

# Builtin functions vs calculating something manually

Let's say you want to calculate the distance between two vectors.

Some of you would probably write something like this :

```
distance = Vector3.Distance(a, b);
```

Others might write something like this (which is how Unity implemented the `Distance` method) :

```
float num = a.x - b.x;
float num2 = a.y - b.y;
float num3 = a.z - b.z;
distance = (float)System.Math.Sqrt(num * num
    + num2 * num2
    + num3 * num3);
```

In regular C#, that wouldn't make a difference, because C# code is compiled and both examples above would probably get compiled into something similar.

That's not the case in Udon, each line of code has an execution time because Udon is an interpreted programming language, so the first example would call a compiled C# function (fast), the second example would calculate the distance in Udon (slow)

This benchmark is mostly to show you that it is better to use builtin function instead of calculating something manually. Before implementing a function, make sure that the function hasn't already been implemented yet, the `Mathf` class implements many math functions!

I executed two methods `Benchmark1` and `Benchmark2` and compared their execution time.

```
private int RandomVal()
{
    return Random.Range(0, 100);
}
```

```

}
public override void Benchmark1()
{
    float distance;
    for (int i = 0; i < 10000; i++)
    {
        Vector3 a = new Vector3(RandomVal(), RandomVal(), RandomVal());
        Vector3 b = new Vector3(RandomVal(), RandomVal(), RandomVal());
        float num = a.x - b.x;
        float num2 = a.y - b.y;
        float num3 = a.z - b.z;
        distance = (float)System.Math.Sqrt(num * num
            + num2 * num2
            + num3 * num3);
    }
}

public override void Benchmark2()
{
    float distance;
    for (int i = 0; i < 10000; i++)
    {
        Vector3 a = new Vector3(RandomVal(), RandomVal(), RandomVal());
        Vector3 b = new Vector3(RandomVal(), RandomVal(), RandomVal());
        distance = Vector3.Distance(a, b);
    }
}

```

Both methods calculate distances, but the second method calls the builtin Distance method.

**B1 : 166.699 ms (1.79 times slower)**

**B2 : 92.678 ms**

**Use builtin functions as much as you can!**

Revision #2

Created 15 February 2023 19:57:00 by MyroP

Updated 16 February 2023 20:50:01 by MyroP