姓名：赵欣玥　　学号：211240065

问题求解（二）课程项目 第一阶段 项目报告

2022 年 2 月 22 日至 2022 年 3 月 27 日，使用 C++语言，以类与对象为载体，完成字符界面下的初级版泡泡堂游戏并使用 git 进行课程项目的版本控制。

完成进度：

• 使用 git 管理代码　使用 git init 创建仓库，对文档进行修改之后使用 git add 将其添加到暂存区，再使用 git commit 将暂存区的内容提交到当前分支，在每次提交时加上文字备注当次修改内容。git remote 建立远程仓库，再将每一次的 commit 记录 git push 到 github 上。（图为 git log 查看 commit 记录，origin2,211240065--proj 代表两个远程仓库）

```
commit a60aa6438e2bccdf966bf7fb9c7ef09fe7908ff2 (HEAD -> main, origin2/main, 211240065--proj/main)
Author: zxyleaf <211240065@smail.nju.edu.cn>
Date:   Thu Mar 24 22:04:51 2022 +0800

    clean it
```

• 设置地图　为了方便，我没有设置地图类，直接采用两个 14 * 20 二维数组分别记录地图和道具，每次移动、设置炸弹都修改 MAP[i][j],涉及道具是修改 MAPTool[i][j],并且 MAP 是固定的，但每次软墙下的 1、2 道具位置通过随机数修改。

```
tool = rand() % 2 + 1;
char MAP[14][20] = {
    {'#','#','#','#',
```

• 设置玩家　定义 Player 类 使用 A、B 两个字符代表两个玩家，初始位置固定（以免初始就无路可走）

```
Player(int x, int y, char sy);
Player player1(5, 1,'A'), player2(1, 18, 'B');
```

可以直接通过键盘操作 A、B，上下左右走动、释放炸弹、捡道具。初始时速度是 1，每按一次按键移动一格，初始时每个玩家拥有三个炸弹，最多可连续释放三个炸弹。在长按键盘时，不会因为按按键的速度影响移动的速度。因为在每次读取键盘输入前，采用 clock()函数先判断了每次输入的间隔时间，若小于 Diff 则不读取该输入。

• 设置机器人　定义 Robot 类，设置两个机器人，都用 R 字符表示，初始位置固定（以免初始就无路可走）。第一阶段机器人不具有智能性，采用随机数走法，间隔一定时间调用机器人移动函数，通过随机数判断走的方向。若该方向是“*”软墙，判断是否可走路线，若有则是放炸弹，保证机器人不会把自己炸死。且该阶段机器人不会捡道具。

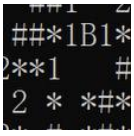
• 设置炸弹　定义 Bomb 类。对于机器人这只固定的炸弹，单次只能释放一个炸弹。对于玩家，建立炸弹链表。在释放炸弹后，该位置图标变成“o”，3 秒过后，炸弹爆炸，显示光束 1 秒，此时判断在炸弹与光束的位置上是否有 A、B、R，对应判断生死。爆炸之后，调用 afterboom 函数，判断被炸开的软墙之后是否有道具。初始炸弹威力是 1 格，之后捡道具依次增加。

• 设置道具　在 MAPTool[i][j]中随机设置道具隐藏在软墙下，1 是增加速度，即减少 Diff 的时间 加快读取频率。2 是增大爆炸威力，初始是 1 格，每捡一次道具增加 1，

• 设置得分显示 在地图的下方显示玩家的实时得分，以及当前的速度，炸点威力，剩余炸弹数目。

```
Player1 score : 0
        Bomb level: 1
        Bomb number:3
        Speed:      1
Player2 score : 0
        Bomb level:1
        Bomb number:3
        Speed:      1
```

释放炸弹



炸开之后

姓名：赵欣玥　　学号：211240065

• **处理定时事件**　主题框架采用 while(1) ONE_SECOND 循环计数 其中关于速度方面采用 clock 函数计时，更加准确。

遇到的困难：

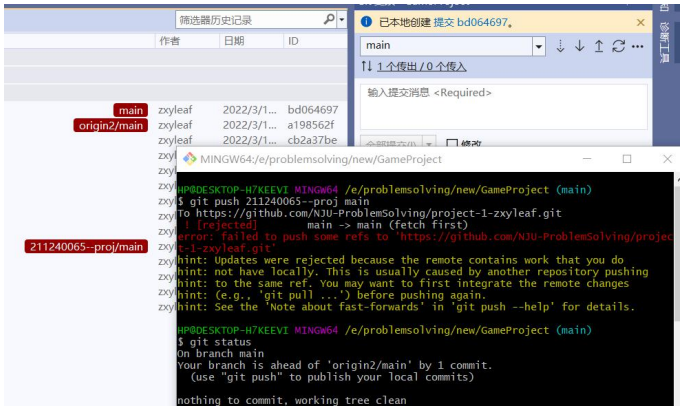
• **屏幕闪烁较快**　在对于玩家机器人每次移动、释放炸弹、分数改变时，都采用 system(“cls”)函数，若连续按键，屏幕闪烁现象较明显，降低游戏体验。

• **机器人不够智能**　由于只能想到随机数方法，导致机器人行动迟缓且有些智障。

• **使用远程仓库提交**　在一开始跟着廖雪峰老师的文字教程学习时，使用 Git Bush 建立远程仓库时，建立错了文件夹，导致在 vs 上会出现无法提交，只有本地的 commit 记录，无法 push。在第一次成功连接 github classroom 并提交上去之后，发现提交的文件夹含过多不必要的文件，便随手在远程删除了这些文档，之后没有注意两个远程仓库的分支出现了冲突，必须-f 强制 push，还采用 git log 查看修改的编号，并回到之前的修改。

心得体会：

在第一次使用类与对象的知识完成有一定代码量的项目，还有许多不足。没有用.h 将每一个类分开，只放在同一个.cpp 文件下，降低了代码的可读性。但初始完地图后，第一次人物可以移动，可以释放炸弹，机器人不会把自己炸死等等，都让我十分开心。尤其是在左支右绌的时候，又学会了更多 debug 的调试方法。当我看第一次写完炸弹类，发现我有着无敌破解版泡泡堂-AB 不会被炸死哈哈哈，R 被炸死又会突然自动复活等 bug。一开始在思考加速道具应该怎样实现时，先采用了最笨拙的按一次按键可以走 2 格的简单方法，但发现这样会有诸多问题，并且实现的非常不精准，于是在和同学交流后，学会了使用 clock()函数，精确记录行走事件，加快行走速度。第一阶段的项目，是我第一次写类与对象的代码，可能封装性方面做得不够完善，比如一些 public 和 private 写得很冗杂，可以在以后中多加学习。当我把完成的代码发给我的一位完全不懂编程和计算机的内测用户时，他给我提了很多建议，比如增加交互，游戏初始界面，游戏规则介绍界面等等，也让我在继承原版泡泡堂游戏时，有了自己的一些创新点。另外，屡次无法提交的 git 也让我对 git status、git diff、git log、git reset、git config --global --unset http.proxy 有了更深刻的理解，并且在报错中不断成长。



图为 失败经历分隔两地的分支 github 的 reject

致谢：

最后的最后，再次感谢朱宇博学长，从 2 月 22 号听我念叨不会写计时到 3 月提交发生错我。 “你的好像 还可以”虽然凸显了他的无奈，但我会在他的鞭策下继续努力学会写 C++的类与对象，在不完美中不断接近完美。