```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerMovement : MonoBehaviour
{
       public bool canMove;

       // Use this for initialization
       void Start ()
       {
              canMove = true;
       }

       // Update is called once per frame
       void Update ()
       {
              if (canMove)
              {
                     var x = Input.GetAxis ("Horizontal") * Time.deltaTime * 150.0f;
                     var z = Input.GetAxis ("Vertical") * Time.deltaTime * 3.0f;

                     transform.Rotate (0, x, 0);
                     transform.Translate (0, 0, z);
              }
       }
}
```

What It Does:

The above script is a simple player movement script.

How It Does It:

- First we need to set the bool canMove.
- In void Start() we set canMove to true. This can be set to false in the update script should you require a way to stop the player moving.
- In voidUpdate() we write var x = Input.GetAxis (“Horizontal”) * Time.deltaTime *150.0f;. What this does is set a variable “x” to collect the horizontal value from the input manager in Unity. This useful tool sets left to a -1 value and right to a +1 value. This is then run once a frame in void Update() multiplied by your speed which is 150.0f in this example.
- We do the same thing again with var z = Input.GetAxis (“Vertical”) * Time.deltaTime * 3.0f;. As above this sets the variable “z” to collect the vertical value from the input manager in Unity.
- We then set a simple transform.Rotate (0, x, 0); This sets the rotation of the player on pressing the left or right inputs.
- Below this we add a transform.Translate (0, 0, z); This sets the translation of the player (where the player is in the scene) to the forward and backward inputs.