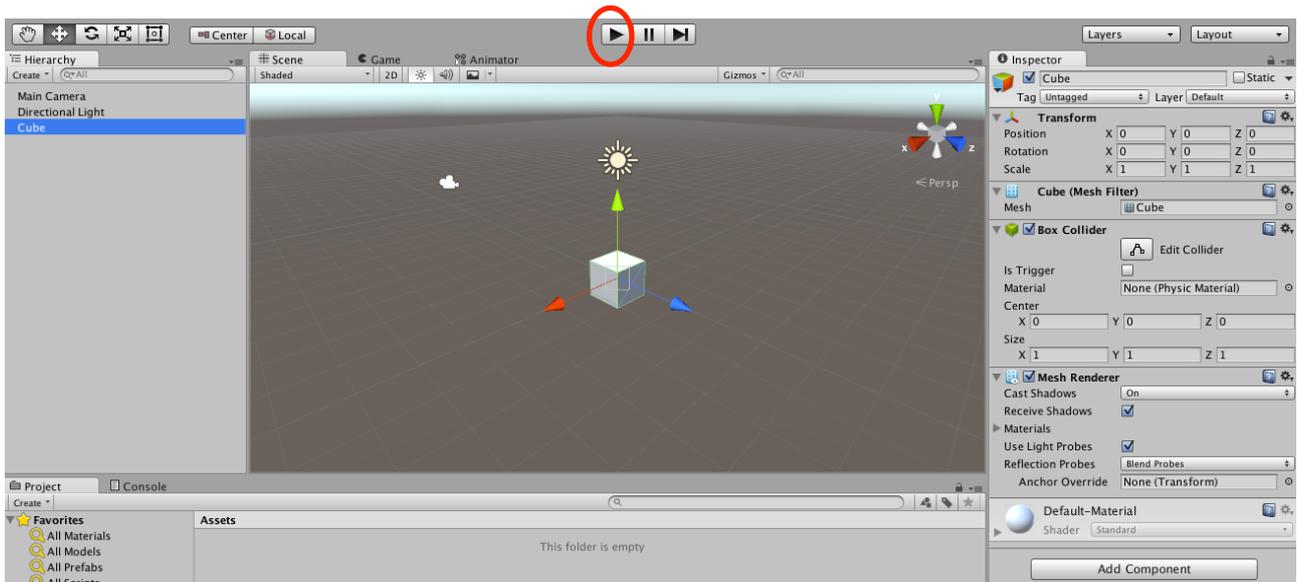
①Hierarchyビューの上にあるCreateをクリックします。  
3D Object の中のCubeを選択します。



Cubeが出来ましたか？

下のようになりましたか？

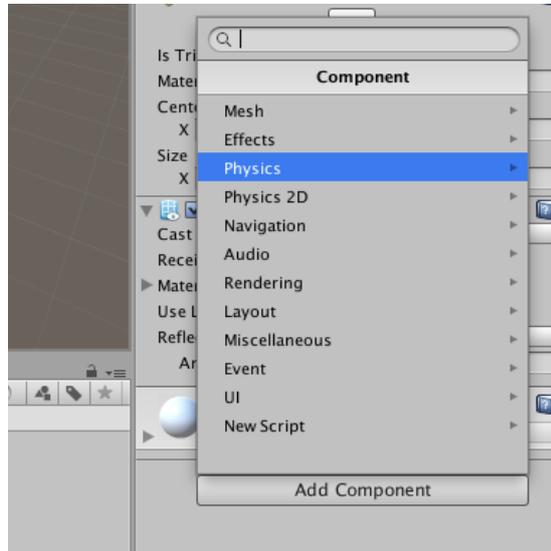
それでは、再生ボタンを押してみましょ。再生を止めるときは、もう一度再生ボタンを押します。



- ①Cubeの名前が、Inspectorビューにあることを確認します。
- ②Cubeの位置を変更します。Inspectorビューの値も変わります。
- ③Cubeを回転させます。Inspectorビューの値も変わります。
- ④Cubeを引き延ばします。Inspectorビューの値も変わります。

⑤Box Colliderがキューブについています。Colliderは、衝突装置という意味です。Box Colliderは、Cubeというオブジェクトに着いているコンポーネント(Component)です。

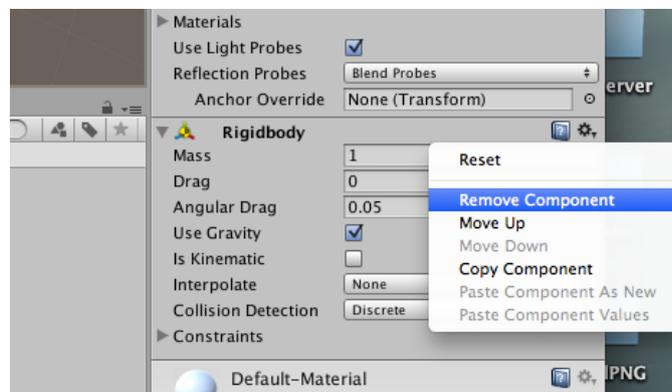
- ⑥Rigidbodyを追加します。



Add Componentをクリックし、Physicsをクリック。  
そして、Rigidbodyをクリックします。

再生ボタンを押してみましよう。

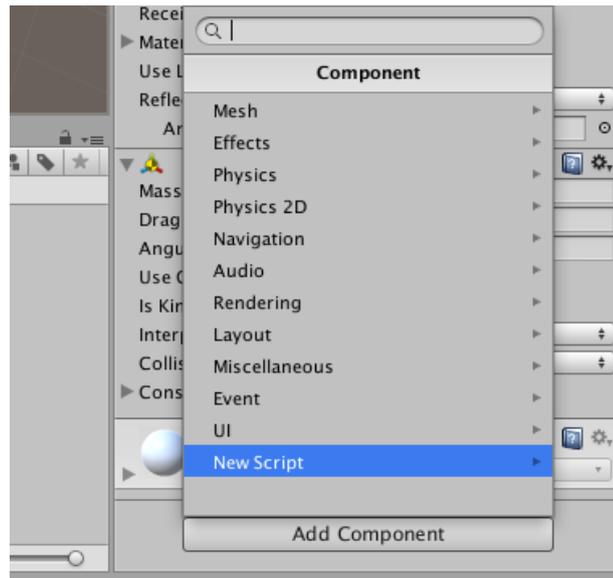
⑦Rigidbodyを取り外します。



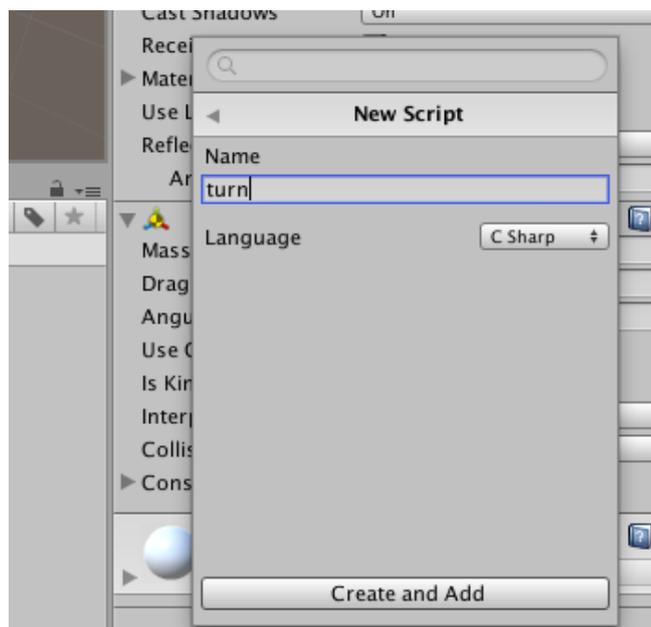
⑧スクリプトを追加します。

記述したプログラムをすぐに実行可能な言語を「スクリプト言語」といいます。ここでは、C#を使います。

Add Componentをクリックし、New Scriptをクリックします。



次に、名前をturn、言語をCsharpとして、Create and Addをクリックします。



## ☆プログラミングします

- ①turnという名前のスクリプトを、Scriptの中に移動しておきます。
- ②turnをダブルクリックします。monodevelopが開きます。

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class turn : MonoBehaviour {
5
6     // Use this for initialization
7     void Start () {
8
9     }
10
11    // Update is called once per frame
12    void Update () {
13
14    }
15 }
16
```

- ③次のように、書き込んでください。書き終わったら、プログラムを保存します。unityのSceneのセーブと同様、Ctrl+Sで保存します。

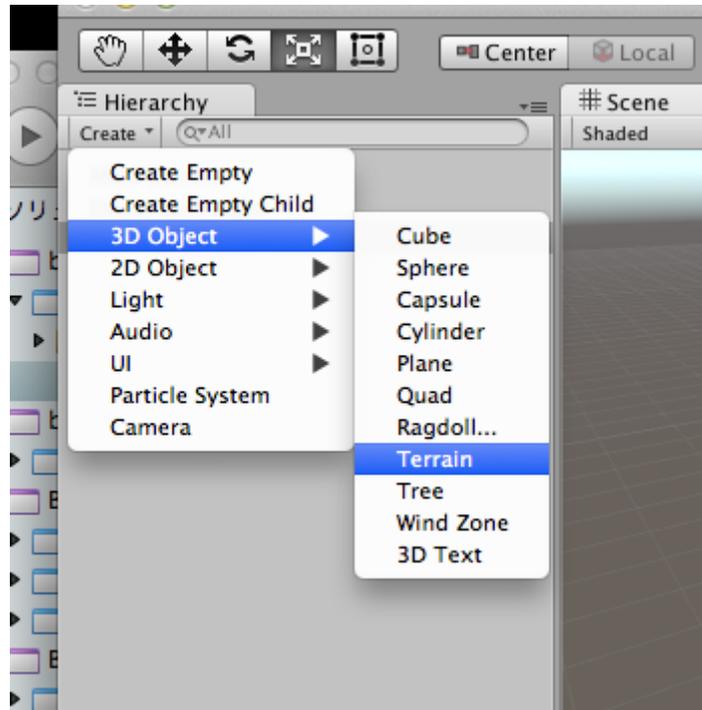
```
1 using UnityEngine;
2 using System.Collections;
3
4 public class turn : MonoBehaviour {
5
6     // Use this for initialization
7     void Start () {
8
9     }
10
11    // Update is called once per frame
12    void Update () {
13        this.transform.Rotate(0f, 1f, 0f);
14    }
15 }
16
```

## 練習問題

- (1)回転速度を速くしてみましょう。
- (2)X軸周りに回転させてみましょう。
- (3)Z軸周りに回転させてみましょう。

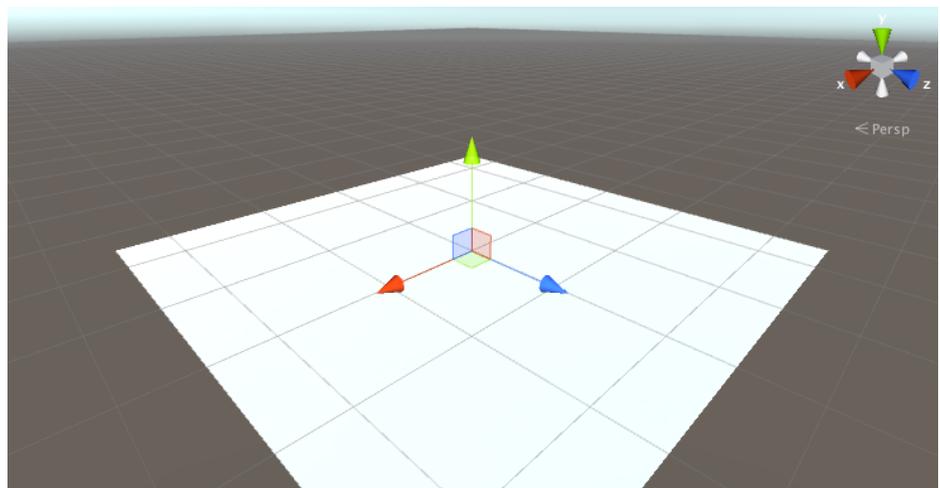
## ☆大地を作る

- ① HierarchyビューのCreateをクリック。  
3D Objectを選択して、Terrainをクリック。



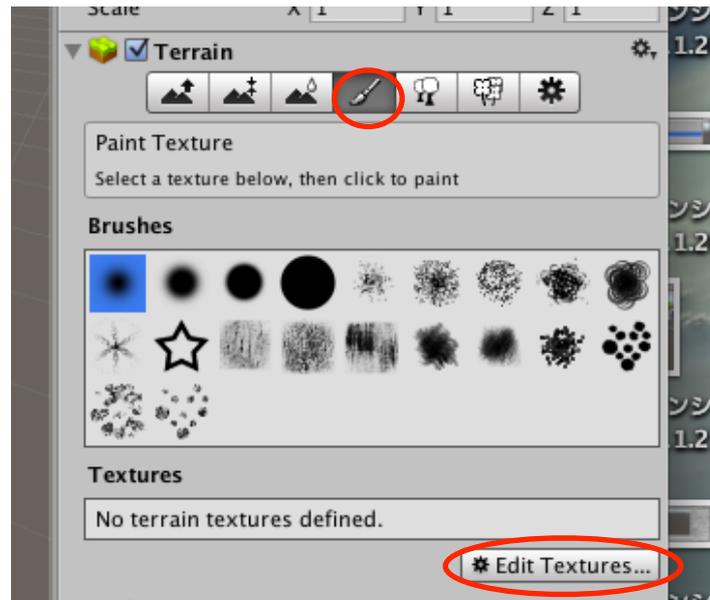
Terrainが出来ましたか？

- ② CubeがTerrainの真ん中に来るように、調整します。  
Terrainを選択して、Inspectorビューを表示します。  
positionを、X : -250 Y : 0 Z : -250にします。  
下のようになります。大きさを調整してください。

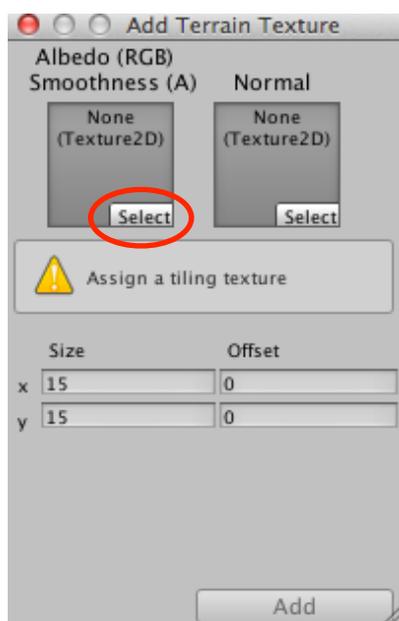


### ③Terrainに色をつけます。

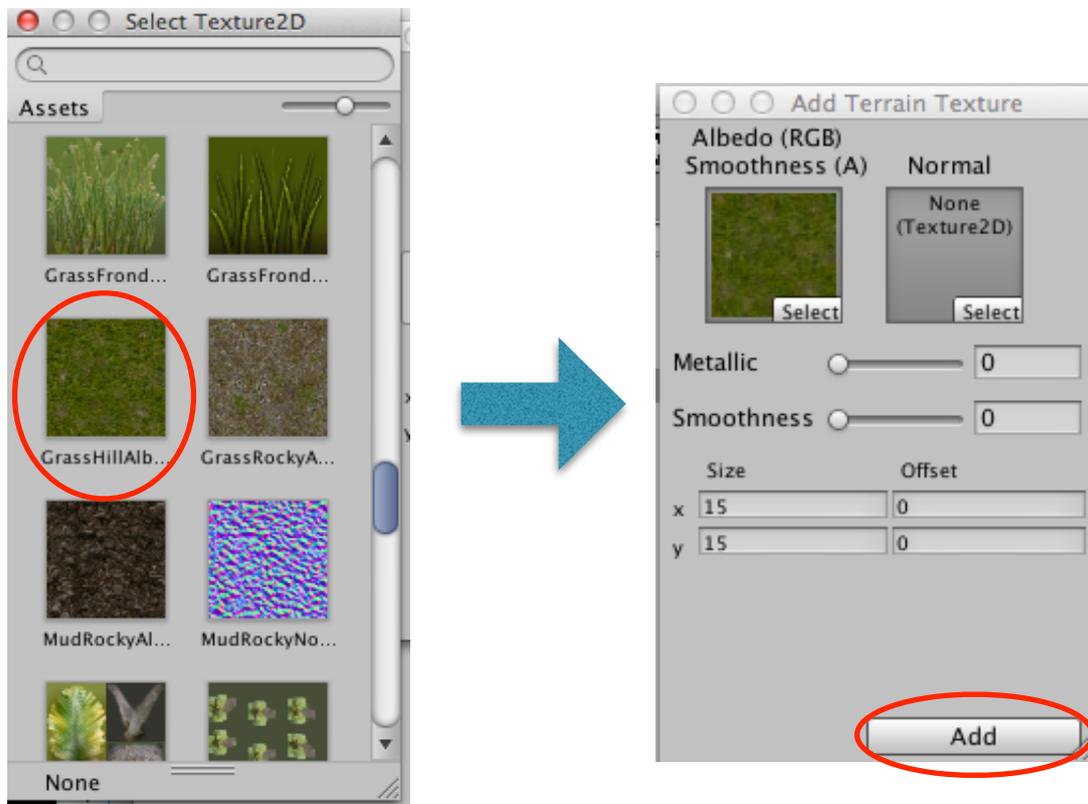
HierarchyビューでTerrainを選択し、Inspectorビューの筆マークをクリック。その後、Edit Textureをクリック。Add Textureをクリックします。



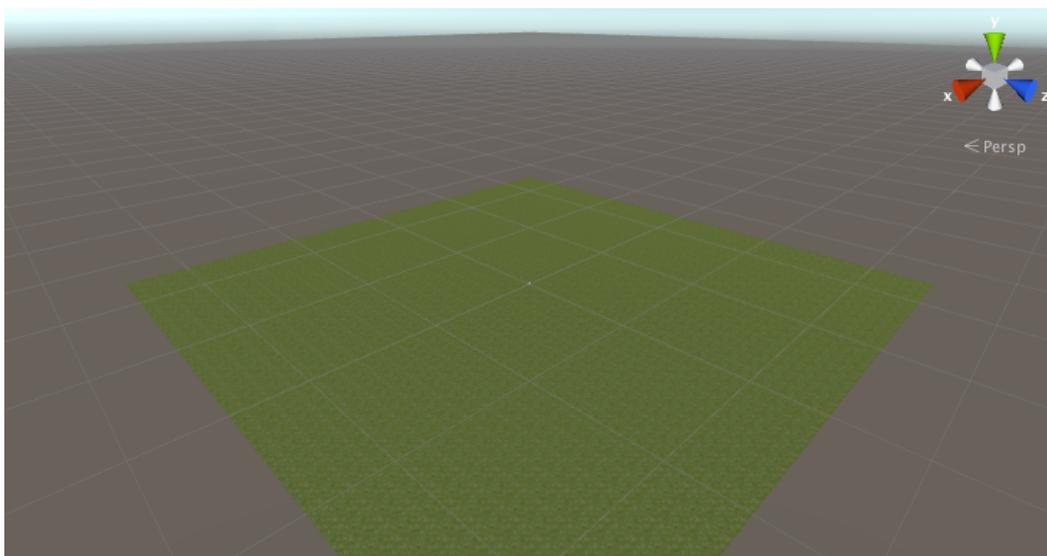
出て来た小さなウィンドウの左側の枠下のSelectをクリック。



テクスチャの中から、GrassHillAlb..をダブルクリック。  
(または、クリックして選択した状態でウインドウを消す)  
最初のウインドウのAddをクリック。



Terrainに色が着いたことを確かめます。

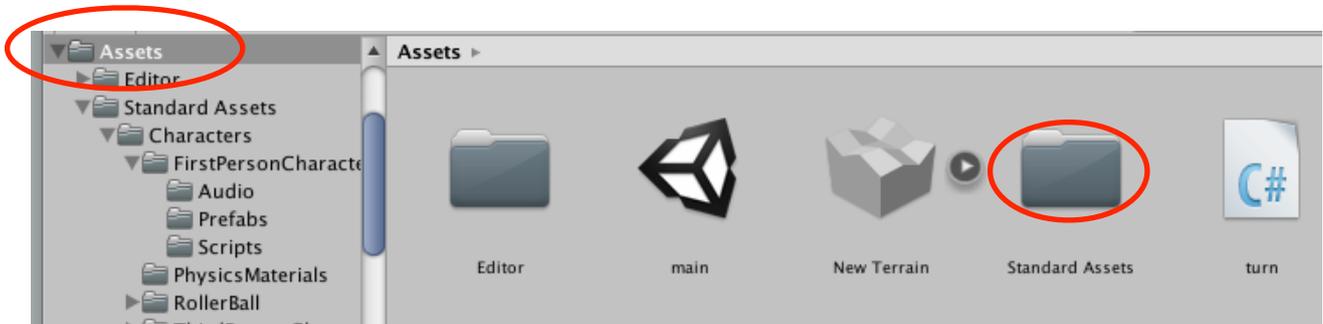


## 練習問題

(1)同じようにして、GrassRockyA..を追加してください。岩肌部分を塗ってみましょう。

☆FirstPersonControllerを追加します。

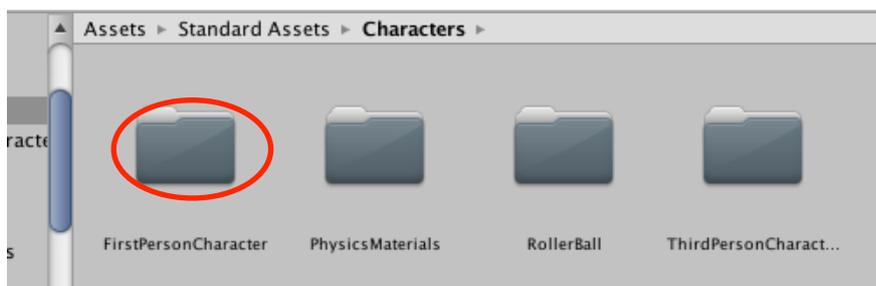
①ProjectビューのAssetsをクリック。Standard Assetsをダブルクリック。



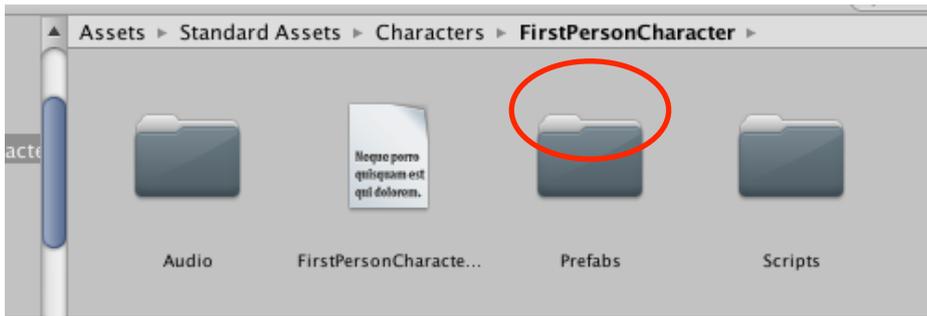
②Charactersをダブルクリック。



③FirstPersonCharacterをダブルクリック。

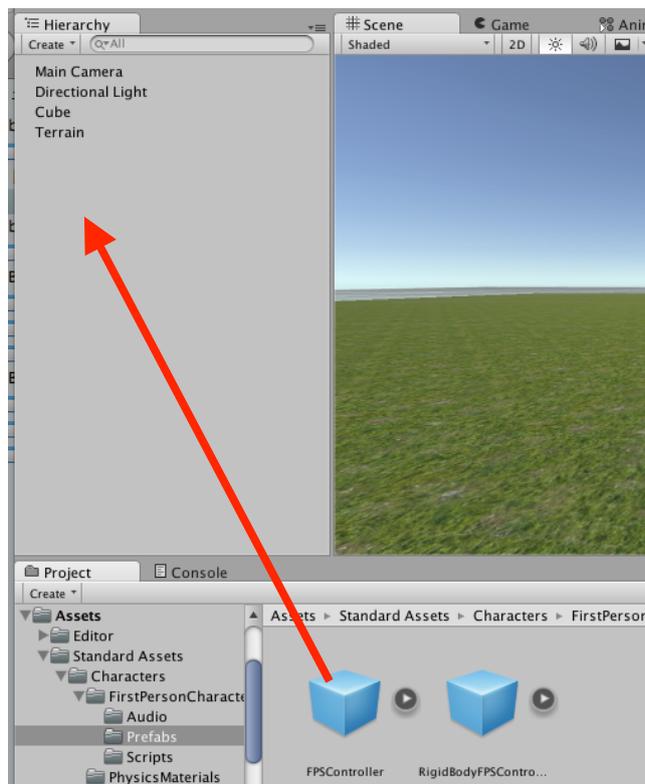


④Prefabsをダブルクリック。



## ⑤FPSController

をHierarchyビューへ移動します。



⑥Main Cameraを選択して、InspectorビューのAudio Listenerのチェックを外してください。

⑦再生ボタンを押して動かします。

Cubeを探しましょう。

## 練習問題

(1)Cubeが埋まっているので、少し上に移動させましょう。

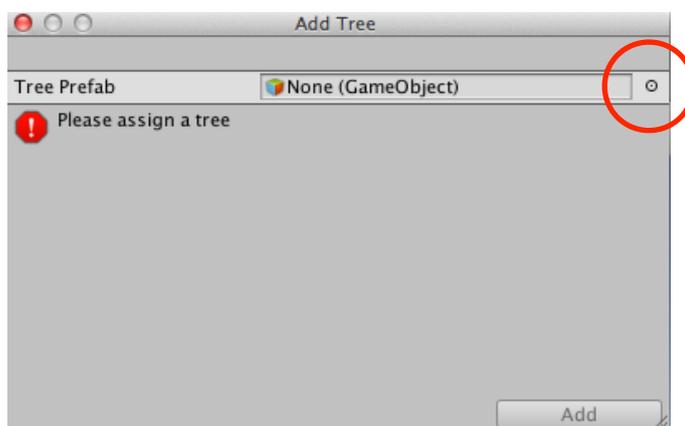
☆大地に木を植えます。

①HierarchyビューでTerrainをクリックし、Inspectorビューの木のボタンを押します。

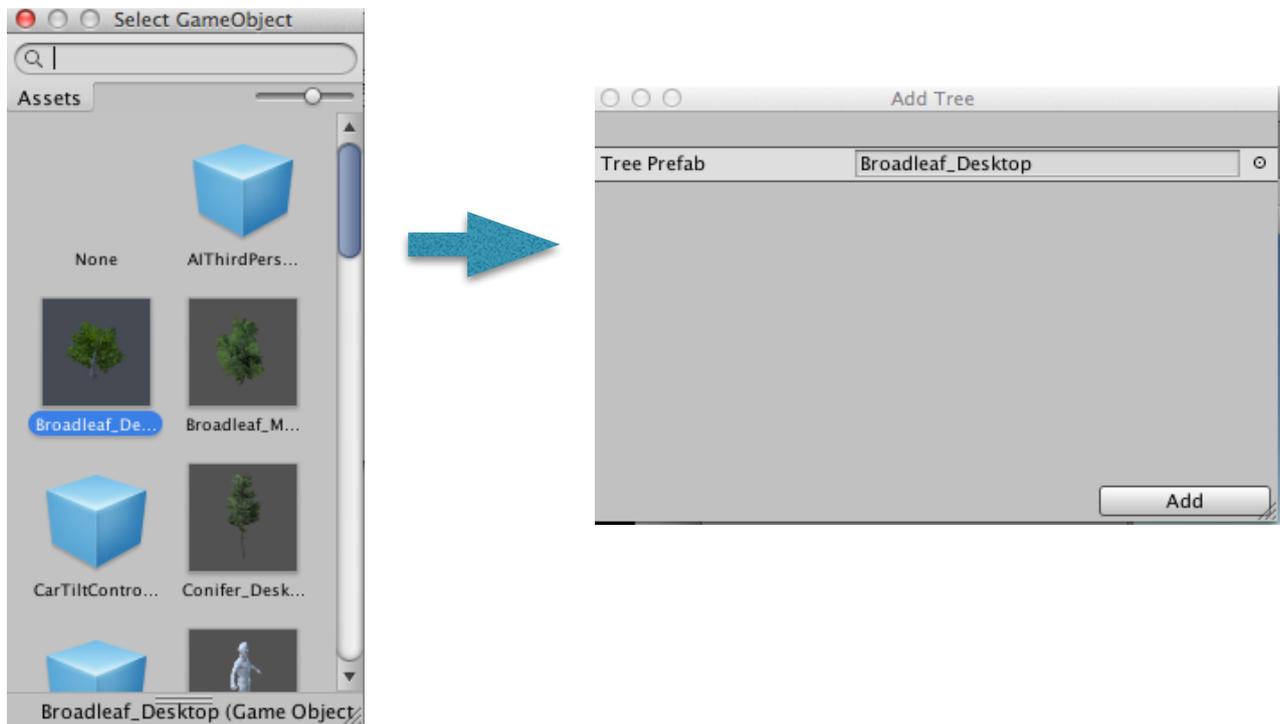
②Edit Trees..をクリックし、Add Treeをクリック。



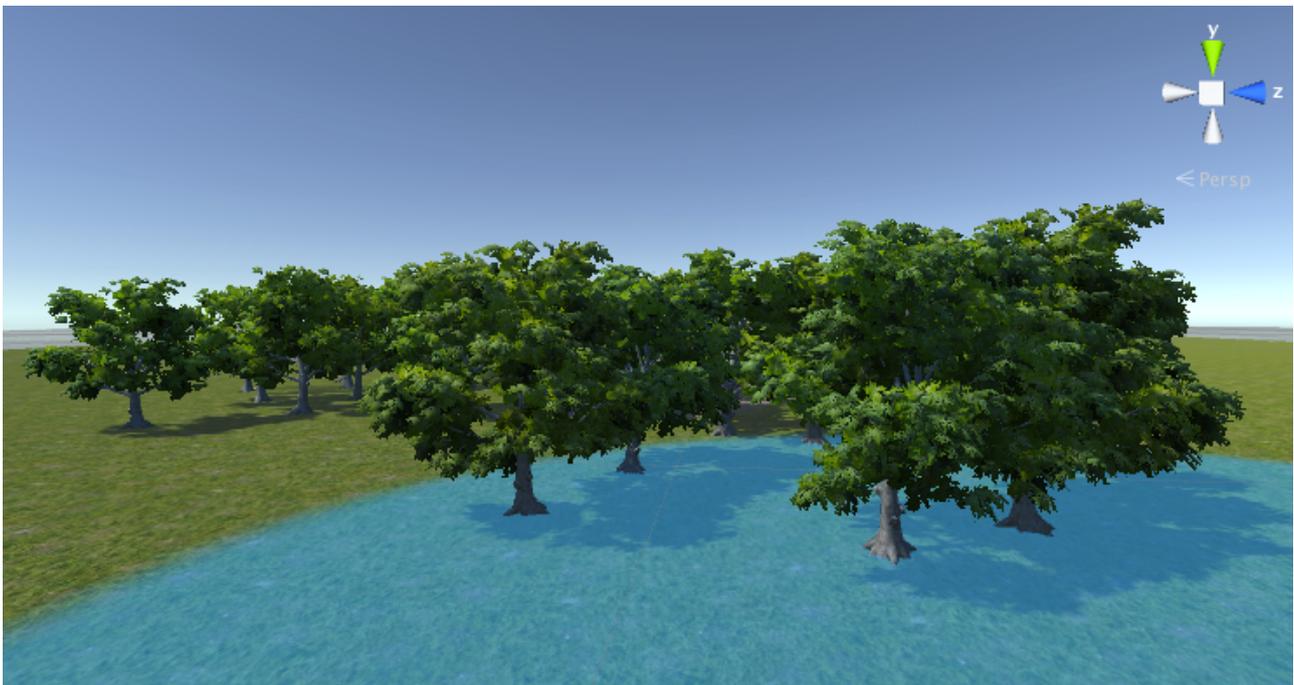
③ウインドウの小さな丸をクリックします。



④好きな木をダブルクリックします。次に、最初に出て来たウインドウのAddをクリックします。



⑤木を植えます。  
SceneビューのTerrainをクリックしてください。



⑥では、再生ボタンを押してみましょう。

植えすぎたときは、shift+クリックで木を消せます。

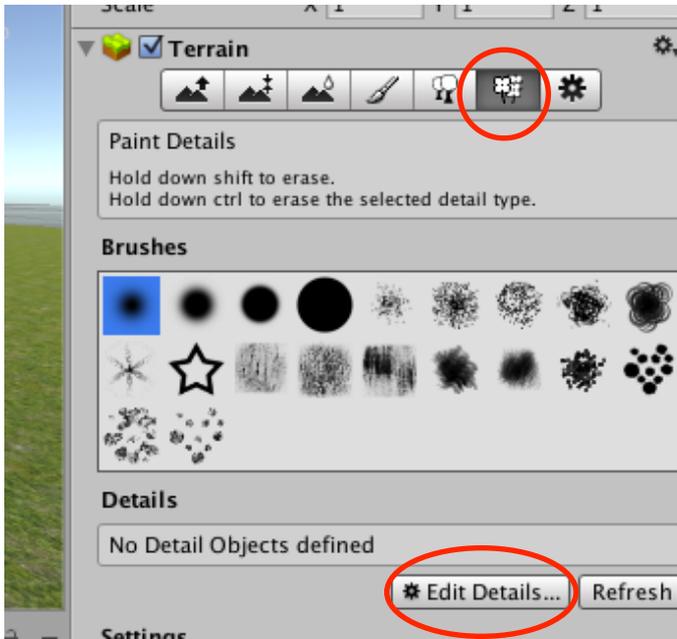
### 練習問題

(1)種類の違う木も追加して、木を植えましょう。

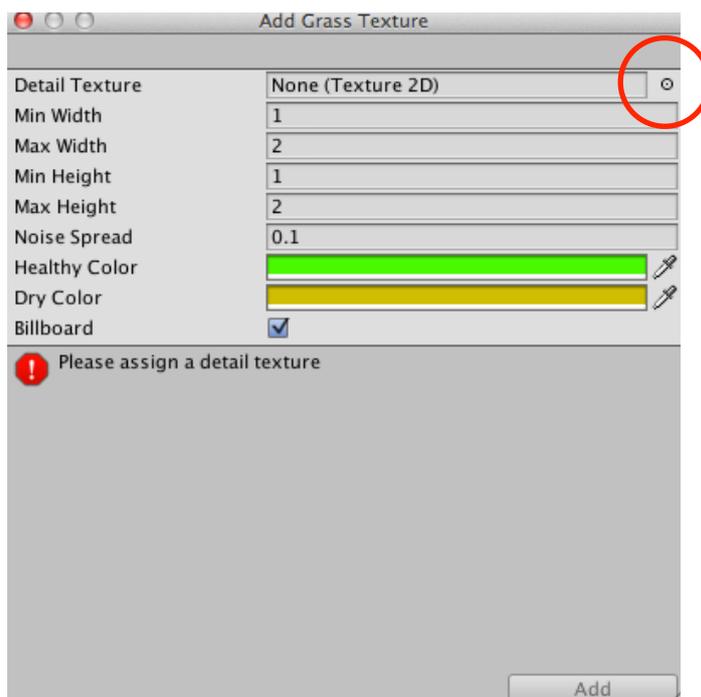
☆大地に草を植えます

①HierarchyビューでTerrainをクリックし、Inspectorビューの花のボタンを押します。

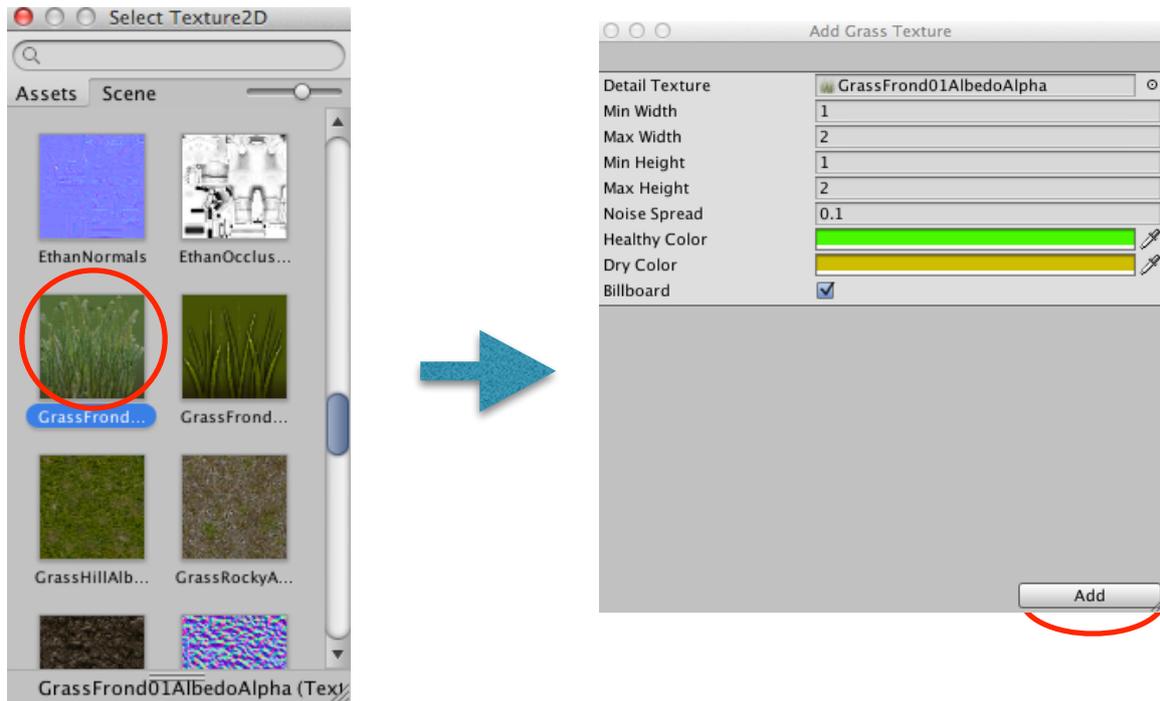
②Edit Details..をクリックし、Add Grass Textureをクリック。



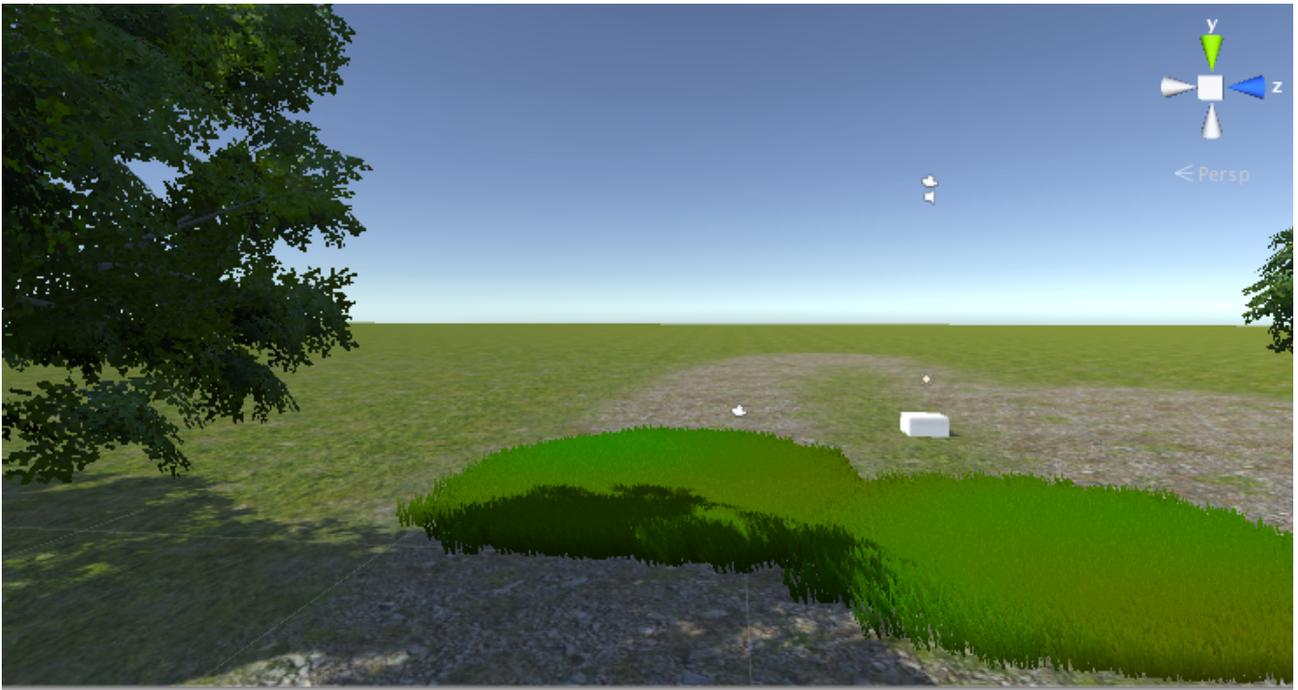
③ウインドウの小さな丸をクリックします。



④草をダブルクリックします。次に、最初に出て来たウィンドウのAddをクリックします。



⑤草を植えます。  
SceneビューのTerrainをクリックしてください。



⑥再生ボタンを押してみましょう。

植えすぎたときは、shift+クリックで草を消せます。

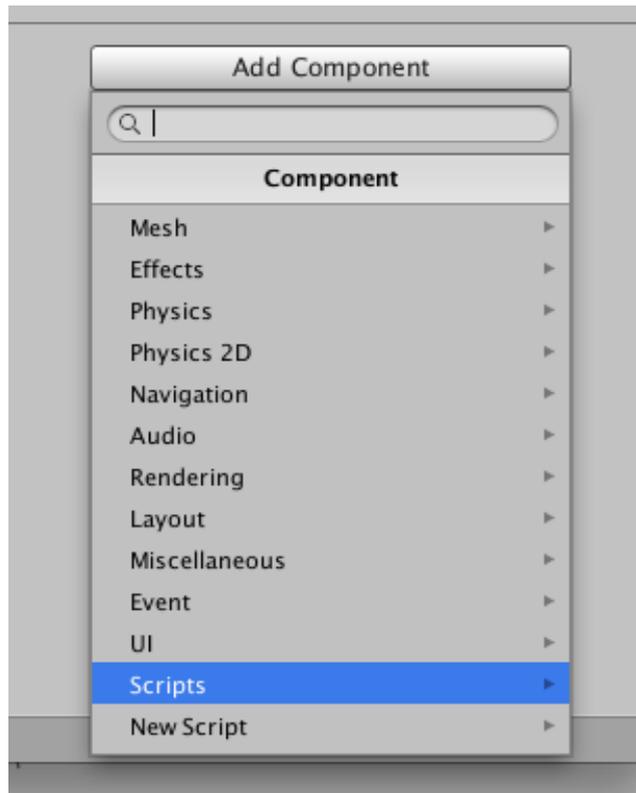
#### 練習問題

- (1)草が風になびいているのを確認しましょう。
- (2)太陽の位置を確認しましょう。

☆太陽を動かす。

①Directional Lightにスクリプトを追加します。

Hierarchyビューで、Directional Lightを選択する。  
Add Componentをクリック、Scriptsをクリック。



次に、Taiyoをクリック。

※予め記載されたスクリプトを用意しましたので、そのスクリプトを Directional Lightに追加します。

## ☆スクリプトの中身を見る

①taiyoをダブルクリックします。monodevelopが開きます。

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class taiyo : MonoBehaviour {
5
6     // Use this for initialization
7     void Start () {
8
9     }
10
11    // Update is called once per frame
12    void Update () {
13        this.transform.Rotate (0.1f, 0f, 0f);
14
15    //     if (Input.GetKeyDown ("m")) {
16    //         this.transform.eulerAngles = new Vector3(0f, 330f, 0f);
17    //     }
18    }
19 }
20

```

②Update関数に、this.transform.Rotate (0.1f, 0f, 0f);とあります。

Update関数が呼ばれる度に、X軸周りに0.1度ずつ回転します。

③再生ボタンを押して、太陽が動くことを確認します。

④もう一度、taiyoをダブルクリックして開きます。

スクリプトを分かりやすくするために、コメントを書きます。

コメント行は、行の最初に「//」を付けます。

簡単に付ける方法：

行のどこかにカーソルをおいて、Ctrl+/とします。もう一度、Ctrl+/とすると、元に戻ります。

- ⑤Update関数の中の if から3行分のコメントを外す。
- ⑥今度は、夜になっても、mキーを押すと、朝に戻ります。  
再生ボタンを押して確認しましょう。

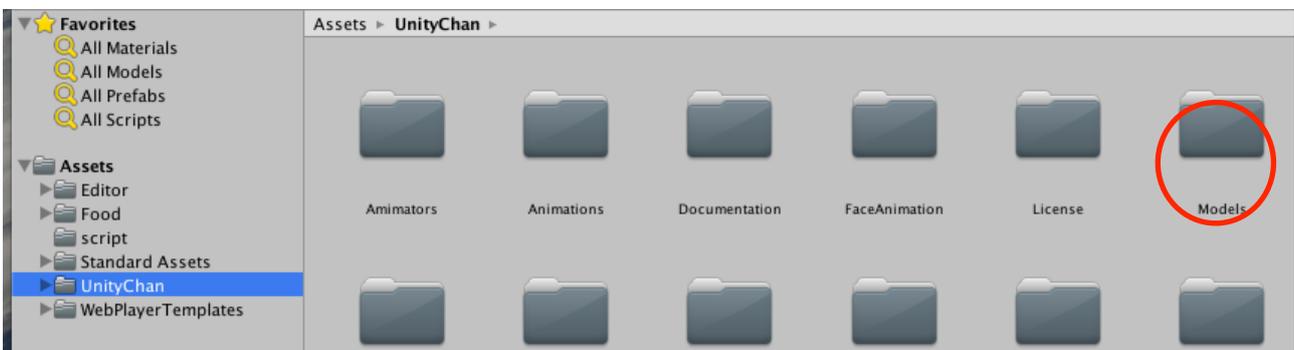
## 練習問題

- (1)太陽の速度を調節してみましょう。

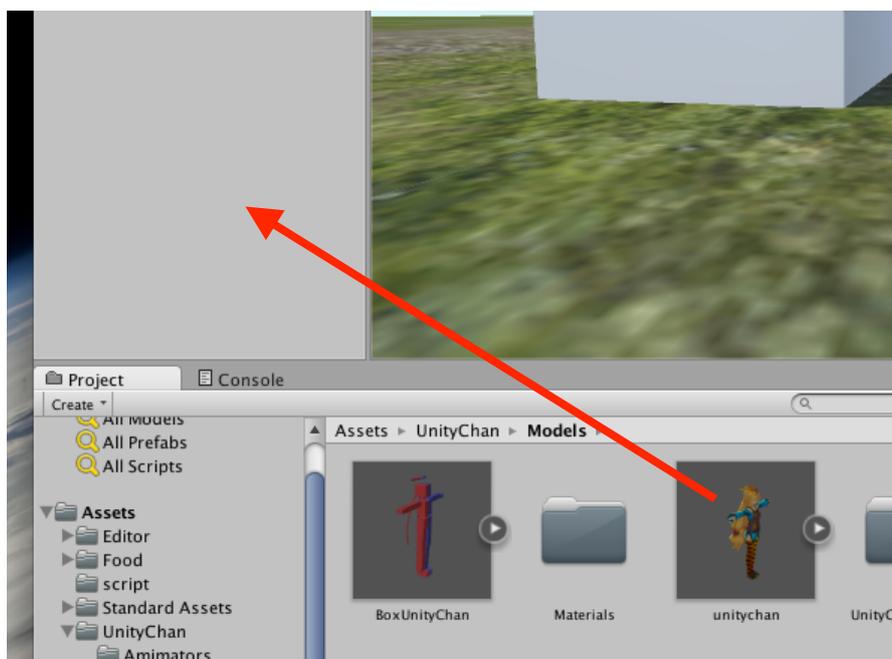
☆UnityChanを配置します。

①HierarchyビューのFPSControllerをDeleteします。FPSControllerをクリックします。そのあと、右クリックをして、Deleteを選択します。

②ProjectビューのAssetsの中のUnityChanをクリックします。次に、Modelsをダブルクリックします。



③unitychanをHierarchyビューへ移動します。



④CubeをDeleteします。

⑤unitychanのサイズを調整します。

Hierarchyビューでunitychanをクリックして、Inspectorビューのsizeを、X:2 Y:2 Z:2とします。

⑥再生ボタンを押してみましよう。

unitychanがカメラに映りましたか？

⑦カメラの位置を近づけます。

HierarchyビューでMain Cameraをクリックして、Inspectorビューのpositionを、X:0 Y:2 Z:-4とします。

⑧再生ボタンを押してみましよう。

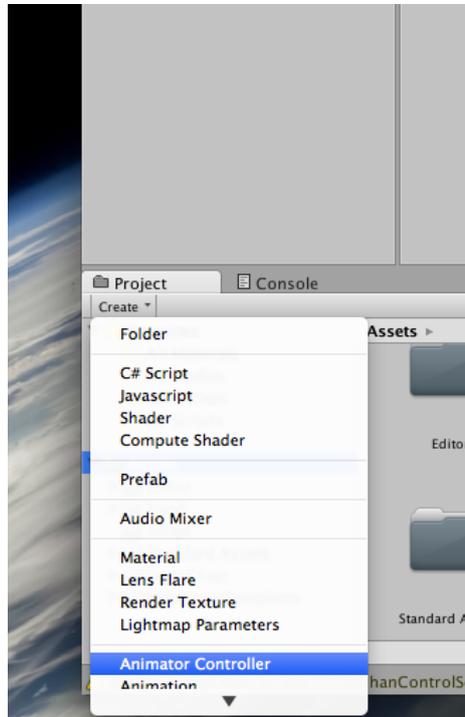
unitychanが大きくカメラに映りましたか？

### 練習問題

(1)unitychanを180度回転させて、カメラのほうを向くようにしましよ  
う。

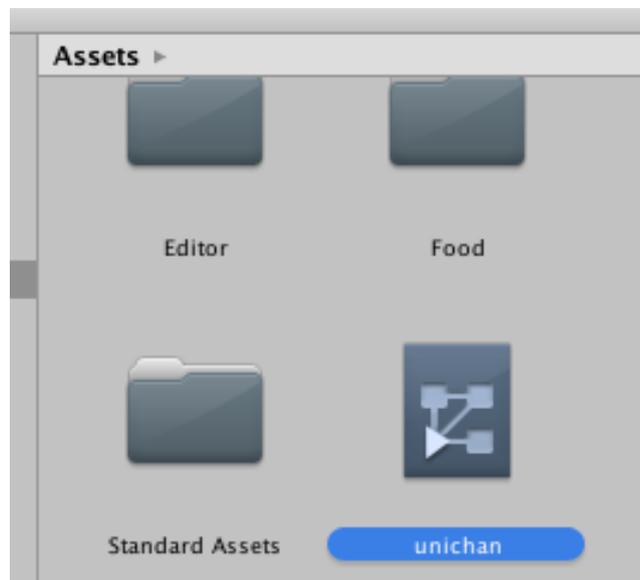
☆UnityChanのアニメーションを作ります

①ProjectビューのAssetsをクリックします。ProjectビューのCreateをクリックし、Animator Controllerをクリックします。

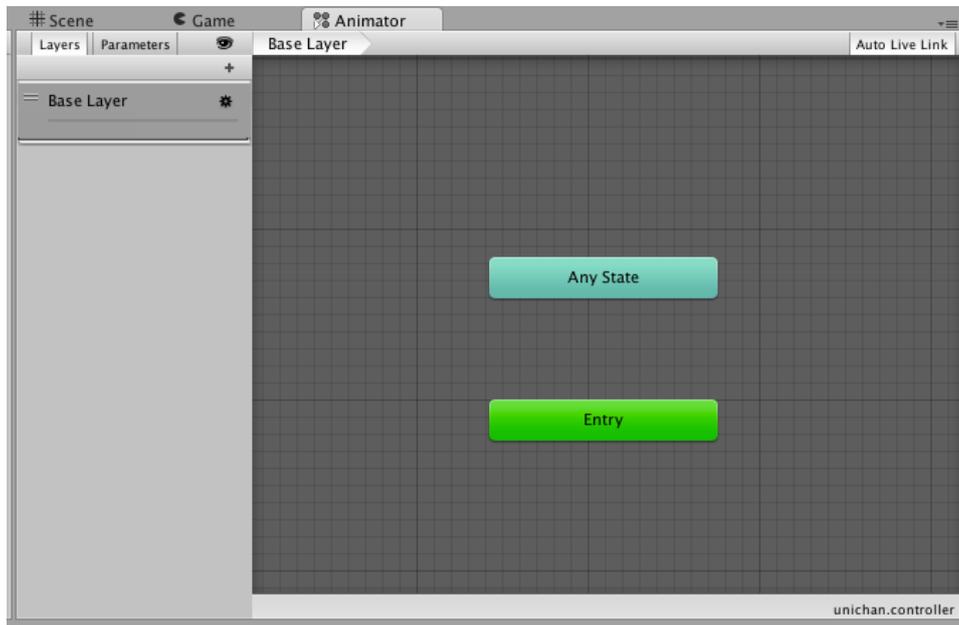


②Animator Controllerの名前は、unichanとしました。名前はなんでも構いません。

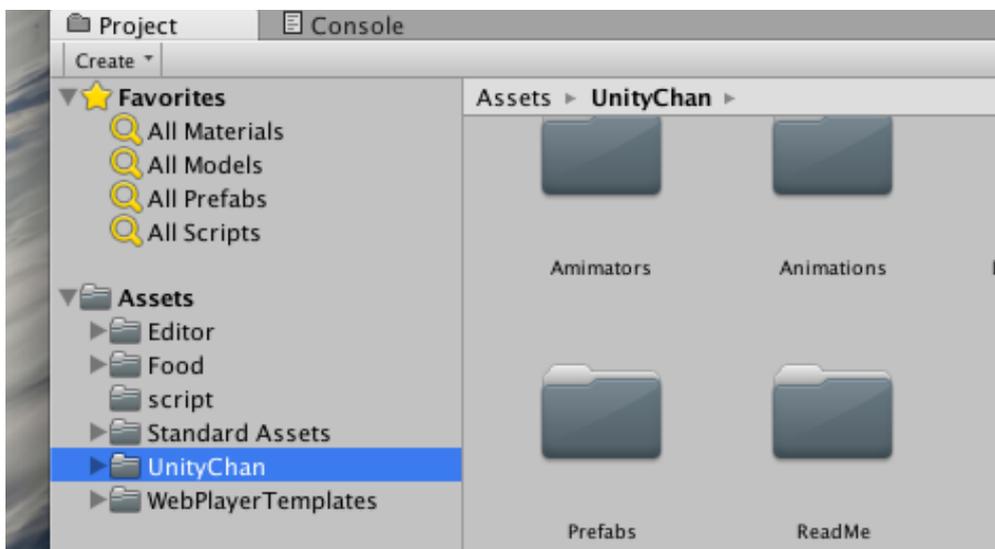
③Animator Controllerをダブルクリックします。



④下のようなAnimatorの画面が現れます。



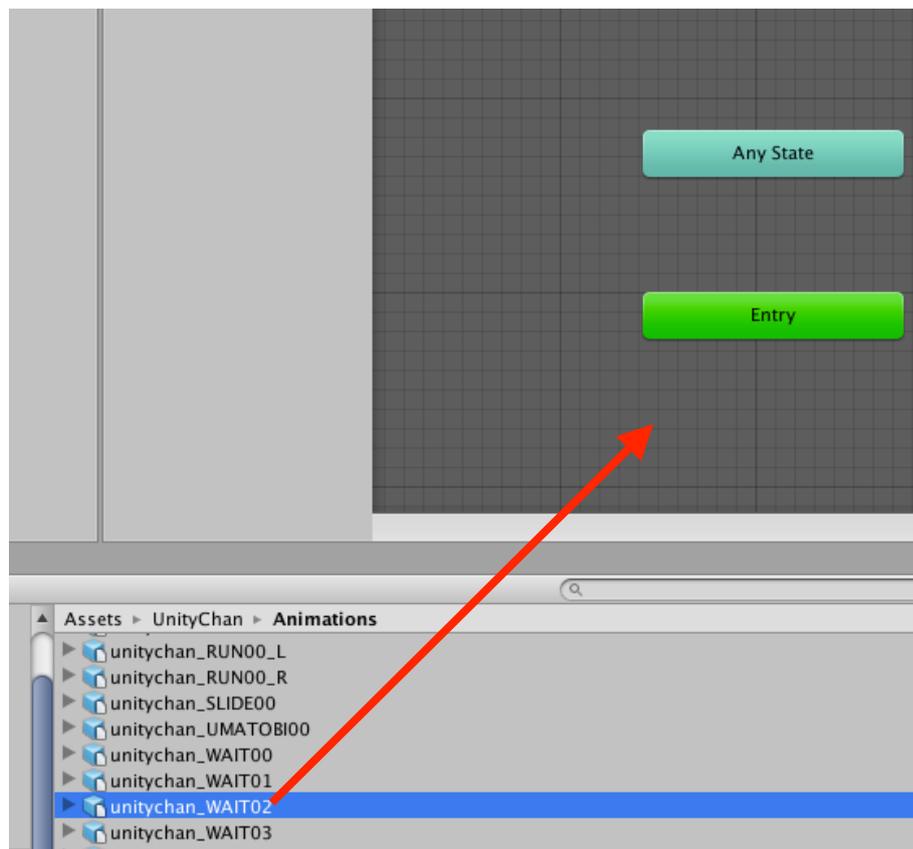
⑤ProjectビューのAssetsの中のUnityChanをクリックします。次に、Animationsをダブルクリックします。



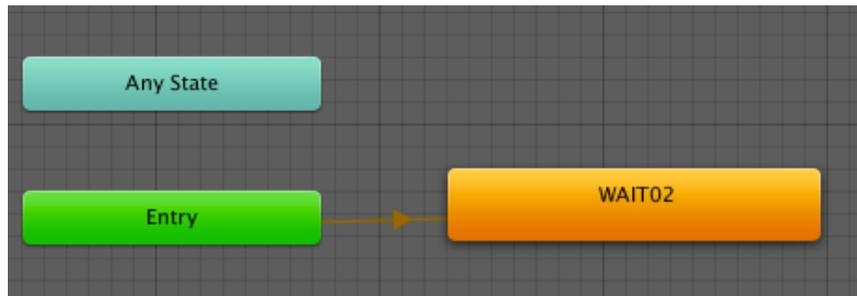
⑥unitychan\_WAIT02を探しますが、探しにくいときは、右下のレバーを一番左まで動かします。



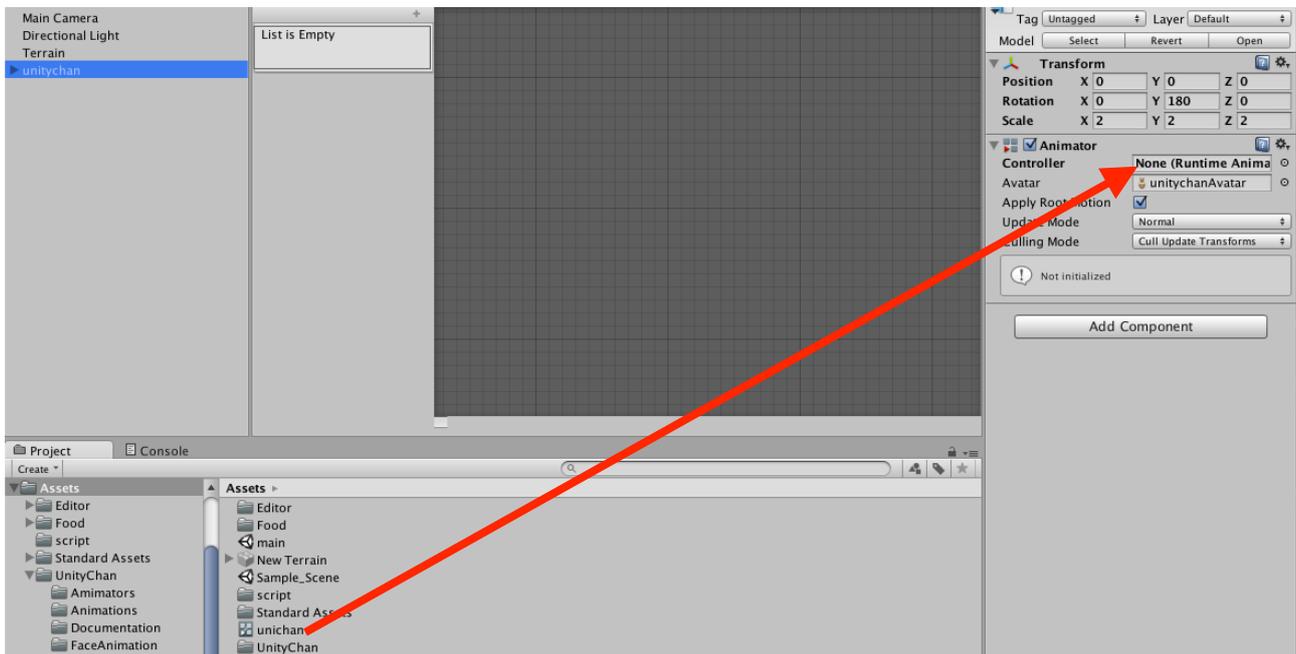
⑦unitychan\_WAIT02をAnimatorの画面の中に移動させます。



⑧WAIT02がEntryにつながります。位置を適当に調整します。



⑨Animatorのunichanを、unitychanに設定します。

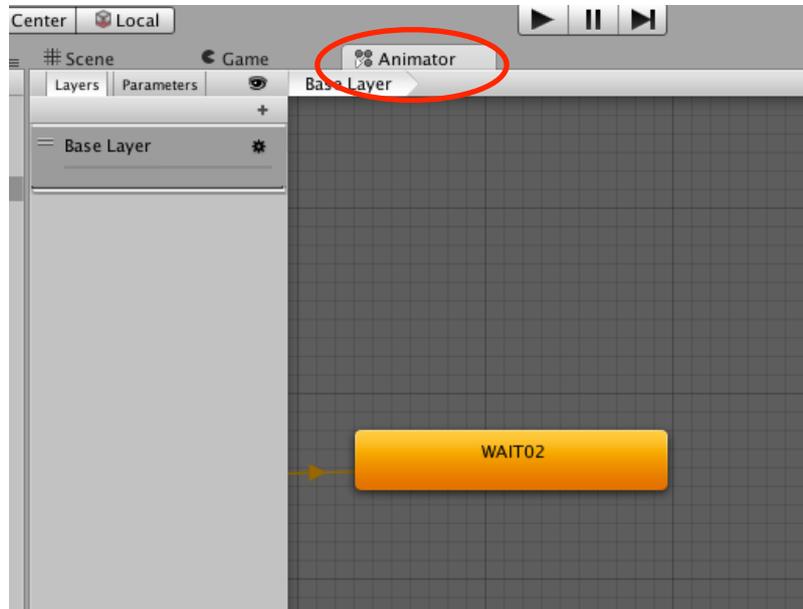


⑩再生ボタンを押してみましよう。unitychanが動きましたか？



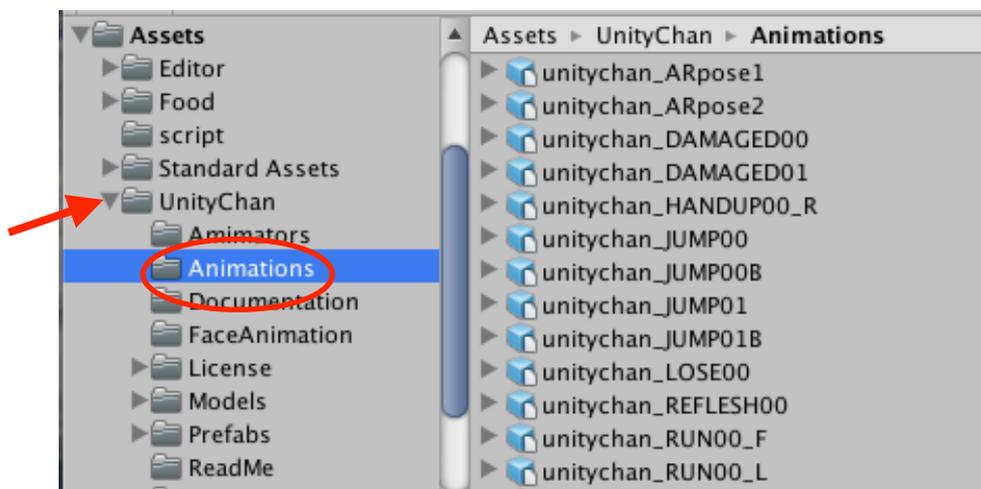
☆走るアニメーションを追加します。

①Animatorのタブをクリックし、Animatorビューに切り替えます。

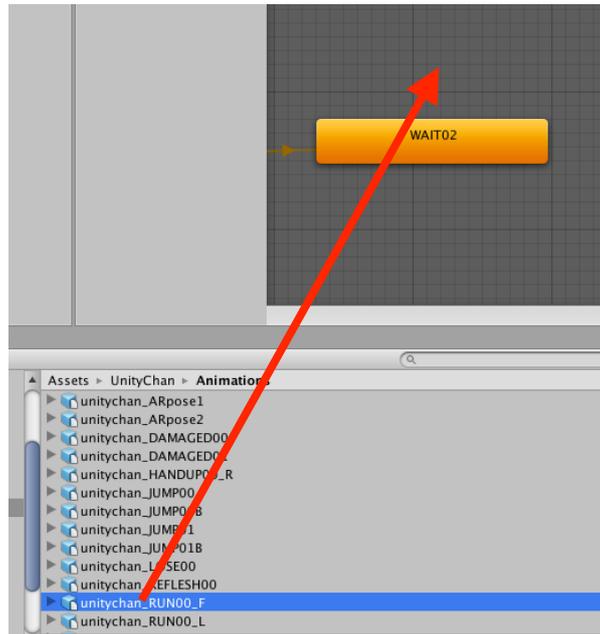


②Assetsの中のUnityChanのAnimationをクリックします。Animationが表示されます。

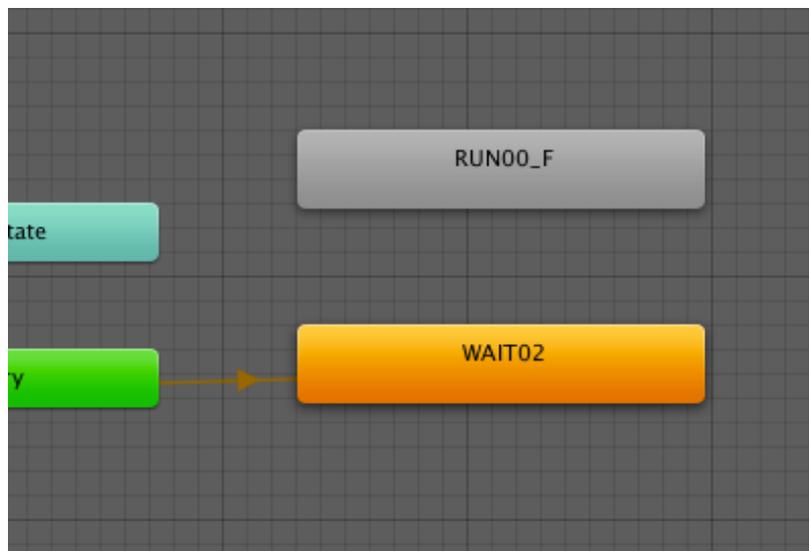
※Assetsの中のフォルダは、左側の三角マーク (▶) をクリックすることで展開されます。展開されているフォルダは、三角マークが下向き (▼) になります。



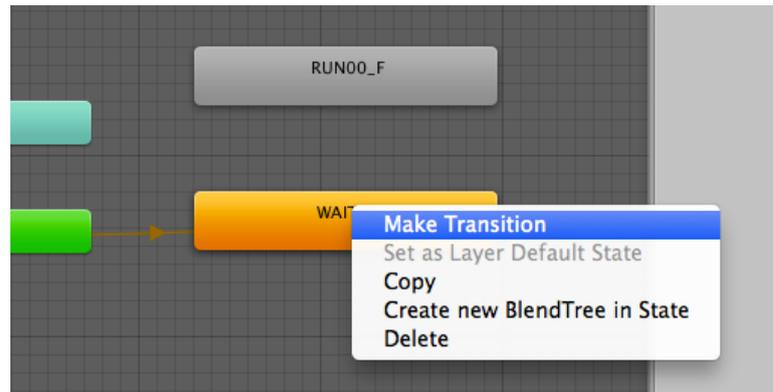
③unitychan\_RUN00\_Fを、Animatorビューに移動します。



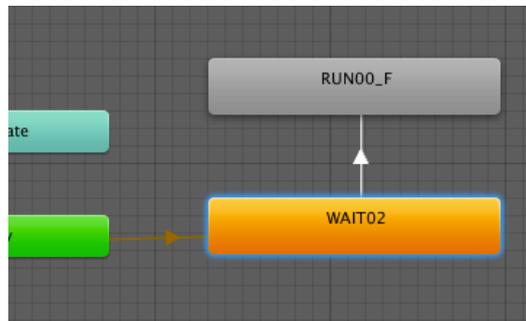
④WAIT02の上方に、RUN00\_Fを配置します。



⑤WAIT02を右クリックし、Make Transitionをクリックします。

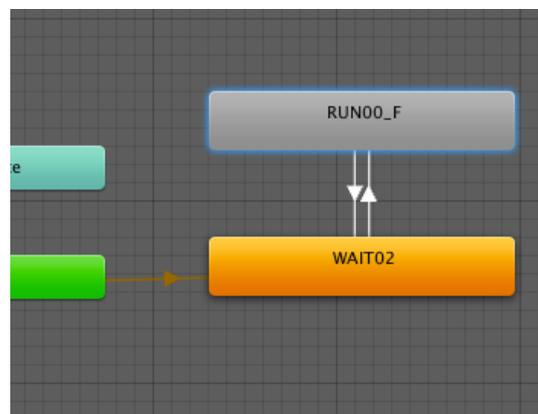


⑥矢印が出てくるので、RUN00\_Fをクリックして矢印をつなぎます。



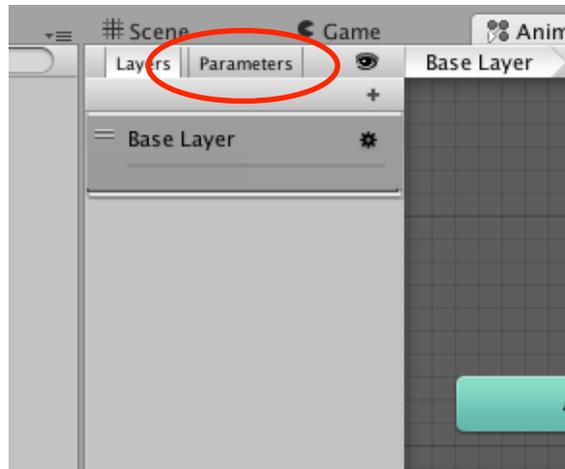
⑦RUN00\_Fを右クリックし、Make Transitionをクリックします。

⑧矢印が出てくるので、WAIT02をクリックして矢印をつなぎます。

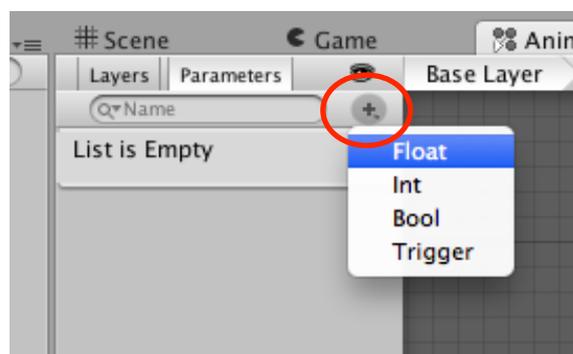


☆アニメーションの切り替えの条件を作ります。

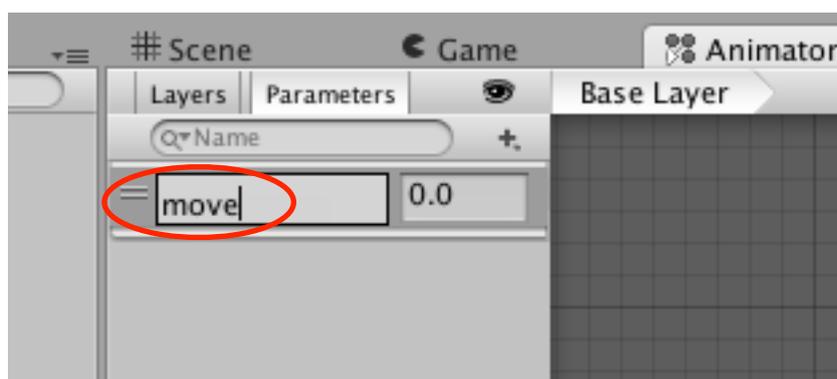
①Animatorビューを表示し、Animatorビューの左上にあるParameterをクリックします。



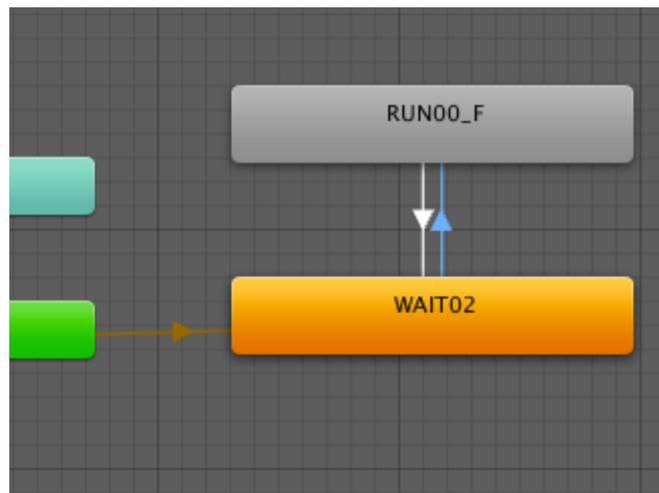
②Parameterの下の+ボタンをクリックし、Floatを選択します。



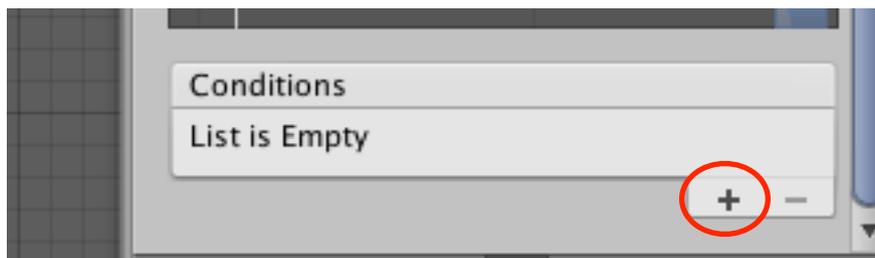
③変数名を、moveにします。



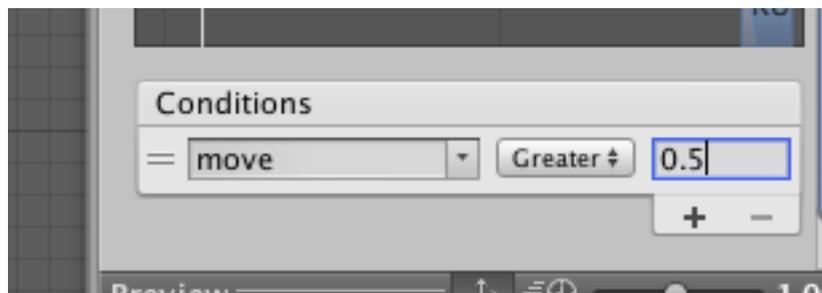
④WAIT02からRUN00\_Fへの矢印をクリックします。



⑤Inspectorビューの下のほうにあるConditionsの+マークをクリックします。

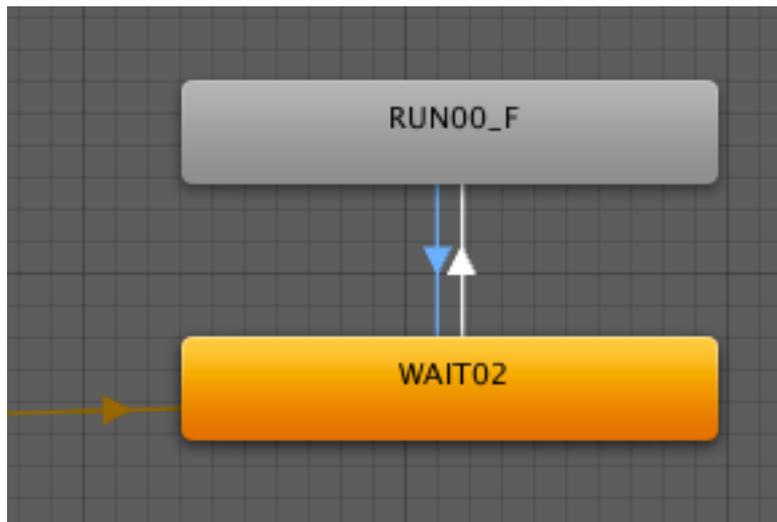


⑥右側の数字を0.5にします。  
move Greater 0.5となります。

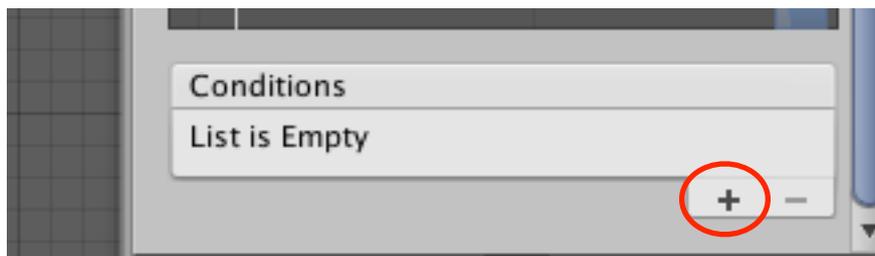


つまり、 $\text{move} > 0.5$  となったとき、RUN00\_Fに遷移します。  
Has Exit Timeのチェックを外します。

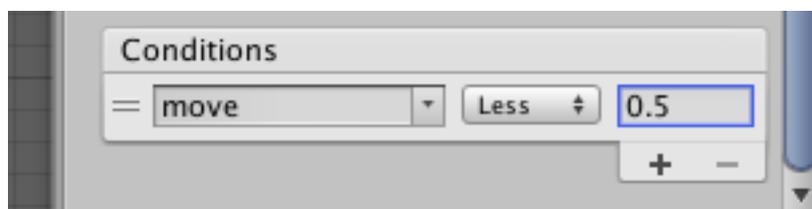
⑦RUN00\_FからWAIT02への矢印をクリックします。



⑧Inspectorビューの下のほうにあるConditionsの+マークをクリックします。



⑨右側の数字を0.5にします。Greaterの部分をクリックして、Lessに切り替えます。  
move Less 0.5となります。



つまり、 $\text{move} < 0.5$  となったとき、WAIT02に遷移します。  
Has Exit Timeのチェックを外します。

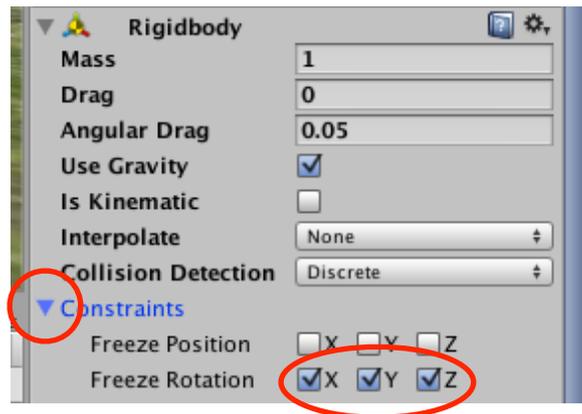
## ☆unitychanの準備

①unitychanに、Rigidbodyを付けます。

Hierarchyビューのunitychanをクリックします。Inspectorビューの下にあるAdd Componentをクリックします。

Physicsをクリックして、Rigidbodyをクリックします。

Constrainの左側の▶マークをクリックし、Freeze Rotationにチェックを入れます。

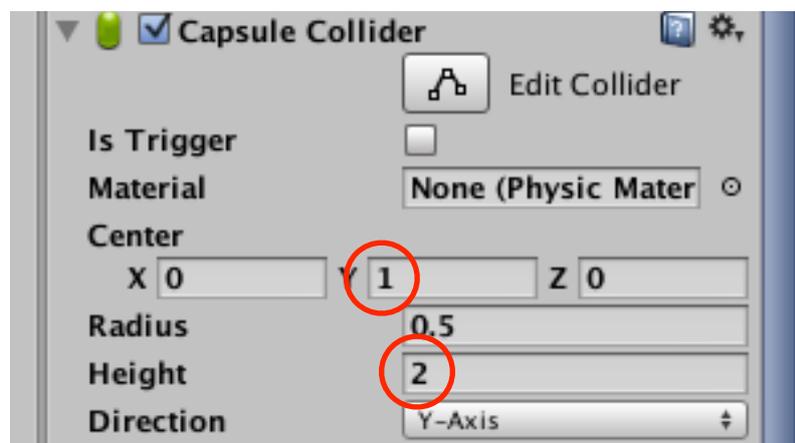


②unitychanに、Colliderを付けます。

Hierarchyビューのunitychanをクリックします。Inspectorビューの下にあるAdd Componentをクリックします。

Physicsをクリックして、Capsule Colliderをクリックします。

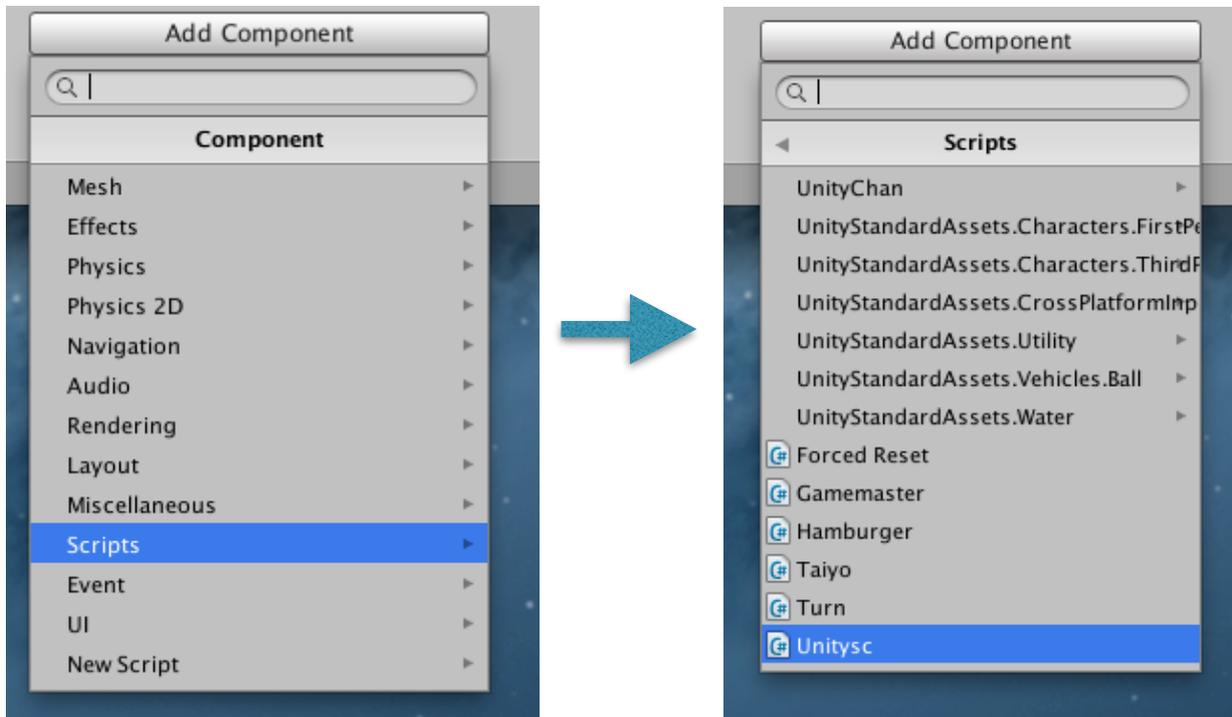
Capsule Colliderの値を調整します。



③unitychanに、スクリプトを付けます。

Hierarchyビューのunitychanをクリックします。Inspectorビューの下にあるAdd Componentをクリックします。

Scriptsをクリックして、Unityscをクリックします。



```
using UnityEngine;
using System.Collections;

public class unitysc : MonoBehaviour {

    Animator anim;
    public float moveSpeed = 500;

    void Start () {
        anim = GetComponent<Animator>();
    }

    void Update () {

        float mov = Input.GetAxis("Vertical");

        anim.SetFloat("move",mov);
    }
}
```

④再生ボタンを押してみましよう。

☆unitychanが走るようにします。

①unityscをダブルクリックします。

```
GetComponent<Rigidbody>().AddForce(transform.forward*mov*moveSpeed);
```

の行の//を外します。

そして、Ctrl+Sで保存します。

②再生ボタンを押してみましよう。

上下のキーを押すと、走りましたか？

③カメラを、unitychanと一緒に動かします。

Hierarchyビューで、unitychanをクリックします。

InspectorビューのTransformのRotationを、X:0 Y:0 Z:0とします。

④HierarchyビューのMain Cameraを、Hierarchyビューのunitychanの上まで移動します。

⑤再生ボタンを押してみましよう。

上下のキーを押すと、カメラも追従しましたか？

⑥unityscをダブルクリックします。

```
public float turnSpeed = 10;  
float turn = Input.GetAxis("Horizontal");  
transform.Rotate(0f, turn*turnSpeed, 0f);
```

の合計3行の//を外します。

そして、Ctrl+Sで保存します。

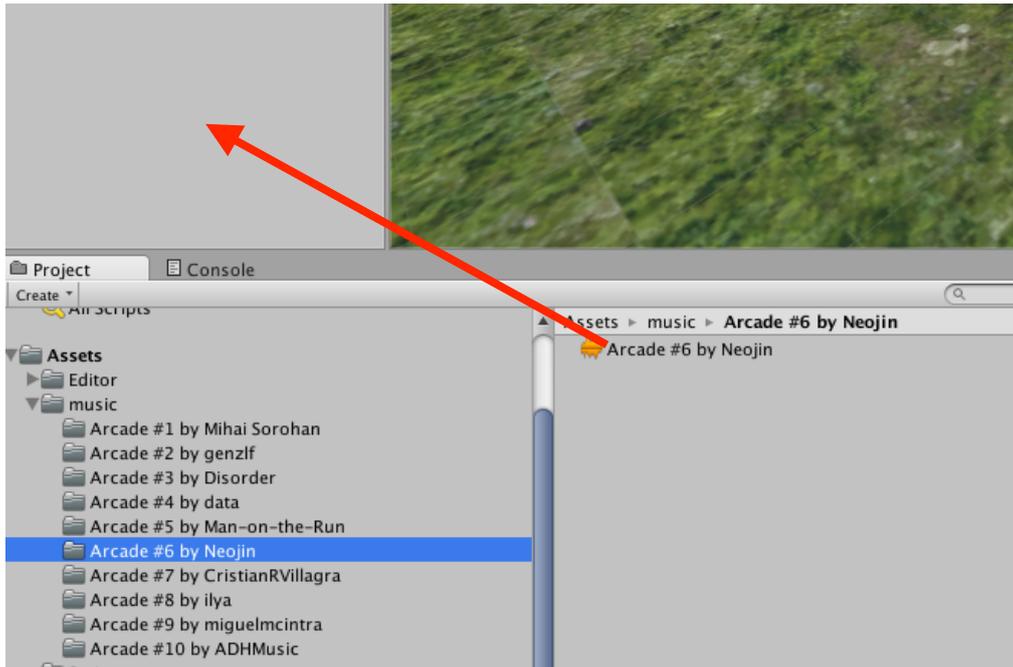
⑦再生ボタンを押してみましよう。

走るスピードと、回転するスピードは、unitychanを選択したときのInspectorビューのMove Speed, Turn Speedで調整できます。

☆音楽を流します。

①ProjectビューのAssetsにあるmusicのArcade #6 by Neojinをクリックします。

Arcade #6 by NeojinのフォルダにあるArcade #6 by Neojinを、Hierarchyビューに移動します。

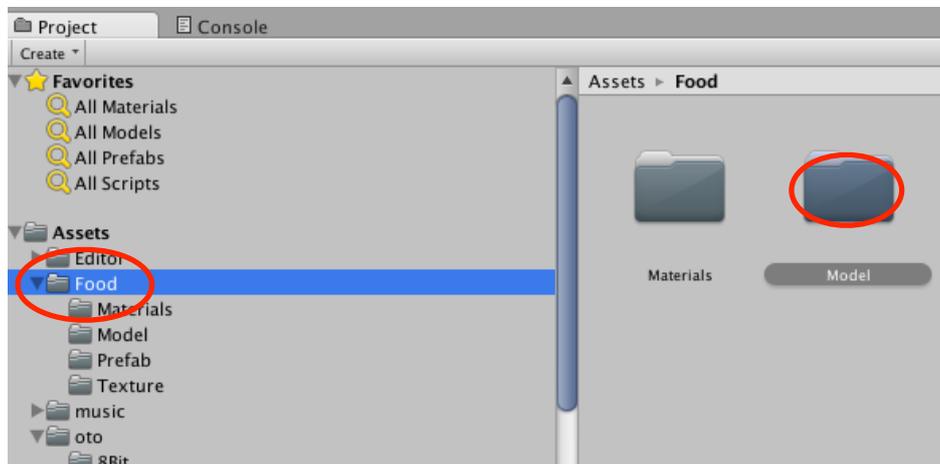


②Main Cameraをクリックし、InspectorビューのAudio Listenerにチェックを入れます。

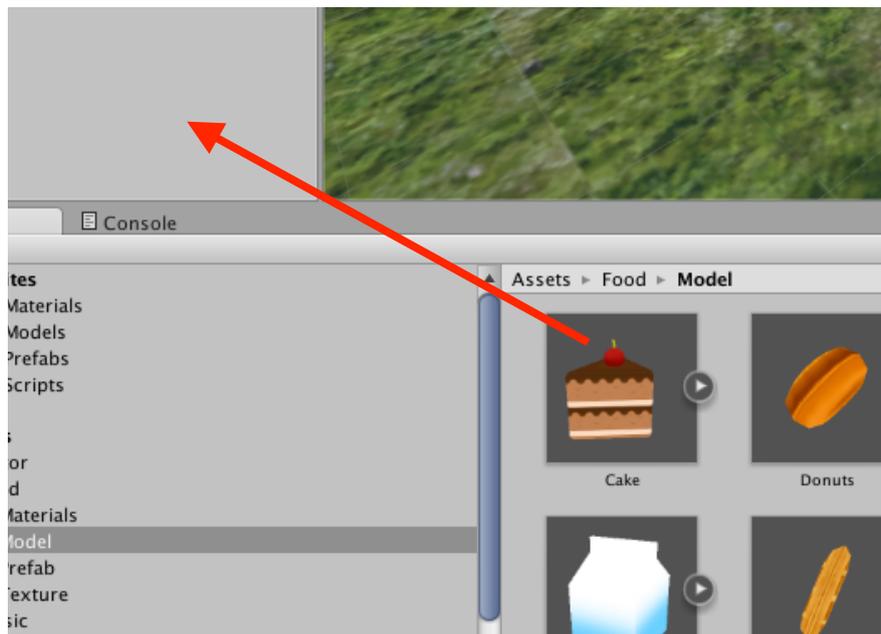
③再生ボタンを押してみましよう。

☆ワールドにアイテムを置いてみます。

①ProjectビューのAssetsの中のFoodをクリックします。続いてModelをダブルクリックします。



②CakeをHierarchyビューへ移動します。

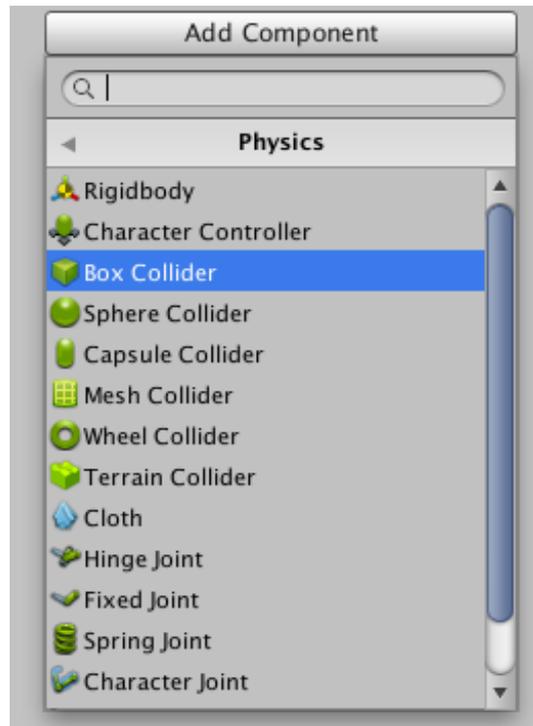


③CakeのScaleを、X:5 Y:5 Z:5とします。

④再生ボタンを押してみましよう。ケーキがありますか？

☆CakeにColliderを付けます。

①HierarchyビューのCakeをクリックします。InspectorビューのAdd Componentをクリックして、Physicsをクリックし、Box Colliderをクリックします。



②SceneビューのCakeに、緑色の枠が付いていることを確認してください。

※SceneビューにCakeが見えないときは、HierarchyビューのCakeをダブルクリックします。

③再生ボタンを押して、Cakeに体当たりしましょう。

### 練習問題

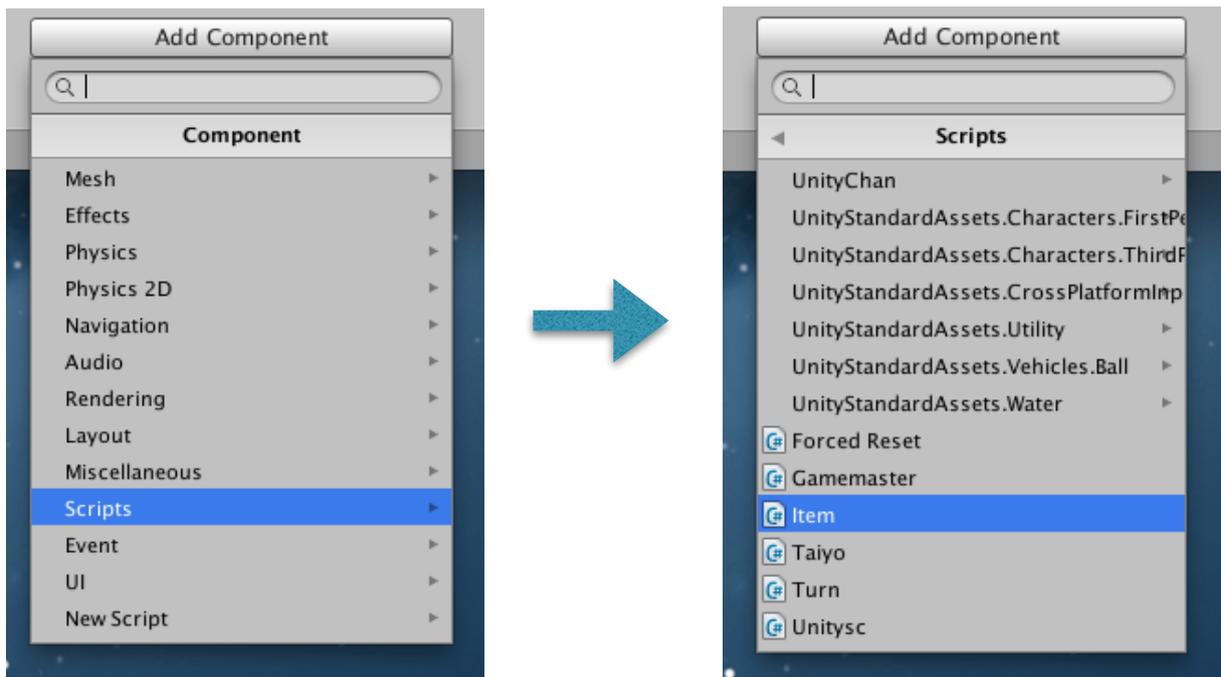
(1)CakeにRigidbodyを付けましょう。

(2)再生ボタンを押して、Cakeに体当たりしましょう。

☆Cakeにスクリプトを付けます。

①HierarchyビューのCakeをクリックし、InspectorビューのAdd Componentをクリックします。

②Scriptsをクリックし、Itemをクリックします。



③Inspectorビューのitemをダブルクリックしてください。スクリプトの内容が表示されます。

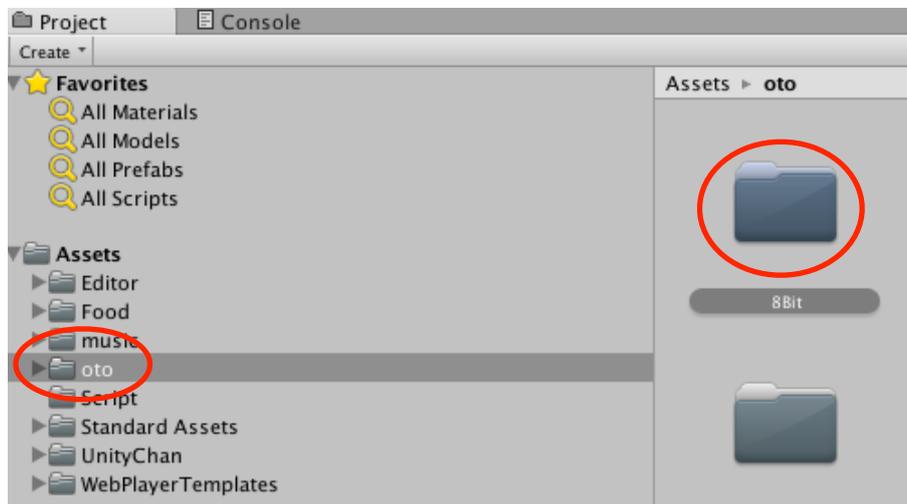
```
void OnCollisionEnter(Collision col){  
    if (col.gameObject.tag == "Player") {  
        //GameObject.Find("Get").GetComponent<AudioSource>().Play();  
        Destroy(gameObject);  
    }  
}
```

④unitychanのTagを、Playerにしておきます。

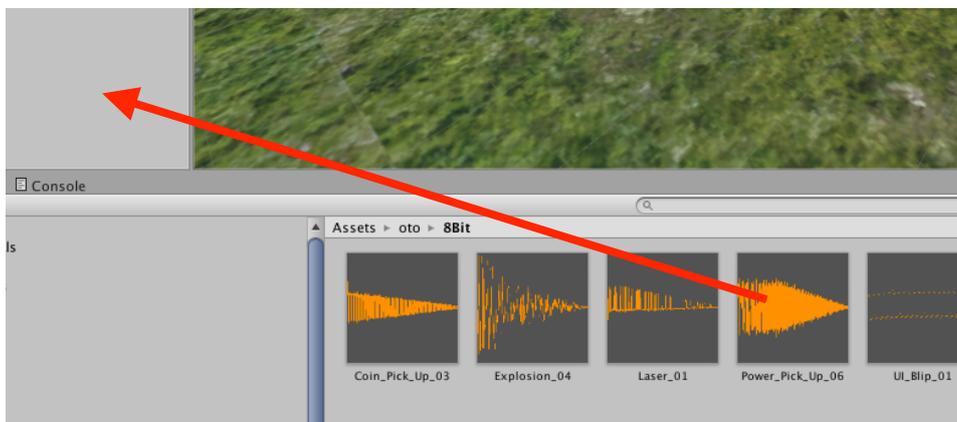
⑤再生ボタンを押して、Cakeに体当たりしてみましよう。

☆Cakeをゲットしたときの効果音を入れます。

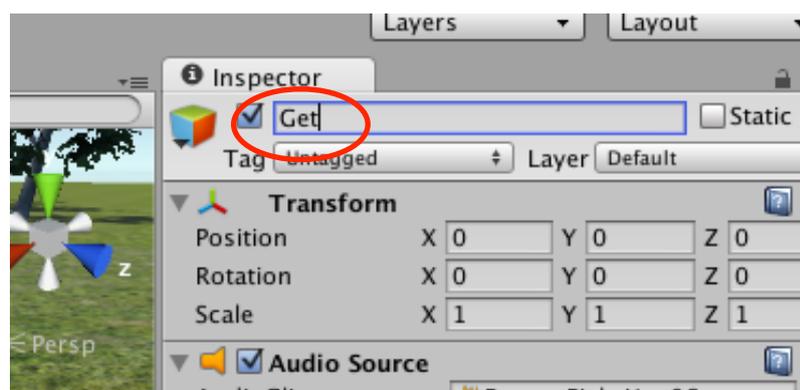
①ProjectビューのAssetsの中にあるotoをクリックし、8Bitをダブルクリックします。



②Power\_Pick\_Up\_06を、Hierarchyビューへ移動します。



③Inspectorビューで名前を、Getに変更します。



④Inspectorビューで、Play On Awakeのチェックを外します。



⑤HierarchyビューのCakeをクリックし、itemをダブルクリックしてスクリプトを開きます。

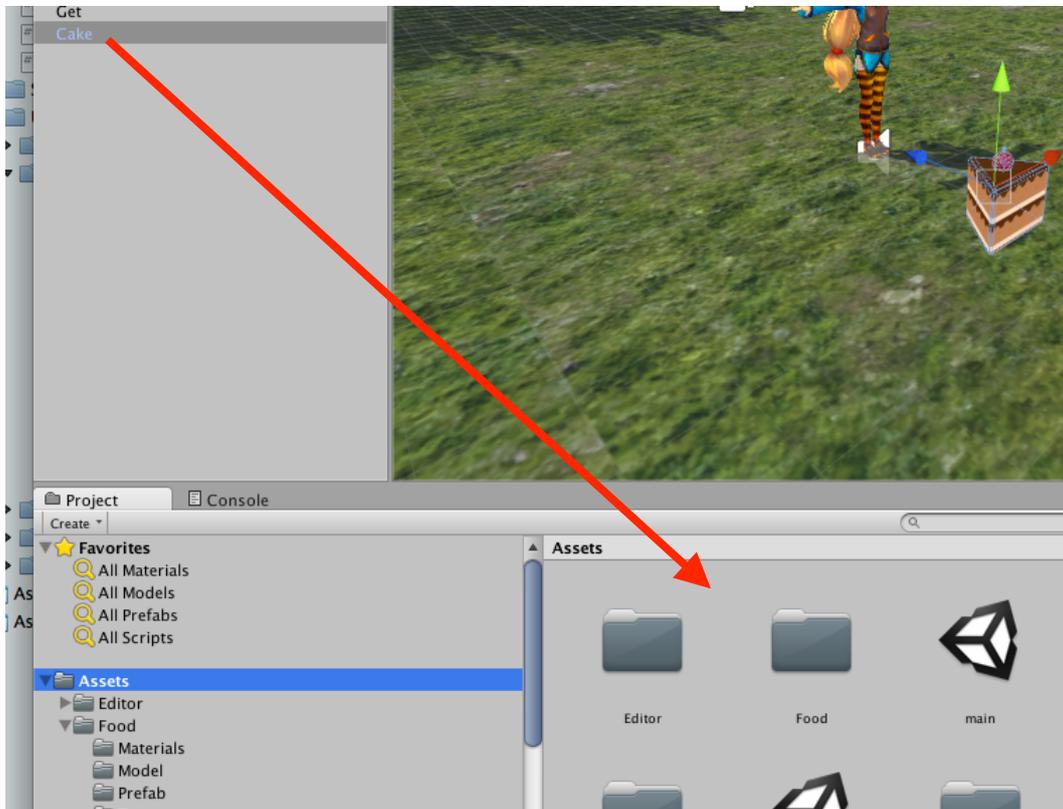
⑥次の行のコメントを外します。Ctrl+Sでセーブします。

```
GameObject.Find("Get").GetComponent<AudioSource>().Play();
```

⑦再生ボタンを押して、Cakeに体当たりしてみましよう。

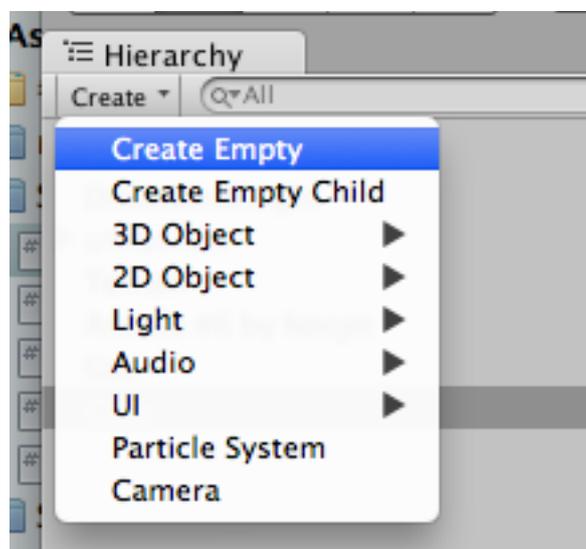
☆Cakeをたくさん作ります。

①ProjectビューのAssetsをクリックし、HierarchyビューのCakeをAssetsの中へ移動します。



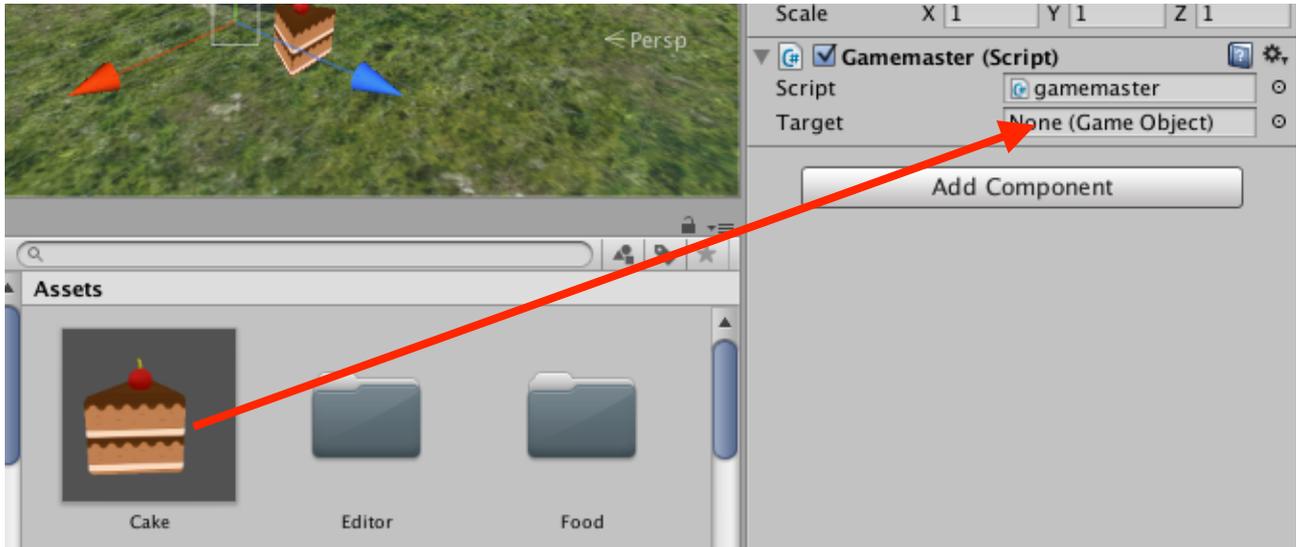
※フォルダの中へ入れないでください。

②HierarchyビューのCreateからCreate Emptyをクリックします。



③HierarchyビューのGameObjectをクリックして、InspectorビューのAdd Componentをクリックし、Scriptsをクリックして、Gamemasterをクリックします。

④Assetsの中のCakeを、Targetの枠の中へ移動します。

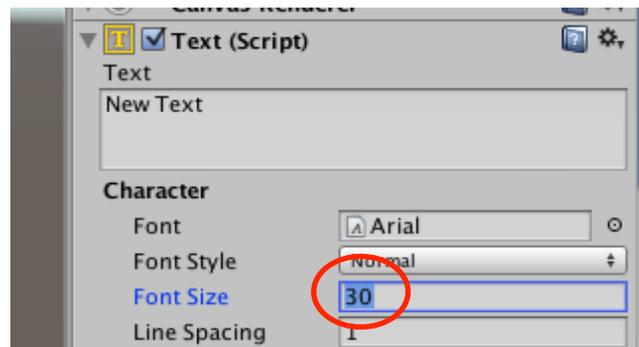
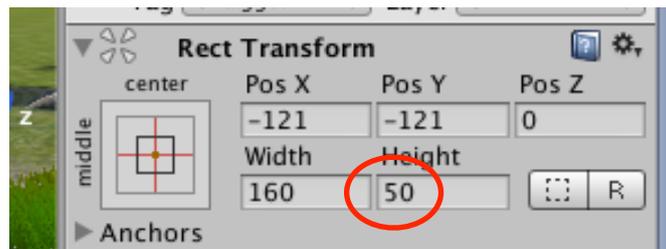


⑤再生ボタンを押してみましよう。Cakeがたくさん出てきましたか？

☆スコア表示を行います。

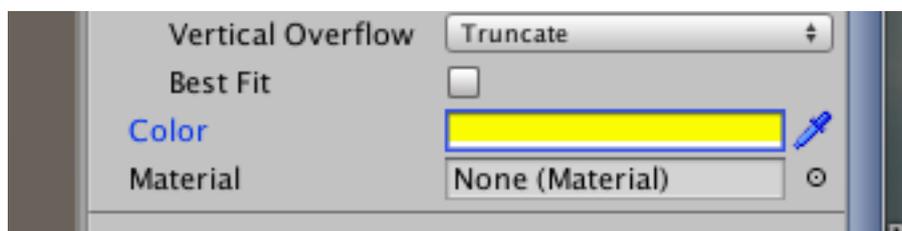
①HierarchyビューのCreateをクリックし、UIを選択してTextをクリックします。

②Textをクリックして、Inspectorビューの情報を書き換えます。



③文字色を目立つ色にします。

Colorの枠の中をクリックすると、色が設定できます。



④Textに、スクリプトを取り付けます。

Add Componentをクリックし、Scriptsをクリックして、Scoreをクリックします。

⑤HierarchyビューのCakeをクリックし、スクリプトのitemをダブルクリックします。

```
void OnCollisionEnter(Collision col){  
    if (col.gameObject.tag == "Player") {  
        GameObject.Find("Get").GetComponent<AudioSource>().Play();  
        GameObject.Find ("Text").SendMessage("ScoreUp");  
        Destroy(gameObject);  
    }  
}
```

コメントとなっている行の//を外します。

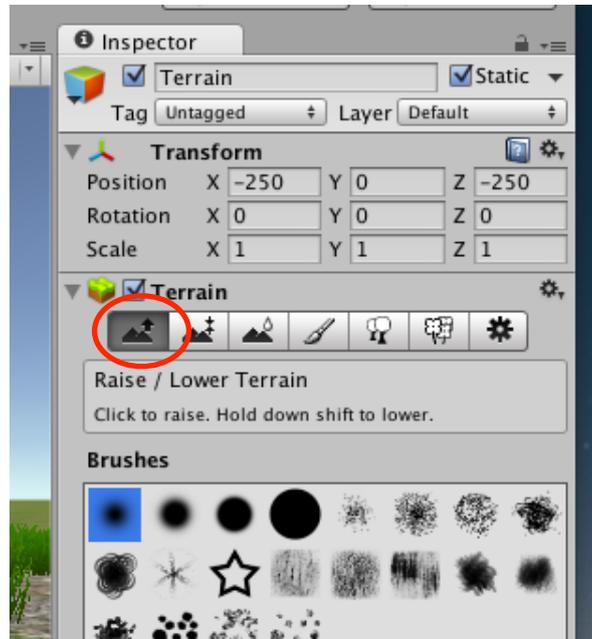
⑥再生ボタンを押してみましよう。スコアが表示されますか？

## ★おまけ～その1～

全部出来てしまった人は、やってみましょう。

☆大地に山を作る。

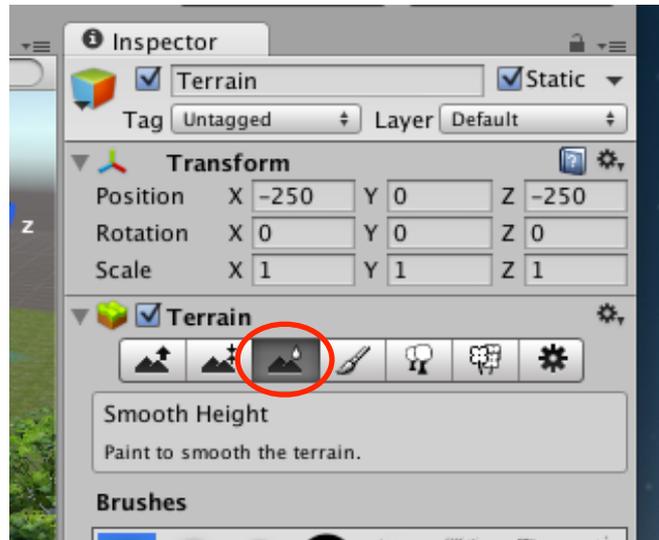
①HierarchyビューのTerrainをクリックし、Inspectorビューの山のボタンを押します。



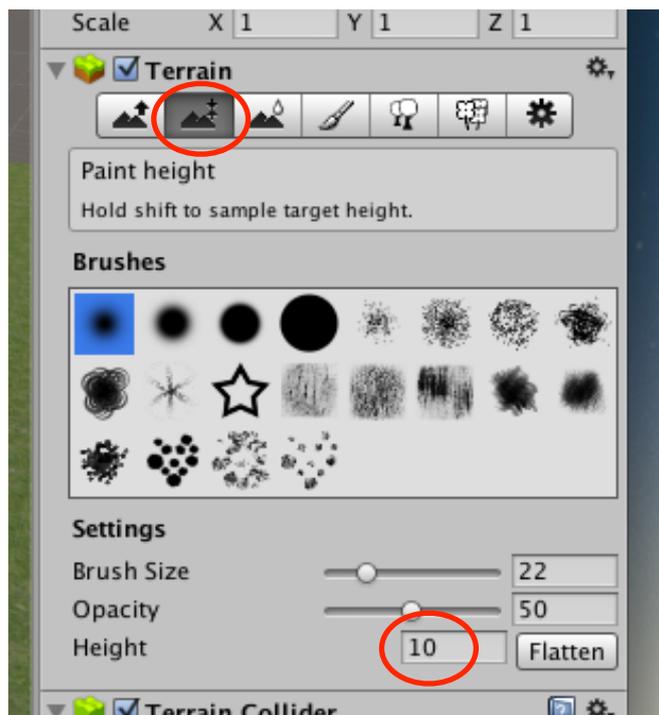
②マウスでSceneビューのTerrainをクリックしながらなぞると、山ができます。



③山が高くなりすぎたときは、Shiftを押しながら、クリックしながらマウスでなぞります。また、尖った部分を滑らかにしたいときは、山に水滴が垂れている？ボタンを押して、クリックしながらマウスでなぞります。

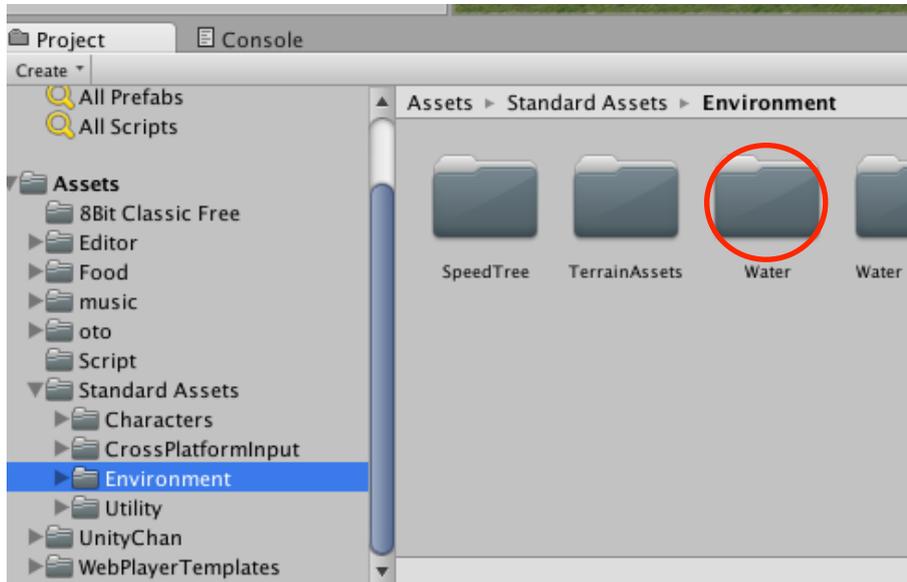


④山の高さをある高さまでにしたいときは、矢印がついているボタンを押します。そして、高さ(height)を設定します。その後、クリックしながら、マウスでなぞります。



## ☆大地に水をつくる

①Standard Assetsの下のEnvironmentの中に、Waterというフォルダがあります。これをダブルクリックします。



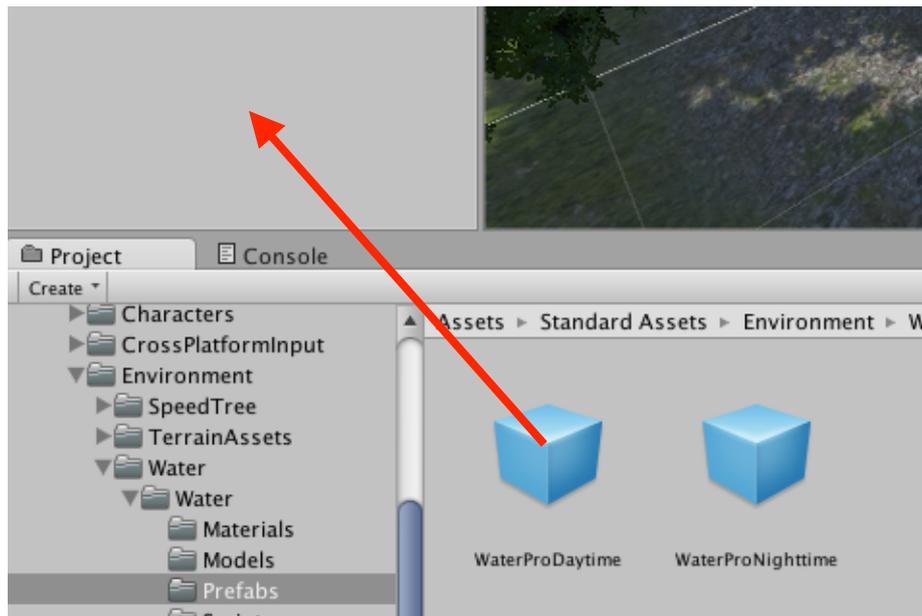
②さらに、その下のWaterをダブルクリックします。



③その下のPrefabsをダブルクリックします。



④WaterProDaytimeを、Hierarchyビューに移動します。



⑤HierarchyビューでWaterProDaytimeをクリックし、InspectorビューでPositionを、X:0 Y:0.1 Z:0とします。

⑥再生ボタンを押してみましよう。水面に、景色が映っているのは確認してください。