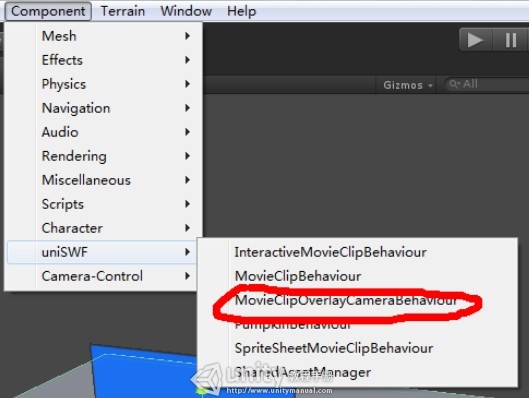
**Unity3D的uniSWF插件动态加载SWF UI资源**

Posted on 2013年07月10日 by U3d / [Unity3D 基础教程](http://www.unitymanual.com/category/manual/unity3d-%e5%9f%ba%e7%a1%80%e6%95%99%e7%a8%8b) /被围观 44 次

uniSWF能把Flash的素材像用AS3编程类似，只不过环境要在C#或者JS中编程。要是想创建一个类似Menu菜单或者像导航一样的菜单，在Unity中首先要选择摄像机，给摄像机添加MovieClipOverlayCameraBehaviour：

[](http://www.unitymanual.com/wp-content/uploads/2013/07/19.jpg)

Unity3D的uniSWF插件动态加载SWF UI资源

因为之后 MovieClipOverlayCameraBehaviour.instance中才包含stage对象的实例，然后需要选择相机的类型：

[](http://www.unitymanual.com/wp-content/uploads/2013/07/28.jpg)

Unity3D的uniSWF插件动态加载SWF UI资源

不然程序就会报错了。

然后我们就可以在程序中编写了，这个程序需要拖放的对象当然也是相机对象，要切记！动态载入和直接给Flash材质有区别需要找的对象放在swf路径的后面：

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 1 | MovieClip minimap = new MovieClip("libufan/swf/UI2.swf:MCArea"); |

要想获得对象的子对象，可以采用getChildByName来获取：

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 1 | playerArea = minimap.getChildByName<MovieClip>("roundArea"); |
| 2 | btn.getChildByName<TextField>("txt") |

当然要是想添加点击事件的话就和AS3中一样了，可以直接添加MouseEvent。下面是所有的程序，以备时间长了回来看看：

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 01 | | | **using** UnityEngine; | |
| 02 | | | **using** System.Collections; | |
| 03 | | | **using** System.Collections.Generic; | |
| 04 | | | **using** pumpkin.display; | |
| 05 | | | **using** pumpkin.events; | |
| 06 | | | **using** pumpkin.tweener; | |
| 07 | | | **using** pumpkin.text; | |
| 08 | | |  | |
| 09 | | | **public** **class** Test1 : MonoBehaviour { | |
| 10 | | | **private** Stage stage; | |
| 11 | | | **private** MovieClip playerArea; | |
| 12 | | | **private** MovieClip btnContainer; | |
| 13 | | | **private** Dictionary<**string**,Handler> menus; | |
| 14 | | |  | |
| 15 | | | **public** **float** scaleRatio = 0.5f; | |
| 16 | | | **public** **float** angle = 0f; | |
| 17 | | | **private** **delegate** **void** Handler(); | |
| 18 | | | **void** Start () { | |
| 19 | | | **if**(!MovieClipOverlayCameraBehaviour.instance)**return**; | |
| 20 | | | stage = MovieClipOverlayCameraBehaviour.instance.stage; | |
| 21 | | | MovieClip minimap = new MovieClip("libufan/swf/UI2.swf:MCArea"); | |
| 22 | | | playerArea = minimap.getChildByName<MovieClip>("roundArea"); | |
| 23 | | | minimap.scaleX = minimap.scaleY = scaleRatio; | |
| 24 | | | minimap.x = stage.stageWidth-minimap.width/2\*scaleRatio-20f; | |
| 25 | | | minimap.y = minimap.height/2\*scaleRatio+20f; | |
| 26 | | | stage.addChild(minimap); | |
| 27 | | | *//Unity3D教程手册：www.unitymanual.com* | |
| 28 | | | BuildMenu(); | |
| 29 | | | } | |
| 30 | | |  | |
| 31 | | | **void** Update () { | |
| 32 | | | angle += Time.deltaTime\*12f; | |
| 33 | | | playerArea.rotation = angle; | |
| 34 | | | } | |
| 35 | | |  | |
| 36 | | | **void** MenuData() | |
| 37 | | | { | |
| 38 | | | menus = new Dictionary<**string**, Handler>(); | |
| 39 | | | menus.**Add**("New Game", HandlerStart); | |
| 40 | | | menus.**Add**("Continue", HandlerStart); | |
| 41 | | | menus.**Add**("Options", HandlerStart); | |
| 42 | | | menus.**Add**("Exit",HandlerStart); | |
| 43 | | | } | |
| 44 | | | *//Unity3D教程手册：www.unitymanual.com* | |
| 45 | | | **void** BuildMenu() | |
| 46 | | | { | |
| 47 | | | MenuData(); | |
| 48 | | |  | |
| 49 | | | btnContainer = new MovieClip(); | |
| 50 | | | stage.addChild(btnContainer); | |
| 51 | | |  | |
| 52 | | | **int** i=0; | |
| 53 | | | **foreach**(KeyValuePair<**string**,Handler> keyvalue **in** menus) | |
| 54 | | | { | |
| 55 | | | MovieClip btn = new MovieClip("libufan/swf/UI2.swf:MCTxt"); | |
| 56 | | | btn.gotoAndStop(1); | |
| 57 | | | btn.looping = **false**; | |
| 58 | | | btn.mouseChildrenEnabled = **false**; | |
| 59 | | | btn.y = i\*(btn.height+10f); | |
| 60 | | | btn.name = keyvalue.Key; | |
| 61 | | | *//btn.getChildByName<TextField>("txt").text = btn.name;* | |
| 62 | | | stage.addChild(btn); | |
| 63 | | | i++; | |
| 64 | | | } | |
| 65 | | |  | |
| 66 | | | btnContainer.x = stage.stageWidth\*0.5f; | |
| 67 | | | btnContainer.y = (stage.stageHeight - btnContainer.height)\*0.5f; | |
| 68 | | | } | |
| 69 | | |  | |
| 70 | | | **void** HandlerStart() | |
| 71 | | | { | |
| 72 | | | Debug.Log("start"); | |
| 73 | | | } | |
| 74 | | | } | |
|  |  | |  |

|  |  |
| --- | --- |
| 001 | **using** UnityEngine; |
| 002 | **using** System.Collections; |
| 003 | **using** System.Collections.Generic; |
| 004 | **using** pumpkin.display; |
| 005 | **using** pumpkin.events; |
| 006 | **using** pumpkin.tweener; |
| 007 | **using** pumpkin.text; |
| 008 |  |
| 009 | **public** **class** L2 : MonoBehaviour { |
| 010 | **private** Stage stage; |
| 011 | **private** MovieClip minimap; |
| 012 | **private** MovieClip background; |
| 013 | **private** MovieClip playerArea; |
| 014 | **private** MovieClip btnContainer; |
| 015 | **private** Dictionary<**string**,Handler> menus; |
| 016 |  |
| 017 | **public** **float** scaleRatio=0.5f; |
| 018 | **public** **float** angle=0f; |
| 019 |  |
| 020 | **private** **delegate** **void** Handler(); |
| 021 |  |
| 022 | **void** Start () { |
| 023 | **if**(!MovieClipOverlayCameraBehaviour.instance)**return**; |
| 024 | stage=MovieClipOverlayCameraBehaviour.instance.stage; |
| 025 |  |
| 026 | background=new MovieClip("learnUniSwf/1/hud2.swf:Background"); |
| 027 | [color=#008ef1]//背景全屏制作方法[/color] |
| 028 | background.x=-2; |
| 029 | background.y=-2; |
| 030 | background.scaleX=(stage.stageWidth+10)/background.width; |
| 031 | background.scaleY=(stage.stageHeight+10)/background.height; |
| 032 | stage.addChild(background); |
| 033 |  |
| 034 | minimap=new MovieClip("learnUniSwf/1/hud2.swf:Minimap"); |
| 035 | playerArea=minimap.getChildByName<MovieClip>("player"); |
| 036 | minimap.scaleX=minimap.scaleY=scaleRatio; |
| 037 | minimap.x=stage.stageWidth-minimap.width/2\*scaleRatio-20f; |
| 038 | minimap.y=minimap.height/2\*scaleRatio+20f; |
| 039 | stage.addChild(minimap); |
| 040 |  |
| 041 | BuildButtons(); |
| 042 | } |
| 043 |  |
| 044 | **void** Update () { |
| 045 | angle+=Time.deltaTime\*12f; |
| 046 | *//动画，模拟minimap 扫描* |
| 047 | playerArea.rotation=angle; |
| 048 | } |
| 049 |  |
| 050 | **void** MenuData(){ |
| 051 | menus=new Dictionary<**string**, Handler>(); |
| 052 | menus.**Add**("New Game",HandlerStart); |
| 053 | menus.**Add**("Continue",HandlerContinue); |
| 054 | menus.**Add**("Options",HandlerOptions); |
| 055 | menus.**Add**("Exit",HandlerExit); |
| 056 | } |
| 057 |  |
| 058 | **void** BuildButtons(){ |
| 059 | MenuData(); |
| 060 |  |
| 061 | btnContainer=new MovieClip("learnUniSwf/1/hud2.swf:SimpleContainer"); |
| 062 | stage.addChild(btnContainer); |
| 063 |  |
| 064 | **int** i=0; |
| 065 | **foreach**(KeyValuePair<**string**,Handler> keyvalue **in** menus){ |
| 066 | MovieClip btn=new MovieClip("learnUniSwf/1/hud2.swf:SimpleButton"); |
| 067 | btn.gotoAndStop(1); |
| 068 | btn.looping=**false** |
| 069 | *//按钮子元素对Mouse 不触发事件，很关键，不然下文evt.target将指向Btn内的某元素* |
| 070 | btn.mouseChildrenEnabled=**false**; |
| 071 | btn.y=i\*(btn.height+10f); |
| 072 | btn.name=keyvalue.Key; |
| 073 | btn.getChildByName<TextField>("txt").text=btn.name; |
| 074 | btnContainer.addChild(btn); |
| 075 | i++; |
| 076 | } |
| 077 | *//Unity3D教程手册：www.unitymanual.com* |
| 078 | btnContainer.x=stage.stageWidth/2; |
| 079 | btnContainer.y=(stage.stageHeight-btnContainer.height)/2; |
| 080 | *//对按钮父容器进行监听，通过事件的target引用找到按钮* |
| 081 | btnContainer.addEventListener(MouseEvent.CLICK,ClickHandler); |
| 082 | btnContainer.addEventListener(MouseEvent.MOUSE\_ENTER,OnEnter); |
| 083 | btnContainer.addEventListener(MouseEvent.MOUSE\_LEAVE,OnLeave); |
| 084 | } |
| 085 |  |
| 086 | **void** HandlerStart(){ |
| 087 | Debug.Log("handler start...."); |
| 088 | } |
| 089 |  |
| 090 | **void** HandlerContinue(){ |
| 091 | Debug.Log("handler continue...."); |
| 092 | } |
| 093 |  |
| 094 | **void** HandlerOptions(){ |
| 095 | Debug.Log("handler options...."); |
| 096 | } |
| 097 | *//Unity3D教程手册：www.unitymanual.com* |
| 098 | **void** HandlerExit(){ |
| 099 | Debug.Log("handler exit...."); |
| 100 | } |
| 101 |  |
| 102 | **void** ClickHandler(CEvent evt){ |
| 103 | Handler h=menus[(evt.target **as** MovieClip).name]; |
| 104 | **if**(h!=**null**){ |
| 105 | h(); |
| 106 | } |
| 107 | } |
| 108 |  |
| 109 | **void** OnEnter(CEvent evt){ |
| 110 | (evt.target **as** MovieClip).gotoAndPlay(1); |
| 111 | } |
| 112 |  |
| 113 | **void** OnLeave(CEvent evt){ |
| 114 | *//(evt.target as MovieClip).gotoAndStop(1);* |
| 115 | } |
| 116 |  |
| 117 | } |