UNIVERSITY OF MALTA
Institute of Digital Games

Creating Virtual Worlds

Lecture 6

Scripts and Components II

Our topics today …

- **Short Recap Round**
  Data types, operators, and methods
- **Arrays and Loops**
  Defining data lists and repeating code blocks
- **The Structure of Unity scripts**
  A look at the basics of object oriented programming
- **Live Walkthrough**
  Example: Creating a warning light

Review: Data Types

- Integers
- Floats
- Booleans
- Strings
- Vector2 / Vector3

Review: Variables and Operators

- Designation of a variable (=)
- Arithmetic operators (+ — / *)
- Relational operators (== > < !=)
- Logical operators (&& || !)

Review: Methods

- Void
- Methods that return a value
- Methods with parameters

```
void Boo () {
    // Do something
}

Boo();  // calling the method
```

Speaking of methods!

```
Debug.Log("Hello!");
```

Debug.Log() displays its input as console text!

```
Mathf.Abs(-10);
```

Returns the absolute value

```
string text = "10";
int number = int.Parse(text);
```

Convert a string to an int

```
string text = number.ToString();
```

... and vice-versa

Arrays

Arrays are series of values, stored in one variable. Think of them as an ordered list from which individual values may be read or written to.

```
int[] somearray = {5, 3, 7};
somearray[1];  // equals 3
```

In C#, the data type of an array is determined at its creation; and all values in an array need to have the same data type.

To access individual values of an array, we refer to the index of the value inside the array.

Index numbers start from 0, indicating the first item in the array.

```
string[] names = {"Archer", "Baboo", "Pam"};
bool[] switches;
float[] moneyz;
Vector3[] positions;
```

Loops

While Loops

Execute a block of code repeatedly as long as the check condition is true. This can very easily cause Unity to lock up!

```
int counter = 0;
while (counter < 10) {
   Debug.Log("Current Loop is: " + counter);
   counter = counter + 1;
}
```

For Loops

Execute a block of code repeatedly, with an initialiser, boolean check, and iterator.

```
for (int count = 0; count < 10; count += 1) {
  Debug.Log("Current Loop is: " + count);
}
```

Note that the variable ‘count’ is only set for use within the loop!

Foreach Loops

Execute a block of code for each item in an array.

```
string[] types = {"rock", "paper", "scissors"};

foreach (var element in types) {
  if (element != "rock") {
    Debug.Log(element);
  }
}
```

The Structure of Unity Scripts

By default, each new script already starts with a few lines:

```
using UnityEngine;
using System.Collections;

public class NewScript : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

}
```

The most important methods of a Unity script are *Start ()* **and** *Update()*

There are a few other predefined methods, such as *OnEnable()* **or** *OnTriggerStay()*

```
1   using UnityEngine;
2   using System.Collections;
3
4   public class Penguin : MonoBehaviour {
5
6     public int age;
7     public string name;
8     public float height;
9
10    public void Speak (string saythis) {
11      Debug.Log(name + " says: " + saythis);
12    }
13
14  }
```


```
1   using UnityEngine;
2   using System.Collections;
3
4   public class PenguinMaster : MonoBehaviour {
5
6     public Penguin peng;
7
8     void Start () {
9       Debug.Log(peng.age);
10
11      peng.Speak ("Sqweak");
12    }
13  }
```

Let’s do more of this in practice