Unit 2: Scripting

Lesson 2: Scope and access modifiers

Activity 3 (10’ minutes): Gap text

Fill the gaps with the words you’ve heard in the video.

The scope of a variable is the area in code which the variable can be used in.

A variable is said to be local to the place in code that it can be used. Code blocks are generally what defines a variable’s scope, and these are denoted by braces.

Variables defined in the class, as opposed to those declared within a function, have an access modifier attributed to them. That modifier is a keyword placed before a datatype in a declaration and its purpose is to define where the variable or function can be seen from. As a general rule, if other scripts need access to a variable or function then it should be “public”, otherwise it should be “private”.

Declaring a variable as public means that it can be accessed from outside the class. It also means that the variable is shown and editable on the component in the inspector. This allows the user to edit the variable whilst they test the game.

Imagine, for example, a value controls the speed of a car. It would be nice to tweak that variable whilst testing it without having to stop, edit the script and play again. As such, it makes sense to have this be a public variable. Note that, if a variable is initialised in the class to a default value, then it will still be overridden by the value that’s written in the inspector.

Private variables can only be edited from within the class. In C# “private” is the default access modifier for any variable that doesn’t have it specified. Setting variables and functions to public can also mean that you can access them from other scripts.

The Intellisense in Monodevelop, or any other code editor, always shows only the variables that are actually available to us.