Development of a manual for mouse-less access to Windows

Final year project submitted for the degree of BSc (Hons) Information Technology, Business Management and Language

Department of Computer Science

University of York

Richard Williamson

March 2002

Word Count: 19,098 words, as counted by Microsoft Word.
Includes Appendices A and B, but excludes Appendices C, D and E.

Table of Contents

Abstract

1 – Introduction

1.1 – History of the Graphical User Interface

The Graphical User Interface (GUI) as we consider it today was first announced in April 1981 by Xerox Parc for their 8010 Star Information System. [Johnson 1989]. In turn, this influenced the development of the Apple Lisa computer, followed soon after by the Apple Macintosh. This interface for accessing a computer was quickly copied by Microsoft for the PC, and Atari and Commodore for their computers.

> *'In many instances the display screen is too small to hold all the information a user may wish to consult at one time, and so we have developed "windows", or simulated display frames within the larger physical display.'*
>
> *'Once the windows have been created they overlap on the screen like sheets of paper.'*

[A.C. Kay 1977].

The concept of a GUI was heralded as a great step forward in making computers more usable by non-computer scientists, especially the home and small business users.

One of the most important parts of all the GUIs developed was the WIMP system – Windows, Icons, Menus, and Pointer. The pointer is used to select the other parts of the GUI, and is almost always controlled by a mouse. The mouse that we use to access our computers today was first developed in 1964 by Doug Engelbart at Stanford Research Institute [Stanford University 1999]. This mouse had two wheels on the base, and could therefore not move freely like modern mice, but was limited to movement in either the X or the Z axes – translating to X and Y axis motion of the pointer on-screen. Only relative motion of the pointer is possible – that is, you cannot 'jump' the pointer directly to an icon you wish to select. A WIMP environment is based on the 'point-and-click' idea – you move the pointer to the icon, or other control, you wish to operate, and click to select it.

A mouse makes it very easy to interact with the Windows, Icons and Menus in a GUI. Without a mouse, it is very difficult to access the GUI. For those people who cannot use a mouse, then – usually due to some disability – it follows that the introduction of domination of the GUI was a significant step backwards. Previous interfaces had relied solely on textual input in the form of syntactically correct commands that were typed using a standard keyboard.

The reason that GUIs are generally considered to be a step forward for most users is that they red

1.2 – Physical impairments

Some users have various physical impairments that might prevent them from using a mouse to access a GUI. The most severe impairments of this kind involve either the lack of a hand, because of amputation or a birth defect, or a paralysis that means they are unable to use their hands, usually due to a spinal injury.

Some people lack the manual dexterity necessary to control a mouse. They may be unable to move the mouse accurately, or have a tremor in their hands that moves the pointer unintentionally. Cerebral palsy is one example of this – it is caused by damage to the brain, and often has effects on motor control. Imagine trying to select a file by moving the pointer to the icon, and clicking the mouse button to select the file. If you lacked fine motor control, then you may not be able to halt the pointer over the icon, and clicking the button may cause the pointer to jump off the icon.

Lack of strength in the hands, or slowness when moving the mouse, can also render the mouse useless for accessing a computer.

The inability to use a mouse can sometimes be temporary; RSI (Repetitive Strain Injury) in the form of Carpel Tunnel Syndrome can cause severe pain which prevents the use of a mouse. This form of RSI is especially prevalent among those who operate a computer with a mouse for large amounts of time.

1.3 – Alternatives to the mouse

There are other devices available that control the pointer movement on the screen in a similar way to a mouse. Trackballs are quite common, but many users find them cumbersome and difficult to use. Essentially, a trackball is a mouse turned upside down, so the user manipulates the ball directly, rather than sliding motion across the mouse mat doing it indirectly. Proportional analogue and switched digital joysticks are available as well, and are suitable for some users. Alternatives that are more expensive include eye-gaze trackers, and head motion sensors, which try to track where the user is looking in order to move the pointer. These are difficult to calibrate, and it is hard to get fine control with them – which is usually the main problem with using a mouse for someone with limited manual dexterity in any case.

Although physical impairments often also have an impact on how a person uses a keyboard, the effects are usually less detrimental to the input the user generates. For people who cannot use their hands due to absence of a limb, there is the option of using other body parts, or prostheses, to press the keys on a keyboard. People have learned to use their feet to press keys, or to hold a stick in their mouth and press the keys that way.

If a user lacks manual dexterity, then keyboards with large keys can be used, or a key guard can be fitted to a standard keyboard to assist the user in pressing the correct key.

Alternative layouts on keyboards are also available, which reduce the amount of movement the hands have to make when pressing keys. This can reduce fatigue for people with lack of strength or slowness in their hands. The standard ‘QWERTY’ keyboard arrangement was developed originally for typewriters, in 1878, to prevent the type heads from jamming when typing at speed. There are much more efficient arrangements, but this is the one that has stayed with us. The Dvorak keyboard layout is one example, and uses an arrangement that attempts to reduce hand movement [Brooks 2000].

There are also devices that simulate keyboard input, but do not require the user to press the individual keys. Scanning keyboard input techniques employ a display that moves through the letters and numbers found on a normal keyboard, requiring the user only to press one key to make a selection. The way the letters are scanned through varies, but the principle is the same – to reduce the number of keys for someone with limited manual dexterity.

An entirely separate issue is those people who do not wish, by choice, to use a mouse to access Windows. Some people find the mouse to be a slow and clumsy way to access the computer. Often, problems arise when people use laptop computers. For the sake of portability, a mouse is not included with a laptop, since it requires a significant amount of desk space to use.

Alternatives are nowadays always provided, and include a miniature trackball below the space bar; a glide pad, which you use your fingers to operate by dragging your finger along the surface, to move the pointer analogously on the screen; and the touch-stick, (also known as a TrackPoint) which is like a tiny joystick positioned between the keys that you manipulate with one finger.

The problem with some of these alternatives is that users often find controlling the pointer with them difficult, especially to gain fine-level control. For these users, a guide on how to operate Windows without using the mouse or mouse-analogues would also be very useful.

There are even some speed issues, in that some tasks can be completed much more quickly using just the keyboard. One example would be if you were using a word processing application, and you wished to save your document. You can do it without even moving your hands from the keyboard using a key combination such as Ctrl + S, and then carry on typing your document. This may seem like a minor saving of time and effort, but if it is a task you repeat several dozen times a day, then it may be worth learning the keyboard alternative.

Since keyboards, modifications to keyboards, and keyboard simulating devices are reasonably ubiquitous, I have chosen to focus on keyboard access when developing this manual on how to access Windows without a mouse. There are numerous keyboard shortcuts available in Windows, and it is possible to perform any task that you could do with a mouse with the keyboard – with the main exception of art and drawing applications, which are hard to control with keys.

Accessing Windows via the keyboard in this way requires a different perspective on performing some tasks. For example, placing a file into the Recycle Bin with the mouse involves moving the pointer to the icon, clicking to select it, moving the pointer over to the Recycle Bin icon, and releasing the button to drop it. With keyboard input, you simply press the Delete key – there is no analogous ‘drag-and-drop’ operation. As mentioned above, a mouse provides relative movement (δx, δy), whereas keyboard manipulation provides the facility to directly select the icon, or other control, you want – such as pressing the ‘D’ key, which moves your selection directly to the first icon whose filename begins with D.

The inspiration for this project initially came from a letter published in *Ability*, a magazine published by the British Computer Society, in which the author bemoaned the fact that there was no literature available for disabled users on how to access Windows without a mouse. The aim of this project is to attempt to rectify that situation, by constructing a useful and easy to use guide, primarily for disabled users, but also for anyone wishing to learn how to access Windows using only the keyboard. The guide will not provide any additions to Windows in the form of any software; all the methods for accessing Windows using the keyboard are built in to the software. Many of these methods are completely unknown to the average – even the expert – computer user, and so the guide will bring them all together, and reveal the features of Windows which remain ‘hidden’ to most users.

The manual will be based only on the Microsoft Windows platform; it will explain the concepts of using the Windows interface through keyboard shortcuts and alternatives. Due to the specific nature of the keyboard alternatives built in to Windows, this guide will not be useful for users of the Macintosh, or any GUI-based Linux/Unix variants.

This report is going to be based on the creation and testing of a guide on how to access Windows without using a mouse, or similar pointing device. The next chapter discusses the design of the manual – how it was created and why it was created in that way. Chapter 3 will contain the evaluation of the manual, and the results from that. Chapter 4 will discuss the conclusions drawn from those results.

The manual itself is available in Appendix D at the end of the report.

2 – Design

2.1 – Information structures in user manuals

There are two distinct types of manual – user manuals or instruction manuals. [Allwood & Kalén 1997]. User manuals are designed to be used for reference, and spontaneous use – users who are having a specific problem, or are looking for specific information, would use this kind of manual. Instruction manuals, on the other hand, are generally designed with the assumption that the reader of the manual will sit down and attempt to use the system based on the examples and information provided. Many manuals are written with the intent that they serve both purposes. This approach provides the opportunity for the novice user and the more expert user to make use of the manual. The novice user can follow the instruction manual approach, since the information therein will all, presumably, be new to him; the expert user can dip into the manual for solutions to particular situations.

These two types of manual contain different types of information – declarative information, and procedural information. Declarative information is factual information. Procedural information is information that directly supports the performance of a task – containing conditions, actions, and the results of the actions.

> *‘Declarative information is often considered to be of little value to software manual users, for two reasons: some research results show that it is consistently skipped by users, and other research results show that declarative information does not enhance task performance.’*

[Ummelen 1997, p. 283]

In the paper quoted above, the conclusion is reached that declarative information *is* useful and important to users of manuals. Users are able to select circumstances where procedural or declarative information would be of more use to them. Declarative information is used spontaneously, and so it does not obstruct users in carrying out tasks. As long as the declarative, factual information is available readily, then it will be of use. The most obvious way to support this use is by inclusion of a detailed index, which will allow the user to find the information they need easily. Declarative information would also seem to be of more use to more expert users, who do not require the examples and step-by-step guides provided by procedural information.

Another argument for including declarative information in the manual is the ACT* theory of cognition (J. Anderson, Carnegie Mellon University), according to which, all knowledge begins as declarative information; procedural knowledge is learned by making inferences from already existing factual knowledge.

Rettig [1991] notes that until not long ago, most software documentation was limited to a reference of all the commands available to the system – not unlike the ‘man’ help system available on Unix systems. He advocates using task-oriented documentation over what he calls ‘software-oriented’ documentation, to help the user perform tasks, rather than list all the things that the software can do.

John Brockman, at SIGDOC ’90, summarized the following points about adult learners:

- Are impatient learners and want to get started quickly on something productive

- Skip around in manuals and online documents and rarely read them fully
- Make mistakes but learn most often from correcting such mistakes
- Are best motivated by self-initiated exploration
- Are discouraged, not empowered, by large manuals with each task decomposed into its subtask minutiae

The problem is how to get around these issues to help people learn to use software well with a minimum of frustration. It would seem that users tend not to sit down and read user documentation before trying out the software, since they are impatient to get started. The manual should therefore support this approach, and enable users to sit down and get started using the Windows interface via the keyboard straight away.

2.2 – Presentation of information

As important as the information presented in the manual, is the format in which it is presented. For any manual, there is the obvious option of a printed document. When considering a manual for a computer application, there is the immediate and additional option of a computer based help file, which in the case of a Windows-based computer would usually be a Windows Help file or a Compiled HTML Help file.

There are some problems with providing an online only manual. Firstly, if the manual is supposed to teach a user who cannot use a mouse how to use a computer via the keyboard, how are they supposed to install the documentation, and access it, without prior knowledge? Secondly, the target audience of this manual is relative novice Windows users. It is less likely that a user who is expert in the Windows environment would decide to learn how to access a computer without a mouse, unless they acquired a disability later in life. New users prefer to get started with software using printed materials. [Smart et al. 1998].

Therefore, the guide will be first produced as a printed manual. This is not to rule out an online version of the guide, and it would make sense to provide such a version for reference purposes, once the basic principles had been learned; if a user wished to look up a keyboard shortcut or a particular method of performing a task, they could access it more quickly via an online help version.

2.3 – Problems with creating the manual

Thimbleby [1996] has said that user manuals are usually written by technical authors after the design of a software product. This is particularly a problem when constructing that manual several years after the production of the product – Microsoft Windows – and basing the information on personal observation while using that product. As the writer of this manual, there may be information that I should include in the manual that I am unaware of, since it is sometimes easy to ‘gloss over’ procedures that you follow without thinking about them, and that have become second nature to you.

To try to avoid this, during the development of the manual, I decided to pilot a draft version of the manual. The structure of this draft manual is discussed below, but one of the aims was to try to gather verbalised information about information that participants felt should be included in the manual. In addition to this, several different sources of information about alternative keyboard usage in Windows were researched.

‘Before setting out to build a help system that ‘explains’ a difficult interface, try to identify what makes the interface difficult, and fix the problems. Once you have made your interface as clear as it can be, develop a help system that aids users as they work.’

[Apple Computers]

The problem with writing a manual for somebody else’s product is that you have no control over the quality of the software, and documenting a badly designed piece of software in an understandable fashion is difficult [Rettig 1991]. If the interface provided by Windows is ‘difficult’, it is not possible to modify the interface to make it easier to access. Microsoft has been making quite a large effort in what it terms ‘Accessibility’ – which involves making their applications accessible to all users, and mainly involves modifications and extra pieces of software designed to make that access easier, or even possible, for disabled users. This began with the Access Utility software available for Windows 2.0 [BusinessWeek 13/06/01].

Despite this, and despite Microsoft being something of a market leader in terms of software accessibility (especially when compared to the Linux variants), there are still problems with the interface – holdovers from previous versions of Windows, and the fact that essentially GUIs were designed for mouse access. Therefore, the design of the manual must work within the limits of the Windows interface. One problem that immediately stands out when considering keyboard access to Windows is the problem of widget focus. In a push button widget, for example, the indicator that the button has the focus, i.e. is the active control, is a set of dotted lines around the inside of the button. These are often extremely hard to see, especially if you have ‘lost’ the focus – in the same way that finding your mouse pointer can sometimes be difficult.

2.4 – Producing a draft manual

After these sources had been researched, the next step was to write down the knowledge that I use naturally while operating Windows. This involved using my computer to perform several tasks, while making a note of the keyboard shortcuts, and procedures I followed to access the computer. A key-monitoring program to record the keys that were pressed was also used. This data was to be used mainly as a backup, since notes were being kept – mainly verbal, but some written – as to what procedures were being followed.

Using this data, a first draft version of the manual was prepared. At this stage of the design, it would have been useless to attempt to prepare a user manual. This would have been too time-consuming, and as yet, there was not enough data on user experiences with keyboard access. The draft mainly followed an instruction manual format, and contained factual information. At the time, the information was mainly raw data; there was little attempt to order it in a particularly logical way, and no attempt to index the information. The draft manual is included in Appendix E. It includes sections on Explorer shortcuts, menus, dialogs and basic widget control, as well as Internet Explorer and Outlook Express keyboard shortcuts, since these applications are a standard part of Windows.

2.4.1 – Testing the draft manual

Testing of the draft manual involved the setting of various tasks for some participants. The tasks they were asked to perform are detailed in Appendix A, and were an attempt to get the participant to utilise as many functions of Windows and as many skills as possible in a reasonable testing period.

The test system was a standard PC loaded with Windows ’98, and which did not have a mouse plugged in to it. It had been set up beforehand to be a completely generic, standard Windows environment. MouseKeys was available for the participants to use if they wished to. The participants were able to refer to the draft manual as much as they needed throughout the testing procedure, and had read it prior to starting the tasks. The participants were encouraged to verbalise their actions as much as possible while completing the tasks, as well as provide any comments they might have on the content. At the end of the tasks, the participants were asked to provide any general comments or insights they might have concerning keyboard access to Windows, from either the tasks just completed or their own prior experience.

There was no attempt to find novice computer users for these tasks. Removing the mouse as a method of input generally causes most computer users to ‘become’ novice users. In addition, the tasks were not intended to be a test of an individual’s ability to use the Windows environment, but rather how they adapt and change their methods of input if the mouse is removed. Following this principal, if a participant did not understand or know how to perform a particular task, due to lack of knowledge about Windows itself – i.e., if they would not know how to perform the task even with the mouse – then assistance was provided in the form of information as to where in the Windows environment the task could be accomplished, e.g. “you can rename a file by opening the context menu for that file”. In addition, the tasks themselves were worded in a way to reduce the cognitive load of the user where possible, in order to concentrate on the differences of keyboard access compared to mouse access.

At the end of the testing, participants were asked to make comments on the structure of the draft manual, and how they believed it should be improved. Five participants were involved, and all said that they believed the best way to learn from a manual was by giving examples of the concepts you were discussing. Three participants also made note that diagrams and illustrations would enable the user to understand the concepts more quickly.

2.4.2 – Results of testing

A consistent feature of the results was that all participants at first attempted to use the MouseKeys interface, before moving on to using keyboard shortcuts and the control keys – Enter, Tab, Alt, Ctrl, Escape, the Windows key and the Context Menu key. When asked about this, the participants said that the MouseKeys interface was too slow and cumbersome. Participants appeared frustrated by how it attempted to emulate the mouse, but did so in a way that slowed their performance considerably.

Menu navigation appeared to be the concept that was grasped most quickly. Participants appeared to have no problems with opening menus using the Alt, Windows and Context Menu keys, and navigating using the Arrow Keys. Only one participant used the letter keys to access a menu option directly, and when asked, said that he had learnt some basic keyboard access skills after being frustrated by the glide pad present on his laptop, which was his main computer.

One participant commented that the format of the draft manual would be “fine if you knew what you were looking for”, but that you would have to read the entire manual to be certain that a piece of information was, or was not, contained. It was suggested that the format of the draft manual be retained for a reference section in the manual. The need for an index was highlighted by all users.

3 – Implementation

Following the testing of the draft manual, the next stage of the project was the creation of the full manual. Since the manual is not intended to be a research paper or similar work, the decision was made early on to adopt an open and informal style of writing, to make it easier for a novice computer user to understand. Many computer software manuals are written in this fashion, and they appear more accessible to the reader, while still containing the requisite information.

3.1 – Layout of the manual

The layout plan initially consisted of a brainstorm of all the items that needed to be included in the manual. Some of these were duplicate ideas, or similar in concept, and so were merged. Next, the items were grouped according to relevance to each other – for example, navigating a menu would be grouped with how to access the menu bar of an application.

The groups were then arranged into a set of chapters. I decided to start with a ‘first principles’ approach, such that the basic concepts of Windows would be dealt with first, before moving on to more complex concepts that require a knowledge of the simpler principles. An example would be the description of the ‘widgets’ available in Windows – push buttons, slider bars, etc. Concepts such as the use of the Windows Explorer application are dealt with later on, and that chapter makes use of the widgets in its explanations, among other basic principles.

As was noted in Section 2.4.2, the participants of the testing for the draft manual considered that it was unlikely they would sit down and read the manual prior to starting out on their computer. This is consistent with the points made by John Brockman about adult learners (see Section 2.1). Therefore, the design of the manual could not be based on this principle – the users would be likely to move around the sections of the manual based on what they needed to know at a particular time. Since these requirements would naturally be unique for each user, it was important that the information in the manual should be easily found, and well indexed.

An important feature of the manual is that it brings out clearly the elements that were marked as 'problem areas' by the participants in the pilot testing. The inclusion of tips in the manual is one way of highlighting items that would be of particular use to the user, or might be essential to successfully accessing Windows via the keyboard. These tips were separated from the rest of the text, and highlighted by marking them with special symbols (i.e. ⓘ). There are a limited number of tips; marking a piece of information as different from the main body of the text draws the reader’s attention, but highlighting too many pieces of information negates this effect. [Benest 2001].

3.1.1 – Chapter One – Introduction

The first chapter of the manual is the introduction. It simply explains briefly what the aims of the manual are – i.e., to teach users how to access Windows via the keyboard. It also states how this will be accomplished, so that the reader knows what to expect.

It is not expected that every reader will choose to read the manual in order, and so the introduction may well be skipped. This should not prove to be a major detriment to the reader.

3.1.2 – Chapter Two – MouseKeys and other Windows Accessibility features

MouseKeys is one of the Microsoft Accessibility features. It allows you to control the mouse pointer in a conventional way, but using the numeric keypad as the controlling device rather than the mouse. This chapter explains the use of it, and other pertinent Accessibility features. It details how to activate the features, and how to control the pointer using the numeric keypad, mainly by use of examples – e.g., how to click on an item, how to drag and drop an item – and a diagram which overlays the key functions while MouseKeys is active.

This chapter is at the start of the book primarily for the benefit of users who are already accustomed to using Windows with a mouse. Although users tend to find the MouseKeys interface extremely slow and cumbersome (see Section 4.x), it allows control of the Windows interface in a manner consistent with mouse access, and so could perhaps be thought of as an intermediate step, or last resort, if a user cannot accomplish what he wants to via normal keyboard access. Most of these problems would naturally occur at the start of the learning period, and so instruction on how to use MouseKeys is at the start of the manual.

3.1.3 – Chapter Three – System focus and navigation

An important concept that is dealt with in the manual is that of system focus, and moving that focus. This is dealt with at the start of the manual, since it is something that is not really considered by mouse users of Windows, as it is not readily apparent. Mouse users are familiar with having one active application window, which is marked out by the title bar of the window being a different colour than the others. They are usually unaware of the fact that individual items within the application can also have focus (see Section 2.4.2).

To illustrate this, contrast the process by which a mouse user selects a radio-button in a dialog window for an application that contains ten controls, and how a keyboard user would select that radio-button. The mouse user simply moves the mouse pointer over to the appropriate button, and clicks the left button. If the window is not active at the time, then clicking in the window shifts the focus to the window. If they then wish to select a check box in the same dialog, they move the pointer to it and click on it.

A keyboard user wanting to accomplish the same tasks would first have to ensure that the window was active (perhaps using the Alt + Tab key combination to cycle through the available windows). Then, once the window was active, they would have to move the focus inside the dialog window from the first widget to the one they wish to 'click'. This is accomplished by pressing the Tab key to cycle the focus around all the widgets in the dialog, and pressing the Space bar on the appropriate one.

Since the process of moving the focus is not apparent to the mouse user, it seems important that in teaching keyboard access to Windows to a user that this concept should be addressed. It is also perhaps the first building block in controlling Windows – without understanding the idea of focus, it is at least problematic, if not impossible, to explain how to select menus and other controls.

3.1.4 – Chapter Four - Menus

Chapter Four deals with the Windows menu system. It begins by differentiating between the two main types of menu – the application menu bar, and the context menus that many items have, which is usually accessed by right clicking on an item. After this, the two 'special' menus available – the

Start Menu, and the System Menu of an application, are explained. The reason for separating these two is clarity – they do not easily fit into either of the other categories.

The different methods of accessing the various menus are dealt with here, with the application menu bar having several ways to access the same particular menu.

After explaining the types of menu available, it is noted that the method of moving to and selecting a menu option is the same in each; thus, they are dealt with together after this point. There is a standard method given, i.e. using the arrow keys to move up and down the different items in the menu. The 'shortcut' methods are also explained – underlined letters being direct shortcuts, or using the initial letter of a menu option to cycle through all the options beginning with that letter.

3.1.5 – Chapter Five – Widgets

This chapter is the second 'building blocks' chapter. It lists and describes all the different widgets that a user will encounter in Windows, and explains how to select or manipulate these controls. It uses the concepts introduced in chapter three to explain how to select the different controls. Each explanation is accompanied by a picture example of the particular widget, in order to ensure that the user will link the method of manipulating the widget to the correct widget.

Specific differences, unusual features, or tips about each widget are highlighted.

3.1.6 – Chapter Six – Windows Explorer

This chapter draws together the principles explained in the previous chapters in order to explain the primary Windows interface – Windows Explorer. This is the main application in Windows, and is used to manipulate the files and folders usually created by other applications. There is an explanation of how to access Explorer, and a step-by-step guide of how to manipulate files and folders in various different ways.

One item that occurred in the testing of the draft manual (see Section 2.4) was how to close and resize an Explorer window without using the standard close, minimize, maximize, restore and resize widgets available in the top and bottom right-hand corners of an Explorer windows (and almost all other applications). These widgets are not available for focusing as are most other widgets in a window, and so it caused some confusion for participants used to a mouse. The alternative of the System Menu (explained in chapter four) is again highlighted here.

The traditional 'drag-and-drop' feature of a GUI, whereby you pick up a window or an icon by clicking on it and moving it to another area of the screen, is also referenced here, but only to say that it is not a method used in keyboard access (unless you wish to use MouseKeys). Cutting, copying and pasting – as you would do in a word processor – is explained instead, and the parallels between this and a word processing application are brought out.

The alternative method of moving and resizing an application window, which would use 'drag-and-drop' with a mouse, is also explained.

3.1.7 – Chapter Seven – The Desktop

Although the Desktop can be treated as simply another folder in Windows (and indeed, this is exactly what it is – stored in the individual's profile), the Desktop has some different properties which merit a separate chapter. It cannot be re-sized or closed; if you minimize all other windows, then the Desktop will still be visible; and it contains several standard – and quite important – system icons, which allow access to your computer. Although the Desktop is actually the area above the Taskbar and System Tray, I decided to include both these items in the description of the Desktop, since the word Desktop is often taken in common usage to mean the whole visible screen if all windows are minimized or closed.

Special mention is made in this chapter of how to move the focus directly to an icon on the Desktop; this was a problem that all participants of the draft manual testing experienced, since Windows seems to 'lose' the system focus occasionally.

Access to the Taskbar, and selection of the applications displayed there, is explained. The fact that the System Tray cannot be accessed via the keyboard, without using MouseKeys, is also made clear. Items for which the system focus cannot be moved to caused frustration in the participants of the draft manual testing, so I have attempted to make it clear if this cannot be done. This was also explained for the toolbar in Explorer and other applications in the previous chapter, since toolbars can similarly not be selected.

3.1.8 – Chapter Eight – Quick reference list of keyboard shortcuts

This section is included to extend the usefulness of the manual after the initial learning period is over. It is useful for those looking for a quick reminder of the particular shortcut required to complete an action. It would also provide a valuable reference for people who have already learned to use Windows through the keyboard – by reading this section, they may well pick up useful hints that could immediately increase their efficiency while using the computer.

3.1.9 – Chapter Nine – Functions in Internet Explorer and Outlook Express

Although this manual concentrates solely on the Windows interface, Microsoft's Internet Explorer web browser has been bundled with the Windows operating system since Windows '95, and the Outlook Express e-mail/newsgroup application since Windows '98. Indeed, the Internet Explorer application shares many common features with the Windows interface, due to the tight integration of the two. A full explanation of access to Internet Explorer and Outlook Express should properly be the subject of another manual, due to the scope of these to applications; however, due to their dominance of the web browser and e-mail sector, I felt it would be useful to include at least a reference section of their functions and keyboard alternatives.

3.1.10 – Index

This section is possibly one of the most important in the manual. Following Rettig [1991] and Brockman's (see section 2.1) conclusions about how people learn, it seems likely that the manual will be often used to locate specific pieces of information. The index is therefore vital to enable these types of readers to find the information they want. In addition, during testing of the draft manual participants expressed the need for an index (see Section 2.x).

3.2 – Testing of the manual

3.2.1 – The 'Think Aloud' protocol

- As users interact with the system, they say out loud what they are thinking and what they are trying to do.
- The observer listens, takes notes, asks questions.
- The session is recorded on tape enabling a formal analysis to be performed identifying the strategies adopted, the types of problems that occur and their frequency (protocol analysis).
- The observer may set specific tasks, or let the user explore.
- The observer varies the amount of prompting so as not to impede the user's performance or encourage them to make things up.
- Thinking aloud helps to **identify** the types of problems that occur and **why** they occur.

[Benest 2001]

This is the protocol used to perform testing of the manual. Participants in the testing were asked to perform a series of tasks, as in the testing of the draft manual.

Allwood [1991] also suggests this protocol as being appropriate for the evaluation of a user manual; i.e., the free verbalization of a user's comments.

3.2.2 – Test procedure

Attempts were made at using key-monitoring software to record the key-presses that the user made. However, a short test of this system found that it was hard to interpret the data, and that detailed notes still had to be taken to understand what the participant was trying to do. In addition, the two pieces of software that were tested seemed to 'miss' keystrokes, particularly if the rate of typing was fast. For this reason, it was decided to rely wholly on the think aloud protocol for the testing.

Prior to the testing, the purpose of the manual was explained to the participant. All of the participants were novice computer users, with one exception. This participant has cerebral palsy, and has been using a computer for several years at college. He cannot use a mouse due to motor impairment, and usually uses a trackball to control his computer. However, he claims not to be particularly expert in accessing Windows via keyboard shortcuts. I felt that the perspective of someone who actually cannot use a mouse (as opposed to the other participants, who normally do use a mouse) would be valuable.

Each session was recorded on Dictaphone to enable later review of the data. The computer set-up used was as in the testing of the draft manual – a generic, standard PC, with Windows '98. The tasks performed by the users are detailed in Appendix A.
Notes taken during the test procedure, as well as transcripts of the participants actions based on the Dictaphone records, are detailed in Appendix B.

The first task involved use of context and application menus, dropdown list widgets, list widgets, the Start Menu, opening and closing applications, and copying and pasting files.

The second task was primarily an exercise involving all the different types of widget available in the Windows interface.

The third task involved use of the Explorer application, and also introduced expandable list widgets, which had not been covered yet.

The final task was mainly use of a dropdown list; however, it also tested the participant's knowledge of how to navigate menus using shortcut keys.

4 – Results

4.1 – Issues raised from the testing procedure

4.1.1 – Problems encountered

The following are points raised about keyboard access during the transcript procedure of the test results.

- Subject assumed the correct text box would be already selected, without any obvious confirmation, and started typing straight away.

Upon opening a Find Files dialog box, the subject did not check that the correct text box was selected before they started typing. When asked, they said that they assumed it would be the right one, since “that was what it’s designed to do”. There was an assumption of the function of the dialog box; in this case, a correct assumption. However, it is easy to imagine problems arising if the function was not as expected – in the Find Files box, for example, typing text into the “Containing text” field might lead to a search of the contents of your hard-drive, which would be confusing to an inexperienced user.

- Subject did not appear to know about the ‘default action’ in a dialog box, which would mean she could have just pressed Enter to begin the search.

This may be due to the insufficient highlighting of the push button with the default action by Windows, or maybe that the ‘default action’ concept is not present while using a mouse.

- Subject appeared to have some problems tracking where the focus was moving to – she said that it was sometimes hard to see where it was currently, since the highlighting was fairly subtle.

This is mainly a problem with Windows; the location of the focus should be made clearer for keyboard users – perhaps an option in the Control Panel that says, “I am a keyboard user, so make the focus on widgets clearer”. This could be accomplished with add-on software.

- Subject had some problems opening the file from the Results widget at first. She could see the focus was on the widget, but did not see why the file had to be selected *inside* the widget. This was mainly due to the fact there was only one item in the list; more items would probably have got her to use the Arrow Keys to select the correct item intuitively.

This is perhaps an error in the test procedure, but nevertheless there will be occasions when the list of search results, or another list, contains only one item. Another example would be an Explorer window displaying a folder containing only one file. The fact that Windows does not automatically select the first item in the list needs to be made clearer in the manual. Whether or not Windows *should* select the first item in the list once the focus is moved to that list widget is probably individual choice.

- Subject did not use Arrow Keys as she has done so far to move the caret to the end of the document, rather going for a direct keyboard shortcut. This is probably due to some previous word processing experience, but may mean that such commonalities should be drawn out in the manual.

Any tips or hints that reduce the reader's cognitive load – using parallels from previous experiences in this case – would be beneficial to the learning process.

- Subject displayed some confusion about the difference between the context menu and the menu bar, and how to access each. This should perhaps be made clearer in the manual.

This was also a comment made by another participant. (see Section 4.2.2).

- Subject pressed Alt + Down to expand the menu, but this did not work, since she had already pressed Alt. However, when it did not work, she repeated the same key combination, apparently without realising it was an error.

The procedure for opening menus seemed confusing to some of the participants. The circumstances under which Alt should and should not be used should be made clearer in the manual.

- Subject did not seem to know at this point about the Ctrl + S combination to save a file.

The different methods of selecting menu options are already explained in some detail; some more clarity may be required.

- Next, subject wanted to close the program using the close widget on the title bar. She tried to move the focus in the normal way with the Tab key. It should be made clearer that this is not an option here.

The items that can never have focus (toolbars and the window manipulation widgets) should the highlighted more explicitly. There seems to be no reason why they cannot have focus; perhaps the sheer number of buttons available on a toolbar would make using the Tab key to cycle the focus through them unmanageable.

- Subject did not know about pressing the letter corresponding to the initial letter of the file/folder name to move the selection directly there.

This feature should perhaps be highlighted.

- Subject could not manage to get the focus onto the desktop in order to use Arrow Keys to select My Computer. This is a temperamental feature in Windows – more detail on how to move the focus to the Desktop is needed.

- Here, subject did not realise the file would not be selected after the previous file had been deleted. This is another 'feature' of Explorer that should be mentioned explicitly – how the focus is often 'lost' if you delete a file, switch to another application, and so on.

Windows often seems to 'lose' the focus. Sometimes it is possible to locate it again by pressing the Tab key repeatedly – after deleting a file, the focus disappears from the Explorer window, but can be regained in this way – but in tasks involving the Desktop, the focus seems to disappear completely. Sometimes resorting to MouseKeys and 'left-clicking' on the Desktop is the only way to regain the focus.

Another problem is how to open the context menu for the Desktop. If an icon on the Desktop is selected, there is no way to ‘deselect’ an icon, and so make it possible to access the Desktop context menu using the context menu key. This is also a problem in Explorer. Again, there seems to be no solution to this, without using MouseKeys to left-click on the Desktop or in the Explorer window.

- Subject could not see where the file had been pasted in the folder. An explicit note saying that pasted files usually appear at the end of the folder listing would be helpful.

This is just a feature of Windows – even if the Explorer window is set to 'Auto Arrange' icons, it will not reorder them until the window is refreshed or reopened.

- Subject’s use of the Delete key here suggests she did not know about overtyping highlighted text.

This is something that occurred with more than one participant. It is not a major point, and possibly personal preference – wanting to make sure the text has actually been deleted.

- Subject realised that menu shortcuts could not be used, since all options started with a parenthesis. The fact that you cannot type further characters to ‘refine’ the shortcut selection was noted – e.g., typing ‘(Be’ would move the selection to Beijing. This should be made clearer in the manual.

This is another feature of Windows. Some similarly constructed interfaces would allow you to press the initial letter, which would constrain the options to those beginning with that letter, and then refine that constraint by typing more characters. This is not the case with these widgets, and this should be made clear.

- Subject showed some confusion about the differing terms ‘Exit’ and ‘Close’.

This is a problem during the early stages of learning, in that you are comparing your current situation to ones you have faced in the past; any differences make you question whether you are following the correct procedure. In this case, the difference is negligible. Ease of use should come with familiarity and practice.

- Subject did not appear to know about pressing Delete key to delete files.

The more novice Windows users in the test group seemed to follow the methods laid out that could be applied to multiple situations; e.g. manipulating a file can always be done via the menus. Using direct shortcuts, such as pressing the Delete key to delete a file rather than selecting the 'Delete' option in a menu, would require more practice.

- Subject used letter shortcuts on this occasion after not having done so before; but only for 'marked' shortcuts, and not for moving the selection t, for example, the 'W' group of options by pressing the 'W' key.

Clarification of the use of letter shortcuts for all options in the Start Menu – and in list widgets as well – is needed.

4.1.2 – Use of the manual

The following is a list of examples uses of the menus for one participant.

- **Referred to manual** – checked how to navigate menus.
- **Referred to manual** – checked how to expand a drop down menu.
- Here, subject wanted to use the save option in the File menu. Finding a way of accessing the toolbar was never an option – she said that she knew this was not possible, having read it in the manual.
- **Referred to manual** – how to access the menu bar.
- **Referred to manual** – how to close a program window.
- Subject had noticed here that some menu options have underlined letters, and can be quickly accessed in this way.
- **Referred to manual** – how to copy and paste files.
- **Referred to manual** – how to create a shortcut in another location.
- **Referred to manual - upon seeing the expandable list widget in the Folders bar of the Explorer window, checked how to use them.**

Based on these initial tests, it seems that reading the manual prior to starting the tasks is only slightly beneficial. Reference to the manual to aid performance of a task is more common than remembering the information; this would presumably change with time and practice, although further testing would be required to ascertain this.

It is interesting to see how one participant in particular applied techniques learned early on in the test exercises to later tasks. Specifically, he seemed to grasp one concept and apply the techniques in similar situations – for example, after reading that you could move the file selection in an Explorer window using the letters corresponding to the files' initials, he applied that principle to navigating the menu system. After asking if he read this in the manual, he said "…it made sense since you can do that to select files". This is an argument for the consistency of user interface design, since the commonalities between the two interfaces (menus, and Explorer lists) clearly reduced this user's cognitive load, and made the performance of his task easier.

Some of the testing data seems like it could have been affected by the actual testing process – despite frequent reassurances that the procedure was not a test of their abilities, some of the participants in the testing seemed slightly nervous, and forgot methods and procedures they had used only minutes before.

4.2 – Free comments made by the participants

There were a number of suggestions for modifications to the manual that were made by the participants after the exercises had been completed, as well as comments on keyboard access in general.

1. "A glossary of terms would be very useful."
2. "I haven't heard of words like applet before."

Ideally, the manual should not use terms that would require detailed explanation. However, the use of some terms is unavoidable. Terms like applet can easily be replaced by other phrases, but a glossary might be more effective than a lengthy phrase that means essentially the same thing.
In order to minimize the reader's cognitive load, the glossary should be kept as short as possible.

3. "Some of the examples are too specific, which is a bit confusing – like the example mentioning the Acrobat Reader in the menus." – referring to Figure 4.1 on page 10.

Examples that are more generic should be used in the manual; if this is not possible or if the explanation loses clarity by not using a specific example (as would be the case with the explanation mentioned by this participant), then it should be made more clear to the user that this is an example case. The problem with this approach is that the reader may form a link between items incorrectly if a specific example is used, which may lead to incorrect understanding in the future.

4. "Why are the menus explained in that order – the menu bar is the main one, so surely that should be explained first."

The relative importance of the context and application menus is hard to accurately judge, but it seemed comments made by the participants while completing the exercises that the context menu was less used than the application menu. Since most of the functions available in the context menu are available in the application's menu bar (albeit less conveniently), it would seem to make sense to call the application menu bar the 'main' menu. Thus, the order of explanation in the manual should be changed.

5. "You need to explain the differences between the two types of menu better." – referring to the context and application menus.

The difference between the context and application menus was only mentioned by one participant as a problem. Making the difference clearer however would only have a positive impact on the other readers as well.

6. "Why can't I use the Ctrl + S shortcut while the menu is open?"

This is really a fault of Windows – the menus display the appropriate keyboard shortcuts beside the menu option, but if the user tries to test that shortcut while the menu is still open, then an error beep is heard. This immediately makes the user think that they have done something wrong, when in fact they have used the correct shortcut, just at the wrong time – which seems unnecessarily complex.

7. "I wish I'd had this manual while I was learning to use Windows – it's probably a bit late now."

This comment was made by the participant in the testing who has cerebral palsy, and had already learnt how to use Windows without a mouse. His point was that having already learnt how to access Windows via the keyboard in a way that works well for him already, he was unlikely to want to take the time to learn a new method of doing things. He admitted that his methods might not be the most efficient, and that he has little knowledge of keyboard shortcuts in particular, and tends to rely on accessing menus to get things done. However, since he already knows how to use Windows, he is not really willing to devote the time necessary to first 'un-learn' his methods, and then learn the methods suggested in the manual.

5 – Evaluation

5.1 – Conclusions

1. The keyboard is a viable alternative form of access for Windows.

There appears to be no task that can be performed using the mouse in Windows which cannot be completed using a keyboard. Some tasks undoubtedly take longer to perform, while others can be completed more quickly. Some functions in each method of access have no parallel in the other. An example of both of these points is the deletion of a file. Using the keyboard, you can delete a file simply by pressing the Delete key. With the mouse, you have to drag and drop the file into the Recycle Bin. This is slower than simply pressing Delete; there is also no parallel to the drag and drop action using the keyboard. Another example, which was highlighted as particularly useful by participants in the testing, is the ability to locate files easily merely by pressing the initial letter of the file name. This has no parallel in mouse access.

An exception to the rule, mentioned before, is computer based graphic design – using the keyboard to draw is not a particularly viable method.

2. The manual successfully teaches users how to access Windows without a mouse via a keyboard-only interface.

All of the participants in the testing completed the tasks without any significant difficulties. The only difficulties they had were due to their lack of knowledge of the Windows interface. The manual is not an attempt at teaching users how to use Windows successfully; there are many books that can accomplish that. However, with reference to the manual, all the participants managed to complete the tasks. In addition, all showed indications of learning and remembering the methods explained in the manual, as well as extrapolating the methods to other situations without reference to the manual.

3. The manual suffers from the same drawbacks as all manuals do.

The fundamental problem with learning from a manual is that the process of learning is not particularly suited to reading large amounts of text. Trying out the principles explained in the manual by sitting in front of a computer and experimenting is one way to get around this problem. However, a taught course is probably going to be more effective, since it is interactive, and students could ask questions and have concepts explained to them.

In addition, people tend not to want to read manuals until they encounter a problem. This could possibly lead to users gaining the bare minimum from the manual that they require to access the computer as quickly as possible. An example of this would be selecting menu options and files in Explorer by using the Arrow keys all the time – the more efficient way is to press the letter associated with the file or menu option. Using the Arrow keys works, but is slower. However, since it is a common method of access to menus, files and widgets, it is easier to learn.

4. A user who already uses a keyboard-based interface to Windows will likely have developed their own style of input, and will probably be unwilling to expend the effort required in learning a new system, even if it is more efficient.

5. The manual is of benefit to novice computer users who have not used a mouse- or other pointing device-based interface to a large extent previously.

Most of the participants in the testing were novice computer users. They demonstrated no problems with accessing the computer via the keyboard, although the methods of access are quite different to those used with a mouse. This may be because they are still learning how to use the computer, and so are more open to different methods of access, and incorporate them into their use of the computer without thinking of the keyboard as an alternative form of input – they see it more as a form of input which supplements, rather than supplants, the mouse – and vice versa.

6. The manual is also of benefit to users who have used a pointer-based interface exclusively.

Users who have not previously used keyboard access, or have not used it extensively, would not have learnt keyboard access methods that may be inefficient. If they are willing to expend the time to read the manual and practice the techniques given, then their efficiency when using Windows can be increased.

7. Users who use both a mouse and a keyboard to access Windows, depending on which is more efficient, will find the manual useful.

If a user access Windows via the mouse and also the keyboard, then it is likely that they do so for reasons of convenience, and efficiency. Switching between the mouse and the keyboard can slow performance of a task considerably; if the same procedure is followed frequently, then learning a keyboard shortcut can reduce the amount of time it takes to complete a task.

5.2 – Further work

5.2.1 - Complied HTML manual

A version of this manual that would potentially be useful would be computer based, so that the user could access it simply by pressing a key combination.

The best way of presenting this information would probably be through the standard Windows Compiled HTML Help (CHM) interface. This is the de facto standard help system on Windows, and is therefore a common interface which could be used easily by most people. This style of help system is written in HTML, and the interface is based on a simplified web browser interface. The commonalities between this system and widely used web applications make it a good choice.

The help file is indexed, so that users can easily find what they are looking for. In addition, the help system includes a search tool, which can search through the whole manual for specific terms, which is a clear advantage over the printed version.

One useful addition would be a custom coded applet that positions a text field unobtrusively on the screen, which could be activated via a key press. This would make the manual faster and more convenient to access.

5.2.2 – Focus highlighting

A further option is a background application that increases the amount of highlighting applied to widgets that have the system focus. The current Windows system is a faint dotted line to indicate focus, and a faint shadow to indicate a default action. These are often difficult to see, and so an application which links in to the operating system to increase the visibility of these indicators would be highly useful.

6 – References

[Allwood & Kalén 1997] C.M. Allwood, T. Kalén, “*Evaluating and improving the usability of a user manual*”, Behaviour and Information Technology, vol. 16, no. 1, pp. 43 – 57 (1997).

[Benest 2001] Ian Benest, “*User Interface Design Engineering*”, University of York, course notes (2001).

[BusinessWeek 13/06/01] BusinessWeek, “*A Chat With Microsoft’s Steve Ballmer*” (13/06/01). <http://www.businessweek.com/bwdaily/dnflash/jun2001/nf20010613_081.htm>

[Johnson 1989] Jeff Johnson et al., “*The Xerox "Star": A Retrospective*” (1989). <http://www.geocities.com/SiliconValley/Office/7101/retrospect/index.html>

[Kay 1977] Alan Kay, *"Microelectronics and the Personal Computer"*, Scientific American, vol. 237, no. 3, pp. 230 – 244 (1977), as quoted in Ian Benest, “*User Interface Design Engineering*”, University of York, course notes (2001).

[Rettig 1991] Marc Rettig, “*Nobody reads documentation*”, Communications of the ACM, vol. 34, no. 7, pp. 19 – 24 (1991).

[Smart 1998] Karl L. Smart, Kristen Bell DeTienne, Matthew Whiting, “*Customers’ Use of Documentation: The Enduring Legacy of Print*”, Annual ACM Conference on Systems Documentation, pp. 23 – 28 (1998).

[Stanford University 1999] Stanford University, “*Technology*” (1999). <http://unrev.stanford.edu/history/technology/body_technology.html>

[Thimbleby 1996] Harold Thimbleby, “*Creating user manuals for use in collaborative design*”, Conference on Human Factors and Computing Systems, pp. 279 – 280 (1996).

[Ummelen 1997] Nicole Ummelen, “*Declarative Information in Software Manuals: What’s The Use?*”, Annual ACM Conference on Systems Documentation, pp. 283 – 296 (1997).

7 – Bibliography

Marcus Brooks, *“Introducing the Dvorak Keyboard”*, (2000).
<http://www.mwbrooks.com/dvorak>

Susan Grimm, *“How to write computer documentation for users”*, Wadsworth (1982).

Jonathon Grudin, *“The case against user interface consistency”*, Communications of the ACM, 1164 – 1173 (1989).

William Horton, *“Designing and Writing Online Documentation: Help Files to Hypertext”*, John Wiley & Sons (1990).

Joseph Lazzaro, *"Adaptive Technologies for Learning & Work Environments"*, American Library Association (2001).

Microsoft Corp., *“Keyboard Assistance”*, (2002).
<http://www.microsoft.com/enable/products/keyboard.htm>

Microsoft Corp., *“Keyboard Shortcuts for Windows”* (2001).
<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q126449>

Microsoft Corp., *“Microsoft Windows 98 Accessibility Features”*, (2002).
<http://www.microsoft.com/enable/download/training/Windows98.doc>

Sarah Morley, *“Window Concepts: An Introductory Guide for Blind and Visually Disabled Users”*, Royal National Institute for the Blind, (1995).

Neff Walker, Judith Reitman Olson, *“Designing Keybindings to be Easy to Learn and Resistant to Forgetting Even When the Set of Commands is Large”*, Conference on Human Factors and Computing Systems, 201 – 206 (1988).

Appendix A – Test procedure

The following is a copy of the tasks that participants in the testing of the manual were asked to perform. If during the test procedure a participant had difficulties due to lack of knowledge about the Windows interface, rather than how to access it without a keyboard, help was given in the form of verbal prompts.

Overview:

You are a disabled computer user, who cannot use a standard computer mouse; however, you can use a standard keyboard. Use the manual provided to help you perform the following tasks; the MouseKeys tool will be enabled, which will allow you to control the mouse pointer using the numeric keypad; however, bear in mind that using this system is cumbersome.

While you are performing the tasks, please try to describe what it is you are attempting to do at any moment, and how you are trying to do it. If an approach does not work, try as many as you like, but try to describe what you do.

- Task 1: Editing a document.
    - Locate the file “testfile.doc” on the hard-drive, using the “Find Files” tool.
    - Open the file.
    - Add some text to the end of it.
    - Save the file, and exit the program.
    - Create a copy of the document.
    - Rename the copy “mytestfile.doc”.

- Task 2: Resetting the display.
    - Reset the resolution of the desktop to 1280 x 1024 pixels, 16-bit colour, using the Display Properties applet. Access this applet in any way you want. Confirm the change to close the applet.

- Task 3: Using expandable lists (in Explorer).
    - Start the Windows Explorer.
    - The Outlook Express window is getting in your way – minimize it.
    - Using the Folders toolbar on the left, browse to the location of “testfile.doc”.
    - Select “testfile.doc”, and delete it.
    - Close Explorer, and Outlook Express.

- Task 4: Using dropdown lists (Date/Time applet).
    - Start up the Date/Time applet in Control Panel.
    - You are taking your computer to China. Reset the time zone to reflect the time in Beijing. Expand the dropdown list to find the correct time zone.

Appendix B – Transcribed Results and Comments

Subject 1

The computer was set up with MouseKeys before the testing started, but was not activated by default.

Task 1:

- Subject did not know how to find files using the Windows interface, so guidance was given as to the location of the Find program (in the Start Menu).

1. Opened the Start Menu using the Windows key.

   ☞ **Referred to manual** – checked how to navigate menus.

2. Up Arrow key to move selection to Find option.
3. Right Arrow to open the submenu.
4. Enter to select ‘Files or Folders’.
5. Entered search string.

   - Subject assumed the correct text box would be already selected, without any obvious confirmation, and started typing straight away.

6. Used Tab to move focus to ‘Find Now’ button.

   - Subject did not appear to know about the ‘default action’ in a dialog box, which would mean she could have just pressed Enter to begin the search.

7. Enter to select the ‘Find Now’ button.

   - Subject had searched in the wrong location – when no results came up, she checked all the fields, and realised this.

8. Used Tab key to move focus to the ‘Look in’ field.

   - Subject appeared to have some problems tracking where the focus was moving to – she said that it was sometimes hard to see where it was currently, since the highlighting was fairly subtle.

   ☞ **Referred to manual** – checked how to expand a drop down menu.

9. Used Alt + Down Arrow to expand the menu.
10. Used Arrow Keys to select location.
11. Used Enter to confirm.
12. Used Tab to move focus to ‘Find Now’ button.
13. Used Enter to select button.
14. Used Tab to move focus to the Results list widget.

- Subject had some problems opening the file from the Results widget at first. She could see the focus was on the widget, but did not see why the file had to be selected *inside* the widget. This was mainly due to the fact there was only one item in the list; more items would probably have got her to use the Arrow Keys to select the correct item intuitively.

15. Used Arrow Keys to select the file in the Results list.
16. Used Enter to open the file.
17. Used End key to move the caret to the end of the text.

- Subject didn't use Arrow Keys as she has done so far to move the caret to the end of the document, rather going for a direct keyboard shortcut. This is probably due to some previous word processing experience, but may mean that such commonalities should be drawn out in the manual.

18. Typed some text.

- Here, subject wanted to use the save option in the File menu. Finding a way of accessing the toolbar was never an option – she said that she knew this was not possible, having read it in the manual.
- ☞ **Referred to manual –** how to access the menu bar.
- Subject displayed some confusion about the difference between the context menu and the menu bar, and how to access each. This should maybe be made clearer in the manual.

19. Used Alt key to select the menu bar.
20. Used Alt + Down Arrow to try to expand the menu.

- Subject pressed Alt + Down to expand the menu, but this didn't work, since she had already pressed Alt. However, when it didn't work, she repeated the same key combination, apparently without realising it was an error.

21. Used Arrow Keys to move to 'Save'.
22. Used Enter to select 'Save'.

- Subject did not seem to know at this point about the Ctrl + S combination to save a file.
- Next, subject wanted to close the program using the close widget on the title bar. She tried to move the focus in the normal way with the Tab key. It should be made clearer that this is not an option here.

23. Pressed Tab key to try to Tab to the close widget.

- ☞ **Referred to manual –** how to close a program window.

24. Used Alt + Spacebar to open System menu.
25. Pressed 'C' to select Close in the menu.

- Subject had noticed here that some menu options have underlined letters, and can be quickly accessed in this way.

26. Used MouseKeys to move pointer to and open My Computer icon on Desktop.

- Subject could not manage to get the focus onto the desktop in order to use Arrow Keys to select My Computer. This is a temperamental feature in Windows – more detail on how to move the focus to the Desktop is needed.

27. Used Arrow Keys to move to C: Drive.
28. Used Enter to open C: Drive.
29. Used Arrow Keys and Enter repeatedly to move to the location of the file.

- Subject did not know about pressing the letter corresponding to the initial letter of the file/folder name to move the selection directly there.
- ☞ **Referred to manual** – how to copy and paste files.

30. Used Ctrl + C to copy the file to the clipboard.
31. Used Ctrl + V to paste the file in the current folder.

- Subject could not see where the file had been pasted in the folder. An explicit note saying that pasted files usually appear at the end of the folder listing would be helpful.
- Subject did not know how to rename files using the Windows interface, so prompting was given by saying the option was available in the right-click context menu.

32. Used context menu key to open menu.
33. Pressed ‘M’ to select ‘Rename’ option.
34. Pressed Delete key to delete text.

- Subject’s use of the Delete key here suggests she did not know about overtyping highlighted text.

35. Pressed Enter to confirm name change.

☞ **In Task 1, Subject 1 referred to the manual 5 times.**

Task 2:

- Subject was already aware of how to access the Display Properties applet from the Desktop context menu.

1. Used context key to open Desktop context menu.
2. Used Arrow Keys to select ‘Properties’.
3. Used Enter to confirm.
4. Used Tab key to move the focus to the Tab panes widget.

- Subject did not at first remember that Shift + Tab moves the widget focus in a dialog back, and that this is the best way to access the Tab panes widget. After moving the focus past the widget required, subject remembered, and used Shift + Tab to move the focus back.

5. Used Right Arrow to move to Settings tab.
6. Used Tab key to move focus to Colors menu.
7. Tried Ctrl + Down Arrow to expand drop down menu.

- This was simply confusion between which meta key to use – Alt or Tab.

8. Used Alt + Down Arrow to expand drop down menu.
9. Used Arrow Keys to select option.
10. Used Enter to confirm.
11. Used Tab key to move focus to Screen Area slider widget.
12. Used Arrow Keys to adjust Screen Area.

- This seemed to be an intuitive way to manipulate this widget.

13. Used Tab key to move focus to OK button.
14. Pressed Enter to confirm.

- Subject did not appear to know about the default action in a dialog box.

☞ **In Task 2, Subject 1 did not refer to the manual at all.**

Task 3:

1. Opened Start Menu using Windows key.
2. Pressed ‘P’ to open Programs submenu.
3. Pressed ‘W’ repeatedly to move selection to Windows Explorer.
4. Pressed Enter to confirm.

- Subject seems to be displaying much more knowledge of menu shortcuts than in previous tasks.

☞ **Referred to manual - upon seeing the expandable list widget in the Folders bar of the Explorer window, checked how to use them.**

5. Used Arrow Keys to select a folder.
6. Used Right Arrow to expand the folder list.
7. Repeated use of Arrow Keys to select folders and expand them.
8. Tab to move focus from Folders bar to Files pane.
9. Used ‘T’ to move selection to the file.
10. Pressed Delete key to delete it.
11. Used Alt + F4 to close the Explorer window.

- Subject used Alt + F4 combination after noticing it in the System menu during previous tasks.

☞ **In Task 3, Subject 1 referred to the manual once.**

Task 4:

1. Used Windows key to open Start Menu.
2. Used Arrow Keys to select and open Control Panel submenu.
3. Used Arrow Keys to select Date/Time.
4. Used Enter to confirm.

- Subject seems to have forgotten use of menu shortcuts.

5. Used Tab key to move focus to Time zone drop down menu.
6. Used Alt + Down Arrow to expand menu.

7. Used Arrow Keys to move selection to correct option.

   - Subject realised that menu shortcuts could not be used, since all options started with a parenthesis. The fact that you cannot type further characters to ‘refine’ the shortcut selection was noted – e.g., typing ‘(Be’ would move the selection to Beijing. This should be made clearer in the manual.

8. Pressed Enter to confirm selection.
9. Pressed Enter to confirm the change and close the dialog.

   - Subject did not move the focus to the OK button as in previous tasks, instead relying on the default action – possibly due to the simpler layout of this dialog.

Questions:

1. Would you ever use keyboard access again?
   - Yes, in some circumstances.

2. In what situations?
   - If you were already using the keyboard, then keyboard shortcuts are often quicker than switching to the mouse.
   - If I couldn’t find a file in Explorer, then I would use the keyboard to move the focus to the file.
   - If the mouse pointer was positioned far from the application menu bar, then I would probably use the keyboard shortcuts.

3. Any other comments on the manual:
   - A glossary of terms used in the manual would be very useful.

Observer's comments:

Subject 2:

Task 1:

1. Opened Start Menu using Windows key.

   - Subject knew how to access the Start Menu, but not how to navigate menus in general.

   ☞ **Referred to manual – how to select a menu option.**

2. Used Up Arrow key to move selection to ‘Find…’.
3. Used Enter to open the submenu.

   - Subject uses the Enter key to open the submenu, as if he were selecting that option, rather than the Right Arrow key, as if to expand the menu.

4. Used Enter to select ‘Files and Folders’.
5. Typed in search string.

- Subject mistyped the search string.

6. Pressed Enter to begin the search.

   - Subject did not attempt to move focus to the ‘Find Now’ button, but also did not visibly check to see if this was the default action.
   - The search was negative since the search string was mistyped. Subject had to move focus back to ‘Named’ text box.
   - ☞ **Referred to manual – how to navigate through widgets in a dialog box.**

7. Used Alt + Tab key combination to attempt to move focus.

   - Alt + Tab switches between applications, not widgets. However, since no other applications were open, the Alt + Tab combination did not have any effect.

8. Used Tab key to move focus to the ‘Named’ text box.

   - Subject did not need to refer to manual, but realised and corrected his own mistake of using Alt + Tab.

9. Used Delete to remove the search string already present.

   - Subject did not appear to know about overtyping selected text in text field.

10. Retyped search string.
11. Pressed Enter to begin search.
12. Used Tab key to move focus to Results list widget.

    - Subject did not hesitate at all or refer to manual when moving focus to the list widget – applying knowledge recently learned about navigating between widgets.
    - Subject did not know how to select an item inside a list widget.
    - ☞ **Referred to manual – how to select items inside a list widget.**

13. Used Down Arrow key to select file.
14. Pressed Enter to open the file.
15. Used Arrow Keys to move caret to end of text.
16. Typed some text.

    - Subject wanted to save the file using the File menu, but did not know how to access application menus.
    - ☞ **Referred to manual – how to open menus**
    - Subject showed confusion over the difference between context and application menus.

17. Used Alt key to select menu bar.
18. Used Down Arrow key to expand the File menu.
19. Used the Arrow keys to move to the ‘Save’ option.
20. Pressed Enter to select ‘Save’.
21. Used Alt key to select menu bar.

22. Used Down Arrow key to expand the File menu.
23. Used Arrow keys to move selection to ‘Exit’.
24. Pressed Enter to select ‘Exit’.

- Subject did not appear to know about using letter shortcuts in menus.
- Next, subject wanted to close the Find Files dialog box.

25. Used Alt key to select menu bar.
26. Used Down Arrow key to expand the File menu.
27. Used Arrow keys to move selection to ‘Close’.
28. Pressed Enter to select ‘Close’.

- Subject showed some confusion about the differing terms ‘Exit’ and ‘Close’.

29. Used MouseKeys to ‘left-click’ on the Desktop to move the focus there.
30. Used MouseKeys to select and open ‘My Computer’.

- Subject used MouseKeys to move the focus to the Desktop since Windows had ‘lost’ it. However, subject then proceeded to use MouseKeys unnecessarily.

31. Used Arrow keys to move selection to C: Drive.
32. Used Enter to open C: Drive.
33. Used Arrow keys and Enter repeatedly to move to the correct folder.
34. Used Arrow keys to move selection to correct file.

- Subject did not appear to know about using the letter keys to move the selection directly to the file group beginning with that letter.

35. Pressed Enter to open file.

- Subject opened the file mistakenly, and so had to close it again.

36. Used Alt to open menu bar.
37. Used Arrow keys to move selection to ‘Exit’.

- Subject noticed the underlining of the shortcut keys in menus, and also the key combination shortcuts.

38. Used Alt + X combination to attempt to close application.

- The use of Alt was unnecessary here, and generated an error beep; subject realised his mistake.

39. Used ‘X’ to select ‘Exit’.

- After the application had closed, the focus in the previously opened Explorer window had been lost. Help was given by saying that this was a common problem, and that the focus had been lost.

40. Used Tab key to relocate focus.

41. Used Alt key to open menu bar.
42. Used Down Arrow to expand File menu.
43. Used Arrows to move selection to ‘Copy to Folder’.
44. Used Enter to select ‘Copy to Folder’.

- Subject had used an unexpected method of copying the file – which involved the use of an expandable list widget.
- ☞ **Referred to manual – expandable list widgets.**

45. Used Arrow keys to move to the first folder.
46. Used ‘+’ key to expand the folder.
47. Used Arrow keys to move selection and ‘+’ key to expand folders repeatedly to the correct location.
48. Pressed Enter to confirm.

- Subject did not check the default action of the dialog.
- Subject noted that the file appeared at the end of the list.
- Subject noted that the focus had been lost again in the Explorer window.

49. Used Tab key to relocate focus.

- Quick application of previous procedures.
- Next, subject wanted to rename the file.

50. Used Alt key to select menu bar.
51. Used Arrow keys to move selection to ‘Edit’ menu.
52. Used Down Arrow to expand the menu.

- The rename command is not in the ‘File’ menu. This was lack of knowledge about the Windows interface.
- Subject wanted to use the context menu of the file.
- ☞ **Referred to manual – how to open context menus.**

53. Used Context menu key to open context menu.
54. Used Arrow keys to move selection to 'Rename'.
55. Pressed Enter to select 'Rename'.
56. Typed the new name.

- Subject did not attempt to delete the text before typing the new name, possibly indicating he has remembered reading about overtyping text.

57. Pressed Enter to confirm name change.
58. Used Alt to select menu bar.
59. Used Down Arrow key to expand File menu.
60. Pressed 'C' to select 'Close'.

Task 2:

1. Pressed Windows key to open Start Menu.
2. Used Arrow keys to move selection to Settings.

3. Pressed Enter to open Settings submenu.
4. Pressed Enter to select Control Panel.
5. Used Arrow keys to move selection to 'Display'.
6. Pressed Enter to open Display Properties applet.
7. Used Tab to move focus to the Tab panes widget.

- Subject did not appear to know that you could use Shift + Tab in a dialog to move directly to the Tab panes widget, as is suggested in the manual.

8. Used Arrow keys to move selection to the 'Settings' pane.
9. Used Tab to move focus to the 'Colors' drop down menu widget.

☞ **Referred to manual – how to expand drop down menu widgets.**

10. Used Alt + Down to expand menu.
11. Used Arrow keys to move selection.
12. Used Enter to confirm.
13. Used Tab key to move focus to the 'Screen Area' slider bar widget.
14. Used Arrow keys to change the screen area.
15. Used Tab key to move focus to 'OK' button widget.
16. Pressed Enter to select the 'OK' button.

Task 3:

1. Pressed Windows key to open Start Menu.
2. Used Arrow keys to move selection to Programs.
3. Pressed Enter to open Programs submenu.
4. Pressed 'W' to move selection to Windows Explorer.

- There seems to be a discrepancy in the way the different menu options are accessed – sometimes directly, sometimes by moving the selection using arrow keys.

5. Pressed Enter to confirm selection.
6. Used Arrow keys to move selection in Folders pane of Explorer window.
7. Used '+' key to expand folders.
8. Used Arrow keys and '+' key repeatedly to select and expand folders.
9. Pressed Tab key to move focus to the files pane of the Explorer window.
10. Pressed 'T' to move selection to 'testfile.doc'.
11. Pressed Alt key to select menu bar.
12. Pressed Down Arrow key to expand File menu.
13. Pressed 'D' to select 'Delete'.

- Subject did not appear to know about pressing Delete key to delete files.

14. Pressed Enter to confirm deletion in pop-up dialog.
15. Pressed Alt key to select menu bar.
16. Pressed Down Arrow key to expand File menu.
17. Pressed 'C' to select 'Close'.

Task 4:

1. Pressed Windows key to open Start Menu.
2. Used Arrow keys to move selection to Settings.
3. Pressed Enter to open Settings submenu.
4. Pressed Enter to select Control Panel.
5. Used Arrow keys to move selection to 'Date/Time'.
6. Pressed Enter to open Date/Time Properties applet.
7. Used Tab key to move focus to 'Time zone' drop down menu widget

- Subject could not recall how to expand drop down menu widgets.
- ☞ **Referred to manual – how to expand drop down menu widgets.**

8. Used Arrow keys to change selection.

- Subject realised that menu shortcuts could not be used, since all options started with a parenthesis.

9. Pressed Enter to confirm selection and close the dialog.

Questions:

1. Would you ever use keyboard access again?

Observer's comments: Subject applied knowledge gained from performing one task to the following tasks much more quickly than the other participants.

Subject 3:

Task 1:

1. Used Windows key to open Start Menu.
2. Used Arrow keys to move selection to Find.

- Subject did not use shortcut letters to select Find submenu.

3. Pressed Enter to open Find submenu.
4. Pressed Enter to select Files or Folders.
5. Typed search string.

- Checked where cursor was located before typing search string.

6. Pressed enter to begin search.

- Subject assumed that beginning the search would be the default action in the dialog.
- ☞ **Referred to manual – how to move focus between widgets.**

7. Used Tab key to move focus to Results list widget.

   ☞ Subject displayed some confusion between Alt + Tab to switch between applications, and Tab, to switch between widgets.

8. Pressed Space bar to select the file in the list widget.

   - This is not a method mentioned in the manual, but is nevertheless valid, and moves the selection to the first item in the list.

   ☞ **Referred to manual – how to open files**

9. Pressed Enter to open file.
10. Used Arrow keys to move caret to end of text in the file.
11. Added some text.

    ☞ **Referred to manual – how to access the application menu bar.**

12. Pressed Alt to select menu bar.
13. Pressed 'F' to expand File menu.
14. Used Arrow keys to move selection to Save.
15. Pressed Enter to select Save.
16. Pressed Alt to select menu bar.
17. Pressed 'F' to expand File menu.
18. Pressed 'X' to select Exit.
19. Pressed Alt to open application menu bar in Find Files dialog.
20. Pressed 'F' to open File menu.
21. Pressed 'C' to select 'Close'.

    - After referring to the manual to see how to select the menu bar, subject showed no difficulties in using menu shortcuts to open menus and select options.
    - Next, subject displayed some difficulties in locating the focus on the Desktop – didn't see how the focus could be 'nowhere'.

22. Used Tab key to move focus to icons on the Desktop.
23. Pressed Enter to open 'My Computer'.
24. Used Arrow keys to move selection to C: Drive.
25. Pressed Enter to open C: Drive.
26. Used Arrow keys and Enter repeatedly to move to location of file.
27. Pressed 'T' to move selection to testfile.doc.

    ☞ **Referred to manual – how to copy and paste files.**

    - Subject decided to use MouseKeys to select copy on the toolbar instead of using a keyboard shortcut.

28. Used MouseKeys to select Copy on the toolbar.
29. Used MouseKeys to select Paste on the toolbar.
30. Used Arrow keys to select the new file.

    ☞ **Referred to manual – how to open the context menu.**

31. Pressed Context key to open context menu.
32. Pressed 'M' to select 'Rename'.
33. Typed new name.

- Subject overtyped the text – said he knew that it would overwrite it as it was highlighted.

34. Pressed Alt to select menu bar in Explorer.
35. Pressed F to open File menu.
36. Pressed C to select Close.

Task 2:

1. Pressed Windows key to open Start Menu.
2. Pressed 'S' to select Settings.
3. Pressed Enter to open Control Panel.
4. Used Arrow keys to select Display.
5. Pressed Enter to open Display Properties applet.
6. Tried Arrow keys to move selection of Tab panes widget.

- Arrow keys had no effect as no widget which would accept input in this way was active.
- Subject remembered how to move focus between widgets without referring to the manual.

7. Pressed Tab key to move focus to Tab panes widget.
8. Used Arrow keys to move selection to 'Settings' pane.

- Subject now had problems in transferring his knowledge – only just used – of how to navigate between widgets, to the next task.

9. Pressed Tab key to move focus to Colors drop down menu widget.

- **Referred to manual – how to expand drop down menu widgets.**

10. Used Alt + Down Arrow key to expand Colors drop down menu widget.
11. Used Arrow keys to change selection.
12. Pressed Escape to close.

- The use of Escape to close the menu is not strictly necessary here, as moving the focus on to the next widget to close it automatically. However, since the tasks are separated, pressing Escape provides a sense of task closure.

13. Pressed Tab key to move focus to Screen area slider bar widget.
14. Used Arrow keys to change selection.
15. Pressed Enter to confirm changes and close the dialog.

- Subject knew that the default action was to press OK, as he had noted the extra highlighting around the OK button.

Task 3:

1. Pressed Windows key to open the Start Menu.

2. Pressed P to open Programs submenu.
3. Pressed W to move selection to Windows Explorer option.

- Here subject tried out the principle of cycling through all the options beginning with 'W' by pressing the key repeatedly.

4. Used Arrow keys to move selection to the correct folder.

☞ **Referred to manual – how to expand expandable lists.**

5. Used Right Arrow key to expand folder.
6. Used Arrow keys to select folders and Right Arrow key to expand folders repeatedly to the correct location.
7. Pressed Tab key to move focus to the Files pane.
8. Pressed 'T' to move selection to the testfile.doc file.
9. Pressed Delete key to delete the file.

- Subject said that he had read how to delete files in the manual previously.

10. Pressed Enter to confirm deletion.

Task 4:

1. Pressed Windows key to open Start Menu.
2. Pressed 'S' to select Settings.
3. Pressed Enter to open Control Panel.
4. Used Arrow keys to select Date/Time.
5. Pressed Enter to open Date/Time Properties applet.

- Subject expressed some uncertainty about using the letter shortcuts to move to menu or list options when the shortcut is not explicitly defined.

6. Used Tab key to move focus to Time zone drop down menu widget.

☞ **Referred to manual – how to expand drop down menu widgets.**

7. Used Alt + Down to expand the drop down menu widget.
8. Tried B to select Beijing in the menu.

☞ Subject attempted to use B to move directly to the Beijing option, but was unsuccessful due to parentheses at start and end of menu options.

9. Used Arrow keys to select correct option.
10. Pressed Enter to confirm and close dialog.

Questions:

Subject 4:

This participant is a 17 year old with cerebral palsy, who has been using a computer for several years in school and college. He routinely uses the keyboard to access computers, and at home uses a trackball rather than a mouse, which he finds easier to use. Due to his

Task 1:

1. Opened the Start Menu using the Windows key.
2. Up Arrow key to move selection to 'Find'.
3. Right Arrow key to expand submenu.
4. Enter to select 'Files or folders'.
5. Entered search string.

    - Subject knew the right text box was already highlighted.

6. Pressed Enter to select the default action (Find Now).

    - Subject knew that Find Now was the default action in this dialog.

7. Used Tab key to move focus to the Results list widget.
8. Used Down Arrow key to select file.
9. Used Enter to open the file.
10. Used Arrow keys to move the caret to the end of the text.
11. Typed some text.
12. Pressed Alt + F key combination to open the File menu.

    - Subject knew that Alt + F would open the menu directly, and did not try to access the file in stages – i.e. select the menu bar, then open the File menu.

13. Pressed 'S' to select 'Save'.
14. Pressed Alt + F to open the File menu.
15. Pressed X to exit the application.
16. Used Tab key to relocate focus in the Explorer window.
17. Used Arrow keys to select the file.
18. Pressed Ctrl + C to copy the file to the clipboard.
19. Pressed Ctrl + V to paste the file.
20. Used Arrow keys to move selection to the new file at the end of the list.
21. Pressed Context key to open file's context menu.
22. Used Arrow keys to move selection to Rename.
23. Pressed Enter to select Rename.
24. Overtyped the new name.
25. Pressed Enter to confirm.

Task 2:

1. Used Windows key to open Start Menu.
2. Used Arrow keys to move selection to Settings.
3. Pressed Right Arrow key to expand Settings submenu.
4. Pressed Enter to select Control Panel.

- Subject did not appear to know about using shortcut letters in the Start Menu to select options.

5. Used Arrow Keys to move selection to Display.
6. Pressed Enter to open Display Properties applet.
7. Used Tab key to move focus to the Tab pane widget.

- Subject did not appear to know that you could move the focus 'back' to the Tab panes widget using Shift + Tab.

8. Used Arrow keys to move selection to 'Settings' pane.
9. Used Tab key to move focus to 'Colors' drop down menu widget.
10. Used Arrow keys to change selection.

- Subject did not attempt to expand the menu. This was due to familiarity with the Windows interface; however, he also said that he would not have known how to expand the menu.

11. Used Tab key to move focus to the 'Screen Area' slider bar widget.
12. Used Arrow keys to change selection.
13. Pressed Enter to confirm changes and close the dialog.

- Subject did not attempt to move focus onto the OK button before pressing Enter. When asked, he said that he knew just pressing Enter would suffice.

Task 3:

1. Used Windows key to open Start Menu.
2. Pressed 'P' to open Programs submenu.
3. Used Arrow Keys to select Windows Explorer option.
4. Pressed Enter to confirm selection.

- Subject used letter shortcuts on this occasion after not having done so before; but only for 'marked' shortcuts, and not for moving the selection t, for example, the 'W' group of options by pressing the 'W' key.

5. Used Arrow keys to move folder selection in the Folders pane of the Explorer window.
6. Used Right Arrow key to expand folders.
7. Used Arrow keys and Right Arrow key repeatedly to navigate to correct location.

- Subject continues the concept of using the Right Arrow key to expand items in Windows.

8. Used Tab key to move focus to the Files pane in the Explorer window.
9. Used Arrow keys to move selection to the file.

- Subject shows further evidence of being unaware that you can use letter shortcuts to move selections even if those shortcuts are not explicitly defined.

10. Pressed Delete key to delete the file.

- Subject used a direct shortcut action.

11. Pressed Enter to confirm deletion.
12. Used Alt + F to select and expand File menu.

    - Subject displays more familiarity with Windows here by going directly to the menu required.

13. Pressed 'C' to select Close.

Task 4:

1. Used Windows key to open Start Menu.
2. Used Arrow keys to move to Settings.
3. Used Right Arrow key to expand Settings submenu.

    - Subject did not use letter shortcuts to select the Settings submenu as he previously had with the Programs submenu.

4. Pressed Enter to select Control Panel.
5. Used Arrow keys to move selection to Date/Time.
6. Pressed Enter to open Date/Time Properties applet.
7. Used Tab key to move focus to Time zone drop down menu widget.
8. Used Arrow keys to select correct option.

    - Subject did not attempt to expand drop down menu, despite this being a large number of options to scroll through.

9. Pressed Enter to confirm the changes and close the dialog.
10. Used Alt + F to open File menu in Control Panel.
11. Pressed 'C' to select Close.

Questions:

1. Do you think that using the manual would change the way you access the computer via the keyboard?
    - Probably not, since I've already figured out how to do it myself, and so it would be like unlearning bad habits in a lot of cases. And I'm already quite happy with how I manage to use the computer anyway.

2. Any other comments on the manual:
    - There seems to be a lot to take in. It might be easier if the manual was presented more as a taught course, rather than reading it yourself.

Observer's comments:

He tended to use the keyboard short cuts that he has always done in the past. If this had been available when he first started using a computer, no doubt he would be more proficient in using the short cuts. He also tended to use commands that could be done with one key press at a time. This is more to do with his physical disability. As pointed out at the beginning, Sticky Keys were turned on before these tests were conducted. However, even though the option of using Sticky Keys was there, he preferred to use more simple short cuts.

Appendix C – Manual (Version 1)

This appendix contains a copy of the manual that was made available to the participants of the testing of the manual.

Appendix D – Manual (Version 2)

This appendix contains the version of the manual as it was updated following the analysis of the testing results.

Appendix E – Draft manual

Functions in Windows Explorer

General keyboard functions:

- ENTER: Activates the selected item with its default function.
- DELETE: Move item to Recycle Bin
- SHIFT+DELETE: Deletes an item immediately without placing it in the Recycle Bin
- SHIFT+F10 Opens a shortcut menu for the selected item (this is the same as right-clicking an object)
- SHIFT: Press and hold down the SHIFT key while you insert a CD-ROM to bypass the automatic-run feature
- BACKSPACE: Switch to the parent folder; or opens the parent folder in the current window if you have selected to display all folders in the same window.
- SHIFT+ALT+F4: For folders, close the current folder plus all parent folders (has no extra effect if you have chosen to display all folders in the same window).
- Left ALT+left SHIFT+NUM LOCK: Toggles MouseKeys on and off
- Application key: (▤) Displays a shortcut menu for the selected item (equivalent to right-clicking)

Function Keys:

- F1: Starts Windows Help
- F2: Rename object
- F3: Find all files
- F4: Selects the Address box (if the toolbar is active in Windows Explorer)
- F5: Refreshes the current window.
- F6: Moves among panes in Windows Explorer
- F10: Activates menu bar options (same as ALT)
- F11: Displays the window in Full Screen Mode

ALT Modifiers:

- ALT: Activates menu bar options
- ALT+TAB: Switch to another running program (hold down the ALT key and then press the TAB key to view the task-switching window)
- ALT+ENTER: Open the properties for the selected object
- ALT+SPACE: Displays the window's System menu (from the System menu, you can restore, move, resize, minimize, maximize, or close the window)
- ALT+DOWN ARROW: Opens a drop-down list box
- ALT+F4: Closes the current window
- ALT+F6: Switch between multiple windows in the same program (for example, when the Notepad Find dialog box is displayed, ALT+F6 switches between the Find dialog box and the main Notepad window)
- ALT+underlined letter in menu: Opens the menu

CTRL Modifiers:

- CTRL+ESC: Opens the Start menu (use the ARROW keys to select an item)
- CTRL+C: Copy selected object
- CTRL+X: Cut selected object
- CTRL+V: Paste copied or cut object (if a folder is selected when Paste is applied, the cut/copied object is pasted into that folder).
- CTRL+Z: Undo last command
- CTRL+A: Select all the items in the current window
- CTRL+F: Find all files

Windows Key (⊞) Modifiers:

- ⊞: Start menu
- ⊞+R: Run dialog box
- ⊞+M: Minimize all
- SHIFT+⊞+M: Undo minimize all
- ⊞+F1: Help
- ⊞+E: Windows Explorer
- ⊞+F: Find files or folders
- ⊞+D: Minimizes all open windows and displays the desktop
- CTRL+⊞+F: Find computer
- ⊞+TAB: Cycle through taskbar buttons
- ⊞+Break: System Properties dialog box

Tabs Control

NB: Tabs refers to the different pages available in a dialog, e.g. the different pages in the Internet Options applet in Control Panel: General, Security, Privacy, etc.

In the dialog, you can use TAB and SHIFT+TAB to move between controls as you would any other; to access the other ‘pages’ in the dialog, move the focus to the label of the current page, then use the arrow keys to navigate through the pages. Press TAB once to enter the page once you have selected it.

Tree Control

- Numeric Keypad *: Expands everything under the current selection
- Numeric Keypad +: Expands the current selection
- Numeric Keypad -: Collapses the current selection.
- RIGHT ARROW: Expands the current selection if it is not expanded, otherwise goes to the first child
- LEFT ARROW: Collapses the current selection if it is expanded, otherwise goes to the parent

Dialog Box Keyboard Commands

- TAB: Move to the next control in the dialog box
- SHIFT+TAB: Move to the previous control in the dialog box
- SPACEBAR: If the current control is a button, this clicks the button. If the current control is a check box, this toggles the check box. If the current control is an option, this selects the option.
- ENTER: Equivalent to clicking the selected button (the button with the outline)
- ESC: Equivalent to clicking the Cancel button
- ALT+underlined letter in dialog box item: Move to the corresponding item – if the corresponding item is a button, this will select that button.

Start Menu Control

When the Start Menu is active (by pressing CTRL+ESC or ⊞), you can navigate the menus by pressing the initial letter of the item you wish to select. If there are multiple items beginning with that letter, you can cycle through them by tapping the letter repeatedly. If the item is runnable, i.e. a link to a program or document, and there is only one item with that initial in the current sub-menu, pressing the letter will run that item. You can use arrow keys to navigate also; left and right will expand or collapse a sub-menu.

Functions in Internet Explorer and Outlook Express

Many of the functions outlined for Windows Explorer are valid in other Microsoft products, including Internet Explorer and Outlook Express. Naturally, both these programs have additional commands, and there are some differences; however, there is a common theme. This is true of many Windows programs. Since Internet Explorer and Outlook Express are both included as a standard part of the Windows package, this section is to help you use these programs without a mouse.

Internet Explorer:

There are many versions of IE, but most share common commands. There are some features which are not available in earlier versions; Print Preview was only added after IE5.5. The current minimum ‘accepted’ standard is IE4, which is packaged as part of Windows 98. Windows 95 was supplied with earlier versions, however.

Frames hint: frames are used to make the layout of a page more attractive; however, for mouseless access, they can be frustrating to navigate. Frames in IE are laid out first top -> bottom, then left -> right. You can navigate between them using (SHIFT +) CTRL + TAB.

General commands:

- F1: Display Internet Explorer Help, or when in a dialog box, display context Help on an item.
- F3: Open the Search bar.
- F4: Display a list of addresses you've typed.

- F5: Refresh the current Web page.
- F6: Move forward between frames.
- F11: Toggle between full-screen and regular views of the browser window.
- TAB: Move forward through the items on a Web page, the Address bar, and the Links bar.
- BACKSPACE: Go to the previous page.
- UP ARROW: Scroll toward the beginning of a document.
- DOWN ARROW: Scroll toward the end of a document.
- PAGE UP: Scroll toward the beginning of a document in larger increments.
- PAGE DOWN: Scroll toward the end of a document in larger increments.
- HOME: Move to the beginning of a document.
- END: Move to the end of a document.
- ESC: Stop downloading a page.
- ENTER: Activate a selected link.
- CTRL+TAB: Move forward between frames.
- SHIFT+CTRL+TAB: Move back between frames.
- CTRL+D: Add the current page to your favorites.
- CTRL+F: Find on this page.
- CTRL+R: Refresh the current Web page.
- CTRL+F5: Force a refresh of the current Web page, even if your settings specify a different refresh period.
- CTRL+L: Go to a new location.
- CTRL+O: Go to a new location.
- CTRL+N: Open a new window.
- CTRL+W: Close the current window.
- CTRL+S: Save the current page.
- CTRL+P: Brings up the Print dialog.
- CTRL+E: Open the Search bar.
- CTRL+I: Open the Favorites bar.
- CTRL+H: Open the History bar.
- CTRL+X: Remove the selected items and copy them to the Clipboard.
- CTRL+C: Copy the selected items to the Clipboard.
- CTRL+V: Insert the contents of the Clipboard at the selected location.
- CTRL+A: Select all items on the current Web page.

Address Bar:

- ALT+D: Select the text in the Address bar.
- CTRL+LEFT ARROW: When in the Address bar, move the cursor left to the next logical break in the address (period or slash).
- CTRL+RIGHT ARROW: When in the Address bar, move the cursor right to the next logical break in the address (period or slash).
- CTRL+ENTER: Add "www." to the beginning and ".com" to the end of the text typed in the Address bar, and goes to that page.
- UP ARROW: Move forward through the list of AutoComplete matches.
- DOWN ARROW: Move back through the list of AutoComplete matches.

Favourites Bar:

- CTRL+B: Open the Organize Favorites dialog box.
- ALT+UP ARROW: Move selected item up in the Favorites list in the Organize Favorites dialog box.
- ALT+DOWN ARROW: Move selected item down in the Favorites list in the Organize Favorites dialog box.

Print Preview:

- ALT+F then V: File menu command to bring up Print Preview window.
- ALT+U: Change paper, headers and footers, orientation, and margins for this page.
- ALT+HOME: Display the first page to be printed.
- ALT+LEFT ARROW: Display the previous page to be printed.
- ALT+A: Type the number of the page you want displayed.
- ALT+RIGHT ARROW: Display the next page to be printed.
- ALT+END: Display the last page to be printed.
- ALT+MINUS: Zoom out.
- ALT+PLUS: Zoom in.
- ALT+Z: Display a list of zoom percentages.
- ALT+F: Specify how you want frames to print. This option is available only if you are printing a Web page that uses frames.
- ALT+C: Close Print Preview.

Outlook Express

Outlook Express shares many common keyboard commands with IE, and Windows. It does have some special commands, and some commands are only available at certain times. All are listed here.

Main window, view message window, and send message window

- F1: Open Help topics
- CTRL+A: Select all messages

Main window and view message window

- CTRL+P: Print the selected message
- CTRL+M: Send and receive e-mail
- DEL or CTRL+D: Delete an e-mail message
- CTRL+N: Open or post a new message
- CTRL+SHIFT+B: Open the Address Book
- CTRL+R: Reply to the message author
- CTRL+F: Forward a message
- CTRL+SHIFT+R or CTRL+G (news only): Reply to all
- CTRL+I: Go to your Inbox
- CTRL+> or CTRL+SHIFT+>: Go to the next message in the list
- CTRL+< or CTRL+SHIFT+<:
- Go to the previous message in the list

- ALT+ENTER: View properties of a selected message
- F5: Refresh news messages and headers
- CTRL+U: Go to the next unread e-mail message
- CTRL+SHIFT+U: Go to the next unread news conversation
- CTRL+Y: Go to a folder

Main window

- CTRL+O or ENTER: Open a selected message
- CTRL+ENTER or CTRL+Q: Mark a message as read
- TAB: Move between the Folders list (if on), message list, preview pane, and Contacts list (if on).
- CTRL+SHIFT+A: Mark all news messages as read
- CTRL+W: Go to a newsgroup
- LEFT ARROW or PLUS SIGN (+): Expand a news conversation (show all responses)
- RIGHT ARROW or MINUS SIGN (-): Collapse a news conversation (hide messages)
- CTRL+J: Go to the next unread newsgroup or folder
- CTRL+SHIFT+M: Download news for offline reading

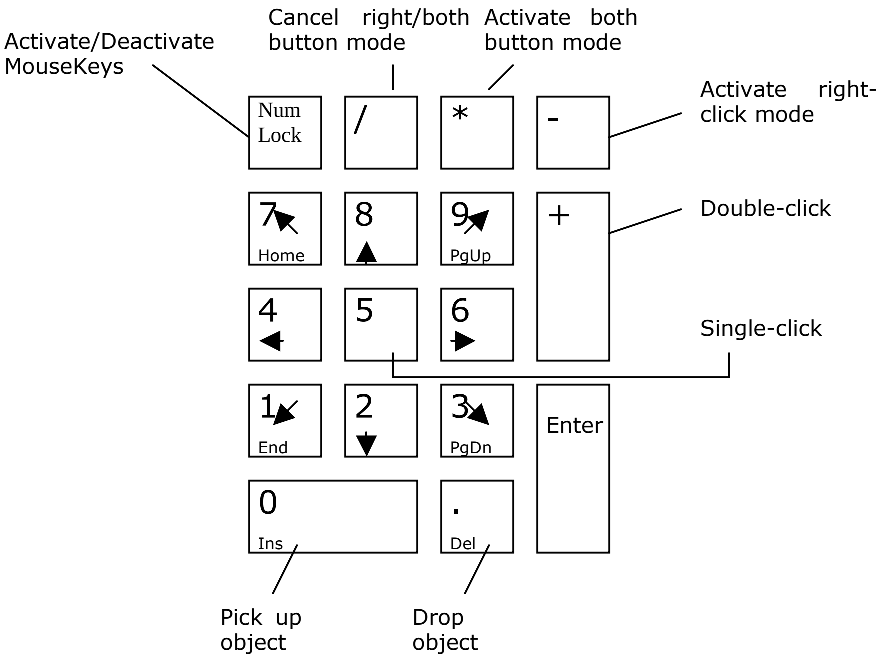
Message window – viewing or sending

- ESC: Close a message
- F3: Find text
- CTRL+SHIFT+F: Find a message
- CTRL+TAB: Switch among the Edit, Source, and Preview tabs

Message window – sending only

- CTRL+K or ALT+K: Check names
- F7: Check spelling
- CTRL+SHIFT+S: Insert a signature
- CTRL+ENTER or ALT+S: Send (post) a message

MouseKeys


Activate/Deactivate MouseKeys
Cancel right/both button mode
Activate both button mode
Activate right-click mode
Double-click
Single-click
Num Lock
/
*
-
7 Home
8
9 PgUp
+
4
5
6
1 End
2
3 PgDn
Enter
0 Ins
. Del
Pick up object
Drop object

Moving:

Use the nine number keys 1-9 to move the cursor vertically, horizontally, or along either diagonal.

Clicking:

- Left-click: Num 5
- Double left-click: Num +

- Right-click: Press Num – to access the right mouse button, then;
  - Num 5 to click once
  - Num + to double-click.

- Both mouse buttons: Press Num *, then;
  - Num 5 to click once
  - Num + to double-click

NB: After using right-click or both mouse buttons mode, press Num / to return to normal (left-button) mode.

- To pick up an object: Press Num INS/0 once.
- To drop an object: Press Num DEL/. once.
- Move the object around using the arrow keys on the Numeric keypad.

Acknowledgements

- Alistair Edwards for supervising this project.
- Glenn Denney for proof reading, and general support.
- Dan Bentley, Tim Essex, Rebecca Williamson, Neil Williamson and Jamal Akbar for participating in the testing of the manual.
- Finally, the BCS publication *Ability*, for providing the inspiration for this project.