**每日日志\_5（2025年7月4日）**

**一、今日已完成**

今日进行了里程碑步骤七和步骤八的任务。

步骤七：实现了五子棋棋盘状态的更新和计算某一位置周围四条线的状态。首先，棋盘用字符串表示，利用 getIndex 函数将棋盘的坐标转换为线性索引，并根据命令序列更新棋盘。每次下棋时，黑方用 'x'，白方用 'o'，根据步数判断轮流下棋。然后，使用 getLine 函数计算每个给定坐标周围的四条线（横线、竖线、斜线）的状态，并将结果拼接成字符串。最后，将计算结果提交给服务器并输出响应。

步骤八：实现了基于五子棋棋局的状态评估和最优下棋点选择。通过与服务器交互，代码首先根据命令序列更新棋盘，并根据规则和得分计算每个可能位置的最佳下棋点。具体步骤包括：首先，通过 getIndexNum 函数将棋盘坐标转换为线性索引，利用 getLine 函数获取每个位置周围的四条线状态；然后，定义了规则和得分系统 RuleWithPoints 来评估每个空位的得分；接着，getMaxCoords 函数根据得分选择最佳下棋点。最后，通过 getNextStep 函数将结果提交给服务器，获取下一题的链接。

**二、遇到的问题、心得体会**

在这过程中，我学到了如何通过加密保证安全、如何优化网络请求以提升效率、如何处理复杂的棋局评估规则，以及如何平衡实时性和系统效率，确保游戏体验更加流畅和高效。