问题求解(二)课程项目-第一阶段项目报告

林朗

完成情况：

2.1 使用 git：共有 7 次 commit 记录，commit 历史将在视频中展示；

2.2 地图：游戏设定为 20*20 的地图，地图为随机+check 的模式生成，开始时每一格有 2/10 的概率生成硬墙，3/10 的概率生成软墙，地图中与人物初始位置相邻的两格固定为空地。在地图生成结束后会跑一遍 dfs 判断四个角（也就是双玩家+双 robot）的初始位置是否连通（此时只有硬墙是不可走的，软墙和空地都视为连通），若不连通则重新生成；

2.3 玩家：玩家以字符'1''2'表示，初始时在地图左上和右上角。玩家内置一个 moveInterval 变量，当两次移动命令的间隔大于 moveInterval 时才执行移动命令，否则不执行。moveInterval 可随游戏情况改变；

2.4 机器人：机器人以字符'3''4'表示，初始时在地图左下和右下角。

机器人算法：

放炸弹：满足以下两个条件时机器人会放置炸弹：a.机器人在当前位置放置炸弹可以炸到软墙或者玩家/其它的机器人；b.机器人距离上次放炸弹已经过去了 3s；

后续会考虑优化解决机器人把自己炸死的问题；

移动：尝试向四个方向走和不动五种策略，利用赋权值的方法找出最优解（保命为先，锁敌攻击为后），并且考虑以下两种特殊情况：

a. 待在原地不动马上就会死，抛弃原地不动的策略；
b. 有多种策略的权值相同且最小（设定最小最优），那么就先锁定离自己曼哈顿距离最小的玩家/其它机器人，用 bfs 跑一遍最短路，如果最短路的第一步移动对应的权值最小，那么就走这一步；

具体算法会在视频中展示；

2.5 炸弹：炸弹用'!'表示，光束用'-''|'表示，其它均已完成；

顺便在这里解释一下本阶段项目的定时事件处理：每个定时时间都存储在对应的 class 而非一个定时队列中，count 仅作为判定是否处理定时事件的单位时间用（提升效率）。该方法效率似乎不如用队列来解决，但思考方式更为简单，将考虑在下一阶段中优化该问题；

2.6 道具：在生成地图时，每个软墙后有 1/10 的概率生成道具，两种道具概率对半分，用一个专门的数组 propsType 存储相关信息。玩家/机器人炸开软墙后道具显示，吃了后获得对应功能；

2.7 得分显示：显示在下方，炸开一个软墙得 1 分，吃一个道具得 10 分，基于游戏公平性及人道主义角度考虑，炸死人不得分。

总而言之就是都完成了。

遇到的困难：

机器人设计问题难以解决，main 函数内繁杂冗长，不知道计时事件如何处理等；

最大的问题莫过于我不知道多个.h 文件和对应的.cpp 文件可以相互调用，导致我刚开始的时候写了一堆东西在 main 函数里……

以及屏闪问题，还没有解决。

实验心得：

还是先把面向对象的框架学好再来写代码，说实话我现在看到我这个 main 函数还是挺头疼的，但是改的话其实也好不了多少，很大程度上是结构不太漂亮，准备第二阶段进行一个大改；

以及没玩过泡泡堂写这个项目真的很难受……四处问些乱七八糟的问题，现在也不知道究竟还有没有游戏原本性冲突，只能说这个游戏设置本身我看的比较赏心悦目了。

致谢：

朱宇博助教，王远博同学，邓振霄同学，胡皓明同学

感谢助教和同学们对我有关代码乃至游戏本身的各种低智问题的耐心解答。