Index :

Support.
Demo and more information, please visit :
http://www.clonefactor.com/wordpress/unity3d-show-room/devmanager/

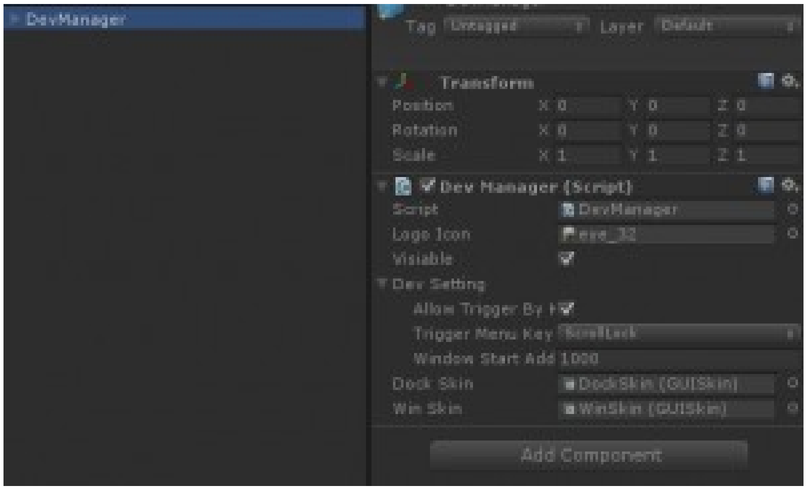
If you have any suggestion & question,
please leave a message on my website(on related project of coz).
OR
send email to canis@clonefactor.com

What is “DevManager"

It’s a tiny tools for developer debug on a machine “without" unity editor.
being test under C#
Features :

- Display you lovely icon on Dock :D
- Dock control all registered GUI window’s visibility.
- Support OnFocus() / OnBlur() function when GUI Window selected.

How to use DevManager :

1. Create a empty gameobject and add “DevManager.cs" script to this object.
2. or you may want to use my standard perfab in package, it’s up to you :D
3. 

Tag Untagged Layer Default
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Script DevManager
Logo Icon eye_32
Visiable ☑
Dev Setting
Allow Trigger By K ☑
Trigger Menu Key ScrollLock
Window Start Add 1000
Dock Skin DockSkin (GUISkin)
Win Skin WinSkin (GUISkin)

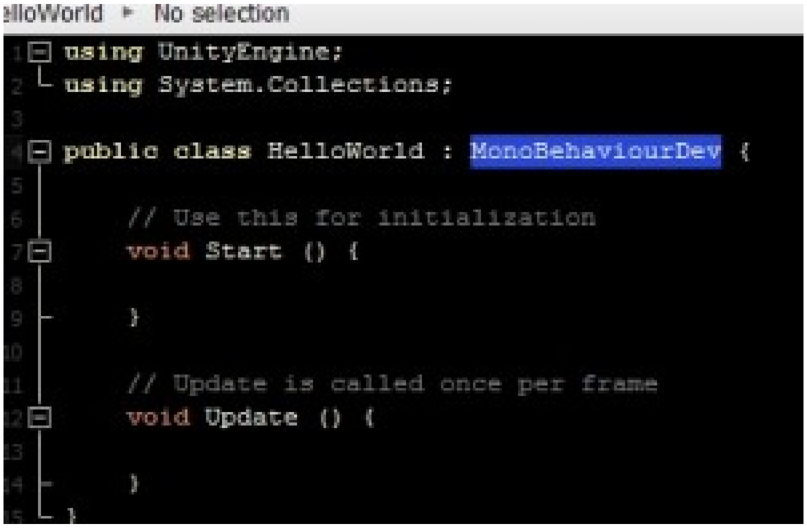
4. It’s finished the basic setup.

Setting of DevManager [Optional]

1. In inspector, you can see a DevSetting.
    1. Logo Icon
        - put your icon in here :D
    2. visible (Default : false)
        - you can control visible of DevManager with other function.
    3. Allow Trigger By Key (Default : Enable)
        - Allow or Disallow player using the assign key to toggle DevManager menu.
    4. Trigger MenuKey (Default : Scroll Lock)
        - A key to toggle DevManager visiable,
        - ~~Don’t forget to use that button to trigger DevManager display in released gameplay.~~
    5. Window Start Address (Default : 1000)
        - each GUIWindow have a unique ID, DevManager start at setting number.
    6. Dock Skin / Win Skin
        - you can design your own screen.

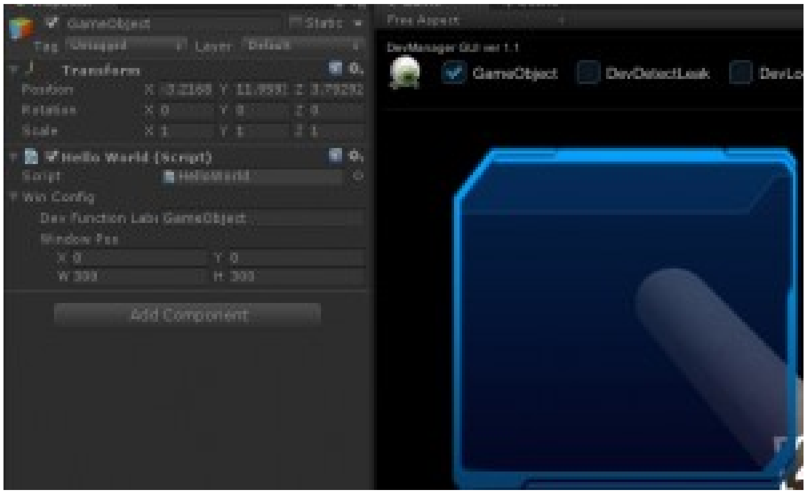
Customize Your own plugin [Optional]

for programmer who need to create thier own GUI window by DevManager.

1. Create your own script, and register GUI window simply just extends a class “MonoBehaviourDev.cs".
2. for example we create a script call “HelloWorld.cs"
3. and add “HelloWorld.cs" to any gameobject on scene.
4. 

elloWorld ▸ No selection

```
using UnityEngine;
using System.Collections;

public class HelloWorld : MonoBehaviourDev {

	// Use this for initialization
	void Start () {

	}

	// Update is called once per frame
	void Update () {

	}
}
```

5. When this script running on scene, it created a empty GUIWindow. (push the button on Dock bar)
6. 

☑ GameObject

Tag Untagged Layer Default

Transform

Position X -3.2168 Y 11.9591 Z 3.79292

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

☑ Hello World (Script)

Script HelloWorld

Win Config

Dev Function Labe GameObject

Window Pos

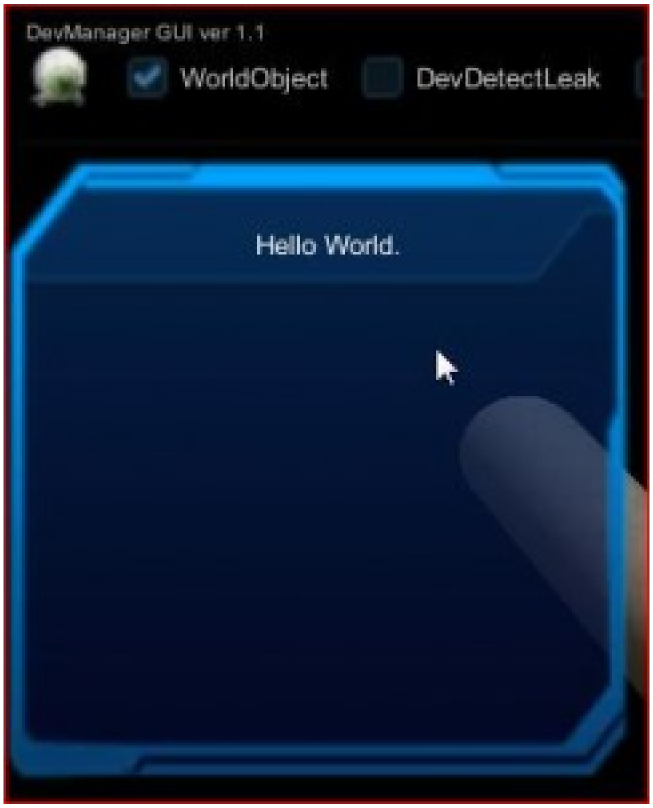
X 0 Y 0

W 300 H 300

7. Now back to the code.
8. override a class call “HandleDebugGUI()" and run the game.

```
1. public override void HandleDebugGUI ()
2. {
3.     GUILayout.Label("Hello World.");
4. }
```

13. Now you got the result like this.



14. about GUI & GUILayout, you may find in here.
15. https://docs.unity3d.com/Documentation/ScriptReference/GUILayout.html
16. One more thing, you may needed to know.
17. when using “MonoBehaviourDev" following method will need extends the parent method.
    - OnEnable()
    - OnDisable()
    - OnGUI()
    - OnFocus()
    - OnBlur()

    extends parent class will look like this.

```
public override void OnEnable ()
{
    base.OnEnable (); // don't remove this line.
    //
    // you can write you code here.
}
```

DevLogViewer intro & Usage

DevLogViewer :

With this tool you can easily check your editor console logs inside the game.

** you will need DevManager to run this script. **

Feature :

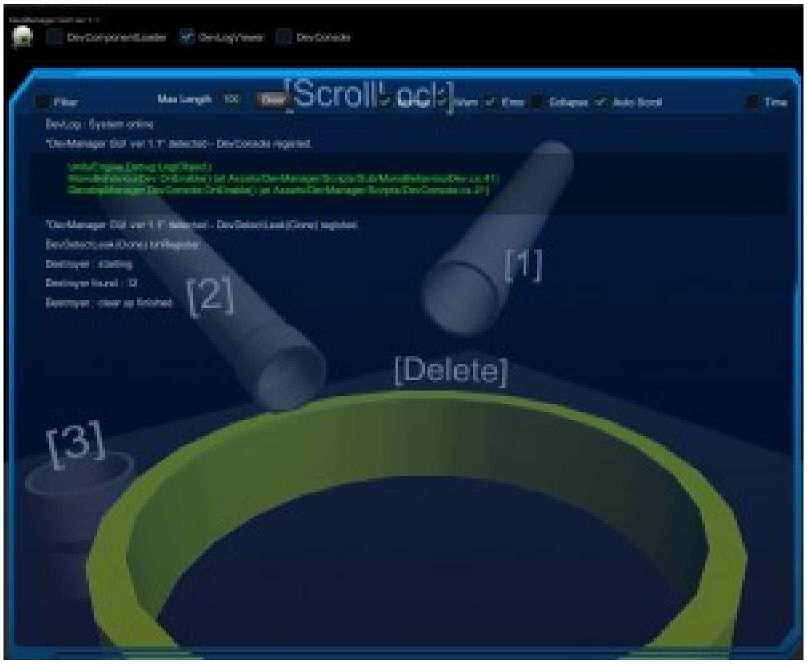
- check your console log in game, on any platform.
- filter base on class / thread name.
- log time & detail trace.
- identity log type by color.
    - Debug.Log = Color.white
    - Debug.LogWarning = Color.yellow
    - Debug.LogError = Color.red
- basic in-game time scale control.

How to use DevLogViewer :

1. You will need DevManager at scene already.
2. Create a empty gameobject, and add “DevLogViewer.cs" on than object.
3. Done.

Usage :

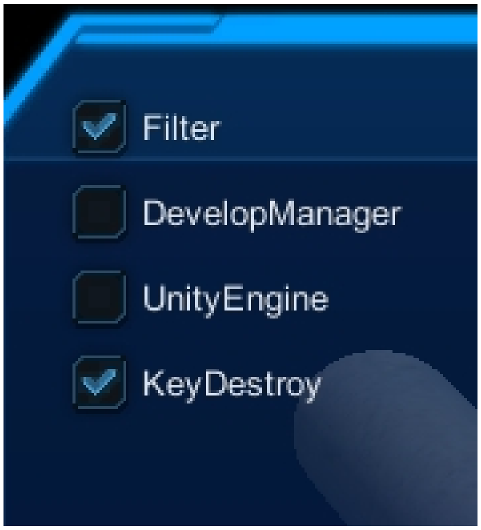
1. Open up DevManager menu (Default key : scroll lock)
2. Switch on “DevLogViewer" on Dock. then you will see the GUIWindow like this.



```
DevLog : System online
"DevManager Dll ver 1.1" detected - DevConsole registed.
    UnityEngine.Debug:Log(Object)
    MonoBehaviourDev: OnEnable() (at Assets/DevManager/Scripts/Sub/MonoBehaviourDev.cs:41)
    DevelopManager.DevConsole:OnEnable() (at Assets/DevManager/Scripts/DevConsole.cs:21)
"DevManager Dll ver 1.1" detected - DevDetectLeak(Clone) registed.
DevDetectLeak(Clone) UnRegister
Destroyer : starting
Destroyer found : 0
Destroyer : clear up finished
```

3. function of those button.

- Filter – toggle on will display a list like this.



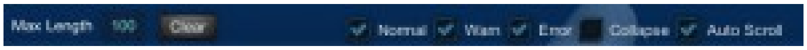
☑ Filter

☐ DevelopManager

☐ UnityEngine

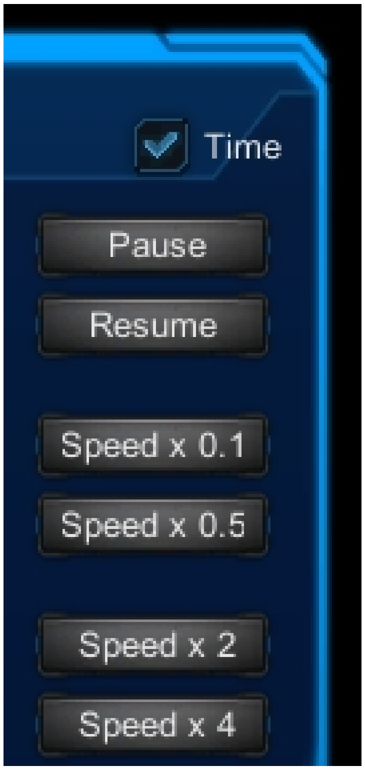
☑ KeyDestroy

- it’s a filter function let user filtering the debug log base on dispatch function class.
- Max length – is the maximum number of log display on GUIWindow.
- Clear – clear all log in this GUIWindow
- Normal – Normal log display
- Warn – Warning log display
- Error – Error log display
- Collapse – if duplicate log keep dispatching in a short period, will display only 1 time
- Auto Scroll – auto scrolling the page to the lastest row of log

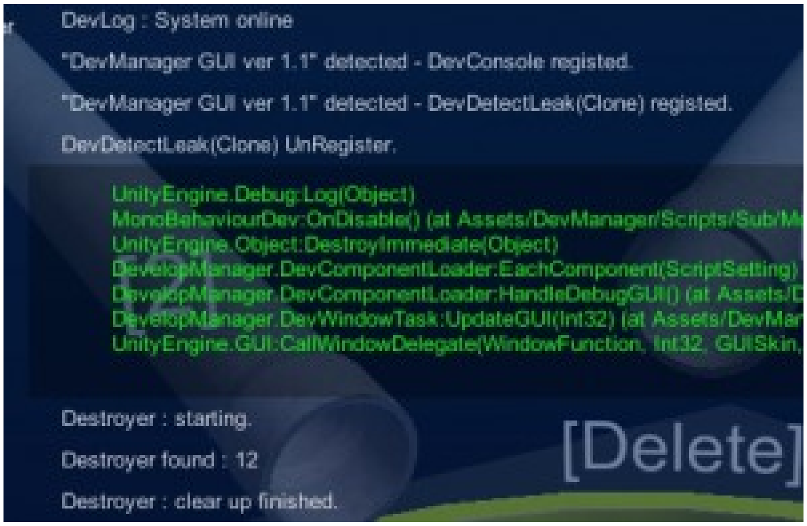


Max Length 100 Clear ☑ Normal ☑ Warn ☑ Error ☐ Collapse ☑ Auto Scroll

- Time – control time scale in-game.



- double click on those log you may found the log stace trace, in green color.



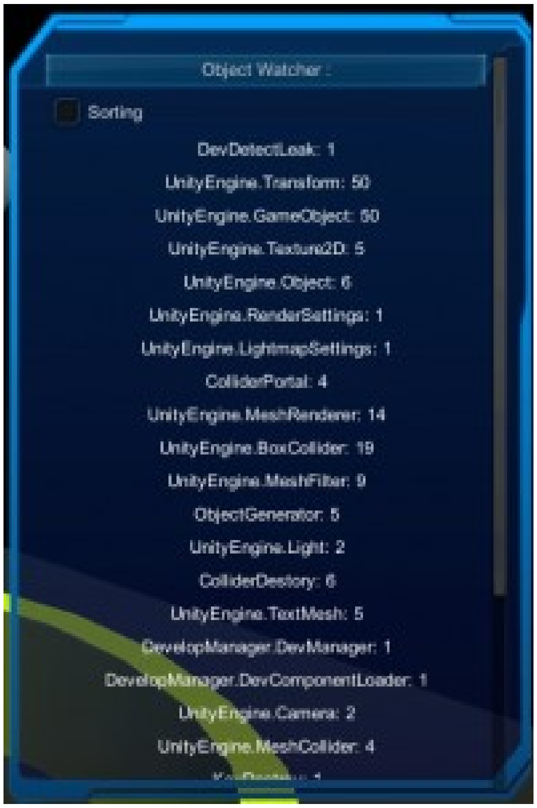
```
DevLog : System online
"DevManager GUI ver 1.1" detected - DevConsole registed.
"DevManager GUI ver 1.1" detected - DevDetectLeak(Clone) registed.
DevDetectLeak(Clone) UnRegister.
    UnityEngine.Debug:Log(Object)
    MonoBehaviourDev:OnDisable() (at Assets/DevManager/Scripts/Sub/M
    UnityEngine.Object:DestroyImmediate(Object)
    DevelopManager.DevComponentLoader:EachComponent(ScriptSetting)
    DevelopManager.DevComponentLoader:HandleDebugGUI() (at Assets/D
    DevelopManager.DevWindowTask:UpdateGUI(Int32) (at Assets/DevMar
    UnityEngine.GUI:CallWindowDelegate(WindowFunction, Int32, GUISkin,
Destroyer : starting.
Destroyer found : 12
Destroyer : clear up finished.
```

The end of DevLogViewer description.

DevDetectLeak

Feature :
A copy of Unity(Pro) > profiler function.
count & list all gameobject, texture number in whole scene. in real time.
you can easier monitor which resource is out of control.
** you will need DevManager to run this script. **
Usage :

1. Open up DevManager menu (Default key : scroll lock)
2. Switch on “DevDetectLeak" on Dock. then you will see the GUIWindow like this.



```
DevDetectLeak: 1
UnityEngine.Transform: 50
UnityEngine.GameObject: 50
UnityEngine.Texture2D: 5
UnityEngine.Object: 6
UnityEngine.RenderSettings: 1
UnityEngine.LightmapSettings: 1
ColliderPortal: 4
UnityEngine.MeshRenderer: 14
UnityEngine.BoxCollider: 19
UnityEngine.MeshFilter: 9
ObjectGenerator: 5
UnityEngine.Light: 2
ColliderDestory: 6
UnityEngine.TextMesh: 5
DevelopManager.DevManager: 1
DevelopManager.DevComponentLoader: 1
UnityEngine.Camera: 2
UnityEngine.MeshCollider: 4
```

3. You can now watch all the asset number in the scene

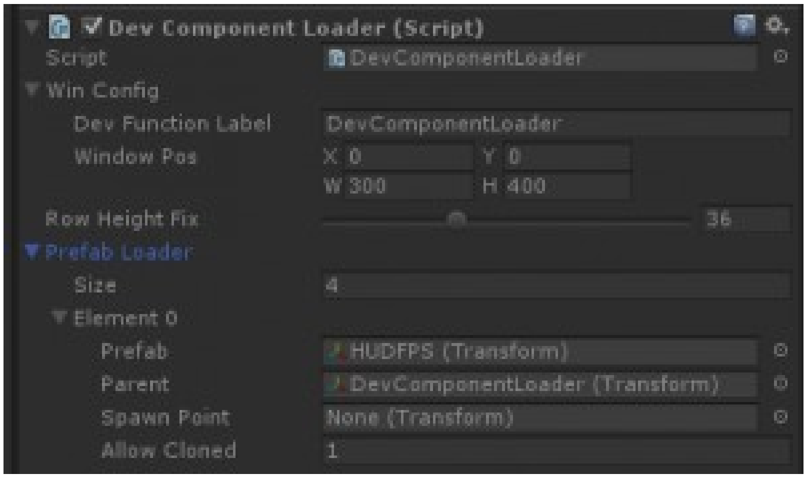
The end of DevDetectLeak description

DevComponentLoader
A tools to dynamic add a perfab/gameobject into the scene.
** you will need DevManager to run this script. **
Feature :

- Add / Remove the selected gameobject into scene, with a single click.
- Assign clone object ‘s parent.
- Assign clone object’s spawn position at scene.
- Config the cloneable maximum number.
- Clone object with rename as “xxxxx (Clone 001)"
- when something go wrong you may easier to found out by their number.

Usage :

1. Create a empty gameobject, add “DevComponentLoader.cs" at than object.
2. Select prefab loader

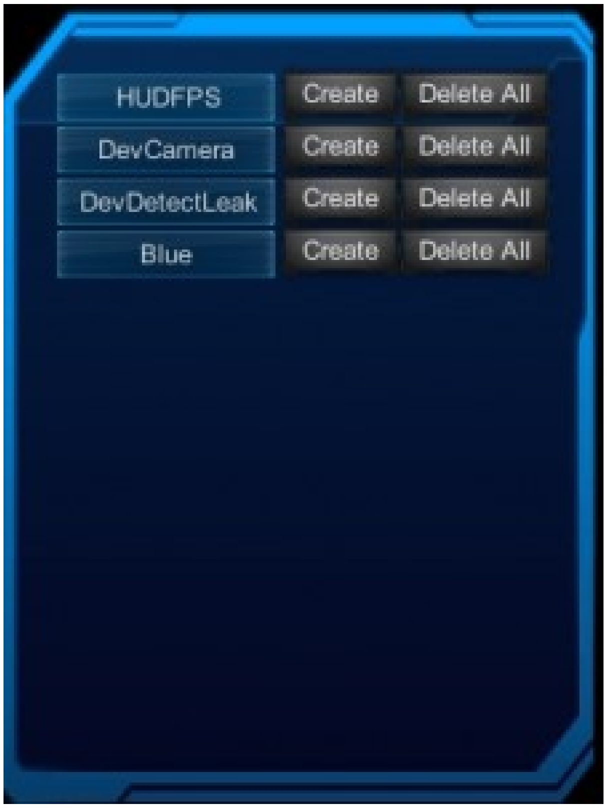


Dev Component Loader (Script)
Script: DevComponentLoader
Win Config
Dev Function Label: DevComponentLoader
Window Pos: X 0 Y 0 W 300 H 400
Row Height Fix: 36
Prefab Loader
Size: 4
Element 0
Prefab: HUDFPS (Transform)
Parent: DevComponentLoader (Transform)
Spawn Point: None (Transform)
Allow Cloned: 1

design how many gameobject / perfab you may want to load in game.
[they wouldn't loaded into scene, until you click it.]

3. Adjust the Loader size.
   - Element > Prefab
   - drag the Gameobject or Perfab you want to be clone in game.
   - Element > Parent
   - setup clone object’s transform parent.
   - Element > Spawn Point
   - the transform location of the global scene.
   - Element > Allow Cloned
   - the number to control the maximum number of cloned object.
4. After setup you DevComponentLoader now you may run the game.
5. Open up DevManager menu (Default key : scroll lock)

6. Switch on “DevComponentLoader" on Dock. then you will see the GUIWindow like this.



7. Now simply just click on “Create" button, there you will found your object was created.
8. Also you can use “Delete All" or “Delete" button to destroy gameobject you created before.

The end of DevComponentLoader description.

DevConsole

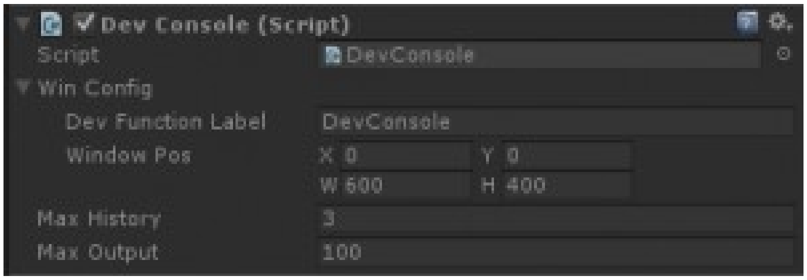
A tools for someone who still love the command console.
allow trigger something by user input.
this is a framework to let developer easier add custom command to this console.
** you will need DevManager to run this script. **

Feature :

- AddCommand() / DelCommand()
- Command with params
- Simply extends methods, Create your tools. :D

Usage :

1. Create a empty gameobject, add “DevConsole.cs" at than object.

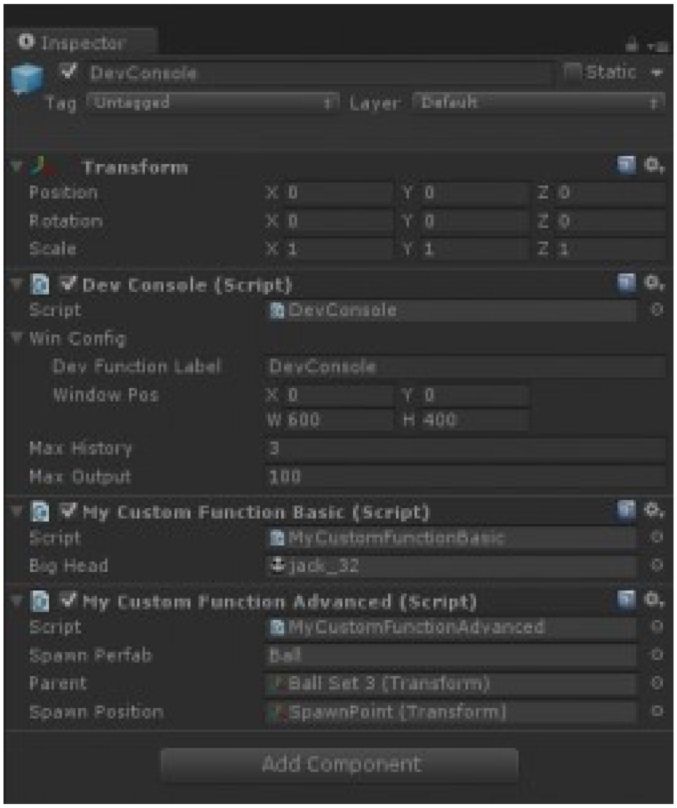


Dev Console (Script)
Script: DevConsole
Win Config
Dev Function Label: DevConsole
Window Pos: X 0 Y 0 W 600 H 400
Max History: 3
Max Output: 100

2. Create and design your own script.
3. we have sample code in pro packet, you may start there.
   a. Example :
   b. MyCustomFucktionBasic.cs

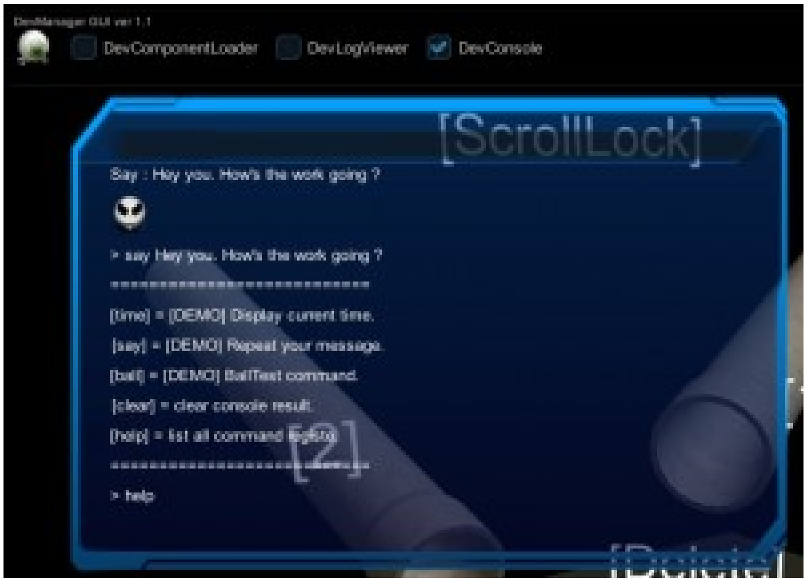
```
using UnityEngine;
using System.Collections.Generic;

public class MyCustomFunctionBasic : DevelopManager.ExtendCommand // <-- extends class
{
    public Texture2D BigHead;


    void Start() {
        AddCommand("say", "[DEMO] Repeat your message.", TalkToMe);
        // DelCommand("say");
        AddCommand("time","[DEMO] Display current time.", CurrentTime);
    }


    // DoSomething should return List<GUIContent> or null
    public List<GUIContent> CurrentTime(string[] _args)
    {
        List<GUIContent> _rst = new List<GUIContent>();
        _rst.Add(new GUIContent("Time : "+ System.DateTime.Now.ToLongDateString() ));
        return _rst;
    }


    public List<GUIContent> TalkToMe(string[] _args)
    {
        List<GUIContent> _rst = new List<GUIContent>();
        // You can use Image. as your output result.
        if( BigHead ) _rst.Add(new GUIContent(BigHead));
        // _args[] is user's input string behind setting command.
        _rst.Add(new GUIContent("Say : "+ string.Join(" ",_args) ));
        return _rst;
    }
}
```

4. After you ready yours script file. add your custom script file on Gameobject. which have “DevConsole.cs”, like this.



```
Inspector
☑ DevConsole                                   □ 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

☑ Dev Console (Script)
Script                  DevConsole
Win Config
  Dev Function Label    DevConsole
  Window Pos            X 0      Y 0
                        W 600    H 400
Max History             3
Max Output              100

☑ My Custom Function Basic (Script)
Script                  MyCustomFunctionBasic
Big Head                jack_32

☑ My Custom Function Advanced (Script)
Script                  MyCustomFunctionAdvanced
Spawn Perfab            Ball
Parent                  Ball Set 3 (Transform)
Spawn Position          SpawnPoint (Transform)

Add Component
```

e.g.
we added “MyCustomFunctionBasic.cs" & “MyCustomFunctionAdvanced.cs" on Gameobject who contain “DevConsole.cs"

5. Start the game.
6. Open up DevManager menu (Default key : scroll lock)
7. Switch on “DevConsole" on Dock. then you will see the GUIWindow like this.
8. 

DevManager GUI ver 1.1
□ DevComponentLoader □ DevLogViewer ☑ DevConsole

```
Say : Hey you. How's the work going ?

> say Hey you. How's the work going ?
============================
[time] = [DEMO] Display current time.
[say] = [DEMO] Repeat your message.
[ball] = [DEMO] BallTest command.
[clear] = clear console result.
[help] = list all command
============================
> help
```

Code Reference :

1. Don’t forget to extends class “DevelopManager.ExtendCommand“
2. AddCommand Method
3. public bool AddCommand(string _name,string _description,TheCommand _callback)
4. will return true when success added.
5. DelCommand Method
6. public bool DelCommand(string _name)
7. will return true when success delete.
8. The Call back function must return a List<GUIContent> result.

```
public List<GUIContent> YourFunction(string[] _params)
{
    List<GUIContent> _rst = new List<GUIContent>();

    // Do something here.

    // return the output GUIContent to DevConsole.
    return _rst;
}
```

The end of DevConsole description.

DevCamera

A tools for the player control the extends camera, and fly anywhere in the scene.

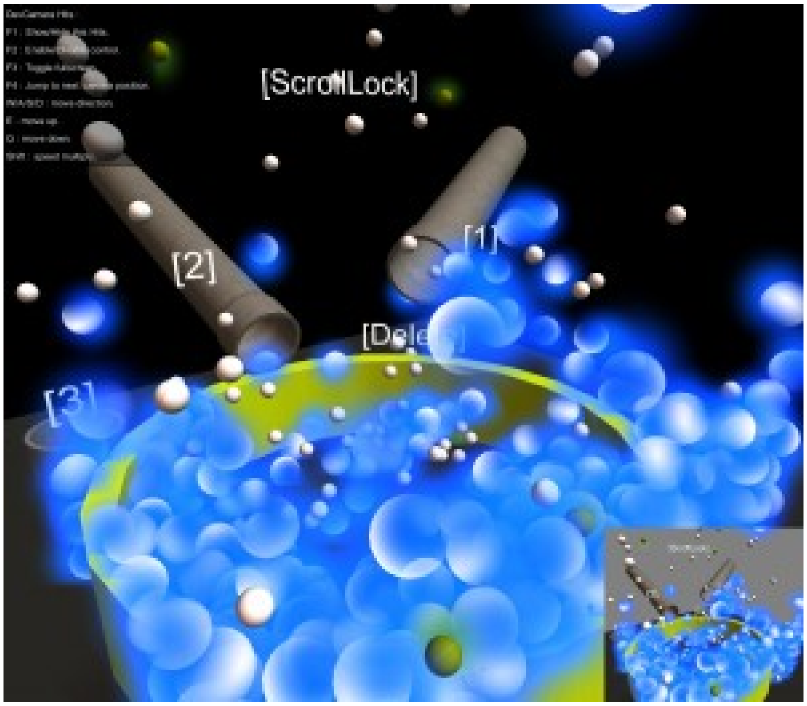
** you will need DevManager to run this script. **

Feature :

- As same as Unity Editor’s scene camera movement method.
- Toggle between fullscreen & your custom preview screen.

Usage :

1. Create a Camera object
2. [Optional] Remove all those “Flare Layer", “GUILayer", “Audio Listener"
3. add “DevConsole.cs" at than object.
4. Start the game.
5. Now you can see the screen like this.



   the camera screen on bottom right is the DevCamera.

6. You will find the tips on the upper-left screen.
   - F1 to toggle display of tips.
   - F2 to Get/Release the movement control of DevCamera
   - F3 to toggle fullscreen.
   - F4 to jump to any other camera in the scene.
   - W,A,S,D for forward,back,left,right direction.
   - Q,E for down, up direction.
   - Shift for Speed multiply
7. [Optional] Suggest you load/unload DevCamera by DevComponentLoader, cause you can’t unload the DevCamera when you directly drag into scene. and DevComponentLoader can handle this.

The end of DevCamera description.