Contents

Getting Started

Getting started with Pure Pool is incredibly simple. Pick one of the managers below, and add it to a GameObject in your scene!

Managers

Rather than deal with individual object pools, it's much easier to work with a manager. A manager is a collection of pools, and can automatically create pools when one doesn't exist. Three different types of manager are included, along with three static classes to help access them:

GameObjectPoolManager

This manager can be placed in your scene, and manages a collection of `GameObjectPool` pools. This is the most commonly-used manager, and is a simple solution for replacing Instantiate and Destroy of game objects. It uses object/prefab references, in the same way that Object.Instantiate does.

To facilitate easier interaction with the pool, you can use the `StaticPoolManager` class. It requires a `GameObjectPoolManager` to be present in your scene, and will automatically create one if it cannot be found.

NamedGameObjectPoolManager

This manager can be placed in your scene and requires a `GameObjectPoolManager` to also be present. It provides access to the collection of `GameObjectPool` pools using **string names**, rather than object/prefab references.

The static equivalent is the `StaticNamedPoolManager` class. It requires a `NamedGameObjectPoolManager` to be present in your scene, and will automatically create one if it cannot be found.

ComponentPoolManager

This manager can be placed in your scene, and manages a collection of `ComponentPool` pools. Component pools are harder to use without problems, and should only be used when you're sure you know what you're doing. Stick to `GameObjectPoolManager` until then.

The static equivalent is the `StaticComponentPoolManager` class. It requires a `ComponentPoolManager` to be present in your scene, and will automatically create one if it cannot be found.