****

《人工智能综合课程设计》报告

|  |  |
| --- | --- |
| 学院名称： | 自动化学院 |
| 专业班级： | 智能科学与技术 |
| 指导教师： |  |
| 姓名（学号）： |  |
|  |  |
|  |  |
| 时 间： |  |

目录

[一、 课程设计题目及要求 1](#_Toc18187)

[二、 小组分工情况 1](#_Toc16597)

[三、 系统方案设计 1](#_Toc23212)

[1. 功能分析 1](#_Toc25015)

[2. 总体方案设计 1](#_Toc20560)

[3.设计方案合理性评价 1](#_Toc13572)

[四、 算法原理 2](#_Toc604)

[1. 问题详述 2](#_Toc16680)

[2. 算法原理描述 2](#_Toc1213)

[五、 算法的改进与实现 2](#_Toc7363)

[六、 实现效果及结果分析 2](#_Toc17489)

[七、 课程设计总结及心得 3](#_Toc30453)

[八、 参考文献 3](#_Toc14988)

1. **课程设计题目及要求**

**步步为营桌游对战智能设计**

**描述**：步步为营，又名围追堵截，英文名为Quoridor，是著名游戏设计师Mirko Marchesi的作品。设计一个人工智能与其他小组对战，在规则允许的范围内尽可能获取胜利。

游戏规则为：

一个9X9的棋盘

起始状态：黑方位于棋盘上方中间格子，白方位于棋盘下方的中间格子

胜利条件：将己方的棋子，走到对方出发的那一行

1、行棋过程中，双方各执10块挡板，挡板用于阻拦对手前进。挡板横跨（或竖跨）两个格子，阻挡时不能围个圈把对方或自己堵死，且无法收回。

2、每一个回合，可以选择控制棋子移动或放置挡板的其中一个操作。可以向上下左右四个方向移动，每次移动一格，但不能越过挡板，挡板一回合只能放置一块。

3、如果正在移动棋子，并且棋子下一步正好是对方棋子的位置，则可以再走一步，此时，只能行棋不能放置挡板。

**具体要求：**

1、分析题目，明确任务，查阅文献，确定方法和开发工具。

2、设计Quoridor对战AI，能够从服务器获取棋盘当前信息，完成移动棋子或放置挡板的决策，并将正确指令发送到服务器端显示出来。

3、服务器端会统计非法操作的数量，如试图将棋子移出棋盘、放置重叠的挡板等，因此需要确保发送了正确的指令，当服务器端反馈错误信息时应能及时重发信息。

4、服务器端有用时信息统计，要求单步用时不超过10秒，最大超时数不超过3次，要求设计方案时考虑时间复杂度满足要求，并在代码优化时引入剪枝、蒸馏、并行计算等方法尽可能提高计算速度。

5、对战AI应以获取胜利为最终目标设计，成果验收阶段需与其他小组完成对战。

1. **小组分工情况**

表 1 成员分工表

|  |  |  |
| --- | --- | --- |
| 姓名 | 具体分工 | 签名 |
|  | 编写代码及调试程序 |  |
|  | 编写代码及调试程序 |  |
|  | 记录数据及编写文档 |  |

1. **系统方案设计**
2. **功能分析**

首先，游戏的棋盘是9x9的，起始位置是黑方在棋盘上方中间（也就是坐标（1,5）），白方在下方中间（坐标（9,5））。胜利条件是走到对方初始所在的行，即黑方需要走到第9行，白方需要走到第1行。

接下来是规则部分，每个玩家有10块挡板，挡板可以横向或纵向放置，覆盖两个格子。放置挡板时需要注意不能形成完全封闭的区域，导致对方或自己被堵死。挡板一旦放置就不能收回，每回合只能选择移动棋子或者放置一块挡板。

移动棋子时，可以向上下左右四个方向移动一格，但不能穿过挡板。如果移动时下一步是对方的位置，那么可以再移动一步，此时只能移动不能放置挡板。

**2. 总体方案设计**

方案结构图如下：

游戏基本信息

中文名：步步为营（围追堵截）

英文名：Quoridor

设计师：Mirko Marchesi

游戏规则

棋盘规格：9X9棋盘

起始状态：

黑方：棋盘上方中间格子

白方：棋盘下方中间格子

胜利条件：将己方棋子走到对方出发那一行

行棋规则：

挡板相关：

双方各执10块挡板；

挡板横跨（或竖跨）两个格子；

阻挡时不能围圈堵死对方或自己，且无法收回；

回合操作：

每回合可选：控制棋子移动 或 放置挡板（只能选其一）；

棋子移动方向：上下左右，每次一格，不能越过挡板；

挡板放置：一回合只能放一块；

特殊规则：

移动棋子时，若下一步是对方棋子位置，可再走一步，此步只能行棋不能放挡板

图 1 总体方案结构图

**3.设计方案合理性评价**

|  |  |
| --- | --- |
| 课程目标 | 达成课程目标的举措和评价 |
| 在人工智能领域复杂工程问题的背景下，针对智能信息处理与应用系统，应用所学人工智能等课程知识分析和调研需求，选择研究路线，制定可行的实验方案，完成具体应用场景中的智能信息处理与应用系统的设计，具备方案设计的创新性思维，能在传统架构中提出改进方案。 | 举措： 深入研究步步为营桌游规则，明确智能对战需解决的关键问题，如路径规划、挡板放置策略等。利用机器学习中强化学习的知识，选择以深度Q - 网络（DQN）为基础的研究路线。  评价：通过多轮与不同策略智能体或人类玩家对战，统计胜率、平局率等指标。对比只使用强化学习的智能体，评估结合模仿学习后策略的有效性与创新性，判断是否达成目标。 |
| 在智能信息处理与应用系统设计中，能够基于科学原理并采用科学方法对数据进行分析与解释,并通过信息综合得到合理有效的结论，并在实验报告中展示分析和调试的过程和结果。 | 举措： 收集步步为营游戏大量对战数据，包括每局棋盘状态变化、双方操作序列等。运用概率论和数理统计知识，分析棋子移动规律、挡板放置偏好等数据特征。综合多组实验数据，得出关于不同策略优劣的结论。在实验报告中，详细展示数据收集过程、分析方法选择依据，以及用图表呈现分析结果和调试中策略优化过程。  评价：以实验报告中数据分析逻辑的严密性、结论对策略改进的指导作用为评价要点。邀请专业人员评估分析方法的科学性，根据结论应用于智能体训练后的效果提升判断达成情况。 |
| 在智能信息处理与应用系统设计中，能够熟练使用Python编程工具，选择适合的实验设备，构建实验系统和测试平台。 | 举措：根据电脑硬件配置，合理分配计算资源用于智能体训练和模拟对战。选用开源游戏框架搭建步步为营游戏实验系统，在系统中集成智能决策模块。  评价：从代码是否符合Python编码规范、实验系统运行稳定性、测试覆盖率等方面评价。通过代码审查工具检查代码规范度，以系统长时间运行无崩溃及测试用例覆盖关键功能判断是否达成目标。 |
| 掌握新一代人工智能的发展趋势，通过技术调研与文献综述，持续跟踪前沿技术，持续改进方案，展望课题的未来发展。 | 举措： 基于调研，尝试将元学习思想引入步步为营智能体训练，使智能体能够更快适应不同对手策略。同时，探索用轻量级Transformer架构替换传统神经网络，提升智能体对复杂棋盘局势的表征能力。  评价：以改进方案是否提升智能体性能、对前沿技术调研是否全面深入、未来展望是否合理可行作为评价依据。通过智能体在新环境下的对战测试和专家评估判断达成情况。 |
| 能够在智能信息处理与应用的规划、设计、开发、部署及运行维护的实践中遵守工程职业规范，履行社会责任，树立正确的工程伦理观和科技报国的情怀。 | 举措： 在步步为营智能对战系统规划设计阶段，遵循软件工程规范，制定详细项目计划，明确需求分析、设计、开发、测试各阶段任务和时间节点。考虑游戏可能带来的影响，确保智能体不会采用恶意策略（如故意拖延游戏 ）。在数据收集和使用中，遵循数据隐私保护原则，不泄露玩家信息。  评价：通过检查项目文档是否规范、完整，调查玩家对游戏公平性和体验感的反馈，评估开发过程是否遵循工程职业规范、是否履行社会责任和树立正确价值观。 |

1. **算法原理**
2. **问题详述**

**如何在棋盘中做出合适的抉择,是否要移动棋子,还是要放置挡板?棋子的移动是哪个方向,哪个位置放置挡板.**

1. **算法原理描述**

**多次采用广度优先搜索算法(BFS),就是暴力遍历棋盘,找到我方到目的地的全部最短路径,然后再以对方的视角找到最短路径,判断地方和我方的最短路径差,如果我方路径更长(需要移动的步数越长),就采用放置挡板的逻辑,在绝对何处放置挡板的时候再次进行bfs,找到可以让对方路径变长并且我方路径不会变长或者我方路径变长远远小于对方路径变长的距离,从而做出决策.**

1. **算法原理**

本算法基于"双重视角路径博弈"思想，采用分层BFS策略实现动态决策：

1. **双路径计算层**：通过两次BFS分别计算我方（P1）和对手（P2）到各自目标边界的最短路径
2. **路径差异分析层**：比较双方路径长度差Δ=|L1-L2|
   * 若L1 < L2（我方路径更优）：优先执行移动
   * 若L1 ≥ L2（对手路径更优）：触发阻挡策略
3. **智能阻挡决策层**：在P2的最优路径上搜索合法阻挡点，需满足：
   * 使P2的新路径长度L2'最大化
   * 保证我方路径长度L1'变化率δ1 < 对手变化率δ2
   * 满足棋盘阻挡规则（不跨现有阻挡器）
4. **算法步骤**



图 2算法流程图

1. **算法的改进与实现**

本课程设计主要从最优挡板放置策略 、游戏时期两个方面对算法进行改进，具体改进包括以下三点：

1. **改进1**

不同于大多数算法的那么局限,目的不单纯的是延长对方的路径,而是在不影响已方路径的情况下去放置挡板,并且也不是单纯的选择一个符合的位置,并且加入的对多条最短路径的判断,可以做出对自己全部最短路径影响最小的选择,让对手的路径变长.并且加入了做优挡板策略作为逻辑的最高层,当出现在某一个地方放置挡板可以让路程差大大增加(n>3),便会优先放置挡板.

1. **改进2**

引入游戏时期概念,规定回合数,定义游戏前中后期,合理规划挡板资源.因为这个游戏就是基于挡板放置,而挡板是非常宝贵的资源,所以说得合理规划,在前期(回合数2之前),不轻易放置挡板,,因为此时放置的挡板都不是最优势挡板,并不能带来收益最大化,所以规划的挡板资源少.中期是游戏的核心博弈阶段,可以更多的去放置挡板,但是同时也要预留出少量的挡板去应对后期的重要抉择.后期也会预留出少量挡板,以便以防万一,有备无患.

1. **实现效果及结果分析**

经过以上算法策略的实现,虽然是纯逻辑程序判断,但是也表现出了一定的智能,并且计算快速,可以再规定时间内完成操作,因为这个棋盘的大小是固定的,状态空间有限,即使是开局状态空间最大的时候,一次bfs所需要的最坏执行次数为81次,计算双方最短路径时间几乎可以忽略不记,最耗费时间的是在挡板放置的这个环节,每次评估挡板都会再进行一次双方的bfs搜索,挡板的放置也以最坏的情况去考虑,再放置第一块挡板的时候所需要执行的判处次数为128次,故最坏执行次数为128\*81次,而计算机完全可以支持这个计算量,经过现实程序表现,程序运行非常快速.

1. **课程设计总结及心得**

**在本次“步步为营桌游对战智能设计”课程设计中，我们收获颇丰，经历了从理论学习到实践操作的完整过程，实现了自我能力的提升与突破。**

**1.知识与技能的增长**

**课程伊始，面对人工智能与博弈论相关知识，我们深感陌生。但通过查阅大量资料、钻研算法原理，我们逐渐掌握了基础概念。在设计智能对战程序时，从搭建基础框架到融入搜索算法，每一步都充满挑战。以极小极大算法为例，为使其适应游戏规则，我们不断调整参数、优化逻辑，过程中深刻理解了算法在策略博弈中的应用机制，编程与算法运用能力得到质的飞跃。**

**2.团队协作与沟通**

**团队协作是项目推进的关键。小组内成员分工明确，高佳愿负责算法设计，张旭阳专注代码实现，田雨稼记录数据。沟通中，我们会因算法选择产生分歧，但通过理性讨论、对比不同方案优劣，最终达成共识。这种思想碰撞不仅完善了设计方案，还让我们学会倾听与表达，提升了团队协作能力，明白良好的团队氛围和高效沟通是项目成功的保障。**

**3.问题解决与自我反思**

**项目进程并非一帆风顺，遇到过诸如棋子移动逻辑混乱、挡板放置规则冲突等问题。面对这些，我们没有慌乱，而是运用调试工具逐步排查，结合理论知识分析错误根源。解决问题的过程就是自我成长的过程，让我们学会冷静思考、灵活应对。同时，我们也反思自身不足，如知识储备不够全面、问题预见能力有待提高，明确了后续学习的重点方向。**

**4.展望与期许**

**此次课程设计虽已结束，但它为我们打开了人工智能应用的新窗口。未来，我们希望深入探索更复杂的博弈场景，尝试将强化学习等前沿技术融入其中，进一步优化对战智能策略。也期待参与更多实践项目，不断积累经验，提升专业素养，在人工智能领域持续深耕，用所学知识创造更具价值的成果。**

**这次课程设计是知识探索之旅、能力提升之途，让我们在实践中成长，为未来发展筑牢根基。**

1. **参考文献**

* **Browne, C.** (2011). Classic Board Games. In "Modern Board Games: Design and Analysis" (pp. 45-60). A K Peters/CRC Press.
  + 对 Quoridor 等经典棋盘游戏的策略分析。
* **Saffidine, A., Cazenave, T., & Méhat, J.** (2012). "Retrograde Analysis for Quoridor." IEEE Transactions on Computational Intelligence and AI in Games, 4(3), 216-223.
  + 使用回溯分析方法优化 Quoridor 的搜索策略。
* **Silver, D., Hubert, T., Schrittwieser, J., et al.** (2018). "A General Reinforcement Learning Algorithm that Masters Chess, Shogi, and Go through Self-Play." Science, 362(6419), 1140-1144.
  + AlphaZero 框架研究，部分方法适用于 Quoridor AI 开发。
* **Silver, D., Hubert, T., Schrittwieser, J., et al.** (2018). "A General Reinforcement Learning Algorithm that Masters Chess, Shogi, and Go through Self-Play." Science, 362(6419), 1140-1144.
  + AlphaZero 框架研究，部分方法适用于 Quoridor AI 开发。
* **Zhang, L., & Li, Y.** (2021). "Developing a Quoridor Game Engine for Algorithmic Strategy Analysis." Journal of Game Development, 16(2), 45-62.
  + Quoridor 游戏引擎的实现及其在算法研究中的应用。