概述

2014-2-14

在游戏开发中，一般会把游戏分为两部分：一部分是游戏界面部分，也就是常说的UI部分（User Interface，用户界面）；另一部分就是游戏本身部分。有时UI有很多页面，在页面中用的图也并不是很多，不需要使用切换场景，只需把不同页面做成不同的布景层，然后切换布景层。那么这就需要一个“管理者”来管理这些界面，这时候就要使用CCLayerMultiplex类。

在很多游戏中都需要在不同的界面中使用相同的几个变量，如果不这样做，就需要做大量的保存工作。

tests项目中MenuTest.cpp的MenuTestScene类的runThisTest函数中有CCLayerMultiplex类的定义初始化方法，如代码清单3-21所示。

void MenuTestScene::runThisTest()

{

CCLayer\* pLayer1 = new MenuLayerMainMenu();

CCLayer\* pLayer2 = new MenuLayer2();

CCLayer\* pLayer3 = new MenuLayer3();

CCLayer\* pLayer4 = new MenuLayer4();

CCLayer\* pLayer5 = new MenuLayerPriorityTest();

CCLayer\* pLayer6 = new BugsTest();

CCLayer\* pLayer7 = new RemoveMenuItemWhenMove();

CCLayerMultiplex\* layer = CCLayerMultiplex::create(pLayer1, pLayer2, pLayer3, pLayer4, pLayer5, pLayer6, pLayer7, NULL);

addChild(layer, 0);

pLayer1->release();

pLayer2->release();

pLayer3->release();

pLayer4->release();

pLayer5->release();

pLayer6->release();

pLayer7->release();

CCDirector::sharedDirector()->replaceScene(this);

}

首先定义并初始化每个布景层类，然后将这些布景层实例以参数形式传给CCLayerMultiplex的create函数，最后以NULL（空）结束。

这里在传入参数之后将这些布景层实例的指针释放，是为了防止内存泄露。至于Cocos2D-x的内存管理，本书将会在后面的章节介绍。

然后把CCLayerMultiplex实例作为子节点传入场景中，最后运行场景。代码清单3-22所示是切换布景层的switchTo函数使用方法。

void MenuLayerMainMenu::menuCallback(CCObject\* sender)

{

((CCLayerMultiplex\*)m\_pParent)->switchTo(1);

}

由于这个函数被CCLayerMultiplex实例的子布景，即初始化CCLayerMultiplex传入的布景类实例调用，所以它的m\_pParent父节点就是CCLayerMultiplex实例本身。获得CCLayerMultiplex实例指针后，调用switchTo函数就可以转换到相应的子布景中。