### 1.2 理解Cocosd-2x中的一些基本概念

本节利用已有的Cocos2d-x，在其中新建一个用于显示场景的新类，以此来理解Cocos2d-x中各部分的运行顺序及启动原理。

**【本节知识点】**

* **如何在Cocos2d-x中新建一个项目并运行**
* **如何创建一个新类并根据需要对项目进行设置**
* **导演、节点、图层、精灵等概念在Cocos2d-x中是怎样使用的，以及它们之间的区别。**

在Cocos2d-x中场景图中，节点是基本元素，换句话说，场景图中的每一个对象都必须是节点对象或者节点对象类的子类生成的对象。

【注意】场景图和场景类的区别

场景图是指屏幕上一切已经显示了的或者要显示的元素的容器，可简单地理解为屏幕。

场景类是节点的一个子类，是在屏幕中被显示出来的一个元素。

图2-1显示了场景类（Scene）、图层类（Layer）等于Node之间的关系。



图2-1 Node类与其子类的关系

通过本实验理解Cocos2d-x中的导演、节点、场景、图层、精灵等概念。

1. **节点**

Node类中并没有提供用来具体显示某个内容的方法，但是提供了显示其子对象的方法和属性。Node类中的属性如表2-1所示。

|  |  |  |
| --- | --- | --- |
| **属性** | **描述** | **默认值** |
| position | 位置 | 0,0 |
| scale(x,y) | 缩放尺寸 | 1,1 |
| rotation(indegrees,clockwise) | 旋转角度 | 0 |
| Camera | 摄像机朝向 | 对象 |
| GridBase | 模型网格 | 对象 |
| anchorPoint | 锚点 | 0,0 |
| size | 尺寸 | 1,1 |
| visible | 是否可见 | TRUE |
| z-order | z坐标 | 0 |
| openGL z position | openGL Z坐标 | 0 |

Node类中并没有提供具体显示某个内容的方法，那么Node类究竟怎样实现物体的显示呢？这就不得不提到Node类及其子类的3个特点：

* 可以包含Node类机器子类对象作为子节点，对应的方法为addChild、getChildByTag、removeChild等。
* 每个Node类及其子类都可以使用定时器，如schedule等。
* 每个Node类及其子类都可以继承动作，如runAciton、stioAction等。

**Node对象可以添加Node类的子类对象作为子节点。只要这些子节点可以显示，Node本身就可以显示。**

学习Node类的子类时需要对以下3个方面加以了解：

* 对象初始化的方式和内容
* 时间回调函数
* 重载绘制函数

**（2）场景**

**可以将场景理解为一个容器，这个容器对游戏界面的管理发挥重要作用。场景将不同的图层（Layer）组织在一起呈现给玩家一个完整的游戏画面。**

在游戏中，节点是一个很大的范围，因为无论场景，图层还是精灵，从本质上来说都是节点，因为它们都是节点的子类。但是场景的范围就要相对小得多了，一般来说，**场景是作为场景切换的基本单位而存在的。**

Cocos2d-x中的场景分为3类：

* 游戏内容场景：主要展示游戏中的内容如地图、怪物等。
* 选项类场景：如游戏设置界面、关卡选择界面等。
* 展示类场景：如游戏的开场动画、RPG游戏战斗结束后的胜利信息等。

在Cocos2d-x中创建一个场景的代码如下：

auto scene = Scene::create(); // 创建一个场景对象

auto layer = HelloWorld::create(); // 创建HelloWorld对象

scene->addChild(layer); // 将layer添加到场景scene中

return scene; // 返回创建好的场景对象

可以将场景理解为游戏状态切换的基本单位。Cocos2d引擎为场景的切换提供了丰富的切换效果。实验1-2是一个用于展示Cocos2d场景切换效果的项目。

其中负责切换场景的代码如下：

Director::sharedDirector()->replaceScene(HelloWorld::createScene());

其中HelloWorld是所要切换的场景所要的类，在切换场景时导演会自动地销毁当前场景并创建新的场景来实现切换的效果。

还可以使用带有动画效果的场景切换，在类HelloWorldScene中定义了15个用于场景切换的方法，分别对应屏幕上的15个菜单按钮，具体实现代码如下：

Director::sharedDirector()->replaceScene(TransitionFade::create(3.0f, MyWorld::createScene()));

其中3.0f是切换动画完成的时间，MyWorld是所要切换的场景所在的类，TransitionFade是选用的切换动画效果。表2-2给出了Cocos2d-x所提供的切换效果。

|  |  |
| --- | --- |
| **类名称** | **说明** |
| TranstionRotoZoom | 当前场景以旋转加缩放的形式退出到屏幕外，新的场景以同样的形式代替元场景占据屏幕内容 |
| TranstionJumpZoom | 缩小跳着切出场景，同时跳着放大传入场景 |
| TranstionMoveInL | 从左侧传入场景 |
| TranstionMoveInR | 从右侧传入场景 |
| TranstionMoveInT | 从顶部传入场景 |
| TranstionMoveInB | 从底部传入场景 |
| TranstionSlideInL | 从左边滑入传入场景 |
| TranstionSlideInR | 从右边滑入传入场景 |
| TranstionSlideInB | 从底部滑入传入场景 |
| TranstionSlideInT | 从顶部滑入传入场景 |
| TranstionShrinkGrow | 当增长传入场景的时候，收缩传出的场景 |
| TranstionFlipX | 水平翻转屏幕，正面是传出的场景，背面是传入的场景 |
| TranstionFlipY | 垂直翻转屏幕 |
| TranstionFlipAngular | 水平垂直翻转一半屏幕 |
| TranstionZoomFlipX | 水平翻转屏幕，做一个传入/穿出缩放。正面是传出的场景，背面是传入的场 |
| TranstionZoomFlipY | 垂直翻转屏幕，做一个传入/穿出缩放。正面是传出的场景，背面是传入的场景 |
| TranstionZoomFlipAngular | 一半水平一半垂直传入/穿出翻转，并一点点地缩放屏幕，正面是传出的场景，背面是传入的场景 |
| TranstionFade | 淡出传出场景，淡入传入场景 |
| TranstionCrossFade | 两个场景交叉淡入淡出 |
| TranstionTurnOffTiles | 随机顺序关闭淡出场景的瓦片 |
| TranstionSplitCols | 奇数列向上推移而偶数列向下推移 |
| TranstionSplitRows | 奇数行从左侧推移，偶数行从右侧推移 |
| TranstionFadeTR | 从左下角到右上角淡出场景的所有瓦片 |
| TranstionFadeBL | 从右上角到左下角淡出场景的所有瓦片 |
| TranstionFadeUp | 从下向上淡出场景的所有瓦片 |
| TranstionPageTurn | 翻页过渡效果 |

**（3）图层**

场景是游戏状态切换的基本单位，而每一个场景都是由一个个图层组成的。可以这么说，玩家所看到的每一个场景实际上都是由若干个图层组合而成的。

游戏“植物大战僵尸”中，要让僵尸在草地上行走，就需用两层：一层用来显示背景的草地，然后在草地上面再加入一层用来显示“僵尸”行走的动画。实际开发中至少要分层三层，因为还要留出一层用来显示分数、消息等功能。

【说明】引入分层后，可以使2D游戏中物体之间的遮罩关系处理变得简单。

图层类在Cocos2d-x开发中要完成的任务分为3类：

* 接收用户操作，比如单击、触屏等
* 作为显示内容的容器，比如在图层中显示游戏角色、选项、文字等内容
* 作为游戏背景使用

相对于场景，图层类增加了对用户交互的处理，并且使多个图层可以在一个场景中共存。任何一个图层都必须要加入到场景中才能发生作用。

**（4）精灵**

精灵是在Cocos2d-x中玩家所能够看到的最基本的单位。

在游戏中，一个妖怪是一个精灵、一个装备是一个精灵、从飞机上发射出来的一个子弹还是一个精灵。

**（5）实验1-2，Cocos2d场景切换**

**（6）实验1-3，Cocos2d场景切换效果**