
# DTS311TC FINAL YEAR PROJECT

## *Player-Aware Intelligent Monitoring and Operations Navigator*

## Proposal Report

In Partial Fulfillment
of the Requirements for the Degree of
Bachelor of Engineering


| | | |
|---|---|---|
| Student Name | : | Taimingwang Liu |
| Student ID | : | 2037690 |
| Supervisor | : | Xihan Bian |

School of AI and Advanced Computing
Xi'an Jiaotong-Liverpool University

November 2025

# Abstract

Apply the font of Times New Roman to the paragraphs of the abstract using font size of 12. An abstract is usually one to three paragraphs long with a length of 150 to 350 words.

# Contents

# 1 Introduction

## 1.1 Problem Setting & Motivation

近年来，面向玩家的智能交互快速涌现：从*AI*游戏主播/虚拟角色到*LLM*驱动的*NPC*/插件，社区与产业侧案例表明"大模型×游戏交互"具备显著关注度与潜在影响（game changer potential）。然而，这些案例多为定制工程，缺乏统一接口与可复现实验协议。本文聚焦伴随式（*companion-style*）实时助手，在统一动作接口与低延迟体验的约束下，探索一条仅基于*GUI*（*GCC*）即可落地、并具备可复现性的系统路径。

**Industry/Community Signals** 除学术工作外，社区与产业侧的"AI×游戏/直播"案例为本研究提供现实动机。例如*Neuro-sama* 及其开源复刻框架[1]–[4]，以及叙事解谜作品*AI2U: With You 'Til The End*[5] 展示了"对话即操作"（dialogue-as-action）与高交互度（LLM-controlled NPCs）的设计可能性。我们据此聚焦"游戏+ 大模型交互"的可复现路径；这些案例不作为方法有效性的学术证据，研究问题将落在统一动作接口（*GUI/GCC*）、低延迟与评测协议上。

## 1.2 Scope & Working Definitions

我们采用**General Computer Control (GCC)** 的*GUI*范式：*screen-in, keyboard/mouse-out*的人类同态接口（human-homomorphic interface）。代表作*Cradle* 证明了在不依赖应用API的前提下完成长链路桌面/游戏任务的可行性[6]。在此基础上，本文明确聚焦**GUI**，不进行应用*API*的单独适配（尽管在可行时API可能进一步提升*determinism & efficiency*，但其工程成本与维护负担超出本文范围）。相应地，我们以***Model Context Protocol (MCP)*** 作为内部**"技能/工具总线"**：将*GUI*技能（*skills/macros*）、规划/记忆/反思（*planning/memory/reflection*）等模块以*MCP*风格进行注册与编排，统一在GCC通道上执行。

作为与本文评测设置相关的代表性工作，*Orak* 提供覆盖多类型真实电子游戏的统一基准，并以*MCP* 实现*plug-and-play* 的代理—环境对接；其*Leaderboard/Battle Arena* 与*agentic modules* 消融，为比较不同模块与输入模态提供了统一框架[7]。我们参考其"统一评测—模块消融—可复现配置"的思路，但将输出接口限定为*GUI*（*GCC*），并把*MCP* 用作内部技能编排协议而非外部应用API适配层。

作为开放式环境下的系统评测参考，我们采用基于*procedural generation*的统一框架来度量*VLA/VLM*在多步轨迹与OOD设定中的表现，并将架构/数据/输出后处理作为可控变量纳入对比[8]。

我们参考*lmgame-Bench* 的"游戏→可靠评测"思路：以统一*Gym-style API* 与轻量*perception/memory scaffolds* 控制提示方差（*prompt variance*）与污染（*contamination*），并度量跨游戏与多步任务的泛化（*generalization*）表现；实现上我们仍统一采用**GUI**执行，*MCP-style* 仅用于内部技能编排（*skill bus*）[9].

**Screenshot-only navigation (GUI).** 我们将“GUI通道”具体化为纯截图输入与结构化动作输出的端到端导航设定。近期工作显示，不依赖*planner*或*A11y*树，仅凭截图亦可在真实平台上取得SOTA（如UI-Venus在AndroidWorld达 65.9% pass@1，并在ScreenSpot-V2/Pro上达到 95.3%/61.9%）[10].

作为视觉中心评测的代表性工作，V-MAGE 以*visual-only/continuous-space* 的游戏设定系统考察定位、轨迹、时机与视觉记忆等要素，并使用Elo 风格排名报告相对强度**v-mage**.

## 1.3 Key Challenges

**TODO:** 长链路稳定性、UI变化鲁棒、延迟预算、权限安全与回滚、跨游戏迁移；并注明我们仅依赖GUI控制带来的特定挑战（如确定性与重试策略）。

## 1.4 Our Positioning & Contributions

**TODO:** (1) **GUI/GCC** 的伴随式助手原型；(2) 以**MCP** 作为“技能/工具总线”统一规划/记忆/反思/技能组并落到GUI执行；(3) 提出*advice adoption, voice RTT, macro success*等伴随式指标与统一评测设置；(4) 讨论API接入的潜在收益但不纳入本文范围。

我们借鉴动作粒度的结构化输出思想（format/type/coord/content），以严格schema减少无效动作与走偏坐标；同时在评测中引入“宏技能成功率（macro success/recall）”以覆盖稀疏关键动作[10].

## 1.5 Design Principles & System Preview

**TODO:** 一句话系统流：*screen/audio* → *VLM* → *LLM/agentic*（planning/memory/reflection）→ *MCP-skill bus*（技能注册/路由）→ *GUI*执行（kb/mouse）→ *safety*（permissions, rollback, kill-switch）。

# 2 Literature Review

## 2.1 Perception: Modalities & Grounding

**TODO:** 视觉为主（screen/video）+ 可选音频（audio）；VLM能力：检测/描述/grounding；可简单对照VLA（直接产出action tokens）与VLM+tool的差别。与本文：选轻量VLM，优先本地（on-device）与流式ASR/TTS。

## 2.2 Action Interfaces: GUI (GCC) & MCP-style Orchestration

在动作接口上，**GUI** 路线以**General Computer Control (GCC)** 为统一通道（*screen-in, keyboard/mouse-out*），强调对不同应用/游戏的可迁移性（portability）与统一的人类同态交互（human-homomorphic interface）。代表性工作*Cradle* 显示出在不依赖应用专用接口的前提下，仍可通过规划—技能整理（*skill curation/registry*）—反思—记忆的管线完成长链路任务（desktop/games），从而为*GUI* 的可行性提供了强证据（evidence）。与此同时，**MCP**（Model Context Protocol）为模块注册/编排（*module registration/orchestration*）提供协议化思路：在不改变输出仍为*GUI*的前提下，可将*skills/macros*、*planning*、*memory*、*reflection*等以统一接口组织起来，便于系统性消融与复用（plug-and-play）。与本文：我们采用*GUI*（*GCC*）作为唯一执行通道，并借鉴*MCP* 的注册/编排思想作为内部“技能总线（skill bus）”，统一路由与调用*skills/macros* 等模块。[6], [7]

**Takeaway** 基于文献可见：*GUI*（*GCC*）为跨应用/跨游戏提供了统一接口与较低的移植门槛；协议化编排（如*MCP*）可在不更改*GUI*输出的条件下提升模块化与可复现性。与本文：坚持*GUI*输出，内部采用*MCP-style* 编排以获得结构化、可消融的系统形态。

...Cradle 以统一的GUI（GCC）通道展示了从“屏幕输入→内在推理→键鼠控制”的闭环，可作为GUI 可行性的强证据（见Figure 1）。

**UI-Venus（screenshot-only）** 端到端GUI导航，无需planner/A11y，强调截图→结构化动作的通道仍能在真实平台达SOTA（AndroidWorld 65.9% pass@1），对我们“GUI优先、统一动作模式”的落地具有直接支撑[10].

## 2.3 Agentic Modules: Planning, Memory, Reflection, Skills

**TODO:** 规划（planning）、记忆（memory, 用户偏好/历史）、反思（self-reflection, 纠错/风格一致）、技能库（skills/macros, 原子→复合）。与本文：直接采纳skills+reflection+memory组合，并通过“技能总线”统一管理。

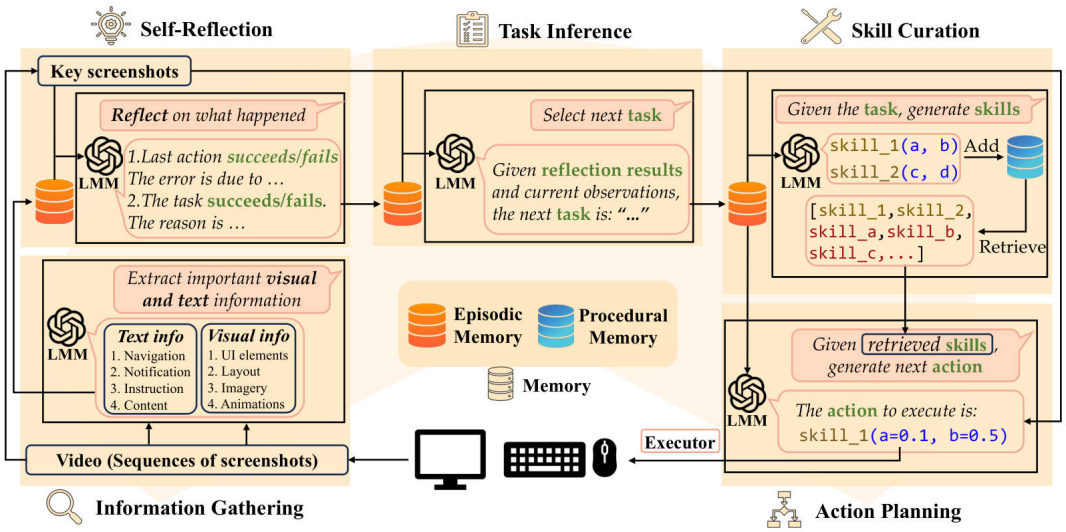


Figure 1: An overview of the CRADLE framework: CRADLE takes video from the computer screen as input and outputs computer keyboard and mouse control determined through inner reasoning (planning, skill curation, reflection, memory) [6].

历史对齐与稀疏动作增强　提出*Self-Evolving Trajectory History Alignment & Sparse Action Enhancement*：用当前模型重写历史"思维—动作"轨迹以对齐风格/细节，并上采样稀疏但关键动作（如LongPress），改善长链路一致性与泛化**uivenus_rft**.

## 2.4 Learning Paradigms: Zero-shot, RAG, Finetune, IL/RL, Distillation

**TODO:** 零样本/提示工程、检索增强（RAG for UI schema/FAQ）、轻量微调（LoRA）、模仿/强化（IL/RL）、蒸馏到小模型。与本文：优先零样本+RAG，必要时小规模LoRA以稳UI。

以"指令化（instructionalization）"增强RL 代理的上下文理解是一条有效路径。*R2-Play* 将多模态游戏指令（MGI）并入*Decision Transformer*，形成*DTGI*，并通过超网络（*SHyperGenerator*）在训练任务与未见任务间共享知识；作者报告多模态指令相较于文本或轨迹单模态在多任务与泛化上更优（动机见Figure 2）。MGI 的三段式结构——*game description*、*game trajectory* 与*game guidance*（含动作、语言引导及关键元素位置）——为"指令卡"提供了清晰模板（见Figure 3）。对我们而言，这更像"数据与提示工程"的启发：为*GUI* 技能准备轻量的"多模态说明卡"（示例帧+要点文本），在*MCP-style* 的内部技能总线下统一注册与调用，输出仍保持*GUI*（kb/mouse）[11].

**RFT（GRPO）与动作粒度奖励**　将奖励拆分为格式/动作类型/坐标/内容四部分并加权，既鼓励结构化输出合规，又提升细粒度定位/文本输入正确性，是近期GUI导航RL-finetune的代表路径**uivenus_rft**.

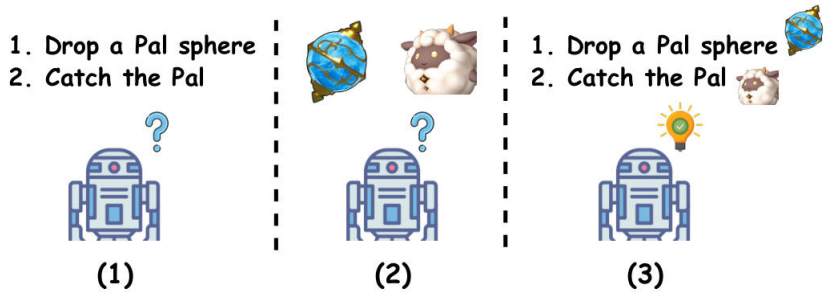


Figure 2: Imagine an agent learning to play Palworld. (1) The agent exhibits confusion when only relying on textual guidance. (2) The agent is confused when presented with images of a Pal sphere and a Pal. (3) The agent understands how to catch a pet through *multimodal guidance*, which combines textual guidance with images of the Pal sphere and Pal [11].


Figure 3: An illustrative example of *multimodal game instructions (MGI)*. Each instruction consists of three sections: *game description*, *game trajectory*, and *game guidance* (including action, language guidance, and the position of key elements) [11].

## 2.5 Benchmarks & Datasets (OS-like, Games, Desktop)

以真实多类型游戏为对象的统一评测框架正在兴起：*Orak* 通过*MCP* 实现*plug-and-play* 的代理—环境解耦，并在统一配置下检验*planning / reflection / memory / skills* 等*agentic modules* 的边际贡献（ablation），配套*Leaderboard/Battle Arena* 的对比体例与用于训练的轨迹数据（fine-tuning trajectories）。这类框架的价值在于：机制—性能—配置三者被一体化呈现，既利于横向（模型/模态）又利于纵向（策略/模块）比较。与本文：我们借鉴其"统一评测+ 消融"的组织方式，但将输出统一为*GUI*，并用*MCP-style* 在内部完成技能与模块的注册/编排。[7]

与以真实多类型游戏构建统一评测与消融的框架相互补充，一条重要的发展脉络是基于*procedural generation*的开放式评测：在可控生成下构造*OOD*与多步任务压力，统一比较*VLA/VLM*在架构/训练数据/输出后处理等变量下的泛化与稳健性，并配套可复用的工具链以保证*reproducibility*。我们采纳其"协议一致、变量可控"的组织方式，但将动作执行统一为*GUI*，并用*MCP-style*在内部编排*skills/macros*与*agentic modules*以适配我们的场景需求[8].

与面向真实多类型游戏的统一评测/消融框架互补，*lmgame-Bench* 将"游戏→评测"系统化：用*Gym-style* 接口与*perception/memory scaffolds* 稳定*prompt*、剔除污染，在多模型下获得良好分离度，并通过相关性分析（*correlation analysis*）展示"各游戏探测的能力混合并不相同"；进一步，单一游戏的*RL* 训练对未见游戏与外部规划任务出现迁移（transfer）。本文沿用其"协议一致、变量可控"的评测方法学，但执行端统一为**GUI**，并以*MCP-style* 在内部编排*skills/macros* 与*planning/memory/reflection* 以做可复现实验**hu2505lmgame**.

.....为减少提示方差并抑制污染，lmgame-Bench 以模块化脚手架稳定"感知—记忆—推理"的交互回路（见Figure 4），从而把"游戏→可靠评测"落到可复现协议之中。

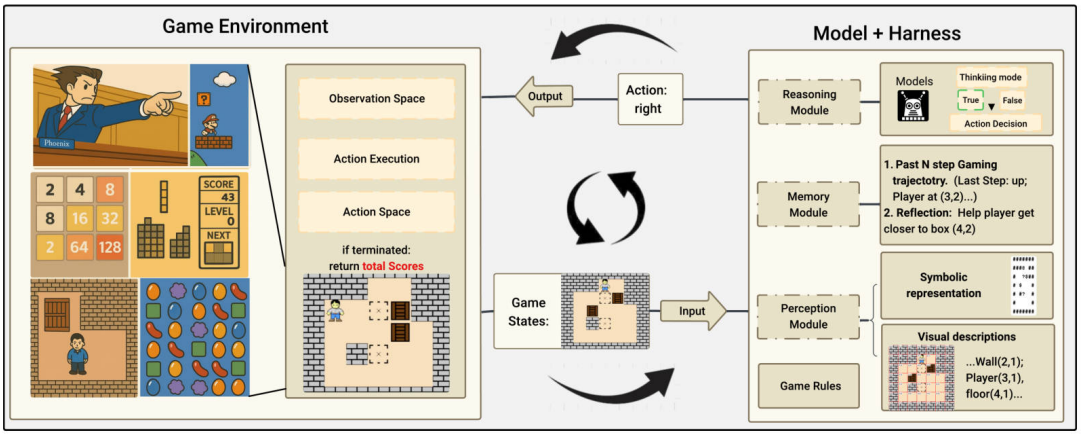


Figure 4: lmgame-Bench uses modular harnesses—such as perception, memory, and reasoning modules—to systematically extend a model's game-playing capabilities, enabling iterative interaction loops with a simulated game environment **hu2505lmgame**.

**Benchmarks** 在线：AndroidWorld（实时多步交互）；离线：AndroidControl、GUI-

Odyssey — UI-Venus在三者上报告系统对比与SOTA/可比结果**uivenus_rft**.

**V-MAGE**（**vision-centric, visual-only, continuous-space**） 该框架以仅视觉输入与连续空间的游戏环境，聚焦评测多模态模型的视觉中心能力，覆盖定位、轨迹追踪、时机、视觉记忆及更高层时序推理等要素；其评测管线支持分离“模型/策略”，并采用Elo风格排名进行相对强度比较。作者报告现有模型与人类表现存在差距，常见感知错误与锚定偏差，且有限历史上下文会限制长时规划**v-mage**.

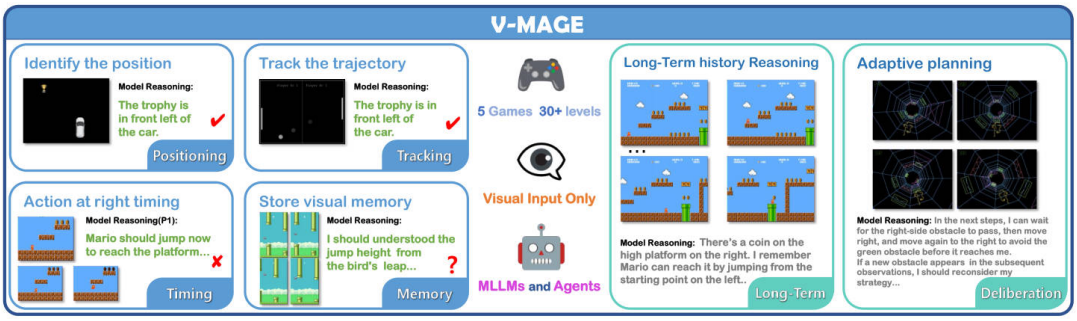


Figure 5: The overview of the V-MAGE benchmark, designed to evaluate vision-centric capabilities and higher-level reasoning of MLLMs across 5 free-form games with 30+ levels. V-MAGE assesses critical abilities in visual reasoning, providing a comprehensive evaluation of model performance in complex, dynamic environments.

## 2.6 Evaluation Protocols & Metrics

**TODO:** 客观：success rate, time-to-completion, no-misclick/rollback rate, latency（voice RTT, frame→hint时间）；主观：advice adoption, user satisfaction。与本文：将advice adoption与macro success设为核心指标，配套延迟与稳定性度量。

为避免实现细节带来的不可比性，我们将输出结构化与解码约束（structured output & constrained decoding）纳入统一协议：动作以JSON Schema 与白名单规范，经MCP-style skill bus 路由后由GUI 执行；并以Invalid Action Rate 作为守门指标（目标近零）。在效果度量上，我们采用跨技能均值（macro-averaged success/recall）为主视角，辅以micro 指标；对稀疏/时序敏感技能，额外报告机会归一化成功率（OAS）、反应时延（**RT**）与每次机会尝试数（**APO**）以刻画稳定性与可用性。[8]

我们将“输出后处理（*post-processing*）”（如技能解码、重试/回滚策略）显式纳入可控变量，配合稳定*prompt* 与污染控制的协议，减少实现细节对可比性的干扰**hu2505lmgame**.

文献中亦有采用*Elo-style ranking* 进行跨任务相对强度比较的做法，V-MAGE 为其中一例**v-mage**.

## 2.7 Deployment & Real-time Considerations

**TODO:** 本地/云混合、量化（INT4/FP8）、流式解码、语音中断（barge-in）、资源占用与帧率影响。与本文：给出延迟预算（如≤ 500ms 提示、≤ 1.5s 语音回路）。

## 2.8 Safety, Permissions & Robustness

**TODO:** 权限模型（whitelist, scope）、操作确认、影子模式（shadow mode）先预测后执行、回滚/急停。与本文：作为系统必要模块并与“技能总线”联动。

如Figure 6 所示，think–action 不一致（mismatch）揭示了MLLM 的“幻觉”（hallucination）风险。[10]

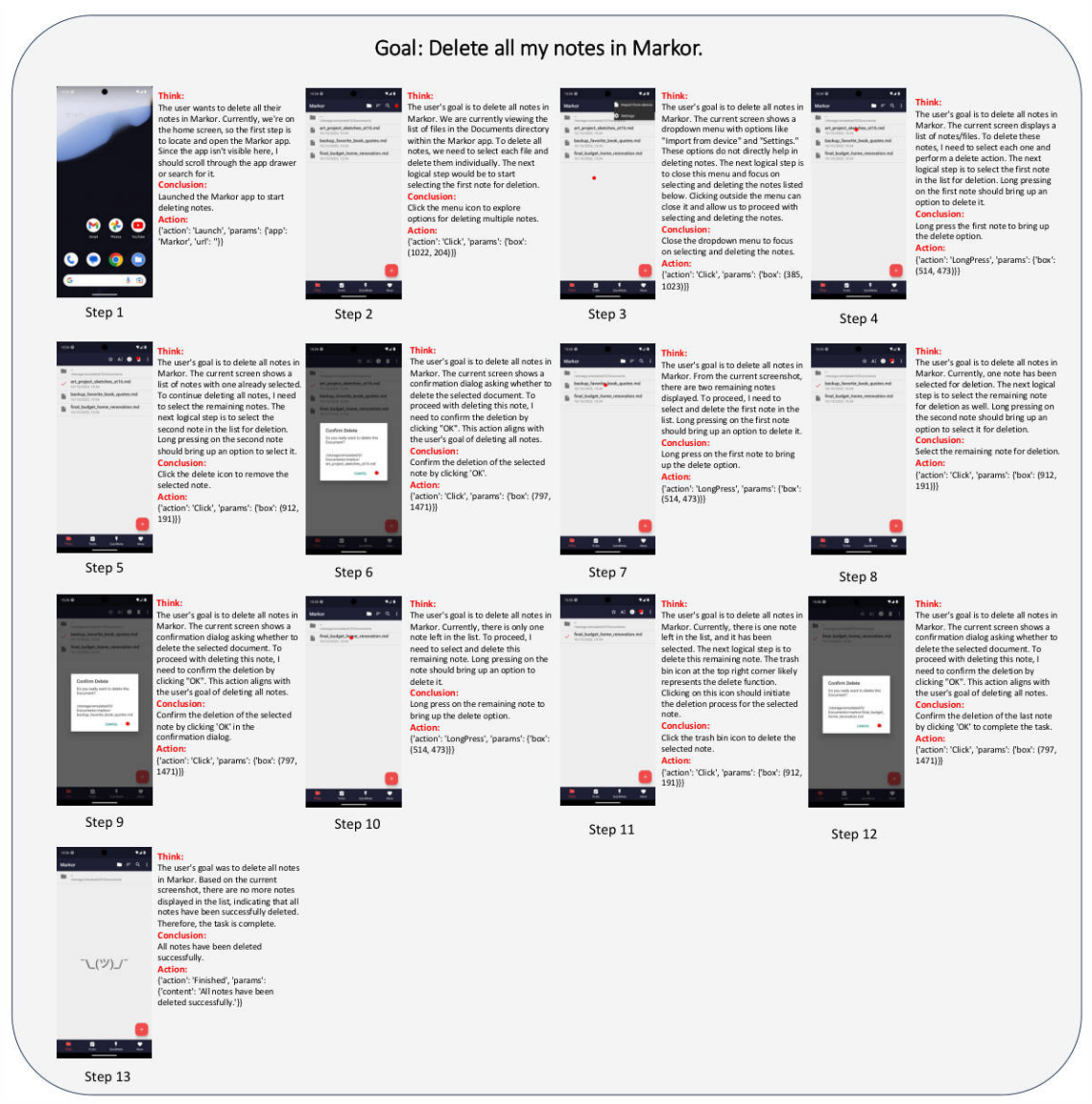


Figure 6: One trace of UI-Venus on the task named MarkorDeleteAllNotes in AndroidWorld. We can observe that UI-Venus successfully achieves the goal and has the reflection ability in Step 3. However, there also exists the conflict between think and action in Step 5, remaining as a future work about how to solve MLLM’s hallucination.[10]

## 2.9 Synthesis: Trends, Gaps & Our Niche

当前趋势是在*GUI*（*GCC*）的统一通道上引入协议化/模块化的编排（如*MCP-style*），以便做机制消融与可复现实验；真实多类型游戏的统一评测（如*Orak*）正在成为共识。缺口在于：实时伴随式（*companion-style*）场景仍缺乏围绕语音互动与低延迟体验的专门指标与协议。与本文：我们将以*GUI*为唯一执行通道，结合*MCP-style* 编排与伴随式指标，给出可复现的小型协议与演示设置。

# 3 Project Plan

## 3.1 3.1 Proposed Solution / Methodology

This section contains the methodology, technical design for the project.

## 3.2 Experimental Design

This section contains the methodology, technical and experimental design for the project.

## 3.3 Expected Results

This section contains the expected results.

## 3.4 Progress Analysis and Gantt Chart

This section contains the progress analysis and Gantt chart.

# 4 Conclusion

# References


[1] Vedal and Neuro-sama, *Neuro-sama official youtube channel*, https://www.youtube.com/@Neurosama, Accessed: 2025-10-10, 2022.

[2] O.-L.-V. contributors, *Open-llm-vtuber: An open-source ai vtuber framework*, https://github.com/Open-LLM-VTuber/Open-LLM-VTuber, Accessed: 2025-10-10, 2025.

[3] kimjammer, *Neuro: A local-model recreation of neuro-sama*, https://github.com/kimjammer/Neuro, Accessed: 2025-10-10, 2025.

[4] moeru-ai, *Airi: Ai waifu / virtual character container inspired by neuro-sama*, https://github.com/moeru-ai/airi, Accessed: 2025-10-10, 2025.

[5] AlterStaff, *Ai2u: With you 'til the end*, https://store.steampowered.com/app/2880730/AI2U_With_You_Til_The_End/, Accessed: 2025-10-10, 2025.

[6] W. Tan *et al.*, "Cradle: Empowering foundation agents towards general computer control," *arXiv preprint arXiv:2403.03186*, 2024.

[7] D. Park *et al.*, "Orak: A foundational benchmark for training and evaluating llm agents on diverse video games," 2025, arXiv:2506.03610. arXiv: 2506.03610.

[8] P. Guruprasad, Y. Wang, S. Chowdhury, H. Sikka, and P. P. Liang, "Benchmarking vision, language, & action models in procedurally generated, open ended action environments," *arXiv preprint arXiv:2505.05540*, 2025.

[9] L. Hu *et al.*, "Lmgame-bench: How good are llms at playing games?, 2025a," *URL https://arxiv. org/abs/2505.15146*,

[10] Z. Gu *et al.*, "Ui-venus technical report: Building high-performance ui agents with rft," *arXiv preprint arXiv:2508.10833*, 2025.

[11] Y. Jin *et al.*, "Read to play (r2-play): Decision transformer with multimodal game instruction," *arXiv preprint arXiv:2402.04154*, 2024.

# Appendix A. Title of Appendix A

## A.1 Appendix Heading 1

Text of the appendix goes here

## A.2 Appendix Heading 2

Text of the appendix goes here

## A.3 Appendix Table and Figure Captions

In appendices, table and figure caption labels and numbers are typed in manually (e.g., Table A1, Table A2, etc.). These do not get generated into the lists that appear after the Table of Contents.

# Appendix B. Title of Appendix B

Text of the appendix goes here if there is only a single heading.