

## Company Style Guide

*Please note this style guide is for the code review part of the mid-level Unity Developer test.*

1. Code should be:
  - a. Fit for purpose/bug free
  - b. Easily readable by humans
  - c. Machine-efficient
  - d. Scalable
2. Give everything useful, non-cryptic names except when its local to a function/algorithm
3. Explicitly state the access modifier on all declarations e.g. private, public, internal etc.
4. For variables/properties we use the following declaration styling to help understand variables at a glance:

```
using System;
using UnityEngine;

public class MyClass : MonoBehaviour
{
    private float _foo;
    protected float _foo;
    public float foo;

    private static float _Foo;
    public static float Foo;
}
```

*So the capitalisation declares whether the variable is static or not, and the underscore prefix declares whether the variable is private or public.*

5. PascalCasing should be used for function and class names
6. Use **[SerializeField] private** where possible if your variable does not need to be public, but you wish to provide Editor access to the variable;
7. Open braces should always be at the beginning of a new line, after the statement that begins the block
8. Please use tabs for consistent spacing and ensure whitespace between lines is consistent

```
if (someExpression)
{
    DoSomething();
}
else
{
    DoSomethingElse();
}
```