Ignition / ObjectPooler

Namespace **Playniax.Ignition**

Script can be found in **Assets/Playniax/Framework/Ignition/Scripts/ObjectPooler.cs**

Class ObjectPooler

Description

Instantiate() and Destroy() are useful and necessary methods during gameplay. Each generally requires minimal CPU time. However, for objects created during gameplay that have a short lifespan and get destroyed in vast numbers per second, the CPU needs to allocate considerably more time. Additionally, Unity uses Garbage Collection to deallocate memory that's no longer in use. Repeated calls to Destroy() frequently trigger this task, and it has a knack for slowing down CPUs and introducing pauses to gameplay. This behavior is critical in resource-constrained environments such as mobile devices and web builds. Object pooling is where you pre-instantiate all the objects you'll need at any specific moment before gameplay — for instance, during a loading screen. Instead of creating new objects and destroying old ones during gameplay, your game reuses objects from a 'pool'.

Public fields	Description
`GameObject prefab`	The object to pre-instantiate.
`int initialize`	Number of objects to pre-instantiate.

Public Methods	Description
`GameObject GetAvailableObject(bool allowGrowth = true)`	Returns the first available object.