

Ignition / GameObjectPooler

Namespace **Playniax.Ignition**

Inherits from **MonoBehaviour**

Script can be found in **Assets/Playniax/Framework/Ignition/Scripts (MonoBehaviour)/GameObjectPooler.cs**

Class GameObjectPooler

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.
bool allowGrowth	Whether to allow the number instantiated objects to grow or not.
bool hideInHierarchy	Whether to hide the objects in the hierarchy or not.

Example

Example can be found in **Assets/Playniax/Framework/Ignition/Examples/01 - Framework/GameObjectPooler.unity**

Example script can be found in **Assets/Playniax/Framework/Ignition/Examples/01 - Framework/Scripts (MonoBehaviour)/GameObjectPooler_Example.cs**

```
using UnityEngine;
using Playniax.Ignition;

public class GameObjectPooler_Example : MonoBehaviour
{
    [Tooltip("Prefab to be used")]
    public GameObject prefab;
    [Tooltip("Parent to be used")]
    public Transform parent;
    [Tooltip("Timer settings")]
```

```
public Timer timer;

void Update()
{
    if (timer.Update())
    {
        // See if object is available.
        var bullet = GameObjectPooler.GetAvailableObject(prefab);
        // Was created?
        if (bullet)
        {
            // Override parent.
            if (parent) bullet.transform.parent = parent;
            // Set position.
            bullet.transform.position = transform.position;
            // Don't forget to activate.
            bullet.SetActive(true);
        }
    }
}
```