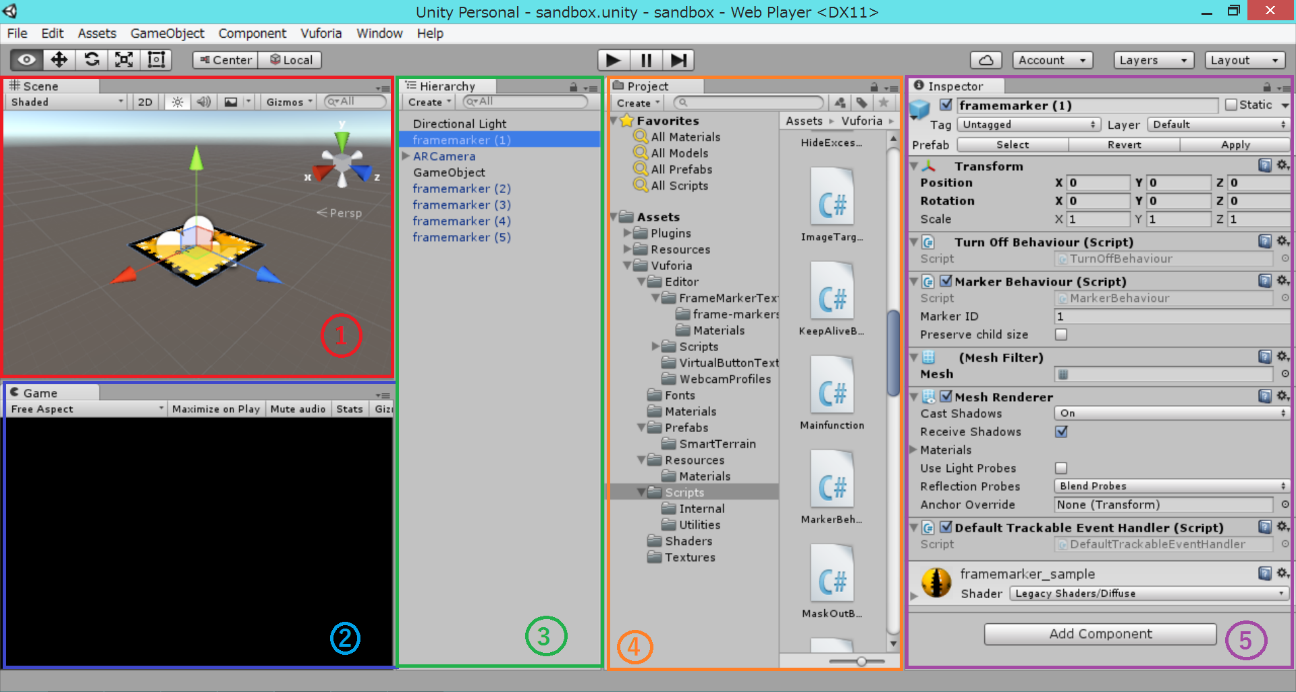
Unity & Vuforia

○基礎画面(黄色→2 by 3)



1…シーン。色々な角度からオブジェクトを見る

2…ゲーム。プログラムを動かした時に実際に見える画面

3…ヒエラルキー。ここにオブジェクトを置くと、シーンに反映される

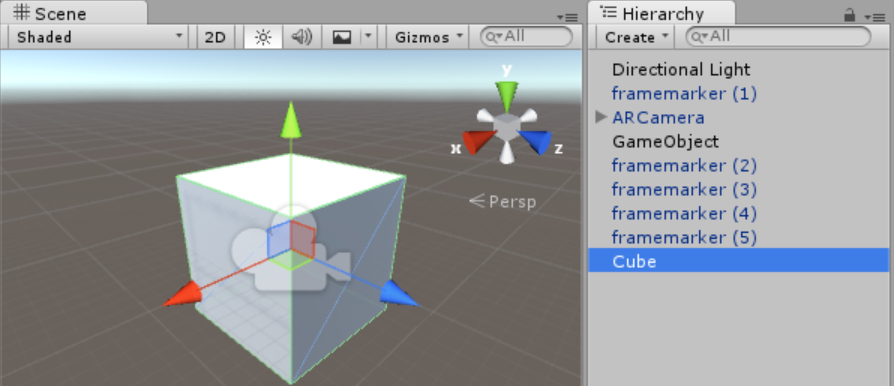
4…プロジェクト。ファイル郡

5…インスペクター。選択しているオブジェクトの詳細

○基本操作

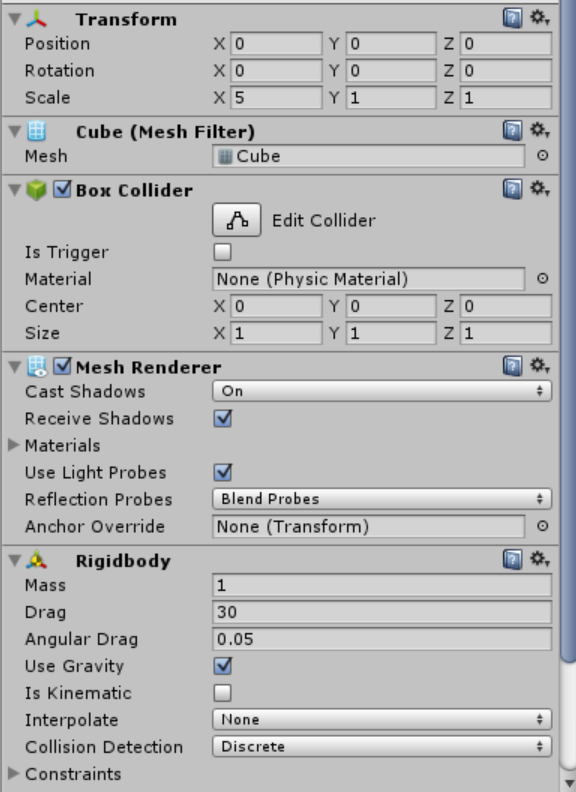
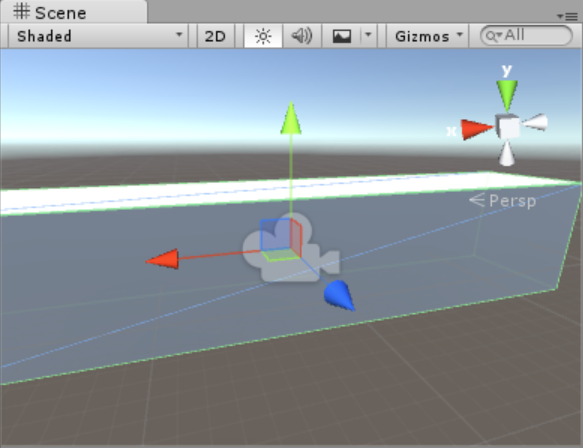
・オブジェクトの配置

1. ヒエラルキーで右クリック
2. 3D object→cube



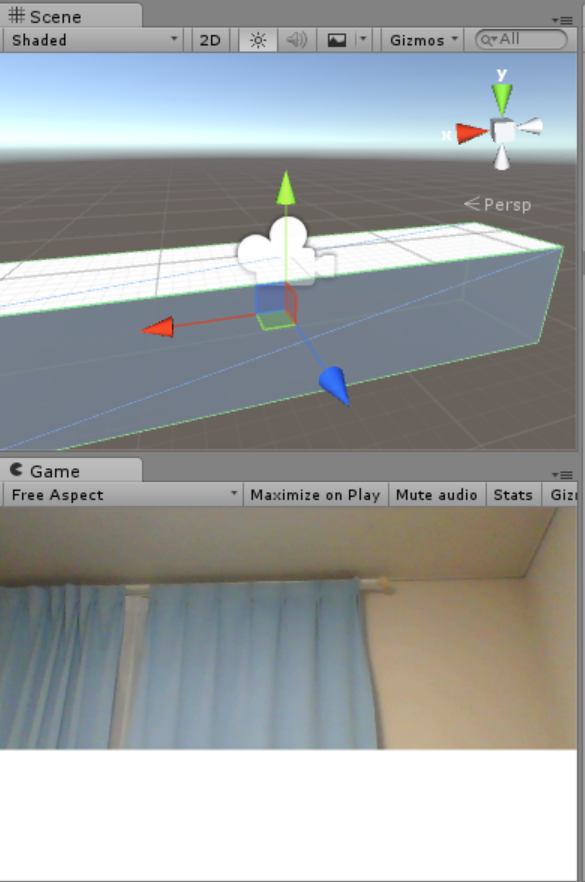
・インスペクターの操作

1. ヒエラルキーでcubeを選択
2. TransformコンポーネントのScaleのXを5にする
3. 一番下のAdd Componentをクリック
4. Physics→Rigidbodyを選択
5. RigidbodyコンポーネントのDrag(空気抵抗)を30にする



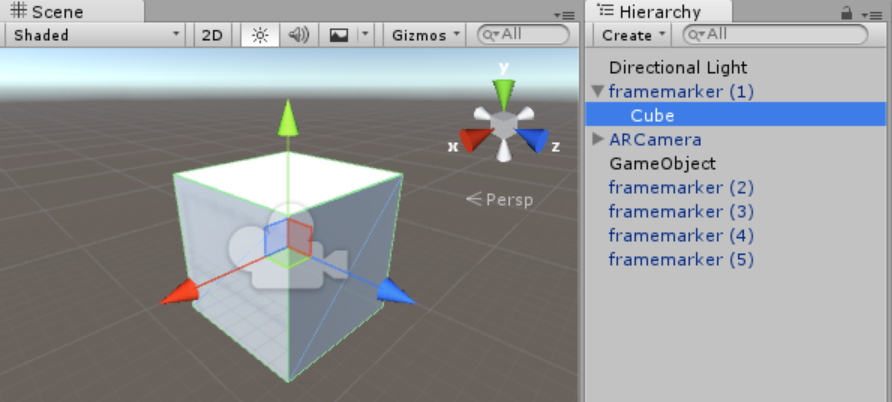
・実行

1. 上の三角かCtrl+P
2. Unity Web Player～が出たら、Always～をクリック



・子要素の作成

1. 作成したcubeをframemarker (1)にD&D

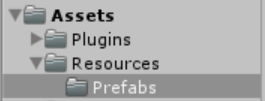


※ framemarker (X)の子要素は、そのマーカー上に表示される立体となる

・プレハブの作成

※ プレハブ…立体のコピー元と覚えておく

1. Assetsフォルダを右クリック
2. Create→Folder
3. Resourcesフォルダを作成
4. Resourcesフォルダを右クリック
5. Create→Folder
6. Prefabsフォルダを作成
7. Prefabsフォルダを右クリック
8. Create→Prefab
9. mycubeを作成
10. ヒエラルキーのcubeをプロジェクトのmycubeにD&D



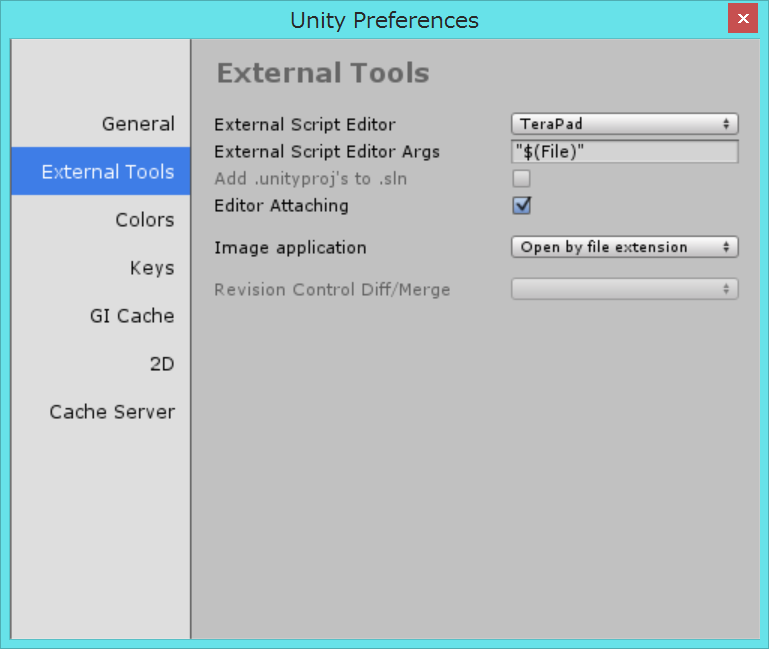
・オブジェクトの削除

1. ヒエラルキーのcubeを右クリック
2. delete

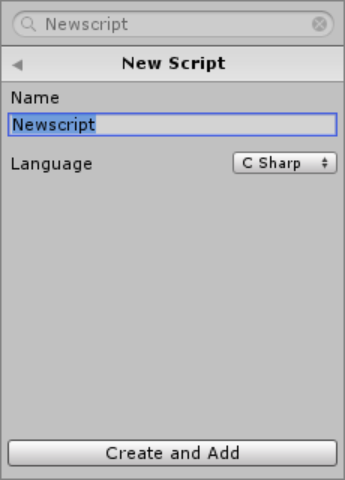
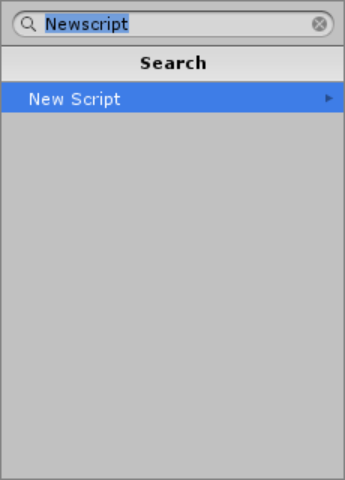
○スクリプト(C#)

・準備

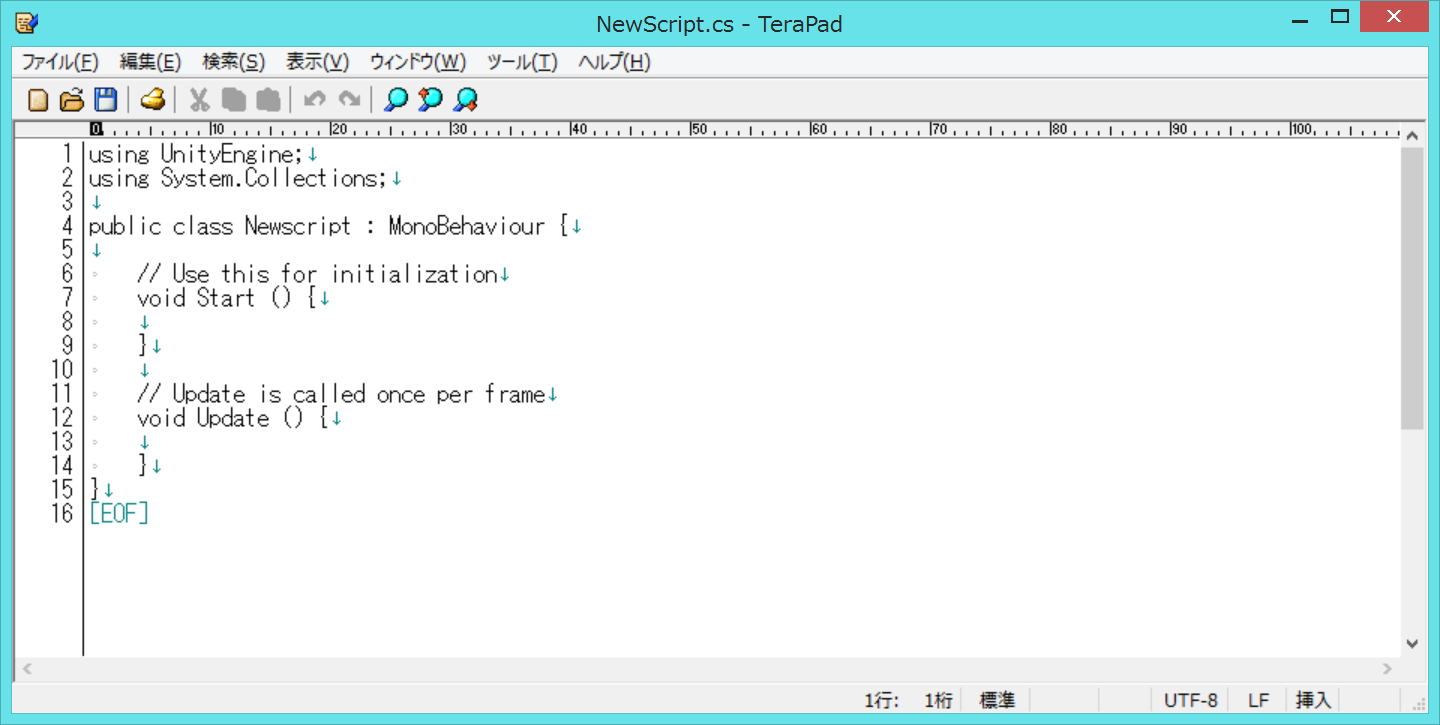
1. メニューバーのEdit→Preferences
2. External Script Editerで使いたいエディタを選択(デフォはMonoDevelop)



1. ×でPreferencesを閉じる
2. ヒエラルキーを右クリック
3. Create Empty(すでにGameObjectがあれば不要)
4. GameObjectを選択
5. インスペクターでAdd Component
6. 検索ボックスで「NewScript」と入力
7. New Script→Create and Add



1. Assets/にあるNewScriptをダブルクリック



↓

一番最初

(1フレーム？)

に実行される

毎フレームごとに実行される

void Start(){

}

void Update(){

}

・試し書き

void Start () {

Debug.Log("Start");

}

void Update () {

Debug.Log("Update");

}

！Start

・・・

！Update

！Update

・・・

1. メニューのWindow→ConsoleまたはCtrl+C
2. 実行
3. コンソールに右が出力(他の出力はあるかもしれないが)
4. GameObjectのインスペクターでNewScripコンポーネント

をRemove＆プロジェクトのNewScriptをDelete

・出力

Debug.Log(<文字列や数値>);

もしくは、

print(<文字列や数値>);

Debug.Log(“Hello,World!”);

print(1+1);

・オブジェクトの検索

GameObject <変数名> = GameObject.Find(<オブジェクト名(シーン)>);

GameObject marker = GameObject.Find("framemarker (1)");

※オブジェクトの変数型はGameObject

・プレハブのロード

GameObject <変数名> =

　　(GameObject)Resources.Load ("Prefabs/" + <プレハブ名>);

GameObject prefab =

　　(GameObject)Resources.Load ("Prefabs/mycube”);

・オブジェクトの生成

GameObject <オブジェクト名> =

Instantiate(<プレハブ変数名>,

transform.position,transform.rotation) as GameObject;

GameObject newcube =

Instantiate(prefab,transform.position,transform.rotation)

as GameObject;

・オブジェクトの子要素化

<子オブジェクト>.transform.parent = <親オブジェクト>.transform;

newcube.transform.parent = marker.transform;

・オブジェクトの名前変更

<オブジェクト>.name = <値>;

newcube.name = "newcube";

・コンポーネントの取得

<コンポーネントクラス> <変数名> =

<オブジェクト>.GetComponent.<<コンポーネントクラス>>;

Rigidbody rigidbody = newcube.GetComponent.<Rigidbody>;

※覚えておいたほうがいいコンポーネントクラス

・Transform…座標関係

・Rigidbody…質量や空気抵抗など物理演算関係

・コンポーネントパラメータの変更

<コンポーネント変数>.<パラメータ名> = <値>

rigidbody.drag = 1

○フレームマーカー

・フレームマーカーとは？

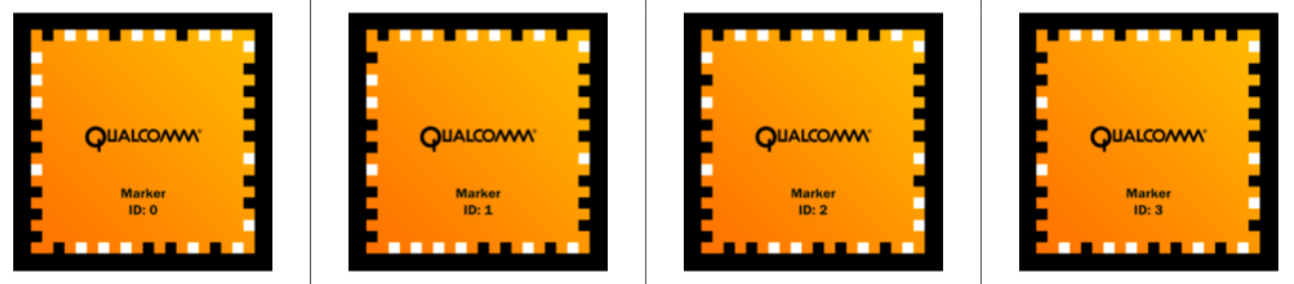
Vuforiaに登録されている512種類のマーカーのこと。縁の点線が重要らしく、中はわりとどうでもいい？認識精度はかなり高いが、イメージマーカー(写真をマーカーにする)とは違い、一部が隠れると認識されにくくなる。

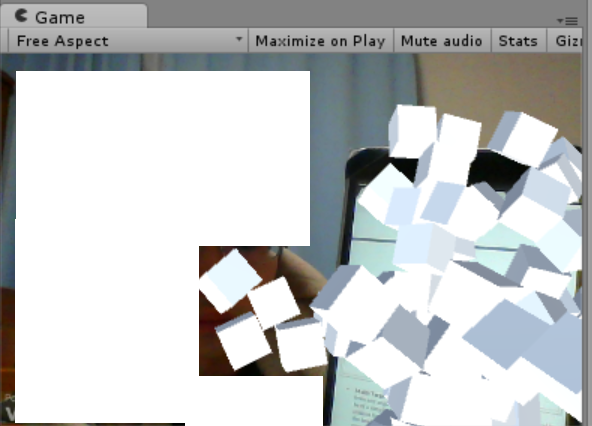
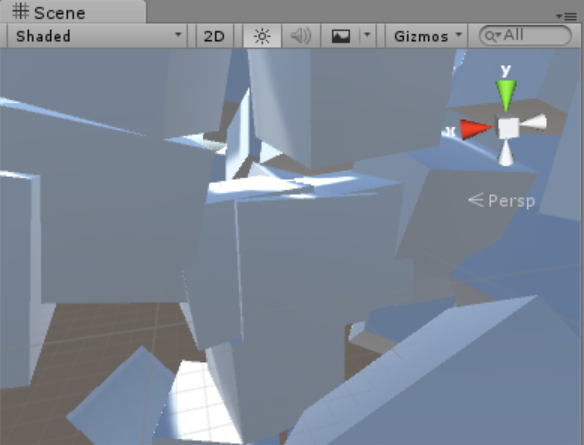
・インスペクター



Marker IDがマーカーの種類を表す。0～511まで設定可能。

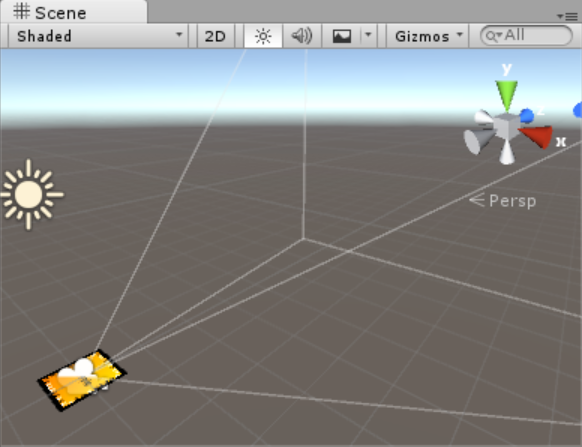
・マーカーと実行例





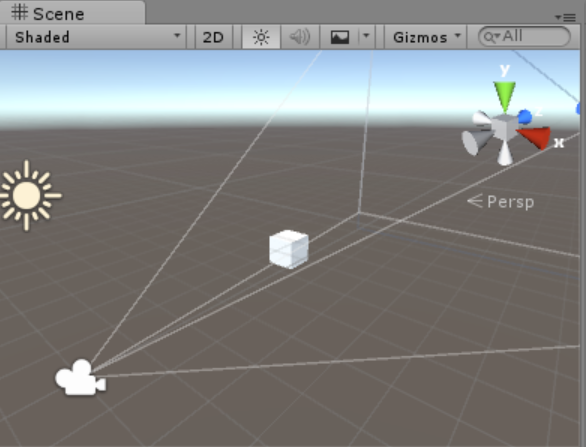
ﾀﾌﾞﾚｯﾄ(マーカーを映している)

○マーカーに頼らないAR(?)



ARマーカーを選択すると放射状に広がる線が見える(見づらかったら、右クリックしながらorホイールを押しながらorホイールを回して視点変更可能)。

これはカメラが捉える範囲である。この範囲内に立体を置くと、



カメラ上に常に立体が配置される。

○保存について

Unityはプロジェクトという単位で保存される。1つのプロジェクトはファイル(スクリプト、プレハブ、テクスチャ、**シーン**…)の集まりである。

シーンとはプロジェクトの中に複数存在できるヒエラルキーやインスペクターなどの状態である。

つまり、1つのプロジェクトでも異なるプログラム(シーン)を保存することができる。ただし、ファイルは共通なので、他シーンで使うスクリプトを消したり、(他のシーンにとって)余計なunityprojectファイルをインポートすると困ることになるかもしれない。