

Pooling Manager Documentation

Introduction

There is no need whatsoever to write this document, but the Asset Store reviewer rejected the package because lack of **“some form of written documentation describing the functionality and intended use of your project”**, so here I am going to explain to you the very things that are already very well explained in the scripts’ comments. You might want to skip this document altogether and go to have a look to the demo scene which is more than enough.

Intended use of this project

Well, I guess you have read the description on the Asset Store, so you know this already... but it seems that I have to tell you this again: “This class allows you to pool any object without need to add a pooling script to the objects you want to pool. Moreover, this class allows you to pool game objects but declare the pool as a pool of components, so that (for instance) you retrieve from the pool directly the Rigidbody (or any component suits you to pool).”

Functionality

The class PoolingSystem encompasses all the functionalities. Basically you create your pool like this:

```
spheres = new PoolingSystem<GameObject>(spherePrefab, 10);
```

or using a different type assuming that it’s a component in the root prefabricate:

```
capsules = new PoolingSystem<Rigidbody>(capsulePrefab, 10);
```

Once you have those pools (of 10 initial objects in this example) instead of using “Instantiate” to get a sphere or a capsule you use

```
spheres.GetElement ();          // This retrieves a GameObject (see above declaration)
```

or

```
capsules.GetElement ();          // This retrieves a Rigidbody (see above declaration)
```

And when it comes to dispose of them, instead of using “Destroy” we do this:

```
spheres.ReleaseElement (gameObject, true);
```

or

```
capsules.ReleaseElement (rigidbody, true);
```

This way the Manager takes care of all and you speed up your game avoiding to Instantiate and Destroy bullets, balls and so on.

Basically that’s it. Please read carefully the comments in the scripts: those are quite explanatory, and being in the context of a script they represent the best way for you to understand how this works behind the scene.

If you need help do not hesitate to contact me at pino@dftgames.com

Enjoy!