**LOGO渐隐渐出的实现方法**

Posted on 2013年04月01日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script)/被围观 271 次

很多游戏大作都出现过LOGO渐隐渐出的效果，以下[**脚本**](http://www.unitymanual.com/category/script)就能直接实现：

using UnityEngine;

using System.Collections;

public class SplashScreen : MonoBehaviour  
{  
public string levelToLoad = ""; // this has to correspond to a level (file>build settings)  
public Texture2D splashLogo; // the logo to splash;  
public float fadeSpeed = 0.3f;  
public float waitTime = 0.5f; // seconds to wait before fading out  
public bool waitForInput = false; // if true, this acts as a "press any key to continue"  
private float timeFadingInFinished = 0.0f;

public enum SplashType  
{  
LoadNextLevelThenFadeOut,  
FadeOutThenLoadNextLevel  
}  
public SplashType splashType;

private float alpha = 0.0f;

private enum FadeStatus  
{  
FadeIn,  
FadeWaiting,  
FadeOut  
}  
private FadeStatus status = FadeStatus.FadeIn;

private Camera oldCam;  
private GameObject oldCamGO;

private Rect splashLogoPos = new Rect();  
public enum LogoPositioning  
{  
Centered,  
Stretched  
}  
public LogoPositioning logoPositioning;

private bool loadingNextLevel = false;

void Start()  
{  
oldCam = Camera.main;  
oldCamGO = Camera.main.gameObject;

if (logoPositioning == LogoPositioning.Centered)  
{  
splashLogoPos.x = (Screen.width \* 0.5f) - (splashLogo.width \* 0.5f);  
splashLogoPos.y = (Screen.height \* 0.5f) - (splashLogo.height \* 0.5f);

splashLogoPos.width = splashLogo.width;  
splashLogoPos.height = splashLogo.height;  
}  
else  
{  
splashLogoPos.x = 0;  
splashLogoPos.y = 0;

splashLogoPos.width = Screen.width;  
splashLogoPos.height = Screen.height;  
}

if (splashType == SplashType.LoadNextLevelThenFadeOut)  
{  
DontDestroyOnLoad(this);  
DontDestroyOnLoad(Camera.main);  
}  
if ((Application.levelCount <= 1) || (levelToLoad == ""))  
{  
Debug.Log("I need to have a level to load or the value of level To load is wrong!");  
return;  
}  
}

void Update()  
{  
switch(status)  
{  
case FadeStatus.FadeIn:  
alpha += fadeSpeed \* Time.deltaTime;  
break;  
case FadeStatus.FadeWaiting:  
if ((!waitForInput && Time.time >= timeFadingInFinished + waitTime) || (waitForInput && Input.anyKey))  
{  
status = FadeStatus.FadeOut;  
}  
break;  
case FadeStatus.FadeOut:  
alpha += -fadeSpeed \* Time.deltaTime;  
break;  
}  
}

void OnGUI()  
{  
if (splashLogo != null)  
{  
GUI.color = new Color(GUI.color.r, GUI.color.g, GUI.color.b, Mathf.Clamp01(alpha));  
GUI.DrawTexture(splashLogoPos, splashLogo);  
if (alpha > 1.0f)  
{  
status = FadeStatus.FadeWaiting;  
timeFadingInFinished = Time.time;  
alpha = 1.0f;  
if (splashType == SplashType.LoadNextLevelThenFadeOut)  
{  
oldCam.depth = -1000;  
loadingNextLevel = true;  
Application.LoadLevel(levelToLoad);  
}  
}  
if (alpha < 0.0f)  
{  
if (splashType == SplashType.FadeOutThenLoadNextLevel)  
{  
Application.LoadLevel(levelToLoad);  
}  
else  
{  
Destroy(oldCamGO); // somehow this doesn't work  
Destroy(this);  
}  
}  
}  
}

void OnLevelWasLoaded(int lvlIdx)  
{  
if (loadingNextLevel)  
{  
Destroy(oldCam.GetComponent<AudioListener>());  
Destroy(oldCam.GetComponent<GUILayer>());  
}  
}

void OnDrawGizmos()  
{  
Gizmos.color = new Color(1f, 0f, 0f, .5f);  
Gizmos.DrawCube(transform.position, new Vector3(1, 1, 1));  
}  
}