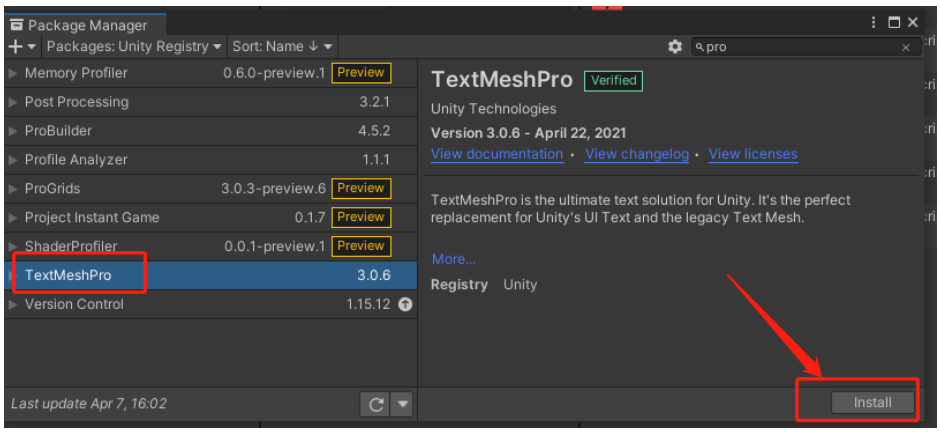
UI Framework

Catalogue

1.Quick Start

1. Install TextMeshPro

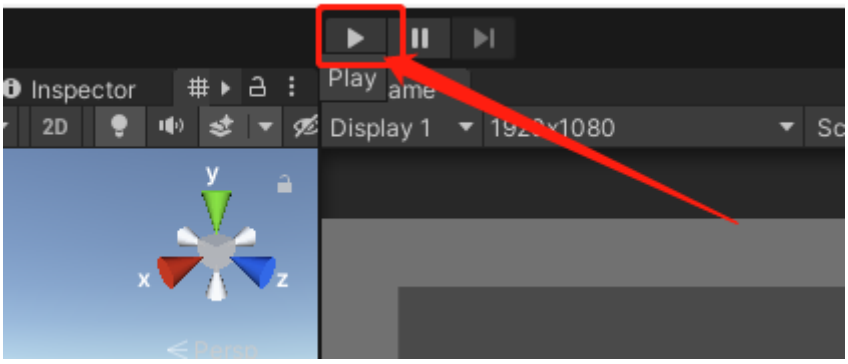
   Open “Window/Package Manager” Install **TextMeshPro**



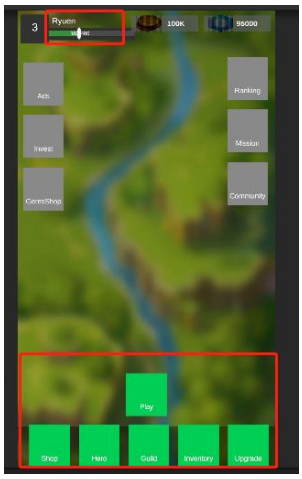
2. Open scene

   “Assets/GamesTan/SuperScrollRect/Examples/**Main.unity**”

3. Click “**Play**” button ,and you will see the demo

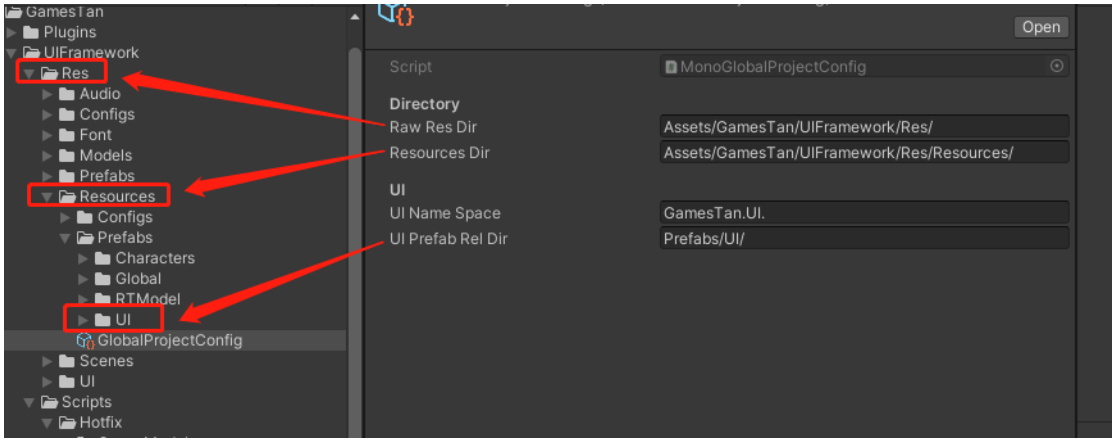


4. All Green Buttons can open some window, try to click them.


3
Ryuton
100K
99000
Acs
Invest
GemsShop
Ranking
Mission
Community
Play
Shop
Hero
Guild
Inventory
Upgrade

2. Start To Use

1. Config Project setting

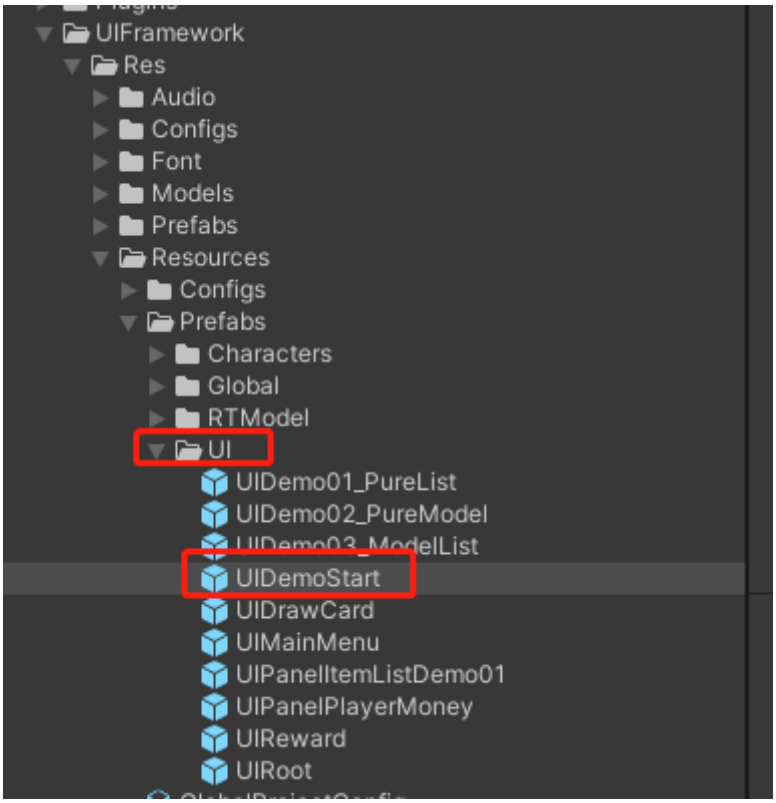


Raw Res Dir	Project global resource directory (include all resource)
Resources Dir	Resources Directory (where code can direct load) Anything that can be load from **ResourceManager** should be in this directory or it's child directory
UI Prefab Rel Dir	The Directory which UI prefab putted into UI Full path = ResourcesDir + UIPrefabRelDir In this demo : "Assets/GamesTan/UIFramework/Res/Resources/Prefabs/UI/"
UI Namespace	UI script's namespace (used to bind with prefab by framework)

2. How to add a new window

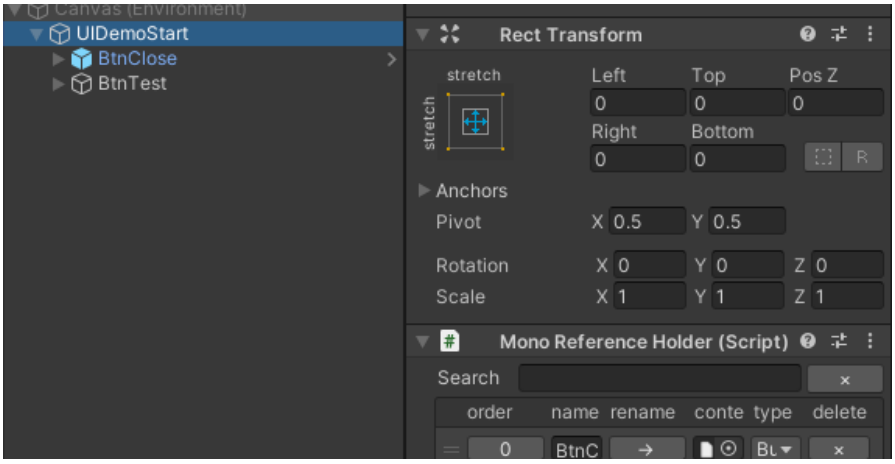
1. Create a Prefab

Create a prefab (eg: "UIDemoStart") in directory
（GlobalProjectConfig. ResourcesDir + GlobalProjectConfig.UIPrefabRelDir）
"Assets/GamesTan/UIFramework/Res/Resources/Prefabs/UI/"



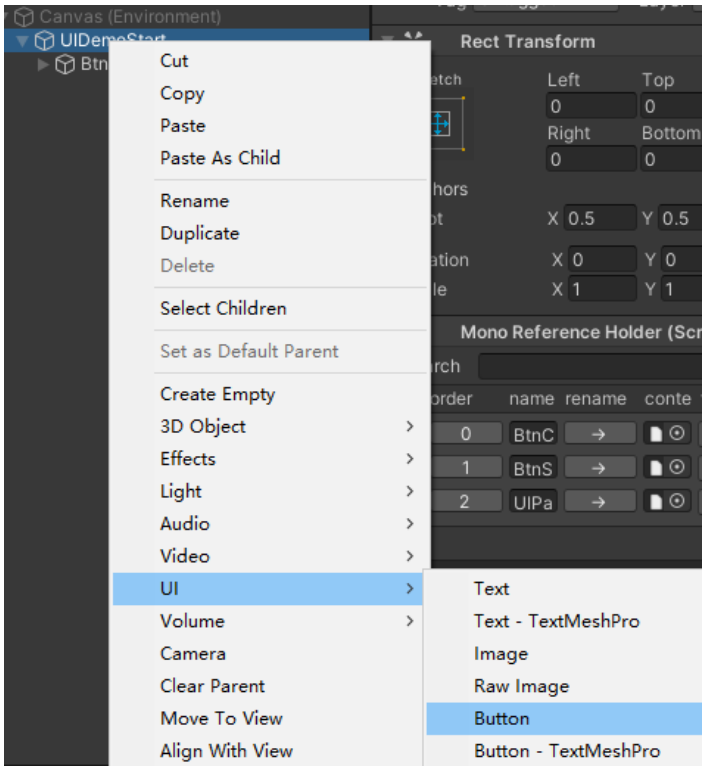
2. Attach Script

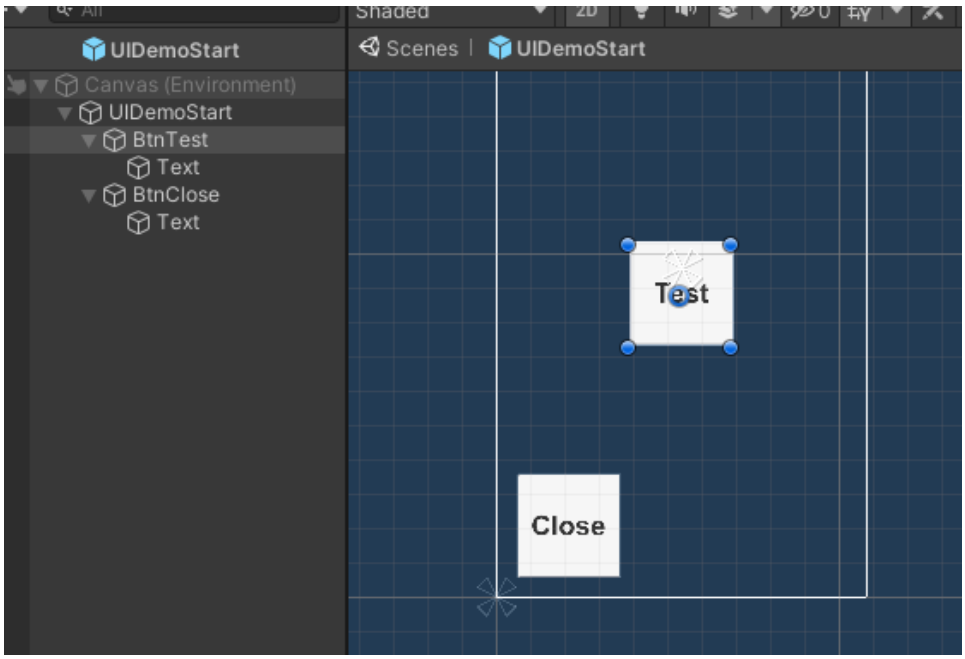
Make sure it has a ” **MonoReferenceHolder**” script



3. Create some UI elements

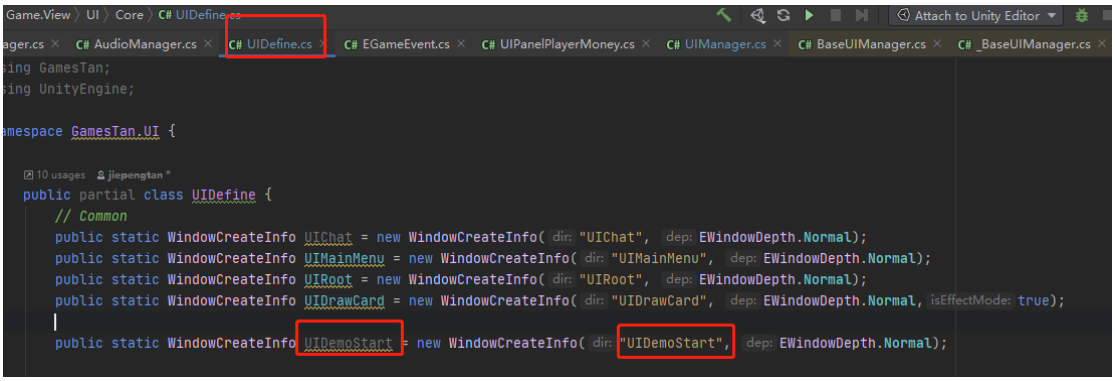
In demo : we create 2 button , “BtnClose” and “BtnTest”





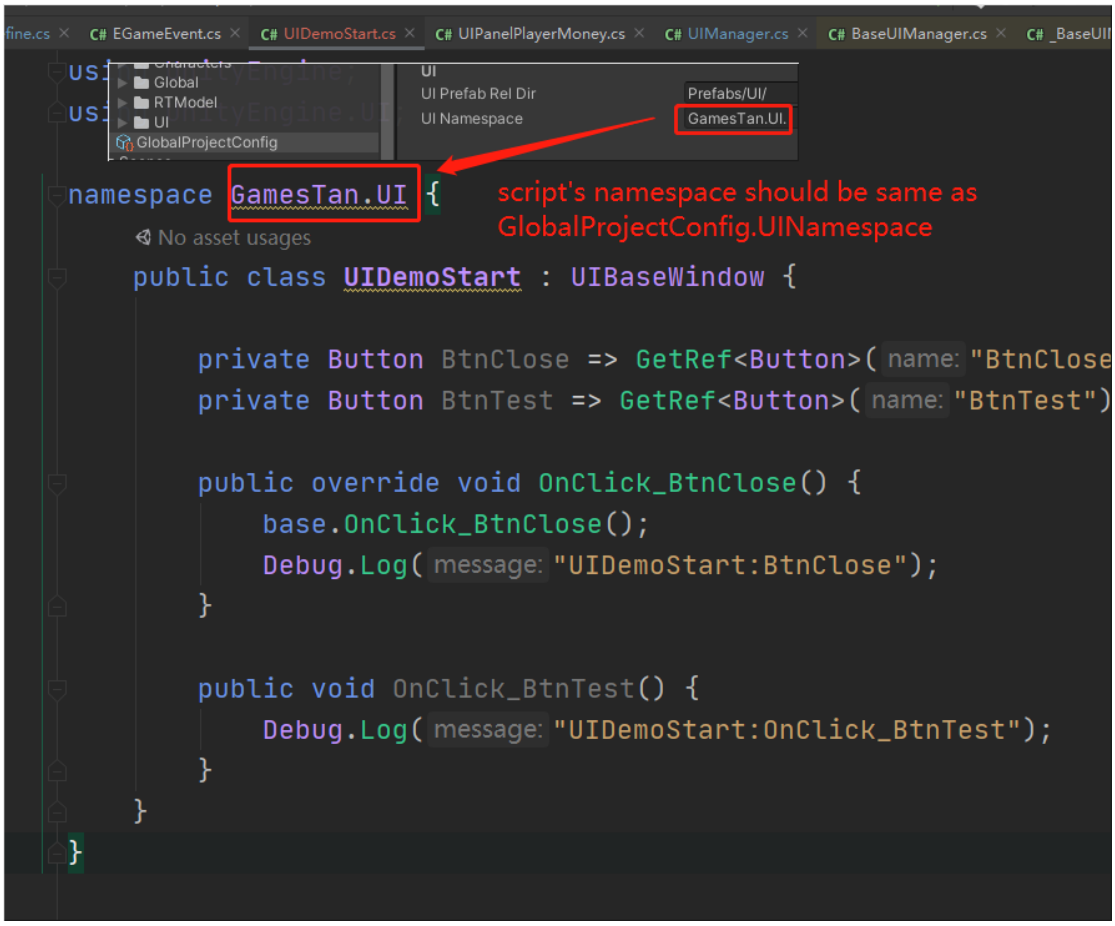
4. Define UI Instantiate info

Define a “WindowCreateInfo” object in file “UIDefine.cs”



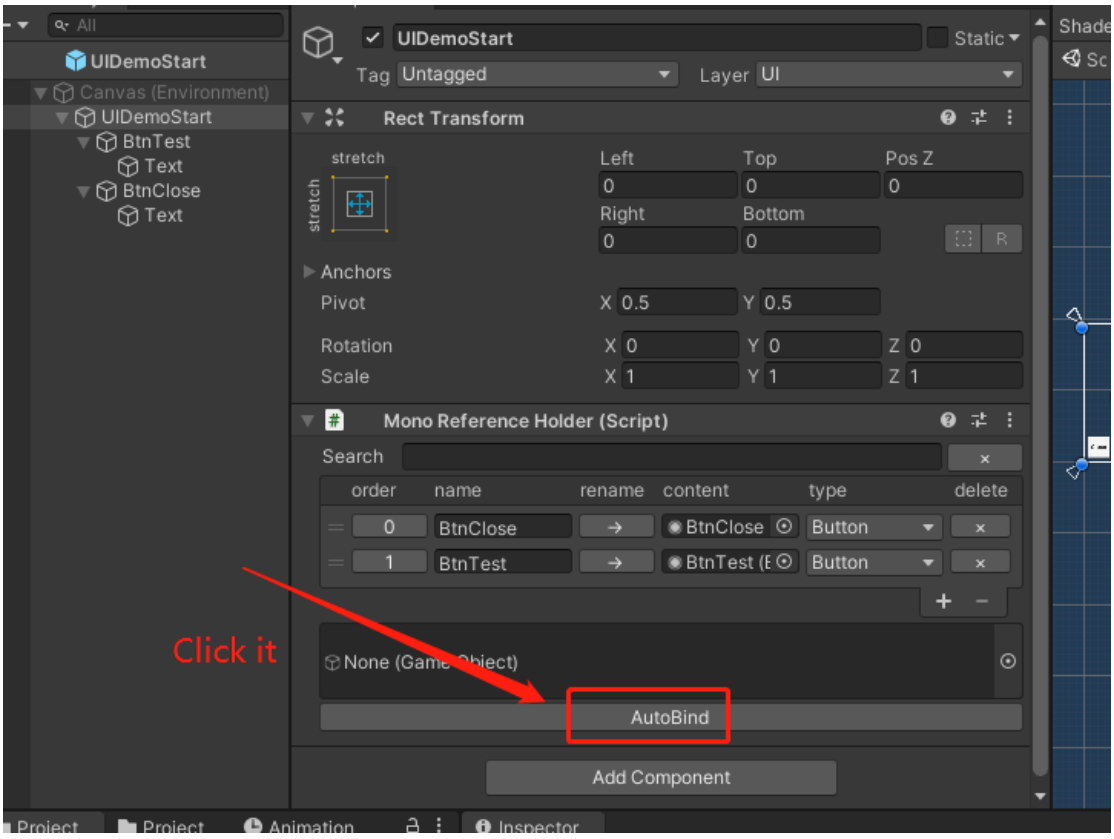
5. Create a Script (script's name has to be the same as prefab's name)

Create a script "UIDemoStart.cs"



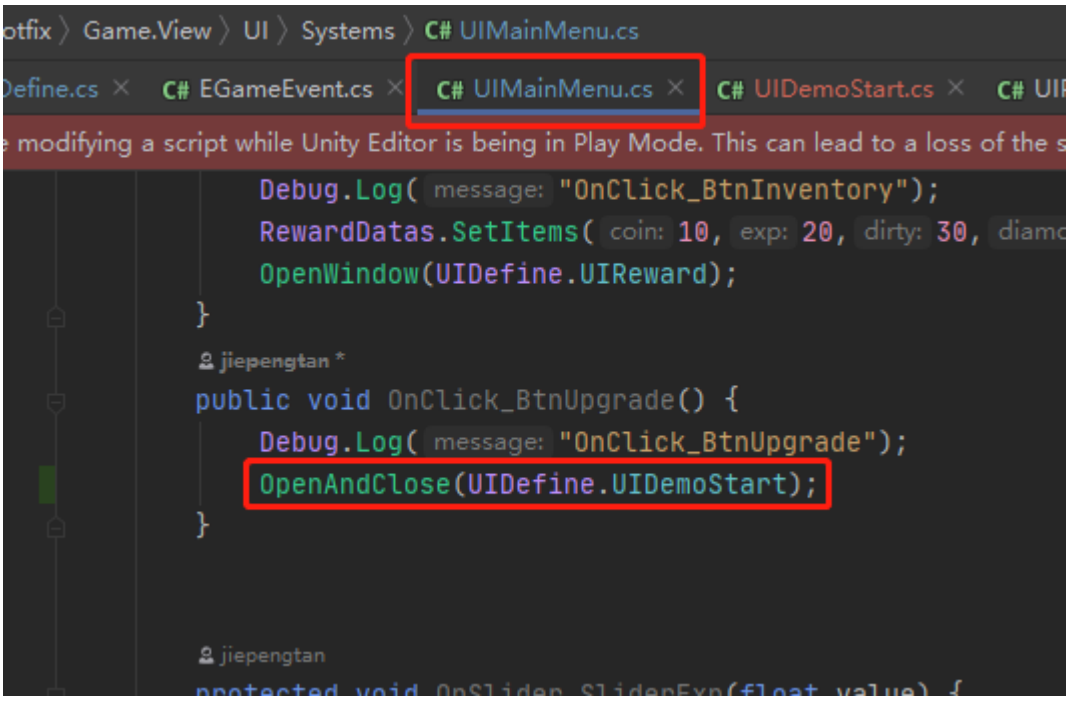
6. Auto Bind Reference

Click “**MonoReferenceHolder**” component’s ”**AutoBind**” button



7. Open UI by scirpt

Use script to open the window wherever you want



A. **Run and Click the button**

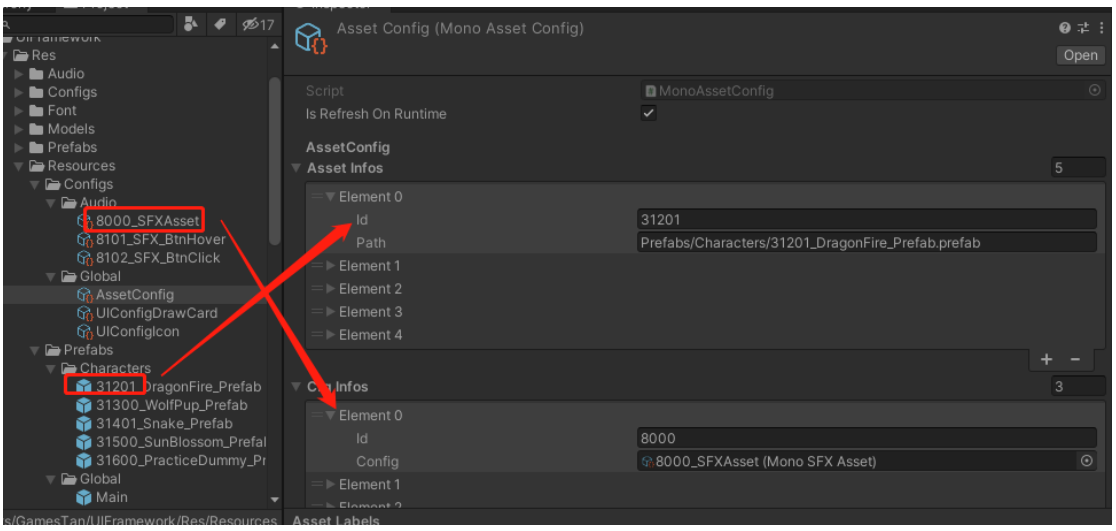
3. Script API

1. Asset Management

1. Naming rule

ConfigDir = GlobalProjectConfig .ResourcesDir + “/Configs”
PrefabDir = GlobalProjectConfig .ResourcesDir + “/Prefabs”

Any Prefab in directory **PrefabDir** or any ScriptableObject in directory **PrefabDir** whose name match the rule (AssetId + “_” + AssetName)
would be added to AssetConfig automatically, and can be loaded by **ResourceManager** with **AssetId**



2. LoadConfig

Eg: AssetId = 8000 and AssetName = TestAsset

8000_TestAsset.asset

You can load the config using the code below:

```
var config = ResourceManager.Instance.LoadConfig<ScriptableObject>( assetId: 8000);
```

3. Load Prefab & Instance Prefab

Eg: AssetId = 31201 and AssetName = DragonFire_Prefab

31201_DragonFire_Prefab.prefab

You can load or instantiate the prefab using the code below:

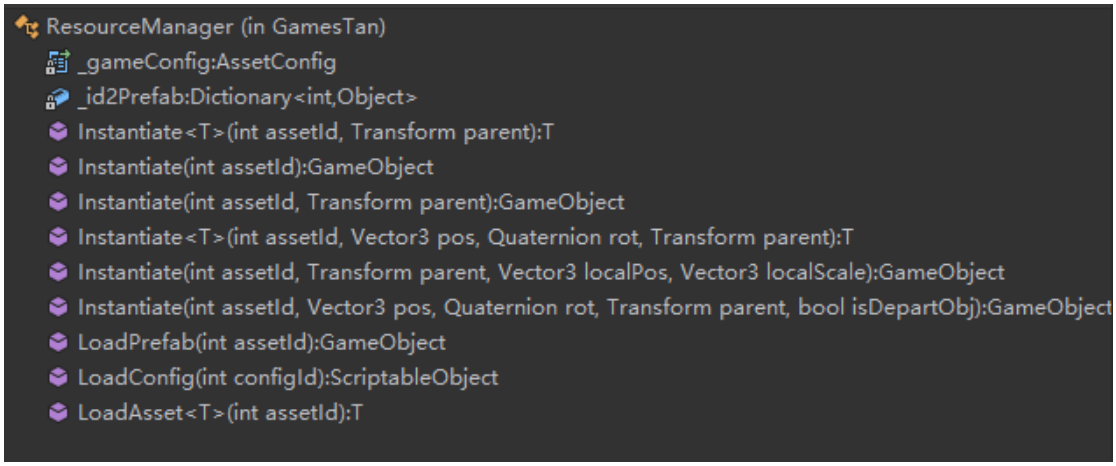
```
// Load prefab then instantiate it
var prefab :GameObject = ResourceManager.Instance.LoadPrefab( assetId: 31201);
var goFab :GameObject = GameObject.Instantiate(prefab);

// Instantiate Prefab directly
var go :GameObject = ResourceManager.Instance.Instantiate( assetId: 31201);
```

1. Naming Rule
   a) Id + "_" + AnyAssetName eg: 8001_SFXBtnHover

2. API

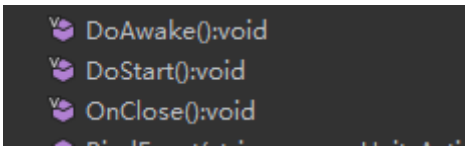
1. ResourceManager API



Instantiate	Initialize a prefab by Id
LoadPrefab	Load Prefab by Id
LoadAsset	Load Any Type Asset by Id
LoadConfig	Load ScriptableObject by Id

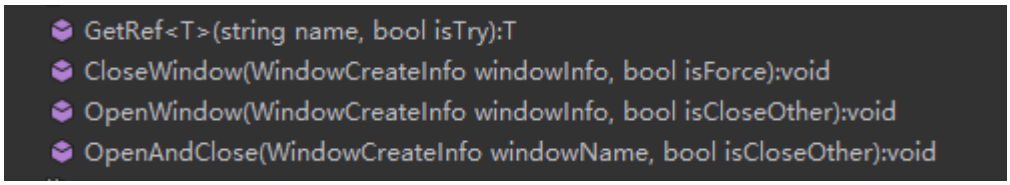
2. UIBaseWindow API

Lifecycle:

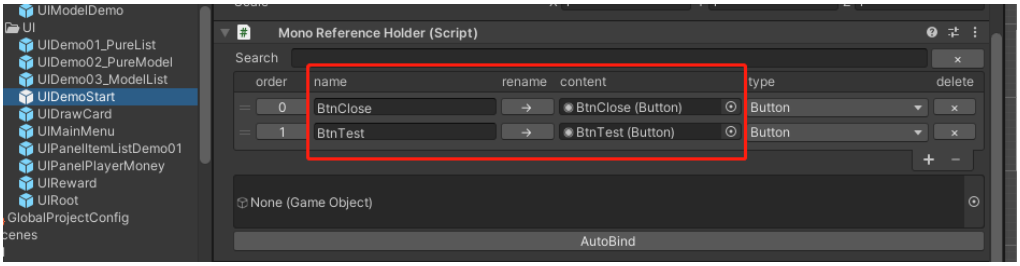


DoAwake	Called when window is opened
DoStart	Called after DoAwake when window is opened
OnClose	Called when the window was closed

API:



GetRef	Get object reference in component “MonoReferenceHold”



CloseWindow	Close the window defined in UIDefine
OpenWindow	Open the window defined in UIDefine
OpenAndClose	Open the window defined in UIDefine and close self

3. Features

1. ScrollView

See demo script “UIDemo01_PureList.cs”

2. UIModel

See demo script “UIDemo02_PureModel.cs”

3. Sub Panel

See demo script “UIDemo04_MutilSubPanel.cs”