### 1.3 在屏幕上显示内容

本节学习在Cocos2d-x中使用自定义字体、中文显示、导入音乐文件等内容。

**【本节知识点】**

* **Cocos2d-x中标签的使用方法**
* **在场景中使用自定义字体及中文字体**
* **Cocos2d-x中显示背景及其他内容**
* **Cocos2d-x中菜单的使用方法**
* **Cocos2d-x中播放音乐**
* **Cocos2d-x中实现屏幕适配功能**

**（1）显示文字**

Cocos2d-x中显示文字需要用到Label控件。

**使用系统字体，在屏幕上输出一行文字的代码如下：**

auto label1 = Label::create("this is the first label","Arial",36,Size::ZERO,

TextHAlignment::LEFT,TextVAlignment::TOP);

label1->setPosition(210, 310);

this->addChild(label1);

create方法6个参数的含义为：

* 标签中所显示文字的内容
* 标签字体
* 标签内容字体大小
* 标签尺寸
* 标签中的横向对其方式
* 标签中的纵向对其方式

一般使用时可以进一步简化，仅保留前3项参数。

**使用TTF文件，加载外部字体文件，在屏幕上输出一行文字的代码如下：**

TTFConfig ttfconfig("fonts/chunkmuffinhollowwide.ttf", 36);

auto label2 = Label::createWithTTF(ttfconfig,"this is the second label");

label2->setPosition(270,250);

this->addChild(label2);

或者

auto label3 = Label::createWithTTF("this is the third label","fonts/Marker Felt.ttf",36);

label3->setPosition(200, 190);

this->addChild(label3);

为字体添加效果

auto label4 = Label::createWithTTF("this is the fourth label", "fonts/Marker Felt.ttf", 36);

label4->setPosition(280,130);

label4->enableShadow(Color4B::GREEN, Size(5, 5)); **//阴影效果**

label4->enableOutline(Color4B::RED, 3); **//边框效果**

this->addChild(label4);

auto label5 = Label::createWithTTF("this is the fifth label", "fonts/Marker Felt.ttf", 36);

label5->setPosition(200, 70);

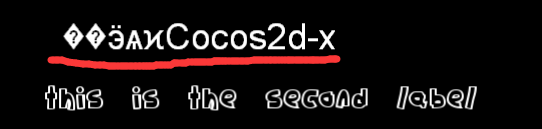
label5->enableGlow(Color4B::GREEN); **//光效**

this->addChild(label5);

上述代码的执行效果如下图所示：

**显示汉字**

在Cocos2d中直接使用中文是无法正确显示的，如果将上面代码中的显示内容修改为：“欢迎学习Cocos2d”就会显示下图的乱码：



**解决方法有两种：**

**第一种：**通过转换为utf-8编码的方式显示出来，需要建立一个用于转换中文到utf-8编码的FontToUTF8方法代码如下：

char\* HelloWorld::FontToUTF8(const char \*font)

{

int len = MultiByteToWideChar(CP\_ACP, 0, font, -1, NULL, 0);

wchar\_t\* wstr = new wchar\_t[len + 1];

memset(wstr, 0, len + 1);

MultiByteToWideChar(CP\_ACP, 0, font, -1, wstr, len);

len = WideCharToMultiByte(CP\_UTF8, 0, wstr, -1, NULL, 0, NULL, NULL);

char\* str = new char[len + 1];

memset(str, 0, len + 1);

WideCharToMultiByte(CP\_UTF8, 0, wstr, -1, str, len, NULL, NULL);

if (wstr) delete[] wstr;

return str;

}

然后显示中文label代码如下：

auto label1 = Label::create(HelloWorld::FontToUTF8("欢迎学习Cocos2d-x"), "Arial", 36, Size::ZERO);

label1->setPosition(210, 310);

this->addChild(label1);

**第二种：**引入iconv库实现中文显示

创建一个GBKToUTF8方法实现编码转换。

在HelloWorldScene.cpp文件首部添加下面代码：

#include "C:\Cocos\frameworks\cocos2d-x-3.9\external\win32-specific\icon\include\iconv.h"

GBKToUTF8方法代码如下：

int HelloWorld::GBKToUTF8(std::string &gbkStr, const char\* toCode, const char\* formCode)

{

iconv\_t iconvH;

iconvH = iconv\_open(formCode, toCode);

if (iconvH == 0)

{

return -1;

}

const char\* strChar = gbkStr.c\_str();

const char\*\* pin = &strChar;

size\_t strLength = gbkStr.length();

char\* outbuf = (char\*)malloc(strLength \* 4);

char\* pBuff = outbuf;

memset(outbuf, 0, strLength \* 4);

size\_t outLenghth = strLength \* 4;

if (-1 == iconv(iconvH, pin, &strLength, &outbuf, &outLenghth)){

iconv\_close(iconvH);

return -1;

}

gbkStr = pBuff;

iconv\_close(iconvH);

return 0;

}

显示中文label代码如下：

std::string hello = "欢迎学习Cocos2d-x";

GBKToUTF8(hello, "gb2312", "utf-8");

CCLabelTTF\* label1 = CCLabelTTF::create(hello.c\_str(), "Arial", 36);

label1->setPosition(210, 310);

this->addChild(label1);

**第三种：**使用XML解析中文内容的方法。创建XML文件，文件格式如下：

<dict>

<key>string1</key>

<string>欢迎学习cocos2dx</string>

<key>string2</key>

<string>使用XML显示中文</string>

<key>string3</key>

<string>key与string标签中的内容是对应的</string>

</dict>

保存文件是修改文件格式为utf-8格式。

**显示上面xml文件的代码如下：**

auto \*chnStrings = Dictionary::createWithContentsOfFile("CHN\_Strings.xml");

const char \*str1 = ((String\*)chnStrings->objectForKey("string1"))->getCString();

auto\* label1 = Label::create(str1, "Arial", 36);

label1->setPosition(320,270);

addChild(label1);

const char \*str2 = ((String\*)chnStrings->objectForKey("string2"))->getCString();

auto\* label2 = Label::create(str2, "Arial", 36);

label2->setPosition(320, 180);

addChild(label2);

const char \*str3 = ((String\*)chnStrings->objectForKey("string3"))->getCString();

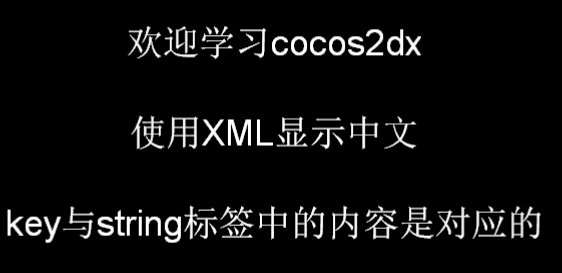
auto\* label3 = Label::create(str3, "Arial", 36);

label3->setPosition(320, 90);

addChild(label3);

return true;

显示结果如下图所示：



**（2）显示图像**

在Cocos2d-x中的图像是通过精灵来显示的。

下面代码的实现效果如下所示。



Size size = Director::getInstance()->getVisibleSize(); //获取屏幕尺寸

Vec2 origin = Director::getInstance()->getVisibleOrigin();

//显示背景图像，文件名为background.jpg

auto\* background = Sprite::create("background.jpg");

background->setPosition(size.width/2,size.height/2);

background->setScale(0.7f);

addChild(background,0);

//显示右侧的3个按钮

**auto\* button1 = Sprite::create("button.png");**

**button1->setPosition(550, size.height / 2);**

**button1->setScale(0.6f);**

**addChild(button1,1);**

auto\* button2 = Sprite::create("button.png");

button2->setPosition(550, size.height / 2-60);

button2->setScale(0.6f);

addChild(button2,1);

auto\* button3 = Sprite::create("button.png");

button3->setPosition(550, size.height / 2 - 120);

button3->setScale(0.6f);

addChild(button3,1);

//显示右侧的人物

auto\* renwu = Sprite::create("renwu.png");

renwu->setPosition(0,0);

renwu->setAnchorPoint(Vec2(0,0));

renwu->setScale(0.5f);

addChild(renwu);

上面代码中使用了一个新的方法setAnchorPoint，该方法的作用是设置当前的锚点。上面代码将人物图像的锚点设置在了其左下角的位置，此后再使用setPosition方法时，其中的参数就变成了图像的左下角与屏幕左下角的相对位置。

**（3）使用菜单**

本节为上面示例中的按钮图片添加点击效果。

下面的代码是实现运行结果中第二个按钮效果的代码。首先创建了一个新的菜单项pButton2，其中第1,2个参数为按钮在默认情况下与单击时所显示的图片素材的文件名，第3个参数this指针指向当前的场景，最后一个参数则为当按钮被单击时所执行的操作，将它们封装在一个函数中。然后在后面的语句中创建Menu类的对象button2

auto\* pButton2 = MenuItemImage::create("button1.png",

"button2.png",

this,

menu\_selector(HelloWorld::menu));

auto\* button2 = Menu::create(pButton2, NULL);

button2->setAnchorPoint(Vec2(0,0));

button2->setPosition(550, size.height / 2 - 60);

button2->setScale(0.6f);

addChild(button2);

auto\* pButton3 = MenuItemImage::create("button1.png",

"button2.png",

this,

menu\_selector(HelloWorld::menu));

auto\* label3 = LabelTTF::create("start", "Arial", 40);

auto\* pLabel3 = MenuItemLabel::create(label3, this, menu\_selector(HelloWorld::menu));

auto\* button3 = Menu::create(pButton3, pLabel3, NULL);

button3->setAnchorPoint(Vec2(0, 0));

button3->setPosition(550, size.height / 2 - 120);

button3->setScale(0.6f);

addChild(button3);

运行之后的效果如下图所示：



在Cocos2d-x中创建一个菜单按钮的方法分为两个步骤：

第1歩：创建新的菜单项MenuItemImage并设置相应的**回调函数。**menu\_selector(HelloWorld::menu)。

第2歩：将菜单添加到菜单创建参数中。如：

// 创建pButton2菜单项的菜单

auto\* button2 = Menu::create(pButton2, NULL);

// 创建pButton3和pLabel3菜单项的菜单

auto\* button3 = Menu::create(pButton3, pLabel3, NULL);

最后以NULL作为最后一个参数结尾。

**（4）实验1-4 ：在cocos中使用菜单**