

## 25 秋季 C 语言程序设计课程中期作业

### 人人对战版五子棋程序设计

提交时间：2025 年 12 月 19 日 23:59 前

#### 重要提示

- 本次作业为 C 语言课程中期作业，占课程总评成绩的 10%
- 请务必**独立完成**，严禁直接抄袭他人代码，一经发现按零分处理
- 本次作业采用人工评阅方式，需提交完整项目源代码和主程序可执行文件
- 提交截止时间后原则上不再接受补交，请合理安排时间
- 程序必须能够正常编译运行，无法运行的程序将严重影响评分

# 目录

<b>1</b>	<b>项目简介</b>	<b>3</b>
1.1	基本介绍	3
1.2	项目目标	3
1.3	最终目标	3
<b>2</b>	<b>功能要求</b>	<b>4</b>
<b>3</b>	<b>五子棋规则详解</b>	<b>5</b>
3.1	基本规则	5
3.2	棋型说明	5
3.3	禁手详解	6
3.3.1	禁手的类型	6
3.4	胜负判定	7
3.4.1	胜局条件	7
3.4.2	和局条件	7
<b>4</b>	<b>提交要求</b>	<b>8</b>
4.1	基本要求	8
4.2	代码规范要求（非强制，仅作为高分建议）	8
4.3	提交方式	9
<b>5</b>	<b>评分标准</b>	<b>10</b>
5.1	基本功能 (70 分)	10
5.2	代码质量 (30 分)	10
<b>6</b>	<b>常见问题解答</b>	<b>11</b>
6.1	技术问题	11
6.1.1	Q1: 如何清空屏幕以实现棋盘刷新?	11
6.1.2	Q2: 如何实现棋盘的美观显示?	11
6.1.3	Q3: 禁手判断太复杂, 有什么简化方法吗?	11
6.1.4	Q4: 对复杂禁手规则还是不太理解, 怎么办?	11
6.2	提交相关问题	11
6.2.1	Q5: 可以使用多个.c 文件吗?	11
6.2.2	Q6: 压缩包里需要包含.o 文件吗?	11
6.2.3	Q7: 我的程序在我的电脑上能运行, 但不确定在其他电脑上是否可以?	11

## 1 项目简介

本次中期作业要求实现一个人人对战版五子棋程序。五子棋，又称连珠、五目、五目碰、五格等，是起源于中国古代的传统黑白棋种之一。现代五子棋称为连珠，是一种两人对弈的纯策略型棋类游戏。

### 1.1 基本介绍

五子棋使用围棋棋盘和棋子，两人对局，各执一色，轮流下一子。棋子下在棋盘的交叉点上，棋子下定后不再移动。率先在棋盘上形成横向、竖向或斜向连续五个同色棋子的一方获胜。

棋盘正中一点为“天元”。棋盘两端的横线称端线。棋盘左右最外边的两条纵线称边线。从两条端线和两条边线向正中发展而纵横交叉在第四条线形成的四个点称为“星”。

以持黑方为准，棋盘上的纵轴线从左到右用英文字母 A~O 标记。横行线从近到远用阿拉伯数字 1~15 标记。纵横轴上的纵横线交叉点分别用纵横线标记的名称合写成。如“天元”为 H8，四个“星”分别为 D4、D12、L12、L4 等。

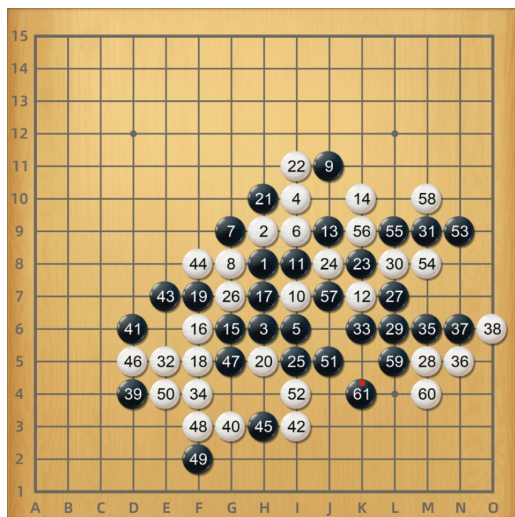


图 1: 五子棋棋盘示意图

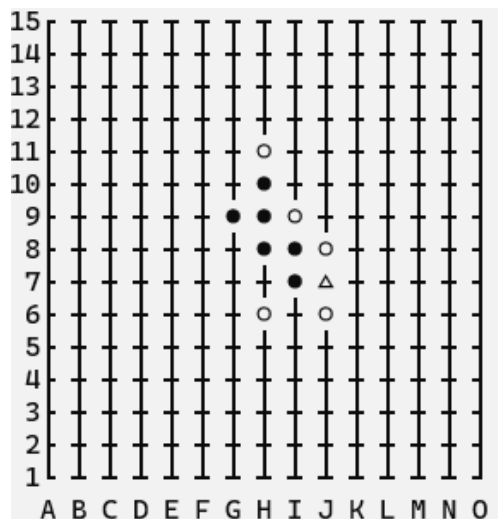


图 2: 控制台棋盘实现示例

### 1.2 项目目标

通过本次作业，要求学生：

- 掌握 C 语言基本语法，能够运用结构化程序设计方法解决实际问题
- 理解并实现棋盘的数据结构表示
- 实现基本的输入输出界面（命令行界面）
- 掌握游戏逻辑的实现，包括落子判断、禁手判断、胜负判定等
- 理解并实现五子棋的特殊规则（禁手规则）
- 培养调试程序和解决问题的能力

### 1.3 最终目标

本项目将会要求大家在期末阶段完成最终的“人机对战”版本五子棋设计并进行对战 PK。完成中期作业的设计后，可尽快开始最终版本的实现。

## 2 功能要求

程序必须实现以下基本功能：

### 1. 使用 C 语言编写

- 必须使用**标准 C 语言**（C99 及以上标准）编写
- **严禁使用 C++**或其他语言

### 2. 命令行界面

- 程序必须在**命令行终端**中运行，按图2所示的方式显示棋盘，不要改动行列编号顺序
- **不允许使用图形界面**（如 GTK、Qt 等）
- 使用标准输入输出（`stdin/stdout`）进行交互
- 界面应清晰美观，易于理解（期末评分的重要依据之一）

### 3. 标准键盘输入

- 只能使用**键盘**进行输入
- **不使用鼠标**进行交互
- 输入方式采用“字母 + 数字”式的坐标输入（如：H7）

### 4. 人人对战模式

- 实现双人对战功能，黑白双方轮流落子
- 明确显示当前轮到哪一方落子
- 提供清晰的落子提示（如输入引导文字）

### 5. 禁手规则实现

- 实现**黑棋禁手**规则（详见第3节）
- 白棋无禁手限制
- 程序应能自动判断禁手并在黑棋尝试下禁手点时给出明确提示，对黑棋判负

### 6. 最后一步特殊显示

- 最新落下的棋子应有**特殊标记**，可以用不同的符号、颜色来区分，如实心三角形/空心三角形
- 便于玩家识别最近的操作

### 7. 输赢判断

- 能够正确判断五连（横、竖、斜向）
- 判断黑棋禁手导致的输局
- 判断棋盘下满的和局
- 胜负确定后显示明确的结果信息，而不是直接退出程序

注：最终版本中，实现机器落子时需要明确输出机器思考所用时长（最多 15s），具体要求将在期末通知。

## 3 五子棋规则详解

### 3.1 基本规则

1. 对局双方各执一色，黑先、白后，从天元（棋盘中心）开始相互顺序落子
2. 白棋第一手应在天元为界自己一侧布子，之后双方可任意行子
3. 最先在棋盘横向、竖向、斜向形成连续的相同色五个棋子的一方为胜
4. 黑棋禁手判负、白棋无禁手
  - 黑棋禁手包括：“三三禁手”、“四四禁手”、“长连禁手”
  - 黑方只能通过“四三”取胜
  - 黑方出现禁手，无论是自愿或被迫走出，判白方胜
  - 若白方在黑方出现禁手后，未立即指出又落下一白子，则黑方禁手不再成立
5. 如分不出胜负（棋盘下满），则定为平局
6. 五连与禁手同时形成，先五为胜（五连优先于禁手）

### 3.2 棋型说明

理解以下棋型定义对于实现禁手规则至关重要：

#### 1. 长连

- 定义：在一条直线或斜线上，连续下成六个或以上的同色棋子
- 说明：黑棋形成长连即为禁手，白棋长连视为胜利

#### 2. 活三

- 定义：若对方不进行必要防守，本方下一手可取得活四的基本子力形式（包括连活三和跳活三）
  - 连活三：在一条横线、竖线或斜线上紧紧相连的同色三子，两端均有空交叉点，且至少有一端有两个或以上的空交叉点（对于黑方则还需左右相邻的两个交叉点在此三形成的同时至少有一个不为禁点，且从两端向外数至少在各两个交叉点上均无黑子）所构成的基本子力形式
  - 跳活三：中间间隔一子的活三，且两端均有空交叉点（对于黑方则还需在此三形成的同时中间的空交叉点不为禁点，且从两端向外数至少在各两个交叉点上均无黑子）所构成的基本子力形式
- 特别说明：1、3、5 式的中间空出两个子的三不叫跳活三，而只能称为跳三。跳三不认为属于活三的范围，所以不纳入禁手判断考虑的情况
- 示例：
  - 连活三：\_\_\_●●●\_\_\_（两端均为空，至少一端有两个空位）
  - 跳活三：\_\_●\_●●\_\_（中间间隔一子）

#### 3. 活四

- 定义：在一条直线或斜线上连续相邻的四枚同色棋子且有两个空白交叉点可以成五的棋型（对于黑方则还需左右相邻的两个交叉点在此四形成的同时至少有一个不为禁点，且从两端向外数至少在各两个交叉点上均无黑子）
- 示例：\_\_\_●●●●\_\_\_（两端均有两个空位）

#### 4. 冲四

- 定义：指除“活四”外的，再下一着棋（下子的位置不能是黑棋的禁点）便可形成五连，并且存在五连的可能性的局面
- 示例：○●●●●\_\_ 或 \_\_●●●●○

#### 5. 五连

- 定义：五个同色棋子在一条线上连成一线
- 这是获胜的基本条件

### 3.3 禁手详解

黑棋先行具有较大优势，因此对黑棋施加了禁手限制。白棋无任何禁手限制。

#### 3.3.1 禁手的类型

##### 1. 三三禁手

- 定义：由于黑方走一着在无子交叉点上同时形成两个或两个以上黑方“活三”的局面
- 注意：此子必须是形成的至少两个活三的共同构成子
- 如图4所示，A 点黑棋一子落下同时形成了两个活三，注意三个黑 1 组成的三，在黑棋 A 点落子后由眠三（假活三）成为活三，这是由于 A 点落下黑子后，B 点被解禁了。但 A 点落子不是这两个活三的共同构成子，所以 A 点不是三三禁手。请同学们务必仔细揣摩。

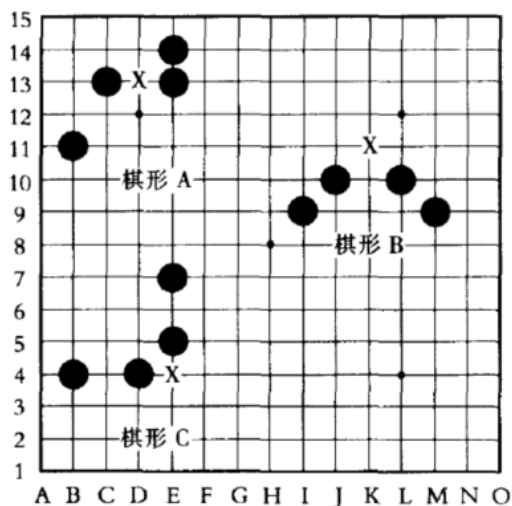


图 3: 三三禁手示例 (X 点为禁手点)

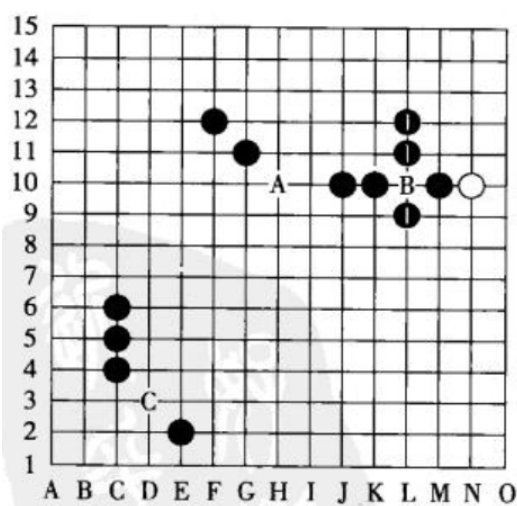


图 4: 三三禁手反例 (A、C 点不为禁手点)

##### 2. 四四禁手

- 定义：当黑棋走一步棋，同时形成两个或两个以上的“四”（包括活四和冲四）且没有形成“五连”时，该落子点为禁手点
- 注意：只要是“四”的棋型都算，不区分活四还是冲四
- 如果同时形成了五连，则五连优先，不算禁手

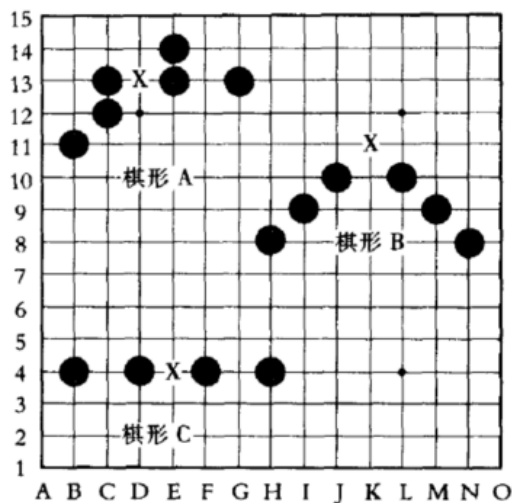


图 5: 四四禁手示例 (X 点为禁手点)

### 3. 长连禁手

- 定义：当黑棋走一步棋，形成六个或六个以上连续同色棋子，该落子点为禁手点
- 注意：只有黑棋受此限制，白棋长连视为胜利

## 3.4 胜负判定

### 3.4.1 胜局条件

#### 1. 五连胜利

- 最先在棋盘上形成五连的一方获胜
- 白棋长连（六个或以上连续棋子）视同五连，白方胜

#### 2. 黑棋禁手判负

- 黑方走出禁手点，白方胜
- 但如果黑方同时形成五连与禁手，禁手失效，黑方胜

#### 3. 对方认输或超过规定时限

- 一方主动认输，对方获胜
- 一方该回合落子思考时长超过规定时限（针对机器思考而言，时限为 15s），对方获胜

### 3.4.2 和局条件

#### 1. 棋盘下满

- 全盘均下满，已无空白交叉点
- 双方均未形成五连，判为和局

#### 2. 双方同意和棋

- 对局双方同一回合均放弃行棋权
- 对局双方一致同意和棋

## 4 提交要求

### 4.1 基本要求

本次作业需要提交一个以学号命名的 **zip 压缩包**，包含以下内容：

#### 1. 源代码文件（必需）

- 所有 `.c` 和 `.h` 源文件（可以建立子文件夹，但结构和命名应该清晰规范）
- 代码必须有**良好的注释**（期末评分的重要依据之一）
- 代码风格应**规范统一**（期末评分的重要依据之一）
- 如果项目包含多个文件，请在 `README` 中说明文件组织结构

#### 2. 可执行文件（必需）

- 编译好的可执行程序（Windows 下为 `.exe` 文件，Linux 下为可执行文件）
- **必须保证可执行文件能够直接运行**
- 如果需要特殊的运行环境，请在 `README` 中说明

#### 3. `README` 文档（必需，期末评分的重要依据之一）

- 可以是 `README.txt` 或 `README.md`，也可以是 `tex` 文件或 `pdf` 格式
- 内容包括：
  - 项目简介（可以简单介绍自己的设计思路和方法，项目的亮点与优势）
  - 编译与运行方法（具体的编译命令等等）
  - 操作说明（如何输入坐标等）
  - 实现的功能列表
  - 已知问题（如果有）
  - 其他说明

#### 4. `Makefile`（可选，但强烈建议）

- 提供编译脚本，方便评阅
- 应包含 `make`、`make clean` 等目标

#### 5. 测试用例（可选）

- 如果编写了测试代码，可以一并提交
- 有助于展示程序的正确性（如复杂嵌套禁手判断的逻辑是否正确）

### 4.2 代码规范要求（非强制，仅作为高分建议）

#### 1. 命名规范

- 变量名、函数名使用有意义的英文单词或拼音
- 建议使用下划线命名法：`check_win`、`game_state`
- 常量使用大写：`BOARD_SIZE`、`BLACK`

#### 2. 注释规范



- 每个函数前应有函数说明注释
- 复杂逻辑处应有行内注释
- 注释可以使用中文或英文
- 注释应清晰说明代码意图

示例：

```
1  /**
2   * 检查指定位置是否为禁手点
3   * @param game 游戏状态指针
4   * @param row 行坐标
5   * @param col 列坐标
6   * @return 1表示禁手，0表示非禁手
7   */
8  int is_forbidden(GameState *game, int row, int col) {
9      // 只有黑棋有禁手限制
10     if (game->current_player != BLACK) {
11         return 0;
12     }
13     // ...
14 }
```

### 3. 代码格式

- 使用一致的缩进（建议 4 个空格或 1 个 Tab）
- 大括号风格保持一致
- 运算符两边留空格
- 适当使用空行分隔逻辑块

### 4. 函数设计

- 每个函数应功能单一，职责明确
- 避免函数过长（建议不超过 50 行）
- 合理使用函数参数和返回值

## 4.3 提交方式

- 提交平台：国科大在线
- 提交方式：直接以附件形式提交即可
- 如有特殊情况，请提前与助教沟通

## 5 评分标准

本次作业总分为 100 分，占本课程总评成绩的 10%。参考评分标准如下：

### 5.1 基本功能 (70 分)

评分项	评分要点	分值
程序可运行性	<ul style="list-style-type: none"><li>程序能够正常编译（5 分）</li><li>程序能够正常运行不崩溃（10 分）</li></ul>	15 分
界面显示	<ul style="list-style-type: none"><li>棋盘显示清晰美观（5 分）</li><li>坐标显示正确（3 分）</li><li>最后一步有特殊标记（2 分）</li></ul>	10 分
基本对局功能	<ul style="list-style-type: none"><li>能够正常落子（5 分）</li><li>黑白轮流对战（5 分）</li><li>落子合法性检查（5 分）</li></ul>	15 分
胜负判定	<ul style="list-style-type: none"><li>正确判断横向五连（2 分）</li><li>正确判断纵向五连（2 分）</li><li>正确判断两条斜向五连（4 分）</li><li>判断和局（2 分）</li></ul>	10 分
禁手规则	<ul style="list-style-type: none"><li>实现三三禁手判断（5 分）</li><li>实现四四禁手判断（5 分）</li><li>实现长连禁手判断（3 分）</li><li>实现复杂嵌套禁手判断（5 分）</li><li>禁手提示信息（2 分）</li></ul>	20 分

### 5.2 代码质量 (30 分)

评分项	评分要点	分值
代码规范	<ul style="list-style-type: none"><li>命名规范（4 分）</li><li>代码格式规范（3 分）</li><li>注释充分（3 分）</li></ul>	10 分
程序结构	<ul style="list-style-type: none"><li>函数划分合理（3 分）</li><li>逻辑清晰（4 分）</li><li>代码简洁高效（3 分）</li></ul>	10 分
文档完整性	<ul style="list-style-type: none"><li>提供完整的 README（5 分）</li><li>说明清晰（5 分）</li></ul>	10 分

注：学术诚信是每个人都应具有的良好品质。我们鼓励同学们学习借鉴往届大神或者身边同学的优秀思路，但这并不意味着抄袭。借鉴思路时，请记得在注释中说明参考来源并致谢。不加以注释地原样照抄他人代码，一经发现将按**零分处理**。

## 6 常见问题解答

### 6.1 技术问题

#### 6.1.1 Q1: 如何清空屏幕以实现棋盘刷新?

A: 可以使用以下方法:

- Windows 系统: `system("cls");`
- Linux/Mac 系统: `system("clear");`

#### 6.1.2 Q2: 如何实现棋盘的美观显示?

A: 建议使用 Unicode 字符: ● (黑子)、○ (白子) 等等

#### 6.1.3 Q3: 禁手判断太复杂, 有什么简化方法吗?

A: 可以先实现基础的连子统计函数, 再分别实现各类棋型 (活三、活四、冲四等) 的识别, 在此基础上组合实现禁手判断。对于复杂嵌套禁手的判断, 可能需要使用递归方法。在编写禁手判断时, 也别忘了充分利用测试用例验证正确性 (建议先完成基本功能, 再逐步完善复杂的禁手判断)。

#### 6.1.4 Q4: 对复杂禁手规则还是不太理解, 怎么办?

A: 可以参考作业附件: 五子棋禁手规则详解.pdf

### 6.2 提交相关问题

#### 6.2.1 Q5: 可以使用多个.c 文件吗?

A: 完全可以。我们强烈建议将代码分成多个文件, 例如:

- main.c: 主程序
- board.c/board.h: 棋盘相关函数
- rule.c/rule.h: 规则判断函数
- display.c/display.h: 显示相关函数

提交时记得包含所有源文件, 并在 README 中说明编译方法。

#### 6.2.2 Q6: 压缩包里需要包含.o 文件吗?

A: **不需要**。只需包含源代码文件 (.c 和 .h)、可执行文件、README 文档和 Makefile (可选)。不要包含中间编译产物 (.o 文件)。

#### 6.2.3 Q7: 我的程序在我的电脑上能运行, 但不确定在其他电脑上是否可以?

A: 建议使用标准 C 库函数, 避免使用编译器特定扩展, 并在 README 中说明编译环境和依赖。如果可能, 在其他电脑或虚拟机上进行检查和测试。

祝各位同学顺利完成作业!