Unit 2: Scripting

Lesson 5: Getting components and GameObjects

Activity 3 (⏱ 15′ minutes): Gap text

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

In Unity, a script is considered as a custom component, and often you will need to access other scripts attached to the same GameObject or even on other GameObjects. You can access other scripts and components using GetComponent.

Note that the references to other scripts are simply variables whose type is the name of the script. This is because what we are actually referencing is an instance of the class defined in the script.

In the Awake function, we initialise our variables. The GetComponent function uses a slightly different style of call to what we are used to. We use a pair of angle brackets before the normal brackets, these symbols are for taking a type as a parameter.

GetComponent will return a reference to any component of the type specified on the GameObject is called upon.

GetComponent however is expensive in terms of processing power and should be called as infrequently as possible. It's often good practice to call it in the Awake or Start functions, or only once when it's first needed.

Tags are a way of identifying GameObjects in Unity. For example, you may have an object called Ork or Tank, but these could all be tagged "Enemy", and in your code you could check for any objects that have the tag Enemy.

To assign a tag to an object, select it and use the drop-down menu at the top of the inspector. If the tag you want is not already present, then add a new tag from the tag manager.

There are a number of functions in code, which will allow you to find objects with tags; the simplest one of these is GameObject.FindWithTag, which allows you to specify a string with the name of the tag inside it. Likewise, you could find multiple objects with the same tag by using FindGameObjectsWithTag.