



# CHAPTER2\_Unityはじめの一步

## 一Unityの画面構成や基本操作を覚える

Unityの基本的な操作方法を学びUnityを操作できるようになってから、オブジェクトを配置して質感を自由に変更させることを学ぶ。C#のコード（Unityではスクリプトと表現する）を作成し、Unityでどのようにスクリプトを適用するのかを学ぶ。

### Chapter2 の目的

- シーンに3Dオブジェクトを配置することができる
- 配置したオブジェクトの質感を変更することができる
- スクリプトを作成し、適用する

## 2-1 シーンとアセット

### シーン

Scene（シーン）はUnityゲームの土台となるもので、シーン上に3Dオブジェクトや背景などを配置してゲームを作っていく。シーンは通常複数作成し、場面（シーン）毎に切り替えていく。例としてタイトルシーン、プレイシーン、リザルトシーン、エンディングシーン等が挙げられる。

図2-1 タイトルシーンの例

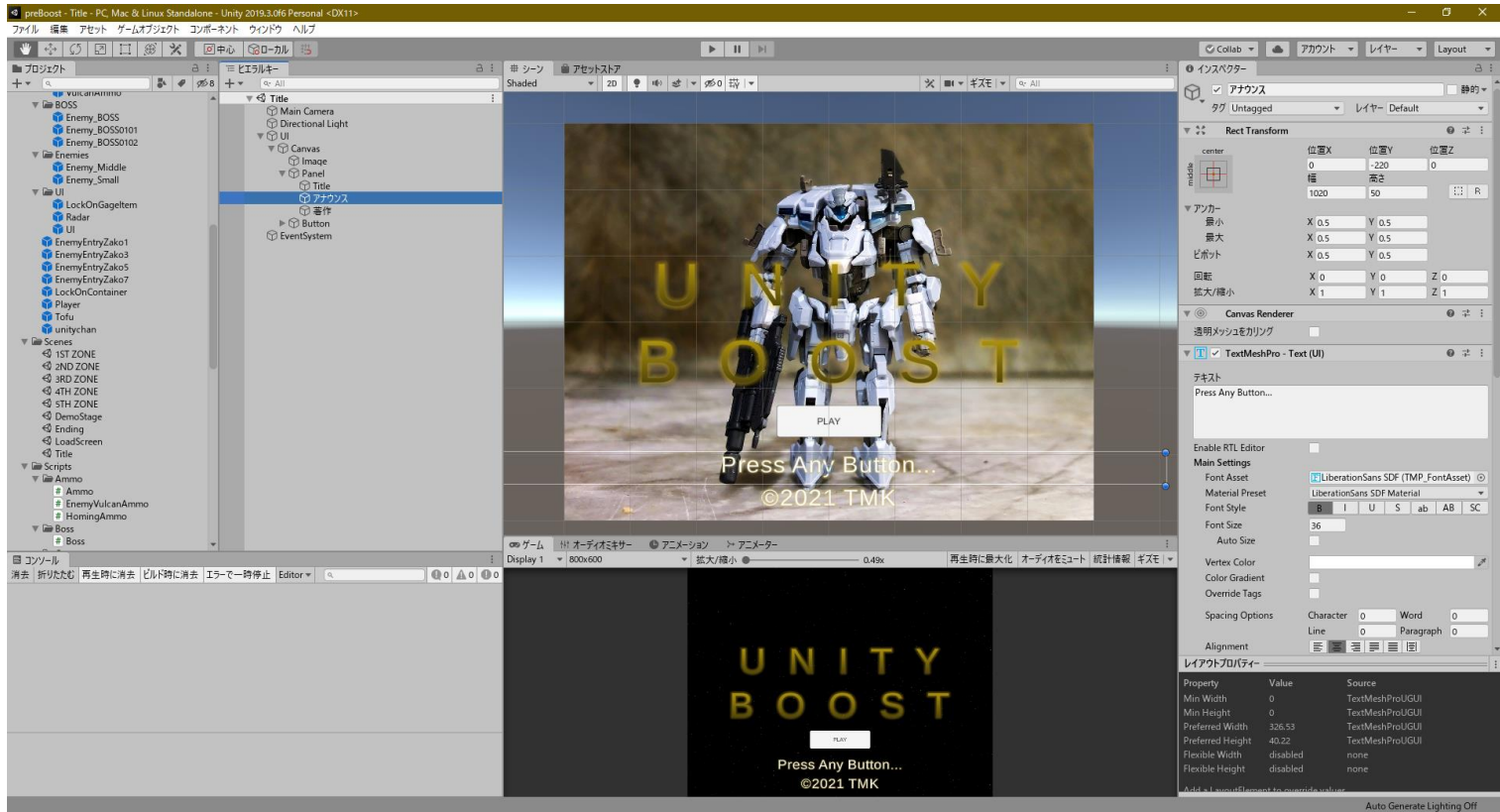


図2-2 プレイシーンの例

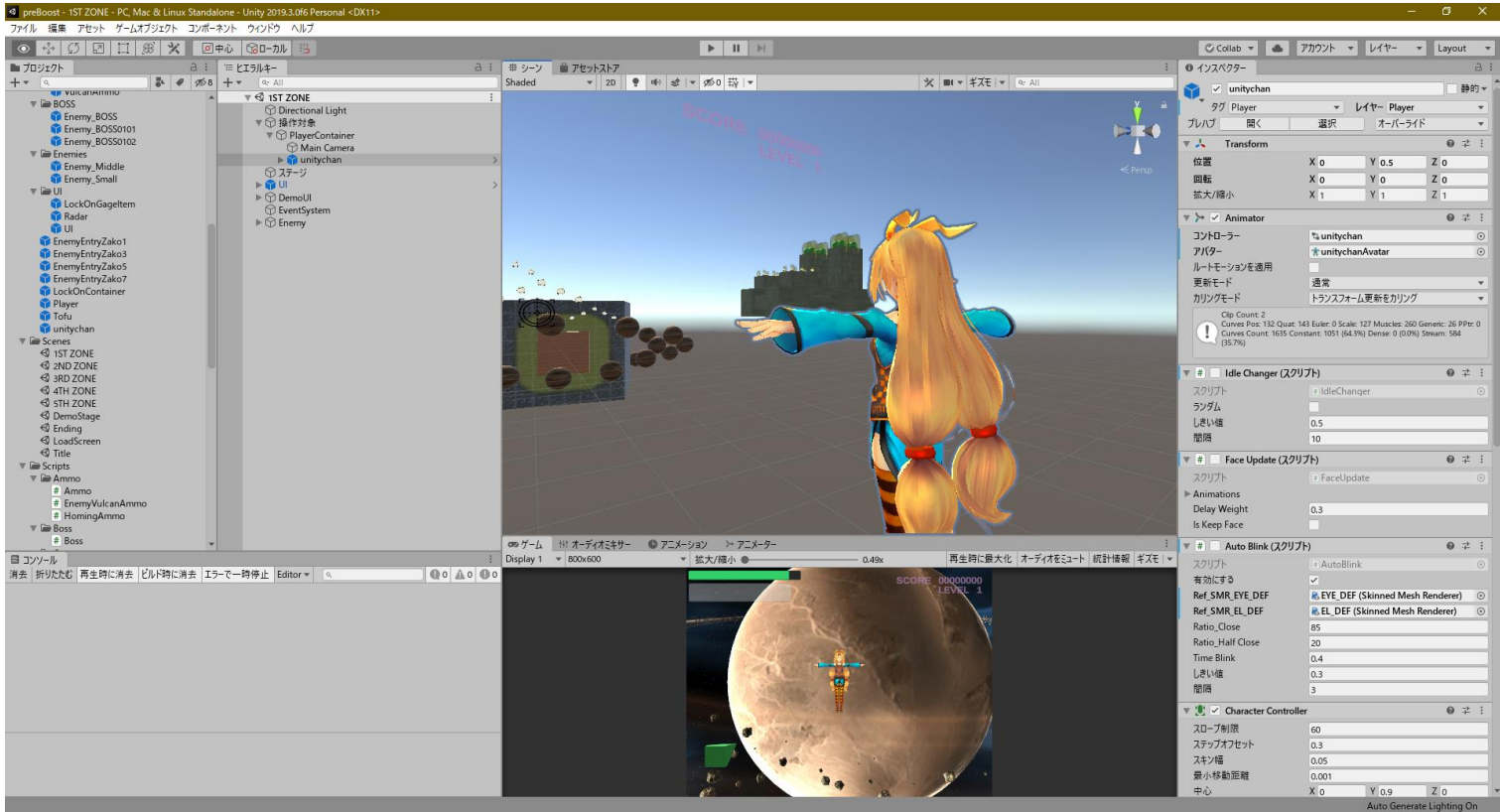
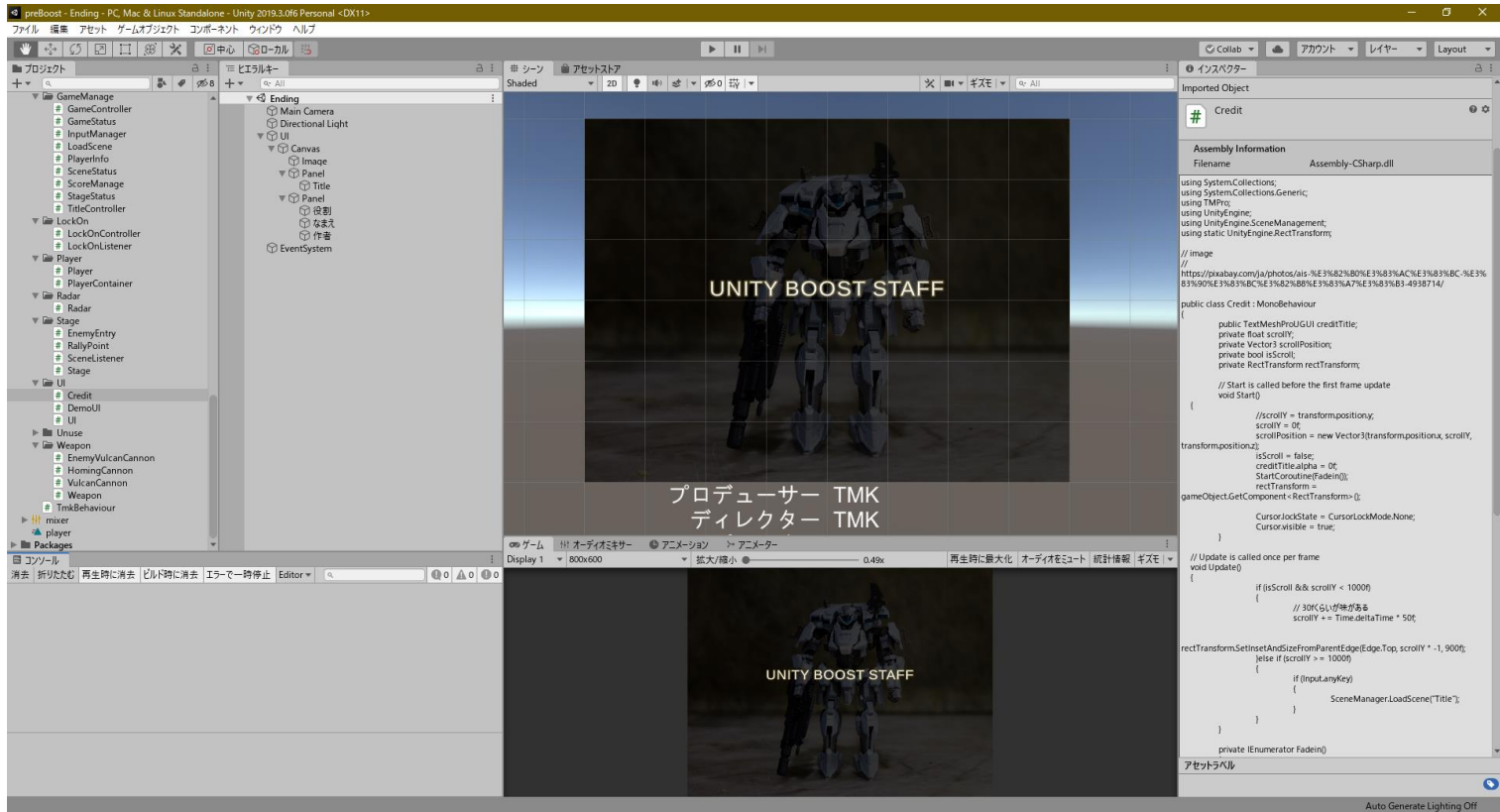


図2-3 クレジットシーンの例

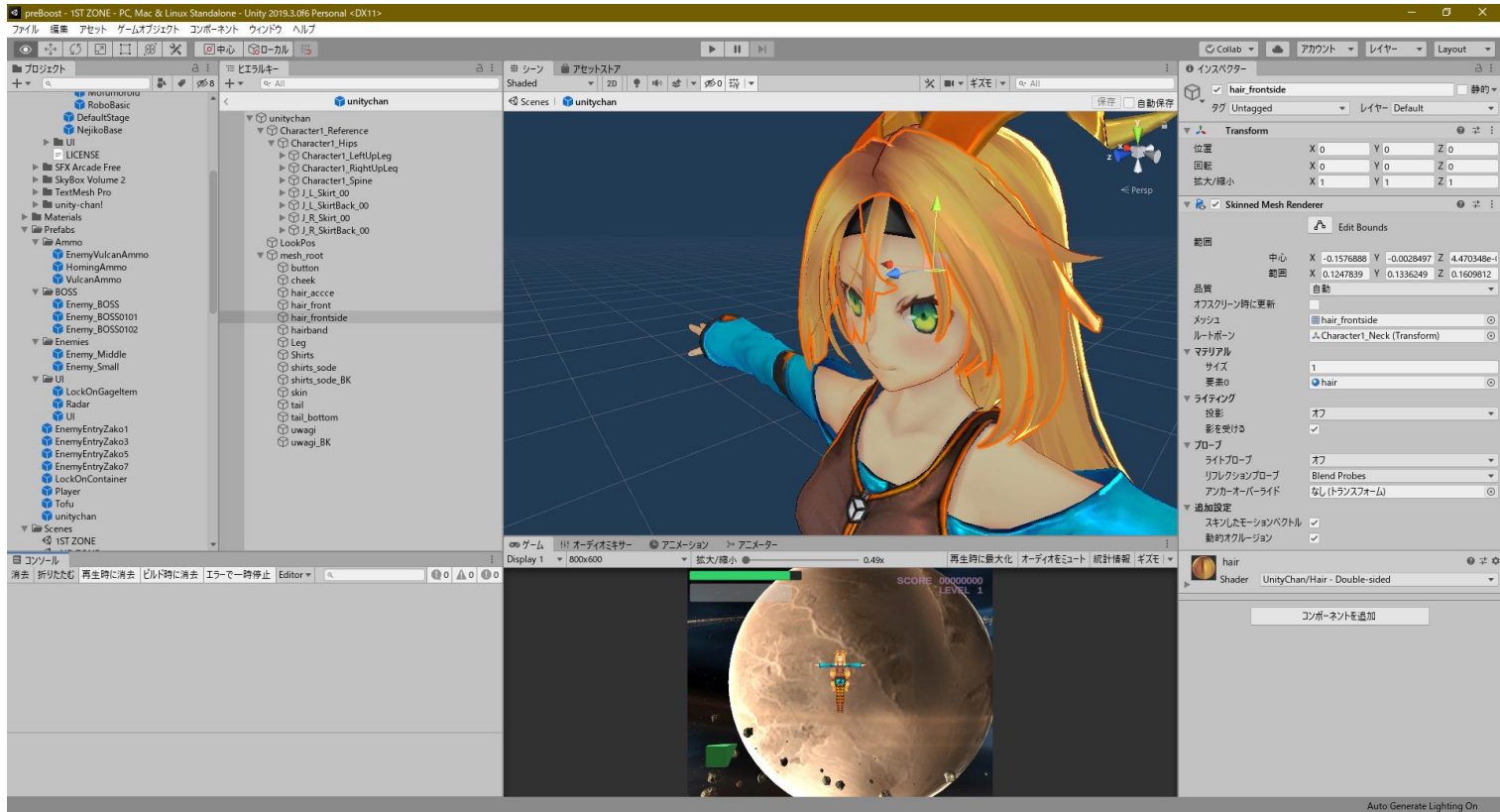


## アセット

Asset（アセット）はUnityでゲームを作る素材を指す。アセットはキャラクター3Dデータや建物などの3Dデータ、背景画像、音声・BGMなどがあり、スクリプトもアセットの一部。アセットは無料公開されているものが多数あり、それらを使うことでゲームを作ることができる。アセットはアセットストアで公開されており、利用方法は後程説明する。

図2-4 ユニティちゃんもアセット





## 2-2 Unityエディターの基本

### 画面構成

Unityは機能が豊富で覚えなければならない事がたくさんあるが、まず覚えるべき点はエディターの画面構成と使い方から学ぶ。

図2-5 Unityエディターの画面構成

2. ゲーム  
ビュー

8. アセットストア  
ウィンドウ

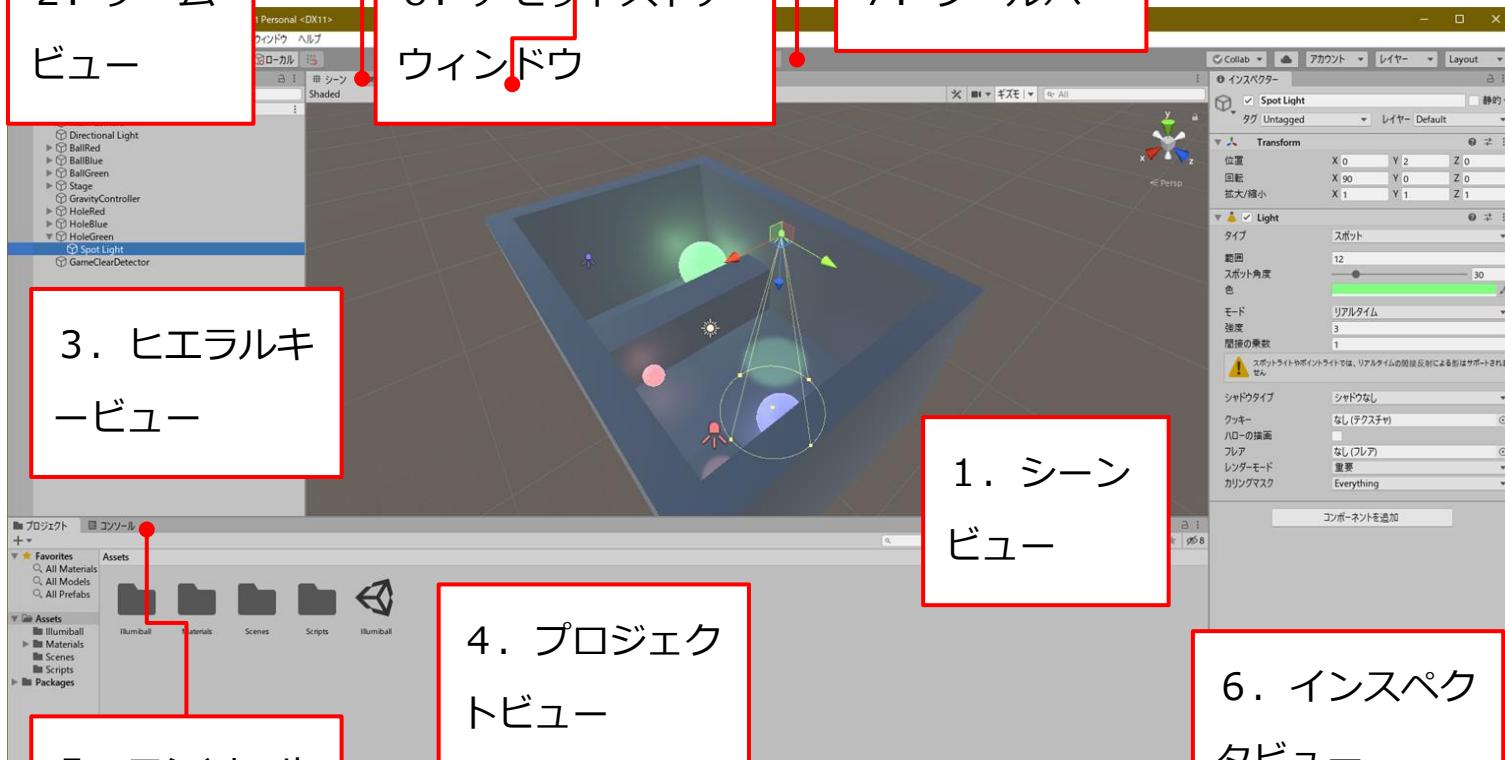
7. ツールバー

3. ヒエラルキ  
ービュー

1. シーン  
ビュー

4. プロジェク  
トビュー

6. インスペク  
タビュー



## 1. シーンビュー

シーンの編集を行うビュー。オブジェクトの配置、位置調整、回転等の調整を行える他に視覚的に全体を把握できる。最も使うであろうビュー。

## 2. ゲームビュー

ゲームのプレビューを行うビュー。作成したゲームがどのように動くか確認することができる。

### 3. ヒエラルキービュー

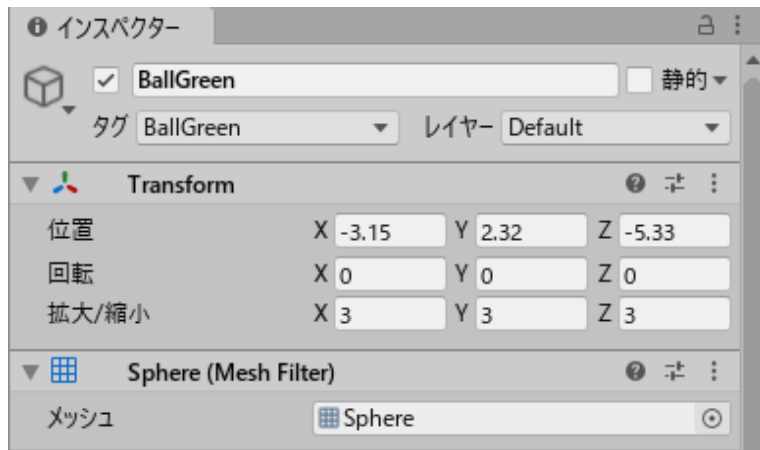
アクティブなシーンのオブジェクトを一覧表示するビュー。オブジェクトの階層状態を確認することができる。

## 4. プロジェクトビュー

プロジェクト全体にインポートされているアセットを表示するビュー。  
「Packages」フォルダはUnityパッケージマネージャーが管理しているもので、手を触れることは無い。

## 5. コンソールビュー

エラーメッセージやデバッグメッセージを表示するビュー。スクリプトからメッセージを表示させることもできるので、デバッグの際に役立つ。



## 6. インスペクタビュー

アクティブなオブジェクトの情報を表示するビュー。C#のフォームアプリケーションのプロパティウィンドウに似ている。このインスペクタビューで各オブジェクトの値を微調整することが多い。非常によく利用するビュー。

## 7. ツールバー

オブジェクトの操作方法を切り替えたり、ゲームのプレビュー用のボタン等のよく使う機能がツールとして集まっている。

## 8. アセットストアウィンドウ

様々なアセットが公開されているアセットストアのページを表示するためのウィンドウ。Unity自体が重いので、アセットストアを見るのであればブラウザで見ることを推奨する。





## レイアウトのカスタマイズ

各ビューはタブをドラッグ&ドロップすることで位置を変更し、カスタマイズすることができる。

## Projectビューのレイアウト

プロジェクトビューは表示形式を切り替える（1列レイアウト or 2列レイアウト）ことができる。好みのレイアウトを使えば良い。

## TIPS Projectビューの検索とお気に入り登録

プロジェクトビューは検索機能が付いているので、アセットが増えても目的のアセットを探しやすい。2列レイアウトであれば、検索結果を保存しておくこともできる。

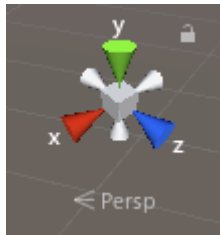
## Gameビューのアスペクト比の変更

ゲームビューではアスペクト比（画面比率）を変更することができる。作るゲームのアスペクト比が決まっている場合は設定するといだらう。そうでないなら「Free Aspect」が良い。

## Sceneビューの操作

シーンビューは配置状態や視覚的効果を映し出すだけでなく、オブジェクトの配置を調整したり、視点を変更したりできる。

## シーングズモと投影モード



シーンビューの右上に表示されているのが**シーングズモ**。X、Y、Z軸の方向が表されており、色が付いている部分をクリックするとその軸に視点が移る。

シーングズモの下部には**投影モード**が表示されている。投影モードは**Perspectiveモード**と**Isometricモード**の2種類があり、Perspectiveモードは奥行きを考慮したパース（俗に言う遠近法）がかかり、Isometricモードは奥行きを考慮したパースがかからない様になっている。パースの詳細については以下リンクを参照。

[パースとは？ 漫画を描くために必要な知識](#)

## マウスとキー操作による視点の変更

マウスとキー操作で視点を変更することができる。

視点移動

中ボタンドラッグ

視点回転

右ボタンドラッグ

ズームIN/OUT      ホイール前後移動

フォーカス      ヒエラルキー項目をダブルクリック  
Shift + F

## 2-3 3Dシーンを組み立てる

### 新規プロジェクトの作成

Unityはゲームをプロジェクト単位で管理する。新規プロジェクト作成はUnity Hubの**新規作成**から行う。

※講義では事前に用意したHelloUnityリポジトリを使用する。

複数のUnityバージョンをインストールしている場合、ターゲットとなるバージョンを選ぶ必要がある。バージョンは新規作成ボタンの右にあるコンボボックスから選ぶことができる。

図2-6 複数Unityバージョンがある場合は選択する



tm

プロジェクト

使い方を学ぶ

コミュニティ

インストール

## プロジェクト

プロジェクト名	Unity バージョン	ターゲット			
<b>87713c9ad304e044893f1dedb78d...</b> C:\Users\ten\AppData\Local\Temp\87... Unity バージョン: 2020.3.17f1	2020.3.17f1 ▼	使用中のプ...			
<b>65262cac4716aa74ea893264b01...</b> C:\Users\ten\AppData\Local\Temp\65... Unity バージョン: 2020.3.17f1	2020.3.17f1 ▼	使用中のプラッ...	▼	2 days ago	⋮
<b>NejikoRun</b> E:\PG\unity\unity_chapter5\NejikoRun Unity バージョン: 2019.3.0f6	2019.3.0f6 ▼	使用中のプラッ...	▼	a day ago	⋮
<b>tmkForce</b> E:\PG\unity\tmkForce\tmkForce Unity バージョン: 2019.3.15f1	2019.3.15f1 ▼	使用中のプラッ...	▼	a day ago	⋮

2020.3.17f1

2019.3.15f1

2019.3.0f6

2019.2.11f1

プロジェクト名に**HelloUnity**と入力し、保存先に**ローカルリポジトリのパス**を指定する。**作成**ボタンをクリック。

図2-7 プロジェクトを新規作成

## テンプレート



2D



アップグレ...



3D



3D With Extras



High Definition RP

## 設定

プロジェクト名 \*

HelloUnity

保存先 \*

E:\PG\kunren\Unity\HelloUnity



キャンセル

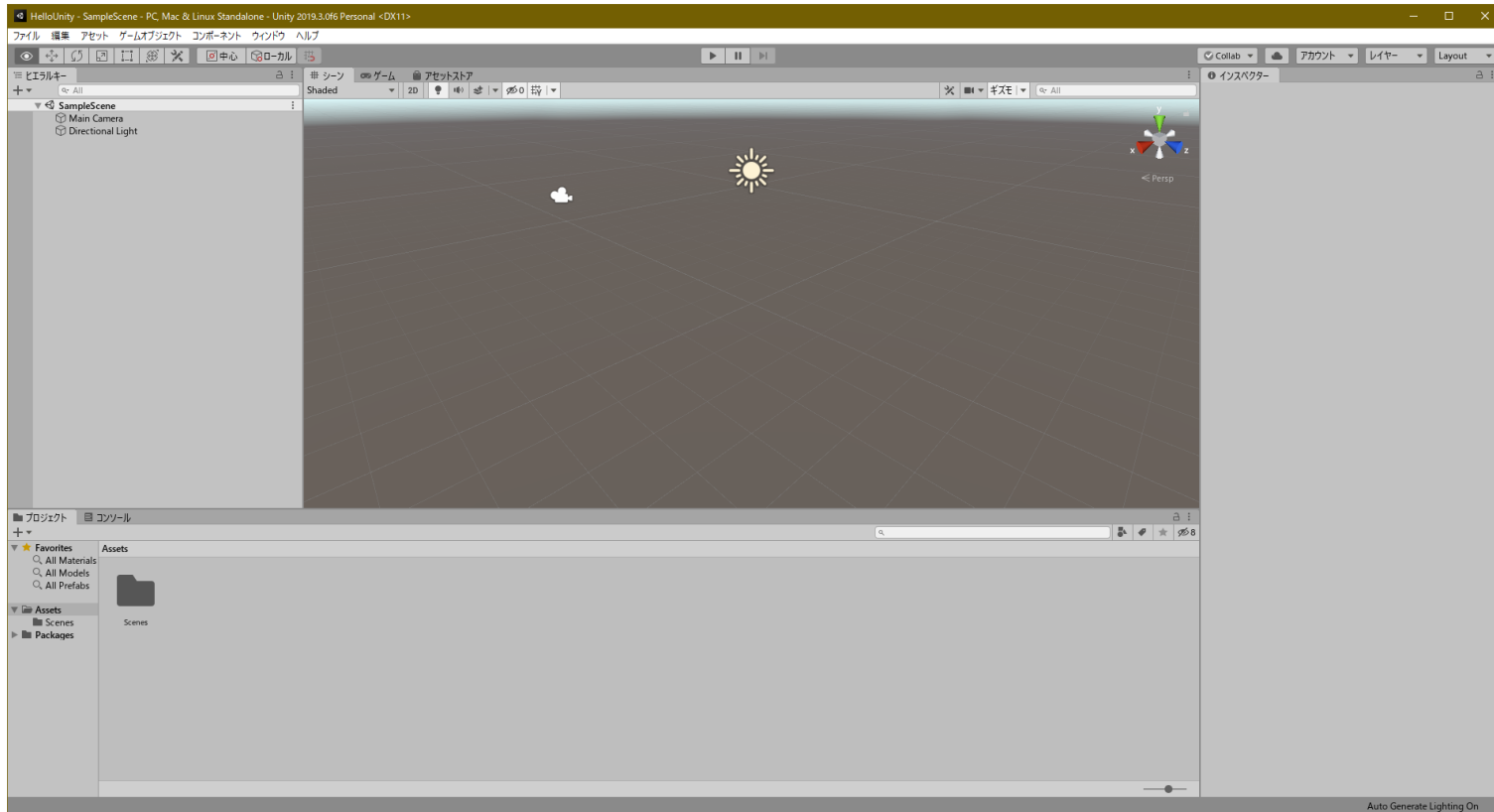
作成

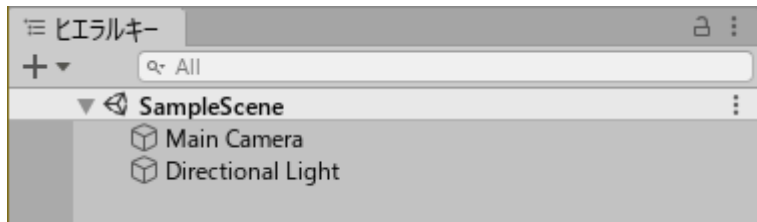


## デフォルトシーンの中身と3D空間の基本

プロジェクトを新規作成するとUnityエディタが起動する。

図2-8 プロジェクト新規作成時のUnityエディタ





ヒエラルキーを見ると**SampleScene**という名前のシーンが追加されている。SampleSceneはシーンビューに表示されている。シーンビューに表示されている項目は次の通り。



## Main Camera（メインカメラ）

ゲームを再生した時に表示される内容を撮影しているカメラ。カメラから見た映像をディスプレイや画像ファイルに映し出す処理を**レンダリング**と呼ぶ。



## Directional Light（ディレクショナルライト）

太陽光の様なもので、一定方向へ光を提供する。



## Skybox（スカイボックス）

シーンの背景。デフォルトでは空の様な描画になっている。

## カメラ・ライトのギズモアイコン

メインカメラやディレクショナルライトは形を持たない（オブジェクトが描画されない）ため、シーン上ではアイコンで表示される。このアイコンを**ギズモアイコン**と呼ぶ。

## 3Dオブジェクトの生成と配置

3Dオブジェクトの生成方法と配置・調整について学ぶ。

### 3Dオブジェクトの配置方法

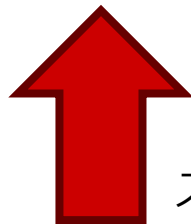
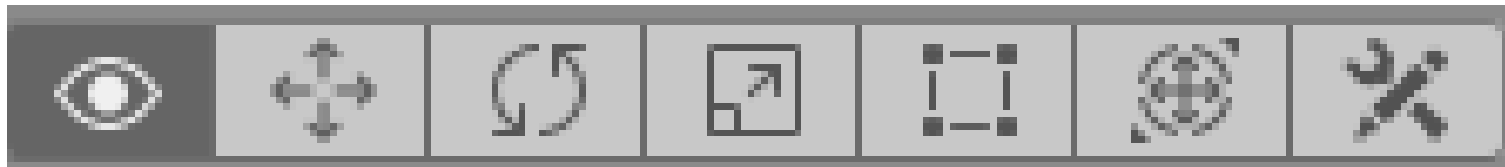
ヒエラルキーの+ボタン、または右クリック > 3Dオブジェクト > キューブ

キューブ以外にスフィア、カプセル等がある。配置後はヒエラルキーにオブジェクトが追加される。追加されたオブジェクトはF2で名前変更可能。

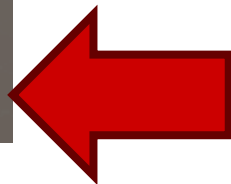
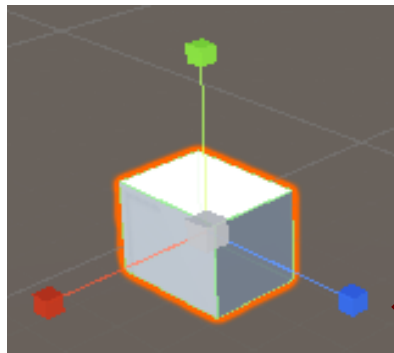
### 3Dオブジェクトのサイズ変更

ツールバーの**スケールツール**を選択後、オブジェクトの**ハンドル**（各軸についた色つきの四角形）をマウス左ボタンドラッグでサイズ変更可能。インスペクタビューでは直接値を指定して変更可能。

図2-9 スケールツールとハンドル

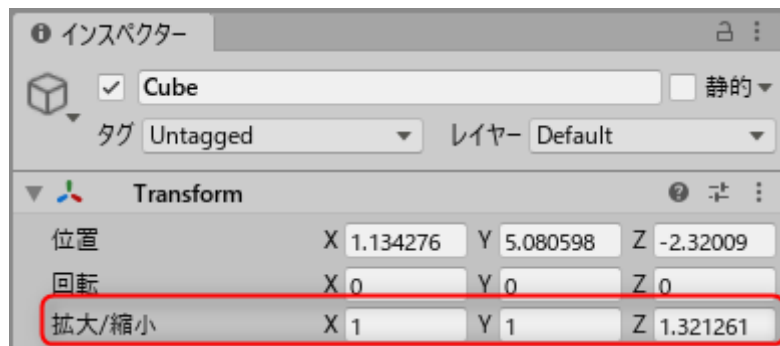


スケールツール



ハンドル

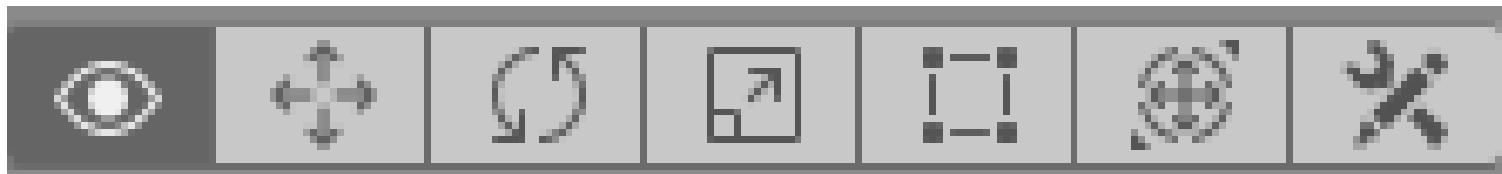
図2-10 インスペクタビューでサイズ変更



### 3Dオブジェクトの移動

ツールバーの**移動ツール**を選択後、オブジェクトの**ハンドル**（各軸についた色つきの円錐）をマウス左ボタンドラッグで移動可能。  
インスペクタビューでは直接値を指定して変更可能。

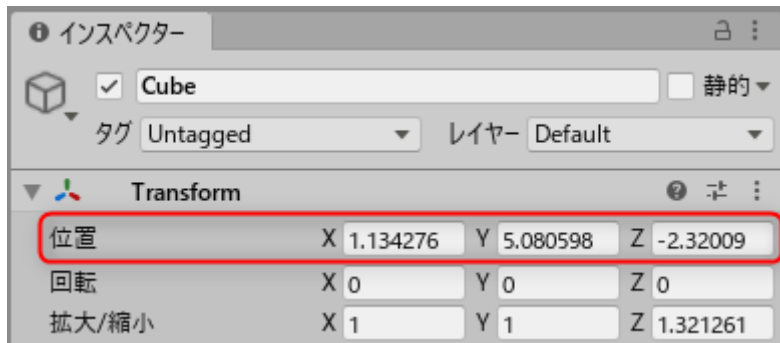
図2-11 移動ツール



移動ツール

図2-12 インспекタビューで移動

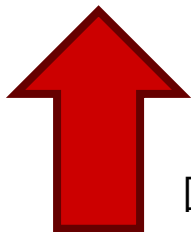
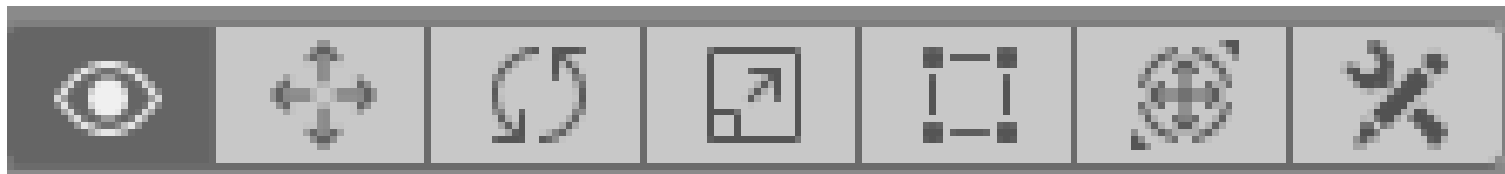




### 3Dオブジェクトの回転

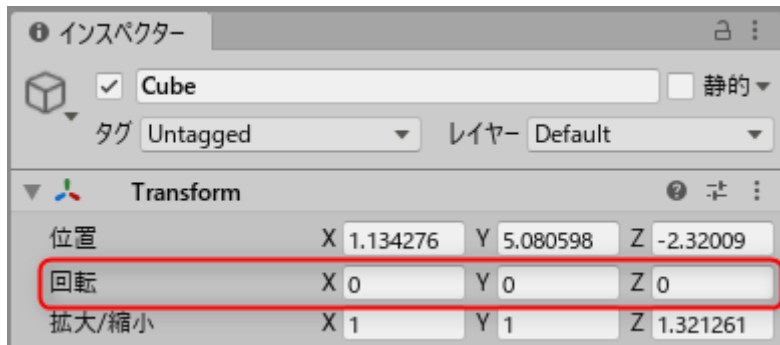
ツールバーの**回転ツール**を選択後、オブジェクトの**ハンドル**（各軸についた色つきの円弧）をマウス左ボタンドラッグで回転可能。  
インスペクタビューでは直接値を指定して変更可能。

図2-13 回転ツール



回転ツール

図2-14 インспекタビューで回転



## カメラの調整

カメラが映し出す映像がゲーム再生時の映像になる。カメラの位置を変更し、見やすい位置へカメラを移動する。

## TIPS プレビュー中のオブジェクトの操作と確認

プレビュー中（ゲーム再生中）でもシーンやインスペクタからオブジェクトを移動・拡大などの操作が可能。しかし、**プレビュー終了後に変更した値は元に戻る**ため注意が必要。

## シーンの保存

シーンはCtrl+Sで保存。

## TIPS プロジェクト全体の設定の保存

基本的にUnityエディターを終了する時に保存されるが、以下の手順でも保存可能。

ファイル > プロジェクトを保存

## COLUMN マルチシーンエディティングの活用

複数のシーンを同時に編集することができる。規模がかなり大きくなってシーン間で連携が必要になった時に試してみると良い。  
講義ではマルチシーンエディティングを行うことは無い。

## 2-4 色と質感を与える

### シェーダーとマテリアル

光の陰影の計算を行うプログラムを**シェーダー**と呼ぶ。シェーダーの直接的な調整は通常行わない。

どのような材質でできているかを表す**マテリアル**と呼ばれるものを使ってシェーダーを指定し、質感を出す。

## 物理ベースシェーダー

現実世界の物理特性を再現したシェーダーを**物理ベースシェーダー**と呼ぶ。シェーダーには様々な種類があり、例えばアニメの様な質感を出すトゥーンシェーダー等がある。講義では物理ベースシェーダー（Unityエディター上では**Standard**シェーダーと表現される）を標準として扱う。

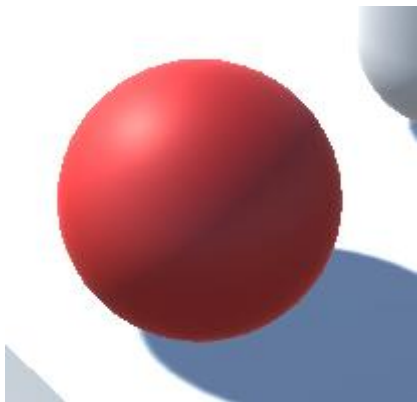
## マテリアルの設定とオブジェクトの色の変更

マテリアルはプロジェクトビュー上で以下の操作で作成できる。

右クリック > 作成 > マテリアル

作成したマテリアルはStandardシェーダーの場合、インスペクタービューの**アルベド**項目で色の指定ができる。設定したマテリアルは各オブジェクトのインスペクタの項目**マテリアル**の**要素0**にドラッグ&ドロップすることで反映できる。シーン上のオブジェクトに直接ドラッグ&ドロップしても適用することができる。

図2-15 マテリアルが適用されたスフィアオブジェクト



### **TIPS** マテリアルの共有

マテリアルは複数のオブジェクトに設定することができ、マテリアルの設定を変更すると設定しているオブジェクト全てに影響が出る。

## フォルダの作成とアセットの整理

プロジェクトビューでフォルダを作成することができる。アセット別にフォルダを作って管理すると良い。

## オブジェクトの質感の変更

Standardシェーダーではマテリアルの項目に**Metallic**と**Smoothness**があり、それぞれ金属らしさと表面の滑らかさを表す。値を変更すると金属のような質感、ゴムのような質感を出すことができる。

### COLUMN 周囲の映り込みとReflection Probe

周囲のオブジェクトを映り込む様にするにはリフレクションプローブをコンポーネント追加する必要がある。



## 2-5 グローバルイルミネーションを体感する

物体（オブジェクト）を光で照らす処理を**ライティング**と呼ぶ。

### 古典的なライティング手法

**直接光**と**環境光**を使ってライティングする。

#### 直接光

ライト等から放出された光のこと。オブジェクトの表面を照らす。

#### 環境光

反射によって無限に拡散された光のこと。無限にライティング処理することは不可能であるので、代わりとして僅かな明かりをすべての場所に与える。

## GIによるライティング手法

直接光と環境光に加えて**間接光**も考慮してライティング処理を行う技術がグローバルイルミネーション（GI）。間接光は物体から反射した光を再現する。デフォルト設定ではグローバルイルミネーションが有効になっていない。

### COLUMN 背景を光源にするImage Based Lighting

Image Based Lightingライティング手法により、スカイボックスの画像色がオブジェクトに反映される。スカイボックスは以下の手順で変更可能。

メニューのウィンドウ > レンダリング > ライティング設定 > スカイボックスマテリアル

## GI有効化の下準備

(特記事項なし)

## 環境光の無効化と影響の確認

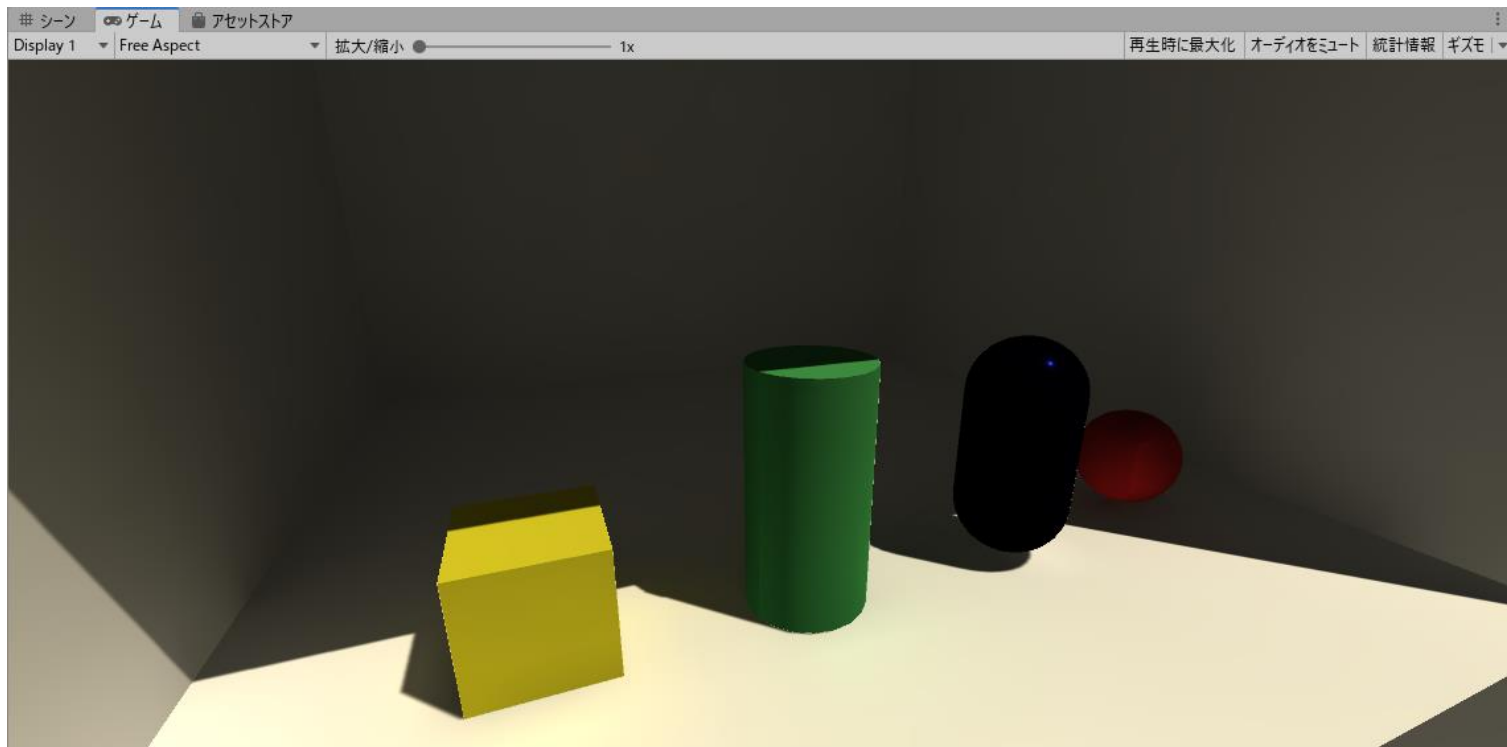
環境光はライティング設定の**環境ライティング**、**環境リフレクション**の**強度の乗数**をゼロにすることで無効化することができる。

## 間接光の有効化

**間接光はスタティック（静的）なオブジェクトにのみ反映される。**

[静的ゲームオブジェクト - Unity マニュアル](#)

図2-16 間接光を有効化したシーンのゲームビュー



## 2-6 初めてのスクリプトを実行する

Hello World的なスクリプトを作成する。

### スクリプト作成とMonoBehaviourクラス

プロジェクトビューのCreateメニュー（+マーク）から**C# スクリプト**を選択して新規スクリプトを作成する。作成したスクリプトをダブルクリックもしくはエンターキーを押すとVisual Studioが起動する。

スクリプトを開くと**MonoBehaviour**クラスを継承していることがわかる。MonoBehaviourクラスはUnityの根幹を担うクラスで基本的には継承してクラスを作成する。

スクリプトには最初から**Start**メソッドと**Update**メソッドが定義されているが、使わない場合は削除しても良い（パフォーマンスの観点からは使っていないのならば削除したほうが良い）。サンプル程度であれば、そのまま残しておいても問題は起きない。

## スクリプト実装とアタッチ

ログを出力するには**Debug.Log**メソッドを使う。

[Debug-Log - Unity スクリプトリファレンス](#)

スクリプト作成後はゲームオブジェクトに**アタッチ**する必要がある。アタッチの方法は2通りある。

1. 対象のゲームオブジェクトを選択し、スクリプトをインスペクタービューにドラッグ&ドロップ
2. インスペクタービューから**コンポーネントを追加**ボタンをクリック > スクリプト > 対象のスクリプトを選択

## COLUMN フレームを理解する

ゲームが再生される時は1秒間に何回もの内部的な演算と画面描画を行っている。この処理の単位をフレームと呼び、映像があるゲームはフレームの積み重ねで動く。

1フレーム目はアタッチされたスクリプトのStartメソッドが呼び出され、毎フレームUpdateメソッドが呼び出される。

[イベント関数の実行順序 - Unity マニュアル](#)

1秒間に処理されるフレーム数をフレームレートと呼び、スマホゲームで40FPS(Frames Per Second)、PCゲームで60FPS程度必要とされている。これはアクション性や滑らかさを重視するかどうかで若干変動する。

## まとめ Unityの画面構成や基本操作を覚える

- Unityエディタの基本的な使い方（ゲームオブジェクトの配置、移動、拡大縮小、回転）
- 物理ベースシェーダーの基本的な利用方法
- 直接光、環境光、間接光（グローバルイルミネーション）
- スクリプト作成、アタッチの方法