时间调度 schedule

2014-2-17

1.Update函数

在Cocos2d-x中的每一帧执行事件，只有Update是不行的，还要在init()函数里面加上scheduleUpdate()，这样才会每一帧都调用update()。

2. Schedule函数

带时间参数的，间隔指定时间，然后执行指定的函数，达到定时刷新的效果。

不带时间参数的，每一帧刷新，执行指定的函数，作用效果就和scheduleUpdate() 函数差不多效果，不过函数可以自己指定。

不带时间参数的写法(写在init()中)，这样每一帧执行update() 函数

schedule(schedule\_selector(CocosRakNet::update));

带时间参数的写法，每隔指定时间执行update() 函数

schedule(schedule\_selector(CocosRakNet::update),3.0f);

注意这里的时间间隔是 秒  为单位

如何让只执行一次函数？

**this**->scheduleOnce(schedule\_selector(HelloWorld::Update), 5.0f);

这样就会延迟5秒之后执行一次Update，而且只执行一次！

那如何停止schedule ？

1.停止scheduleUpdate()关联的update()函数

unscheduleUpdate();

2.停止schedule() 关联的自己的函数

schedule(schedule\_selector(CocosRakNet::update),3.0f,10,5.0f);

unschedule(schedule\_selector(CocosRakNet::update));

3.停止所有的schedule，不管是自己的还是默认关联的

//更新函数

scheduleUpdate();

schedule(schedule\_selector(CocosRakNet::update));

schedule(schedule\_selector(CocosRakNet::update),3.0f);

unschedule(schedule\_selector(CocosRakNet::update));

unscheduleAllSelectors();

在游戏中，时常需要隔一段时间更新一些数据或者是人物位置，Cocos2D-x中提供了这些时间调度的函数，所有CCNode类的子类都有这样的函数，定义方法如代码清单3-50所示。

schedule(schedule\_selector(SchedulerAutoremove::autoremove), 0.5f);

这是一个按时调用一个函数的方法。第一个参数使用schedule\_selector选择器将autoremove函数名称传进来。第二个参数是时间间隔。定义这个参数以后就会隔一段时间调用一次该函数，直到unschedule被调用。

unschedule(schedule\_selector(SchedulerAutoremove::autoremove));

这句被调用，之前schedule的时间调度将结束。使用如代码清单3-52所示的代码分别暂停并重新启动schedule。

//暂停schedule

m\_pPausedTargets = pDirector->getScheduler()->pauseAllTargets();

CC\_SAFE\_RETAIN(m\_pPausedTargets);

//重新启动schedule

pDirector->getScheduler()->resumeTargets(m\_pPausedTargets);

CC\_SAFE\_RELEASE\_NULL(m\_pPausedTargets);

使用unscheduleAllSelectors()或者如代码清单3-53所示的代码，可以使所有schedule停止。

有一种固定的调用方式，就是使用scheduleUpdate会在0.01s左右调用一次update方法，只要重载update方法即可。