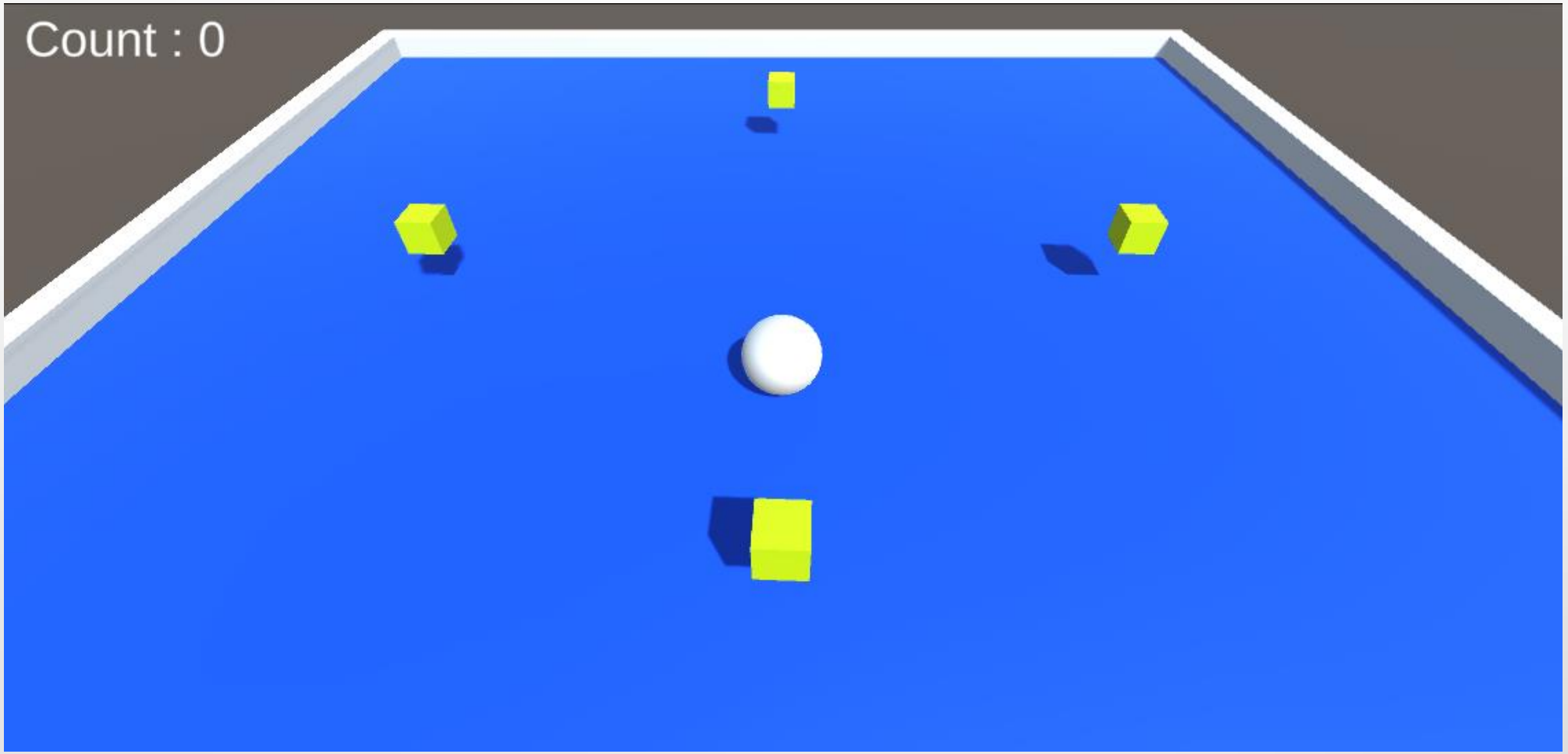
3Dゲーム作成＃1

目次

0．制作するゲームの紹介/覚えて欲しい事
1．ステージ作成
2．プレイヤー(玉)の作成
3．追尾するカメラ機能の作成
4．アイテムの作成
5．UI(テキストの作成) ← 2Dと同じ、復習

０．制作するゲームの紹介/覚えて欲しい事



０．制作するゲームの紹介/覚えて欲しい事

＜覚えて欲しいこと＞

・3Dゲーム作成時の操作方法

・RigidbodyのAddForceとVelocityの違いについて

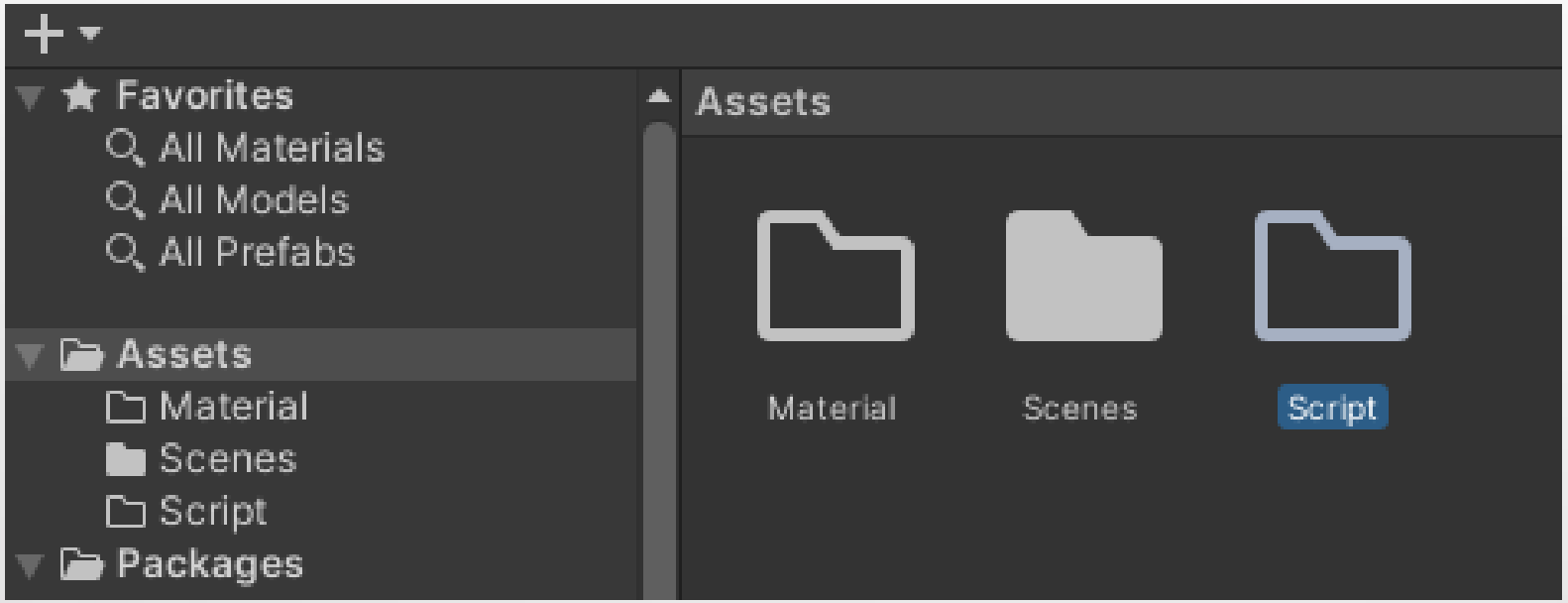
・スクリプト内におけるGameObjectの探し方

・カメラ追尾の考え方(ベクトル)

０．前準備

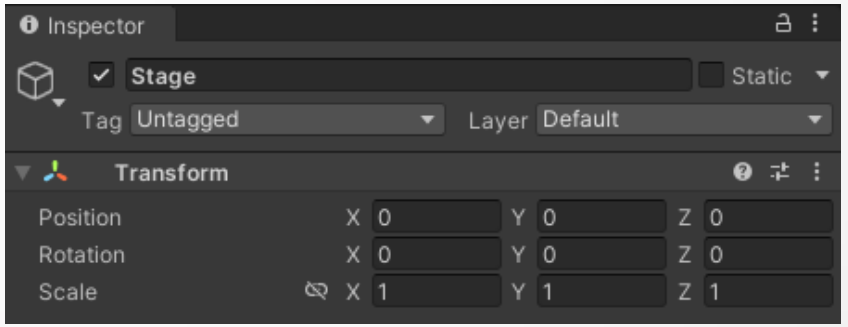
Asset直下に以下のフォルダを作成

- Material
- Script



１．ステージ作成

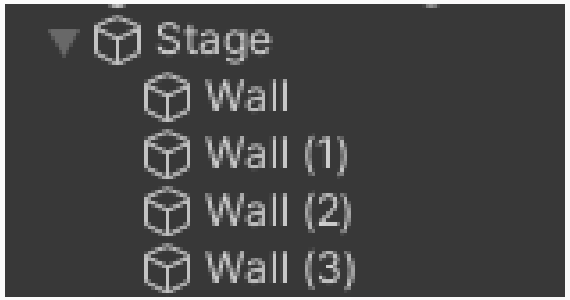
ヒエラルキー上で右クリック→CreateEmptyでStageと入力。
StageのPositionは0,0,0であることを確認。
(ステージのGameObjectをまとめる)



次に、右クリック→3D Object→Cubeを選択、以下のように設定
出来たら、Stageの中に格納する。名前はWallに設定。
Position：-10 , 0 , 0
Rotation：0 , 0 , 0
Scale：0.5 , 1 , 20.5

１．ステージ作成

WallのGameObjectを3回コピペする。



Wall(1)はPositionのXを10に変更。

Wall(2)はPositionのXを0(中心に戻す)、Z(奥行)を10に変更
RotationのYを90に変更

Wall(3)はPositionのXを0、Zを-10に変更
RotationのYを90に変更

１．ステージ作成

右クリック→3D Object→Planeを選択、XとZのScaleを2に設定する。

Materialフォルダを選択し、右クリック→Create→Materialを選択。

Material：色や光沢、テクスチャの情報によって、金属やプラスチックなどの物質の材質を表現する概念

Albedoを選択し、色を設定。設定出来たらPlaneのゲームオブジェクトにドラッグ＆ドロップ。
地面の色が変わればOK。

２．プレイヤー(玉)の作成

右クリック→3D Object→Sphereを選択、Positionは0，0.5，0を選択。
Rigidbodyをアタッチする。

プログラムを作成する。
　スクリプト名：Player

ただし、Update内の
「Vector3 movement = new Vector3(0,0,0);」
は現状のままだと動かないので、カーソルキーの入力に合わせて動くように
修正をお願いします。

２．プレイヤー(玉)の作成

＜プログラム解説＞
今まで2Dでやってきた内容と同様。(～2Dが無くなっただけ)
カーソルキーの入力をX(横方向)とZ(奥行き)に設定し、設定したベクトルを
AddForceに係数を掛けて出力することでボールが動く。

```
Vector3 movement =
new Vector3(moveHorizontal, 0, moveVertical);
```

AddForceはVelocityとは異なり、現実的な動きをするようになる。
(ゆっくり加速したり、曲がるときも摩擦で滑ったり・・・)

３．追尾するカメラの作成

カメラの位置を以下のように設定。

Position：0，6，-6

Rotation：45，0，0

今のまま再生を実施するとカメラは固定の位置となってしまう。

プログラムで書くとどのような記載になるのか、一度考えてみてください！

　スクリプト名：Camera

ヒント：ベクトルの減算を使うと出来そう・・・？

　Startメソッド → 2点の距離を算出

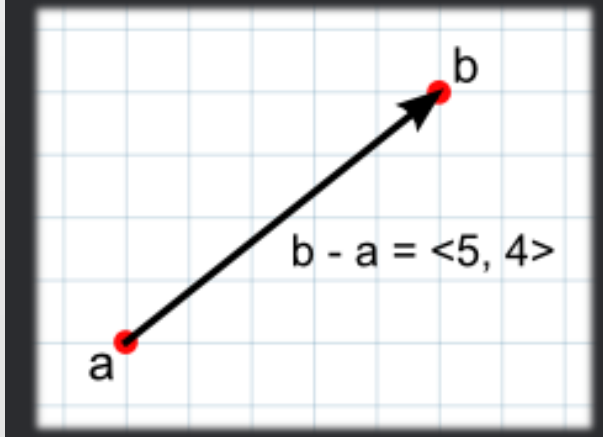
　Updateメソッド → カメラの位置を設定

３．追尾するカメラの作成

以下の点aは(0,0)、点bは(5,4)であり、
2点の減算を行うと、距離を出すことが出来る。

減算

ベクトルの減算は、ひとつのオブジェクトから別のオブジェクトへの向きと距離を出す場合にもっともよく使われます。減算の場合は、2つのパラメーターの順番は計算結果に **影響を与える** ことに注意ください。



４．アイテムの作成

右クリック→3D Object→Cubeを選択、Scaleを全て0.5に選択。
地面と同様にMaterialを作成し、アタッチする。

1秒間にY軸を起点に90度回るようなプログラムを作成し、アタッチする。
　スクリプト名：ItemRotate
指定方向に回転するにはUpdate内で以下のプログラムに、ある係数を掛けることで実現可能。

```
transform.Rotate(new Vector3(0, 90, 0) * xxx);
```

TagをItemに設定すること！

５．UI(テキストの作成)

画面上にアイテムを獲得した数を表示する。
アイテムの総数は可変とし、その上で全て獲得した場合にシーンを読み込むようにしていく。

＜やること＞

１．TextMeshProでUIを作成
２．Playerのスクリプトを変更、GameManagerのスクリプトを追加する。
３．PlayerにGameManagerをアタッチ、GameManagerにテキストをアタッチする

５．UI(テキストの作成)

＜プログラム解説＞

```
private GameObject[] item;

item = GameObject.FindGameObjectsWithTag("Item");
totalCount = item.Length;
```

Itemが付いているタグを全てitemというGameObjectに格納する。(配列)
配列の長さを取得することで、個数を取得。