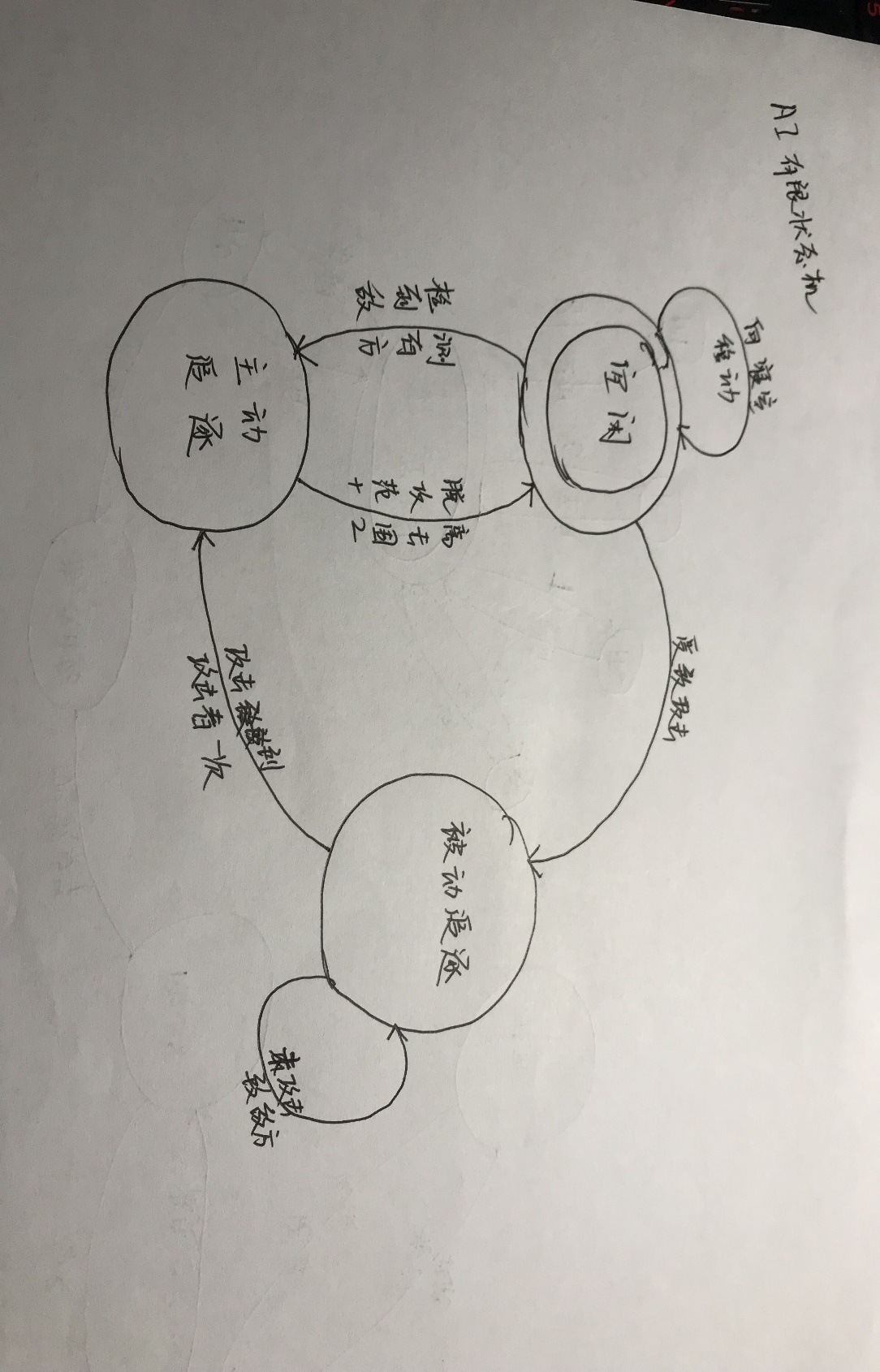
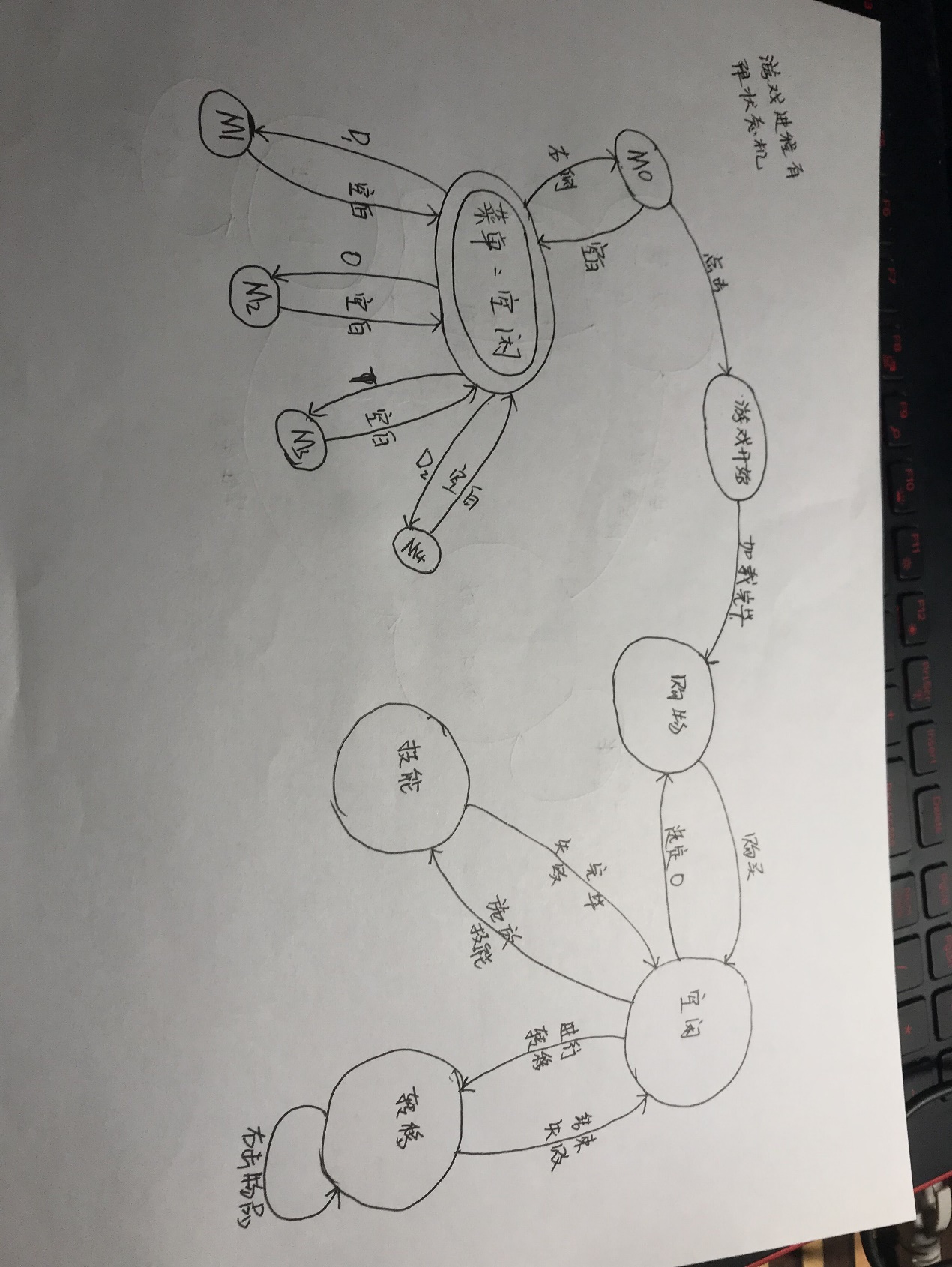
游戏的整体框架是有限状态机+多线程并行

游戏中使用了两类有限状态机 分别是AI有限状态机和游戏进程（鼠标）有限状态机

AI状态转移图：



游戏进程状态转移图：



然后 每个玩家英雄会拥有大约6-7个子线程

每个AI英雄会拥有3-4个子线程

寻路方面依然是使用的A\*寻路

在代码中对应Astar类

开启列表是用最小堆 也就是代码中的MinHeap类

关闭列表因为无需进行排序 采取了可变数组 对应的Aector类

采用虚继承模拟有限状态机

因为很多地方不得不使用虚继承 所以代码量会有些虚高……

而且很多装备描述 不得不采用Switch case 也使代码看起来有点多

识别方面采用RTTI

采用\_beginthreadex来开启线程

线程通信方面

采用 信号量（Event）来确保事件响应的同步

采用互斥量（Mutex）来确保写入不发生冲突

但是 由于采用高并发的框架 部分事件响应可能有些怪异

除了播放音效不得不采用MFC的mciSendstring函数

其他方面均采用API 原则上不调用第三方库

总体理念是

输入事件（键盘/鼠标）->鼠标有限状态机->分发到线程->做出响应->玩家英雄与AI英雄交互->做出应答