LESSON IDEAS

APPROACHES TO TEACH DIGITAL TECHNOLOGIES

# Level up: Explaining Unity code

The following code is taken from the Roll-a-ball tutorial[1] provided with Unity.

Use this document to annotate the code with your notes so that you can understand them, and apply this to new programs.

### Code

```
using UnityEngine;
using System.Collections;

public class PlayerController : MonoBehaviour {

    public float speed;

    private Rigidbody rb;

    void Start ()
    {
        rb = GetComponent<Rigidbody>();
    }

    void FixedUpdate ()
    {
        float moveHorizontal = Input.GetAxis ("Horizontal");
        float moveVertical = Input.GetAxis ("Vertical");

        Vector3 movement = new Vector3 (moveHorizontal, 0.0f, moveVertical);

        rb.AddForce (movement * speed);
    }
}
```

[1] https://unity3d.com/learn/tutorials/projects/roll-ball-tutorial/moving-camera?playlist=17141

# Level up: Explaining Unity code – Answers

```
1.  using UnityEngine;
2.  using System.Collections;//these two lines import program libraries that we will be using.
3.
4.  public class PlayerController : MonoBehaviour {
5.
6.      public float speed;//this declares a variable of float data type called speed. This is a public variable that is then accessible outside of this class.
7.
8.      private Rigidbody rb;//this declares a private variable, of type Rigidbody (this is a special data type used by Unity) called rb. The fact that this is private means that it is only accessible inside this class.
9.
10.     void Start ()//what is between the {} is run on starting the program only.
11.     {
12.         rb = GetComponent<Rigidbody>();//this calls the RigidBody that exists in the program already and stores it in the variable created in line 8.
13.     }//end of the start procedure.
14.
15.     void FixedUpdate ()//runs before physics calculations are called
16.     {
17.         float moveHorizontal = Input.GetAxis ("Horizontal");//gets the position of the Horizontal input (set by the input manager)  and assigns this to a variable (moveHorizontal)
18.         float moveVertical = Input.GetAxis ("Vertical");//gets the position of the vertical input (set by the input manager)  and assigns this to a variable (moveVertical)
19.
20.         Vector3 movement = new Vector3 (moveHorizontal, 0.0f, moveVertical);  //creates an object called movement, that is a type of Vector (x, y, z) and uses the variables from line 17 and 18
21.
22.         rb.AddForce (movement * speed);  //uses a pre-defined function called addForce to create movement on the horizontal and vertical (set in line 20) at a set speed.
23.     }//end of the FixedUpdate Procedure.
24. }//end start
```