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，探讨一条仅基于*GUI*（*GCC*）的可复现实证路径。

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。这些案例不作为方法有效性的学术证据；本文的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]。本文聚焦*GUI*（*GCC*）通道；*Model Context Protocol (MCP)* 被视为内部模块/技能编排的思路（registration/orchestration），与输出通道无关。

与本文话题相关的代表性工作包括：*Orak*（统一评测/模块消融/对接协议）[7]；基于*procedural generation* 研究OOD与变量可控性的评测方法[8]；以及将真实游戏转化为可复现评测的*lmgame-Bench*[9]。此外，*UI-Venus* 强调纯截图（*screenshot-only*）输入与结构化动作输出的端到端导航，不依赖planner或A11y树[10]；*V-MAGE* 强调*visual-only/continuous-space* 的视觉中心评测**v-mage**。本文在相关工作部分对其进行系统梳理。

Screenshot-only navigation (GUI). 文献显示，在真实平台上，纯截图输入+ 结构化动作输出也可实现端到端导航并取得有竞争力的结果（如AndroidWorld *pass@1* 指标，以及ScreenSpot系列定位能力）[10]。

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** 的伴随式助手设定；(2) **MCP** 作为内部“技能/工具总线”进行注册/编排（与输出通道无关）；(3) 本文提出的评测协议与指标（advice adoption, voice RTT, macro success 等）；(4) 论文结构与开源计划（如有）。

1.5 Design Principles & System Preview

TODO: 一句话系统流：*screen/audio* → *VLM* → *LLM/agentic*（planning/memory/reflection）→ *MCP-style*（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。与本文关系：本节仅界定术语与能力范畴。

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*），强调跨应用/跨游戏的可迁移性（portability）与人类同态交互（human-homomorphic interface）。代表性工作*Cradle* 展示了在不依赖应用专用接口的前提下，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），为*GUI* 可行性提供了实证支持。与此同时，**MCP**（Model Context Protocol）提供了模块注册/编排（*module registration/orchestration*）的协议化思路：在不改变输出仍为*GUI* 的条件下，*skills/macros*、*planning*、*memory*、*reflection* 等可在统一接口下组织，便于可复现实验与消融比较。[6], [7]

Takeaway 文献显示：*GUI*（*GCC*）提供统一接口与较低移植门槛；协议化编排（如*MCP*）有助于模块化与复现性。与本文关系：本节作为动作接口背景与术语界定。

…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（见Figure 1）。

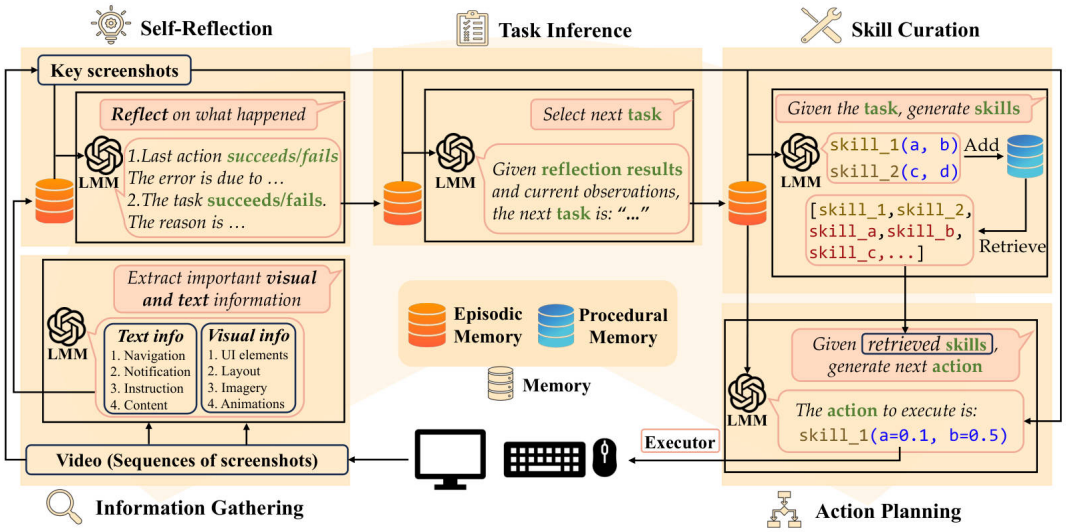

Figure 1: An overview of the CRADLE framework: CRADLE takes video from the computer screen as input and outputs computer keyboard and mouse control determined through inner reasoning (planning, skill curation, reflection, memory) [6].

UI-Venus（screenshot-only） 端到端GUI 导航，无需planner/A11y；强调截图→结构化动作的通道在真实平台可达到具有竞争力的结果（如AndroidWorld pass@1与ScreenSpot 系列定位）[10]。与本文关系：作为文献实例表明screenshot-only 路线的可行性。

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）。与本文关系：仅作机制分类与代表性做法的回顾。

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），以改善长链路一致性与泛化**uivenus_rft**。与本文关系：作为处理长链路与长尾动作的文献做法。

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。与本文关系：范式综述，不含实现承诺。

以"指令化（instructionalization）"增强RL 代理的上下文理解是一条代表性路线。*R2-Play* 将多模态游戏指令（MGI）并入*Decision Transformer*（*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较文本/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*、*game guidance*（含动作、语言引导及关键元素位置）——给出了指令模板（见Figure 3）[11]。与本文关系：作为"指令化/Decision Transformer"方向的文献背景。

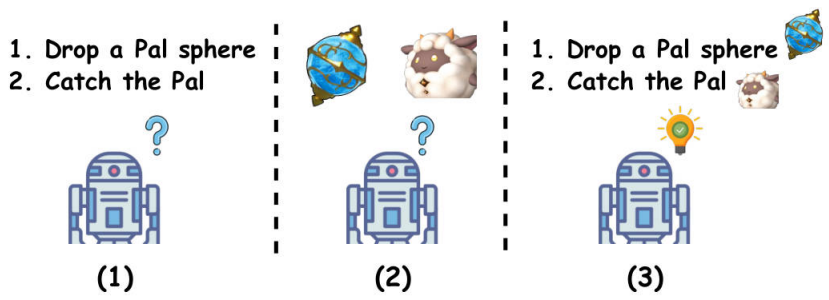

Figure 2: Imagine an agent learning to play Palworld. (1) The agent exhibits confusion when only relying on textual guidance. (2) The agent is confused when presented with images of a Pal sphere and a Pal. (3) The agent understands how to catch a pet through *multimodal guidance*, which combines textual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

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，以同时度量结构化输出合规性与细粒度定位/文本输入正确性，作为GUI 导航中RL-finetune 的代表做法之一**uivenus_rft**。与本文关系：作为RL-finetune 在GUI 导航中的奖励设计范例。

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

Orak（统一评测/消融/MCP思想） *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* 与训练轨迹数据（fine-tuning trajectories），将机制—性能—配置一体化呈现[7]。与本文关系：作为"统一评测与消融"的代表性基准。

Procedural-generation（OOD方法学） 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，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，并配套工具链以保证*reproducibility*[8]。与本文关系：作为OOD/变量可控的评测方法学背景。

lmgame-Bench（脚手架与污染控制） *lmgame-Bench* 将“游戏→评测”系统化：用*Gym-style* 接口与*perception/memory scaffolds* 稳定*prompt* 并剔除污染，在多模型下获得良好分离度，并通过相关性分析展示“各游戏探测的能力混合不相同”；另报告单一游戏的*RL* 训练对未见游戏/外部规划任务存在迁移**hu2505lmgame**。与本文关系：作为“脚手架/污染控制/迁移观察”的评测文献。

…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）。

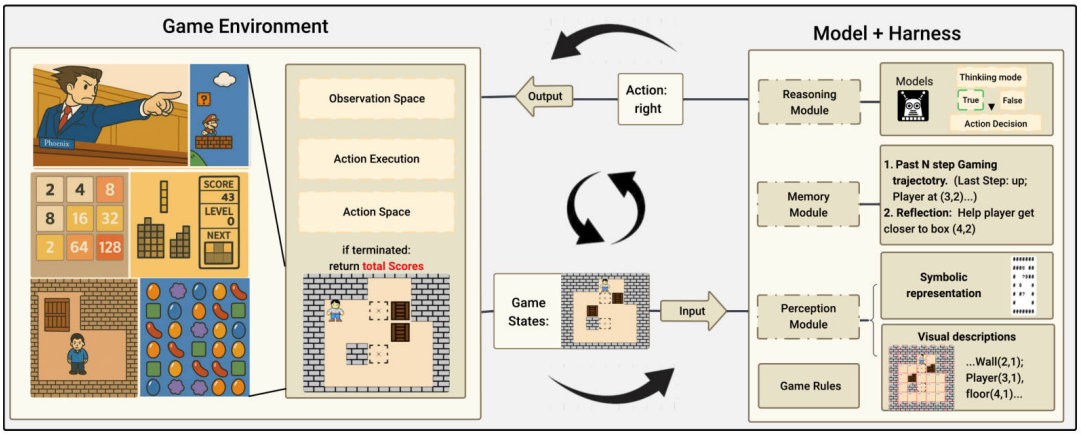

Figure 4: lmgame-Bench uses modular harnesses—such as perception, memory, and reasoning modules—to systematically extend a model’s game-playing capabilities, enabling iterative interaction loops with a simulated game environment **hu2505lmgame**.

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，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；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**。与本文关系：作为视觉中心评测的代表性基准。

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。与本文关系：术语与度量背景。

文献中常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，经内部路由后由GUI 执行；并以*Invalid Action Rate* 作为守门指标。对稀疏/时序敏感技能，亦有工作报告机会归一

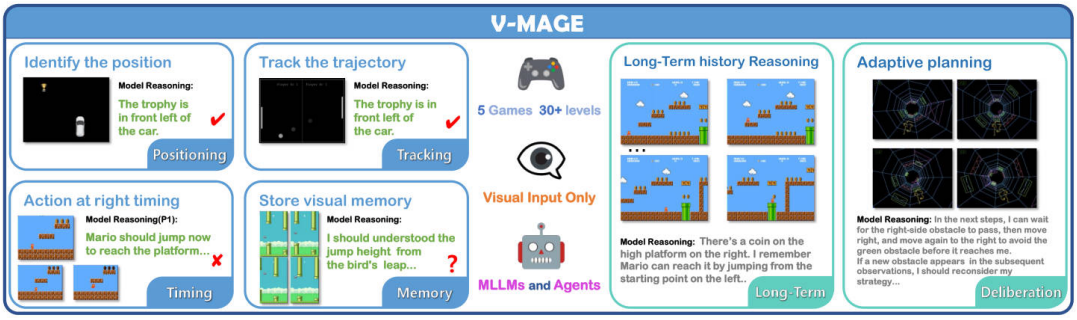

Figure 5: The overview of the V-MAGE benchmark, designed to evaluate vision-centric capabilities and higher-level reasoning of MLLMs across 5 free-form games with 30+ levels **v-mage**.

化成功率（*OAS*）、反应时延（*RT*）与每次机会尝试数（*APO*）等指标[8]。此外，关于*post-processing*（技能解码、重试/回滚）的显式记录亦见于将游戏转化为可靠评测的文献**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**。

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。与本文关系：工程背景。

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。与本文关系：安全与鲁棒性背景。

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

2.9 Synthesis: Trends, Gaps & Our Niche

当前趋势是在*GUI*（*GCC*）通道上引入协议化/模块化编排以支撑可复现实验与消融；真实多类型游戏的统一评测（如*Orak*）与*visual-only/continuous-space* 的视觉中心评测（如*V-MAGE*）并行发展；把脚手架/污染控制引入评测协议（如*lmgame-Bench*）成为常见做法。与本文关系：本文研究定位于伴随式（*companion-style*）场景的文献回顾与术语/评测背景梳理。

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]

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.