# 实验目的

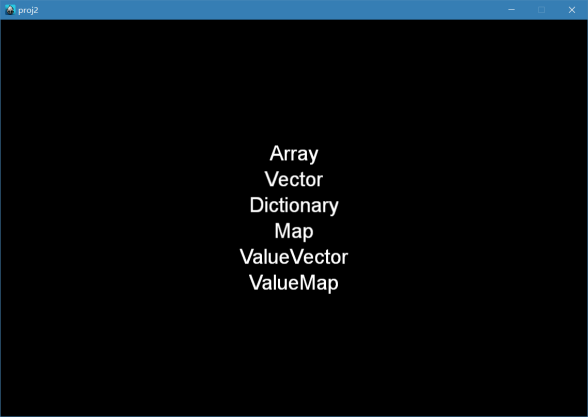
熟悉并掌握cocos2dx中的数据结构类。

# 实验内容

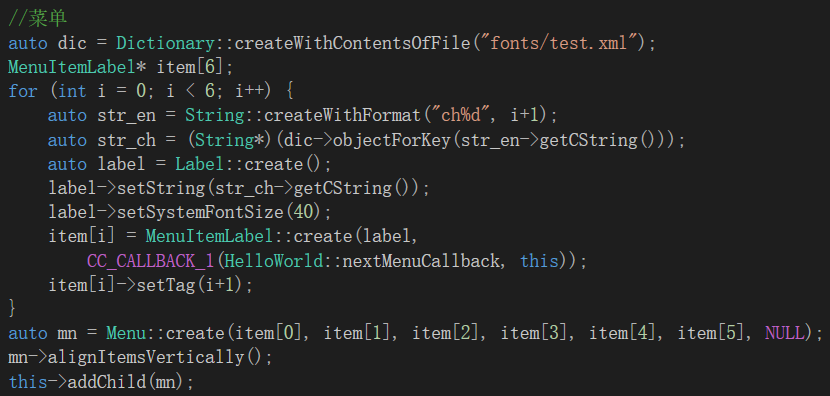
根据数据结构类的讲义及核心代码，**用多种方法实现两个物体点击随机出现的小项目**，着重理解REF类和c++11的类的内存管理机制及其区别，对于容器和字典类，特别要注意容器元素和字典值对象的类型与容器和字典本身类型的关系。要求上交实验报告（内附相关代码说明）。

# 实验步骤

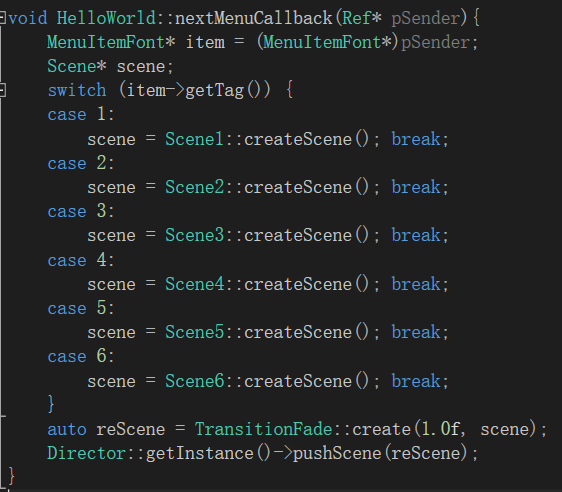
首先创建一个菜单场景如下图，然后通过菜单场景进入六个子场景



init函数创建菜单

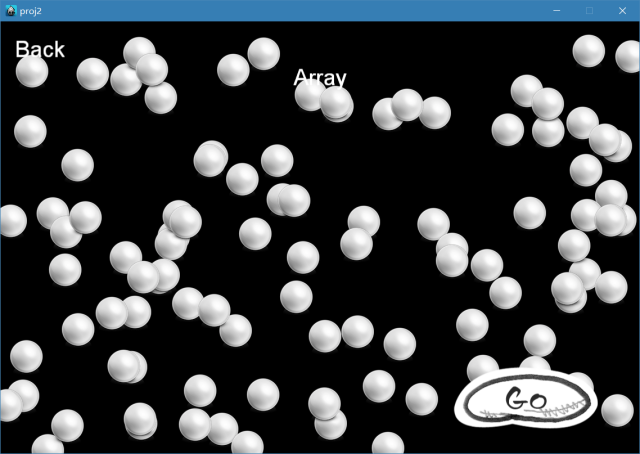


回调函数



## Array

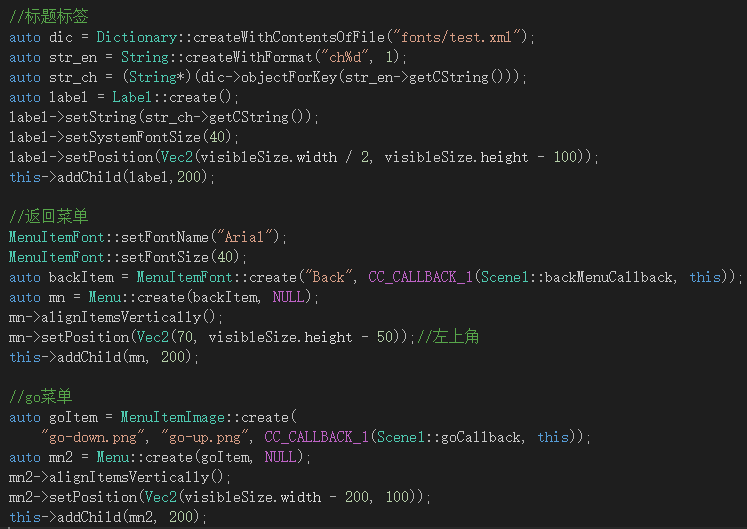
效果图



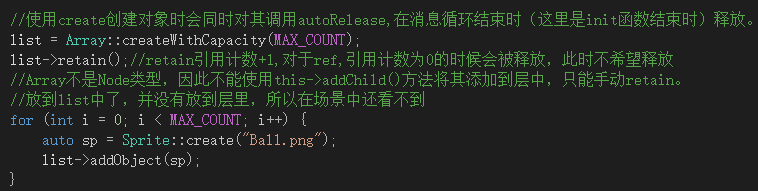
Scene1.h



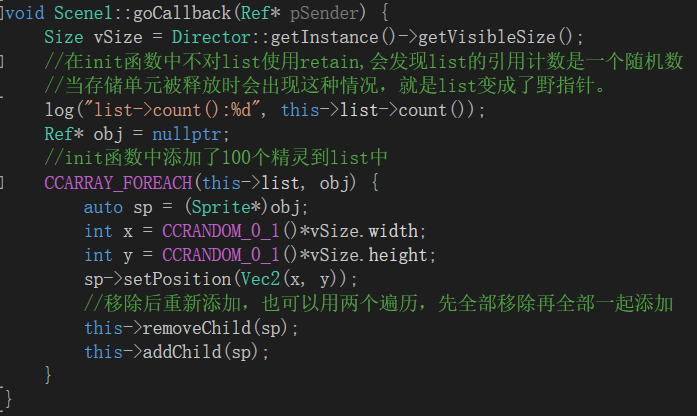
init函数中标签和菜单



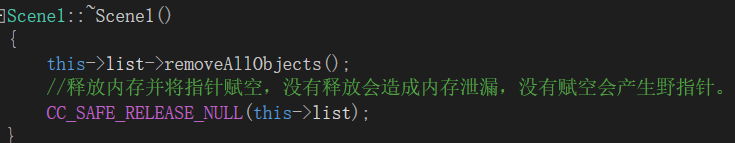
Init函数中将精灵添加到Array



go回调函数



析构函数

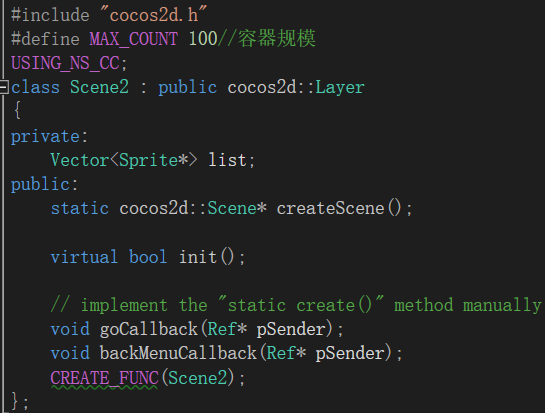


## Vector

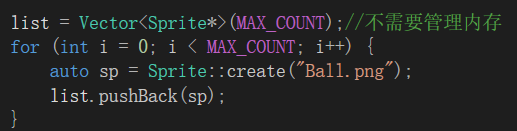
效果图

## 

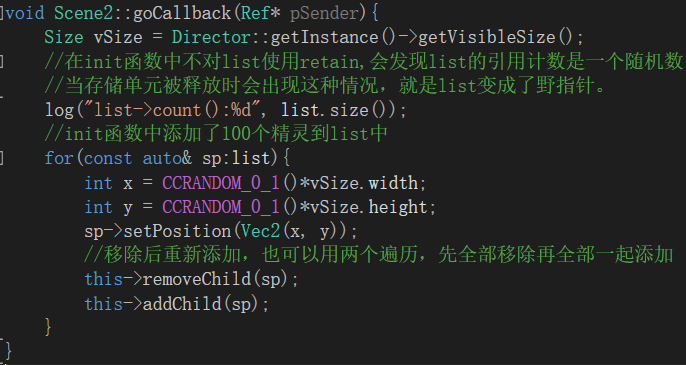
Scene2.h



init函数

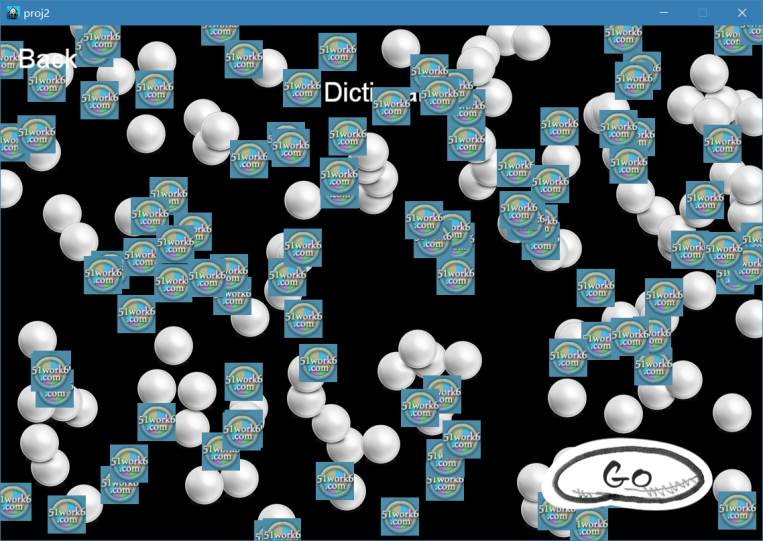


go回调函数

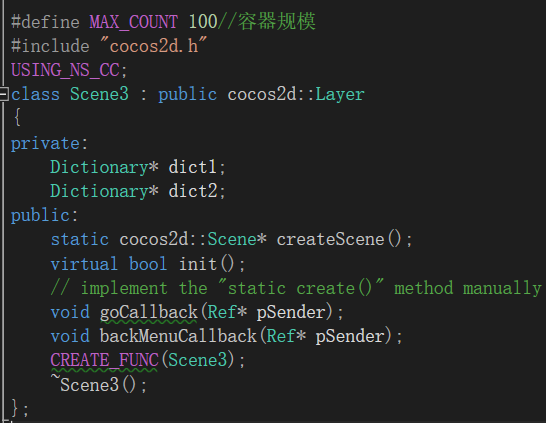


## Dictionary

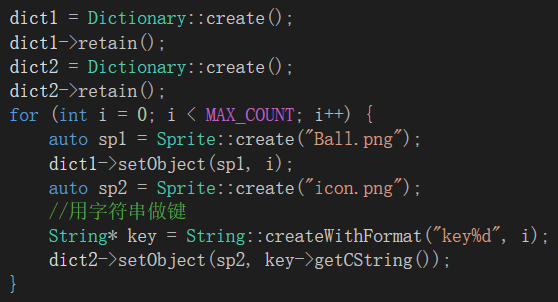
效果图



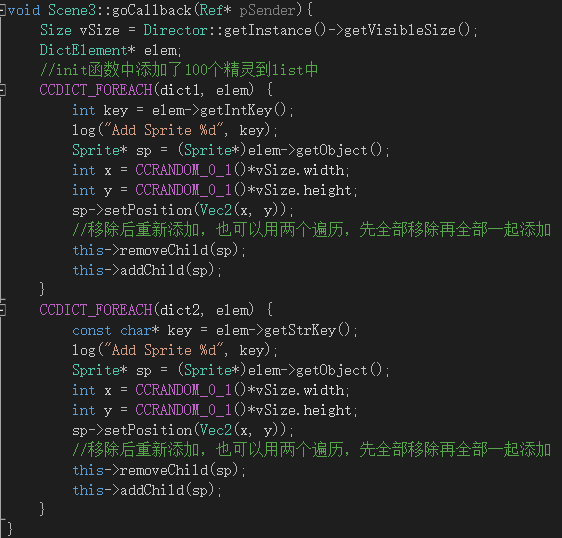
Scene3.h



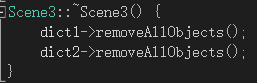
init函数



回调函数

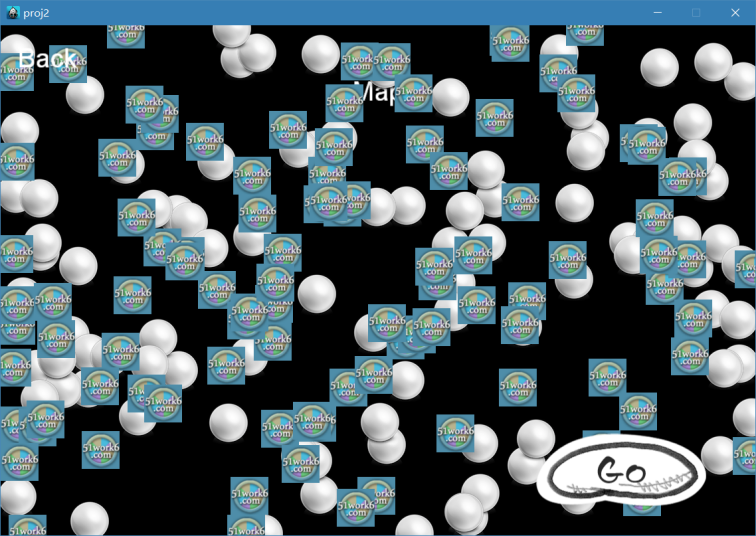


析构函数

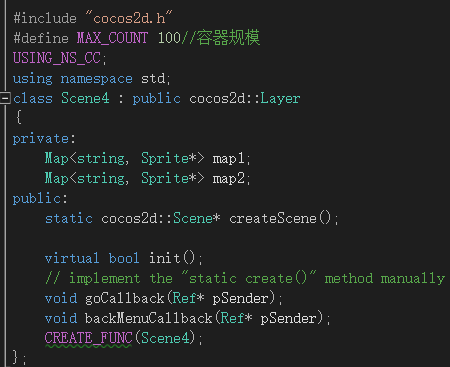


## Map

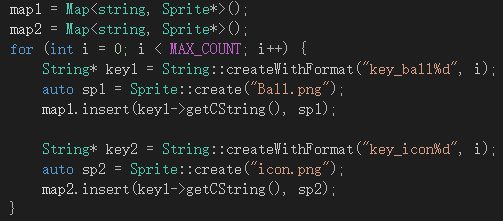
效果图



Scene4.h



init函数



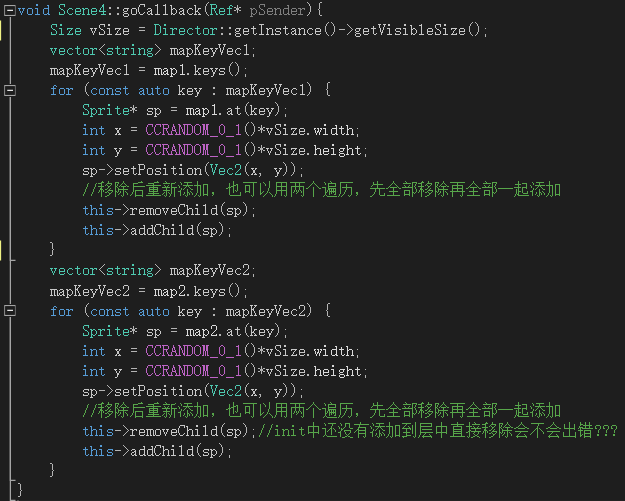
回调函数

### 问题：使用removeChiled函数时，如果没有该子对象会怎么样

查看源码会有输出信息

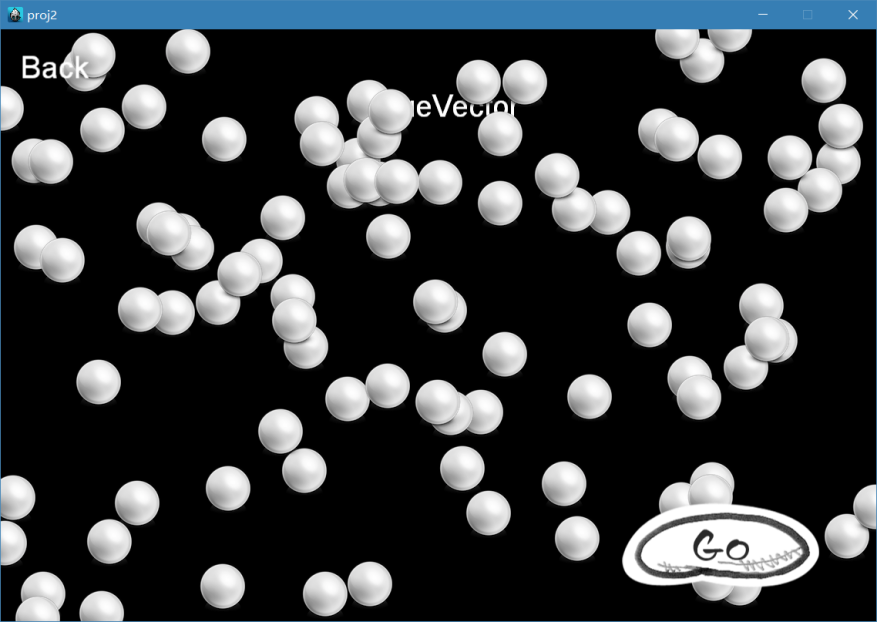






## ValueVector

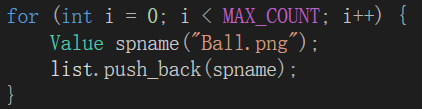
效果图



Scene5.h



init函数

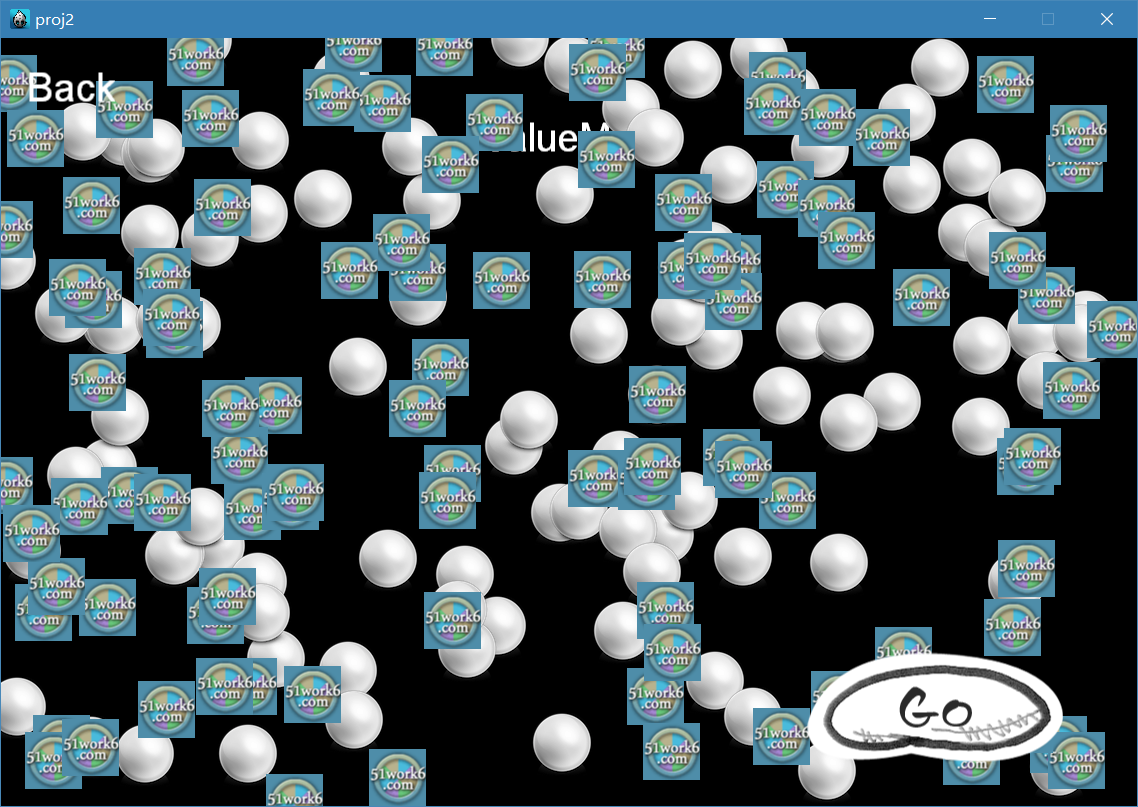


回调函数

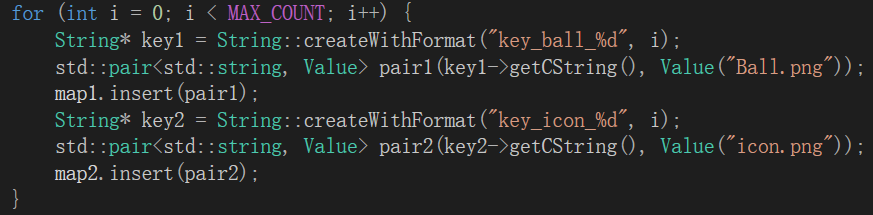


## ValueMap

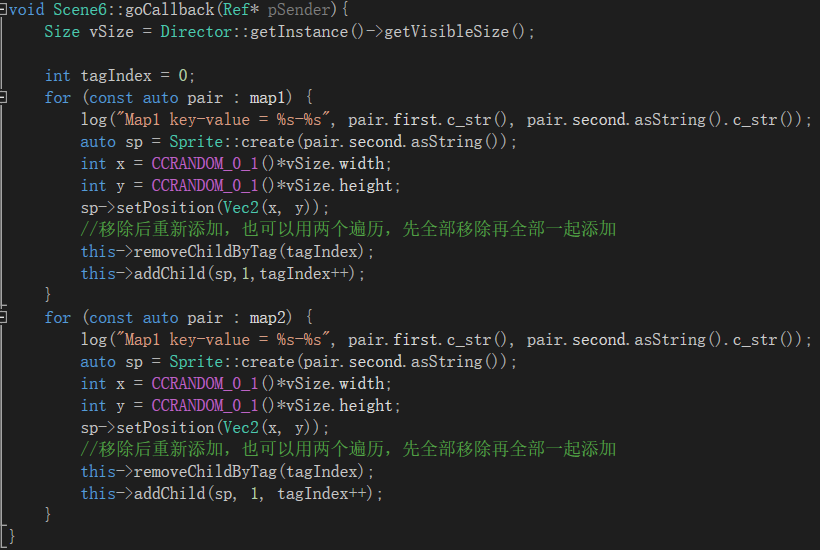
效果图



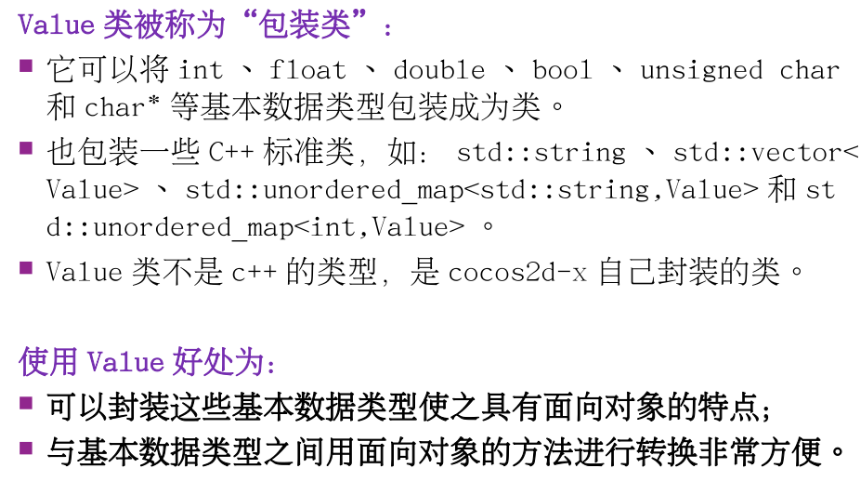
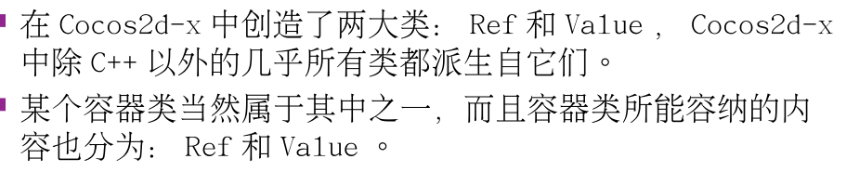
init函数



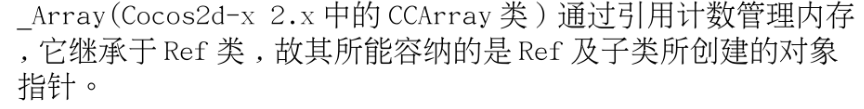
回调函数

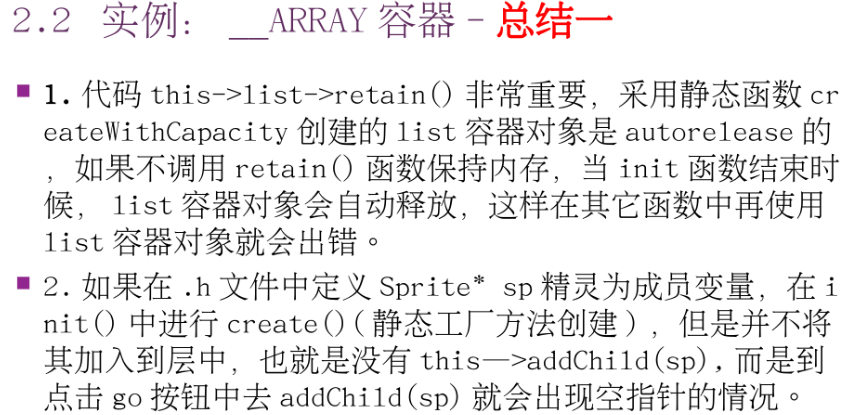


# 实验心得



## Array





### 问题：如果是这样，那么Init函数岂不是每一帧都要绘制一次（是的）。

有些元素是不需要重新绘制的（位置大小没有发生改变），每帧都绘制一次岂不是会进行很多不必要的计算,比如创建精灵然后把精灵添加到层中。

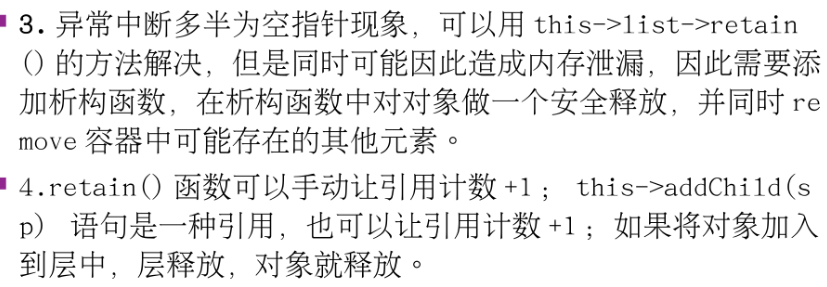
### 问题：当init函数结束时，其子对象list容器会被自动释放，那么监听器、精灵这些怎么解释？Layer的成员变量不是在层的所有函数中都可以访问吗？？

Create新建一个对象时，其引用计数初值为1，但是同时会调用autoRelease，在消息循环结束时该对象就会被释放。

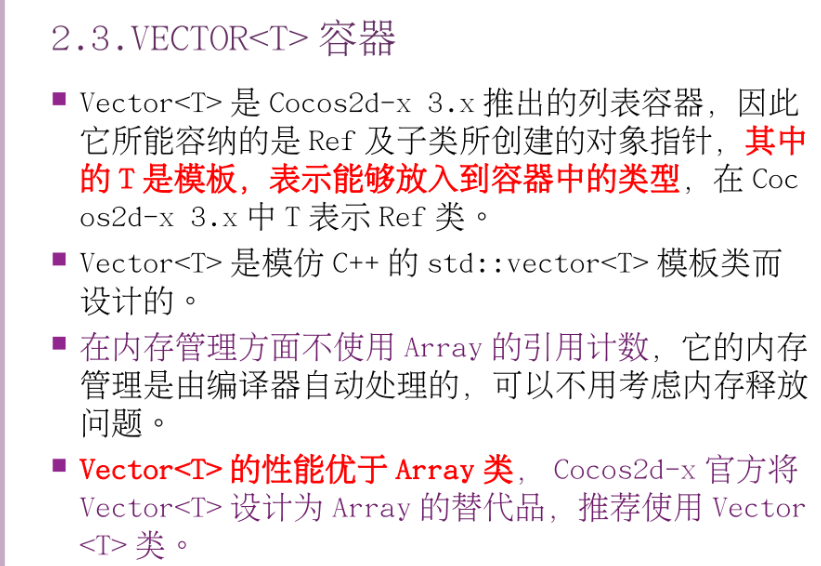
监听器、精灵这些在init结束后没有被释放是因为添加成为了层的子对象，隐式调用了retain。

所以这里也可以把Array添加为层的子对象，这样就不会在init结束时就被释放了。实际上不行，Array不是Node类型，因此不能使用this->addChild()方法将其添加到层中，只能手动retain。

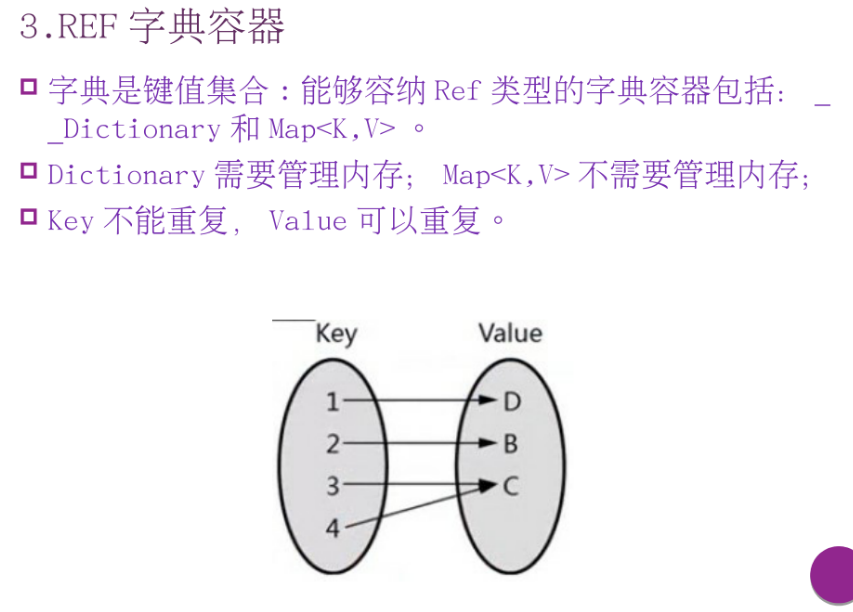
Cocos2dx引擎一帧中做的事：<https://blog.csdn.net/dooms_guardian/article/details/52610511>



## Vector



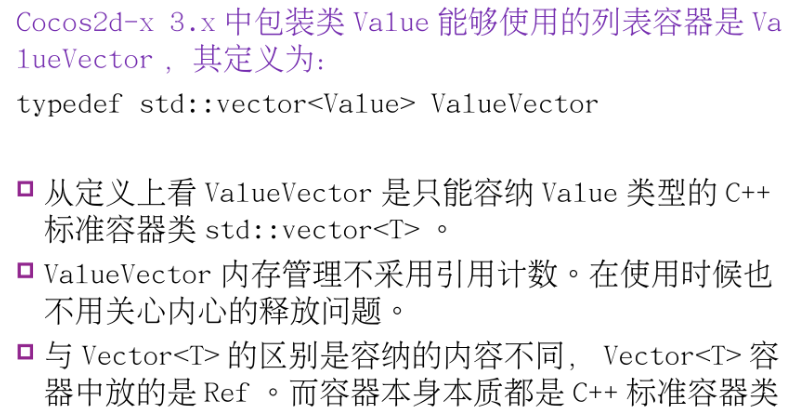
## Dictionary



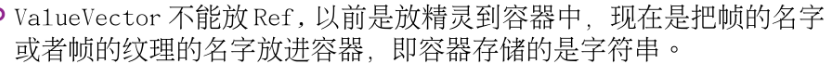
## Map



## ValueVector







## ValueMap

