

# 吉林大学 2021 年春季学期

## 数据结构课程设计 A2 题

### 不围棋锦标赛

不围棋（NOGO）由围棋衍生而来，棋盘比围棋小，且规则与围棋相反，是“中国大学生计算机博弈大赛暨中国计算机博弈锦标赛”的正式比赛项目。在本题中，你的任务是编写不围棋 AI 程序，即让程序自动下棋，并与其他同学对战博弈，进而决出冠军及名次。

#### 一、游戏规则与术语：

1. 棋盘为  $9 \times 9$  围棋棋盘。

2. 黑子先手，双方轮流落子，落子后棋子不可移动。

3. “连”的定义：“连”就是棋子和棋子接在一块，成为一个整体。横是连，竖是连，斜线不算连。例如图 1 中带方形的两颗棋子是相连的，带三角形的三颗棋子是相连的，右边的带圆形的三颗棋子不相连。

4. “气”的定义：“气”是围棋和不围棋里最核心的概念。一个棋子在棋盘上，与它直线紧邻的空点是这个棋子的“气”。棋子直线紧邻的点上，如果有同色棋子存在，则它们便相互连接成一个不可分割的整体，它们的气也应一并计算。棋子直线紧邻的点上，如果有异色棋子存在，这口气就不复存在。如所有的气均为对方所占据，便呈无气状态。例如图 2 中，角上一颗黑子有 2 口气，其右侧上边的一颗黑子有 3 口气，中间一颗黑子有 4 口气。右边两个白子相连有 6 口气，下边三个白子相连有 8 口气。

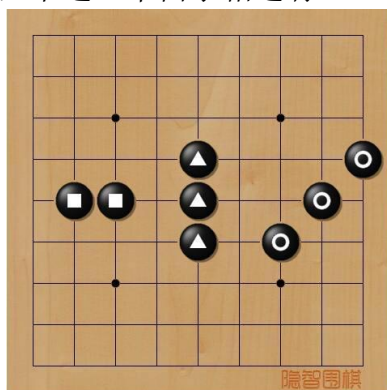


图 1 连

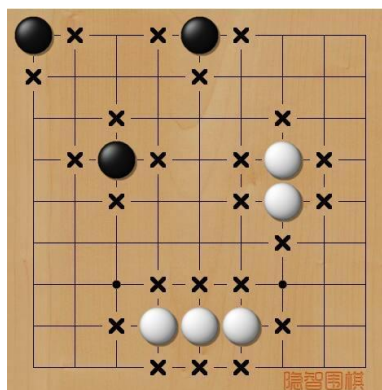


图 2 气

5. “吃子”的定义：落子后，使对方的棋子变成无气状态就是吃子。例如在图 3 中，如果白棋下在“×”标记点上，就会把黑棋变成无气，把黑棋吃掉。

6. 不围棋对弈的目标不是吃掉对方的棋子，恰恰相反，如果一方落子后吃掉了对方的棋子，则落子一方判负。

7. 如果一方在棋盘上某个点落子后，该子将呈现无气状态，那么相当于自杀，例如图 4 中，白棋若在“×”处落子即为自杀。不围棋对弈禁止自杀，落子自杀一方判负。

8. 对弈禁止空手（Pass），即轮到本方走棋时，若无棋可走，则本方判负。

9. 对弈结果只有胜负，没有和棋。

10. 每步走棋限时 1 秒，超时判负。内存限定 256MB。

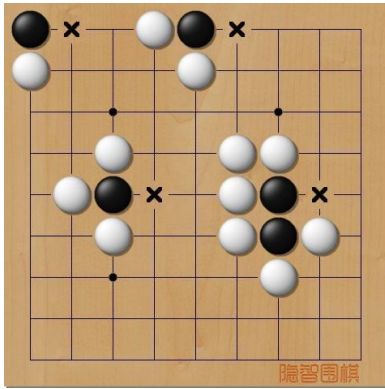


图 3 吃子

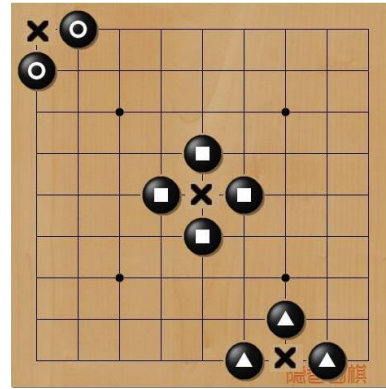


图 4 自杀

建议大家实际玩玩这款游戏，玩个十几局就掌握其规则及玩法了。PC 版已发给大家（若无法打开，可能需要安装 visual studio），手机版可在 app store 或安卓市场下载。大家也可以查阅相关网络资料，进一步了解不围棋的下棋技巧和经验。

## 二、代码实现：

大家无需掌握图形界面编程技术，只需将代码上传至北京大学 Botzone 对战平台（<https://www.botzone.org.cn/>），对战平台将自动进行图形化对战比赛，并给出排名。本题编程语言为 C/C++ 语言，允许使用 C++ STL。

**请大家注册 Botzone 账号，并加入“吉林大学2021年数据结构课程设计”小组。**

### （1）棋盘坐标设定

9×9 的二维棋盘分为横轴和纵轴两个维度，以左上角为坐标原点 (0, 0)。坐标系如图 5 所示。在程序中棋盘信息存储在数组 `int board[9][9]` 中，下标从 0 开始，元素 `board[x][y]` 有 0、1、-1 三种可能取值，分别表示棋盘 (x, y) 处为空白（没有任何棋子）、对方棋子、我方棋子。

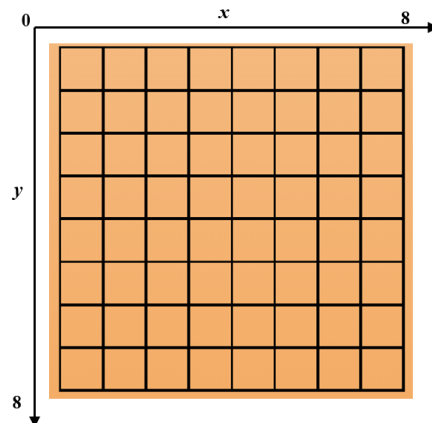


图 5 棋盘坐标系

### （2）你的程序与对战平台的交互方式

我们提供给大家 **样例程序**，样例程序已经包含了程序与对战平台的交互功能，你只需在样例代码基础上开发，基于当前棋局，输出本回合的决策结果。你的程序（在 botzone 网站里也称为 bot）的生命周期仅为一次决策：读入当前棋盘状态，输出本回合你的下棋（落子）位置。在整个对弈过程中，对战平台会多次调用你的程序（每下一步棋调用你一次，直至对弈结束）。

**了解本小节下面内容有助于理解程序的运行原理和流程。若不想或无法理解下面内容，可**

跳过而直接看（3），对完成本题没有影响。

对战平台每次调用你的程序时：首先将之前对方和本方落子的位置作为输入传送给你的程序，从而建立起棋盘信息board数组，然后你的程序根据棋盘信息进行决策，并向对战平台输出你的决策结果，即本方的落子位置。

你的bot与对战平台有两种交互方式：JSON交互和简单交互（你可任选其一）。

### ➤ JSON交互

你可以理解为程序通过一系列“特殊格式的字符串”与对战平台进行交互，其格式形如：

```
"requests":[{"x":-1,"y":-1},{"x":0,"y":8},{"x":6,"y":8},{"x":1,"y":4}],
"responses":[{"x":2,"y":3},{"x":7,"y":5},{"x":5,"y":8}]}
```

其中requests后面是对方以往落子的位置，如果是黑方的第一回合，则 request 为 {"x": -1, "y": -1}。responses后面是本方以往的落子位置。你的程序可以对上述字符串进行自动解析，解析出x坐标和y坐标（样例程序已经包含了解析代码，无需你来编程解析），这样你可以根据这些x和y坐标，建立当前棋盘状态信息（样例程序已经实现该功能，并将棋盘信息存入board数组，你只需根据board数组所表示的棋盘信息，进行决策即可）。详见示例程序。

例如本方是黑方，对于上面的信息，可知之前的棋应该是这么下的：我方第一回合收到(-1,-1)，然后我方下棋(2,3)，然后对方下棋(0,8)，然后我方下棋(7,5)...以此类推，形成图6的棋盘。

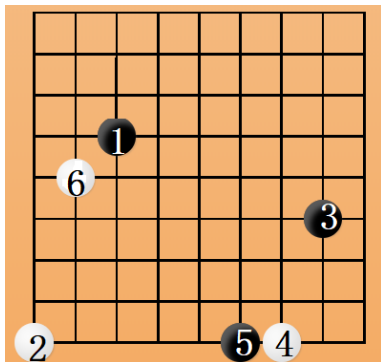


图6 对战平台与博弈程序通信示例

### ➤ 简单交互

使用这种交互方式，你的程序通过一系列数字与对战平台进行交互，格式形如下表左侧所示。而你的程序就是要读取这些数字，建立棋盘信息，并基于棋盘信息做决策，最后输出第k回合本方落子位置。详见示例程序。例如图6的棋盘状态对应下表右侧的输入。

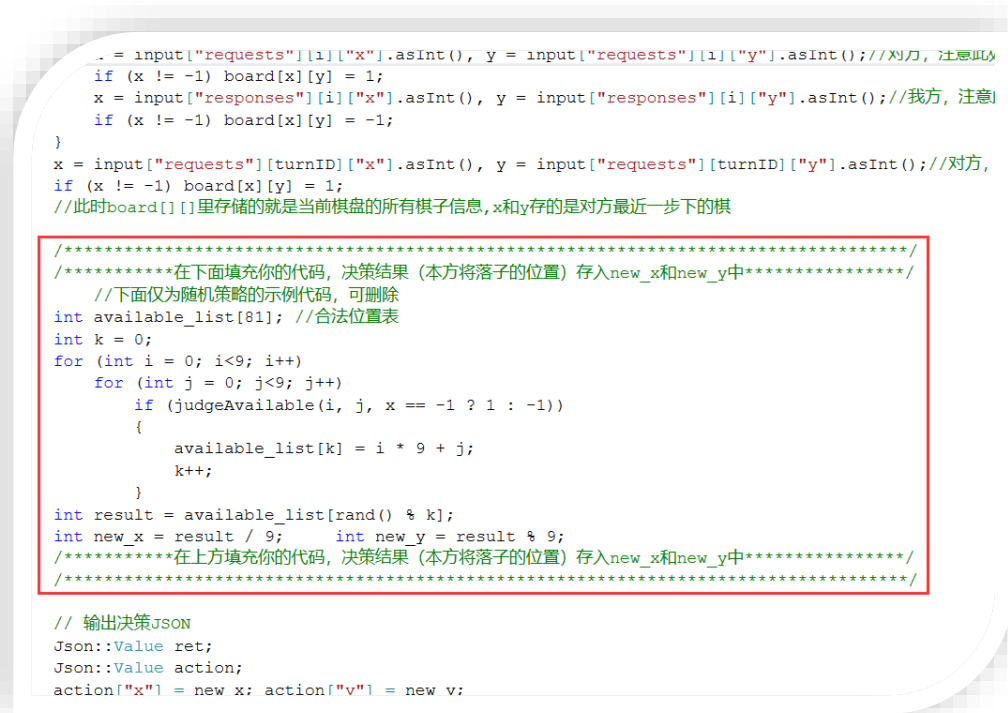
$k$	//双方已下棋的回合数	4
$x_1^{\text{对方}} y_1^{\text{对方}}$	//第1回合对方落子位置	-1 -1
$x_1^{\text{本方}} y_1^{\text{本方}}$	//第1回合本方落子位置	2 3
$x_2^{\text{对方}} y_2^{\text{对方}}$	//第2回合对方落子位置	0 8
$x_2^{\text{本方}} y_2^{\text{本方}}$	//第2回合本方落子位置	7 5
.....		6 8
$x_{k-1}^{\text{对方}} y_{k-1}^{\text{对方}}$	//第k-1回合对方落子位置	5 8
$x_{k-1}^{\text{本方}} y_{k-1}^{\text{本方}}$	//第k-1回合本方落子位置	1 4
$x_k^{\text{对方}} y_k^{\text{对方}}$	//第k回合对方落子位置	

关于bot与平台的交互方式的更详细描述可参考：

<https://wiki.botzone.org.cn/index.php?title=Bot#.E4.BA.A4.E4.BA.92>  
<https://wiki.botzone.org.cn/index.php?title=NoGo>

### (3) 博弈程序编写

提供给大家JSON交互和简单交互的两个样例程序，分别为nogoai.cpp和nogo\_simple.cpp，里面包含了详细注释，容易阅读。你需要实现cpp中下图红框部分。



样例程序在 visual studio 2017 下可直接运行，其中 nogo\_simple\_inter 项目为简单交互，nogo 项目为 JSON 交互，想选择哪种交互方式，右键将对应项目设为启动项目即可。若在其他 IDE 下开发，可将样例代码文件 nogoai.cpp 和 nogo\_simple.cpp 拿出来到你的 IDE 下进行开发即可，注意代码文件要与 json 文件夹和 jsoncpp.cpp 在同一文件夹下。详见：

<https://wiki.botzone.org.cn/index.php?title=%E8%B0%83%E8%AF%95#.E6.9C.AC.E5.9C.B0.E8.B0.83.E8.AF.95>

### (4) 博弈程序的调试

#### ➤ 本地调试

在本地编译运行程序，本地调试时可采用“简单交互”方式，并结合输入输出重定向，以数字形式作为输入，并观察程序输出结果。

#### ➤ 在Botzone网站上在线调试

Botzone 提供了比较完善的调试功能。点击网页右上角的“我的 Bot”，然后点击下方“创建新 Bot”，如图 7。然后填写 bot 信息，上传代码文件，如图 8，注意每步走棋限时 1 秒。请不要勾选“开源”，为了避免抄袭，将对开源者扣分。**Botzone 提供了较好的版本管理功能，方便管理程序的各个版本。**若想对已有的 bot 增加新的版本，可如图 9 所示点击加号。



图 7 创建新 Bot



图 8 配置 Bot 信息上传代码



图 9 为已有 bot 增加新版本



图 10 创建游戏桌

在线调试，可回到首页，手动创建游戏桌，如图 10 所示。然后指定玩家进行对局，如图 11 所示，对局双方可以是已有的 Bot，也可以是房主（人类），即你自己。



图 11 指定玩家进行对局调试

在对局页面（也可通过右上角菜单“我的对局”选择对局点击“回放”）在线调试，点击下方“切换到调试模式”，如图 12 所示。详细调试信息可以点击右下角“log 查看工具（调试



用) ” 查看每回合每方具体的输入和输出信息, 如图 13, 选择“查看完整 Log”, 也可以查看程序的运行时间。为防止程序超时, 可以尝试进行卡时, 比如在每次迭代/循环时, 通过 `clock()` 或者 `time()` 函数来判断程序运行了多久, 从而决定是否提前返回。

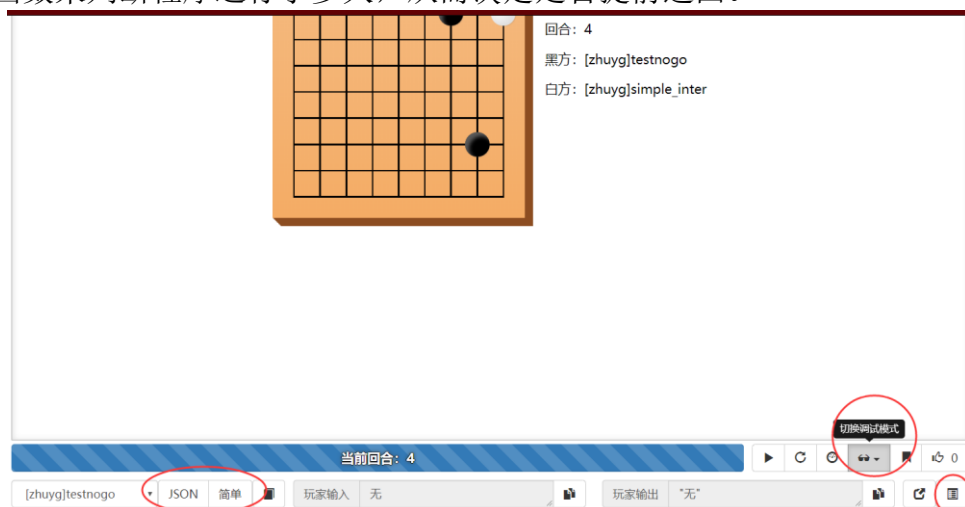


图 12

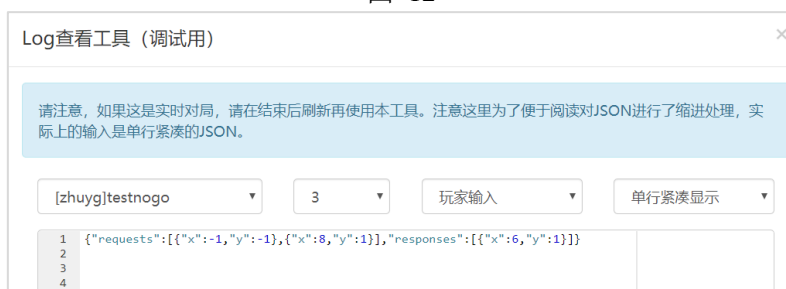


图 13

在 JSON 交互模式下, 可将程序中的调试信息写入代码中的 `ret["debug"]` 处, 在线调试的 log 查看工具中的“玩家输出”可显示出运行过程中的调试信息, 如图 14 所示。



图 14

**你的程序在本地的输出和平台的输出不一样怎么办?** (1) 你的程序可能存在漏洞, 在平台上可能会暴露出数组越界、地址越界、除零、未初始化等问题。建议仔细检查自己的代码。TScanCode 是一个很好的工具。(2) 如果采用了随机数, 可能你本地生成的随机数和平台生成的随机数不一致。

### 三、实验报告提交方式:

通过超星作业的方式提交实验报告 **PDF 版**, 报告模板另行发布。文件以“**班号\_题号\_姓名**”命名, 例如“21 班\_A2\_张无忌.pdf”, **若没按要求命名文件, 将被扣掉一定分数, 且系统将无法保证文件一定会被老师收到。**对于少数民族姓名较长的同学, 可简写姓名, 能辨别出是你即可。

## 四、程序提交方式：

注册 Botzone 账号并加入吉大课设小组，按规定时间参加对抗积分赛。**参加比赛的 Bot 名必须按“计/软\_班号\_姓名”命名**，如“计 25 班\_张无忌”、“软 6 班\_令狐冲”等，**注意这里说的是 Bot 名，而不是用户昵称，未按要求命名者成绩将会受到影响**。对于少数民族姓名较长的同学，由于系统限制了 bot 名长度，所以建议简写姓名，能辨别出是你即可。



图 15 加入小组

可以将用户昵称设置为个性化的队名，但不要通过图 16(a)设置“组中昵称”，那样会使最终的排行榜格式变乱。设置昵称可通过如图 16(b)(c)所示的页面右上方菜单“账户修改”处修改昵称。



图 16 修改昵称

## 五、评测方法：

以程序对战的方式进行评测，比赛分为两阶段。

- **第一阶段：班内赛。**每班内部采用循环积分赛，每场比赛胜者积 1 分，负者 0 分。大家在本课程的成绩将由本阶段比赛决定。
- **第二阶段：诸神之战。**各班冠军（计算机学院 15 人、软件学院 10 人）进入第二阶段。若各班竞争激烈比分接近，也可能适当放宽每班出线名额。第二阶段仍通过循环积分赛，决出两院总冠军。本阶段仅供娱乐，与成绩无关。
- **前情回顾：**在去年棋类博弈项目的两院诸神之战 PK 中，计算机学院包揽前 6 名。今年计算机学院能否延续辉煌？软件学院能否为学长报一箭之仇？我们拭目以待。

## 六、成绩评定方法：

本题满分为 100 分，程序质量（比赛结果）占 90 分，实验报告占 10 分。第一阶段比赛中，若本班冠军的分数为 k，则你的分数乘以 90/k，即将每位同学的分数映射到[0, 90]区间。实验报告的分数由各带班老师评定。

提交的文件没按照题目要求的格式命名的，视情节轻重进行一定扣分，尤其是 bot 名（非玩家昵称）没按规则命名导致无法辨别出学生班级姓名的，则比赛结果无效，得分全部扣减。

通过查重和现场答辩确定是否抄袭或雇佣他人完成。作弊者视情节轻重倒扣分，完全抄袭他人或找他人代做的，本课程记为不及格。查重时样例代码和注释不算在内。查重并不一定在题目结束后立即进行，可能在课程全部结束后统一进行。答辩形式可能是大家上讲台答辩，也可能是老师下到同学中间检查，不用准备 ppt，只需讲解程序并回答老师提问即可。

如果仅提交老师给的示例程序，则无论比赛结果如何，本题均记 0 分。

我们鼓励大家创新思维，若对传统算法进行了创新性的改进，提出了原创性很高的解法，或在算法中融入了新颖的技巧、策略等，例如提出了一种新的估值函数、搜索策略等，可给予适当加分。

以上规则也可能在课程进行中有所调整，如果同学们有好的建议，也欢迎提出来，好的建议我们会采纳。

## 七、诚信要求：

我们允许并鼓励大家查阅资料、文献、学术论文，甚至自学一些高级技术完成本题。

在 Botzone 排行榜上有的用户公开了自己的代码，他们的代码可以参考、借鉴思路，但绝不允许照搬。不允许将其他同学或网络上的代码改头换面后作为自己开发的程序参赛。修改变量名或函数名、变换语句结构或函数位置等均无效。

可能会有同学发现有的用户有多个 bot 都公开了源代码，而这些开源 bot 并没有全出现在天梯排行榜上，抄这些没在排行榜上的 bot 老师就不会发现吧？或者有的用户公开的 bot 有几十个版本，排行榜上只是最新版，那么不抄最新版，抄前面的几个版本老师不会发现吧？

请注意：**我们会在后台统一下载大家参赛的 bot 源代码，同时下载该游戏所有公开的源代码，无论是它否出现在排行榜上，同时也搜集了 GitHub、CSDN 等网络上的所有相关代码，上述代码都一并放入查重软件一起查重。**抄袭者与抄袭者双方同论，不做区分，无论是否同班、同专业、同学院、同年级。任何形式的破解、攻击、在提交的代码中植入恶意代码等不正当竞争行为均视为作弊。

任何时候若发现有违纪情况可发邮件到 [jludatastruct@163.com](mailto:jludatastruct@163.com) 进行举报。

## 八、提示：

**本题开放性很强，可以说没有完全标准的答案**，同学们可以充分发挥聪明才智和想象力，给出自己的解法。任何基础、任何层次的学生都有能力给出解决方案。

比如可以通过不断的玩游戏，找出一些经验技巧，构思一些规则，按照固定的规则下棋。亦或是可以考察在当前局面下所有可能的下法中，执行哪种下法后产生的新局面对本方最有利（可以设定一个估值函数对每个棋盘局面进行评估，算一个分值），然后选取对本方最有利的下法。更高一点，可以采用博弈树搜索。甚至机器学习方法（当然我们并不太建议采用此法，因为这已超出了本课程的范畴）。亦或将上述几种方法相结合。

希望大家不要随波逐流，不要看别人怎么做自己就怎么做，我更希望大家有自己的见解。

对于 Botzone 的使用、游戏规则与编程的问题除了问本班的老师外，还可直接在 Botzone 讨论区（NOGO 游戏）发帖，会有更为专业的北大 Botzone 开发团队人员回答。

## 九、本题难么？

大家可以在 Botzone 中注意到，本题也是去年北京大学《计算概论》课的编程大作业（单人完成），该课相当于咱们学校的《程序设计基础》，讲的主要是 C 语言基础，也就是说这是北大大一上学期的学生做的题。所以大家要有充分的信心。



另外，对于本题，普通同学和竞赛大佬的差距可能会缩小。如果是 OJ 题目，竞赛选手经历了长期的训练，做题数量是普通同学的 10~20 倍以上，所以有着极为丰富的经验。但对于这种编程对战的形式，相信不少竞赛同学也是头一回经历。从去年的情况看，比赛中不乏普通同学战胜竞赛大佬的情况出现。所以对于普通同学来说，要有信心，只要努力，你也有机会。对于竞赛的同学来说，也不要轻敌，不要觉得简单弄弄就能夺冠，若想取得好成绩，还是需要精心投入。

当然，只要是比赛，就难免有不确定性和偶然性，大家也不要吧结果看得很重，结果仅供娱乐，在这期间编程能力得到的锻炼才是收获。

## 十、激励措施：

对于两院总冠军、亚军、季军，如果程序质量较好，且同学们积极性较高，我们会发放一个纪念奖杯，作为留念。以上所产生的所有费用，均为我自掏腰包。



获得两院总冠军和亚军的队伍，在其自愿且不违反学校疫情防控规定的前提下，我们可推荐其代表吉林大学参加 2021 年中国大学生计算机博弈大赛。去年课设的棋类博弈项目是六子棋，我全额资助了我们的冠军和亚军参加 2020 年全国大赛，两队同学表现不俗，均获得二等奖，其中 1 队与一等奖仅差一点点，失之交臂。

