Universidad de los Andes

Faculty of Engineering
Undergraduate Program in Systems Engineering
Graduation Project
Period 2024-20

EyeNav: A Novel Accessibility-Driven System for Interaction and Automated Test Generation Using Eyetracking and Natural Language Processing

Author: Juan Diego Yepes-Parra
Advisor: Camilo Andrés Escobar-Velásquez

November of 2024
Bogotá, Colombia

Contents

1 Abstract

EyeNav is a novel system that combines eye tracking and natural language processing (NLP) to enhance accessibility and enable automated test generation. This research demonstrates the integration of these technologies for intuitive web interaction, enabling pointer control via gaze and natural language processing for interpreting user intentions. Additionally, it presents a record-and-replay module for generating automated test scripts. Preliminary user evaluations yielded positive results in terms of usability. The ultimate goal is to demonstrate that eye tracking combined with NLP can be effectively used not only as a possible assistive technology but also as an innovative approach to software testing.

Keywords: Eye-tracking, Automated Test Generation, Assistive Technology, Natural Language Processing, Web Applications, Accessibility.

2 General Objective

The primary objective of this project is to leverage eye tracking combined with natural language processing technologies as user input methods for interacting with computers. By combining eye tracking for pointer control and natural language processing for specifying commands, the project seeks to create assistive technology that facilitates computer use. Additionally, the project aims to explore the potential of this interaction model for automating the generation of test scripts in a record-and-replay manner for web applications.

3 Specific Objectives

1. To develop an **eyetracking** system that can be used as an alternative input method for computer interaction.
2. To integrate **natural language processing** capabilities for interpreting user commands and execute the corresponding actions.
3. To develop a framework for **automated test generation** for web applications, in a record-and-replay style.
4. To evaluate the effectiveness of the developed system in terms of **accessibility**, **usability** and accuracy in generating **test scripts**.

4 Literature Review

4.1 Eye Tracking

Eye Tracking refers to "the use of proper techniques and tools for the identification of a subject's gaze direction; in other words, eye tracking allows detecting and recording what users watch, typically, but not necessarily, on a screen" (Cantoni & Porta, 2014). This technique can be measured and applied in different disciplines, which will be explained in the following sections.

Dondi & Porta (2023) point out that there are mainly two different kinds of Eye Trackers:

1. *Remote*, the selected type for the project, which are non-intrusive devices positioned at the bottom of the display.
2. *Wearable*, that are installed in a pair of glasses, for instance, and are able to track outside of digital environments.

4.1.1 Gaze

Gaze is usually associated with a fixed or steady look, however in this field it is used to describe the approximate point on screen where the subject is fixating their eyes. Cantoni & Porta (2014) have explained that to obtain this information in Eye Trackers, gaze direction is obtained during an initial calibration phase, which determines the correspondences between the points observed by the subject on the screen, and the positions of the cornea.

4.1.2 Accuracy and Precision

Accuracy and precision have various applications, like in data science for example, but can be used in this context to understand how the Eye Tracker is performing. Following the manufacturer definitions, accuracy refers to the discrepancy between the true gaze position and the location recorded by the eye tracker, reflecting the systematic error or spatial accuracy of the data. On the other hand, precision indicates the consistency of the eye tracker in capturing the same gaze point across successive samples. It is quantified using the Root Mean Square (RMS) of the recorded points to assess the variation in the data. (Tobii-AB, 2023)

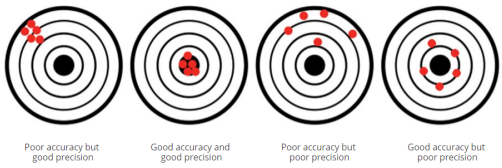


Figure 1: Accuracy and Precision. Source: Tobii-AB (2023)

4.1.3 Applications

- Usability testing

  In this field, eye tracking allows for the analysis of the behavior of the user. This is a process that occurs after they have used the product, which means it does not take an active role in the interaction per se. (Jacob & Karn, 2003)

- Psychology experiments

Many different psychology experiments can be done using eye tracking, for instance, the cognitive load students using a Learning Management System (LMS) has been analyzed using eye tracking, and how it had an effect on their learning outcomes. (Sáiz-Manzanares et al., 2024)

- Input method

  As Jacob & Karn (2003) point out, the application of eye trackers in HCI (Human-Computer Interaction) has long been studied and researched. The approach of using eye tracking as an input method is yet to be applied massively in a commercial environment.

- Artificial Intelligence

  In the rising field of Generative Artificial Intelligence, Taieb-Maimon et al. (2024) used eye tracking to perform text summarization, mining data from attention metrics, which outperforms regular text summarization.

4.2 Natural Language Processing

Natural Language Processing (NLP) is a field of artificial intelligence that focuses on the interaction between computers and humans through natural language. The goal of NLP is to enable computers to understand, interpret, and generate human language in a way that is both meaningful and useful. As Kochmar (2022) describes it, "[Natural Language Processing] addresses the various ways in computers can deal with natural—that is human—language"

NLP encompasses several core tasks. These techniques provide the foundation for building more complex systems capable of handling a wide range of applications. For instance, here are some that Kochmar (2022) describes:

- Tokenization

  Breaking text into smaller units such as words, sentences, or subwords to facilitate analysis.

- Syntactic Parsing

  Analyzing sentence structure to identify grammatical relationships between words.

- Semantic Analysis

  Understanding the meaning of text by interpreting context, word sense, and relationships.

- Sentiment Analysis

  Determining the emotional tone or attitude expressed in text.

4.2.1 Applications

- Assistive Technologies

  Assistive tools such as screen readers, language translation systems, and real-time transcription services leverage NLP to improve accessibility for individuals with disabilities. For instance, Girón Bastidas et al. (2019) used it to create a hearing aid for students with disabilities, to carry their academic tasks with normalcy.

- Smart Voice Assistants

  Smart assistants like Siri and others are powered by natural language processing, enabling them to understand voice commands, process natural language queries, and generate spoken responses (Terzopoulos & Satratzemi, 2020).

4.3 Record And Replay Testing

A record and replay tool for web applications operates by capturing user interactions, including mouse clicks, keyboard inputs, and navigation commands, during the application's usage. These recorded actions are then replayed by reapplying the captured inputs to the browser engine during playback. (Hammoudi et al., 2016)

4.3.1 Gherkin Syntax

All of the testing features developed were possible thanks to the Kraken framework. These tests are made using the Gherkin syntax and WebdriverIO, which enables controlling the browser for automated testing scenarios. (Ravelo-Méndez et al., 2023)

Gherkin syntax is a language that allows for instructions to be written in natural language. The language can bridge the gap between technical and non-technical team members, promoting a unified approach to defining requirements (Cucumber-Docs, 2023). It employs a set of keywords such as `Given`, `When` and `Then` that each outline preconditions, actions and expected outcomes respectively.

5 Related Work

5.1 Eyetracking in a chrome extension for behavioral analysis

Zelinskyi & Boyko (2024) developed a Chrome Extension that leverages eye tracking combined with session recording (also in a record and replay matter) to be able to use for behavior analysis. Their goal was to create a tool that could make behavioral analysis that combine eye tracking and session recording studies easier.

Their extension showed to be highly usable and successful, nevertheless their eye tracking was not as accurate which impacted the usability overall.

This study is interesting in this context because it handles eye tracking combined with a chrome extension and record and replay style sessions, which is very similar to the solution developed.

5.2 Eyetracking as an Input Method

Recent advancements in consumer-grade AR and VR headsets, such as the Meta Quest Pro, Pico 4 Pro, and Apple Vision Pro, have significantly increased the use of eye tracking technologies Huang et al. (2024). Notably, some of these devices, like the Apple Vision Pro, eliminate the need for physical peripherals, relying solely on hand gestures and eye tracking for control.

This study is relevant as it aims to explore the use of eye tracking and voice controls, similar to the

aforementioned technologies. While accuracy is crucial for gaze-based interactions, Huang et al. (2024) emphasize that it is not the only factor influencing user experience.

Additional considerations for this novel input method include the natural tendencies of users. Fernandes et al. (2023) found that users do not typically fixate on a target before selecting it. Instead, they quickly shift their gaze from one point of interest to another, complicating the estimation of gaze events.

Furthermore, navigating the web using eye tracking technologies, such as those in the Apple Vision Pro, requires web design adaptations. Apple (2024) highlight that larger, rounder elements are easier to interact with compared to smaller clickable elements like hyperlinks. This insight underscores the need for web interfaces to accommodate virtual reality environments.

5.3 The future of the web

Panwar (2024) discusses the evolution of web applications from their static origins to their current dynamic and complex state.

The study also provides a forward-looking perspective on the future of the web, particularly in the context of AR/VR. A key point highlighted is the necessity for future web applications to support multi-modality, as "interfaces that combine touch, voice, text, and gesture are expected to rise" (Panwar, 2024).

Understanding the future of the web is crucial for this study, as it provides insights into usability and potential challenges arising from new interaction methods. This knowledge is invaluable for developers and companies aiming to cater to users in these evolving environments.

6 Method

In this section, the tools and the implemented system are specified.

6.1 Peripherals

The external hardware required for developing the project will be described in detail.

6.1.1 Tobii Pro Nano Eye Tracker

Figure 2: Tobii Pro Nano Eye Tracker. Source: (Tobii-AB, n.d.)

The Tobii Pro Nano Eye Tracker, shown in figure 2, uses a corneal reflection, dark and bright pupil combination, one-camera system. It supports screen sizes up to 19 inches, introduces a 17 ms system latency. (Tobii-AB, n.d.)

It is characterized by its lightweight and compact form; which makes it easy to carry, and ultimately enabled an easier usability testing process. On the other hand, Tobii Eye Trackers are compatible with the `tobii-research` SDK for python, which allows for controlling the device without the need for a graphical user interface.

This device prompted the idea for the project.

6.1.2 Microphone

The microphone serves as a mediator of interaction, enabling the capture of verbal cues essential for the analysis of words that then get turned into commands thanks to the natural language processing. It also serves as an way of qualitative data acquisition, for recording each participant response for usability testing.

6.2 System Design

Given the system requirements and all of the peripherals it has to handle, a software architecture design was needed, in order to satisfy all the functional and non-functional requirements.

6.2.1 Software Architecture

The software components are shown in figure 3. The important takeaway here is the understanding of which component communicates with each other, and how this comunication is happening (protocol, procedure, etc) and in which environment. This also showcases how the architecture of the system works in a fullstack-like matter, where the frontend (chrome extension) communicates with the backend, and the additional testing module.

Figure 4 displays a sequence diagram where a user starts a simple session, in which they click an element on screen and scroll down. The purpose of this illustration is to showcase how the system behaves and which component, from the previous component diagram, is responsible for handling each instructions.

This system works through the power of multithreading and concurrency, therefore, showcasing when each process is being called and by who helps to illustrate the functionality better.

6.3 Implementation

A prototype for this idea was developed, which lets the user control the web browser using their gaze visualization and voice for command instructions. The prototype is started using the frontend, and the backend needs to be running prior to said initialization.

All of the source code for this project is hosted in a GitHub repository.[1] The repository holds the code for this documentation, the backend module, the chrome extension and the testing module.

6.3.1 Chrome Extension

In Chrome versions 114 and later, it is possible to develop a sidepanel, which effectively gives the ability to create a secondary and completely independent webpage (or at least from the primary webpage)

[1] `https://github.com/juanyepesp/EyeNav`

that is always visible to the user. This is an awesome capability since it allows for the system to display important information relevant to the user. For instance, the voice command that the voice recognition module detected, or the different commands that can be used and how, etc.

Developing a chrome extension, or at least one with a sidepanel, is effectively the same as doing a webpage. The only extra step to creating one is to use a `manifest.json` that tells the browser the code will be packaged into an extension. The rest is plain javascript for functionality, HTML for structure and CSS for styles.

There are also some tricky parts that needed to be adressed. Since the system needs to register which HTML tag the user clicks in a given moment, the extension needs to handle each click using an event listener. However, this needs to be implemented carefully for dynamically created elements, since adding an event listener on top can eliminate certain functionalities of the webpage.

6.3.2 Backend

The backend needed to include several components in order to achieve all of the functional requirements. Following the diagram detailed in figure 3, here the implementation of each one will be described.

As a disclaimer, every part of this system is running locally on a single computer, ensuring data privacy, minimal latency and overall a strong performance.

- `API`

  Implemented in the `main.py` file, this component is the primary way of talking with the backend and exposes the messages necessary for starting and stopping a session, and registering each click tag.

- `commandQueue`

  The command queue, is the tunnel of information for each voice command that the voice control module recognizes. It uses websockets instead of HTTP for ensuring a fast and streamlined command display; ensuring minimal latency perceived by the user in the chrome extension.

- `loggerModule`

  Similarly, this module also uses a queue for handling instruction writing on the testing files, called features. Each feature is created once the session is started, and needs to be handled on a queue for impeding data races.

- `eyeTrackingModule`

  The eyetracking module uses the `tobii-research` module part of the tobii pro sdk for Python. It enables for the programatic control of the eyetracker whilst smoothly moving the cursor to each new position it detects.

- `voiceControlModule`

  The voice control module was developed using `Vosk`, an open sourced framework for loading language modules locally. `Vosk` allows for downloading different models based on language. This module is also responsible for translating each correct voice command into an actual computer command, while adding to the queue said commands for the logging module to process.

6.3.3 Testing

The testing module is much more straightforward and concrete, primarily due to the Kraken module handling most of the responsibilities. The only necessary files that had to be created were the step definition file and each feature (those are the tests to be executed). The step definitions are written in Gherkin syntax and include each action the user took; they need to be defined for the webdriver to execute them. On the other hand, the features were created previously by the logger module.

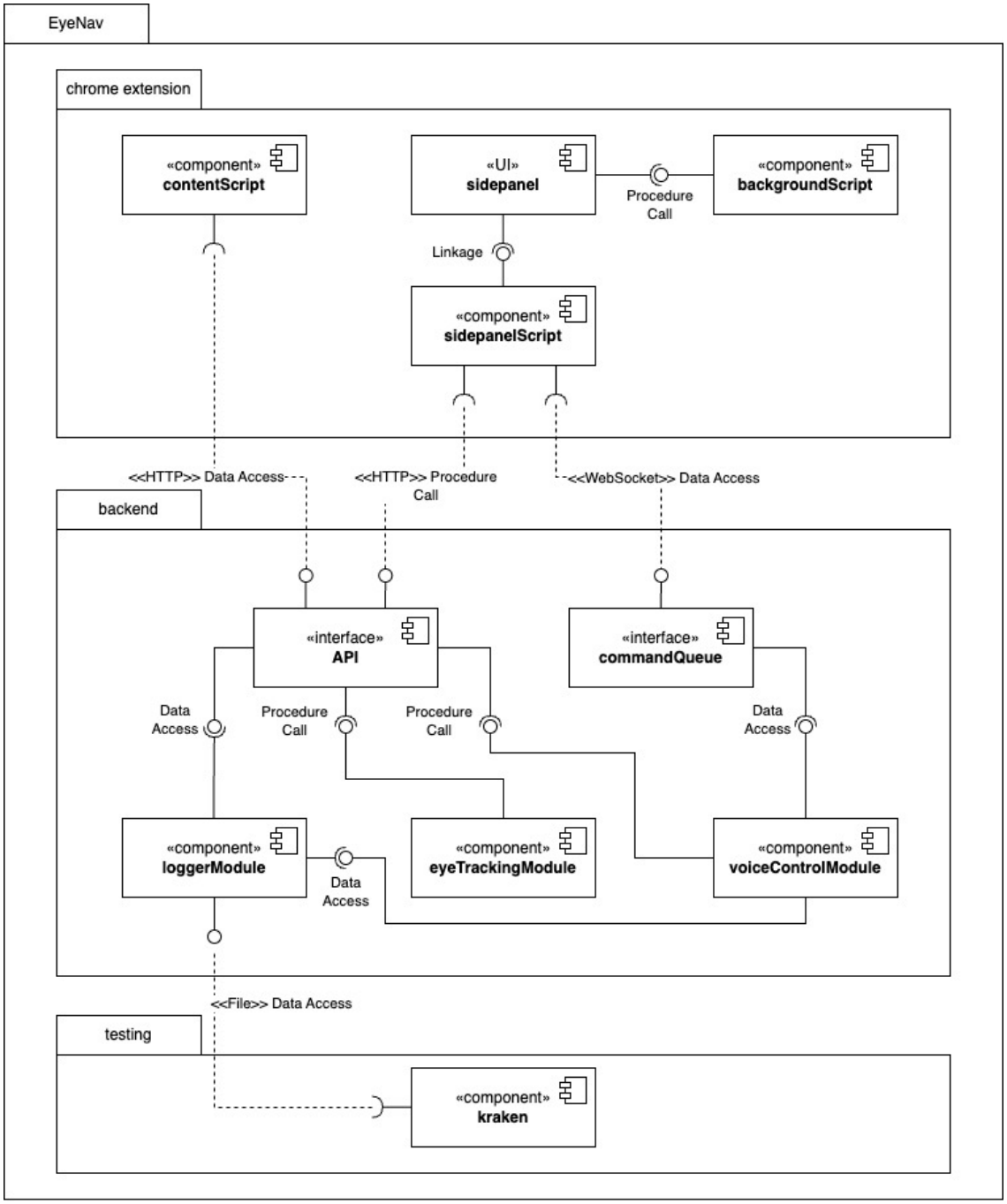


Figure 3: Architecture described in components

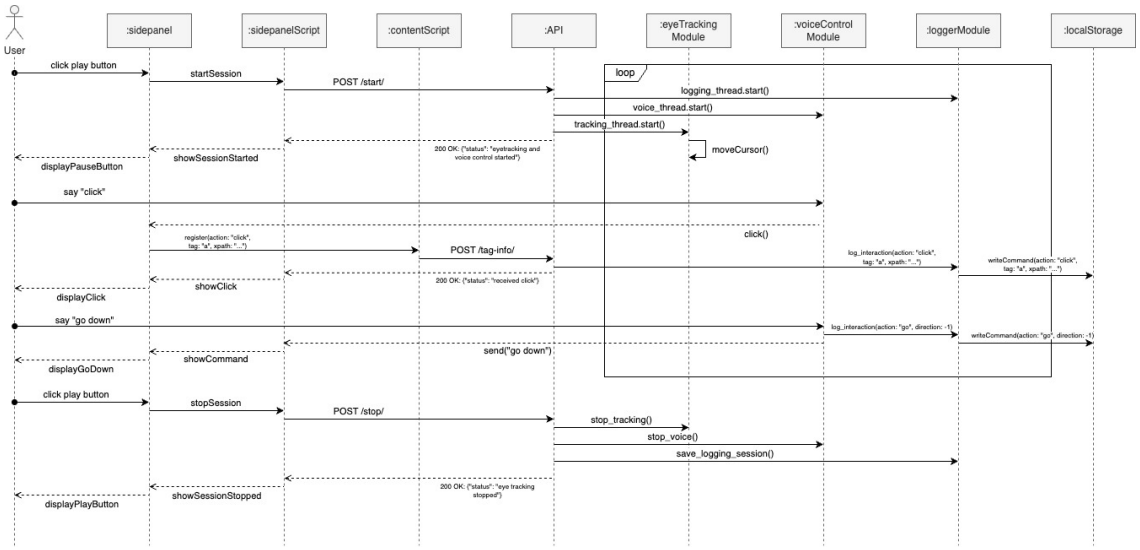


Figure 4: An user clicking and scrolling sequence

7 Results

7.1 System

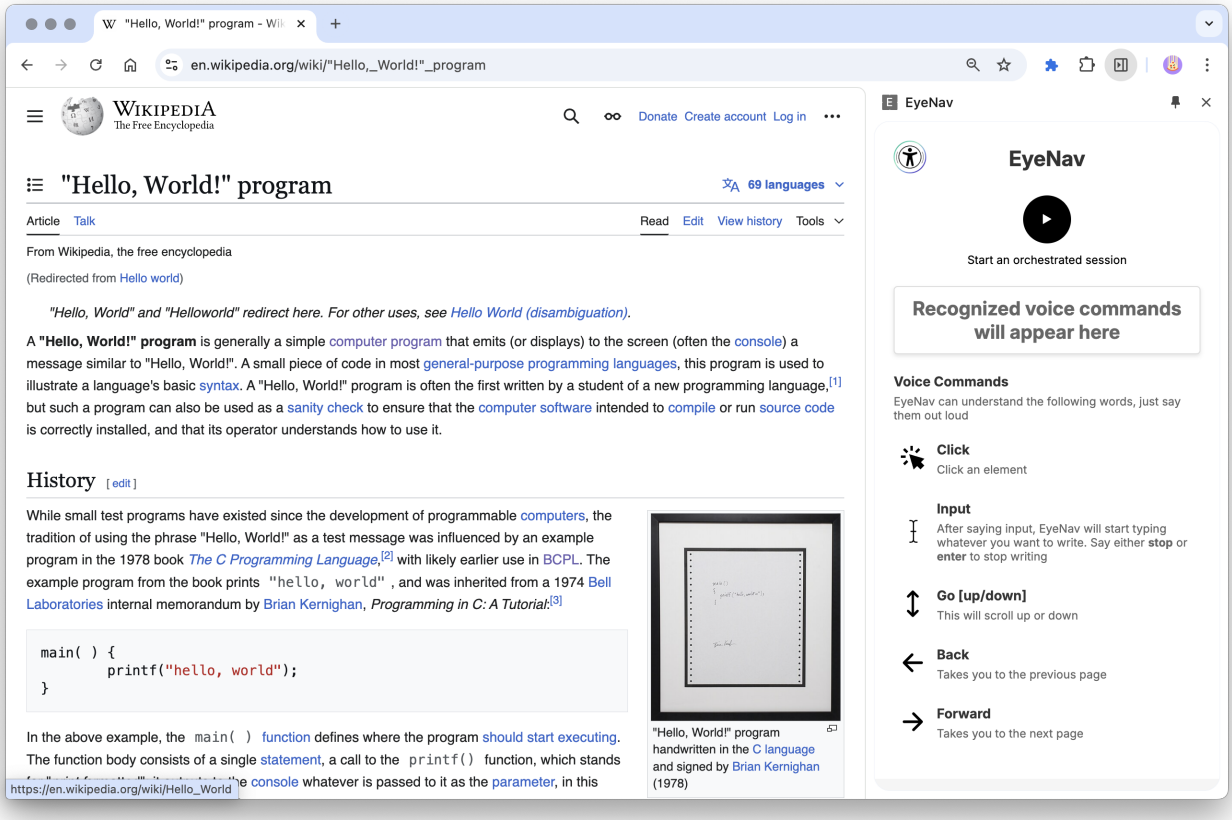
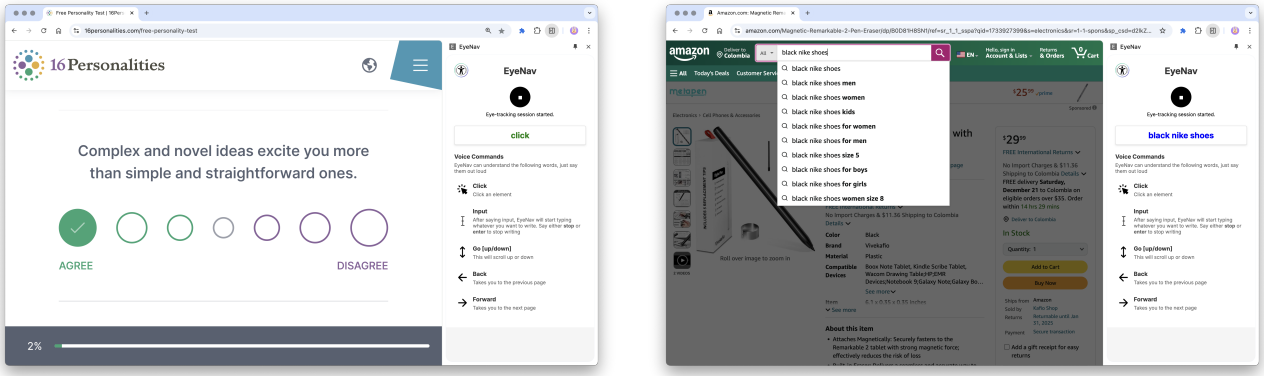


Figure 5: Chrome extension

Figures 5 and 6 illustrate the system in operation. The left side of the browser shows the webpage as it would appear in any web browser, while the right side features the side panel of the Chrome extension. This side panel must remain visible at all times for proper orchestration. Below the play button, the voice commands box displays recognized commands, with control words in green and input (typing mode) words in blue. At the bottom, a menu lists all available commands.



(a) Clicking an item

(b) Inputting text

Figure 6: Other actions that can be done

7.2 Usability

Usability in itself is not an absolute metric, and needs to be defined in particular contexts where it is being applied. In that sense, ways of measuring this non-functional requirement are not straightforward and do not depend on a single method, but are rather cualitative data that need to be analysed in this way.

7.2.1 System Usabillity Scale

The system usability scale (SUS) is useful for understanding broad and general measurements of usability. Proposed by Brooke (1996), this scale is widely used in systems engineering, and consists of 10 questions that the user will assess subjectively, but ultimately can point out how the system performed in three points:

- effectiveness (the ability of users to complete tasks using the system, and the quality of the output of those tasks)
- efficiency (the level of resource consumed in performing tasks)
- satisfaction (users subjective reactions to using the system). (Brooke, 1996)

These are the statements proposed in the scale, where the subject can mark from 1 (strongly disagree) to 5 (strongly agree):

1.	I would like to use this system frequently.	1 2 3 4 5
2.	I found the system unnecessarily complex.	1 2 3 4 5
3.	I thought the system was easy to use.	1 2 3 4 5
4.	I think I would need technical support to use this system.	1 2 3 4 5
5.	The functions of the system were well integrated.	1 2 3 4 5
6.	There was too much inconsistency in the system.	1 2 3 4 5
7.	Most people would learn to use this system very quickly.	1 2 3 4 5
8.	I found the system very cumbersome to use.	1 2 3 4 5
9.	I felt very confident using the system.	1 2 3 4 5
10.	I needed to learn a lot of things before I could get going with this system.	1 2 3 4 5

SUS have scores ranging from 0 to 100. Each individual score can be calculated like formula 1 shows, where each s_n represents each statement

$$2.5\left(20\sum(s_1, s_3, s_5, s_7, s_9) - \sum(s_2, s_4, s_6, s_8, s_{10})\right) \tag{1}$$

Furthermore, in the usability tests conducted, the average SUS was 76.7, with the highest score being 95 and the lowest being 55. Indicating that the system was found to be quite usable.

7.2.2 User Feedback

Another method to evaluate the system's usability was through user interviews. Key insights from these interviews include:

- Scenario-specific usability: The navigability and ease of use of the system are directly related to the specific scenario or webpage being interacted with. Users found it easier to use when performing straightforward tasks like purchasing a product or reading news, but more challenging when completing complex actions such as filling out forms.

- Icon size: The ease of reaching icons using eye-tracking is highly dependent on their size. Larger icons are generally easier to navigate to.
- Environmental conditions: These significantly impact the system's usability, especially due to its reliance on voice recognition. In noisy environments, users found the system harder and less enjoyable to use.
- Interaction accuracy: Users expressed a desire for a feature that indicates the exact element the system is about to click, which could enhance interaction accuracy.
- User comfort: Most users felt comfortable using their voice and eyesight for interaction, describing this method as intuitive and user-friendly.
- Response time: The response time for each action was deemed satisfactory by all users, indicating no significant delays.
- Voice control: Voice control seemed to be easier whenever control words were clearer and shorter. For instance, when the first prototype was developed, the user had to say in units how much they wanted to scroll, like "scroll down five units". This became a hassle and was ultimately decided that scrolling would be done in an arbitrary number of units that is intuitive.

7.3 Accessibility

7.3.1 User Feedback

- According to WebAIM (2024), 95.9% of webpages have at least one detectable accessibility error. Accessibility is an attribute of the webpage itself, so many tester users felt that this system indeed leverages a new way of interacting with webpages. However, to become more accessible, it would need to align with the actual design of the webpage, such as making icons easier to reach and text more readable.
- Due to the scope and time limitations of the project, testing with individuals with motor impairments was not possible. However, all users who tested the system suggested that this method of interaction could be beneficial for individuals with motor impairments. They even mentioned that if the system were more polished, they would consider using it regularly to be more productive.

7.4 Test Script Generation

The system's tests are a series of actions recorded during user interactions. These actions are documented in Gherkin syntax, which are then interpreted and mapped to corresponding Webdriver actions that the testing module can replay. Below is an example of a test script:

```
1    Feature: Replay of session on Nov 11 at 02:48:38 PM
2
3    @user1 @web
4    Scenario: User interacts with the web page named "Amazon.com. Spend less. Smile more."
5
6        Given I navigate to page "https://www.amazon.com/"
7        And I click on tag with id "twotabsearchtextbox"
8        And I input "nike black shoes"
9        And I click on tag with id "nav-search-submit-button"
10       And I scroll down
11       And I click on tag with xpath "/html[1]/body[1]/div[1]/div[1]/div[1]/div[1]/div[1]/span
     [1]/div[1]/div[9]/div[1]/div[1]/span[1]/div[1]/div[1]/div[1]/span[1]/a[1]/div[1]"
```

These actions are replayable thanks to the Kraken module. However, as webpages dynamically change and evolve, these tests have a limited lifespan. Despite this, they can be valuable for specific testing scenarios. For example, they can help determine how a webpage behaves when its size changes, how many clicks are required to perform the same action, and whether the flow of actions breaks if the webpage layout changes.

8 Discussion

Overall, the results indicate success in achieving the research objectives. The integration of the eye tracking system with NLP capabilities proved to be an intuitive and responsive method for computer interaction, as demonstrated by user testing.

However, in terms of usability, the system still requires adjustments to better accommodate diverse user bases. This includes improving accessibility for individuals who wear glasses and addressing challenges faced by users speaking different languages or accents, as these groups reported higher levels of frustration when completing tasks.

On the other hand, the testing module provides a flexible and accessible solution for conducting usability tests on any webpage. Its ease of use, accuracy generating the test and adaptability make it a valuable tool for assessing and improving web interfaces in a dynamic environment.

Finally, this study highlights emerging challenges in web interaction brought about by advancements in AR and VR technologies. As these technologies become more prevalent, it is increasingly important to understand how eye and voice interactions can be effectively utilized in web environments. This research serves as a foundational approach to exploring the usability of such interactions, resolving for future studies to build upon these findings and further enhance user experience in immersive web contexts.

9 Future Work

Considering the scope and impact of the project, the following enhancements and additions can be made to improve the system's accessibility. Furthermore, this study has paved the way for new research areas where the discovered concepts can be applied.

- To improve eye tracking for individuals who wear prescription eyeglasses by using their prescription formula to adjust the system for optimal performance.
- To expand the system's language support and implement internationalization (i18n) features.
- To extend the system capabilities to other environments, such as mobile platforms.
- To include a walkthrough page on how to use the system.

- To incorporate individuals with motor disabilities in future usability testing sessions, as the system has shown potential to be a beneficial solution based on the system usability scale scores and interview feedback.
- To include visual cues for the eyetracking system, to make the system more intuitive
- To leverage record-and-replay test generation as a standalone extension, that can be used with these or other interaction methods.
- To explore the use of these technologies for testing virtual environments, such as those enabled by Meta Quest or Apple Vision Pro mixed reality glasses. This could involve evaluating the responsiveness and usability of these systems, ensuring accessibility features can accommodate a diverse range of users.

10 Conclusions

This study has demonstrated the successful integration of eye-tracking and NLP to create an accessible and intuitive user interaction system. The EyeNav system provides a compelling system, facilitating automated test generation. Despite limitations in accommodating diverse user conditions, the results underscore its potential for broad applications across various domains, such as Human-Computer Interaction, record and replay testing, NLP for voice commands, etc. Future research should aim to expand its usability among underrepresented user groups and explore applications in virtual and augmented reality, since eyetracking and voice controls are also being used in these areas.

References

Apple. (2024). *Wwdc24: Optimize for the spatial web.* Retrieved from https://www.youtube.com/watch?v=5tjPBF2qoY4

Brooke, J. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, *189*(194), 4–7.

Cantoni, V., & Porta, M. (2014). Eye tracking as a computer input and interaction method. In *Proceedings of the 15th international conference on computer systems and technologies* (pp. 1–12).

Cucumber-Docs. (2023). *Gherkin reference documentation.* Retrieved from https://cucumber.io/docs/gherkin/

Dondi, P., & Porta, M. (2023). Gaze-based human–computer interaction for museums and exhibitions: technologies, applications and future perspectives. *Electronics*, *12*(14), 3064.

Fernandes, A. S., Murdison, T. S., & Proulx, M. J. (2023). Leveling the playing field: A comparative reevaluation of unmodified eye tracking as an input and interaction modality for vr. *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS*, *29*.

Girón Bastidas, J. P., Salcedo Parra, O. J., & Espitia R, M. J. (2019). Natural language processing services in assistive technology. *Juan Pablo Girón Bastidas, Octavio José Salcedo Parra and Miguel J. Espitia R., Natural Language Processing Services in Assistive Technology. International Journal of Mechanical Engineering and Technology*, *10*(7).

Hammoudi, M., Rothermel, G., & Tonella, P. (2016). Why do record/replay tests of web applications break? In *2016 ieee international conference on software testing, verification and validation (icst)* (pp. 180–190).

Huang, Z., Zhu, G., Duan, X., Wang, R., Li, Y., Zhang, S., & Wang, Z. (2024). Measuring eye-tracking accuracy and its impact on usability in apple vision pro. *arXiv preprint arXiv:2406.00255*.

Jacob, R. J., & Karn, K. S. (2003). Commentary on section 4. eye tracking in human-computer interaction and usability research: Ready to deliver the promises. *The mind's eye*, *2*(3), 573–605.

Kochmar, E. (2022). *Getting started with natural language processing*. Manning.

Panwar, V. (2024). Web evolution to revolution: Navigating the future of web application development. *International Journal of Computer Trends and Technology*, *72*.

Ravelo-Méndez, W., Escobar-Velásquez, C., & Linares-Vásquez, M. (2023). Kraken 2.0: A platform-agnostic and cross-device interaction testing tool. *Science of Computer Programming*, *225*, 102897.

Sáiz-Manzanares, M. C., Marticorena-Sánchez, R., Martin Anton, L. J., González-Díez, I., & Carbonero Martín, M. Á. (2024). Using eye tracking technology to analyse cognitive load in multichannel activities in university students. *International Journal of Human–Computer Interaction*, *40*(12), 3263–3281.

Taieb-Maimon, M., Romanovski-Chernik, A., Last, M., Litvak, M., & Elhadad, M. (2024). Mining eye-tracking data for text summarization. *International Journal of Human–Computer Interaction*, *40*(17), 4887–4905.

Terzopoulos, G., & Satratzemi, M. (2020). Voice assistants and smart speakers in everyday life and in education. *Informatics in Education*, *19*(3).

Tobii-AB. (n.d.). Tobii pro nano: Enter the world of eye tracking research [Computer software manual]. Stockholm, Sweden. Retrieved from `https://tobiipro.com`

Tobii-AB. (2023). *Eye tracker accuracy and precision.* Retrieved from `https://connect.tobii.com/s/article/eye-tracker-accuracy-and-precision`

WebAIM. (2024). *The techreport:webaim-2024 million: The 2024 report on the accessibility of the top 1,000,000 home pages.* (Tech. Rep.). techreport:webaim-2024. Retrieved from `https://techreport:webaim-2024.org/projects/million/`

Zelinskyi, S., & Boyko, Y. (2024). Integrating session recording and eye-tracking: development and evaluation of a chrome extension for user behavior analysis. *Radioelectronic and Computer Systems*, *2024*(3), 38–54.