GTK: A Gaze-Based Interaction Toolkit in XR

László Kopácsi*,[1]   Abdulrahman Mohamed Selim[1]   Michael Barz[1]   Daniel Sonntag[1,2]

[1]Interactive Machine Learning, German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany
[2]Applied Artificial Intelligence, University of Oldenburg, Oldenburg, Germany

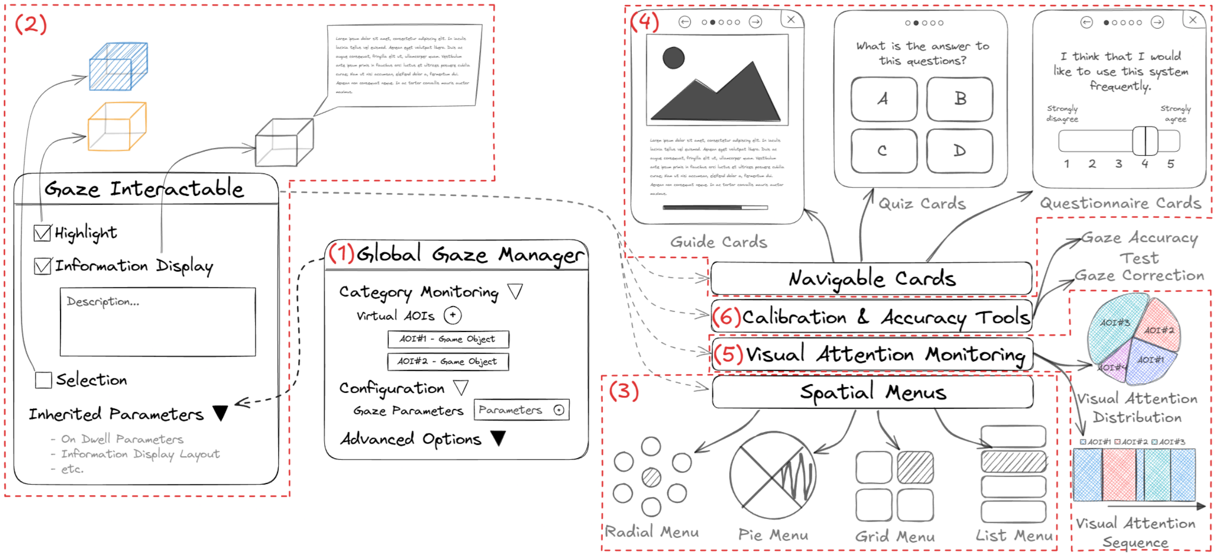

Figure 1: Module design of GTK. GTK provides modular components, including gaze-aware interactables, spatial context menus, as well as tools for onboarding, visual attention monitoring, and eye-tracking accuracy assessment.

ABSTRACT

Gaze interaction is becoming increasingly common in Extended Reality (XR), with eye-tracking sensors being integrated more frequently into head-mounted displays. However, software support for developing gaze-based interaction remains fragmented, often forcing developers to reimplement common interaction patterns. In this paper, we present the Gaze Interaction Toolkit (GTK), an open-source Unity package that combines active gaze interaction and passive attention monitoring within a single modular framework. GTK extends Unity's XR Interaction Toolkit with components for multimodal selection, gaze-contingent displays, spatial menus, and structured onboarding. The toolkit was developed through iterative consultation with academic and industrial partners in robotics, education, XR research, and eye-tracking applications. To validate GTK's usability, we conducted a two-part user study situated in an industrial sorting task with guided onboarding. Pilot results (N=6) informed protocol refinements for the ongoing evaluation. We report on the toolkit's architecture, demonstrate its application through the sorting scenario, and discuss planned extensions.

Index Terms: Gaze-based Interaction, Unity Toolkit, Eye Tracking, Virtual Reality.

1 INTRODUCTION

Although eye-tracking hardware in Head-Mounted Displays (HMDs) has matured, software support for rapid prototyping remains fragmented. Usually, developers often rely on commercial SDKs like Tobii[1], or general-purpose frameworks such as the Mixed Reality Toolkit (MRTK)[2] and Unity's XR Interaction Toolkit (XRI)[3]. While these tools provide access to gaze data and basic dwell-time selection, they often lack high-level, modular components for more complex interaction patterns. As a result, researchers frequently implement features such as gaze-contingent displays, robust multimodal selection, and attention analytics from scratch [5], increasing development time.

A further challenge is that current frameworks typically prioritise either *active* gaze interaction or *passive* gaze data acquisition. General-purpose toolkits (e.g., XRI and MRTK) mainly support gaze as a pointing mechanism for selection and targeting, whereas research-oriented toolkits such as *ARETT* [13] and *UnitEye* [24] focus on robust logging and event extraction but are often decoupled from interaction logic. This separation complicates applications that require both in the same system. In addition, gaze interaction is often unfamiliar to users [3], which increases the need for structured onboarding to avoid frustration.

To address this gap, we introduce the Gaze Interaction Toolkit (GTK), a modular and extensible toolkit for the Unity game engine[4]. GTK is built on OpenXR and supports a wide range of eye-tracking HMDs. It extends Unity's XRI with pre-configured prefabs for gaze-responsive User Interface (UI) elements, validation

---

*e-mail: laszlo.kopacsi@dfki.de

[1] https://developer.tobii.com/xr/ (Accessed 10 Jan 2026)

[2] https://github.com/microsoft/MixedRealityToolkit-Unity (Accessed 10 Jan 2026)

[3] https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@3.0 (Accessed 10 Jan 2026)

[4] https://github.com/DFKI-Interactive-Machine-Learning/de.dfki-iml.xr-gaze-interaction-toolkit

tools, and an onboarding system. This streamlines the development of XR training applications using multimodal interaction. In this paper, we present the core features of GTK, including gaze-aware interactables, spatial menus, attention monitoring and onboarding tools. Additionally, we discuss how GTK has been utilised in research projects to demonstrate its effectiveness in complex Virtual Reality (VR) scenarios, and how we plan to further evaluate some of its functionalities in future user studies.

2 GAZE INTERACTION TOOLKIT (GTK)

GTK extends Unity's XRI with a component-based architecture, as shown in Figure 1, to ensure seamless integration into existing projects. The framework is controlled by the **Global Gaze Manager** (Element (1) in Figure 1), which handles global interaction states. This component enables developers to configure gaze parameters, such as dwell time, and define default behaviours, such as object highlighting. These settings can be overridden by individual components when necessary. Additionally, the manager allows for the definition of Virtual Areas-of-Interest (AOIs) to support visual attention monitoring.

2.1 Interactable Components

We provide interactable components (Element (2) in Figure 1) that wrap Unity's standard XRI interactables, simplifying their interface while extending their features. The toolkit distinguishes between two abstraction levels based on the required functionality. The *SimpleGazeInteractable* handles basic, low-level gaze events with limited overhead. In contrast, the full *GazeInteractable* supports advanced features, including gaze-based object highlighting, selection, and information displays. Both interactables support multiple interaction methods: gaze-only (dwell or border-crossing), controller-only, and multimodal (gaze and controller combined). In addition, they enable active interaction features such as gaze-contingent highlighting (i.e., visual feedback when fixating on an object), information overlays inspired by Rivu et al. [23], and custom event triggers upon selection.

2.2 Spatial Menus

GTK includes prefabs for common **context menu layouts** (Element (3) in Figure 1), including Radial, Grid, and Pie menus inspired by the Lattice Menu [14], as well as List menus based on designs by Monteiro et al. [20]. While the system architecture supports multi-level structures, the current implementation focuses on single-level menus. By default, menus are optimised for world-referenced placement, as Lediaeva et al. [18] suggest this is faster and less physically demanding than body-referenced placement for single-level structures. Furthermore, recent literature [17] indicates that menu shape has minimal impact on performance; therefore, layout choices in GTK can be driven by user preference. Although other layouts exist, such as Ring menus found in VRMenuDesigner [12], GTK prioritises the most common designs.

2.3 Navigable Cards

To structure user progression, we introduce the **Navigable Card** module (Element (4) in Figure 1), which includes three card types:

- **Guide Cards:** Designed following the framework by Chauvergne et al. [7], these cards provide formation-specific contexts for active onboarding.
- **Quiz Cards:** These assess conceptual understanding using embedded Grid or Pie menus.
- **Questionnaire Cards:** These collect user feedback using standard 5, 7, or 9-point Likert scales to support common questionnaires, such as the System Usability Scale (SUS) [6], NASA-TLX [11], and Mental-effort rating scale by Paas [21].

To support guided learning and enforce skill acquisition, **Guide Cards** combine mixed-media assistance and partially enabled interactivity. They can present visual aids, such as embedded video demonstrations, alongside verbal instruction. To manage complexity, they use partially enabled interactivity and limit interaction to task-relevant objects (e.g., highlighted AOIs); this helps avoid unexpected behaviours and keeps the user's attention on the current step. Beyond highlighting, the cards can display an indicator that points to the nearest AOI when it is outside the user's field of view. AOI highlighting can also be animated using different effects, e.g., a blink effect, where object borders transition from white to yellow and back. If progress stalls, the system can trigger audio reminders (e.g., repeating the task at hand) to help the user continue.

2.4 Visual Attention Monitoring

The **Visual Attention Monitoring** module (Element (5) in Figure 1) records data on gaze and controller interactions. It exports CSV logs for analysis and displays real-time visualisations of attention sequences. Currently, the toolkit supports pie charts to show attention distribution and scarf plots to visualise the sequence of attention.

2.5 Calibration and Accuracy Tools

Finally, GTK includes tools for assessing eye-tracking quality (Element (6) in Figure 1). We implemented a configurable gaze-accuracy test based on Clemotte et al. [8], where targets appear sequentially for the user to fixate on. This process yields correspondences between known target positions $\mathbf{p}_i = (x_i, y_i)^\top$ and measured positions $\mathbf{m}_i = (u_i, v_i)^\top$. We then apply a linear transformation, following Padikal et al. [22], to map $\mathbf{m}_i$ to $\mathbf{p}_i$, generating corrected gaze points that minimise system error. This correction step complements the headsets' built-in calibration procedure by reducing the number of calibration attempts needed to achieve acceptable eye-tracking performance.

3 ITERATIVE AND DEVELOPER-CENTRIC MODULE DESIGN

GTK was established through an iterative design process involving internal evaluations and consultations with domain experts in robotics, education, and eye-tracking research involved in the MASTER-XR project [2, 15]. This evaluation process focused on assessing the toolkit's suitability for creating diverse research and training scenarios in XR. To verify the toolkit's capabilities, we utilised GTK to implement various research prototypes, validating its modular architecture in different contexts. For example, in [4], we used GTK to develop a multimodal-multisensor interaction interface in VR, investigating how users interact with deep learning models in virtual environments through a co-creation task; while in [16, 17], we used GTK as the basis for a controlled study of gaze-based menu navigation, using GTK's spatial menu components to implement and compare various layouts and selection modalities.

3.1 Planned Developments

Building on the feedback from industrial and academic partners, we identified requirements for the next iteration of the toolkit. In future development, we will prioritise the following developments:

1. A desktop application to support post-hoc analysis of attention and interaction telemetry.
2. Extending the visual attention monitoring module to capture additional ocular measures, such as pupil dilation. This would enable the inference of cognitive load and related mental states [19].
3. Implementing low-latency data streaming for gaze and controller inputs to enable real-time machine learning pipelines [1].

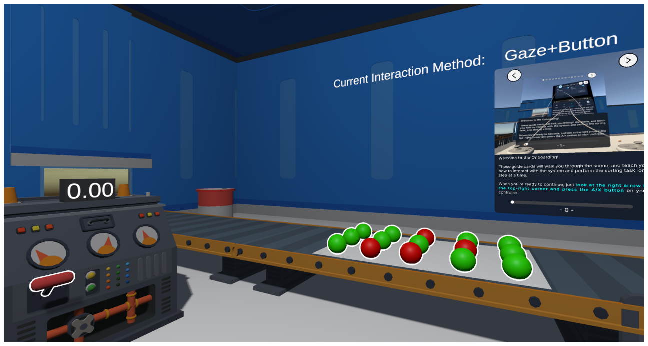

Figure 2: Scene layout of the VR Onboarding and Sorting Task. Guide cards (on the right) introduce the task and monitor progression. Trays of objects arrive from the left gate via the conveyor belt, and after completing the sorting, the participant activates the lever on the control station to start the next round. All interactable objects are highlighted with a white border to guide user attention.

4. Integrating speech as a complementary input modality to support contextual voice commands [4]. For example, the system could interpret spoken queries in relation to recent interactions (e.g., attended objects or controller pointing) or predefined AOIs.

5. Extending the Guide Card module to facilitate content creation via generative AI (e.g., automatic narration synthesis and image generation) and provide writing assistance for onboarding instructions.

6. Integrating GTK components into immersive authoring tools utilising visual scripting environments [9] or programming by demonstration approaches [10] to allow the creation of interactive gaze-enabled applications by non-technical content creators.

4 USER STUDY EVALUATION

GTK is a general-purpose toolkit; however, to assess the usability and efficiency of the reusable components for active gaze-based interaction in a realistic workflow, we designed a two-part VR user study. **Part 1** assesses the usability of GTK components during an onboarding procedure, including Guide Cards, Information Displays, and menu navigation. **Part 2** compares the efficiency of gaze-assisted and controller-only selection in a time-constrained industrial sorting task. In this section, we describe the revised user study design, including the tasks and procedures. We then report findings from an initial pilot study and explain how they informed revisions to the final study.

4.1 Study Design and Measures

We use a within-participant design. Both parts take place in the same scene layout (Figure 2) and use Guide Cards to present instructions, track progress, and collect questionnaires. Across tasks, we collect subjective measures (usability and perceived mental effort) and qualitative feedback. In Part 2, we also record performance measures, such as task time and errors, as well as interaction logs.

4.1.1 Part 1: VR Onboarding (Component Usability)

Part 1 introduces participants to the interaction techniques used in the study and evaluates three GTK components in the onboarding procedure. This part is designed to capture early usability feedback and identify issues before participants proceed to the time-critical sorting task. The three GTK components are:

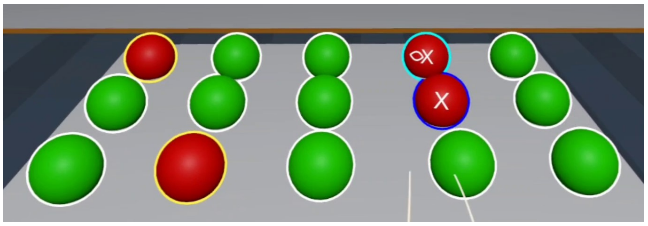

Figure 3: Sorting of defective (red) objects. When the user hovers over or selects an object, the highlight changes to provide visual feedback. During the onboarding phase, defective objects are additionally highlighted with a blinking yellow border.

- **Guide Cards:** Participants follow step-by-step instructions and navigate card content using either gaze-assisted or controller input. Guide Cards may restrict navigation to enforce task-based progression and can guide attention using highlighting and directional cues (e.g., arrows).
- **Information Displays:** Participants explore the scene and inspect several AOIs with attached displays, which appear when the user looks at an AOI and fade out when attention shifts.
- **Menu navigation (Quiz Cards):** Participants complete a short quiz using embedded Pie or Grid menus to rehearse key details for Part 2.

In this part, we collect three measures: (1) Mental effort (Paas) [21]: collected after each condition; (2) Usability (SUS) [6]: collected after each condition; and (3) Qualitative feedback: collected after completing all conditions through semi-structured interviews.

4.1.2 Part 2: VR Sorting Task (Selection Performance)

In Part 2, participants perform an industrial quality-control sorting task using two selection methods: multimodal gaze-assisted selection (gaze to point + button to confirm) and controller-only selection. Each condition includes a training phase and a timed test phase. In each round, a tray arrives with 15 objects arranged in a 5×3 grid, consisting of 11 correct (green) and 4 defective (red) items, as shown in Figure 3. Participants select all defective objects, and then activate a lever to advance to the next round. When an object is selected, its highlight changes to provide immediate visual feedback. Incorrect selections can be corrected by selecting the object again. Throughout the task, participants follow instructions provided by the Guide Cards. Upon entering the scene, the Guide Cards greet the participant, introduce the task, and specify the first interaction method to use; in addition, the Guide Cards control user progression and present questionnaires similar to Part 1.

In this part, similar to Part 1, we collect mental effort and usability scores after each condition, as well as qualitative feedback after completing both conditions. In addition, we measure workload (NASA-TLX [11]) and task performance using *Task completion time* (i.e., Total time to complete the test phase of each condition), and *Error rate* (i.e., Number of incorrect selections during the test phase of each condition).

4.2 Procedure

The procedure consists of five phases, as shown in Figure 4. First, we introduce the study, obtain consent, collect demographic information (including prior VR, eye tracking, and gaming experience), and set up the headset. Next, participants complete Part 1, which covers VR onboarding using Guide Cards, Information Displays, and a short quiz with menus. This is followed by a short break and a brief interview focused on the onboarding components. Participants then complete Part 2, i.e., the VR sorting task, under two

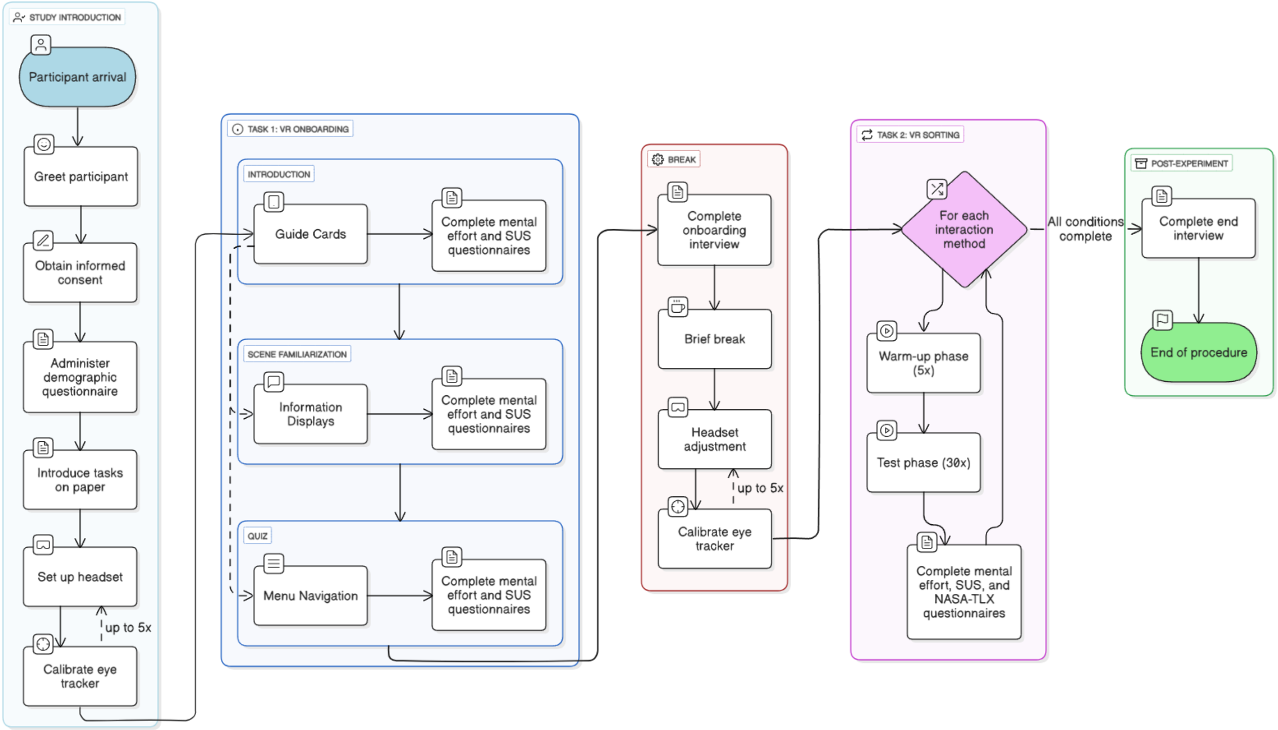

Figure 4: User study procedure for validating the active gaze-based interaction modules.

counterbalanced selection conditions. Finally, we conduct a post-study interview to gather preferences, perceived performance, and any issues that may have been encountered.

4.3 Pilot Study

We conducted two pilot studies to finalise the VR scenes and refine the study protocol. In the pilot studies, we initially used questionnaires instead of semi-structured interviews for qualitative feedback; this was revised based on our findings, as explained below.

4.3.1 First Pilot Study

We first recruited (N=4) eye-tracking and VR expert users. To address reliability concerns in the eye-tracking signal, we compared two headsets in the sorting-task scenario to assess user preferences and practical eye-tracking performance: the VIVE XR Elite[5] equipped with the VIVE Full Face Tracker[6] and the VIVE Focus Vision[7]. Both headsets provide automatic IPD (inter-pupillary distance) adjustment and report similar accuracy (0.5°–1.1°) within a 20° field of view. The XR Elite features diopter dials, whereas the Focus Vision offers a wider facial interface to accommodate larger eyeglasses.

We initially set the sorting task to ten rounds. Each participant completed the onboarding (i.e., Part 1) using one headset. Afterwards, they completed the sorting task (i.e., Part 2) using the same headset. After using the first headset, participants provided qualitative feedback using a post-study questionnaire. Then, the participants wore the second headset and repeated the sorting task (i.e., Part 2). The order of the headsets and the starting selection method conditions were alternated between participants. After using both headsets, we conducted a semi-structured interview to gather qualitative feedback regarding the comfort of the headset, the study content, and the interface.

Key Findings Two participants wore glasses during the study. Three participants reported blurry peripheral vision with the VIVE XR Elite, which may be related to the Full Face Tracker add-on. Comfort preferences were split: two participants (one with glasses and one without) preferred the VIVE XR Elite (mainly due to weight), while two preferred the VIVE Focus Vision (mainly due to overall comfort). During headset setup, both headsets required several calibration attempts to achieve acceptable eye-tracking accuracy (below 2°) and precision (below 0.5°). However, for one participant wearing glasses, we were unable to achieve acceptable eye-tracking accuracy with the VIVE XR Elite; this issue can be attributed to factors such as eye surgery, eye disease, heavy makeup, and high myopia, as noted in the VIVE Full Face Tracker specifications.

Based on these observations, we decided to use the VIVE Focus Vision headset for the remainder of the study, as it provided better visual clarity while maintaining similar comfort. We also revised our data collection to shorten the session and obtain more detailed qualitative feedback. Specifically, we replaced the post-study free-text questionnaire with a semi-structured interview, as some participants provided very brief responses and were reluctant to elaborate. An interview can help mitigate this issue and gather more meaningful insights. To limit setup time, we capped the number of calibration attempts during headset setup to five when acceptable tracking could not be achieved. Beyond these changes, we incorporated minor feedback (e.g., small wording changes in Guide Card instructions and UI adjustments), but no major redesigns were required.

Finally, we improved visual feedback for object selection: in addition to the border highlight, we added a separate selection indi-

---

[5] https://www.vive.com/eu/product/vive-xr-elite/overview/ (Accessed Jan 10 2026)

[6] https://www.vive.com/eu/accessory/vive-full-face-tracker/ (Accessed Jan 10 2026)

[7] https://www.vive.com/eu/product/vive-focus-vision/overview/ (Accessed Jan 10 2026)

cator overlay, i.e., an `X` symbol as shown in Figure 3, to help users distinguish *selected* objects during gaze-assisted input.

4.3.2 Second Pilot Study

In the first pilot study, we did not observe a decline in selection time or accuracy over repeated rounds. Instead, participants often improved by adapting their behaviour (e.g., using both hands, keeping one hand near the lever, stepping back to view the full scene, or stepping closer to improve gaze accuracy). Therefore, to further investigate this, we conducted a second pilot to identify when repeated rounds begin to degrade selection performance. We set the number of rounds to 50 and recruited (N=2) participants without prior eye-tracking or VR experience.

Key Findings We observed degraded selection performance after more than 20 repeated rounds. Therefore, we decided to set the number of rounds to 30 for the main studies. We also revised the acceptable eye-tracking accuracy threshold to 2.5°. Overall, based on both pilots, we made the following changes:

- **Hardware:** Use VIVE Focus Vision for the main study.
- **Data collection:** Use semi-structured interviews in place of some written free-text questionnaire prompts.
- **Setup constraints:** Cap calibration attempts during headset setup.
- **Task length:** Set the number of sorting rounds to 30.
- **Minor UI adjustments**: Revise instructions and clarify selection indicator.

4.4 Planned Main Studies and Hypotheses

In future work, we plan to conduct two within-subject user studies. We will use Study 1 to establish a baseline for the current GTK prototype, then Study 2 will incorporate feedback from Study 1 to assess whether the revised version improves on this baseline. To reduce learning effects (especially for the time-critical sorting task), we will recruit different participants for the two studies. Both studies will follow the setup as reported above. Based on our statistical analysis, we plan to recruit around 20 participants per study.

5 CONCLUSION

In this paper, we introduced GTK, a modular and extensible Unity toolkit for gaze-based interaction in XR. We described GTK's core modules, including gaze-aware interactables, spatial menus, navigable cards for onboarding and questionnaires, visual attention monitoring, and calibration and accuracy tools. We also reported how GTK has been used across internal demonstrations and research prototypes, and outlined planned developments informed by feedback from industrial and academic partners. Finally, we presented a two-part user study design to evaluate GTK's active gaze-based components in an onboarding workflow and an industrial sorting task, and reported pilot findings (N=6) that informed revisions to the study protocol. These revisions prepare the planned main studies (approximately 20 participants per study) for a more robust assessment of usability, workload, and task performance across gaze-assisted and controller-only interaction conditions.

ACKNOWLEDGMENTS

This work was funded by the European Union under grant number 101093079 (MASTER) and the Federal Ministry of Research, Technology and Space (BMFTR) under grant number 16IW23002 (No-IDLE), the Lower Saxony Ministry of Science and Culture (MWK) in zukunft.niedersachsen program, and the Endowed Chair of AAI at University of Oldenburg.

REFERENCES

[1] M. Barz, O. S. Bhatti, B. Lüers, A. Prange, and D. Sonntag. Multisensor-Pipeline: A Lightweight, Flexible, and Extensible Framework for Building Multimodal-Multisensor Interfaces. In *Companion Publication of the 2021 International Conference on Multimodal Interaction*, pp. 13–18. ACM, Montreal QC Canada, Oct. 2021. doi: 10.1145/3461615.3485432 2

[2] M. Barz, P. Karagiannis, J. Kildal, A. Rivera Pinto, J. R. de Munain, J. Rosel, M. Madarieta, K. Salagianni, P. Aivaliotis, S. Makris, and D. Sonntag. MASTER-XR: Mixed Reality Ecosystem for Teaching Robotics in Manufacturing. In *Integrated Systems: Data Driven Engineering*, pp. 167–182. Springer Nature Switzerland, Cham, 2024. doi: 10.1007/978-3-031-53652-6_10 2

[3] S.-J. Bittner, M. Barz, and D. Sonntag. Eliciting Multimodal Approaches for Machine Learning–Assisted Photobook Creation. In C. Ardito, S. Diniz Junqueira Barbosa, T. Conte, A. Freire, I. Gasparini, P. Palanque, and R. Prates, eds., *Human-Computer Interaction – INTERACT 2025*, pp. 576–601. Springer Nature Switzerland, Cham, 2026. doi: 10.1007/978-3-032-05002-1_31 1

[4] S.-J. Bittner, R. A. Leist, L. Kopácsi, O. S. Bhatti, A. Mohamed Selim, M. Barz, and D. Sonntag. Interactive Multimodal Photobook Co-Creation in Virtual Reality. In *Companion Proceedings of the 30th International Conference on Intelligent User Interfaces*, IUI '25 Companion, pp. 146–151. Association for Computing Machinery, New York, NY, USA, Mar. 2025. doi: 10.1145/3708557.3716355 2, 3

[5] Y. Bondarenko, V. Kuts, S. Pizzagalli, K. Nutonen, N. Murray, and E. O'Connell. Universal XR Framework Architecture Based on Open-Source XR Tools. In *XR and Metaverse*, pp. 87–98. Springer Nature Switzerland, Cham, 2024. doi: 10.1007/978-3-031-50559-1_7 1

[6] J. Brooke. SUS: A 'Quick and Dirty' Usability Scale. In *Usability Evaluation In Industry*, p. 6. CRC Press, 1st edition ed., 1996. doi: 10.1201/9781498710411-35 2, 3

[7] E. Chauvergne, M. Hachet, and A. Prouzeau. User Onboarding in Virtual Reality: An Investigation of Current Practices. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, pp. 1–15. Association for Computing Machinery, New York, NY, USA, Apr. 2023. doi: 10.1145/3544548.3581211 2

[8] A. Clemotte, M. Velasco, D. Torricelli, R. Raya, and R. Ceres. Accuracy and Precision of the Tobii X2-30 Eye-tracking under Non Ideal Conditions. In *Proceedings of the 2nd International Congress on Neurotechnology, Electronics and Informatics - NEUROTECHNIX*, pp. 111–116. SciTePress, 2014. Backup Publisher: INSTICC. doi: 10.5220/0005094201110116 2

[9] S. Eroglu, P. Schmitz, K. Sinke, D. Anders, T. W. Kuhlen, and B. Weyers. Choose Your Reference Frame Right: An Immersive Authoring Technique for Creating Reactive Behavior. In *Proceedings of the 30th ACM Symposium on Virtual Reality Software and Technology*, VRST '24, pp. 1–11. Association for Computing Machinery, New York, NY, USA, Oct. 2024. doi: 10.1145/3641825.3687744 3

[10] L. Gramoli, F. Nouviale, A. Reuzeau, A. Audinot, M. Risy, T. Marchand-Guerniou, M. Mavromatis, B. Arnaldi, and V. Gouranton. Xareus: a Framework to Create Interactive Applications without Coding. In *2025 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 1658–1659, Mar. 2025. doi: 10.1109/VRW66409.2025.00474 3

[11] S. G. Hart and L. E. Staveland. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In P. A. Hancock and N. Meshkati, eds., *Advances in Psychology*, vol. 52 of *Human Mental Workload*, pp. 139–183. North-Holland, Jan. 1988. doi: 10.1016/S0166-4115(08)62386-9 2, 3

[12] S. Hou, B. H. Thomas, and X. Lu. VRMenuDesigner: A toolkit for automatically generating and modifying VR menus. In *2021 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pp. 154–159, Nov. 2021. doi: 10.1109/AIVR52153.2021.00036 2

[13] S. Kapp, M. Barz, S. Mukhametov, D. Sonntag, and J. Kuhn. ARETT: Augmented Reality Eye Tracking Toolkit for Head Mounted Displays. *Sensors*, 21(6):2234, Jan. 2021. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute. doi: 10.3390/s21062234 1

[14] T. Kim, A. Ham, S. Ahn, and G. Lee. Lattice Menu: A Low-Error Gaze-Based Marking Menu Utilizing Target-Assisted Gaze Gestures on a Lattice of Visual Anchors. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, pp. 1–12. Association for Computing Machinery, New York, NY, USA, Apr. 2022. doi: 10.1145/3491102.3501977 2

[15] L. Kopácsi, P. Karagiannis, S. Makris, J. Kildal, A. Rivera-Pinto, J. R. de Munain, J. Rosel, M. Madarieta, N. Tseregkounis, K. Salagianni, P. Aivaliotis, M. Barz, and D. Sonntag. The MASTER XR Platform for Robotics Training in Manufacturing. In *Proceedings of the 30th ACM Symposium on Virtual Reality Software and Technology*, VRST '24, pp. 1–2. Association for Computing Machinery, New York, NY, USA, Oct. 2024. doi: 10.1145/3641825.3689514 2

[16] L. Kopácsi, A. Klimenko, M. Barz, and D. Sonntag. Exploring Gaze-Based Menu Navigation in Virtual Environments. In *Proceedings of the 2024 ACM Symposium on Spatial User Interaction*, SUI '24, pp. 1–2. Association for Computing Machinery, New York, NY, USA, Oct. 2024. doi: 10.1145/3677386.3688887 2

[17] L. Kopácsi, A. Klimenko, A. M. Selim, M. Barz, and D. Sonntag. Gaze-Based Menu Navigation in Virtual Reality: A Comparative Study of Layouts and Interaction Techniques. In *Human-Computer Interaction – INTERACT 2025*, pp. 520–543. Springer Nature Switzerland, Cham, 2026. doi: 10.1007/978-3-032-04999-5_31 2

[18] I. Lediaeva and J. LaViola. Evaluation of Body-Referenced Graphical Menus in Virtual Environments. pp. 9 pages, 3128. Canadian Human-Computer Communications Society / Société canadienne du dialogue humain-machine, 2020. Artwork Size: 9 pages, 3128 ISSN: 0713-5424 Medium: application/pdf. doi: 10.20380/GI2020.31 2

[19] A. Mohamed Selim, M. Barz, O. S. Bhatti, H. M. T. Alam, and D. Sonntag. A review of machine learning in scanpath analysis for passive gaze-based interaction. *Frontiers in Artificial Intelligence*, 7:1391745, June 2024. doi: 10.3389/frai.2024.1391745 2

[20] P. Monteiro, H. Coelho, G. Gonçalves, M. Melo, and M. Bessa. Comparison of Radial and Panel Menus in Virtual Reality. *IEEE Access*, 7:1–1, Aug. 2019. doi: 10.1109/ACCESS.2019.2933055 2

[21] F. G. W. C. Paas. Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive-load approach. *Journal of Educational Psychology*, 84(4):429–434, 1992. Place: US Publisher: American Psychological Association. doi: 10.1037/0022-0663.84.4.429 2, 3

[22] V. Padikal, A. Plonkowski, P. F. Lawton, L. K. Young, and J. C. A. Read. Gaze Error Estimation and Linear Transformation to Improve Accuracy of Video-Based Eye Trackers. *Vision*, 9(2):29, June 2025. Publisher: Multidisciplinary Digital Publishing Institute. doi: 10.3390/vision9020029 2

[23] R. Rivu, Y. Abdrabou, K. Pfeuffer, A. Esteves, S. Meitner, and F. Alt. StARe: Gaze-Assisted Face-to-Face Communication in Augmented Reality. In *ACM Symposium on Eye Tracking Research and Applications*, pp. 1–5. ACM, Stuttgart Germany, June 2020. doi: 10.1145/3379157.3388930 2

[24] T. Wagner, M. Colley, D. Breckel, M. Kösel, and E. Rukzio. UnitEye: Introducing a User-Friendly Plugin to Democratize Eye Tracking Technology in Unity Environments. In *Proceedings of Mensch und Computer 2024*, MuC '24, pp. 1–10. Association for Computing Machinery, New York, NY, USA, Sept. 2024. doi: 10.1145/3670653.3670655 1