Assisted Reading

Hochschule
Augsburg University of
Applied Sciences

Interaction Engineering
Winter Semester 18/19
Ryan McErlean, Emil Meyn Mathiassen
Prof. Dr. Michael Kipp
Augsburg University of Applied Sciences

Februar 8th 2019

Abstract

We would like to introduce our project titled "Assisted Reading"; The aim of this project is to make the online reading or E-book experience more convenient and enjoyable to the user. We present gaze-assisted reading techniques which use the traditional linear reading style. This approach enables continuous gaze-assisted reading without requiring manual input to scroll, along with text highlighting for increased reading clarity: We implemented two different page scrolling techniques in order to try and get an impression for which technique is most optimal. We also applied a text highlighting feature in order to test if it makes reading the text more clearer, and improves the overall reading experience. Testing our project with users and gathering their general opinion gave us an insight into which features were preferred and most effective.

Motivation

In our modern society we live in a very much digitalised world; From online articles to reading E-books, people have an enormous library of text all available on one device. We want to tap into this ever growing field by making reading dynamic and effortless. Previous research papers such as "AutoPager: Exploiting Change Blindness for Gaze-Assisted Reading" (Wilson and Williams) change the text while the user is reading in order to make reading seamless; However this can be problematic if the user wants to re-read a section at the top of the page. It also doesn't experiment into making text easier to read, it only tries to improve the "page turning" effect. Our reading assistant allows the user to scroll through the text at a pace which suits their reading level. Also if the user wishes to re-read previous text they can simply scroll back through the text. For people who have trouble reading, or poor vision; We highlight or enhance text, which makes the section they are reading more clear.

Concept

For this project the main interaction technique is through the readers eye gaze and dwell time. We used a Tobii Eye Tracker 4C in order to determine where the reader was gazing at any given moment.

Figure 0.2. Tobii Eye Tracker 4C

Once the readers eye gaze is tracked we can use this in order to determine what section or line of the text they are reading; When the section of text being read is indicated we use this to implement our text highlighting and scrolling features. For example when the user is reading a certain line, then that area of text will be highlighted or made more clear so that they can concentrate on the particular text they are reading, and it is made easier to interpret. We also use the reader's gaze area for text scrolling; The first gaze scrolling technique is when the reader is focused on the middle of the page. Once the reader reachers this point then the text is automatically adjusted to scroll so that more text is revealed. The second gaze scrolling technique is implemented when a user is gazing at the buttons on the side of the page the page; This also

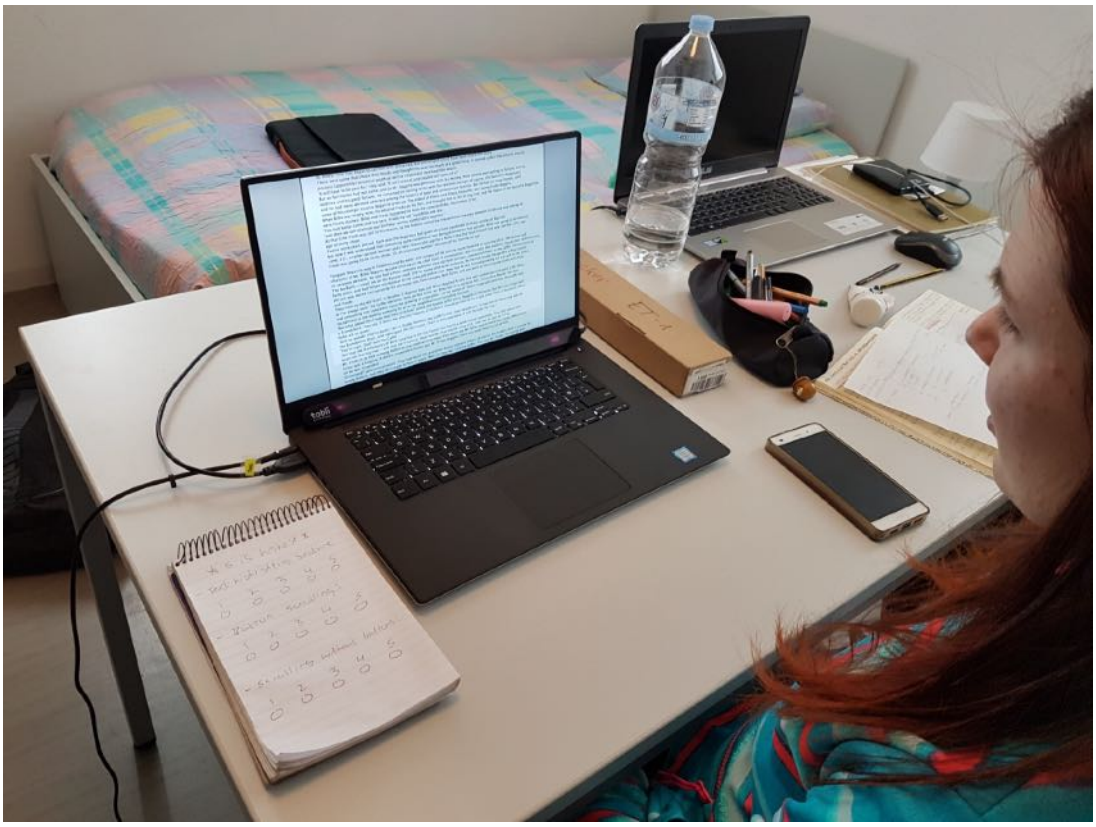

Figure 0.1. User test

adjusts the page automatically so that more text is revealed; However the main difference with these two methods is that the first option allows you to scroll at a controlled pace (which can be increased depending on how far down the page the reader gazes. The button scrolling option is much faster and allows users to flick throw multiple pages at a fast rate.

Implementation

We initially started off with all three features working together, but we later decided it would be better to test the two scrolling options separately, and test the programme with/without the text highlighting. This gave us a feel of what features the users prefer and what features are distracting or unhelpful. The hardware that we used was a Tobii Eye Tracker 4C along with a windows pc/laptop. The programme we used to work with the eye tracker was Processing; This enabled us to write the assisted reading programme compatible with the Tobii eye tracker.

Dynamically scrolling

The dynamically scrolling feature is created with the y-position of the gaze squared, multiplied with a factor to increase the speed of the scroll dynamically depending on how far down on the page you look. While scrolling backwards the y-value will be decreasing and therefore the y-value is minussed with the value of the position of the page where it is supposed to start scrolling back and thereafter multiplied with a negative number to create the increasing scrolling speed. Scrolling backwards is slower at maximum speed because it does not scale with a squared value, this was evaluated because scrolling backwards is done less than scrolling forwards.

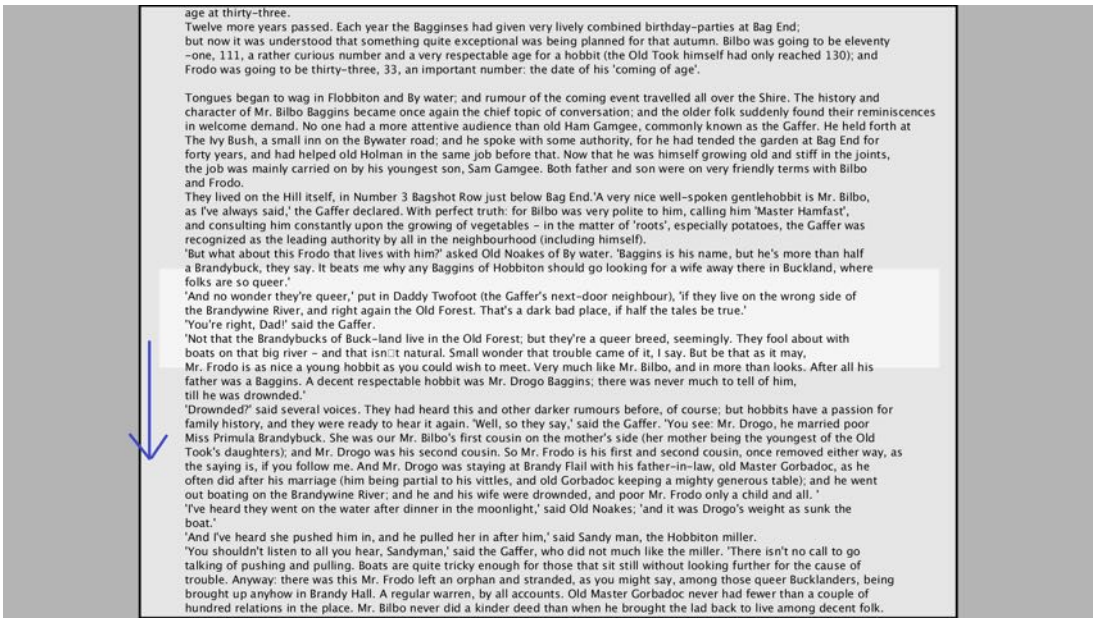

Figure 0.3. Dynamic scrolling

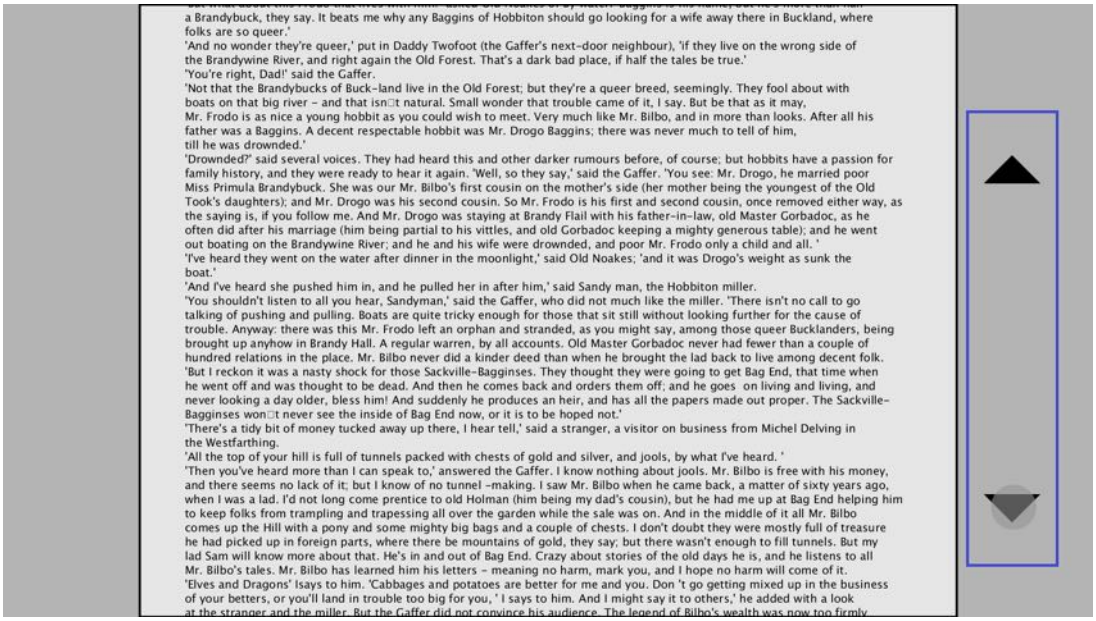

Figure 0.4. Button scrolling

Button Scrolling

The buttons scrolling is a more direct and concise way of scrolling through the text. Two arrows (one pointing up, the other down) on the right side of the page indicate scrolling back and forth (up and down). This is simply achieved with a static 'scrolling' factor added when the arrows are gazed upon/within.

Text-highlighting

For the text-highlighting feature the gazes Y-value is used to determined the middle of the highlighted area. From this middle value a rectangle is created on to the gaze position. This

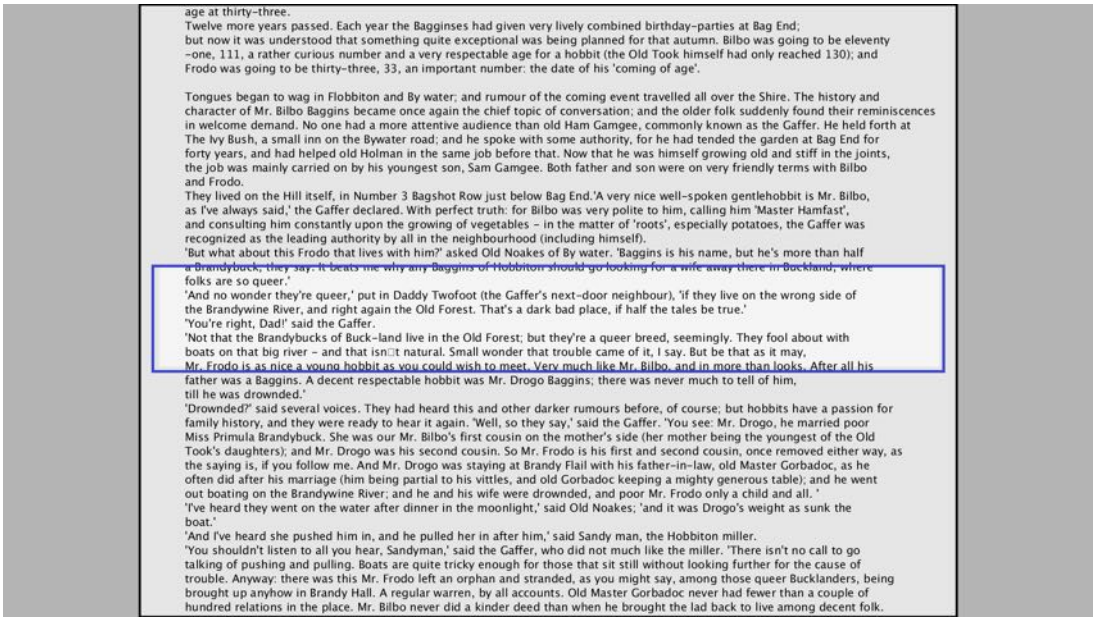

Figure 0.5. Text highlighting

position is only changed if the gaze reaches the beginning of the next line, which is when the x-value crosses a certain threshold. This was done for two reasons. First of all to avoid the box from jittering too much and confuse the reader more than it benefitted. The second reason was that looking away would not change the position of the highlighted text. Before the highlighted area would follow as soon as the gaze entered the screen and therefore would not help in finding the spot that was reached

Evaluation

Feedback

The feedback overall show a positive response to the prototype; We did not carry out an intensive study with comparable data but we thought it was still important to see how users would respond to the different features of the programme. Two users were asked to test the prototype and rate each feature on a scale of 1-5, we also asked for their general opinion of the features. Both users enjoyed the gaze scrolling; However they had disagreeing opinions on which scrolling option was most effective or efficient. The text highlighting was rated as 3 by both participants which may indicate it is less effective as the two scrolling features. We feel that both scrolling options have benefits and therefore they could both be implemented and use simultaneously. The button scrolling feature seems to be more effective for fast page flipping, where as the dynamic scrolling feature is more suited for practical reading and slow page flipping.

Test Feature	Highlighting	Button Scrolling	Dynamic Scrolling	Overall Comments
User 1	3	3	5	*"The text highlighting isn't annoying but I would prefer reading without it"*
User 2	3	4	2	*"I think the dynamic gaze scrolling is great but the button gaze is too fast"*

Improvements and Limitations

The prototype in the given form is very limited to a straight text without too much formatting or longer texts, split into sections or chapters. Talking about chapters and sections another feature to add would be jumping to the next/previous chapter and could also involve how to display a menu of sections/chapters. It would have been very interesting to test this in a disturbing/stressed setting like a full canteen or a busy train station to see if the highlighted zone would actually improve focus and keep gaze longer or what effects it would have shown. Looking at the different parameters for reading, like reading speed and focusing threshold could open a personalised feature, where scrolling speed, scrolling position and highlighting area, could be adjusted by the user either by manual inputs or somewhat automatically. With a reading speed tracker, automatically adjusting scrolling speed based on an average calculated reading speed could be implemented as a further development.

Conclusion

With the few responses we had, it is hard to state any result, but with the responses in mind and our own observations, the current functions of the product work well and from the feedback it would seem that implementing both scrolling options would make sense, since they don't overlap with each other. The difference in preference could be reading speed, which could mean a personalisation could give different results for the same two features. Testing the product in a distracting setting would properly give a better indication towards if the highlighting is helpful or not, whereas in the current tested scenario it is not. Adding more functionalities, like chapter menus or chapter skips, to make it more similar to a book or article would increase the use-cases.