Unity version: [2020.3.25](https://unity3d.com/unity/whats-new/2020.3.25)

# Dependency Injection

Increases complexity and creates hard dependencies, but can catch dependency errors immediately, provides structure to explicitly construct scripts in the desired order, and replaces the need to insert dependencies through the inspector.

## Construct Method

1. Classes should receive it’s pre-constructed dependencies through its constructor or the custom Construct method parameters if it is a Monobehaviour
2. The Construct method most of the time replaces the purpose of the Awake and Start Monobehaviour methods for getting the GameObject’s components
3. The Construct method body should explicitly check every incoming dependency to catch null dependencies immediately

## **Higher Level Organization**

1. A pseudo composition root Monobehaviour called GameManager (or similar name) should on Awake find/receive the next lower scope level of classes and Construct them
2. The GameManager should become the root of a dependency tree composed of all classes needed on Awake and have no looping dependencies
3. A Manager class should be made for a GameObject or system composed of several scripts, or many instances of an object.
   1. These managers should have the sole purpose of passing dependencies and act as the root of the script collection
   2. They may also double as controllers (classes that take input and cause game actions in response) or factories/ instance trackers (classes that create and track all instances of a type)
4. If the dependency tree nodes have read access for child nodes, a child class may traverse the tree upward and back down to retrieve a reference to a class
   1. This may be dangerous as it creates crossing dependancy paths
   2. The retrieved class must have been constructed prior to retrieval