

## 问题求解（二）课程项目 第三阶段 项目报告

从 2022 年 5 月 22 日至 2022 年 7 月 1 日，使用 C++ 语言，以类和对象为载体，使用图形化框架 QT，完善的泡泡堂游戏功能及图形界面并使用 git 进行课程项目的版本控制。

完成进度：

- **使用 git 管理代码** 将第二阶段项目本地对应的远程仓库重定向至第三阶段仓库。更加熟悉 git add . git commit git push 的用法。

- **进一步优化** 在已经完成第二阶段全部任务的基础之上，进行了细节上的优化。人物走动时动态效果有时不展现的 bug 已经被解决（由 getImage 函数 不同的 stage 切换不同图片）

```
const char *Player::getImage(int id, int dir, int sta)
{
```

增加了摆臂的图片以更好地达到动态效果。增加背景音乐种类，除了主背景音乐，在放炸弹、爆炸、捡道具、胜利等有不同的音效和音乐。修改了计分板有时候的 bug。

```
if (yes == 1)
{
    auto musictool = new QSoundEffect;
    musictool->setSource(QUrl::fromLocalFile(":/res/tool.wav"));
    musictool->setVolume(0.15f);
    musictool->play();
}

if (item->getComponent<Bomb>()->get_x() == idy && item->getComponent<Bomb>()
{
    auto musictool = new QSoundEffect;
    musictool->setSource(QUrl::fromLocalFile(":/res/movebomb.wav"));
    musictool->setVolume(0.15f);
    musictool->play();
    item->getComponent<Bomb>()->hit = dir;
    break;
}
```

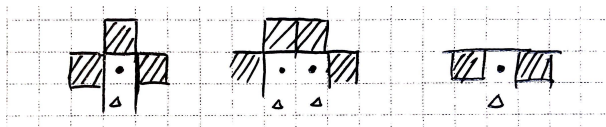
- **机器人** 修改第二阶段的机器人，使其具有一定的“智能性”，由于之前行走和炸弹等都是通过维护二维数组实现的，因此就利用二维数组实现智能性。新在 map 类中添加两张相同大小的地图 `int NOWTool[15][20]={}` `int NOWBomb[15][20]={}` 分别记录地图上出现的道具、正在爆炸的地方、

人物移动的位置，之后通过 bfs 搜索路径，然后通过 path 记录路径节点、pathdir 记录每一步走的方向。由于想还原原游戏的操作手感，并没有采用离散型行走（按一下按键走一格），所以在记录路径上会显得比较复杂 `if (Find_Tool(i + idy, j + idx, 0))` 若成功找到路径，则放入属于 robot 的记录路径的 vector 中

当 tempDiraction 不为空时，就先走该 vector 中储存 `for (auto it : tempDiraction)` 的路线，若为空，则继续执行随机游走。 `Diraction.emplace_back(it);`

```
int n = QRandomGenerator::global()->generate() % 4 + 1; (auto it : tempLocation)
while (n == walk) Location.emplace_back(it);
    n = rand() % 4 + 1;
```

在第二阶段的基础上，增加了机器人也可以捡道具，所以对于不放炸弹把自己炸死、控制自己的炸弹威力等做出了修改。除了间隔一定的时间放炸弹，还增加了对特殊地形的分析，如图 1，走到三面都是软墙的“死路”时，会释放炸弹。并且在炸弹数量大于 1 时，会在复杂的地形连续放几个炸弹。对于攻击人物，采用同样的方法，定位人物的位置（且找人的优先级应该大于找道具）当走到人物附近时，释放炸弹。以及在上述地形时，释放炸弹将人物锁死。



### 遇到的困难：

- **机器人入手** 开始入手机器人是非常痛苦的，慢慢阅读了网上一些 bfs 的代码，才理解了整体思路。然后发现每走一步搜索一次很不现实，因为不是离散的行走方式。就必须记录直到每个格子走完。
- **图片形状问题 导致人物移动有视觉差** 由于没有用碰撞检测函数导致视觉差。在用二维数组记录机器人和玩家的位置时，不是一进入就改变二维数组，而是视觉上进入后才改变二维数组，这样会使视觉效果好一些。
- **机器人主动攻击人物的算法** 由于没有采用很复杂的算法实现机器人的智能性，而是列举多种可能预见的情况，用特判来解决，来事先模拟机器人所遇到的环境，再做出相应判断。
- **机器人记录路线后若被干扰则会有 bug** 由于一开始采用直接定位是锁定一个目标位置之后，找到合法路径，就把路径储存下来，然后沿着记录的方向走。但如果中途路径被炸弹或光束阻碍，就无法实现。解决方法是每走一格进行一次搜索。

### 心得体会：

经过无比忙碌的考试周之后，才开始我的第三阶段，完成的有些许 struggle。和 qt 的战役，算取得阶段性成功了吧。从 qt 小白、oop 小白到完成了一个看起来还挺好看的小游戏；从第一次把 .cpp .h 文件分离、创建 class 到一步步写完几千行代码；从阅读生涩的代码框架到熟练使用接口完成操作；从新创建 github 账号、一个一个字母的敲着黑窗到熟练的 git 指令会返回到某次修改；从只会摇骰子随机游走的智障机器人到有规划路线看起来稍微智能一些的机器人... 当我已经可以面对九百行代码气定神闲的找到 bug，当我真正把 print assert 调试法运用其中是我更迅速地 debug，当我尝试把新学到的算法运用在自己的 project 里... 感觉真的学到了很多，也算体会到学期大作业的“乐趣”，也会在下次大作业中收获更多的经验，虽然还有很多做的不足的地方，但我看到我那还算好看的作品和同学快乐的玩我写的游戏时，还是十分开心你的。

### 致谢：

感谢与我交流机器人算法的同学与助教！从你们不同的方法中，我也学会了很多东西！谢谢！