Today's Progress Summary of Mecha Chess Project

Date: December 15, 2025

Core Promotion Direction: Clarify the implementation path of the core gameplay "Move-Driven Plot", finalize a lightweight implementation plan, avoid early development complexity, and prioritize verifying the functional closed loop.

I. Confirmation of Core Gameplay Direction

It is determined that "Move-Driven Plot" will be the core characteristic gameplay of the project, with the core logic anchored in the strategic essence of "The art of war is based on deception": the player's move action + the current game state (advantage/disadvantage/draw) triggers the corresponding plot. The plot must be bound with the dual effects of "tactical gain + tactical risk", directly affecting subsequent chess game rules (such as exposing pieces, prohibiting movement, evading attacks, etc.). Numerical development is rejected throughout to ensure a pure strategic game experience.

II. AI Plot Scheme Planning (Focusing on the Core Path of Local Training)

The core direction of the AI plot scheme is clarified as local training, focusing on the model development of the "move-plot" correlation logic. The specific plan is as follows:

1. Core Path (Local Training): Prioritize the advancement of local AI model training. The training data are paired samples of "move action + game state + faction + plot + tactical effect", focusing on the model's adaptability to the generation of tactical plots;
2. Alternative Scheme (Lightweight Backup): If obstacles are encountered in the early stage of local training, mature large model APIs can be temporarily used to quickly verify functions, and switch back to the core path when local training conditions are mature.

III. Plot Presentation: Function First, Lightweight Implementation

It is clarified that complex 3D plot animations will not be made in the early stage, focusing on function verification and adopting a low-cost presentation plan:

1. Core Prompt: UI pop-up + typewriter text effect (distinguished by faction-specific color matching) + appropriate sound effects (industrial heavy faction's steel collision sound, sci-fi tech faction's current sound, etc.) to intuitively convey plot information;

2. Tactical Feedback: Enhance the plot's impact on the game through simple special effects such as highlighted apertures on the chessboard (exposing enemy pieces) and piece status markers (such as red warning icons, evasion buff indicators);
3. Atmosphere Supplement (Optional): Create 3-5 static illustrations for each faction, use them as pop-up backgrounds when triggering plots, and add slight zoom animations to enhance immersion at low cost.

IV. Minimum Viable Function Closed-Loop Planning

Determine the core early development path to quickly verify the gameplay logic:

1. Build Basic Framework: Use Excel/Notepad to establish a comparison table of "move action - game state - faction - plot - tactical effect" to realize a preset plot library;
2. Develop Trigger Logic: Implement condition detection through simple scripts, and automatically match the corresponding content in the preset plot library after the player makes a move;
3. Rule Implementation: Convert plot effects into chess game rule tags (such as "prohibit movement" and "evade one attack") to ensure that the plot effectively affects the game process.

V. Late-Stage Iteration and Upgrade Path

After the core functions are verified, conduct phased advanced optimization:

1. Plot Richness Upgrade: With the locally trained model as the core, continuously optimize the model based on the "move-plot" data accumulated in the early stage to realize infinite dynamic plot generation; if API backup was used in the early stage, the high-quality plot data generated by the API can be supplemented as training samples for the local model to improve the model effect;
2. Presentation Effect Upgrade: Combine modeling and Unreal Engine technology to upgrade the lightweight presentation plan to 3D plot animation and enhance visual immersion.

VI. Follow-Up Promotion Plan

Next, prioritize the development of the first plot demo (such as the "Vanguard Reconnaissance" plot of the industrial heavy faction), focus on script writing and UI material production, and complete the full functional verification of "move triggering plot - effect feedback".

中文翻译 / Chinese Translation

机甲象棋项目今日进展总结

日期：2025-12-15

核心推进方向：明确「走子驱动剧情」核心玩法落地路径，敲定轻量化实现方案，规避前期开发复杂度，优先验证功能闭环。

一、核心玩法方向敲定

确定以「走子驱动剧情」为项目核心特色玩法，核心逻辑锚定“兵者诡道”的策略内核：玩家走子动作+当前棋局状态（优势/劣势/平局）触发对应剧情，剧情需绑定“战术收益+战术风险”的双重效果，直接影响后续棋局规则（如暴露棋子、禁止移动、闪避攻击等），全程拒绝数值养成，保障纯策略博弈体验。

二、AI剧情方案规划（聚焦本地训练核心路径）

明确AI剧情方案核心方向为本地训练，聚焦“走子-剧情”关联逻辑的模型开发，具体规划如下：

1. 核心路径（本地训练）：优先推进本地AI模型训练，训练数据为「走子行为+棋局状态+派系+剧情+战术效果」的配对样本，聚焦模型对战术剧情的生成适配性；
2. 备选方案（轻量化兜底）：若本地训练初期遇阻，可临时采用成熟大模型API快速验证功能，待本地训练条件成熟后切换回核心路径。

三、剧情呈现：功能优先，轻量化落地

明确前期不做复杂3D剧情动画，聚焦功能验证，采用低成本呈现方案：

1. 核心提示：UI弹窗+打字机文本效果（派系专属配色区分）+ 适配音效（重工派钢铁碰撞声、科技派电流声等），直观传递剧情信息；
2. 战术反馈：通过棋盘高亮光圈（暴露敌方棋子）、棋子状态标记（如红色警告图标、闪避buff标识）等简单特效，强化剧情对棋局的影响；
3. 氛围补充（可选）：制作各派系3-5张静态插画，触发剧情时作为弹窗背景，添加轻微缩放动画，低成本提升沉浸感。

四、最小可行功能闭环规划

确定前期核心开发路径，快速验证玩法逻辑：

1. 搭建基础框架：用Excel/记事本建立「走子动作-棋局状态-派系-剧情-战术效果」对照表，实现预设剧情库；
2. 开发触发逻辑：通过简单脚本实现条件检测，玩家走子后自动匹配预设剧情库中的对应内容；
3. 规则落地：将剧情效果转化为棋局规则标签（如“禁止移动”“闪避一次攻击”），确保剧情切实影响博弈过程。

五、后期迭代升级路径

待核心功能验证通过后，分阶段进阶优化：

1. 剧情丰富度升级：以本地训练模型为核心，基于前期积累的“走子-剧情”数据持续优化模型，实现剧情无限动态生成；若前期使用过API兜底，可将API生成的优质剧情数据补充为本地模型的训练样本，提升模型效果；
2. 呈现效果升级：结合建模与虚幻引擎技术，将轻量化呈现方案升级为3D剧情动画，提升视觉沉浸感。

后续推进计划

下次优先推进第一个剧情demo开发（如工业重工派“先锋侦查”剧情），聚焦脚本编写与UI素材制作，完成“走子触发剧情-效果反馈”的完整功能验证。

（注：文档部分内容可能由 AI 生成）