ChibiPoint: Accessible Pointing for Web Applications

Alexander Birch

November 2013

Abstract

Contents

1 Introduction **3**

1.1 Context and Motivation . . . 3

1.2 Literature & Technology Review . . . 3

1.3 Problem statement and Hypothesis . . . 3

1.3.1 Problem statement . . . 3

1.3.2 Hypothesis . . . 3

1.4 Goals and Methods . . . 4

1.4.1 Goals . . . 4

1.4.2 Methods . . . 4

1.5 Results / Contributions . . . 5

1.6 Thesis overview . . . 5

2 Literature & Technology Review **6**

3 Requirements & Design **7**

4 Implementation **8**

5 Evaluation **9**

5.1 Overview . . . 9

5.2 Study 1: Usability Study . . . 9

5.2.1 Intuitivity . . . 10

5.2.2 Interface problems . . . 10

5.2.3 Implementation problems . . . 11

5.2.4 Feedback . . . 11

5.3 Study 2: Quantitative Comparison of Pointing Systems . . . 12

5.3.1 Outline . . . 12

5.3.2 Demographic . . . 12

5.3.3 Methodology . . . 13

5.3.4 Limitations . . . 14
5.3.5 Metrics . . . 14
5.3.6 Pilot Study . . . 15
5.3.7 Navigation tasks . . . 16

6 Results of larger study **19**

6.1 Overview . . . 19
6.1.1 Keystrokes . . . 19
6.1.2 Time . . . 19
6.2 Analysis used . . . 19
6.3 Keystroke Performance . . . 20
6.3.1 Task 1: [Within form] Immediate related traversal . . . 20
6.3.2 Task 2: [Within form] distant related traversal . . . 20
6.3.3 Task 3: Visually early element . . . 20
6.3.4 Task 4: Visually late element . . . 21
6.3.5 Task 5: [Within form] Spatially close, markup distant traversal . . . 21
6.3.6 Task 6: Low visual indicator of focus . . . 22
6.3.7 Task 7: Scrolled element . . . 22
6.3.8 Task 8: Link surrounded by links . . . 22
6.4 Summary . . . 23

7 Conclusions & Future work **24**

8 Glossary **25**

8.1 Acronyms . . . 25

9 Figures **26**

A Usability Study **29**

Chapter 1

Introduction

1.1 Context and Motivation

1.2 Literature & Technology Review

1.3 Problem statement and Hypothesis

1.3.1 Problem statement

Websites are designed to be used with mice. Pointing at them without a mouse is difficult, because tabbing navigation — the standards-blessed mechanism for mouseless navigation — is ineffective, unpredictable and inefficient. It is challenging to optimise websites to accommodate tabbing navigation. In practice the web remains inaccessible, despite the availability of standards for accessible development.

Even using tabbing navigation within a tab-accessible website is not an ideal browsing experience. This pointing mechanism is inherently limited, so **a more suitable mouseless pointing mechanism needs to be developed: one that is designed to cope with real websites, rather than idealized ones.**

As for why a mechanism for mouseless pointing is needed, this is due to input restrictions:

1. Some users are too disabled to use the input device offered to them
2. More suitable input hardware can be hard to prescribe or procure
3. Some devices inherently enforce a standard, limited input hardware

To solve mouseless pointing on a wide variety of input hardware, a mechanism is required that maps well to that wide variety of input hardware.

1.3.2 Hypothesis

We believe that our keyboard navigation system for the web, ChibiPoint, provides an improved mechanism for pointing as compared to tabbing.

Hypothesis 1: ChibiPoint is generally more efficient at web navigation than tabbing

We hypothesise that our system, **ChibiPoint, requires significantly fewer keypresses to navigate webpages, than does 'tabbing navigation'** (the current standards-prescribed method for keyboard navigation). 'Navigate webpages' is a broad term, as there are many classes of pointing that need to be analysed. The more detailed hypothesis is that, **given several distinct classes of pointing, ChibiPoint requires fewer (or equal) keypresses, for a large majority (>80%) of these scenarios**. Since we analyse two versions of ChibiPoint, it should be understood that we require at least one of these to outperform tabbing in a majority of tasks.

Hypothesis 2: ChibiPoint's 'flyouts' feature, reduces further the required keypresses for web navigation

We hypothesise that the novel 'flyouts' feature of ChibiPoint, which assigns hotkeys to suggested buttons on the webpage, contributes to reducing the number of keypresses required by ChibiPoint for web navigation, compared to using just its standard pointing method of hierarchically drilling through the page and pointing at elements using crosshairs. In other words, we hypothesise that ChibiPoint 'with crosshairs and flyouts enabled' performs a large majority (>80%) of pointing scenarios in fewer keypresses than ChibiPoint 'with just crosshairs enabled'.

1.4 Goals and Methods

1.4.1 Goals

We intend to demonstrate that a more effective, efficient, and predictable mouseless pointing mechanism than 'tabbing navigation' can be made for the web. The mechanism needs to be capable of achieving widespread availability — across many browsers and input systems.

The input mechanism used by the web browser extension should be input hardware-agnostic: that is, it must accept as input a language that can be mapped to by a wide variety of input methods.

1.4.2 Methods

We will produce a proof-of-concept web browser extension (for Google Chrome), that provides mouseless pointing in a variety of real-world websites. The implementation will recruit only technologies that are available to all web browsers (i.e. HTML5, CSS and Javascript). This foundation enables it inherently to be completely cross-platform and cross-browser. Any external libraries used must be free in cost, so as not to impose a price barrier to availability.

The input language for the system will be an instance of symbol-based input: keystroke sequences. Thus a keyboard-based interface theoretically serves as a proof-of-concept for other input systems that can produce a vocabulary of discrete inputs, for example gestures or spoken words. The more limited this proof-of-concept's vocabulary is, the more devices it allows to map to its space.

We will improve upon the efficiency of tabbing navigation's linear trawl, by using a hierarchical traversal.

We will solve tabbing navigation's effectiveness problems: we avoid falling into 'focus traps' by avoiding a relative traversal. Additionally we solve the feedback problem represented by 'lack of visual indicator of focus', by painting our own visual indicators on elements.

We will improve predictability of navigation by using a spatial traversal rather than tabbing navigation's markup-driven traversal.

1.5 Results / Contributions

1.6 Thesis overview

The remainder of the thesis is composed as follows:

Chapter [2], *Literature & Technology Review*, confirms the need for an alternative web pointing mechanism, and explores the approaches recommended, as well as reviewing what solutions have been attempted already.
Chapter [3], *Requirements & Design*, details the design of our system and pointing mechanism.
Chapter [4], *Implementation*, describes how we built our web browser extension.
Chapter [5], *Evaluation*, determines how our web browser extension compares to tabbing navigation, and also determines the contribution of our novel.
Chapter [7], *Conclusions & Future Work*, discusses our contribution to the field, and ways to extend the work.

Chapter 2

Literature & Technology Review

Chapter 3

Requirements & Design

Chapter 4

Implementation

Chapter 5

Evaluation

5.1 Overview

The primary system comparison we sought to pursue was 'ChibiPoint' versus 'tabbing navigation'. We aimed also to evaluate the contribution made by the 'flyouts' feature of ChibiPoint: whether it helps or hinders. As such, we evaluated three systems:

1. Tabbing navigation
2. ChibiPoint (with just the crosshairs feature enabled)
3. ChibiPoint (with both the crosshairs feature AND the flyouts feature enabled)

We aimed to quantitatively evaluate the following things:

1. Which system is the most efficient for each class of navigation tested
2. Which system is the fastest for each class of navigation tested
3. Which system is preferred by the participants
4. Which system is considered by the participants to be easiest to use
5. How reasonable is the amount of keypressing demanded by each system, according to the participants

Qualitative data was useful also, as it offered feedback on specific advantages or disadvantages of each system, as well as providing insight to usability and the reasons for user preferences.

We performed first a usability study, as a precursor to the more detailed quantitative testing. The usability study was expected to reveal how a new user approaches the system — what level of instruction is necessary, and what problems are encountered with usage. Addressing issues here would reduce problems in the larger, quantitative study that followed.

Regarding the quantitative study: a pilot study was conducted first, to ensure that the study approach was effective, before widening the participation. Omissions from that pilot study were addressed. A larger study then recruited 12 users to evaluate the systems in a counterbalanced 3×8 ANOVA (three systems, 8 navigation types).

5.2 Study 1: Usability Study

We recruited a lecturer of Computer Science for this study. The participant was chosen due to her being a known power user of keyboard navigation. Additionally she had large experience with using accessibility hardware and software. We hoped that this participant could bring out the full potential of our system, and reveal what an expert of keyboard navigation expected from a system, as well as what strategies they tried when learning said system.

A full transcript of the session can be found in Listing [A.1].

The participant controlled the system using a complex accessible keyboard, with which she was already well-trained. This was an opportunity to confirm the concept that ChibiPoint's key input does not need to come from a conventional keyboard input device. No problems piloting ChibiPoint arose from this input choice, although the mapping of 'numpad layout' to drilling direction was not possible, since there was no numpad available (hotkeys had to be mapped elsewhere). It was found that, with the chosen drilling mapping (left hand of Qwerty, QWE|ASD|ZXC), the participant did not notice the spatial connection until it was pointed out explicitly. Perhaps if the user configured the mapping themselves, they would learn the relation (although this is out of scope for the proof-of-concept). For the evaluation study, instructions will be given to the participant, pointing out the mapping.

5.2.1 Intuitivity

Generally the system was found to be quite intuitive. Exploratory use was encouraged, with the researcher disclosing only the ChibiPoint initial invocation shortcut (Cmd K).

The participant understood that the labels upon each quadrant were keyboard shortcuts. She understood immediately that these shortcuts directed ChibiPoint's search into that region. Additionally, unlabelled shortcuts were able to be guessed; she worked out within two attempts that 'Enter' was the shortcut to click the element under the crosshairs.

The participant soon noticed the flyouts, and understood them to be keyboard-activated suggestions. Even on her first use of these, her efficiency was just one keystroke short of expert performance.

The participant understood intuitively that page scrolling keys co-operate with the ChibiPoint interface; she was able to click off-screen elements by scrolling to them and then invoking ChibiPoint, without instruction.

5.2.2 Interface problems

The BBC website had complicated nesting of clickables within its navigation bar. It became difficult to distinguish which flyout referred to which button. Judgement of which element the flyout was bound to, appeared to be based on the position of the label. This was an incorrect mental model; the label's position is only a heuristic for which interface element is bound. In reality, the arrow coming off the label is the indicator of which element will be clicked. Observation of the video suggests that the arrows were barely visible in this situation; their dark arrows were flush with the (also dark) frame of the webpage. Ultimately the participant was able to recover from the situation by relying on ChibiPoint's other pointing mechanism, crosshairs. This showed that the participant learned that both could be used situationally.

More contrast would help highlight the existence of the flyout arrows. Another solution is to uniquely color-match the flyout with the element it was bound to, so that the pairing is visualized. This led us in the next version of ChibiPoint to 'paint' flyout-bound elements, in the same way ChibiPoint's crosshairs paint their target. Flyouts were also given discretely different colors to each other, for further identity.

The participant was seen to crane toward the screen to read flyouts. We decided to respond to this by increasing the size and legibility.

The participant noted that the crosshairs did not wholly emulate a mouse, since they did not trigger mouseover or hover events, nor display a 'mouseover' cursor. A cursory effort was made to address this, but no fix could be found at the time. Further work could pursue a solution for this, to improve the feedback of the system and complete the cursor metaphor.

The participant assumed incorrectly that drilling could not continue beyond the point where the quadrant labels were hidden. The grid gets smaller each time, and past a threshold the text and gridlines are removed so as not to occlude the content in the quadrants. However, drilling remains possible provided the user recalls the shortcuts. To address any misconceptions, we decided to explictly cover this behaviour in future training.

5.2.3 Implementation problems

ChibiPoint could not click buttons within Flash Player interfaces (such as 'play' buttons on web-based video). For a proof-of-concept this is unimportant. More work could investigate communicating with Flash elements.

Additionally key listener conflicts were seen when ChibiPoint was used on the Google Search page; input to ChibiPoint was intercepted by the webpage, rendering the system unusable. Again this is not important at proof-of-concept stage. Possibly the reason lies in the order in which key listeners are bound; ChibiPoint waits for the page to be loaded in its entirety before setting up its event listeners. Future work could investigate whether establishing event listeners earlier makes a difference.

A problem was encountered with detecting clickables on the *The Bath Chronicle* (`http://bath.thisisads.co.uk/register.php`) registration page. Flyouts were not able to be provided for several buttons on this page. A later investigation revealed that the clickables were created using inline DOM level 0 events, which ChibiPoint does not look for. Future work could extend clickable detection to include these.

5.2.4 Feedback

Overall, the participant was enthusiastic about ChibiPoint. She felt that she was able to learn the system "pretty fast", especially compared to "quite a lot of usability stuff". She liked the flyouts feature, expressing that "they were pretty good guesses". She understood how to point with it, and felt that the grid drilling was a "great" way to point. She said that she would prefer this system to tabbing "of course", and that it was "obviously faster because it's hierarchical. My first-years should be able to answer that question". She described the system as "pretty predictable [excepting bugs]", and compared to tabbing "absolutely [more predictable]".

As for extending the system, the participant expressed that it would be useful to be able to hide the interface when it gets cluttered. However she also conceded that the existing feature for closing the interface sufficed. She advised that a cheatsheet or documentation would be useful for discovering hidden features. We resolved to print all controls on the evaluation study instructions.

The participant also assented with the notion that she would desire sometimes a method to undo 'drilling down' ("there was, like, one time"). However this feature already existed, so we resolved simply to improve its visibility with future instructions.

5.3 Study 2: Quantitative Comparison of Pointing Systems

5.3.1 Outline

A pilot study set the sequence for the full study to follow. The main output of the study was the evaluation of pointing systems; participants were instructed to complete several tasks of the following ilk:

1. Go to a specified website.
2. Using the specified pointing system, click a specified button/page element.

After completing this set of tasks with one pointing system, the participant was asked to repeat it with another of the pointing systems under test, until all three systems had been tested. We counterbalanced the order in which pointing systems were allocated to participants, to reduce biases in learning the tasks or systems.

In addition to this evaluation of system performance, participants were asked to fill out two questionnaires. The first questionnaire was posed before the pointing tasks: participants declared their proficiencies with the computing concepts involved, as well as describing their demographic. The second questionnaire was posed after the pointing tasks, and asked the participants to give subjective feedback on the systems they used.

5.3.2 Demographic

All participants were known personally by the researcher. Participants were asked to volunteer, and offered a chocolate biscuit as recompense. A majority (9) explicitly identified as being current Computer Science students. One further was known to be a graduate of Computer Science, another pursuing an 'EngD in Digital Media' and the final participant a Physics student.

The mean average age for participants was 22.58, with a standard deviation of 2.151. The range was 21–28. These data are tabulated in Figure [9.4]. A quarter of participants were female, the rest were male (Figure [9.5]).

Proficiencies

Most participants were frequent users of computing devices; 5 using computers or smartphones for 13+ hours per day, 5 using the same for 7-9 hours, and just 2 using devices for 4-6 hours. Other categories (0-1, 2-3, 10-12) were offered, but no participants identified with these. These frequencies can be seen in Figure [9.3].

Many participants (5) preferred to navigate using a mouse, with a majority (6) recruiting hotkeys in addition to the mouse. Only one participant actually preferred keyboard controls, and no participant identified with completely requiring full mouse support or full keyboard support (Figure [9.6]).

Touch-typing proficiency was rated on a five-point scale ranging from the sentiment 'I look at every key before pressing' to 'I always type without even a reference glance at the keyboard' (Figure [9.7]. Participants on (mean) average considered themselves a 3.42. A standard deviation of 1.084 was observed. Two primary users of the Dvorak keyboard layout were asked to use the (familiar, but not preferred) Qwerty layout during this study, and rated their touch-typing with respect to the constraint that they would be typing in Qwerty during this time.

A majority (11) used mainly Windows as their desktop Operating System. Browser choice was more divided, with 7 participants using mainly Google Chrome, 3 Firefox and 1 'Other' (at the time disclosed to be Safari, which was not an option offered on the questionnaire).

Disability

Regarding participant's exposure to accessibility needs or solutions, three had used Dragon NaturallySpeaking speech-to-text. The two who elaborated on the extent of their use of speech-to-text described only limited use. No participant had — at the time, or ever — had computer-relevant vision difficulties. Two had had in the past minor experiences with RSI, with one describing "RSI from mouse use, but nothing too severe to require a large change in typing/clicking", the other elaborating "Have changed input device due to pain in very specific circumstances.". No participants had RSI at the time of the study.

5.3.3 Methodology

The following describes the methodology for the full study. Deviations exclusive to the pilot study are disclosed in its separate section.

The researcher followed a script to ensure that all instances of the study were performed the same way.

Participants were briefed on the purpose of the dissertation and of the study, as well as what participation in the study would entail. They were assured that no identifying information would be kept about them, that it was the system performance being measured rather than their personal performance, and that they were free to withdraw participation at any point. They were told that there was no desire for them to generate any particular outcome, and that we simply wanted to compare the performance of the systems. With these explained, participants were asked to sign a permission slip, and were offered also a copy in case they wished to contact the researchers after the fact.

Participants were assigned a unique number. This enabled both the questionnaires they would fill in to be related as being from the same participant. Additionally participants were assigned a sequence in which to evaluate the three systems. Biases pertaining to the order in which systems are used, were counterbalanced by testing using all sequence permutations of the three systems equally. There were 6 permutations, so the study recruited a multiple of this: 12 participants.

The study began with the first questionnaire's being allocated. This asked participants to disclose their demographic (occupation, gender and age).

The participants began the testing activity by reading instructions on the system they had been allocated. The researcher then guided them through a training scenario to confirm that they had understood the instructions. During this training period, the researcher would answer any questions and correct any mistakes that occurred. In the case of learning the system 'ChibiPoint with crosshairs AND flyouts' before learning the system 'ChibiPoint with crosshairs ONLY', participants were trained in the use of both features in turn.

After training, all test websites were opened, and the participant was directed to the attend to the first browsing context, which was navigated to the website for task 1. The researcher would point out to them the page element that needed to be clicked in this task. The pointing objectives were also printed on a piece of paper that they could refer to.

The participant completes a task by clicking the specified element using the pointing system they have been allocated. At this point, telemetry (of keypress count and timing) is automatically downloaded by the browser extension (that is, ChibiPoint — which is repurposed as a keylogger — whether its pointing features are in use or not). The researcher then confirms whether the correct button was clicked.

If the correct button was clicked (or some equivalent, such as a caption with the same hyperlink), the participant moved to the next task. If some other button was clicked by mistake, then the telemetry was discarded by the researcher, and the participant was directed to attempt that task again.

Once the participant completed all navigation tasks, they were allocated the next system to use for pointing, and given instructions and training in it (in the same way as before). They would repeat the array of tasks from before with this new system. This process was completed until all three systems were tested.

With all systems tested, the participants were finally asked to fill in a post-experiment questionnaire, describing their preferences and thoughts on the systems tested.

Upon completion, the participants were thanked, and were offered a chocolate biscuit.

5.3.4 Limitations

Only successful task completion is logged; mistakes are discarded. Thus any measurements will not necessarily be a model of 'first-time' usage; we capture instead 'beginner' usage of ChibiPoint.

In the same way, users of tabbing could perhaps improve their performance if they retried after an attempt where mistakes are made. So again this does not model 'first-time' usage of tabbing. Here we cannot assert that users are 'beginners' of tabbing, either: general keyboard navigation proficiency was assessed in the questionnaire, but from this no confident assertion can be made about the tabbing navigation experience. However, tabbing navigation is very deterministic in journey, so provided no overshooting occurs, the number of keypresses from a beginner is no different to an expert's.

We perform no analysis to assess the tabbing skill level of our participants (that is, how close to expert performance they achieve), but necessarily the tabbing performance measured is 'at least that of a beginner', and ChibiPoint performance is 'at most that of a beginner', so our comparison should be interpreted in the context of those boundaries.

Skiplinks are one exception to the non-determinism of tabbing journeys. We chose to disallow the use of these, as they are website-dependent, and would therefore not represent the performance of a general browsing mechanism. Thus variance for tabbing is arguably not as large as it could be. However the tests chosen were not ones whose performance could have been improved upon with the available skip links (with the possible exception of Test 8 on Wikipedia, but this Skiplink is broken on Google Chrome, the browser in use, so provides no benefit).

5.3.5 Metrics

The metric that was most important to study was efficiency — how many keypresses (and by extension, how much exertion) are required for a given navigation task. Secondary to this metric was time taken by each system to perform a navigation task — it is preferred, but not essential, for the system to be fast; the main priority is reducing exertion, which is connected to efficiency.

During the navigation tasks, user keypresses were recorded, as well as the times that they occurred. We recorded only keypresses relevant to pointing with the current system. For example, attempts to use the 'tab' key during use of ChibiPoint were intercepted, and answered with an alert to the user that tabbing was disallowed. We disregarded keyboard input that did not invoke functionality within tabbing or the currently activated version of ChibiPoint. Page scrolling via the keyboard was considered unrelated to pointing, and so was not recorded. Attempts to invoke flyouts were only recorded when said flyout was present; if no flyouts were created (for example because no clickables are in the specified area, or if flyouts are disabled altogether), then the keypress was disregarded.

Time was measured from the first to last keypress, so no time was measured during page load or task explanation.

The post-task questionnaire asked participants to give subjective feedback on the systems they used. Some of the feedback was quantitative — which was the preferred pointing system, how easy was each system to use, and how reasonable was the amount of key pressing required — and some of the feedback was qualitative, asking for general thoughts on the task and systems.

5.3.6 Pilot Study

The pilot study existed as a 'dry run' to catch problems in the method of the larger study, before scaling up.

We recruited a Software Research Engineer with experience developing and assessing accessibility software. His preliminary questionnaire response (Figure [9.1]) describe a proficient computer user (with 13+ hours/day of device usage, 5/5 self-rating for touch typing, and a user of hotkeys).

A list of pointing tasks was created, but these proved to be uninformative — many of them duplicated classes of navigation already tested, and overall they did not test a wide enough spread of navigation types — to the extent that tabbing's strengths in form traversal were not represented. Additionally it was not clear what the delineation was between task setup and the actual recorded portion. The larger study uses a clearer separation of actions and setup.

Though these inspired the eventual classifications we would use for tasks in the full study, they do not in themselves paint a full picture of system performance.

These tasks were as follows:

1. Search 'piano' on YouTube
   (a) Then, select first result
   (b) Then, select 'Favorite videos'
2. Search 'sport' on BBC
   (a) Then, select first result
3. Search 'pillow' on Amazon
   (a) Then, select first result
4. View first article on Engadget.com
5. Google 'slam'
   (a) Then, select first result
6. Select 'Archives' on Megatokyo.com
7. Search 'computer' on Wikipedia
   (a) Then, select 'Section 2.3: The modern computer'

Some tests (for example the Google search) were found to be invalid, as ChibiPoint could not activate on this website due to key listener conflicts. The later study withdrew such tests.

Bugs in the instrumentation were found — certain actions, such as invoking flyouts — were not being recorded, so some keypress counts were artificially low. Luckily these were predictable and could be detected before reporting results. Additionally, it highlighted the need for waiting for the first keypress before starting a timer; explanations given at the start of each task were being accounted in the measurements, but less explanation was needed for each repeat of a task.

The study highlighted the need for automation. A lot of intervention was needed to navigate to the webpages used for tasks, and start the focus in consistent places. As well, extra unmeasured pointing tasks were added just to set up the task that followed. This was labour-intensive and distracting. The larger study designed tasks to use webpages where focus began in the desired starting point (for example, inside a form). Additionally, a bookmark folder was created so that all tasks could be opened in batch. The later versions of the instrumentation also added support for detecting form traversal (which effects a 'focus' event rather than a 'click' event), so more types of navigation could be studied.

The need was seen for a record of what the next task was.

Questionnaire errors were found, with some copy-paste mistakes being identified in the questions.

The pilot study also caught the omission of instructions for the use of tabbing navigation, which was needed for a fair comparison. Additionally it was decided that an explicit training period would be necessary to ensure that all participants started with the same minimum amount of knowledge.

For shorthand, the ChibiPoint systems will be named: CX (ChibiPoint with just crosshairs feature enabled) and CX+F (ChibiPoint with crosshairs and flyouts enabled).

The participant's post-questionnaire response can be seen in Figure [9.2]. His favourite system was CX+F. Overall he found easier the use of both ChibiPoint modes (CX = 4/5, CX+F = 5/5) to tabbing (1/5). The amount of keypressing required by ChibiPoint was considered reasonable ('unreasonability' rating CX = 2/5, CX+F = 1/5), and tabbing considered completely unreasonable (5/5). He refers in feedback to its being problematic to select the intended element with crosshairs ("Crosshairs without flyouts can be difficult to accurately pinpoint the desired text."). This was a known problem with how crosshairs paints even elements which are not 'clickables'. We decided to address this by mentioning it in future training.

5.3.7 Navigation tasks

For the larger study, a concrete set of tasks was chosen, to try to represent many discrete classes of pointing.

1. Go to (bookmarked) Halifax.co.uk login page
   (a) Then, click the 'password' field
2. Go to (bookmarked) Kickstarter.com signup page
   (a) Then, click the 'Re-enter password' field
3. Go to BBC.co.uk
   (a) Then, click the 'Sport' tab
4. Go to (bookmarked) University of Bath Library's 'Computer Science' resources page
   (a) Then, click 'Useful websites'
5. Go to Youtube.com
   (a) Search for 'piano' *This is just setup; not measured.*

   (b) *The page will change, which is the true start point of the test*
   (c) Then, click the first search result

6. Go to (bookmarked) Amazon.co.uk search page for 'pillow'
   (a) Then, click the first search result

7. Go to Megatokyo.com
   (a) *Using the arrow keys, scroll the page until you see the 'Archives' button.*
   (b) Then, click 'Archives'

8. Go to (bookmarked) Wikipedia.org page for 'Computer'
   (a) Then, click 'Section 2.3: The modern computer'

Each task's recorded portion is the 'Then' step; everything before is just setup to get the focus in the desired starting point. A list of URLs used are provided in Figure [A.1].

An explanation of the task classifications follows:

1. **[Within form] Immediate related traversal**: the Halifax login form places focus automatically inside the 'Username' field. Move focus to the next item in the form, 'password'.
2. **[Within form] distant related traversal**: the Kickstarter signup page places focus automatically inside 'Full Name' field. Move focus to the final field in the medium-sized form, 'Re-enter password'. *Note: Computer must not be logged into Kickstarter at the time, as an existing login will cancel the registration process*
3. **Visually early element**: the BBC front-page has a navigation bar, which is the first visual element. Click one of the buttons within this, 'Sport'.
4. **Visually late element**: the University of Bath Library's 'Computer Science' resources page has many. many hyperlinks on it, in content and also in sidebars. Click 'Useful websites', a hyperlink that is relatively far down the list.
5. **[Within form] Spatially close, markup distant traversal**: navigating to a YouTube search page from the front-page search guarantees that your focus starts in the 'search' field. From here, the first search result is right next to the search form, but it is a far longer journey in the page markup. Click that first search result.
6. **Low visual indicator of focus**: Amazon is one of many websites where, in Google Chrome, focusing a page element will provide no visual feedback. With this being the case, click the first search result.
7. **Scrolled element**: Megatokyo hosts a tall webcomic, which fills some screens. There are buttons beneath this, but it is necessary to first scroll to see these. Click one of these off-screen elements, 'Archive'.
8. **Link surrounded by links**: the Table of Contents for a Wikipedia article contains many hyperlinks cramped together. Click one of these, '2.3 the mdern computer'.

An explanation of the pointing ramifications of these tasks follows. We describe why these classes of navigation create different situations for tabbing's relative, linear, markup traversal, compared to ChibiPoint's absolute, hierachical navigation.

1. **[Within form] Immediate related traversal**: focusing an adjacent element in an already-focused form is trivial for tabbing, which traverses the markup adjacent to the focused element. ChibiPoint does not use relative navigation, and has no such advantage.

2. **[Within form] distant related traversal**: focusing distant elements in forms is also trivial for tabbing, which traverses contiguous markup in a linear fashion. But hierarchical navigation can theoretically catch up quickly with a linear traversal — ChibiPoint may be able to perform comparably over this distance.
3. **Visually early element**: in lieu of any other starting focus, tabbing traversal begins at the start of a webpage's markup. Assuming that early visual elements will be earlier in markup (admittedly not guaranteed, since styling is powerful), navigation bars will be very early in the tab order of a webpage. Thus tabbing can have an advantage navigating to elements in navigation bars. Again, absolute hierarchical navigation does not have this advantage.
4. **Visually late element**: this is the polar opposite of the previous test. The relative nature of tabbing forces it into a longer journey when the element is visually far away. Absolute navigation is not harmed by this.
5. **[Within form] Spatially close, markup distant traversal**: this test shows that visual proximity is, at best, only a heuristic for how far away an element is in markup. Tabbing may not do as well here as it appears it should. Absolute navigation is not harmed by this.
6. **Low visual indicator of focus**: this test is likely to harm tabbing, where seeing the currently focused element is crucial to estimating how far the user is from the target, and also whether they are already on the target. Mistakes and backtracking may be seen if the situation is confusing enough. ChibiPoint produces its own focus indicators, so should not be so affected by the page markup.
7. **Scrolled element**: this is mainly a confirmation of effectiveness; early iterations of ChibiPoint did not follow the page as it scrolled, so it was seen that this was a class of navigation that a system can have varying success at. The only reason this differs from 'Visually late element' is because the position ChibiPoint is launched from affects the pointing journey; an expert user could scroll an element into the middle of the screen so that the crosshairs would begin in the right place, saving keypresses.
8. **Link surrounded by links**: this class of pointing could cause problems for ChibiPoint flyouts; many suggestions need to be squeezed into a small space. Though the keypress count might not be affected, reading time may be harmed. Additionally, in the case of tabbing, there will be more elements to tab past.

Admittedly these are not completely isolated classes; for example, visual indicator of focus happens also to use an element that is late in markup, as well as testing the general lack of visual feedback. So some keypress count will be attributed to unrelated factors. However, we feel this correctly represents the real usage context of a navigation system — the web is not so kind as to be designed as a set of isolated pointing cases.

Ultimately what we test is 'at least' the described pointing case. Unrelated factors such as the starting point in the page, or how many clickables there are along the way, will always add or subtract from the difficulty. We cannot know whether a keypress difference is caused primarily by unrelated factors or by the actual task intention.

Were a large discrepancy to be caused by such unrelated factors, then the tested system would be seen to be so vulnerable to outside influence that it could not complete the task without being overwhelmed by those factors. Perhaps capturing this is similar enough to finding that the system is bad at that class of navigation — since we found we cannot rely on this system to always cope with that class of navigation, due to its existing vulnerabilities.

Chapter 6

Results of larger study

6.1 Overview

We compared, for each of the eight navigation tasks, the number of keystrokes each system required to complete that navigation. Time was compared also, but as this was of secondary interest, less analysis was performed.

6.1.1 Keystrokes

Each system was found to have different strengths. Tabbing was found to excel significantly at just one task: traversing short distances in forms (for example, from username field to password field). For longer form traversals, it was matched or beaten by the various ChibiPoint systems, and it even failed to surpass ChibiPoint at accessing visually early elements. In the remaining five cases, both versions of ChibiPoint greatly outperformed tabbing. In all but one case, the 'flyouts' feature of ChibiPoint reduced the number of keypresses. For the remaining case, 'Visually Early Element', there was no significant difference between the versions of ChibiPoint.

6.1.2 Time

6.2 Analysis used

Keystroke results were analysed, for each task, using a repeated measures ANOVA. Greenhouse-Geisser correction was used to determine how significantly the mean keystrokes differed between systems. This correction was used because sphericity could not be assumed of our data; there is no homogeneity of variance, as tabbing variance emerges very differently to ChibiPoint variance (tabbing is deterministic excepting mistakes, whereas ChibiPoint navigation can take multiple valid routes).

After ascertaining whether there existed within that task a significant difference in performance between the systems, we performed — using Bonferroni correction — post-hoc tests pairwise between each combination of systems used. This told us whether, between any two systems, there existed a significant difference in keystrokes.

6.3 Keystroke Performance

For shorthand, the ChibiPoint systems will be named: CX (ChibiPoint with just crosshairs feature enabled) and CX+F (ChibiPoint with crosshairs and flyouts enabled).

We consider statistical significance for $p < 0.01$.

6.3.1 Task 1: [Within form] Immediate related traversal

System	Keypresses Mean (s.d.)
CX	5.67 (.492)
CX+F	3.00 (.000)
Tabbing	1.00 (.000)

Mean keystrokes differed statistically significantly between systems.
($F(1, 11) = 814.000$, $p < 0.001$).

CX keystrokes significantly greater than Tabbing ($p < 0.001$).
Analysis failed to produce significance value for CX+F vs Tabbing.

CX+F keystrokes significantly fewer than CX ($p < 0.001$).

Hypothesis 1 refuted in this case.
Hypothesis 2 supported.

6.3.2 Task 2: [Within form] distant related traversal

System	Keypresses Mean (s.d.)
CX	4.67 (.492)
CX+F	2.25 (.452)
Tabbing	4.33 (1.155)

Mean keystrokes differed statistically significantly between systems.
($F(1.574, 17.311) = 33.543$, $p < 0.001$).

CX keystrokes non-significantly different to Tabbing ($p = 1.000$).
CX+F keystrokes significantly fewer than Tabbing ($p = .001$).

CX+F keystrokes significantly fewer than CX ($p < .001$).

Hypothesis 1 supported (for CX+F only).
Hypothesis 2 supported.

6.3.3 Task 3: Visually early element

System	Keypresses Mean (s.d.)
CX	5.00 (.603)
CX+F	4.67 (.888)
Tabbing	7.00 (.000)

Mean keystrokes differed statistically significantly between systems.
($F(1.590, 17.489) = 54.057$, $p < 0.001$).

CX keystrokes significantly fewer than Tabbing ($p < .001$).
CX+F keystrokes significantly fewer than Tabbing ($p < .001$).

CX+F keystrokes non-significantly different to CX ($p = .797$).

Hypothesis 1 supported.
Hypothesis 2 not supported.

6.3.4 Task 4: Visually late element

System	Keypresses Mean (s.d.)
CX	5.25 (.452)
CX+F	3.17 (.577)
Tabbing	72.50 (1.732)

Mean keystrokes differed statistically significantly between systems.
($F(1.246, 13.701) = 15056.088$, $p < 0.001$).

CX keystrokes significantly fewer than Tabbing ($p < .001$).
CX+F keystrokes significantly fewer than Tabbing ($p < .001$).

CX+F keystrokes significantly fewer than CX ($p < .001$).

Hypothesis 1 supported.
Hypothesis 2 supported.

6.3.5 Task 5: [Within form] Spatially close, markup distant traversal

System	Keypresses Mean (s.d.)
CX	4.42 (.996)
CX+F	2.92 (.289)
Tabbing	29.75 (9.324)

Mean keystrokes differed statistically significantly between systems.
($F(1.017, 11.188) = 91.990$, $p < 0.001$).

CX keystrokes significantly fewer than Tabbing ($p < .001$).
CX+F keystrokes significantly fewer than Tabbing ($p < .001$).

CX+F keystrokes significantly fewer than CX ($p = .002$).

Hypothesis 1 supported.
Hypothesis 2 supported.

6.3.6 Task 6: Low visual indicator of focus

System	Keypresses Mean (s.d.)
CX	6.42 (3.204)
CX+F	2.33 (.492)
Tabbing	48.50 (30.485)

Mean keystrokes differed statistically significantly between systems.
($F(1.012, 11.128) = 24.176$, $p < 0.001$).

CX keystrokes significantly fewer than Tabbing ($p = .002$).
CX+F keystrokes significantly fewer than Tabbing ($p = .001$).

CX+F keystrokes significantly fewer than CX ($p = .001$).

Hypothesis 1 supported.
Hypothesis 2 supported.

6.3.7 Task 7: Scrolled element

System	Keypresses Mean (s.d.)
CX	5.83 (1.115)
CX+F	3.50 (1.087)
Tabbing	23.33 (11.155)

Mean keystrokes differed statistically significantly between systems.
($F(1.349, 14.837) = 1304.682$, $p < 0.001$).

CX keystrokes significantly fewer than Tabbing ($p < .001$).
CX+F keystrokes significantly fewer than Tabbing ($p < .001$).

CX+F keystrokes significantly fewer than CX ($p = .001$).

Hypothesis 1 supported.
Hypothesis 2 supported.

6.3.8 Task 8: Link surrounded by links

System	Keypresses Mean (s.d.)
CX	6.67 (1.371)
CX+F	3.75 (.452)
Tabbing	48.67 (2.934)

Mean keystrokes differed statistically significantly between systems.
($F(1.387, 15.257) = 2296.627$, $p < 0.001$).

CX keystrokes significantly fewer than Tabbing ($p < .001$).
CX+F keystrokes significantly fewer than Tabbing ($p < .001$).

CX+F keystrokes significantly fewer than CX ($p < .001$).

Hypothesis 1 supported.
Hypothesis 2 supported.

6.4 Summary

Hypothesis 1 is supported for both versions of ChibiPoint in tasks 1–6.
Hypothesis 1 is supported also for CX+F in task 2.
Hypothesis 1 is refuted in task 2.
CX+F reduced keypresses compared to tabbing in 7 of 8 tasks, which we consider to be a large majority (>80%).
CX reduced keypresses compared to tabbing in 6 of 8 tasks (75%), which is certainly overall an improvement, but not the large majority sought by the hypothesis.
Since one mode of ChibiPoint (CX+F) meets the requirements, Hypothesis 1 holds.

Hypothesis 2 is supported for tasks 1–2 inclusive, and 4–8 inclusive.
Hypothesis 2 is not refuted in task 3 — it is merely not supported. Thus CX+F reduced keypresses compared to CX in 7 of 8 tasks, which we consider to be a large majority (>80%), as required.
It should also be observed that in the non-supported task, no refute was seen to the claim; in all cases, CX+F either outperforms or matches CX.

Chapter 7

Conclusions & Future work

Chapter 8

Glossary

8.1 Acronyms

browsing context a window or tab in a web browser that hosts a webpage. 13

Chapter 9

Figures

Question	Response
Gender	Male
Age	23
How many hours a day do you use a computer or smartphone?	13+
How proficient are you at controlling computers with just the keyboard?	I use hotkeys in addition to a mouse/pointing device
Do you /currently/ have any motor impairments/disabilities that hinder computer usage? If so, please explain.	None
Do you presently have any difficulty reading the screen? If so, please explain.	None
Have you ever had any motor impairments/disabilities that hinder computer usage? If so, please explain.	None
What is the main desktop OS you use?	Windows
What is the main web browser you use?	Internet Explorer
Have you ever used any accessibility software or hardware to control computers? If so, please explain which, and the extent to which you used them.	Developed and used adaptive interface software for World of Warcraft.
Is there any additional information you wish to declare?	None
How proficient are you at touch-typing?	5
What is your occupation?	Software research engineer

Figure 9.1: Questionnaire response of Pilot Study participant

Question	Response
Which system did you prefer using?	ChibiPoint (with crosshairs AND flyouts)
How easy was it to point using 'tabbing navigation'?	1
How was the amount of keypressing required in 'tabbing navigation'?	5
How easy was it to point using 'ChibiPoint (with just crosshairs)'?	4
How was the amount of keypressing required in 'ChibiPoint (with just crosshairs)'?	2
How easy was it to point using 'ChibiPoint (with crosshairs AND flyouts)'?	5
How was the amount of keypressing required in 'ChibiPoint (with crosshairs AND flyouts)'?	1
Have you any other feedback?	Crosshairs without flyouts can be difficult to accurately pinpoint the desired text.

Figure 9.2: Post-experiment questionnaire response of Pilot Study participant
Ratings were all on a five-point scale, 1–5.

How many hours a day do you use a computer or smartphone?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	4-6	2	16.7	16.7	58.3
	7-9	5	41.7	41.7	100.0
	13+	5	41.7	41.7	41.7
	Total	12	100.0	100.0	

Figure 9.3: Device usage by participant

Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation
Age	12	21	28	22.58	2.151
Valid N (listwise)	12				

Figure 9.4: Age of participant

Gender of Participant

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Female	3	25.0	25.0	25.0
	Male	9	75.0	75.0	100.0
	Total	12	100.0	100.0	

Figure 9.5: Gender of participant

Participant Keyboard Navigation Proficiency

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	I prefer a mouse/pointing device	5	41.7	41.7	41.7
	I use hotkeys in addition to a mouse/pointing device	6	50.0	50.0	100.0
	I prefer keyboard controls where possible	1	8.3	8.3	50.0
	Total	12	100.0	100.0	

Figure 9.6: Keyboard Navigation Proficiency of Participants

Participants were asked to pick which of these sentiments they identified best with. The statements attempted to describe a proficiency scale for keyboard usage:

1. I need a mouse/pointing device
2. I prefer a mouse/pointing device
3. I use hotkeys in addition to a mouse/pointing device
4. I prefer keyboard controls where possible
5. I strongly prefer full keyboard support

Votes were cast only for the middle descriptors, 2–4.

Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation
Touch-Typing Proficiency	12	1	5	3.42	1.084
Valid N (listwise)	12				

Figure 9.7: Touch-Typing Proficiency of Participants

Touch-typing proficiency was rated on a five-point scale ranging from the sentiment ‘I look at every key before pressing’ to ‘I always type without even a reference glance at the keyboard’.

Appendix A

Usability Study

Listing A.1: Transcript of Usability Study

```
R: (Researcher)
P: (Participant)

====Flyouts On====

    First system, Flyouts ON.

R:  Start the system by pressing Apple K.
P:      Okay, is there an Apple on here?
R:  Yeah, I've mapped Ctrl to Apple.
P:      Okay, great.
P:      I'll do what I have to do to get started, and...
        *Presses Apple K*
        *Chibipoint interface appears*
P:      Okay.

R:  Your mission, should you choose to accept it, is to search for the Olympics.
P:      Oh, search, okay, so...
        *Presses indicated key to drill toward top-right quadrant, where search is*
        // No instruction required on how to point, or what keys to use!
P:      And now, the question is... oh, I see.
        *Continues drilling toward search box*

P:      Uh, it keeps coming...
        *Continues drilling toward search box.*
    [Field is now in crosshairs.]
P:      Ah!
P:      Am I there yet?
        *Guesses key*
P:      Nope. Do I have to hit return or something?
        *Hits return*
        *Search field is selected*
        // No instructions required!
        *Types Olympics, submits search*

P:      Who would search for the olympics... haha! Okay, now what?
P:      I might have to do the same again. Ctrl K?
        *Presses Apple K*
        *Chibipoint interface appears*
```

```
P:      Okay, what have I got to do?

R:  Let me think, what would be a good idea.. try and hit that article, Sochi 2014 [top result].
P:      Hm, oh the first Sochi 2012? Sochi 2014, dude! Haha.
P:      The one with that guy, the one with the picture.
        *Drills toward article*
        *Drills toward article*
P:      And it's missed the link..
P:      Oh, I see.
        *Activates an appropriate flyout*
        // Unprompted! Only one lookup later than expert perf.
P:      It does that. Very clever of you, that extra thing.

P:      Hm, okay. This is great! I would totally take this.
        *Presses Apple K*
        *Chibipoint interface appears*
P:      Especially as it's keyboard, and it saves you from moving to it.
P:      And it's not slower than, uh, mouse; you don't have to move it to. So that's great.
P:      I used to keep the mouse on this; it's obnoxiously slow. I used to use that in 1998, when I was working for Lego, I was so screwed up I did use that sometimes.
P:      Anyway, yes.

R:  Now click on Sport.
R:  Err, that sport.
P:      Oh, that sport up there.
        *Drills toward*
P:      Mm, (counting noises)
        *Presses wrong flyout*
P:      Oops, missed. So what's the back button?
R:  Backspace.
        *Presses back*

P:      Let's try again.
        *Presses Apple K*
        *Chibipoint interface appears*
P:      Okay.
        *Drills toward*
P:      Mm.
        *Again activates inappropriate flyout*
P:      Ah, it's, it's.. there's something wrong.
P:      I hate to tell you this, but I'm definitely clicking on the right thing. Haha.

R:  Okay, uh..
P:      I guess I could therefore try to compensate for that one, cause..
        *Drills toward*
R:  I dunno how much I should say..
        *Activates correct flyout*
P:      Yeah, okay. I compensated, but there's definitely something wrong. Haha.

R:  Okay, how about a different website; go to Amazon.
        *Changes website*
        *Presses Apple K*
        *Chibipoint interface appears*
R:  Let's do a... I dunno; let's do a search.
P:      Okay.
```

```
        *Drills toward*
        *Many flyouts appear close together with tags tying them to their buttons in various directions*
P:      Hah! That's really all over.
        *Presses appropriate flyout for search field*
R:  Hm, what's your favourite book?
        [search term redacted]
        [discussion of how to select suggested products using arrow keys on accessible input device]
        *search submitted*

R:  Try and click on [author for first result redacted].
        *Some unproductive inputs [accessible input device typing in wrong mode]*
P:      Hello?
P:      Oh, I've got to...
        *Presses Apple K*
        *Chibipoint interface appears*
        *Drills toward*
        *Presses appropriate flyout*
        //  Expert performance already!

R:  I wonder if that's enough for Amazon. Let's go to Wikipedia.
R:  Looks like it's already selected the input field. Shall we pretend it didn't?
P:      Why? Why don't you just have me go to the German [Wikipedia] or something?
        *Presses Apple K*
        *Chibipoint interface appears*
P:      Let's go to the German one.
R:  Okay.
        *Presses appropriate flyout*
        // No drilling required!

P:      Okay, now what?
        *Presses Apple K*
        *Chibipoint interface appears*
        *Drills toward search*
        *Drills toward search*
        *Drills toward search*
R:  German foods?
        *Presses appropriate flyout*
        *Searches currywurst*

R:  Let's go to the bottom; there should be some interesting markup there.
        *Scrolls to end*
R:  Alright, let's go to Über Wikipedia.
P:      Alright.
        *Drills toward*
        *Presses appropriate flyout*

    [Discussion to find an inaccessible website to navigate next]

        *Google searches Bath Chronicle.*
        *Presses Apple K*
        *Chibipoint interface appears*
        *Presses key for appropriate flyout*
        *Bug in Chibipoint means the key press is caught by Google instead*
P:      That's interesting. So the 'J' got captured.
        *More typing*
```

P: Yeah, every time you type something, Google's capturing it.
P: Yeah, so, can't... it's just totally failing now.

R: Yeah, there's still other ways to use Google with the keyboard.
R: I found this earlier. I don't know how to make my key listeners the highest priority.
R: It's JavaScript injected into the page, so it's fighting the rest of the page.
P: So we can cheat briefly.
Clicks news article of Bath Chronicle

P: So now..
Presses Apple K
Chibipoint interface appears
P: So what do you want me to click?
Drills toward top-left
R: Click 'place an advert'.
Presses appropriate flyout

R: Register. Let's see how it does with forms.
P: Okay.
Presses Apple K
Chibipoint interface appears
Drills toward Register
Closes Chibipoint accidentally
P: Oops.
Presses Apple K
Chibipoint interface appears
(Immediately) drills to original position
// Appears to be able to reproduce practiced spatial traversals quickly without pausing to think.
P: ...That's a 'Z'; okay. [Text was obscured, so appeared as numeral 7]
Drills toward
Drills toward
[No flyouts have appeared for this button; clickable detection failed, so crosshairs are the only option. It's now in crosshairs.]
Presses enter

[A form appears]
Tabs a few times to enter form
// Candidate seems to notice that both systems can co-exist. This tab journey was a short one.
P: Oops.
Presses Apple K
Chibipoint interface appears
[First 6 fields of form have flyouts suggested, without drilling.]
Presses flyout for first form element
Focuses drop-down menu
P: Heh. Yeah, thanks. [Presumably wanted menu to open, as with a real click].
Starts filling out form (unprompted) without Chibipoint
Accidentally closes window.

Restores window.
Presses Apple K
Chibipoint interface appears
Drills toward 'Address lookup' button
Drills toward [non-trivial because it's on an edge after each drill, and also because no flyouts are offered; clickables detection failed]
Drills toward [now in crosshairs]

Presses Return
Address lookup is clicked; panel appears overlaying existing content

A focusing keypress is made, possibly tab or Apple L (select address bar). It is not clear if this is intentional.
Keyboard focus leaves browser
Presses Apple K
Chibipoint interface appears
Attempts to drill, but keyboard focus is in the address bar; types into address instead.
[Much attempting to regain focus and resume drilling, but even tabbing fails]
[Resort to mouse click to put focus back in browser content pane]

Drills toward address
Drills toward
Drills toward address [now in crosshairs, though never detected by flyouts]
Presses return
Click sent, element flashes, but no change in form
// Without mouseover probing, we cannot get affordances for which element needs to be clicked, and confused results can occur.
[Attempt to verify using mouse which element is meant to be clicked]
[Panel closes]

[Recover panel by refreshing page and re-opening Address lookup]
[Clicks same element again with Chibipoint; form option remains unclicked]
P: It must want [me to click] the checkbox next to it.
Presses Apple K
Chibipoint interface appears
Drills toward
Drills toward
Drills toward
[Button is not yet in crosshairs. Drilling labels now hidden since grid is small.]
P: Hmh.
Presses return
P: Yeah, I have no idea. It doesn't seem to be..
P: So, does that..
Hovers mouse over to confirm feedback
P: [The button] doesn't seem to notice [with Chibipoint] that something's there.
P: Obviously there's no signal that [an emulated mouse cursor is above].
P: 'Cause the [cursor] changes into a hand when you hover over it.
[Repeats previous navigation:]
Presses Apple K
Chibipoint interface appears
Drills toward
Drills toward
Drills toward
[Button is not yet in crosshairs. Drilling labels now hidden since grid is small.]
Presses return
// Seems to think that when the drilling labels are gone, it is not possible to continue drilling.

Reopens Chibipoint, drills lightly into an alternative journey.
Closes Chibipoint without clicking

Reopens Chibipoint, redoing a previous journey.

P: Yeah, nothing I do seems to get the tick.
R: Try pressing [bottom-left quadrant (now unlabelled)]
Drills
R: And again.
Drills
R: You can inch closer.

R: Boxes aren't perfect sub-boxes; they grow a little bit.
P: Okay. I didn't see Z as an action, so...
R: Ah. Yes, it... I can't make text any smaller.
P: Well, is that enough information?

====Flyouts Off====

R: Now try a version with the Flyouts turned off.
P: Okay.
[On BBC homepage]
Presses Apple K
Chibipoint interface appears
R: Try and watch Top Gear. [This is at the very bottom edge of the grid]
P: Okay, down there.
Drills toward
Drills toward
Drills toward
Presses return
[Page navigates]
// Even buttons in tricky positions were able to be clicked easily.

Presses Apple K
Chibipoint interface appears
[Top Gear Flash pane is painted wholly by Chibipoint; inner structure not detected]
Presses space bar to attempt to play video
[Page scrolls upon space bar since video does not have focus]
Drills toward Flash pane
Drills toward centre of Flash pane
Presses return
[Flash pane is clicked, but no meaningful change occurs]
Presses key to scroll up
[Page does not scroll, despite input]
P: How do you scroll up?
R: Your focus is trapped by Flash.
[Mouse recruited to recover focus]
[Page scrolled back to beginning]

Presses Apple K
Chibipoint interface appears
P: So, is the big yellow thing..? I'm sure if it could just see which button I'm trying to click..
P: No, it's not doing anything.
Presses Apple K
Chibipoint interface appears
Closes Chibipoint

R: So I'll spoil that; Flash players can't take clicks [from ChibiPoint].
P: Oh, okay.

R: So you've discovered that.

[We return to BBC homepage]
R: Try to click something on the bottom of the page - that's not on-screen.
R: Music.
Presses Apple K
Chibipoint interface appears
Scrolls using Page Down
// Unprompted, understands that page scrolling keys co-exist with ChibiPoint.
Drills toward an element in the same list as 'Music'
P: Too late; I'm going to go into 'Learning'.
P: Although it's actually not what you said.
[Actual 'Learning' text is not highlighted, but clicking the box that contains it worked in this case; did not have to drill too specifically]
Presses return
[Navigates to Learning]

R: Let's see if there's anything hard to click on; keep scrolling.
Presses Page Down
R: Click the 'Accessibility Help'.
P: Okay.
Presses Apple K
Chibipoint interface appears
Drills toward
P: Hmm..
Drills toward
[Painting of link container obscures the white hyperlinks within]
R: Oh god.
Drills toward
[Presently highlighted element is the link below Accessibility Help]
P: And then, how do you move it around? Did you say the Z key..
R: The keybindings continue to be exactly the same.
P: Oh, but I didn't memorise them.
R: Uhh.. they're this grid.
[Demonstrates grid of 9 keys on accessibility hardware that map to on-screen controls]
P: Oh, okay, so..
Drills toward
Presses enter
[Accessibility Help is clicked]

R: So, did you notice that they were...
P: No, I totally didn't notice that they were,
P: and I shouldn't, since there was a cue.
// Mapping was not noticed nor memorised; singular lack of incidental information.
P: Okay.
P:
P: Okay, so..
[Decides to click one of the webpage accessibility articles]
Presses Apple K
Chibipoint interface appears
Drills toward
Drills toward
Drills toward
Presses return

P: Well anyway, this doesn't seem to be a problem, so..
R: Right, I'll end.

```
====Feedback====

R: What do you think of the two systems you tried (flyouts on, off)?
P:     I couldn't tell the difference, I'm afraid.
P:     What was the... oh, I dunno... oh, the flyouts with commas and periods and things?
R: Yeah.
P:     Yeah, I liked the commas and periods and things. They were pretty good guesses. Although it was a bit cluttered. I'd like to be able to switch back and forth depending on what I was trying to do.
R: Uhh, by switch back and forth, do you mean 'hide them'?
P:     Yeah. Exactly, so like mostly it's great. Although anyway you can hide it with Ctrl-K, so I guess that's enough.

R: Was the problem that [the flyouts] obscured [what you were trying to point at]?
P:     It just looks a bit cluttered sometimes. But it was really useful, so I think I'd probably mostly just have them on. I could just imagine, I dunno some time I might want to turn them off...

R: Would it be useful to have, like, holding down SHIFT dispels the interface?
P:     What?
R: Uhm, there's a lot of heads-up display on the page; you could hold down a modifying key to hide stuff temporarily, but not lose where your grid is.
P:     Uhh, I don't know what... one thing is that Ctrl-K doesn't work very well for me, 'cause I use emacs; that's why it was after a little while when I was doing the wrong things because I doing the wrong commands, because I was confusing all the commands.
P:     So for me it would be better if it was something different, like Alt+ something, just because I hardly ever use Alt. Then I'd remember it; I could remember that. And also, when I'm on emacs, I'd be able to control it.

R: There's a limit to what keyboard shortcuts I'm allowed to choose. Actually, it's a limit to what I can 'suggest' as a keyboard shortcut, but the user can override the system ones if they manually choose them.
P:     Okay. Yeah.
R: So that could be done.

P:     Yeah, but it was really nice. It was a nice way to do it.
P:     I could imagine using that a lot.
P:     Although now I'm pretty much just using the pen tablet.
P:     But it saves you find the tablet - the pen, tablet; it's just right there.

R: Were there any things that you didn't understand?
P:     Oh yeah, like I said, I didn't notice that the grid was continuing; or rather I didn't pick up that - they just looked like random letters to me. I didn't know why you were doing it.
P:     But had I noticed... 'cause, the thing is that in the old days, my first guess would've been, was the right hand.
P:     In the old days, the arrow keys used to be right around there.
P:     I should've picked up that you were using the left hand instead, but I didn't.
R: I'm less worried about that, because it usually would be the numpad, but it's [accessible input device] mode.
```

P: Yeah, okay. The problem would be the – because they were slowing down the typist, when you're in Qwerty, all the characters that you use are mostly on the left, so you wouldn't normally expect them to use the right hand for the arrow keys.
R: Okay, so they can be swapped.

R: Did you ever desire to be able to 'undo' a drill-down, and retreat?
P: Yeah, once, and what I did was just Ctrl+K to start over, but there was , like, one time.
R: Okay. There's a feature I should've disclosed to you; there is a key that can 'back up'.
P: Okay. Yeah, no, I would've wanted that if I was, I got it.
P: But that was pretty fast; considering how long it took me to get... quite a lot of usability stuff downloads, the first time you try using a pen tablet(!), yeah, so that was, I was able to get into that; that was fine.
P: Again, I am old, and I did used to navigate with the arrow keys on the other hand(!)

R: Would you say it's intuitive?
P: It was largely intuitive; I mean, there was a couple things you didn't know. It might be nice to have, like, a little helping thing that could pop up and tell you the cheat sheet, to tell you if there's any other stuff.
P: 'Cause usually what I do is, I hack around a bit with the command, and then I go and read the manual, and see if there's anything I've missed.
P: So this was a very successful first hack-around, but then you would want somewhere to be able to read and find out what you'd missed.
R: Maybe I'll print a cheat-sheet rather than coding one.
P: Yeah, sure.

R: Were the 'clickable' suggestions – made by the flyouts – were they good enough guesses?
P: Yeah! Yeah, they were highly accurate; I was very impressed by that.
P: I especially liked that they were sometimes, you picked the wrong region, and they still guessed that it was just slightly off the region; it was like, way! Thinking outside the box.

R: Did you find that there were any things that it didn't seem to be able to see?
P: I didn't notice any in particular – except for something, you know, that one thing that we tried over and over again, and couldn't get anything to see.
R: That button didn't seem to be marked up properly.
P: Yeah.
R: It has a lot of intelligence to work out if something's clickable, but...
P: But somehow, like I said with the mouse you could hover over it and it would change into a pointy thing, so there must be something there, but it wasn't much.
R: I put in a feature to force it to 'mouseover' anything that the crosshairs are on. But it doesn't seem to... I told it to trigger the mouseovers, but it never happens.
P: I never really 'got' the crosshairs, so I hadn't realised – right until the end you showed it to me – but I didn't earlier pick up on that.

R: Yeah, it's problematic that as well as the flyouts, there's also the crosshairs.
P: Okay. I wasn't really noticing [the crosshairs], because the other system was working so well for me, so I was sort of reflecting, and concentrating on what I saw.
R: And yet, once you were switched to the system without flyouts, you were happy to use crosshairs.

P: Well, once you showed me how to use them, that you could still navigate even if there's no letters, and then I was like, okay, well that was like what I was used to before with the [accessible input device], so I learned my lesson.

R: Would you say you could understand how to get to a place?
P: Yeah, obviously; you just saw me do it(!)
R: Was the grid-splitting an intuitive way?
P: Yeah, no, I thought that was fine; that was great. Uh, yeah, that made sense. I liked that it was very bright, and there was this little {unclear}.
P: I'm probably not your best - it depends how you think about it. I'm probably not the most stereotypical - it's not like taking home a heavier[?] - somebody who doesn't use a computer - to try it.
R: Well, it's designed for [people with accessibility requirements]. You'd probably be a pretty good sample for accessibility needs.
P: Well, if it's designed for disabled people with a lot of experience with different tech, so that's what I'm saying - you'd need to find some other people who haven't had the same breadth of experience of available tech .

R: If we want to model performance of an expert, that's fine though.
P: Yeah, no, that's fine. I'm just saying that you should realise I'm not a typical user. In fact, that's one of the things that Google will tell you, probably(!) is that their problem is that all the people they hire are not at all - can't even conceive of the problems that their users have. Figuring out, for example, that keyboards {unclear}(!)
P: Uhm, yeah, we're not the normal. I'm particularly mechanically inclined , so I tend to figure these things out pretty quickly.
P: But still, it was good. I'm just saying that the hacking through it might not... well of course, to be fair, who does hack through your stuff? People who are good at hacking. Look at the manual.

R: How do you feel this system compares to navigating using tabbing around a page?
P: Oh, it's obviously faster because it's hierarchical. My first-years should be able to answer that question.
R: Would you prefer this system?
P: To tabbing? Of course, yeah. As long as it didn't interfere with anything else.

R: Bonus question: how does it compare to the mouse?
P: Uh, like I said, the main benefit is... well, there's actually two benefits. The main benefit is that for me, because of whatever - I don't know if everybody has this - is that especially by the time you get your hand over, and you find the mouse, that sort of thing, it was just much faster to have it right on the keyboard.
P: So, that's... and the other thing is that, honestly, sometimes it's really hard to click on, so being able to choose a letter is just better than , than just, life, you know, even with a pointing device.
P: But, yeah. So, basically, dunno, I'd have to try it. Uh, it might be... the other, the downside is that one of the problems with typing injuries is that you are too much in the same posture doing the same thing all the time, so it is good to use other devices if you have them, just to move around a bit.
P: In fact especially when you have to go searching for the pen, that's always good.

R: Would you use it alongside tabbing?
P: Tabbing? No, probably I would just use this, once I was used to it. I probably wouldn't use tabbing unless [Chibipoint] was breaking.

P: But if you see that [what you want to focus] is the next thing, then you tab because you have no reason to go through a hierarchy. 'Cause it was one, if it is in a form, then it's one keystroke to get to the next thing. If you [use Chibipoint, it's] always 2 keystrokes because you first have to call it up, and then you have to go again.

P: So I would use the tab - I would always go for the fewest keystrokes. So you can answer your own question by saying how many times would you have to press tab to get to the next point.

R: Okay. So you would see yourself using the two systems together?

P: Yeah, I think so because - but, well, I - yeah, depending on what you're trying to do. But for the average page, you would never ever tab through it, but it's gonna be just how many - like I said - I'm always gonna try and use the fewest keystrokes in the shortest amount of time. So, but, that's, you know, for a form that's gonna be tab, because you have to fill everything in.

P: But for any normal page, like on the BBC page, where you're skipping around, then of course you don't wanna use the tab.

R: And, how predictable was it?

P: I thought it was pretty predictable, I found. Except from, you know, there were bugs. The capture problems.

R: Was it more predictable than tabbing?

P: Yeah. Absolutely. Who knows what's gonna - yeah, when people {unclear}, I never figured it out. Use the tab to, like, {unclear}, you're just like, why did they need that? You know. And sometimes they skip over things, that were just not so {unclear}. I do a lot of {unclear}, so you know {unclear}.

1. `https://www.halifax-online.co.uk/personal/logon/login.jsp`
2. `https://www.kickstarter.com/signup?ref=nav`
3. `http://www.bbc.co.uk/`
4. `http://www.bath.ac.uk/library/subjects/comp-sci/`
5. `http://www.youtube.com/`
6. `http://www.amazon.co.uk/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=pillow&rh=i%3Aaps%2Ck%3Apillow`
7. `http://www.megatokyo.com/`
8. `http://en.wikipedia.org/wiki/Computer`

Figure A.1: URLs used in tasks (full study)