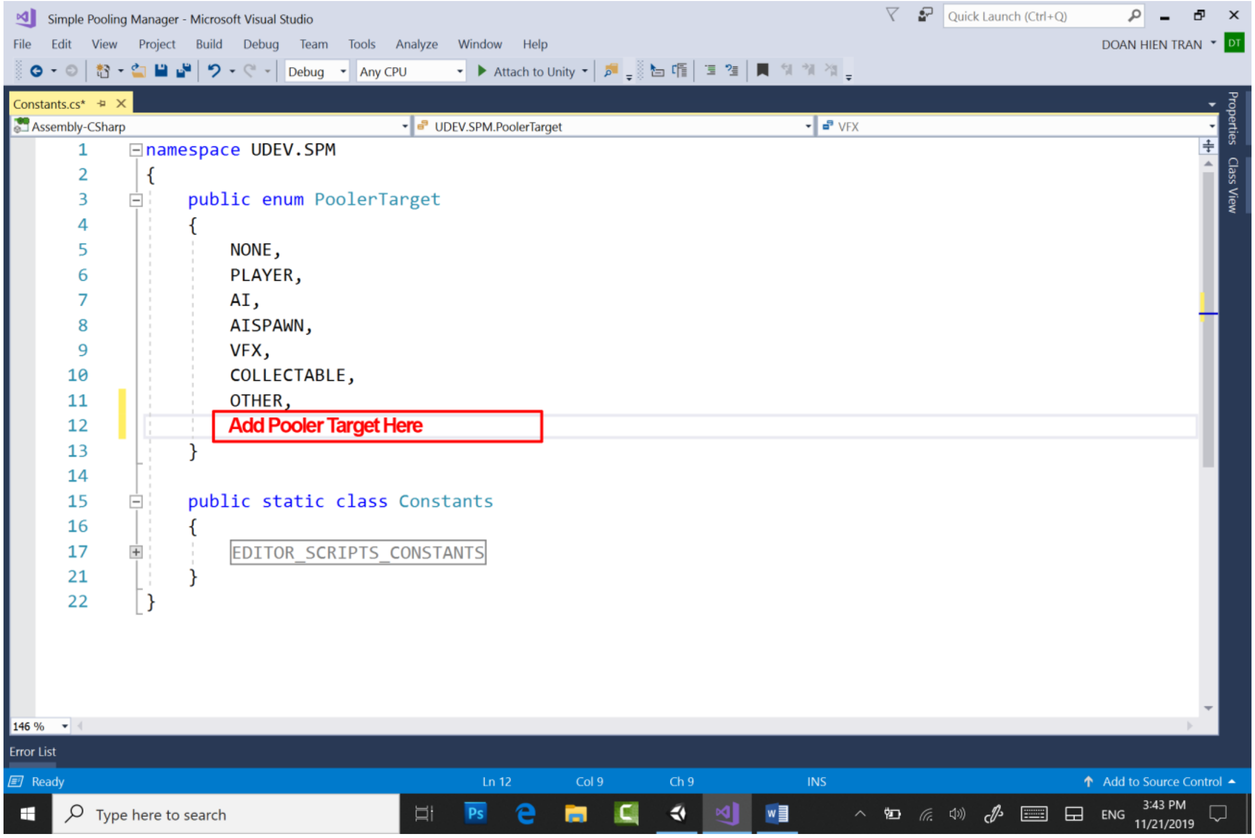
Simple Pooling Manager (SPM)

I – Config

Open ***"Constants.cs"*** you will see:



```
namespace UDEV.SPM
{
    public enum PoolerTarget
    {
        NONE,
        PLAYER,
        AI,
        AISPAWN,
        VFX,
        COLLECTABLE,
        OTHER,
        Add Pooler Target Here
    }

    public static class Constants
    {
        EDITOR_SCRIPTS_CONSTANTS
    }
}
```

You can modify pooler target in here.

Warning: *You can remove or add pooler target to any place in enum class if you don't have any pooler yet.If you already have poolers you must add new pooler target to bottom of enum class and can rename any pooler but don't remove it.*

If you already have poolers and you add, remove pooler target at new place not bottom place.Please update pooler target in all of your pooler files.

You also can modify save data path of SPM in ***"Constants.cs"*** file.

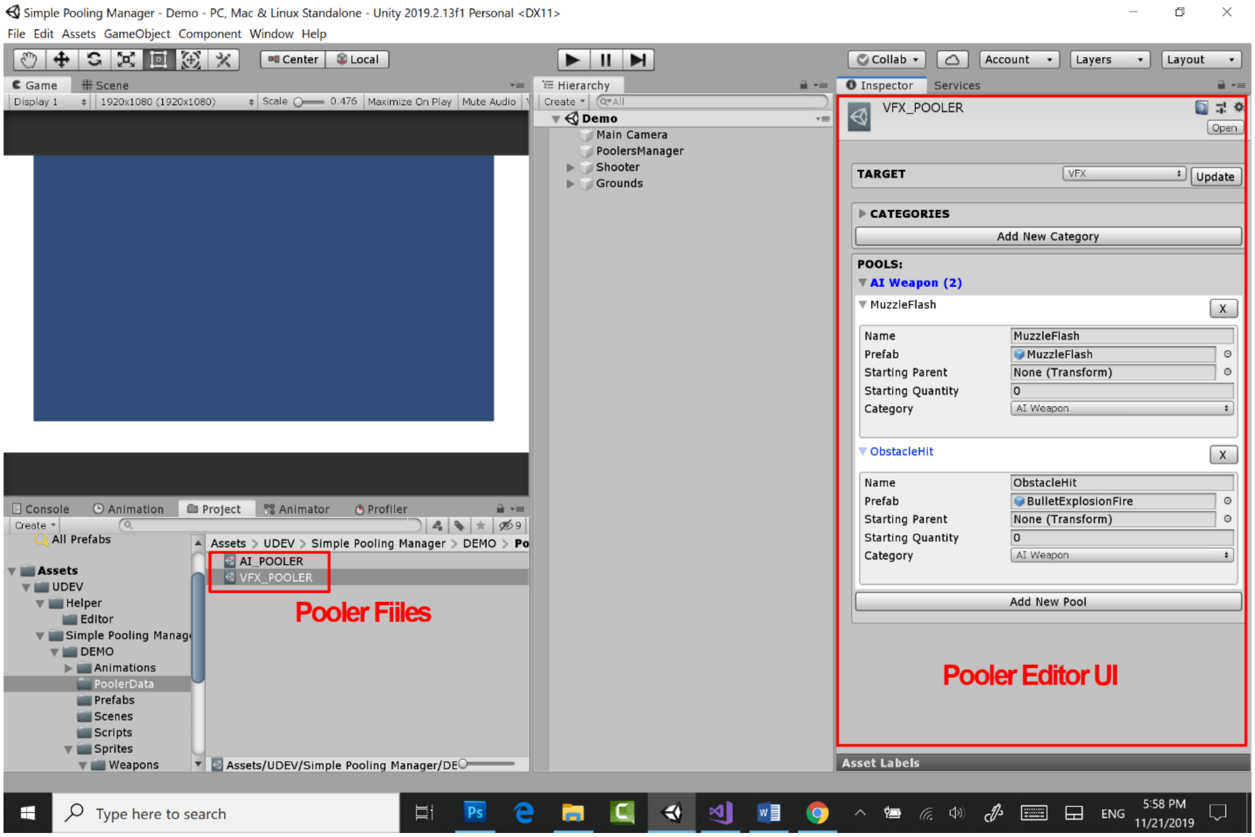
II – Create Pooler

Have two way to create a pooler:

1) Select **Assets/Create/UDEV/SPM/Create Pooler**

2) Right click your mouse and select **Create/UDEV/SPM/Create Pooler**

After a pooler created you will see:


Simple Pooling Manager - Demo - PC, Mac & Linux Standalone - Unity 2019.2.13f1 Personal <DX11>

Pooler Fiiles

Pooler Editor UI

VFX_POOLER

TARGET: VFX

CATEGORIES

POOLS:

AI Weapon (2)

MuzzleFlash

Name	MuzzleFlash
Prefab	MuzzleFlash
Starting Parent	None (Transform)
Starting Quantity	0
Category	AI Weapon

ObstacleHit

Name	ObstacleHit
Prefab	BulletExplosionFire
Starting Parent	None (Transform)
Starting Quantity	0
Category	AI Weapon

Assets > UDEV > Simple Pooling Manager > DEMO > Po

AI_POOLER

VFX_POOLER

Assets/UDEV/Simple Pooling Manager/DE

Follow editor of a pooler you can add your own prefabs you need to pool very easy.

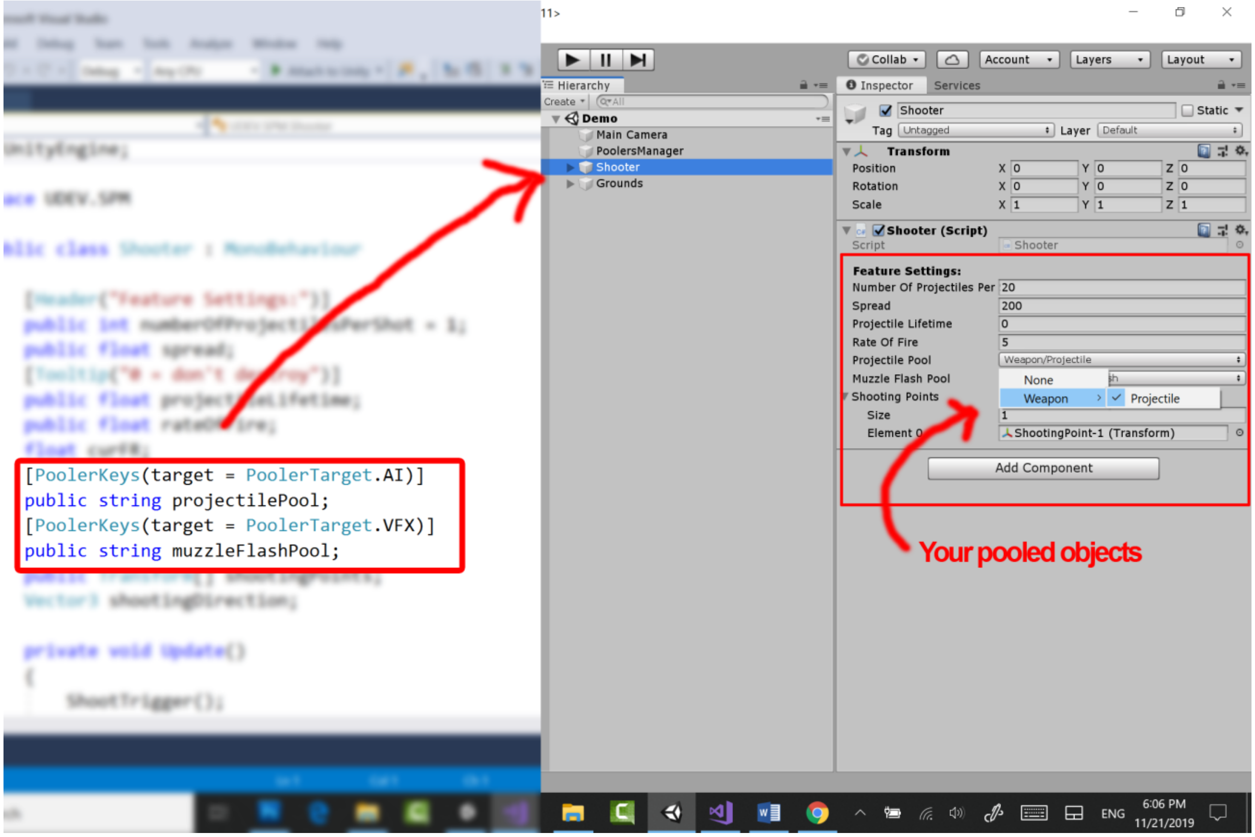
Warning: *If you switch target of a pooler file with other pooler file or move pooler file to other folder you must push update button on Pooler Editor to update SPM data.*

* *You can use same target for multi pooler file.*

III - Reference to pooled object in script

***** Make sure SPM namespace imported to your script everytime you work with SPM (add this line using UDEV.SPM to on top your script) *****

To reference to pooled object in script you can create a public string variable and add custom attribute above it.Let's see picture below:



```
using UnityEngine;

namespace UDEV.SPM

public class Shooter : MonoBehaviour

    [Header("Feature Settings:")]
    public int numberOfProjectilesPerShot = 1;
    public float spread;
    [Tooltip("0 = don't destroy")]
    public float projectileLifetime;
    public float rateOfFire;
    float curFR;
    [PoolerKeys(target = PoolerTarget.AI)]
    public string projectilePool;
    [PoolerKeys(target = PoolerTarget.VFX)]
    public string muzzleFlashPool;
    public Transform[] shootingPoints;
    Vector3 shootingDirection;

    private void Update()
    {
        ShootTrigger();
```

Shooter (Script)

Feature Settings:
Number Of Projectiles Per: 20
Spread: 200
Projectile Lifetime: 0
Rate Of Fire: 5
Projectile Pool: Weapon/Projectile
Muzzle Flash Pool
Shooting Points
Size: 1
Element 0: ShootingPoint-1 (Transform)

Remember: Use attribute [PoolerKeys(target = "target of your pooler")] above public string variable to get list of your pooled objects.

IV – Spawn Pooled Object

1 – Create empty game object in scene.

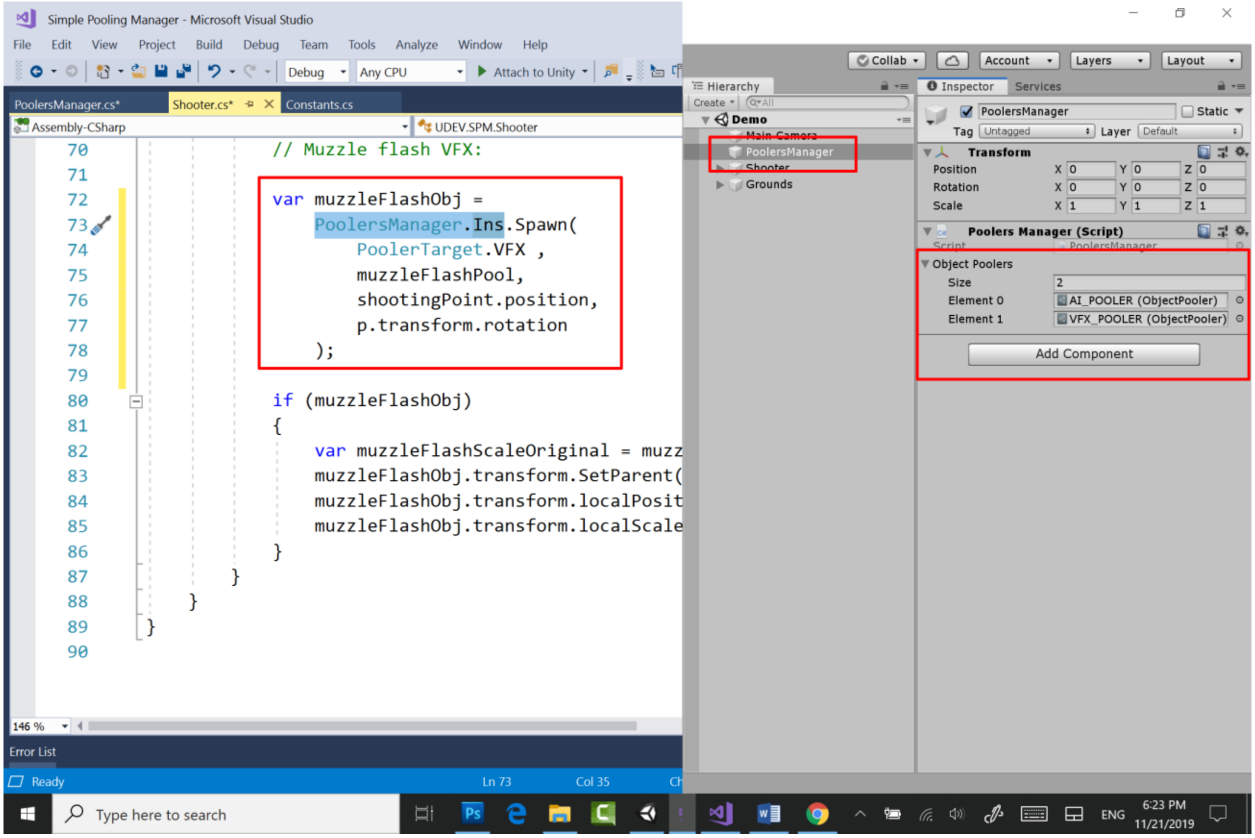
2 – Add ***"PoolersManager.cs"*** script to it.

3 – Add all your pooler object files to ***"Object Poolers"*** Array.

4 – In your script let add a line to spawn your pooled object: "PoolersManager.Ins.Spawn(pooler_target ,poolId, position, rotation, parentTransform)".

It return a gameobject.

Let's see picture below:



```
            // Muzzle flash VFX:

            var muzzleFlashObj =
                PoolersManager.Ins.Spawn(
                    PoolerTarget.VFX ,
                    muzzleFlashPool,
                    shootingPoint.position,
                    p.transform.rotation
                );

            if (muzzleFlashObj)
            {
                var muzzleFlashScaleOriginal = muzz
                muzzleFlashObj.transform.SetParent(
                muzzleFlashObj.transform.localPosit
                muzzleFlashObj.transform.localScale
            }
        }
    }
}
```

PoolersManager ☑ □ Static
Tag: Untagged Layer: Default

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

Poolers Manager (Script)
Script: PoolersManager
Object Poolers
Size: 2
Element 0: AI_POOLER (ObjectPooler)
Element 1: VFX_POOLER (ObjectPooler)

To unspawn a pooled object so simply.Let's use built in Unity's method ***"gameobject.SetActive(false)"*** on pooled object spawned.

If you want to unspawn pooled object after a few time you can see how to do that in **"AutoDestroy script"** section:

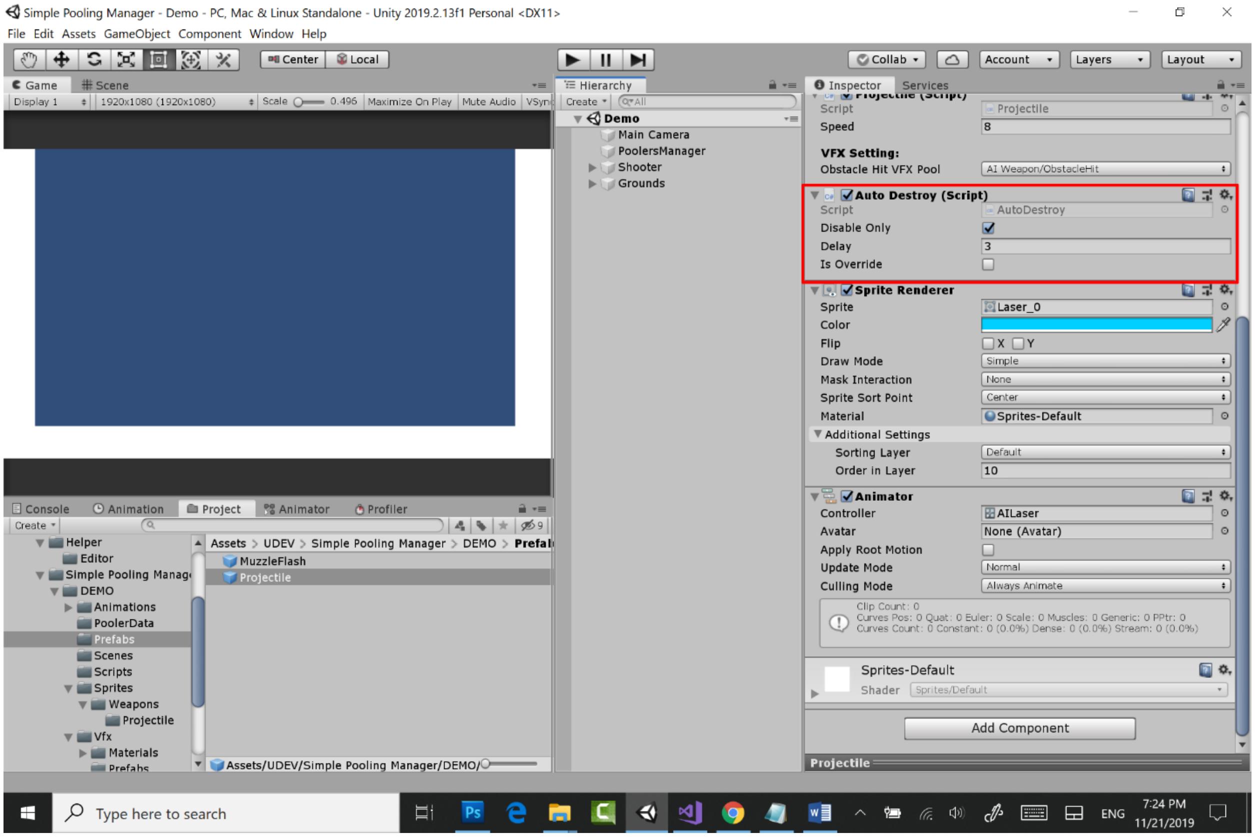
To clear pooled objects of a pooler let add a line:

"PoolersManager.Ins. Clear(pooler_target)".

To clear pooled objects of all pooler let add a line:

"PoolersManager.Ins. ClearAll()".

V – AutoDestroy script



Projectile (Script)
Script: Projectile
Speed: 8

VFX Setting:
Obstacle Hit VFX Pool: AI Weapon/ObstacleHit

☑ **Auto Destroy (Script)**
Script: AutoDestroy
Disable Only: ☑
Delay: 3
Is Override: □

☑ **Sprite Renderer**
Sprite: Laser_0
Color: 🟦
Flip: □ X □ Y
Draw Mode: Simple
Mask Interaction: None
Sprite Sort Point: Center
Material: Sprites-Default
Additional Settings
Sorting Layer: Default
Order in Layer: 10

☑ **Animator**
Controller: AILaser
Avatar: None (Avatar)
Apply Root Motion: □
Update Mode: Normal
Culling Mode: Always Animate

```
Clip Count: 0
Curves Pos: 0 Quat: 0 Euler: 0 Scale: 0 Muscles: 0 Generic: 0 PPtr: 0
Curves Count: 0 Constant: 0 (0.0%) Dense: 0 (0.0%) Stream: 0 (0.0%)
```

Sprites-Default
Shader: Sprites/Default

- **"Disable Only" :** Check it if your object was pooled in pooler file.If you don't check it your object will be really destroy.
- **"Delay" :** Waiting time before your object will be unspawn or really destroy.
- **"Is Override" :** Check it if you don't want ***AutoDestroy.cs*** script excute automatic and you must call method ***"DestroyTrigger()"*** in your code to excute AutoDestroy feature.

You can check out DEMO folder to understand how to use SPM Tool more clearly.

That's All. Thank you for purchased our tool.

If have any question or need support about this tool feel free email to us: ***udev2020@gmail.com***