Reading With Your Eyes

Kyle Diederich
The University of Iowa
Iowa City, USA
kyle-diederich@uiowa.edu

Ashkan Bozorgzad
The University of Iowa
Iowa City, USA
ashkan-bozorgazad@uiowa.edu

Ziqi Su
The University of Iowa
Iowa City, USA
ziqi-su@uiowa.edu

Suhas Kumar
The University of Iowa
Iowa City, USA
suhas-kumar@uiowa.edu

ABSTRACT

Eye tracking software can be used to provide positive user experiences when reading story books. This paper details initial results and summarizes the lessons learned from our initial experimentation using the Tobii EyeX Controller and associated software.

1. INTRODUCTION

The purpose of our paper is to discuss the use of the Tobii EyeX Controller and associated software as a methodology to improve user experience with storytelling. As part of this project, we developed a technique to utilize eye tracking software to convert user gaze on a story book page into precision text detection. Our experiments included performing trials using various amounts of text on a screen to test user satisfaction and determine the appropriate text size and quantity.

We encountered several challenges, such as appropriate size of text box to use for accurate results and proper delay amount between text box activations. Our goal was to support an overall positive user experience.

This project does not represent a comprehensive study of all applications of eye tracking usage. There are significate future applications for this technology which can be the focus of future work.

2. RELATED WORK

In order to provide an improved reading experience for children, we created a story book application incorporating eye tracking control. Stories utilizing eye tracking software allow children to experience reading without requiring the assistance of additional individuals.

Prior work by Feng demonstrated that there are two major limitations of current eye tracking technology [4]. First, eye tracking may not be accurate enough to measure the actual gaze target of a person. Second, even though the eye tracking software accurate is very accurate, people's gaze position may constantly move due to involuntary eye movement. This rapid, random, involuntary eye movement makes it difficult to solely rely on gaze as an indicator of intent.

Researchers from University of Auckland created a program called *actigaze* which allows users to access Internet websites without mouse interaction [5]. The *actigaze* environment requires user to gaze at the certain objects (for example, web-links), then confirm that gaze by fixating on its associated color [5]. Their work was instrumental in designing our experiments.

Evans and Saint-Aubin demonstrate that children spend more time looking at illustrations than the printed text in a storybook when parents or adults read them a story [1]. The study also showed that text highlighted will encourages children pay more attention to certain details [1]. We used this result to structure an environment where every word triggers a reaction from the storybook.

3. GOALS

This project focuses on developing techniques to assimilate eye tracking software into precision text activation during storytelling. In order to achieve, this goal three things are required:

(1) Create a software environment which seamlessly integrates eye tracking and story book reading.

(2) Utilize eye tracking with the Tobii EyeX Controller and Tobii EyeX Engine to drive the software.

(3) Provide a positive experience for users.

In order to provide a positive user experience the software:

(1) Should highlighted the text corresponding to user gaze,

(2) Should play a sound file of the target word.

Challenges considered and addressed by the project include:

(1) Discovering techniques to track eye movement based on intended user gaze.

(2) Physical differences between user eye shape and detection by the eye tracker software should not negatively impact user experience.

(3) There are variety of interaction methodologies provided as part of the Tobii software development kit. We must select the interaction type which allows for greatest accuracy.

(4) There are inherent limitations with using a software designed primarily for the gaming industry.

The following sections describe the process we followed to overcome these challenges.

4. DESIGN

The main goal in the creation of this project was to demonstrate how current advancements in eye tracking could be utilized in conjunction with story book reading. In order to accomplish this goal, we created an electronic book, in which all interactions only require the use of eye tracking. Taking inspiration from currently available applications for children aged 5 and under such as *Go Away*, *Big Green Monster! for iPad* and *Barnyard Dance – Boynton*, we decided to adapt an existing Aesop fable, *The Lion and the Mouse*.

Our story features multiple interactable elements on each story page. When the user's gaze falls on a particular object (most objects correspond to individual pieces of text – specifically single words), we trigger an event which results in both the word becoming highlighted and the word being spoken through playing a pre-recorded voice file for each individual word. We added inertia to the activation of these interactors following our experimentation, which allows the user to read at a reasonable speed without errant activations of neighboring words. For this application, we consider a reasonable speed to an amount of delay between onset of gaze and activation of event, where the user can activate each interactor with a timing sequence consistent with reading aloud. The process for determining the precise amount of delay required to obtain this balance is discussed in detail in the experimentation section.

In order to provide control of the story using the Tobii EyeX Tracker, we created a .NET application which incorporated the Tobii EyeX Engine. This approach requires the use of a software development kit (SDK) provided by Tobii. In the application, the EyeX Engine is required to maintain the connections between the three main components: Windows operating system, EyeX tracking hardware, and our client application.

Our application is built using Gaze-Aware Behavior, a service interaction provided by the Tobii SDK. This type of interaction was chosen because it allows for the fastest response to eye movements of the variety of interaction types available through the Tobii SDK. Gaze-Aware Behavior requires a clear set of interactable elements (the individual words and page turning mechanisms in our application) and sufficient spacing so that multiple items from the available set are not activated at the same time.

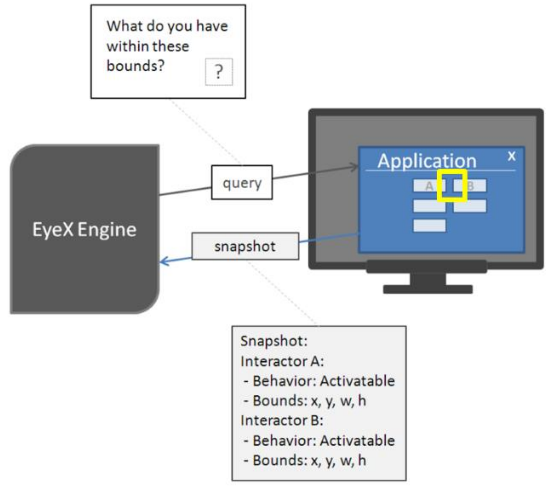

Figure 1. Query Snapshot Cycle

All interactions between the three components of the project follow the Query Snapshot Cycle (See Figure). Periodically, the EyeX Engine will issue a query command to the active application. This application will respond with a snapshot. The boundaries of this snapshot are fully controlled by Tobii and were not available for manipulation. Essentially, this snapshot contains information relating to any interactors (and their boundaries) which occur within the snapshot range. The EyeX Engine compiles the snapshot information with results of the eye tracking device to provide the application with data on which, if any, interactors should be activated.

5. IMPLEMENTATION

Because we are coding using both .NET and the EyeX Engine, our program requires the use of Microsoft Visual Studio. The Tobii EyeX Controller (eye tracker) will only work with Windows 8 or later.

In order to use any of our software, the user must first activate the EyeX Controller software provided by Tobii. This software is used to create a user profile, containing relevant information like seating position and whether the user wears corrective vision devices. When the program has this information, the software proceeds to calibrate the device to track the user's gaze. Our experimentation revealed that the success of the experience relies heavily on this calibration period. Changes in seating position or inaccurate gaze during the calibration, result in the device misinterpreting the intended gaze target of the user.

Once a profile has been selected and the calibration is complete, the Tobii EyeX Engine will automatically connect to any application running its framework. We then launch our application, which is automatically connected to the EyeX Controller, and the user can begin to interact with the story using only gaze.

Upon activation of our application, the user is greeted with a story scene similar to those found in children's books. Our intention was to not provide specific instructions, because the user only needs to look about the page to activate the various elements of the story. (See Figure)

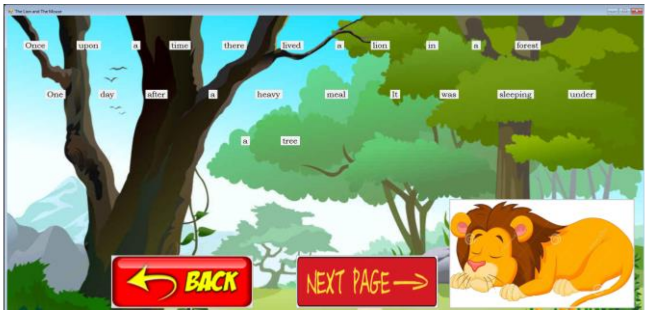

Figure 2. Story Main Page

On each page, the spacing between words and the spacing between lines of text was chosen as a result of our experimentation. As the user directs her/his gaze about the application, the EyeX Engine will periodically issue query commands to the application based on current gaze data provided by the EyeX Controller. (See Figure)

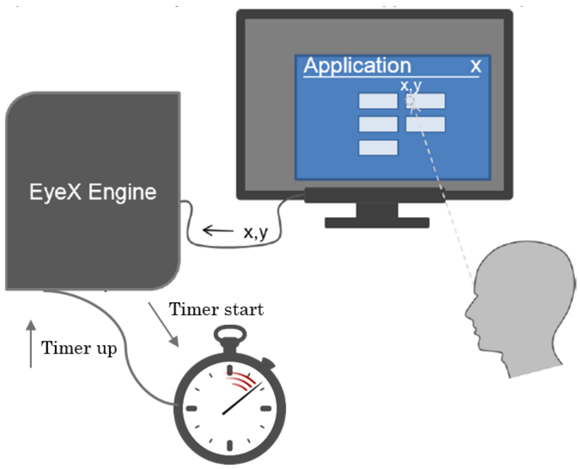

Figure 3. Control Flow of Application

When this data indicates that the user gaze has crossed the boundary of an interactor, the EyeX Engine will issue a command to our application informing the application that it should launch an event corresponding to that interactor. The white background in the story figure, indicates the boundary for a particular interactor. The actual command sent to the application by the EyeX Engine contains only the information on whether the current interactor is or is not the target of the user's gaze according to the internal algorithm of the EyeX suite. This feature allows our application to activate a word or deactivate a word depending on whether or not it is the active gaze target.

Activation of a word in our application consists of two features:

1. We modify both the background color and the text color to yellow and red, respectively.
2. We launch a prerecorded sound file of the word using a Windows Media Player Control object.

When activating a word, we decided that we should have minimal processing, so as to not introduce artificial delay prior to the display updates. In the current version of the application, the modification of the text color requires negligible delay. However, we do introduce a penalty of approximately 85 milliseconds on average to launch a sound file. Combined with another slight delay due to communication between the EyeX Controller, EyeX Engine, Windows OS, and our application, it can take up to 100 milliseconds to go from gaze to activation of sound file.

In order to support a positive user experience with the story application, we introduced inertia to each interactor. In the context of this application, inertia corresponds to a delay between when the EyeX Engine says an interactor is the gaze target and when the program activates that particular interactor. In the case where the engine provides an additional indication that a particular interactor is no longer the gaze target prior to the expiration of the delay parameter, the interactor event is not launched by the application. This feature was introduced to account for the variations between saccades (eye movement from one location to another) and fixations (eye position is constant and information is perceived) in normal eye movements. Many readers move their eyes approximately every quarter second [2]. Therefore, our delay parameter had to be set to a threshold which would remove unintended fixations of the normal reading process, such that, the application would only activate specifically intended words.

Our experimentation indicated that a delay of 305 milliseconds allowed for a reasonably smooth reading experience; however, this amount was widely contested based on personal preference. We did observe that 200 milliseconds was the minimum amount required, but this time delay required specific focus and practice.

Because our page turn buttons included text, we created a different threshold which allows the user to read the text on the button without triggering the page to advance.

6. EXPERIMENTS

In order to increase the accuracy of the system, two sets of experiments were designed. The results of the first experiment were used to determine an appropriate size for each textbox. The second experiment was structured to determine the amount of delay between then the user's gaze falls on an interactor and the event associated that interactor become activated.

6.1 Experimentation for the Textbox Size

In this project, textboxes (rectangular object shapes) were used a boundaries for interactors that the EyeX Engine would allow application to recognized as detecting user gaze on a word. The ideal size of these textboxes is consistent with the size of the word. However, some small words such as the article: a, an, and the, are too small and cannot be detected with high accuracy by the eye tracker. In order to determine the most reasonable textbox size, different textbox sizes were created and the accuracy of the software was measured. Our goal in this experiment was to determine the size that maintain a high accuracy while also allowing for the majority of a sentence to fit on a single line.

To create different textbox sizes, horizontal and vertical grid lines were used. The constant window size was divided to multiple columns and rows with these horizontal and vertical grid lines. The great the number of horizontal and vertical lines, the smaller the textbox size and vice versa. For example, 9 * 23 means the window has 9 rows and 23 columns. Figure 4 shows the 11 * 11 textbox size as a sample.

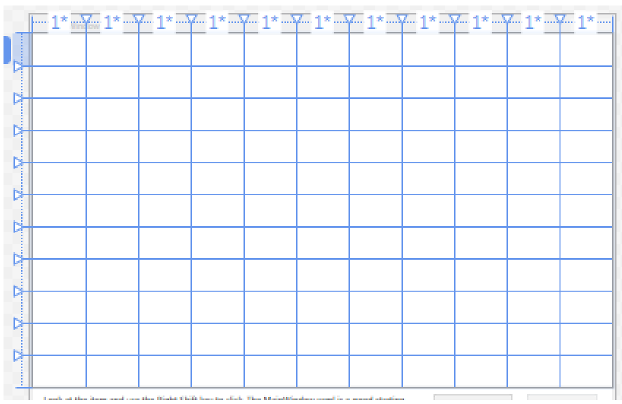

Figure 4. 11 * 11 textbox size

In order to determine the appropriate minimum textbox size, seven different text box sizes were chosen. First, the 9 * 9, 11 * 11 and 13 * 13 were used to determine a suitable number of rows (the height of the textbox). Based on the results of that experimentation, the following sizes: 9 *15, 9 *17, 9 *19 and 9 * 23, were used to calculate the appropriate number of columns (the width of the textbox). Two trials were conducted for each textbox size by 4 participants. In each trial, 10 sets of random numbers (row, column) were generated, and the user was instructed by the experimenter to gaze at a particular cell. When the participant fixated on the desired location, she/he manually pressed a key which highlighted the gaze target as determined by the EyeX Controller. Therefore, for each grid size, a total of 80 cells were tested for detection (4 participant * 2 trials each * 10 gaze targets). The accuracy was calculated using the following formula. This accuracy ratio was used to determine which textbox size should be utilized in the final version of the software:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{Correctly\ detected\ words}}{\boldsymbol{Total\ number\ of\ words}} \qquad (\mathbf{1})$$

Table 1 shows a sample of requested versus actual gaze targets result by a participant for the 11 * 11 textbox size trial. If the detected and requested cell was same, the results were considered to have been interpreted correctly. If the result were not in agreement, the results were consider to be wrongly detected and are highlighted with red color in Table 1. As previously noted, calibration was performed prior to each trail throughout the experiment.

Table 1. A sample of a single participant results for and 11*11 textbox size.

Trial 1		Trial 2	
Request cell	**Actual Detected Cell**	**Request cell**	**Actual Detected Cell**
(9, 9)	(9, 9)	(10, 11)	(10, 11)
(8, 1)	(8, 1)	(5, 4)	(5, 4)
(10, 7)	(10, 7)	(7, 6)	(6, 6)
(3, 6)	(2, 6)	(2, 4)	(2, 4)
(9, 9)	(9, 9)	(8, 9)	(8, 9)
(2, 8)	(2, 8)	(5, 7)	(5, 7)
(1, 8)	(1, 8)	(4, 3)	(4, 3)
(2, 1)	(1, 1)	(7, 1)	(5, 1)
(7, 2)	(6, 2)	(5, 1)	(4, 1)
(9, 2)	(8, 2)	(8, 7)	(7, 7)

As demonstrated by this table, the participant achieved 60% accuracy for both trials. Figure 5 illustrates the accuracy of three sample sizes.

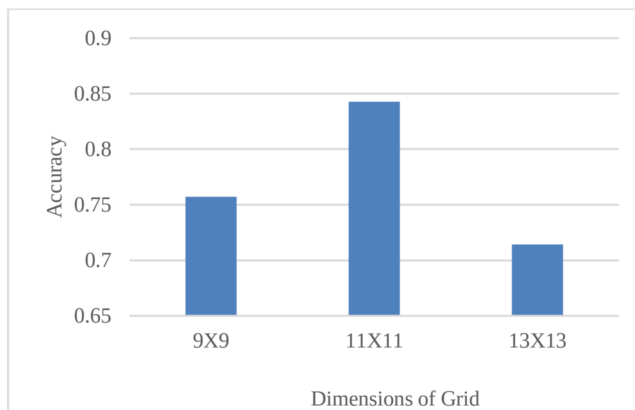

Figure 5. The accuracy of three sample sizes

As shown in Figure 5, the results of our experimentation resulted in lower accuracy than expected for the 9 * 9 grid trials. That is, the 11 * 11 trials used a smaller textbox size, but had higher accuracy results than the 9 * 9 size trials. Our interpretation of the cause this discrepancy in the experimentation results, most likely stems from poor calibration. In later trails, the participants became more familiar with the calibration process, and our results were more consistent with our expectations. Additionally, a closer look at the actual errors in the 11 * 11 trails indicated that the overwhelming majority of inaccurate activations were based on selection of a different row, rather than a different column. We determined that 9 rows was the most logical amount to use in the remainder of our experimentation.

After initial determination of the amount of grid rows, additional testing was conducted for 9 *15, 9 *17, 9 *19 and 9 * 23 sized grids. These trials were used to determine a reasonable width of potential textboxes. Figure 6 indicates the results of these experiments and the 9 * 9 case.

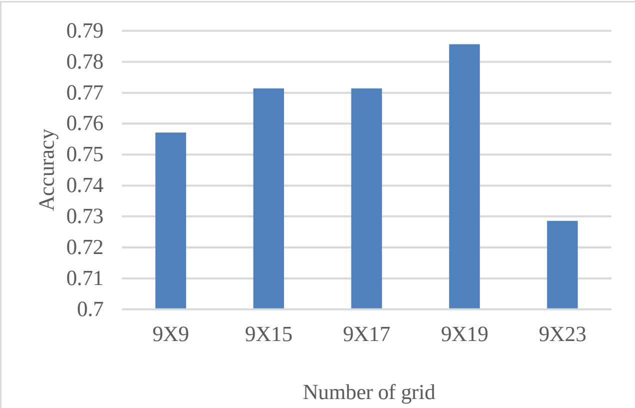

Figure 6 the accuracy of 9 rows samples with different number of columns

As illustrated by Figure 6, the 9 * 19 trials has the highest accuracy. Therefore, 19 columns were selected for the width of the grid. The results of this experimentation indicate that a maximum sentence length of 19 words will not negatively impact user experience.

It is important to mention some of the potential causes of the variability in our experimentation.

(1) Learning process of users during the trails: for all participants, the 9 * 9 case represented the first time interacting with the Tobii EyeX Controller. A signification portion of the errors was due to unfamiliarity on the part of participants with the hardware. After the initial trial, participants obtained some experience and performed better on subsequent trials.
(2) High sensitivity of the calibration process: prior to each trial, the participant completed the calibration process required by the Tobii EyeX Engine. This process is extremely sensitive and can drastically affect the results.
(3) Limited number of the trials: our experimentation consisted of a limited group number of participants. A greater number of participants would result in less variability in results.
(4) Limited number of experiments: the 4 participants in this experiment have differences between their eye shapes. One participant wore corrective lenses, another one contact lenses, a third had a narrow eye opening, and the final participant had a large eye opening without any vision correction.

6.2 Experiment for Inertia

Our final experiment was designed to determine the appropriate amount of inertia to add onto each word interactor. This delay is necessary to control for the rapid nature of eye movements. Adding inertia to each interactor assists in avoiding random selection of gaze targets. This experiment tested 6 different delay amounts. The values were: 200, 300, 400, 450, 500 and 600 milliseconds. Each participant was instructed to read a selection from the story for each delay amount. Following each trials, the experiments asked the participant to rate their user experiment using a Likert Scale, the results of which are displayed in Table 2.

Table 2. Likert Scale for User Experience

Scale	Good Experience?
1	Strongly Disagree
2	Disagree
3	Neither agree nor disagree
4	Agree
5	Strongly Agree

The results of the delay experiment are illustrated in Figure 7.

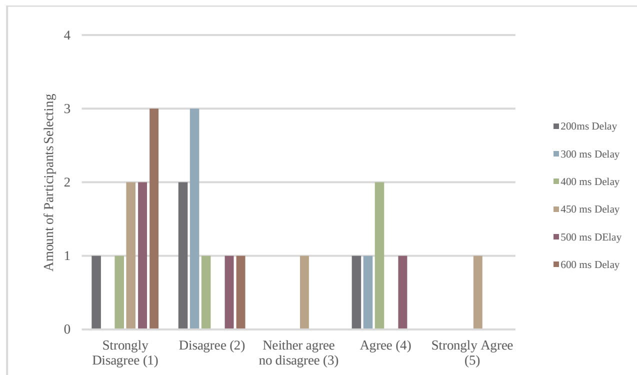

Figure 7. The results of delay experimentation

In Figure 7, each color represents each delay amount and the number of participants who selected that Likert Scale amount are shown on the vertical axis. For example, 3 participants strongly disagreed that 600ms delay provided a good user experience. By using the weighted average method, and average value of preference for each delay amount was calculated and is displayed shown in the Figure 8.Figure 8

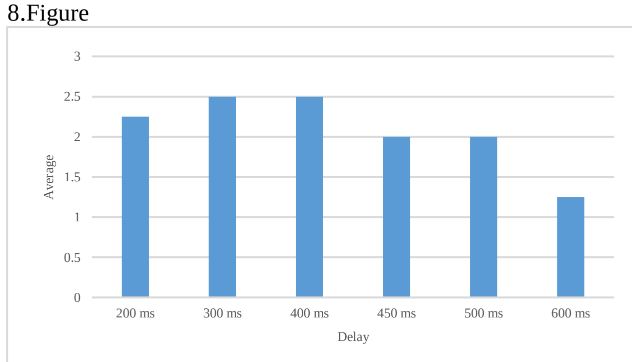

Figure 8. Weighted Average of Participant Preference

As demonstrated in Figure 8, participants selected both 300ms and 400ms delays as most favorable. Therefore, the delay of 305ms was selected as the final per word interactor delay.

7. LESSONS LEARNED

Throughout our experimentation and development with the Tobii EyeX Controller and Tobii EyeX Engine, we learned a number of important lessons:

(1) There are not many resources available for inexperienced developers who wish to create eye tracking software (other than games) using the Tobii EyeX engine.

(2) The preferred size of text in a story environment that allows for greatest user satisfaction.

(3) How to control for rapid, random eye movements to support a positive user experience.

(4) This type of environment can support children in learning the proper pronunciation of words in a story.

8. CONCLUSION

Based on the results of our experimentation, we made the following discoveries:

(1) The Tobii EyeX Controller's accuracy depends greatly on the calibration process.
(2) In our experiment results, we discovered that a sentence length of approximately 20 words was optimal for performance and user experience.
(3) An optimal delay amount is between 200ms and 400ms, depending on user familiarity with the Tobii EyeX suite.
(4) As users' fluency increased the variability declined and their experience was enhanced.
(5) Development is limited because the Tobii EyeX suite is designed for gaming and only runs on Windows 8 or later.
(6) Our group was limited to four participants; therefore, our conclusions may be based on inaccurate results.

9. FUTURE WORK

Our experimentation provides a first attempted at utilizing eye tracking technology to improve user experience with story reading.

We can enhance the user interface (UI) to more closely resemble a printed story book.

Incorporating a method to automate the process of recording each individual word, would allow users to add their own voice to the story.

10. REFERENCES

[1] Evans, Mary Ann, and Jean Saint-Aubin. "What Children Are Looking at during Shared Storybook Reading." Psychological Science. U.S. National Library of Medicine, 16 Nov. 2005. Web. 10 Dec. 2016.

[2] Eye movement in reading. https://en.wikipedia.org/wiki/Eye_movement_in_reading.

[3] Eye Tracking. https://www.usability.gov/how-to-and-tools/methods/eye-tracking.html. (Retrieved Nov 11).

[4] Gary Feng (2011) Eye Tracking: A Brief Guide for Developmental Researchers, Journal of Cognition and Development, 12:1, 1-11, DOI: 10.1080/15248372.2011.547447

[5] Lutteroth, Christof, Moiz Penkar, and Gerald Weber. "Gaze vs. Mouse : A Fast and Accurate Gaze-Only Click Alternative." Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15 (2015): n. pag. ACM New York, 5 Nov. 2015. Web. 1 Nov. 2016.