## Manual to the

# Object Pool - multitype pooling system

FALLencake studio

### Summary

- 1. What is it?
- 2. How to use it?
- 3. Demos

#### What is it?

Creation of a lot of game objects, eg. coins, bullets, enemies and even environment elements, we need to often reuse, is a very expensive operation.

Object Pool will cache all those objects for the using of next time. It will reduce the overhead of a memory-intensive application.

Directory: FC Object Pooler > Scripts > Fallencake > Tools > ObjectPool

FCObjectPooler.cs - a basic pooler class, uses to be extended for SingleTypeObjectPooler and MultiTypeObjectPooler classes

FCSingleTypeObjectPooler.cs - a simple object pooler that can output only a single type of objects

FCMultiTypeObjectPooler.cs - a complex object pooler that can output multiple types of objects

FCPoolableObject.cs - this class should be added to the object that you need to be pooled from Object Pooler

#### **MultiTypeObjectPooler Methods** you can pull objects from the pool with:

- SequentialOrder the pooler will get all object of current type before moving to the next type object
- InTurnTypeBased objects will be spawned in the order has setted in the inspector (from top to bottom)
- **InTurnPriorityBased** tries to get an object from the pool in order based on the type Priority value, probability to be get picked depends on the Priority value too (the larger the Priority value, the higher the probability it'll be chosen)
- RandomPoolBased the pooler will get one object from the whole pool, at random, each object has equal chances to be chosen
- RandomTypeBased randomly chooses the type of the object (the larger the amount of object of the specific type, the higher the chances it'll be chosen)
- **RandomPriorityBased** randomly chooses one object from the pool, based on Priority value of its type probability (the larger the Priority, the higher the chances it'll be chosen)

#### How to use it?

- 1. Add FCSingleTypeObjectPooler or FCMultiTypeObjectPooler script to any gameobject you need.
- 2. Set any game object you need as the **Object To Pool** parameter (**GameObject ObjectToPool**). Object you need to pool should have **FCPoolableObject.cs** component, add this script aforehand.
- 3. Set the **Pool Size** parameter (int **PoolSize**). This is the amount of objects that will be added to the pool.

#### How to Get a GameObject from the Pool?

- Use the method **GetPooledGameObject()** to return a pooled gameobject

#### How to Destroy / Deactivate a pooled GameObject ?

- Use the method Destroy() to deactivate the instance to reuse it

**Poolable Objects** have a **Life Time** parameter (**float lifeTime**) - the time, in seconds, that the pooled object will be active on scene. If set to any positive value, it'll be set inactive after that time. Value of 0 means the object will live forever.

Pool Can Increase parameter (bool PoolCanIncrease) - if true, the pool will automatically add more objects to the itself if needed

#### **Demos**

FC Object Pooler > Demo folder contains Animations, Prefabs, Scenes and Sprites folders needed to show the Object Pooler work.

There are two demo scenes (Coins Demo and Runner Demo) in the folder FC Object Pooler > Demo > Scenes showcases the usage of the Object Pooler scripts that will help you to create the gorgeous worlds!

