概述

2014-2-14

要定义按钮对象，首先调用create函数。该函数传入的参数可以是图片的路径，也可以是精灵对象。两个参数代表的分别是按钮标题字和背景。之后可以设置按钮的参数，包括设置按钮位置和加入定义的回调函数等。这里的回调函数可以有多个，根据需要的操作定义，包括按下、拖动和抬起等。运行效果如图3-22所示。

按钮类CCControlButton的定义和初始化如代码清单3-28所示。代码是tests项目中ControlExtensionTest \ CCControlButtonTest目录下CCControlButtonTest.cpp中的CCControlButtonTest\_Event的init函数。

bool CCControlButtonTest\_Event::init()

{

if (CCControlScene::init())

{

CCSize screenSize = CCDirector::sharedDirector()->getWinSize();

// Add a label in which the button events will be displayed

setDisplayValueLabel(CCLabelTTF::create("No Event", "Marker Felt", 32));

m\_pDisplayValueLabel->setAnchorPoint(ccp(0.5f, -1));

m\_pDisplayValueLabel->setPosition(ccp(screenSize.width / 2.0f, screenSize.height / 2.0f));

addChild(m\_pDisplayValueLabel, 1);

// Add the button

CCScale9Sprite \*backgroundButton = CCScale9Sprite::create("extensions/button.png");

CCScale9Sprite \*backgroundHighlightedButton = CCScale9Sprite::create("extensions/buttonHighlighted.png");

CCLabelTTF \*titleButton = CCLabelTTF::create("Touch Me!", "Marker Felt", 30);

titleButton->setColor(ccc3(159, 168, 176));

/// 定¡§义°?按ã¡ä钮£¤

CCControlButton \*controlButton = CCControlButton::create(titleButton, backgroundButton);

controlButton->setBackgroundSpriteForState(backgroundHighlightedButton, CCControlStateHighlighted);

controlButton->setTitleColorForState(ccWHITE, CCControlStateHighlighted);

controlButton->setAnchorPoint(ccp(0.5f, 1));

controlButton->setPosition(ccp(screenSize.width / 2.0f, screenSize.height / 2.0f));

addChild(controlButton, 1);

// Add the black background

CCScale9Sprite \*background = CCScale9Sprite::create("extensions/buttonBackground.png");

background->setContentSize(CCSizeMake(300, 170));

background->setPosition(ccp(screenSize.width / 2.0f, screenSize.height / 2.0f));

addChild(background);

// Sets up event handlers

controlButton->addTargetWithActionForControlEvents(this, cccontrol\_selector(CCControlButtonTest\_Event::touchDownAction), CCControlEventTouchDown);

controlButton->addTargetWithActionForControlEvents(this, cccontrol\_selector(CCControlButtonTest\_Event::touchDragInsideAction), CCControlEventTouchDragInside);

controlButton->addTargetWithActionForControlEvents(this, cccontrol\_selector(CCControlButtonTest\_Event::touchDragOutsideAction), CCControlEventTouchDragOutside);

controlButton->addTargetWithActionForControlEvents(this, cccontrol\_selector(CCControlButtonTest\_Event::touchDragEnterAction), CCControlEventTouchDragEnter);

controlButton->addTargetWithActionForControlEvents(this, cccontrol\_selector(CCControlButtonTest\_Event::touchDragExitAction), CCControlEventTouchDragExit);

controlButton->addTargetWithActionForControlEvents(this, cccontrol\_selector(CCControlButtonTest\_Event::touchUpInsideAction), CCControlEventTouchUpInside);

controlButton->addTargetWithActionForControlEvents(this, cccontrol\_selector(CCControlButtonTest\_Event::touchUpOutsideAction), CCControlEventTouchUpOutside);

controlButton->addTargetWithActionForControlEvents(this, cccontrol\_selector(CCControlButtonTest\_Event::touchCancelAction), CCControlEventTouchCancel);

return true;

}

return false;

}