

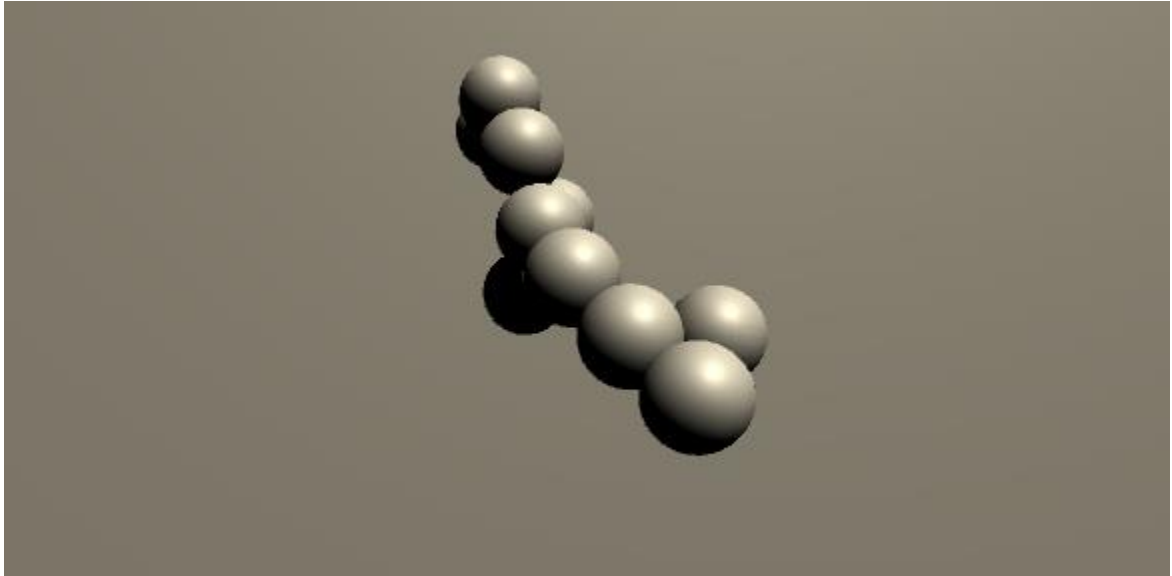
Unity勉強会

目次

1. 今回作るゲーム
2. Unity の準備
3. ゲーム作成

1. 今回作るゲーム

球を飛ばして沢山くっつけるゲームを作成します



完成イメージ

●今回のゲームを作るうえで、以下の内容を学ぶことができます。

- ・素材を配置して背景を作る。
- ・クリックした方向に球を飛ばす。
- ・球が落ちた時の処理を作る。
- ・球同士がくっつく処理を作る。

●所要時間；00分

2. Unity の準備

●Unity Hub のインストール

まずは Unity のインストールを行います。

以下の URL から Unity Hub をダウンロード・インストールしてください。

<https://unity3d.com/jp/get-unity/download>

※Unity Hub とは

Unity Hub は Unity 本体ではなく、Unity のバージョンを管理してくれるツールになります。

Unity もバージョンアップにより、今まで動いていた機能が動かなくなったり、オブジェクトなどが使用できなくなったりします。

それを防ぐためのツールになります。Unity Hub から Unity 本体のインストールが可能です。

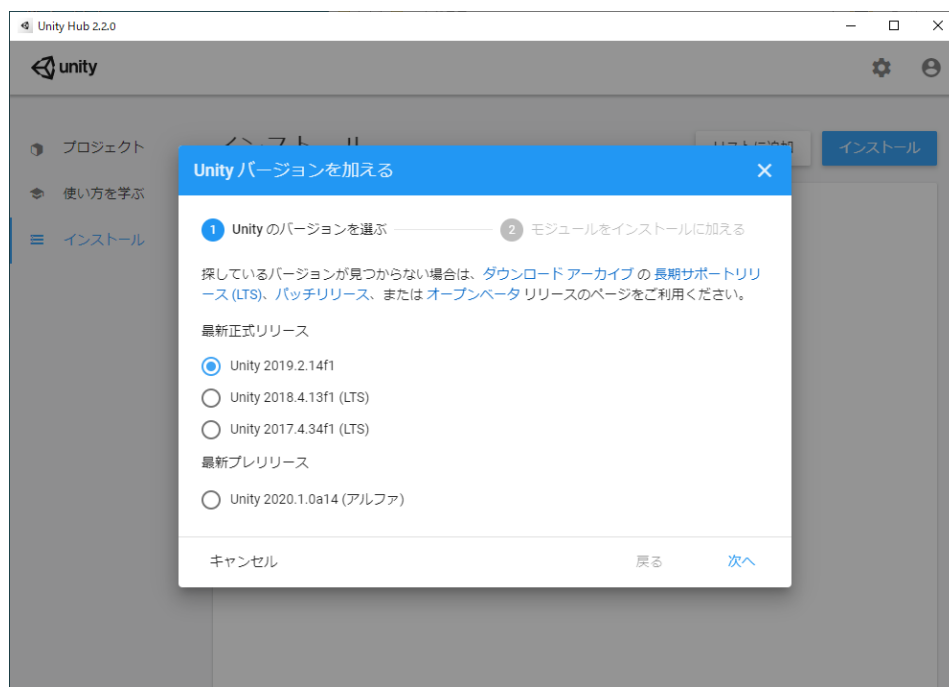
Unity Hub のインストールが完了したら、そのまま起動します。



左側にある、インストールを選択します。



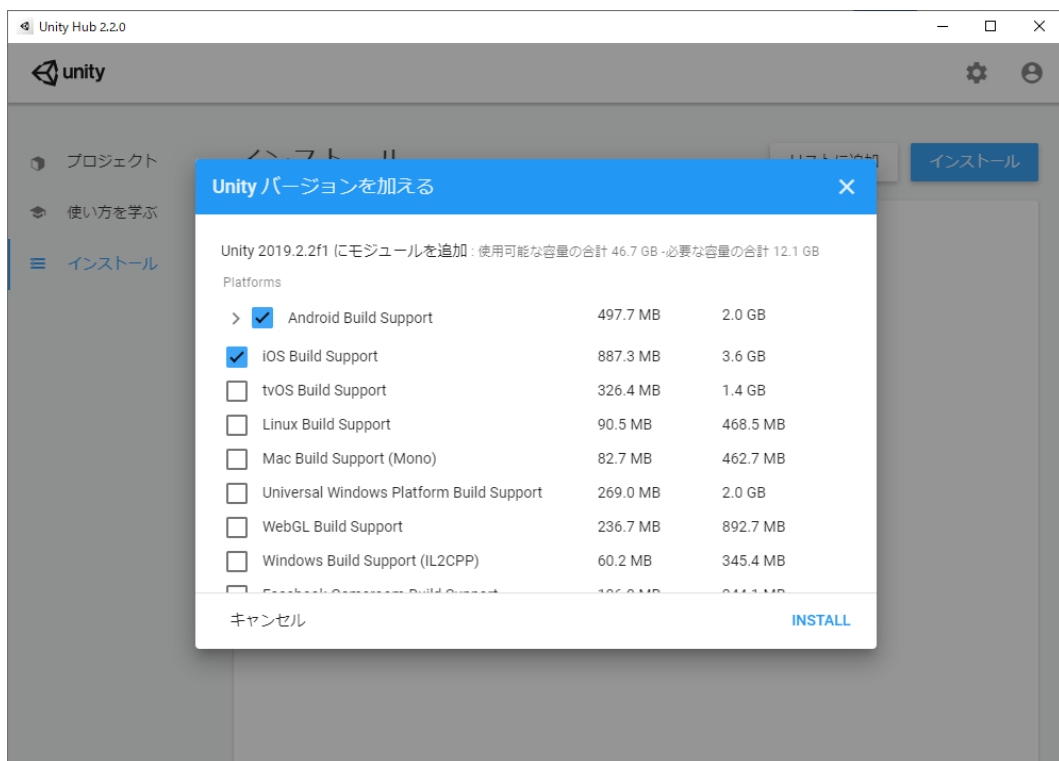
インストールタブに移動したら、右上にあるインストールを選択します。



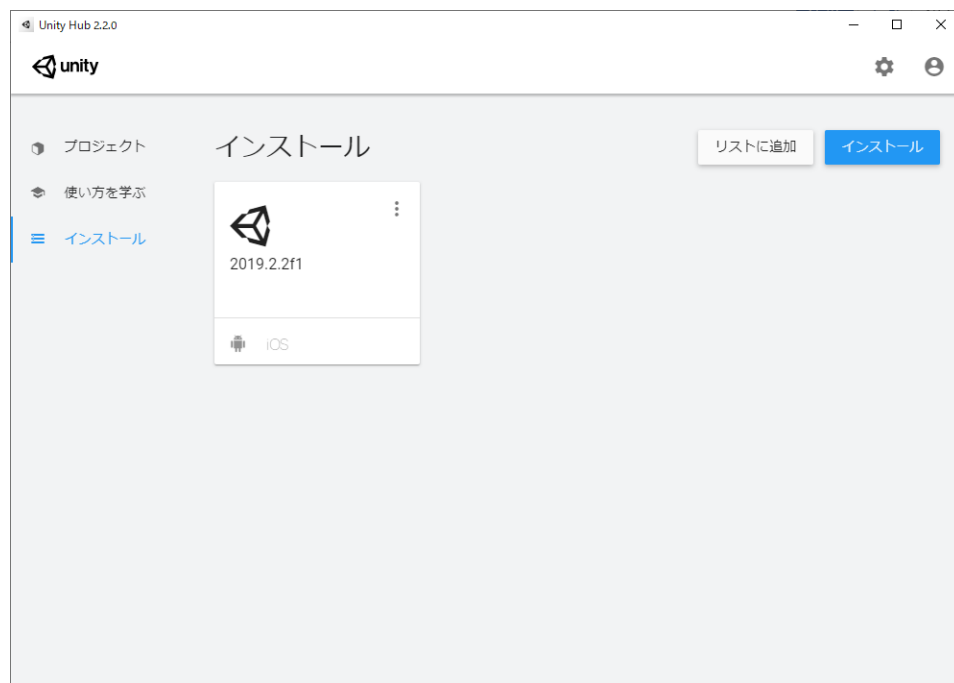
今回のバージョンは少し古い、2019.2.2f1 を使用したいので、「ダウンロード アーカイブ」を押下してください。

Unity 2019.2.4 5 Sep, 2019	Unity Hub	Downloads (Win)	Downloads (Mac)	Release notes
Unity 2019.2.3 28 Aug, 2019	Unity Hub	Downloads (Win)	Downloads (Mac)	Release notes
Unity 2019.2.2 21 Aug, 2019	Unity Hub	Downloads (Win)	Downloads (Mac)	Release notes
Unity 2019.2.1 14 Aug, 2019	Unity Hub	Downloads (Win)	Downloads (Mac)	Release notes

Unity のダウンロード アーカイブページに移動しますので、少し下にスクロールして Unity 2019.2.2 の Unity Hub を選択します。Unity Hub を開くか聞かれますので開きましょう。



必要なモジュールを選択し、(よくわからなければそのままでもいいです)右下の **INSTALL** を押下します。
※インストールには少し時間がかかります。



インストールタブに 2019.2.2f1 の Unity が追加されて準備は完了です。

3. ゲーム作成

●プロジェクトの作成

プロジェクトタブに移動して、右上の新規作成を選択します。



テンプレートは「3D」を選択し、任意のプロジェクト名と保存先を選択します。



設定が完了したら、右下の作成を押下します。

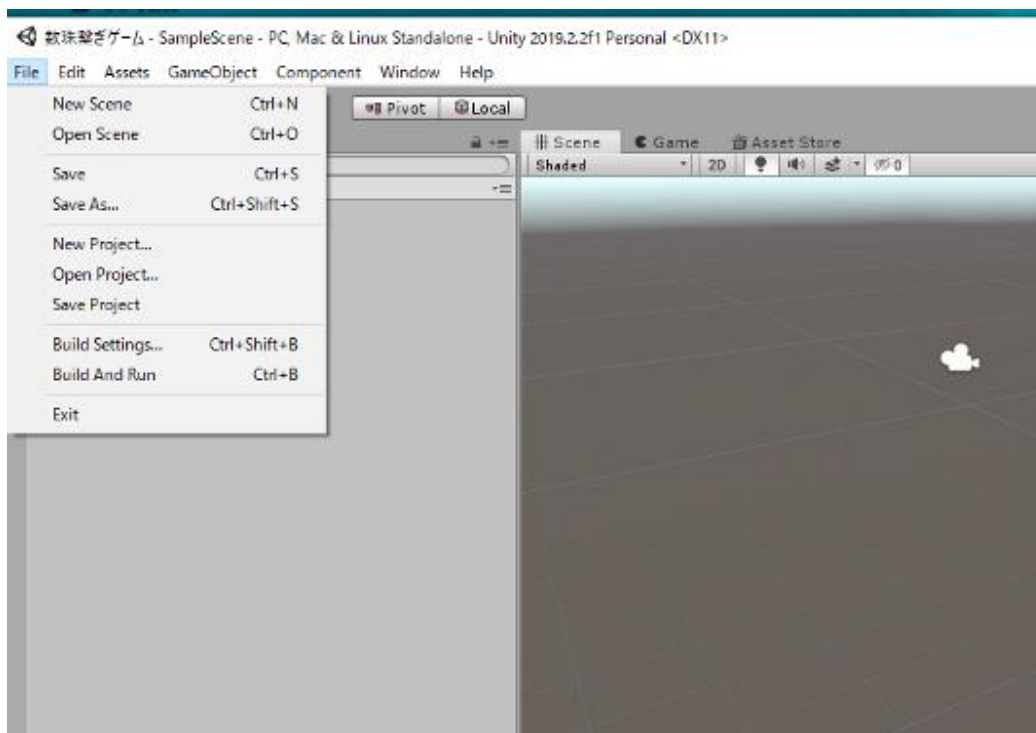
自動的に Unity が起動します。

●シーンの作成

プロジェクトを作成したら、まずはシーンを作成します。

※シーンとは？

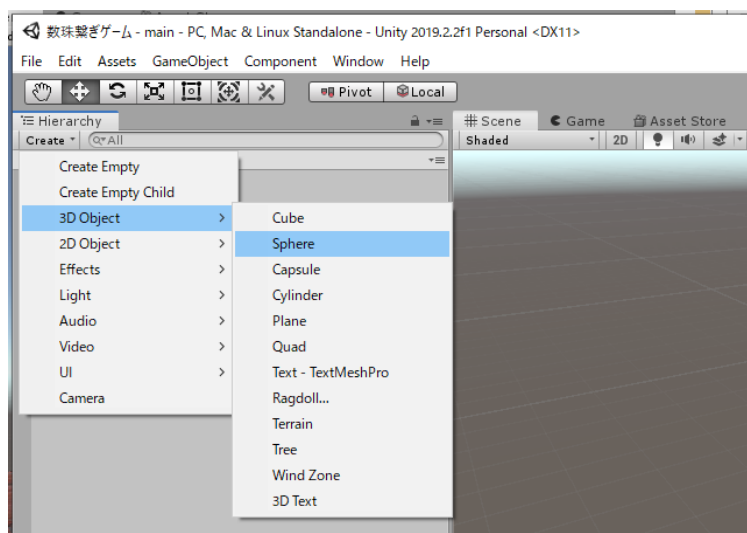
シーンとはその名の通り場面を設定する部品です。ゲームというと町やフィールド、ダンジョンと考えるとわかりやすいと思います。まずシーンを作成して壁や床などの部品をさらに追加していきます。



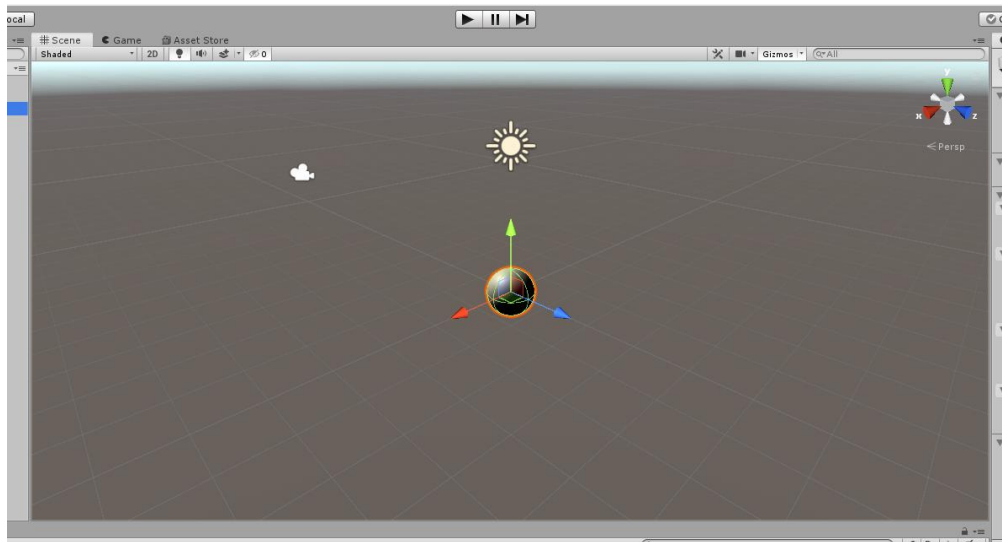
[File]>[New Scene]を選択して新しいシーンを作成し、[File]>[Save]でシーンを保存します。シーン名は[main]としてください。

●「Sphere」を追加する。

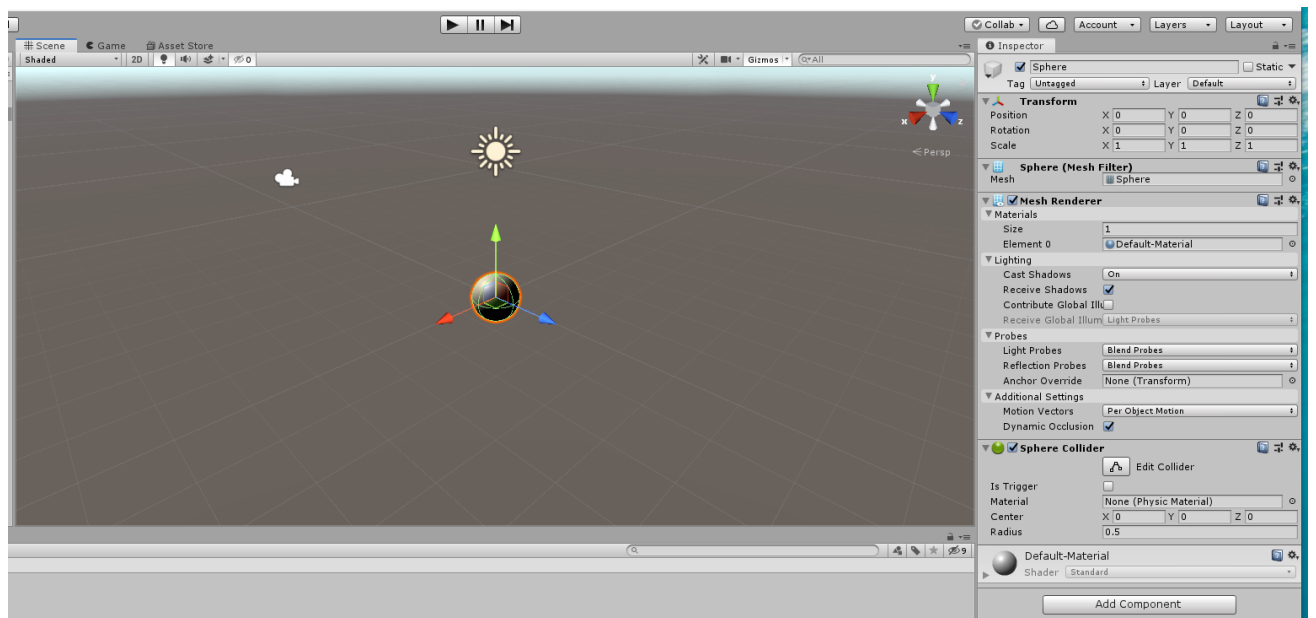
発射する球のぶつけ先を用意します。そのためにオブジェクト「球」を作成します。



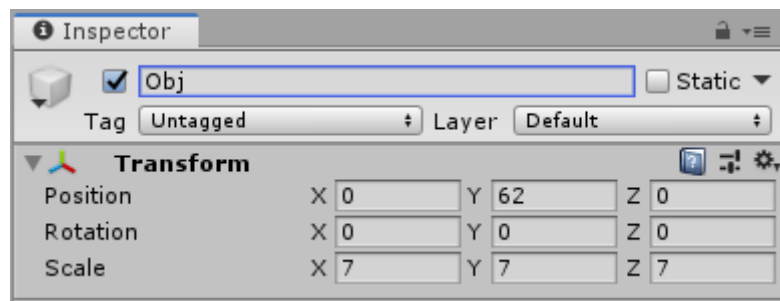
[Create]>[3D Object]>[Sphere]と選択します。



中央に球体が現れました。このように部品は[Create]から簡単に追加することができます。球の位置や大きさを調整します。



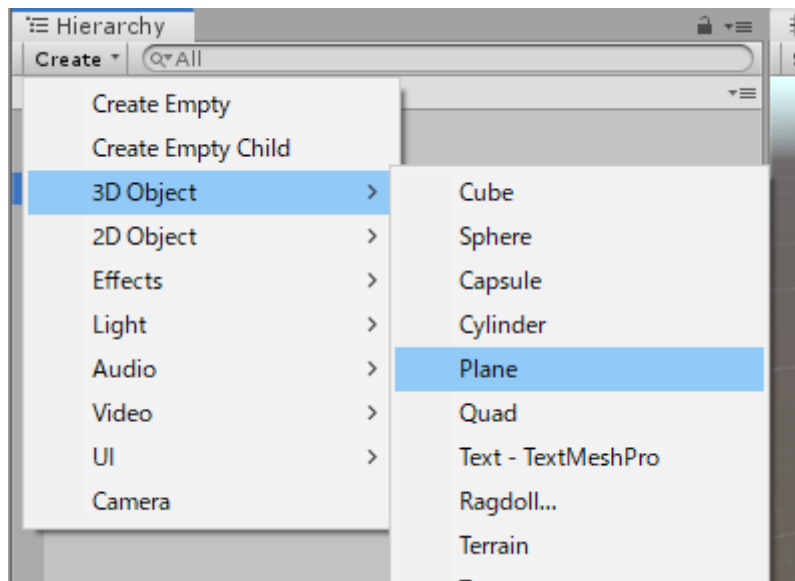
部品を選択していると右側にプロパティが表示されます。その中の以下の値を次のように変更します。



Position は座標、Scale は大きさの設定となります。

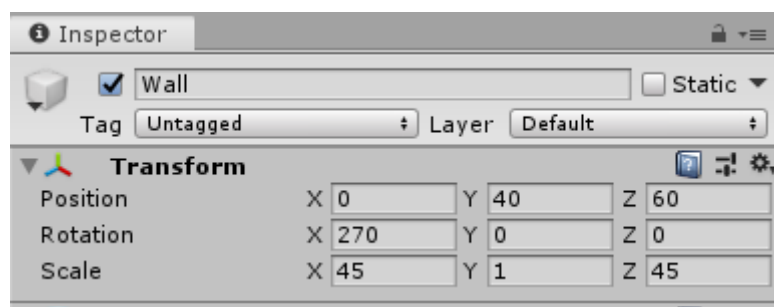
● 「Plane」を追加する。

今まだと見た目が味気ないうえに、飛ばした球が際限なく奥に飛んで行ってしまうため、それを防止するために、オブジェクト「壁」を追加します。

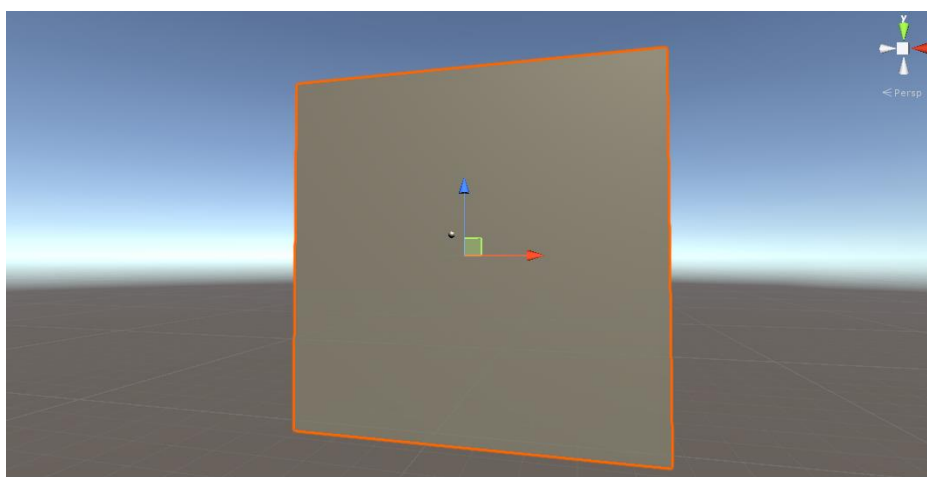


[Create]>[3D Object]>[Plane]と選択します。

先ほど追加した Sphere と同じように位置と大きさの調整を行います。



Rotation は回転になります。X を 270 にすることで、縦向きに配置することができました。

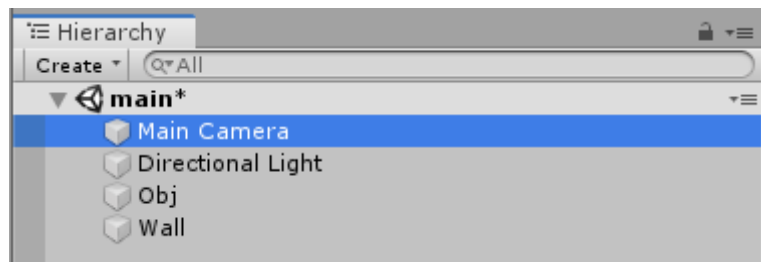


現在のゲーム画面はこのようなになっています。

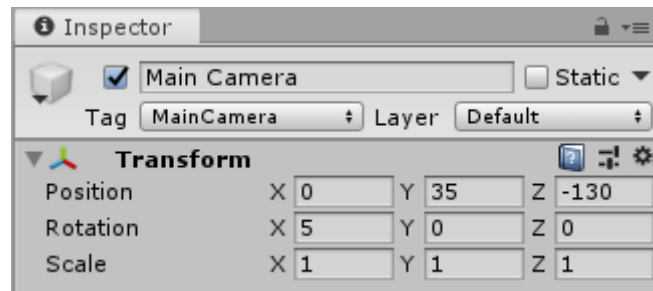
※マウスホイールで画面の拡大・縮小。右クリックをしながら移動で、スクロールをすることができます。

●「Camera」の設定を行う。

カメラの調整を行い、ゲームを見やすくします。今回は少し見下ろすようなカメラアングルにします。



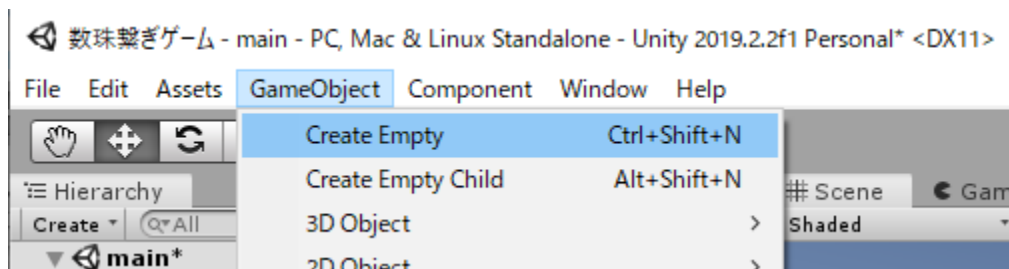
左側にある「Main Camera」を選択して、



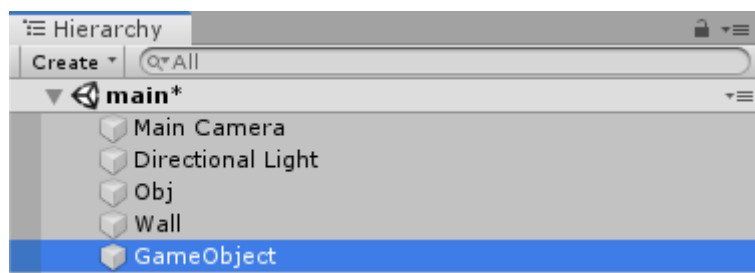
プロパティを上のように変更します。

●ぶつける球を作成する。

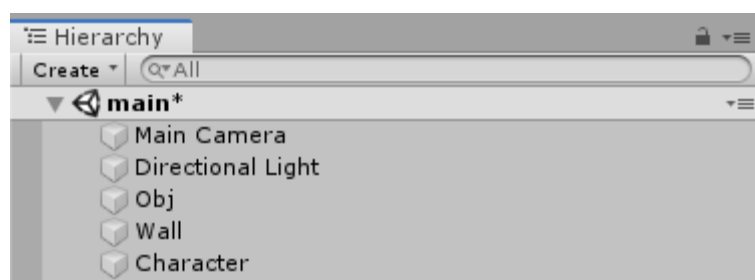
発射する球を作成します。



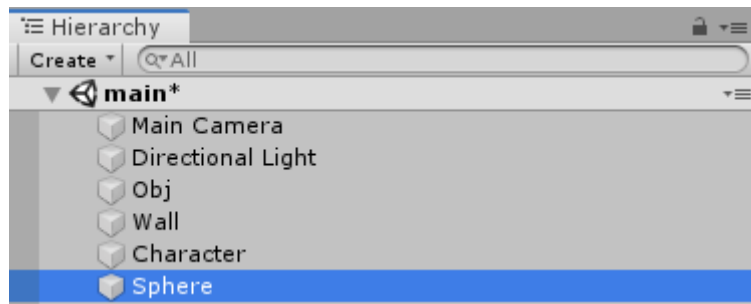
[GameObject]>[Create Empty]と選択して空のオブジェクトを作成します。



名前を「GameObject」から「Character」に変更しましょう。



続けて[Create]>[3D Object]>[Sphere]と選択して新しい球体のオブジェクトを追加します。



追加した「Sphere」を「Character」にドラッグします。



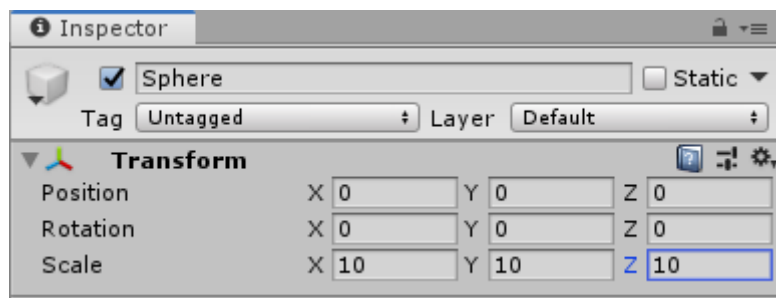
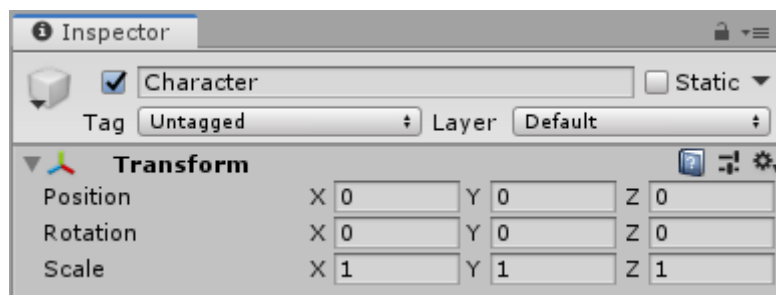
このような状態になっていれば OK です。

これで **Character** と **Sphere** は親子関係になりました。

※親子関係にする理由

親子関係を設定したのは、重力や当たり判定などは **Character** で行い、見た目である、**Sphere** には何も影響されないようにするためです。このようにしておくと、別のオブジェクトに差し替えるのが非常に簡単になります。

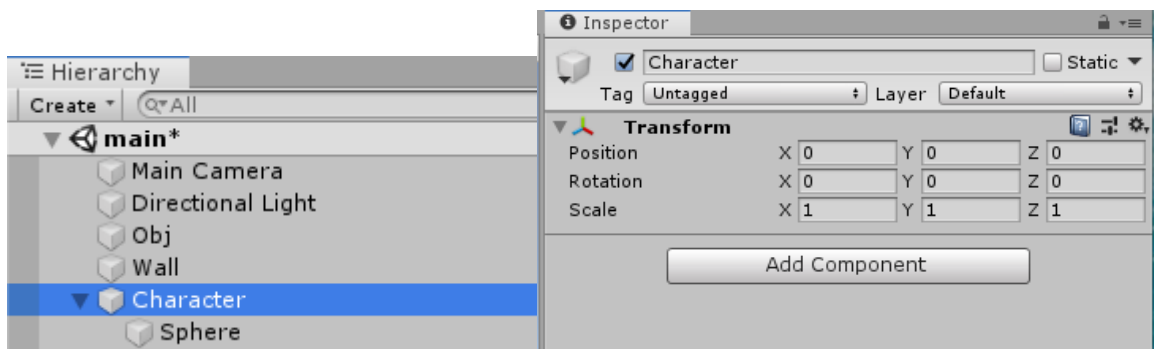
Character と **Sphere** のプロパティも変更しましょう。



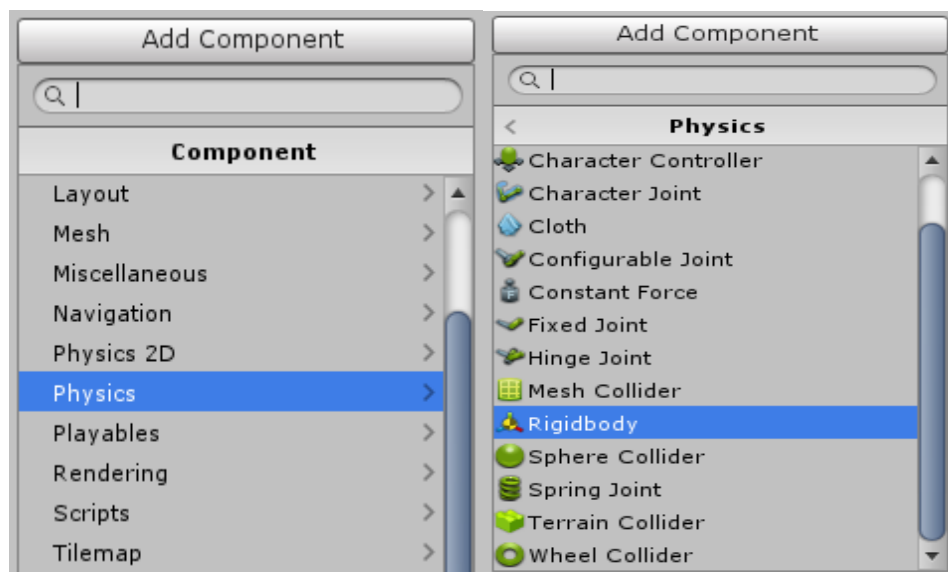
先ほど追加した **Obj** より一回り大きいサイズの球体にしました。

●物理挙動を設定する。

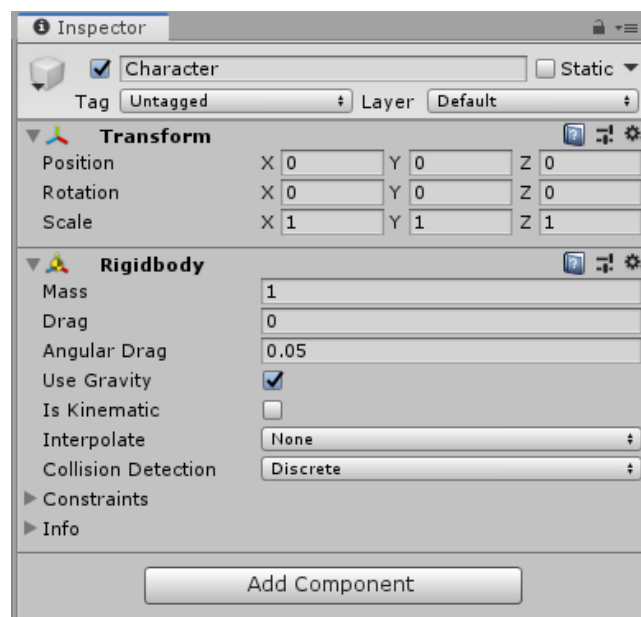
発射された球が重力に従って落ちていくように重力や球同士がぶつかったときのあたり判定を設定します。



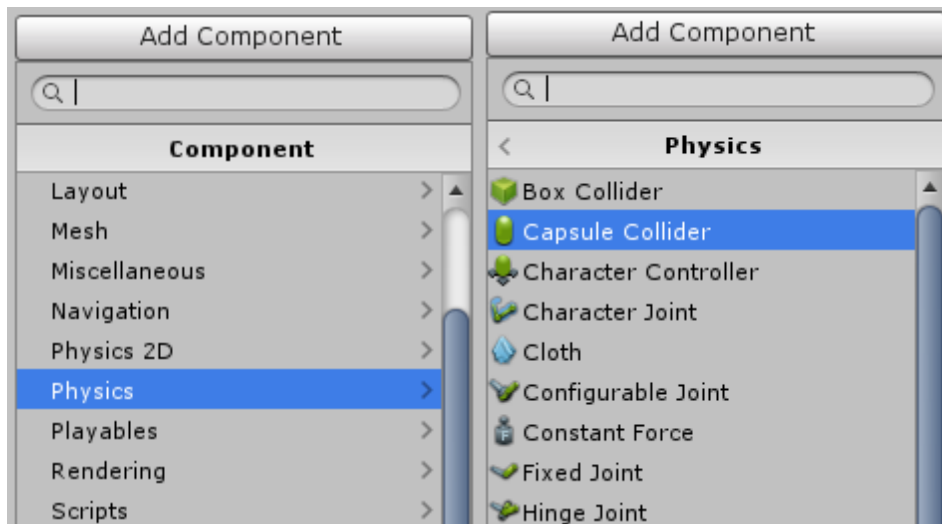
Character を選択している状態で「Add Component」を押下します。



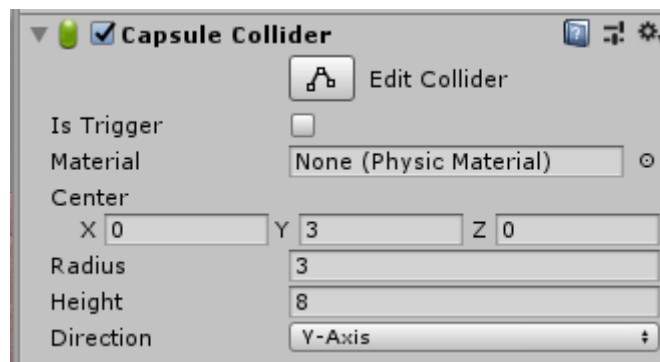
[Physics]>[Rigidbody]を選択します。



重力の設定が追加されました。



続けて「Add Component」から[Physics]>[Capsule Collider]を選択します。
Capsule Collider のプロパティを次のように変更します。

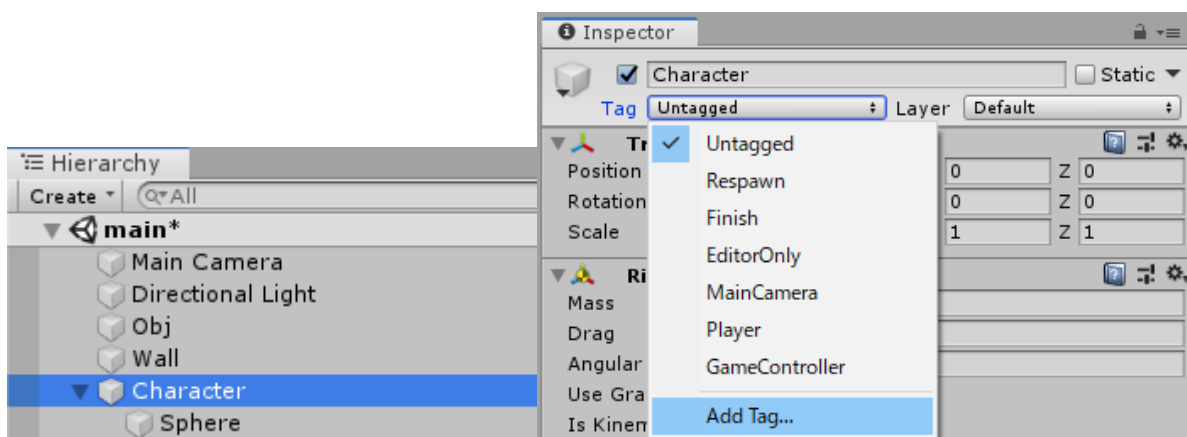


Center は当たり判定の中央の位置になります。

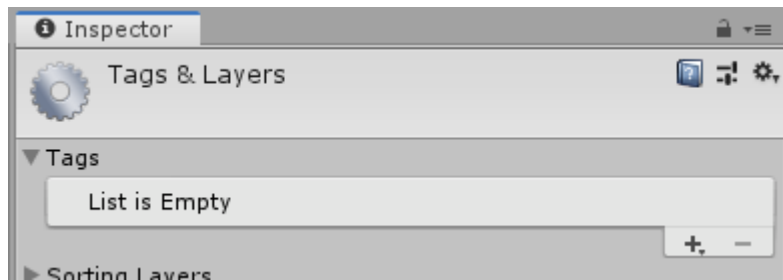
Radius、Height はそれぞれ、横、縦の判定の大きさになります。

● タグを変更する。

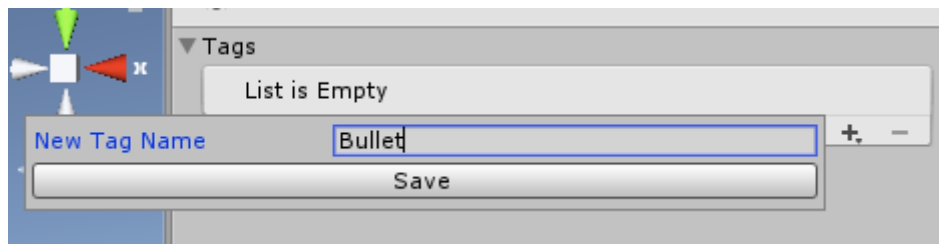
タグはスクリプト上での判別に使用されます。今回は、オブジェクト同士がぶつかったときや落ちた球を削除するために使用しています。



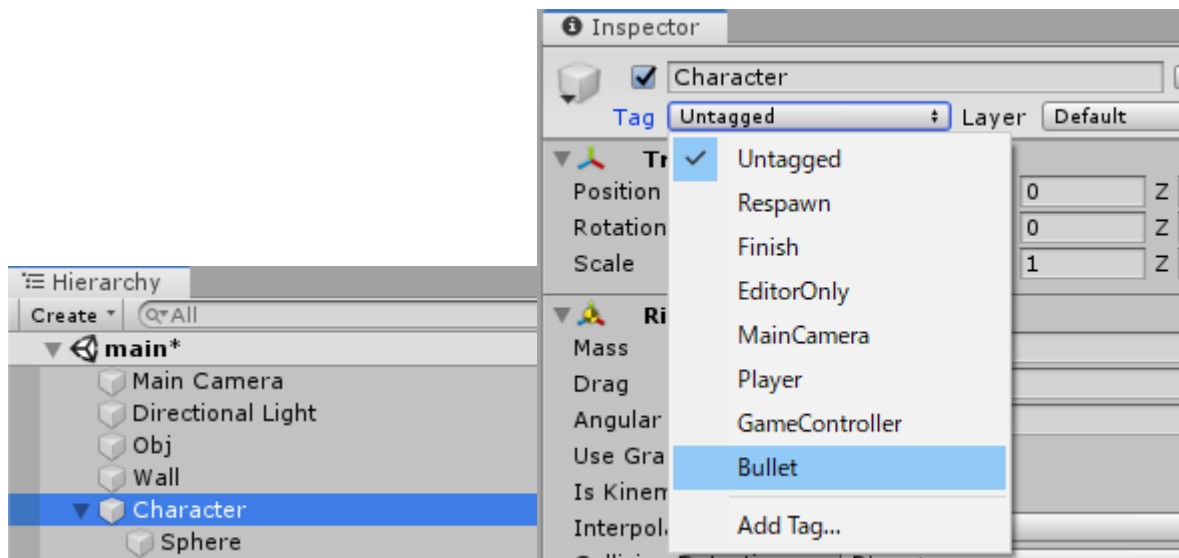
Character を選択している状態で、[Tag]>[Add Tag]を選択します。



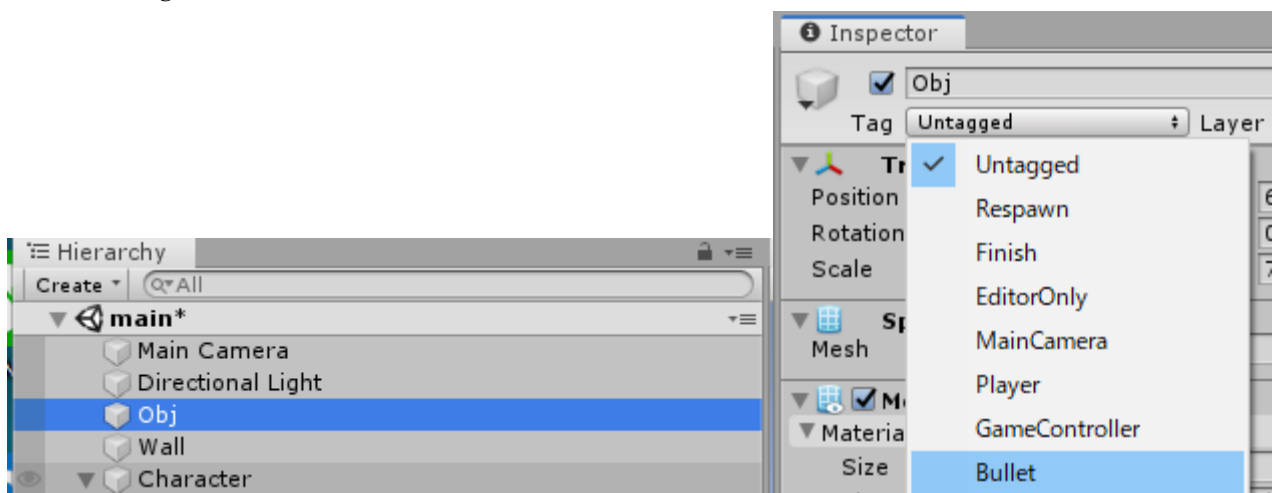
List is Empty の右下にある「+」を押下します。



タグ名は「Bullet」として Save を押下します。



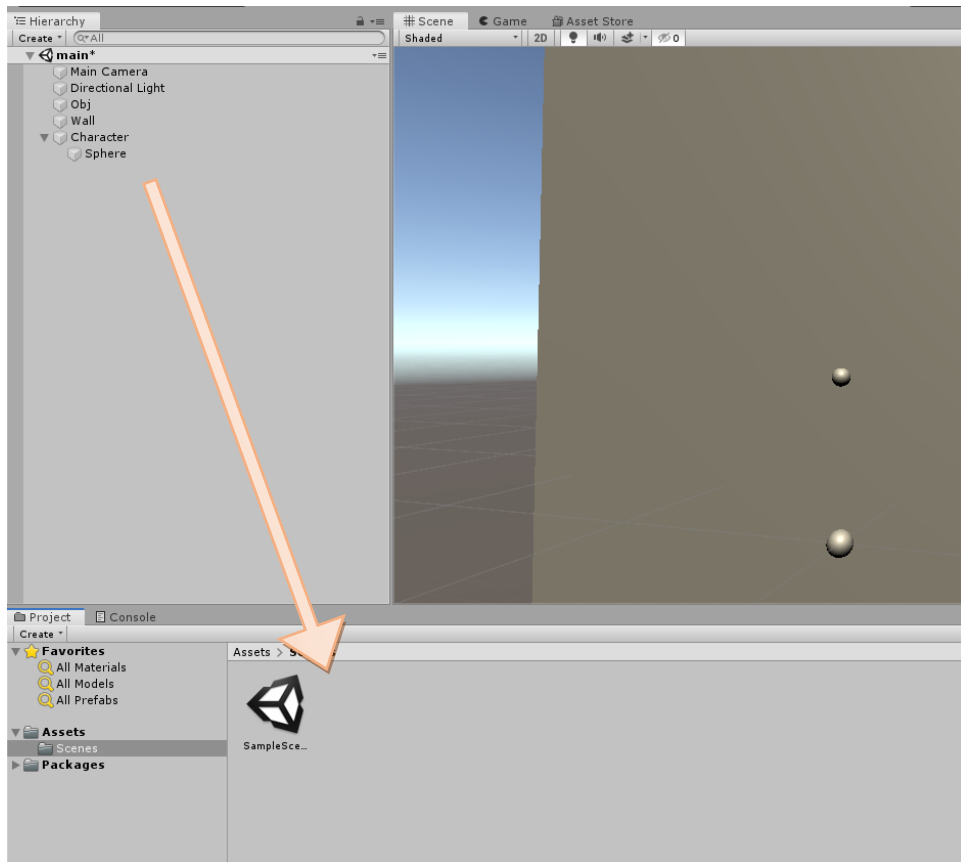
改めて Tag を押下すると「Bullet」が増えていますので、選択しましょう。



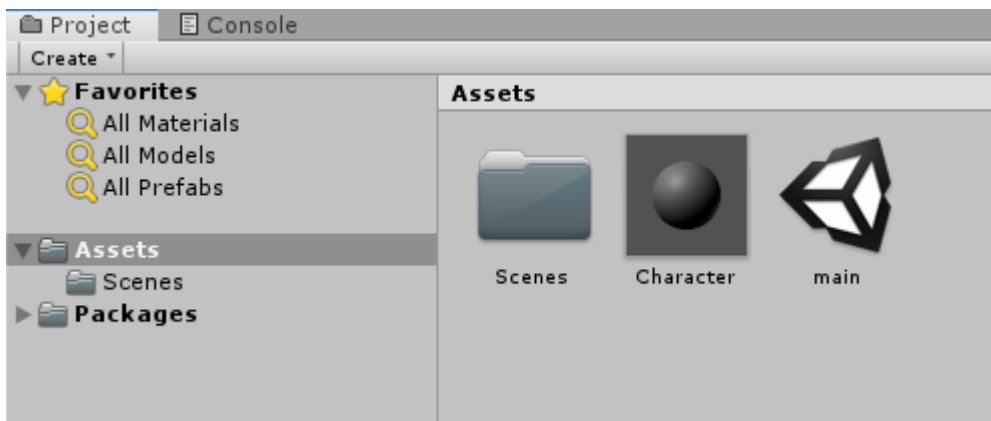
Obj にも同様に設定します。

● キャラクタを Prefab 化する。

出来上がった Character を Prefab 化します。こうすることで、後から再利用できるようになります。

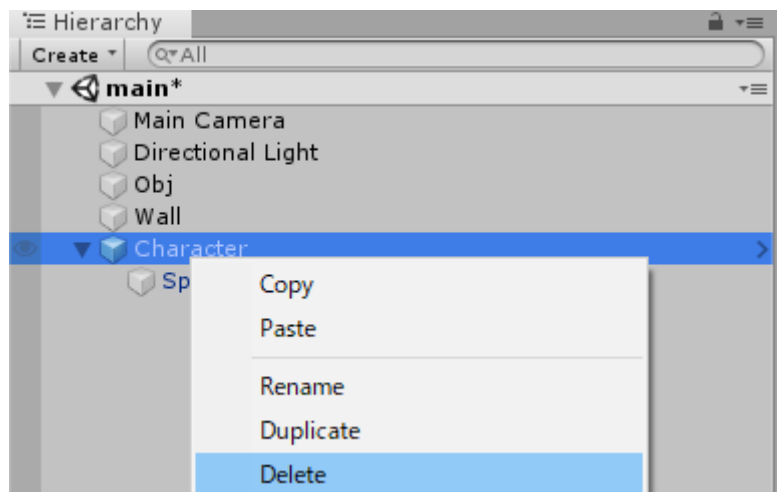


Character を下の Project ビューにドラッグ&ドロップします。



ビューに Characteraer が追加され、Prefab 化が完了しました。

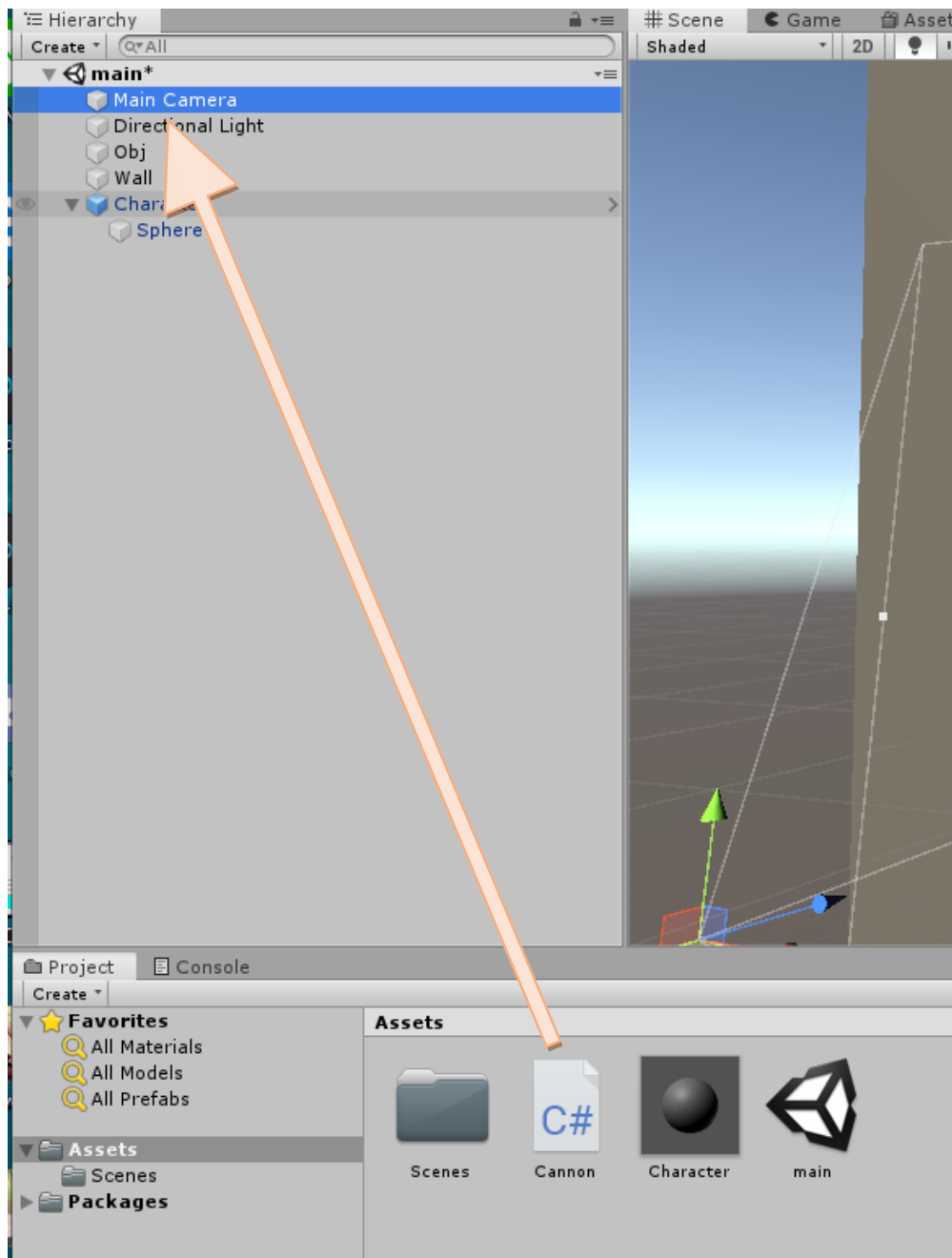
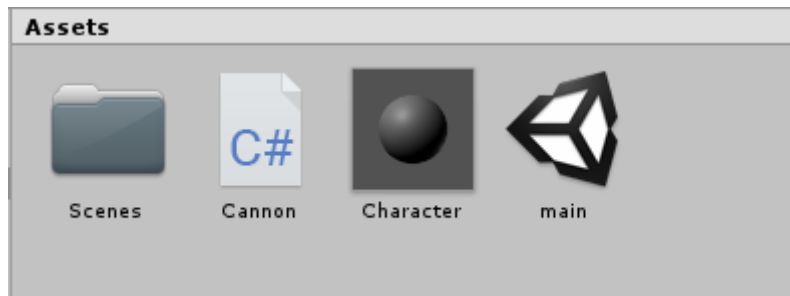
上の図は Assets フォルダで行っています。



Character はとりあえず必要なくなったので、削除します。

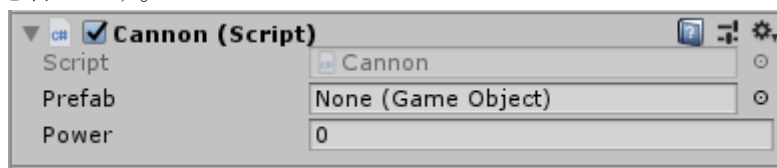
●球を発射できるようにする。

今までの作業で場面は整いました。次はスクリプトを使用して球を発射できるようにします。
素材フォルダにある、「Cannon.cs」を Project ビューに D&D します。

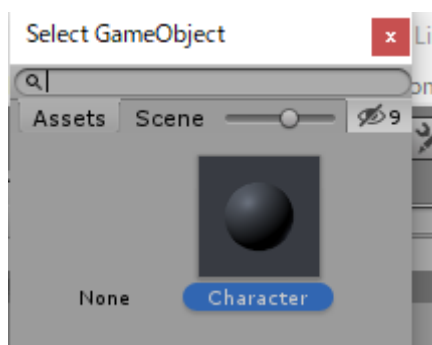


追加した「Cannon.cs」を「Main Camera」に D&D します。

打ち出す球の設定を行います。



「Main Camera」を選択してる状態で、右側に追加された Cannon(Script)から Prefab の右側にある丸いマークを選択します。

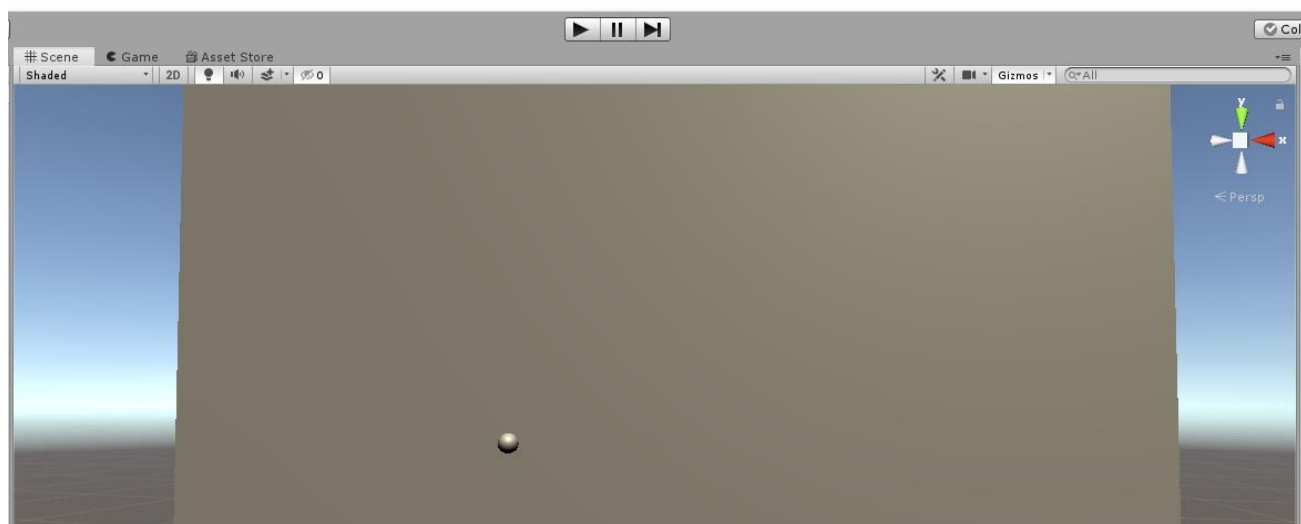


Assets タブから Character を選択します。これで設定した Character が発射されるようになります。



続けた「Power」を変更します。Power は大きいほど早く発射されます。勢いがつきすぎている場合は小さい値に変更して調整してください。

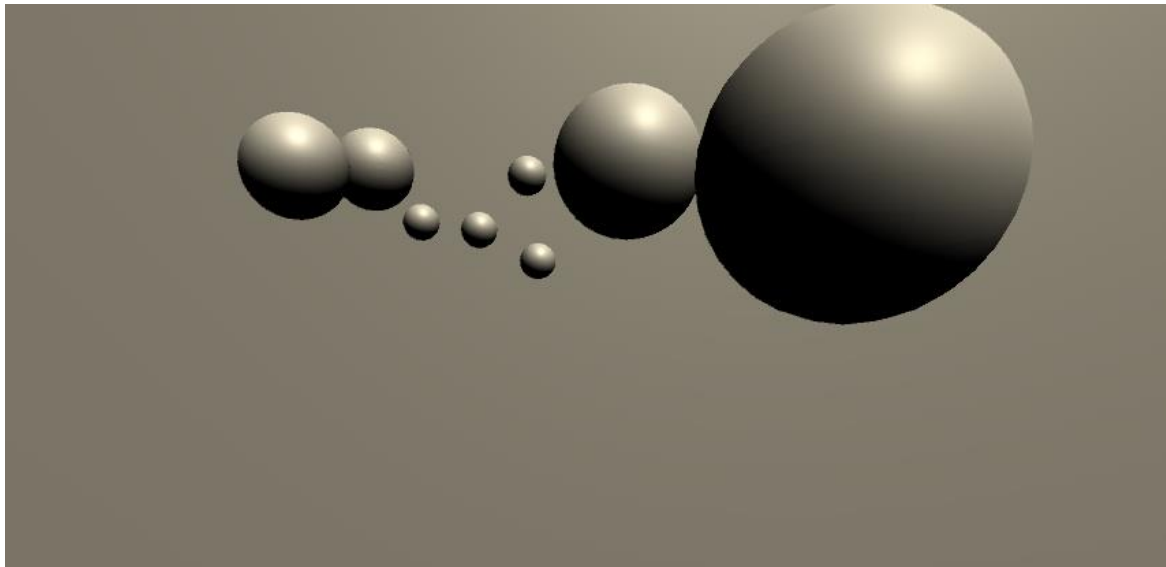
★現在の状態



球が発射できるようになったところで、一度動作を確認してみましょう。

中央上部にある再生ボタンから実行することができます。

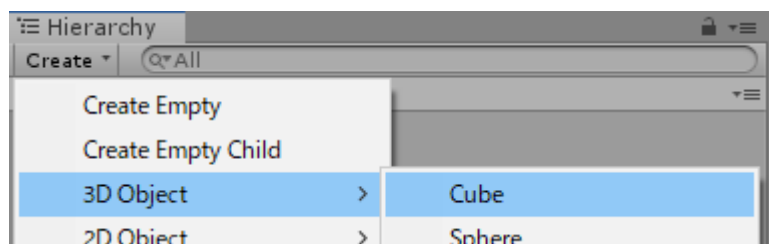
再度再生ボタンを押下すると停止します。



クリックすると球が発射されるように！(色が味気ない。。。)

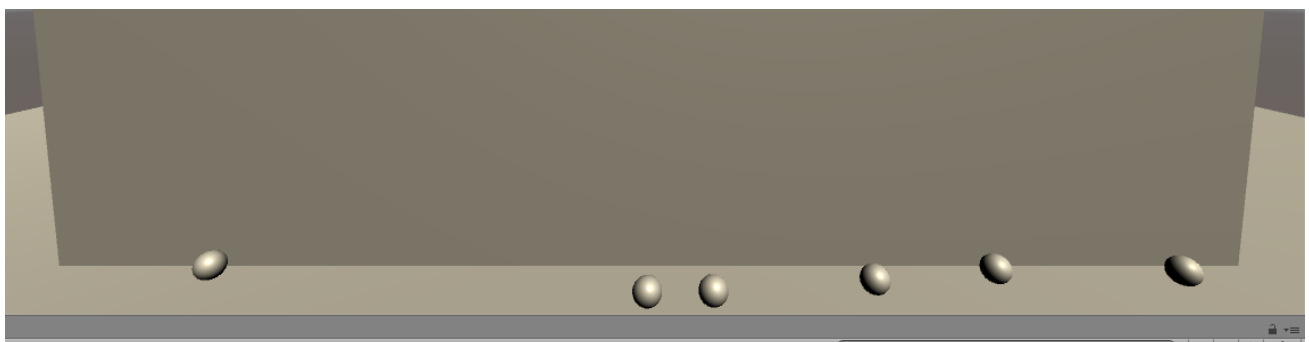
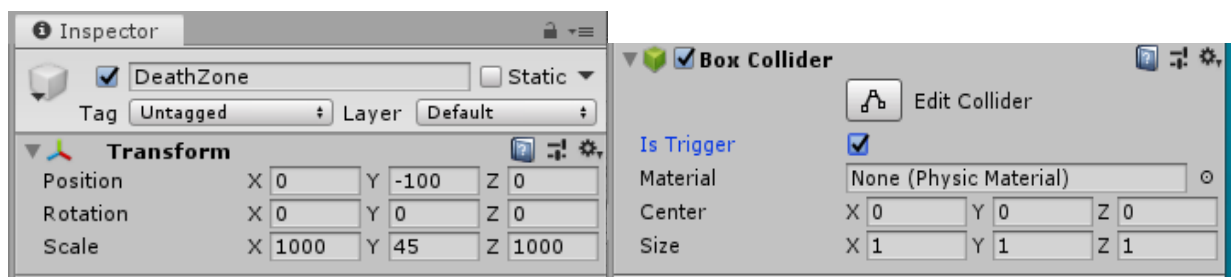
●球が落下した時の処理を作成

球が下に落下したら消えるようにオブジェクトを作成します。



[Create]>[3D Object]>[Cube]を選択します。

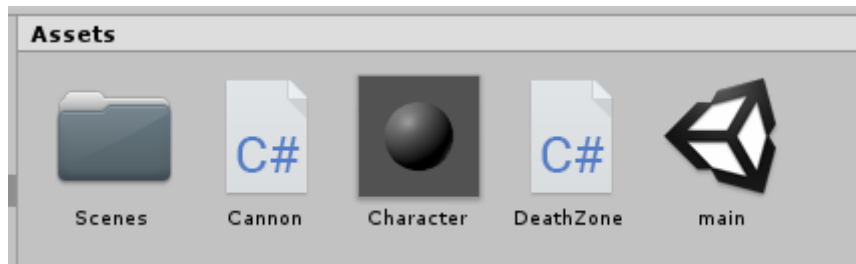
プロパティを下のように変更します。



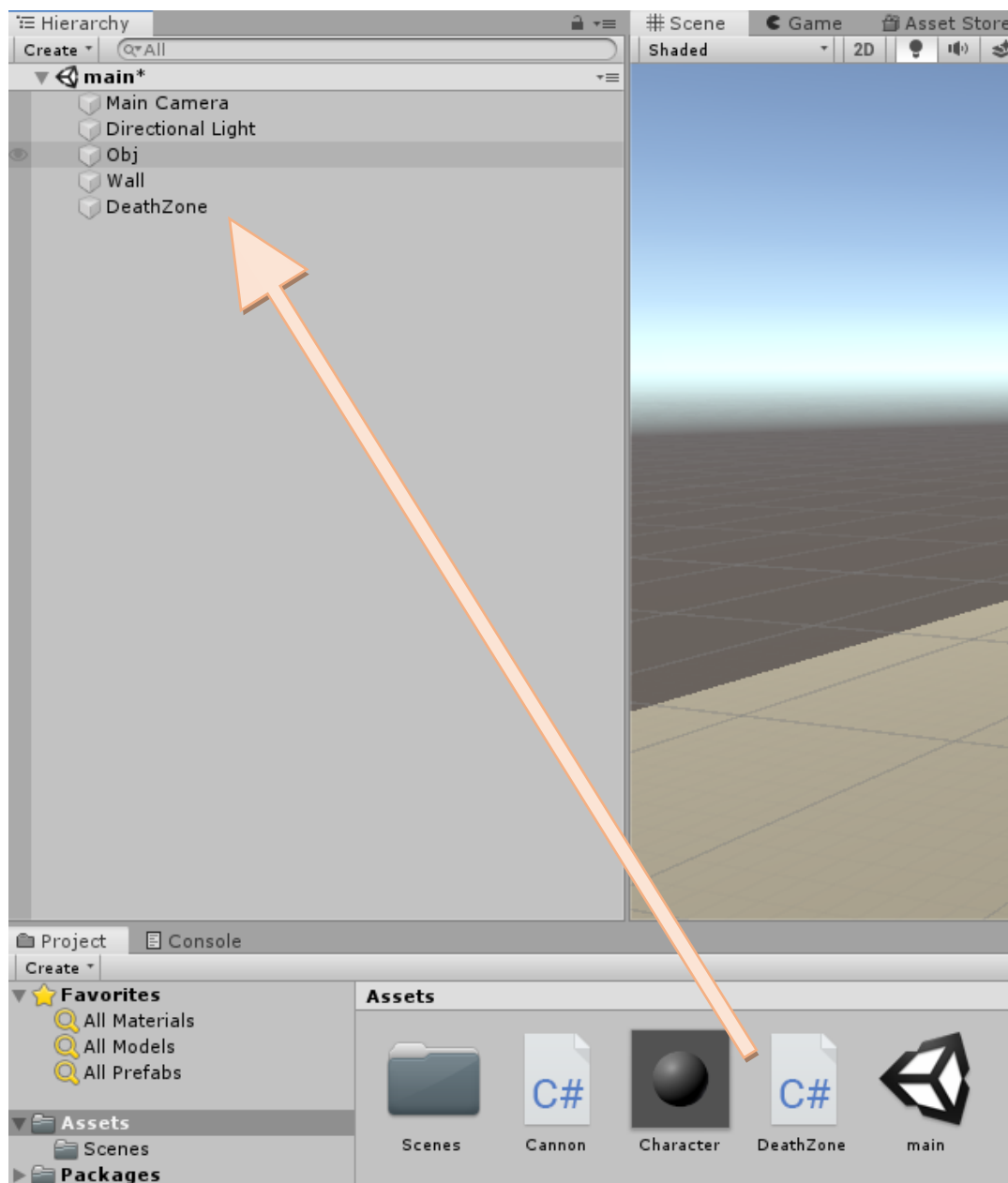
床に球が残るようになりました。

●球が消える処理を追加

素材フォルダにある、「DeathZone.cs」を Project ビューに D&D します。



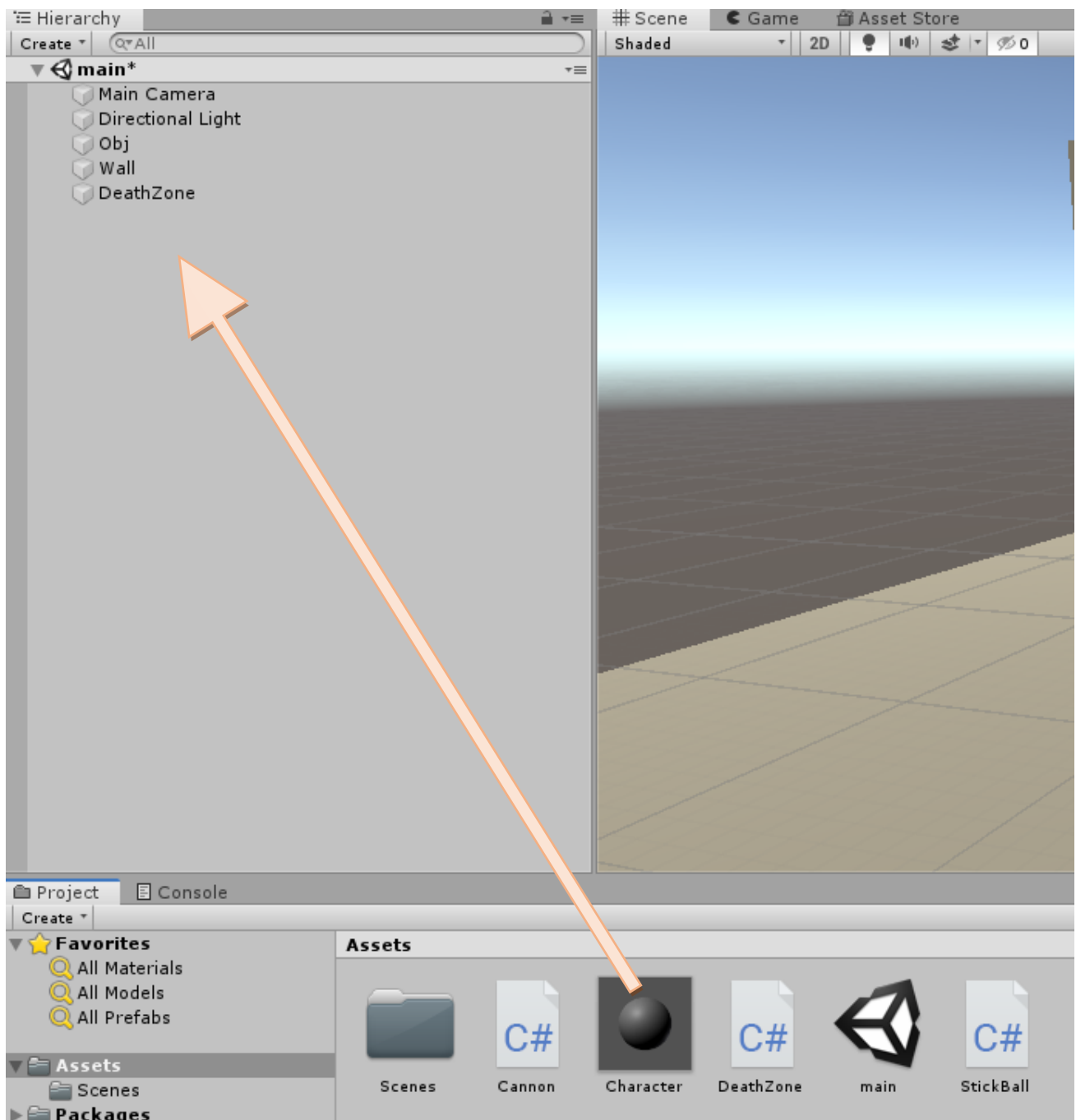
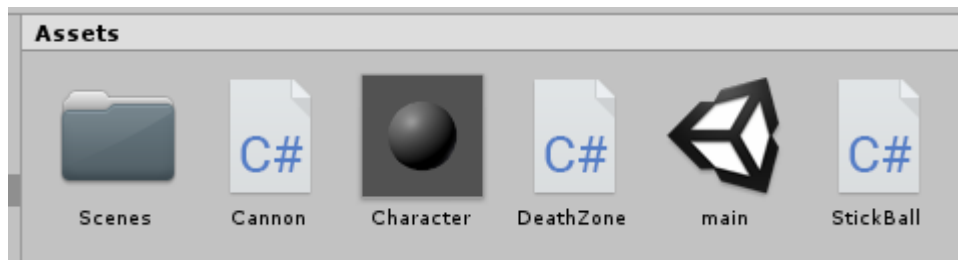
追加した[DeathZone.cs]を DeathZone に D&D します。



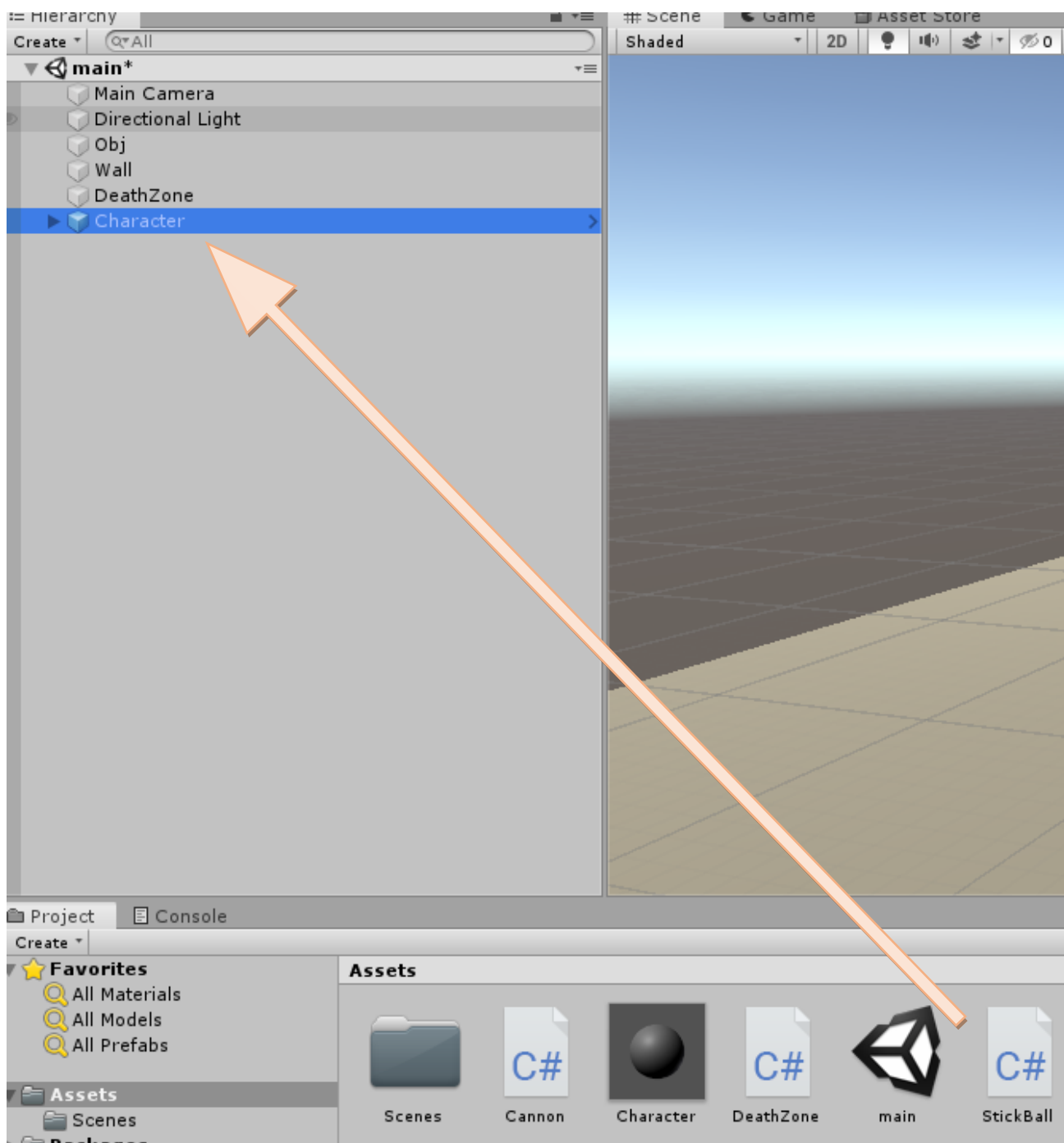
これで下に落下した球が消えるようになりました。

●球同士がくっつくようにする。

素材フォルダにある、「StickBall.cs」を Project ビューに D&D します。

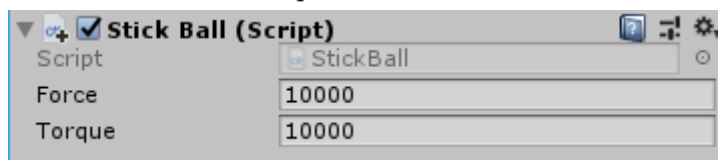


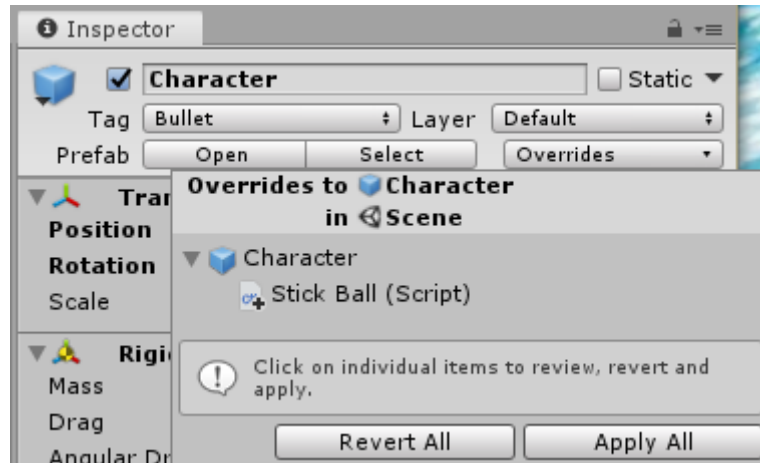
Character を Hierarchy に D&D します。



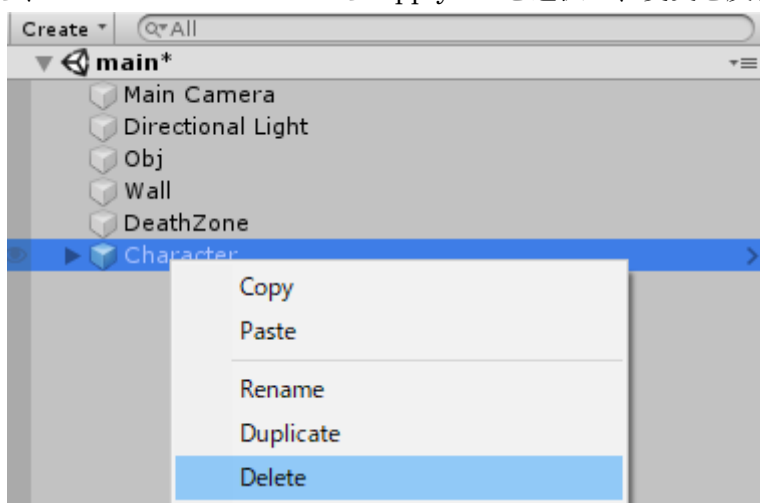
「StickBall.cs」を Character に D&D します。

Character に追加された、Stick Ball(Script)のプロパティを変更します。





ここまで変更したら、Prefab の Overrides から Apply All を選択し、変更を反映させます。

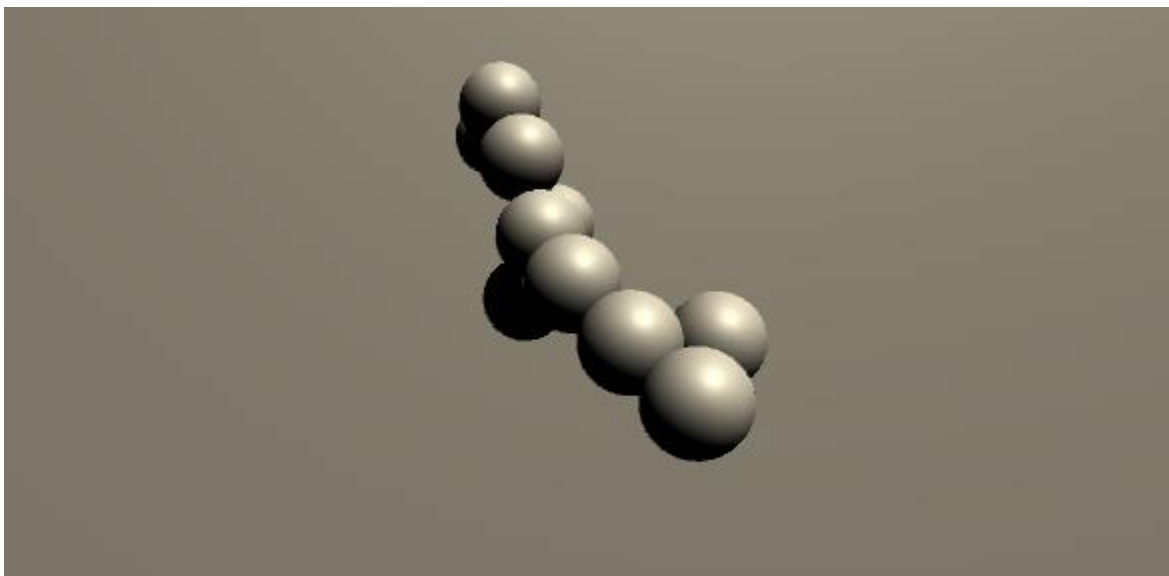


Character はまた不要になったので削除します。

これで球同士がくっつくようになりました。

※球がくっつかない場合は、Force、Torque の値を大きくしてみてください。(Apply All を忘れずに！)

これで完成になります！



うーむ。味気ない。

●ゲームを作ってみて

球を飛ばしてくっつけるという単純なゲームの作成でしたが、いかがだったでしょうか。

おそらく、この程度なの？や、こういうことができそう、など様々な感想があると思います。

ここから改良を加えていってより面白いゲームにするのもありだと思いますし、新しいゲーム作成にチャレンジするのもありだと思います。

今回で終わりにせずに、Unityの可能性を広げていってほしいと思います。