

## Unity によるプログラミング

### Unity のインストール

ここでは、無料で使える Personal エディションを紹介する。まず UnityHUB をインストールする  
公式サイト の Download ページ

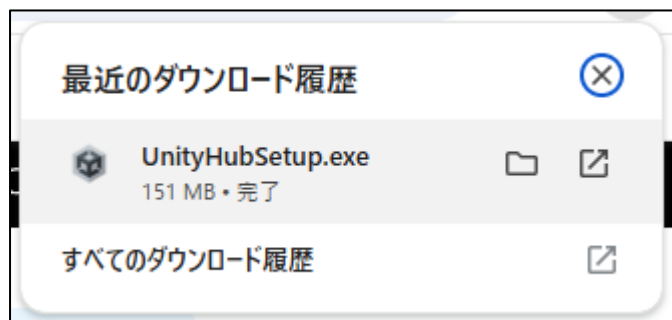
<https://unity.com/ja/download>



→ WINDOWS 用ダウンロード

をクリックする

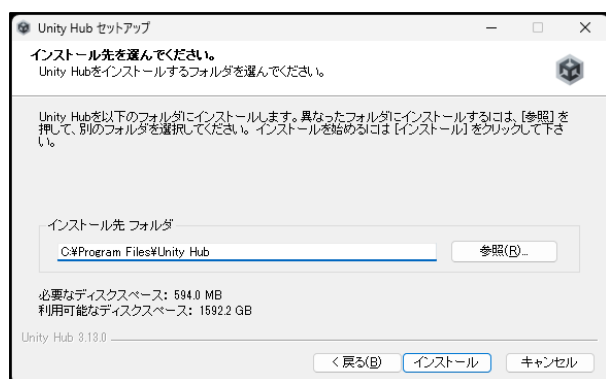
ダウンロードされたファイルをクリックする



同意するをクリックする

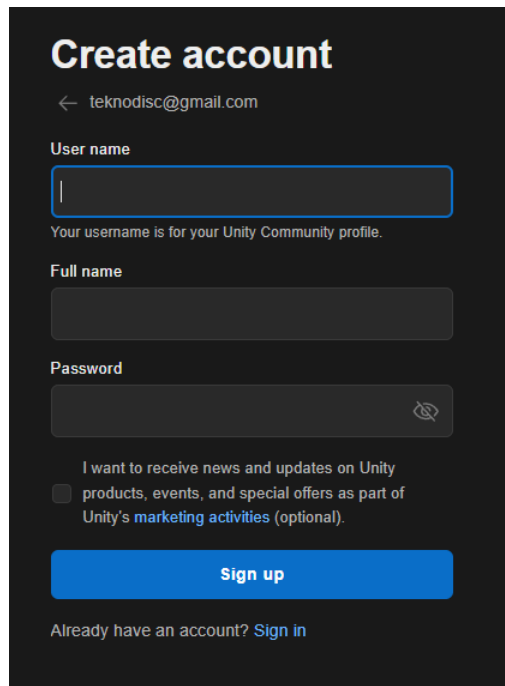
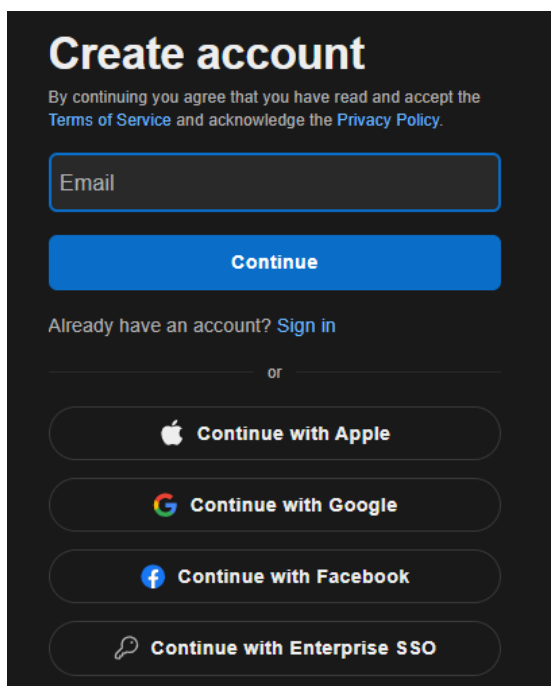
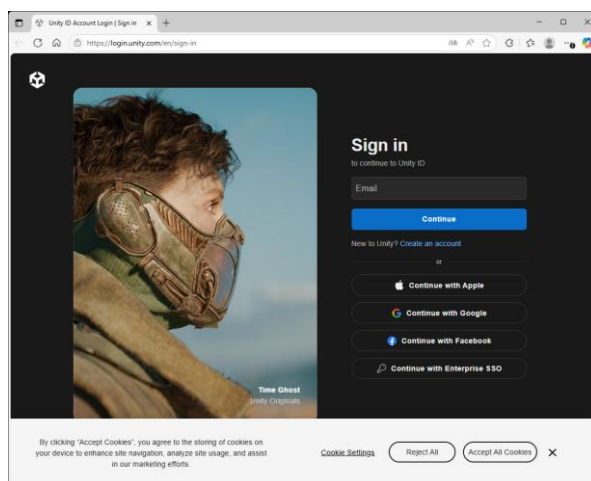
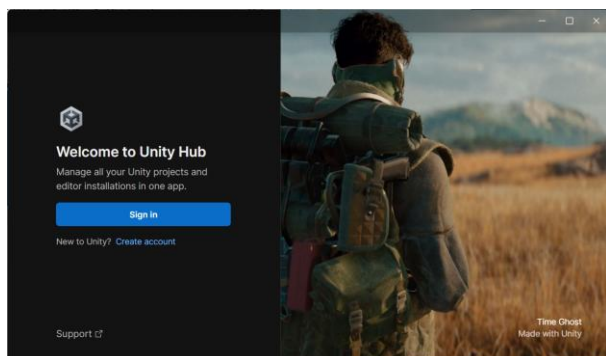


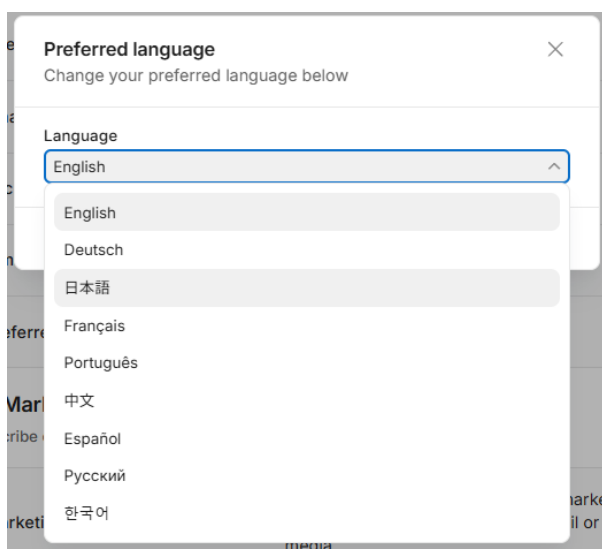
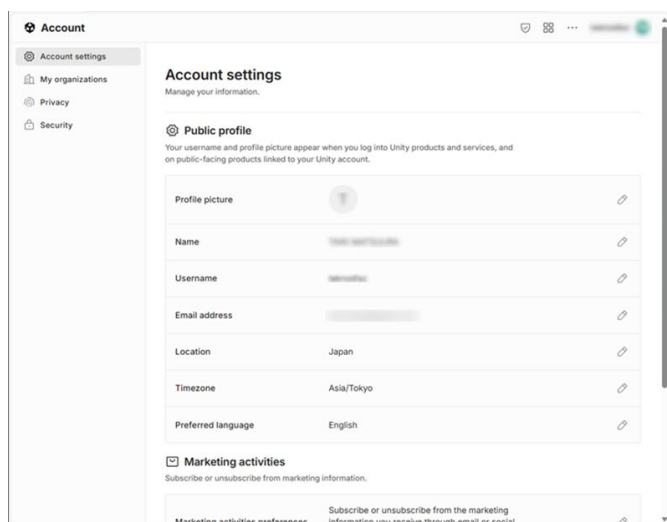
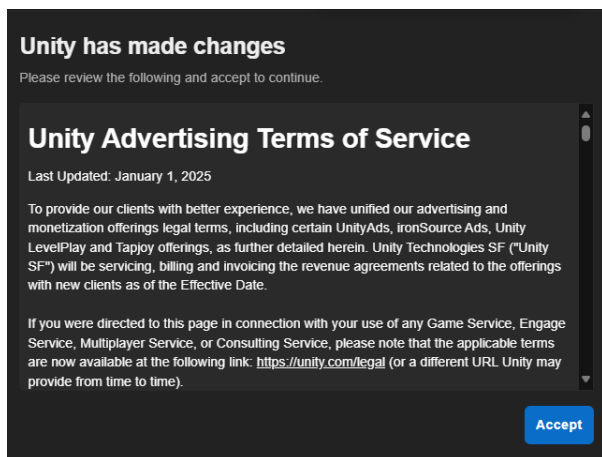
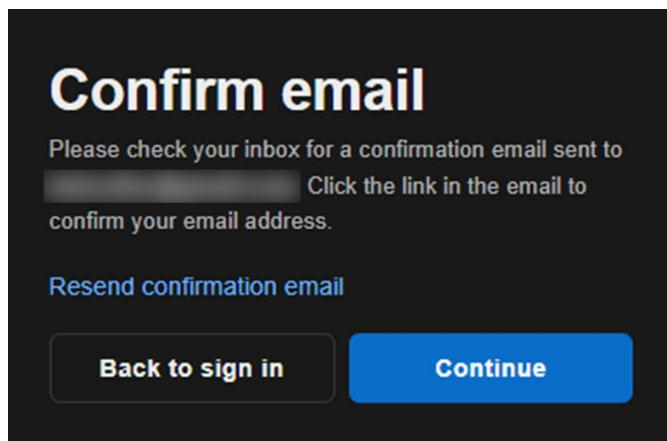
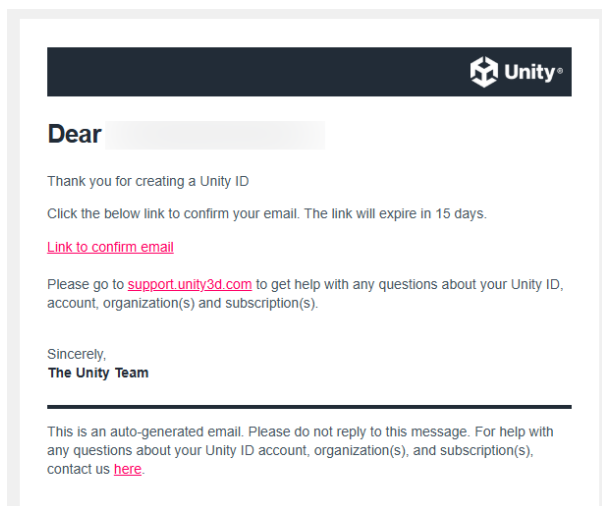
セットアップの指示に従い進めていく。

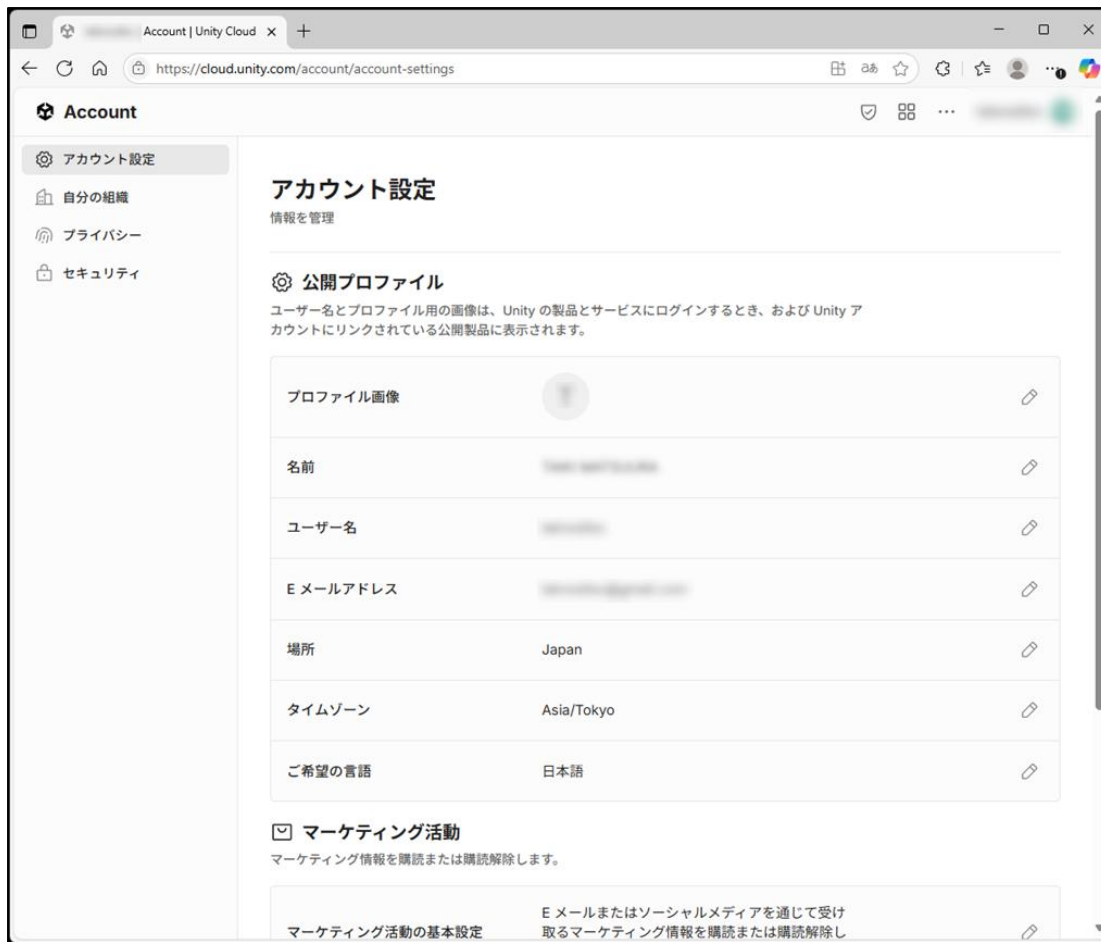


Unity Hub でアカウントの作成

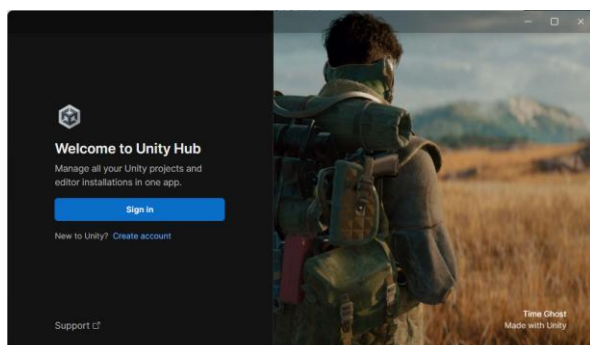
ここでは、e-mail で作成をしている







アカウントの設定をしたら、Unity Hub にサインインする

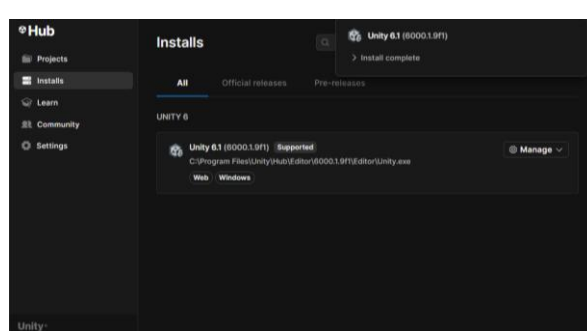
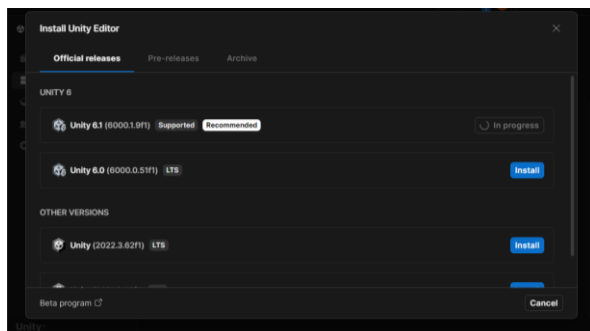
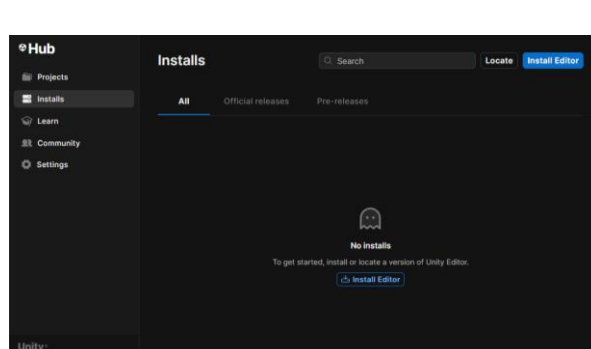
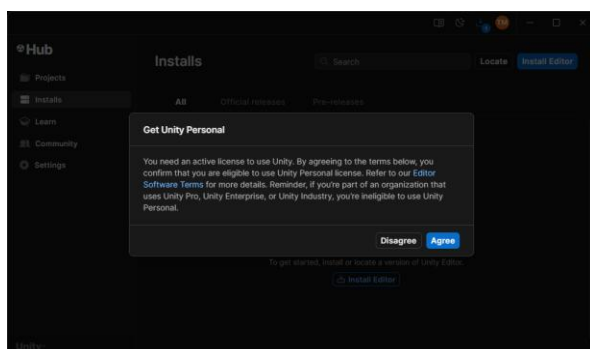
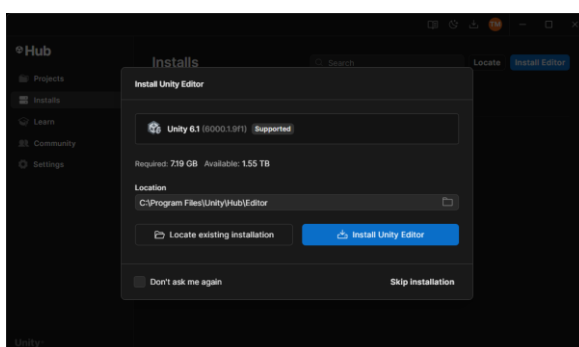
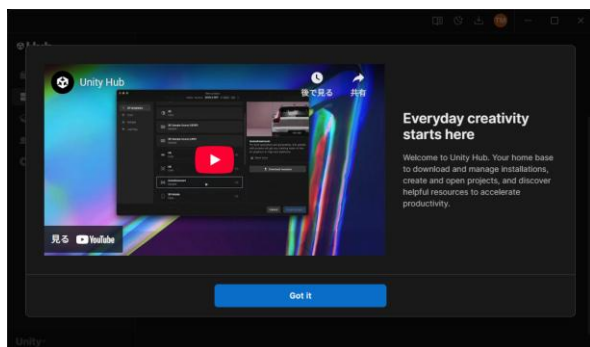


このサイトは、Unity Hub を開こうとしています。  
https://api.unity.com では、このアプリケーションを開くことを要求しています。  
☐ api.unity.com が、関連付けられたアプリでこの種類のリンクを開くことを常に許可する

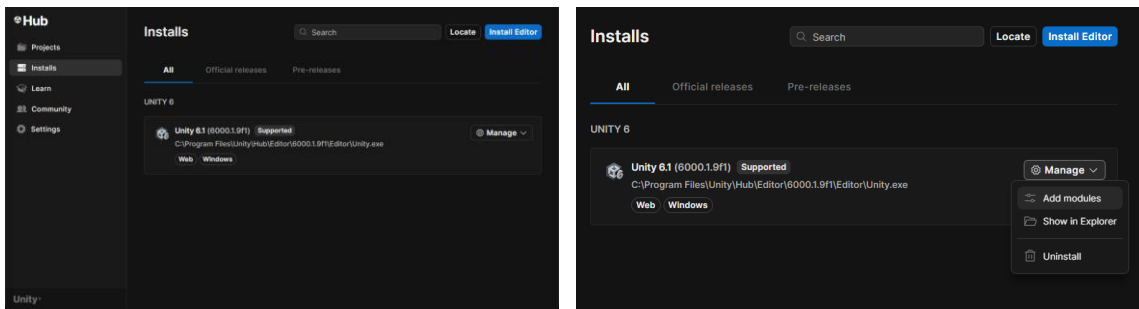
開く キャンセル

## Launching Unity Hub

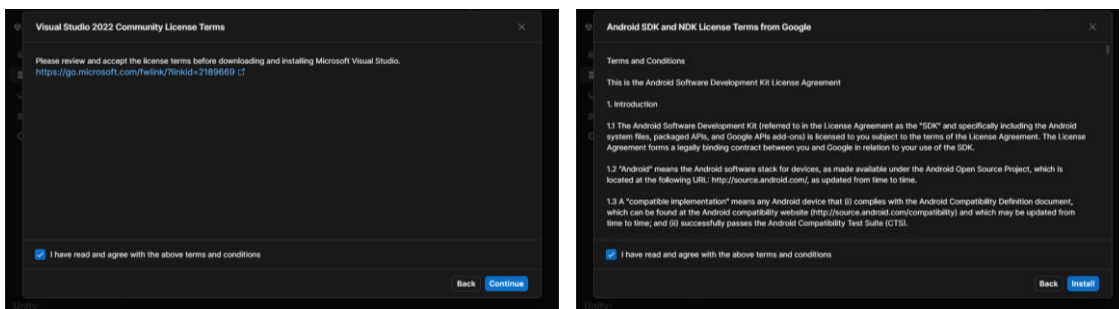
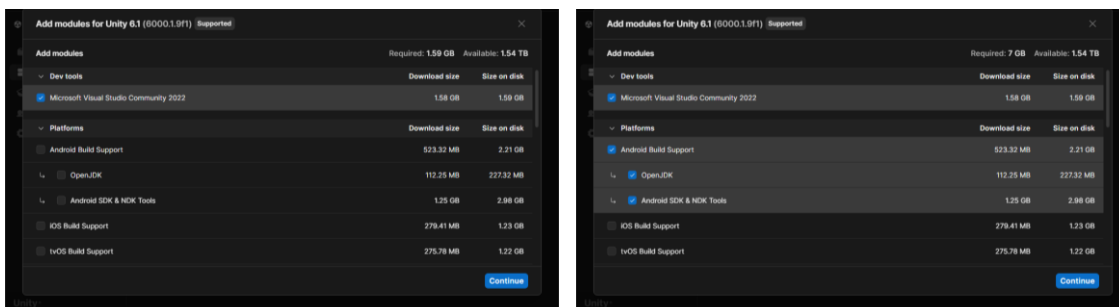
If you are not redirected automatically, follow this [link to login page](#).



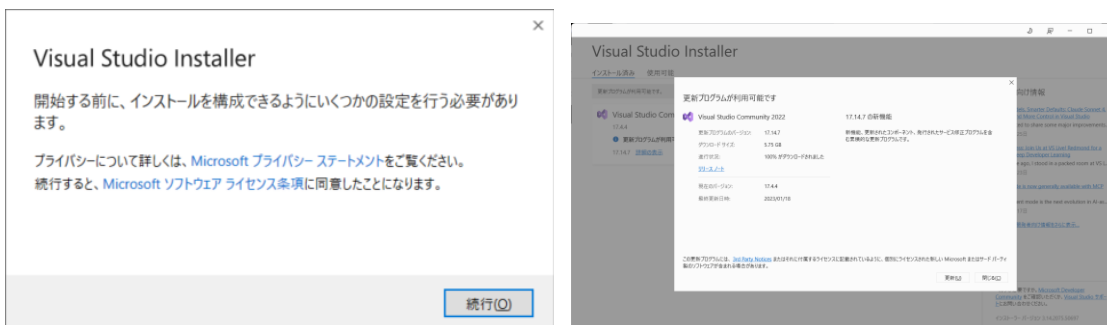
右側の Manage の Add modules を選択して、

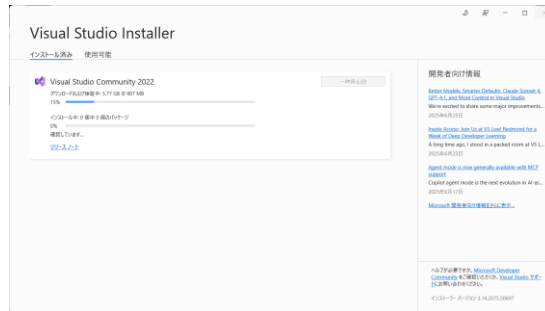


下記のように、Android Build Support OpenJDK Android SDK &NDK Tools を選択して Continueを押して進めていく

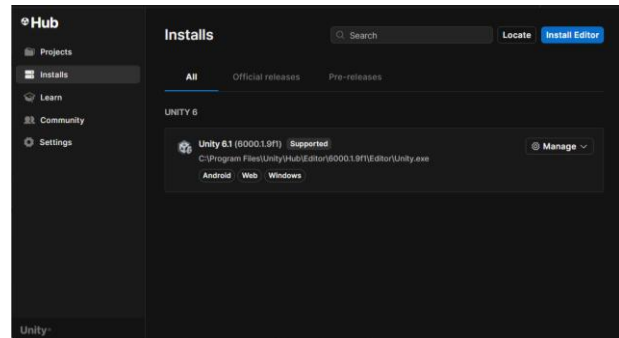
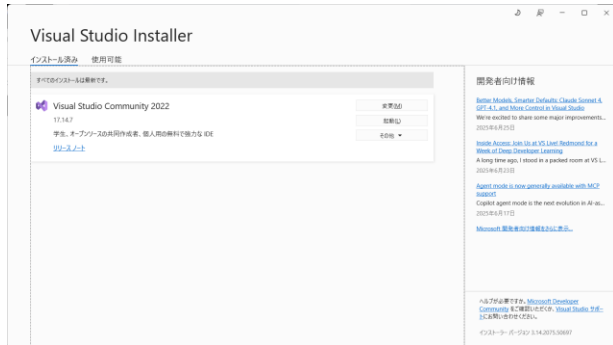


指示に従ってインストールを進めていく。(下記はすでにVisual Studioが入っている状態での画面)

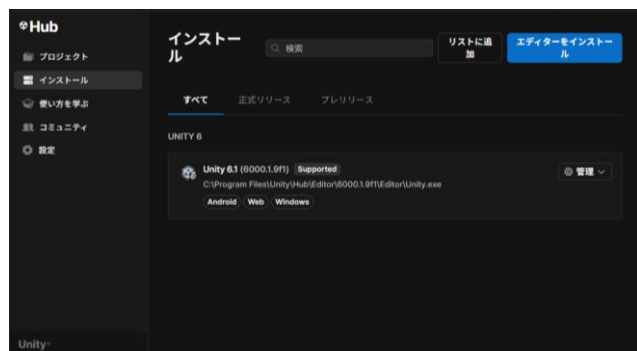
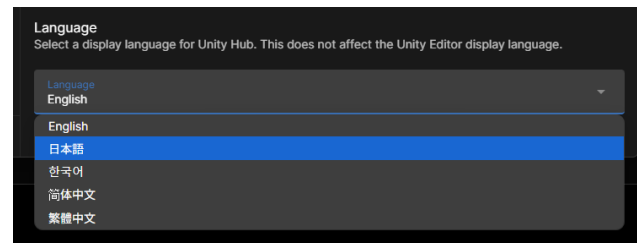
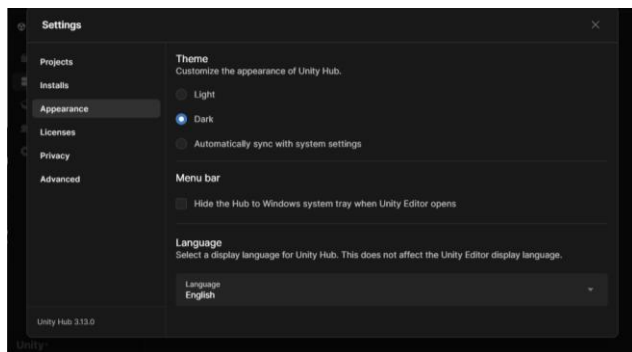




## インストールが完了したら、Unity Hubに戻る



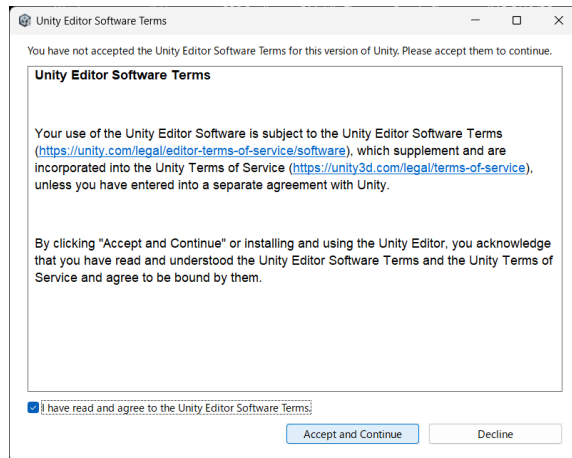
## AppearanceのLanguageを日本語にする



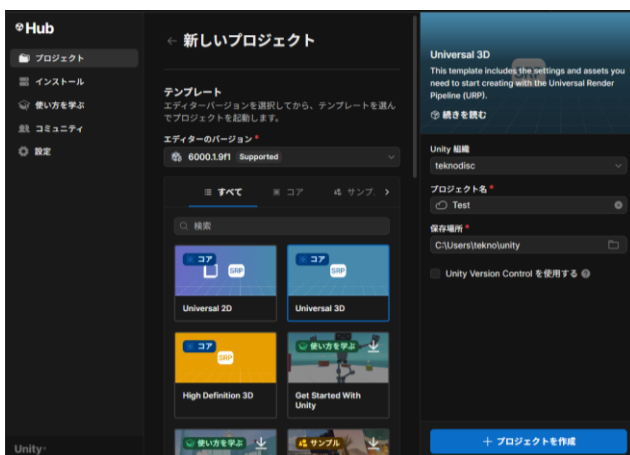
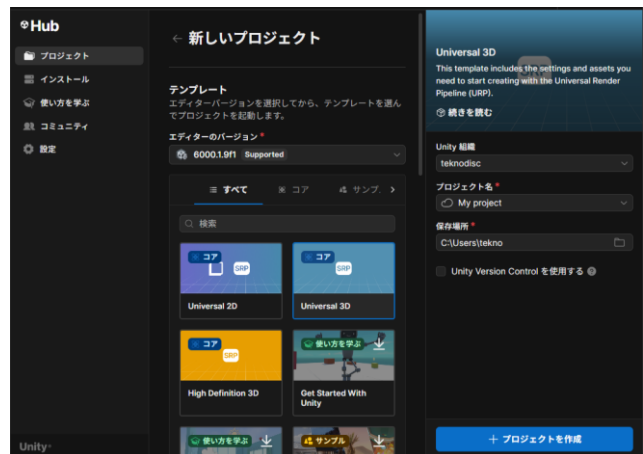


## UNITY の基本操作

### UNITY のアイコンをクリックする

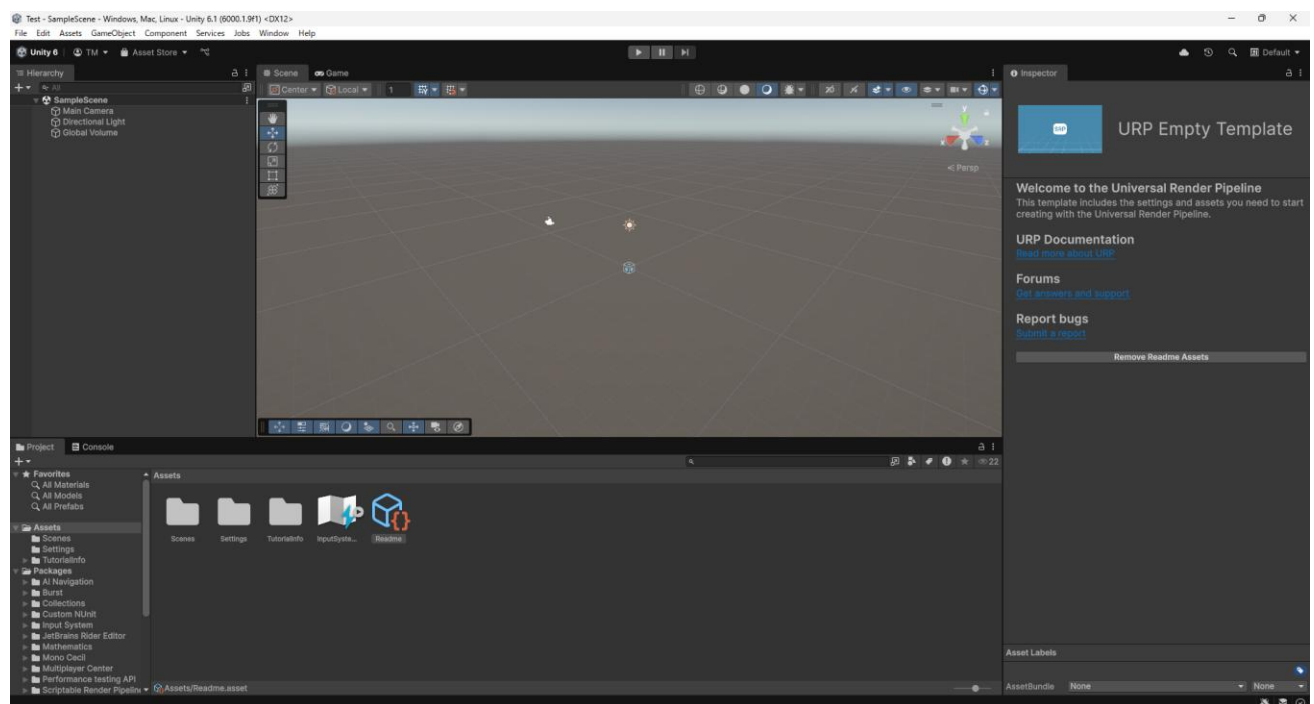


### 新しいプロジェクトボタンを押す

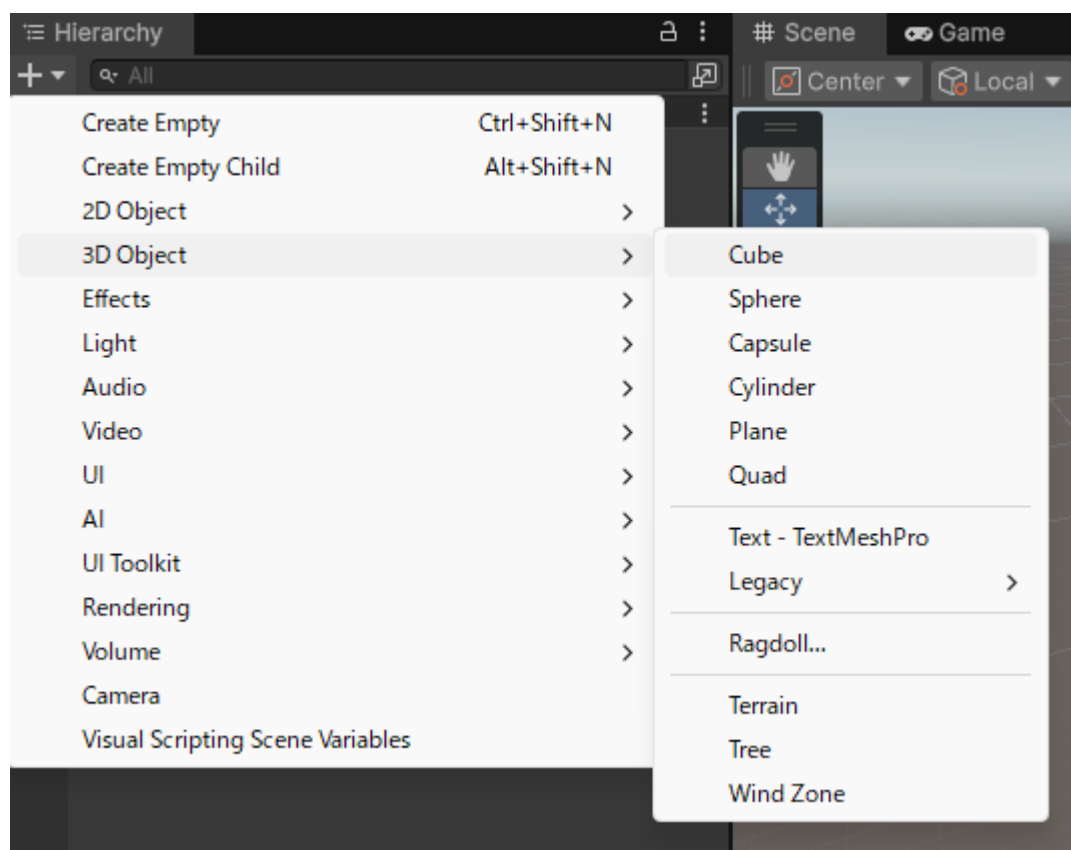


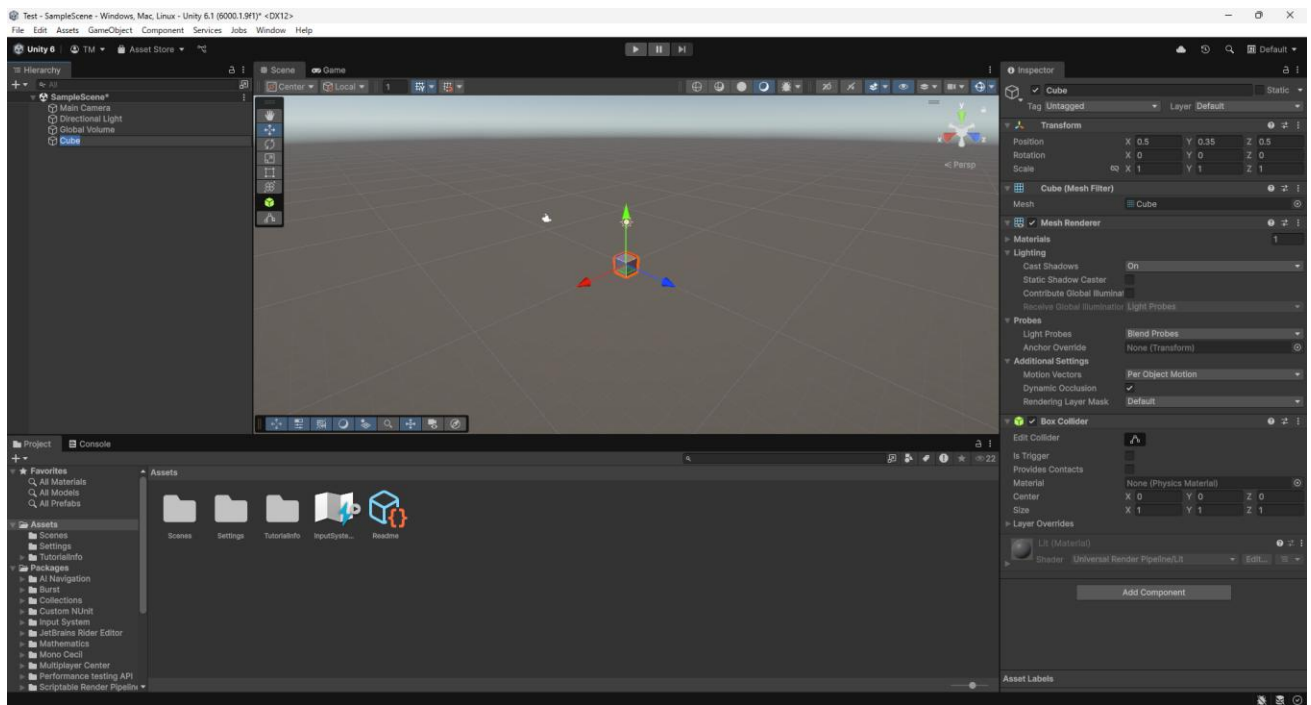


## 作業画面

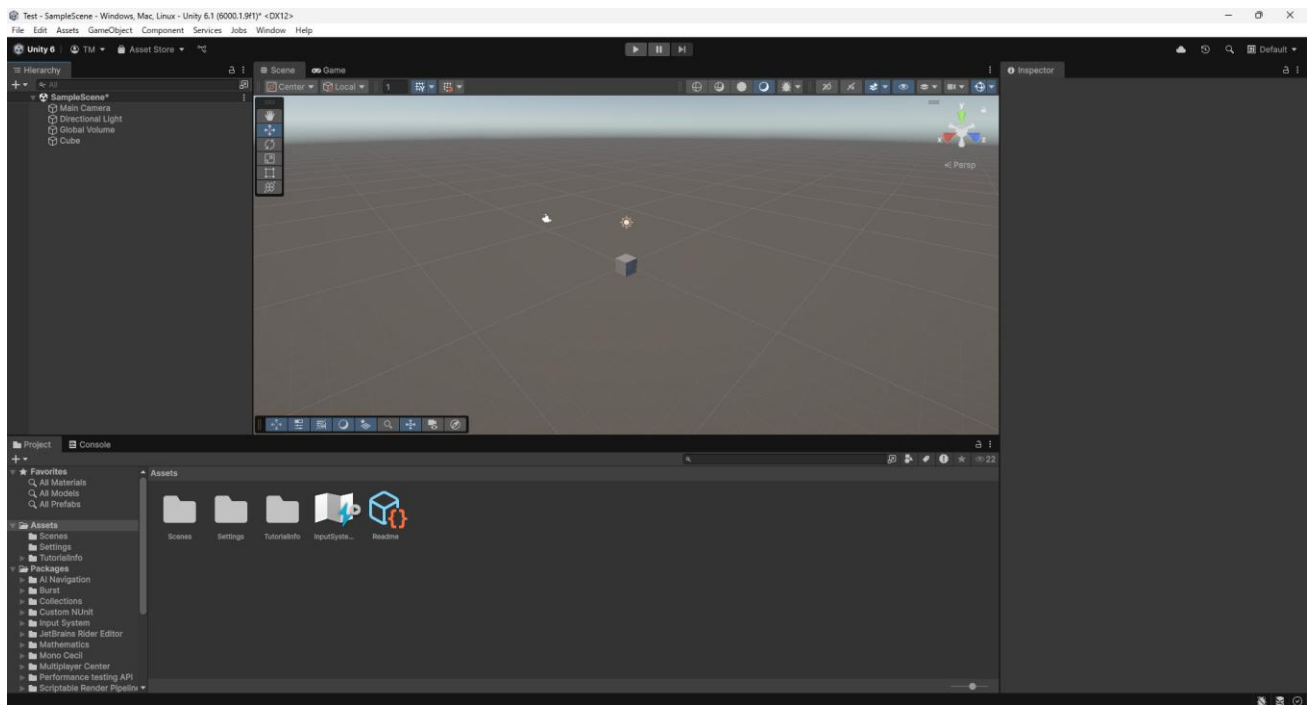


左上の+から3D Object のCube を選択する

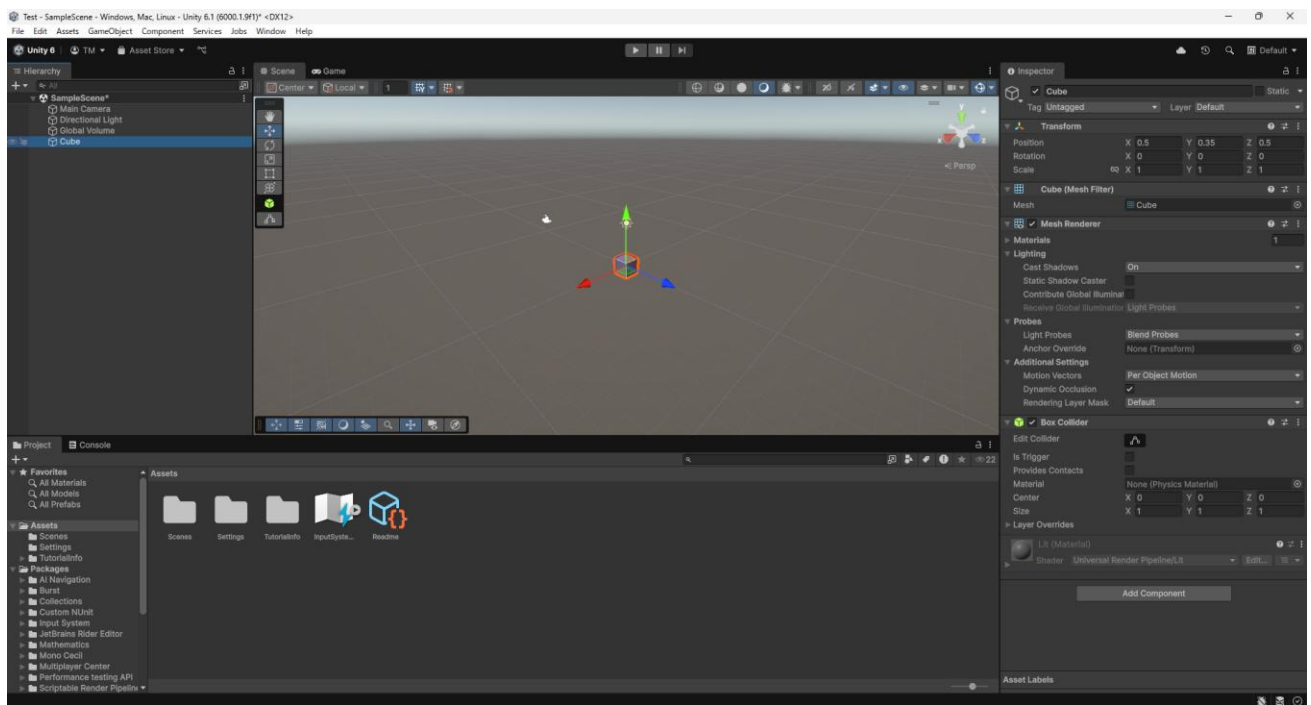




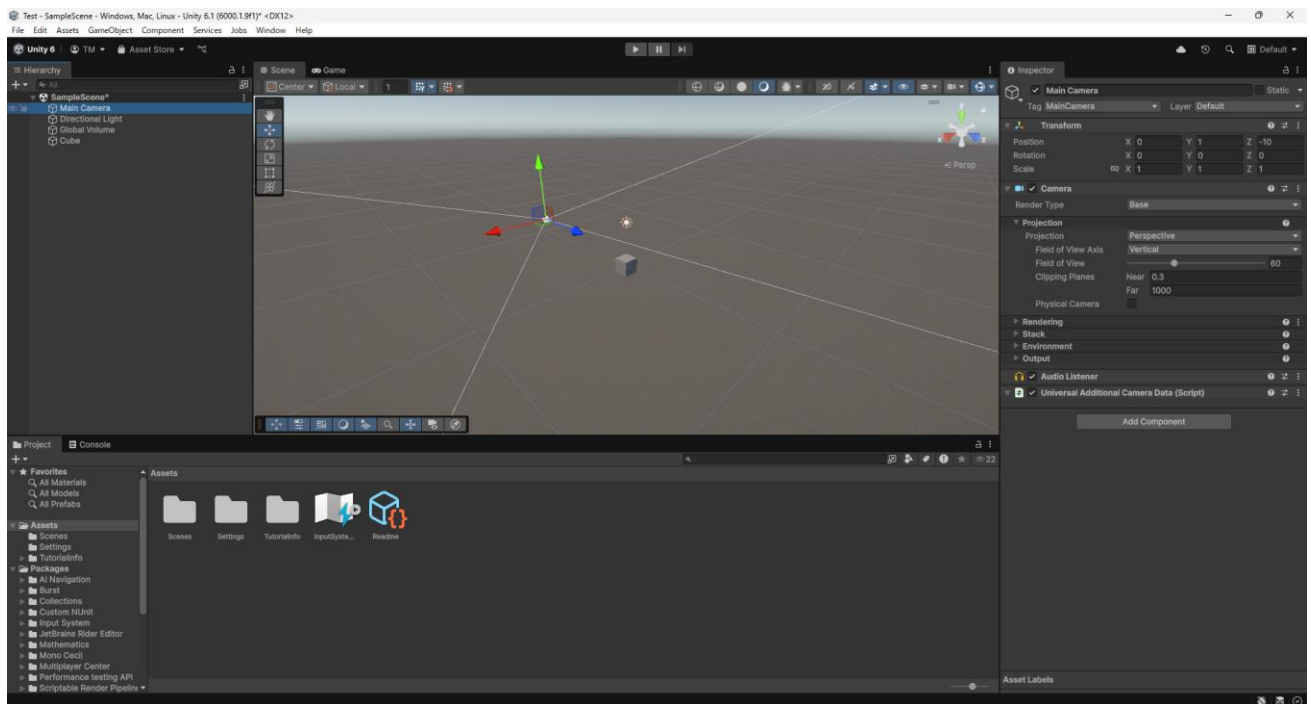
何もないところをクリックして選択を解除する

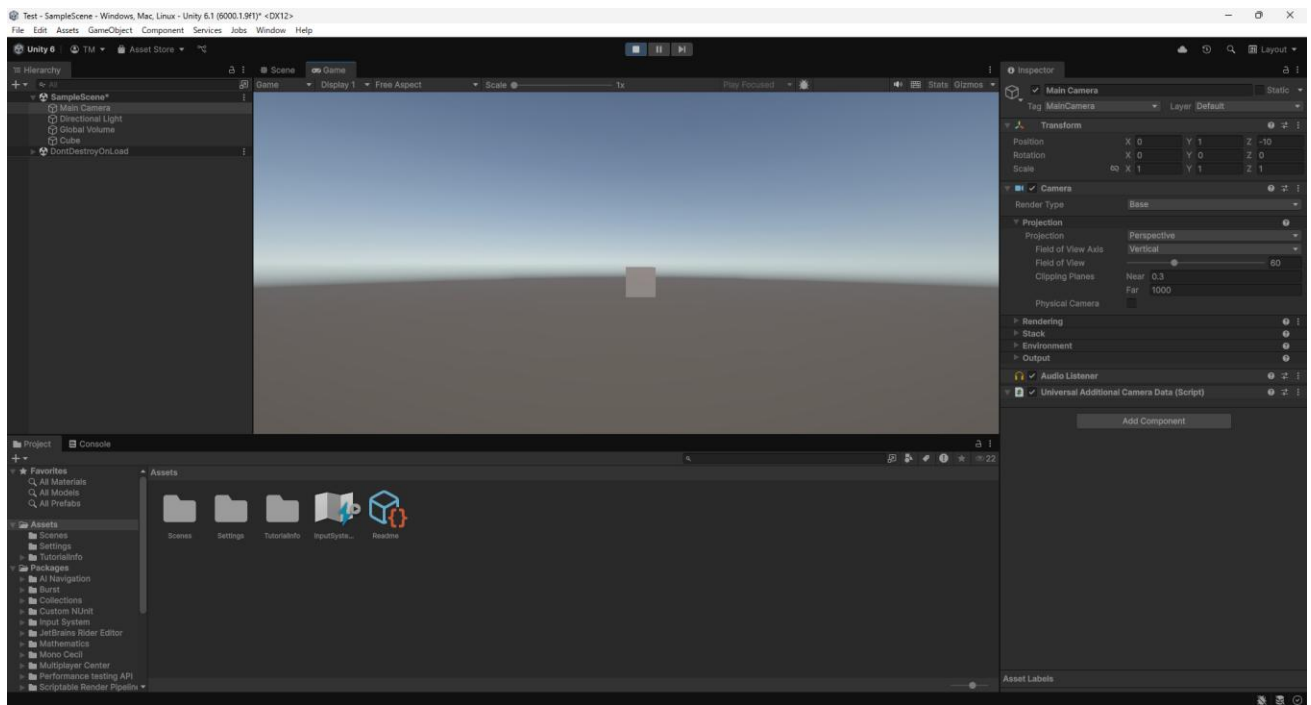


## もう一度 Cube をクリックする

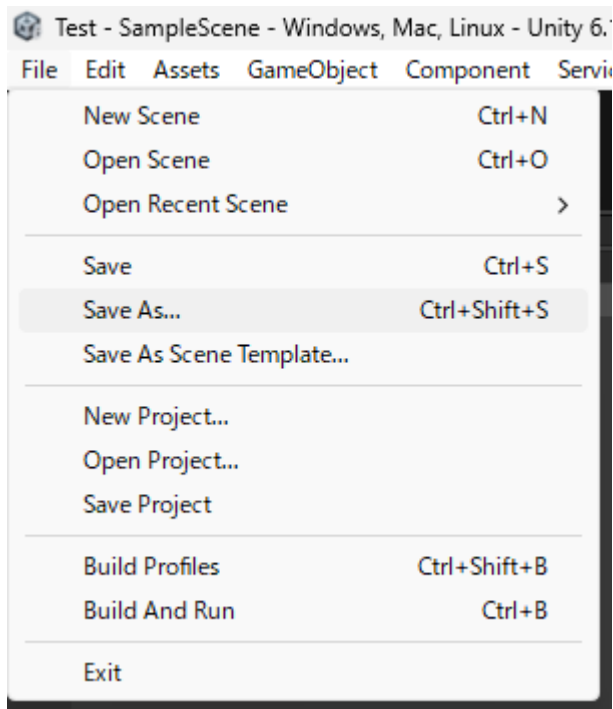


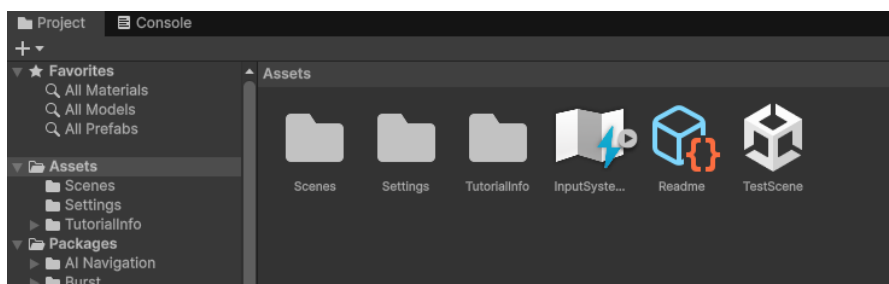
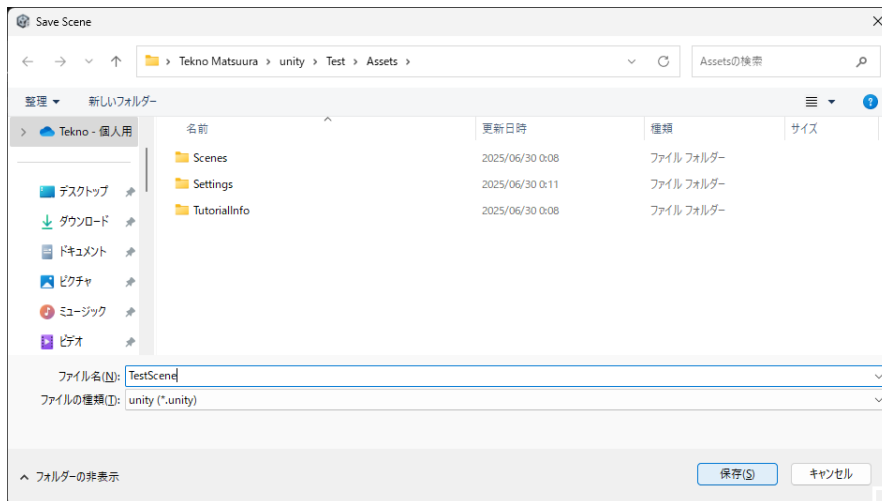
## Main Camera をクリックする





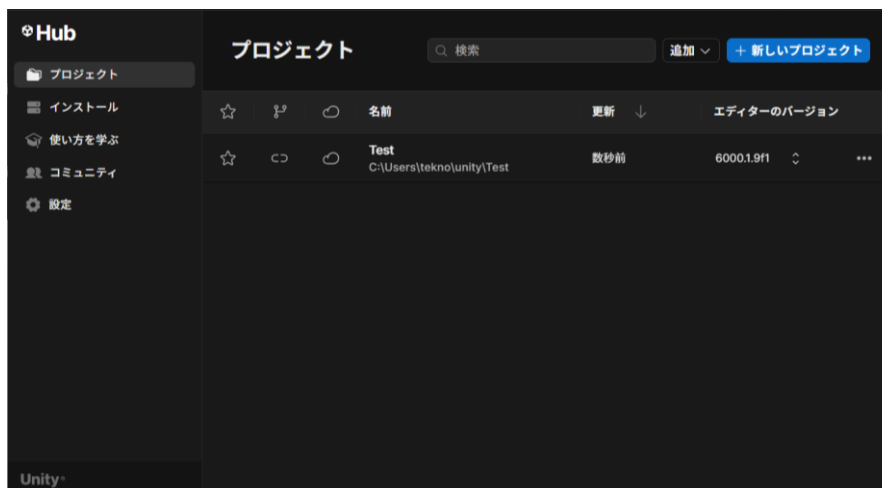
FileメニューのSave As... からプロジェクトを保存する。

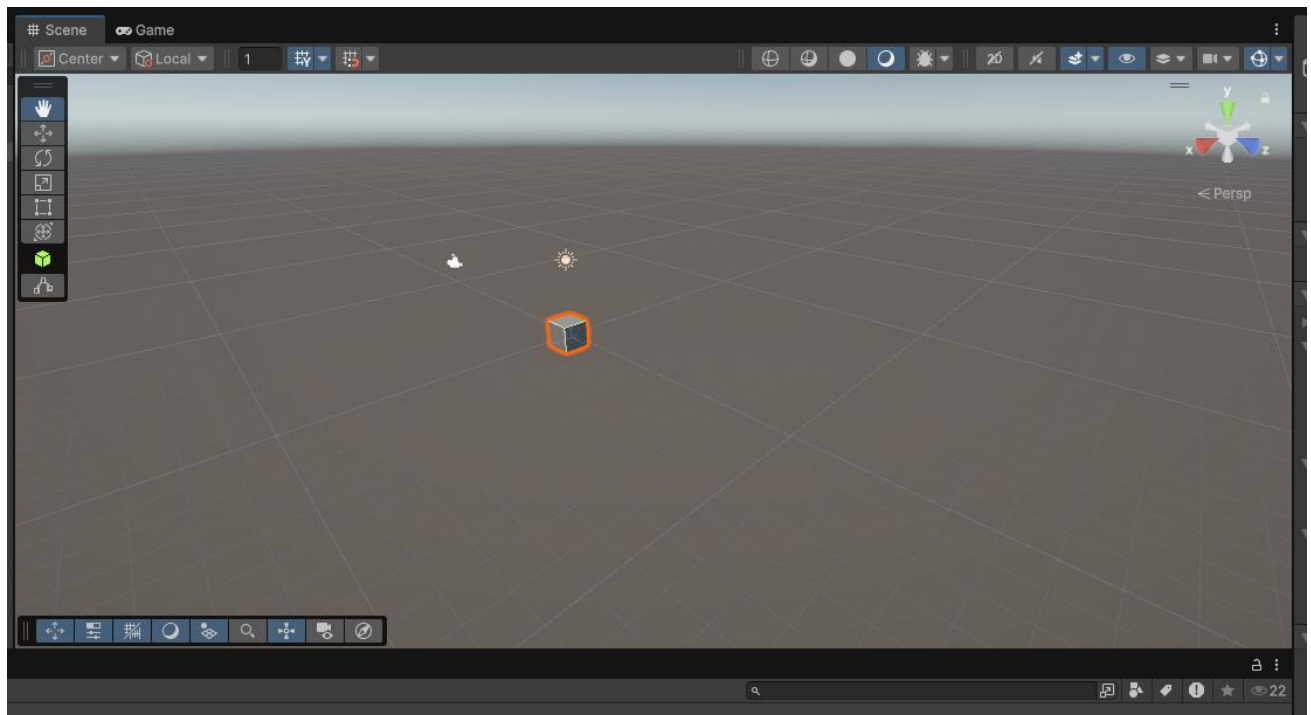




保存したプロジェクトを開く

Unity Hub を起動して、保存したファイルを選択する





左上の Scene タブを選択した状態で



を見る。

## ■視点の変更

・マウスホイールを転がすと、拡大縮小



を選択してドラッグすると平行移動

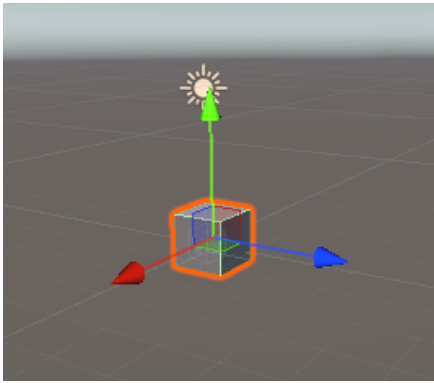
Alt を押しながらドラッグすると、視点の回転

## ■オブジェクトの変形

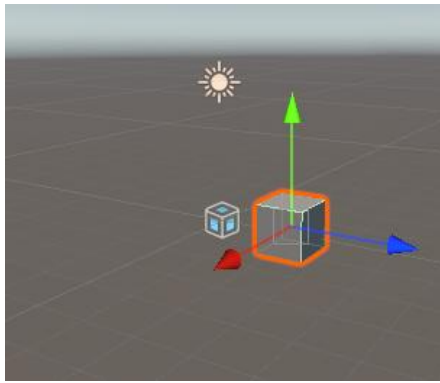


は移動

青い矢印をドラッグするとポジションが変わる



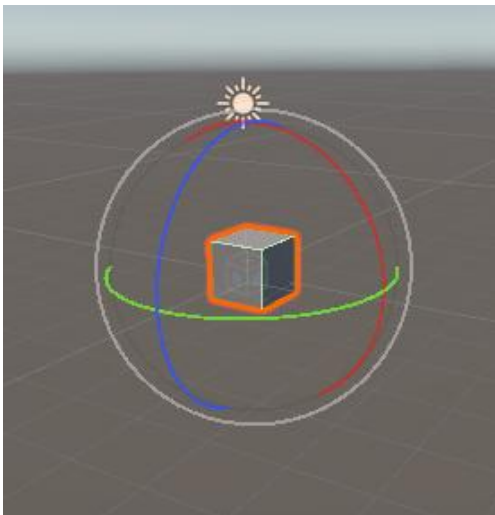
Transform						
Position	X	0.5	Y	0.35	Z	0.5
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1



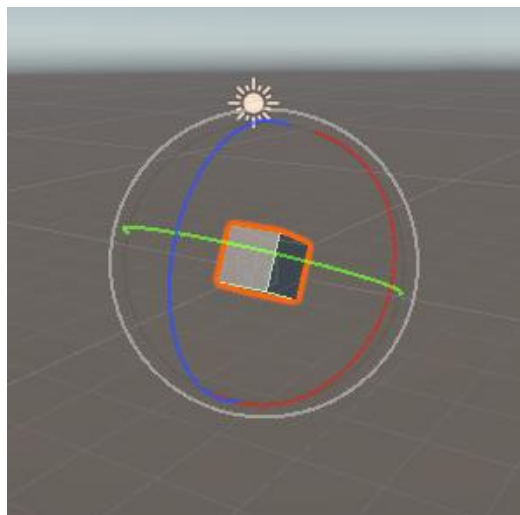
Transform						
Position	X	0.5	Y	0.35	Z	2.02
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1



は回転



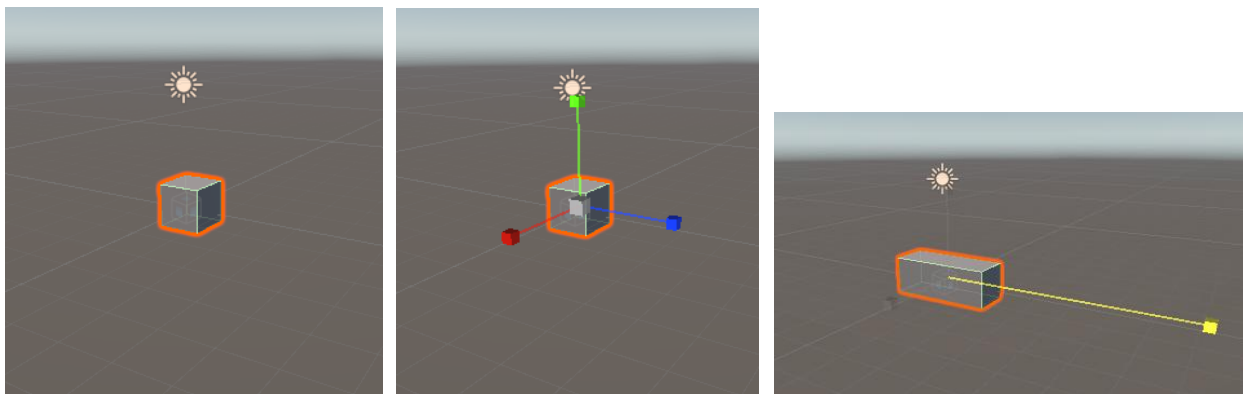
Transform						
Position	X	0.5	Y	0.35	Z	0.5
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1



Transform						
Position	X	0.5	Y	0.35	Z	0.5
Rotation	X	0	Y	0	Z	25.568
Scale	X	1	Y	1	Z	1



 は拡大縮小

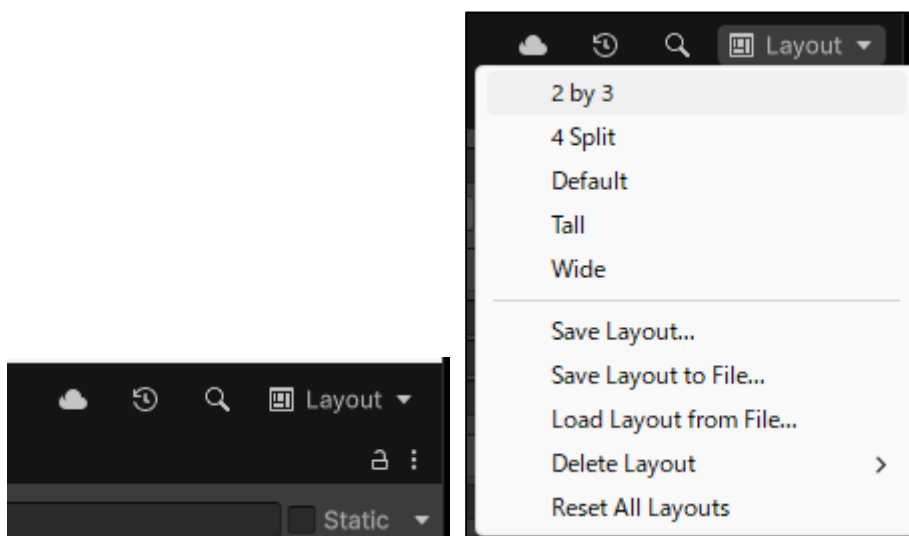


青の軸を伸ばすと Z の値が大きくなる

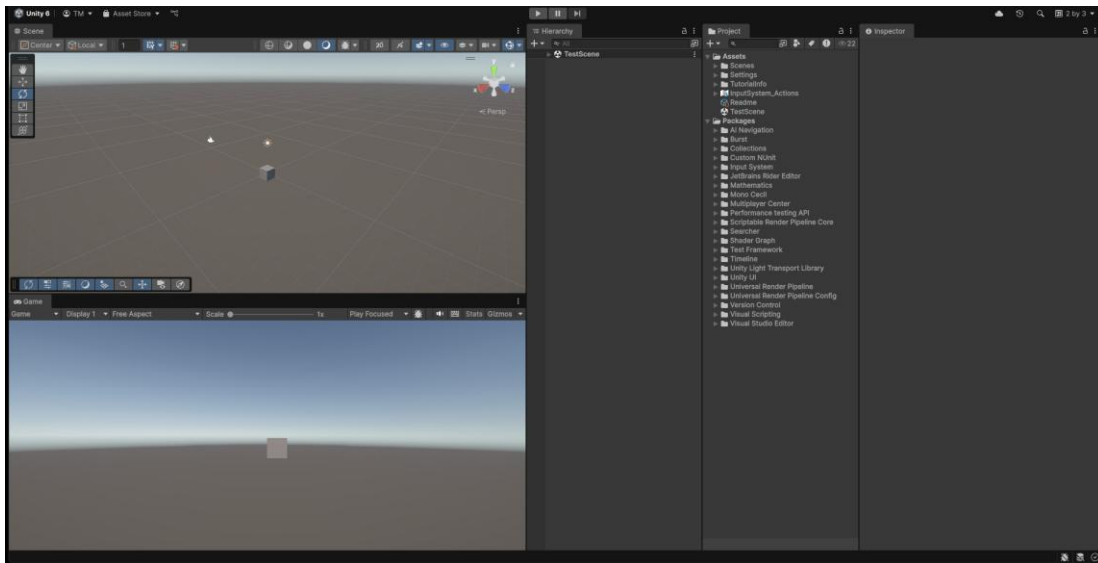


ここで、  
Position、Rotation、Scale の XYZ の値を変えて変形させて、元に戻す。

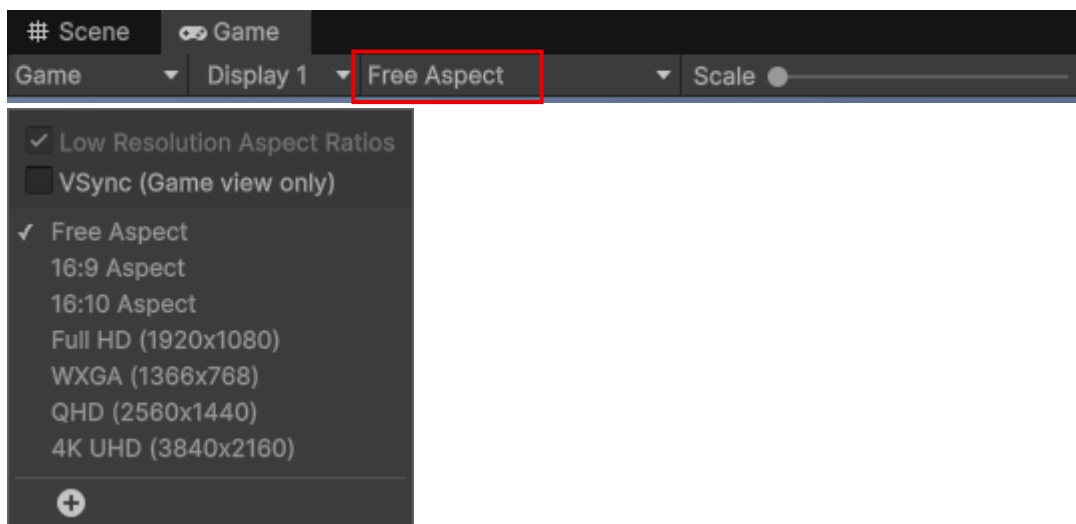
■レイアウトを変える



2 b y 3 にすると下記のようなレイアウトになる。他も試してみる。

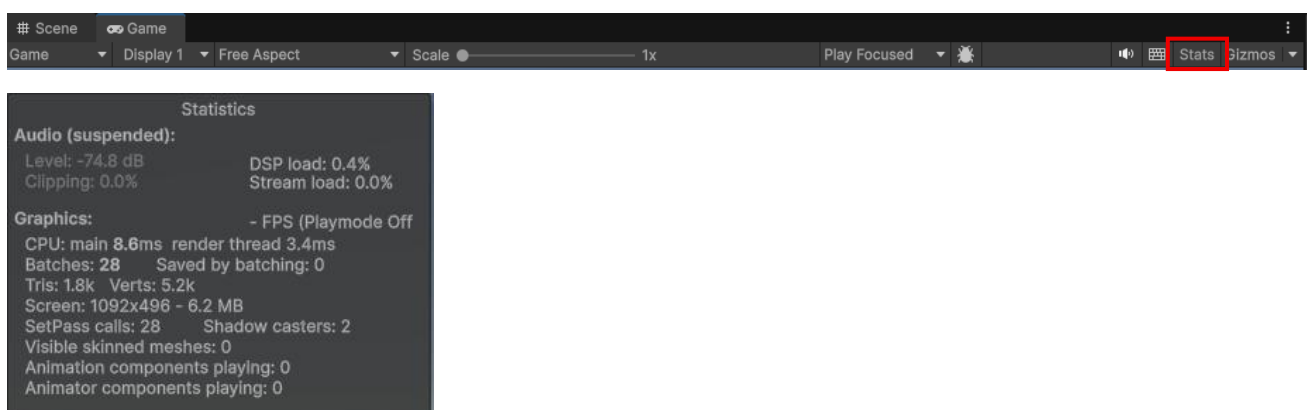


## 実行画面サイズの変更



## ■ プロファイラ

ゲーム実行時のプロファイルは Stats ボタンをクリックすると見ることができる。



## ■ゲームの実行

Ctrl+P またはツールバーの再生ボタンを押すとゲームが再生できる。今の段階では、まだ動きを付けていないので、Game ビューが表示されるだけになる。Game ビューを表示させるだけなら、Gama タブをクリックする。もう一度 Ctrl+P を押すと再生が停止できる。

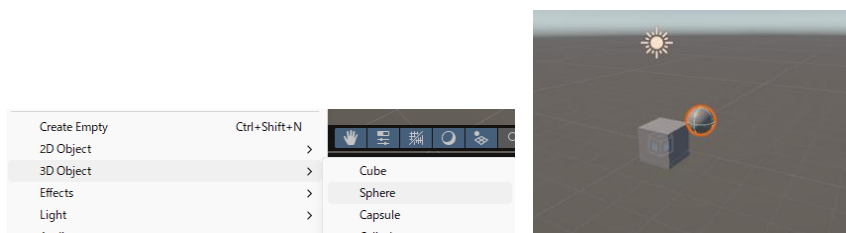
## ■物理エンジンで遊ぶ

ゲームオブジェクトを物理の法則に則って動かすことが可能。ここでは、ボールを地面でバウンドさせる。

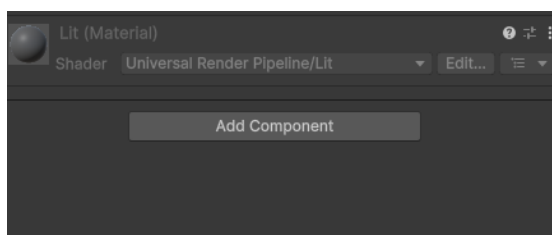
### ・Rigidbody をアタッチする

物理演算を適用したいゲームオブジェクトには、Rigidbody コンポーネントをアタッチする。アタッチするだけで物理法則が適用される。

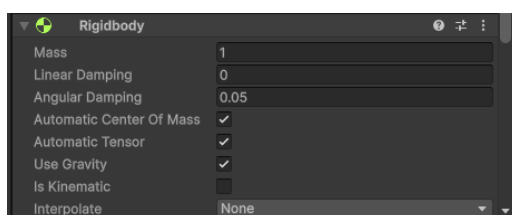
左側の Hierarchy ウィンドウで右クリックして 3D Object から Sphere を選択して



追加して、右側の Inspector ウィンドウの中の一番下の Add Component ボタンをクリックする



Physics→Rigidbody



### ・Rigidbody の主なプロパティ

Mass オブジェクトの重さ

Drag オブジェクトの空気抵抗

Angular Drag 回転に対する空気抵抗

Use Gravity 重力を適用するかどうか指定する

Is Kinematic 建物の壁など固定された物に使用するプロパティ

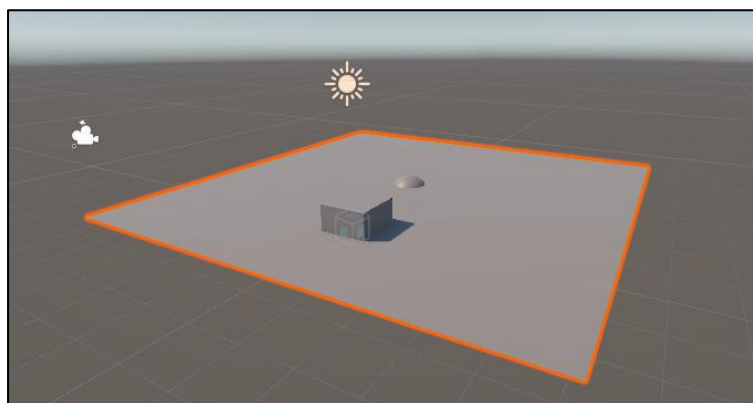
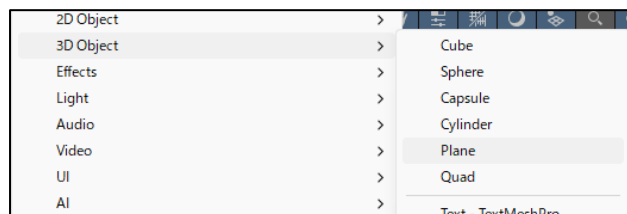
Interpolate Unity では描画処理と物理演算処理が別々に実行されるため、描画と物理演算にズレが生じる場合がある。これらを設定すると、物理演算の補完を行い、これらのズレを軽減させることができる。

Collisions Detection 物理演算で移動するオブジェクトを高速で動かすと壁などを貫通してしまう場合、Continuous にすると、高速移動させても貫通しなくなる。

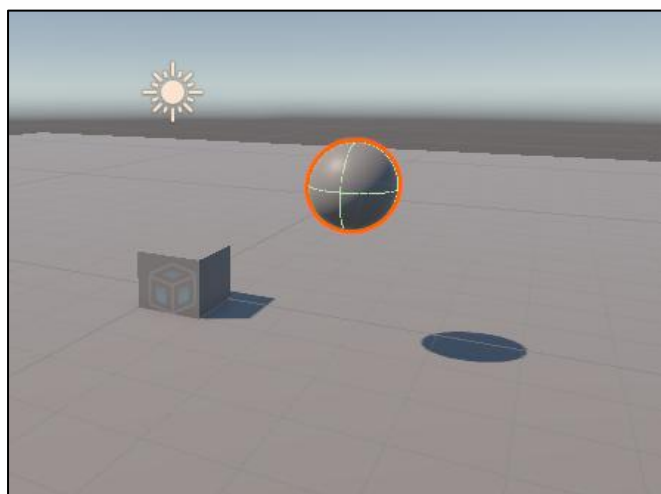
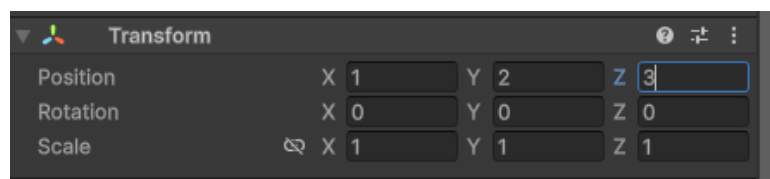
Constraints 各座標軸に対して、Freeze Position は移動、Freeze Rotation では回転をしないよう制御が可能

## ■平らな床を配置する

床を配置してみましょう。Hierarchy ウィンドウで右クリックし、3D Object → Plane を選択する

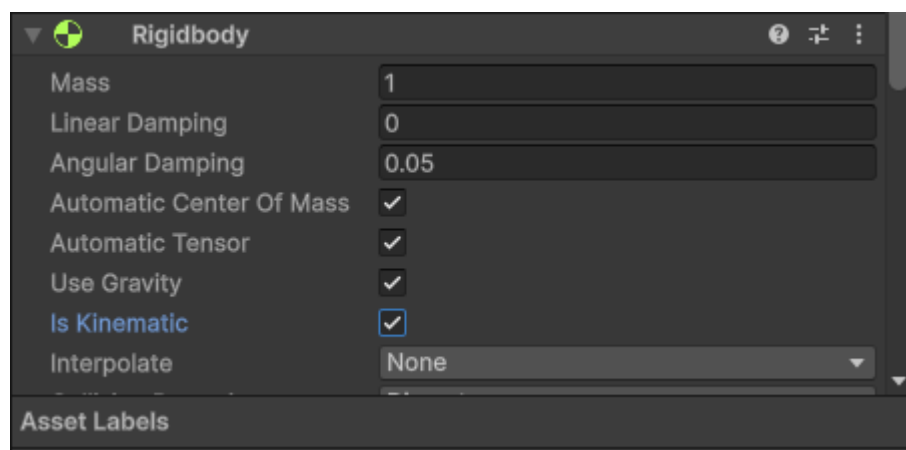


Sphere の設定を



にする。

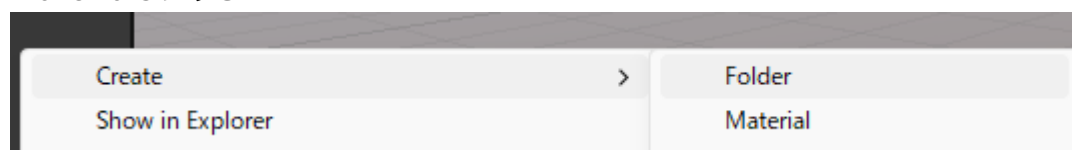
Plane を選択して、Rigidbody を追加して Is Kinematic にチェックを入れる



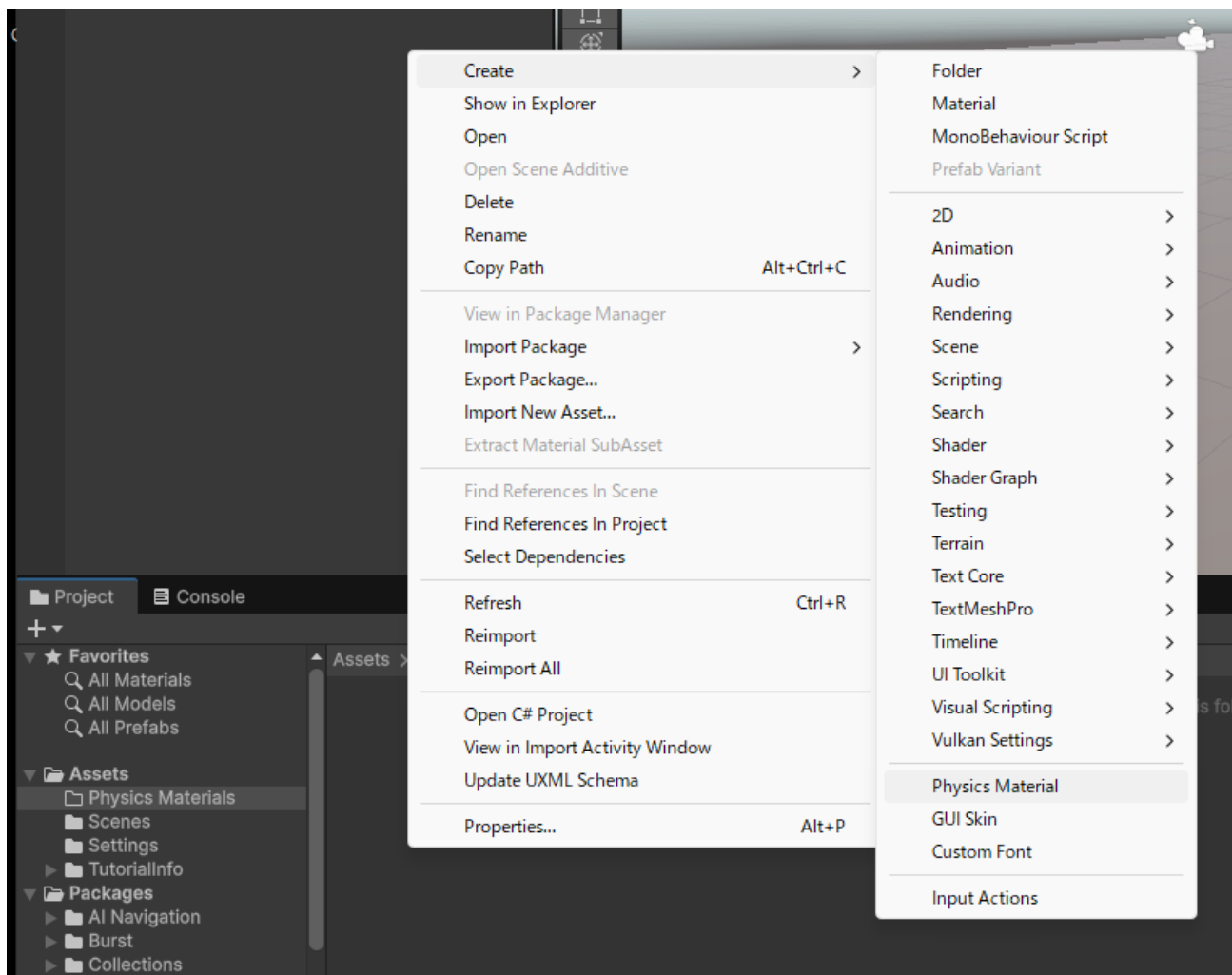
### ■バウンドする球を用意する

Physics Material を作成する。

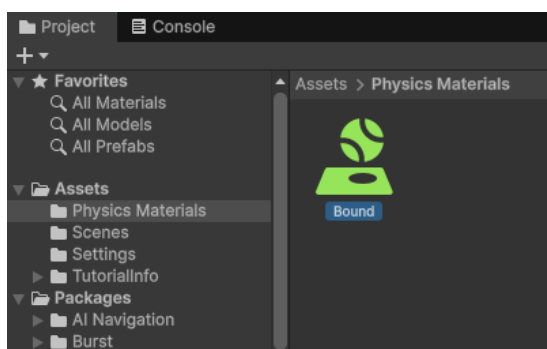
Assets ウィンドウの中で右クリックして、Create→Folder を選択して、フォルダの名前を Physics Materials にする



Physics Materials

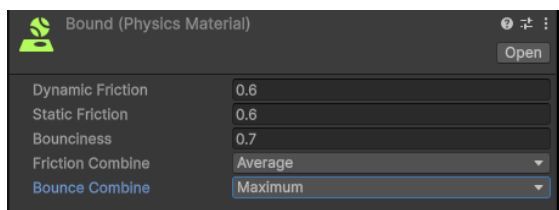


名前を Bound にする



設定を Bounciness が0.7

Bounce Combine を Maximum にする



・Physics Material の主なプロパティ

Dynamic Friction 摩擦抵抗の値で動いている物体に対して適用される

Static Friction 摩擦抵抗の値で、動いていない物体に対して適用される

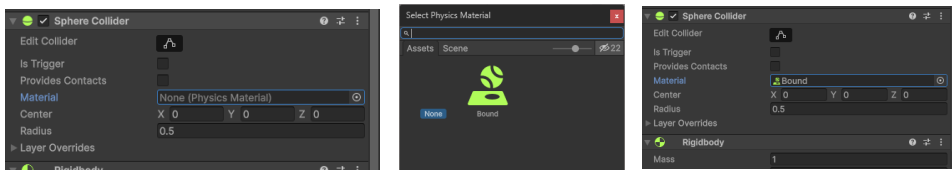
Bounciness 弾性の値で範囲は0～1

Friction Combine 実際に適用される摩擦抵抗の計算方法

Bounce Combine 実際に適用される弾性の計算方法

作成した Physics Material を Collider に設定する

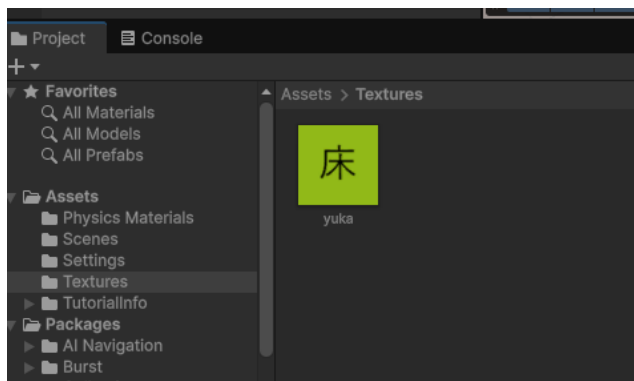
Sphere を選択し、InspectorウィンドウのSphere ColliderコンポーネントにあるMaterialに、今回作成した「Bound」を設定する





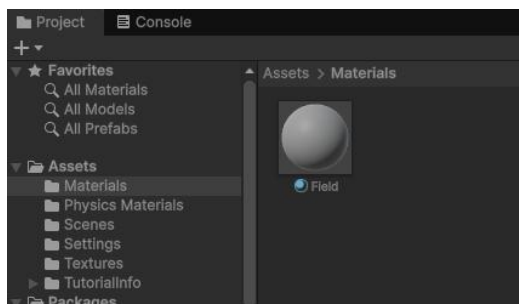
## ■床に色や模様を設定する

画像を Asset として取り込む。ここでは yuka.png というファイルを用意した。Assets フォルダの下に Textures フォルダを作り、その中に画像をドラッグ&ドロップする

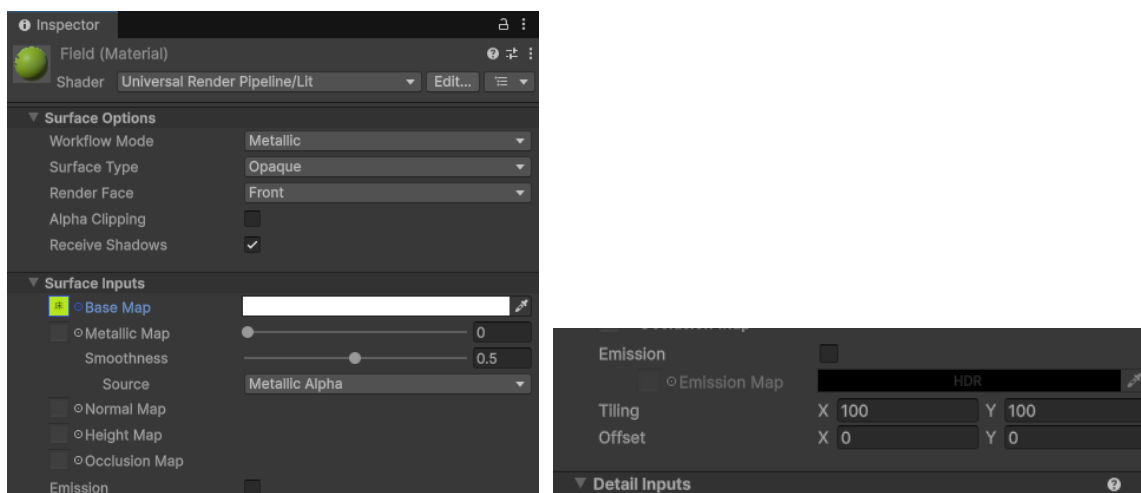


3D オブジェクトに画像を貼り付けるために、Material を用意する。

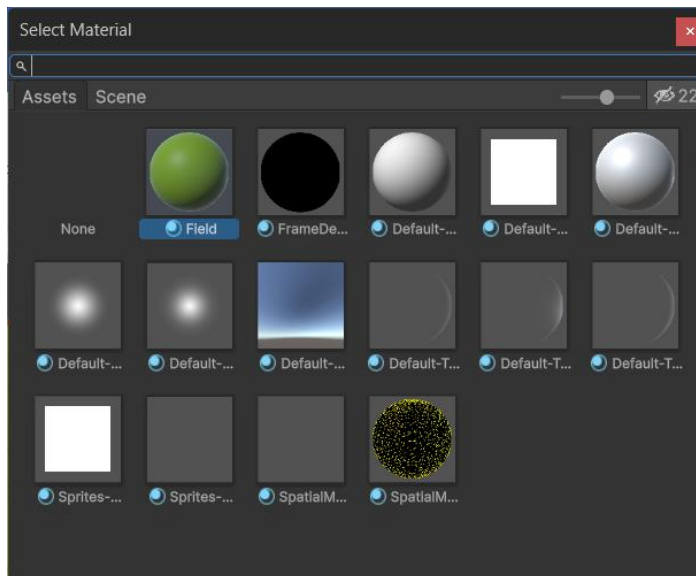
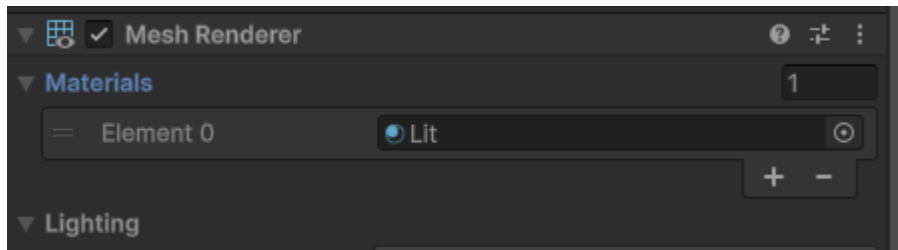
Project ウィンドウで Assets フォルダの下に Materials フォルダを作り、右クリックして、Create → Material を選択する。名前は Field にする



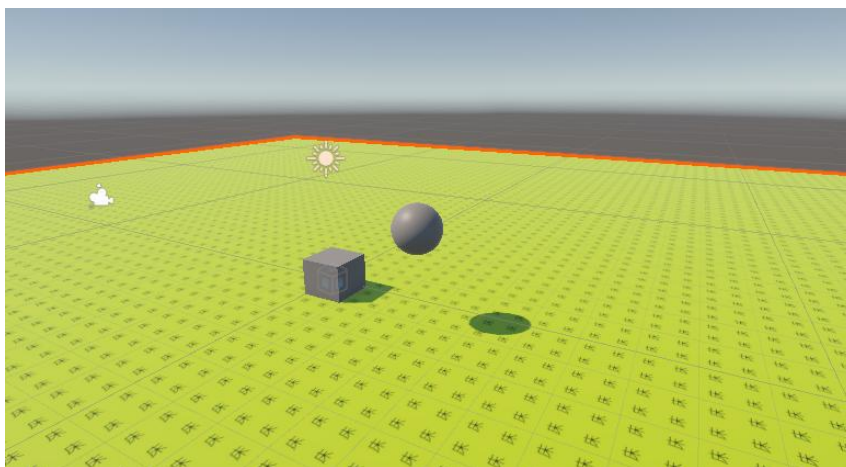
Base Maps の左側の○のアイコンをクリックして、yuka.png を選択する。Tiling の値は XY とともに 100 とする



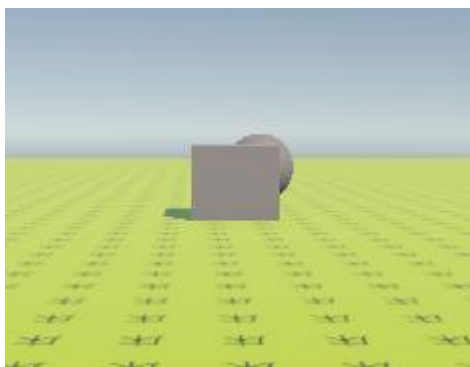
Plane を選択して、Element 0 の右側の🔗のボタンをクリックして、Field を選択する。



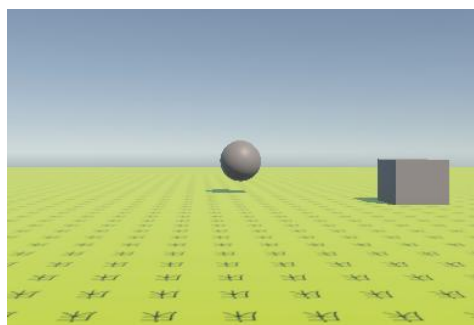
床が yuka.png で埋め尽くされる。



再生ボタンを押すと、立方体の後ろで球がバウンドする



立方体や球の位置を変えて、もう一度実行する。立方体を消してもよい。



分かってきたら、画像を変えたり、球や立方体の方に画像や色を設定する。