§.综合题 II - 消除类游戏伪图形工具函数集的实现及应用

【注意:】

1、 除明确要求外，已学过的知识中，不允许使用 goto、不允许使用全局变量，不允许使用 C++的 string 变量，不允许使用 C++的 STL 容器等后续知识

2、 本作业仅要求 VS2022 编译通过即可（“0 errors, 0 warnings”）

3、 不允许使用 scanf/printf 进行输入/输出

综合题 2：消除类游戏伪图形工具函数集的实现及应用

【消除类游戏伪图形工具函数集的实现（cmd_graphics_middleware = cmd_gmw_tools):】

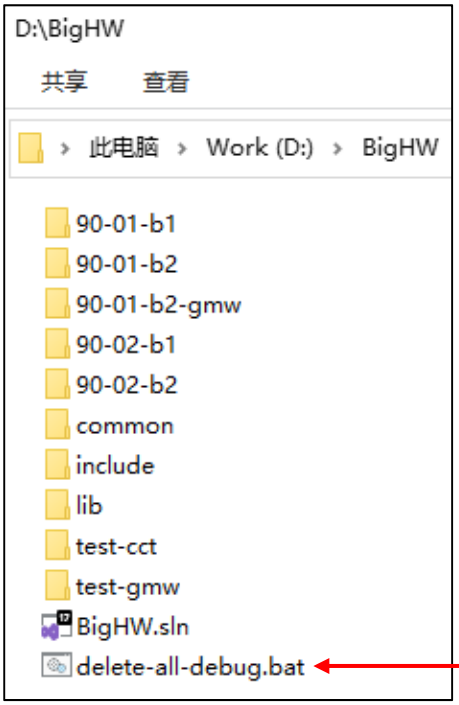
1、完成一套在 cmd 窗口下的消除类游戏伪图形工具函数工具集，包括：

- 若干设置函数
  - 设置整个窗口的字符、颜色
  - 设置游戏主框架区域的大小、颜色、线型、是否需要行号列标等
  - 设置附加行列
  - 设置一个色块的大小、线型、颜色、是否需要边框等
  - 设置画框架、色块、色块移动时的延时
  - 设置状态栏的显示与否、颜色等
- 画游戏主框架的函数
- 画一个色块的函数
- 移动一个色块的函数
- 状态栏信息显示函数
- 读键盘和鼠标的函数

2、附件给出的大作业模板 BigHW.rar 对应的解决方案说明

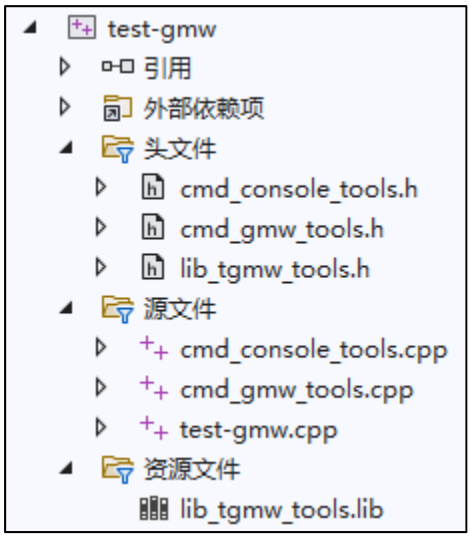
- 90-01-b1：汉诺塔对应的项目（无变化）
- 90-01-b2：合成十对应的项目（无变化）
- 90-01-b2-gmw：用本次的 gmw 工具集重做的合成十项目（新增，目前为空）
- 90-02-b1：数独对应的项目（无变化）
- 90-02-b2：用本次的 gmw 工具集新做的 2048/消灭星星/扫雷/其它消除类项目（新增，目前为空）
- common：公共函数集的源码（含义不变，有新增文件，本项目已设置为跳过，不需要编译，仅为方便查看）
- include：公共函数集对应的头文件（含义不变，有新增文件，本项目已设置为跳过，不需要编译，仅为方便查看）
- test-cct：cct 工具集的测试用例（无变化）
- test-gmw：本次需要实现的 gmw 工具集的测试用例（新增，已有内容）

3、附件给出的大作业模板 BigHW.rar 对应的文件目录及替换说明



注：附送 delete-all-debug.bat 大礼包
干什么用的自己揣摩，记得之前做好备份

- 共 12 个目录，但 BigHW.sln 打开后应有 9 个项目（注意缺省为 x64，要改!!!）
- lib 目录在解决方案中无对应的项目，其中放了 lib_tgmw_tools.lib 文件，是老师实现的工具集对应的静态库文件（因为既不需要编译，也不需要查看，所以 sln 中无项目对应）
- common：新增 cmd_gmw_tools.cpp 文件
- include：新增 cmd_gmw_tools.h 和 lib_tgmw_tools.h 文件
- test-gmw：源文件只有 test-gmw.cpp 一个



注意：

1、打开的项目中包含的文件如左图所示，想想这些文件时如何包含进来的（虽然目前的 sln 中已给出，但仍然需要掌握方法）

2、再次强调：公共函数的源文件及头文件只允许存在于（common/include）中，不允许在各项目的目录下再次存在，否则得分为 0!!!

- 本次给出的模板和第一次作业已有的 BigHW 的合并

  方法 1：在已完成作业的 BigHW 基础上加入本模板中的新增项
  - 先把对应目录(lib/test-gmw/90-01-b2-gmw)整体复制过来，再在 sln 中添加现有项目的方法（解决方案“BigHW”上右键菜单-“添加”-“已有项目…”）来添加这些已有项目
  - 将 common/include 中的 cmd_gmw_tools.cpp/cmd_gmw_tools.h/lib_tgmw_tools.h 这三个新增文件复制到对应目录中（必做），再在 common/include 的项目中对应添加这三个文件(非必做项,因为这两个项目是不编译的,放进去只是为了查看方便而已)

  方法 2：以本模板为准，将已完成作业的对应内容复制过来（包括 sln 和目录两方面）
  - 先把对应目录中的文件复制过来，再在 sln 的各项目添加，使能编译通过
  - 需要变化的有 90-01-b1/90-01-b2/90-02-b1/common/include

  注意：无论哪种方法，必须保证所有已完成项目可再次编译成功

4、完成要求

- 保证 test-gmw 下的测试用例通过（即 ENABLE_LIB_TGMW_TOOLS 为 0/1 时的表现一致）
  - 作业完成过程中可根据自己的需要随意修改测试内容
  - 要求 cmd_gmw_tools.cpp 实现后，本文件的原始测试用例的表现与 test-tgmw-demo.exe 的表现一致
  - 附件给出了 Colorlinez 和 MagicBall 两个小游戏，是因为测试用例中有这两个游戏需要用到的一些显示/移动效果（这两个游戏本身不需要实现）
- 将原 90-01-b2（合成十）用本套工具集重写（仅完整版即可）
  - 放入新项目 90-01-b2-gmw 中，不要改动生成的 exe 文件名（即生成为 90-01-b2-gmw.exe）
  - 要求伪图形界面为不带分隔线的版本
- 完成一个新的消除类游戏
  - 放入新项目 90-02-b2 中，不要改动生成的 exe 文件名（即生成为 90-02-b2.exe）
  - 可以在附件给出的 2048、消灭星星、扫雷中任选一个完成（仅完整版即可）
  - 也可以实现 Color linez/MagicBall 的伪图形界面版
  - 也可以自选一个消除类游戏（在 90-02-b2 目录中附加一个简单的文档说明即可）
  - 实现细节可以与 demo 不同，允许简化，主要关注工具函数的使用

5、关于静态链接库的简单说明

- *.lib 是由源代码编译而成的静态链接库文件，用户可以调用其中的函数，但是看不到函数的具体实现过程，适用于需要给他人提供工具函数的使用而不提供源代码的应用场景
- 附件 BigHW.rar 模板中给出的 lib_tgmw_tools.lib 包含了所有 tgmw_* 函数的实现，对应头文件是 lib_tgmw_tools.h（所有 gmw_*和 tgmw_*的形参表的个数和类型完全一致，因此可简单的通过宏替换互换）
- 为了使 lib_tgmw_tools.lib 中的 tgmw_* 函数运行正确，要保证与 gmw_* 函数两者共用的 CONSOLE_GRAPHICS_INFO 等四个结构体大小必须一致，因此在 main 函数开始添加了对四个结构体大小的验证，要求如果在这四个结构体中添加内容，只能使用最后预留的 64 字节空间，即必须保证添加的结构体成员占用 char pad[64];的空间，添加后整个整个结构体的大小保持不变
- 想看 lib_tgmw_tools.lib 中提供的 tgmw_*函数的实现效果，修改 ENABLE_LIB_TGMW_TOOLS 宏定义即可
  - 不要 gmw_*和 tgmw_*交叉使用，否则不保证正确性；每个测试函数中，不要在 tgmw_*之前调用 gmw_*，否则不保证正确性（即不要将测试用例中的 gmw_*单独改为 tgmw_*，而要通过宏定义 ENABLE_LIB_TGMW_TOOLS 整体切换）
  - 在宏定义 ENABLE_LIB_TGMW_TOOLS 为 1 时，可以通过修改 gmw_*的实参值，观察 tgmw_*函数的运行结果和你的期望是否相同（这是 test-tgmw-demo.exe 做不到的，exe 只能按固定值演示，无法修改）
  - 也可以将每个测试函数复制一次，一个显示 gmw_*的效果，一个显示 tgmw_*的效果，对比查看

【实验报告：】

1、本次作业暂时不需要提交单独的实验报告

2、后续会有综合性实验报告的要求，重点在公共函数的提炼

【作业要求：】

1、 仅需要在VS2022下编译通过即可，要做到“0 errors, 0 warnings”

2、 **4月3日前（周日，2.5周）**网上提交本次作业

3、 每题所占平时成绩的具体分值见网页

【控制台属性设置：】

统一要求为旧版并取消“快速编辑模式”和“插入模式”，否则鼠标读取会有问题