Eye Tracker – A Reading Assistant

1st Author

1st author's affiliation  
1st line of address  
2nd line of address  
Telephone number, incl. country code

1st author's E-mail address

2nd Author

2nd author's affiliation  
1st line of address  
2nd line of address  
Telephone number, incl. country code

2nd E-mail

3rd Author

3rd author's affiliation  
1st line of address  
2nd line of address  
Telephone number, incl. country code

3rd E-mail

**ABSTRACT**

The use of Eye tracker in everyday life is still in development stages. In recent years, it is being used in gaming industry. This paper focuses the use of Eye-Tracker to recognize the word and reproduce the sound of the word. All the discussed experiments are done using Tobii eye tracker.

**Keywords:**

*Eye tracker, gaming, word, sound, Tobii*

# INTRODUCTION

Eye Tracking is a technology that allows computers to understand what gamers are looking at – using an eye tracker. An eye tracker consists of illuminators and cameras, and with it, you now can complement keyboards, mice, and gamepads as an input method for a new game experience. This paper is intended for audience who would like to explore the benefits of eye tracker and inherit possible potential application ideas.

We are using Tobii eye tracker for our experiments. There are several challenges associated with it, like, we must connect the tracker to USB 3.0 port only. We always need to have Tobii software to turn on /use the eye tracker, eye tracker should be mounted, very limited development support etc.

Given the fact that eye tracker is majorly used in gaming industry, currently eye tracker is used in games like Assassin creed, watch dogs 2 and other high level games. Our goal was to inherit the other use of eyes tracker and create an application which can be of great help to people with speech disability.

We have developed a technique to convert eye tracking into a precision text detection machine. However, we need to consider these technical challenges like how to detect eyes continuously and constantly, best way of detecting object(text), limitation of Tobii eye engine.

We have countered those challenges by performing various experiments. We have performed experiments on text box size – which determine what is the right size of text box. Experiment on delay amount – which determines, what is the right delay for a word and to navigate between pages. This paper provides the proof of to pick right size of text box and right amount of delay. So, that user will have a smooth experience.

This paper is not a finished article of our intentions. There are lot of other things which can be improved. Future works discusses the potential application ideas we can implement using eye tracker.

# Lessons Learnt

* As a developer, we have limited support from Tobii.
* You need to use SDK provided by Tobii. Building a custom engine is highly challenging.
* There are various options to engage with Eye tracker depending on user requirement.
* Eye tracker is greatly accurate which helps to achieve your experiments.
* Since Eye tracker is mostly used in gaming industry, there are not much resources to help you get started with implementation of new ideas.
* We learnt what is the right size of text box which user can be comfortable seeing it.
* We learnt that what is the right amount of delay required for words and as well as delay required to navigate to other pages.
* We also learnt how to make an effective storytelling. This mainly helps people who are having speech disability.
* There are lot of things that eye tracker can achieve. Like making payments by detecting the user’s eye, Automate many manual processes etc.

## Future Works

This paper is not a finished article of our intentions. There are lot of benefits of eye tracker that are un exposed. We would like to inherit those benefits and improve our application.

Following are our ways we can improve our application further:

1. To deploy our application on windows platform. Currently we need to run using visual studio.
2. We can enhance the UI to a true story book. Currently windows form doesn’t provide much options to tweak the UI.
3. Automate the sound production. Currently we have recorded the sound and we have hardcoded to play the sound upon word highlight. We can possibly provide the URL and get the sounds online, rather than manual recording.
4. You can replace the touch screen. User no more has to touch the screen to perform the intended action. User can develop a system which uses the eye tracker and can perform all the touch actions by eye sight.

SAMPLE: Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*Conference’10*, Month 1–2, 2010, City, State, Country.

Copyright 2010 ACM 1-58113-000-0/00/0010 …$15.00.

DOI: http://dx.doi.org/10.1145/12345.67890

## Conclusion

We have seen the problems associated with eye tracker, technical challenges faced with eye tracker functionality, solution to resolve few associated problems. Our experiments have provided us the right value to perform the intended actions. This paper is intended to for people who are new to using eye tracker and wants to understand the power of eye tracker. This also helps experienced people to enhance or integrate these features into their applications.

## First Page Copyright Notice

Please leave 3.81 cm (1.5") of blank text box at the bottom of the left column of the first page for the copyright notice.

## Subsequent Pages

For pages other than the first page, start at the top of the page, and continue in double-column format. The two columns on the last page should be as close to equal length as possible.

Table . Table captions should be placed above the table

|  |  |  |  |
| --- | --- | --- | --- |
| **Graphics** | **Top** | **In-between** | **Bottom** |
| Tables | End | Last | First |
| Figures | Good | Similar | Very well |

## References and Citations

Footnotes should be Times New Roman 9-point, and justified to the full width of the column.

Use the “ACM Reference format” for references – that is, a numbered list at the end of the article, ordered alphabetically and formatted accordingly. See examples of some typical reference types, in the new “ACM Reference format”, at the end of this document. Within this template, use the style named *references* for the text. Acceptable abbreviations, for journal names, can be found here: <http://library.caltech.edu/reference/abbreviations/>. Word may try to automatically ‘underline’ hotlinks in your references, the correct style is NO underlining.

The references are also in 9 pt., but that section (see Section 7) is ragged right. References should be published materials accessible to the public. Internal technical reports may be cited only if they are easily accessible (i.e. you can give the address to obtain the report within your citation) and may be obtained by any reader. Proprietary information may not be cited. Private communications should be acknowledged, not referenced (e.g., “[Robertson, personal communication]”).

## Page Numbering, Headers and Footers

Do not include headers, footers or page numbers in your submission. These will be added when the publications are assembled.

# FIGURES/CAPTIONS

Place Tables/Figures/Images in text as close to the reference as possible (see Figure 1). It may extend across both columns to a maximum width of 17.78 cm (7”).

Captions should be Times New Roman 9-point bold. They should be numbered (e.g., “Table 1” or “Figure 2”), please note that the word for Table and Figure are spelled out. Figure’s captions should be centered beneath the image or picture, and Table captions should be centered above the table body.

# SECTIONS

The heading of a section should be in Times New Roman 12-point bold in all-capitals flush left with an additional 6-points of white space above the section head. Sections and subsequent sub- sections should be numbered and flush left. For a section head and a subsection head together (such as Section 3 and subsection 3.1), use no additional space above the subsection head.



Figure 1. Insert caption to place caption below figure.

.

## Subsections

The heading of subsections should be in Times New Roman 12-point bold with only the initial letters capitalized. (Note: For subsections and subsubsections, a word like *the* or *a* is not capitalized unless it is the first word of the header.)

### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized and 6-points of white space above the subsubsection head.

#### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized.

#### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized.

# ACKNOWLEDGMENTS

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

# REFERENCES

1. Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (Nov. 1993), 795-825. DOI= <http://doi.acm.org/10.1145/161468.16147>.
2. Ding, W. and Marchionini, G. 1997. *A Study on Video Browsing Strategies*. Technical Report. University of Maryland at College Park.
3. Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM, New York, NY, 526-531. DOI= <http://doi.acm.org/10.1145/332040.332491>.
4. Tavel, P. 2007. *Modeling and Simulation Design*. AK Peters Ltd., Natick, MA.
5. Sannella, M. J. 1994. *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.
6. Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.* 3 (Mar. 2003), 1289-1305.
7. Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology* (Vancouver, Canada, November 02 - 05, 2003). UIST '03. ACM, New York, NY, 1-10. DOI= <http://doi.acm.org/10.1145/964696.964697>.
8. Yu, Y. T. and Lau, M. F. 2006. A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions. *J. Syst. Softw.* 79, 5 (May. 2006), 577-590. DOI= <http://dx.doi.org/10.1016/j.jss.2005.05.030>.
9. Spector, A. Z. 1989. Achieving application requirements. In *Distributed Systems*, S. Mullender, Ed. ACM Press Frontier Series. ACM, New York, NY, 19-33. DOI= <http://doi.acm.org/10.1145/90417.90738>.

Columns on Last Page Should Be Made As Close As Possible to Equal Length