超级玛丽详析

超级玛丽是我从暑假就开始做的，感觉难度比斗地主游戏更大，主要表现在处理动画以及程序的架构方面。刚开始做的时候很长一段时间都不知道从哪里下手，后来干脆不想了，直接开始写，但是因为一开始写没想清楚，架构很混乱，漫无目的的写，具体的一个个算法都花了不少时间，但是之前写的算法到最后因为改动又要重新写过。后来和高莺一起对照网上一个人的方法讨论了很多次，想着干脆重新写一遍，谁知道写着 写着发现竟然是这么简单，后悔当时浪费的那么多时间。 最后三个人的超级玛丽从流程到具体算法都十分不一样，并且我的超级玛丽已经不像超级玛丽了，我要把它做成一个跑酷游戏了

哈哈 哈哈哈，如果做成跑酷的游戏的话体验应该比超级玛丽好。

我的超级玛丽有两大优点：

1：架构清晰，代码简洁

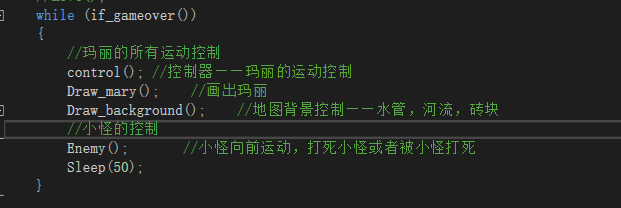
2：动画处理流畅

接下来分结构，数据，操纵控制三个方面总结。

# 程序结构部分

说到底我的结构就两块——Control and Draw

控制器控制玛丽的参数变化画图函数根据参数画出玛丽。



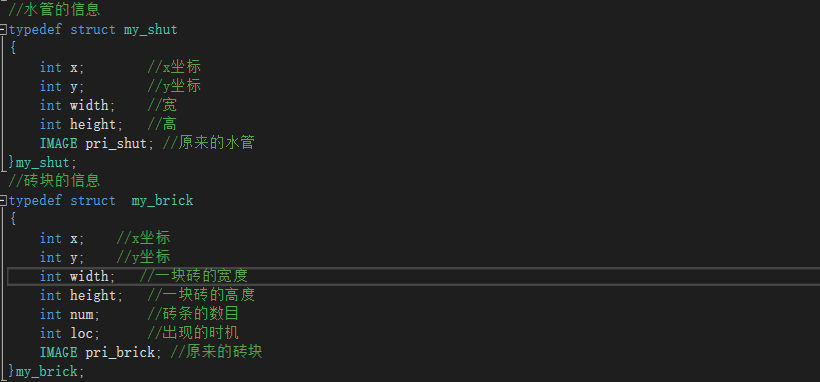
# 数据处理部分

玛丽的数据不用多说，自然是用结构体存储其各种属性，难点在于地图背景数据的处理方式

我了解到的绝大部分是用二维数组存储砖块，河流，水管等各种背景环境，使用这种方式只要随着玛丽的运动更新数组的内容即可。但是我采用的是另一种方法。

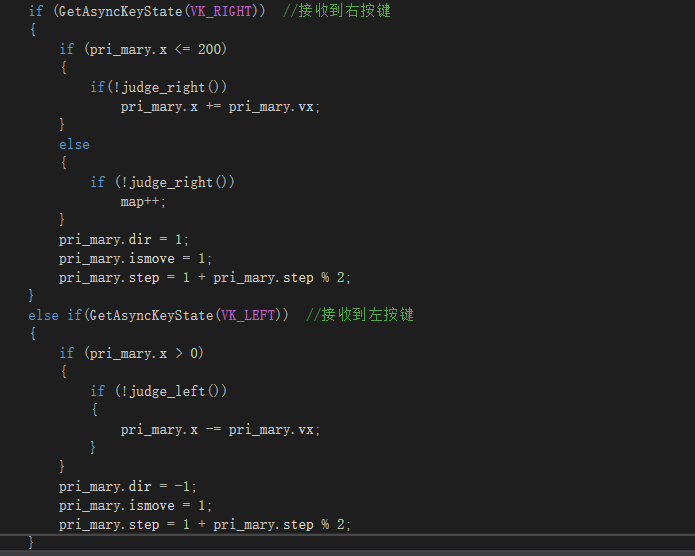
我将砖块，水管都单独划分出来建立一个结构体，整张地图的所有水管和砖块分别存在相应的vector容器里，时机到了则出现，时机过了则删去，不浪费空间。

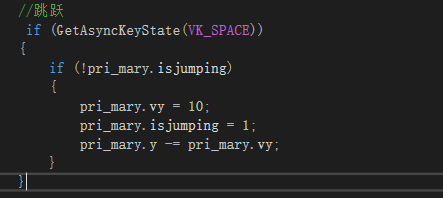
我这样的好处是对地图背景的改变较为简单，可以随意安排水管砖块的位置，并且不会闪屏。



# 动作处理部分

一般的做法是一次读取两个键盘输入形成一次方向跳跃，但是这样的话还要区分单独跳跃，单独跳跃后前进，单独跳跃后后退，这样的话横向坐标很混乱。我直接将横向的坐标控制和跳跃控制分开，两方互不影响，因此可以做到跳跃时也能前进后退（冥思苦想很久，最后的解决方法仅仅是将跳跃的else if 改成 if）





我就总结出了一个道理：

把流程和数据搭好了剩下的算法不过是体力活而已