

一. 编译环境及编译选项

1. 编译平台：采用**VSCode**。
2. C语言环境：采用**mingw**，版本为 version **12.1.0** (MinGW-W64 x86_64-msvcrt-posix-seh, built by Brecht Sanders)
3. 编译指令: `gcc chess.c -o chess -lSDL2 -lSDL2main -lSDL2_image -lws2_32 -lSDL2_mixer`
4. 使用的第三方库：**SDL2**图形库，需要配置SDL2，SDL2_image，SDL2_mixer这些环境;引用了ws2_32.dll，用于支持网络编程。

二. 游戏方式

1. 窗口大小：**850 * 850** (像素)。
2. 开始游戏：**先**打开server.exe,**再**打开client1.exe和client2.exe,两个客户端的打开顺序**可以交换**。
3. 退出游戏：**点击任意一客户端界面右上角的红叉**即可退出，另一客户端也会随着一方的退出自动退出。
4. 游戏规则：
 1. 无论是代表黑方的客户端先进入还是代表红方的客户端先进入游戏，游戏开始后**都是红方先手**。
 2. 游戏中的合法操作：
 1. **按照原生象棋的规则在一回合内对己方棋子进行一次合法移动。**
注：如果游戏中双方的将出现对脸情况，一方可以**直接操控老将跨越整个棋盘**吃掉对方老将。
 2. **关于护摩之杖：**
 1. 在游戏中，玩家在对局过程中可以调用护摩之杖高贵的**爆伤**之力，先点击游戏中的护摩之杖图标，再点击想要升级的己方棋子，便可以在本回合内对己方棋子进行一次**升级**。升级后棋子的**外观和行为规则**也会有所改变。**变身后棋子增加的能力如下：**
 1. 马：不再蹩脚。
 2. 象/相：不再蹩脚。
 3. 车：与国际象棋中的象一般可以走斜线（也可以走直线）。
 4. 兵/卒：可以一下行走两格。
 5. 炮：可以隔着两个棋子进行攻击（升级后就不能隔着一个棋子攻击了）。
 6. 士/仕：无法加强。
 7. 将/帅：无法加强。
 2. **需要注意的是**，如果你不想对某一棋子进行加强了，只需要以**升级操作的手法对己方目标棋子进行操作**，便可以对其进行**降级**。同时，**一回合内**或对棋子进行升降级，或

对棋子进行合法移动。由于没点暴击率，只要对升级状态的棋子进行了一次合法移动，它便会自动降级恢复原状。

3. **关于悔棋**：游戏中支持悔棋操作，且双方都可以**无限次**悔棋。如果在走完一步后意识到自己走错了，便可以点击游戏中**箭头**状的图标发送悔棋**申请**，等待对方的回应。收到请求的一方既可以**给个机会**，当然也可以**残忍拒绝**。
4. **关于选中提示**：游戏过程中若**选中了**棋子或护摩之杖，其周围都会出现蓝色的选中框。
5. **关于再来一局**：如果双方都有再来一局的意愿，便可以在对局结束后点击**再来一局**。此时**上一局的败者将获得先手**。
6. **关于背景音乐**：将猫尾酒馆的钢琴纯音bgm循环播放，营造游玩“对弈游戏”时的美妙氛围（只在client1.c中加载了音乐，避免双声道）。

三. 些许思路

1. 采用**帧同步**的方式进行网络编程，客户端负责采集每一次鼠标点击事件的坐标并通过服务器转发给对手，服务端则负责处理信息。**一旦前后两次的点击事件能构成一次合法行动**，双方客户端就改变棋盘状态并交换回合。
2. 通过二维数组维护棋盘状态，并将每次的点击坐标转化为数组对应的下标。

四. 一些问题及感想

1. 图片有点糊~ 😊
2. 如果一个客户端在下棋的时候在棋盘上疯狂乱点，传一堆无用的数据，会导致另一个的客户端反应十分缓慢。（每一次send receive judge貌似都需要一定的时间，目前还没有找到解决方案）（如果对面半天没反应试着在对面戳戳棋子，可能会有反应）。
3. 代码中还有很多可以优化的地方，继续加油罢！ 😊