

# Tui 教程

## 目录

Tui 简介 .....	2
环境需求.....	2
安装 TuiEditor 插件 .....	3
预览 Tui Flash Sample.....	10
运行 Tui Sample .....	17
从 HelloTui 开始 .....	19
新建 HelloTui Flash CS 工程.....	19
标签层.....	20
添加 iphone 分辨率.....	21
新建 iphone 版 panel_start .....	22
添加一个按钮.....	29
新建 iphone 版 panel_game.....	36
把 HelloTui 导入程序 .....	40
多语言资源管理.....	48

# Tui 简介

- Tui 是什么?  
Tui 是基于 FlashCS Pro6 的游戏 UI 编辑器。
- Tui 支持的控件类型有哪些?  
面板: flash 中元件类型为影片剪辑, 面板里可以嵌套面板, 图片或按钮  
图片: flash 中为图片资源如 background.png  
按钮: flash 中元件类型为 button
- Tui 的定位是什么?  
Tui 专门面向小游戏, 由于中大规模游戏中用到的控件太多, Tui 目前并不支持太多控件。
- 使用 Tui 开发的已上线产品  
piper1.0.1 : <https://itunes.apple.com/cn/app/piper/id560114562?ls=1&mt=8>
- 更多问题请联系我  
仝鹏飞 27001909@qq.com

## 环境需求

### [Flash CS Professional6.0](#)

Download: [http://www.adobe.com/cfusion/tdrc/index.cfm?product=flash&loc=en\\_us](http://www.adobe.com/cfusion/tdrc/index.cfm?product=flash&loc=en_us)

### [AdobeExtensionManager6.0](#)

Download: [http://www.adobe.com/exchange/em\\_download/](http://www.adobe.com/exchange/em_download/)

### [cocos2d-x-2.0.3](#)

Download: <http://cocos2d-x.googlecode.com/files/cocos2d-2.0-x-2.0.3.zip>

### [Tui](#)

Download: <https://github.com/tongpengfei/tui/archive/master.zip>

### [TinyXML](#)

Download: <http://sourceforge.net/projects/tinyxml/files/latest/download>

# 安装 TuiEditor 插件

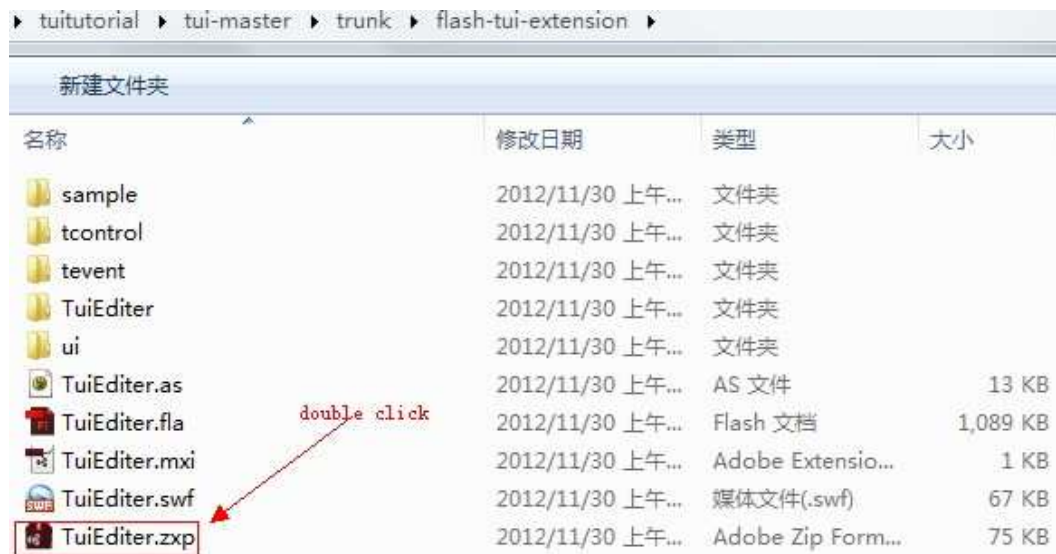
安装插件有两种办法:

## 1. 如果 FlashCS6.0 是完整版:

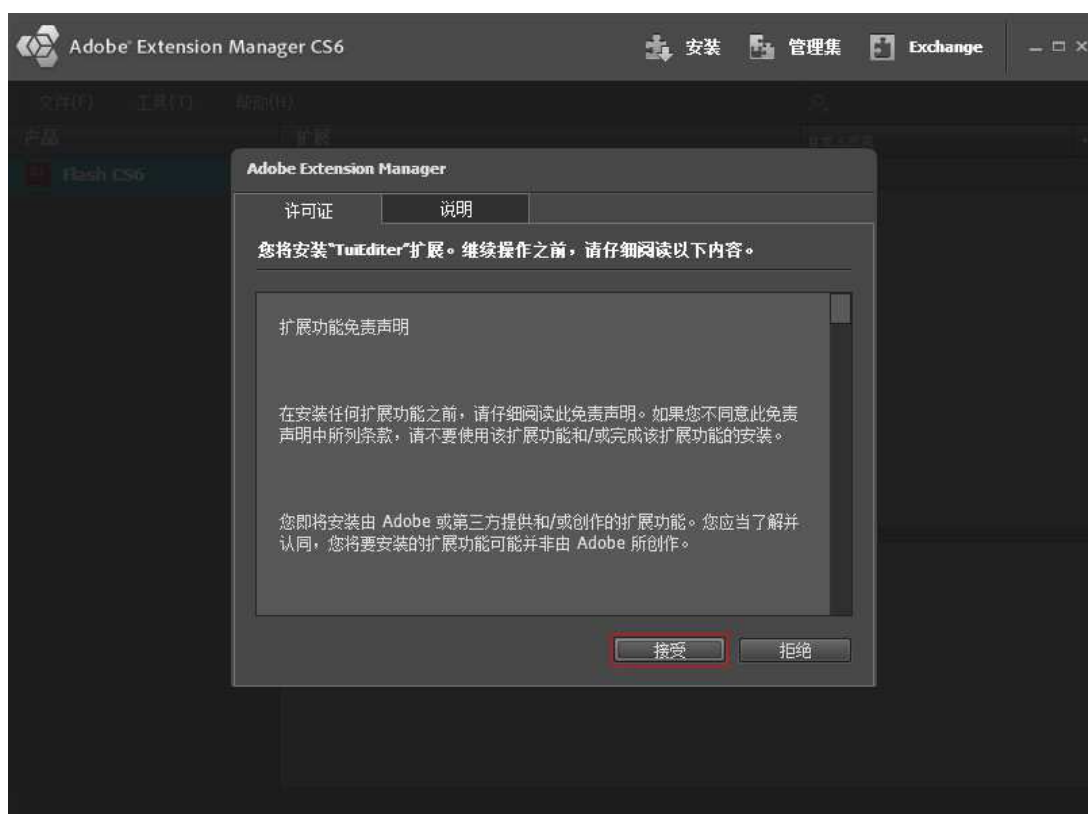
- a) 安装 AdobeExtensionManager6.0
- b) 安装 TuiEditor 插件

解压 tui-master.zip 后我们把 tui-master/trunk 做为\$(TUITRUNK)。

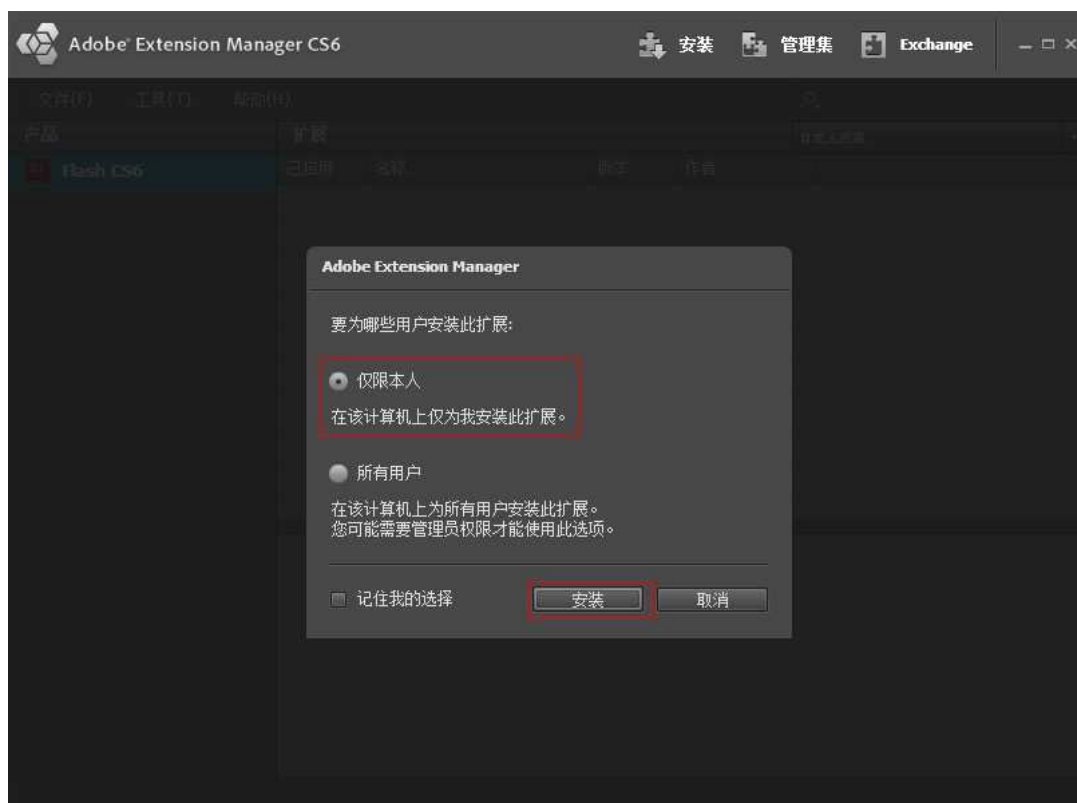
进入\$(TUITRUNK) \flash-tui-extension 目录双击 TuiEditor.zxp 文件



点击接受



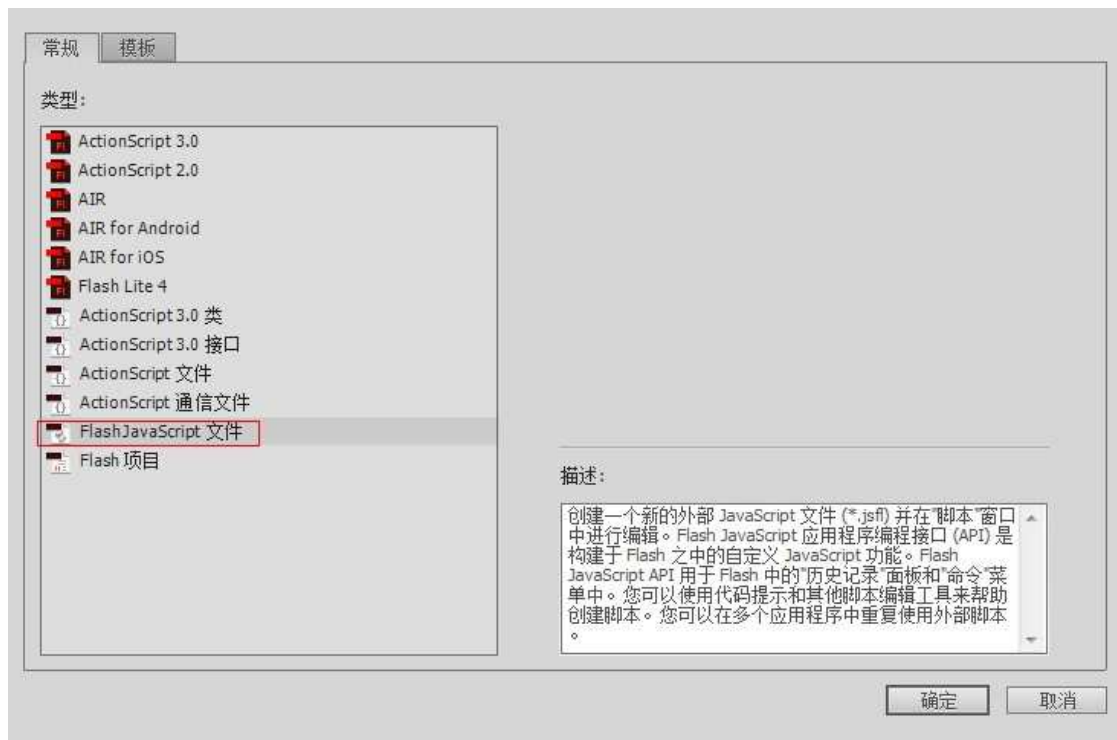
点击安装



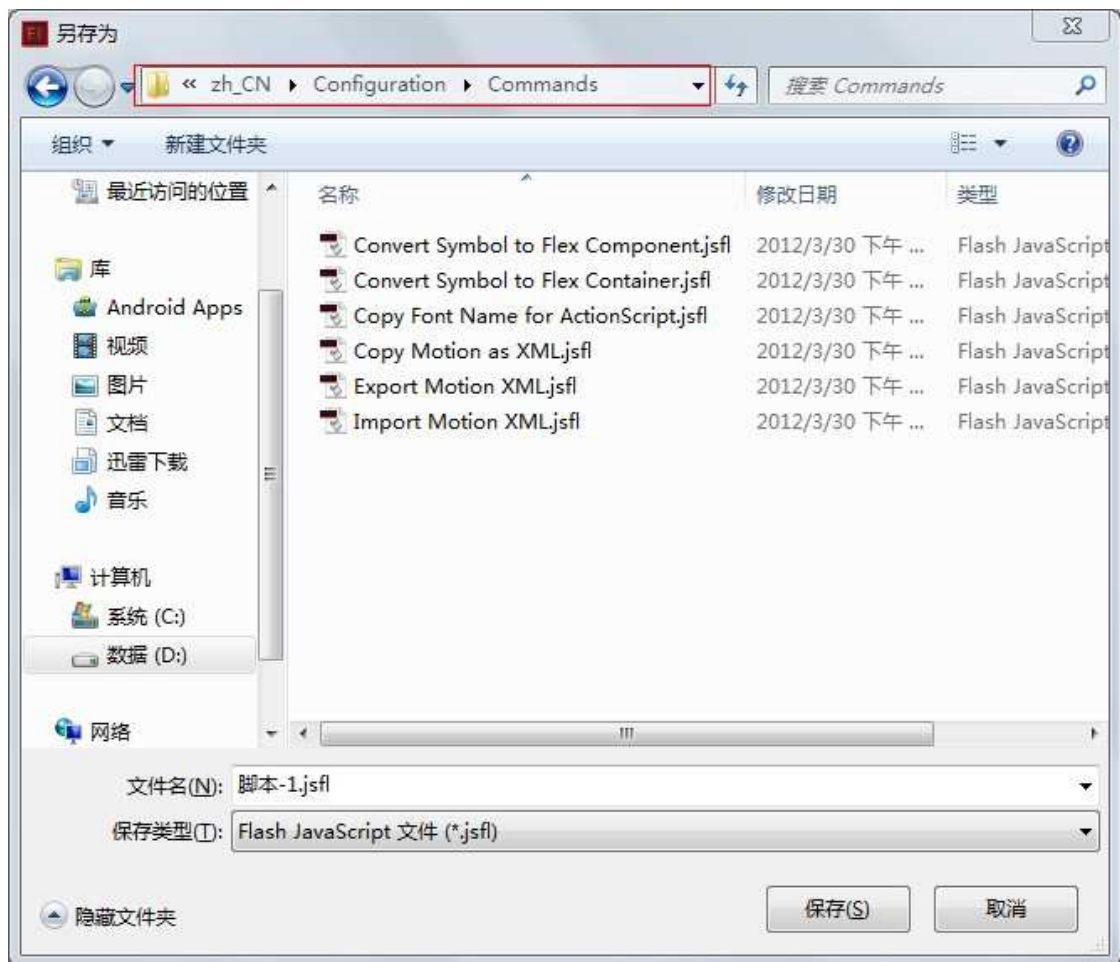
## 2. 如果 FlashCS6.0 是绿色精简版

a) 找到 Flash 插件的安装目录

打开 FalshCS->文件->新建->Flash JavaScript 文件



这时会建立一个空的 jsfl 文件,然后点击: 文件->另存为



复制地址栏里的目录地址, 进入该目录的上一级, 找到 WindowSWF 目录

Local > Adobe > Flash CS6 > zh_CN > Configuration >			
共享 新建文件夹			
名称	修改日期	类型	大小
include	2012/12/23 下午...	文件夹	
Javascript	2012/12/23 下午...	文件夹	
Keyboard Shortcuts	2012/12/23 下午...	文件夹	
Kinds	2012/12/23 下午...	文件夹	
Libraries	2012/12/23 下午...	文件夹	
LivePreviewClasses	2012/12/23 下午...	文件夹	
Motion Presets	2012/12/23 下午...	文件夹	
Players	2012/12/23 下午...	文件夹	
ProcScripts	2012/12/23 下午...	文件夹	
Project	2012/12/23 下午...	文件夹	
Publish Profiles	2012/12/23 下午...	文件夹	
Screen Types	2012/12/23 下午...	文件夹	
StartPage	2012/12/23 下午...	文件夹	
SwfTagCache	2013/2/26 下午 ...	文件夹	
Tools	2012/12/23 下午...	文件夹	
WebServices	2012/12/23 下午...	文件夹	
WindowSWF	2013/2/26 下午 ...	文件夹	
Workspace	2013/2/26 下午 ...	文件夹	
XULControls	2012/12/23 下午...	文件夹	
AutoRecoverFileNames.txt	2013/1/28 下午 ...	文本文档	0 KB
filelist.txt	2012/3/30 下午 ...	文本文档	515 KB
First Run.dat	2012/12/23 下午...	媒体文件(.dat)	369 KB
First Run.log	2013/2/26 下午 ...	文本文档	100 KB
fontinfo.txt	2013/1/28 下午 ...	文本文档	31 KB
LastFirstRunInstall.dat	2012/7/13 上午 ...	媒体文件(.dat)	1 KB
missfont.map	2012/3/30 下午 ...	Linker Address ...	0 KB

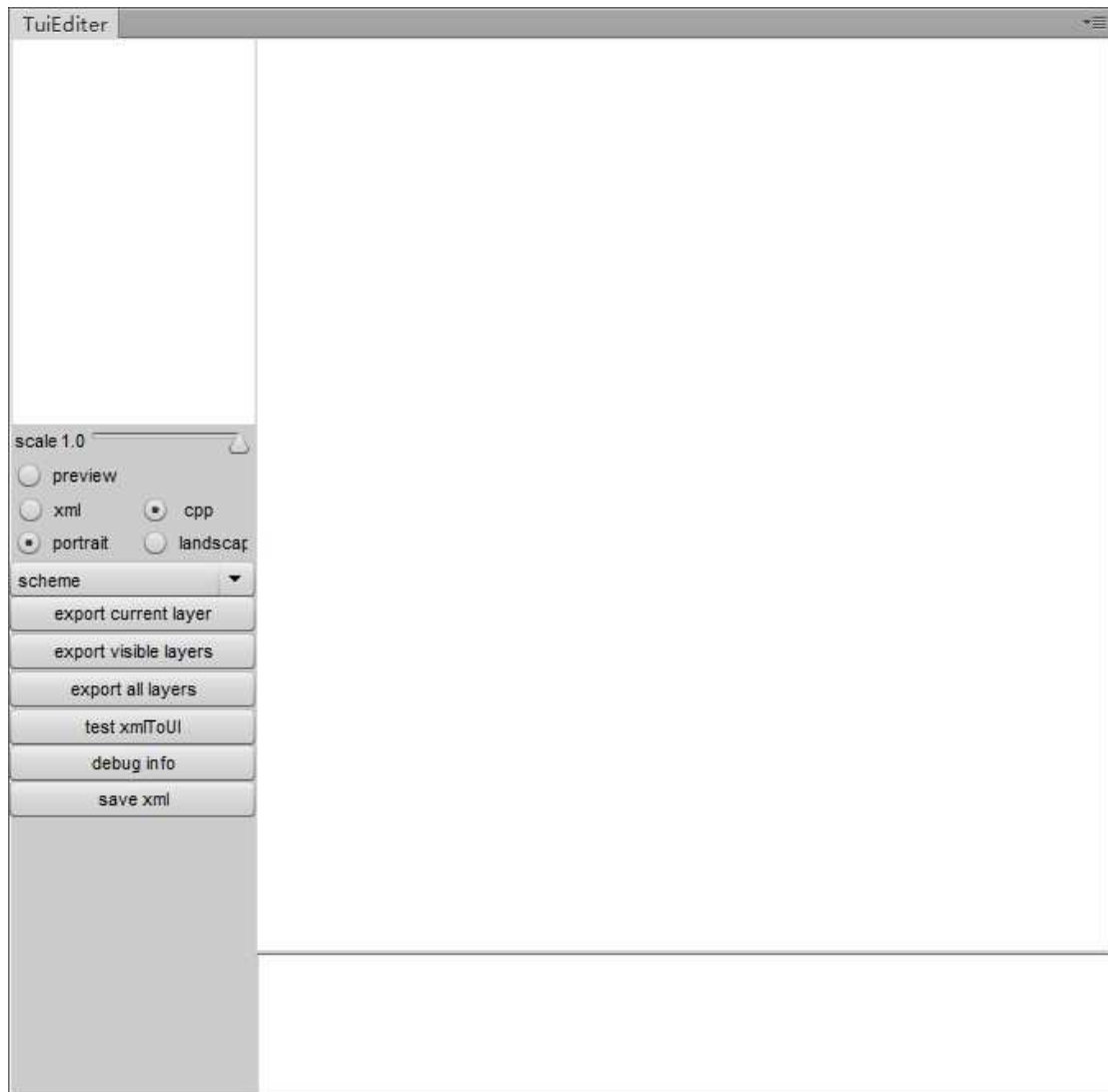
- b) 进入\$(TUITRUNK) \flash-tui-extension 目录把 TuiEditor 复制到 WindowsSWF 目录下

tuitutorial > tui-master > trunk > flash-tui-extension >			
新建文件夹			
名称	修改日期	类型	大小
sample	2012/11/30 上午...	文件夹	
tcontrol	2012/11/30 上午...	文件夹	
tevent	2012/11/30 上午...	文件夹	
TuiEditor	2012/11/30 上午...	文件夹	
ui	2012/11/30 上午...	文件夹	
TuiEditor.as	2012/11/30 上午...	AS 文件	13 KB
TuiEditor fla	2012/11/30 上午...	Flash 文档	1,089 KB
TuiEditor.mxi	2012/11/30 上午...	Adobe Extensio...	1 KB
TuiEditor.swf	2012/11/30 上午...	媒体文件(.swf)	67 KB
TuiEditor.zxp	2012/11/30 上午...	Adobe Zip Form...	75 KB



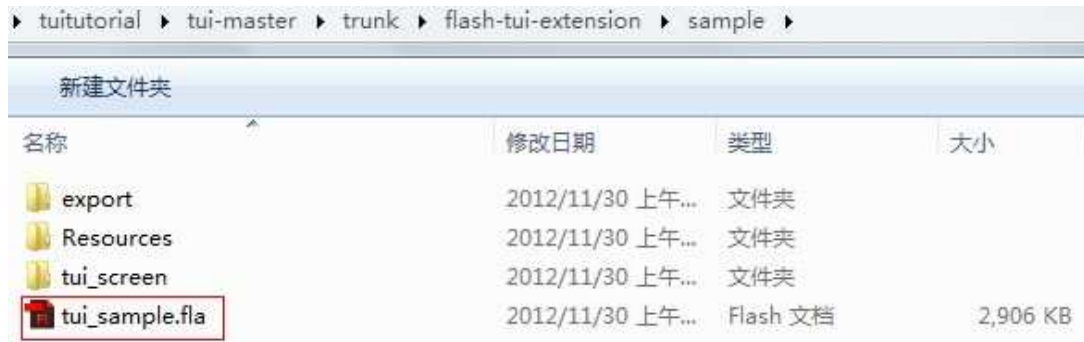
### 3. 打开插件窗口

打开 FlashCS->窗口->其他面板->TuiEditor

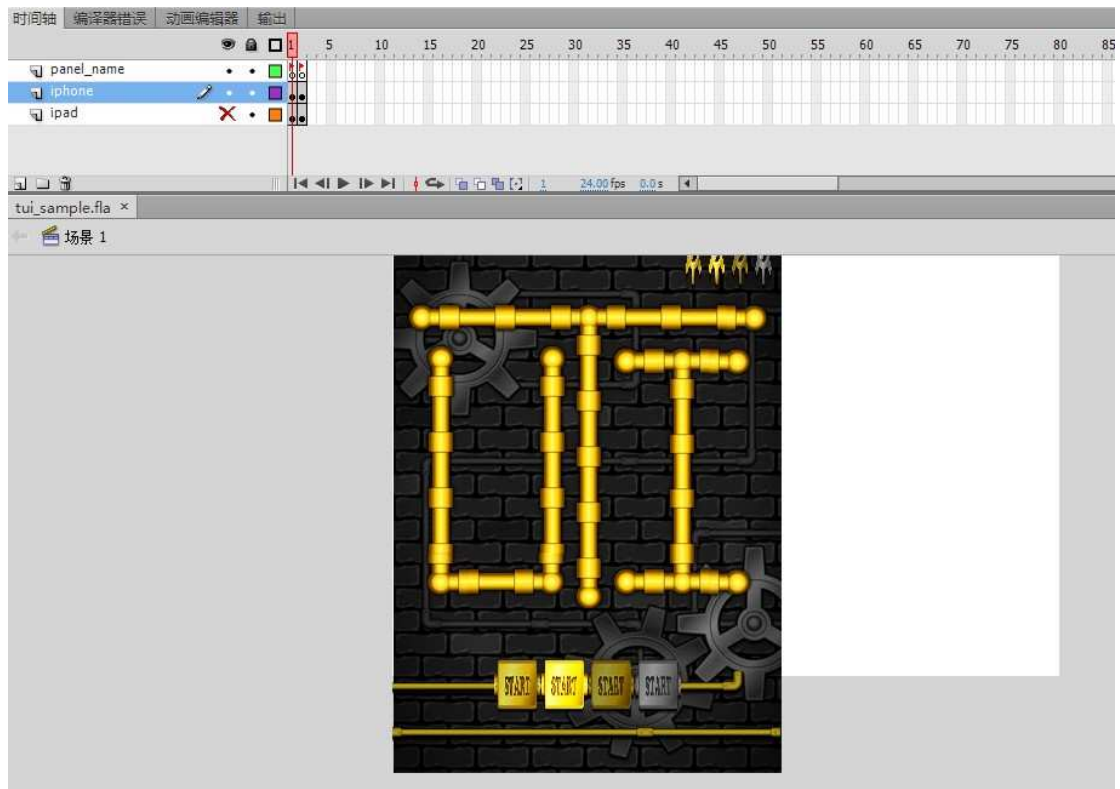


# 预览 Tui Flash Sample

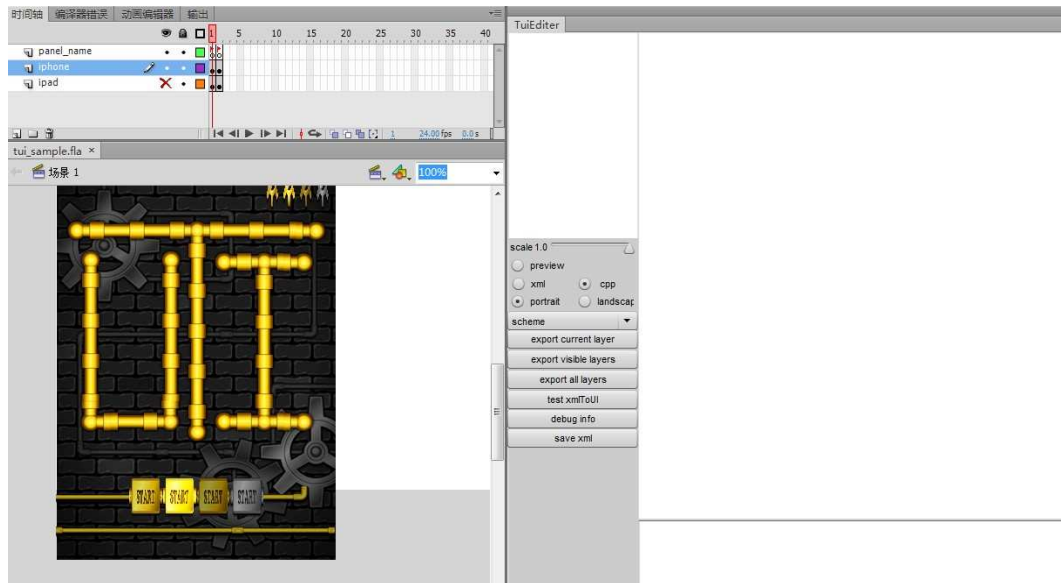
1. 进入目录\$(TUITRUNK)\flash-tui-extension\sample 双击 tui\_sample fla



2. 打开后显示的画面为

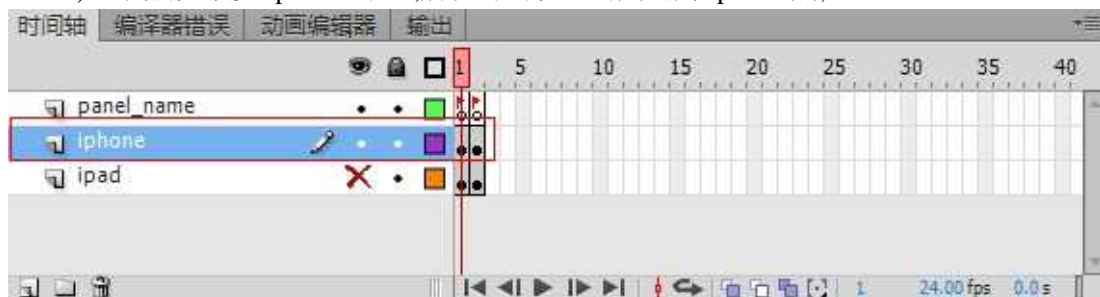


3. 打开插件: 窗口->其他面板->TuiEditor

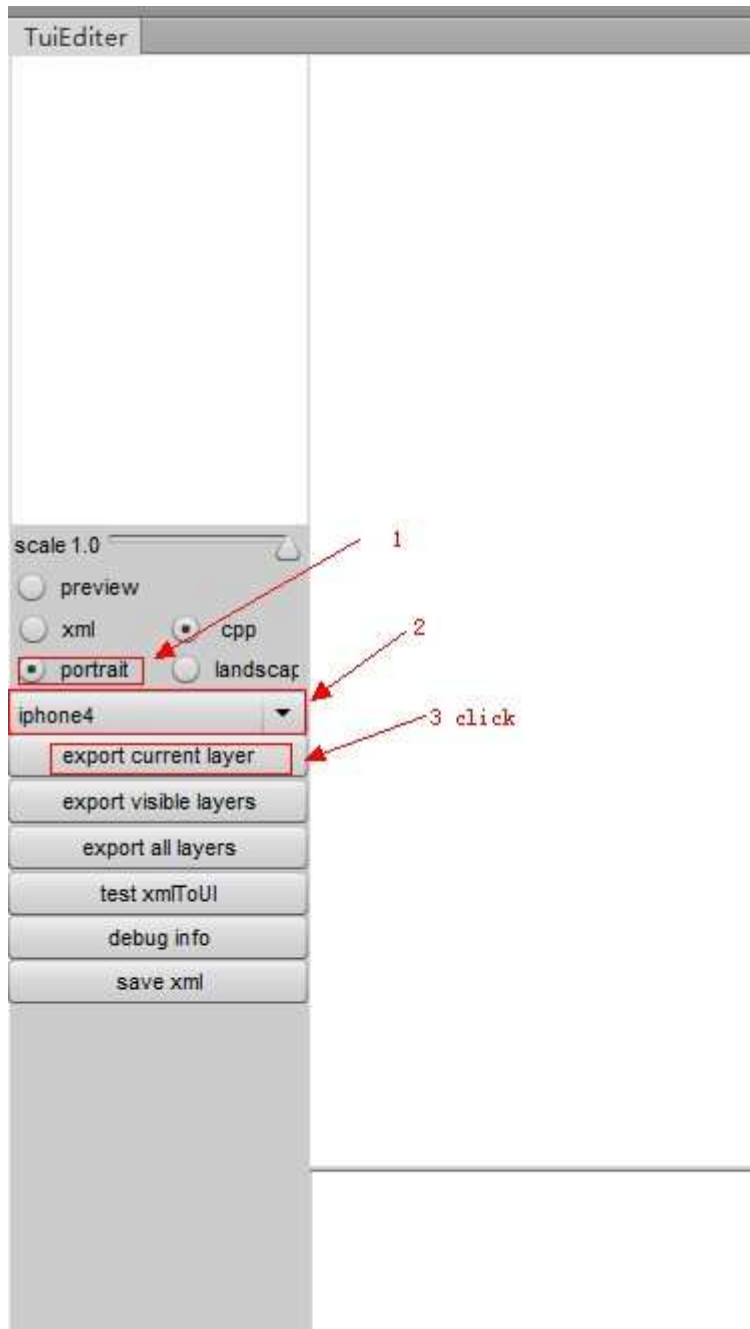


4. 预览 Tui

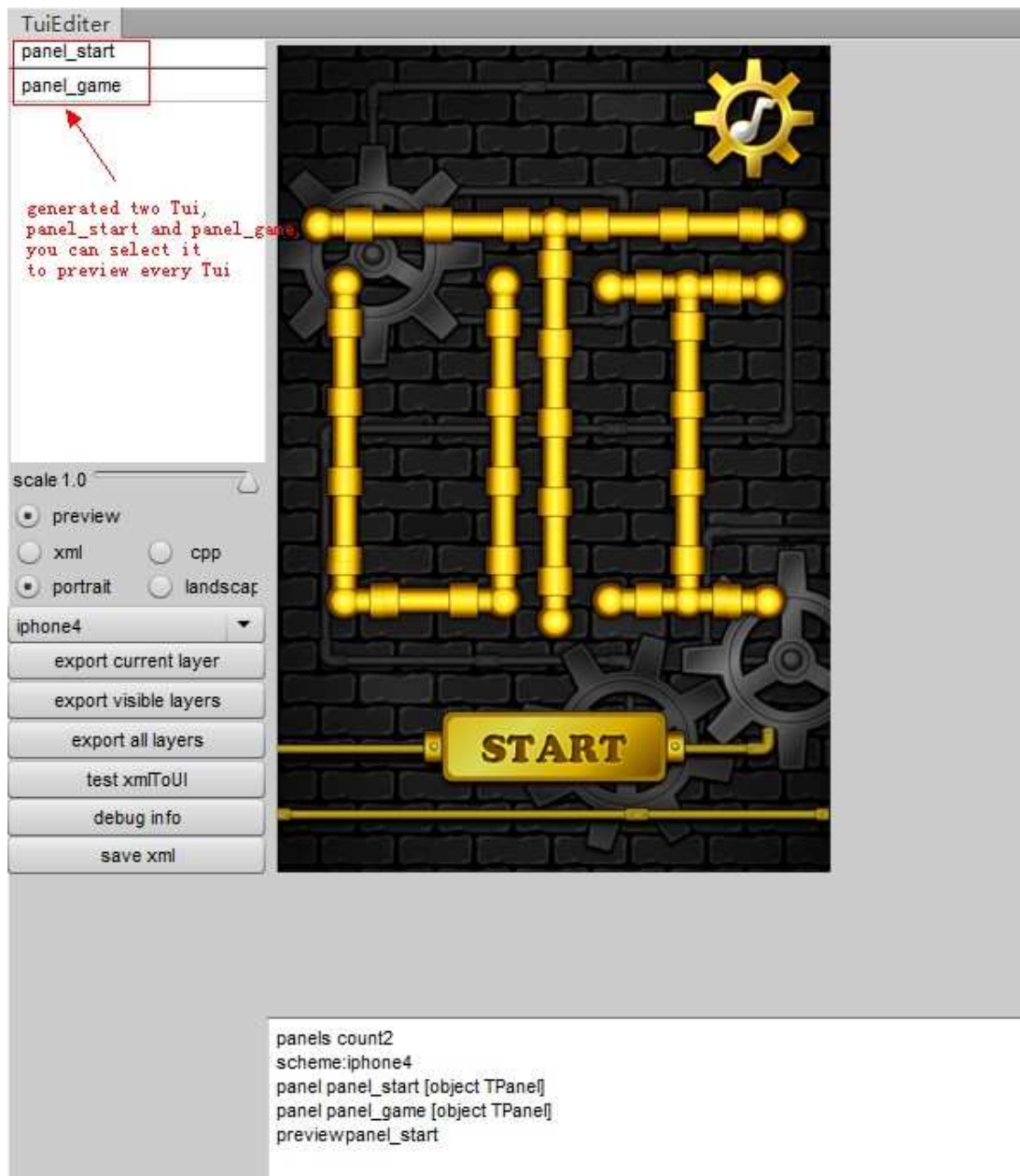
a) 我们先预览 iphone 的 UI,所以时间轴里的层选为 iphone 层,



b) 在插件面板里把相应的选项选为:

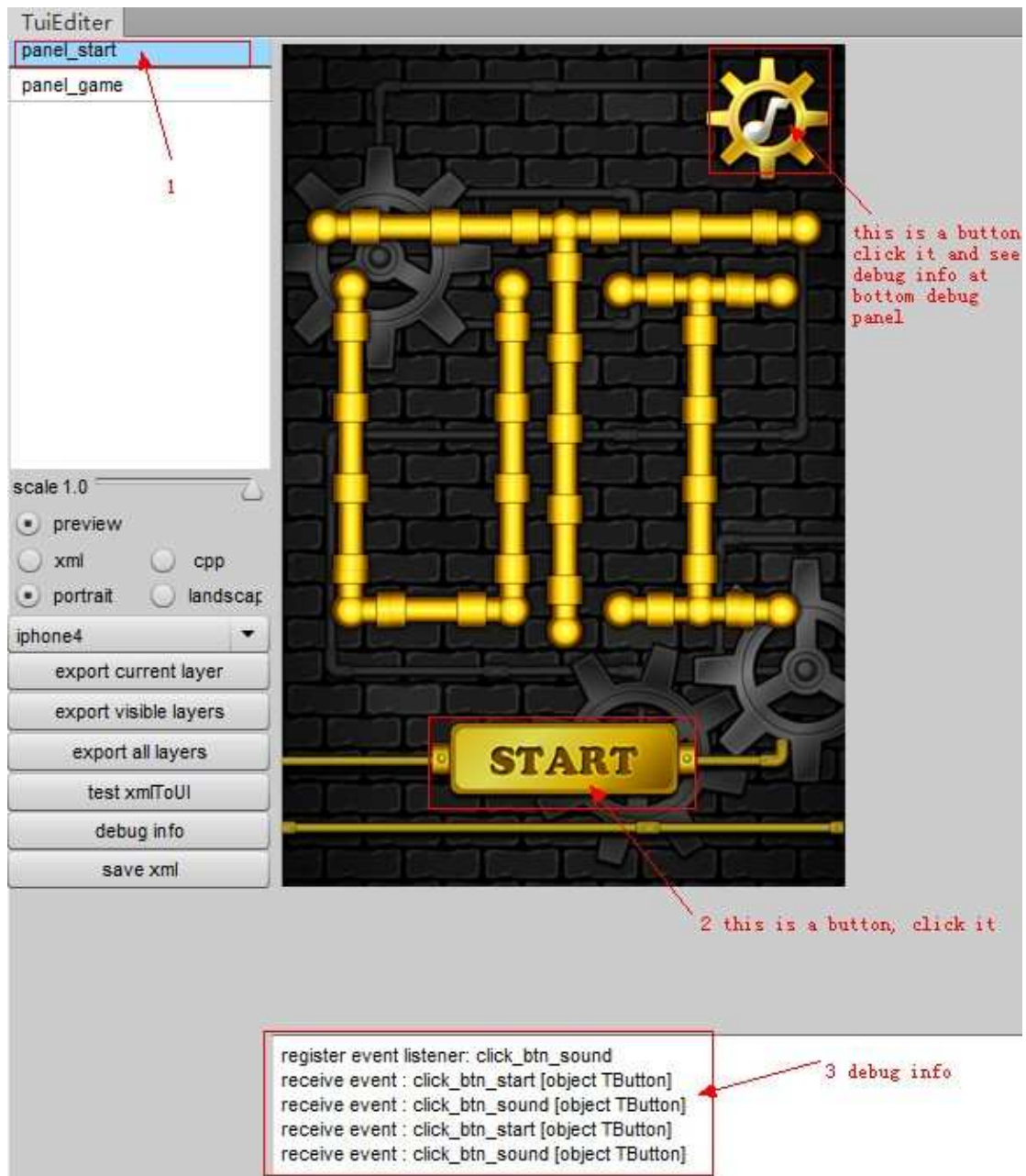


- c) 插件显示 UI 预览效果, 左上角的列表为 iphone 下一共有哪些 Tui, 在这里有 panel\_start 和 panel\_game 两个 UI, 可以点击每个列表选项预览各个 Tui

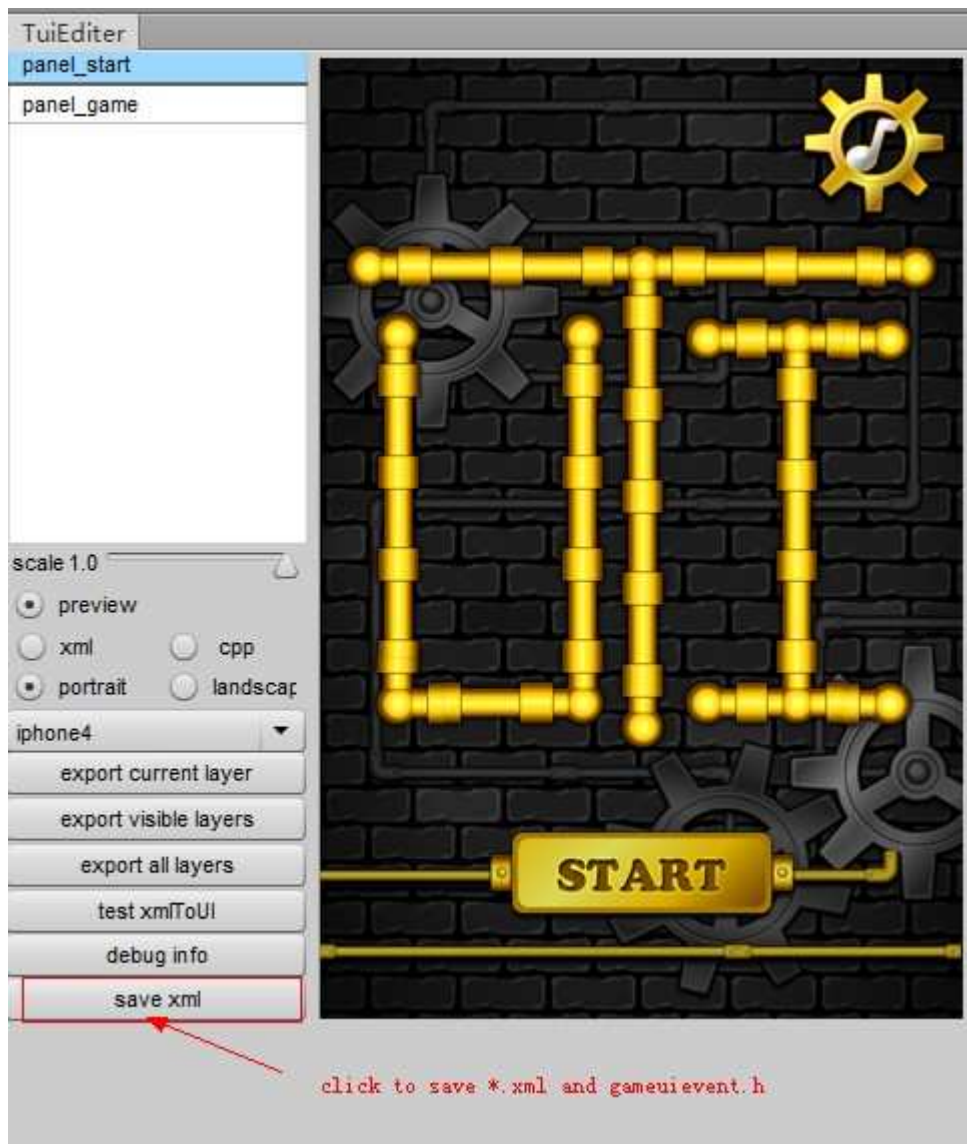


d) 接受按钮响应事件

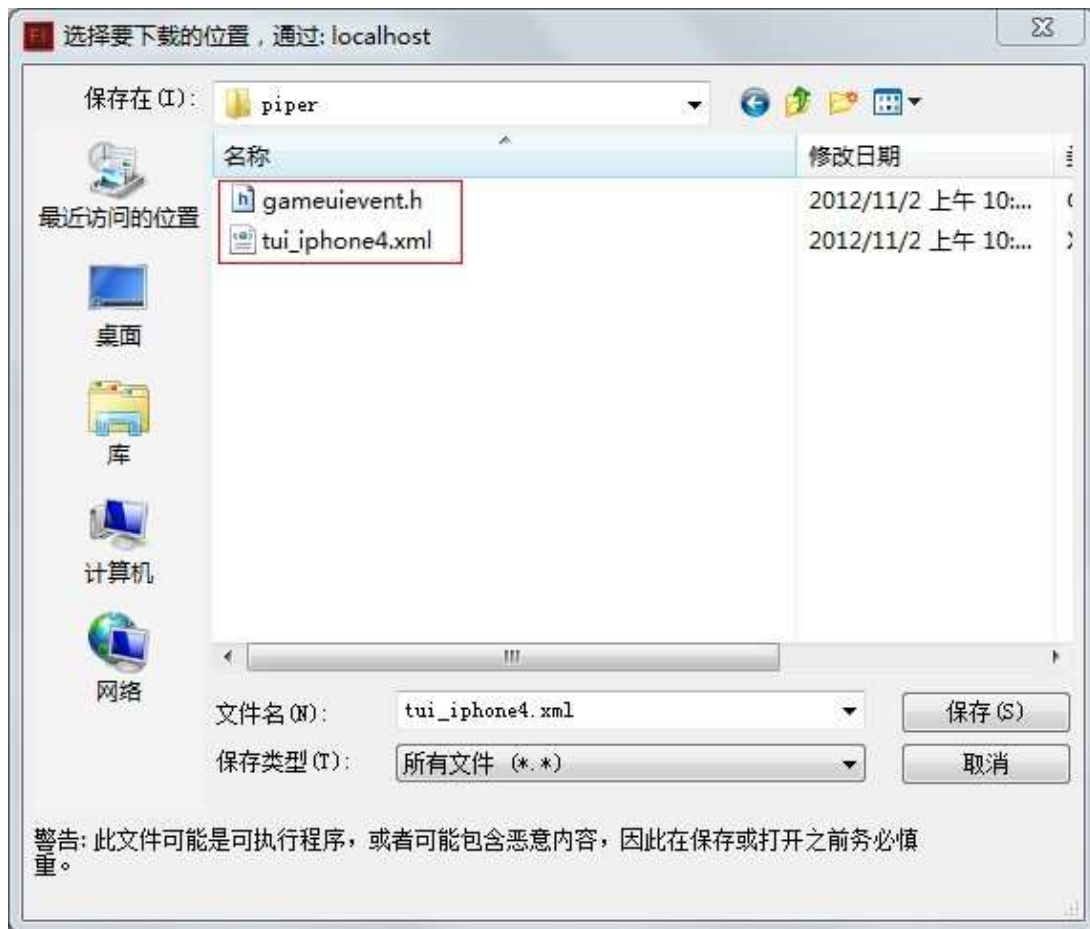
在列表里选择 panel\_start -> 点击 START, 会在 debug 面板显示出点击 start 按钮的 debug 信息



5. 生成 xml 文件和 gameuievent.h 文件



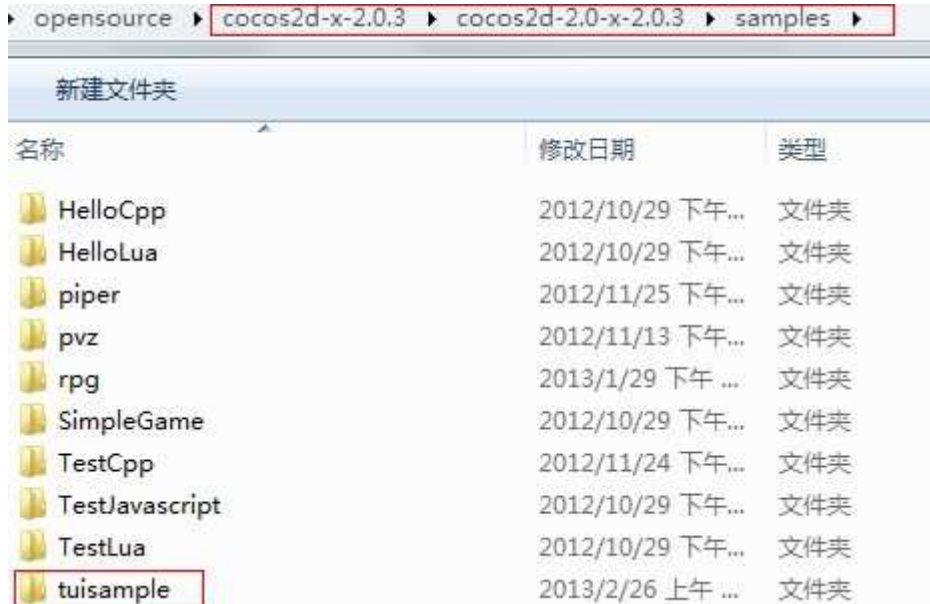




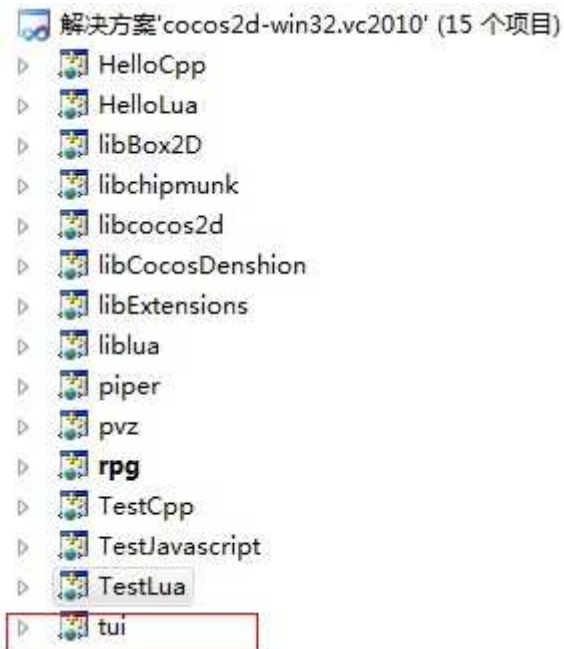


# 运行 Tui Sample

1. 进入目录 \$(TUITRUNK)/ cocos2dx-tui/samples, 复制 tuisample 到 \$(COCOS2D-X-2.0.3)/samples 目录下



2. 填加 tuisample 工程  
进入目录 cocos2d-2.0-x-2.0.3, 打开 cocos2d-win32.vc2010.sln 工程, 文件->填加->现有项目->\$(COCOS2D-X-2.0.3)/samples/tuisample/proj.win32/tui.vcproj



### 3. 运行 Tui 项目



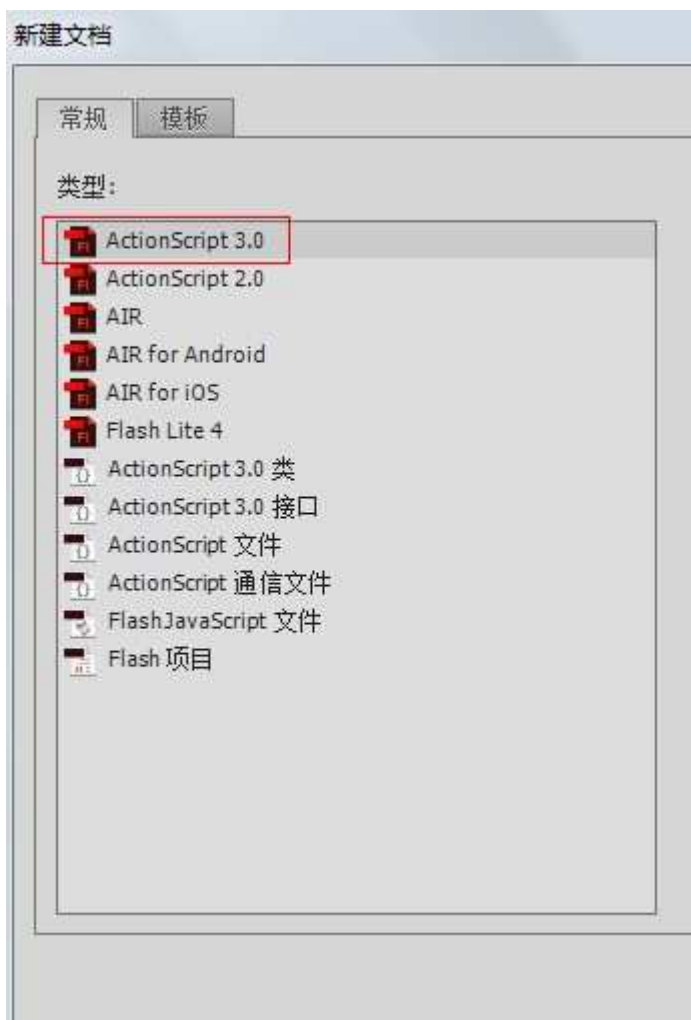
# 从 HelloTui 开始

下面我们新建一个 HelloTui 工程，从头到尾用 Tui 做一个游戏 UI 要求:

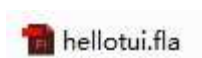
1. 包括两个 UI 界面，panel\_start, panel\_game
2. 要求每个 UI 上至少有一个按钮且两个 UI 可以来回切换
3. 支持 iphone 和 ipad，为了体现 Tui 对多方案的支持，要求 iphone 和 ipad 的 UI 排版不同

## 新建 HelloTui Flash CS 工程

- 打开 FlashCS6 -> 文件 -> 新建-> ActionScript3.0

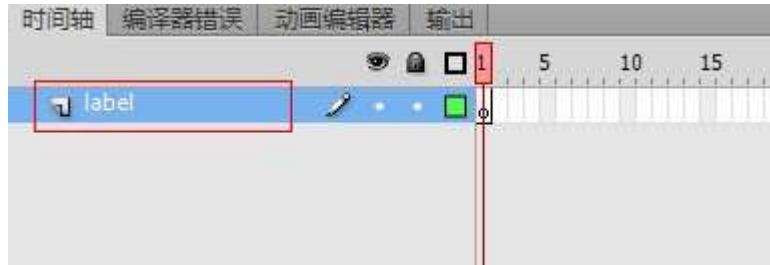


- 保存为 hellotui.flas



## 标签层

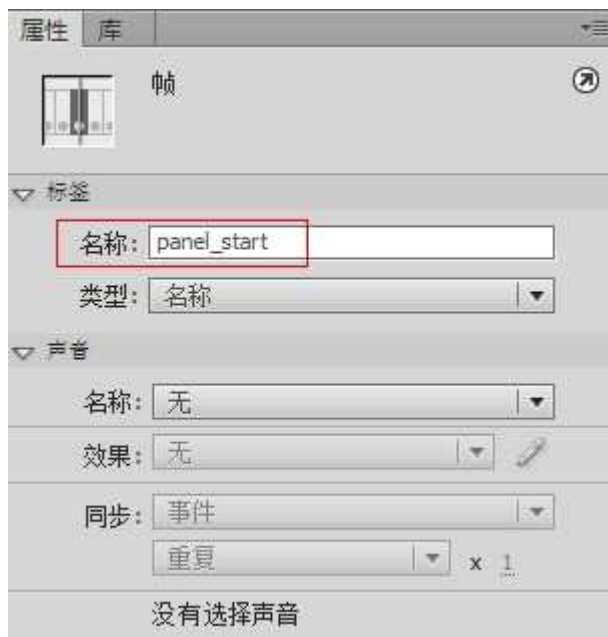
- 为了方便同事间传阅编辑 flash 文件,建议加入标签层,标记下某一帧的 ui 是什么 ui,不加也不影响程序,为方便编辑,建议每一帧只包含一个影片剪辑,一个影片剪辑就是一个 Tui 把默认的图层重命名为: label



- 选择第一帧



- 这一帧我们打算做 panel\_start 面板，所以把这一帧命名为 panel\_start

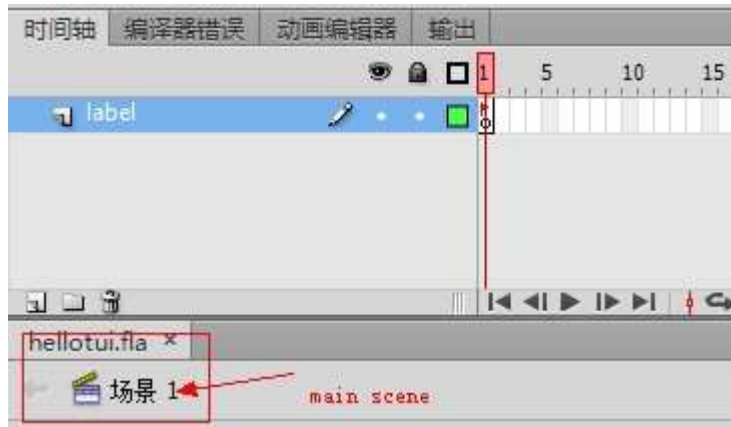


- 命名的帧被加上了红旗标志



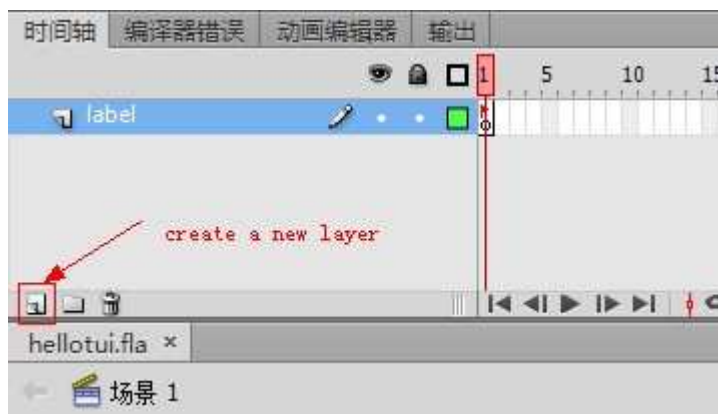
## 添加 iphone 分辨率

- Tui 在管理分辨率时按 Flash 主场景(下面图标出了什么是主场景)里时间轴的图导分, 每一层为一个分辨率, 该层里的每一帧包含一个 UI 面板



场景 1 就是主场景

- 新建一个层



- 重命名为 iphone, 并调整层的顺序, 把 label 放在最上面



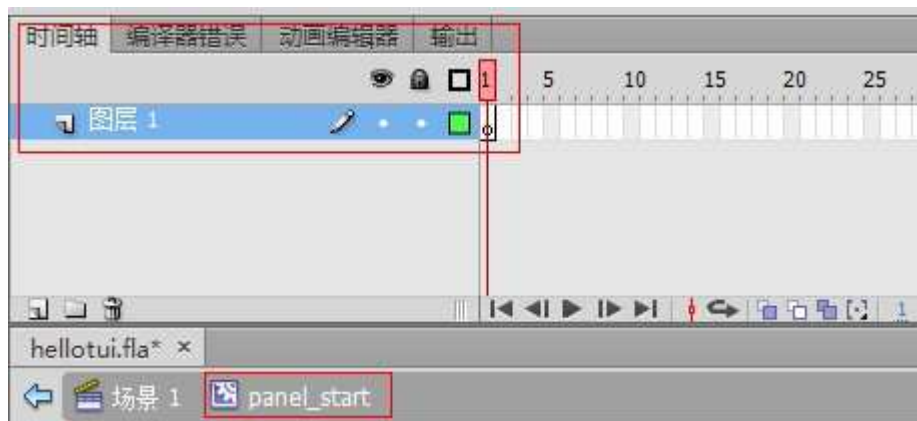
Ok,iphone 分辨率就建好了。

## 新建 iphone 版 panel\_start

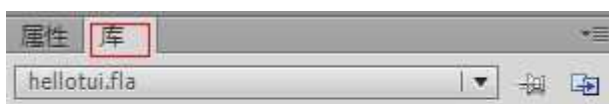
- 因为在 flash 里一个影片剪辑就是一个 Tui 面板，所以我们新建一个影片剪辑做 panel\_start 面板  
按 ctrl + f8 打开新建元件面板，填好参数后，点击确定



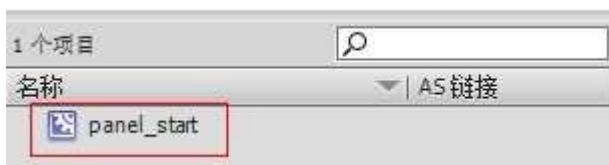
- 点击确定后 flash 会进入 panel\_start 的编辑界面，注意场景的层次里多了 panel\_start 层，时间轴也变成了 panel\_start 的时间轴，我们可以在这个 panel\_start 的时间轴里加入多个图层，图片，影片剪辑或者按钮



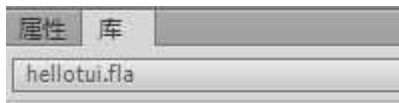
- 可以看一下库里多了 panel\_start 的影片剪辑，如果看不到库可以按 ctrl+L 打开库面板



+



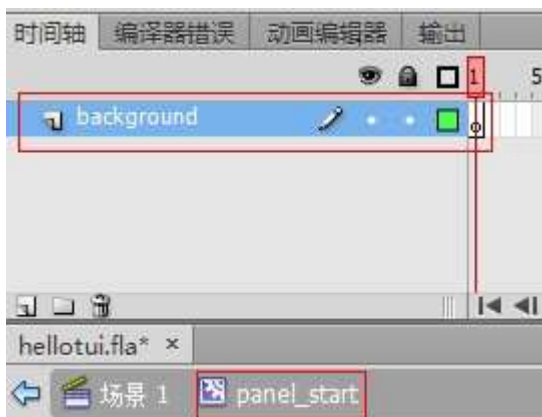
- 为了方便管理,我们在库里新建一个 `iphone` 的目录专门存放 `iphone` 的元件, 然后把 `panel_start` 放到 `iphone/panel` 下面, 目录结构为:



预览不可用



- 在库里双击 `panel_start` 进入到 `panel_start` 的时间轴, 填加一个背景层, 这层我们放一张背景图



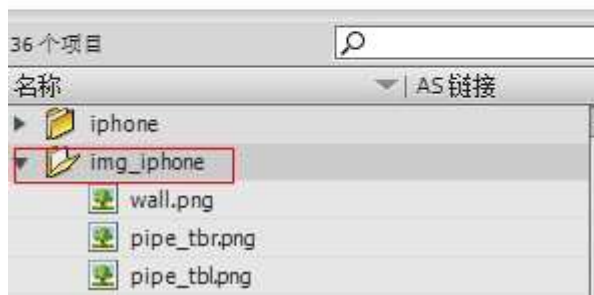
- 准备 `iphone` 的图片资源, 这里我们直接把 `tuisample` 里的资源复制到同 `hellotui.fla` 目录下



- 把 res\_iphone 资源导入到 flash 库里,目录结构为,注意这里的目录名必须为 **img\_iphone**,插件在生成 xml 时会在路径里固定去掉 **img\_iphone**, **img\_ipad** 目录层

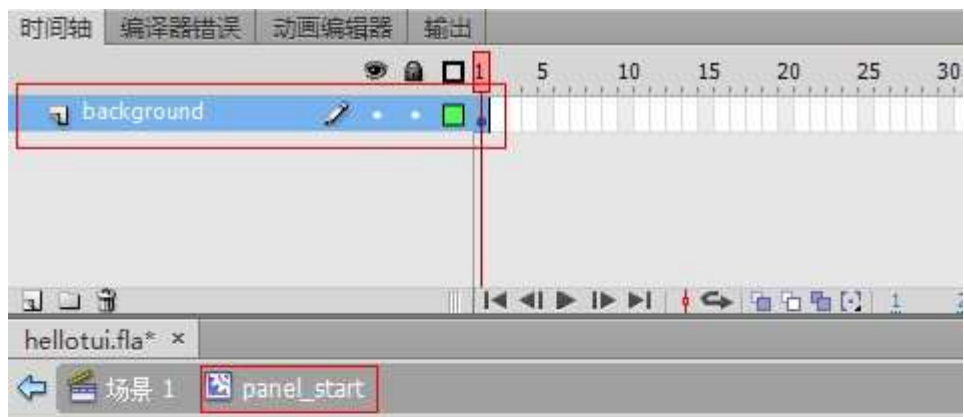


预览不可用





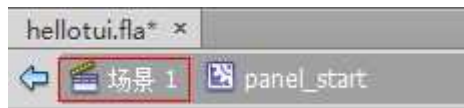
- 把 menu\_background.png 托到 panel\_start 影片剪辑的 background 图层



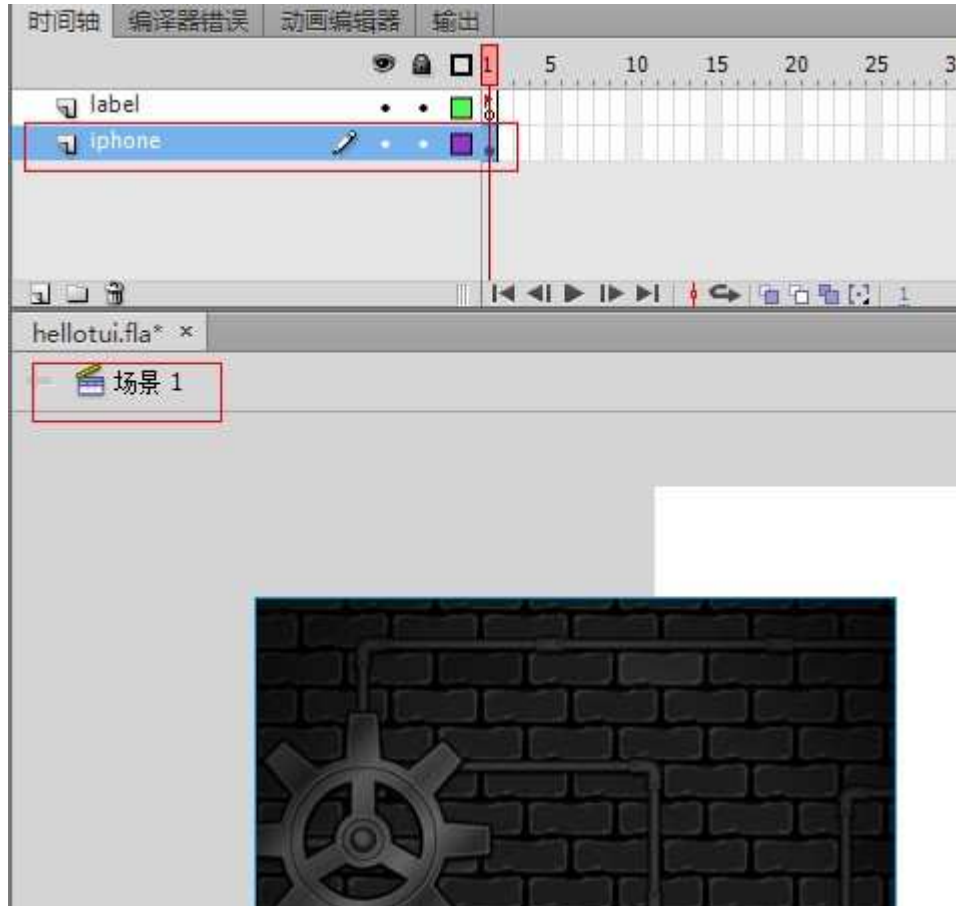
- 把图片居中



- 点击场景 1，回到主场景



- 把 panel\_iphone 面板填加到主场景的 iphone 层，第 1 帧



- 左上对齐



- 这时在主场景里的面板为 `panel_start` 的一个实例。举个例子：库里的 `panel_start` 就是 `c++` 里的类，主场景里的 `panel_start` 是一个对象，因为是一个对象所以我们需要给对象命名，我们把这个实例还命名为 `panel_start`，叫什么名都行，只不过我们在这里为了识别方便。这个名字用于程序里访问面板。

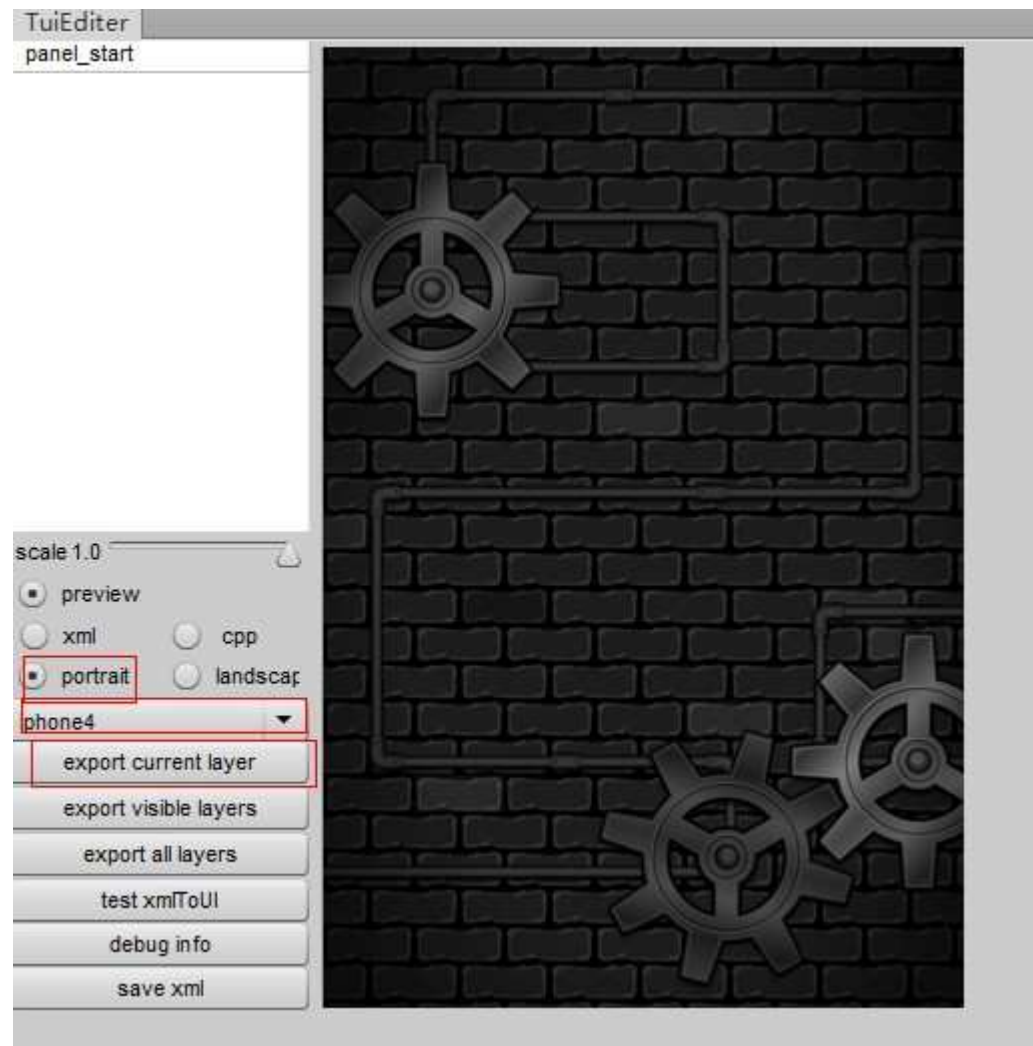
选中刚刚放在主场景里面的 `panel_start` 面板，在属性面板里为实例命名为 `panel_start`



- 在插件里预览效果  
点击菜单：窗口->其他窗口->TuiEditor 打开插件  
确定场景为主场景，时间轴为 `iphone` 层



在插件中选择 scheme 为 iphone4, 屏幕为 portrait, 然后点击 export current layer, 插件显示如下图:



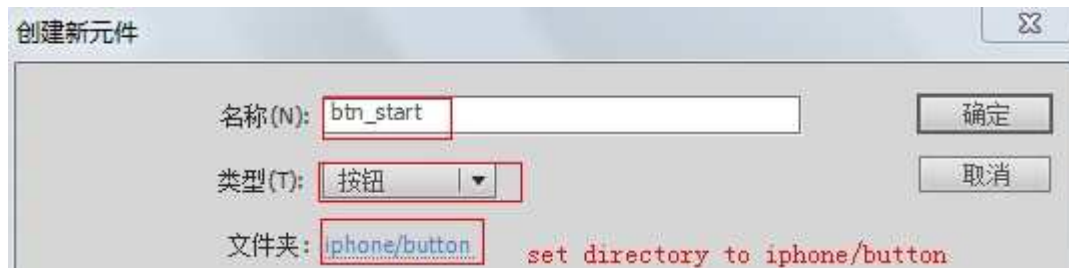
- **Ok, 至此, 我们有了第一个 UI, 需要特别注意的地方有:**
  - **创建 Tui 面板用影片剪辑, 在影片剪辑里放图片时需要居中对齐**
  - **在主场景里创建面板实例时需要左上对齐**
  - **面板实例一定要命名**

## 添加一个按钮

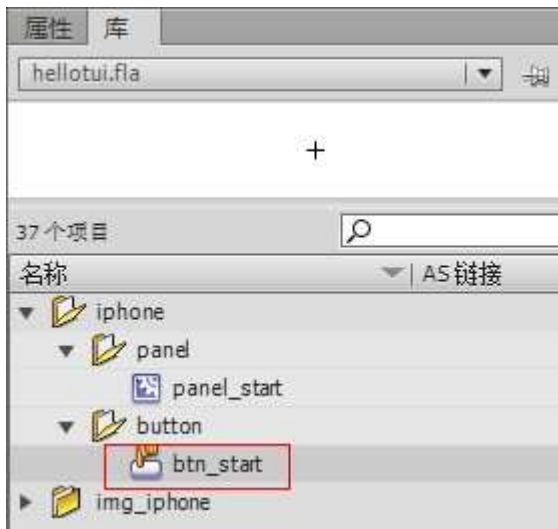
- 按钮的图片格式，Tui 的按钮图片分为四个状态：正常，高亮，按下，禁用，这里延续端游的做法是为了方便手游再移植为端游。



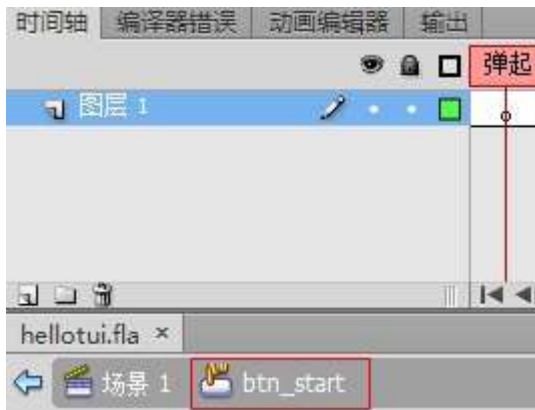
- 在 flash 里创建一个按钮元件  
按 CTRL + F8 打开新建元件面板，名称为 btn\_start，类型为按钮，文件夹设为 iphone/button:



- 在库里看到 btn\_start 如下图:



- 在库面板里双击 btn\_start 进入编辑状态



- 为按钮加上图片

在库里的 img\_iphone 里找到 menu\_start.png 并托到场景里(找 menu\_start.png 的时候一定会看到里面还有一个 cn\_menu\_start.png, 这个是国际化时候用的, 后面会讲到), 如下图:



+

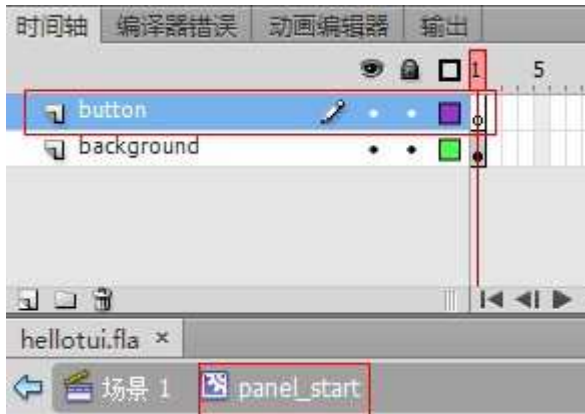
- 居中并横向缩放为 25%, 因为原图为正常按钮的 4 倍大, 所以这里需要缩为 1/4



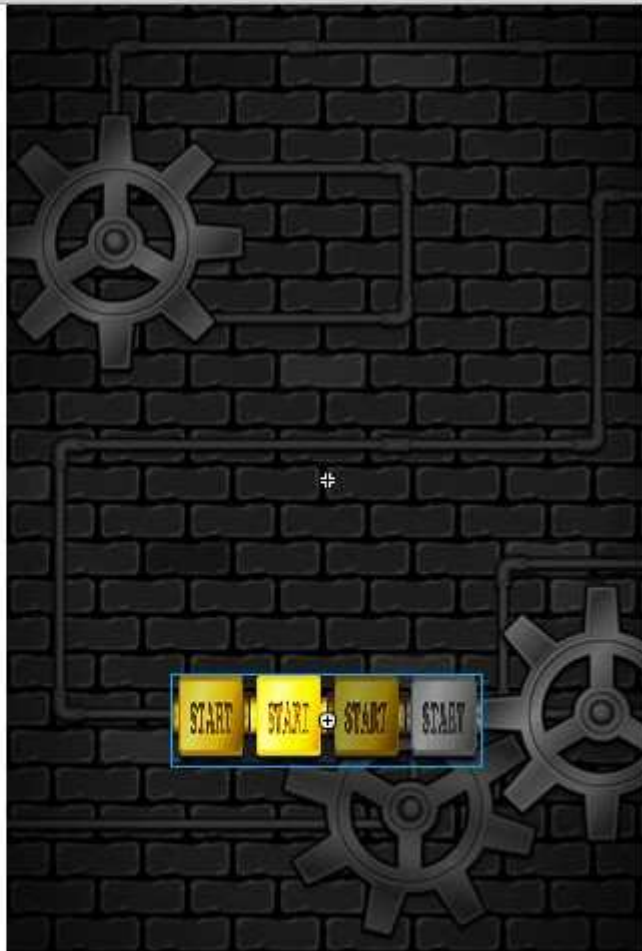
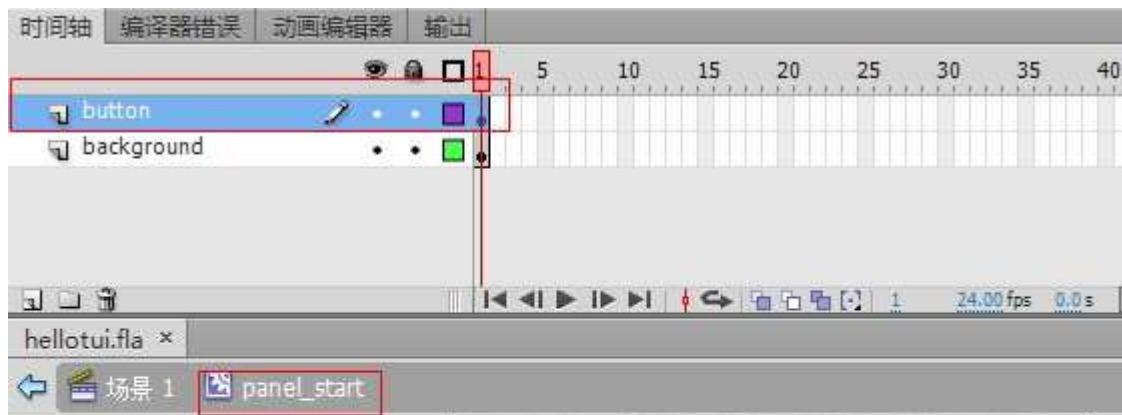
- 一个按钮做好了，看看效果图



- 为 panel\_start 填加一个按钮，在库面板里双击 panel\_start 影片剪辑进入编辑状态，并新建一个层，命名为 button:



- 选中 button 层第一帧，在库面板里把 btn\_start 按钮托到 panel\_start 里，放置如图：

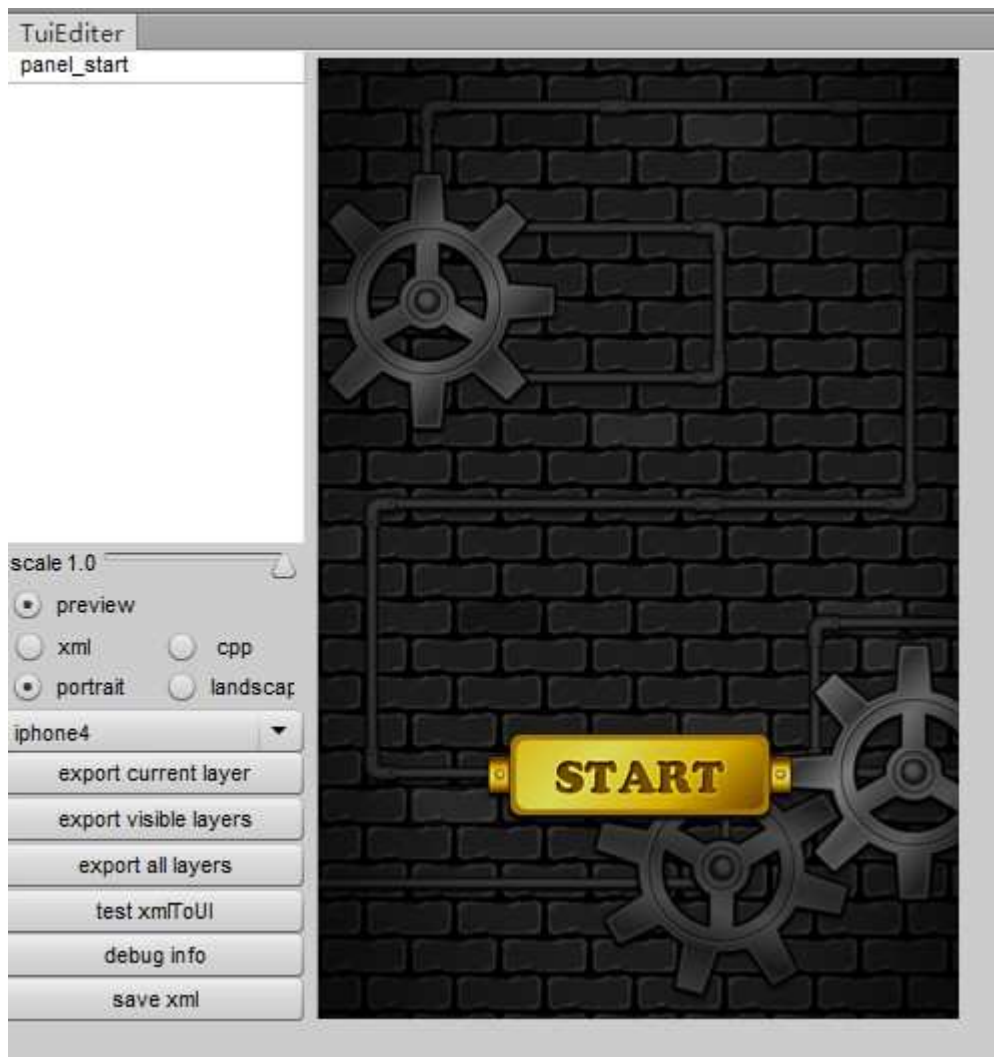


- 同样的为刚刚填充到 panel\_start 里的按钮实例命名为 btn\_start,名字可以随便取,该名字用于程序中取得该按钮,点击该按钮触发的事件为面板名+click+按钮名:

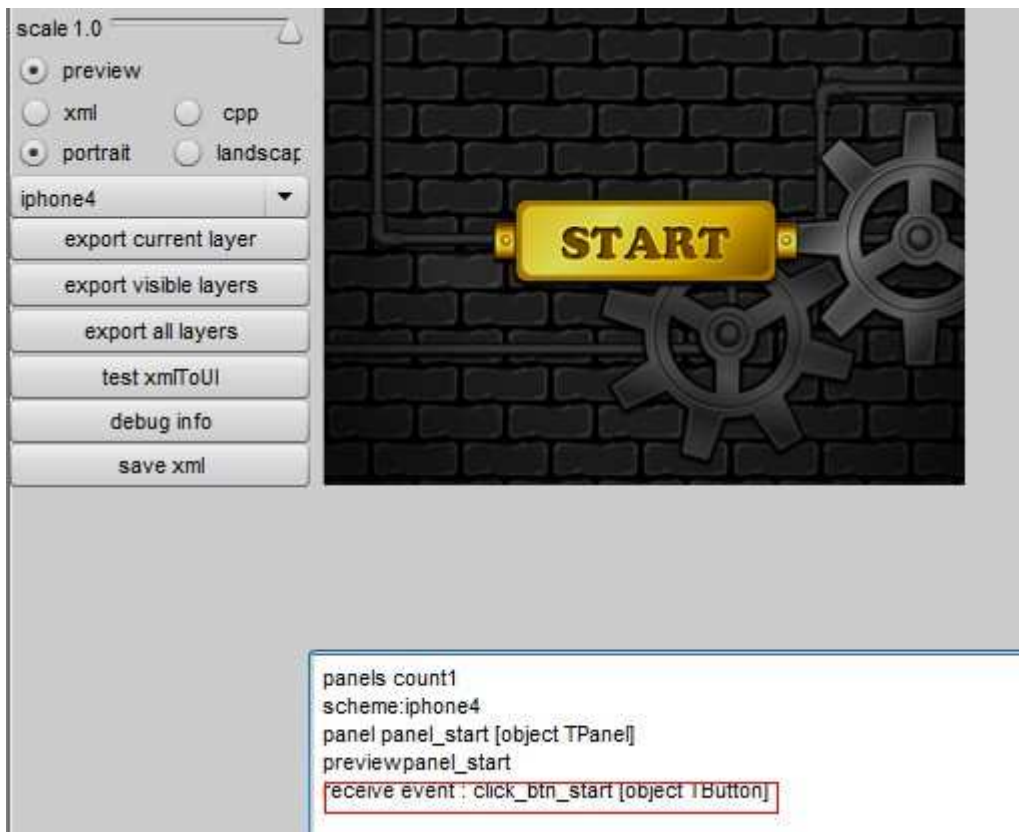




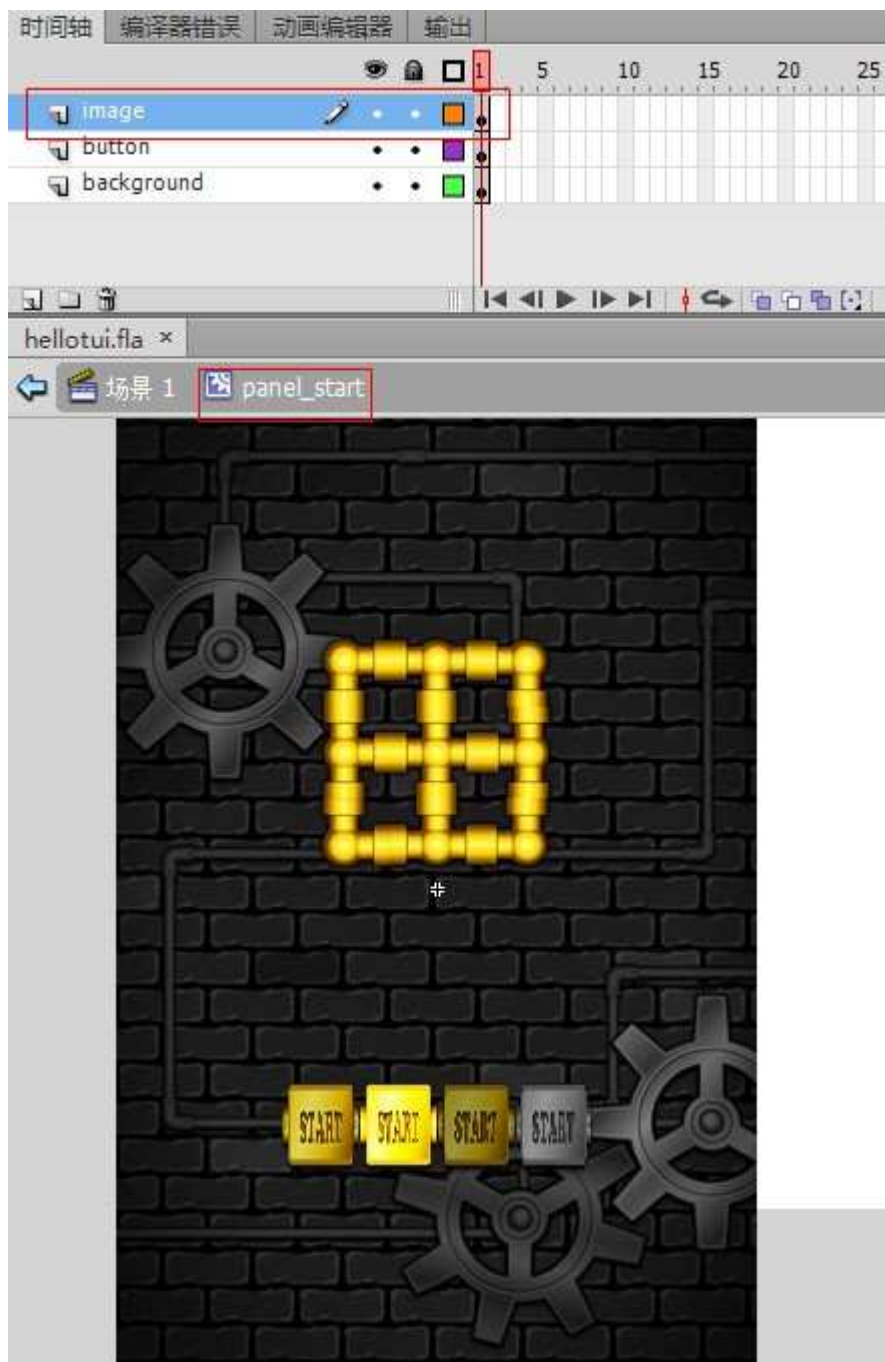
- 预览按钮，一定要回到[主场景]后再用插件导出 UI，否则预览不正确



- 点击按钮可以会打印出 debug 信息: receive event : click\_btn\_start [object TButton]



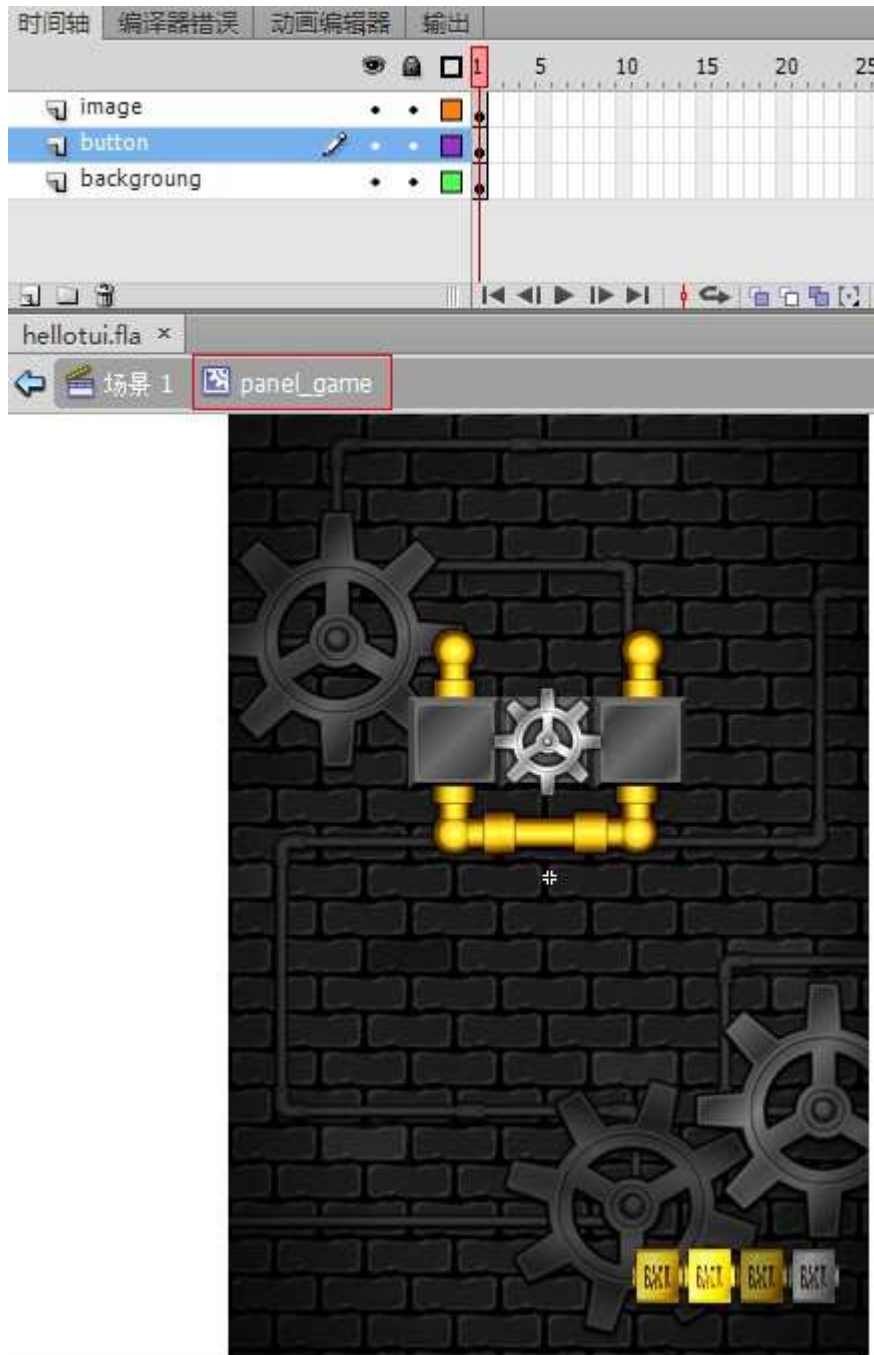
- OK, 为 panel\_start 加了一个 start 按钮, 看上去单调些哈, 下面为 panel\_start 加些图片点缀一下吧
- 在 panel\_start 的时间轴里新加一层, 命名为 image, 选中 image 层, 托几张图片进来, 效果如下:



。。。看上去依然很单调，亲，时间有限，意思意思哈~  
OK, panel\_start 面板就全部完工了，下面会同样的方法再新建一个 panel\_game 面板。

## 新建 iphone 版 panel\_game

- 按 CTRL+F8 创建 panel\_game 影片剪辑，这里我就不一步一步的建 panel\_game 面板，遇到问题请参考建 panel\_start 的步骤，最后的影片剪辑的效果为，填加的按钮为 menu\_back.png:

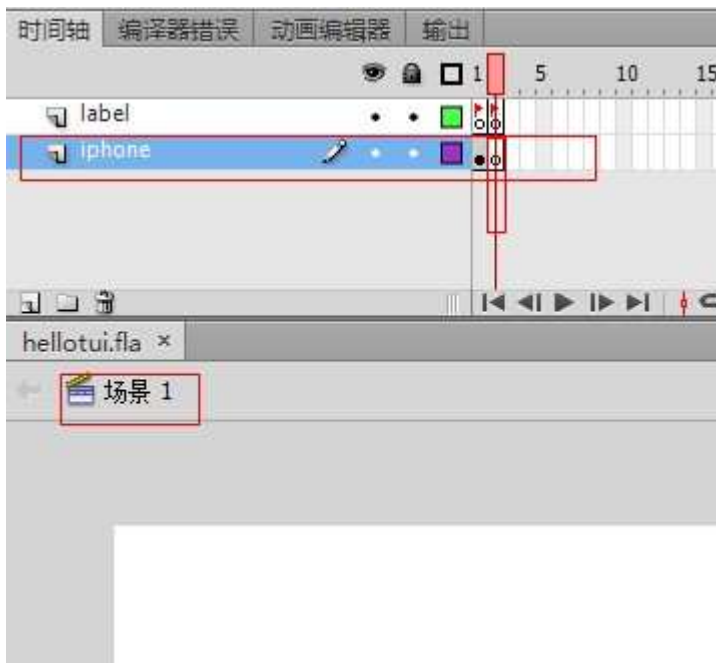


注意: 背景图为居中对齐, 放到 panel\_game 里面的按钮实例要命名

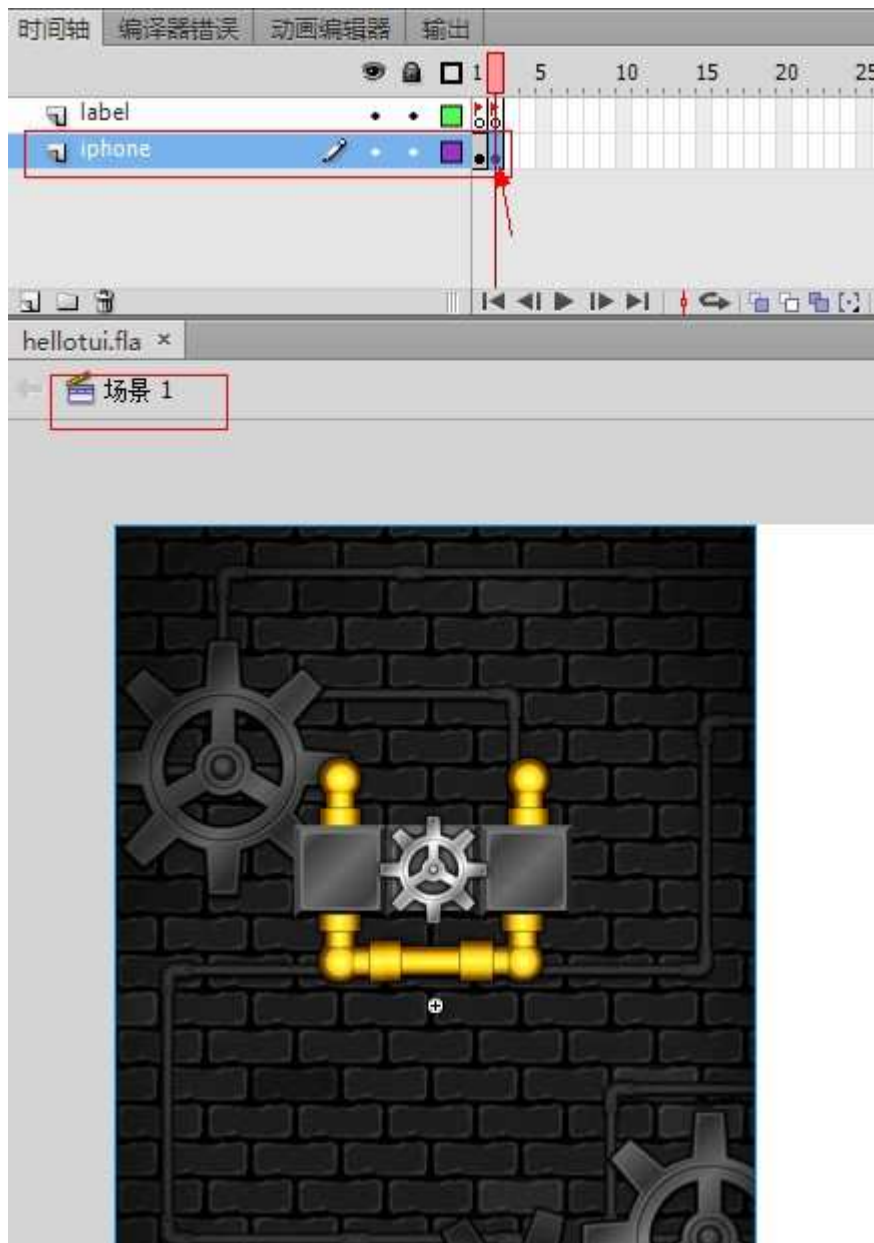
- 选中主场景时间轴的标签层的第 2 帧按 F6 填加一个关键帧，命名为 panel\_game



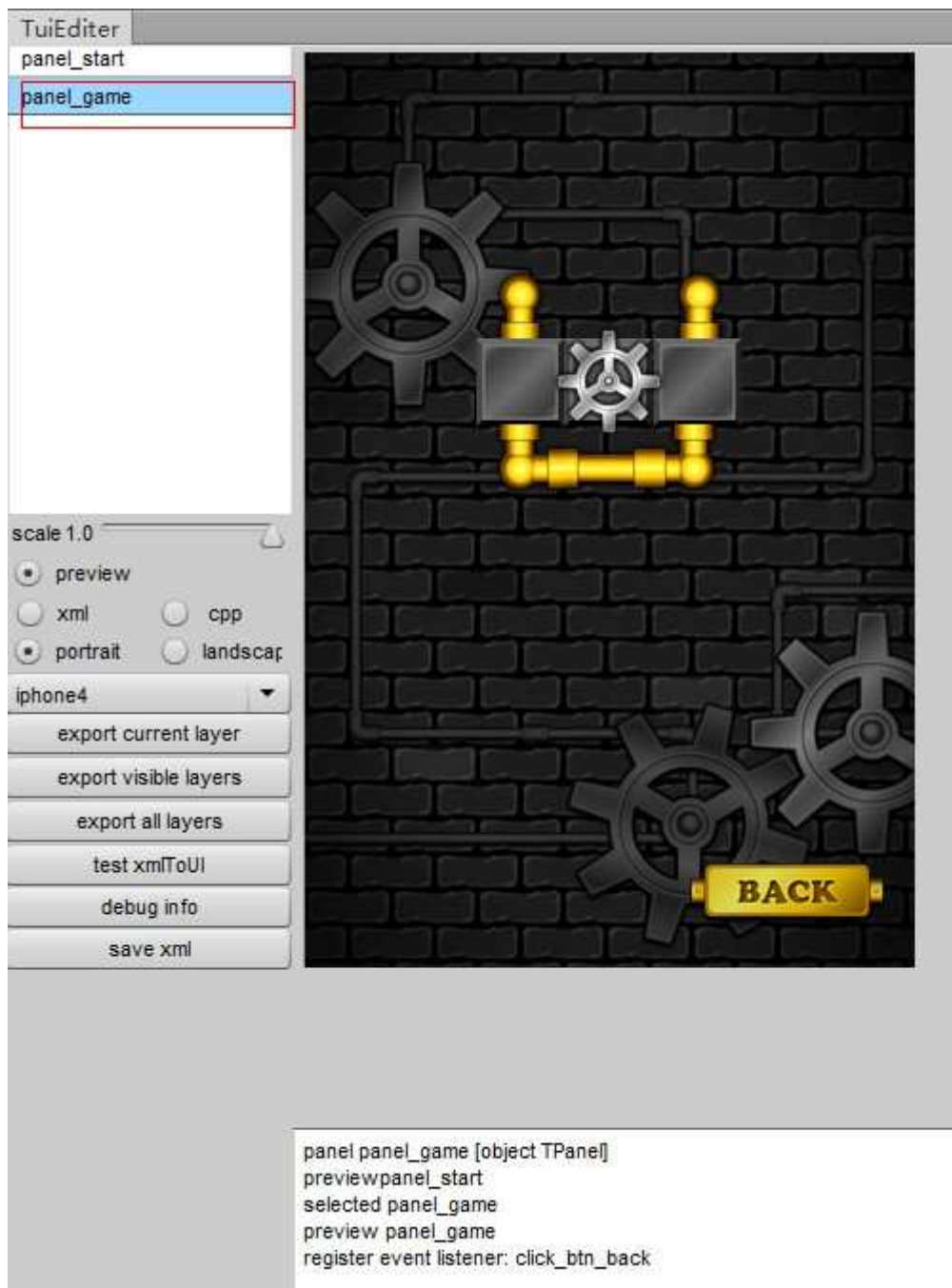
- 在主场景的 iphone 层第二帧按 F6 填加一个关键帧，并把第二帧里的面板去掉，使第二帧为空心圆



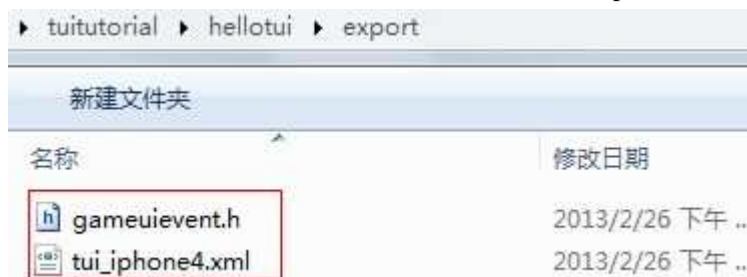
- 把 panel\_game 影片剪辑托到主场景，创建 panel\_game 的实例，并把该实例命名为 panel\_game



- 在插件里预览



- OK,两个面板都建好了，点击 save xml，生成 tui\_iphone4.xml 和 gameuievent.h 文件

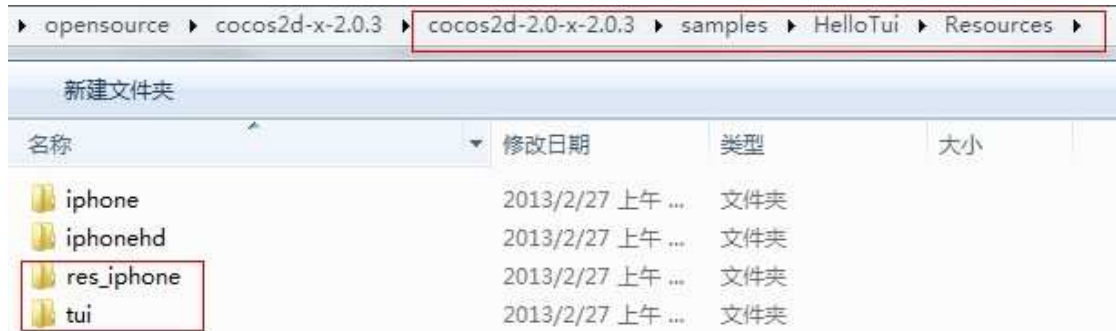


- Flash 工作就完成了，后面我们看如何在程序中使用这两个文件

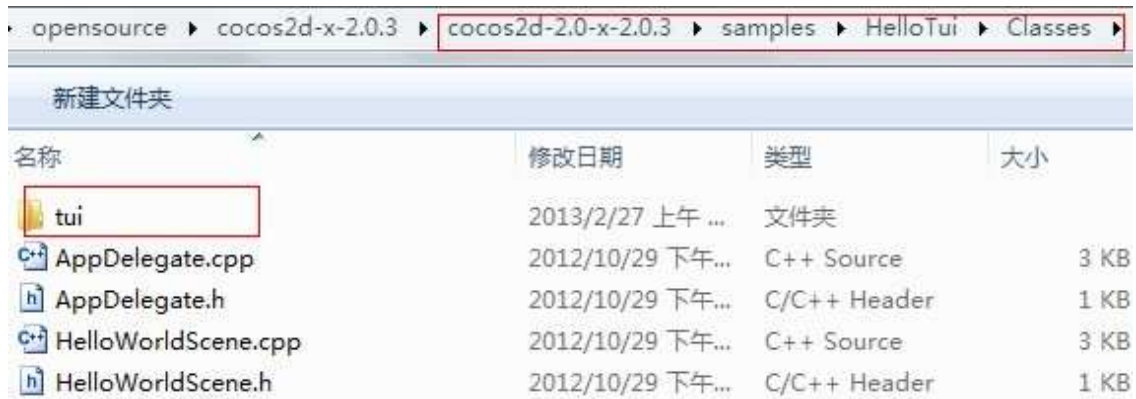


## 把 HelloTui 导入程序

- 我们在 cocos2d-x 的 HelloCpp 的基础上改为 HelloTui，这个过程就不用说了吧。
- 在 HelloTui 工程的 Resources 下建个 tui 目录，把刚刚插件生成的 tui\_iphone4.xml 文件复制到 Resources/tui 目录下
- 把 res\_iphone 复制到 Resources 下面

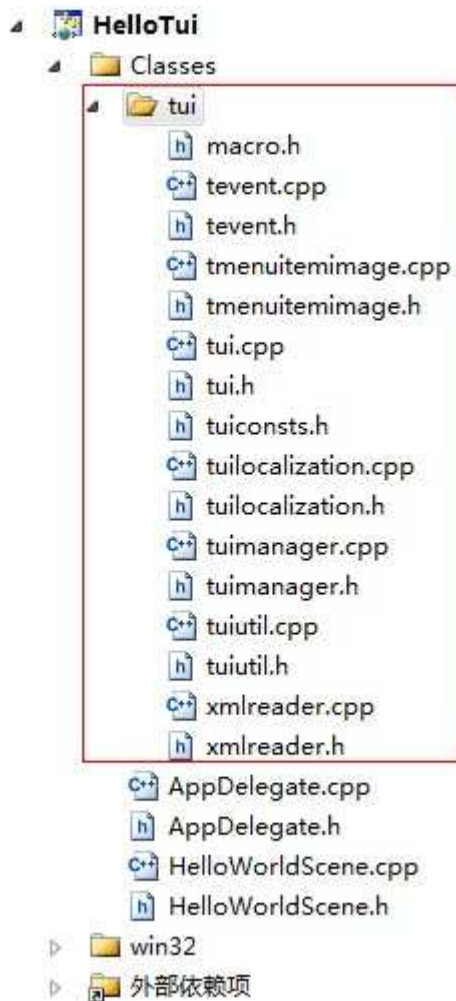


- 把\$(TUITRUNK) /cocos2dx-tui/tui 复制到 HelloTui/Classes 目录下

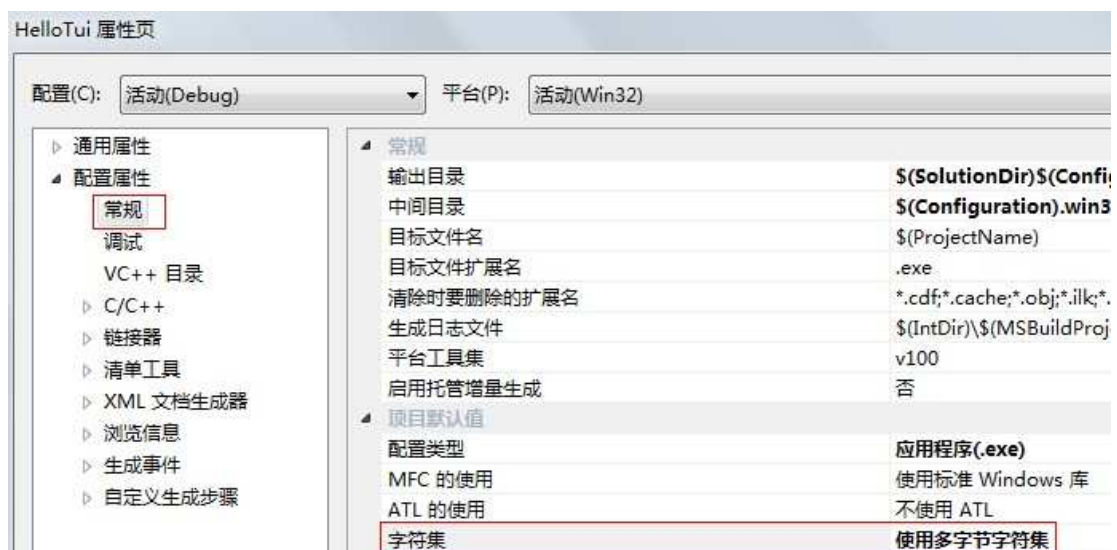




- 在工程里的 Classes 下添加 tui 目录，并把 tui/下的源码加进来



- ALT+F7 打开工程属性面板,设置字符集为: 使用多字节字符集



- 链接 tinyxml 库，这里为了方便我直接把 tinyxml 源文件放到工程里用，哈，偷个懒~  
新建目录 HelloTui/Classes/tinyxml,把 tinyxml 里的源码除了 xmltest.cpp 以外，全部复制到该目录下

tuutorial > hellotui > tinyxml\_2\_6\_2 > tinyxml >

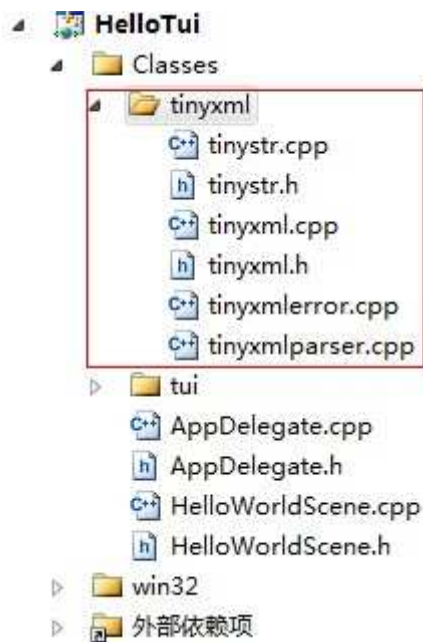
名称	修改日期	类型	大小
docs	2011/5/14 下午 ...	文件夹	
tinyst.h	2011/5/14 下午 ...	C/C++ Header	9 KB
tinyxml.h	2011/5/14 下午 ...	C/C++ Header	64 KB
tinyst.cpp	2011/5/14 下午 ...	C++ Source	3 KB
tinyxml.cpp	2011/5/14 下午 ...	C++ Source	37 KB
tinyxmlerror.cpp	2011/5/14 下午 ...	C++ Source	2 KB
tinyxmlparser.cpp	2011/5/14 下午 ...	C++ Source	37 KB
xmltest.cpp	2011/5/14 下午 ...	C++ Source	42 KB
utf8test.gif	2011/5/14 下午 ...	GIF 图像	15 KB
tinyxml.sln	2011/5/14 下午 ...	Microsoft Visual...	3 KB

opensource > cocos2d-x-2.0.3 > cocos2d-2.0-x-2.0.3 > samples > HelloTui > Classes > tinyxml

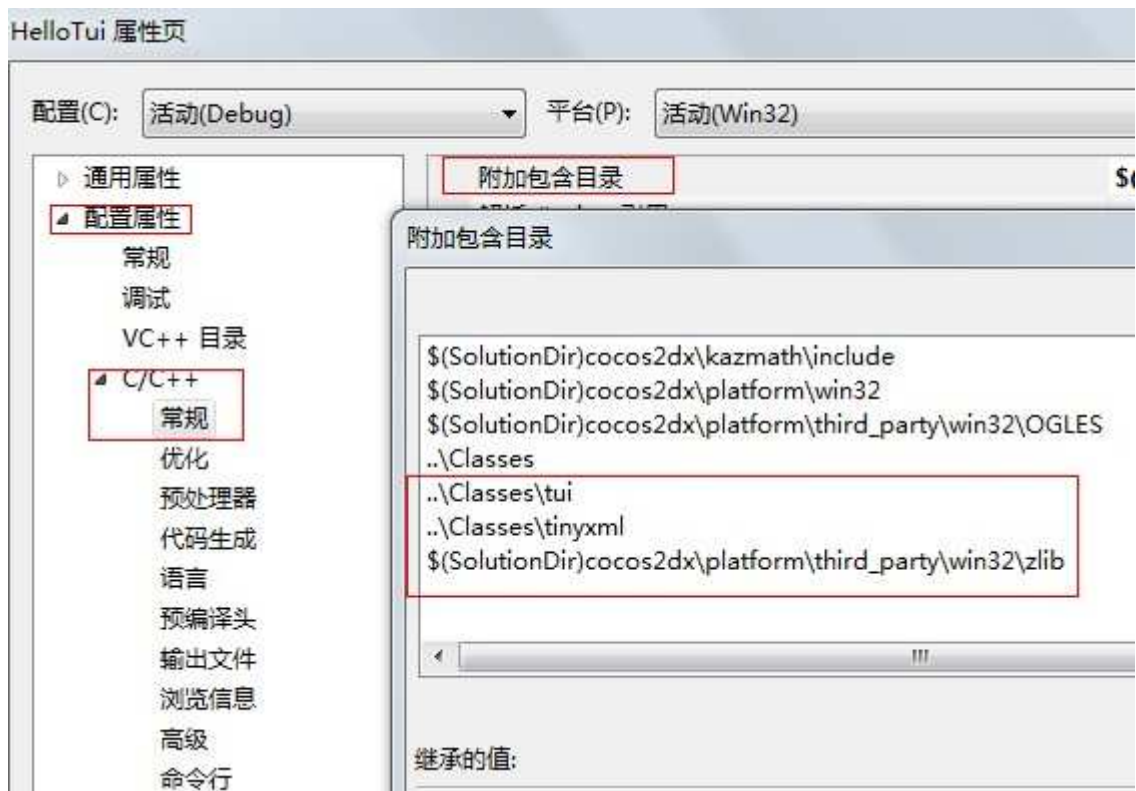
新建文件夹

名称	修改日期	类型	大小
tinyst.cpp	2011/5/14 下午 ...	C++ Source	3 KB
tinyst.h	2011/5/14 下午 ...	C/C++ Header	9 KB
tinyxml.cpp	2011/5/14 下午 ...	C++ Source	37 KB
tinyxml.h	2011/5/14 下午 ...	C/C++ Header	64 KB
tinyxmlerror.cpp	2011/5/14 下午 ...	C++ Source	2 KB
tinyxmlparser.cpp	2011/5/14 下午 ...	C++ Source	37 KB

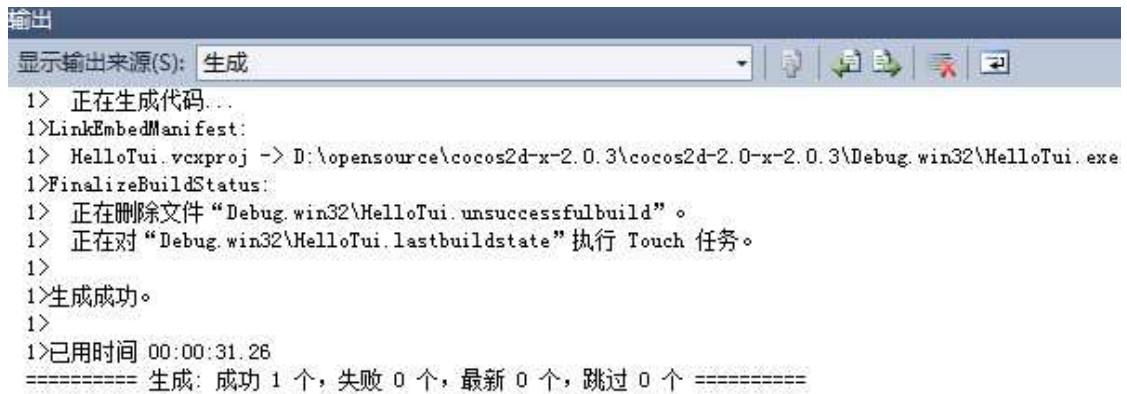
- 工程里添加 tinyxml 源码



- HelloTui 工程里添加相关头文件目录:
  - ..\Classes\tui
  - ..\Classes\tinyxml
  - \$(SolutionDir)cocos2dx\platform\third\_party\win32\zlib

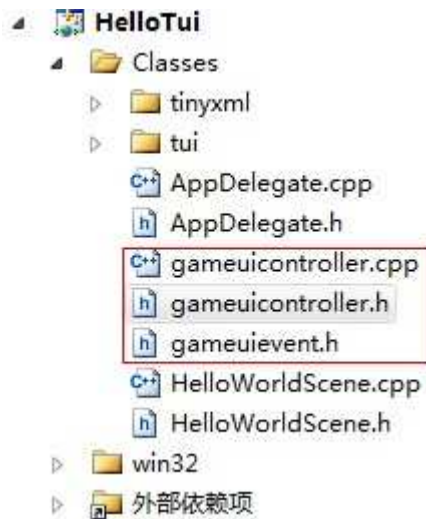


- 编译工程可以通过



```
输出
显示输出来源(S): 生成
1> 正在生成代码...
1>LinkEmbedManifest:
1> HelloTui.vcxproj -> D:\opensource\cocos2d-x-2.0.3\cocos2d-2.0-x-2.0.3\Debug.win32\HelloTui.exe
1>FinalizeBuildStatus:
1> 正在删除文件“Debug.win32\HelloTui.unsuccessfulbuild” 。
1> 正在对“Debug.win32\HelloTui.lastbuildstate”执行 Touch 任务。
1>
1>生成成功。
1>
1>已用时间 00:00:31.26
====生成: 成功 1 个, 失败 0 个, 最新 0 个, 跳过 0 个 =====
```

- 复制\$(TUITRUNK)/ cocos2dx-tui/samples/tuisample/Classes 下的 gameuicontroller.cpp 和 gameuicontroller.h 文件到 HelloCpp/Classes 下面，并加入到工程
- 把插件生成的 gameuievent.h 复制到 HelloCpp/Classes 下面，并加入到工程



- 去掉 gameuicontroller 里没用的事件处理方法，变成下面的样子

```

10  */
11
12  class GameUIController : public TuiManager
13  {
14      GameUIController();
15  public:
16      static GameUIController* I();
17      virtual ~GameUIController();
18
19  public:
20
21  private:
22      static GameUIController* s_instance;
23  };
24
25  #endif//GAMEUICONTROLLER_H
26

```

- 把gameuevent.h里的函数全部填加在gameuicontroller里，格式如下

```

class GameUIController : public TuiManager{
public:
    void on_event_panel_start_click_btn_start( void* o, TEvent* e );
    void on_event_panel_game_click_btn_back( void* o, TEvent* e );
};

```

当 btn\_start 被按下时，会调用 on\_event\_panel\_start\_click\_btn\_start 函数

当 btn\_back 被按下时，会调用 on\_event\_panel\_game\_click\_btn\_back 函数

- 我们暂时先把这两个函数写为空函数，什么也不处理，先加载 tui\_iphone4 的配置文件

修改 AppDelegate.cpp

加上头文件:

```
#include <gameuicontroller.h>
```

修改 AppDelegate::applicationDidFinishLaunching，红色为 Tui 相关代码:

```

bool AppDelegate::applicationDidFinishLaunching() {
    // initialize director
    CCDirector *pDirector = CCDirector::sharedDirector();
    pDirector->setOpenGLView(CCEGLView::sharedOpenGLView());

```

```
GameUIController* uic = GameUIController::I();
```

```

    uic->retain();

    TargetPlatform target = getTargetPlatform();
    if (target == kTargetIpad){
        //暂略
    }
    else if (target == kTargetIphone){
        //暂略
    }else {
        // android, windows, blackberry, linux or mac
        // use 960*640 resources as design resolution size
        uic->loadTuiXML( "tui/tui_iphone4.xml" );
        CCFileUtils::sharedFileUtils()->setResourceDirectory("res_iphone");
        int w = uic->schemeWidth();
        int h = uic->schemeHeight();
        CCEGLView::sharedOpenGLView()->setDesignResolutionSize(w, h,
kResolutionShowAll);
    }

    pDirector->setDisplayStats(true);
    pDirector->setAnimationInterval(1.0 / 60);
    CCScene *pScene = HelloWorld::scene();
    pDirector->runWithScene(pScene);
    return true;
}

```

- 修改 HelloWorldScene.cpp

填加头文件:

```

#include <gameuicontroller.h>
#include <gameuievent.h>

```

修改 HelloWorld::init, 红色为 Tui 相关代码

```

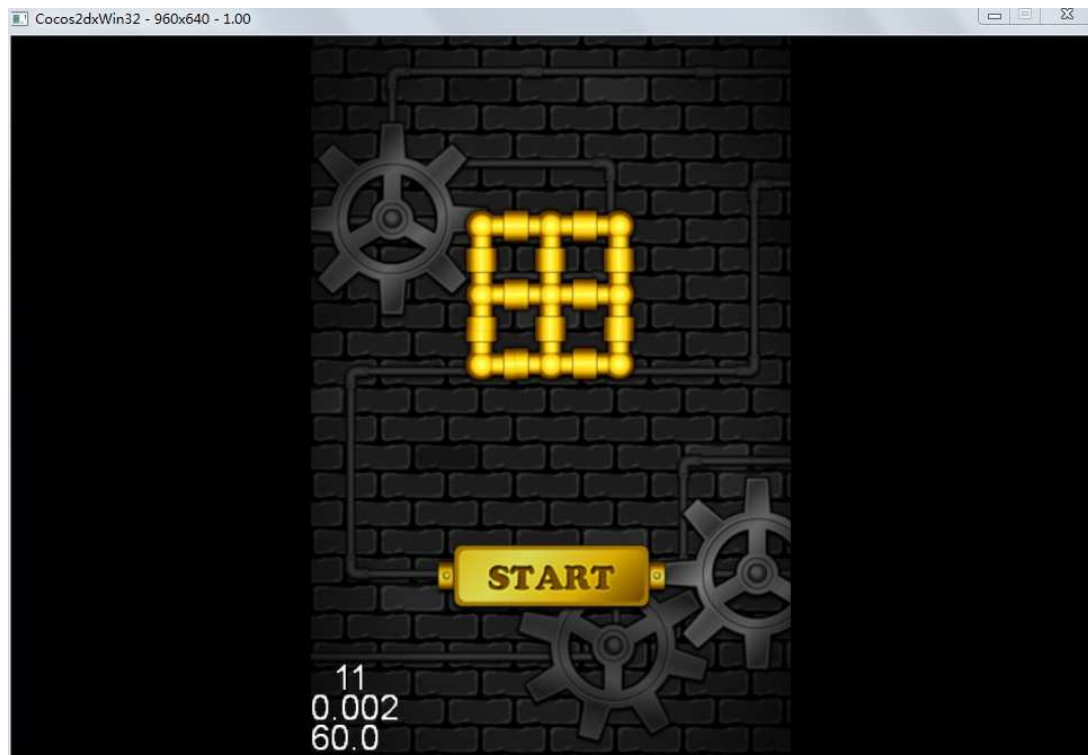
bool HelloWorld::init(){
    //////////////////////////////////////
    // 1. super init first
    if ( !CCLayer::init() ){
        return false;
    }

    GameUIController* uic = GameUIController::I();
    addChild( uic );
    uic->showTui( s_t_panel_start );
    // enable standard touch
    this->setTouchEnabled(true);
    return true;
}

```



- 编译运行后显示下图:



这时我们点start是没有反应的，因为我们的事件处理是空函数，急性子同学可以现在在 `GameUIController::on_event_panel_start_click_btn_start` 里打个log看看效果

- 写 `GameUIController::on_event_panel_start_click_btn_start` 的处理，这里我们想在点start后切换到 `panel_game`，函数体为：

```
void GameUIController::on_event_panel_start_click_btn_start( void* o, TEvent* e ){
    GameUIController::I()->closeAll();
    GameUIController::I()->showTui( s_t_panel_game );
}
```

- 同样的点击back按钮后，我们希望切回 `panel_start` 面板：

```
void GameUIController::on_event_panel_game_click_btn_back( void* o, TEvent* e ){
    GameUIController::I()->closeAll();
    GameUIController::I()->showTui( s_t_panel_start );
}
```

- 在使用Tui过程中，可能想为图片或面板加上一些action,如移动，旋转等效果，可以使用函数 `Tui* pui = GameUIController::I()->getTui( s_t_panel_start );` 得到该面板，然后为该面板 `runAction` 加上相应的效果。
- 如果是想得到Tui里的子控件，可以在得到Tui后使用 `Tui::getControl( name )` 的方法得到子控件。
- OK,这样一个近乎完整洁的Tui使用做完了。

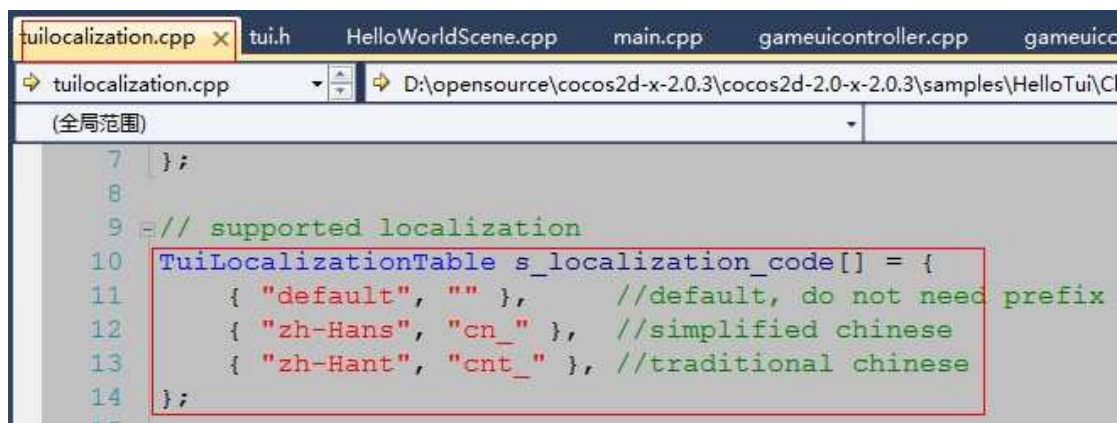
## 添加 ipad 版本

- Ipad 版与 iphone 的制作方法相同
- FlashCS 里打开 hellotui.fla，为主场景添加一个 ipad 层，然后为 ipad 放在与 iphone 相同的 UI，注意每个控件实例的名字都要与 iphone 版本的一样，比如开始面板实例在 iphone 里叫 panel\_start，在 ipad 里也得叫 panel\_start，这样程序在不同的分辨率时才会根据相同的名字控件。这里我就不专门做 ipad 版本了，大家可以看 tuisample 里的 ipad 做法。

## 多语言资源管理

之前我们在找按钮图片资源的时候提到过多语言控制。下面我们介绍一下如果在不同的语言环境下显示不同的资源。

先看一下 tuilocalization.cpp 源文件吧



```
7 };
8
9 // supported localization
10 TuiLocalizationTable s_localization_code[] = {
11     { "default", "" }, //default, do not need prefix
12     { "zh-Hans", "cn_" }, //simplified chinese
13     { "zh-Hant", "cnt_" }, //traditional chinese
14 };
```

文件里定义了三个语言，默认为default，前缀是空的，比如有个按钮的图片用的是 menu\_start.png，当没有调用 `TuiLocalization::l()->setLocalization`，或者设置了Tui里没有定义的语言时，Tui会加载menu\_start.png，如果把语言设为zh-Hans，如 `TuiLocalization::l()->setLocalization("zh-Hans")`，那Tui在加载menu\_start.png时会试图加载cn\_menu\_start.png，如果没有找到cn\_menu\_start.png那么Tui会加载 menu\_start.png。所以我们只需要做一套相应语言的按钮就可以了。

本地化设置可以在 AppDelegate:: applicationDidFinishLaunching 里加上

`const char* code = 得到设备的当前语言码;`

`TuiLocalization::l()->setLocalization(code);`



# 结束

哇哇哇~~~终于写完啦!!!

能耐着性子看到这里的都是”THE GOD LIKE”的选手!!!

谢谢大家!!!