FlashAdventure: A Benchmark for GUI Agents Solving Full Story Arcs in Diverse Adventure Games

Jaewoo Ahn[*,†,1,2] **Junseo Kim**[*,1] **Heeseung Yun**[1] **Jaehyeon Son**[†,2,3]
Dongmin Park[2] **Jaewoong Cho**[2] **Gunhee Kim**[1]
[1]Seoul National University [2]KRAFTON [3]Georgia Institute of Technology
{jaewoo.ahn, junseo.kim, heeseung.yun}@vision.snu.ac.kr
jaehyeon.son@gatech.edu, {dongmin.park, jwcho}@krafton.com, gunhee@snu.ac.kr
https://ahnjaewoo.github.io/flashadventure

Abstract

GUI agents powered by LLMs show promise in interacting with diverse digital environments. Among these, *video games* offer a valuable testbed due to their varied interfaces, with *adventure games* posing additional challenges through complex, narrative-driven interactions. Existing game benchmarks, however, lack diversity and rarely evaluate agents on completing entire storylines. To address this, we introduce FlashAdventure, a benchmark of 34 Flash-based adventure games designed to test full story arc completion and tackle the *observation-behavior gap*: the challenge of remembering and acting on earlier gameplay information. We also propose CUA-as-a-Judge, an automated gameplay evaluator, and COAST, an agentic framework leveraging *long-term clue memory* to better plan and solve sequential tasks. Experiments show current GUI agents struggle with full story arcs, while COAST improves milestone completion by bridging the observation-behavior gap. Nonetheless, a marked discrepancy between humans and best-performing agents warrants continued research efforts to narrow this divide.

1 Introduction

Recent advances in Large Language Models (LLMs) and multimodal LLMs have enabled the emergence of graphical user interface (GUI) agents, or Computer-Using Agents (CUA) - autonomous systems that perform diverse digital tasks by controlling mouse and keyboard inputs to interact with visual elements on platforms such as web, mobile, and OS (Nguyen et al., 2024; Tang et al., 2025). This approach holds great promise, as GUIs are ubiquitous in all computing devices humans use in work and daily life.

Figure 1: FlashAdventure consists of 34 Flash-based classic adventure games and supports automatic evaluation of the GUI agent using CUA-as-a-Judge.

Among various applications of GUI agents, *video gaming* agents play a crucial role. Video games offer a balanced environment between device user interfaces and real-world perception. Unlike standard GUIs such as web pages or widgets, game environments feature non-standard, less obvious layouts and complex interaction modes, making them an excellent testbed for evaluating the generalizability of GUI agents (Hu et al., 2025b). In particular, classic *adventure games*[2], characterized by narrative-driven exploration, introduce additional complexity through diverse visual layouts and interactions, such as managing inventories, engaging in multi-turn dialogues with branching outcomes, and performing context-sensitive actions. Moreover, understanding the story arc requires conceptual reasoning and lateral thinking (De Bono and Zimbalist, 1970), as players should interweave collected items and narrated problems with creativity.

Recent studies of GUI agents in video games have explored several adventure game characteris-

[*]Equal contribution.
†Work done during an internship at KRAFTON.

[2]https://en.wikipedia.org/wiki/Adventure_game.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 23354–23384
November 4-9, 2025

tics, such as performing low-level tasks (*e.g.*, renaming a card deck) (Hu et al., 2024b), executing sequences of actions (*e.g.*, combatting a boss monster) (Chen et al., 2024), or completing specific missions (*e.g.*, searching barn) (Tan et al., 2025). While such capabilities allow for meaningful gameplay, a critical gap remains in both (1) the diversity of tasks and (2) the completion of full story arc. Although these agents excel in specific game tasks, it is unclear whether they can generalize to diverse scenarios. Moreover, due to the extremely long story arcs in AAA games or the lack of clear narrative conclusions often found in MMORPG-ish games, the ability of GUI agents to complete entire story arcs has yet to be verified.

To address these limitations, we introduce **FlashAdventure**, a new benchmark based on 34 adventure games to evaluate GUI agents solving full story arcs from the start to finish. To assess agents in games with self-contained, compact story arcs, we use *Flash* games[3] (See Figure 1), which offer compact playtime (approximately one hour per game, compared to over 30 hours for AAA games like *Red Dead Redemption 2*), and are free to play. A key challenge in this context is the "observation-behavior gap", which refers to the *time lag* between when an agent *observes information* and when it can *act upon it*. As shown in Figure 2, adventure games require agents to manage long-term time lags, such as interrogating a suspect and later discovering their innocence. These long-term dependencies are crucial when solving full story arcs. Tolman's theory on latent learning (Tolman and Honzik, 1930; Tolman, 1948) suggests that humans can retrieve and apply clues after a long delay, which can also be explored in agents to assess whether similar emergent behaviors occur.

Based on our benchmark, we propose two technical components: **CUA-as-a-Judge** and **COAST**. While prior studies of GUI agents in video games relied on manual assessment, we address this limitation by introducing CUA-as-a-Judge, a new *judge agent*. This agent acts as an oracle with access to predefined success milestones for each game and actively interacts with the game environment to verify whether these milestones have been achieved. Additionally, to address observation-behavior gap issues in FlashAdventure, we propose a new agentic framework, COAST (Clue-Oriented Agent for Sequential Tasks), which leverages long-term *clue memory* by proactively maintaining it and applying it during the problem-solving stage.

[3]Flash games refer to browser-based games developed using the Adobe Flash platform.

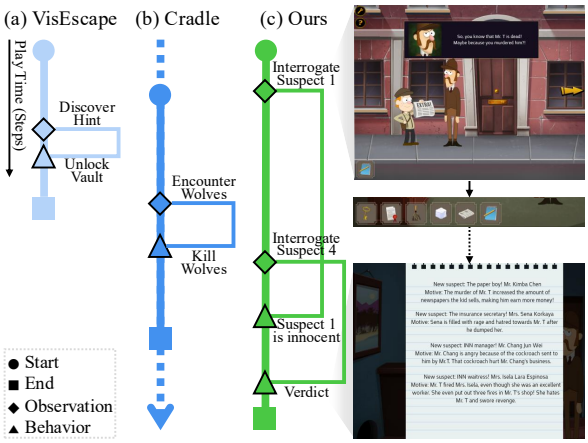

Figure 2: Comparison of gameplay progression across (a) VisEscape (Lim et al., 2025), (b) Cradle (Tan et al., 2025), and (c) FlashAdventure. Prior benchmarks focus on short-term objectives or include short story arcs, limiting their ability to fully evaluate agents' capacity to manage the long-term observation-behavior gap. In contrast, FlashAdventure emphasizes completion of full story arcs involving long-term objectives, exemplified by suspect interrogations leading to a verdict.

Using FlashAdventure, we evaluate multiple GUI agents and empirically show that they greatly struggle to complete full story arcs due to weak planning capabilities, limited visual perception in non-standard layouts, and the lack of lateral thinking. In contrast, COAST enhances the planning by effectively managing clue memory, thereby bridging the observation-behavior gap and improving problem-solving capabilities. However, all GUI agents still show a significant gap compared to human performance, demonstrating the need for further improvement in associating time-dependent observations with the full story arcs.

2 Related Work

GUI Agents. Multimodal LLMs enabled GUI agents to perform tasks on various digital platforms (Nguyen et al., 2024). GUI agents require explicit action planning and execution, necessitating *GUI grounding* problem (Cheng et al., 2024; Wu et al., 2025) to map language instructions to specific GUI elements and executions. Another key problem is *input modality selection*: whether GUI agent should rely on platform-specific structured data (*e.g.*, , DOM trees for web, accessibility trees for OS, view hierarchies for mobile) (Deng et al.,

Benchmark / Framework	# Games	Environment	Free?	Automatic Evaluation	Complete Story Arc	Featured Games
Code/API-based						
PokeLLMon (2024a)	1	API	✓	✓	✗	Pokémon
TextStarCraft II (2024)	1	API	✓	✓	✗	StarCraft II
MineDojo (2022)	1	API & Screen	✓	✓	✗	Minecraft
BALROG (2025)	6	API & Screen	✓	✓	✗	MiniHack, NLE, Baba Is AI, Crafter, BabyAI, TextWorld
LVLM-Playground (2025a)	6	API & Screen	✓	✓	✗	TicTacToe, Reversi, Sudoku, Minesweeper, Gomoku, Chess
V-MAGE (2025)	5	API & Screen	✓	✓	✗	FlappyBird, RaceGame, Super Mario, PongGame, Tempest Run
VisEscape (2025)	-*	API & Screen	✓	✓	✓	*Room escape game created for research, instead of adapting existing ones
Orak (2025)	12	API & Screen	✗	✓	✗	2 Games × 6 Genres (Action, Adventure, RPG, Simulation, Strategy, Puzzle)
Pixel/Screenshot-based						
OOTB (2024b)	2	Screen Only	✓	✗	✗	Hearthstone, Honkai: Star Rail
VARP (2024)	1	Screen Only	✗	✗	✗	Black Myth: Wukong (AAA game)
Cradle (2025)	4	Screen Only	✗	✗	✗	RDR2 (AAA game), Stardew Valley, Cities: Skylines, Dealer's Life 2
FlashAdventure	**34**	Screen Only	✓	✓	✓	Classic Adventure Games (Mystery/Detective, Hidden Object, Room Escape, Visual Novel, Life/Management Simulation)

Table 1: Overview of video game benchmarks. *Complete Story Arc* indicates whether the benchmark evaluates an agent's ability to complete a self-contained story arc from beginning to end. Our FlashAdventure evaluates agents on completing full story arcs in diverse adventure games.

2023; Zhou et al., 2024; Li et al., 2020), vision-only input (screenshots) (Gou et al., 2025; Hong et al., 2024; Rawles et al., 2023), or a hybrid combination of both (He et al., 2024; Koh et al., 2024; Furuta et al., 2024; Xie et al., 2024). Although numerous benchmarks have been proposed to evaluate GUI agents, they focus on standardized templates, restricting the range of interaction behaviors and visual structures that agents encounter (Fan et al., 2024; Sun et al., 2025).

Video Game Benchmarks. Early gameplaying agents relied on reinforcement learning (RL) for solving relatively simple, rule-based environments (Bellemare et al., 2013; Brockman et al., 2016; Silver et al., 2016). Recent attention has shifted toward tackling complex video games via LLM-based agents (Wu et al., 2024; Team et al., 2024; Kanervisto et al., 2025; Yuan et al., 2025; Zhang et al., 2025). To evaluate game-solving capabilities, several game benchmarks have emerged, broadly categorized into two types.

(1) *Code/API*-based benchmarks provide structured representations of game states or internal reward signals through direct API interactions (*e.g.*, , Pokémon (Hu et al., 2024a), StarCraft II (Ma et al., 2024)). Certain variants augment these interactions with visual inputs, such as screenshots (*e.g.*, , Minecraft (Fan et al., 2022), video game collections (Wang et al., 2025a; Paglieri et al., 2025; Hu et al., 2025a)), yet the fundamental interaction paradigm remains reliant on APIs rather than GUI manipulations (see Table 1). While such benchmarks guarantee precise and immediate feedback, they are inherently limited to the games that expose accessible APIs, necessitate substantial preliminary setup efforts (*e.g.*, modding[4] or reverse engineering) to integrate new games.

(2) *Pixel/screenshot*-based benchmarks adopt a more realistic vision-action interface, where agents perceive raw visual input (*i.e.*, screenshot) and interact via actions such as clicks or keystrokes. While this paradigm holds promise for general-purpose GUI agent evaluation, existing studies (Chen et al., 2024; Tan et al., 2025; Chen et al., 2025) often rely on *buy-to-play* AAA games, which limit accessibility and reproducibility, and require manual evaluation, making broader assessments difficult.

Overall, existing game benchmarks lack task and game diversity, featuring at most 12 games (see Table 1), making it difficult to evaluate agents across a wide range of gaming scenarios beyond specific tasks or a limited set of games. Furthermore, except for room escape games, none of these benchmarks test agents on complete story arcs, either because AAA games have extensive length (Chen et al., 2024) or because some games lack clear endings (Hu et al., 2024b; Tan et al., 2025). Although room escape games evaluate agents on complete story arcs, they fall short in assessing the extensive observation-behavior gap due to their substantially low average human playtimes (*e.g.*, 5.7 steps for Wang et al. (2025b), 52.8 for Lim et al. (2025), and 257.8 for Qian et al. (2025)), as illustrated in Figure 2. This limitation stems from the fact that the games were designed specifically for research rather than adapted from existing games for entertainment purposes.

[4] https://en.wikipedia.org/wiki/Modding

3 The FlashAdventure Benchmark

We introduce **FlashAdventure**, a new benchmark comprising 34 diverse Flash-based classic adventure games that evaluates agents' ability to solve full story arcs.

3.1 Problem Formulation

We cast FlashAdventure gameplay as a partially-observable Markov decision process (POMDP) $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \Omega, \mathcal{T}, \mathcal{R} \rangle$. The hidden state space $\mathcal{S}$ denotes the internal configuration of the game runtime (*e.g.*, memory, object stacks), while the observation space $\mathcal{O}$ is the RGB frame buffer rendered each step. A deterministic rendering function $\Omega : \mathcal{S} \to \mathcal{O}$ returns the visible frame $o_t = \Omega(s_t)$ presented to the agent. The action set $\mathcal{A}$ comprises low-level GUI inputs (*e.g.*, `left-click(x,y)`; see Table 5 in Appendix A.1 for the complete set) that the agent can issue at every step. Upon executing $a_t \in \mathcal{A}$, the game engine advances according to the transition kernel $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, yielding the next hidden state s_{t+1} and its observation o_{t+1}. A GUI agent policy π_θ conditions on a task query q (*e.g.*, "escape the room within 1,000 steps"), and the complete interaction history to output the next action a_t, thereby generating the trajectory $\tau = (o_1, a_1, \ldots, o_T, a_T)$. Finally, a reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \{0, 1\}$ produces a binary success indicator at the end of an episode, which is directly observable to the agent: $\mathcal{R}(s_T, a_T) = 1$ if the agent completes the task successfully, and 0 otherwise.

Separately, within the environment, we define a milestone reward function $\mathcal{R}_{\mathrm{m}} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ that assigns intermediate rewards based on progression milestones. Unlike $\mathcal{R}$, the values of $\mathcal{R}_{\mathrm{m}}$ are not directly observable by the agent.

3.2 Game Selection

We utilize the *FlashPoint Archive*[5] that enables secure playback of Flash-based browser games. We specifically focus on *classic adventure games* characterized by narrative-driven exploration. From the extensive library of Flash-based adventure games, we selected 34 games based on the following criteria:

1. *Free-to-play* games covering diverse subgenres (*e.g.*, mystery/detective, hidden object, room escape, visual novel, simulation).
2. Games that emphasize reasoning over reaction speed by avoiding strict time limits and rapid-action requirements.
3. Games with validated human walkthroughs and clearly defined progression milestones, excluding fully open-ended narratives. All games are solvable within at most 1-2 hours by human players, ensuring agents can complete the full story arcs within practical evaluation timeframes.

The resulting 34 game collection makes FlashAdventure the largest among existing video game benchmarks (see Table 1). Appendix A.2 lists all games and includes a representative human walkthrough for each subgenre. Appendix A.3 further analyzes inter- and intra-subgenre diversity, demonstrating that FlashAdventure provides substantial diversity across games.

3.3 Human GamePlay

To validate that the selected games are neither too easy nor too difficult to complete full story arcs within feasible timeframes, we collected a set of human gameplay demonstrations. Specifically, we recruited 13 participants to play through all 34 games. Detailed procedures for collecting human gameplay demonstrations are described in Appendix A.5.

As summarized in Table 13 (in Appendix B.2), participants completed full story arcs with an average of 1,142 steps, an average playtime of 26 minutes, and a success rate of 97.1%, demonstrating that the games can be fully solved within practical time limits. Compared to prior room escape benchmarks, which require 5.7-257.8 steps to complete the story arcs, our selected games require substantially longer playtime (1,142 steps), reflecting greater complexity.

Moreover, we observed several instances of the long-term observation-behavior gap during gameplay. As shown in Figure 10 (Appendix A.5.1), players exhibit substantial step gaps of 99 and 421 between acquiring items and receiving rewards, demonstrating their ability to manage extended dependencies. Appendix A.5.1 reports statistics of the observation-behavior gap, with an average of 251.1 steps, indicating its substantial magnitude.

3.4 Evaluation

Defining milestones and scores. To systematically evaluate gameplay performance in FlashAdventure,

[5] https://flashpointarchive.org/

it is essential to clearly define milestones and subtasks reflecting agents' progression towards the narrative end-goal. Following prior video game benchmarks using milestones to measure progress (Chen et al., 2024; Tan et al., 2025; Lim et al., 2025), we define a set of milestones for each game. To ensure the validity of the defined milestones, three of the authors independently review the full walkthroughs of each game and finalize the milestone list by filtering out those without consensus. Specifically, for the 27 non-simulation-based games, we define a *discrete* number of milestones (average: 6.7, maximum: 12, minimum: 4). For the 7 simulation-based games, we dispense with discrete milestones and instead use the continuous score shown in the game's head-up display (HUD). This score, which can be revealed by clicking the score icon when necessary, is normalized by the maximum attainable value. We provide the full list of milestones along with a brief introduction to the game in Appendix A.4.

Evaluation metrics. We adopt standard evaluation metrics commonly employed in benchmarks for games and embodied agents (Chen et al., 2024; Tan et al., 2025; Lim et al., 2025):

- **Success Rate**: A binary metric $\mathcal{R}(s_T, a_T)$ indicating whether the final narrative goal (*e.g.*, mystery solving, room escape) is successfully achieved (1) or not (0).
- **Milestone Completion Rate**: The proportion of milestones completed $\mathcal{R}_{\text{m}}(s_T, a_T)$ at the point when an agent reaches the maximum allowed steps (*e.g.*, , 1,000). For simulation-based games, this metric is represented by normalizing the continuous score achieved at the maximum step count against the maximum attainable score.
- **Steps**: The total number of actions taken by the agent (t), capped by a predefined maximum number.

3.5 CUA-as-a-Judge

Existing pixel/screenshot-based video game benchmarks for evaluating progression milestones of GUI game agents lack automatic evaluation methods and have relied on manual assessments by human annotators (Chen et al., 2024; Tan et al., 2025).

To overcome this limitation, we introduce a novel automatic evaluation framework, *CUA-as-a-Judge*, implemented using Claude-3.7-Sonnet computer-use (Anthropic, 2024). This *judge* agent functions as an oracle with direct access to success milestones for each game, simulating a human judging process (*e.g.*, manually checking puzzle completion or character progress). After a game agent π_θ finishes gameplay (either by successfully achieving the final goal or reaching the maximum allowed number of steps), the CUA-as-a-Judge resumes from the game's final state s_T. The judge interacts directly with the game environment by executing actions a_{T+j} (*e.g.*, `mouse-click`(x, y), `key-type`(key)) and observing the resulting observations o_{T+j}, thereby verifying the accomplishment of necessary milestones. For instance, in Figure 11 (right) in Appendix A.6, CUA-as-a-Judge can click on the notebook to reveal the number of discovered suspects, directly confirming milestone progression.

Specifically, to determine the degree of completion, the judge sequentially checks whether each of the N_{m} predefined milestones has been achieved, starting from the first. If a milestone is not satisfied, the evaluation halts, and the completion score is computed as $K_{\text{m}}/N_{\text{m}}$, where K_{m} is the number of milestones successfully verified. This process continues until failure or full completion of all milestones. Note that in other cases, milestone verification can instead be performed in a single pass by counting cumulative game states such as collected items or unlocked locations. Further implementation details on evaluation strategy are provided in Appendix A.6.

Evaluation agreement vs. human judge. We evaluate the reliability of CUA-as-a-Judge by comparing its judgments with human judgments across all 34 games. For each game, we sampled 8-9 episodes, resulting in a total of 300 samples. Our comparison shows a high agreement, with an accuracy of 94.00%, Spearman correlation of 0.9912, and Pearson correlation of 0.9999. The judge performed particularly well on step-by-step milestone verification, while showing some limitations with counting-based milestones that require assessing multiple items at once. See Appendix A.6 for validation details.

4 Approach

As shown in §3.3, it is essential to address long-term observation-behavior gap in FlashAdventure. To this end, we propose the COAST (*Clue-Oriented Agent for Sequential Tasks*) framework, which

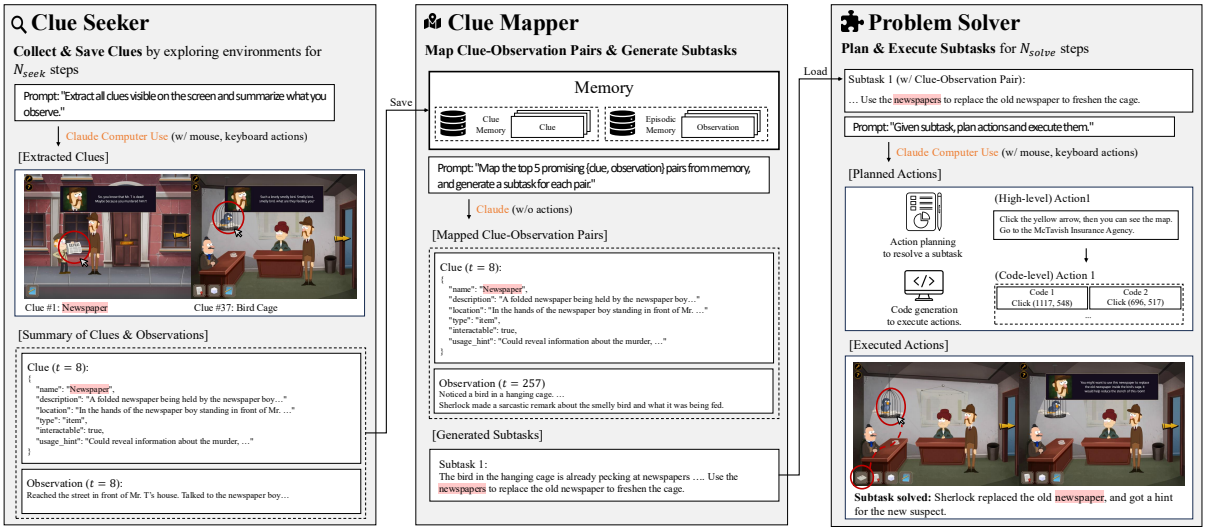

Figure 3: Overview of COAST Framework with Seek-Map-Solve Cycle.

leverages long-term *clue memory* that stores relevant clues for future use. Then, COAST guides agents to actively seek clues and generate diverse subtasks with plausible hypotheses for solving complex tasks by following a *Seek-Map-Solve* cycle (see Figure 3 and Algorithm 1):

Step 1: Clue Seeking. The **Clue Seeker** module (π_θ^{seek}) explores the environment for N_{seek} steps to collect potential clues (*e.g.*, "A folded newspaper held by the boy."), rather than solving tasks directly. All gathered information is stored in a shared clue memory $\mathcal{M}$.

Step 2: Clue-Observation Mapping. The **Clue Mapper** (π_ϕ^{map}) analyzes the accumulated memory and trajectory to identify at most K promising pairs of *clues* and corresponding *observations* (o_t) at specific time steps. Based on these pairs, it generates subtasks (*e.g.*, "Use the newspaper to replace the old newspaper to freshen the bird cage.") that represent plausible hypotheses, thus forming the goal candidate set $\mathcal{G} = \{g_1, g_2, \ldots, g_K\}$, where each g_k represents a goal defined by $(\mathcal{M}_k, o_{t_k}, \text{subtask}_k)$.

Step 3: Problem Solving. The **Problem Solver** (π_ψ^{solve}) executes the proposed subtasks for N_{solve}. Completed goals are stored in a resolved set $\mathcal{G}_R$ to avoid duplication.

The agent cycles through clue seeking, subtask generation, and problem solving until the step limit T is reached, continually updating memory and filtering resolved goals. Note that the clue memory $\mathcal{M}$ has no strict size limit. It does not cause memory issues storing all clues during gameplay, since each clue is small (*e.g.*, a few dozen tokens). On average, 150 clues $\times$ 85 tokens per clue yields about

Algorithm 1: COAST Framework with Seek-Map-Solve Cycle.

Input: task query q, maximum step T
Data: $\mathcal{M}$ (clue memory), τ (trajectory), $\mathcal{G}_R$ (resolved-goal set)

$\mathcal{M} \leftarrow \varnothing, \tau \leftarrow \varnothing, \mathcal{G}_R \leftarrow \varnothing, t \leftarrow 0$
while $t < T$ **do**
  `// 1. Clue Seeker`
  for $i = 1$ **to** N_{seek} **do**
    Observe frame o_t
    $a_t \leftarrow \pi_\theta^{\text{seek}}(o_t, \mathcal{M}, q)$
    $\Delta\mathcal{M} \leftarrow$ Execute action a_t
    $\mathcal{M} \leftarrow \mathcal{M} \cup \Delta\mathcal{M}$
    $\tau \leftarrow \tau \oplus (o_t, a_t)$ `// append` (o_t, a_t) `to` τ
    $t \leftarrow t + 1$
    if $t \geq T$ **then**
      break
  if $t \geq T$ **then**
    break
  `// 2. Clue Mapper`
  $\mathcal{G} \leftarrow \pi_\phi^{\text{map}}(\mathcal{M}, \tau, q)$
  $\mathcal{G} \leftarrow \mathcal{G} \setminus \mathcal{G}_R$ `// filter out resolved goals`
  if $\mathcal{G} = \varnothing$ **then**
    continue `// restart Seek block`
  else
    `// 3. Problem Solver`
    $n_{\text{solve}} \leftarrow 0$
    foreach $g \in \mathcal{G}$ **do**
      $a_t \leftarrow \pi_\psi^{\text{solve}}(o_t, g, q)$
      `success` $\leftarrow$ Execute action a_t
      $\tau \leftarrow \tau \oplus (o_t, a_t)$
      if `success` **then**
        $\mathcal{G}_R \leftarrow \mathcal{G}_R \cup \{g\}$
      $t \leftarrow t + 1, n_{\text{solve}} \leftarrow n_{\text{solve}} + 1$
      if $t \geq T$ *or* $n_{solve} \geq N_{solve}$ **then**
        break

12.8K tokens, which is well below the maximum context length of modern LLMs (*e.g.*, Claude-3.7-Sonnet supports 200K tokens).

5 Experiments

5.1 Baseline Agents

We evaluate five representative GUI agents, grouped by the degree of modularization between the backbone VLM, the GUI Grounding/action-execution layer, and the agentic framework.

End-to-end agents. (1) Claude-3.7-Sonnet Computer-Use (Anthropic, 2024) and (2) OpenAI CUA (OpenAI, 2025b) are proprietary agents that integrate perception, planning, and action execution within a unified architecture. (3) UI-TARS-1.5-7B (Qin et al., 2025) is an open-source alternative, built upon Qwen-2-VL 7B (Wang et al., 2024).

Modular agents. In contrast, (4) Cradle (Tan et al., 2025) and (5) Agent S2 (Agashe et al., 2025b) exemplify modular agents that explicitly separate each module. We adopt GPT-4o (OpenAI et al., 2024) and Claude-3.7-Sonnet (Anthropic, 2025) as backbone VLMs. For GUI Grounding/action-execution, we utilize either UGround-V1-7B (Gou et al., 2025) or Claude-3.7-Sonnet combined with `pyautogui` (*i.e.*, Python scripts that control the mouse and keyboard.). We exclude GPT-4o as it frequently failed to localize UI elements.

The agentic framework for Cradle comprises six modules: *Information Gathering* process multimodal input, *Self-Reflection* reconsiders past trajectories, *Task Inference* selects the subsequent task, *Skill Curation* generates relevant skills for a given task, *Action Planning* determines executable actions, and *Memory* stores and retrieves past experiences and skills. Agent S2 extends this modular design by further decoupling grounding and planning, routing each sub-goal to a *Mixture-of-Grounding* ensemble and employing *Proactive Hierarchical Planning* that refines action sketches as new observations arrive.

5.2 Experimental Settings

We set the maximum number of action steps per agent to 1,000. For the main experiments, we conducted evaluations on all 34 games using seven agents. In addition, we report detailed results for each game in Table 13 and Table 14 (Appendix B.2). Please refer to Appendix B.1 for further details.

5.3 Experimental Results

As shown in Table 2 and Table 13 in Appendix B.2, all agents exhibit near-zero success rates and significantly lower milestone completion rates compared to humans. Even the highest-performing agents, measured by success rate, only completed two hidden-object games (*i.e.*, the Grim Tales series) and failed to complete any other games. We identify three common failure patterns that contribute to the agents' inability to complete full story arcs, as follows:

Weak planning capability. Agents exhibit repetitive behaviors, such as revisiting the same locations or repeating identical actions, that consume excessive turns. This arises from insufficient planning abilities and limited memory, preventing agents from effectively leveraging past interactions to guide their actions. As a result, even when agents have relevant clues, they often fail to generate appropriate subtask plans, reflecting a significant gap between their observations and behavior.

Poor visual perception. Agents often fail to correctly interpret non-standard layouts, resulting in inappropriate interactions. For example, hidden object games (see Figure 6 in Appendix A.2), which have simple structures solvable through visual recognition, still pose challenges for most baselines.

Deficient lateral thinking. When subtasks are not explicitly defined, flexible problem-solving in uncertain environments becomes essential; however, agents often exhibit inflexible thinking, which ultimately reveals weaknesses in planning and reasoning. For instance, in an escape room game where the goal is to find clues to get out, an agent might focus only on the final goal and keep roaming around the door, missing the important clues nearby.

In contrast, COAST improves planning by incorporating clue memory: it first seeks to gather as many clues as possible, then explores the available situations and uses this information for subtask planning. This approach helps to address the observation-behavior gap and experimentally demonstrates some degree of lateral thinking. As a result, it achieves the highest success and milestone completion rates among the baselines, improving success rate by 5.88 percentage points and milestone completion by up to 2.78 percentage points compared to Claude-3.7-Sonnet Computer-Use. However, in visual novels, our method does not consistently outperform the baseline Claude-

Model	GUI Grounding / Action Execution	Agentic Framework	Success↑ (%)	Milestone↑ (%)	# Steps
GPT-4o	UGround-V1-7B / `pyautogui`	Cradle	0.00	4.56	1000.0
Claude-3.7-Sonnet	UGround-V1-7B / `pyautogui`	Cradle	0.00	6.59	1000.0
	Claude-3.7-Sonnet / `pyautogui`	Cradle	0.00	10.60	1000.0
		Agents S2	0.00	1.20	1000.0
UI-TARS-1.5-7B			0.00	6.93	1000.0
OpenAI CUA			**5.88**	15.39	954.1
Claude-3.7-Sonnet Computer-Use			0.00	17.11	992.4
Claude-3.7-Sonnet Computer-Use + COAST **(Ours)**			**5.88**	**19.89**	966.8
Human Performance (max 1,000 steps)			50.98	78.98	815.5
Human Performance (unlimited)			97.06	100.00	1142.0

Table 2: Comparison of different GUI agents across all 34 video games.

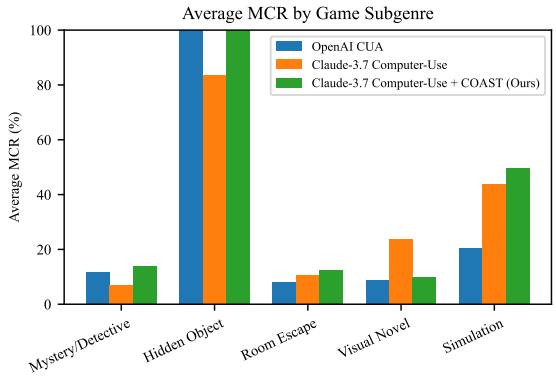

Figure 4: Comparison of average milestone completion rates (MCR) across different game subgenres for three GUI agents.

Game (Subgenre)	Metric	Seeker	Seeker + Solver	COAST
Sherlock Holmes 2 (Mystery/Detective)	Success (%)	0.0	0.0	0.0
	Milestone (%)	37.5	37.5	62.5
	# Steps	1000	1000	1000
Grim Tales: The Bride (Hidden Object)	Success (%)	0.0	100.0	100.0
	Milestone (%)	66.7	100.0	100.0
	# Steps	1000	790	225
Camping Room Escape (Room Escape)	Success (%)	0.0	0.0	0.0
	Milestone (%)	22.2	33.3	44.4
	# Steps	1000	1000	1000
Idol Days Sim Date (Visual Novel)	Success (%)	0.0	0.0	0.0
	Milestone (%)	2.3	25.6	25.6
	# Steps	1000	1000	1000
Sort the Court (Simulation)	Success (%)	0.0	0.0	0.0
	Milestone (%)	83.2	93.0	95.5
	# Steps	1000	1000	1000
Total	Success (%)	0.0	**20.0**	**20.0**
	Milestone (%)	42.4	57.9	**65.6**
	# Steps	1000	958	845

Table 3: Comparison across subgenres under different agent configurations, based on five sampled games.

3.7-Sonnet Computer Use, as shown in Figure 4. Since the observation-behavior gap is smaller in this subgenre, clue-based reasoning provides less advantage, highlighting differences across subgenres.

5.4 Ablation Study

We conducted an ablation study of COAST using Claude to isolate the contributions of the Seeker, Mapper and Solver components.

The results in Table 3 demonstrate that integrating all three components yields the highest overall performance. When the Seeker was used alone, performance was notably poor across all five tested games. This indicates that merely identifying clues without a subsequent planning and solving phase is insufficient. Furthermore, the Seeker + Solver revealed suboptimal milestone completion rate (MCR), particularly in clue-rich mystery/detective and room escape games. This highlights the critical role of the Mapper in formulating effective subtask plans by connecting raw observations and clues.

In conclusion, this ablation study confirms that all three modules are indispensable and contribute to COAST's optimal performance, thereby validating the necessity of our proposed architecture.

5.5 Failure Analysis and Mitigation

To analyze how COAST mitigates agent failures, we manually monitored gameplay and categorized errors into (1) weak planning, (2) poor visual perception, (3) deficient lateral thinking, and (4) inefficient resource management (Table 4).

COAST improved **planning** in Grim Tales (hidden object), where it successfully used prior failed attempts to locate hidden objects. It also allevi-

Game (Subgenre)	GPT-4o + UGround + Cradle	Claude-3.7 Computer-Use	Claude-3.7 Computer-Use + COAST
Sherlock Holmes 2 (Mystery/Detective)	1,2,3	1,2,3	1,2
Grim Tales: The Bride (Hidden Object)	1,2	1	-
Camping Room Escape (Room Escape)	1,2,3	1,2,3	1,2
Idol Days Sim Date (Visual Novel)	4	4	4
Sort the Court (Simulation)	4	4	-

Table 4: Comparison of failure patterns across agents, with codes denoting failure types: 1 = planning, 2 = perception, 3 = lateral thinking, 4 = resource management.

ated **lateral thinking** issues in Sherlock Holmes 2 (mystery/detective) and Camping Room Escape (room escape), where it bridged long-term observation–behavior gaps (*e.g.*, using a blue ball to retrieve a card on the ceiling).

In contrast, results were mixed for **resource management**: while COAST made more efficient yes/no choices in Sort the Court (simulation), it failed to gain high experience score in Idol Days Sim Date (visual novel). Finally, COAST did not mitigate **visual perception** failures, as it inherits the same backbone (Claude-Sonnet-3.7). GPT-4o + UGround + Cradle was especially prone to such perception errors in Grim Tales, likely due to UGround's limited GUI grounding.

Overall, these findings highlight our method's strength in mitigating planning and lateral thinking failures, while limitations remain in perception and resource management.

6 Further Analysis

6.1 Analysis on Contaminations

We investigate whether LLMs were exposed to game-specific information during pretraining and how such contamination affected their in-game performance (Paglieri et al., 2025). We analyze Claude-3.7-Sonnet and GPT-4o. For each subgenre, we selected a representative game and constructed questions targeting specific in-game situations (*e.g.*, actions, interactions, problem-solving steps). These questions assessed whether the models possessed pretrained knowledge of the scenarios and could answer correctly without genuine reasoning.

Claude-3.7-Sonnet showed no signs of contamination, suggesting fair problem solving without memorized content. In contrast, GPT-4o exhibited contamination and hallucination in 3 of 10 questions. In one case, it revealed knowledge of game-specific content but failed to execute the corresponding action in gameplay. This highlights a *knowing-doing gap* (Paglieri et al., 2025): despite contaminated knowledge, GPT-4o's gameplay performance (Table 2) lagged behind Claude-3.7-Sonnet, showing its inability to translate knowledge into action. See Appendix C.1 for details.

6.2 Additional Hint Injection

We evaluate Claude-3.7-Sonnet Computer-Use with and without injected hints (*i.e.*, oracle subtasks), representing an upper performance bound.

In *Sherlock Holmes: The Tea Shop Murder Mystery*, hints enabled the agent to complete all milestones in 758 turns. Without them, it only achieved one milestone after 1,000 turns. This shows that explicit subtasks are highly effective for navigation and problem-solving in well-structured environments. In contrast, for *Computer Office Escape*, which demands complex pattern recognition, the agent failed to clear the first milestone even with hints. This suggests that while hints can streamline task execution, they do not overcome fundamental bottlenecks in complex spatial-logical reasoning over patterns in unstructured environments. Further details and results using a Large Reasoning Model (o4-mini) are available in Appendix C.2 and Appendix C.3.

7 Conclusion

We presented FlashAdventure, a benchmark of 34 diverse Flash-based adventure games designed to evaluate GUI agents' ability to complete full story arcs. To enable reliable evaluation, we introduced CUA-as-a-Judge, an automatic milestone verification agent that closely aligns with human judgments. Our experiments showed that current state-of-the-art GUI agents struggle with planning, perception, and lateral thinking required for full game completion. Therefore, we proposed COAST, a clue-oriented framework that manages long-term clue memory and generates subtasks, significantly improving performance and bridging the observation-behavior gap. Despite these advances, a substantial gap remains between agent and human performance, requiring further improvements.

Limitations

A limitation lies in the reliance on manually defined milestones, which is common across prior video game benchmarks (Chen et al., 2024; Tan et al., 2025; Zhang et al., 2025). To mitigate concerns about *labor intensity*, we designed a lightweight process requiring only 10-15 minutes per game, making it unlikely to become a bottleneck when scaling. For each game, (1) an annotator first selected initial milestones (*e.g.*, clues, items, events) while reviewing the full walkthrough (3-5 minutes), and (2) evaluated whether each was indispensable for completing the story, discarding the rest (7-10 minutes). To enhance the *objectivity* of milestone selection, future work could leverage narrative flow charts (Paschali et al., 2018) to derive milestones directly from structured story representations, providing a more principled evaluation framework.

In addition, our evaluation method, CUA-as-a-Judge, is not fully genre-agnostic. It is suitable within FlashAdventure because selected games lack strict time constraints and narrative progression does not directly depend on character movements (§ 3.2). However, this approach is less applicable to fast-paced action games (*e.g.*, , *Super Mario Bros*), where character movement directly impacts the game state or narrative outcome. Exploring extensions to such reflex-oriented genres, such as pausing gameplay to report stats via clicks, would be an interesting direction for future work.

As mentioned in § 4, we didn't implement any memory management for COAST, since each clue is small enough and the total input didn't exceed the maximum memory limit. However, this may lead to scalability issues as the number of action steps increases. To address this, one can potentially apply summarization (Tan et al., 2025), retrieval (Agashe et al., 2025a), or forgetting (Park et al., 2023) mechanisms to effectively manage memory.

Lastly, running all games for 1,000 steps using proprietary APIs (*e.g.*, GPT-4o, Claude-3.7-Sonnet) incurs substantial cost, posing a barrier to reproduction and large-scale experimentation. However, this issue is not inherent to our benchmark itself, but rather reflects the current need for more efficient open-source GUI agents to reduce reliance on expensive API calls.

Ethics Statement

Regarding game copyright, all games included in our benchmark remain the intellectual property of their original creators and publishers. We emphasize that our use of these games is strictly for **non-commercial, academic research purposes** only. Before including the games, we contacted the respective rights holders (such as developers and publishers) to obtain permission and clarify our intended use. Our usage is limited to evaluation rather than model training, and we do not redistribute or modify the original software. Instead, the games are accessed via authorized platforms like Flashpoint Archive, preserving the original experience. We rely solely on screenshot-based observations for the gameplay, without reverse-engineering or modding any low-level game code. This approach minimizes potential copyright infringement and respects the integrity of the original works.

For the collection of human gameplay demonstrations, we recruited 13 adult participants who provided informed consent prior to the study. Participants were compensated fairly at or above the national minimum wage for each game played, regardless of completion. All procedures, including data collection via input logging and screen recording, were conducted with respect for participant privacy and autonomy. Participants were free to withdraw at any time without penalty. We designed this process to minimize any risks, such as fatigue, and ensure ethical treatment throughout the gameplay sessions. Please refer to Appendix A.5 for further details.

Acknowledgments

We thank all annotators for their dedication and invaluable contributions to the development of FlashAdventure. We are also grateful to Euihyun Tae, Junseo Koo, Wonkwang Lee, Sangwoo Moon, Seyeon Choi, Sieon Park, Kangwook Lee, and the anonymous reviewers for their insightful feedback. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2019-II191082, RS-2021-II211343, No. RS-2022-II220156), the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2023-00274280), and the Korea Radio Promotion Association (Development of Intelligent Docent Service for Information-Disadvantaged Groups). Gunhee Kim is the corresponding author.

References

Saaket Agashe, Jiuzhou Han, Shuyu Gan, Jiachen Yang, Ang Li, and Xin Eric Wang. 2025a. Agent s: An open agentic framework that uses computers like a human. In *ICLR*.

Saaket Agashe, Kyle Wong, Vincent Tu, Jiachen Yang, Ang Li, and Xin Eric Wang. 2025b. Agent s2: A compositional generalist-specialist framework for computer use agents. In *COLM*.

Anthropic. 2024. Claude computer use.

Anthropic. 2025. Claude 3.7 sonnet.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. 2013. The arcade learning environment: An evaluation platform for general agents. *JAIR*.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv:1606.01540*.

Peng Chen, Pi Bu, Jun Song, Yuan Gao, and Bo Zheng. 2024. Can vlms play action role-playing games? take black myth wukong as a study case. *arXiv:2409.12889*.

Peng Chen, Pi Bu, Yingyao Wang, Xinyi Wang, Ziming Wang, Jie Guo, Yingxiu Zhao, Qi Zhu, Jun Song, Siran Yang, Jiamang Wang, and Bo Zheng. 2025. Combatvla: An efficient vision-language-action model for combat tasks in 3d action role-playing games. *arXiv:2503.09527*.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. 2024. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *ACL*.

Edward De Bono and Efrem Zimbalist. 1970. *Lateral thinking*. Penguin London.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. In *NeurIPS Datasets and Benchmarks*.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. 2022. Minedojo: Building open-ended embodied agents with internet-scale knowledge. In *NeurIPS*.

Yue Fan, Lei Ding, Ching-Chen Kuo, Shan Jiang, Yang Zhao, Xinze Guan, Jie Yang, Yi Zhang, and Xin Eric Wang. 2024. Read anywhere pointed: Layout-aware GUI screen reading with tree-of-lens grounding. In *EMNLP*.

Hiroki Furuta, Kuang-Huei Lee, Ofir Nachum, Yutaka Matsuo, Aleksandra Faust, Shixiang Shane Gu, and Izzeddin Gur. 2024. Multimodal web navigation with instruction-finetuned foundation models. In *ICLR*.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. 2025. Navigating the digital world as humans do: Universal visual grounding for GUI agents. In *ICLR*.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024. WebVoyager: Building an end-to-end web agent with large multimodal models. In *ACL*.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxuan Zhang, Juanzi Li, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. 2024. Cogagent: A visual language model for gui agents. In *CVPR*.

Lanxiang Hu, Mingjia Huo, Yuxuan Zhang, Haoyang Yu, Eric P Xing, Ion Stoica, Tajana Rosing, Haojian Jin, and Hao Zhang. 2025a. lmgame-bench: How good are llms at playing games? *arXiv:2505.15146*.

Sihao Hu, Tiansheng Huang, Gaowen Liu, Ramana Rao Kompella, Fatih Ilhan, Selim Furkan Tekin, Yichang Xu, Zachary Yahn, and Ling Liu. 2025b. A survey on large language model-based game agents. *arXiv:2404.02039*.

Sihao Hu, Tiansheng Huang, and Ling Liu. 2024a. Pokellmon: A human-parity agent for pokemon battles with large language models. *arXiv:2402.01118*.

Siyuan Hu, Mingyu Ouyang, Difei Gao, and Mike Zheng Shou. 2024b. The dawn of gui agent: A preliminary case study with claude 3.5 computer use. *arXiv:2411.10323*.

Anssi Kanervisto, David Bignell, Linda Yilin Wen, Martin Grayson, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Tabish Rashid, Tim Pearce, Yuhan Cao, Abdelhak Lemkhenter, Chentian Jiang, Gavin Costello, Gunshi Gupta, Marko Tot, Shu Ishida, Tarun Gupta, Udit Arora, Ryen W. White, and 3 others. 2025. World and human action models towards gameplay ideation. *Nature*.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. 2024. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In *ACL*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *SOSP*.

Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. 2020. Mapping natural language instructions to mobile UI action sequences. In *ACL*.

Seungwon Lim, Sungwoong Kim, Jihwan Yu, Sungjae Lee, Jiwan Chung, and Youngjae Yu. 2025. Visescape: A benchmark for evaluating exploration-driven decision-making in virtual escape rooms. *arXiv:2503.14427*.

Weiyu Ma, Qirui Mi, Yongcheng Zeng, Xue Yan, Runji Lin, Yuqiao Wu, Jun Wang, and Haifeng Zhang. 2024. Large language models play starcraft II:benchmarks and a chain of summarization approach. In *NeurIPS*.

Dang Nguyen, Jian Chen, Yu Wang, Gang Wu, Namyong Park, Zhengmian Hu, Hanjia Lyu, Junda Wu, Ryan Aponte, Yu Xia, Xintong Li, Jing Shi, Hongjie Chen, Viet Dac Lai, Zhouhang Xie, Sungchul Kim, Ruiyi Zhang, Tong Yu, Mehrab Tanjim, and 10 others. 2024. Gui agents: A survey. *arXiv:2412.13501*.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *arXiv:2410.21276*.

OpenAI. 2025a. o3 and o4-mini.

OpenAI. 2025b. Operator.

Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wolczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, Jakob Nicolaus Foerster, Jack Parker-Holder, and Tim Rocktäschel. 2025. BALROG: Benchmarking agentic LLM and VLM reasoning on games. In *ICLR*.

Dongmin Park, Minkyu Kim, Beongjun Choi, Junhyuck Kim, Keon Lee, Jonghyun Lee, Inkyu Park, Byeong-Uk Lee, Jaeyoung Hwang, Jaewoo Ahn, Ameya S. Mahabaleshwarkar, Bilal Kartal, Pritam Biswas, Yoshi Suhara, Kangwook Lee, and Jaewoong Cho. 2025. Orak: A foundational benchmark for training and evaluating llm agents on diverse video games. *arXiv:2506.03610*.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *UIST*.

Maria Eleni Paschali, Nikolaos Bafatakis, Apostolos Ampatzoglou, Alexander Chatzigeorgiou, and Ioannis Stamelos. 2018. Tool-assisted game scenario representation through flow charts. In *ENASE*.

Cheng Qian, Peixuan Han, Qinyu Luo, Bingxiang He, Xiusi Chen, Yuji Zhang, Hongyi Du, Jiarui Yao, Xiaocheng Yang, Denghui Zhang, Yunzhu Li, and Heng Ji. 2025. EscapeBench: Towards advancing creative intelligence of language model agents. In *ACL*.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, Wanjun Zhong, Kuanye Li, Jiale Yang, Yu Miao, Woyu Lin, Longxiang Liu, Xu Jiang, Qianli Ma, Jingyu Li, and 16 others. 2025. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv:2501.12326*.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy P Lillicrap. 2023. Androidinthewild: A large-scale dataset for android device control. In *NeurIPS Datasets and Benchmarks*.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, L. Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. Mastering the game of go with deep neural networks and tree search. *Nature*.

Yuchen Sun, Shanhui Zhao, Tao Yu, Hao Wen, Samith Va, Mengwei Xu, Yuanchun Li, and Chongyang Zhang. 2025. Gui-xplore: Empowering generalizable gui agents with one exploration. *arXiv:2503.17709*.

Weihao Tan, Wentao Zhang, Xinrun Xu, Haochong Xia, Ziluo Ding, Boyu Li, Bohan Zhou, Junpeng Yue, Jiechuan Jiang, Yewen Li, Ruyi An, Molei Qin, Chuqiao Zong, Longtao Zheng, YuJie Wu, Xiaoqiang Chai, Yifei Bi, Tianbao Xie, Pengjie Gu, and 9 others. 2025. Cradle: Empowering foundation agents towards general computer control. In *ICML*.

Fei Tang, Haolei Xu, Hang Zhang, Siqi Chen, Xingyu Wu, Yongliang Shen, Wenqi Zhang, Guiyang Hou, Zeqi Tan, Yuchen Yan, Kaitao Song, Jian Shao, Weiming Lu, Jun Xiao, and Yueting Zhuang. 2025. A survey on (m) llm-based gui agents. *arXiv:2504.13865*.

SIMA Team, Maria Abi Raad, Arun Ahuja, Catarina Barros, Frederic Besse, Andrew Bolt, Adrian Bolton, Bethanie Brownfield, Gavin Buttimore, Max Cant, Sarah Chakera, Stephanie C. Y. Chan, Jeff Clune, Adrian Collister, Vikki Copeman, Alex Cullum, Ishita Dasgupta, Dario de Cesare, Julia Di Trapani, and 75 others. 2024. Scaling instructable agents across many simulated worlds. *arXiv:2404.10179*.

Edward C Tolman. 1948. Cognitive maps in rats and men. *Psychological review*.

Edward Chace Tolman and Charles H Honzik. 1930. Introduction and removal of reward, and maze performance in rats. *University of California publications in psychology*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv:2409.12191*.

Xinyu Wang, Bohan Zhuang, and Qi Wu. 2025a. Are large vision language models good game players? In *ICLR*.

Ziyue Wang, Yurui Dong, Fuwen Luo, Minyuan Ruan, Zhili Cheng, Chi Chen, Peng Li, and Yang Liu. 2025b. Escapecraft: A 3d room escape environment for benchmarking complex multimodal reasoning ability. *arXiv:2503.10042*.

Yue Wu, Xuan Tang, Tom Mitchell, and Yuanzhi Li. 2024. Smartplay : A benchmark for LLMs as intelligent agents. In *ICLR*.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, and Yu Qiao. 2025. OS-ATLAS: Foundation action model for generalist GUI agents. In *ICLR*.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. 2024. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *NeurIPS Datasets and Benchmarks*.

Yuan Yuan, Muyu He, Muhammad Adil Shahid, Jiani Huang, Ziyang Li, and Li Zhang. 2025. Turnabout-llm: A deductive reasoning benchmark from detective games. *arXiv:2505.15712*.

Alex L Zhang, Thomas L Griffiths, Karthik R Narasimhan, and Ofir Press. 2025. Videogamebench: Can vision-language models complete popular video games? *arXiv:2505.18134*.

Xiangxi Zheng, Linjie Li, Zhengyuan Yang, Ping Yu, Alex Jinpeng Wang, Rui Yan, Yuan Yao, and Lijuan Wang. 2025. V-mage: A game evaluation framework for assessing visual-centric capabilities in multimodal large language models. *arXiv:2504.06148*.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. Webarena: A realistic web environment for building autonomous agents. In *ICLR*.

A Details on FlashAdventure

A.1 Problem Formulation

Table 5 presents the complete action set $\mathcal{A}$ used by GUI agents.

Action	Description
`left_click(x, y)`	Left-click at coordinates.
`right_click(x, y)`	Right-click at coordinates.
`middle_click(x, y)`	Middle-click at coordinates.
`double_click(x, y)`	Double-click at coordinates.
`triple_click(x, y)`	Triple-click at coordinates.
`drag_from_(x1,y1)_to_(x2,y2)`	Drag with mouse to coordinates.
`scroll(dir, amount)`	Scroll screen (up, down, left, right).
`key: enter`	Press a specific key (*e.g.*, , Enter).
`type text: hello`	Type text "hello".
`hold_key(ctrl, 2)`	Hold key for duration (*e.g.*, , 2s).
`finish`	The task is finished.

Table 5: Unified action space $\mathcal{A}$.

A.2 Game Selection

We provide a complete list of games available in FlashAdventure organized by subgenre:

Subgenre	Games
Point-and-Click Adventure (Mystery/Detective)	Sherlock Holmes: The Tea Shop Murder Mystery Sherlock Holmes 2 Vortex Point 1 Vortex Point 2 Vortex Point 3 Pierre Hotel Small Town Detective Dakota Winchester's Adventures Saucy Devil Gordon Ray and Cooper 2 Nick Bounty: A Case of the Crabs
Hidden Object	Grim Tales: The Bride Grim Tales: The Legacy Collector's Edition
Room Escape	Computer Office Escape Crimson Room Camping Room Escape Chemical Room Escape Space Museum Escape Vending Machine Room Wood Workshop Escape Geometric Room Escape Game Cafe Escape Machine Room Escape VideoStudio Escape Design House Escape Paint Room Escape Mirror Room Escape Elevator Room Escape
Visual Novel (Dating Sim)	Pico Sim Date Festival Days Sim Date Kingdom Days Idol Days Sim Date
Simulation: Life	Community College Sim
Simulation: Management	Sort the Court

Table 6: List of games included in FlashAdventure.

In addition, we provide human walkthroughs for one selected game per subgenre, including annotated screenshots and step counts. Specifically, Figure 5 presents a *point-and-click adventure (mystery/detective)* example (*Sherlock Holmes: The Tea Shop Murder Mystery*), Figure 6 shows a *hidden object* example (*Grim Tales: The Bride*), Figure 7 depicts a *room escape example* (*Computer Office Escape*), Figure 8 provides a *visual novel (dating sim)* example (*Pico Sim Date*), and Figure 9 illustrates a *simulation* example (*Sort the Court*).

A.3 Analysis on Game Diversity

To clarify that our benchmark offers substantial diversity and presents a challenging testbed for modern GUI agents, we explain the diversity of FlashAdventure along two dimensions: *inter-subgenre* and *intra-subgenre diversity*.

Inter-subgenre diversity: FlashAdventure includes five subgenres, each of which is thematically and structurally distinct, as shown in the Table 7. Mystery/detective games require social interactions with NPCs to gather key clues (*e.g.*, characters explore towns and engage with various people), and room escape games focus on solving complex puzzles in isolated environments. The hidden object subgenre is unique in that it heavily relies on precise visual perception to identify hidden items within a scene. These games also tend to have the shortest playtime (less than 5 minutes) among all subgenres. Finally, visual novel and simulation games both emphasize efficient resource management to achieve high scores, However, they differ in their core objectives: visual novels typically focus on optimizing a single key metric, such as an affection with a specific character, while simulation games often require balancing multiple factors (*e.g.*, wealth, population, happiness), demanding a higher degree of logical reasoning.

Intra-subgenre diversity: Among our 34 games, a fraction display thematically similar narratives, *e.g.*, mystery/detective games. Nevertheless, seemingly relevant games display considerably diverse structures; excelling in one game does not necessarily lead to excelling in another. We provide a game-play statistics collected from three human players who solved 10 mystery/detective games and 10 room escape games in common, as shown in Table 8. The results indicate that each player tends to excel in different games.

- Across both subgenres, rankings and normalized step scores varied substantially across games. These fluctuations were reflected in the sign and magnitude of each player's score, which shifted significantly from game to game, indicating high performance variability.

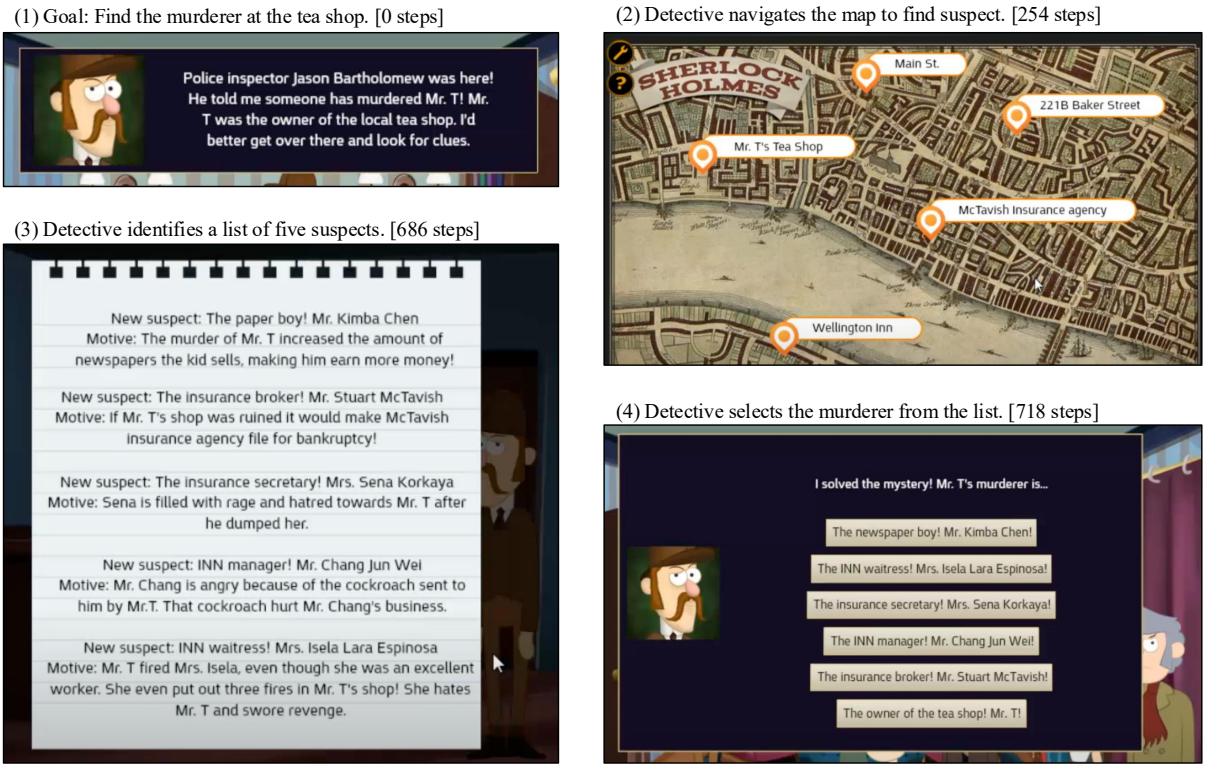

Figure 5: An illustration of the *Point-and-Click Adventure (mystery/detective)* subgenre, showing a human player's walkthrough of *Sherlock Holmes: The Tea Shop Murder Mystery*. The cumulative number of steps is written at the end of each subfigure caption; the game ends at 718 steps.

Figure 6: An illustration of the *Hidden Object* subgenre, showing a human player's walkthrough of *Grim Tales: The Bride*. The cumulative number of steps is written at the end of each subfigure caption; the game ends at 55 steps.

Subgenre	Perceptual Precision	Logical Reasoning	Lateral Thinking	Social Reasoning (Conversation with NPC)	Resource Management
Mystery/Detective	High	High	High	High	Low
Hidden Object	High	Low	Mid	Low	Low
Room Escape	High	High	High	Low	Low
Visual Novel	Mid	Mid	Low	High	High
Simulation	Mid	High	Low	High	High

Table 7: Inter-subgenre diversity: comparison of subgenres across cognitive dimensions.

- **Mystery/Detective:** Player A outperformed the others most frequently, but the standard deviations of normalized scores were still large (A=236, B=302, C=249).
- **Room Escape:** Performance variability was even greater, with standard deviations of normalized scores (A=314, B=361, C=964).

To sum it up, FlashAdventure exhibits both inter- and intra-subgenre diversity, as evidenced by distinct gameplay requirements across subgenres and

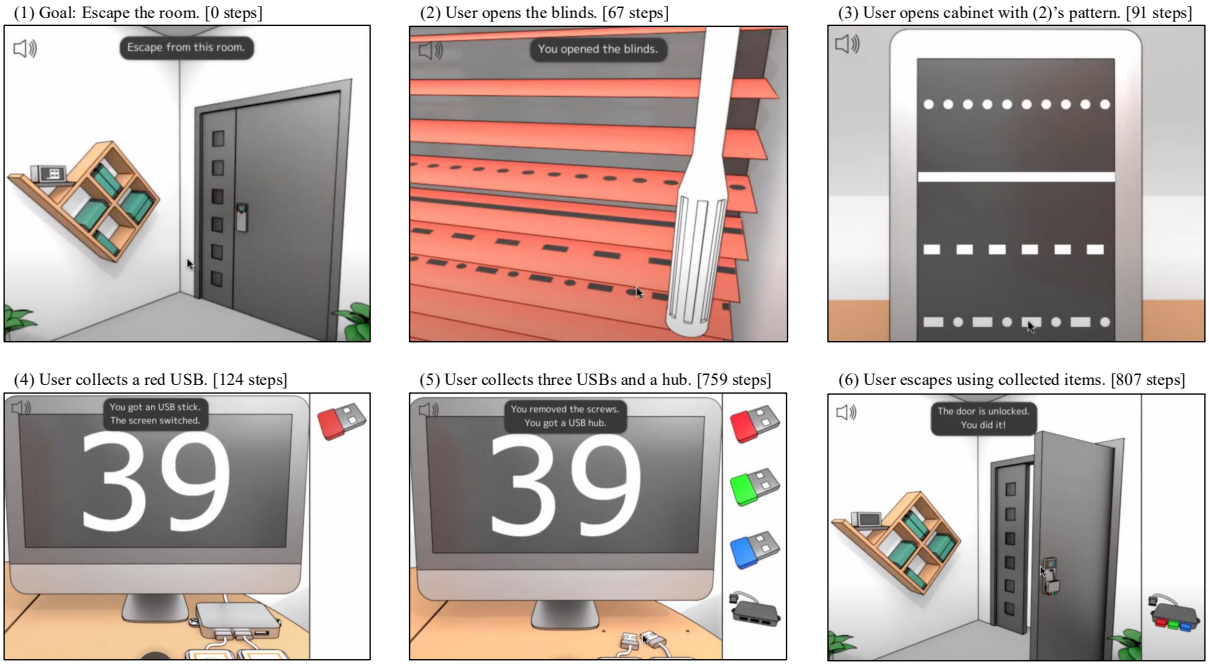

Figure 7: An illustration of the *Room Escape* subgenre, showing a human player's walkthrough of *Computer Office Escape*. The cumulative number of steps is written at the end of each subfigure caption; the game ends at 807 steps.

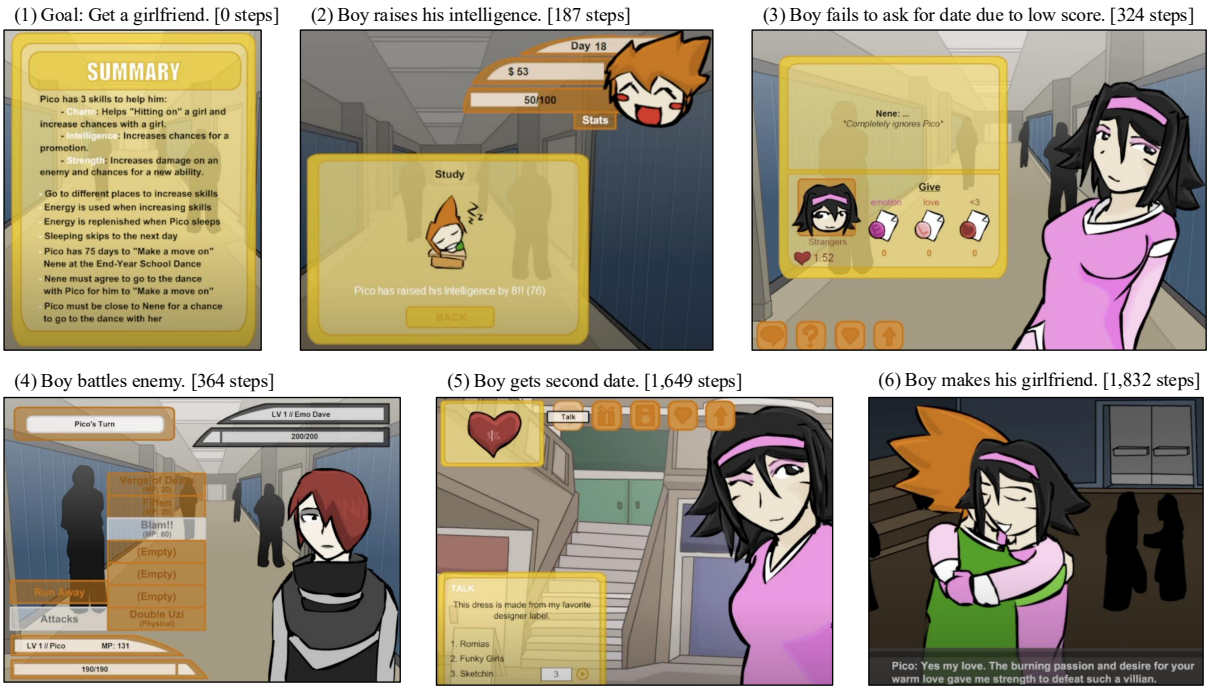

Figure 8: An illustration of the *Visual Novel (Dating Sim)* subgenre, showing a human player's walkthrough of *Pico Sim Date*. The cumulative number of steps is written at the end of each subfigure caption; the game ends at 1,832 steps.

substantial performance variability among human players within the same subgenre.

A.4 Evaluation

Full List of Milestones and Game Introductions. In this section, we provide the full list of milestones for each game, along with a brief introduction to each game.

1. **Sherlock Holmes: The Tea Shop Murder Mystery (6 milestones)**: Detective/Mystery game where a player investigates the accidental death of a tea shop owner. Milestone: (1) Verify that the player checks the number of new suspects via the notebook item (5 total), (2)

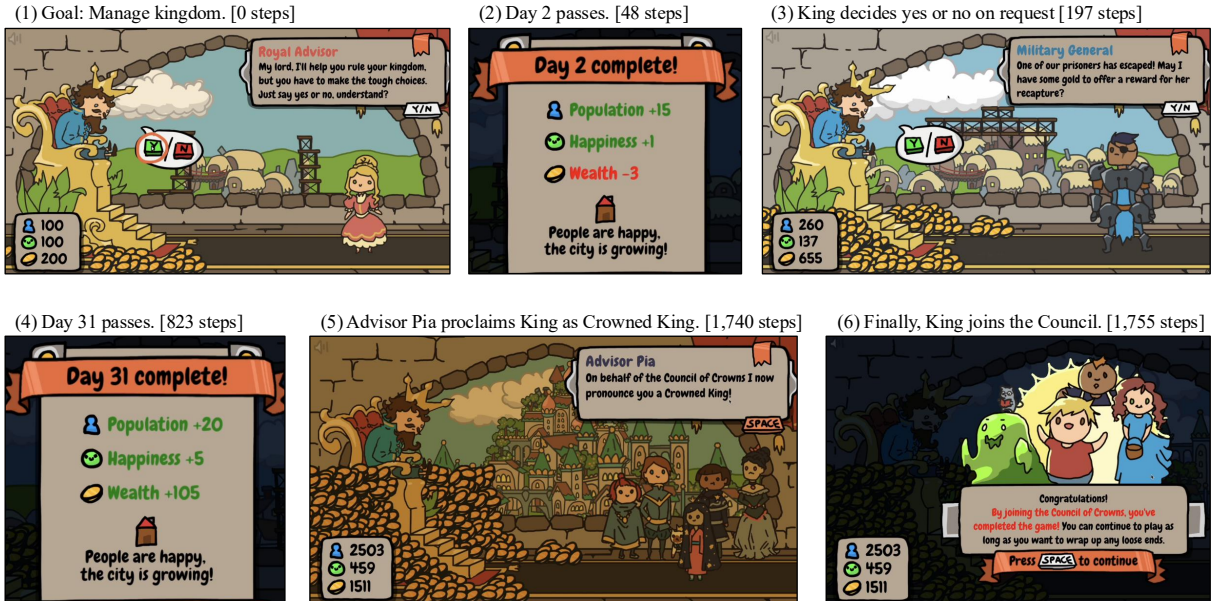

Figure 9: An illustration of the *Simulation* subgenre, showing a human player's walkthrough of *Sort the Court*. The cumulative number of steps is written at the end of each subfigure caption; the game ends at 1,755 steps.

Games	Steps (A)	Steps (B)	Steps (C)	Rank (A)	Rank (B)	Rank (C)	Best Player	Norm. Steps (A)	Norm. Steps (B)	Norm. Steps (C)
Mystery/Detective										
Sherlock Holmes: The Tea Shop Murder Mystery	718	439	752	2	1	3	B	81.7	-197.3	115.7
Sherlock Holmes 2	399	1037	823	1	3	2	A	-354.0	284.0	70.0
Vortex Point 1	707	1461	794	1	3	2	A	-280.3	473.7	-193.3
Vortex Point 2	1147	595	646	3	1	2	B	351.0	-201.0	-150.0
Vortex Point 3	466	588	503	1	3	2	A	-53.0	69.0	-16.0
Pierre Hotel	862	318	1013	2	1	3	B	131.0	-413.0	282.0
Small Town Detective	437	1103	905	1	3	2	A	-378.0	288.0	90.0
Dakota Winchester's Adventures	326	568	502	1	3	2	A	-139.3	102.7	36.7
Saucy Devil Gordon	297	387	455	1	2	3	A	-92.7	7.3	75.3
Nick Bounty: A Case of the Crabs	1292	1660	531	2	3	1	C	131.0	499.0	-630.0
Room Escape										
Computer Office Escape	1625	747	741	3	2	1	C	587.3	-290.7	-296.7
Camping Room Escape	1430	2376	1141	2	3	1	C	-219.0	727.0	-508.0
Space Museum Escape	1283	1119	1936	2	1	3	B	-163.0	-327.0	490.0
Wood Workshop Escape	848	1452	947	1	3	2	A	-234.3	369.7	-135.3
Geometric Room Escape	1247	2221	1099	2	3	1	C	-275.3	698.7	-423.3
Game Cafe Escape	1280	1091	1262	3	1	2	B	69.0	-120.0	51.0
Machine Room Escape	896	903	1206	1	2	3	A	-105.7	-98.7	204.3
VideoStudio Escape	1513	2341	2330	1	3	2	A	-548.3	279.7	268.7
Design House Escape	1671	1402	1482	3	1	2	C	152.7	-116.3	-36.3
Elevator Room Escape	1227	1194	2362	2	1	3	B	-367.3	-400.3	767.7

Table 8: Intra-subgenre diversity: Performance of human participants (A, B, C) in mystery/detective and room escape games, measured by steps, ranks, and normalized steps (*i.e.*, # steps - # average steps per game).

Verify correct suspect selection in the final ending.

2. **Sherlock Holmes 2 (8 milestones)**: Detective/Mystery game where a player investigates a murder at a mansion. Milestone: (1) Check collection of essential items (5 total), (2) Confirm that the fire alarm system on the second floor of the crime scene is opened, (3) Confirm whether the outdoor area of the crime scene has been set on fire, (4) Verify task completion through to the final ending.

3. **Vortex Point 1 (7 milestones)**: Detective/Mystery game where a player solves a supernatural theft in the town of Vortex Point. Milestone: (1) Verify the discovery of 4 hidden locations in the given map, (2) Confirm successful entry into 2956 Vineyard Drive, (3) Confirm successful entry into C. Razy Mental Hospital, (4) Verify task completion through to the final ending.

4. **Vortex Point 2 (5 milestones)**: Detective/Mystery game where a player investigates the paranormal disappearance of a girl linked to a mysterious photo booth. Milestone: (1) Check if the man inside the hidden pub at the crime scene has disappeared, (2) Check if the magician inside the yellow-sign building on Main Street has disappeared and if water is spilling, (3) Verify disappearance of the guard at Town's History Museum, (4) Confirm that the fence at Marshall Square has been opened, (5) Verify task completion through to the final ending.

5. **Vortex Point 3 (5 milestones)**: Detective/Mystery game where a player investigates the disappearance of a tourist at Vortex Lake Hotel. Milestone: (1) Check if the man at Vortex Lake Hotel is unexpectedly eating a hamburger, (2) Confirm whether the toilet in the Souvenir Shop is accessible, (3) Verify the disappearance of the man in the Souvenir Shop, (4) Check if the player can board the boat at Vortex Lake Pier (*i.e.*, reach Private Property), (5) Verify task completion through to the final ending.

6. **Pierre Hotel (6 milestones)**: Detective/Mystery game

where a player searches for their missing girlfriend in a hotel full of vampires. Milestone: (1) Verify that the wine bar staff on the first floor is on a phone call, (2) Confirm access to the broom room on the second floor, (3) Check if the fireplace on the first floor has been extinguished, (4) Confirm that the elevator on the first floor operates properly, (5) Verify disappearance of the vampire at the first floor (front counter), (6) Verify task completion through to the final ending.

7. **Small Town Detective (6 milestones)**: Detective/Mystery game where a player helps Paul the Detective find a missing journalist in a small town. Milestone: (1) Verify the discovery of 5 hidden locations (excluding Paul's Office) in the given map, (2) Verify task completion through to the final ending.

8. **Dakota Winchester's Adventures (6 milestones)**: Detective/Mystery game where a player guides archaeologist Dakota Winchester on a quest to find cursed rubies and unlock Hilda's box. Milestone: (1) Confirm that the stepping stones to the bomb shelter have been placed, (2) Check if a fire is lit in the inner village area, (3) Verify that the temple entrance is open, (4) Confirm that a monkey is eating a banana inside the temple, (5) Check if a yellow ceiling light is illuminating the temple, (6) Verify task completion through to the final ending.

9. **Saucy Devil Gordon (5 milestones)**: Detective/Mystery game where a player helps Gordon become a pirate. Milestone: (1) Confirm that a coconut has been retrieved from a palm tree, (2) Check whether a pineapple has been picked next to the door of the left house in the scene with a tiki statue, (3) Verify that the door in the skull-marked area is open, (4) Confirm that light is shining vertically on the grave, (5) Verify task completion through to the final ending.

10. **Ray and Cooper 2 (6 milestones)**: Detective/Mystery game where a player controls investigator Paul Maxstrong to find the missing Ray and Cooper. Milestone: (1) Check whether the upper window frame inside Cooper's apartment is open, (2) Confirm that "MATTEO'S PIZZA" is closed, (3) Verify that the door under the "NEW WING! COMING SOON!" banner inside the Museum of Oddities is accessible, (4) Confirm the museum manager of the Museum of Oddities has disappeared, and a fence has appeared, (5) Check whether the woman on the box in Snack Shop 1162 has disappeared, (6) Verify task completion through to the final ending.

11. **Nick Bounty: A Case of the Crabs (5 milestones)**: Detective/Mystery game where a player helps detective Nick Bounty solve the murder of a seafood salesman. Milestone (1) Verify the discovery of 4 hidden locations in the given map, (2) Verify task completion through to the final ending.

12. **Grim Tales: The Bride (12 milestones)**: Puzzle (hidden object) adventure game where a player finds hidden objects. Milestone: Verify how many of the 12 hidden objects the player has found in the initial Toolbox scene.

13. **Grim Tales: The Legacy Collector's Edition (12 milestones)**: Puzzle (hidden object) adventure game where a player finds hidden objects. Milestone: Verify how many of the 12 hidden objects the player has found in the initial Barrels on the Road scene.

14. **Computer Office Escape (5 milestones)**: Room escape game where a player solves puzzles and uses items to escape from an office with computers. Milestone: (1) Verify the collection of three colored USBs (red, green, blue) and a USB hub, (2) Verify task completion through to the final ending.

15. **Crimson Room (14 milestones)**: Room escape game where a player wakes up in a locked crimson-colored room and must find a way out. Milestone: (1) Verify the collection of all 13 required items, (2) Verify task completion through to the final ending.

16. **Camping Room Escape (9 milestones)**: Room escape game where a player solves puzzles and use items to escape from a room with a tent. Milestone: (1) Verify the collection of all 8 required items, (2) Verify task completion through to the final ending.

17. **Chemical Room Escape (8 milestones)**: Room escape game where a player solves puzzles and uses items to escape from a laboratory filled with chemical equipment. Milestone: (1) Verify the collection of all 7 required items, (2) Verify task completion through to the final ending.

18. **Space Museum Escape (6 milestones)**: Room escape game where a player is trapped in a space-themed museum and must solve puzzles from exhibits to escape. Milestone: (1–5) Check success based on color-based sequence (red = 0, yellow = 1, green = 2, light blue = 3, blue = 4, pink = 5), (6) Verify task completion through to the final ending.

19. **Vending Machine Room (9 milestones)**: Room escape game where a player must use and manipulate vending machines and related items to escape. Milestone: (1) Verify the collection of all 8 required items, (2) Verify task completion through to the final ending.

20. **Wood Workshop Escape (7 milestones)**: Room escape game where a player solves puzzles using woodworking tools to escape from a workshop. Milestone: (1) Verify the collection of all 6 required items, (2) Verify task completion through to the final ending.

21. **Geometric Room Escape (6 milestones)**: Room escape game where a player must solve puzzles based on geometric patterns to escape. Milestone: (1) Verify the collection of all 5 required items, (2) Verify task completion through to the final ending.

22. **Game Cafe Escape (5 milestones)**: Room escape game where a player solves puzzles to escape from a game-themed café. Milestone: (1) Check if the game console has been acquired, (2) Verify whether the red box was opened in the mini game, (3) Confirm that the player used the gemstone to solve the puzzle and turned off the lights in the mini game, (4) Check if the door key was obtained and used in the mini game, (5) Verify task completion through to the final ending.

23. **Machine Room Escape (4 milestones)**: Room escape game where a player solves mechanical puzzles to escape from a room filled with machines. Milestone: (1–3) Assess the player's progress in replicating the three distinct door pattern based on tablet illustrations, (4) Verify task completion through to the final ending.

24. **VideoStudio Escape (4 milestones)**: Room escape game where a player must solve complex puzzles to

escape from a video studio. Milestone: (1–3) Check how many green lights are lit above the door's 1, 2, and 3 markers, (4) Verify task completion through to the final ending.

25. **Design House Escape (5 milestones)**: Room escape game where a player solves creative design-themed puzzles to escape from a modern house. Milestone: (1) Check whether a paper cube appears in the room with "2" marked on the wall, (2–4) Confirm whether three specific doors have been opened, (5) Verify task completion through to the final ending.

26. **Paint Room Escape (8 milestones)**: Room escape game where a player uses art supplies and solves color-based puzzles to escape from a paint-themed room. Milestone: (1–7) Confirm the color sequence of doors: red → light blue → orange → light blue (with pen present) → navy → yellow → green, (8) Verify task completion through to the final ending.

27. **Mirror Room Escape (5 milestones)**: Room escape game where a player solves reflection and light-based puzzles to escape from a room filled with mirrors. Milestone: (1) Check whether the colorful door has been opened, (2) Confirm opening of the glass bathroom door, (3) Verify that a message appears on the mirror when exposed to steam, (4) Check if the right-side door has been opened, (5) Verify task completion through to the final ending.

28. **Elevator Room Escape (4 milestones)**: Room escape game where a player must repair and operate an elevator to escape from a building. Milestone: (1–2) Confirm which floors are accessible via the elevator buttons (2nd and 3rd floors), (3) Verify the existence of a ceiling passage leading to the pulley room on the 3rd floor, (4) Verify task completion through to the final ending.

29. **Pico Sim Date (continuous score)**: Visual novel game where a player controls Pico, a boy who must win the affection of Nene, a girl he hopes to make his future wife, within 75 days. A Player balances relationship-building activities with battles against romantic rivals. Milestone: Measure the affinity score shown next to the heart icon in the "Stats" menu.

30. **Festival Days Sim Date (continuous score)**: Visual novel game where a player has 30 days to get a boyfriend before the school festival, interacting with four unique boys and experiencing multiple endings. Milestone: Measure the maximum experience value among the four male characters, viewable via the "Stat" option in the Home menu.

31. **Kingdom Days (continuous score)**: Visual novel game set in a medieval kingdom, where a player, as Princess Rose, must build relationships with five characters in 30 days after fleeing her home and seeking refuge in a foreign land. Milestone: Measure the maximum experience value among the five male characters, accessible through the "Menu."

32. **Idol Days Sim Date (continuous score)**: Visual novel game in which a player has 30 days to date boys, improve skills, and prepare for a concert, aiming for love and success as an aspiring idol. Milestone: Measure the maximum experience value among the five male characters, accessible through the "Menu."

33. **Community College Sim (continuous score)**: Life simulation game where a player has 100 days to survive community college, balancing partying, studying, and socializing as a dropout, nerd, foreign kid, or grownup. Milestone: Measure the combined total of Intelligence and Social Skills, accessible via the "Stats" menu at the top of the screen.

34. **Sort the Court (continuous score)**: Management simulation game where a player acts as a king, making yes-or-no choices that affect the kingdom's wealth, population, and happiness while managing events and relationships with citizens. Milestone: Measure the final values of population and happiness, then normalize each by the corresponding maximum values obtained via human walkthroughs; the final score is computed as the average of these normalized values.

A.5 Human GamePlay

To evaluate human performance, we recruited 13 participants over the age of 18 who are fluent in English and have at least an undergraduate-level education. Each participant reviewed and signed the consent form shown in Table 9. Before playing, the participants confirmed that they had never played the target games before. They also completed a short questionnaire (Table 10) assessing their prior experience or familiarity with classic adventure games.

FlashAdventure Human Study Consent Form

This study involves collecting gameplay demonstrations from human participants for the *FlashAdventure* benchmark. By signing this form, participants agree to the following:

1. **Study Overview**: Gameplay demonstrations, including input logs and screen recordings, will be used for game-playing AI research.
2. **Eligibility**: Participants must be 18 or older and provide informed, voluntary consent.
3. **Data Collection**: Collected data includes (a) mouse/keyboard input logs, (b) screen recordings with timestamps, and (c) questionnaire responses. Data will be used solely for research purposes.
4. **Anonymization & Privacy**: All data are anonymized; no personally identifiable information will be disclosed.
5. **Voluntary Participation**: Participants may withdraw at any time and request data deletion without penalty.
6. **Risks**: No major risks are anticipated. Minor fatigue may occur due to screen exposure.
7. **Compensation**: Participants will receive {national minimum wage} per game.
8. **Play Duration**: Recommended play time per game is 15 to 90 minutes, regardless of game completion.
9. **Use of Results**: Results will be published in anonymized form for academic use. Participants may request access to final publications.

Consent Statement: I have read and understood the above. I voluntarily agree to participate and allow my data to be used for research purposes.

Table 9: Consent form provided to human participants.

The participants then used their own desktop/laptop to play the games. They enabled input logging (*i.e.*, mouse and keyboard actions) and screen recording before starting each game.

After completing each game, the participants submitted their input logs and screen recordings to the authors. We then evaluated the demonstrations

Participant Background Questionnaire
Q1. Familiarity with Classic Adventure Genres Have you previously played classic adventure games (*e.g.*, , Mystery/Detective, Hidden Object, Room Escape, Visual Novel, Life/Management Simulation)? For example, have you played escape room games or similar titles more than three times in the past three years? If yes, please list any games you've played extensively or consider yourself skilled at.
Q2. General Gaming Experience Regardless of genre, are you generally familiar with playing video games? If yes, please list any games you've played extensively or consider yourself skilled at.
Q3. Inexperience with Games Are you unfamiliar with video games or have little to no experience playing them?

Table 10: Pre-task questionnaire completed by each participant.

based on predefined milestones. For each game, we recorded the number of steps taken to reach each milestone. In simulation-based games, we tracked the score at specific steps.

For each game, we collected gameplay demonstrations from three different participants. On average, each participant completed approximately seven to eight games, with each game taking about 26 minutes.

A.5.1 Statistics of Observation-Behavior Gap

To quantify the observation-behavior gap, we first focused on games from subgenres that feature discrete milestones. Specifically, we selected 7 games (2 mystery/detective, 2 hidden objects, and 3 room escape). For each game, we randomly sampled 4 milestone-relevant clues that are essential for story progression. Using gameplay logs from 3 human players, we measured the observation-behavior gap as the number of steps between when a clue was first observed and when it was acted upon. We then computed the average gap across the players for each clue.

The overall average observation-behavior gap in Table 11 is **251.1±142.1** steps, which supports the "substantial" step gaps discussed in § 3.3. This value is nearly five times greater than the average human playtime in VisEscape (Lim et al., 2025) (52.8 steps). We also found genre-specific trends: (1) Mystery/detective and room escape games showed larger and more variable gaps across games. (2) Hidden object games had shorter average playtimes and correspondingly smaller, more consistent gaps between games.

Additionally, we provide examples from the game *Sherlock Holmes: The Tea Shop Murder Mystery* illustrating the long-term observation-behavior gap from human player data in Figure 10.

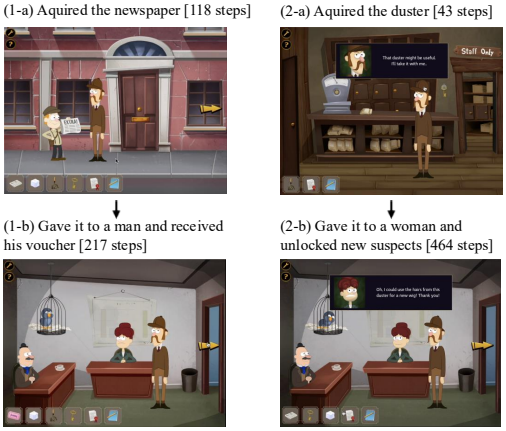

Figure 10: Examples illustrating the long-term observation-behavior gap from human player demonstrations in *Sherlock Holmes: The Tea Shop Murder Mystery*. (1-a) The player acquired a newspaper after 118 steps. (1-b) The player gave the newspaper to a man and received a voucher after 217 steps, resulting in a step gap of 99 between acquiring the newspaper and receiving the voucher. (2-a) The player acquired a duster after 43 steps. (2-b) The player gave the duster to a woman, unlocking new suspects after 464 steps, with a step gap of 421 between acquiring the duster and unlocking new suspects.

A.6 CUA-as-a-Judge

We provide Figure 11 as an example of CUA-as-a-Judge automatically evaluating game progression by interacting with game interfaces to verify milestone completion through checking affection scores and suspect counts.

Milestone structure and evaluation strategy. Games in FlashAdventure differ in how milestones can be observed and assessed. Some follow a step-by-step structure, where completion can only be confirmed by sequentially checking each individual event or condition (*e.g.*, whether a specific interaction occurred or an NPC disappeared). In these 11 games, CUA-as-a-Judge verifies milestones one by one, halting at the first unmet milestone and reporting progress as $K_{\mathrm{m}}/N_{\mathrm{m}}$, as described in § 3.5.

In contrast, 23 games, including visual novel and simulation games, expose cumulative game state (*e.g.*, number of suspects unlocked, locations discovered, affection scores), allowing all N_{m} milestones to be assessed in a single pass. In these cases, the final score reflects the total number of milestones completed.

Evaluation agreement vs. human judge. For each of the 34 games, we evaluate CUA-as-a-Judge on 8 or 9 sampled evaluation instances. If a game

Game	Gap #1	Gap #2	Gap #3	Gap #4	Average
Sherlock Holmes: The Tea Shop Murder Mystery	95.3	561.7	321.7	655.3	408.5 ± 251.7
Dakota Winchester's Adventures	113.7	322.0	211.0	229.3	219.0 ± 85.4
Grim Tales: The Bride	57.0	62.0	72.0	91.7	70.7 ± 15.3
Grim Tales: The Legacy Collector's Edition	58.3	78.7	95.7	122.0	88.7 ± 27.0
Computer Office Escape	1001.3	584.3	120.3	47.3	438.3 ± 444.3
Camping Room Escape	307.0	632.3	150.3	57.3	286.8 ± 252.4
Space Museum Escape	303.0	157.3	240.0	284.0	246.1 ± 64.8

Table 11: Observation-behavior gap analysis across games. Values show four gaps and their average ± standard deviation.

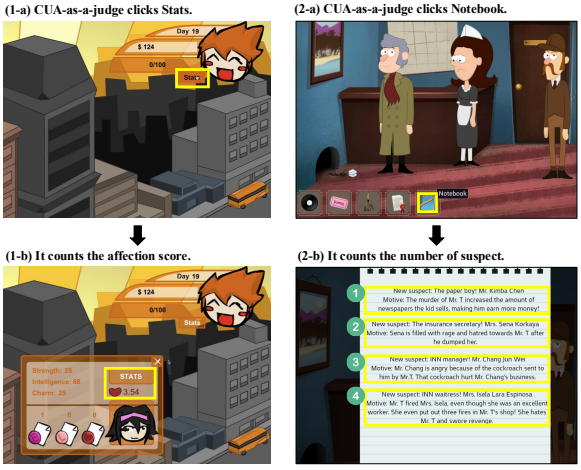

Figure 11: CUA-as-a-Judge verifies game progress by interacting with the environment. Left: (1-a) The judge clicks the "Stats" button in *Pico Sim Date* to (1-b) verify the character's affection score. Right: (2-a) The judge clicks the "Notebook" item in *Sherlock Holmes: The Tea Shop Murder Mystery* to (2-b) count the number of suspects (5) identified by the player, confirming milestone completion.

has 4 milestones, for example, the possible milestone completions range from 0 to 4 (*i.e.*, 5 levels). To account for variations in final gameplay states (even when the milestone count is the same), we construct two distinct examples per level, yielding up to 10 evaluation instances per game. From these, we randomly sample 8 or 9 cases and assess whether CUA-as-a-Judge correctly estimates the milestone completion level, resulting in a total of 300 evaluation samples across all games.

Our comparison shows a high agreement, with an accuracy of 94.00%, Spearman correlation of 0.9912, and Pearson correlation of 0.9999. Per-game accuracy results are presented in Table 12.

Game	Accuracy
Sherlock Holmes: The Tea Shop Murder Mystery	9/9
Sherlock Holmes 2	6/8
Vortex Point 1	9/9
Vortex Point 2	8/9
Vortex Point 3	9/9
Pierre Hotel	9/9
Small Town Detective	9/9
Dakota Winchester's Adventures	8/9
Saucy Devil Gordon	8/9
Ray and Cooper 2	9/9
Nick Bounty: A Case of the Crabs	8/8
Grim Tales: The Bride	7/9
Grim Tales: The Legacy Collector's Edition	7/9
Computer Office Escape	9/9
Crimson Room	6/8
Camping Room Escape	9/9
Chemical Room Escape	9/9
Space Museum Escape	9/9
Vending Machine Room	6/8
Wood Workshop Escape	6/8
Geometric Room Escape	9/9
Game Cafe Escape	9/9
Machine Room Escape	9/9
VideoStudio Escape	9/9
Design House Escape	9/9
Paint Room Escape	9/9
Mirror Room Escape	8/9
Elevator Room Escape	6/8
Pico Sim Date	9/9
Festival Days Sim Date	9/9
Kingdom Days	9/9
Idol Days Sim Date	9/9
Community College Sim	9/9
Sort the Court	9/9
Total	282/300

Table 12: Per-game accuracy of CUA-as-a-Judge compared to human judgment.

B Details on Experiments

B.1 Implementation Details

GPT-4o (`gpt-4o-2024-08-06`), Claude-3.7-Sonnet (`claude-3-7-sonnet-20250219`), OpenAI CUA (`computer-use-preview-2025-03-11`), and Claude-3.7-Sonnet Computer-Use (`claude-3-7-sonnet-20250219` with tool version `20250124`) were all accessed via propri-

etary APIs using default hyperparameters. All experiments were conducted on 28 personal computers capable of running the Flash games (7 Macs and 21 Windows). For UGround-V1-7B and UI-TARS-1.5-7B, we ran them on a single NVIDIA RTX A6000 GPU and accessed the APIs via VLLM (Kwon et al., 2023). Each experiment typically took about 12 to 18 hours per game.

For all GUI agents, the most recent 10 trajectories were explicitly provided to the agent. Beyond this, we made efforts to preserve each baseline's original configuration as much as possible. Some baselines (Tan et al., 2025; Agashe et al., 2025b; Qin et al., 2025) implemented retrieval or summary modules to realize long-term memory, while for OpenAI CUA and Claude-3.7-Sonnet Computer-Use, the internal mechanisms are not publicly known, so whether they implement long-term memory remains unclear.

For our COAST framework, we use Claude-3.7-Sonnet Computer-Use for Clue Seeker and Problem Solver, and Claude-3.7-Sonnet for Clue Mapper. For the hyperparameters $\{N_{\text{seek}}, N_{\text{solve}}\}$, we set them to (5,2) for visual novels and simulation genres, and (15,5) for all others. We set $K = 5$ for Clue Mapper.

Due to the high API cost, all experiments were conducted as single runs.

B.1.1 Detailed Prompts

We provide detailed prompts of (1) CUA-as-a-Judge, (2) baseline agents, and (3) COAST for the game 'Sherlock Holmes: The Tea Shop Murder Mystery' as an example:

CUA-as-a-Judge Prompt

Your task is to locate and click the blue notebook icon in the game interface (usually found at the far right of the inventory bar). Once the notebook is opened, do not perform any additional actions. Carefully read the displayed text (do not scroll) and count how many times the phrase `New Suspect` appears in the note. If there is nothing written in the note, it means there is no new suspect. Once counted, output the result in the following format:

Output Format

`New Suspect: [Num of occurrences]`

Do not perform any further interactions after counting. Your task ends once the count is provided.

Basic Prompt (for baseline agents)

[Instruction]
You're a game agent solving an adventure game. Adventure games can involve variables and often require lateral thinking and creative problem-solving. Rather than focusing solely on the ultimate goal, try to solve problems using common sense and imaginative thinking.

[Important Notice]
To ensure a smooth experience and prevent unexpected issues, please read and follow the instructions below carefully:

- Do not repeatedly interact with objects that may not be clickable. Even if nothing happens, your action might have already been registered-or the object may simply not be interactive.
- Avoid repeating the same action excessively. A lack of visible response or an unexpected result does not mean the action failed.
- Do not click the setup or question buttons in the top-left corner of the screen.
- Do not leave the game screen.
- Do not access Settings or Help:
  - Do not click the wrench icon (Settings).
  - Do not click the question mark icon (Help or Hints).

[Prompt]
You are now the lead agent in "Sherlock Holmes: The Tea Shop Murder Mystery" - a point-and-click detective game. Everything in this game is controlled with simple mouse clicks. Search the tea shop carefully, interact with objects, uncover hidden clues, and solve the murder mystery.

[Basic Rules]

- Observe carefully. Don't rush or search randomly - every detail could matter. What seems ordinary might hide secrets.
- Think logically and laterally. Don't brute-force or guess wildly. Follow the trail of clues using reason, insight, and creativity.
- Investigate step by step. Whether solving a mystery or unlocking a puzzle, progress comes from careful observation and deduction.
- Trust your instincts. Sometimes the answer is hidden in plain sight - don't overlook the obvious.

[Information]
Mr. T has been murdered by someone. But we don't know who the criminal is.
Mr. T was the owner of the local tea shop.
...

[Completion Condition]
The game is considered complete only when you

have gathered enough evidence, logically identified the culprit, and correctly selected the true suspect during the final decision. A clear message or scene will confirm that the case has been solved.

[Completion Signal]
Once you've solved the case, type the following to signal completion:
`[Done]`

COAST: Clue Seeker Prompt

{Basic Prompt}

[Action Prompt]
Your job is to extract all clues visible on the screen and summarize what you observe.

[Expected Behavior]
Return the following:

- A list of clues found. Each clue should include:
  - `clue`: short name or label for the clue
  - `description`: what the clue seems to represent or imply
  - `location`: where the clue was found - include both the specific spot and a short description of the surrounding environment
  - `type`: categorize the clue using one of the following:
    - `item`: Tools or objects the player can collect or interact with
    - `note`: Written information such as signs, notes, or documents
    - `code`: Visible numbers, passcodes, symbols, or puzzle sequences
    - `visual cue`: Visual cues like arrows, lighting, gaze direction, or environmental emphasis
    - `status`: UI state indicators
    - `conversation`: Any meaningful dialogue or internal monologue text shown on screen
  - `interactable`: true if the player can interact with this clue, false otherwise
  - `usage_hint`: how this clue might be used or why it is important
- A short summary of the current observation-action pair (episodic memory). Each memory should include:
  - `action`: what you did and what was observed as a result
  - `place`: what the current area/room looks like, its notable features

[Result Format]
Respond in this exact JSON format, wrapped in `<RESPO>` tags, like this:

```
<RESPO>
{
  "clues": [
    {
      "clue": "<Short name or label for
      ↪  the clue>",
      "description": "<What the clue seems
      ↪  to represent or imply>",
      "location": "<Specific spot +
      ↪  surrounding context>",
      "type": "<item | note | code | visual
      ↪  cue | status | conversation>",
      "interactable": <true | false>,
      "usage_hint": "<How this clue might
      ↪  be used or why it could be
      ↪  important>"
    }
  ],
  "episodic_memory": [
    {
      "action": "<What the player did and
      ↪  what was observed as a result>",
      "place": "<Description of the room
      ↪  or environment where the action
      ↪  occurred>"
    }
  ]
}
</RESPO>
```

If your response does not strictly follow this format, it will be discarded. Do not store the same clue more than once in memory.

[Clues]
{clues}

COAST: Clue Mapper Prompt

{Basic Prompt}

[Action Prompt]
You are a reasoning agent matching current clues with episodic memory from past gameplay. Your goal is to find meaningful — and possibly non-obvious — connections between clues and past events using:

- *abductive reasoning*: inferring the most plausible explanation from incomplete or ambiguous information
- *lateral thinking*: creative, indirect associations beyond surface similarity

For each clue:

- Identify an episodic memory where the clue could plausibly have helped — even if the connection is indirect or interpretive.
- Determine the concrete action the player should now take based on that match.

[Expected behavior]

- Use each clue's `description`, `type`, and `usage_hint` to inform your reasoning.
- Go beyond surface-level similarity — prioritize plausible, creative mappings.
- Favor abductive reasoning — what might this clue explain or reveal?

- Explore lateral connections — metaphorical, thematic, or functional.
- Only match when a memory clearly presents a situation where the clue could have been helpful.
- Be specific and grounded. If uncertain, omit the match.
- Return up to 5 of the most insightfully plausible matches — quality over quantity.
- If no valid matches are found, return:

```
<RESPO>[Nobody]</RESPO>
```

Do not fabricate connections.

[Result Format]
Respond in JSON format like this:

```
<RESPO>
[
  {
    "clue": {
      "name": "<short clue name>",
      "description": "<what the clue seems
      ↪  to represent or imply>",
      "location": "<specific spot +
      ↪  environment context>",
     "type": "<item | note | code | visual
      ↪  cue | status | conversation>",
      "interactable": <true | false>,
      "usage_hint": "<how this clue might
      ↪  be used or why it could be
      ↪  important>"
    },
    "related_memory": "<a specific past
    ↪  observation where this clue would
    ↪  have been useful>",
    "expected_action": "<concrete action
    ↪  the player should now take using
    ↪  this clue>"
  }
]
</RESPO>
```

If the format is not strictly followed, the response will be discarded.

[Clues]
{clues}

[Episodic Memory]
{episodic memory}

Do not generate mapping memory that has already succeeded.

[Success Memory]
{success memory}

COAST: Problem Solver Prompt

{Basic Prompt}

[Action Prompt]
You are now trying to solve games using previously discovered clues and their related observations. Each clue is paired with a related episodic memory from your past exploration. Use this information to decide on the most logical and effective next action to progress in the game.
Each episodic memory is structured as:

- `action`: what you did and what you observed
- `place`: description of the environment or scene

[Expected behavior]

- Choose a clear goal based on the clue-to-memory mapping you are given.
- Take a meaningful action in the game to pursue that goal.
- Summarize what happened after the action.
- Return the result in the format below:
  - `episodic_memory`: what happened during this step (at least one item)
  - `mapping_result` (optional), if your action clearly relates to a clue, include:
    - `goal`: what problem you were trying to solve
    - `reasoning`: why that clue and memory were relevant to the goal
    - `result`: one of `"Success"` or `"Fail"` depending on whether your action clearly used the clue to solve a problem

[Result Format]
Respond in JSON format like this:

```
<RESPO>
{
  "episodic_memory": [
    {
      "action": "<What the player did and
      ↪  what was observed as a result>",
      "place": "<Description of the room
      ↪  or area where it happened>"
    }
  ],
  "mapping_result": [
    {
      "clue": "<Clue name used in this
      ↪  action>",
      "related_memory": "<The relevant
      ↪  episodic memory entry this clue
      ↪  connects to>",
      "goal": "<What the player was trying
      ↪  to achieve by using the clue>",
      "reasoning": "<Why this clue and
      ↪  memory logically support that
      ↪  goal>"
    }
  ],
  "result": "<Success | Fail>"
}
</RESPO>
```

If your response does not strictly follow the format, it will be discarded.

[Important about Success]
An action is only considered a `"Success"` if the clue

was effectively used to solve a specific puzzle or problem — for example, using a pattern from books on a shelf to open a secret compartment based on a past observation. Simply interacting with objects is not enough.
To qualify as a true `Success`, the outcome must include a meaningful in-game change, such as:

- Obtaining an item
- Unlocking a new area
- Updating a stat
- Triggering story progression

[Action Process]

- Select a goal based on the mapping between clue and memory. Explain why this goal is relevant.
- Act accordingly in the game world.
- At the final turn, assess the outcome.
- If the clue helped solve a problem, it's a `"Success"`; otherwise, it's a `"Fail"`.
- When selecting a goal, be sure to reference the clue's metadata:
  - `type`: the kind of clue (*e.g.*, , item, code, note, etc.)
  - `interactable`: whether the player can use it
  - `usage_hint`: what the clue suggests it might be useful for

[Mapping History]
Clue: {clue 1}
Related Memory: {related memory 1}
Expected Action: {expected action 1}
...

B.2 Experimental Results

We report the full experimental results on all 34 games using OpenAI CUA, Claude-3.7-Sonnet Computer-Use, and ours with human performance in Table 13. In addition, we report the results of other GUI agents in Table 14.

Overall, despite having the smallest parameter size, UI-TARS-1.5-7B achieves a higher milestone completion rate than proprietary LLM-based models like GPT-4o + UGround-V1-7B + Cradle and Claude-3.7-Sonnet + Claude-3.7-Sonnet + Agent S2, somehow consistent with its original claim of effectively solving browser-based games. UGround-V1-7B, which incorporates a GUI grounding module, demonstrated weaker grounding ability compared to Claude-3.7-Sonnet. Finally, although Agent S2 performs well in desktop GUI tasks (Xie et al., 2024), it shows the lowest performance among the compared agents on FlashAdventure, highlighting the gap between everyday digital interactions and the complexities of adventure games.

B.2.1 Multi-Run Experiments

We additionally conducted multi-run experiments with three trials, and the results are shown in Table 15. Claude-3.7 Computer-Use + COAST achieves a higher SR and MCR than baseline agents. These trends are consistent with the single-run results in Table 2.

An exception occurs in *Sort the Court*, where both agents complete the full story arc (SR > 0%), suggesting stronger performance in NPC conversations (social reasoning) and resource management. However, this result is expected, as COAST already achieved a near-perfect MCR (95.5%) in Table 2.

MCR values can exceed 100% in visual novel and simulation sub-genres because the continuous score is normalized relative to human performance. For instance, an MCR of 101.2% in *Sort the Court* indicates that the agent performs on par with, or even better than, a human player.

B.2.2 Fair Comparison between COAST and Baselines

To demonstrate that the success of COAST stems from its novel Seek-Map-Solve architecture rather than improved prompt engineering, we conducted an additional control experiment in which the baseline agent (*i.e.*, Claude-3.7 Computer-Use) was instructed using COAST-style prompts in Appendix B.1.1. Specifically, we concatenated prompts from the Clue Seeker, Clue Mapper, and Problem Solver modules, made minor adjustments for coherence, and then provided the final combined prompt to the Claude-3.7 Computer-Use.

Table 16 shows that prompt-optimized one is slightly worse than the naive Claude-3.7 Computer-Use. The prompt-optimized version fails to solve additional milestones in the mystery/detective and room escape subgenres involving long-term observation-behavior gaps, where COAST demonstrates clear gains. These findings support our claim that COAST 's performance improvements stem not from better prompting, but from its structured Seek-Map-Solve framework.

Game	OpenAI CUA			Claude-3.7 Computer-Use			Claude-3.7 Computer-Use +COAST (Ours)			Human (max 1K steps)			Human (unlimited)		
	SR (%)	MCR (%)	Stp (#)	SR (%)	MCR (%)	Stp (#)	SR (%)	MCR (%)	Stp (#)	SR (%)	MCR (%)	Stp (#)	SR (%)	MCR (%)	Stp (#)
Point-and-Click Adventure (Mystery/Detective)															
Sherlock Holmes: The Tea Shop Murder Mystery	0.0	50.0	1000	0.0	16.7	1000	0.0	50.0	1000	100.0	100.0	636.6	100.0	100.0	636.6
Sherlock Holmes 2	0.0	37.5	1000	0.0	37.5	1000	0.0	62.5	1000	66.7	95.8	740.7	100.0	100.0	753.0
Vortex Point 1	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	66.7	81.0	833.7	100.0	100.0	987.3
Vortex Point 2	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	66.7	86.7	747.0	100.0	100.0	796.0
Vortex Point 3	0.0	20.0	1000	0.0	0.0	1000	0.0	20.0	1000	100.0	100.0	519.0	100.0	100.0	519.0
Pierre Hotel	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	66.7	88.9	726.7	100.0	100.0	731.0
Small Town Detective	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	66.7	94.4	780.7	100.0	100.0	815.0
Dakota Winchester's Adventures	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	100.0	100.0	562.7	100.0	100.0	562.7
Saucy Devil Gordon	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	100.0	100.0	379.7	100.0	100.0	379.7
Ray and Cooper 2	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	100.0	100.0	952.5	100.0	100.0	952.5
Nick Bounty: A Case of the Crabs	0.0	20.0	1000	0.0	20.0	1000	0.0	20.0	1000	33.3	80.0	843.7	100.0	100.0	1161.0
Hidden Object															
Grim Tales: The Bride	100.0	100.0	274	0.0	91.7	1000	100.0	100.0	225	100.0	100.0	91.7	100.0	100.0	91.7
Grim Tales: The Legacy Collector's Edition	100.0	100.0	164	0.0	75.0	1000	100.0	100.0	647	100.0	100.0	121.0	100.0	100.0	121.0
Room Escape															
Computer Office Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	66.7	80.0	851.3	100.0	100.0	987.0
Crimson Room	0.0	35.7	1000	0.0	35.7	1000	0.0	28.6	1000	0.0	64.3	1000.0	100.0	100.0	3133.0
Camping Room Escape	0.0	22.2	1000	0.0	22.2	1000	0.0	44.4	1000	0.0	44.4	1000.0	100.0	100.0	1734.0
Chemical Room Escape	0.0	25.0	1000	0.0	25.0	1000	0.0	25.0	1000	0.0	79.2	1000.0	100.0	100.0	1303.3
Space Museum Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	33.3	77.8	994.3	100.0	100.0	1128.3
Vending Machine Room	0.0	22.2	1000	0.0	33.3	1000	0.0	44.4	1000	100.0	100.0	664.3	100.0	100.0	664.3
Wood Workshop Escape	0.0	14.3	1000	0.0	42.9	1000	0.0	42.9	1000	33.3	90.5	949.3	100.0	100.0	1104.0
Geometric Room Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	33.3	55.5	882.3	100.0	100.0	1371.7
Game Cafe Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	33.3	73.3	933.3	100.0	100.0	993.7
Machine Room Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	100.0	100.0	819.3	100.0	100.0	819.3
VideoStudio Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	25.0	1000.0	100.0	100.0	1697.7
Design House Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	33.3	86.7	971.1	100.0	100.0	1183.7
Paint Room Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	91.7	1000.0	100.0	100.0	1171.7
Mirror Room Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	53.3	1000.0	100.0	100.0	2352.7
Elevator Room Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	66.7	1000.0	100.0	100.0	1161.7
Visual Novel (Dating Sim)															
Pico Sim Date	0.0	0.1	1000	0.0	4.6	1000	0.0	0.9	1000	0.0	13.0	1000.0	33.3	100.0	1751.0
Festival Days Sim Date	0.0	15.7	1000	0.0	0.6	1000	0.0	11.2	1000	0.0	41.7	1000.0	100.0	100.0	2798.7
Kingdom Days	0.0	2.2	1000	0.0	19.5	1000	0.0	1.5	1000	100.0	100.0	958.0	100.0	100.0	958.0
Idol Days Sim Date	0.0	17.4	1000	0.0	69.8	760	0.0	25.6	1000	100.0	100.0	873.0	100.0	100.0	873.0
Simulation															
Sort the Court	0.0	37.4	1000	0.0	78.1	1000	0.0	95.5	1000	33.3	88.1	894.0	66.7	100.0	1145.7
Community College Sim	0.0	3.2	1000	0.0	9.2	1000	0.0	3.9	1000	0.0	27.4	1000.0	100.0	100.0	1992.0
Average	**5.9**	15.4	954.1	0.0	17.1	992.9	**5.9**	**19.9**	966.8	51.0	79.0	815.5	97.1	100.0	1142.0

Table 13: Detailed results for all 34 games across three GUI agents (*i.e.*, (1) OpenAI CUA, (2) Claude-3.7-Sonnet Computer-Use, and (3) Claude-3.7-Sonnet Computer-Use with COAST) and human baselines.

C Further Analysis

C.1 Analysis on Contaminations

For the contamination check, we composed 1 to 3 questions per game(5 games) and assessed the extent of contamination in each LLM model through human evaluation. The specific questions and their expected answers are provided in Table 17. The evaluation criteria are as follows:

1. **Answer Accuracy**: How well does the response align with information from the walkthrough?

2. **Conclusion Accuracy**: Is the judgement about the outcome in each response correct?

3. **Behavior Match**: Does the agent's actual behavior during evaluation align with the correct standard?

The results of this evaluation for each questions are presented in Table 18. These results show the contamination observed in several questions for GPT-4o. For example, in Question 2, the model correctly identified that there is a boy on the street and that a newspaper can be obtained. However, instead of stating that the newspaper should be acquired from the boy, the model hallucinated alternative actions-such as purchasing it from a store or suggesting other implausible interactions-indicating that its knowledge was partially contaminated or imprecise.

C.2 Additional Hint Injection

To enable hint injection, we constructed hints from complete walkthroughs by decomposing them into subtasks aligned with each game's milestone structure. Rather than embedding these hints in the initial prompt, we adopted a conditional injection strategy: hints were introduced only when the

Game	Claude-3.7 + UGround + Cradle			Claude-3.7 + Claude-3.7 + Cradle			Claude-3.7 + Claude-3.7 + Agent S2			GPT-4o + UGround + Cradle			UI-TARS-1.5-7B		
	SR (%)	MCR (%)	Stp (#)	SR (%)	MCR (%)	Stp (#)	SR (%)	MCR (%)	Stp (#)	SR (%)	MCR (%)	Stp (#)	SR (%)	MCR (%)	Stp (#)
Point-and-Click Adventure (Mystery/Detective)															
Sherlock Holmes: The Tea Shop Murder Mystery	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Sherlock Holmes 2	0.0	0.0	1000	0.0	12.5	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	12.5	1000
Vortex Point 1	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Vortex Point 2	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Vortex Point 3	0.0	20.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	20.0	1000	0.0	0.0	1000
Pierre Hotel	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Small Town Detective	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Dakota Winchester's Adventures	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Saucy Devil Gordon	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Ray and Cooper 2	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Nick Bounty: A Case of the Crabs	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Hidden Object															
Grim Tales: The Bride	0.0	0.0	1000	0.0	83.3	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	75.0	1000
Grim Tales: The Legacy Collector's Edition	0.0	0.0	1000	0.0	75.0	1000	0.0	16.7	1000	0.0	0.0	1000	0.0	83.3	1000
Room Escape															
Computer Office Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Crimson Room	0.0	14.3	1000	0.0	28.6	1000	0.0	0.0	1000	0.0	14.3	1000	0.0	0.0	1000
Camping Room Escape	0.0	22.2	1000	0.0	22.2	1000	0.0	11.1	1000	0.0	11.0	1000	0.0	22.2	1000
Chemical Room Escape	0.0	25.0	1000	0.0	25.0	1000	0.0	0.0	1000	0.0	12.5	1000	0.0	0.0	1000
Space Museum Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Vending Machine Room	0.0	22.2	1000	0.0	11.1	1000	0.0	0.0	1000	0.0	11.1	1000	0.0	11.1	1000
Wood Workshop Escape	0.0	14.3	1000	0.0	28.6	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Geometric Room Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Game Cafe Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Machine Room Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
VideoStudio Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Design House Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Paint Room Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Mirror Room Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Elevator Room Escape	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000	0.0	0.0	1000
Visual Novel (Dating Sim)															
Pico Sim Date	0.0	0.9	1000	0.0	1.1	1000	0.0	0.0	1000	0.0	0.5	1000	0.0	0.0	1000
Festival Days Sim Date	0.0	10.6	1000	0.0	0.6	1000	0.0	0.0	1000	0.0	5.4	1000	0.0	8.5	1000
Kingdom Days	0.0	4.5	1000	0.0	4.5	1000	0.0	0.0	1000	0.0	1.5	1000	0.0	3.0	1000
Idol Days Sim Date	0.0	23.3	1000	0.0	11.6	1000	0.0	0.0	1000	0.0	17.4	1000	0.0	7.0	1000
Simulation															
Sort the Court	0.0	67.0	1000	0.0	54.4	1000	0.0	12.8	1000	0.0	64.6	1000	0.0	11.6	1000
Community College Sim	0.0	2.0	1000	0.0	1.9	1000	0.0	0.2	1000	0.0	2.4	1000	0.0	1.3	1000
Average	0.0	6.6	1000	0.0	10.6	1000	0.0	1.2	1000	0.0	4.6	1000	0.0	6.9	1000

Table 14: Detailed results for all 34 games across five GUI agents (*i.e.*, (1) Claude-3.7-Sonnet + UGround-V1-7B / `pyautogui` + Cradle, (2) Claude-3.7-Sonnet + Claude-3.7-Sonnet / `pyautogui` + Cradle, (3) Claude-3.7-Sonnet + Claude-3.7-Sonnet / `pyautogui` + Agent S2, (4) GPT-4o + UGround-V1-7B / `pyautogui` + Cradle, and (5) UI-TARS-1.5-7B).

model failed to make progress toward a milestone over a predefined number of steps.

Specifically, in *Sherlock Holmes: The Tea Shop Murder Mystery*, a hint was injected if no milestone progress was made after 100 steps. Without hints, the model achieved only a single milestone across 1,000 turns. In contrast, with hint injection, it successfully completed all milestones within 758 turns. In *Computer Office Escape*, hints were provided every 50 turns. Despite offering guidance on relevant patterns and subsequent tasks in Phase 1, the model failed to achieve even the first milestone.

The hints used for each game and the final prompt with injected hints are summarized below:

Hints for "Sherlock Holmes: The Tea Shop Murder Mystery"

Phase 1

- Go to Main Street and talk to the newspaper vendor. He will give you a new suspect.
- Get the following items at the tea shop: a gold key and an insurance document.

Phase 2

- Go to McTavish Insurance agency. Talk with people in there. You can get some suspects.
- Give the newspaper to bird.

Phase 3

- Go to Wellington Inn and give sugar at hole. After that, talk with people in there. You can get some suspects.

Game (Subgenre)	Metric	GPT-4o + UGround + Cradle	Claude-3.7 + Claude-3.7 + Cradle	Claude-3.7 Computer-Use	Claude-3.7 Computer-Use + COAST
Sherlock Holmes 2 (Mystery/Detective)	Success (%)	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	Milestone (%)	4.2 ± 7.2	16.7 ± 7.2	37.5 ± 0.0	54.2 ± 7.2
	# Steps	1000.0 ± 0.0	1000.0 ± 0.0	1000.0 ± 0.0	1000.0 ± 0.0
Grim Tales: The Bride (Hidden Object)	Success (%)	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	100.0 ± 0.0
	Milestone (%)	0.0 ± 0.0	75.0 ± 8.3	83.3 ± 8.3	100.0 ± 0.0
	# Steps	1000.0 ± 0.0	1000.0 ± 0.0	1000.0 ± 0.0	281.3 ± 61.6
Camping Room Escape (Room Escape)	Success (%)	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	Milestone (%)	3.7 ± 6.4	7.4 ± 12.8	25.9 ± 6.4	40.7 ± 6.4
	# Steps	1000.0 ± 0.0	1000.0 ± 0.0	1000.0 ± 0.0	1000.0 ± 0.0
Idol Days Sim Date (Visual Novel)	Success (%)	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	Milestone (%)	17.1 ± 4.8	10.1 ± 2.7	79.1 ± 35.8	45.3 ± 36.3
	# Steps	1000.0 ± 0.0	1000.0 ± 0.0	920.0 ± 138.6	1000.0 ± 0.0
Sort the Court (Simulation)	Success (%)	0.0 ± 0.0	0.0 ± 0.0	33.3 ± 57.7	66.7 ± 57.7
	Milestone (%)	55.9 ± 8.0	53.9 ± 0.4	83.4 ± 7.1	101.2 ± 6.2
	# Steps	1000.0 ± 0.0	1000.0 ± 0.0	990.7 ± 16.2	973.0 ± 23.5
Total	Success (%)	0.0 ± 0.0	0.0 ± 0.0	6.7 ± 11.5	**33.3 ± 11.5**
	Milestone (%)	16.2 ± 2.0	32.6 ± 4.4	62.0 ± 4.4	**68.3 ± 7.2**
	# Steps	1000.0 ± 0.0	1000.0 ± 0.0	982.1 ± 26.2	850.9 ± 8.6

Table 15: Comparison of GUI agents in multi-run experiments on five sampled games. Values represent mean ± standard deviation.

Game (Subgenre)	Metric	Claude-3.7 Computer-Use	Claude-3.7 Computer-Use (prompt-optimized)	Claude-3.7 Computer-Use + COAST
Sherlock Holmes 2 (Mystery/Detective)	Success (%)	0.0	0.0	0.0
	Milestone (%)	37.5	37.5	62.5
	# Steps	1000	1000	1000
Grim Tales: The Bride (Hidden Object)	Success (%)	0.0	0.0	100.0
	Milestone (%)	91.7	75.0	100.0
	# Steps	1000	1000	225
Camping Room Escape (Room Escape)	Success (%)	0.0	0.0	0.0
	Milestone (%)	22.2	22.2	44.4
	# Steps	1000	1000	1000
Idol Days Sim Date (Visual Novel)	Success (%)	0.0	0.0	0.0
	Milestone (%)	69.8	17.4	25.6
	# Steps	760	1000	1000
Sort the Court (Simulation)	Success (%)	0.0	0.0	0.0
	Milestone (%)	78.1	80.8	95.5
	# Steps	1000	1000	1000
Total	Success (%)	0.0	0.0	**20.0**
	Milestone (%)	59.8	46.6	**65.6**
	# Steps	952	1000	845

Table 16: Comparison of Claude-3.7 Sonnet Computer-Use variants (*i.e.*, baseline, prompt-optimized, and COAST) across five sampled games.

Phase 4

- Go to McTavish Insurance agency and give duster to Sena.
- Go to the tea shop, put record on the gramophone.

Phase 5

ID	Game	Question	Expected Answer
Q1	Sherlock Holmes: The Tea Shop Murder Mystery	Is there anything of interest inside the insurance agency? If so, what should the player do there to progress in the game?	There are two employees and a bird in the room. The player should talk to the employees to obtain new suspects and can replace the newspapers in the birdcage.
Q2	Sherlock Holmes: The Tea Shop Murder Mystery	When the player arrives at Main Street, please describe exactly what they should do, including interactions, items to obtain, and how those items contribute to progressing the story.	Talk to the boy who is selling newspapers to get one, and unlock new suspects.
Q3	Grim Tales: The Legacy Collector's Edition	What should the player do when they reach the barrels on the board? Please describe any required interactions.	The player should check the item list at the bottom of the screen and search for the hidden items within the game scene.
Q4	Grim Tales: The Legacy Collector's Edition	What should the player find items at barrels on the board? List up please.	Bee, Dwarf, Pepper, Clower, Star, Shell, Crab, Tomato, Train, Darts, Mushroom, and Machete
Q5	Grim Tales: The Legacy Collector's Edition	Where is the bee at the barrels on the board?	The bee is sitting on the rock directly above the toad, in the center of the screen.
Q6	Camping Room Escape	What color is the tent? Can I enter it? If I can, what item can I find inside?	The tent is yellow-green. Yes, you can enter and find a sheet of paper.
Q7	Camping Room Escape	There is a case secured with a padlock. What is the combination to unlock it, and what happens after the case is opened?	The code is 6428. It opens to get cardboard tubes.
Q8	Pico Sim Date	There's a place where you can buy drinks. What happens when you do? Does it affect any of your stats or relationships?	Buying drinks increases charm.
Q9	Pico Sim Date	When the player goes to work at the game, what actions do they perform, and what are the outcomes or benefits?	You can work like a mailroom tech and earn money from it. If your intelligence score is above a certain threshold, you have the chance to get promoted. (Note: You can attempt promotion only once per day.)
Q10	Community College Sim	Which role should the player choose to start the game with Job Level 2 already unlocked?	Choose the "Adult" role.

Table 17: Questions for contamination check with expected answers.

LLM		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
GPT-4o	Correct	✗	∼	✗	✗	✗	∼	✗	✗	∼	✗
	Conclusion	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
	Behavior	✓	✓	✓	-	✓	✗	✗	-	✓	-
Claude-3.7-Sonnet	Correct	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Conclusion	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Behavior	✗	✓	✓	-	✓	✗	✗	-	✓	-

Table 18: Evaluation results for contamination check questions. Responses that are only partially correct or contain hallucinated content are marked with a '∼' symbol. If the agent does not encounter a situation where the relevant knowledge is needed, it is marked with a '-'.

- Use crab to cut the chain at the tea shop.
- Enter the door which is opened by cutting the chain.

Phase 6

- Enter the door which is opened by cutting the chain at the tea shop and click the portrait.
- Use the clues and the suspect note to select the culprit.

Hints for "Computer Office Escape"

Phase 1

- To unlock the computer on the bookshelf, observe the pattern of the books and click the matching pattern on the computer.
- Find the orange blind. Look at the pattern on it and use that to unlock the gray box beneath it.

Phase 2

- Go to the desk with the quarter-circle shape. Use the circular patterns on the desks to unlock the pattern on the computer.
- Connect the two items you've obtained to the computer to rotate it.

Phase 3

- Use the red USB you found to open a drawer.

Phase 4

- Place the cup you obtained on the coffee machine. When it's filled with coffee, drink it.
- Use the pattern on the bottom of the coffee cup to unlock the box directly below the coffee machine.

Phase 5

- Use the brown key to unlock the closed door.

Phase 6

- On top of the dollar-sign cabinet, there's a gray box. Look at the pattern on the sticky note next to it, and press the buttons on the tablet on the floor to receive a code.
- Input the code into the gray box on top of the dollar-sign cabinet.

Phase 7

- Place the object with red squares on top of the computer keyboard with the green USB.
- From the revealed information, enter the letters in alphabetical order into the computer.

Phase 8

- Obtain the tablet with the blue USB inserted.

Phase 9

- Find the sign with the rotating arrow.
- Use this to solve the similar pattern on the computer.

Phase 10

- Find the drawer that can be opened with the green USB.
- Using the information from the tablet and the computer in the room with the orange blinds, solve the puzzle on the newly obtained black tablet.

Phase 11

- Use the black tablet to activate a flashlight. Shine it on the sunflower.
- Memorize the directions shown and find the flower-patterned box. Enter the directions in order.

Phase 12

- Find the box that can be opened with the silver key. It's located beyond the brown door.
- Inside the opened box, look at the color pattern of the electric wires and use it to solve the problem on the tablet labeled A B C D outside.

Phase 13

- Check the email on the black tablet.
- To enter the password, match each symbol with the corresponding table shape and input accordingly.

Phase 14

- The QR code from the tablet is only half. Find the missing half and place the tablet there.
- Use the tablet with the blue USB to scan the completed QR code, as it has the scanning function.

Phase 15

- Find the drawer that can be opened with the blue USB.
- Use a screwdriver to open the hub on the computer labeled "39."

Phase 16

- Use the 3 USBs and the hub you obtained to open the final door.

Hint Injection Prompt for "Sherlock Holmes: The Tea Shop Murder Mystery"

[Instruction]
You're a game agent solving an adventure game. Adventure games can involve variables and often require lateral thinking and creative problem-solving. Rather than focusing solely on the ultimate goal, try to solve problems using common sense and imaginative thinking.

[Important Notice]
To ensure a smooth experience and prevent unexpected issues, please read and follow the instructions below carefully:

- Do not repeatedly interact with objects that may not be clickable. Even if nothing happens, your action might have already been registered, or the object may simply not be interactive.
- Avoid repeating the same action excessively. A lack of visible response or an unexpected result does not mean the action failed.
- Do not click the setup or question buttons in the top-left corner of the screen.
- Do not leave the game screen.
- Do not access Settings or Help:
  - Do not click the wrench icon (Settings).
  - Do not click the question mark icon (Help or Hints).

[Prompt]
You are now the lead agent in "Sherlock Holmes: The Tea Shop Murder Mystery", a point-and-click detective game. Everything in this game is controlled with simple mouse clicks. Search the tea shop carefully, interact with objects, uncover hidden clues, and solve the murder mystery.

[Basic Rules]

1. Observe carefully. Don't rush or search randomly. Every detail could matter. What seems ordinary might hide secrets.
2. Think logically and laterally. Don't brute-force or guess wildly. Follow the trail of clues using reason, insight, and creativity.
3. Investigate step by step. Whether solving a mystery or unlocking a puzzle, progress comes from careful observation and deduction.

4. Trust your instincts. Sometimes the answer is hidden in plain sight. Don't overlook the obvious.

[Information]

- Mr. T was murdered by someone. But we don't know who the criminal is.
- Mr. T was the owner of the local tea shop.
- You must find the true criminal of this case.
- You can find items at crime scenes and collect clues by traveling to different locations using the map.
- Clues you find in the environment will be added to the inventory at the bottom of the screen. To use an item, click on it and place it in the appropriate location.

[Completion Condition]
The game is considered complete only when you have gathered enough evidence, logically identified the culprit, and correctly selected the true suspect during the final decision. A clear message or scene will confirm that the case has been solved.

[Completion Signal]
Once you've solved the case, type the following to signal completion: `[Done]`

[Hint]

- Go to Main Street and talk to the newspaper vendor. He will give you a new suspect.
- Get the following items at the tea shop: a gold key and an insurance document.

C.3 Large Reasoning Models

To evaluate whether a reasoning-specialized model improves action planning, we replaced GPT-4o + UGround-V1-7B + Cradle with o4-mini (OpenAI, 2025a) + UGround-V1-7B + Cradle, which is designed for stronger high-level reasoning.

In *Sherlock Holmes: The Tea Shop Murder Mystery*, o4-mini achieved 0 milestones, and in *Camping Room Escape*, it achieved 2: showing no improvements over GPT-4o's results. While it generated more complex and structured reasoning, it did not improve action planning.