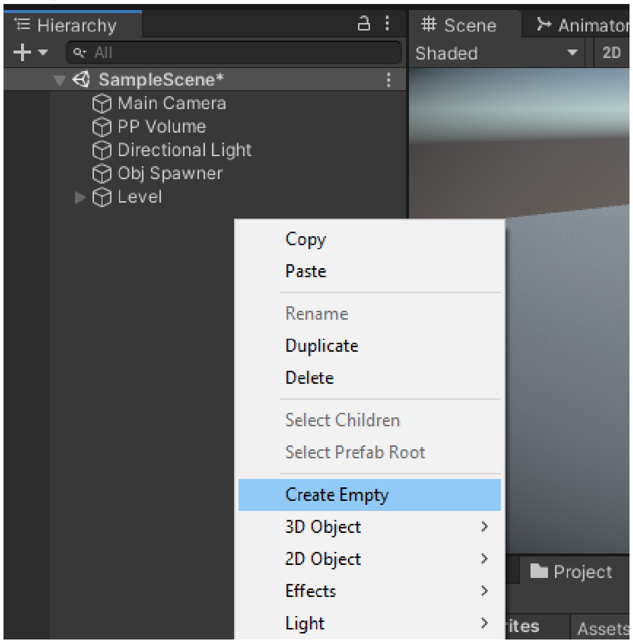
ADVANCED OBJECT POOLING

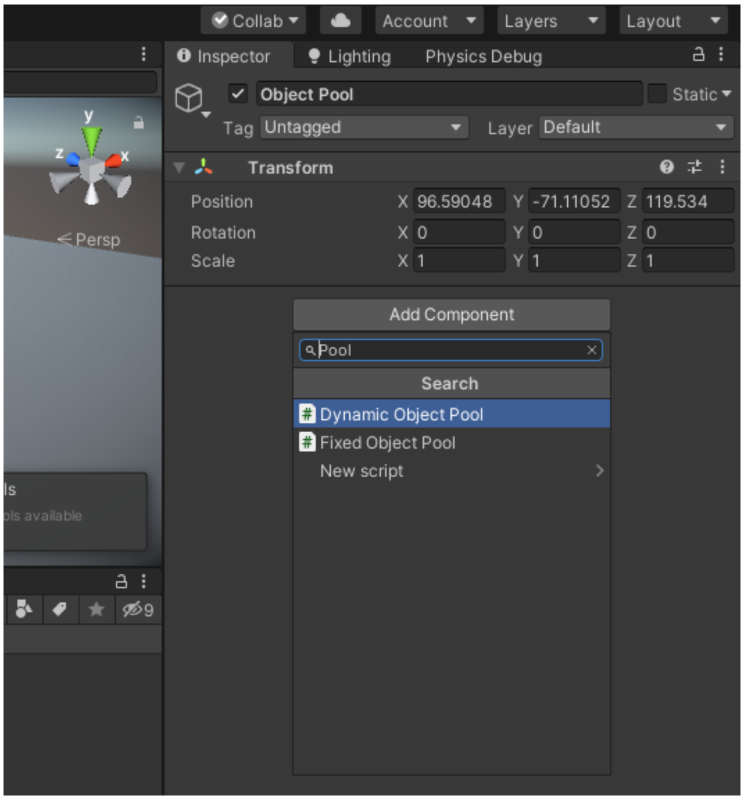
Thank you for downloading **Advanced Object Pooling**! Here you will find more information about the asset and how to set it up!

How to use

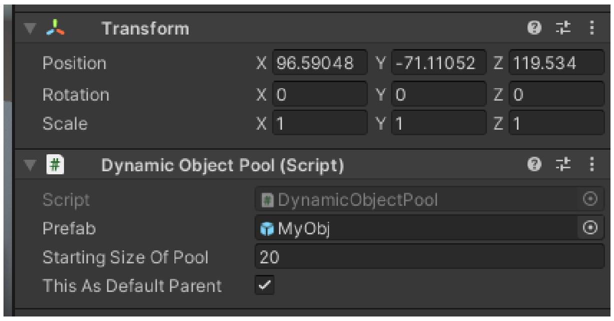
1. Create an Empty that will store all the objects in the pool



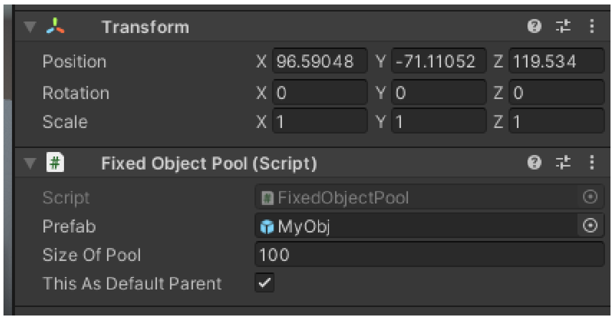
2. Add Component > **Fixed Object Pool** or **Dynamic Object Pool**
- **Fixed Object Pool** – creates an object pool of a fixed size
- **Dynamic Object Pool** – creates an object pool with flexible size



3. Place the prefab you want to use in this pool, set a size and choose if you want all objects to be automatically parented to the Pool Game Object.

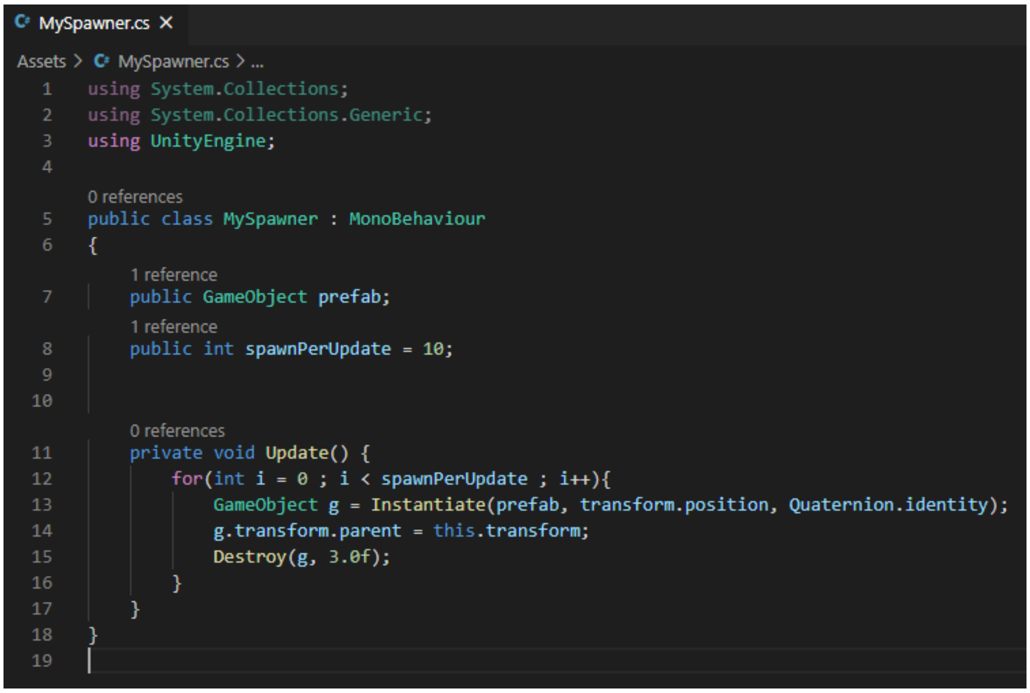


Transform
Position X 96.59048 Y -71.11052 Z 119.534
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Dynamic Object Pool (Script)
Script DynamicObjectPool
Prefab MyObj
Starting Size Of Pool 20
This As Default Parent ✓



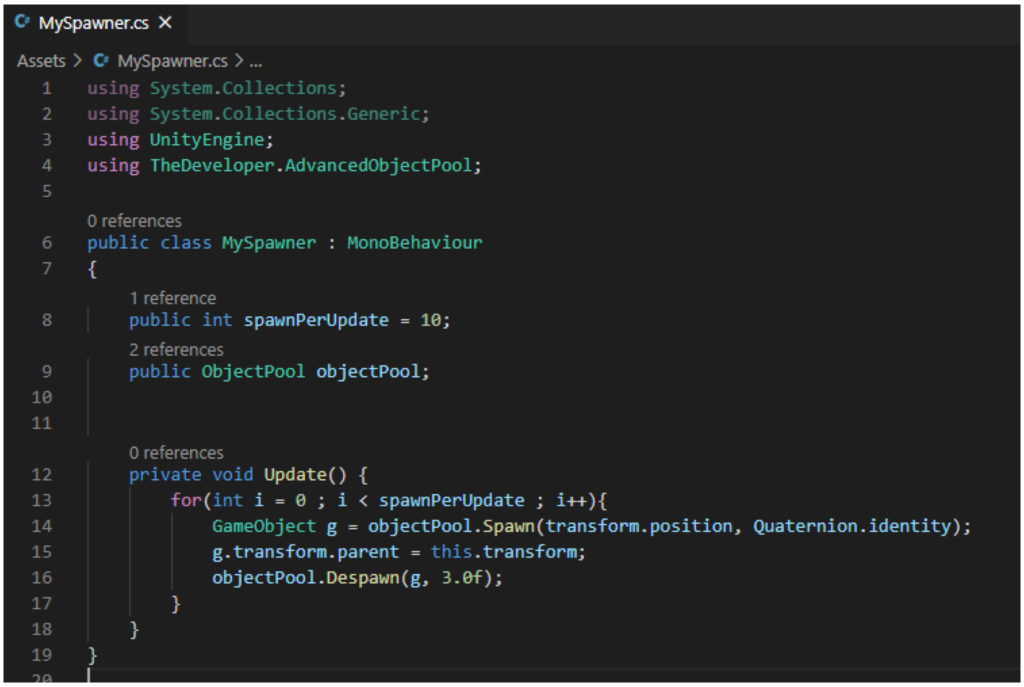
Transform
Position X 96.59048 Y -71.11052 Z 119.534
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Fixed Object Pool (Script)
Script FixedObjectPool
Prefab MyObj
Size Of Pool 100
This As Default Parent ✓

4. Now you have to make your code:



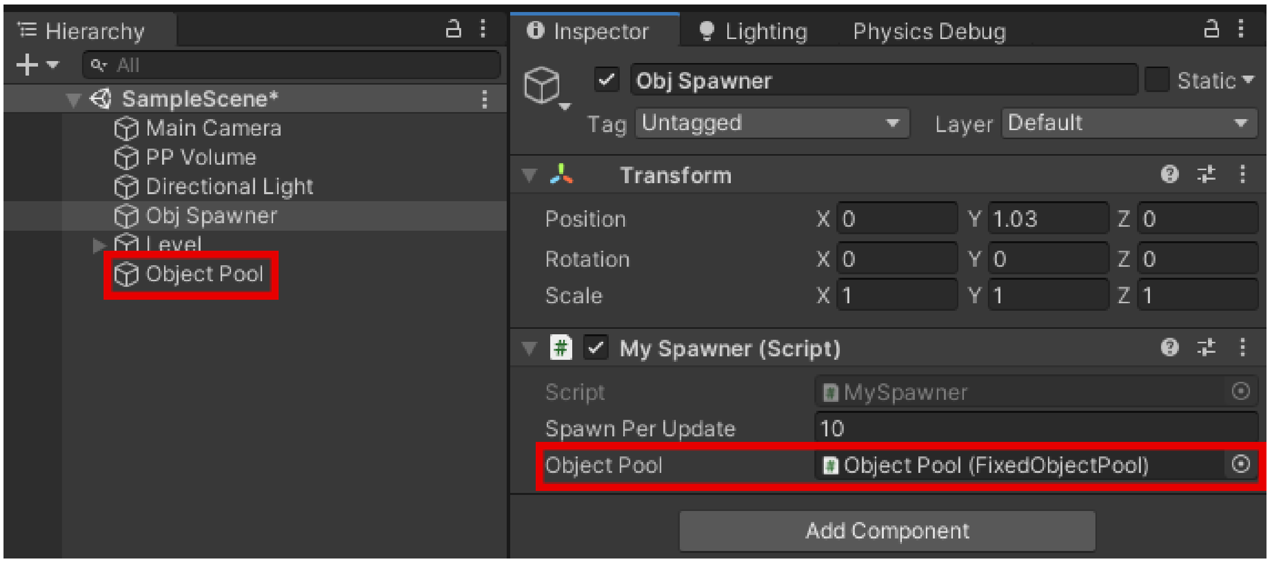
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

0 references
public class MySpawner : MonoBehaviour
{
    1 reference
    public GameObject prefab;
    1 reference
    public int spawnPerUpdate = 10;


    0 references
    private void Update() {
        for(int i = 0 ; i < spawnPerUpdate ; i++){
            GameObject g = Instantiate(prefab, transform.position, Quaternion.identity);
            g.transform.parent = this.transform;
            Destroy(g, 3.0f);
        }
    }
}
```

use the Object Pooling:



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TheDeveloper.AdvancedObjectPool;

0 references
public class MySpawner : MonoBehaviour
{
    1 reference
    public int spawnPerUpdate = 10;
    2 references
    public ObjectPool objectPool;


    0 references
    private void Update() {
        for(int i = 0 ; i < spawnPerUpdate ; i++){
            GameObject g = objectPool.Spawn(transform.position, Quaternion.identity);
            g.transform.parent = this.transform;
            objectPool.Despawn(g, 3.0f);
        }
    }
}
```

5. Final Step! Assign the field Object Pool.



Obj Spawner

Tag: Untagged Layer: Default

Transform	X	Y	Z
Position	0	1.03	0
Rotation	0	0	0
Scale	1	1	1

My Spawner (Script)

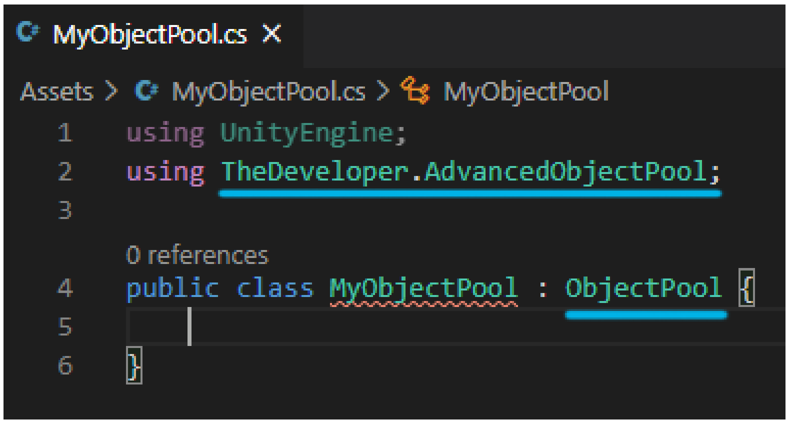
Script: MySpawner

Spawn Per Update: 10

Object Pool: Object Pool (FixedObjectPool)

Creating your own Object Pool

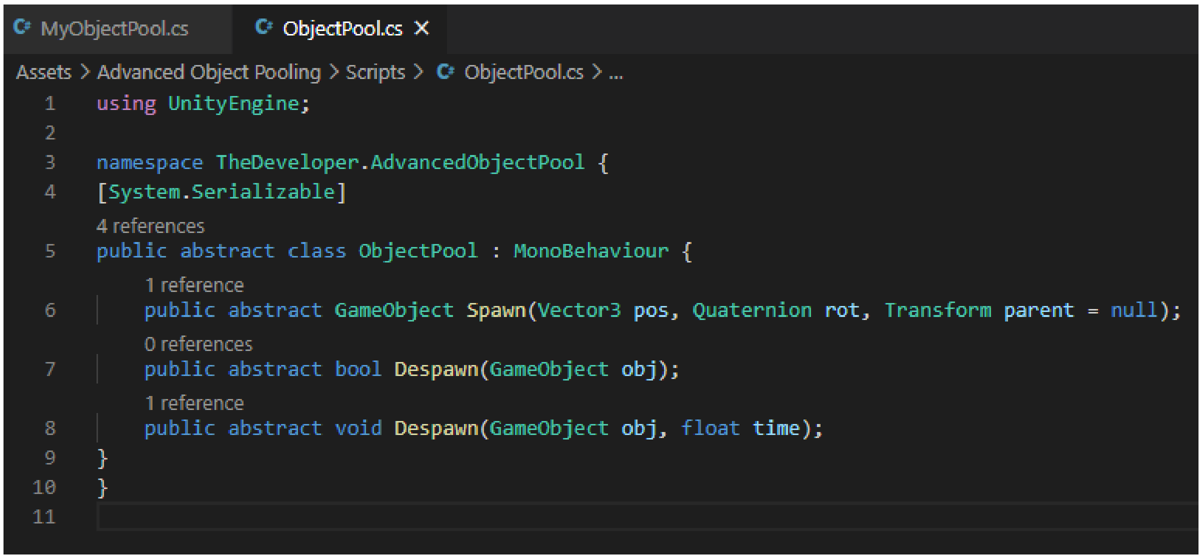
1. Create a class that inherits from the abstract class ObjectPool.



```
using UnityEngine;
using TheDeveloper.AdvancedObjectPool;

public class MyObjectPool : ObjectPool {

}
```

As we can see it gives an error. This error is caused because our class still doesn’t have any implementations of the methods in ObjectPool.

2. Let’s see what methods ObjectPool contains



```csharp
using UnityEngine;

namespace TheDeveloper.AdvancedObjectPool {
[System.Serializable]
public abstract class ObjectPool : MonoBehaviour {
    public abstract GameObject Spawn(Vector3 pos, Quaternion rot, Transform parent = null);
    public abstract bool Despawn(GameObject obj);
    public abstract void Despawn(GameObject obj, float time);
}
}
```

As we see Object Pool inherits from MonoBehaviour and contains 3 abstract methods:

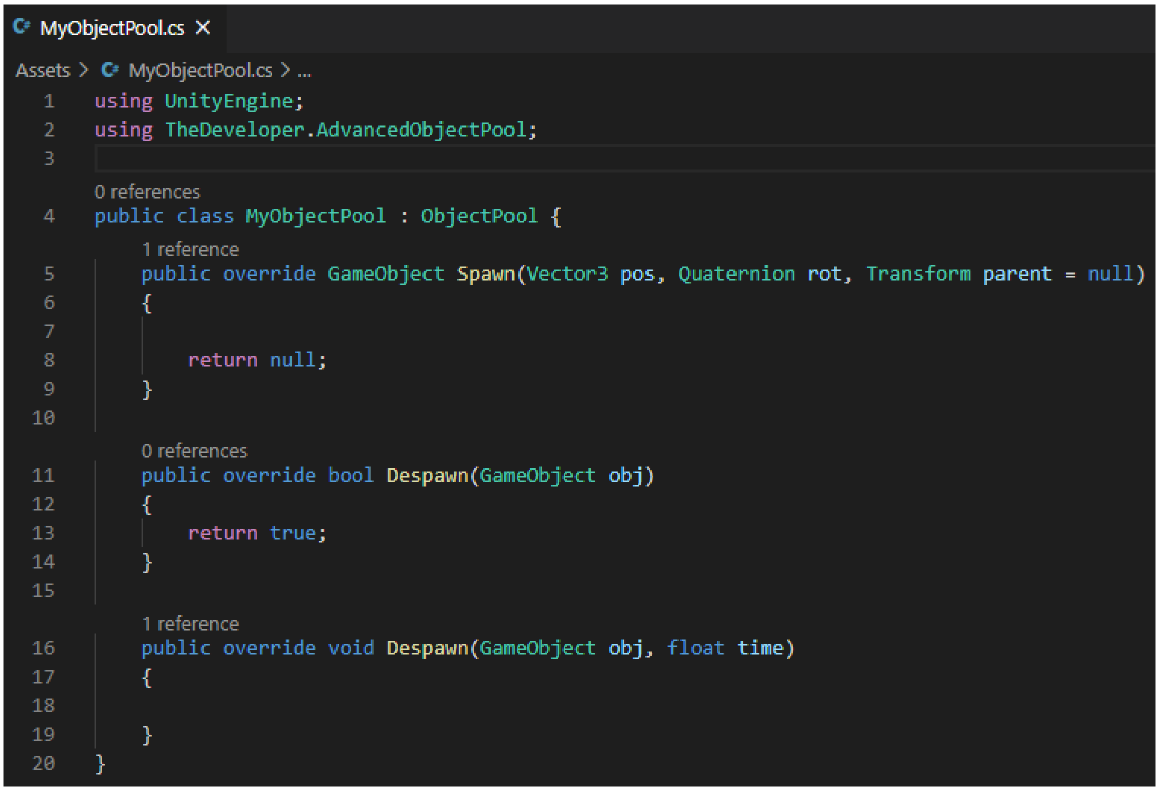
- Spawn(pos, rot, parent) – Spawns an object at a specific world position and rotation.

You can also give a parent to the object you spawn.

- Despawn(obj) – Simple despawns an object.

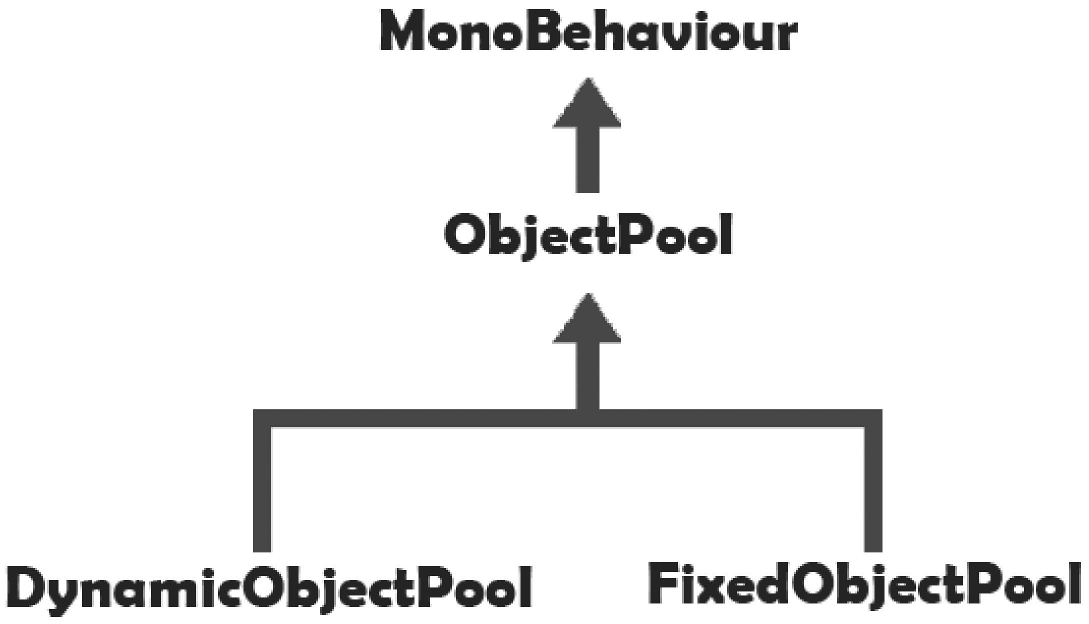
- Despawn(obj, time) – Simple despawns an object after given time.

3. Now it’s up to you to fill these methods ;)



```csharp
using UnityEngine;
using TheDeveloper.AdvancedObjectPool;

public class MyObjectPool : ObjectPool {
    public override GameObject Spawn(Vector3 pos, Quaternion rot, Transform parent = null)
    {

        return null;
    }

    public override bool Despawn(GameObject obj)
    {
        return true;
    }

    public override void Despawn(GameObject obj, float time)
    {

    }
}
```

Inheritance Tree



Contacts

Email – support@thedevelopers.tech

WEBSITE

YOUTUBE

FACEBOOK

INSTAGRAM

TWITTER

LINKEDIN