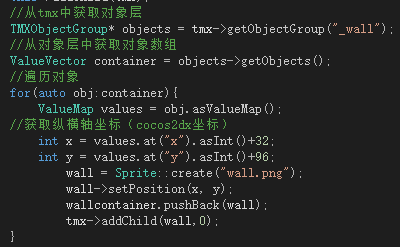
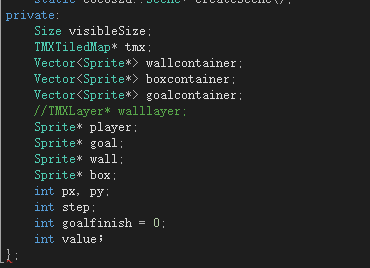
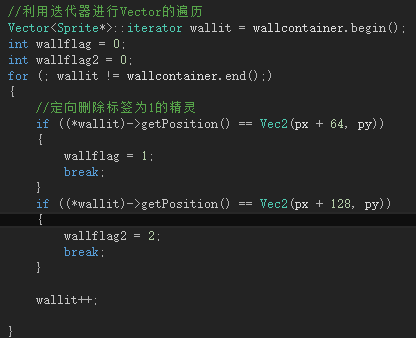
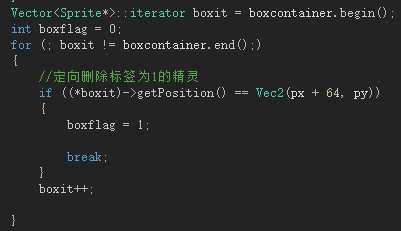
1. **步骤**
   1. 利用demo所给的图片和tmx地图文件，制作地图。
   2. 通过遍历对象和定义sprite显示地图，下图是显示tmx地图里wall的代码，显示box，player和goal的代码基本相同。



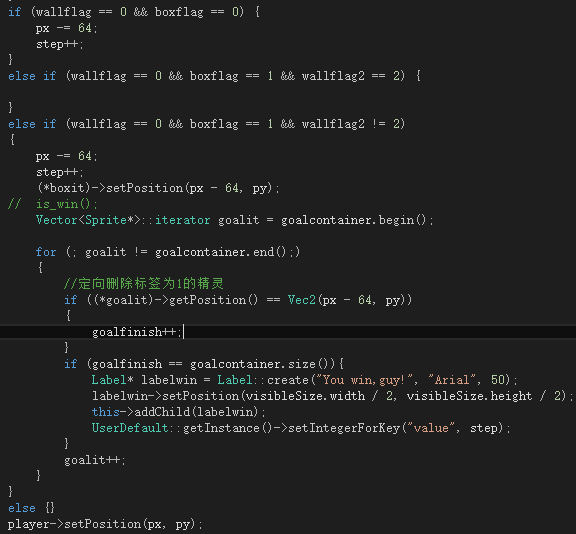
Pushback操作是我之前定义 了一个vector来存储这些对象，以便后面移动操作的位置判断

* 1. Touch按钮的UI设置demo中已给出
  2. 编写点击各个方向按钮的函数，右图为头文件的成员定义：

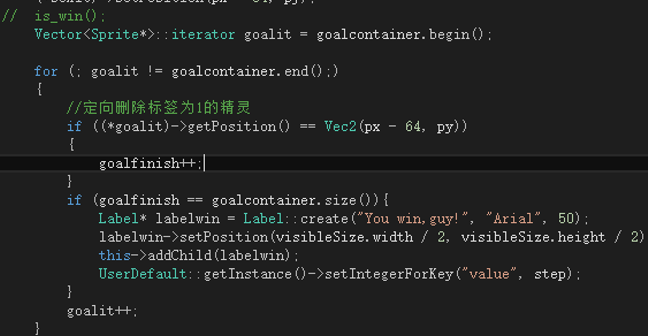


首先利用迭代器对之前存储有各种对象的vector进行遍历，我使用（px，py）作为player的坐标，通过遍历所有的wall和box对象，判断player的下一步位置不是wall和box时便让移动标记wallflag和wallflag2获得相应值。

然后对box对象遍历判断箱子的下一步是否为wall和player的下一步是否为box，从而赋予移动标记boxflag相应值。



最后，根据之前获得的一些移动标记的值判断player应该如何移动。

胜利的判断条件设在player移动箱子到goal后面。通过比较box与goal位置相同的数量的值和boxcontainer.Size（）的值是否相等来判断是否胜利。

当胜利时，新建一个label提示胜利，并把移动步数通过userdefault存储到userdefault.xml的value里；

* 1. 在每一个按钮函数里，只要发生移动，就有一个记录步数的step++；
  2. 游戏部分截图：

