# [KSFramework：Unity3D开发框架快速入门](https://www.cnblogs.com/zhaoqingqing/p/5658826.html)

本文为作者原创，转载请注明出处：<https://www.cnblogs.com/zhaoqingqing/p/5658826.html>

## KSFramework知识

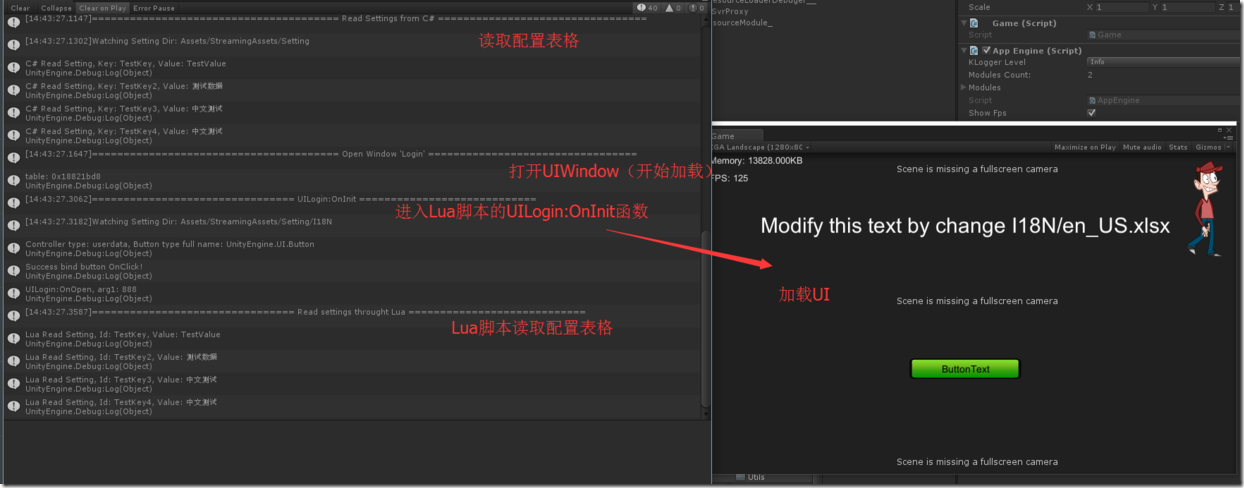
<https://github.com/mr-kelly/KSFramework>

KSFramework是一个整合KEngine、SLua和一些开发组件组成的全功能Unity 5开发框架，适合有一定规模的团队使用。

热重载是KSFramework的开发重点——在不重启游戏的前提下，重载代码、配置表可立刻看到修改效果，最大限度的提升开发、调试的速度，并且在运营阶段方便的进行产品热更新。

## 看看Demo！

双击打开Assets/Game.unity场景，点击播放。

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092138732-1583176925.png)

图：开始Game.unity后的日志输出

### KSFramework做了这些事

这时候KSFramework的默认Demo开始，做了这些事情：

* 基础模块的启动
* Lua模块的启动
* 尝试读取并打印Excel表格GameConfig.xlsx内容
* 加载UI界面Login
* 执行UI界面Login的Lua脚本
* Lua脚本绑定UI控件、设置UI控件
* Lua脚本读取并打印Excel表格GameConfig.xlsx内容

### 囊括KSFramework核心模块的使用

总而言之，这个Demo囊括了KSFramework中的几个核心模块的使用：

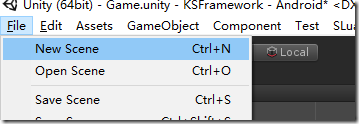
* KEngine中的Setting模块的使用
* KEngine中的UI的资源加载
* SLua脚本引擎与UI的联合

接下来，将模仿这个Demo，创建/加载一个新的UI界面、创建/读取一个新的配置表格。

# 尝试做一个公告界面Billboard

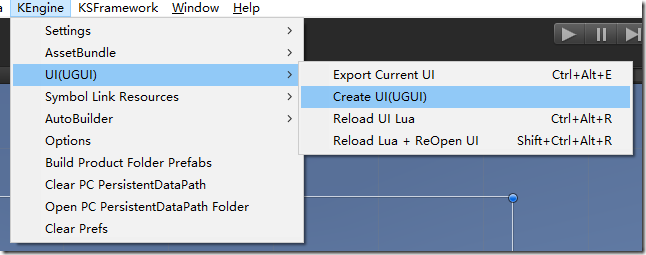
接下来，我们将创建一个UI公告牌（Billboard），使用Lua脚本，并从配置表读取公告内容。

## 1. 创建UI资源

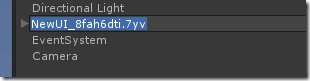
[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092140436-992642223.png)

图：创建New Scene

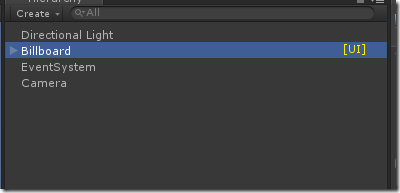
### Create UI

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092143811-947912366.png)

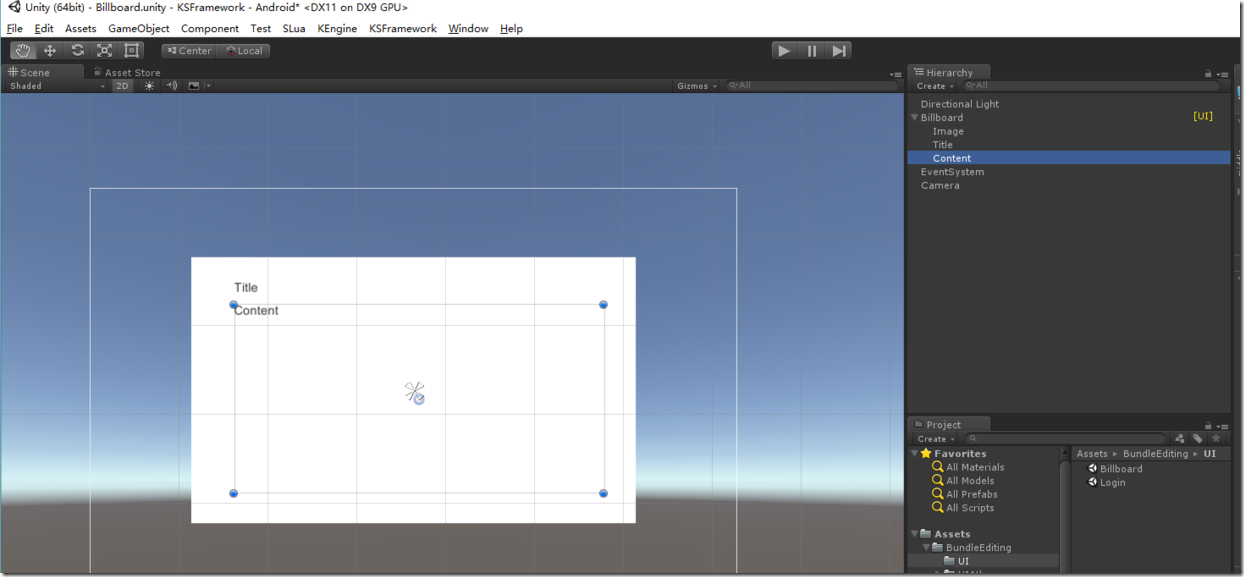
图：**KEngine** – **UI** - **Create UI** 创建UI布局

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092144779-531343280.png)

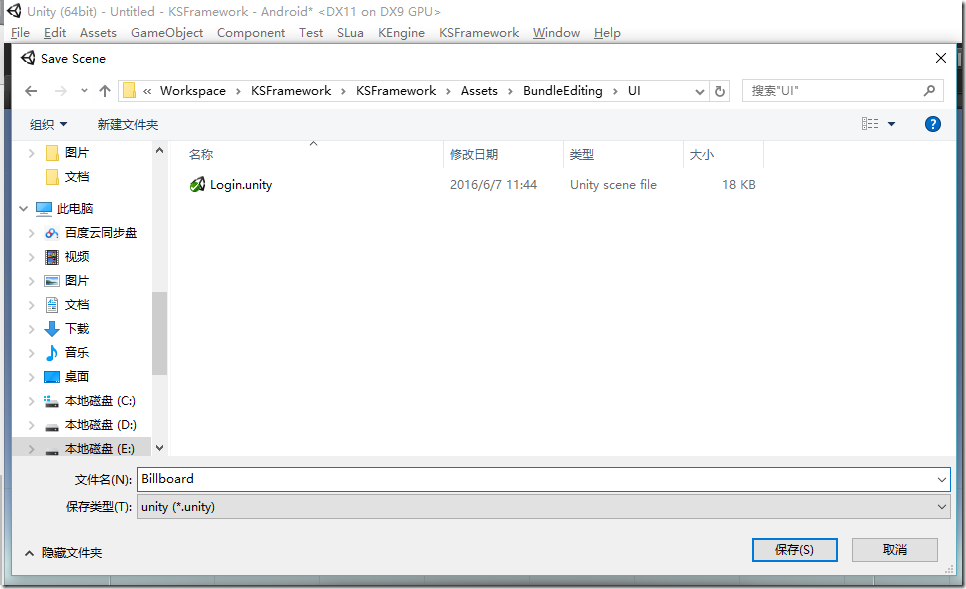
图：点击Create UI后，默认随机名字，把UI名字修改为**Billboard**

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092145686-211366706.png)

图：修改UI名字为Billboard，UI界面右边带有**黄色UI标识**

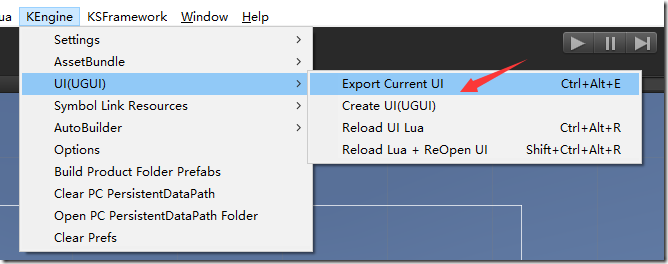
[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092146764-2077714389.png)

图：编辑一下UI场景，一个背景Image，两个Label

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092148482-974173789.png)

保存一下场景，保存到Assets/BundleEditing/UI/Billboard.unity

### Export UI To Assetbundle

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092149904-1775728005.png)

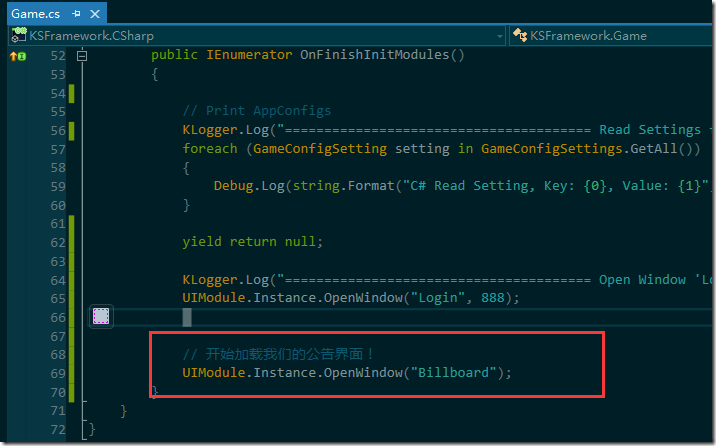
图：导出 — 打包AssetBundle，快捷键Ctrl+Alt+E

## 2. 加载UI界面

好了，Billboard界面创建好了，也导出成了AssetBundle。

### 显示UI面板(OpenWindow方式)

接下来，我们通过代码打开界面。

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092151014-1588514034.png)

### 代码示例

编辑 **Assets**/**Code**/**Game.cs**

在 **OnFinishInitModules** 函数的末端，加上这样的一句：

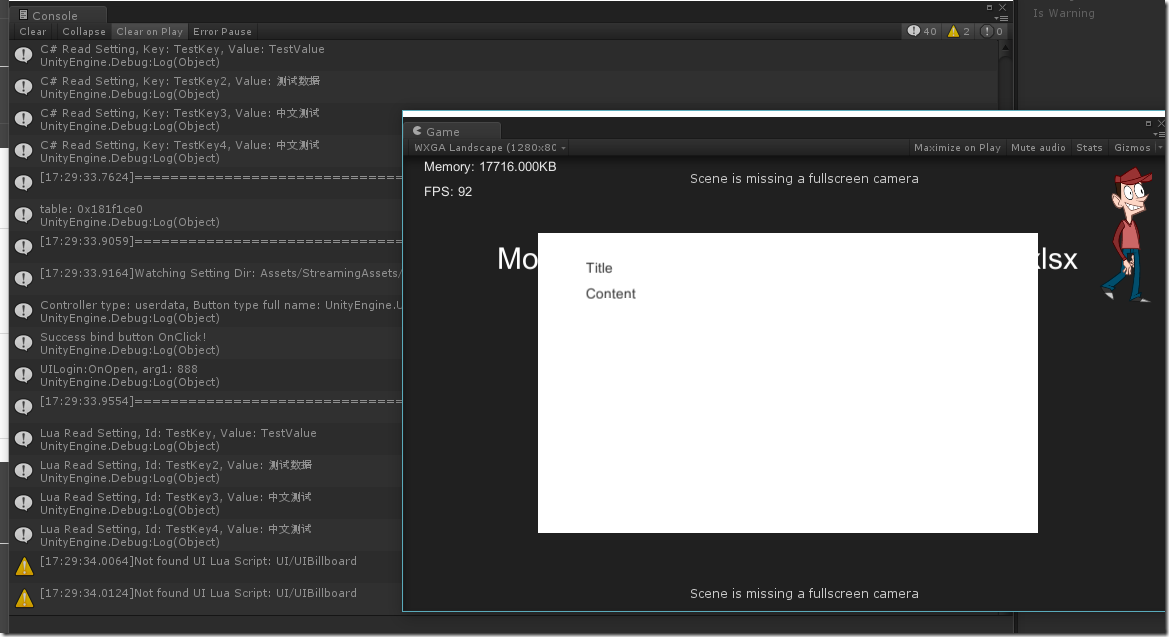
// 开始加载我们的公告界面！

UIModule.Instance.OpenWindow("Billboard");

完成并保存。

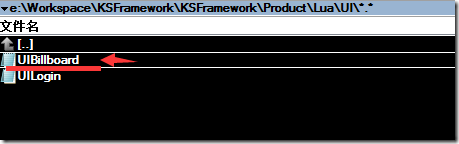
### 运行效果

打开场景 Assets/**Game.unity**，点击播放按钮：

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092152279-687509114.png)

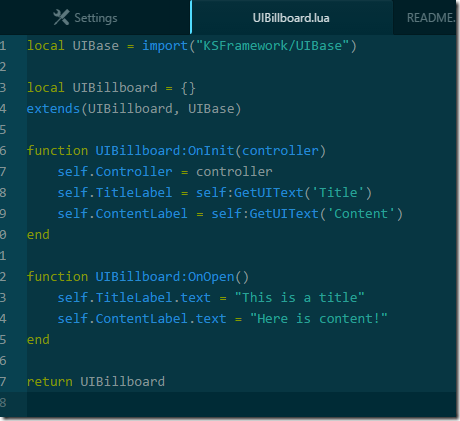
我们的UI通过AssetBundle打开了，弹出提示找不到UI Lua脚本，接下来我们创建Lua脚本吧

## 3. 创建Lua脚本

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092155404-210795028.png)

图：在目录 **Product**/**Lua**/**UI** 中新建一个 **lua文件**

### Lua脚本示例

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092159404-1680136155.png)

写一段Lua代码：UIBillboard的执行逻辑

[IMG_268](https://www.cnblogs.com/zhaoqingqing/p/javascript:void(0);)

local UIBase = import("KSFramework/UIBase")

local UIBillboard = {}

extends(UIBillboard, UIBase)

function UIBillboard:OnInit(controller)

self.Controller = controller

self.TitleLabel = self:GetUIText('Title')

self.ContentLabel = self:GetUIText('Content')end

function UIBillboard:OnOpen()

self.TitleLabel.text = "This is a title"

self.ContentLabel.text = "Here is content!"end

return UIBillboard

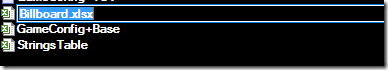
[IMG_269](https://www.cnblogs.com/zhaoqingqing/p/javascript:void(0);)

这段lua中，创建了一个**Table叫UIBillboard**，这个table必须有**OnInit(controller)**函数。它通过代码设置了UI中的文字。

好了，接下来，我们要为策划准备配置表了。

## 4. 创建配置表

打开 **Product**/**SettingSource** 目录，复制一份 **StringsTable.xlsx**，并改名叫 **Billboard.xlsx** 吧

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092200482-299213714.png)

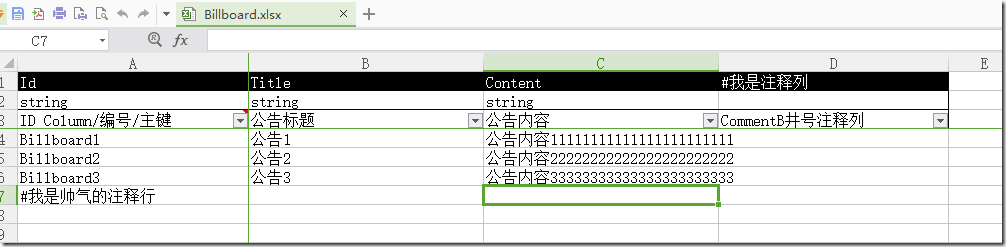
### 公告需求

用Excel打开我们新的 **Billboard.xlsx**，编辑我们的公告。

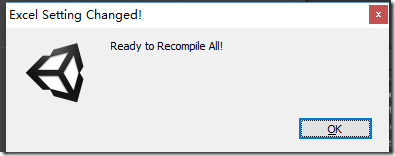
我们大概定一下需求，我们假设写入3条公告，每次打开公告随机显示其中一条。每个公告，我们给它一个英文ID，一列中文标题，一列中文内容。

### 配置公告Excel

Excel表修改如下： （增加公告内容 ）

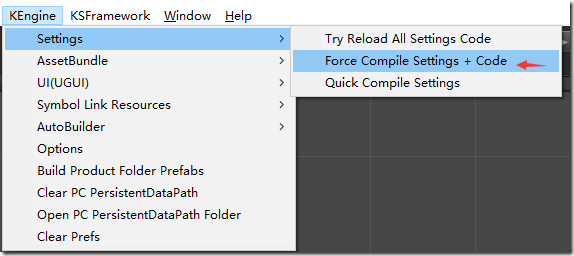
[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092201576-1272662856.png)

### Unity自动监测Excel变动

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092203217-1637328597.png)

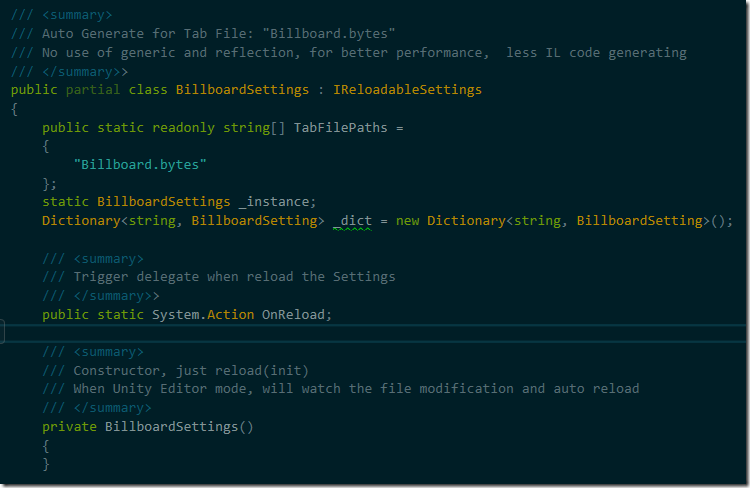
图：回到Unity，监测到Excel变动。点击OK。

### 自动生成配置表代码

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092204514-595568038.png)

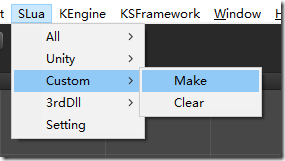
上一步监测到变动，只编译Excel表，手动执行一些重新编译，并生成配置表代码

### tab生成的BillboardSettings类

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092205670-586330155.png)

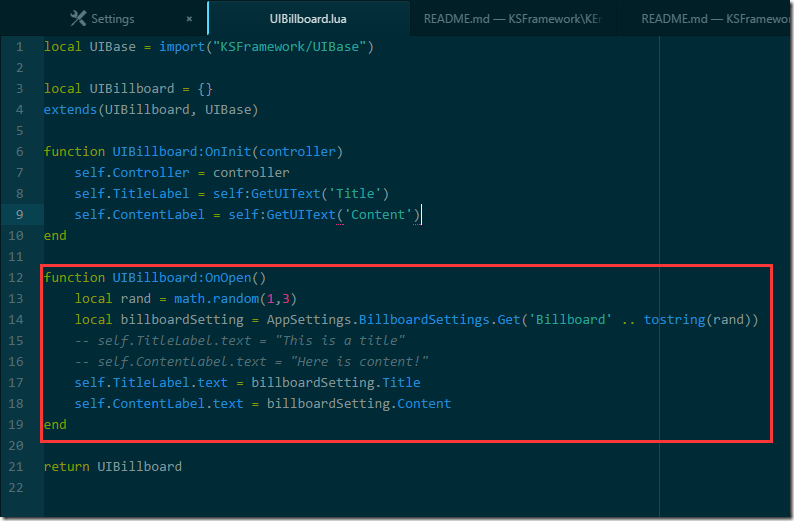
图：这时候，打开AppSettings.cs代码文件，我们可以发现，已经生成名叫BillboardSettings的类了

### 生成SLua的静态代码

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092208514-1390044605.png)

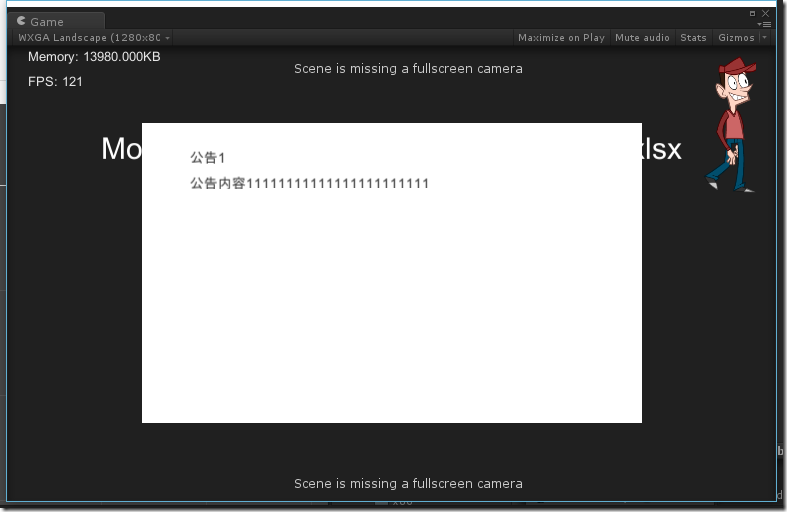
因为我们要在Lua使用BillboardSettings读取配置表，这里需要重新生成一下SLua的静态代码

### 修改Lua随机读取一条公告

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092209436-1604934757.png)

接下来修改Lua代码，随机读取一条公告，并设置Content、Title

### 公告内容随机显示

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092211451-1588073550.png)

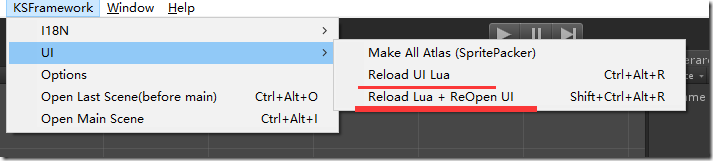
图：运行播放 **Game.unity**，我们的公告界面完成了

公告界面完成了。我们创建了一个UI、写了C#和Lua代码加载它、然后创建了一张配置表，再从Lua读取配置表，设置UI的显示。

## 玩玩热重载

### 热重载Lua

接着我们刚才运行的 **Game.unity** 。 我们尝试一下对Lua热重载：**在不重启游戏的情况，快速重载Lua代码，方便程序调试**。

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092213170-1713037747.png)

图：菜单 **KSFramework** –> **UI** –> **Reload+ReOpen**：热重载Lua脚本，并重新打开目前处在打开状态的UI界面

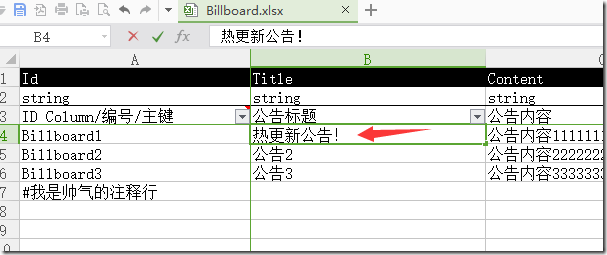
我们可以从菜单执行热重载并重新打开UI界面，或者使用快捷键 **Ctrl+Alt+Shift+R**。   
由于我们的Lua脚本中，每次执行随机获取一条公告。因此可以看到公告内容在不停的变动着。

### 热重载的实现原理

热重载的实现，主要依赖于每个lua脚本的最后一行——return Table；C#层在执行完Lua脚本后，会把这个返回的Table缓存起来，每次需要使用table时，直接从缓存拿；而热重载，实际就是把这个缓存table删掉，这样当需要用到这个table时，就会重新执行Lua文件，来获取Table，这样就达到了热重载得目的。

### 热重载Excel表格

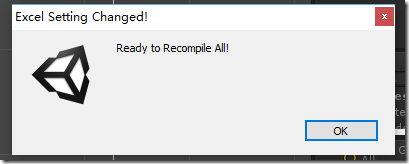
我们保持运行刚刚的 **Game.unity**，不要停止掉。这时候我们去修改Excel表格。

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092214186-623235467.png)

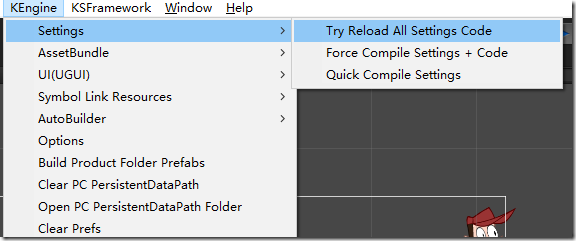
图：修改Excel表格

### 修改配置表并重载

保存后，回到Unity，提示表格有改动。

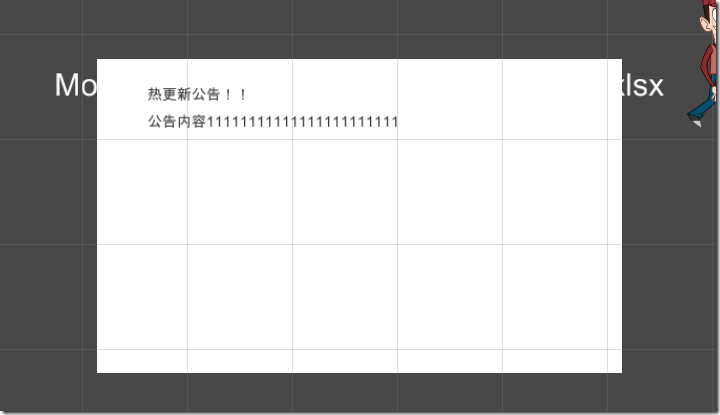
[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092215529-1390957878.png)

图：发现表格有变动，点击OK编译表

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092219295-297050292.png)

图：从菜单中心一下重载配置表格吧

### 重载效果

[](http://images2015.cnblogs.com/blog/363476/201607/363476-20160711092220482-1144296806.png)

图：Ctrl+Alt+Shift+R刷新Lua

重载Lua，我们的新修改的配置表内容生效了。

至此，我们的Lua和配置表的改动，都可以在不重启、不重新转菊花的情况下快速修改。

## 版权说明

文／公的Kelly[mr-kelly]（简书作者）   
原文链接：<http://www.jianshu.com/p/ccb491ed4260>   
著作权归作者所有，转载请联系作者获得授权。

### KSFramework系列

github地址：<https://github.com/mr-kelly/KSFramework>

欢迎大家到 [github提issues](https://github.com/mr-kelly/KSFramework/issues)