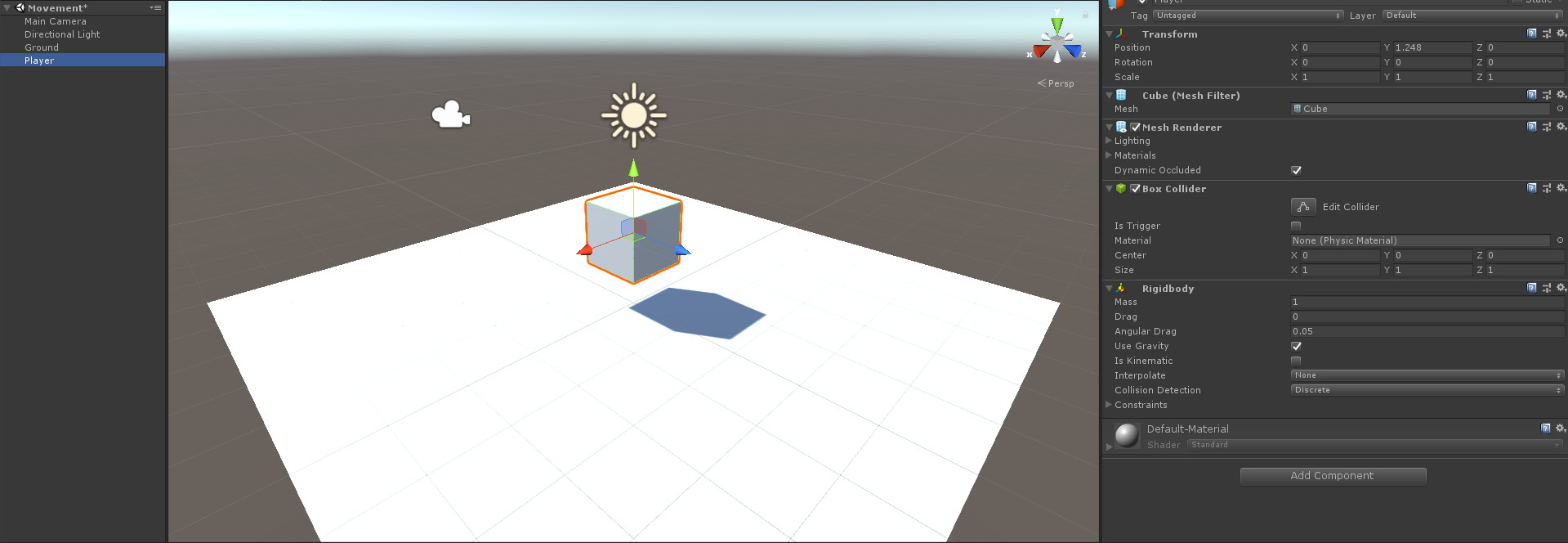
This is a simple tutorial that is going to tell you on how to add movement to the game that your player can use. I will be using the GetAxis method. So just like last time the first thing we are going to do is create a new project and we are going to call it something like **Basic Movement**.

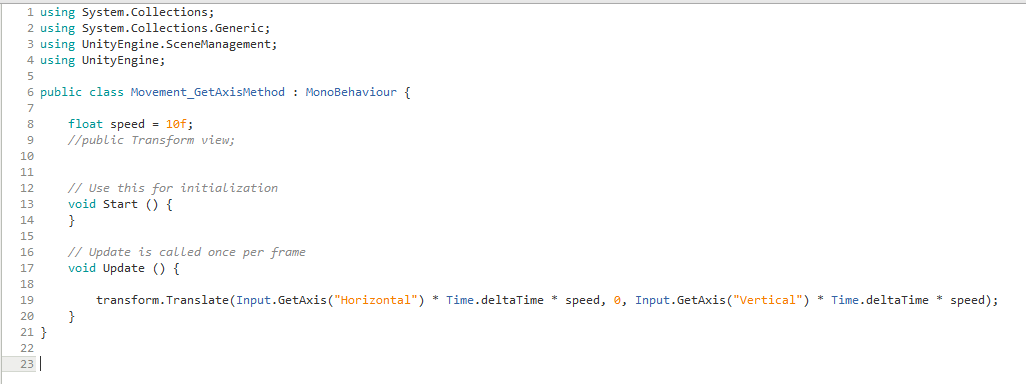
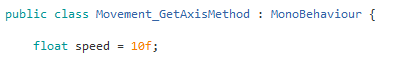
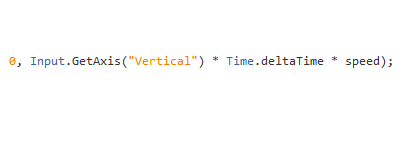
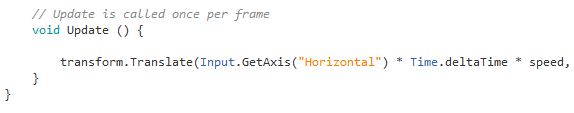
**Step by Steps:**

1. E:\UnityTutorials\Movement\PrintScreens\PS1.PNGSo we are now on a blank scene called the sample screen, now the first thing I am going to do is create a new scene as previously mentioned in the last tutorial of **Colour-Change** I prefer to make the sample scene empty as it lets me load the scenes I want to when everything is loaded rather than a scene I won’t be editing on that day. (I will be acting as if you have read my previous tutorial on **Colour-Change**, so I will assume you know some of the basics of Unity’s interface) I am going to call this scene **Movement**.
2. E:\UnityTutorials\Movement\PrintScreens\PS2.PNGSo after that lets go into how new scene, then I am going to create a new folder called **Scripts**, so I have somewhere to store scripts.
3. Then lets create a new C# Script and call it something like Movement\_GetAxisMethod and as usual double click it to open the script itself

E:\UnityTutorials\Movement\PrintScreens\PS3.PNG

1. Right before we go into the script lets edit our scene to test our script when done. First let’s create a 3D **GameObject** and use the **Plane** shape. Then let’s get something get something to represent the player themselves like a cube for instance(just as a quick not if you want to add a simple form of gravity to the player but don’t want to code it in you can just use the **RigidBody** feature when using **GetComponent** as it adds an element of gravity). Let’s just name the **Plane Ground** and the **Cube Player**.
2. E:\UnityTutorials\Movement\PrintScreens\PS5.PNGNow let’s make sure that we can tell the difference between the Player and the Ground. So let’s make a new folder called Materials, then create two new materials for them.(Choose any colours you wish)

****

1. Now let’s get back to the script, also be sure to first attach the script to the Player so you get a moment where you are wondering on why it does not work. So this is the script as a whole I will try my best to describe it.
2. So first let’s just clean this up a bit and get rid of any un-needed part of the script such as the start function and so on. I am going to keep the “// Update is called once per frame” part just as a slight reminder for myself on what it does. So after that the first part is adding a variable for the **Speed** of our **Player** as we don’t want the default which is 0. For this we need a simple float and we are then going to call it speed and make it equal something basic like 10. It is also good to note that at the end of most numbers in **C# Scripts** that you want to add an **f** to the end of it like “**10f**”.
3. ****Then underneath the void Update we want to add the image of text that is shown below. What this script those will be explained in PowerPoint that is in then Zip file as well.
4. So after all of that here is the script in text form as I tried to get an image of the script but it wouldn’t be in a decent quality. It should be working but if not send I a message and I will look into making corrections at some point.

using System.Collections;  
using System.Collections.Generic;  
using UnityEngine.SceneManagement;  
using UnityEngine;  
  
public class Movement\_GetAxisMethod : MonoBehaviour {  
  
    float speed = 10f;  
      
    *// Update is called once per frame*  
    void Update () {  
  
        transform.Translate(Input.GetAxis("Horizontal") \* Time.deltaTime \* speed, 0, Input.GetAxis("Vertical") \* Time.deltaTime \* speed);  
    }  
}