Argus: A Game-Agnostic Accessibility Overlay for Visually Impaired Players

Victor Olojo
Algoma University
Sault Ste. Marie, Ontario, Canada
dolojo@algomau.ca

Wenjun Lin*
Algoma University
Brampton, Canada
randylin@algomau.ca

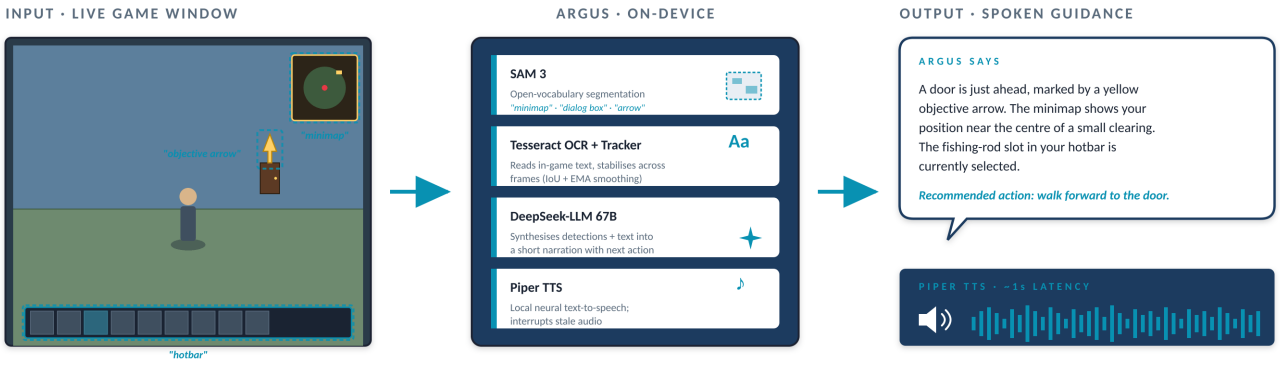

Figure 1: Overview of the Argus pipeline. A live game frame is captured via XDG Desktop Portal, segmented with open-vocabulary text prompts using SAM 3, processed by Tesseract OCR over detected UI regions, stabilised by an IoU-based tracker, and condensed by DeepSeek-LLM into a short narration spoken in near real time by Piper TTS. The system runs as a non-invasive overlay and requires no game-side modification.

Abstract

More than 2.2 billion people globally live with near or distance vision impairment, yet video game accessibility depends almost entirely on per-game effort: studios that do not invest in built-in features leave their titles permanently unreachable, and community-made accessibility mods, though valuable, are confined to games that expose a modding API. We present Argus, a game-agnostic accessibility overlay that delivers real-time spoken narration for any video game without game-side modification or developer cooperation, operating solely on the rendered screen output. Argus captures the live game window, segments UI regions using open-vocabulary prompts (SAM 3), extracts text via OCR, and synthesises context-aware guidance through a local LLM and neural TTS engine—a pipeline that generalises across art styles without per-title training data. We demonstrate Argus on tutorial gameplay from *Old School RuneScape* and *Minecraft*, two visually and mechanically distinct titles, confirming that the same prompt vocabulary transfers across both rendering styles without reconfiguration.

CCS Concepts

• **Human-centered computing** → **Accessibility systems and tools**; • **Computing methodologies** → *Image segmentation.*

*Corresponding author.

Keywords

video game accessibility, visual impairment, foundation model segmentation, SAM 3, OCR, large language models, text-to-speech, real-time overlay, game-agnostic

1 Introduction

Video games reach more than 3 billion players worldwide, yet they remain largely inaccessible to the estimated 2.2 billion people who live with a near or distance vision impairment~[10]. The medium communicates spatial and narrative information through visual UI elements such as minimaps, health bars, and objective markers that standard screen readers cannot parse.

Two categories of tools address this gap. Developer-cooperative approaches, seen in *The Last of Us Part II* and *Forza Motorsport*, require per-title studio investment and leave the back catalogue permanently unreachable~[2, 5]. Community-made mods extend specific games but are confined to titles with modding APIs, depend on individual volunteers, and face DMCA legal risk~[9]. Game-agnostic tools sidestep the cooperation requirement: *Surveyor* supports audio-based spatial exploration without engine integration~[6], and the concurrent *GamerAstra* applies multi-agent vision-language models to generate real-time audio cues~[7]. None delivers continuous, context-aware spoken narration across arbitrary game UIs without game-side modification.

We present *Argus*, a game-agnostic accessibility overlay that narrates any video game in real time without game-side modification or developer cooperation. Argus captures the live game window and operates purely on the rendered screen output: it segments UI regions using open-vocabulary prompts with SAM_3[1], extracts text via OCR, and synthesises context-aware spoken guidance through a local LLM and neural TTS engine. We demonstrate the system on two visually distinct titles in Section 3 and outline a roadmap for formal evaluation with blind and low-vision participants in Section 4.

2 System Design

Argus runs as a background process and exposes three operating modes (single-image inspection, offline video transcription, and live overlay), all sharing one backend pipeline shown in Fig.~1. **Frame capture and overlay.** In live mode, Argus uses XDG Desktop Portal and PipeWire to capture the game window without a game-side hook or kernel driver. A floating semi-transparent panel renders narration text above the game at a configurable opacity.

Open-vocabulary segmentation. Captured frames feed into SAM_3[1], which accepts short natural-language prompts such as "minimap", "health bar", and "yellow objective arrow", and returns pixel-accurate masks with instance identities. Because the prompts describe visual semantics rather than sprite coordinates, the same vocabulary transfers across art styles and genres without per-game training data. This prompt-driven generalisation is the core mechanism behind Argus's game-agnosticism.

OCR and temporal stabilisation. Each mask from SAM~3 crops a UI region, which Argus converts to grayscale, upscales bicubically, and passes to Tesseract~[8] with an adaptive threshold. A `StableBoxRegistry` matches incoming masks against a short detection history using IoU overlap and an exponential moving average hit-count; it emits a stable (label, bounding-box, OCR-text) triple only after the hit-count crosses a configurable threshold, suppressing per-frame detection flicker.

Context-aware narration. Stable triples accumulate into a detection history that, combined with the latest OCR text, forms a structured prompt for DeepSeek-LLM 67B~[3]. The model returns a short scene description and a recommended next action; new narrations interrupt any currently playing audio immediately.

Speech synthesis. Piper~[4] synthesises each narration locally with neural TTS, reaching approximately 1,s end-to-end latency after the LLM response.

3 Demonstration

We ran Argus on an AMD Radeon RX 6800 XT under ROCm/HIP. The full segmentation-and-narration update cycle completes in approximately 3,s end-to-end. After the LLM responds, Piper synthesises and delivers spoken audio within approximately 1,s, acceptable latency for adventure, RPG, and turn-based titles.

That 1,s response time held across two titles chosen to stress style invariance. *Old School RuneScape* (*OSR*) is a pixel-art tile-based RPG; Minecraft is a voxel 3D sandbox. We used identical prompt vocabulary for both with zero reconfiguration. SAM_3[1] correctly localised the minimap, hotbar, and objective markers on every sampled *OSR* frame. Tesseract recovered on-screen instructional text. DeepSeek-LLM named visible UI elements, surfaced the active objective, and proposed a next action across two scenarios: navigating a door indicated by an objective marker, and walking to a fishing spot via a minimap waypoint. The same pipeline transferred to Minecraft without reconfiguration. The key result is style invariance: identical prompt vocabulary transfers across fundamentally different rendering engines.

Prompt vocabulary and narration format were shaped through collaboration with XRnavigation, an accessible navigation technology organisation. Brandon Biggs and their visually impaired team members directly influenced prompt wording, narration structure, and TTS interrupt behaviour.

4 Limitations and Future Work

The current update cycle completes in approximately 3,s end-to-end, which suits adventure, RPG, and turn-based genres but limits Argus for fast-paced action games where UI state changes within fractions of a second.

Closing that latency gap motivates our four planned directions: (i) model quantisation and prompt batching to bring the update cycle into action-game range; (ii) formal user studies with blind and low-vision participants measuring task completion, perceived workload, and qualitative preference; (iii) community-authored expansion of the prompt vocabulary to cover a wider range of genres and UI conventions; and (iv) a port of the frame-capture layer to Windows and macOS to reach a broader player base.

References

[1] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, et al. 2025. SAM 3: Segment Anything with Concepts. arXiv:2511.16719 https://arxiv.org/abs/2511.16719

[2] Laura Dale. 2025. 2025 Video Game Accessibility Recap. Access-Ability. https://access-ability.uk/2025/12/05/2025-video-game-accessibility-recap/

[3] DeepSeek-AI. 2024. DeepSeek LLM: Scaling Open-Source Language Models with Longtermism. arXiv:2401.02954 https://arxiv.org/abs/2401.02954

[4] Michael Hansen. 2023. Piper: A fast, local neural text-to-speech system. https://github.com/rhasspy/piper.

[5] María Eugenia Larreina and Carme Mangiron. 2025. Advancing Game Accessibility With Audio Description: User Perspectives and Recommendations. *Journal of Audiovisual Translation* 8, 2 (2025). doi:10.47476/jat.v8i8.2025.355

[6] Vishnu Nair, Hanxiu Zhu, Peize Song, Jizhong Wang, and Brian A. Smith. 2024. Surveyor: Facilitating Discovery Within Video Games for Blind and Low Vision Players. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. ACM. doi:10.1145/3613904.3642615

[7] Tianrun Qiu, Changxin Chen, Sizhe Cheng, Xuyang Liu, Xumeng Wang, Zhicong Lu, and Yuxin Ma. 2025. GamerAstra: Supporting 2D Non-Twitch Video Games for Blind and Low-Vision Players through a Multi-Agent Framework. arXiv:2506.22937 https://arxiv.org/abs/2506.22937

[8] Ray Smith. 2007. An Overview of the Tesseract OCR Engine. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, Vol. 2. IEEE, 629–633. doi:10.1109/ICDAR.2007.4376991

[9] Yuan Tao and Xin Xu. 2025. Long-Tail Crowdsourcing for Game Accessibility: A Study of Player Co-Creation in Hearthstone's Accessibility Mods. ICMRED 2025 Symposium. https://www.researchgate.net/publication/392881933

[10] World Health Organization. 2023. Blindness and vision impairment. Fact Sheet. https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment