WIDGET MANAGER

Introduction

Thank you for downloading this package stay tuned this is just beginning or you may called it proof of concept more features are in queue.

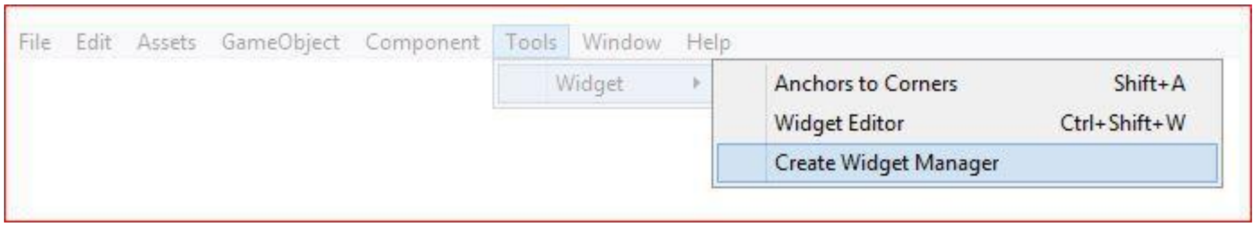
After spending one year in Gaming industry I realize managing UI's in larger projects is headache, you have to keep following things in mind while developing some UI management system for your projects.

- Must have low memory footprint (have pool system)
- Easy to Manage & Plug n Play
- Easy way of communication with UI's
- UI's Persists between scenes, don't destroy on scene change
- When you create new UI, UI Manager class remains same means no code addition required, this is useful if multiple people are working on UI's no conflicts occurs in version control.
- Keep UI's separate from Games scenes it also prevents from version control conflicts.

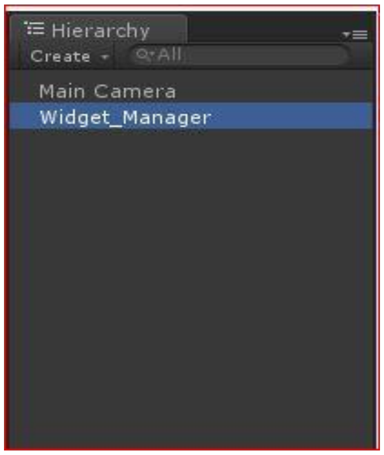
Setup

Setting Widget Manager is very simple just follow following 5 simple steps.

1. From menu Items Click Tools->Widget->Create Widget Manager.

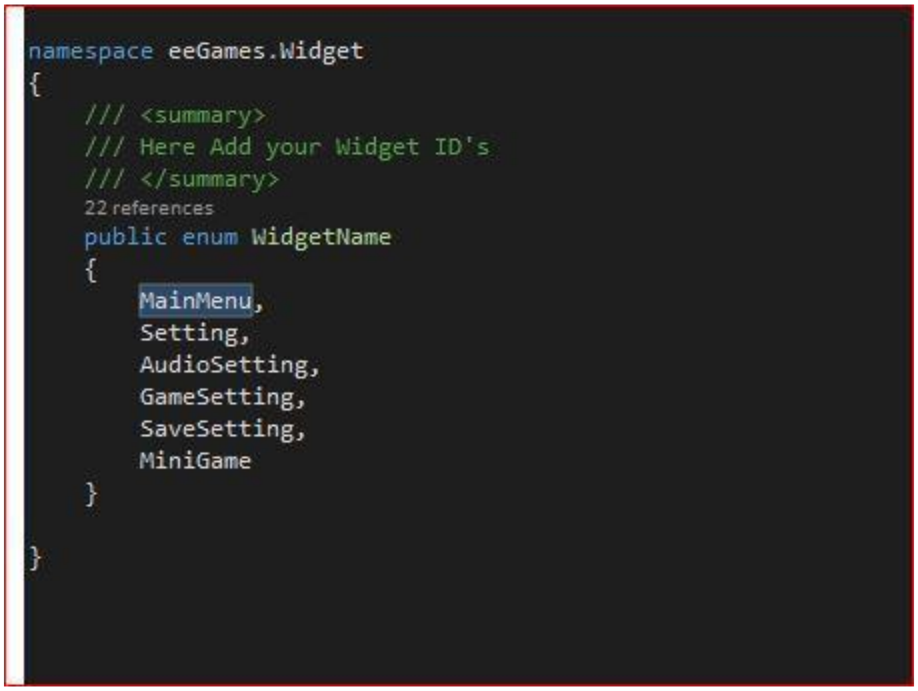


It will create Widget_Manager game object in scene like this.



2: Make new Tag "Parent" and assign it to Canvas, under which you want to Instantiate you UI's.

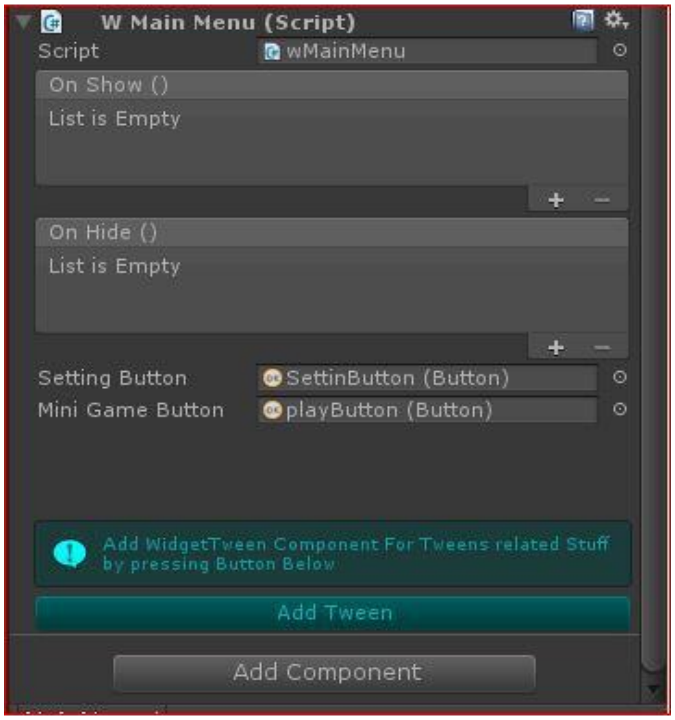
3: Open WidgetName Script (Widgets->Source Code->Common->WidgetName) and add your Widget name in it. NOTE: always add new enums at end.



```
namespace eeGames.Widget
{
    /// <summary>
    /// Here Add your Widget ID's
    /// </summary>
    22 references
    public enum WidgetName
    {
        MainMenu,
        Setting,
        AudioSetting,
        GameSetting,
        SaveSetting,
        MiniGame
    }

}
```

4: Create your UI in my case I name it MainMenu and attack a Script wManiMenu with it. All you have to do is inherit this class with Widget class. Now you inspector look like this, save prefab of this UI in Resources Folder.



W Main Menu (Script)

Script: wMainMenu

On Show ()

List is Empty

On Hide ()

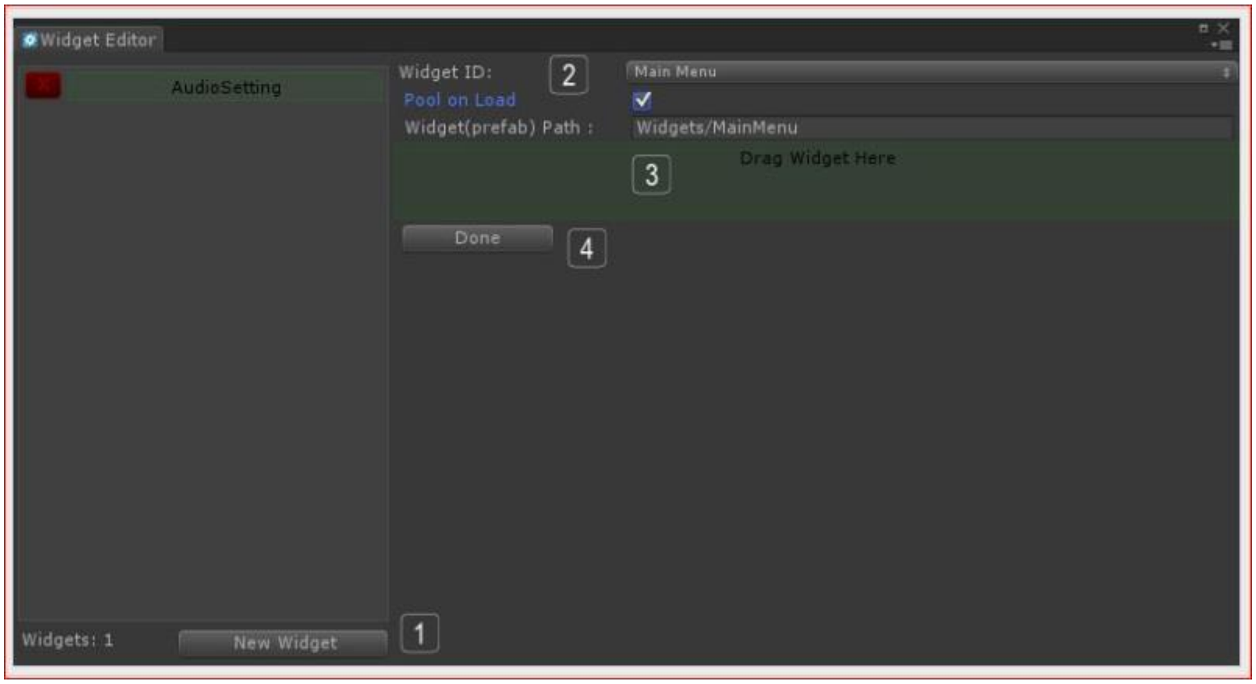
List is Empty

Setting Button: SettinButton (Button)

Mini Game Button: playButton (Button)

Add WidgetTween Component For Tweens related Stuff by pressing Button Below

5: Now you are one step away to see this Widget in action now open widget editor window (**Ctrl + Shift + w**)



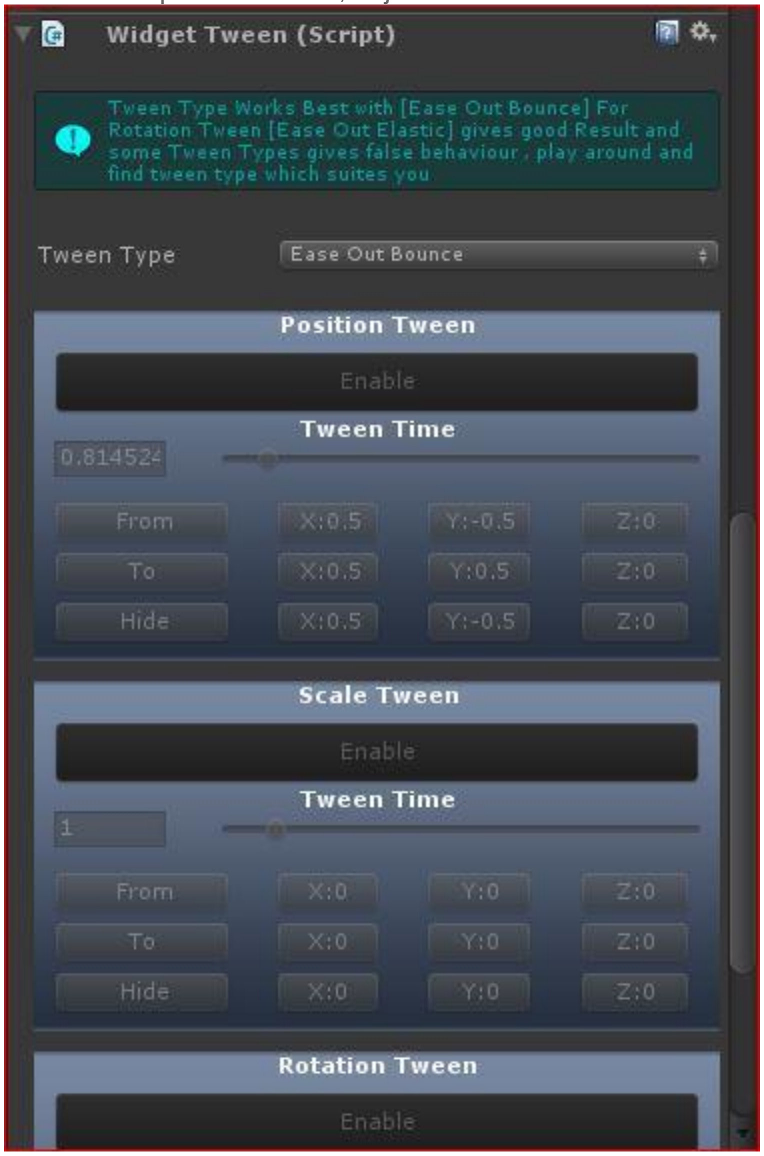
1: click New widget button
2: select widget id from drop down and check Pool on Load if you want to
3: drag you Widget prefab to green area.
4: click “Done” button
that’s all you are ready now load you widget by calling

```
    WidgetManager.Instance.Push(WidgetName.MainMenu)
```

Adding Tween

Click Add Tween button Widget Tween Component Gets added to your UI.

Now enable position tween, adjust tween time and set Tween type.



Widget Tween (Script)

Tween Type Works Best with [Ease Out Bounce] For Rotation Tween [Ease Out Elastic] gives good Result and some Tween Types gives false behaviour , play around and find tween type which suites you

Tween Type: Ease Out Bounce

Position Tween

Enable

Tween Time: 0.814524

From	X:0.5	Y:-0.5	Z:0
To	X:0.5	Y:0.5	Z:0
Hide	X:0.5	Y:-0.5	Z:0

Scale Tween

Enable

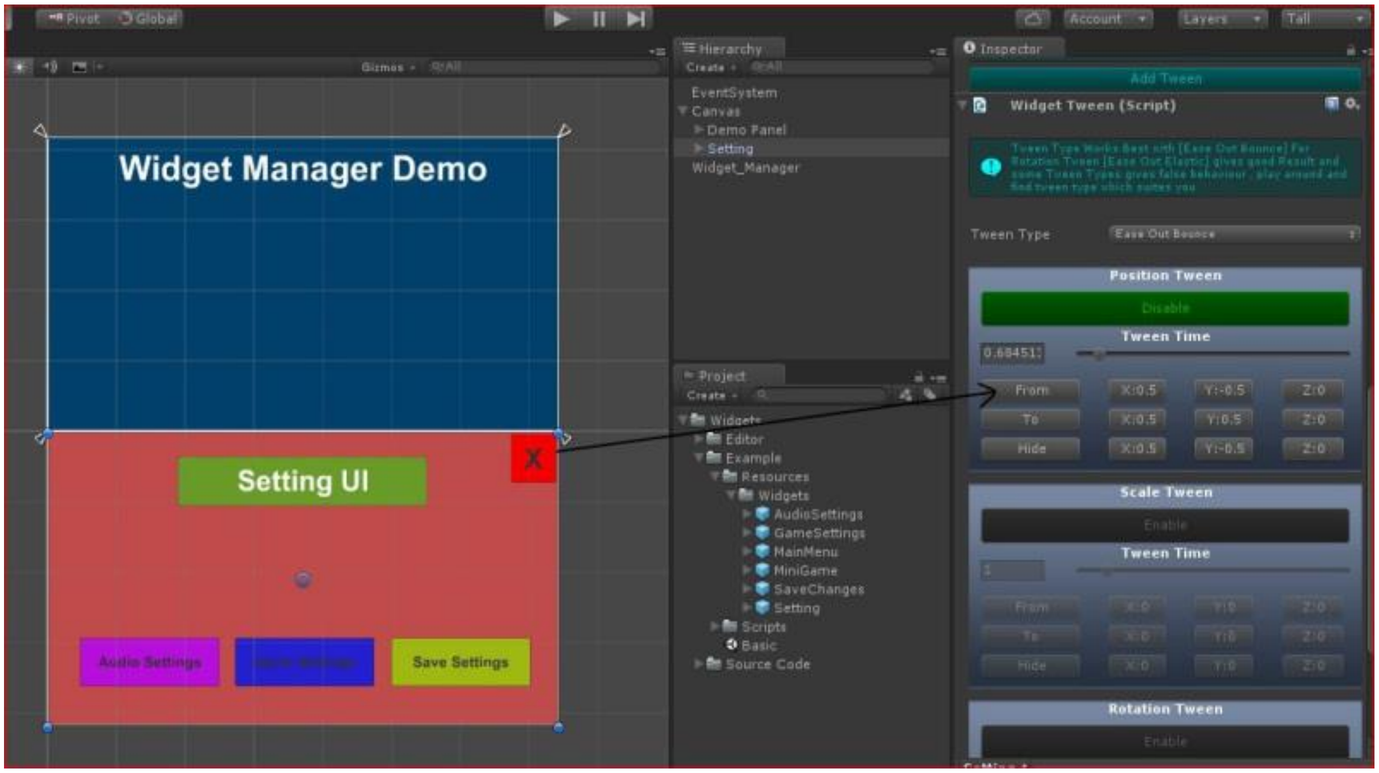
Tween Time: 1

From	X:0	Y:0	Z:0
To	X:0	Y:0	Z:0
Hide	X:0	Y:0	Z:0

Rotation Tween

Enable

Now Capture OnShow Tween positions by clicking From & To button



Tween Type Works Best with [Ease Out Bounce] For Rotation Tween [Ease Out Elastic] gives good Result and some Tween Types gives false behaviour , play around and find tween type which suites you

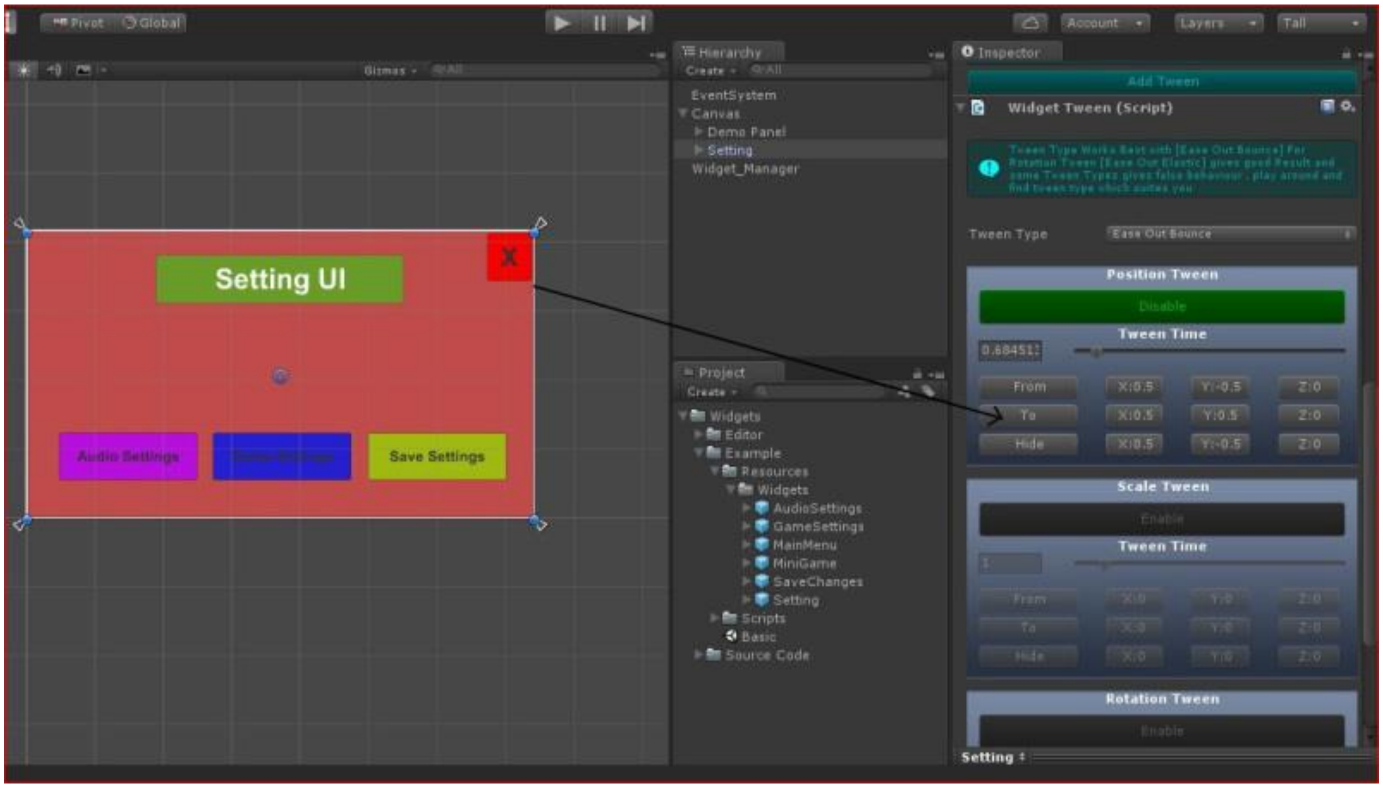
Tween Type: Ease Out Bounce

Position Tween – Tween Time: 0.684511

From	X:0.5	Y:-0.5	Z:0
To	X:0.5	Y:0.5	Z:0
Hide	X:0.5	Y:-0.5	Z:0

Scale Tween – Tween Time: 1

From	X:0	Y:0	Z:0
To	X:0	Y:0	Z:0
Hide	X:0	Y:0	Z:0



Tween Type Works Best with [Ease Out Bounce] For Rotation Tween [Ease Out Elastic] gives good Result and some Tween Types gives false behaviour , play around and find tween type which suites you

Tween Type: Ease Out Bounce

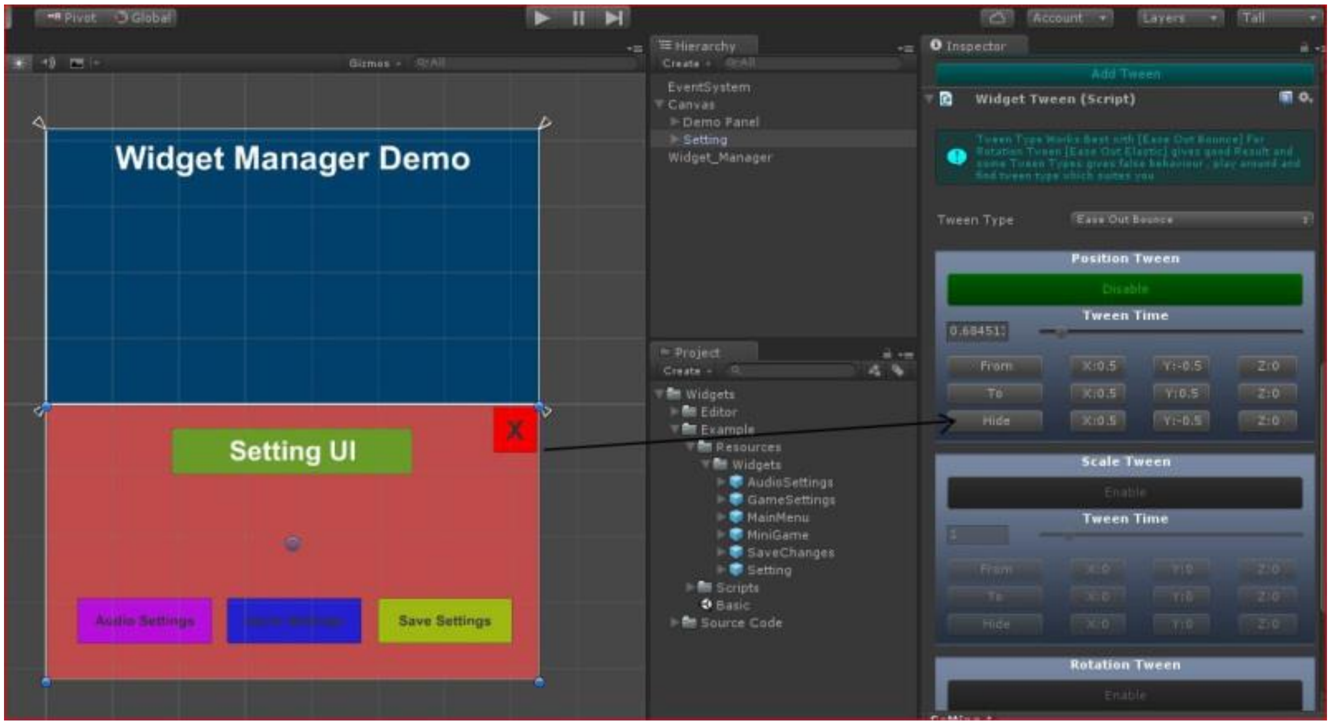
Position Tween – Tween Time: 0.684511

From	X:0.5	Y:-0.5	Z:0
To	X:0.5	Y:0.5	Z:0
Hide	X:0.5	Y:-0.5	Z:0

Scale Tween – Tween Time: 1

From	X:0	Y:0	Z:0
To	X:0	Y:0	Z:0
Hide	X:0	Y:0	Z:0

Here capture OnHide Tween Position by clicking hide button.



Widget Tween (Script)

Tween Type Works Best with [Ease Out Bounce] For Rotation Tween [Ease Out Elastic] gives good Result and some Tween Types gives false behaviour , play around and find tween type which suites you

Tween Type: Ease Out Bounce

Position Tween — Tween Time: 0.68451

From	X:0.5	Y:-0.5	Z:0
To	X:0.5	Y:0.5	Z:0
Hide	X:0.5	Y:-0.5	Z:0

Scale Tween — Tween Time: 1

From	X:0	Y:0	Z:0
To	X:0	Y:0	Z:0
Hide	X:0	Y:0	Z:0

Passing Data to Widgets

just make the function public in widget class you want to call for example you have Populate method in Inventory class which takes List of UserItem as argument you can call it from anywhere simple by couple of line like

```
var widget = WidgetManager.Instance.GetWidget(WidgetName.Inventory) as wInventory;
if(widget != null) widget.Populate(new List<UserItem>());
```

Update Widget

override UpdateWidget method in your class and write you update logic in it, you can call this method from anywhere simply by calling

```
WidgetManager.Instance.UpdateWidget(WidgetName.MainMenu);
```