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

The demand for **companion-style** AI assistants that operate in real-time gaming environments is expanding rapidly. This trend signals a shift in player expectations, moving beyond simple automation scripts or static information overlays toward interactive, persistent, and engaging AI partners.

The rise of AI-driven virtual streamers, particularly the **Neuro-sama** phenomenon, highlights a significant shift in both technology and community-driven commercialization. Neuro-sama, an AI-powered virtual streamer (VTuber), engages in real-time conversations and dynamic gameplay, capturing the attention of audiences [1]. Although Neuro-sama remains closed-source, its success has inspired a vibrant open-source ecosystem, with developers aiming to replicate or expand upon its capabilities [2]–[4]. This technological innovation has been accompanied by strong commercial traction on platforms like Twitch, indicating a growing market demand for AI that offers both utility and companionship [5], [6]. Through persistent, low-latency **voice loops**, these AI systems create engaging player experiences, driving both user engagement and monetization.

This project is timely. While the market demand is clear, the technological feasibility for a **reproducible, non-API-dependent** assistant has only recently emerged. Foundational research is now converging on the key methodologies required for productization. This includes the development of **unified evaluation protocols** and **modular ablation frameworks**, which are essential for ensuring that experiments are reproducible and comparable [7], [8]. Furthermore, recent work has demonstrated the viability of **General Computer Control (GCC)** pathways (i.e., screen-in, keyboard/mouse-out), confirming that capable agents can operate without relying on game-specific APIs [9], [10].

extbfInspired by this clear convergence of market demand and emerging academic feasibility, this project aims to bridge the gap. The core objective is to define and build a prototype for a **companion-style assistant**. This assistant is envisioned as a persistent, in-game partner that delivers **event spotting** and **tactical guidance**, leveraging these new, reproducible methodologies to enhance the player's experience without interrupting gameplay.

1.2 Project Scope & Feasibility

To address the issue of inconsistent action interfaces and the challenge of reproducibility caused by reliance on scene-specific APIs, as mentioned in previous sections, this project makes the following clear definition of its technical scope:

- **Unified action interface via GUI (GCC)**: This refers to a human-homomorphic channel with screen-in, keyboard/mouse-out.

- **Structured output as the default path**: The default action generation and execution path is defined as **structured output + constrained decoding**.

This scope is defined based on its demonstrated feasibility. Existing studies have shown that, without relying on application-specific APIs, long-horizon tasks can be completed through a pipeline consisting of "planning, macro/skill organization, self-reflection, and memory". Additionally, on real platforms, the "screenshot to structured action" end-to-end navigation path has been proven to be reproducible, and **legal move constraints** significantly reduce invalid actions [9]–[11].

1.3 Design Principles & System Preview

Design principles.

This work follows four principles: **Unified Input** (ensures portability across applications), **Structured Output** (reduces invalid actions and is easy to audit), **Protocol Consistency** (ensures reproducibility and ablative evaluation), **Low-Coupling Orchestration** (MCP-style, facilitates modular insertion/removal of skills/tools).

System preview.

The system flow is as follows: **screen/audio** capture, lightweight **VLM** perception, **agentic** (planning/memory/reflection) modules, **MCP-style** skill/tool routing (including OCR/retrieval/computation **tool use**), **GUI execution** (keyboard and mouse), and finally passing through the **safety** module (confirmation/rollback/emergency stop). To reduce end-to-end latency, the deployment strategy will combine the **tool-augmented MLLM** approach with **on-device inference** quantization/caching strategies as key engineering tactics [12], [13].

1.4 Key Challenges

In real player scenarios, the challenge is not just the individual model score, but the overall experience that is **stable, accurate, fast, and controllable**. Below, the challenges are listed according to **measurable** factors.

Long-Horizon Stability (GCC). When using only GUI (GCC), errors and misalignments can accumulate along the interaction chain, leading to the problem of “drifting off track.” This can be measured with **pass@k**, rollback rate, and APO (attempts per opportunity). The combination of “**planning + skills (macro) + reflection + memory**” can alleviate drift, but it is not a panacea [9].

Vision-Centric Grounding & Memory. In purely visual or continuous space setups, **localization/tracking/counting, timing control, and long-term visual memory** are current model limitations. These can be statistically measured by **OAS** (opportunity-normalized success), broken down by opportunity types such as “pickable items/time points/path nodes” [14].

Invalid Actions & Think-Action Mismatch. Converting free text into actions often results in

"thinking correctly but clicking the wrong place." **Structured output + legal move constraints** can reduce **Invalid%**, and **Brier/MAE** can be used to evaluate calibration. On the training side, rewards based on four granularities—"format/type/coordinates/content"—align execution details [10], [11]. At the implementation level, output strategies such as "**semantic, →, allowed actions**" mapping or **probability modeling over the set of allowed actions** can be used to suppress out-of-bounds and misaligned actions from the start [15].

OOD & Protocol Consistency. When the environment or version changes, comparison results become difficult. **Procedural generation** is needed for out-of-distribution (OOD) evaluation, with fixed **post-processing** and prompt scaffolding to control variables, and reporting **macro/micro** metrics under a **unified protocol** and **leaderboard/battle arena** [7], [8], [11].

Latency & User Experience (UX). An assistant must "respond instantly." The key metrics are **RT** (reaction time per opportunity) and **voice RTT** (round-trip voice latency). Engineering strategies to reduce latency include **quantization/pruning/KV caching/streaming decoding** and **on-device/edge-cloud collaboration**, along with single-flight and barge-in strategies [13].

Safety & Robustness. High-risk actions must be "confirmable, rollbackable, and traceable." The approach includes **permission whitelists**, **double confirmation**, **shadow execution**, and **rollback/emergency stop**, with **logging/auditing** to locate **think-action mismatch** [16]–[18].

1.5 Project Objectives & Expected Deliverables

Objectives.

(i) Develop a **GCC-based assistant** real-time prototype that covers event prompting, strategy suggestions, and voice loops.

(ii) Organize **skills/macros, planning, memory, reflection** using **MCP-style orchestration**, enabling plug-and-play and auditability without relying on application-specific APIs.

(iii) Define a set of **small and reproducible** evaluation elements (task scripts and metric families), focusing on **pass@k/TTC/Invalid%/macro-micro** and **OAS/RT/APO** experience-related metrics [7], [8].

Expected Deliverables.

(a) **System Prototype:** Screen capture and lightweight perception, agentic modules, MCP-style skill bus, GUI executor, and basic safety safeguards.

(b) **Evaluation Scripts and Configuration:** Reproducible task scripts, metric calculation, and logging/audit tools (including module switches for comparison).

(c) **User Documentation and Demos:** Installation/operation instructions, configuration templates, and demo videos.

1.6 Assumptions & Out-of-Scope

Working assumptions.

The default engineering posture is **single-flight + event-triggered + frame-window (3–5 frames) + text-first** to reduce latency and variance. Evaluation records **post-processing** and environment versions will be maintained to ensure protocol consistency.

Out-of-scope.

This project does not involve **adapting application/game-specific APIs** on a per-application basis. It does not commit to **large-scale end-to-end training and data collection** in this report. It will not rely on platform-level enhancement privileges (such as A11y/private DOM hooks). **VLA direct-action** will be used only as a benchmark comparison and not as the default path [11].

1.7 Glossary & Terminology

TODO: Add more . . .

Since this research area is relatively new, the terminology and naming conventions across different works are not yet unified. Therefore, before entering the literature review, this work aligns key terms and definitions:

GCC (General Computer Control): A human-homomorphic action interface defined as screen-in + keyboard/mouse-out; this is the default execution channel in this work [16].

LAM (Large Action Models): A family of models where structured actions are treated as first-class outputs; referenced as a comparative paradigm in this work [16].

VLM vs VLA: Text/JSON output mapped into action space vs direct action vectors/distributions; evaluation will consistently use legal move mapping + constrained decoding approach [17].

Scaffold vs Orchestration (MCP-style): The former refers to the stable interaction "scaffolding" during evaluation, while the latter refers to module/tool registration and routing; both are complementary [18].

Metric Definitions: **pass@k**, **TTC**, **Invalid%**, and **macro/micro** will be reported together; opportunity-driven **OAS/RT/APO** serve as core supplements in companion-style scenarios [19].

Memory–Reasoning–I/O (M-R-I/O): The internal working division and terminology anchor of this work. Here, planning and reflection align with reasoning, skills/macros represent action output forms on the I/O side, and memory remains independent. The output side will default to a "semantic-to-allowed action" mapping or a compliance strategy that models probabilities over the set of allowed actions [15].

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

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

[5] "Vedal's ai vtuber neuro-sama sets new twitch hype train world record." Accessed: 2025-10-11, Streams Charts. [Online]. Available: https://streamscharts.com/news/vedals-ai-vtuber-neuro-twitch-hype-train-record.

[6] "Vedal987 — streamer overview & stats." Accessed: 2025-10-11, TwitchTracker. [Online]. Available: https://twitchtracker.com/vedal987.

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

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

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

[11] 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.

[12] W. An, J. Nie, Y. Wu, F. Tian, S. Lu, and Q. Zheng, "Empowering multimodal llms with external tools: A comprehensive survey," *arXiv preprint arXiv:2508.10955*, 2025.

[13] J. Xu *et al.*, "On-device language models: A comprehensive review," *arXiv preprint arXiv:2409.00088*, 2024.

[14] X. Zheng *et al.*, "V-mage: A game evaluation framework for assessing vision-centric capabilities in multimodal large language models," *arXiv preprint arXiv:2504.06148*, 2025.

[15] S. Hu *et al.*, "A survey on large language model-based game agents," *arXiv preprint arXiv:2404.02039*, 2024.

[16] C. Zhang *et al.*, "Large language model-brained gui agents: A survey," *arXiv preprint arXiv:2411.18279*, 2024.

[17] F. Tang *et al.*, "A survey on (m) llm-based gui agents," *arXiv preprint arXiv:2504.13865*, 2025.

[18] X. Hu *et al.*, *Os agents: A survey on mllm-based agents for computer, phone and browser use*, 2024.

[19] Z. Durante *et al.*, "Agent ai: Surveying the horizons of multimodal interaction," *arXiv preprint arXiv:2401.03568*, 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.