

Cocos2dx-UI-Builder 说明文档

构造说明

现在支持 cocos2dx 里面的 CCLayer, CCMenu, CCMenuItem, CCMenuItemSprite, CCMenuItemToggle, CCLabelTTF, CCSprite, CCNode 等类的构造, 如果需要支持其它 cocos 2d 中类的构造, 需要自己进行添加。

常量信息:

```
typedef enum
{
    kCCTextAlignmentLeft,
    kCCTextAlignmentCenter,
    kCCTextAlignmentRight,
} CCTextAlignment;

typedef enum
{
    kCCVerticalTextAlignmentTop,
    kCCVerticalTextAlignmentCenter,
    kCCVerticalTextAlignmentBottom,
} CCVerticalTextAlignment;

enum 按钮状态
{
    CCControlStateNormal = 1 << 0, // The normal, or default state of a control; that is, enabled but neither selected nor highlighted.
    CCControlStateHighlighted = 1 << 1, // Highlighted state of a control. A control enters this state when a touch down, drag inside or drag enter is performed. You can retrieve and set this value through the highlighted property.
    CCControlStateDisabled = 1 << 2, // Disabled state of a control. This state indicates that the control is currently disabled. You can retrieve and set this value through the enabled property.
    CCControlStateSelected = 1 << 3 // Selected state of a control. This state indicates that the control is currently selected. You can retrieve and set this value through the selected property.
};
```

自定义场景

CCControlLayer:

1、自定义场景必须从 CCControlLayer 继承过来，通过 XML 文件来构造自定义场景。
示例：

H 文件	<pre>class SceneMain : public CCControlLayer { public: SceneMain(const char* xmlfile); virtual ~SceneMain(); virtual bool init(); static CCScene* scene(std::string xmlFile); }</pre>
CPP 文件	<pre>CCScene* SceneMain::scene(std::string xmlFile) { CCScene *scene = CCScene::create(); CCLayer *pLayer = new SceneMain(xmlFile.c_str()); if (pLayer && pLayer->init()) pLayer->autorelease(); else CC_SAFE_DELETE(pLayer); scene->addChild(pLayer); return scene; } SceneMain::SceneMain(const char* xmlfile) : CCControlLayer(xmlfile) { } SceneMain::~~SceneMain() { }</pre>

	<pre>bool SceneMain::init() { if (!CCControlLayer::init()) return false; return true; }</pre>
场景获取	<pre>CCScene *pScene = SceneMain::scene("scene/test.xml"); // run pDirector->runWithScene(pScene);</pre>

2、自定义场景请实现如下方法，在合适的方法中去进行初始化和反初始化

`virtual bool init();`//场景第一次构造

`virtual void onEnter();`//场景激活，适合手动注册事件

`virtual void onExit();`//场景退出，要清理掉手动注册事件

CCLayerLoading:

该场景是为了让游戏画面运行流程，实现在后台加载数据。

属性		说明	示例
loading		根节点	<pre>{ "loading": [{"type":"animation", "file":"json/ani_qizhi.json"}, {"type":"animation", "file":""}, {"type":"animation", "file":""}, {"type":"animation", "file":""}, {"type":"spriteframe", "file":"animation/1.plist"}, {"type":"texture", "file":""}, {"type":""," "file":""}] }</pre>
	type	缓存类型	
	file	文件路径	

使用方法:

`CCScene *pScene = SceneMain::scene("scene/main.xml");`

`// run`

`pDirector->runWithScene(CCLayerLoading::scene("json/loading.json", pScene));`

节点构造

CCNode:

一般不会直接去创建 CCNode 节点，该节点属性均为公共属性

参数	说明	示例
tag	int, 节点 ID	tag="1000"
visible	0:不可见, 1:可见	visible="1"
zorder	"0", 越小优先级越高	zorder="0"
size	size "200, 300"	size="200, 300"
skewx	float, 沿 x 轴扭曲	skewx="0.5"
skewy	float, 沿 y 轴扭曲	skewy="0.5"
pos	节点位置, 以 anchor 为参考点	pos="400, 240"
anchor	节点的参考点 (锚点)	anchor="400, 240"
rotate	逆时针为正, 单位为角度 (0-360)	rotate="180"
scalex	float, 沿 x 轴缩放	scalex="2.0"
scaley	float, 沿 y 轴缩放	scaley="0.5"

CCNodeRGBA:

一般不会直接去创建 CCNode 节点，该节点属性均为公共属性

参数	说明	示例
opacity	透明值0-255	opacity="100"
color	显示颜色	color="#ffff00"
cascade_color	Color 值可以传递子控件	cascade_color="1"
cascade_opacity	Opacity 可以传递子控件	cascade_opacity="1"

CCLayer:

示例:

```
<CCLayer tag="103" visible="0">
</CCLayer>
```

CCMenu:

需要设置 pos 属性, anchor 默认为(0.5,0.5), 调整 anchor 属性无效, 所以不要去设置 anchor 属性。

示例:

```
<CCMenu tag="1001" pos="400,50" >
</CCMenu>
```

CCMenuItem:

一般不会直接去构造该节点, 该节点只具有点击事件。

参数	说明	示例
m_enable	控件状态	m_enable="1000"
m_selected	0:未选中, 1:选中	m_selected="1"

示例:

```
<CCMenuItem tag="1001" click=1 size="100,50" pos="400,50" >
    <CCSprite tag="102102" file="help/arrow_s.png" size="100,63" />
</CCMenuItem>
```

CCMenuItemSprite:

参数	说明	示例
anchor	默认为 (0.5, 0.5)	<CCMenuItemSprite tag="10210" visible="1" click="1" anchor="0.5,0.5" pos="50,0">
click	必须带上该属性以注册点击事件	<CCSprite tag="102100" file="help/arrow_n.png" size="100,63" > <CCLabelTTF tag="1021001" text="上一页" font_size="20" font_color="#ffffff" font_name="微软雅黑" anchor="0.5,0.5" pos="80,30"/> </CCSprite> <CCSprite tag="102102" file="help/arrow_s.png" size="100,63" > <CCLabelTTF tag="1021021" text="上一页" font_size="22" font_color="#ff0000" font_name="微软雅黑" anchor="0.5,0.5" pos="80,30"/> </CCSprite> </CCMenuItemSprite>
CCSprite	1、该节点为 CCMenuItemSprite 的子节点, 可以设置 size 大小调整点击区域, 2、该节点 anchor 不用设置, 始终为 (0,0)	
rotate	 箭头旋转, 文字不旋转时, 可以先旋转 CCSprite, 然后再反方	<CCMenuItemSprite tag="10221" visible="1" click="1" pos="500,0" anchor="0.5,0.5" rotate="180"> <CCSprite tag="102210" file="help/arrow_n.png" size="100,63" > <CCLabelTTF tag="1022101" text="下一页" font_size="20" font_color="#ffffff" font_name="微软雅黑" pos="80,30" rotate="-180"/> </CCSprite> <CCSprite tag="102212" file="help/arrow_s.png" size="100,63" > <CCLabelTTF tag="1022121" text="下一页" font_size="22" font_color="#ff0000" font_name="微软雅黑" pos="80,30" rotate="180"/> </CCSprite> </CCMenuItemSprite>

	向旋转文字进行恢复	<pre>font_color="#ff0000" font_name="微软雅黑" pos="80,30" rotate="-180"/> </CCSprite> </CCMenuItemSprite></pre>
--	-----------	---

CCMenuItemToggle:

参数	说明	示例
CCMenuItemToggle::anchor	两者的 anchor 最好不要去设置，默认为 (0.5,0.5)，如果设置不好，会导致点击区域错误	<pre><CCMenu tag="1001" pos="400,240"> <CCMenuItemToggle tag="10310" pos="0,0" selectedIndex="0"> <CCMenuItemSprite tag="10010" pos="1,1"> <CCSprite tag="100101" file="btn_normal.png"> <CCLabelTTF tag="1001011" text="Play" font_size="20" font_color="#ffffff" font_name="fonts/FZLBJW.TTF" anchor="0.5,0.5" pos="84,15"/> </CCSprite> <CCSprite tag="100102" file="btn_sel.png"> <CCLabelTTF tag="1001021" text="Play" font_size="22" font_color="#ffcc00" font_name="fonts/FZLBJW.TTF" anchor="0.5,0.5" pos="84,15"/> </CCSprite> <CCSprite tag="100103" file="btn_sel.png"> <CCLabelTTF tag="1001031" text="Play" font_size="22" font_color="#ffcc00" font_name="fonts/FZLBJW.TTF" anchor="0.5,0.5" pos="84,15"/> </CCSprite> </CCMenuItemSprite> <CCMenuItemSprite tag="10020" pos="1,1"> <CCSprite tag="100201" file="btn_normal.png"> <CCLabelTTF tag="10021011" text="Help" font_size="20" font_color="#ffffff" font_name="fonts/FZLBJW.TTF" anchor="0.5,0.5" pos="84,15"/> </CCSprite> <CCSprite tag="100202" file="btn_sel.png"> <CCLabelTTF tag="1002021" text="Help" font_size="22" font_color="#ffcc00" font_name="fonts/FZLBJW.TTF" anchor="0.5,0.5" pos="84,15"/> </CCSprite> <CCSprite tag="102103" file="btn_sel.png"> <CCLabelTTF tag="1002031" text="Help" font_size="22" font_color="#ffcc00" font_name="fonts/FZLBJW.TTF" anchor="0.5,0.5" pos="84,15"/> </CCSprite> </CCMenuItemSprite> </CCMenuItemToggle> </CCMenu></pre>
CCMenuItemSprite::anchor		
CCSprite	1、该节点为 CCMenuItemSprite 的子节点，可以设置 size 大小调整点击区域， 2、该节点 anchor 不用设置，始终为 (0,0)	

		<pre> </CCSprite> </CCMenuItemSprite> </CCMenuItemToggle> </CCMenu> </pre>
--	--	--

CCSprite:

参数	说明	示例
file	图片地址	file="help/frame.png"
rect	图片纹理区域	rect="588,0,90,190"
flip	1: 沿 x 轴翻转 2: 沿 y 轴翻转	flip="1" flip="2"

示例:

```
<CCSprite tag="10231" file="help/frame.png" anchor="0,1" pos="100,190" rect="588,0,90,190"/>
```

支持从 plist 文件读取纹理（纹理名称为 plist 文件中的 frame 名字）

```
<CCSprite tag="10231" plist="help/frame.png" anchor="0,1" pos="100,190" rect="588,0,90,190"/>
```

CCScale9Sprite:

参数	说明	示例
file	图片地址	file="help/frame.png"
rect	图片纹理区域	rect="588,0,90,190"
inrect		
size	拉伸后的大小	

示例:

```
<CCScale9Sprite tag="1000" file="frame1.png" rect="10,10,10,10" inrect="0,0,0,0" size="300,60" anchor="0,0" pos="200,150"/>
```

支持从 plist 文件读取纹理（纹理名称为 plist 文件中的 frame 名字）

```
<CCScale9Sprite tag="1000" plist="frame1.png" inrect="0,0,0,0" size="300,60" anchor="0,0" pos="200,150"/>
```

CCLabelTTF:

参数	说明	示例
text	文字内容	text="Play"
font_size	文字大小	font_size="22"

font_name	字体库	font_name="fonts/FZLBJW.TTF"
font_color	文字颜色	font_color="#ffcc00"
dimension	显示区域	dimension="200,40"

示例：

```
<CCLabelTTF tag="1001031" text="Play" font_size="22" font_color="#ffcc00" font_name="fonts/FZLBJW.TTF" anchor="0.5,0.5" pos="84,15" dimension="200,40"/>
```

CCLabelAtlas:

参数	说明	示例
text	文字内容	text="0123456789:", 图片对应坐标以起始字符的 ascii 码自增长
item_w	宽度	item_w="14"
Item_h	高度	item_h="21"
Start_ch	开始字符	start_ch="0"
file	图片文件名	file="set1_player_hud3.png"

示例：



```
<CCLabelAtlas tag="1000" text="123456789:0" item_w="14" item_h="21" start_ch="0" file="set1_player_hud3.png" pos="200,200" />
```

CCLabelBMFont:

参数	说明	示例
text	文字内容	text="中国"
width	宽度	width="14"
align	对齐方式	align="0"
offset	偏移	offset="0,0"
file	图片文件名	file="bitmapFontChinese.fnt"
color	文字颜色	color="#ff0000", 可以设置文字颜色

示例：

```
<CCLabelBMFont tag="1001" file="bitmapFontChinese.fnt" text="中国" width="14" align="0" offset="0,0" pos="400,200" color="#ff0000"/>
```

CCProgressTimer:

参数	说明	示例
file	图片文件名	file="pt.png"
type	进度条类型	Type="0" 0: Radial 1: Bar

percent	比率	percent="0.15"
midpos	中心点坐标	midpos="0.5,0.5"
rate	变化率	rate="1.0,1.0"
reverse	是否翻转	reverse="0"

示例：

```
<CCProgressTimer tag="1002" file="pt.png" type="0" percent="0" midpos="0.5,0.5" rate="1,1" reverse="0" pos="114,135" anchor="1,1" color="#ffffff"/>
```

支持从 plist 文件读取纹理（纹理名称为 plist 文件中的 frame 名字）

```
<CCProgressTimer tag="1002" plist="pt.png" type="0" percent="0" midpos="0.5,0.5" rate="1,1" reverse="0" pos="114,135" anchor="1,1" color="#ffffff"/>
```

CCMotionStreak:

参数	说明	示例
file	图片文件名	file="streak.png"
fade	间隐的时间	fade="1.0"
segment	间隐片断的大小	segment="16"
stroke	贴图的宽高	stroke="16"
blend	混合模式	blend="10000,10001"
color	混合颜色	color="#ff0000"

示例：

```
<CCMotionStreak tag="1003" file="streak.png" fade="1.0" segment="16" stroke="16" pos="114,135" blend="1000,10001" anchor="1,1" zorder="1" color="#ff0000"/>
```

CCEditBox

参数	说明	示例
spr9_normal	背景图片-常态	spr9_normal="frame1.png"
spr9_press	背景图片-按下态	spr9_press="frame1.png"
spr9_disable	背景图片-不可用	Font_name="16"
tag_normal	常态 CCScale9Sprite 的 tag	tag_normal=""
tag_press	按下态 CCScale9Sprite 的 tag	tag_press=""
tag_disable	不可用 CCScale9Sprite 的 tag	tag_disable=""
spr9_rect	背景图片区域	spr9_rect="0,0,44,44"
spr9_inrect	图片拉伸区域	spr9_inrect="10,10,10,10"
ph_text	空白提示文字	ph_text="abcdf"
ph_font_color	空白提示文字颜色	ph_font_color="#ffff00"
ph_font_name	提示字体	ph_font_name="Arial"

ph_font_size	提示文字大小	ph_font_size="20"
text	文字内容	text=""
font_name	文字字体	font_name="微软雅黑"
font_size	文字大小	font_size="24"
font_color	文字颜色	font_color="#ff0000"
input_flag	0-kEditBoxInputFlagPassword 1-kEditBoxInputFlagSensitive 2-kEditBoxInputFlagInitialCapsWord 3-kEditBoxInputFlagInitialCapsSentence 4-kEditBoxInputFlagInitialCapsAllCharacters	input_flag="4"
input_mode	0-kEditBoxInputModeAny 1-kEditBoxInputModeEmailAddr 2-kEditBoxInputModeNumeric 3-kEditBoxInputModePhoneNumber 4-kEditBoxInputModeUrl 5-kEditBoxInputModeDecimal 6-kEditBoxInputModeSingleLine	input_mode="0"
max_length		max_length="256"
return_type	0-kKeyboardReturnTypeDefault 1-kKeyboardReturnTypeDone 2-kKeyboardReturnTypeSend 3-kKeyboardReturnTypeSearch 4-kKeyboardReturnTypeGo	return_type="0"
color	背景填充色	color="#ffffff"
size	区域大小	size="300,60"

示例：

```
<CCEditBox tag="1006" spr9_normal="frame1.png" spr9_press="frame1.png" spr9_rect="0,0,44,44" spr9_inrect="10,10,10,10" ph_text="abcd" ph_font_color="#ffff00" ph_font_name="Arial" ph_font_size="20" text="" font_name="微软雅黑" font_size="24" font_color="#ff0000" align="1" input_flag="" input_mode="" max_length="256" return_type="0" color="#ffffff" size="300,60" anchor="0,0" pos="400,50"/>
```

CCTextFieldTTF

```
<CCTextFieldTTF tag="1005" placeholder="frame1.png" align="1" font_name="微软雅黑" font_size="12" size="300,60" anchor="0,0" pos="400,150"/>
```

CCControl

参数	说明	示例
enable	是否可用	enable="1"

selected	是否选中	selected="1"
highlight	是否高亮	highlight="1"
bOpacityModifyRGB		bOpacityModifyRGB="1"

注意：该类不会直接构造，该类的参数为公共参数。

CCControlButton

参数	说明	示例
adjustbg	是否自适应文字大小	
p_size	显示区域大小，设置后会固定大小	
pos		
bg_normal		bg_normal="file=frame1.png^tag=10095^rect=0,0,44,44^inrect=10,10,10,10"
bg_highlight		同上
bg_disable		同上
bg_select		同上
ttf_normal		tftlabel="tag=10081^text=Touch Me!^font_size=30^font_name=Arial^font_color=#ff0000"
ttf_highlight		同上
ttf_disable		同上
ttf_select		同上
bmf_normal		bmfontlabel="tag=10091^file=bitmapFontChinese.fnt^text=A 中国 B^width=14^align=0^offset=0,0^font_color=#ff0000"
bmf_highlight		同上
bmf_disable		同上
bmf_select		同上

示例：

```
<CCControlButton tag="1008" ttf_normal="tag=10081^text=Touch Me!^font_size=30^font_name=Arial^font_color=#ff0000"
    bg_normal="file=frame1.png^tag=10085^rect=0,0,44,44^inrect=10,10,10,10"
    pos="400,400" p_size="300,100"/>
```

注意：这里的子控件信息采用^符号进行连接，属性值不能带双引号，例如：ttf_normal="tag=10081^text=Touch Me!^font_size=30^font_name=Arial^font_color=#ff0000"

CCControlSlider

参数	说明	示例
----	----	----

spr_bg	轨道图片	spr_bg="sliderTrack2.png"
spr_progress	进度条图片	spr_progress="sliderProgress2.png"
spr_thumb	滑块图片	spr_thumb="sliderThumb.png"
max_value	最大值	max_value="999"
min_value	最小值	min_value="0"
value	当前值	value="555"
rotate	旋转	rotate="-90"

示例：

```
<CCControlSlider tag="1011" spr_bg="sliderTrack2.png" spr_progress="sliderProgress2.png" spr_thumb="sliderThumb.png" max_value="999" min_value="0" value="555" enable="1" pos="200,200" rotate="-90"/>
```

CCControlColourPicker


参数	说明	示例
color		

示例：

```
<CCControlSlider tag="1011" spr_bg="sliderTrack2.png" spr_progress="sliderProgress2.png" spr_thumb="sliderThumb.png" max_value="999" min_value="0" value="555" enable="1" pos="200,200" rotate="-90"/>
```



CCControlPotentiometer

参数	说明	示例
spr_bg	轨道图片 	spr_bg="sliderTrack2.png"
spr_progress	进度条图片	spr_progress="sliderProgress2.png"

		
spr_thumb	中心图片 	spr_thumb="sliderThumb.png"
max_value	最大值	max_value="999"
min_value	最小值	min_value="0"
value	当前值	value="555"
rotate	旋转	rotate="-90"

示例：

```
<CCControlPotentiometer tag="1012" spr_bg="extensions/potentiometerTrack.png" spr_progress="extensions/potentiometerProgress.png" spr_thumb="extensions/potentiometerButton.png" max_value="999" min_value="0" value="555" enable="1" pos="100,400" rotate="0"/>
```



CCControlStepper

参数	说明	示例
spr_bg	轨道图片	spr_bg="sliderTrack2.png"
spr_progress	进度条图片	spr_progress="sliderProgress2.png"
spr_thumb	中心图片	spr_thumb="sliderThumb.png"
max_value	最大值	max_value="1000"
min_value	最小值	min_value="0"
value	当前值	value="500"
step	步进	step="100"
repeat	自动重复：非零	repeat="0"
rotate	旋转	rotate="-90"

示例：

```
<CCControlStepper tag="1013" spr_minus="extensions/stepper-minus.png" spr_plus="extensions/stepper-plus.png" max_value="1000" min_value="0" value="500" step="100" repeat="0" enable="1" pos="100,400" rotate="0"/>
```

CCControlSwitch（添加多个有 bug）

参数	说明	示例
spr_mask	背景图片	spr_mask="extensions/switch-mask.png"
spr_on	打开状态	spr_on="extensions/switch-on.png"
spr_off	关闭状态	spr_off="extensions/switch-off.png"
spr_thumb	按钮	spr_thumb="extensions/switch-thumb.png"
ttf_on	打开文字	ttf_on="tag=10081^text=On^font_size=30^font_name=Arial-BoldMT^font_color=#ff0000"
ttf_off	关闭文字	ttf_off="tag=10081^text=Off!^font_size=30^font_name=Arial-BoldMT^font_color=#ff0000"
is_on	状态(0 为关)	is_on="0"
rotate	旋转	rotate="-90"

示例：

```
<CCControlSwitch tag="1014" spr_mask="extensions/switch-mask.png" spr_on="extensions/switch-on.png" spr_off="extensions/switch-off.png" spr_thumb="extensions/switch-thumb.png" ttf_on="tag=10081^text=On^font_size=30^font_name=Arial-BoldMT^font_color=#ff0000" ttf_off="tag=10081^text=Off!^font_size=30^font_name=Arial-BoldMT^font_color=#ff0000" is_on="0" enable="1" pos="100,400" rotate="0"/>
```

CCControlHuePicker

CCControlSaturationBrightnessPicker

CCScrollView

CCScrollView 只可以设置一个子节点，子节点设置为 scroll 的 container 对象。

参数	说明	示例
offset	子控件的偏移量	offset="0,0"
direction	kCCScrollViewDirectionNone = -1, kCCScrollViewDirectionHorizontal = 0, kCCScrollViewDirectionVertical=1 kCCScrollViewDirectionBoth=2	direction="1"
view_size	视图大小（会自动设置 contentsize 为 view_size 大小）	view_size="100,100"
touch	Touch 事件是否可用	touch="1"

示例：

```
<CCScrollView tag="1001" pos="600,150" view_size="100,100" offset="0,0" direction="1" touch="1">
    <CCLayer tag="10011" size="100,150" pos="0,0">
        <CCControlButton tag="1008" ttf_normal="tag=10081^text=Touch Me!^font_size=20^font_name=Arial^font_color=#ff0000" bg_normal="file=frame1.png^tag=10085^rect=0,0,44,44^inrect=10,10,10,10" pos="0,0" p_
```

```

size="100,50" anchor="0,0"/>
    <CCControlButton tag="1008" ttf_normal="tag=10081^text=Touch Me!^font_size=20^font_name=Arial^font_color=#ff0000" bg_normal="file=frame1.png^tag=10085^rect=0,0,44,44^inrect=10,10,10,10" pos="0,50" p_size="100,50" anchor="0,0"/>
    <CCControlButton tag="1008" ttf_normal="tag=10081^text=Touch Me!^font_size=20^font_name=Arial^font_color=#ff0000" bg_normal="file=frame1.png^tag=10085^rect=0,0,44,44^inrect=10,10,10,10" pos="0,100" p_size="100,50" anchor="0,0"/>
    </CCLayer>
</CCScrollView>

```

CCTableView

参数	说明	示例
fill_order	kCCTableViewFillTopDown = 0 kCCTableViewFillBottomUp = 1	fill_order="1"
参考 CCScrollView		

示例：

```

<CCTableView tag="1022" view_size="250,60" direction="0" touch="1" datasource="scene/cell.xml" pos="300,100"/>

```

● CCTableViewCell 模板

下面为 scene/cell.xml 内容：

```

<?xml version="1.0" encoding="utf-8"?>
<!--      TableViewCell 单元模板      -->
<CCTableViewCell tag="1001" >
    <CCSprite tag=[spriteTag] file=[spriteFile] anchor="0,0" pos="0,0"/>
    <CCLabelTTF tag=[ttfTag] text=[text] font_size="20" font_name="Helvetica" anchor="0,0" pos="0,0"/>
</CCTableViewCell>

```

- cell 子控件的 tag 必须相同（第一个和第二个的 cell 子控件 tag 要相同，在修改子控件时需要根据 tag 值去获取子控件，如果不相同的话会获取失败）

例如下面为 TableViewCell 参数值，用 json 格式表示：

```

{
    "params":[
        {
            "spriteTag": "1",
            "spriteFile": "Icon.png",
            "ttfTag": "2",
            "text": "1"
        },
        {
            "spriteTag": "1",
            "spriteFile": "Icon.png",

```

```

        "ttfTag": "2",
        "text": "2"
    }
}
]
}

```

第一个 cell 的 spriteTag=1，ttfTag=2 与第二个 cell 的 spriteTag=1，ttfTag=2 相同。

自定义 Layer 类继承自 CControlLayer，需要实现以下方法（更新 cell 内容）：

```

tableCellContentAtIndex(CCTableView *table, unsigned int idx, CCTableViewCell* cell,
    Json* json)

```

使用说明：

std::map<int, std::string> mDSTemplateMap;——保存 cell 模板数据，该数据在读配置文件时会自动保存；如果是手动添加 CCTableView，则需要手动添加模板数据。

std::map<int, Json*> mJsonMap;——保存 cell 参数信息，一般从服务器获取，获取之后保存在 map 中，然后再调用 ccTableView 的 reloadData 方法进行界面刷新。

动画精灵

CCAnimateSpx:

自定义动画精灵从继承 CCAnimateSpx 进行继承，在自定义动画类中添加属性成员进行扩展。

该类的使用方法：

1、直接绘制

调用 CCAnimateSpx* create(const char* jsonFile)方法，将返回值 addchild 到相应的节点中。

2、通过 CCSpriteBatchNode 进行绘制

调用 CCAnimateSpx* create(CCSpriteBatchNode** node, const char* jsonFile)方法，需要传入 CCSpriteBatchNode 对象，CCSpriteBatchNode 对象需要 addchild 到批量渲染的图层中。（返回的 CCAnimateSpx 对象最好也 addchild 到和 CCSpriteBatchNode 同一节点）

示例：

```

this->mArmyLayer=(CCLayer*)ArmyLayer::create();
this->addChild(mArmyLayer,0);
for(int n=0; n<1; n++)
{
    CCAnimateSpx* pUnit = CCAnimateSpx::create(&m_pBatchNode, "scene/animation.json");
    pUnit->setPosition(ccp(rand()%800, rand()%480));
    mArmyLayer->addChild(pUnit, 0, 700000+n);
}
mArmyLayer->addChild(m_pBatchNode);

```



```
for(int n=0; n<1; n++)
```

```
(CCAnimateSpx*)mArmyLayer->getChildByTag(700000+n))->switchAnimate("UnitWalk");
```

Json 文件格式:

```
{
  "animations": ——动画数组
  [
    {
      "name": "UnitWalk", ——动画名称
      "delay": "0.1", ——间隔时间
      "frames": ——动画帧
      [
        "a0.png",
        "a1.png",
        "a2.png",
        "a3.png",
        "a4.png",
        "a5.png",
        "a6.png",
      ]
    },
    {
      "name": "UnitRun",
      "delay": "0.1",
      "frames": [
        "a7.png",
        "a8.png",
        "a9.png",
        "a10.png",
        "a11.png",
        "a12.png",
        "a13.png",
        "a14.png",
        "a15.png",
        "a16.png"
      ]
    }
  ],
  "image": "unit/1.png", ——纹理图片
  "plist": "unit/1.plist" ——plist 文件
}
```

注意事项

1、不要继承 `cocos2d` 的类，这会需要重载很多父类的函数，而且还容易产生错误。如果需要扩展 `cocos2d` 中的类，采用组装的形式来进行封装，将 `cocos2d` 中的类 `addChild()` 到自定义类中，然后通过定义成员变量来保存自己所需要的信息。