2022

Scrolling vs Paging: Reading Performance and Preference of Reading Modes in Long-form Online News

Richard Herlihy

Scrolling vs Paging: Reading Performance and Preference of Reading Modes in Long-form Online News

Richard Herlihy

A dissertation submitted in partial fulfilment of the requirements of
Technological University Dublin for the degree of
M.Sc. in Computer Science (Advanced Software Development)

2022

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Advanced Software Development), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Technological University Dublin and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: Richard Herlihy

Date: **16 June 2022**

ABSTRACT

This study explores the impact of scrolling and dynamic pagination in long-form online documents on reader performance and reader experience.

Previous research has produced mixed results, indicating no difference between modes, or a positive effect favouring scrolling. Recent advances in web standards have enabled simpler, dynamic, performant methods of pagination to tailor content responsively to any screen, meriting renewed study in this area. This paper uses one such method to load subsequent online news pages instantly without buffering.

In an online browser experiment with 38 participants, an increase in reading speed in the scrolling mode was found at a level of significance. This follows previous research which has suggested that while a scrolling presentation style exacts extra demands on working memory capacity (WMC), many current web users have developed compensatory strategies and cognitive flexibility for navigating scrolling web documents.

Key words: Digital reading, long-reads, continuous imaginary reading, CSS, dynamic pagination

ACKNOWLEDGEMENTS

I'm very grateful for all of the assistance and support from TU Dublin academic staff. In particular my research supervisor, Mariana Rocha, whose guidance and feedback was invaluable in recent months. Also to Andrea Curley who taught a pair of web design and user experience modules which greatly influenced this dissertation.

A big thank you to all of my family and friends for their constant and patient backing (some of whom helped pilot test this study!). In particular Maeve, Ruby and Tammy - whose help was crucial in the final weeks!

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1. INTRODUCTION

What makes for a good, focused reading experience online? This paper explores how different presentation styles (scrolling and paging) in long online documents can impact upon reading performance and subjective preferences. The primary method of research is an online browser experiment.

In the following subsections, the research design is outlined. This includes a short description of the background in this area of study: both in practice and in the research literature. This chapter continues by stating the research problem, along with a research question and a pair of testable hypotheses. The research objectives of this study are then defined, followed by a description of scope, limitations, and delimitations, before the chapter concludes with an outline of the document.

1.1 Background

Nearly all websites today are rendered as scrollable, responsive documents. This trend has increased recently, with the average user now scrolling vertically more than ever before (Fessenden 2018).

This is especially the case for articles found on online news websites, which now almost exclusively present articles in a scrolling style. However, an alternative presentation style has been used intermittently in the past: pagination, where articles are split into distinct web pages which users have to linearly navigate through. Previously, in the 2000s, pagination enjoyed moderately widespread use among websites which published long-form articles (i.e. features or essays running to thousands of words) such as *The New York Times*, *The Washington Post*, and *Business Insider*.

Critics, however, have argued that this navigation style "irritated users" and was primarily motivated by "pageview juicing", a means of increasing "opportunities for ads" (Davidson, 2007; Manjoo, 2012[1]). In 2011, Google revealed through its *Google Search Central Blog* that it would automatically direct search traffic to single-page versions of online articles, if it detected one existed, instead of the default paginated version. "User testing has taught us that searchers much prefer the view-all,

[1] Stop Pagination Now - "Why websites should not make you click and click and click for the full story." (Manjoo, 2012)

single-page version of content over a component page containing only a portion of the same information with arbitrary page breaks" (Google Search Central Blog, 2011). Since then, pagination on popular websites which publish long-form articles has almost entirely disappeared, though it is still occasionally found in long articles with discrete sections on enthusiast or special interest websites such as *Tom's Hardware*.

Nevertheless, previous research suggests there may be mechanisms through which paging style presentation of texts aids factors such as reading performance, working memory and overall comprehension (Brady et al., 2018; Singer and Alexander, 2017; Sanchez and Wiley, 2009). However, findings at a level of significance have been elusive, and past research has been either inconclusive (Dyson and Kipping, 1998) or suggested benefits to paging or scrolling in specific groups only (Brady et al., 2018; Singer and Alexander, 2017).

One possible explanation for elusive findings and a perceived backlash against pagination (in part precipitated by Manjoo's 2012 article and the wider response to it) are user experience limitations inherent in how pagination has been achieved to date.

Firstly, since the predominant style of web pagination was based on a simple multiple-page application (MPA) structure - every chunk is a separate web page - it entailed both delay and disorientation for the reader between every page turn, since a new web page needs to be loaded and the browser has to re-render the layout. Secondly, this caused inconsistent user experiences since an arbitrary cut-off was used to delineate each page, meaning some users on smaller displays might see a scroll bar or, in the case of very large displays, might only see a limited morsel of text surrounded by white space.

Previously, dynamic reflowable pagination (where the text is fitted to fill a user's screen appropriately) has only been possible through technologies such as Adobe Flash or JavaScript libraries. However, new standards in Cascading Style Sheets (CSS; the syntax language used to style web documents) now allow for simple, performant standards-based reflowable dynamic pagination with minimal JavaScript.

Lastly, while past research has explored paging presentation in the context of short text passages (Dyson and Kipping, 1998; Chaparro and Baker, 2003), this style of presentation has not previously been studied in long-form online news articles (which require sustained, focused attention) - though some research into presentation style and working memory has focused on lengthy educational texts (Sanchez and Wiley, 2009).

While there has been a bounty of research into the impact of pagination on reader performance and usability in contexts such as mobile devices, ebook readers (Vanderschantz et al., 2018), and web-based technologies such as Adobe Flash (Liesaputra et al., 2019) – there has been limited research into web-based full-browser reading.

1.2 Research problem

1.2.1 Research problem

Web standards have evolved and advanced significantly in the past two decades, but little research has focused on how new ways of implementing dynamic reflowable pagination affects digital reading. By dynamic reflowable pagination, we refer specifically to the splitting up of long texts page-by-page so that each page fits inside a container in a display area which is sized relative to the available viewport so that no scrolling is necessary and no content flows below the fold.

Despite voluminous research into digital reading, there is a gap in existing literature pertaining to how technologies and techniques like AJAX (Asynchronous JavaScript and XML) or CSS3, which both enable instantaneous transitions between dynamic pages, might augment online reading and provide a reassessment of the value of pagination, especially in long-form online news.

One gap in previous work in this area is that many research designs exploring presentation style (e.g. Dyson and Kipping, 1998; Chaparro and Baker, 2003) have utilised relatively short text samples on low-resolution displays, which may not have provided enough continuous reading for the benefits of either pagination or scrolling to accrue to a level of statistical significance. The design in this present study aims to address this problem by utilising very long samples of approximately 1500 words (3000+ words in total).

1.2.2 Research question

Thus the motivation of the present study can be summarised in the following research question:

RQ1: What impact does dynamic reflowable pagination on web pages have on the digital reading experience of longform news articles?

To explore this question, this study considers measures of reading speed (Dyson and Kipping, 1998; Chaparro and Baker, 2003; Shaikh, and Chaparro, 2005) and reading mode preference (Rello and Baeza-Yates, 2012) which have been studied as variables in earlier research into digital reading. It also uses a number of other variables (described in *1.4. Research methodologies* below) to validate and understand findings.

1.2.3 Hypothesis testing

To answer the research question and provide a framework for testing and validation, this study poses the following null and alternative hypotheses:

H_0: Presenting a 1500-word long-form-style text using **dynamic reflowable pagination** to online readers **does not result in a statistically significant reduction** in *reading time* and an indication of subjective *preference* for this mode when compared with a standard web presentation style using **scrolling**.

H_1: Presenting a 1500-word long-form-style text using **dynamic reflowable pagination** to online readers **causes a statistically significant reduction** in *reading time* and an indication of subjective *preference* for this mode when compared with a standard web presentation style using **scrolling**.

The level of statistical significance refers to a confidence interval of 95% ($p < 0.05$).

Long-form text refers to non-fiction journalism articles (e.g. features, or essays) and also to creative narrative style *new journalism* (Britannica, 2022) which demands continuous focused reading.

1.3 Research objectives

This section describes research objectives used to plan this thesis and schedule activities.

1.3.1 Objective 1: Develop minimal web reader & experimental application

- Review appropriate CSS and JavaScript features and libraries which can be used to achieve pagination and carry out the experiment. This included React

JS, a number of pagination packages, as well as so-called “vanilla” implementations of JavaScript and its native features.

- Develop a minimal reader which can dynamically render and paginate news articles so that selected texts can expand and reflow to responsively fill each page (i.e. the viewport height). The reader implementation should provide near-instantaneous navigation (i.e. no discernable loading time between pages). Additional navigation options should be provided to maximise user experience such as keyboard navigation (advance using arrow keys) and mouse (scroll mousewheel to turn the page).
- Develop experimental application. Code or utilise an existing suite for carrying out a user evaluation browser experiment which captures reading times and responses to questionnaires.

1.3.2 Objective 2: Create experimental protocol and procedure.

- Select and research a platform (i.e. Prolific) for recruiting participants for the experiment and collecting additional data.
- Select two news articles or samples (1500 words each). Both articles will be narrative non-fiction in nature (i.e. a cohesive, linear narrative which necessitates reading from beginning to end) and will be sourced from an open API or Creative Commons source such as The Guardian or ProPublica.
- Create test questions. Assessment methodology will use existing research designs (for example, Chaparro and Baker, 2003). Time taken to read each article sample will be captured for each text.
- Develop a small question bank for comprehension questions to evaluate whether participants have fully read each text.
- Develop a small likert scale set of questions to prime users and partially evaluate navigability, learnability, and preference.

1.3.3 Objective 3: Pilot test application and survey (5-10 participants)

- Write detailed, specific instructions to guide participants through the experiment (for example, how to navigate through each condition, including keyboard shortcuts).

- A trial run will be conducted with colleagues and acquaintances to identify bugs and review if timings are accurate and the difficulty of comprehension questions. Informal qualitative interviews will be conducted over ZOOM or in-person after each session and used to guide refinements to the study design.
- Develop training materials.

1.3.4 Objective 4: Run experiment and collect data.

- Recruit 40+ online participants via the Prolific.co platform. A budget of €200 will be applied to pay subjects for their time.
- Write a study description which will initially brief participants. This will cover:
    - The study aim (i.e. digital reading in the browser). (NB: this will not explicitly refer to the pagination or scrolling conditions ahead of time).
    - Tasks required. (In this case reading two articles and responding to them).
    - Completion conditions: users will be required to read both articles in full and follow on-screen instructions. (At the end, users will be asked whether they have responded truthfully and completed all tasks, with the proviso that payment will not be withheld regardless of their response).
    - Data usage: how information will be collected and stored.
    - Publication and purpose: the study is being carried out as part of an MSc dissertation.
    - Estimated completion time of the study: 25 minutes.
- Executing the experiment. This research will mirror aspects of the experimental design of the study by Chaparro and Baker (2003). It will use a repeated measures, within-groups design and apply counterbalancing. Dependent variables (i.e. the speed metrics and test scores) will be saved to the SPA backend itemised by participant.

1.3.5 Objective 5: Prepare and process collected data

- Download data and review responses by importing to Excel. Screen out participants who have not answered truthfully, or for whom there is missing or aberrant data.

- Clean and filter the data using R Studio so that it is in a long format.
- Wrangle data, calculate variables and consolidate fields by experimental condition (i.e. scrolling or paging) using R Studio and Microsoft Excel.
- Produce descriptive data (i.e. frequencies) and carry out statistical tests using Jamovi to investigate hypotheses and validate findings.

1.4 Research methodologies

This study uses a concurrent nested design mixed methods approach. A user evaluation experiment to explore the impact of different reading presentation styles (scrolling and paging) was conducted in two phases:

- **A pilot testing phase** with six volunteers and in-depth interviews
- **An online phase** with 42 anonymous participants from the Prolific recruitment platform

Both test phases utilised a within-subjects design where each participant experienced both reading mode styles. The online phase and the quantitative data generated from it represents this study's primary research method.

In the pilot testing phase, limited quantitative experimental data was also generated, along with qualitative data in the form of in-depth interviews carried out over video conferencing or in-person with each participant. Both qualitative data and quantitative data from this pilot test was used to adjust the experimental design. This data was also later used to inform the interpretation stage of this study.

During the online phase, quantitative data was collected using repeated measures (with reading samples in both scrolling and pagination modes). To counterbalance ordering effects, two groups were designated (scrolling-paging, paging-scrolling) in which the order of reading mode was reversed in either group. Additionally, participants were invited to optionally leave comments at the study's conclusion about their responses and preferences, which were then also used in the interpretation phase.

This study will focus on measures identified in previous studies: reading performance and subjective reader preference. A simple comprehension test will be used, alongside attention checks, to assess whether users have followed instructions and fully read each text. (NB: users will not be able to navigate back at this stage).

Our hypotheses can be broken down in terms of their two dependent variables:

- *Reading speed.* Also referred to as reading performance in the literature (Dyson and Kipping, 1998). This is the words-per-minute rate at which participants read a sample of text in a given mode.
- *Reading mode preference.* This metric follows the methodology applied in Rello and Baeza Yates' study (2012) exploring optimal background colours for readability, where participants were asked for their overall preference.

The hypotheses also features a single[2] independent variable (categorical):

- *Reading mode.* This is the presentation and navigation style used to read a particular article sample: continuous scrolling (scrolling), or dynamic pagination (paging).

In addition, the research design involves the collection of additional variables used to validate responses and explain findings:

- *Comprehension score.* For each reading sample and reading mode, participants were asked five comprehension questions which tested their recall or recognition of information and phrases in the article. A minimum score of 0.6 (three questions out of five correct) in either reading sample was set for inclusion in the final analysis.
- *Reading efficiency.* Reading speed multiplied by comprehension score. This method is used in Shaikh, A., and Chaparro, B. (2005). Because of this present study's sample size (n = 38) and the limited precision of the comprehension score (5 questions for each reading mode, with a minimum score of 3 required for screening purposes), this measure is not expected to reach the level of significance.
- *Custom usability score.* At the study's conclusion, users rated four statements on a Likert scale for each reading mode in order to prime them before delivering their overall *Reading mode preference* (*s*ee above). This measure is solely used to help participants recall the different aspects of each mode (navigability, learnability, and physical effort) and weigh them to ensure they

[2] The variability of the article text itself has been controlled via repeated measures and counterbalancing.

give a considered preference, limiting the impact of proximity bias (i.e. tendency to focus on the most recent condition experienced).

1.5 Scope and limitations

Because of the nature of digital reading and feasibility considerations for recruiting participants, this research uses a browser-based online user evaluation experiment. Specifically, this study addresses how pagination and dynamic rendering in web apps can impact the reader experience of long texts (1500+ words).

1.5.1 Scope

While there has been focus in recent years on e-reading devices such as smartphones, tablets, and e-ink readers which this study draws upon, this research is concerned with scrolling and pagination as they apply to desktop-sized computing devices.

Focus is also exclusively applied to lengthy documents (i.e. long-form news articles) which entail sustained, “continuous imaginary reading” (Hillesund 2010). The research’s scope does not include short-to-medium news stories, reference documents, and manuals - all of which are documents that encourage or demand short bursts of skimming and searching, which Hillesund (2010) refers to as “discontinuous fragmented reading”. Nor does the scope include discourse around infinite scrolling and pagination of results in mediums like eCommerce and social media.

1.5.2 Assumptions

As the research needs to be conducted online, the following assumptions are made: a large enough user sample will be recruited to generate statistically significant findings, the evaluation text is sufficiently (but not overly) complex, and users will participate fully (i.e. read to the end, answer truthfully). Several measures have been included to verify participation such as attention checks and comprehension questions, but these may not be exhaustive.

1.5.3 Limitations

There are several limitations that may impact the results. These include: application performance (i.e. whether the app works properly), which may be a product of client-side software or configuration; internet latency, which may be influenced by distance from a server used to host the app; and hardware of study participants, which may have an impact on reading speed due to increased load times. (Note that latency has been partially controlled for by deploying the experiment as a single application which fetches and pre-builds each screen before the participant reaches it).

This study may be subject to potential bias due to novelty in the user interface of the pagination mode. Other digital reading research (Schneps et al. 2013; Mangen et al. 2013) controlled for this by carrying out pre-training with participants on the device or software to be studied. However, this extended level of training (at least 300 minutes over 10 days per participant in the Schneps' study) - or indeed any training - was not practical due to the online nature of the experiment.

1.5.4 Delimitations

Finally, delimitations for this research, which are expected to reduce the impact of confounding variables, include: that participants will be recruited via an online user study platform (Prolific.co) and that they will be screened for browser type (Chromium-based, Webkit, or Firefox browser engines only), device (laptop or desktop only), and language proficiency (native English speaker, UK resident, A-level education). Limiting participants to this location is also intended to minimise latency as the experiment will be hosted on a server in the UK. Further details on delimitations and other aspects relating to the research design are provided in *3.2 Experimental design*.

1.6 Document outline

The main body of this thesis is divided into three chapters:

- *Chapter 2 - Literature review*: This chapter explores existing research across a number of domains, beginning with a short history of digital (and analogue reading), explaining the genesis of different modes of presentation in print and digital texts. It explores findings from domains such as human-computer interaction, cognitive science, and psychology in order to develop an understanding of how digital reading and presentation style might affect working memory and comprehension. Subsection 2.4, *scrolling vs. pagination*, explores methodologies and results from previous work in this area. Lastly, subsection 2.5, *Web standards*, reviews current technologies and standards which may be used to achieve dynamic pagination in the browser.

- *Chapter 3 - Design & methodology:* Here the research design is fleshed out in two stages. First, a reader application built on current web standards is outlined in detail. Secondly, the design of the pilot testing and online phases of the experiment is described (including procedure and other considerations).
- *Chapter 4- Results, evaluation, and discussion*: This penultimate chapter features all results from the online experiment phase, including details on the participant sample and statistical tests used to validate the hypothesis. Finally, results are interpreted and synthesised with findings from previous studies reviewed in *Chapter 2*.

In the final chapter, *Chapter 5 - Conclusion*, a summary of the research is provided along with recommendations for future work.

2. LITERATURE REVIEW

This chapter is divided into five parts. The sub-sections **2.4 Scrolling vs. pagination** and **2.5 Web standards: a changing landscape** are the primary locus of this review and feed directly into the design of the experiment. However, other sections which appear below provide motivation for the research and scaffolding for the alternative hypothesis suggested.

2.1 Digital reading: a short pre-history

The recent evolution of digital reading has paralleled many of the developments in different phases of the history of the written and printed word (Liesaputra et al., 2009).

For example, scrolls (i.e. paper rolls) were originally used to store long texts by Egyptians, Romans, and Aztecs (Liesaputra et al., 2009). Scrolls provided a similar, flowing, vertically-oriented reading experience to that we see on the modern HTML page. Advances throughout antiquity eventually saw the development and rise in use of the codex, an early form of manuscript book, which would later form the basis for the gradual emergence of what we recognise as the modern book following the Dark Ages with attendant features such as word and line spacing, punctuation, nad paragraphs (Hillesund, 2010).

Even though books are often considered an ancient kind of a technology, the modern method of reading printed books has only existed for a few hundred years (Brady et al., 2018) and it has been consistently improved upon over time. “Over the centuries, typographers have refined the design of books, making the printed book an effective reading technology.” (Hillesund, 2010) One crucial milestone in book navigation was firstly the introduction of page numbers, and later the index in the 16th century. With these new methods of search and retrieval, readers were able to skip directly to specific content and access it at their leisure. This allowed a new kind of “discontinuous reading” (Hillesund, 2010). It also marked the beginning of a departure, Hillesund argues, from reading as a primarily linear, focused activity.

In parallel, despite this new non-linear way of using books, Brady et al. 2018 point out that in the centuries since, a certain kind of long-form “printed text meant to be read from beginning to end” became popular along with improved mediums for efficient reading of long-form texts such as the novel.

More recently, a paradigm shift in methods of reading emerged in the 1990s, mediated by hyperlinks and search engines and a flexible new scrolling based presentation mode. This marked a resumption of the trend in discontinuous reading which was initially sparked by book indexes and newspaper formats. On the other hand, the popularity of ebook readers in the late 2000s, which were “designed as a metaphor for [printed] books” so that users could better understand how to use them (Pearson et al. 2010), hinted at an ongoing desire for the continuous, page-by-page style first achieved in the original codex.

Because digital reading is still so new (not even three decades have passed since mainstream adoption), it follows that efficient reading technologies and navigation techniques are still in flux, and the best approaches remain to be conclusively determined.

In the 1990s, the early web began to re-establish the primacy of a scroll-like reading medium - this time with responsive web pages steered by vertical scroll bars. A key limitation of computer screens are their flat, fixed dimensions. The available real estate to show text to readers at any one time is limited. But scrollable documents (and swipeable devices) effectively turn screens into an infinite container for continuous consumption.

2.2 The science and practice of reading on screens

2.2.1 Digital reading and distraction

As touched upon in *2.1*, Hillesund (2010) says that text can be defined by degrees of continuousness (both temporal and spatial) and by thematic connectedness. He characterises two approaches to reading long texts: 1) *continuous imaginary reading*, and 2) *sustained reflective reading*.

For example, novels are read in a continuous mode, “from beginning to end”. This form of reading requires a level of fluency - the ability to read with both speed and accuracy. He says that this kind of concentrated reading is “typically done on paper, especially [the] reading of long-form text”, pointing to the “uninterrupted reading” involved in lengthy magazine or journal articles as “an example of continuous reading” (Hillesund (2010).

By contrast, multimodal or sustained reflective reading “is characterised by discontinuous and often treacherous reading”. This, Hillesund says, is how certain readers (e.g. students and academics) use a textbook or the internet for research: “browsing and skimming” in a fragmented way, reading in parts and out of order, underlining and annotating. Websites and apps add extra efficiencies (and distractions) to this method of reading through search functionality and hyperlinks.

By contrast, the challenge in digital reading of long-form texts lies in the need “to replicate conditions for continuous imaginary reading,” Hillesund argues (2010).

It is nearly impossible to become “immersed in hypertext or online reading in the same way we get lost in a book,” says Anne Mangen (2008, cited in Hillesund, 2010). It requires more mental energy, she says, to resist distractions and remain focused on a text than the mental energy needed to discontinuously click the mouse and rekindle our attention with something new. This creates a type of online reader behaviour which Hillesund (2010) says is dominated by “shallow reading”.

By contrast, the kind of reading this present study focuses upon (long-form), requires a level of immersion. Borrowing Hillesund’s description of immersive imaginary reading, readers should be “involved in [the] story, conjuring up vivid images of persons and places, living through situations, empathising with characters”. To facilitate this, the user interface should also eliminate distracting elements, providing a built in “read-mode” for continuous reading. The digital experience, Hillesund adds, needs to make the text and the reader’s imagined representation of it “the primary focus of attention, [leaving the] technology [to fade] into the background.” (Hillesund 2010)

2.2.2 Screen, scrolling, and memory

When we read something on paper (or on screens), in order to keep track of what we’re reading, we model its structure in our memory. This reliance was discovered, say Brady et al. (2018), through research by Ernst Rothkopf and others who found that humans have an incidental memory which spatially maps the location of information in texts. Further, when we read a book, its very physical properties, says Liesaputra

(2010), “[provide] a frame of reference for navigation because relative positions in the text, such as before and after, have tangible physical associations.”

This is thought to be mediated by working memory, which is the brain’s capacity to temporarily store information. Working memory allows humans “to guide behaviour with information that is not present in the immediate environment” (D’Esposito and Postle, 2015). When we read on paper, this applies to words and ideas not only on the current page, but also to content which has vanished from our peripheral vision and into the previous pages. Nevertheless, according to research by Payne and Reader (2006), even when we read from screens our brains are actively constructing a “structure map” of the content which persists when text moves off of the immediate screen or we navigate to different documents. Thus as we read, meaning becomes anchored to structure (Jabr, 2013).

When working memory is applied to reading, it involves distinct “spatial markers” between typographic elements and other means of orientation which help us to remember what we have read” (Brady et al. 2018). It follows, according to Brady et al., that when these markers shift around as a reader scrolls, readers may later have a reduced ability to remember what they read or to locate previous passages.

So is reading on paper necessarily any different? Returning to Liesaputra’s (2009) account of reading, when we use a book in a physical environment, manipulating real pages, the brain receives visual and tactile feedback which enhances its retention of short-term memory about the semantic structure of the book’s content. This can be thought of as a topology, or the way in which we mentally map physical terrain (Jabr, 2013) or as a kind of “visual cue” (Li et al., 2013). The same memory facility which allows us to map and navigate our immediate environments helps us to remember where words and ideas are located on each page.

But, due to the transient nature of digital representation, this relationship does not necessarily map well to screens. Research indicates that people struggle to “construct an effective cognitive map” on digital devices because of the missing contextual cues normally found on paper documents and books (Li et al., 2013). Screens don’t really have a topology: they’re simply a flat viewing portal. When you scroll lengthy documents on a screen, you can only see a tiny portion of the whole at any one time, and your knowledge of structure and length is usually mediated by a thin scroll bar in your peripheral vision. It's difficult to step back and grasp a document as a whole, contiguous thing. When we navigate from one area of a document to another on-screen, the previous area effectively disappears into the ether with little or no residual trace. This format requires us to map and constantly update a whole additional virtual cognitive representation of the full page in our working memory just to keep track (Sanchez and Wiley, 2009). Moreover, when the page is scrolled, it also moves in a protean, often janky, manner that requires us to use fine motor skills in coordination with our eyes to keep us on target.

Thus while in a meta-analysis Singer and Alexander (2017) found that students reading long texts tended to do worse on comprehension tests while reading on screens as opposed to paper, they found no such effect in shorter texts. They speculated that this might be due to increased scrolling required in longer texts and effects which that might have on comprehension. On a similar note, Brady et al. says that because text on a scrollable document isn't fixed, scrolling may result in worse performance on comprehension tests because readers have "to remember content rather than map content". By contrast, those who do less (or no scrolling) "may have a lighter cognitive load" because they can rely on spatial memory to map the content (Brady et al., 2018).

In a study by Sanchez and Wiley (2009) which looked at how scrolling and paging formats affected comprehension, the authors suggest that because scrolling pages don't have static markers, readers may have to "reallocate resources from the comprehension process to maintain their surface memory for the text," which might cause increased load on the reader. They add that since scrolling while reading a complex text requires both comprehension and the maintenance of a surface representation of a text (i.e. the parts of the text which overflow offscreen above or below the current scroll position), this may increase the cognitive load on readers and stretch their working memory capacity (WMC).

Thus in their study, Sanchez and Wiley (2009) found the scrolling format resulted in reduced learning from texts in individuals with lower WMC. Though they found little to no impact in learners with higher WMC.

Nevertheless, they argue that participants with WMC "[might] show the largest decrease in comprehension caused by scrolling" since they may be less capable of creating a "surface representation" of the document. Two explanations they pose for this are: 1) that some readers “may have difficulty controlling their attention” and are more prone to become disoriented while reading from screens (i.e. lose their place), and 2) that scrolling texts may inhibit the “consolidation or integration” processes afforded by page breaks which enable a “wrap-up process critical for comprehension”. (Sanchez and Wiley, 2009)

Likewise in an experiment by Brady et al. (2018) which examined whether scrolling disrupts comprehension while reading on screens compared to paper, they found only a partial effect specific to participants who had no preference for either digital or paper readings. (Those with no specific preference performed worse on a comprehension test in the scrolling mode). Brady et al. suggest that this, in line with other studies, indicates that “the [negative] impact of scrolling on reading processes and comprehension” may have declined over time because users are more familiar with scrolling and the user experience has improved. That is to say that because many readers are familiar with and highly skilled at navigating the web, Brady et al. argue that they “may have developed compensatory strategies” for dealing with the transient properties of on-screen documents. These strategies might include mapping words onto other content on a page (also subject to scrolling) which forms part of the layout or a prominent component like a heading or image.

2.2.4 In defence of scrolling

One key issue with ebook readers and other devices that present documents in a paginated style, Pearson et al. argue (2010), is that text reflowing is often troublesome and requires more work from the user. It’s difficult to fit text to a screen which may have different proportions depending on the exact consumer device being used. Whereas this kind of text fitting is easily solved for in responsive reflow in web documents with a scrollbar that can grow infinitely.

One factor to consider is that pagination can break up the text and enforce an intermediate action on the user before they can continue. In fact, Nielsen (2013), a researcher in the field of human-computer interaction and user experience, says that linear content flows like web articles “should almost never be broken up into multiple screens”. He says that it’s preferable to place a full article on a single long screen, rather than “inflict the pain of additional steps on users”.

This account of pagination is consistent with the MPA (multi-page application) style of pagination of how news media websites used to break up long-form articles. The so-called "pain" of pagination with this older pattern is that when users click a button to advance the page, they firstly lose their scrolled place on the current page because it effectively vanishes, potentially dislocating its place in the structure map they are building in their working memory. Secondly, they see a flash of white and perhaps an extended pause before the next page loads. And then the "new page jumps into view" (Web Incubator CG, 2022). As is common with nearly all MPA websites, users will see page "jank"[3] as the various elements in the DOM[4] of the new page are populated and the CSS styles are applied to the layout. Lastly the page is reconfigured, perhaps at a disorientating new scroll position. All told, this creates a jarring and disconnected experience which increases cognitive load for the user. (Web Incubator CG, 2022)

As we will see in *3.1 Application design*, this present study proposes a solution to these usability issues to deliver a kind of pagination that does not inflict any "pain" on the user.

2.3 Pagination vs. scrolling in research

There is a significant body of research into the presentation of digital documents in scrolling and paging reading modes which reaches back into the 1980s and earlier. However, comparing current-day devices and design patterns with historical results begs caution.

2.3.1 Performance and preference

Dyson and Kipping (1998) point out that "as late as 1990" one research study was still using screens which "took nine seconds to refill in a paging mode". Other research reviewed by Dyson and Kipping from this time imposed restrictions on users such as continuous scrolling (where the page autonomously advanced down at a fixed speed). They add that, at the time of writing (in 1998), there had been no other recent empirical work which "compared reading performance using current paging and scrolling mechanisms".

[3] "'Jank' is any stuttering or choppiness that a user experiences when there is motion on the screen—like during scrolling, transitions, or animations." (Maust, 2015)
[4] Document Object Model (DOM). A tree and API which holds the structure and content of a web page in memory.

Therefore we will use Dyson and Kipping's paper as a starting point and focus on research which is more relevant to the characteristics of modern computers screens and interfaces.

Dyson and Kipping's 1998 study looked at two features of reading on screens: scrolling versus paging, as well as line length. In their experiment, they recorded time taken to read a sample document and administered comprehension tests which asked participants whether certain questions were answerable based on the document. In one of their experiments they found a difference between scrolling and paging conditions which indicated readers read faster while paging. However, the difference fell just shy of the level of significance they had defined ($p = 0.051$). Additionally, the study instructed users to scroll by tapping the down arrow on their keyboard rather than using a mouse, which the authors say may have contributed to a slower reading rate whilst scrolling.

One interesting finding in this study is that:

> Participant's perceptions of ease of reading, whatever their interpretation of this variable, do not correlate with their performance (1998 Dyson and Kipping).

That is to say that while people actually tended to read faster in certain conditions (such as line lengths over 100 characters per line), later when asked for their subjective response, they often said that a different mode of reading was easier (such as a much shorter line length).

In another study which built on Dyson and Kipping's work, Chaparro and Baker (2003) compared reading speed and comprehension in 15 participants in passages of approximately 700 words. This study featured three conditions: a fully-scrolling mode (single page), a scrolling mode spread over 2 pages, and a paging mode spread over 4 pages. The study also included comprehension questions (five multiple-choice questions per passage), as well as five search task items requiring participants to return to the passage to find a piece of information. Unlike Dyson and Kipping's earlier experiments, Chaparro and Baker found a large and significant difference ($p < .05$) in reading time between passages. In the paging condition, the mean reading completion time was 225.25 seconds, compared with 205.52 in the scrolling condition. The paging condition also resulted in a higher comprehension score (3.2 questions correct versus 2.6 in the scrolling condition), but this did not reach the level of significance ($p = .31$).

Nor did they find any significant difference in perceptions between the conditions in a perceived ease of use questionnaire. Chaparro and Baker suggest that a difference in scrolling versus reading times between theirs and previous research may be because, as of 2003, participants had had more exposure to scrolling documents on the web.

2.3.2 Physical demands and feedback from scrolling and paging

So how does scrolling or paging impact users physically?

In a study which examined interactions between memory and digital reading, Wästlund (2007, as cited in Brady et al., 2018) found that participants in a scrolling mode reported more stress and mental fatigue whilst using the mouse than those using the keyboard to navigate page by page. Additionally, Wästlund found that when a page's layout is adapted to fit perfectly on the screen, it produced less mental workload in readers.

Pagination can also be physically cumbersome for users. Where the "next" button is placed is "critical to the ease in which" digital reading devices like ebook readers are used, say Pearson et al. (2010). Awkward placing can be uncomfortable, "[potentially leading] to Repetitive Strain Injury (RSI)".

Nevertheless, paged reading may require less consistent physical input. "Turning the pages of a novel is very different," say Vanderschantz et al. (2018), than skim-reading and browsing to find snippets of information. Continuously reading a novel or a long article may demand only infrequent button presses as one reaches the end of each page.

Vanderschantz et al. add that "turning pages on a physical book is fast and intuitive", providing readers the opportunity to anticipate page turns by lifting the corner of the page before they have even finished reading the current one (O'Hara and Sellen, 1997 cited in Vanderschantz et al., 2018). Such affordances "are impossible to recreate on digital equivalents", say Pearson et al. (2010). And yet, at the time of writing this present study, mobile devices have innovated considerably in the past decade in the areas of gesture-based navigation, providing both visual and haptic feedback. Indeed, past research by Liesaputra (2010) has shown that properties related to physical books (page turning and bookmarks) can be emulated in engaging ways that improve user experience and performance at tasks like seeking out information.

Further details of previous digital reading experiments which explored paging and scrolling are discussed in *3.2 Experimental design and procedure* and in *4.2 Discussion*.

In the next section we look at how the core feature related to the book mental model, pagination, might be implemented using current web standards.

2.4 Web standards: a changing landscape

While dynamic reflowable pagination (often called "paging") is widely implemented in ebook readers and so-called paged media documents (like PDF), it is rarely used in web design. "Reflowable pagination" refers to the presentation of documents so that when one page (or container) is filled with text, excess text "flows" into another page dynamically. This means that different sized containers (for example, computer screens with bigger or smaller dimensions) will reflow text and generate a variable page count depending on the device or configuration used.

The implementation of dynamic pagination in web documents closely parallels the history and technical challenges of implementing separate columns of text in web pages, which we will explore in this subsection.

In the early days of the World Wide Web in the 1990s, web documents could only contain "static information and limited formatting" (Laiola Guimarães et al., 2014). Within a few short years, CSS (Cascading Style Sheets), a syntax language for styling web pages, was introduced. However, for a long time, CSS lacked the ability to define document layouts which could reproduce the advanced typesetting and layout features of traditional paginated print media.

2.4.1 AJAX and JavaScript

JavaScript, a programming language for adding dynamic behaviour to web pages, emerged at around the same time that CSS was introduced. Later releases of JavaScript would see the introduction of AJAX (asynchronous JavaScript and XML), a technique for requesting page updates from a web server without the client needing to reload the page.

Throughout much of its history, CSS lacked features which could manage text flow between layout boxes in a dynamic way to support pagination or text columns. For example, during the 1990s and 2000s, there was no way to declare that excess text in one container (such as a column or a <div> content container modelled on a "page") should flow into a second container when the amount of text overflowed the first.

This feature, which is common to all desktop publishing software, was eventually solved for in myriad ways by JavaScript developers who wrote scripts to overrule the browser's natural way of rendering text and layouts. For example, one solution called *lib_columns.js* (Van Ouwerkerk, 2005) was a JavaScript library which used a relatively simple method of chopping innerHTML (i.e. entire containers of text) into separate <div> elements. Another plugin based on the *jQuery* library, *Columnizer.js*, went a step further and allowed further customisation in the areas of column width and the number of columns.

However, these kinds of JavaScript solutions had attendant problems:

1. Firstly, they require additional code to be included on every webpage where column support is needed.
2. Secondly, they represent a non-standard way of doing things which overrules the browser's way of rendering layouts, and thus may be less performant. This is because JavaScript that reforms a page's layout can affect the "pixel pipeline," the staged process through which the browser interprets HTML/CSS/JavaScript and "paints" it on-screen (Lewis, 2015). This in turn might introduce so-called "jank" and "negatively [impact] the user's experience" (Lewis, 2015).
3. Thirdly, they can lead to inconsistent behaviour across browsers. For example, one developer reported that both l*ib_columns.js* and *Columnizer.js* can split text in inconsistent ways across browsers.
4. Finally, the inclusion of popular JavaScript libraries such as *jQuery* can entail security debt and potential vulnerabilities. Libraries like *jQuery* have indirectly resulted in widespread security vulnerabilities on many websites (Lauinger et al., 2018) due to the introduction of dependencies and a failure by maintainers or users to update packages. Moreover, many features of *jQuery* have been superseded by recent improvements in the core JavaScript standard (ES5 and ES6).

Another way to implement reflowable pagination is through HTML5 Canvas (MDN, 2022a). Unlike other JavaScript legacy methods for creating pagination or columns, the JavaScript Canvas API provides a relatively performant means to "paint" text or images on screen. However, a key drawback of Canvas is that it does not follow the normal interaction paradigm of HTML text (by default, Canvas text cannot be highlighted).

One proposed solution found on the Stack Overflow question and answer website[5] suggests using the scalable vector graphic (SVG) <text> tag to render text (Mulder, 2019). However, this HTML tag is designed for representing text in vector graphics (i.e. typically in document images), not for full text document representation.

Lastly, while it does not solve for page reflow, JavaScript does provide another feature which can enhance the experience of traditional pagination: AJAX. For example, AJAX, as implemented by the JavaScript API, allows the client (i.e. the browser) to send requests to the web server for new information without needing a full page load. This enables alternative architecture to the previous web paradigm which entailed full page reloads when navigating between extant pages in an article. Instead, AJAX facilitates the creation of single-page applications (SPA), which results in an app-like experience on websites because only necessary component parts of the layout get updated and rendered when, for example, the user advances to a new page.

Moreover, a major benefit of an SPA approach for reading experiences is that it provides little or no delay between page loads, and much less of the characteristic "jank" associated with most MPA websites.

2.4.2 CSS

Cascading Style Sheets (CSS) is a language "used to format the layout of web pages" (Hissom, 2011). It combines with other languages (HTML and JavaScript) which are used respectively to define the content and behaviour of web pages.

First released in 1996, CSS has enjoyed an open-ended, iterative process of standards development (known as CSS3) since 1999. Its ongoing, continuous development involves completing and releasing segments of the standard in chunks (Kyrnin, 2020), also known as modules. In parallel, browser engines (such as Blink or WebKit) are

[5] https://stackoverflow.com/a/23556701/18648429

continually (albeit slowly) updated by vendors like Google and Apple to adopt features and behaviour defined in new and continually revised CSS "modules". One such module relevant to this paper is *Multi-column Layout*, which enables the aforementioned text reflow functionality that is required for defining columns, but also potentially for achieving dynamic pagination in web documents.

The original specification for CSS 2 in the 1990s distinguished between two media groups for text: paged and continuous (World Wide Web Consortium, 2011). Paged refers to the mode in which a document has discrete pages and margins - typically documents intended to be physically printed on paper. Continuous refers to media which appears on screens, where the content flows continuously (i.e. a scrolling presentation style).

Proposals to include multi-column layout styles in CSS date back to the first rough sketches of CSS3 in 1999 (World Wide Web Consortium, 1999). This first sketch of the column standard did not specifically address reflowable pagination. But as this area continued to be developed through further draft iterations of CSS3, specific modules have been defined to potentially address different kinds of layout styling relevant to pagination and reflowable text, including: *Paged Media, Generated Content*, *Multi-Column Layout*, and *Fragmentation*.

CSS3 Fragmentation Module Level 3 (World Wide Web Consortium, 2021) is a Candidate Recommendation as of May 2021. It specifies the process "by which content becomes broken up into different boxes", allowing continuous content (such as text) to reflow between those boxes. Specifically, it adds CSS attributes to define how and where breaks occur, and how widows and orphans should be handled. (At time of writing in 2022, however, CSS3 Fragmentation is not fully supported by any major browser).

The *Paged Media* (World Wide Web Consortium, 2018) and *Generated Content* (World Wide Web Consortium, 2014) CSS Modules are two specifications relevant to styling documents as pages. However, the definition of paged in these modules relates specifically to the generation of static documents which can map directly to print media (print previews, using HTML to generate PDFs), and both modules remain (at time of writing) in Working Draft status with only missing or limited support across browsers. Kimber (2018) points out that, as of 2018, there is no "sufficiently-complete open-source solution" for print media pagination in CSS - only commercial tools. He adds that the CSS specification for pagination was still in flux at this time: underspecified or with features missing.

One other relevant CSS3 module is Regions - however, as of 2020 (Andrew, 2019; Deveria, 2022a), there are currently no usable implementations in any of the major browsers , and the 2014 release of the specification remains as a Working Draft.

2.4.3 Pagination libraries and implementations

As of 2022, several commercial and open-source libraries or software packages exist to implement dynamic pagination in web applications. These are summarised below:

- **Readium**[6]. A comprehensive architecture and set of toolkits (Readium 2022) for developing desktop, web, and mobile reading applications designed for .EPUB and .CBZ formats catering to ebooks and digital comics. The toolkit includes a JavaScript-based viewer for online books. Readium also makes use of the CSS3 multi-column format (Readium, 2019), recommending this approach because of its cross-platform support and responsiveness.
- **Epub.js**[7]. Like Readium, this library provides an architecture for rendering .EPUB content and is used to power many other applications, including a number of implementations using ReactJS.
- **Vivliostyle Viewer**. This software is also designed to take HTML content and output it as pages via a web reader. It uses JavaScript to render content directly.
- **JavaScript libraries and scripts**[8]**:** *Lib_columns.js*, *jquery.easyPaginate.js*, and *Columnizer* jQuery Plugin. These libraries allow users to create reflowable columns (i.e. similar to CSS3 Multicol) and/or to automatically paginate content; a proposed benefit[9] of this solution is providing more control using the DOM. Note however that these packages are non-standard and in some cases no longer maintained (*lib_columns.js* was released in 2002), and many do not support images or certain CSS styling features.

[6] https://github.com/readium/readium-js-viewer
[7] https://github.com/futurepress/epub.js/blob/master/src/layout.js#L107-L110
[8] http://13thparallel.com/archive/column-script/lib_columns.js
http://st3ph.github.io/jquery.easyPaginate/
https://adamwulf.me/columnizer-jquery-plugin/
[9] https://stackoverflow.com/a/44491305/18648429

- **Custom JavaScript for splitting and reflowing**. Several examples and solutions[10] [11] [12] were found during this review on StackOverflow. These approaches typically involve splitting an article into a series of word chunks (using a space as a separator), and then rendering each word to the page until the current container/viewscreen is full. However, several issues persist with these approaches:
    1. **1.** They are not standard, and involve reimplementing features already present in CSS or the native JavaScript API.
    2. **2.** In sample code, reflowing was inconsistent (sometimes, while navigating back and forth between pages, words fall on a different page or are repeated both at the end and beginning of adjoining pages).
    3. **3.** This approach struggles with images and may break other types of CSS formatting. One advantage, however, is that it would enable streaming, i.e. dynamic loading, via AJAX.

Nearly all existing implementations of pagination on the web are designed for ebooks using the standardised EPUB file format. Though EPUBs were originally conceived for use on dedicated readers, many browser implementations have now been created. Several use *Epub.js* for rendering the books with a React-based UI as a wrapper such as react-pub-viewer[13] or react-epub-reader[14]. While these implementations were very similar to the paginated style required for this present study, each package included a significant amount of overhead for rendering EPUB and are not compatible with the experimental library required to deploy this study.

2.4.4 An elegant solution

Finally, an article about new features in the then recently revised CSS3 Columns standard (Mirza, 2013) offers a basic description of how CSS and JavaScript can be used to achieve the dynamic reflowable pagination necessary for this present study.

A key benefit of the approach suggested by Mirza compared to the others reviewed is that it is a largely CSS-based approach and is highly performant. Only basic HTML

[10] https://stackoverflow.com/questions/20690834/text-pagination-inside-a-div-with-image
[11] https://stackoverflow.com/a/52194914/18648429
[12] https://stackoverflow.com/a/23640207/18648429
[13] https://github.com/altmshfkgudtjr/react-epub-viewer
[14] https://github.com/alex1504/react-epub-reader

and CSS and the browser engine are used to render and reflow all text. The JavaScript used is only required to programmatically "scroll" through the content. Moreover, this approach requires minimal code and maximises its use of current CSS standards.

Because CSS is processed before JavaScript in the "pixel pipeline" (Lewis 2015), this means that the browser parses both HTML and the CSS to calculate element proportions and spacing, constructing the layout before it does any JavaScript transformations. Moreover, "waiting to obtain CSS doesn't block HTML parsing or downloading," but it does block JavaScript. (MDN, 2022c) Thus by implementing reflow for pagination using a native CSS feature rather than using supplementary JavaScript code, pages using this mechanism should be rendered more quickly than other predominantly JavaScript-based text reflow techniques.

This approach, however, does generate overhead because the full document needs to be loaded into memory. Using a different approach via JavaScript and a database would allow an article to be streamed on-demand as the reader pages through it, thus reducing bandwidth. However, because the documents being loaded for this present research's area of focus are news articles (5000 words is roughly equivalent to 10kB), the CSS3 Columns approach of preloading the full document is superior since the vast majority of consumer devices can easily handle this memory overhead. This solution and other considerations in its deployment will be covered in more detail in section *3.1.1 Implementing pagination.*

3. DESIGN & METHODOLOGY

This chapter is divided into two major sections. In *3.1 Application design*, development of the web application and its attendant features and properties is described in detail. Following that, in *3.2 Experiment design*, an account is given of the research methods and procedures deployed for this study, including some preliminary findings from the pilot testing phase which were fed back into a modified final version of the application and experiment for the online phase.

3.1 Application design

In the first subsection of application design below (*3.1.1 Implementing pagination*), the technical challenge and solution is outlined. In the remaining sections (*3.1.2 - 3.1.5*) several design considerations are addressed regarding readability (for example, background colour). While participants were not exposed to different conditions for these features in the experiment, each one was optimised based on previous research in order to limit the impact of confounding variables and to maximise reading experience.

3.1.1 Implementing pagination

As discussed in *2.4.4 An elegant solution*, this study uses the CSS Columns feature for pagination. It does so using a simple series of manipulations suggested by Mirza (2013) which have been implemented using the CSS and JavaScript documented in this section.

The current specification for CSS columns (which is now supported by all major browser engines), allows developers to set a number of columns with a predefined width for each:

```
#article {
  column-count: 1;
  column-width: var(--article-width);
  max-width: var(--article-width)
```

Note that in our example above a CSS helper variable (--article-width) has been used. This was set using the min() CSS function to provide a fallback for smaller screens that are narrower than the article container[15]:

```css
:root {
  --article-width: min(650px, 92vw);
}
```

When a CSS column is defined with the above properties, the text flows inside the column vertically (top to bottom), extending the page downwards and generating a vertical scroll bar in the usual way as follows:

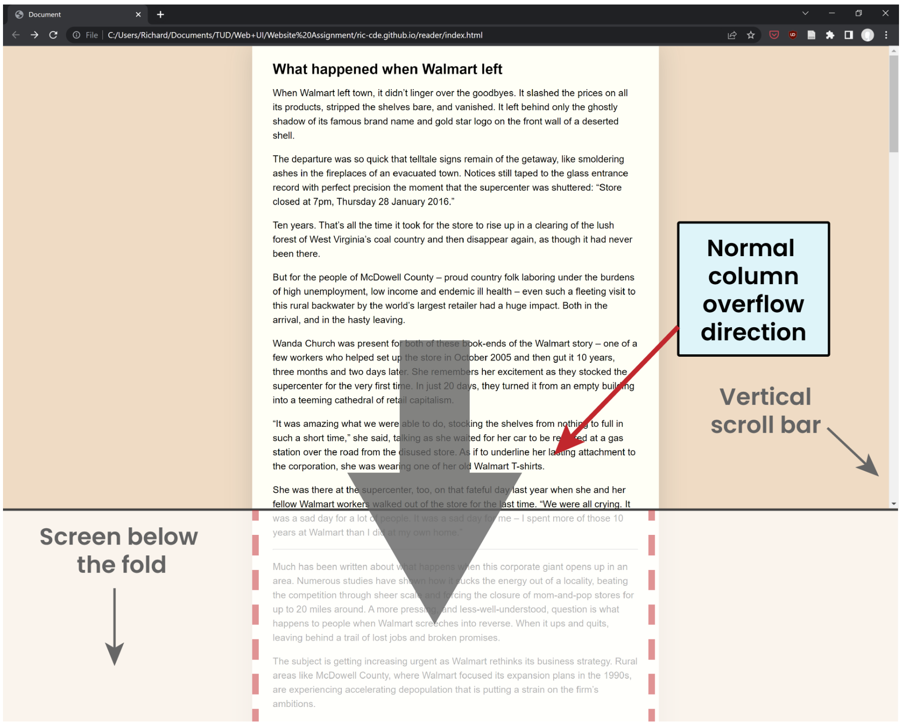


Figure 3.1: Article column container with default overflow.

This is identical to how text in a normal HTML container flows when it has a fixed width. However, we can get a step closer to pagination by adding a fixed height to the article container[16]:

```css
#article {
  /* ... */
  max-height: calc(100vh - 3.2rem);
```

[15] This maximum column width was set based on a widely recommended line width of 65-85 characters-per-line.

[16] Note that Safari, which uses the WebKit browser engine as of 2022, currently implements this feature slightly differently to other major browsers. While in other browsers a relative height in the container such as `max-height: 100%` and a fixed height in a parent container creates an inline overflow, Safari requires an explicit unit value for the height in the column container to achieve this.

Because the container now has a height limit (which is less than the viewport height), the scrollbar disappears and text can no longer flow vertically below the article container. Instead, the text is flowed into new columns which are populated in an "inline" or horizontal, left-to-right direction. Note that even though we originally specified `column-count: 1`, the browser engine generates additional column-like containers (which match the fixed column-width that we specified) for the text to flow into so long as the container's overflow value is set to its default of `overflow: visible`.

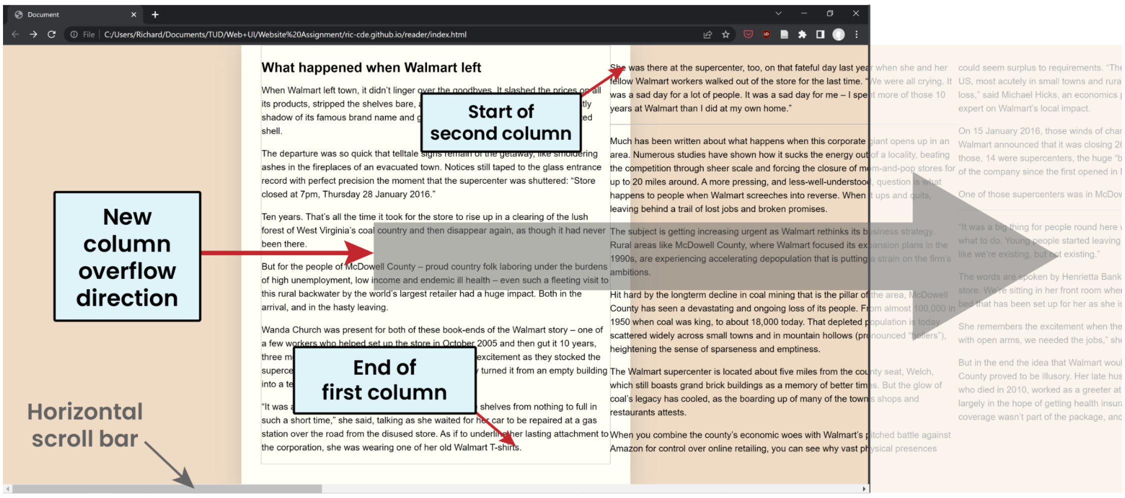


Figure 3.2: Article container with inline (horizontal) columns overflow.

However, if the article container's `overflow` property is set to `hidden` (while also reducing the gap between our overflow columns to keep columns flush):

```
#article {
  /* ... */
  column-gap: 0px;
  overflow: hidden;
```

The inline overflow columns disappear and we are left with a single "page" (see: Figure 3.3 below). The previous overflow section of our container element, which extended to the right and could be scrolled, still exists in the DOM. However, this overflow section (along with the scrollbar[17]) is now hidden from and inaccessible by the user because it extends beyond the bounds of the container.

[17] Note that when the scrollbar is hidden, the container cannot be scrolled. Mousewheels that allow horizontal scrolling and the mouse middle-button "auto-scroll" feature do not work.

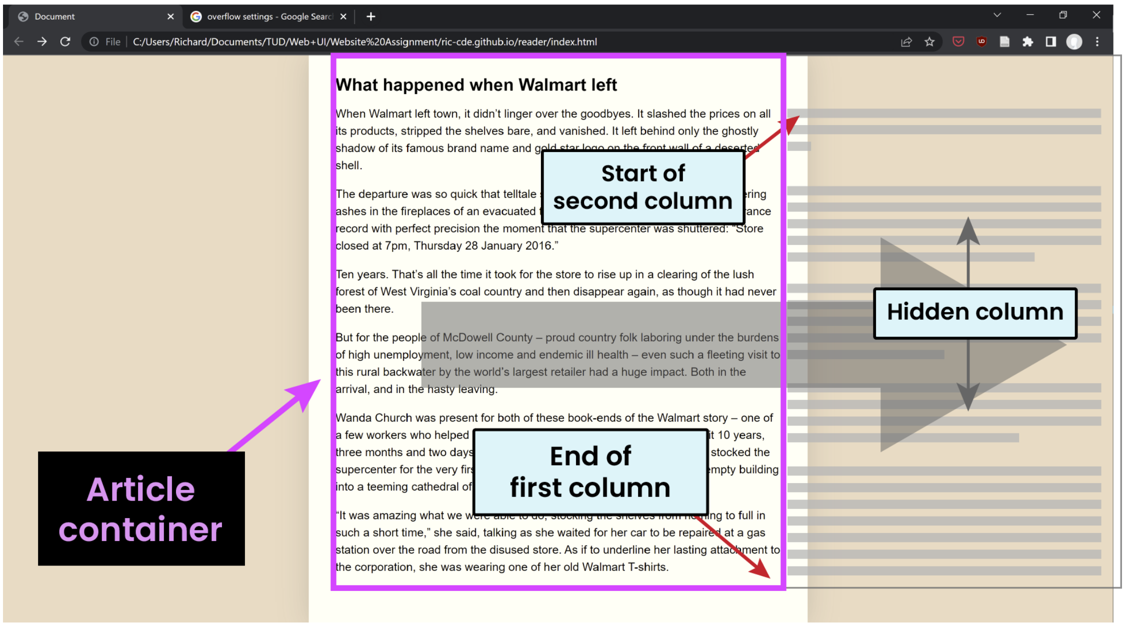


Figure 3.3: Article container with fixed height/width and overflow:hidden.

Each HTML container on the DOM has a `scrollLeft` property which can either get or set how many "pixels an element's content is scrolled from its left edge" (MDN, 2022b). Each element also has an `offsetWidth` attribute which returns its width in pixels. In this case, the article container is the same width as the CSS variable defined previously (650 px); however, `offsetWidth` can be used to access this dynamically in case the container size is updated[18]. Lastly, an `offsetWidth` property tells us the full width in pixels of all columns or "pages" including both the visible and overflow areas. Dividing this by the `scrollWidth` value returns the total number of generated pages and can be used to calculate the current page number.

From here, "next" and "previous" buttons are added along with event listeners with callback functions that increment or decrement the container's `scrollLeft` attribute so that overflow columns (i.e. adjacent "pages") can be moved into view.

```
<!-- HTML -->
<button id="next" class="nav_button">Next →</button>
```

[18] During testing a bug was discovered in how this value is set on certain screens with pixel scaling. Occasionally, adding a whole number value like 650 to `scrollLeft` results in the property being incremented by a fractional number like 649.636352 while using Chrome and Windows 10. After many page turns, the rounding error accumulates and pushes the edge of the current page fractionally out of bounds. However, this error, when it occurred, was negligible and no pilot testers or online phase participants reported seeing it.

```
// JavaScript
const article = document.getElementById('article');

// Retrieve the next button
const next = document.getElementById('next');

function nextPage() {
    article.style.animation = '';

    // Incr. article element's horizontal position & show next page
    article.scrollLeft = article.scrollLeft + pageWidth;
    calcCurrPage();
    article.style.animation = 'fadeIn ease 0.5s';
}
```

A number of additional functions and event handlers to track page progression, render a progress bar, and govern what happens when the user resizes the screen have been defined in the JavaScript code for the reader. Many CSS styles were also defined to, for example, maintain consistent margins across different screen sizes and other aesthetics in order to improve user experience.

To increase user experience and the efficiency of page turns, additional keyboard shortcut hotkeys were added to the reader using the `document.onkeyup` event handler via the Document API. A callback function for the `onwheel` event handler of the article container was also defined so that scrolling via the mousewheel in pagination mode carries out page turns[19].

The code for all features above can be viewed in Appendix A of this document.

Jump to Appendix A ↓

View live demo on Github.io | https://ric-cde.github.io/reader/msc

View Github repository | https://github.com/ric-cde/ric-cde.github.io/tree/master/reader/msc

Note that vendor prefixes such as -webkit and -moz (e.g. -webkit-column-count), originally required in some browsers when the CSS column feature began to be

[19] During testing, this mousewheel-mediated page turning feature proved very user friendly and efficient since it builds on the existing mental model and design pattern that web users expect when navigating texts. Additional visual feedback is issued in the application to help signpost this behaviour for new users who try to scroll. For each page turn, the entire text updates instantly, the page number and the progress bar reflect the change, and a gentle fade-in animation is applied. Additionally, it still allows the user to quickly "scroll through" an entire article fluidly from beginning to end without any intermediate delays, just as efficiently as scrolling in a normal web document.

released in 2011, have been omitted as they are no longer needed in all major browsers (O'Leary, 2021).

As part of this approach, the browser client receives all article text from the server, storing it in memory and rendering the hidden text during page load. Because the additional columns have already been rendered in this way, turning pages is instantaneous.

A potential drawback to this approach, however, is the extra bandwidth overhead during initial load and indirect wastage when a user does not navigate through the full document. In the case of a full book, this might entail several hundred kilobytes or even megabytes.

For this use case, a streaming-type method deployed in an SPA was considered where the server would send a buffer of text sufficient to populate several pages for any screen dimensions. As the reader advances, the client could progressively issue AJAX requests to the server for more text to be added to its buffer so that the user experiences little or no loading pauses.

However, for a long-form article (e.g. 5000 words), loading the full article text typically requires no more than 20-30 kB. (By comparison, loading a normal long-form article on the NewYorker.com website required 7.9 MB of content to be transferred, including JavaScript packages, ads, images, and more). Further, this kind of bandwidth minimisation was unnecessary because the version of the article presented in the experiment did not include any images[20] and the study application pre-fetched and prepared all content before readers reached reading sample screens.

While testing the initial code, it was also observed that the code did not work on the Safari browser (the reader stalled on the first page). Safari uses the Webkit browser engine, which implements CSS3 Columns in a slightly different way where columns need an explicitly fixed width. When this property was added, the reader worked normally in Safari. Additional tests were carried out on browsers that use the Blink/Chromium (Google Chrome and Microsoft Edge) and Gecko (Mozilla Firefox) engines. Though CSS Columns is supported on virtually all other browsers, including

[20] If this reader implementation were to be used in production with images, lazy-loading on images could be used to preserve bandwidth. However, images would need to have a fixed size - otherwise lazy-loading of images could cause text overflow into new pages and increase the page count dynamically as the user progresses, resulting in poor user experience.

versions of the older Internet Explorer browser (MSHTML engine), testing was waived on these engines due to a combination of deprecation (end of life), low market share, and lack of or inconsistent support for CSS features.

3.1.2 Navigation considerations

The reader interface utilises a subdued, minimal navigation bar:

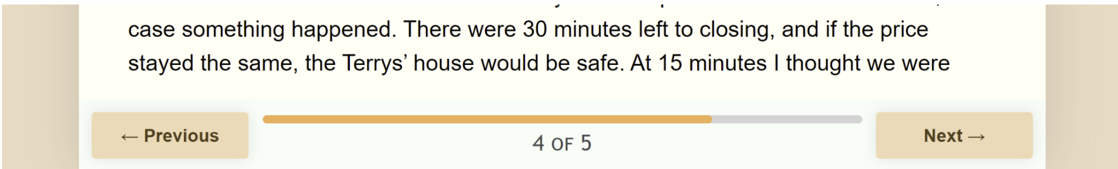


Figure 3.4: The reader's navigation bar in paging mode.

The navigation bar was scaled down in order to both maximise the available vertical space for lines of text, and to reduce its visibility in the reader's eyeline and thus minimise the potential for distraction from the text.

In fact, a more minimalist approach was considered (with a greyscale colour palette and shorter height). However, as the interface would be relatively new to participants in the experiment (as opposed to their default method of scrolling and using the scrollbar), the use of larger touch points and a peach colour to indicate active buttons and progress was retained in order to reduce any cognitive overhead.

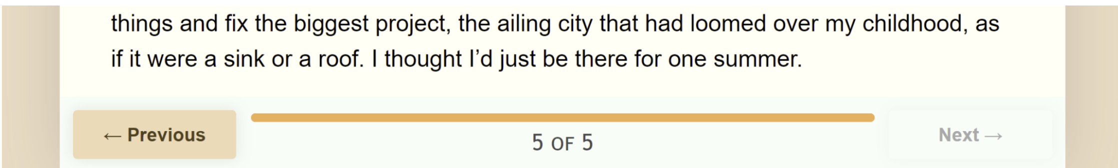


Figure 3.5: The reader in paging mode at the end of an article..

3.1.3 Typography and aesthetics

Background colour

The reader uses a cream coloured background behind the text, paired with a muted peach coloured backdrop to either side of the text.

To maximise contrast, black text on a full white background was considered. However, previous research indicates that readers, particularly those with dyslexia, prefer or are more efficient readers when text is presented with a cream coloured background (Rello and Baeza-Yates, 2012; British Dyslexia Guide, 2022).

Nevertheless, the body text remains very high contrast (20.9:1 contrast ratio) in order to maximise readability for those with reduced vision and to ensure a consistency of viewing experience on screens with limited contrast ratio or which are subject to glare from reflected light.

In their experiments, Dyson and Kipping (1998) suggested that large areas of white screen could produce glare "and may [slow] down reading", and thus they switched to a grey backdrop in a subsequent experiment in their study. Therefore an additional peach coloured backdrop with lower luminance has been used on either side of the text in the reader. In a study of both people with and without dyslexia, the use of this peach colour behind text resulted in improved reading performance relative to other background colours (Rello and Bigham, 2017).

Previous research has shown certain affordances or mental models can improve readability. (Liesaputra et al., 2009) To provide visual feedback, in line with emerging design patterns for including contextual page transitions on the web (Web Incubator OG, 2022), a minimal animation has been added between page turns. When the page is turned, the text of the next page appears instantly but only at 70% opacity, and then fades in to 100%. This minimal feedback serves to signpost that a whole-page change in the text has occurred, without introducing a delay which might inhibit navigation.

Font face, size, and line-height

Each article's body text is presented at the default text size associated with the user's device, which takes into account any font size preferences set by the OS or browser. In this experiment, font-size is set at 1 rem (i.e. root element unit), which is equivalent to the font-size used on websites like Wikipedia, and fits the recommendations of many UX style guides which suggest a font-size of 16-18px for body text on the web (Scacca, 2018; British Dyslexia Association, 2022).

During testing, Windows 10 and Google Chrome with default settings and a font-size of 1 rem in the pilot reader was found to equal exactly 16px on both a 1920 x 1080 monitor and a 3840 x 2160 monitor (with default 250% font scaling). All pilot testers were asked whether the font size was too small or too large. Five said that they had no issues; one participant (who was not using their own personal device) said that the font was slightly too small, but added that they simply had to use their reading glasses. One tester added that, when they read on their Kindle ebook reader device, “I find myself [reducing] the font size in the Kindle [to minimise page turning] so that I can read faster.”

Note, however, that at the time of writing many websites (including long-form news magazines such as The New Yorker and The Atlantic) use significantly larger fixed font sizes in the range of 21-22px. Such a large font size has a basic usability advantage: it may increase legibility in readers, particularly those with reduced vision (and those who have not configured device font preferences). However, because it is not based on the root size which a user has specified on their device, specifying an exact px value for font-size (or adding additional scaling based on one) runs the risk of being either too large or too small on certain reader’s devices.

Added to this, using larger font sizes has a distinct disadvantage for reading long texts on the web because it reduces the effective number of words visible on screen at once, necessitating more scrolling or page turning. We can suppose as well, based on a consideration of working-memory capacity and reduced cognitive load, that showing more text on screen at once decreases the effort involved in creating a surface representation of the text, particularly in the scrolling mode.

Line-height may also exhibit a similar effect (increased line height reduces available text per screen). However, a too-large line-height does not result in the same scaling disadvantage of too-large font sizes. Additionally, UX guidelines for the web routinely recommend a line-height of 1.4 to 1.5 (British Dyslexia Association, 2022), and thus 1.5 has been used in the reader to provide for maximum readability.

A number of different fonts were considered - including several modern, new fonts designed for readability (Britton, 2020). However, Arial was selected because it is the most popular and widely supported typeface and more likely to be familiar to participants. The reader uses fallback fonts of Helvetica and “sans-serif”.

Justification

The text in both reading modes is left-aligned with a ragged right edge, a format found on most online news websites.

CSS has 4 options for alignment: left, center, right, or fully justified text (where text is aligned to both left and right margins and padding with spaces between words such as the text in this document). Fully justified text was considered for its aesthetic appeal. Most print magazine journalism utilises left-justification (which left-aligns the final line, and/or makes use of hyphens and other techniques to keep word-spacing consistent). CSS currently lacks the option for left-justified text as found in desktop publishing software such as Adobe InDesign.

Furthermore, during testing, using CSS full justification resulted in inconsistent behaviour on Safari where the final line would be stretched to reach the right margin creating large gaps even when this line only contained a handful of words.

3.1.4 Scrolling mode

The reader was designed so that it could be deployed to show articles in either pagination or scrolling mode. In both modes, articles are rendered identically (same margins, typography, colours, spacing) except that scrolling mode's bottom navigation bar is absent and the article extends downwards with a standard browser-generated right-hand scroll bar.

A dynamic navigation bar fixed to the right edge of the article text box was considered for inclusion in the scrolling mode. This would have helped match the potential navigation UI advantage which the pagination mode has by having article progress information closer to the text when viewing the web on a widescreen device. (Added, most browsers typically use a very narrow, muted scrollbar on the right-hand side of the screen to minimise the screen real estate occupied). However, this would have duplicated the functionality of the browser scroll bar (potentially introducing a disorientating change to the standard user experience of scrolling a web document), and was avoided.

3.2 Experimental design and procedure

This experiment was designed to be conducted in two phases: a pilot test (featuring six testers), and an online study with 40 paid participants recruited from a research recruitment platform.

The pilot test was used to trial both the application and the research design (particularly reading speed timings and comprehension questions). During pilot testing, each participant was interviewed in-person or over video conferencing and asked to provide open–ended-feedback about their reading habits, their experience of the experiment and the reading modes, and to evaluate each of the questionnaire pages within the experiment which required responses.

3.2.1 Experimental software

There are a number of frameworks and software solutions for running experiments in the browser for use in psychology and user experience research. However, due to the nature of this study's reading experiment, a tool was required which could embed the existing JavaScript implementation of the reader described in section *3.1* of this chapter. The implementation needed to be seamless, allowing complete control over HTML, CSS and JavaScript, whilst also allowing the experiment to be deployed as a single-page application in which the participant progressed between different screens and reading conditions whilst storing user responses and automatically tracking time spent on various screens.

Lab.js (Henninger et al, 2022) is a relatively new open-source tool which was selected for this purpose. It provides both a JavaScript library and a study builder tool with a GUI in which different screens of the experiment can be created as interlinked components. Importantly, it allows the code of each screen and component to be fully modified. Moreover, it is extremely flexible for deployment: it was possible to deploy the complete study both as an offline website for testing and to deploy it on Netlify where the experiment was hosted for the pilot and online phases.

A research paper by the *Lab.js* creators (Henninger et al., 2022) details its ability to record timings - which Henninger et al. identify as a major concern in online experimental research. Timing can be considered in two dimensions: 1) accuracy, which relates to bias or lag, and precision, which refers to "freedom for measurement noise" (Henninger et al., 2022). *Lab.js* purports to reduce both of these factors by using algorithms to capture high-resolution timings (with sub-millisecond precision) using frame synchronisation, which aligns timings to the browser animation cycle, and the use of an optimised rendering engine which, for example, pre-loads and prepares stimuli prior to presentation on screen.

A key benefit of the approach of *Lab.js* is that the entire study's logic can run on the client side. This eliminates any network latency (lag time of sending HTTP responses across the network from the client to the server).

3.2.2 Selecting a text and assessing readability

Approximately 30 long-read texts were reviewed via The Guardian newspaper's Open Platform API which allows programmatic access to its content archive for use in the experimental conditions in this study.

Initially, two separate articles were sought to satisfy the repeated measures design of this study. However, no suitable article pairs with similar readability and style could be found (i.e. articles with the same author, subject matter, and relative length).

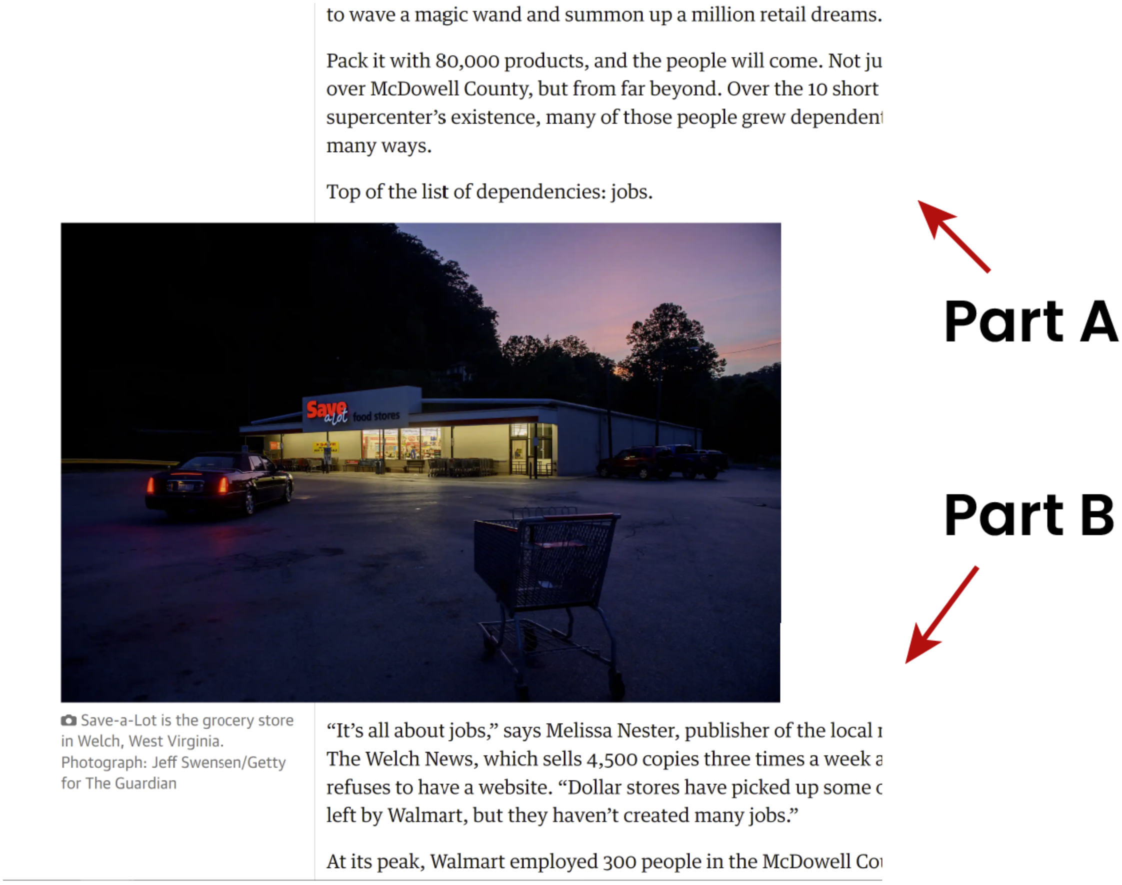
Instead the design was modified so that both conditions would be carried out on two sequential parts of the same article. This altered design provided clear benefits: both text samples would have the same author, same style, and matching content. Both samples were later edited so that word counts and readability measures matched exactly. This approach also helps provide a sense of continuity for participants during the experiment which more closely resembles the real experience of reading a long-form journalism article.

Several longer texts were excluded (4,500 to 7,000 words) because they would have necessitated a much longer reading experiment, increasing fatigue in participants and limiting potential sample size.

Other texts, which may have been more stimulating or interesting to readers (such as *The mystery of the man who fell from the sky*, and *Why have young people in Japan stopped having sex?*) were excluded because they featured graphic or objectionable content which might affect some readers' disposition towards the text.

Ultimately, the selected text: *What happened when Walmart left* was chosen in part because the story and setting (a small American town) was unlikely to have been covered in Irish and British media. Additionally, the cultural context of America is generally familiar to Irish and British audiences. Added, narrative pacing is consistent throughout: the article consists entirely of various characters being introduced every few paragraphs in short, focused vignettes - sprinkled with facts relevant to the overall theme.

To facilitate the repeated measures design, the article was split in two at a point which happened to fall on a natural pause and transition in the article.


to wave a magic wand and summon up a million retail dreams.

Pack it with 80,000 products, and the people will come. Not ju
over McDowell County, but from far beyond. Over the 10 short
supercenter's existence, many of those people grew dependent
many ways.

Top of the list of dependencies: jobs.

Save-a-Lot is the grocery store in Welch, West Virginia. Photograph: Jeff Swensen/Getty for The Guardian

"It's all about jobs," says Melissa Nester, publisher of the local r
The Welch News, which sells 4,500 copies three times a week a
refuses to have a website. "Dollar stores have picked up some c
left by Walmart, but they haven't created many jobs."

At its peak, Walmart employed 300 people in the McDowell Co

Figure 3.6: Original Guardian article with break point selected to divide experimental conditions.

To ensure the two sections were equivalent in difficulty, the Flesch-Kincaid readability tests were used to assess each. These tests purport to assess a text's difficulty using a readability score and the relative educational level required for it (Flesch, 2016). They work by inferring difficulty from the number of words, syllables, sentences, and the average numbers of syllables per words and words per sentence.

It should be noted that readability tests have a number of technical weaknesses, such as originally being formulated for children's school books and failing to account for things like tone, content, or imagery (Redish, 2000). Redish recommends user testing as a superior alternative, and two SEQ scales will also be applied in the final experiment which will be evaluated in the results section.

Article section	Grade level	Reading ease	Reading level	Words	Words per sentence	Syllables per word	Sentences
Part A	9.6	60.4	8th & 9th grade	1592	19.2	1.5	83
Part B	7.6	70.9	7th grade	1581	17.2	1.4	92

Table 3.1 Flesch-Kincaid measures for parts of the original article. (Good Calculators, 2022)

Following Dyson and Kipping's (1998) research design, text samples were selected to be "approximately" equal in length. In the sample text used in this present study, however, it was possible to edit both parts to be exactly equal in both word count and number of syllables.

Following pilot testing with six testers, 11 words were removed from Part A. Many sentences were broken up, or changed from passive to active, in order to reduce the number of independent clauses and shorten sentence lengths.

A number of words were directly replaced with simpler synonyms (or, in the case of Part B, replaced with more comprehensive adjectives and nouns). A sample of these changes can be found in the tables below:

Original text	Revised text for online phase
"She remembers the feeling of excitement and expectation as…"	"She remembers feeling excited as…"
"... the gargantuan "big boxes" that have become…"	"... the huge "big boxes" that have become…"
"...treated for congenital heart disease…"	"...treated for heart disease…"
"...Goodsons, is too expensive, she says, and other Walmarts…"	"...Goodsons, is too expensive, she says. Other Walmarts…"
"...overpowering the competition…"	"...beating the competition…"

Table 3.2 Sample of text changes made to Part A to increase readability.

Original text	Revised text for online phase
"...and doggedly refuses to have a website.…"	"...and obstinately refuses to have a website."
"At its peak, Walmart…"	"At its zenith, Walmart…"
"There were knock-on effects, too,"	"There were second-order effects, too,"
"There are some rays of hope piercing through..."	"There are some glimmers of hope penetrating through..."

Table 3.3: Part B changes - reading ease was decreased by introducing more complex words, increasing syllables.

After these changes (carried out over two repeated passes of each article section), both sections matched each other in terms of word count, syllables per word, reading ease, and other metrics in the Flesch-Kincaid reading scale:

Article section	**Grade level**	**Reading ease**	**Reading level**	**Words**	**Words per sentence**	**Syllables per word**	**Sentences**
Part A	8.7	62.7	8th & 9th grade	1582	17	1.5	93
Part B	8.7	62.7	8th & 9th grade	1582	17	1.5	93

Table 3.4: New Flesch-Kincaid scores after text adjustments.

Finally, in the online experiment, this study used two prompts after each text to capture participants' perception of a text's difficulty and readability by rating them on a 7-part SEQ scale:

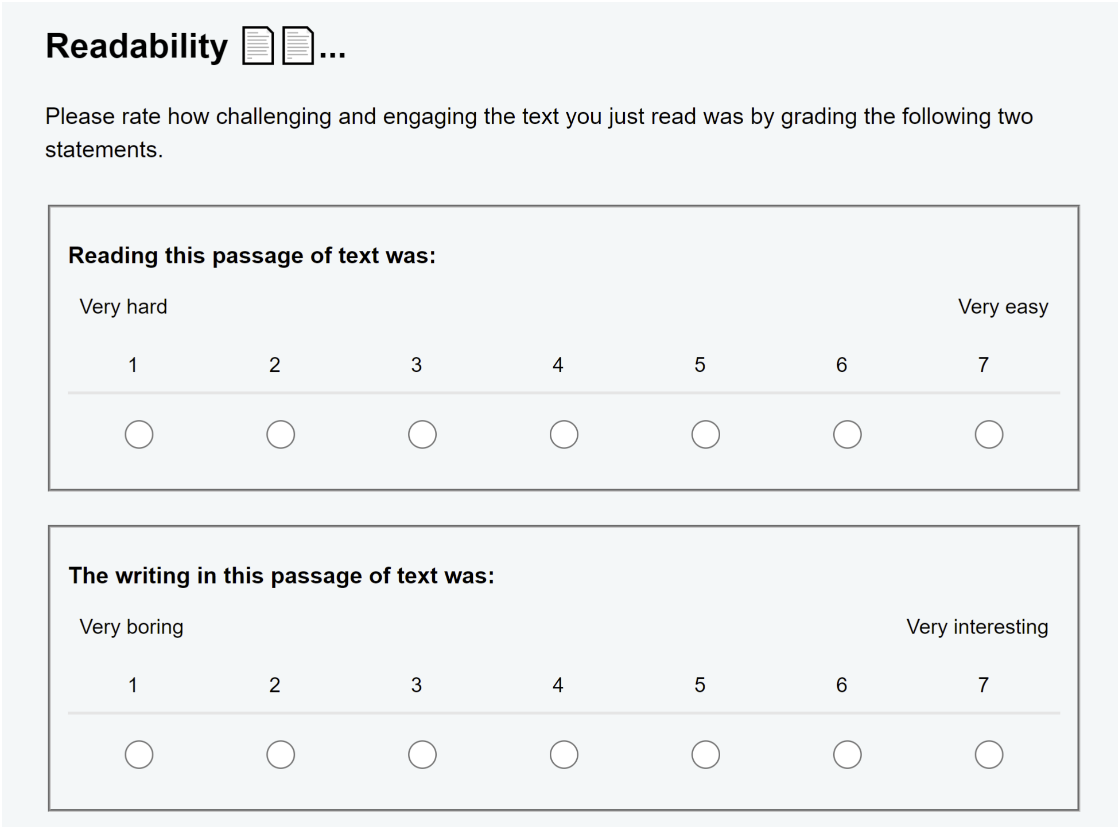


Figure 3.7: 7-part readability scales (SEQ) presented after readings.

In the analysis section, this scale is used to validate that there was no significant difference between the difficulty or readability of either text (irrespective of reading mode).

3.2.3 Comprehension test

In their study on digital reading, Shaikh and Chaparro (2005) used a comprehension test with 19 questions per article (375 words each). This enabled them to calculate reading efficiency (speed * % correct on comprehension).

Another potential method for assessing comprehension, used by Wiley and Sanchez (2009) in their study of working memory capacity in scrolling and paging modes, required participants to write short essays summarising information from a text. This metric not only measures recall but also assesses how participants have processed and interpreted information. Unfortunately, as with using a high number of comprehension questions, the limited timespan (< 25 minutes) and difficulties in guiding participants in an online study made this method for assessing comprehension unviable.

Thus in this present study a multiple-choice question design is used which is similar to that deployed by Chaparro and Baker (2003) featuring five questions per text sample.

As in Copeland and Gedeon (2015), this allows a comprehension score to be calculated for each sample that a participant reads (which can be aggregated across all participants by reading mode), as well as an overall comprehension score covering both texts.

In order to filter out inattentive participants who did not follow the instructions (i.e. skip or skim sections of the text), a minimum comprehension score of 0.6 was set (3 out of 5 questions correct for each sample) for the inclusion of a participant's data in the final analysis. Note: a downside of this approach is that it might result in good faith participants inadvertently being excluded from the study. Thus it is necessary to write comprehension questions in such a way that they provide multiple hooks or breadcrumbs for recall and recognition of details.

Comprehension questions were set by reviewing the text and taking notes on named individuals, dates, figures, and important quotes or phrases which tie in to the overall narrative. These questions and associated correct answers were worded so as to engage users in recognition (i.e. presenting familiar combinations of words or phrases), which is an easier form of memory retrieval than pure recall (Budiu, 2014). Budiu says that since memory is organised in interlinked chunks, retrieval of chunks is influenced by recency and context. Separate chunks can be related by context such that when one chunk is activated, it "spreads activation" to another chunk, allowing that chunk to also be recalled. (Budiu, 2014) Since recognition is based on cues (i.e. partial pattern recognition), it can be easier to activate memory chunks related to recognition by writing comprehension questions which include multiple familiar cues from the texts.

Nevertheless several questions were also designed to be answered via recall (i.e. specific facts or details, in this case names and numbers), with either breadcrumbs and contextual clues, or a high number of appearances in the article. One question in each also makes use of interpretation (for example, correctly being able to parse something a person actually said from what they necessarily might believe).

Question	Type	Assessment	Notes
A Q. 1	Date	Recall/recognition	Can be deduced from other details present in article
A Q. 2	Name	Recall	Appears many times
A Q. 3	Name	Recall/recognition	Appears a few times
A Q. 4	Complete a sentence	Recognition	
A Q. 5	Choose a description	Recognition/interpretation	

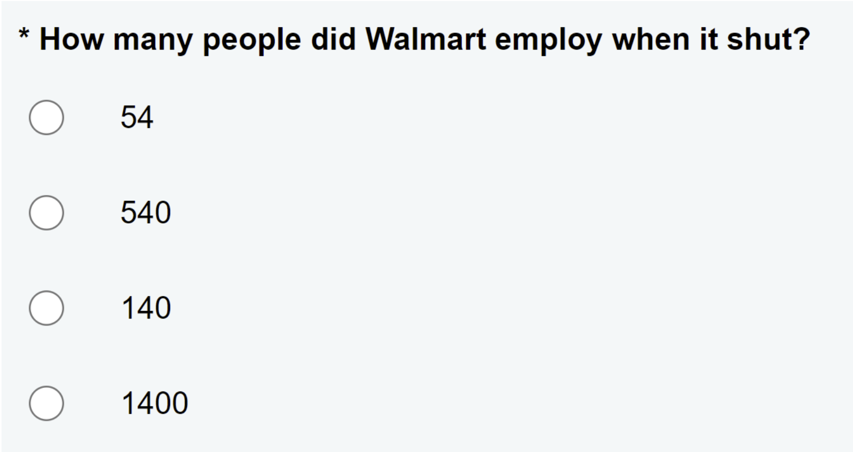
Table 3.5: Classification of Part A comprehension questions.

Question	Type	Assessment	Notes
B Q. 1	Name	Recall/recognition	Appears a few times close together
B Q. 2	Long name	Recall/recognition	Appears once, memorable cultural context
B Q. 3	Number/description	Recall/deduction	Appears a few times
B Q. 4	Complete a sentence	Recognition	Very memorable
B Q. 5	Choose description	Recognition/interpretation	

Table 3.6: Classification of comprehension questions in Part B.

During pilot testing (n = 6), the lowest participant score was 0.7 (7 questions out of 10 correct). In follow-up interviews, each participant was asked to do a walkthrough of the comprehension tests and to point out excessively difficult questions or responses which were either confounding or obviously (through deduction) false.

Several pilot testers flagged question 3 for part B of the article. The article had mentioned both the correct answer (140) and the fact that 300 was the peak number of employees at the Walmart store during its 10 year existence, so it was thought this additional cue would allow most people to deduce the answer.

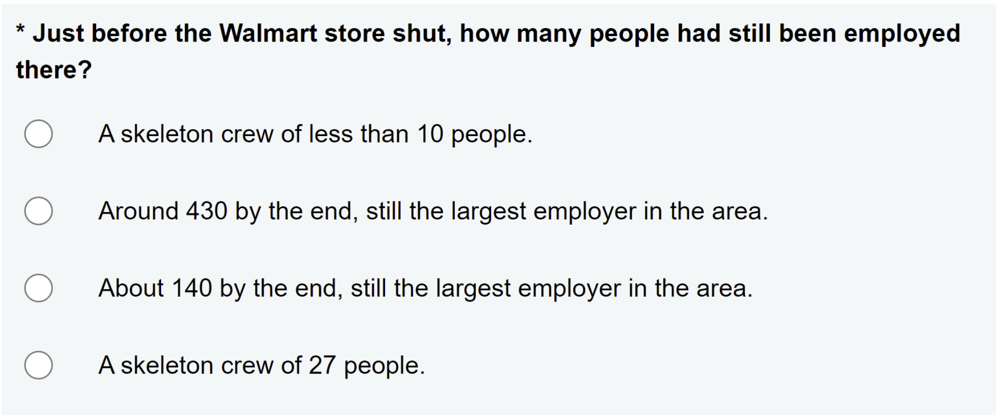

* How many people did Walmart employ when it shut?

- 54
- 540
- 140
- 1400

Figure 3.8: Pilot test version of Part B Question 3 (comprehension).

However, only 50% of responses selected the correct response for this question, and during interviews two testers said they had to guess the answer.

For the online phase, this question was rewritten to include more contextual cues, including the exact phrasing used in the article:


* Just before the Walmart store shut, how many people had still been employed there?

- A skeleton crew of less than 10 people.
- Around 430 by the end, still the largest employer in the area.
- About 140 by the end, still the largest employer in the area.
- A skeleton crew of 27 people.

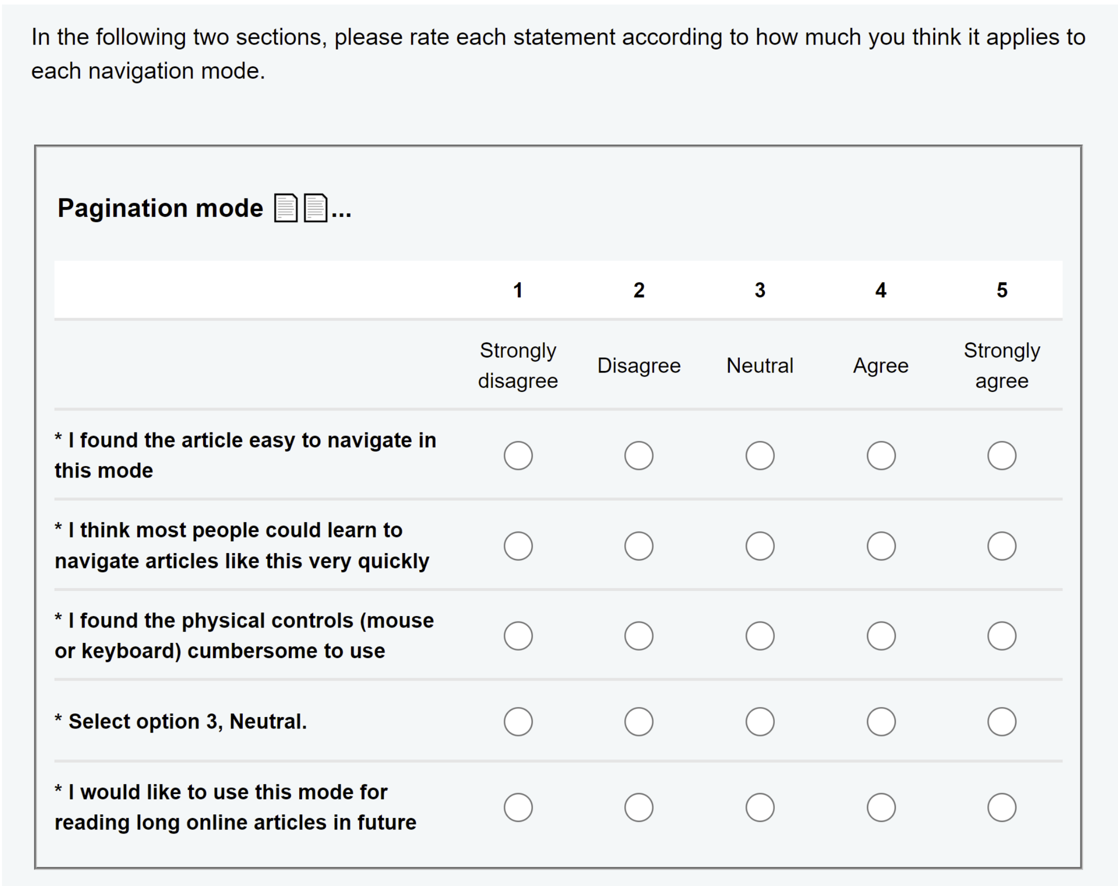
Figure 3.9: Revised version of Part B Question 3 (comprehension).

Other questions and their possible responses were also modified to add extra contextual clues.

It should be noted that the comprehension score is not one of the primary dependent variables of this study. This in part is because of the limited precision involved: 5 questions for a 1500-word article, as opposed to 19 questions for Shaikh and Chaparro's design (2005). Because of the limited nature of the online experiment, replicating such a lengthy questionnaire (38 comprehension questions in total) might have contributed to fatigue effects and reduced participants' compliance with instructions.

3.2.4 Usability and reader preference questionnaire

While brief comprehension tests and two article readability single ease questions (SEQ) were issued after each reading sample (the latter was used post hoc to validate that the readings were equivalent in difficulty and interest), the overall preference question and a number of usability scales were administered on the final screen of the experiment. Questions were staged in this order to mitigate the peak-end effect and to allow participants a separate part of the study in which they could give a considered opinion, completing two side-by-side custom 4-scale usability questionnaires before indicating an overall preference.


In the following two sections, please rate each statement according to how much you think it applies to each navigation mode.

Pagination mode 📄📄...

	1	2	3	4	5
	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
* I found the article easy to navigate in this mode	○	○	○	○	○
* I think most people could learn to navigate articles like this very quickly	○	○	○	○	○
* I found the physical controls (mouse or keyboard) cumbersome to use	○	○	○	○	○
* Select option 3, Neutral.	○	○	○	○	○
* I would like to use this mode for reading long online articles in future	○	○	○	○	○

Figure 3.9: Usability Likert scales applied to pagination.

Note that these questions may suffer from a partial acquiescence bias (Rosala, 2020): a person’s tendency to agree with others. In this case, because the statements are framed with salience in one direction (“I found the article **easy** to navigate”), respondents may be more likely to agree with the statement. Ideally, the statements would normally be counterbalanced with equivalent negatively framed statements (e.g. “I had a difficult time reading the article”), but these were omitted to avoid overburdening respondents and reduce the potential for fatigue.

These questions are based on one of two standard usability and cognitive load questionnaires originally considered for this study: the System Usability Scale (SUS) and the NASA Task Load Index (NASA TLX).

Neither questionnaire was found to be among the research methods of earlier examples of digital reading research which were reviewed for this study.

Nevertheless, parts of the SUS questionnaire (outlined by Laubheimer, 2018) provided a model of how to measure usability and learnability in the reader system. However, given that the system in question (in the scrolling or pagination conditions) only involves a single minimal activity (pressing a button or scrolling the mousewheel), most of the standard SUS statements were inappropriate for the present study, and were thus modified.

The NASA TLX was also ruled out. While it offers a useful way to measure mental workload in digital experiences, the number of questions required (19 for each condition; 38 in total) would have significantly burdened participants. Note that previous digital reading experiments have modelled cognitive load and related measures in a number of ways - including saccadic eye movements, brain imaging, and complex pre and post cognitive tests - not practical in a limited remote experiment.

3.2.4 Recruitment

Participants for the study were recruited via the Prolific platform (https://www.prolific.co/). Prolific is an online service which allows researchers to recruit participants on-demand for browser experiments. Participants can be screened or filtered according to demographics and various other screening responses they have provided to Prolific.

Users recruited from Prolific for this study were awarded compensation of £3.34 (€4.70), along with bonuses (see *3.2.5 Attention checks & repeated instructions*) for an estimated 25 minutes of their time.

Several pre-screening filters were applied on the Prolific platform in order to partially control for variables that could affect the results. These filters included:

- **Nationality**: United Kingdom.
- **First Language**: English.
- **Country of Birth**: United Kingdom.
- **Place of most time spent before turning 18**: In England, or Elsewhere in the UK.

While participants could have been recruited from other countries, people from the UK were chosen to control for different educational experiences or cultural attitudes which might affect their disposition to reading the selected article. (In this case, the sample article was published in a British newspaper).

Other pre-screening filters applied included:

- **Highest education level completed**: High school diploma/A-levels.
    - Fixed at a minimum level (people typically complete A Levels at age 18 or above) to ensure all participants had a reading level above that of the text (Grade 9 according to the Flesch-Kincaid Grade Level, which is typically age 14 and above in the equivalent US school system).
- **Vision**: Normal or corrected-to-normal vision.
    - Participants who do not have normal or corrected-to-normal vision were excluded to rule this out as an extraneous variable. (In this experiment it was not possible to balance the two participant groups in a representative way and thus this may have affected the validity of statistical tests).
- **Approval rate**: 100%
    - Approval rate is the percentage of previous studies for which participants' submissions have been approved by other researchers.

After applying these criteria, the Prolific recruitment tool adjusted the estimated available pool from 137,142 participants down to 18,362 participants.

3.2.4 Informed consent and data protection

This study was designed to align with Prolific's recommendations (Prolific Team, 2022b) regarding consent as well as data protection best practices which conform to the standards of the EU General Data Protection Regulation (GDPR) (Prolific Team, 2022a).

- No personally-identifying information (PII) was recorded. Participants were, however, asked to provide their age. To avoid the collection of PII such as date of birth, this was phrased as "Age (in current years)".
- Participants were asked on a pre-screening page within the experiment to provide demographic information including education level, first language, and country of residence. A "Prefer not to say" option was included.
- Prolific provides separate demographic data (including age, education, language, and sex) which was processed in aggregate (i.e. to produce statistics about the participant sample as a whole). A summary of this additional demographic information is shown in the results section.
- Participants were shown a screen (Figure 3.10 below) which described the study and how their data would be used and the option to consent or opt out of the study without penalty. The experiment was designed so that participants could withdraw consent at any time by simply closing the tab (which would result in no data being stored). At the study's conclusion, the debrief screen provided additional information about the study and its purpose, and participants were again given the option to withdraw consent before their data would be transmitted.

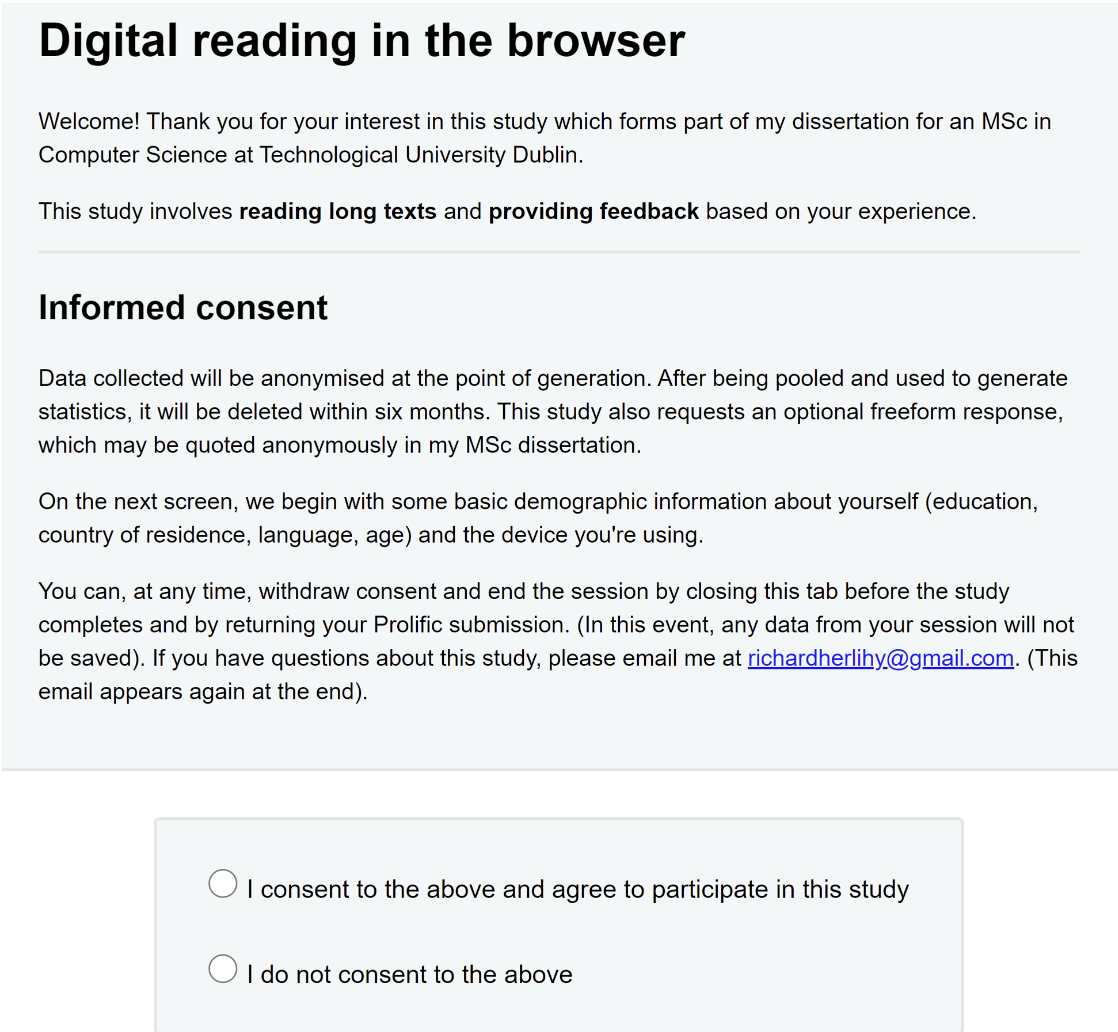


Figure 3.10: Consent screen shown at the beginning of the experiment.

3.2.5 Attention checks & repeated instructions

Participants were briefed with an instruction screen written in a plain English language style, broken down by bullet points and illustrated with icons (as recommended by Dörnyei and Taguchi, 2010) in order to reduce cognitive load and promote cooperation in participants.

Brief instruction screens were also displayed on how to navigate the subsequent reading modes (i.e. mouse scrolling or page turning) before each reading task.

In the pilot phase, one older tester read out loud because she found this easier for the sustained concentration required (20 minutes of reading). The design used in earlier in-person reading studies (Dyson and Kipping, 1998) specifically asked participants to read silently. However, in this present research, no instructions were given regarding verbalisation in the online phase as it would be impossible to enforce and might disrupt natural behaviours.

Following pilot testing, several participants' reading time data exhibited differences between reading modes. For example, one person took 265 seconds to read Part A using pagination, whilst taking 397 seconds to read Part B whilst scrolling. When questioned the participant said that they were interrupted several times by their daughter during the first reading and hadn't realised that the experiment involved timing. Inconsistent timings can also be seen in the data generated by two other participants, one of whom said that they had felt tired because they were doing the experiment at midnight and so speed-read through part B. Later, when two additional random users on the Prolific pattern were enrolled in the study for inclusion in pilot testing, another response also exhibited unusual timings.

As a result of these inconsistencies, additional attention checks were added to the study (see below), as well as an explicit reminder before each reading activity that it would be timed. Lastly the study description on Prolific was modified to include an additional reward: "A bonus of £1 will be added if you pass all attention checks and follow instructions precisely." While this additional incentive may have impacted the experimental conditions by introducing an additional performance bias into the design, it would likely impact both experimental conditions (scrolling and paging) equally.

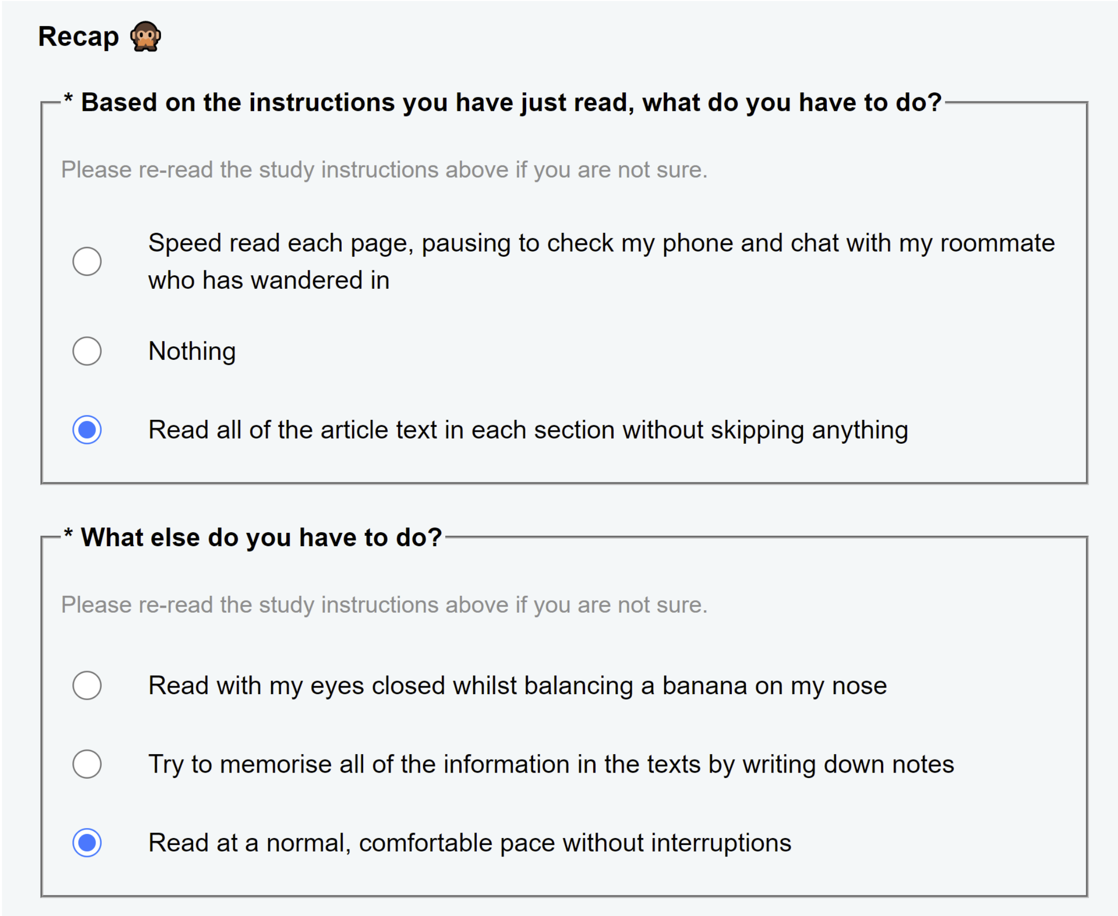


Figure 3.11: Instruction screen attention checks exhibit "nonsensical item" design and two chances for a correct answer.

These checks were modelled upon examples from Prolific's Attention Check Policy (Prolific Team, 2022c), which are themselves based upon examples and instructions from Gordon et al. (2019).

3.2.6 User device constraints

Previous in-person research designs have typically controlled for variables such as the reading device, software, input devices, and screen. (Dyson and Kipping, 1998)

Thus in the online study platform used to recruit participants for this paper, potential participants were screened according to device and browser type by requesting that they meet the following minimum requirements:

- Desktop or laptop computer
- Physical keyboard and pointing device such as a mouse or trackpad

- An up-to-date version of Google Chrome, Safari, Microsoft Edge, or Mozilla Firefox.

This was further validated by a screener questionnaire within the study itself and by checking the user agent.

An interesting aspect of older research, which may be a confounding variable, has been the relatively limited image quality of screens and differences in input devices and interaction paradigms. For example, Dyson and Kipping’s 1998 study utilised a cathode ray tube (CRT) monitor with a resolution of 800 x 600 and 256 colours; Chaparro and Baker (2003) had a system with a resolution of 1024 x 768. Higher resolution facilitates greater fidelity and less blurriness of characters, which may improve readability. Devices from previous decades also typically have reduced contrast ratio, viewing angle, and lower refresh rates (the frequency at which the screen image is updated at). Refresh rates above 60 Hz have been shown to marginally increase reading speed at a level of significance (Montegut et al., 1997), and also to increase reaction time when using a pointing device (Murakami et al., 2014). Research by Behler and Lush (2010) also found that participants in an ebook reader experiment were frustrated by the slow screen refresh rate when turning pages on an eInk device.

Because participants were in a natural environment (i.e. on everyday devices they might typically use to read online), no effort was made to screen participants by display characteristics or to have them modify their display directly. However, the experimental design records the screen resolution of each participant (as reported by the user agent), and the application design has attempted to follow a seamless instantaneous interaction paradigm.

Some studies have attempted to control for different kinds of user interaction by placing constraints such as covering the keyboard to hide irrelevant keys or by putting the mouse out of reach. (Dyson and Kipping, 1998)

In this experiment, to partially control for potential differences in how people advance through web documents, participants were directed to navigate using their mouse (or laptop trackpad) in the scrolling mode and the right arrow key on their keyboard in the paging mode. However, owing to different capabilities or issues participants might have with fine motor control and input devices, participants were also told that they could navigate in whatever way they felt most comfortable or familiar if they found using a mouse or trackpad challenging

The final version[21] of the experimental design and the reader application can be viewed at the link below:

View live demo on Github.io | https://ric-cde.github.io/reader/msc/study

[21] NB: This is a static, client-side-only version which does not record responses.

4. RESULTS, EVALUATION, AND DISCUSSION

4.1 Results

4.1.1 Sample

A total of 43 participants were recruited for the online study. After screening and testing, 38 of these responses were used for the final analysis.

Because Prolific allows for on-demand recruitment of participants, places on the study were released in batches and responses were validated batch by batch.

Firstly, the study was released to a batch of two random participants for testing purposes. On review, the timings of one participant exhibited a large discrepancy between reading modes: the time taken to read Part B, 230 seconds, as reported by the lab.js package, was approximately 57% shorter than the time taken to read Part A, 526 seconds.

Following this, the study was adjusted to include more attention checks and reminders that each reading sample would be timed (see: *3.2.5 Attention checks and repeated instructions*). Responses from the other participant in this batch were validated (they received a comprehension score of 0.7 or 7 questions out of 10 correct and had a small differential between times). However, their response was also discarded due to the adjustments made in the experimental protocol after this batch.

The revised experiment was then released in four batches of 10 (for a total of 41 participants) on the Prolific platform between 5pm and 10pm on a Tuesday and a Wednesday. Each batch belonged to one of two groups with different orderings for the experimental condition in order to counterbalance for the repeated measures design (i.e. Part A: pagination, then Part B: scrolling, and vice versa). The ordering of questions on the questionnaire screen regarding preference for pagination or scrolling was also reversed in each of these groups.

Of the 41 responses received from these batches, submissions by 3 participants were excluded for the following reasons:

- One participant had a significant discrepancy in timings. Time taken to read Part B was just 21 seconds (3,868 WPM), whereas time taken on the comprehension screen for Part B was 320 seconds.

- Similarly, another participant's submission exhibited unusual times (396 seconds on Part A, and 140 seconds on Part B) and had the second-fastest completion time of the overall study.
- No form submission was received from one participant, even though they had progressed to the end and submitted the correct completion code.

Of 38 remaining participants, 19 participants belonged to one group (Part A: scrolling, Part B: paging) and 19 belonged to the other (Part A: paging, Part B: scrolling).

The mean age of participants was 41.3 years (sd = 13.7), while the median age was 41 and the youngest and oldest participants were 19 and 68 respectively.

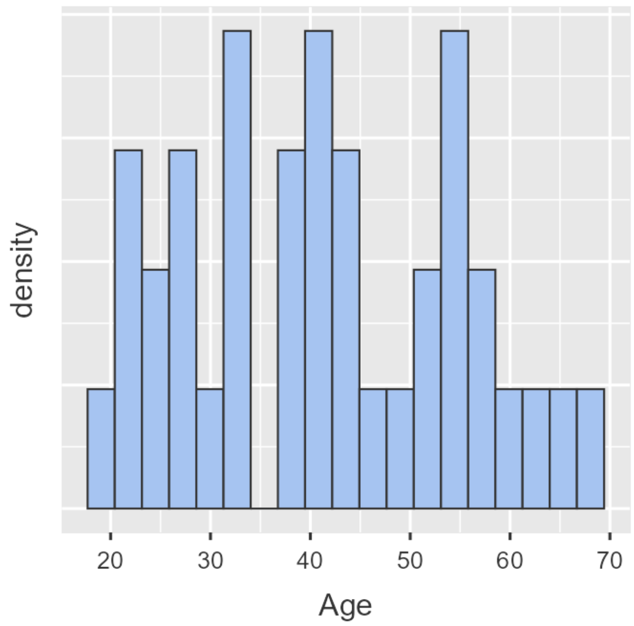


Figure 4.1: Histogram of age distribution of participants (n = 38).

In line with pre-screening filters, all 38 participants reported their country of birth, the place where they spent most of their time before turning 18, and country of residence as the United Kingdom,

The sex of participants was recorded by Prolific, for which participants had answered the question "What sex were you assigned at birth, such as on an original birth certificate?". 30 answered female (78.9%) to this question, whilst 8 answered male (21.1%).

All users reported using a personal computer, with 11 (28.9% of total) indicating their device was a laptop and 27 (71.1% of total) reporting they used a desktop. According to the platform property as reported by each client device, 28 users (73.7%) used Windows, 7 (18.4%) used Mac OS, and 3 (7.9%) used Linux.

Every user in the final sample correctly passed two attention checks posed during the instruction screen[22]. One user incorrectly responded to an attention check posed in the post-test questionnaire ("Select option 2, Disagree"). In addition, they selected the same scale on all other usability statements (4, Agree) on this screen, thus their response has been excluded from analyses[23] using this scale.

A total of 31 (81.6%) of participants said that they were using Chrome in response to a screening question in the experiment.

Browsers used by participants		
Browser	**Counts**	**% of Total**
Chrome	31	81.6 %
Edge	3	7.9 %
Firefox	2	5.3 %
Safari	2	5.3 %

Table 4.1: Browser share among participants.

Data was also collected[24] about the screens which users used to view the experiment (as well as window and scroll height). Screen height may affect reading performance or experience as it determines how many lines of text appear on each page, as well as the overall number of pages generated per sample. Most users had a standard screen resolution (e.g. 1920 x 1080, 1600 x 900, or 1280 x 768), though seven users had a unique resolution.

[22] Note that the study showed an exit screen asking users to return their submission if they failed both attention checks.
[23] This participant's responses have been retained in all other analyses, however, as their timings in the reading conditions (284 seconds and 224 seconds) fell within 2 standard deviations of the sample and they received a comprehension score of 7/10.
[24] Note that screen resolution as reported by browsers can be unreliable. For example, in recent years some 1920 x 1080 displays are frequently mis-identified as 1536 x 864.

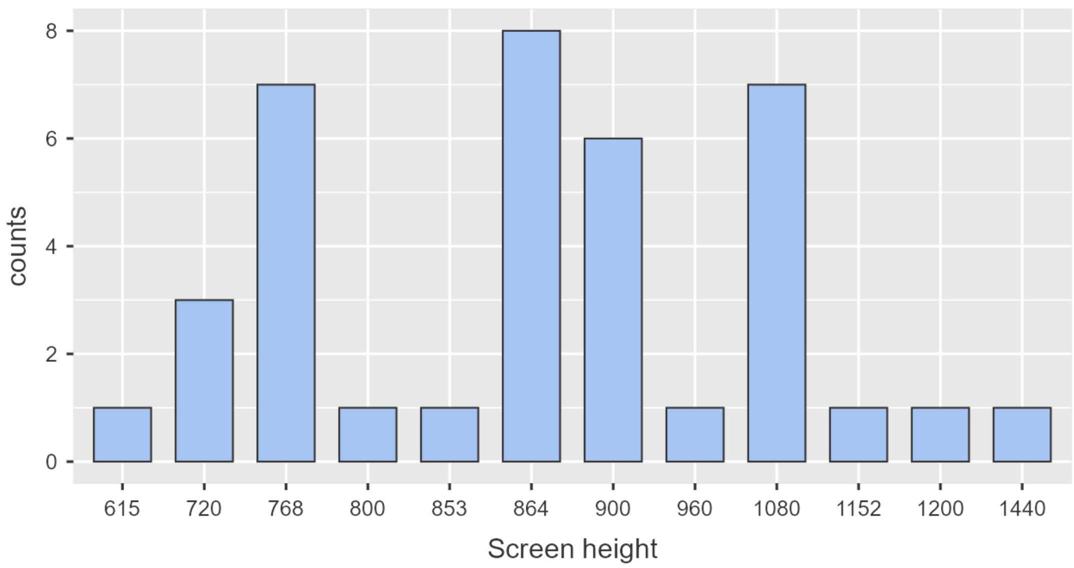


Figure 4.2: Distribution of screen resolution heights of participants in the experiment. (n = 38)

Finally, subjective readability scores on a 7-point scale (based on measures of whether the text was challenging or engaging) were aggregated across Part A and Part B.

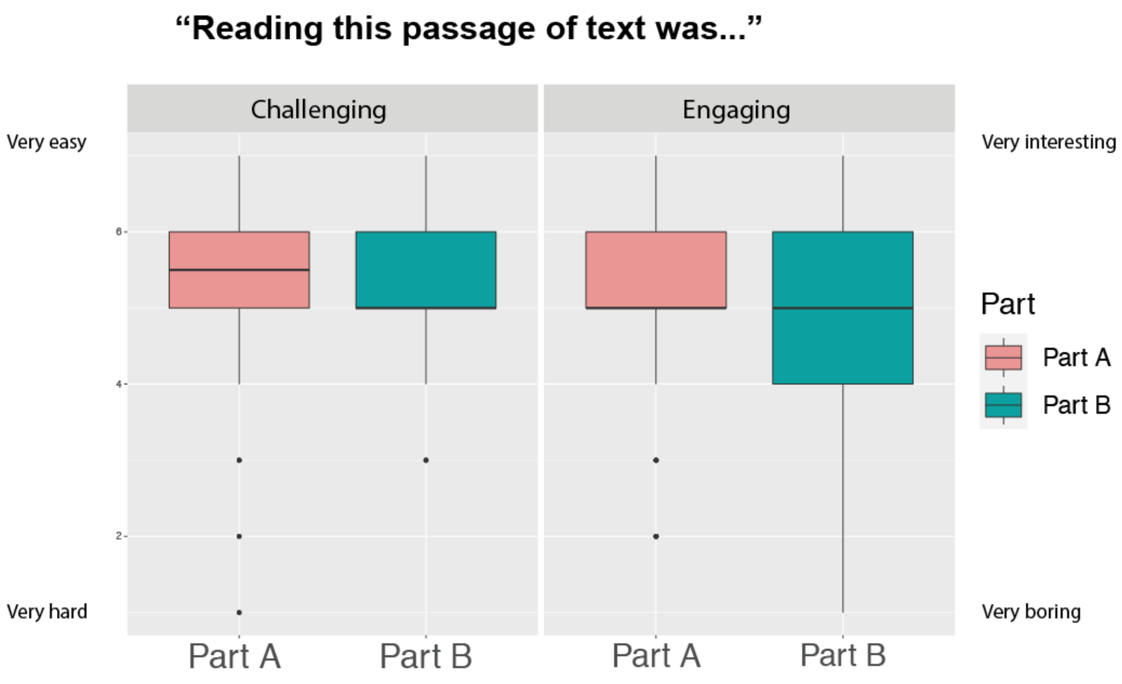


Figure 4.3: Box plot showing distribution of readability scores for article sections.

The mean scores across these two measures were found to have no significant difference according to a paired samples t-test.

	Part A Challenging	Part A Engaging	Part B Challenging	Part B Engaging
N	38	38	38	38
Mean	5.26	5.13	5.24	5.08
Median	5.5	5	5	5
Standard deviation	1.39	1.44	0.998	1.42
Variance	1.93	2.06	0.996	2.02

Table 4.2: Table showing mean scores for readability

Paired Samples T-Test					
			statistic	**df**	**p**
Part A Challenging	Part B Challenging	Student's t	0.119	37	0.906
Part A Engaging	Part B Engaging	Student's t	0.467	37	0.644

Table 4.3: Student's t-test comparing readability scores across parts.

As no significant difference was found across the parts, this provides support for earlier efforts to normalise the sections by using the Flesch-Kincaid tests and indicates that the article sections were extremely similar in terms of quantitative measures of readability.

4.1.2 Primary dependent variables

Reading speed

At the outset of this study an alternative hypothesis to test the research question was defined as follows:

> H_1: Presenting a 1500-word long-form-style text **using dynamic reflowable pagination** to online readers causes a statistically significant reduction in

reading time and an indication of subjective *preference* for this mode when compared with a standard web presentation style using scrolling.

Meanwhile, the null hypothesis (H_0) predicted that there would be no significant difference ($p < 0.05$) among these variables.

To evaluate these hypotheses, statistical tests were carried out across the mean differences within each of the dependent variables, *reading time* (words-per-minute, WPM) and *preference* (%). As the study involved repeated measures and was counterbalanced to mitigate the learning effect and other confounding variables, each participant's *reading time* in either the scrolling or paging condition was paired directly.

Mode	**N**	**Mean**	**Std. error mean**	**Median**	**Standard deviation**	**Min**	**Max**	**Shapiro-Wilk W**	**Shapiro-Wilk p**
Scrolling (WPM)	38	273	13.3	267	82.2	106	460	0.975	0.546
Paging (WPM)	38	262	14.1	255	86.7	107	523	0.942	0.049

Table 4.4: Mean and standard deviation *reading time* for participants in different modes.

Participants read at an average speed of 273 WPM in the scrolling mode, and 262 WPM in the paging mode - indicating that time taken to read the paginated texts was longer (i.e. slower) than the scrolling presentation. This suggests that the original alternative hypothesis (H_1) favouring pagination should be rejected.

These results also indicate that we should consider a new alternative hypothesis regarding *reading speed*, partially defined as follows:

> H_2: Presenting a 1500-word long-form-style text using **the standard web presentation style with scrolling** to online readers causes a statistically significant reduction in *reading time* [...] when compared with dynamic reflowable pagination.

Shapiro-Wilk tests were carried out separately to indicate whether the paging (n=38, p = 0.049, W = 0.942) and scrolling (n=38, p = 0.546, W = 0.049) samples each had normal distributions. However, the low p-value for the paging condition indicates that participants' paging mode times departed from the normal distribution (Choueiry, 2022). This is illustrated in the subsequent charts (Figures 4.4, 4.5, and 4.6).

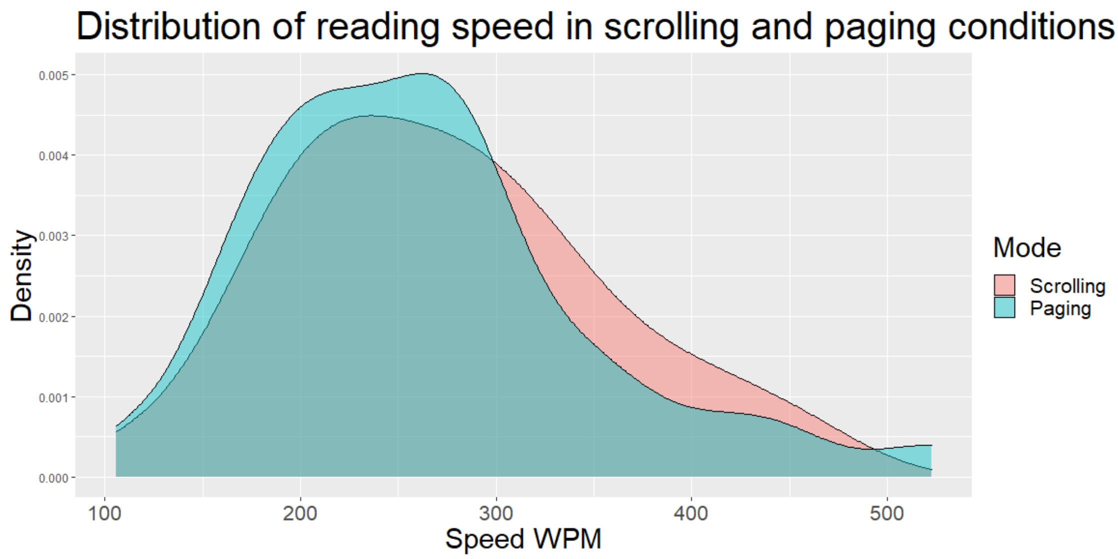


Figure 4.4: Density plot showing distribution of reading times per mode.

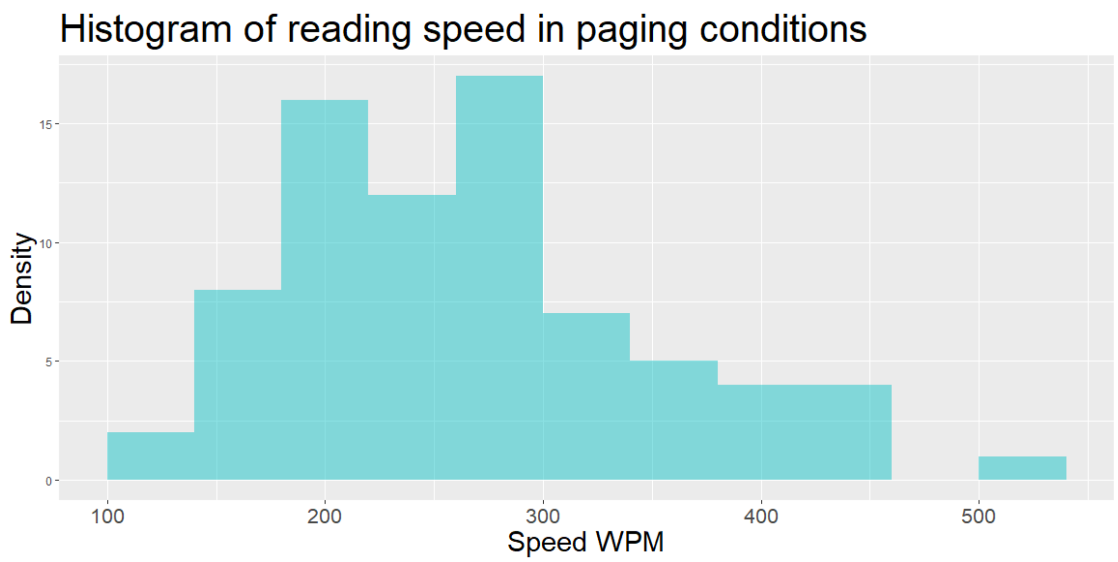


Figure 4.5: Histogram showing distribution of times in paging condition only.

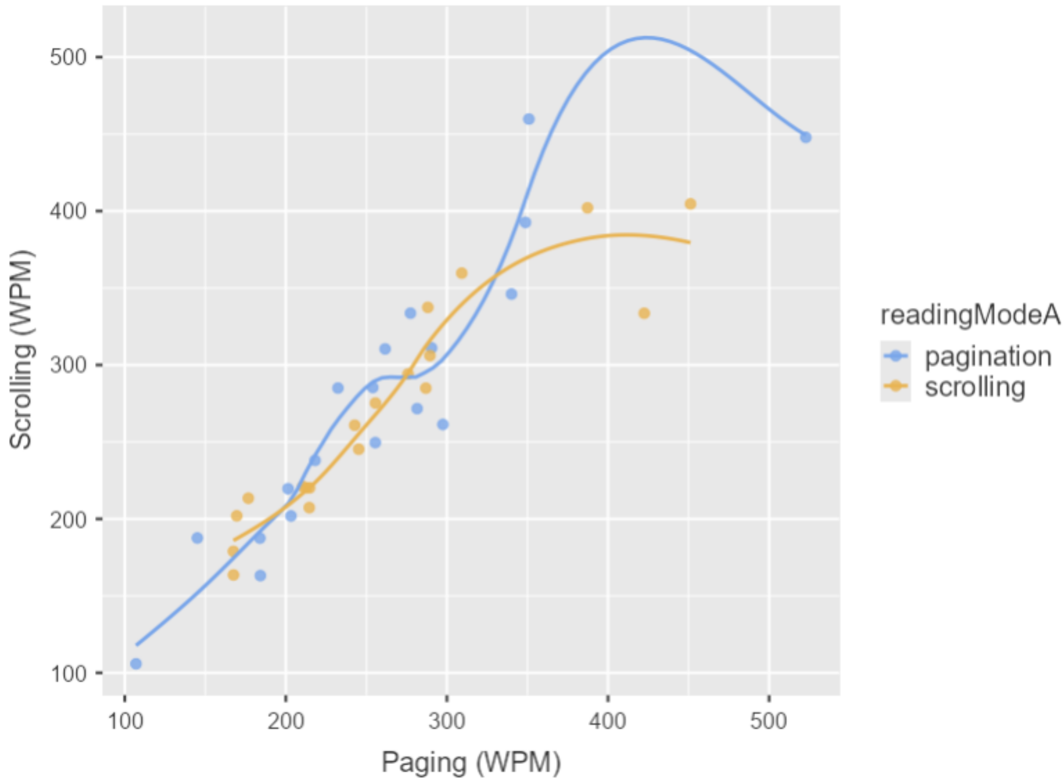


Figure 4.6: Scatterplot showing participants' reading speed in each mode. Colour indicates which mode users read Part A in (i.e. blue denotes readers who read a sample in the paging mode first). Smooth regression lines have been added.

A Shapiro-Wilk test was also carried out for the paired WPM differential between each individual's own times in the different modes (i.e. `scrolling_wpm - paging_wpm = differential`), which also indicated a non-normal distribution (n=38, W = 0.93966, p-value = 0.04084).

Thus rather than rely on a parametric paired samples t-test to explore the significance of the higher mean reading speed in the scrolling mode, a non-parametric Wilcoxon signed rank test was performed (see Figure: 4.5 below).

			Statistic	df	p		Effect Size
Scrolling (WPM)	Paging (WPM)	**Student's t**	2.01	37	0.03	**Cohen's d**	0.326
		Wilcoxon W	544		0.01	**Rank biserial correlation**	0.468

Table 4.5: Wilcoxon signed rank test results. For completeness, the paired samples t-test (student's t) is shown as well as other coefficients.

This test shows that the mean increase in *reading speed* in the scrolling mode over the paging mode reaches the level of significance ($p = 0.01$, $W = 544$).

Reading preference

A second component of the original alternative hypothesis (H_1) is whether users would indicate an overall subjective preference for the pagination mode. (I.e. did more than 50% prefer pagination?). In the post-test questionnaire, users indicated the following preferences:

Levels	Counts	% of Total
Pagination	24	63.2 %
Scrolling	14	36.8 %

Table 4.6: Frequencies of preference in response to the question: "Overall, which reading mode did you prefer?"

Of the total sample ($n = 38$), 24 (63.2% of the total) said that they preferred the pagination mode and 14 indicated a preference for scrolling (36.8%).

To test the significance of this result, a binomial exact test was carried out according to the hypothesis that greater than 50% ($H_1 > 0.5$) of respondents would prefer pagination.

	Level	Count	Total	Proportion	p
Reading preference	Pagination	24	38	0.632	0.07
	Scrolling	14	38	0.368	0.96
Note. H_1 is proportion > 0.5					

Table 4.7: Binomial exact test.

The results of this binomial test ($p = 0.07$, $n = 38$) indicates that while the proportion of participants preferring pagination exceeded 50%, this fell short of the level of significance required ($p < 0.05$) by the original alternative hypothesis H_1.

4.1.3 Secondary dependent variables

The following analyses were conducted post hoc to validate findings and further explore outcomes from this research.

Comprehension score

	Mean	Median	Standard deviation	Min	Max
Part A Comprehension score (/5)	4.76	5	0.431	4	5
Part B Comprehension score (/5)	4.39	5	0.79	3	5
Total (/10)	9.16	9	.973	7	10

Table 4.8: Summary of reading comprehension scores. (n = 38; 5 questions per reading sample).

No participants received a score lower than the previously defined minimum threshold in each part of 3.00 (i.e. participants had to get a minimum of 3 questions out of 5 correct to pass a post-hoc attention check threshold for inclusion in these findings).

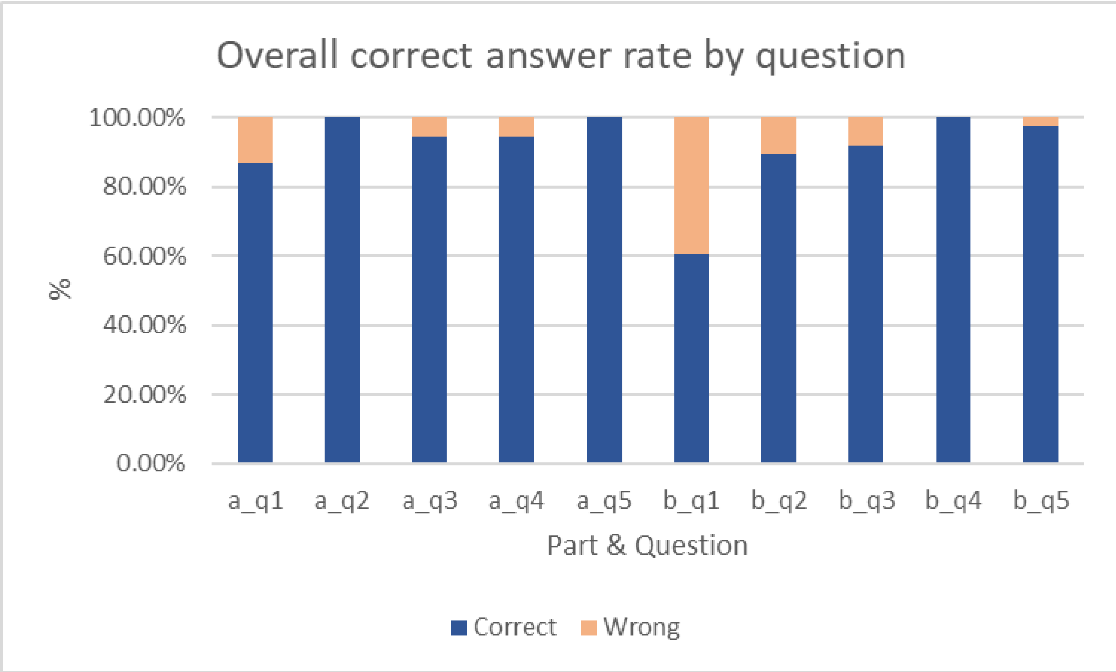
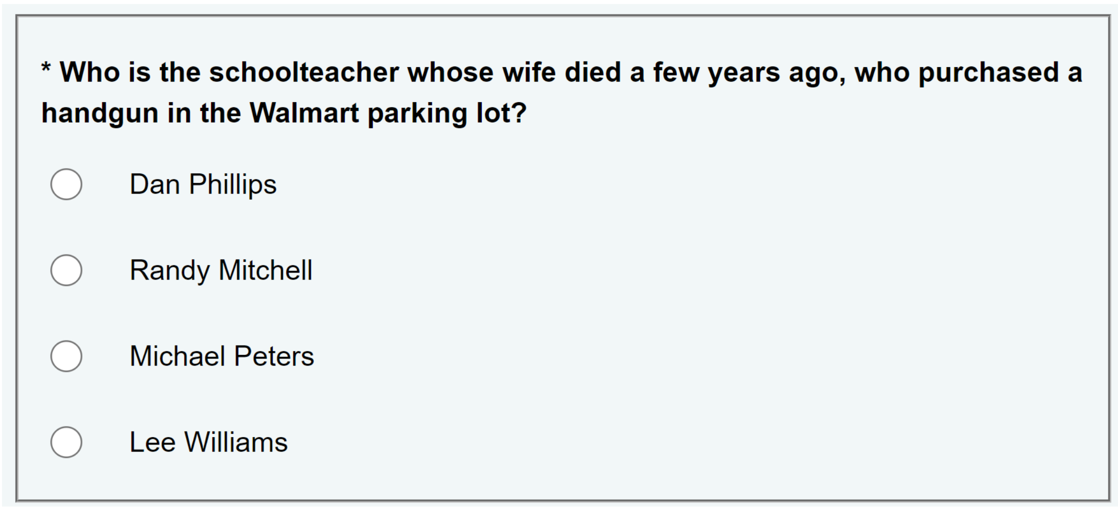


Figure 4.7: % of correct and wrong answers by all participants per question.

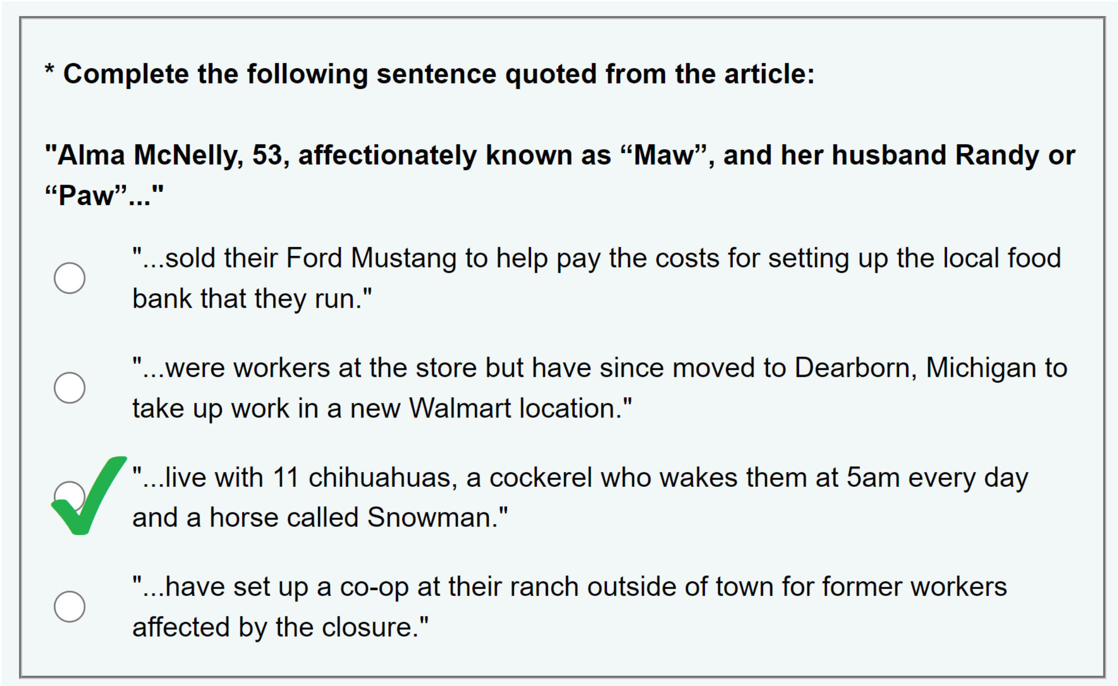
Notably, approximately 40% of respondents got Part B Question 1 incorrect. Pilot testers had also flagged this question for its difficulty, though it roughly mirrored Part A Question 3 (~95% correct answers) which also asked participants to recall the name of someone interviewed for the story.


*** Who is the schoolteacher whose wife died a few years ago, who purchased a handgun in the Walmart parking lot?**

- Dan Phillips
- Randy Mitchell
- Michael Peters
- Lee Williams

Figure 4.8: Part B Question 1 (comprehension)

By contrast, 100% of participants answered Part B Question 4 correctly[25].


* **Complete the following sentence quoted from the article:**

"Alma McNelly, 53, affectionately known as “Maw”, and her husband Randy or “Paw”..."

- "...sold their Ford Mustang to help pay the costs for setting up the local food bank that they run."
- "...were workers at the store but have since moved to Dearborn, Michigan to take up work in a new Walmart location."
- "...live with 11 chihuahuas, a cockerel who wakes them at 5am every day and a horse called Snowman."
- "...have set up a co-op at their ranch outside of town for former workers affected by the closure."

Figure 4.9: Part B Question 4 (comprehension)

In the following table, the mean comprehension score associated with each reading mode (irrespective of part) is displayed.

	N	**Mean**	**Median**	**Standard deviation**	**Min**	**Max**
Scrolling Comprehension score (/5)	38	4.53	5	0.687	3	5
Paging Comprehension score (/5)	38	4.63	5	0.633	3	5
Total (/10)		9.16	9	0.973	7	10

Table 4.9: Mean comprehension score according to reading mode.

[25] Interestingly, the correct response was perhaps the most absurd (and thus memorable), while the staged incorrect responses for this question were written in a style consistent with the overall theme of the article. This extract lay roughly ¾ of the way through the full article, which, since 100% got it correct, perhaps indicates high engagement.

From this we can see that participants in the paging condition had a higher mean comprehension score (mean = 4.63, sd = 0.633, n = 38) for sections presented in that style compared with questions based on sections presented in the scrolling style (mean = 4.53, sd = 0 .687, n = 38). In this case, the distribution is not normal (p = 0.01) according to the Shapiro-Wilks test, so we again use a Wilcoxon signed rank test.

			Statistic	**df**	**p**
Scrolling Comprehension score	Paging Comprehension score	**Student's t**	-.726	37	.236
		Wilcoxon W	70.5		.252

Table 4.10: Wilcoxon signed rank test on participants comprehension scores in different reading modes.

This test shows that the mean increase in *comprehension score* in the paging mode does not reach the level of significance (p = 0.252, W = 70.5).

Finally, one other measure is calculated using the *comprehension score* and *reading speed*. *Reading efficiency* was one of the measures calculated by Shaikh and Chaparro (2005) to explore differences between paging and scrolling presentation.

	N	**Mean**	**Standard deviation**
Scrolling reading efficiency	38	245	80.1
Paging reading efficiency	38	239	77.2

Table 4.11: Mean reading efficiency: `(comprehension score/5) * reading speed`

This metric shows a slightly higher mean reading efficiency in the scrolling mode. However, this again falls far short of a level of significance according to a Wilcoxon signed rank test (p = 0.365, W = 434, n = 38).

4.1.4 Qualitative responses

This section contains a summary of participants' freeform responses and their associated preference and performance. (Some responses have been lightly edited for brevity). At the study's conclusion, participants were invited to comment about their experience reading in different modes and how that relates to existing reading habits.

Many responses related to opinions on how navigation worked in either mode:

Category	Responses	Preference	Fastest Mode	Highest score
Navigation comment	I found both methods equally comfortable to use, I think scrolling makes it harder to see the length of the article as you are reading but the pagination method shows you the length.	Paging	Scrolling	Tied
Navigation comment	I personally preferred pagination as I knew how much there was to read and it helped me to keep reading and maintain my focus and concentration. In scrolling my eyes were strained more to look up and down the page. Whereas when I used pagination it was less straining on my eyes as the pages were smaller and easier to read.	Paging	Scrolling	Paging
Navigation comment	I struggle a bit with scrolling as I'm more prone to losing my place or the track of what I'm reading as the text moves up the screen. I prefer a static text that I can just move on by the page.	Paging	Scrolling	Tied
Navigation comment	1. One thing I liked about the pagination was knowing how far through the article I was (for example, that I was on page 2 of 4).	Paging	Paging	Tied
Navigation comment	Either way was ok but page by page shows how much you have read & how much is left	Paging	Scrolling	Scrolling
Navigation comment	Prefer scrolling because I can go up again as well as down easily. Only catch is that it can seem a long way to the bottom - there's a sense of achievement associated with pagination.	Scrolling	Paging	Tied

Table 4.12: Participant comments about navigation preference.

Despite only a third of participants favouring scrolling, many responses elaborated on why they preferred this mode:

Category	**Responses**	**Preference**	**Fastest Mode**	**Highest score**
Scrolling advantages, Navigation	I preferred scrolling as it was a continuous read. Using pagination there was a halt at the end of the page and it broke the flow.	Scrolling	Scrolling	Tied
Scrolling advantages	I do read long-form journalism quite often and only have the option to scroll. I enjoyed the pagination option - I read ebooks that sometimes use this option - but it may not be as useful as scrolling for articles that also include other elements, such as images.	Paging	Scrolling	Tied
Scrolling advantages	I generally preferred scrolling mode for ease, however I did find that because I could control how much I scrolled down, my eyes did end up wandering further down, so had to make conscious effort to keep on the same line.	Scrolling	Scrolling	Scrolling
Scrolling advantages	I am used to reading articles on a screen with the mouse. I found it marginally distracting to use a key to advance.	Scrolling	Scrolling	Scrolling
Scrolling advantages	I prefer to move text up the screen gradually so the area upon which I am focussed does not change very much. You can only do that with scrolling.	Scrolling	Scrolling	Tied
Scrolling advantages	Scrolling just feels more natural to me and I felt more in control of where the words appeared on the screen. I read regularly on a kindle and laptop	Scrolling	Paging	Scrolling
Scrolling advantages	I find when you have to read a long document by pagination, it doesn't always display the full page on screen without reducing text size which then makes it uncomfortable to read.	Scrolling	Scrolling	Tied

Table 4.13: Participant comments about scrolling advantages.

Other respondents who regularly read print materials or who grew up with traditional written media gave accounts of why they preferred the pagination mode.

Category	Responses	Preference	Fastest Mode	Highest score
Pagination advantages or print preference	I would rather read pages as I was brought up to read traditional books. I find it easy to lose my place scrolling through a long, long document. I find it easier if it's segmented into pages.	Paging	Scrolling	Tied
Pagination advantages or print preference	Although I found both methods easy to use, I have a preference for pagination, which makes the experience more like reading a magazine or book, which would be my preferred medium.	Paging	Paging	Tied
Pagination advantages or print preference	I prefer pagination because it feels like reading a 'real' book and feels more natural than scrolling through a long article. I also prefer to click through rather than scroll as it simply feels more comfortable. I also feel that it's somehow easier to assimilate the information in the 'bite-size' pieces of paginated text.	Paging	Scrolling	Tied
Pagination advantages or print preference	It felt less daunting to use the pagination mode. It felt like it took longer to read when I was scrolling. I prefer reading physical books, but if I'm using technology, it's a laptop (I'm getting old, so reading extensively from a smaller screen irritates my eyes).	Paging	Paging	Paging
Pagination advantages or print preference	I think I preferred reading the article in pagination mode purely due to the ease of it! One click of a button and it was done.	Paging	Scrolling	Paging

Table 4.14: Participant comments about pagination advantages.

4.2 Discussion

Though finding a significant increase in reading speed in the scrolling mode was not anticipated at this study's outset, it does follow results in some existing studies reviewed in *Chapter 2*. Based on the results and an earlier review of the literature, we can offer three potential explanations for this finding.

One is that the participants, like many web users, have developed unconscious strategies and/or a capacity for mapping content on screens in the scrolling mode. In the words of one participant, "scrolling just seemed easier". This idea of adaptive strategies is used by Brady et al. (2018) to explain similar results in their study. They had posited that the lack of fixed spatial markers while scrolling would force readers to remember rather than map content. However, when they found limited impacts from scrolling in their experiment (and no effect in most groups surveyed), they posited that web users have now developed "compensatory strategies" to deal with extra working memory demands required by scrolling. Similarly, accounts from participants in the present study suggest some web users employ other strategies to reduce eye strain or maintain their attention (such as using scrolling to fix the text in one place).

A second explanation may be related to a learning bias introduced by the novelty of the pagination mode. While many users are accustomed to pagination on ebook reader devices and smartphones, instructing participants to use a keyboard shortcut to turn a web page may have created an additional delay as they became accustomed to it. In the present study, one response provides support for this:

> "Reading the entire page [in pagination mode] before moving on with the keypad [...] wasn't entirely comfortable [...] as I had to look down to find the key I needed"

Previous research has solved for this problem by applying pre-training; however, that was not possible in this study.

A third explanation may be that properties of the scrolling navigation style lend themselves to fluid reading or perhaps even skim reading. For example, in this study, when participants were first presented with the pagination mode, the text is chunked so that users can only see a discrete part of the text that they need to "complete" before they move forward. Not so with the scrolling mode, where users can continually scroll ahead of their line by line progress on the text itself to "peek" ahead.

Though all of these effects may have contributed to increased speed in the scrolling condition, the first explanation - that of compensatory strategies - seems the most likely. Another study by Chaparro and Baker (2003) also found that its paging condition produced significantly slower times than that study's scrolling condition. Evidently, even as far back as 2003 participants were reporting that they had already become used to scrolling whilst navigating web documents, adding that paging was "too broken up". Given the continued growth and ubiquity of web and computer usage in the years since, it is possible that capacity for compensatory strategies may have continued to increase among the typical respondent, particularly in an online experiment. Likewise, in freeform responses in this present study, some participants implied that scrolling allowed them to progress more fluidly: "[while] using pagination there was a halt at the end of the page and it broke the flow of the article."

5. CONCLUSION

5.1 Research Overview

This research explored how scrolling and pagination navigation modes impacted upon reader performance and on reader experience. In the literature review, this study presented an account of continuous imaginary reading and how the design of digital reading experiences may impact working memory. Further, reviews were conducted of a large number of JavaScript libraries, recent CSS standards, and different methods were assessed for emulating dynamic reflowable pagination. A minimal performant reader application was developed and deployed, along with a step-by-step blueprint of how to replicate this feature.

In the sample studied under experimental conditions, this study produced the unusual finding that the scrolling presentation style resulted in an increased rate of reading among participants at a significant level. The research also showed weak evidence (below the level of significance) that presentation in a paginated style may be preferred by a majority of users, offering potential advantages in comprehension.

5.2 Problem Definition

5.3 Design/Experimentation, Evaluation & Results

The outcomes from this study suggest that there is no single ideal mode to present long-form news articles on the web to sustain attention and provide for efficient reading. Some readers may prefer to read in an efficient, timesaving way (which scrolling may afford), while others may prefer to read in a way that optimises for comprehension or provides more pauses that reduce eye strain (which either paging or scrolling might afford depending on existing strategies and competencies digital readers may have).

As Chaparro & Baker (2008) suggest in their study which looked at optimal column counts, the best solution may be to provide users with options to “tailor information displays” to suit both their reading needs and existing competencies. Thus the best reader experience for online long-form articles might not be derived from designing websites with either a fixed scrolling or pagination style navigation. Instead, with the help of JavaScript, AJAX, and emerging CSS standards, the optimal solution may be to give users the choice by providing user interface options that allow them to seamlessly (and instantly) switch between pagination and scrolling styles on the fly, without losing their place in online documents.

5.4 Contributions and impact

This study is unique in that it is the first browser-based, fully-remote research design to explore this intersection of digital reading and navigation presentation style.

Both the research design and the code for the minimal paginated reader application offer a template for future researchers to replicate this study and iterate upon its advantages and limitations.

5.5 Future Work & recommendations

Interestingly, this study showed a higher mean reading rate than previous research into reading. In a meta-analysis of 190 studies on reading, Brysbaert (2019) estimated an average reading rate for adults in English of 238 WPM. By contrast, the average reading rate found in this present study was 267 WPM. While this reading rate may be related to the performance-based rewards deployed in recruiting participants, it might also indicate that certain fixed modes of reading (i.e. long reads) result in higher reading performance.

Several of the measures used in this experiment suffered from a limited sample size or precision in tools. Future research into presentation style should avoid these limitations, particularly by increasing the quantity and rigour of comprehension questions and applying pre-training.

While researching the two conditions studied in this thesis, a third condition appeared frequently in the literature: multi-column page layouts. Traditionally, websites have shied away from column-based layouts due to the lack of CSS standards supporting reflowing text in these kinds of layouts. The current CSS standard largely supports this on a single screen, but lacks a feature to enable text reflow within a container of a fixed height to inject into a further container below the fold. However, with the paginated presentation style used in this research, it may be that a combination of pagination and columns which make full use of widescreen displays can further enhance reader experience, and this merits consideration for further research.

BIBLIOGRAPHY

Andrew, R. (2019). *Breaking Boxes With CSS Fragmentation*. Smashing Magazine. https://www.smashingmagazine.com/2019/02/css-fragmentation/

Behler, A., & Lush, B. (2010). Are You Ready for E-readers? *The Reference Librarian*, *52*, 75–87. https://doi.org/10.1080/02763877.2011.523261

Brady, K., Cho, S. J., Narasimham, G., Fisher, D., & Goodwin, A. (2018). Is Scrolling Disrupting While Reading? *Rethinking Learning in the Digital Age: Making the Learning Sciences Count*, *1*, 8. https://doi.dx.org/10.22318/cscl2018.152

British Dyslexia Association. (2022). *Dyslexia friendly style guide*. British Dyslexia Association. https://www.bdadyslexia.org.uk/advice/employers/creating-a-dyslexia-friendly-workplace/dyslexia-friendly-style-guide

Britton, N. (2020). What Are The Best Fonts For Reading Online. *Thrive.Design*. https://thrive.design/best-fonts-for-reading-easiest-to-read-online-design-fonts/

Brysbaert, M. (2019). How many words do we read per minute? A review and meta-analysis of reading rate. *Journal of Memory and Language*, *109*. https://doi.org/10.1016/j.jml.2019.104047

Budiu, R. (2014). *Memory Recognition and Recall in User Interfaces*. Nielsen Norman Group. https://www.nngroup.com/articles/recognition-and-recall/

Chaparro, B., & Baker, J. R. (2003). The Impact of Paging vs. Scrolling on Reading Online Text Passages. *Software Usability Research Lab*. https://www.semanticscholar.org/paper/The-Impact-of-Paging-vs.-Scrolling-on-Reading-Text-Chaparro-Baker/535a87a6697b6c83730acd3ddd0867a4c15e92a0

Chaparro, B., & Baker, J. R. (2008). Is Multiple-Column Online Text Better? It Depends! *Ronmental Science*. https://www.semanticscholar.org/paper/Is-Multiple-Column-Online-Text-Better-It-Depends!-Chaparro-Baker/375f65ad6644d3c242f3e1dcf0c969bf3da346f9

Choueiry, G. (2022). How to Report the Shapiro-Wilk Test – Quantifying Health. *Quantifying Health*. https://quantifyinghealth.com/report-shapiro-wilk-test/

Copeland, L., & Gedeon, T. (2015). Visual Distractions Effects on Reading in Digital Environments: A Comparison of First and Second English Language Readers. *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction*, 506–516. https://doi.org/10.1145/2838739.2838762

Davidson, M. (2007). *Pagination and Page-View Juicing are Evil » Mike Industries*. Mike Industries. https://mikeindustries.com/blog/archive/2007/04/pagination-is-evil

D'Esposito, M., & Postle, B. R. (2015). The Cognitive Neuroscience of Working Memory. *Annual Review of Psychology*, *66*, 115–142.

Deveria, A. (2022a). *'columns' | Can I use... Support tables for HTML5, CSS3, etc*. Can I Use. https://caniuse.com/?search=columns

Deveria, A. (2022b). *CSS Regions | Can I use... Support tables for HTML5, CSS3, etc*. https://caniuse.com/css-regions

Dörnyei, Z., & Taguchi, T. (2010). *Questionnaires in second language research: Construction, administration, and processing*. Routledge.

Dyson, M. C., & Kipping, G. J. (1998). The Effects of Line Length and Method of Movement on Patterns of Reading from Screen. *Visible Language*, *32*(2), 150–181.

Encyclopedia Britannica. (2022). *New Journalism | American literary movement | Britannica*. https://www.britannica.com/topic/New-Journalism

Fessenden, T. (2018). *Scrolling and Attention*. Nielsen Norman Group. https://www.nngroup.com/articles/scrolling-and-attention/

Flesch, R. (2016). *How to Write Plain English: Chapter 2*. University of Canterbury. https://web.archive.org/web/20160712094308/http://www.mang.canterbury.ac.nz/writing_guide/writing/flesch.shtml

Good Calculators. (2022). *Flesch Kincaid Calculator | Good Calculators*. Flesch Kincaid Calculator. https://goodcalculators.com/flesch-kincaid-calculator/

Gordon, A., Quadflieg, S., Brooks, J. C. W., Ecker, U. K. H., & Lewandowsky, S. (2019). Keeping track of 'alternative facts': The neural correlates of processing misinformation corrections. *NeuroImage*, *193*, 46–56. https://doi.org/10.1016/j.neuroimage.2019.03.014

Henninger, F., Shevchenko, Y., Mertens, U. K., Kieslich, P. J., & Hilbig, B. E. (2022a). lab.js: A free, open, online study builder. *Behavior Research Methods*, *54*(2), 556–573. https://doi.org/10.3758/s13428-019-01283-5

Henninger, F., Shevchenko, Y., Mertens, U., Kieslich, P. J., & Hilbig, B. E. (2022b). *lab.js: A free, open, online experiment builder*. Zenodo. https://doi.org/10.5281/zenodo.6505226

Hillesund, T. (2010). Digital reading spaces: How expert readers handle books, the Web and electronic paper. *First Monday*. https://doi.org/10.5210/fm.v15i4.2762

Hissom, A. (2011). *History of HTML and CSS*. http://amyhissom.com/HTML5-CSS3/history.html

Jabr, F. (2013). *The Reading Brain in the Digital Age: The Science of Paper versus Screens*. Scientific American. https://www.scientificamerican.com/article/reading-paper-screens/

Kimber, W. E. (2018). *Introduction to CSS Pagination* [Division]. https://drmacro.github.io/css-pagination-book/

Kyrnin, J. (2020). *What Is CSS3? Cascading Style Sheets Level 3*. ThoughtCo. https://www.thoughtco.com/what-is-css3-3466973

Laiola Guimarães, R., Bulterman, D., Cesar, P., & Jansen, J. (2014). Synchronizing Web Documents with Style. *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web*, 151–158. https://doi.org/10.1145/2664551.2664555

Laubheimer, P. (2018). *Beyond the NPS: Measuring Perceived Usability with the SUS, NASA-TLX, and the Single Ease Question After Tasks and Usability*

Tests. Nielsen Norman Group. https://www.nngroup.com/articles/measuring-perceived-usability/

Lauinger, T., Chaabane, A., & Wilson, C. (2018). Thou Shalt Not Depend on Me: A look at JavaScript libraries in the wild. *Queue*, *16*(1), 62–82. https://doi.org/10.1145/3194653.3205288

Lewis, P. (2015). *Rendering Performance*. Web.Dev. https://web.dev/rendering-performance/

Li, L.-Y., Chen, G.-D., & Yang, S.-J. (2013). Construction of cognitive maps to improve e-book reading and navigation. *Computers & Education*, *60*(1), 32–39. https://doi.org/10.1016/j.compedu.2012.07.010

Liesaputra, V. (2010). *Realistic electronic books*. University of Waikato.

Liesaputra, V., Witten, I. H., & Bainbridge, D. (2009). Creating and Reading Realistic Electronic Books. *Computer*, *42*(2), 72–81. https://doi.org/10.1109/MC.2009.46

Mangen, A. (2008). Hypertext fiction reading: Haptics and immersion. *Journal of Research in Reading*, *31*(4), 404–419. https://doi.org/10.1111/j.1467-9817.2008.00380.x

Manjoo, F. (2012, October 1). Stop Pagination Now. *Slate*. https://slate.com/technology/2012/10/website-pagination-stories-should-load-into-a-single-page-every-time.html

Maust, S. (2015). *What the Jank? | A Faster Web*. A Faster Web. https://www.afasterweb.com/2015/08/29/what-the-jank/

MDN. (2022a). *Canvas API - Web APIs*. Mozilla Developer Network. https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API

MDN. (2022b). *Element.scrollLeft—Web APIs*. Mozilla Developer Network. https://developer.mozilla.org/en-US/docs/Web/API/Element/scrollLeft

MDN. (2022c). *Populating the page: How browsers work - Web Performance*. Mozilla Developer Network. https://developer.mozilla.org/en-US/docs/Web/Performance/How_browsers_work

MDN. (2022d). *SPA (Single-page application) - MDN Web Docs Glossary: Definitions of Web-related terms*. Mozilla Developer Network. https://developer.mozilla.org/en-US/docs/Glossary/SPA

Mirza, I. (2013). *CSS3 Columns and Paged Reflowable Content—SitePoint*. https://www.sitepoint.com/css3-columns-and-paged-reflowable-content/

Montegut, M. J., Bridgeman, B., & Sykes, J. (1997). High refresh rate and oculomotor adaptation facilitate reading from video displays. *Spatial Vision*, *10*(4), 305–322. https://doi.org/10.1163/156856897x00230

Mulder, D. (2019, July 21). *Split text into pages and present separately (HTML5)* [Forum post]. Stack Overflow. https://stackoverflow.com/q/12202324/18648429

Murakami, K., Miyashita, K., & Miyachi, H. (2021). A Study on the Relationship Between Refresh-Rate of Display and Reaction Time of eSports. In L. Barolli, M. Takizawa, T. Yoshihisa, F. Amato, & M. Ikeda (Eds.), *Advances on P2P, Parallel, Grid, Cloud and Internet Computing* (pp. 339–347). Springer International Publishing. https://doi.org/10.1007/978-3-030-61105-7_34

Nielsen, J. (2013). *Users' Pagination Preferences and 'View All'*. Nielsen Norman Group. https://www.nngroup.com/articles/item-list-view-all/

O'Hara, K., & Sellen, A. (1997). A comparison of reading paper and on-line documents. *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, 335–342. https://doi.org/10.1145/258549.258787

O'Hara, K., Sellen, A., & Bentley, R. (1999). Supporting memory for spatial location while reading from small displays. *CHI '99 Extended Abstracts on Human Factors in Computing Systems*, 220–221. https://doi.org/10.1145/632716.632853

O'Leary, R. (2021, May 5). *Is Vendor Prefixing Dead?* CSS-Tricks. https://css-tricks.com/is-vendor-prefixing-dead/

Payne, S. J., & Reader, W. R. (2006). Constructing structure maps of multiple on-line texts. *International Journal of Human-Computer Studies*, *64*(5), 461–474. https://doi.org/10.1016/j.ijhcs.2005.09.003

Pearson, J., Buchanan, G., & Thimbleby, H. (2010). HCI design principles for ereaders. *Proceedings of the Third Workshop on Research Advances in Large Digital Book Repositories and Complementary Media*, 15–24. https://doi.org/10.1145/1871854.1871860

Prolific Team. (2022a). *Data protection and privacy*. Prolific. https://researcher-help.prolific.co/hc/en-gb/articles/360009094594-Data-protection-and-privacy

Prolific Team. (2022b). *Getting consent*. Prolific. https://researcher-help.prolific.co/hc/en-gb/articles/360009500773-Getting-consent

Prolific Team. (2022c). *Prolific's Attention and Comprehension Check Policy*. Prolific. https://researcher-help.prolific.co/hc/en-gb/articles/360009223553-Prolific-s-Attention-and-Comprehension-Check-Policy

Readium. (2019). *Inject and paginate EPUB contents*. Readium-Css Documentation. https://readium.org/readium-css/docs/CSS03-injection_and_pagination.html

Readium. (2022). *Readium*. GitHub. https://github.com/readium

Redish, J. (2000). Readability formulas have even more limitations than Klare discusses. *ACM Journal of Computer Documentation*, *24*, 132–137. https://doi.org/10.1145/344599.344637

Rello, L., & Baeza-Yates, R. (2012). *Optimal Colors to Improve Readability for People with Dyslexia*. Text Customization for Readability Online Symposium. https://www.w3.org/WAI/RD/2012/text-customization/r11

Rello, L., & Bigham, J. P. (2017). Good Background Colors for Readers: A Study of People with and without Dyslexia. *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, 72–80. https://doi.org/10.1145/3132525.3132546

Rosala, M. (2020). *Rating Scales in UX Research: Likert or Semantic Differential?* Nielsen Norman Group. https://www.nngroup.com/articles/rating-scales/

Sanchez, C. A., & Wiley, J. (2009). To Scroll or Not to Scroll: Scrolling, Working Memory Capacity, and Comprehending Complex Texts. *Human Factors*, *51*(5), 730–738. https://doi.org/10.1177/0018720809352788

Scacca, S. (2018). *A Reference Guide For Typography In Mobile Web Design*. Smashing Magazine. https://www.smashingmagazine.com/2018/06/reference-guide-typography-mobile-web-design/

Schneps, M., Thomson, J., Chen, C., Sonnert, G., & Pomplun, M. (2013). E-Readers Are More Effective than Paper for Some with Dyslexia. *PloS One*, *8*, e75634. https://doi.org/10.1371/journal.pone.0075634

Shaikh, A. D., & Chaparro, B. S. (2005). The Effects of Line Length on Reading Performance of Online News Articles. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *49*(5), 701–705. https://doi.org/10.1177/154193120504900514

Singer, L. M., & Alexander, P. A. (2017). Reading on Paper and Digitally: What the Past Decades of Empirical Research Reveal. *Review of Educational Research*, *87*(6), 1007–1041. https://doi.org/10.3102/0034654317722961

Van Ouwerkerk, M. (2005). *Column Script*. http://13thparallel.com/archive/column-script/

Vanderschantz, N., Timpany, C., & Huang, J. (2018). EReader interface design for page-turning. *Proceedings of the 32nd International BCS Human Computer Interaction Conference*, 1–11. https://doi.org/10.14236/ewic/HCI2018.3

Walsh, D. (2016, August 29). Get Query String Parameters with JavaScript. *David Walsh Blog*. https://davidwalsh.name/query-string-javascript

Web Incubator CG. (2022). *Shared Element Transitions* [HTML]. Web Incubator CG. https://github.com/WICG/shared-element-transitions (Original work published 2020)

World Wide Web Consortium. (1999, June 22). *Multi-column layout in CSS, 'W3C Working Draft 22 June 1999'*. https://www.w3.org/1999/06/WD-css3-multicol-19990623

World Wide Web Consortium. (2011). *Cascading Style Sheets Level 2—7: Media types*. https://www.w3.org/TR/CSS2/media.html#continuous-media-group

World Wide Web Consortium. (2014). *CSS Generated Content for Paged Media Module*. https://www.w3.org/TR/2014/WD-css-gcpm-3-20140513/

World Wide Web Consortium. (2018). *CSS Paged Media Module Level 3*. https://www.w3.org/TR/2018/WD-css-page-3-20181018/

World Wide Web Consortium. (2021). *CSS Fragmentation Module Level 3*. https://drafts.csswg.org/css-break/

Wulf, A. (2008). Columnizer jQuery Plugin. *Adam Wulf*. https://adamwulf.me/columnizer-jquery-plugin/

APPENDIX A

Return to *3.1.1 Implementing pagination* ↑

JavaScript for reader application

```
const article = document.getElementById('article');
const previous = document.getElementById('previous');
const next = document.getElementById('next');
const progress_bar = document.getElementById('progress_bar');

// Add event listeners for navigation buttons
previous.addEventListener("click", prevPage);
next.addEventListener("click", nextPage);

// Initialise variables to track pages/columns
let columnsWidth, pageWidth, totalPages, currPage;

// Refresh all article container dimension values and counts
function calcPage() {
    // Round container width to mitigate scaling bug
    article.style.maxWidth = Math.floor(article.offsetWidth) +
"px";

    // Total width of all pages or columns
    columnsWidth = article.scrollWidth;
    document.querySelector('#columnsWidth').textContent =
columnsWidth;

    // Width of page container on current screen
    pageWidth = article.offsetWidth;
    document.querySelector('#pageWidth').textContent = pageWidth;

    // Calculate and show page count
    totalPages = Math.ceil(columnsWidth / pageWidth);
    document.querySelector('#totalPages').textContent =
totalPages;

    // Update page progress
    calcCurrPage();
}

calcPage();

// Event handler for user-initiated screen resize
window.onresize = calcPage;

// Keyboard shortcuts
document.onkeyup = (e) => {
```

```
    switch (e.which) {
        case 39:
        case 40:
        case 34:
            nextPage();
            break;
        case 37:
        case 38:
        case 33:
            prevPage();
            break;
    }
}

// Event handler for mousewheel page turns
article.onwheel = handleScroll;

function handleScroll(event) {
    event.preventDefault();
    if (event.deltaY > 0) {
        // console.log('down');
        nextPage();
    } else if (event.deltaY < 0) {
        // console.log('up');
        prevPage();
    }
}

function prevPage() {
    article.style.animation = '';
    // Increment container's horizontal position to show last
page
    article.scrollLeft = article.scrollLeft - pageWidth;
    calcCurrPage();
    article.style.animation = 'fadeIn ease 0.5s';
}

// JavaScript
function nextPage() {
    article.style.animation = '';

    // Increment container's horizontal position to show next
page
    article.scrollLeft = article.scrollLeft + pageWidth;

    // Update page progress
    calcCurrPage();

    article.style.animation = 'fadeIn ease 0.5s';
}

// Calculate page progress & update nav bar
function calcCurrPage() {
```

```
    // Convert scroll values into discrete page numbers
    let currPageNum = Math.round(article.scrollLeft / pageWidth);
    currPage = (article.scrollLeft != 0 ? (currPageNum) + 1 : 1);
    document.querySelector('#currPage').textContent = currPage;
    document.querySelector('#scrollLeft').textContent =
article.scrollLeft;

    // Update progress bar
    let currPagePercent = (100 * ((article.scrollLeft) /
(columnsWidth - pageWidth))

    progress_bar.style.width = currPagePercent + "%";
    console.log(progress_bar.style.width);
    disableButtonCheck();
}

// Disable relevant navigation button at beginning/end
function disableButtonCheck() {
    if (article.scrollLeft + pageWidth >= columnsWidth) {
        next.disabled = true;
    } else if (next.disabled === true) {
        next.disabled = false;
    }
    if (article.scrollLeft <= 0) {
        previous.disabled = true;
    } else if (previous.disabled === true) {
        previous.disabled = false;
    }
```

CSS for reader application

*Note: a modified version of this CSS was used for the reader in the online experiment in order to prevent conflicts with the CSS defined by the *lab.js* study builder.

```
* {
    box-sizing:border-box;
    margin: 0;
    padding: 0;
}

body {
    height: 100vh;
    background-color: #eedbc4;
}
.container {
  margin: 0 auto;
  padding: 0;
  border: 0;
}
```

```
main {
  padding: 0;
}

:root {
  --article-width: min(650px, 92vw);
}

#article_container {
  height: 100vh;
  background-color: #fffff6;
  margin: 0 auto;
  max-width: 710px;
  max-width: calc(var(--article-width) + var(--article-padding) +
var(--article-padding));
  box-shadow: 0 0 40px rgba(81, 81, 81, 0.2);
}

@keyframes fadeIn {
  0% {
    opacity: 0.7;
}
100% {
    opacity: 1;
 }
}

#article {
    font-family: '', Arial, Helvetica, sans-serif; /* Rubik, IBM
Plex Sans, Lexend  */
    max-width: var(--article-width);
    margin: 0 var(--article-padding);
    padding-top: calc(var(--article-padding) - 0.5rem);
    padding-bottom: 0.5rem;
    max-height: calc(100vh - 3.2rem);
    column-count: 1;
    column-width: var(--article-width);
    column-gap: 0px;
    overflow: hidden;
    background-color: #fffff6;
    line-height: 1.5;
  }

#article p {
  margin-block: 0px 0px;
  font-size: 1rem;
}

#article h1 {
  margin: 0;
  font-size: 24px;
  line-height: 1;
}

nav {
```

```
  background-color: rgba(213, 247, 255, 0.2);
  height: 3.2rem;
  width: 100%;
  align-items: center;
  display: grid;
  grid-template-columns: auto 1fr auto;
  font-size: 1rem;
  margin: 0;
  padding: 9px;
}

.nav_button {
  padding: 3px 10px;
  font-weight: bold;
  border: 0px solid #f7b05f;
  border-radius: 3px;
  color: #52411E;
  background-color: #efac606f;
  height: 100%;
  font-size: 0.8rem;
  line-height: normal;
  margin: auto;
  width: 7rem;
  box-shadow: 0 0 10px rgba(102, 102, 102, 0.15);
  transition-timing-function: ease; */
  transition: all ease-in-out .25s;
}

.nav_button:hover {
  background-color: #fff;
  box-shadow: 0 0 10px rgba(102, 102, 102, 0.15);
}

.nav_button:active  {
  background-color: #fff;
  transform: translateY(1px);
  box-shadow: 0 0 10px rgba(102, 102, 102, 0);
}

.nav_button:disabled {
  transition: none;
  background-color: #e7e3de00;
  color: darkgrey;
  border: none;
  box-shadow: 0 0 10px rgba(102, 102, 102, 0.05);
}

#status {
  height: 100%;
  width: 100%;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: space-around;
  padding: 0 10px;
  font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida
```

```
Grande', 'Lucida Sans', Arial, sans-serif;
  font-variant: small-caps;
  font-size: 1rem;
  color: rgba(0, 0, 0, 0.7);
}

#progress {
  width: 100%;
  background-color: lightgray;
  border-radius: 10px;
}

#progress_bar {
  width: 0%;
  height: 6px;
  background-color: #f7b05f;
  border-radius: 10px;
  transition: width 0.75s;
  transition-timing-function: ease;
}

#debugging {
  display: none;
}

@media only screen and (min-width: 920px) {
  #debugging {
    display: block;
    position: absolute;
    right: 2%;
    top: 2%;
    background: pink;
    padding: 10px;
  }
}
```

HTML for reader application

*Note: as above, HTML tags and certain classes or IDs below differ slightly from the experimental version.

```
<body>
    <div id="debugging">
        scrolled: <span id="scrollLeft"></span> of <span
id="columnsWidth"></span><br />
        pageWidth: <span id="pageWidth">
    </div>
    <div id="article_container">
        <article id="article" class="pagination">
            <p>Article goes here</p>>
        </article>
```

```
        <nav>
            <button id="previous" class="nav_button">←
Previous</button>
            <div id="status">
                <div id="progress">
                    <div id="progress_bar"></div>
                </div>
                <div>
                    <span id="currPage"></span> of <span
id="totalPages"></span>
                </div>
            </div>
            <button id="next" class="nav_button">Next →</button>
        </nav>
    </div>
</body>
```