Cocs2dx-x核心概念

一、基本概念

1. 导演

导演类Director用于管理场景对象，采用单例设计模式，在整个工程中只有一个实例对象。

auto director =Director::getInstance();

导演的职责：

1. 访问和改变场景
2. 访问Cocos2d-x的配置信息
3. 暂停、继续和停止游戏、
4. 转换坐标
5. 场景（派生于Node类）

构成游戏的界面，可分为

展示类场景。播放视频或简单地在地图上输出文字，来实现游戏的开场介绍，胜利，失败提示、帮助介绍。

选项类场景：主菜单，设置游戏参数等。

游戏场景：游戏的主要内容

1. 层（派生于Node类）

写游戏的重点。层的管理类类似于PS中的图层

1. 精灵（sprite，派生于Node类）

包括敌人、静态物体和背景等

1. 菜单

提供操作的集合，派生于Layer

每个菜单都有三种基本状态：正常，选中和禁止

二、Node中的重要操作

1.创建节点：

Node \*childNode=Node::create();

2．增加新的子节点

node->addChild(chileNode,0,123);

3．查找子结点

Node \*node =node->getChildByTag(123);

4.删除子节点,并停止该节点上的一切操作

node->removeChildByTag(123,true);

5.通过Node指针删除节点

node->removeChild()

1. 删除所有节点，并停止这些节点上的一切动作

node->removeAllChildrenWithCleanup(true);

1. 从父节点中删除node节点，并停止该节点上的一切动作。

node->removeFromParentAndCleanup(true);

三、Node中的重要属性

1.position与anchorPoint

anchorPoint 是相对于position的比例，默认值是（0.5,0.5）//中心在横纵坐标的占比

2.Node中调用的主要函数//游戏循环与调度

(1)void schedule Update(void)

每个函数只要调用这个函数，那么这个Node对象就会定时地每帧回调一次自己的update(float dt)函数。

(2)void schedule (SEL\_SCHEDULE selector，float interval)

与（1）函数功能一样，不同的是可以指定回调函数（通过selector指定），也可以更改需要指定回调时间间隔。

(3)void unscheduleUpdate(void)

//停止update(float dt)函数调度

(4)void unscheduled(SET\_SCHEDULE selector)

可以指定具体函数停止调度

(5)void unscheduleAllSelectors(void)

可以停止调度

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. update (float dt)通常需要自己定义，可以在其中设定物体的位置变换使得物体动起来
2. 使字体动起来的一般步骤是：
3. 创建一个label对象并设置字体，
4. 使用label->setTag(123)给label确定位置，如123
5. 在HelloWorldSCene.cpp中的init函数中只需调用一次schedule Update()即可
6. 新建一个update(float dt)函数，在其中定义诸如

auto label1 = this->getChildByTag(123);来找到最前位置上存的label(通过标签找到label)

(5)label1->setPosition(label1->getPosition() + Vec2(0, -1));

设定123位置上label1的位置

(6)可以在void HelloWorld::menuCloseCallback(Ref\* pSender)

函数中调用unscheduleUpdate();停止对update的调用

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

3.Node中调用的主要函数//世界坐标与模型坐标的转换

(1)Vec2 convertToNodeSpace(const Vec2& worldPoint)

将世界坐标转换为模型坐标

(2)Vec2 convertToNodeSpaceAR(const Vec& worldPoint)

将世界坐标转换为模型坐标。AR表示相对锚点

(3)Vec2 convertTouchToNodeSpace(Touch \* touch)

将世界坐标触摸点转换为模型坐标

(4)Vec2 convertTouchToNodeSpaceAR(Touch \*touch)

将世界坐标中触摸点转换为模型坐标。AR表示为相对于锚点

(5)Vec2 convertToWorldSpace(const Vec2& nodePoint)

将模型坐标转换为世界坐标

(6Vec2 convertToWorldSpaceAR(const Vec& nodePoint)

将模型坐标转换为世界坐标，AR表示相对于锚点

四、调整屏幕宽度

在AppDelegate::applicationDidFinishLaunching(){}中，if(!glview){}中，director->setOpenGLView(glview);前

添加glview->setFrameSize(900,640);//设置窗口大小