Virtual Keyboard Guidelines for Single-Switch Access

By Katherine Jamieson

NOTES:

This is one of the most excellent projects completed in CMPT 480. It is an example of a project that received 100% in this course.

The design in this project has been submitted to ISO/IEC JTC1/SC35/WG1 to form the basis for an international standard (ISO/IEC 22121-3) which is currently under development.

Table of Contents

[Analysis 4](#_Toc19876500)

[1. Introduction 4](#_Toc19876501)

[1.1 What is single switch input? 4](#_Toc19876502)

[1.2 How does it work? 4](#_Toc19876503)

[1.3 Why should a virtual keyboard support single-switch access? 4](#_Toc19876504)

[1.4 Challenges to Keep in Mind 4](#_Toc19876505)

[2. Definitions 5](#_Toc19876506)

[3. Overview of Existing On-Screen Keyboards 7](#_Toc19876507)

[3.1 Windows 10 7](#_Toc19876508)

[3.2 Click-N-Type [7] 7](#_Toc19876509)

[3.3 WiViK [8] 8](#_Toc19876510)

[3.4 DiscoverPro [9] 8](#_Toc19876511)

[4. Scanning 9](#_Toc19876512)

[4.1 Scanning Method 10](#_Toc19876513)

[4.2 Scanning Speed 11](#_Toc19876514)

[4.3 Scan Start 12](#_Toc19876515)

[4.4 Re-scan and skip scan 13](#_Toc19876516)

[4.5 Scanning Indicators 13](#_Toc19876517)

[4.6 Selection Indicators 14](#_Toc19876518)

[5. Keyboard Arrangement and Customization 14](#_Toc19876519)

[5.1 Keys 14](#_Toc19876520)

[5.2 Keyboard Layout 15](#_Toc19876521)

[6. Modifier Keys, Navigation Keys, and Macros 17](#_Toc19876522)

[6.1 Modifier Keys 17](#_Toc19876523)

[6.2 Navigation Keys 17](#_Toc19876524)

[6.3 Macros 17](#_Toc19876525)

[7. Prediction 18](#_Toc19876526)

[7.1 Character Prediction 18](#_Toc19876527)

[7.2 Word Prediction 19](#_Toc19876528)

[8. Accessible Help 20](#_Toc19876529)

[8.1 Accessible Help 20](#_Toc19876530)

[Design 21](#_Toc19876531)

[1. Scope 21](#_Toc19876532)

[2. Definitions 21](#_Toc19876533)

[3. Scanning 22](#_Toc19876534)

[3.1 Scanning Methods 22](#_Toc19876535)

[3.2 Scanning Speed 24](#_Toc19876536)

[3.3 Starting Scan 24](#_Toc19876537)

[3.4 Re-Scan and Skip Scan 25](#_Toc19876538)

[3.5 Scanning Indicators 26](#_Toc19876539)

[3.6 Selection Indicators 27](#_Toc19876540)

[4. Keyboard Arrangement and Customization 27](#_Toc19876541)

[4.1 Keys 27](#_Toc19876542)

[4.2 Keyboard Layout 28](#_Toc19876543)

[5. Modifier Keys, Navigation Keys, and Macros 30](#_Toc19876544)

[5.1 Modifier Keys 30](#_Toc19876545)

[5.2 Navigation Keys 31](#_Toc19876546)

[5.3 Macros 31](#_Toc19876547)

[6. Prediction 32](#_Toc19876548)

[7. Accessible Help 34](#_Toc19876549)

[7.1 Accessible Help 34](#_Toc19876550)

[Evaluation 34](#_Toc19876551)

[Part A Instructor Evaluation 34](#_Toc19876552)

[Method of Discussion 34](#_Toc19876553)

[Discussion of Instructor Evaluation 35](#_Toc19876554)

[Part B SelfEvaluation 48](#_Toc19876555)

[Method of Discussion 48](#_Toc19876556)

[Discussion of Self-Evaluation 48](#_Toc19876557)

[Part C Proof by Examples 58](#_Toc19876558)

[Discussion of Examples 58](#_Toc19876559)

[Bibliography 60](#_Toc19876560)

# Analysis

## 1. Introduction

### 1.1 What is single switch input?

There are many people with severe physical impairments that rely on computers and other assistive technology to allow them to communicate and interact with the world. As Koester and Simpson write, "Single-switch scanning allows people with severe physical impairments, who may also be unable to speak, to independently use a computer or augmentative and alternative communication (AAC) device" [1].

The input method of single-switch is a switch device. This is an umbrella term for different kinds of devices that, in their simplest form, have two states, on and off. There are many devices that can be used for single switch access. These take the form of buttons, levers, pedals, touch pads, sip and puff, wobble devices, and more. These devices hook up to a computer, usually through a switch controller, and can then be used with switch compatible software.

### 1.2 How does it work?

There are many ways that systems can be implemented to work with these devices. There are various scanning methods, mouse emulation, and in some cases Morse code could even be used. However, many of these are beyond the physical and/or cognitive capabilities of the users of these devices.

The easiest form of access for these devices is automatic scanning. Automatic scanning is the "successive highlighting of items that are laid out spatially on-screen, and the selection of a desired item by activating a switch when the target item is highlighted" [2].

### 1.3 Why should a virtual keyboard support single-switch access?

Everything that can be done with a mouse on a computer should be able to be done with a keyboard. This means that if single-switch users have a keyboard that is accessible to them they, theoretically, should be able to perform any action necessary on a computer. A virtual keyboard with single-switch accessibility features allows single-switch users to perform necessary tasks on a system.

Another reason virtual keyboards should support single-switch access is because a "virtual keyboard … could be readily changed according to both the users and its use" [3]. Virtual keyboards are not constrained by physical space or the physical world. They can have as many buttons as necessary, they can be resized, the keys can be changed and rearranged, extra features can be present on the keyboard, and more. This flexibility and lack of restrictions makes virtual keyboards great for the level of customization that can be done for single-switch access.

### 1.4 Challenges to Keep in Mind

In Jones' paper he mentions that, "There were changes in a single user throughout the day, for example through boredom or fatigue. The virtual keyboard would need to be able to be readily adaptable to these difference" [3]. This highlights a few challenges that a virtual keyboard needs to be able to accommodate. Long scan times can make progress on a task slow going, and the user might become too bored to want to continue the task.

In regards to fatigue, there is a lot a user must do to use single-switch input. The user “must monitor both the scan process and the application where his output is directed. When in word-prediction mode the cognitive load is even higher, since the user has to additionally monitor the progress of the word-prediction process and at the same time decide whether to perform a selection out of the looked-up words and prefixes, or to proceed with the selection of another character" [2]. This "requires focusing attention on several things at once: locating the desired item, rejecting unwanted items, focusing on the partner or the cursor to indicate the desired item before it is passed over" [4]. As the user must focus on so many things, the cognitive load is very high. This in combination with the physical exertion required to repeatedly press a button will lead to fatigue. The virtual keyboard must do as much as it can to prevent fatigue, and to accommodate a user who is starting to get fatigued.

As noted by Treviranus and Tannok note, "the process of entering selections needed to be quick, simple, and as direct as possible to allow [the user] to concentrate on the actual task" [5]. The focus required to use the virtual keyboard should not be so large that the user is unable to also focus on their task.

Lastly, Steriadis and Constantinou provide a couple of guidelines from their work on designing human-computer interfaces for quadriplegic people that have been kept in mind whilst creating this document. "(a) the provided services that are not so useful must not get in the way; (b) unknown existing functionality must be accessible whenever needed; and (c) commonly used functionality should be easy to learn, use and remember" [2].

## 2. Definitions

**2.1**

**Alphanumeric key**

"keys used in normal text typing including letters, numbers and some punctuation keys" [6]

**2.2**

**Modifier key**

"keys which modify the meaning of other keys pressed down at the same time" [6]

**2.3**

**Navigation key**

“keys used to move within and between user interface objects. The arrow keys, tab, space, enter, home, end, page up and page down are navigation keys” [6]

**2.4**

**Scan Cursor**

the current row, group, column, or individual item currently able to be selected

**2.5**

**Scanning**

successively moving scan cursor through items

**2.6**

**Scan Level**

the depth the scan is currently narrowed to

**2.7**

**Scan cycle**

the completion of the cursor visiting every item currently available to be selected

**2.8**

**Scanning speed**

length of time the cursor stays on an item before going to the next one

**2.9**

**Automatic Scanning**

scanning that does not require user to control progress through items

**2.10**

**Scanning Method**

the pattern the scan cursor takes while scanning items

**2.11**

**row/column**

cursor scans through each row, and moves to scan each column when switch is triggered

**2.12**

**row/group/column**

cursor scans through each row, then through groups in selected row, and then columns within those groups

NOTE Group and row can be interchanged depending on the method. This scanning method would be referred to as group/row/column.

**2.13**

**Word prediction**

generation of a set of words, of which one might be used next

**2.14**

**Character prediction**

generation of a set of individual characters, of which one might be used next

**2.15**

**Macros**

an instruction that expands automatically into a string of characters and keystrokes

## 3. Overview of Existing On-Screen Keyboards

To gain a better understanding of the features virtual keyboards already have for single-switch access four existing on-screen keyboards were researched. Focus was placed on how they supported single-switch access, what settings for this they supplied, and what are some features that were supplied by some keyboards, but not others.

Two of these virtual keyboards have been specifically directed at single-switch users, while the other two would be considered more general purpose keyboards.

### 3.1 Windows 10

The first keyboard looked at was the on-screen keyboard that comes packed with Windows 10 accessibility features. This keyboard is not specifically designed for single-switch access; however, it does supply some features to allow switch users access.

Some features that are available with this keyboard:

* This keyboard is available right from the log in screen, and it can be set to run automatically on system startup.
* Single-switch scanning can be turned on from the settings menu of the keyboard. These settings will persist beyond the current session once set.
	+ These settings include:
		- Setting of scanning speed from 0.5 seconds to 3 seconds.
		- Device to use for selection (keyboard key, joystick/physical switch, click).
* Keyboard has predictive text. These words are at the top of the keyboard (first thing scanned), and there are up to 8 words suggested at once.
* Provides audio and visual cues to indicate the progression of the scan and selection of a key.
* Gives user three scans through a group/row/column before returning to the higher level or stopping scanning. This is to allow users a chance to select the item they need if they miss it on the first scan.

A few of the features that are not included in this keyboard:

* Does not easily supply a way to change the layout of the keyboard keys. QUERTY is the default.
* There is only one scanning method supplied. This keyboard uses the group/row/column method of scanning, meaning any key on the keyboard will take, at minimum, 3 switch presses to be input.
* There is no option bypass the three scans to go back a level. Users that accidentally choose an incorrect selection will be required to wait the full three scans before continuing their task.
* There is no option to store macros in buttons on the keyboard for later.

Overall this keyboard does supply switch access, but lacks many customizations that would provide further accessibility.

### 3.2 Click-N-Type [7]

This keyboard was created for a person with limited hand motor control, and provides more customization options than the Windows keyboard.

Some features of this keyboard:

* There are multiple keyboard layout that can be used. It comes default with QWERY, alphabetical, and speed (where most used keys are at the start). Users can also create their own custom key layouts and use those with this keyboard.
* Contains word prediction. This appears in a separate window from the keyboard.
* Has macros abilities. User can store string of up to 255 keypresses in a single button. These keys are typed into a buffer and then stored as macros. These macros can contain non-characters such as the enter key or shift key as well.
* Window can be resized for larger or smaller keys.

Some complications with this keyboard:

* There is only one scanning method for this system. It uses a block/group scan followed by radar scanning.
	+ This is a complex scanning method and requires quite a bit of cognitive effort.
	+ The smaller the keyboard on the screen, the more accurate a person would have to be with their crosshair selection and the more keys there would be in each group.
* It is a very involved process to allow this keyboard to run on the login screen.

This keyboard has a lot of great features, but the biggest downside is the scanning method it uses. For anyone who struggles with cognitive intense tasks may struggle with the pattern available.

### 3.3 WiViK [8]

This keyboard is a keyboard focused on accessibility for a large range of users. It does have some features that are useful for single-switch access.

These features include:

* There are multiple scanning methods available that the user can pick from. User can select between automatic or inverse/step then for each of those choose from row/column, item, or quadrant scanning patterns.
* The scan rate can be set to the user's needs.
* There are additional buttons for desktop commands such as scroll, resize, move windows, scan menus, emulate mouse movements, etc.
* Provides several key layouts for different scanning methods. One of these is frequently used letters arranged in the upper left corner to allow them to be first in the scan order.
* Provides word prediction in a list along the side of the keyboard.

From the review done on literature on this keyboard it does not appear to:

* Store macros for future use.
* Have a way to go back when the incorrect quadrant/row was selected.
* Adjust how many times the section is scanned before stopping if a mistake is made.

### 3.4 DiscoverPro [9]

This keyboard is designed for single-switch use. All information comes from the introduction video for the software.

The features shown include:

* Sets of application specific keys and the ability to add custom macros for operations that are not already included. These special keys can be given custom audio ques for quick recognition.
* Large set of preferences including:
	+ Scanning speed
	+ Number of cycles before stopping scan
	+ Auditory and visual cues for scanning and selection
* Custom key layouts can be created with more or less keys then the default. With these layouts, the scanning groups can be edited, and the size of keys can be adjusted as needed.
* There is a stop scanning button, to allow user to skip the wait for the scan cycles to complete if they make a mistake.
* Provides word prediction with automatic period and space placement. Predicted words are placed in the top row (first in scanning order).
* There is an option to simulate a mouse. This can be done with buttons on the keyboard or by creating markers on the screen that are mapped to buttons, which when pressed, move the mouse to that location and perform any tasks that are set to be done after the movement.

This was the most comprehensive keyboard for single-switch use, but is unavailable for use. This keyboard looks to require quite a bit of assistance in the beginning to get the keyboard set up.

## 4. Scanning

 Scanning is a method that successively moves through items, and gives the user a chance to select the current item, or group of items, that the cursor is currently on. There are different ways to scan items such as automatically or manually. Manual scanning involves the user controlling the forward, or backward, movement of the cursor and selecting the input when they have reached the one they wish to select. Manual scanning tends to need the user to have the ability to use multiple keys or switches therefore making it not very suitable for single-switch users. Automatic scanning is the type used by most single-switch users. Automatic scanning puts the control of the movement of the scanning into the system’s control. The system controls the cursor, and the user only has to trigger their input device when the cursor reaches the item, or group of items, they wish to select.

Scanning has many advantages that make it a good way to provide single-switch accessibility to keyboards. As Jones says "scanning keyboards have the advantage of being visible and use recognition rather than recall. This is less cognitive effort when compared to the systems using some kind of coding" [3]. Due to the reduction of cognitive effort from other methods, scanning will help to reduce the onset of physical fatigue that the user may experience. "Another advantage of scanning is that it can be used by visually impaired individuals who cannot see vocabulary items on a display" [4]. As Fishman goes on to say, this "can be done through the use of auditory rather than visual presentation of items to the individual" [4]. Scanning also has the advantage of there being many different scanning patterns. Systems that implement many of these allow users to choose a scanning pattern that suits their needs and preferences.

There are also some disadvantages of scanning that need to be kept in mind. As stated by Dalrymple, "scanning mode somewhat analogous to 'hunt and pick' typing" [10]. This implies that scanning can be slow, and as Jones mentions, "this disadvantage is the waiting time for the scan to reach the desired letter" [3]. If this time is too long, then users may get bored and unwilling to continue their task. Another disadvantage, one that happens often with new scanning users, is that "frequently, they follow the cursor stepping through each item and are not prepared to activate the switch when the desired item is highlighted" [11]. With scanning, users need to learn how to keep track of both the item they want to input and where the cursor currently is.

However, "despite its limitations, scanning is often the only alternative for individuals who cannot use the other interfaces. Technologies such as eye gaze and speech recognition may not be useable for individuals with severe spasticity, poor head control, or limited verbal abilities" [1].

### 4.1 Scanning Method

#### 4.1.1 Provide a scanning method

As the system needs to provide at least one scanning method to make the virtual keyboard accessible to single-switch users, it then becomes a matter of which method should be the default. The very basic scanning technique is single key scanning, where there is no grouping, the cursor just moves to each individual item, one after another. However, as Lee and Thomas explain, "the client's rate of input becomes greatly decreased if this element scanning technique is used with larger selections sets" [11]. This means that other methods, such as row/column or row/group/column, will need to be looked at to find the one most suited for users.

#### 4.1.2 Provide a default scanning method

To assist the user in getting started using a keyboard right away, possibly without assistance, the system should supply a default scanning method. As discussed in the previous subsection, the default method needs to be one that will work, not perfectly, but enough for the widest range of users to be able to start with. The best choices for this is row/column or row/group/column.

#### 4.1.3 Provide an appropriate scanning method

"One way of speeding up the selection process is to move away from a linear array of choices, to one of grouping" [3]. Grouping can be done by moving to row/column scanning, or even to row/group/column. Row/column scanning is where the scan cursor move over the rows of the keyboard first, and then scans through the items in the row once a row has been selected. Row/group/column is where the cursor moves over the rows, then when a row is selected the items in the row are grouped into sections and the cursor moves over each section, only moving to individual items when a group has been selected. In the case of a keyboard, the groups of a row might be groups of 3-4 keys next to each other in the row.

 It is mentioned in Lee and Thomas’ book that, "It is suggested that two-dimensional scanning as in row-column scanning be used for sets that have 10-20 items, and that three-dimensional scanning such as page/row/column be used for sets that exceed the space of one selection screen" [11]. This would indicate that the best method for a large space, such as a full set of keyboard keys, would be to have 3 levels of scanning. In using more levels of grouping, one needs to keep in mind that "multi-stage selection modes are employed at the cost of more user activations" [2] and that "the software developer must take into account the tradeoff of the saved time with the additional activations, in relation to the disabled person's capability of -providing these activations" [2].

#### 4.1.4 Provide multiple scanning methods

With just one scanning method every user must use the same one regardless of if that method is too difficult for them to use or too simple and slow. It also prevents users from being able to change the scanning method they are using to something less cognitively intensive, or physically intensive, when they are getting fatigued. This is supported when Fishman says, "In order to provide greater flexibility, many electronic aids offer more than one selection technique. The advantage of this is the greater range of options available to the user, who may still be able to use the same aid when physical skills improve or deteriorate" [4]. More support of multiple scanning methods appears in a study with children done by Treviranus and Tannock where they write, "because scanning is considerably slower than writing or typing, these children will require the fastest scanning method that is commensurate with their physical and cognitive ability; thus, a system that provides a range of scanning methods is required" [5].

Some scanning methods, such as those involving high cognitive demands like crosshair selection, may be too cognitively intensive for some users. [7] The user needs to have the option of switching to a less demanding method in an accessible way. A user should not get stuck on a scanning method they cannot use while switching scanning methods.

Some users will find that switching methods of scanning is too cognitively intensive to do. Some users may be unable to understand switching between methods, or the be able to learn how to use multiple methods. For this reasons the option to switch scanning methods should be up to the user, and the system should not change scanning methods on the user.

#### 4.1.5 Provide method to switch scanning method

When multiple scanning methods are provided, a user will need to be able to switch between the scanning methods if they so choose to for reasons discussed about. The scanning method can be set in the settings menu, however, to increase independent use of the virtual keyboard it would be better to provide the user with a method to change the scanning method that can be selected from the scan of the keyboard.

As noted in the previous subsection, this switching must be accessible, and the user should not be able to get stuck on a scanning method that they are unable to use while switching scanning methods.

####  4.1.6 Equal scanning groups

 When the scanning method involves grouping items the groups these items are in should be relatively equal in size. This will keep the scan moving through the groups in equal sections. It prevents the scan from doing something weird like waiting for a second on a group of 7 keys and then waiting another second on one specific key because the groups in a row were not equal.

 From looking at the on-screen keyboards reviewed scanning methods that used groups all group items as equally as possible and the groups usually did not contain more than 5 items per group. [8] [9]

### 4.2 Scanning Speed

#### 4.2.1 Allow change to scanning speed

Per Thompson, "scanning can be fatiguing due to the amount of time one has to wait to select a word/concept symbol" [12]. This means that if the scanning speed is slower than the user requires can result in more fatigue than necessary. At the same time, "the cursor must be set at a speed that the consumer can casually use" [13]. If the scanning speed is too fast the user will not be able to activate the switch in time. This will result in the user making errors and could prevent them from being able to use the system.

"Many electronic aids offer as much flexibility as possible within a specific technique. For example, if an aid is controlled through scanning, a variable rate of scanning allows adjusting the device to the user's abilities" [4]. Allowing the user to set the scanning speed from a range of values is a feature supplied by all on-screen keyboards examined. The ability of a user to adjust scanning speed allows users to adjust the scanning speed to their unique abilities. Of the virtual keyboard examined the scan rate range tended to be from between 0.5 seconds up to 3 seconds.

#### 4.2.2 Provide appropriate default scanning speed

The default speed on the virtual keyboards examined tend to be set around 1 second, but the speed a user will need to use will depend on many factors. Some of these factors are the user’s fatigue (both cognitive and physical), how quickly the user is able to activate their switch, activation method of the switch, and the user’s comprehension of whether or not the item the cursor is currently on is the method they wish to input. [13]

#### 4.2.3 Automatic adjustments of scanning speed

To fine tune the scanning speed to a user's specific abilities, and change in abilities over time the scanning speed "could be automatically adjusted during use" [2]. Having the system monitor user response speed and error rates could allow the system to adjust the speed to fit the user's current needs. With this type of automatic adjustment, the user would have to be made aware that it is being done to avoid surprises, and must be able to reject the use of this automatic adjustment if they are not comfortable with it.

### 4.3 Scan Start

#### 4.3.1 Wait for switch

Users of scanning may not want the keyboard to scan constantly. While performing tasks such as reading a page, or concentrating on a picture, the scanning of the keyboard may be a distraction preventing them from completing their task. The virtual keyboard can prevent this problem by waiting for the user to activate their switch. This allows the user to control when the keyboard is scanning. This method is employed by the DiscoverPro keyboard, and in a paper about single-switch games the One Switch Invaders game had "the aliens wait until the first 'click' to start falling" [14].

#### 4.3.2 Short delay after input

One of the priorities of scanning is to "be sure there is enough time for the user to use any prediction features effectively, i.e., to search and select the list on the first pass" [12]. Beyond the predictive features, there also needs to be enough time for the user to find the next item they wish to input. This can be done as a scan delay after the previous input. The scanning pauses to let the user find the next thing they wish to input.

#### 4.3.3 Disable short delay

The short delay after input will increase the time that it takes for the user to reach their next selection. Some users may not need this extra time, and would prefer to not be slowed down by it. This means that the system should provide a way for the user to disable this if they want to. [2]

#### 4.3.4 Provide adjustment to delay time

Scan delay should be adjustable so users who require more time to read predictive items or search for next item have the time they require [2].

### 4.4 Re-scan and skip scan

For many users "it is very frustrating to have just missed your selection and have to wait for a complete repeat of the whole scan" [3].

#### 4.4.1 Provide re-scan method

López, Corno, and Russis advise that for switch based games the system should "reduce consequences of errors, ensuring that errors due to difficulties completing rapid or time-sensitive actions do not impair fun" [14] in their single-switch games. This reduction of consequences of error can be applied to the scanning of a virtual keyboard by providing a way for the user to re-scan a section if they missed the selection. This is done in the Windows 10 keyboard and DiscoverPro by cycling through the same scan a few times before returning to a higher scan level. In Jones’ paper the virtual keyboard has "a 'go back' symbol on the end of each scanning row ... the user can select the 'go back' as they reach the end of the row in which they missed making the selection" [3].

#### 4.4.2 Provide skip scan method

During the selection process a user may make a mistake and select the wrong group or row. To reduce the cost of these mistakes the DiscoverPro keyboard implemented a skip scan button at the beginning of each group and row. This button gives users a chance to go back a level of scanning without having to wait the number of cycles of re-scan that will happen before automatically going up a scan level. [9]

### 4.5 Scanning Indicators

#### 4.5.1 Provide visual indication of scanning

Users need to "be able to coordinate visual tracking of the cursor and control list" [11]. To assist users in keeping track of where the scan cursor currently is visual indication of what is currently being scanned needs to be present. These visuals are supplied with every on-screen keyboard reviewed.

#### 4.5.2 Provide auditory indication of scanning

Some users may benefit from auditory indication of scanning. This could be a simple tone, or various tones depending on what is being scanned, or even audio description of individual keys being scanned. This audio may be distracting for some users, and the user should be able to turn it off if needed [2]. The Windows 10 keyboard and DiscoverPro keyboard both provided audio cues during scanning.

#### 4.5.3 Provide customization of audio cues

The DiscoverPro keyboard provided options for custom simple audio cues or text to speech. [9] Custom audio cues give the users the ability to match certain keys to certain sounds. This can be useful for users who are unable to see the keyboard.

### 4.6 Selection Indicators

As visual keyboards do not have physical indications of selection acceptance, such as the feel of the key being pressed, users using virtual keyboards will have to rely on other forms of feedback.

#### 4.6.1 Provide visual indication of selection

Visual indication of selection will allow sighted users to see that a selection was made. This feedback notifies users that their input has been accepted so they do not have to check. This could be done by colour change or popup.

#### 4.6.2 Provide auditory indication of selection

As found by Steriadis and Constantinou, users "had no problem with the production of a short sound whenever an input was accepted. This operated as a notification of their input's acceptance so they did not have to check it by themselves" [2]. For some users, this may not be a necessity, but some users, especially visually impaired users, this audio feedback on selection might be needed.

#### 4.6.3 Disable auditory indication of selection

Some users may not wish to hear the audio indication of selection. They may find it distracting or unnecessary [2]. Therefore, the system needs to provide a method for the user to turn off this audio que.

## 5. Keyboard Arrangement and Customization

The way the keys on the virtual keyboard, and additional features such as predication are arranged on the keyboard influences the efficiency of use. This efficiency is also affected by the keys presented, and the look of the keys presented. This is echoed by Treviranus and Tannock when it is said that efficiency of the scan is "affected by the appearance of the scan-items (e.g. dimensions, font, color, boldness, etc.)" [2]. For users to use the keyboard "there must be a clear simple presentation of choices" [5].

### 5.1 Keys

#### 5.1.1 Adjustment of all key sizes

Some users may need all the keys to be larger to see them. The virtual keyboard used in “Virtual keyboard with scanning and augmented by prediction” allowed the window to be resized if the user needed larger letters [3]. This is a technique that is employed by all the on-screen keyboards that were reviewed.

It is important to remember that "there are tradeoffs between sharing screen space with the applications (that is, you need a small floating window as practical) and coping with users who have poor eyesight (where a large window would help)" [3]. Being able to resize the window as many times as needed will allow the user to adjust the keyboard to the different tasks they may be undertaking.

It is also important to only change the size of the keys when the keyboard is resized. Changes in key location may change the scan order, and "changes in the scan order can result in selection errors" [2].

#### 5.1.2 Provide individual key size adjustment

Users may wish to be able to adjust the size of individual keys. Making a key bigger relative to the keys around it will cause the key to stand out and make it easier for the user to see this key. The DiscoverPro keyboard allowed users to increase the space certain keys take [9]. While this is not a necessity for keyboard use, this customization can be very helpful for some users.

#### 5.1.3 Provide key appearance settings

As mentioned earlier, the appearance of the items being scanned can affect efficiency [2]. Some users may not be able to read the default font, or colour. Being able to change these factors can make the system more accessible to them. In “Virtual keyboard with scanning and augmented by prediction” the virtual keyboard used a keyboard where "keys are colour coded to aid searching" [3]. By giving the user a way to customize key colours they can design a colour coded pattern that will help them find keys needed faster.

#### 5.1.4 Keys reflect their input

Keys on the virtual keyboard should represent what they would input if selected. This is especially important when looking at character keys. If 'Caps Lock' is on or the 'shift' key has been pressed "the key tops actually show what character will be entered when selected" [3]. Having a mismatch between what the user thinks is being entered on selection and what is actually entered can cause the user confusion and they may make mistakes. Selecting 'shift' should change the keys to their shifted form [3].

### 5.2 Keyboard Layout

For single-switch users "order of the graphical keys in scanning keyboards is crucial for their performance, since it defines the access time of each character and thus the overall typing rate" [2]. This means that the typically QUERTY layout is unlikely to be the best choice for a single-switch user. Having a virtual keyboard means that key appearances and locations are not bound to a physical system. This means that giving users different key layouts, and even allowing them to design their own should be a feature easily included.

#### 5.2.1 Provide multiple keyboard layouts

As discussed QUERTY is not likely to be the best choice for most single-switch users. However, if the keyboard is to not just be used by single-switch users it would be helpful to include the QUERTY layout. The virtual keyboard being developed might be used by users with a wide range of abilities, or the helper of a single-switch user might also be using the keyboard at times, and these users could be familiar with, and are efficient with, the QUERTY layout.

Providing multiple layouts will allow users to pick one of these preset layouts that works best for them. If customization of layouts is supported, then these can also serve as a basis for these modifications.

#### 5.2.2 Switch between layouts

 The user might wish to use different layouts for different tasks, the user should be able to switch between layouts when they need to. Different layouts might be more useful or effective for different tasks, such as a layout that has commonly used shortcuts for word processing earlier in the scan pattern specifically used for word processing.

 Having a key on the keyboard would allow for the user to always be able to switch between layouts independently. They would not have to go into a settings menu and change it (this might be a difficult or impossible task for some users to complete). If there is a key on the keyboard, then the key for switching layout must always be in the same place. Having it move every time the layout changes could make it difficult to find, and might prevent some users from being able or willing to change the layout.

#### 5.2.3 Provide frequency based layout

There is a certain trade off that happens when a user needs to learn a layout different then the 'standard' layout. However, "the speed advantage with a layout optimized for English more than compensates of the unusual layout and only needs a short initial learning time" [3].

The best of these optimized layouts for scanning is one that is based on frequency. According to Lee and Thomas, "revising the item layout is another revision that can enhance efficiency. The item layout defines the positions of each item in the scanning grid. And efficient layout presents the most frequently used items before less frequently used items" [11]. The paper goes on to explain that "a scanning grid with letters laid out in frequency-based order is 25 percent more efficient than a layout in alphabetical order" [11]. This idea is supported by Jones when it is said that "by placing the more frequently used letter earlier in the scan sequence the overall performance can be further improved" [3]. Reducing the number of keys that need to be scanned through will increase the input speed of the user.

#### 5.2.4 User can customize, save, load, and share key layout

Some users may benefit from being able to customize the layout of the keyboard beyond the provided layouts. They may have personal preferences of locations of certain keys, or use some keys more often than others. Providing a way for the users to create their own layout to use is a solution to this. The DiscoverPro keyboard and Click-N-Type keyboard both supplied users with methods to change the layout of the keyboard, and allowed the user to make this custom keyboard the default. [7] [9]

To provide the user the ability to keep and use multiple customized keyboards the virtual keyboard will need to provide a way for the user to save the customized keyboards and load them again. The keyboard, and user, would benefit from the ability to share customized keyboards with others. One user might design a keyboard that works great for a specific application and if they are able to share it other users could take that keyboard, load it, and use it without having to do all the work of designing it themselves. This increases the accessibility of the system, and also promotes the idea of an inclusive community where users help each other to accomplish their tasks.

#### 5.2.5 Layout does not change on own

"The fixed layout of the remainder of the keyboards is one that the users will gradually learn and spend less time visually searching" [3]. Being able to learn the layout of the keyboard is important when it comes to efficiency. As put by Steriadis and Constantinou, "frequent changes in the keyboard layout will delay the user in finding the target character" [2]. A constantly reordered layout will prevent the users from being able to use recall to speed up the selection process. "Reordering requires that users either learn many gird arrangements or search the grid for their desired option upon each reordering" [15]. This means that the layout should not change unless the user changes the layout themselves.

#### 5.2.6 Placement of Predictive Features

Word and letter prediction can be used to help user finish pieces or writing faster. However, the effectiveness of these tools is dependent on their placement. Character prediction will only be useful if it is near the start of the scanning order because those characters will already be somewhere in the scan. Having word prediction at the top allows users to select from those first. This can result in the user being able to skip the rest of the scan.

These ideas are supported by Koester and Simpson when they write, "For group organization … it general works best to put the letters+prediction group first" [1], and when Lazzaro writes that, "based on the frequency of certain words, the word predictor moves commonly used words to the top of the menu choices, permitting you to select words and phrases with increased speed" [16].

## 6. Modifier Keys, Navigation Keys, and Macros

### 6.1 Modifier Keys

#### 6.1.1 Modifier keys on-screen

The user should be able to navigate their computer with just a keyboard. This is done through sequences of modifier keys and character keys. For a user to be able to control the system with just a keyboard the modifier keys must be on the virtual keyboard, and able to be selected with scanning. This is seen in many of the on-screen keyboards reviewed such as the Windows 10 on-screen keyboard, and the Click-N-Type keyboard. These keys were also included in the keyboard used in “Virtual keyboard with scanning and augmented by prediction”. This is explained with the quote, "the last row has a range of keys that users may need, for instance you can send alt-sequences that can manipulate operating system functions without the need to go through ordinary menu selections" [3].

#### 6.1.2 Sticky Keys default

With a virtual keyboard, and especially with single-switch input, only one key can be selected at once. Without Sticky Keys always being on, the user would not be able to use modifier keys. This is a default for all the virtual keyboards that were reviewed.

### 6.2 Navigation Keys

#### 6.2.1 Navigation keys on keyboard

 The users may wish to use the navigation specific keys that come on a regular keyboard. These keys, such as the arrow keys, home, and end, allow a user to navigate some interfaces and therefore should be included on the keys that can be used from the virtual keyboard.

 Not every user will want to use these keys, so even though they are to be there by default, the user should have the option to remove these keys from the keys displayed on the virtual keyboard if they so choose.

### 6.3 Macros

"Macros you define are permanently stored on the computer's hard disk drive and can be used repeatedly when necessary. The ability to define hundreds of macros can assist you for word processing, database management, spreadsheets, browsing, book-keeping, programming, and even for games" [16].

#### 6.3.1 Ability to add macros

"Macro programs remember keystrokes and mouse movements and let you play them back whenever you need. You can assign a macro to almost any key on the keyboard and have it play back hundreds of keystrokes and mouse movements in exact sequence" [16]. Being able to store whatever sequence of keys the user wants is a good way to decrease the time the user needs to spend on tasks. With single-switch users, macros can be created with frequently used sequences of keys that can be access by one or a couple of key strokes. This helps reduce physical and mental fatigue because the user does not have to trigger their switch as many times and they do not have to go through the, possibly very long, process of entering the entire sequence key by key. Good macros can help the user input information, and perform tasks much faster than they would be able to if they had to scan for and select every item in the sequence. While macros are not essential to the use of the system, macros "are an inexpensive method of making software programs more accessible and easier to use" [16].

#### 6.3.2 Add macro buttons to keyboard

There are many ways to support macro use. The Clink N Type keyboard allows user to store macros in <ALT>+<character> key combinations, or by using their 'macro' key which acts in the place of <ALT> [7]. The DiscoverPro keyboard provided a way for users to create custom buttons on the keyboard [9]. These additional buttons meant that the users would only have to scan to and select one item instead of multiple to use the macros.

#### 6.3.3 Ability to delete macros

Over time some of the macros a user creates may no longer be used. These macros might take up key combinations, or button space that the user may want to be able to use for other macros.

#### 6.3.4 Ability to edit macros

Some macros might become out of date, or need to be modified slightly to be useful to the user. Without the ability to edit a macro, the user will be forced to remove and re-add a macro. If the macro is very long this might be a struggle. The user might forget the exact sequence of keys in the macro, and it could be very time consuming to re-enter the entire macro. The ability to edit a macro allows the user to keep the parts of the macros that are correct, and just adjust the parts that need to be updated.

## 7. Prediction

The paper “Virtual keyboard with scanning and augmented by prediction” offers two types of prediction, it recommends that a virtual keyboard "provide both character and word completion" [3]. Character predication determines what the next character a user might want to enter is and offers those letters as suggestions. Word prediction tried to determine what the next word the user may wish to enter is and offer those suggestions to the user.

### 7.1 Character Prediction

#### 7.1.1 Provide character prediction

Character prediction is efficiently when done by "predicting what the next most likely set of characters are and offering that always in the first group" [3]. If character predictions are not near the start of the scan there is a risk that the predications for characters will come after the actual character.

Character predication will only reduce the scan time of selecting a letter. It will not reduce the number of switch triggers needed to reach a letter. This type of prediction also required user to quickly read the letters predicted. These two down sides to character prediction make it a feature that may not be suitable for everyone. If character prediction is added, there should be an option to turn it off if the user does not wish to use it.

#### 7.1.2 Provide appropriate number of predictions

When characters are predicted, these predictions are likely to change after every input. This means that a user must read these each time and determine if any of them are the input they want to select. Because there is often not much time before scanning starts up again after a selection there cannot be so many items predicted that the user cannot read them all before the scan cursor moves beyond the group they are contained in.

The keyboard detailed in Jones’ paper uses the prediction of the four most likely letters follow the last selection [3]. Too few characters predicted and the wanted one may not often be there, but too many and the user will not be able to read them. There explanation for four was that "limiting the predictions to only four means that visual search times are kept brief as they are constantly updated and so cannot be learnt" [3]. The on-screen keyboards reviewed, that included character predictions, offered 4-7 characters.

As there may not be any words predicted at times, there is no minimum number of predictions required. However, if there are predictions found then at least one of those predictions should be shown to the user.

#### 7.1.3 Ability to turn off character prediction

Not every user is going to want to use character prediction. All the characters predicted will still be avaliable on the virtual keyboard, so the only gain from character prediction is time and possibly some savings of physical fatigue. Some users might be unable to keep up with the consant changing letters predicted and wish to not use them. Being able to turn on or off character prediction will allow the user a greater level of individualization and a wider range of accessibility.

### 7.2 Word Prediction

#### 7.2.1 Provide word prediction

"Word-prediction software can speed up the data-entry process for an individual using a keyboard, an adapted switch, or another form of alternative input device. The aim is to obtain the maximum data entry for the fewest number of keystrokes" [16]. Unlike character prediction, word prediction can save the user many switch triggers. This helps to reduce the onset of physical fatigue. These predications should be at the top of the scan order to reduce time to reach items. Word prediction was included in all four virtual keyboards that were reviewed.

#### 7.2.2 Prediction based on frequency

Jones explains prediction based on frequency as "word completion rather than prediction, using a pre-loaded dictionary that is added to as the user types" [3]. "This dictionary contains an expected frequency of use" [3]. This means that predictions are not given purely based on alphabetical order, but uses the frequency of word use as the basis of predicting words. The paper explains that other methods tend to predict words that the user is unlikely to want to use [3].

#### 7.2.3 Ability to add to dictionary

For word prediction, there may be some words not initially predicted that the user wants to have offered as a prediction at times. "Many word predictors also contain a user-definable dictionary that allows individuals to add their own custom words and phrases" [16]. This is a customization that can help to reduce the amount of switch triggers needed for words users might commonly use.

#### 7.2.4 Use appropriate number of predictions

As with character prediction, too many words and the user will not be able to read them all before the scan moves past them. Too few predictions and there is a larger chance that the word the user wants will not be in the predicted words. Number of words predicted may be restricted by the space on the virtual keyboard to display predicted words.

Jones also uses four word predictions. The explanation for this is the same as was given for character prediction. [3]

#### 7.2.5 Adding space after selected prediction

All on-screen keyboards reviewed used a feature where after a predicted word was selected a space was added after. This addition reduces the need for the user to manually select this space which saves single-switch users the time and switch triggers needed to add the space.

## 8. Accessible Help

Not every problem encountered by a switch-user can be resolved by the user themselves. Single-switch users may require assistance in finding help and understanding and applying the ideas presented by help. However, the tools should be there for a user to be able to find and browse help independently if they are able to.

### 8.1 Accessible Help

#### 8.1.1 Help available with scanning

For a single-switch user to independently access help for a virtual keyboard, the keyboard must have the help option available. This could take the form of a key or keyboard shortcut that can be selected. A key would be preferred as a user would not need to remember a command in order to access help, which goes by to recognition over recall [3].

#### 8.1.2 Help navigated with keyboard

If the user can enter the help context without assistance, then they should also be able to navigate the help without assistance. If help can be controlled by the virtual keyboard, then the single-switch user will be able to browse help as they need.

# Design

All guidelines have been drawn together from the analysis of the topic presented above. Specific guideline titles match their section and title in the analysis for easy reference. The analysis contains the explanation or reasoning for a specific guideline and why it is present in the design.

## 1. Scope

These guidelines aim to cover functionality that can be added onto an existing on-screen keyboard or one being newly created. They focus on what functionality is required to allow single-switch users to use the virtual keyboard, and for the keyboard to be efficient for them. These guidelines may benefit non-single-switch users as well, and none of the guidelines should prevent non-single-switch users from being able to use the virtual keyboard if implemented.

## 2. Definitions

**2.1**

**Alphanumeric key**

"keys used in normal text typing including letters, numbers and some punctuation keys" [6]

**2.2**

**Modifier key**

"keys which modify the meaning of other keys pressed down at the same time" [6]

**2.3**

**Navigation key**

“keys used to move within and between user interface objects. The arrow keys, tab, space, enter, home, end, page up and page down are navigation keys” [6]

**2.4**

**Scan Cursor**

the current row, group, column, or individual item currently able to be selected

**2.5**

**Scanning**

successively moving scan cursor through items

**2.6**

**Scan Level**

the depth the scan is currently narrowed to

**2.7**

**Scan cycle**

the completion of the cursor visiting every item currently available to be selected

**2.8**

**Scanning speed**

length of time the cursor stays on an item before going to the next one

**2.9**

**Automatic Scanning**

scanning that does not require user to control progress through items

**2.10**

**Scanning Method**

the pattern the scan cursor takes while scanning items

**2.11**

**row/column**

cursor scans through each row, and moves to scan each column when switch is triggered

**2.12**

**row/group/column**

cursor scans through each row, then through groups in selected row, and then columns within those groups

NOTE Group and row can be interchanged depending on the method. This scanning method would be referred to as group/row/column.

**2.13**

**Word prediction**

generation of a set of words, of which one might be used next

**2.14**

**Character prediction**

generation of a set of individual characters, of which one might be used next

**2.15**

**Macros**

an instruction that expands automatically into a string of characters and keystrokes

## 3. Scanning

### 3.1 Scanning Methods

#### 3.1.1 Provide a scanning method

The virtual keyboard shall provide at least one scanning method. [3]

EXAMPLE 1 The virtual keyboard supplied a method of scanning that scans keys by rows and then columns.

EXAMPLE 2 The virtual keyboard supplies a method that scans by rows, then groups, then columns.

#### 3.1.2 Provide a default scanning method

The virtual keyboard shall supply a default scanning method that can be used by users immediately without having to manually select one.

NOTE For information on which scanning method to use as default see guidance on providing an appropriate scanning method (3.1.3)

EXAMPLE When single-switch scanning is turned on, without any settings being changed, it immediately starts scanning using the row/column pattern.

#### 3.1.3 Provide an appropriate default scanning method

a) Default scanning method supplied by the virtual keyboard should not be a complex method. [2]

EXAMPLE 1 If the default scanning method is not complex, users with cognitive disabilities may be able to use the system without having to change from the default scanning method.

b) Default scanning method should be row/column scanning or row/group/column. [10]

NOTE row/column is general better for 10-20 items, where row/group/column is better suited for more than 20 items [11]

c) Scanning methods involving cross hairs or relative sized quadrants are more cognitively intensive methods, and should not be included as the default. [7]

EXAMPLE 2 Some users may have cognitive disabilities that would prevent them from understanding how to operate multiple types of selection in one scanning method. Not having those the default scanning method will allow more users to be able to use the system without individualization.

#### 3.1.4 Provide multiple scanning methods

The virtual keyboard should provide additional scanning methods that the user can choose from. [4] [5]

NOTE A user might find the current scanning method too simple or too difficult to use. Supplying multiple methods allows them to adjust system to their capabilities. [5]

EXAMPLE Virtual keyboard could come supplied with row/column, row/group/column, individual item, and cross hair scanning where any of them can be selected from a preferences or settings section.

#### 3.1.5 Provide method to switch scanning method

If multiple scanning methods are provided, the virtual keyboard should provide a method to change which scanning method is active. [4]

EXAMPLE 1 A button is placed within the scan that, when selected, will change to the next technique.

EXAMPLE 2 A button placed within the scan that, when selected, produces a list of scan methods that is scanned through and lets the user select the method they would like.

#### 3.1.6 Equal scanning groups

The scanning methods provided by the virtual keyboard should group items in roughly equally sized groups with no more than 5 items per group.

EXAMPLE A row of keys has 12 keys in the row. The keys in the row are grouped into groups of 4 keys.

### 3.2 Scanning Speed

#### 3.2.1 Allow change to scanning speed

The virtual keyboard shall allow user to change the scanning speed from, at minimum, between 0.5 seconds to 3 seconds. [13]

EXAMPLE A range of speeds, from 0.5 seconds to 3 seconds, is supplied in settings that the user can adjust as needed by increments of 0.5 seconds.

#### 3.2. Provide default scanning speed

The virtual keyboard should set default scanning speed that lets as many users as possible use the system without having to change the scanning speed, such as 1-2 seconds.

EXAMPLE The default scanning speed, without changing any settings, is set to 1 second.

#### 3.2.2 Automatic adjustment of scanning speed

a) The virtual keyboard may provide method to automatically adjust scanning speed for user.

EXAMPLE 1 The virtual keyboard monitors user's response time and errors and adjusts the speed of scanning to better meet the user's needs.

b) If automatic adjustment is being applied, the virtual keyboard shall make users aware that this automatic adjustment is happening.

EXAMPLE 2 When system speeds up the scanning speed the action and amount the speed is being increase is displayed briefly overlaid on top of the virtual keyboard.

c) If automatic adjustment is being applied, the virtual keyboard shall allow users the option to disable and re-enable the use of automatic adjustments to scanning.

EXAMPLE 3 Within settings for the virtual keyboard there is a checkbox to enable, or disable, the automatic speed adjustment.

EXAMPLE 4 Some users might not be able to understand the changing speed of automatic scanning, disabling it allows them to be able to continue to use the virtual keyboard to accomplish their tasks.

### 3.3 Starting Scan

#### 3.3.1 Wait for switch

On initial start of virtual keyboard use, when scanning has not already been started, the keyboard should wait for user to trigger switch before starting scan. [1] [14]

NOTE 1 This wait will allow users to start scanning only when they are ready to start scanning.

NOTE 2 This allows the virtual keyboard to stop scanning when the user is not using it. Users might not want the keyboard scanning while they are performing a task that does not involve using the keyboard. They can then choose when to have the keyboard scan again.

EXAMPLE 1 The virtual keyboard has just been started so it waits until the user signifies they are ready to start scanning by activating their switch once. [9]

EXAMPLE 2 Scanning has stopped after period of inactivity, and waits until the user signifies they would like to start scanning again by activating their switch. [9]

#### 3.3.2 Short delay after input

The virtual keyboard should provide an option for a short delay after input, before restarting the scan. [1]

NOTE 1 This delay allows user to read predictive items or search for the next item they wish to select. [1]

NOTE 2 This delay will increase the overall time it takes to reach an item therefore some users might not wish to have this delay. See 3.3.3 for further information on letting users disable the delay.

EXAMPLE The user makes a selection and the scan cursor returns to the top and waits for 2 seconds before moving the cursor.

#### 3.3.3 Disable short delay

If virtual keyboard supplies a short delay after input, the virtual keyboard should provide a way for users to turn this delay on and off. [2]

EXAMPLE A user does not wish to use the short delay deselects the option for this delay in the settings menu.

#### 3.3.4 Provide adjustment to delay time

If short delay after input is provided, the system should provide the user with a way to adjust how long the delay lasts to assist the users who require more time or less time to read predictive text, or find items. [2]

EXAMPLE User is unable to make use of predictive text as they are unable to read all the items before the cursor moves past the section they are contained in. To change this they would be able to go into the settings for the virtual keyboard and increase the amount of time the scan waits after input.

### 3.4 Re-Scan and Skip Scan

#### 3.4.1 Provide re-scan method

a) The virtual keyboard should provide functionality to re-scan a section. [3] [14]

NOTE 1 This function is for when a user misses their selection. Without it the user will be required to wait an entire scan cycle causing wasted time and frustration. [3]

EXAMPLE 1 Scanning of a section is automatically repeated multiple times, 2-3, before returning to higher scan level or stopping scan.

NOTE 2 The scan level corresponds to which part of the scan the user is currently in. For row/column the first scan level would be row and second would be column. Going up a scan level from the second level (columns) would result in the scan returning to scanning rows.

EXAMPLE 2 A button is placed at the end of each section that user can select to return the scan cursor to the start of the section. [3]

b) If automatic repetition of scan level is included (referenced by 3.4.1.a EXAMPLE 1), the virtual keyboard may supply a customization setting to change the number of cycles before returning to the previous level.

EXAMPLE 3 Some users may not require more than one re-scan of a section, but the default is set to 3. They enter the settings menu and change the value for the scan repeat to 1 instead of 3.

#### 3.4.2 Provide skip scan method

The virtual keyboard may provide a button to skip scan cycles.

NOTE 1 This would be used if a user selected the incorrect section by mistake and wished to return immediately instead of waiting for all cycles of that scan section to be completed. [9]

EXAMPLE A button provided at the start of each scan section that will return the scan to the previous level if selected. [9]

### 3.5 Scanning Indicators

#### 3.5.1 Provide visual indication of scanning

The virtual keyboard shall provide visual indication of scanning. [11]

EXAMPLE 1 Set of keys being scanned are indicated by a distinct box around them.

EXAMPLE 2 Individual keys being scanned within a group are outlined when the scan cursor is on them, while the entire group that contains the keys is indicated by a distinct box.

#### 3.5.2 Provide auditory indication of scanning

a) The virtual keyboard should provide auditory indication of scanning. [9]

EXAMPLE 1 A short tone sounds each time the scan cursor moves.

EXAMPLE 2 The current key being scanned is read out by text-to-speech.

b) If the virtual keyboard provides auditory indication of scanning, it should allow ability to turn off these audio cues if they wish to~~, as the audio might become a distraction for users~~. [2]

EXAMPLE 1 User enters the virtual keyboard’s setting menu and disables audio for scanning completely.

EXAMPLE 2 The virtual keyboard has an audio key that, when selected, will mute or unmute all audio from the virtual keyboard.

#### 3.5.3 Provide customization of audio cues

The virtual keyboard may provide user method to customize auditory indicators of scanning. [9]

EXAMPLE 1 User being able to change the tone that sounds for different keys or scan levels.

EXAMPLE 2 User being able to select a tone or text-to-speech for individual keys.

EXAMPLE 3 User being able to select a specific tone for specific keys.

### 3.6 Selection Indicators

#### 3.6.1 Provide visual indication of selection

The virtual keyboard shall provide visual indication of selection.

EXAMPLE 1 The key selected lights up or changes colour briefly to indicate selection.

EXAMPLE 2 The icon or character pops up briefly above the key to indicate selection as shown in Figure 1.



**Figure 1 - Popup of letter on selection** [17]

#### 3.6.2 Provide auditory indication of selection

The virtual keyboard shall provide user the option of auditory indication of selection. [2]

EXAMPLE 1 Short tone sounds when a selection has been made.

EXAMPLE 2 The item that has been selected is spoken by the system using text-to-speech.

#### 3.6.3 Disable auditory indication of selection

The virtual keyboard shall provide a method to disable the auditory indication of selection.

EXAMPLE 1 Some uses might not be able to hear the auditory indication on selection. To ensure that their device is not making noise that might disturb others they need to be able to disable the sound produced when an item is selected.

EXAMPLE 2 A button is provided on the virtual keyboard that, when selected, toggles on or off the audio output.

## 4. Keyboard Arrangement and Customization

### 4.1 Keys

#### 4.1.1 Adjustment of all key sizes

a) The virtual keyboard shall provide way to increase or decrease the size of all keys at once. [3]

EXAMPLE 1 Key sizes are adjusted uniformly when the keyboard window is adjusted.

EXAMPLE 2 In the virtual keyboard settings there is an option to scale the size of the keys.

b) The virtual keyboard should not change the key order on key size change.

NOTE This can confuse those who have memorized the order of the keys, and could change the scan order. [2]

EXAMPLE 3 When the window containing the virtual keyboard is made longer horizontally the keys do not rearrange themselves to fill the space created, instead all key’s increase in size to fill the created space.

#### 4.1.2 Provide individual key size adjustment

a) The virtual keyboard may provide the user a method to customize the size of individual keys. [9]

EXAMPLE 1 User can edit a specific key and make it the width of two keys. [9]

b) The virtual keyboard should provide the user a method of specifying how the scan pattern handlers uniquely sized keys.

EXAMPLE 2 The customization process includes a method to manually edit the scanning groups. With this the user can specify which group, and therefore which order, the new key is placed in.

#### 4.1.3 Provide key appearance settings

The virtual keyboard should provide the option to adjust the appearance of keys. [2]

EXAMPLE Allow users to adjust font, boldness, contrast, and colour of items on the keyboard. [3]

NOTE Being able to do these customizations at both the individual key level, and whole keyboard level will be the most beneficial to the user.

#### 4.1.4 Keys reflect their input

The visuals on keys representing a single character, shall match what will be input upon their selection. [3]

NOTE When the key top does not match the input the user can become confused and make mistakes.

EXAMPLE 1 Selection 'shift' changes the keys to their shifted form. [3]

EXAMPLE 2 With no modifiers currently active, keys appear in their default state (likely this would be lower case characters).

### 4.2 Keyboard Layout

#### 4.2.1 Provide multiple keyboard layouts

The virtual keyboard should provide multiple keyboard layouts, including frequency based and QUERTY. [2]

EXAMPLE A virtual keyboard comes with frequency based, alphabetical, and QUERTY keyboard layouts.

#### 4.2.2 Switch between layouts

a) The virtual keyboard should provide a key for user to switch between keyboard layouts as needed.

EXAMPLE 1 A key is available on the virtual keyboard which, when pressed, brings up a list of layouts that are available to the user. This list is scanned through, allowing the user to select the one they wish to use. Are there other ways that this layout change could be done? Should there be a guideline about not making the user cycle through layouts?

b) The key provided to switch between keyboard layouts should offer user choice of layout to use. It should not just switch the keyboard to the next layout in the list of available layouts.

EXAMPLE 2 Some users may have many layouts available to use and having to navigate to the layout key for each layout between the current one and the target one could be time consuming and frustrating. The button bringing up a list of layouts that are cycled through and can be selected to be applied by the user prevents this problem.

c) The key to change the layout of the virtual keyboard should always be located in the same place.

EXAMPLE 2 The key for changing layouts is always placed in the upper right hand corner of the virtual keyboard. This is locked into the keyboard template and cannot be removed or relocated through keyboard customization.

#### 4.2.3 Provide frequency based layout

The virtual keyboard should provide a static frequency based layout. [3] [1]

NOTE If location of keys change after each input, then users will be unable to learn the layout and will have to search for selection each time. This might negate the advantages of using a frequency based layout. [2] [15]

EXAMPLE Keys that a most frequently used by people are placed closer to the top-left of the virtual keyboard.

  [3]

#### 4.2.4 Users can customize, save, load, and share key layout

a) The virtual keyboard should supply a method to customize the layout of keys on the virtual keyboard. [7] [9]

NOTE This can include both the arrangement of keys, and what keys are present on the keyboard.

EXAMPLE 1 From the settings of the virtual keyboard the user of the virtual keyboard can select customize keyboard which opens a new window allowing user to change the layout and keys on the keyboard.

EXAMPLE 2 The virtual keyboard comes with a secondary system to allow users of it to customize keyboard layouts from the default templates.

b) The virtual keyboard should allow user to save their customized keyboards.

EXAMPLE 3 Once the user has customized the keyboard they can save the keyboard to a file on their local machine.

EXAMPLE 4 The virtual keyboard supplies a cloud service to save virtual keyboards to a user account.

c) The virtual keyboard should allow users to load custom keyboards into the virtual keyboard.

EXAMPLE 5 Customized keyboard layouts are loaded from the user’s machine’s local file system.

EXAMPLE 6 Virtual keyboard uses the internet and cloud services, and the user can load customized keyboard layouts from their user account.

d) The virtual keyboard should supply users a way to share customized keyboards with other users.

EXAMPLE 7 Users can upload their customized layouts to the internet, where other users may download and load them into their own virtual keyboard.

EXAMPLE 8 If cloud storage of layouts is provided the user can share layouts they have stored with other users who have accounts.

#### 4.2.5 Layout does not change on own

The virtual keyboard shall not change the layout of keys on virtual keyboard without the user’s permission. [2] [3]

EXAMPLE The user is asked to confirm selection of a different layout and if they do not confirm within 30 seconds the virtual keyboard will reset back to the previous layout.

#### 4.2.6 Placement of predictive features

If predictive features, such as word and letter, are included, they shall be placed at the start of the scanning order. [1] [16]

EXAMPLE Scanning starts in the upper left hand corner of the virtual keyboard and so predictive features are the first thing in that area that is scanned.

NOTE See section 6 for more information on predictive features.

## 5. Modifier Keys, Navigation Keys, and Macros

### 5.1 Modifier Keys

#### 5.1.1 Modifier keys on-screen

a) The virtual keyboard shall allow all modifier keys to be present on screen. [3]

NOTE 1 Single-switch users rely on modifier keys to operate the system they are using, and therefore need to have these keys available on the keyboard.

EXAMPLE 1 Shift, alt, control, and other modifier keys are placed in a row after rows of character keys.

b) The virtual keyboard should allow the user to remove these keys through an individualization function, or key layout customization, if they wish.

EXAMPLE 2 User never uses the caps lock key and removes it through a keyboard layout customization feature.

EXAMPLE 3 The settings for the virtual keyboard provides the option to hide modifier keys from the layout.

#### 5.1.2 Sticky Keys default

The virtual keyboard shall, by default, always have Sticky keys on.

### 5.2 Navigation Keys

#### 5.2.1 Navigation keys on keyboard

a) The virtual keyboard shall allow all navigation keys to be present on the virtual keyboard.

EXAMPLE 1 Arrow keys and other navigation keys are placed on the virtual keyboard in all provided keyboard layouts by default.

b) The virtual keyboard should allow the user to remove these keys through an individualization function, or key layout customization, if they wish.

EXAMPLE 2 User can create a custom keyboard layout that does not include arrow keys.

EXAMPLE 3 The settings from he virtual keyboard provides and option to not include navigation keys in the layout.

### 5.3 Macros

#### 5.3.1 Ability to add macros

a) The virtual keyboard should proivde the ability for user to add custom macros to keyboard. [16]

EXAMPLE 1 Macros can be set up and bound to key combinations through the settings menu of the virtual keyboard.

EXAMPLE 2 Virtual keyboard has a macros button that, when selected, records all keys entered until the button is selected again. The system stores this sequence in the next avaliable hotkey and displays this hotkey to the user.

b) The virtual keyboard should allow user to store modifier keys, navigation keys, character keys, and combinations of all three in macros. [7] [9] [16]

EXAMPLE 3 The user has stored the key sequence <CTRL>b111-111-1111<CTRL>b in a macro. This would turn on bold font, write 111-111-1111, then turn off bold font.

#### 5.3.2 Add macros buttons to keyboard

The virtual keyboard may supply the user a way to add custom macro buttons to the keyboard layout. [9]

NOTE These are different then storing macros in key combinations. This is adding an additional button to the keyboard that a macro can be specifically assigned to.

EXAMPLE 1 Macro buttons are added to a row at the bottom of the virtual keyboard when the user creates a new macro.

#### 5.3.3 Ability to delete macros

If macros can be added, the virtual keyboard shall supply a way for user to delete macros.

EXAMPLE List of macros can be found from the settings menu. Each macro has the option to delete it present beside the name and action of the macro.

#### 5.3.4 Ability to edit macros

If macros can be added, the virtual keyboard shall supply a way for user to edit macros.

EXAMPLE List of macros can be found from the settings menu. Each macro has the option to edit it present beside the name and action. Part or all of the macro can be edited.

## 6. Prediction

6.1 Character Prediction

#### 6.1.1 Provide character prediction

The virtual keyboard may supply character prediction capabilities. [3]

EXAMPLE After the user inputs a letter the virtual keyboard identifies a few characters that the user might wish to enter next and presents them to the user as items that can be selected.

#### 6.1.2 Provide appropriate number of predictions

If character prediction is provided, the virtual keyboard shall provide an appropriate number of predictions. [3]

NOTE 1 Too many predictions, more than 7, will prevent the user from reading them all before the scanning cursor moves beyond the prediction group.

NOTE 2 The number of predictions presented might be restricted by the size of the keyboard.

NOTE 3 There is not a minimum number of predictions suggested, as there may be cases where there are no predictions based on the previous inputs. However, if there are predictions then at least one should be displayed to the user.

EXAMPLE Five characters are predicted and presented to the user.

#### 6.1.3 Ability to turn off character prediction

If character prediction is provided, the virtual keyboard should provide the use a method to turn character prediction on and off.

EXAMPLE In the settings for the virtual keyboard an option is provided to toggle character prediction.

6.2 Word Prediction

#### 6.2.1 Provide word prediction

The virtual keyboard should supply word prediction capabilities. [16]

EXAMPLE Once the user has input the first letter of a word the system attempts to predict the word the user might want to enter and presents those options as items the user can select from the virtual keyboard. If the input is at the end of a word, the prediction attempts to predict the next word the user may wish to enter.

#### 6.2.2 Prediction based on frequency

If word prediction is supplied, predictions shall be based on frequency of user use with more recently used words being first. [3]

NOTE Words frequently used by the user are more likely to be useful over alphabetical order.

EXAMPLE 1 The user selects 't' as the start of a word. The word prediction gives words such as 'the' and 'there' instead of 'ta', 'taa', or 'taal'.

EXAMPLE 2 The user uses the word ‘bandana’ more frequenctly than ‘banana’ so when the user inputs ‘ban’ the word that shows up first in the prediction list is ‘bandana’ over ‘banana’.

#### 6.2.3 Ability to add to dictionary

a) If word prediction is provided, the virtual keyboard should provide a method to allow the user to add words to the dictionary. [16]

EXAMPLE 1 The user's name is not included in the dictionary. They are able to add this word to the dictionary so that it can be offered by the word prediction system.

b) If word prediction is provided, the virtual keyboard may allow the dictionary to learn words the user enters in order to be able to provide those words back to the user as predictions.

EXAMPLE 2 The user often manual inputs the word CMPT, and the system stores this word in the dictionary allowing it to be used as a prediction for the user.

#### 6.2.4 Use appropriate number of predictions

If word prediction is provided, the virtual keyboard shall provide an appropriate number of predictions. [3]

NOTE 1 Too many predictions, more than 7, will prevent the user from reading them all before the scanning cursor moves beyond the prediction group.

NOTE 2 Number of predictions presented may be restricted by the size of the keyboard.

EXAMPLE Five words are predicted and presented to the user.

#### 6.2.5 Adding space after selected prediction

a) If word prediction is provided, the virtual keyboard should input a space after each word input from the words predicted.

EXAMPLE 1 The user selects the word ‘school’ from the predicted items. The virtual keyboard inputs ‘school ‘ to the place the virtual keyboard is currently sending input to. This space should be removed if there is punctuation added after.

b) If space after selection is included, then in the event a user selects a piece of punctuation after a predicted word, the space that had been added after the word is replaced by the punctuation.

EXAMPLE 2 The user selectes the word ‘school’ from the predicted items. The virtual keyboard inserts a space after it. The user then selects a ‘.’. This results in the space previously added being replaced by the ‘.’. The final input would then be ‘school.’

c) If word prediction and space after prediction selection is provided, the virtual keyboard should give the user the ability to turn adding a space after a predicted word is input off.

EXAMPLE 2 From the settings menu the user can deselect the option to add space after predicted words. This will prevent the system from adding additional spaces.

## 7. Accessible Help

### 7.1 Accessible Help

#### 7.1.1 Help available with scanning

The virtual keyboard shall make help available through scanning.

EXAMPLE Add help button to the keyboard so that it can be selected while scanning.

#### 7.1.2 Help navigated with keyboard

The help provided by the virtual keyboard shall be able to be navigated through the keyboard.

EXAMPLE User can use the scanning on the virtual keyboard to input the keys required to navigate through the help sections. Help should provide instructions on how to navigate to ensure that the keyboard can be used to navigate.

#### 7.1.3 Help provides navigation instructions

The help provided by the virtual keyboard should convey to the user the methods that the user can use to navigate the help.

EXAMPLE When the user enters help, the first page contains instructions on how to navigate the help system.

# Evaluation

## Part A Instructor Evaluation

### Method of Discussion

For the instructor evaluation, I will be listing the comments and suggestions made on my analysis and design and where they are located. For each item, I will discuss what I have done to improve my analysis and design from this feedback. This will be done in a format similar to the class critiques, slightly modified to fit the situation.

**Format:**

Background:

Comment:

* *What the original comment is.*

Context:

* *What was being discussed or what the guideline commented on was*

Location:

* *Where the comment is*

Significance:

* *Why is this comment important to the analysis or design?*
* *Why should the document be updated?*

Solution/Improvement:

* *What was done to the document to fix this?*

In the interest of keeping this discussion focused on the major ideas, I will be omitting the comments that are just grammar corrections or consistency items. These will be fixed in the revised analysis and design document. For any comments that were made multiple times in different places, I will include the first one in this discussion and include the location of others if there is no change to the handling of these. If there is a significant difference between the solution to it, these differences will be noted.

Locations are pre-rearrangement locations.

Version of the Design and Analysis with just the corrections made for this part of the evaluation is contained in file named CMPT480-RevisedAnalysisandDesignPartA.docx. Tracked changes were used to make changes to the document. Evaluator comments remain in document as well.

### Discussion of Instructor Evaluation

1.

Background:

Comment:

* Concepts used in the overview of on-screen keyboards have not yet been introduced. Consider moving further down in the analysis.

Context:

* On-screen keyboard overview comes before analysis for the specific guidelines meaning the observations made about the keyboards were not introduced, or defined yet.

Location:

* Analysis 2. Overview of Existing On-Screen Keyboards

Significance:

* Readers may not know what some of the concepts in the overview mean yet.

Solution/Improvement:

* Moving this section down proved to be a difficult thing. It is referenced in every section following it, and moving it down breaks up the organization of the document with general information then specific guideline information after.
* Instead I chose to include a section of definitions before the overview and included the definitions for concepts that are used in the overview of on-screen keyboards.
	+ Allows section to stay in place, while still providing readers at least some information on what things being referenced to in the overview are.

2.

Background:

Comment:

* “While you briefly described scanning near the start, it would be good to have an introductory paragraph elaborating on what scanning is all about before jumping into the details here.”

Context:

* At the start of the section of scanning.

Location:

* Analysis 3

Significance:

* The section of scanning assumed that the reader would already have understood scanning from introduction. This may not be the case, and results in the section on scanning not having an introduction before jumping straight into the material.

Solution/Improvement:

* Added a short introduction to what scanning is and the differences between manual and automatic scanning.

3.

Background:

Comment:

* “How would users choose between methods?”

Context:

* Questioning how a user would choose between scanning methods.

Location:

* Analysis 3.1.1

Significance:

* There was no mention in the scanning section on how a user would be able to switch between scanning methods.

Solution/Improvement:

* Instead of adding a description of how to switch methods in a section discussing just having a method, I created another guideline later with the title “provide method to switch scanning method” (This is now guideline 4.1.5)

4.

Background:

Comment:

* “What does single key scanning mean?”

Context:

* Provide a scanning method section mentioned single key scanning without defining it.

Location:

* Analysis 3.1.1

Significance:

* Readers may not know what single key scanning is already.

Solution/Improvement:

* Added a short description of what single key scanning is to the subsection.

5.

Background:

Comment:

* “So what are the alternatives?”

Context:

* What are the alternatives to using single key scanning?

Location:

* Analysis 3.1.1

Significance:

* Subsection discusses how single key scanning is slow with large numbers of keys, but does not offer information on what else could be used. These alternatives are what the reader is going to be interesting in knowing.

Solution/Improvement:

* Added suggestions on what some of the alternatives for single key could be.

6.

Background:

Comment:

* “What do each of these mean? Please provide explanations and examples.”

Context:

* Comment on what row/column and row/group/column scanning is.

Location:

* Analysis 3.1.3

Significance:

* Terms that were not previously defined so the reader may not know what they mean. This may prevent them from understanding the analysis.

Solution/Improvement:

* Added explanation of the terms to the subsection.
* These are also included in the section of definitions added (now section 2 of the analysis).

7.

Background:

Comment:

* “Of course changing or selecting also needs to be cognitively accessible. And thus it deserves some discussion here”

Context:

* Under subsection on providing multiple scanning methods. There is discussion on being able to switch methods that is lacking on information about making sure switching and scanning is cognitively accessible to the user.

Location:

* Analysis 3.1.3

Significance:

* In order for the system to be accessible the user will need to be able to switch scanning methods and they should be able to do that in a way that is cognitively accessible to them.

Solution/Improvement:

* Added further analysis on the cognitive side of providing multiple scanning methods.
* This also resulted in additional information being added to the guideline on providing method to switch scanning method, about the accessibility of the scan method switching.

8.

Background:

Comment:

* “What are typical speeds? Are they only related to the user, or are they influenced by the task and/or type of switch?”

Context:

* Asking about what the typical speeds of scanning and what affects them.

Location:

* Analysis 3.2.1

Significance:

* Readers using the guidelines and analysis will need a general idea of what speeds are and what they need to consider when setting the range of speeds available.

Solution/Improvement:

* Consulted on-screen keyboards examined for this project and found what the default scanning speed and speed ranges were. Included this information in the subsection.
* Consulted sources used in the project to see if there was further information on what affects single-switch access. Included short summary of findings in the analysis section.

9.

Background:

Comment:

* “A recommendation belongs as a guldeline and not as a NOTE. "Should" indicates a recommendation within a guideline.”
	+ This is the first one, further comments about this were worded similarly.

Context:

* This came up many times within the design portion. Most times it was due to a NOTE for a guideline including something that wasn’t really a note, but something that could be used as another guideline.

Location:

* 3.1.2, 3.1.3, 3.2.2, 3.3.2, 3.5.2, 4.2.1

Significance:

* These notes were recommendations or further guidance that was not put into a separate piece of guidance. Notes are there to explain the guidance further, not for giving more guidance.

Solution/Improvement:

* Made use of the suggestion, “You could use a list of a), b), c), under the single heading. It would be important to ensure that the wording flowed smoothly” and made some of the notes into subparts of a piece of guidance
* 3.1.2
	+ was broken down into a), b), c) with NOTE 1 and NOTE 2 becoming further guidance. These were re-worded from recommendations to guideline wording.
* 3.1.3
	+ NOTE contained guidance on including way to change the scanning method. This was turned into a separate piece of guidance (“3.1.5 provide method to switch scanning method”).
* 3.2.2
	+ NOTE 1 and NOTE 2 were turned into b) and c) under the subsection 3.2.2. They were reworded to be guidance, rather than a note.
* 3.3.2
	+ NOTE 2 was turned into a separate guideline – now “3.3.3 Disable short delay”
	+ NOTE 3 was turned into a separate guideline – now “3.3.4 Provide adjustment to delay time”
* 3.5.2
	+ Broke the guideline down in to a) and b). NOTE 1 becoming part b.
* 4.2.1
	+ NOTE 1 said that “User should be able to switch between these layouts as needed” this was turned into its own piece of guidance – “4.2.2 Switch between layouts”

10.

Background:

Comment:

* “What guidance is available to select between these two?”

Context:

* It was recommended that the default scanning method was either row/column or row/group/column

Location:

* 3.1.2

Significance:

* There was no guidance given on how a person was supposed to choose which of the two to use. The reader may not know what method is the best of the two, or when they should use one over the other.

Solution/Improvement:

* Added a note to the guideline explaining that row/column is better for 10-20 items, but for anything greater row/group/column is a better choice.

11.

Background:

Comment:

* “you have not had a recommendation about having a default”

Context:

* Comes with a sentence saying, “A user may find the default scanning methods too simple or too difficult to use”

Location:

* 3.1.3

Significance:

* Assumes that there is a default scanning method, but there is no guidance that says a person has to have a default scanning method. This could create a conflict.

Solution/Improvement:

* Created a new piece of guidance titled “Provide a default scanning method”. This piece of guidance was place near the start of the guidance on scanning methods (3.1.2).
* Reworded the recommendation to use the word current rather than default. This guideline should be applicable to more than just the default one.

12.

Background:

Comment:

* “"May" in a standard is a weak recommendation”

Context:

* Some of the notes in the guidelines used the words shall, should, and/or may.

Location:

* 3.1.3, 3.3.1, 3.3.2

Significance:

* These words are used specifically for the guidance, and when used in notes for pieces of guidance imply more guidance.

Solution/Improvement:

* Replace these word uses with different wording.
* In self-evaluation, I will be looking for more cases of this problem.

13.

Background:

Comment:

* “What about a default and/or the typical range that needs to be supported?”

Context:

* Question about a piece of guidance stating, “the virtual keyboard shall allow user to change the scanning speed”

Location:

* 3.2.1

Significance:

* The readers of this guidance may not know what the typical ranges are, or how to determine what a large enough range would be for single-switch access.

Solution/Improvement:

* Consulted research and existing on-screen keyboards to get an idea of what the typical speeds and ranges are. Determined that 0.5 seconds to 3 seconds was the smallest range of scanning speeds that would allow the majority of single-switch users to use the system.

14.

Background:

Comment:

* “While examples are good, what does this example add? What could it add?”

Context:

* Example was, “A range of speeds is supplied in settings that the user can adjust as needed”

Location:

* 3.2.1

Significance:

* The guideline was already saying that the user should be able to change the scanning speed. This example as it is did not add much more information.

Solution/Improvement:

* Added extra information on the step size of the setting adjustment.

Note: Still not happy with this example, will revisit in self-evaluation pass.

15.

Background:

Comment:

* “Is this really a "shall" or would being a "should" be good enough? It would be useful to consider the subset of your guidance where you are using "shall" to see if they truly deserve to be a "shall" and if they are of consistent importance.”
* “Is this important enough to be a "shall"?”

Context:

* This comment is in regards to guidance that says, “The virtual keyboard shall provide functionality to re-scan a section”

Location:

* 3.4.1

Significance:

* I had been using shall only for items that were critical to allow a single-switch user access to the system. While the system not re-scanning items might result in some frustration or lost time, the virtual keyboard could be used without it.

Solution/Improvement:

* Downgraded this item to a should instead of a shall.

16.

Background:

Comment:

* “You have not discussed the design / layout / function of various levels of scan. While you referred to it in your analysis, you could do much more to analyze it and to find guidance on how levels should be used.”

Context:

* Section was discussing repetition of scan levels.

Location:

* 3.4.1

Significance:

* Scan levels are mentioned, but there isn’t any guidance on them specifically. This gets deeper into how specific scanning patterns work, which is not discussed in the document.
* Levels are completely dependent on the scanning method implemented.

Solution/Improvement:

* Provided guidance on the layout of scanning patterns in a new item 3.1.6

17.

Background:

Comment:

* “How many times do you recommend? Should the number of times be under user control?”

Context:

* Comment on an example discussing how many times the scan should be repeated.

Location:

* 3.4.1

Significance:

* Example was not clear about how many times the scan repeated, or if the user could control this. Users of this documentation may need an idea of how many times this repetition should be done.

Solution/Improvement:

* Gave a number based on the on-screen keyboards overviewed.
* Added another part to the guideline allowing the option to include customization of automatic re-scan cycles if automatic re-scanning is implemented. s

18.

Background:

Comment:

* “This is rather weak, as is indicated by the "may". And don't you really mean a button/key or are there other methods that you could provide in further examples?”

Context:

* Comment on the guideline, “The virtual keyboard may provide a method to skip scan cycles”

Location:

* 3.4.2

Significance:

* It does not specify button, but that seems to be the only way to accomplish this task with a single-switch. This could confuse the readers of the document.
* May in guidelines is weak and often is not included as it is seen as going above and beyond.

Solution/Improvement:

* Added the word button specifically to the guideline
* Looked at research about the skip scan method and the rest of the mays and shoulds in the guideline.
	+ There was only one instance of this skip scan button being used that I could find (DiscoverPro keyboard).
	+ When matched up against the rest of the shoulds this was likely the least important to the overall usability of the system. It would be a great addition, but the concept is more complicated, consumes screen space, and would increase the non-error scan time.

19.

Background:

Comment:

* “It is better to use a consistent format for all guidelines that is based on subject, action, object: The virtual keyboard shall provide a visual …”

Context:

* Comment on an inconsistently worded guideline.

Location:

* 3.5.1

Significance:

* Consistency makes the guidelines easier to read and understand.

Solution/Improvement:

* Re-word this guideline and any others that do not match the earlier ones to be consistent with their format.
* In self-evaluation look for more cases where this consistency is broken.

20.

Background:

Comment:

* “Always?”

Context:

* Comment on the guideline, “Virtual keyboard shall provide auditory indication of selection”

Location:

* 3.6.2

Significance:

* Some users might not want to hear audio indication of scanning. They might find it annoying or unnecessary.
* This needs to be included to allow users who might have reduced vision to be able to use the keyboard, however there needs to be a way to disable this.

Solution/Improvement:

* Keep the guideline, but add another guideline that indicates that there needs to be a method of disable this audio.
	+ This becomes the guideline “3.6.3 Disable audio indication of selection”

21.

Background:

Comment:

* “this only applies to keys that result in a single character”

Context:

* Comment on a guideline stating that, “visuals on keys shall match what will be input upon their selection”

Location:

* 4.1.4

Significance:

* Keys that have multiple characters input at once may be labeled with a name to describe the action, or is a symbol to represent the multiple characters. These keys should not be changed to show all the keys included in it as it may be confusing and take up too much space.

Solution/Improvement:

* Clarify in the guideline that this guideline is only for single keys, not a key that represents multiple keys.

22.

Background:

Comment:

* “Is this really a shall? Which ones? At least recommend which ones”

Context:

* Comment on guidance about providing multiple keyboard layouts.

Location:

* 4.2.1

Significance:

* Shall is a strong recommendation, where multiple layouts are not a necessity to the usability of the system.
* A reader may need help to know which keyboard layouts should be included by default. They might not have the knowledge needed on which layouts work best to be able to make those decisions themselves.

Solution/Improvement:

* Drop this guideline to a should as it is more at the level as the rest of the should’s instead of the shall’s.
* Checked my research to try and find which layouts would be best to include. This results in recommending QUERTY and frequency based.
	+ QUERTY so that non-single switch users can still comfortably use the system and those single-switch users that might already know QUERTY
	+ Frequency as it is faster to use while scanning than alphabetical

23.

Background:

Comment:

* “You have not even recommended them first. This makes them both required and their location required.

“A NOTE could be used to refer to the section on Prediction below”

Context:

* Comment on guidance for where predictive features should be placed on the keyboard.

Location:

* 4.2.5

Significance:

* Predictive features should be included not shall. The way the guideline is worded assumes that these features are there and therefore required.
* Predictive features have not been talked about yet so the reader might not know where to look for more information.

Solution/Improvement:

* Re-word the guideline to say where predictive features should be placed if they are included in the system.
* Add a note to where the section on predictive features can be found.

24.

Background:

Comment:

* “How is the developer to determine an appropriate number? Can't you suggest a number?”

Context:

* Comment on guidance for providing appropriate number of predictions.

Location:

* 6.1.2

Significance:

* Readers of the design may not have the background knowledge to make a good decision on how many characters to predict, or what appropriate means in this case. This could result in too many characters being predicted by the system.

Solution/Improvement:

* Examined research and on-screen keyboards to get an idea of how many characters are predicted by the products and papers that used it. Character prediction isn’t as common, but as all the characters are already on the screen so the characters shown should only be the top predicted ones.
	+ Less than 7 characters is recommended by me

25.

Background:

Comment:

* “what do you mean word frequency? Frequency in common English, frequency in the program, frequency by the user, recent frequency or all time frequency, etc. etc. etc.?”

Context:

* The guideline this is asking about is, “If word prediction is supplied, predictions shall be based on word frequency”.

Location:

* 6.2.2

Significance:

* Frequency is too vague and too easy to be interpreted differently by people. This will cause inconsistent systems to be created and the word prediction included to be less effective.

Solution/Improvement:

* Specify in the guideline that the predictions should be based on frequency of user use with more recently used words being first.

## Part B SelfEvaluation

### Method of Discussion

In order to evaluate my own work, I have read through the document myself coming up with challenges, opportunities, and questions about the analysis and design. I have also take two documents on planned ISO guidance for virtual keyboards and compared the ideas from those documents to my own guidance.

To show comments and changes in the document, I took the version of the analysis and design that contains the changes made that resulted from the evaluator comments, then removed the evaluator comments and accepted the changes that had been made. From this comment-free document, I went through and used tracked changes to insert my own comments inline. These comments made are the basis of the evaluation discussion presented in this evaluation document.

Using the same format for items in Part A, comments made in the analysis and design will be discussed and then changes suggested will be made to the design and analysis document (CMPT480-RevisedAnalysisandDesign-PartB). Updates resulting from this discussion will be added to the document in the colour green to ensure they can be differentiated from the tracked change comments.

### Discussion of Self-Evaluation

1.

Background:

Comment:

* “Can the user re-enable this once they have rejected it?”

Context:

* Comment on whether a user can re-enable automatic adjustments to scanning after they have rejected its use.

Location:

* 3.2.2 c

Significance:

* The user may wish to use the automatic adjustments sometime in the future. If there is no way to re-enable this feature, then it may prevent users from completing their tasks.

Solution/Improvement:

* Instead of wording it as reject, say that the virtual keyboard shall allow users to disable and enable automatic adjustment to scanning speed.

2.

Background:

Comment:

* “Can this be re-enabled?”

Context:

* Guideline suggests allowing the user to disable the short delay after input.

Location:

* 3.3.3

Significance:

* Some users might want to be able to re-enable this delay at a future time. Not specifying that the developer should include this option could result in a system that does not allow the user to re-enable the feature.

Solution/Improvement:

* Update guideline to include language specifying re-enable as well as disable.

3.

Background:

Comment:

* “Has scan cycle been defined previously?”
* “Is scan cycles defined somewhere above?”

Context:

* Guidelines mention waiting for scan cycles and skipping scan cycles.

Location:

* 3.4.1, 3.4.2

Significance:

* There is no definition or explanation of what a scan cycle is, the reader might not know what the guideline is trying to explain.

Solution/Improvement:

* Add scan cycle to the list of definitions so that the reader can look up this concept and better understand the meaning of the guidelines.

4.

Background:

Comment:

* “Colouring of keys? – Standard doc part 1”

Context:

* Section contains guidance on key appearance on the virtual keyboard. The ISO document contains an empty section 5.2.1 that would deal with the layout and colouring of the keys.

Location:

* 4.1

Significance:

* Different coloured keys can help users to recognize specific keys, and/or help them learn the locations of keys on the keyboard.

Solution/Improvement:

* Double checked the guidelines in this section and there is one for providing key appearance settings.

5.

Background:

Comment:

* “[Is this another guideline?] [If not move up to go with the guideline]”

Context:

* Commenting on a note that says, “Key order should not be changed on key size change. This can confuse those who have memorized the order of the keys, and could change the scan order”

Location:

* 4.1.1

Significance:

* Should implies that there is guidance there. Guidance should not be in a note, but instead in a spate guideline.
* This is another guideline that goes in part with the adjustment of all key sizes.

Solution/Improvement:

* Turn 4.1.1 into a multi-part guideline and make the NOTE into a full guideline.

6.

Background:

Comment:

* “[Move note up to go with the guideline not the example] [Needs can be turned into guidelines, is this another guideline instead of a note?]”

Context:

* Comment on a note that says, “Scanning system needs to be able to compensate for an item that is not the same size as others. User must be aware of how this resize will affect scanning”

Location:

* 4.1.2

Significance:

* In order for the user to be able to know how the resize will affect scanning they should be in control of how the scan pattern handles the uniquely sized key.
* The note should not be a note, and instead it should be made into a separate guideline.

Solution/Improvement:

* Create a new guideline out of the note saying, “the virtual keyboard should provide the user a method of specify how the scan pattern handles uniquely sized keys”
* This guideline will be part of 4.1.2

7.

Background:

Comment:

* “Reword in the style of the virtual keyboard”

Context:

* There are some guidelines that are not worded in a consistent format.

Location:

* Various places

Significance:

* For consistency, all the guidelines should take the same form. Guidelines marked with this type of comment do not match the format of the guidelines from earlier and need to be re-worded.

Solution/Improvement:

* Re-word guidelines to start with the virtual keyboard ….

8.

Background:

Comment:

* “[find a different word for should]”

Context:

* Some notes in the notes for guidelines use the words should, shall, or may.

Location:

* Various places

Significance:

* These words are reserved for guidelines. Having them in the notes implies that the note contains another guideline.
* Notes with this kind of comment has been determined to not be another guideline so they shouldn’t have those words in them.

Solution/Improvement:

* Replace these cases with different words.

9.

Background:

Comment:

* “[is this another guideline or is the should misleading?]”

Context:

* Comment on a note that says, “With no modifiers currently active keys should appear in their default state (likely this would be lower case characters)”

Location:

* 4.1.4

Significance:

* The should implies a guideline, but the example just illustrates what the keys would look like if there was no modifier key active. This means that it is not a guideline and shouldn’t use the word should.

Solution/Improvement:

* Change the wording to remove the use of the ‘should’.

10.

Background:

Comment:

* “Can the user get layouts from elsewhere? Should they be able to share layouts?”

Context:

* Under the section keyboard layout. The idea of the comment comes from the ISO document on virtual keyboards part 2 (6.3.5.6- load-save and-share the customized keyboard).

Location:

* 4.2

Significance:

* There is guidance given on being able to customize layouts and switch layouts, but no guidance on being able to share the customized keyboard with others.
* The sharing is important so some users can customize keyboards for certain tasks and other users can use those keyboards without having to do all the customization themselves.

Solution/Improvement:

* Update guidelines 4.2.4 to include more than just the customization step. This will include customizing, saving, loading, and sharing.

11.

Background:

Comment:

* “If it is done with a key on the keyboard then this key should always be in the same place – iso doc part 1”

Context:

* Comment on a guideline that says, “System should provide a way for user to switch between keyboard layouts as needed”
* 5.9.3 Switching between keyboards - On on-screen keyboards, the toggle key for switching from one layout to another shall always be visible and located at the same place on the user interface.

Location:

* 4.2.2

Significance:

* The user needs to always be able to find the button that lets them switch key layouts regardless of the layout the are currently on. If the button always moves with the layout, then the user may not be able to find it.

Solution/Improvement:

* Expand guideline to include a part about ensuring that the change layout key is always located in the same place.

12.

Background:

Comment:

* “[Should implies another guideline.]”

Context:

* Comment is on the first part of a note that says, “This frequency based layout should be a static layout”. This is part of a guideline that says the keyboard should supply a frequency based layout.

Location:

* 4.2.3

Significance:

* Should implies another guideline, but in this case I don’t find the note to be enough to create a whole new guideline out of. It is however, something that needs to be done and should go as part of a guideline.

Solution/Improvement:

* Instead of making into a whole guideline on its own, I’ve added it directly into the guideline for provide frequency based layout.
	+ The virtual keyboard should provide a frequency based layout. -> The virtual keyboard should provide a static frequency based layout.

13.

Background:

Comment:

“- Make other function keys part of this? – iso doc part 2

 - navigation keys?

 - home/end/etc keys”

Context:

* This was a comment on the section called Macros and Modifiers. The ideas come from part 2 of the ISO document on virtual keyboard guidance (6.3.6).

Location:

* 5.1

Significance:

* Navigation keys have not been mentioned in this analysis and design. This means that there is no guidance on if there should be navigation keys on the keyboard or not. Some users may need to use those keys and therefore they should be available.

Solution/Improvement:

* Updated the Macros and Modifiers section to include Navigation
* Created a guideline that indicates that the virtual keyboard shall allow all navigation keys to be present on the virtual keyboard.
* Updated the definitions to include a definition of navigation keys.

14.

Background:

Comment:

* “[is this another guideline or should the should be a different word?]”*.*

Context:

* Comment on a note that says, “If virtual keyboard can be customized then user can remove these keys if they wish, but it should be the default to include these modifier keys.”

Location:

* 5.1.1

Significance:

* The note uses the word should which implies guidance. This note could be re-worded into a piece of further guidance on the default keys on the virtual keyboard

Solution/Improvement:

* Broke the guidance 5.1.1 into multiple parts.
	+ First part – The virtual keyboard shall, by default, present all modifier keys to be present on the virtual keyboard.
	+ Second – The virtual keyboard should allow individualization function to remove these keys from the virtual keyboard.

15.

Background:

Comment:

* “[Should. Another guideline?]”

Context:

* Comment is on a note that says, “User should be able to store a sequence of modifier keys, character keys, and combinations of both modifiers and characters in a macro”

Location:

* 5.2.1

Significance:

* Should implies guidance, but it is being used in a note.
* This note can be turned into further guidance. In addition to being able to add macros, the macros need to be able to contain more than just characters in order to be able to perform complex actions with a few key presses.

Solution/Improvement:

* Broke the ability to add macros guidance down. Turned the note into another piece of guidance to go along with the subsection.

16.

Background:

Comment:

* “Should there be an option to turn off character prediction?”

Context:

* This is a comment that falls under the guidance on character prediction.

Location:

* 6.1

Significance:

* Not every user will be able to, or willing to, use character prediction. Some might find it unnecessary, or getting in the way of the word prediction. These users might wish to be able to turn off character prediction.
* This is a guideline that is not covered in the existing ones.

Solution/Improvement:

* Created a new guideline that specifies that the virtual keyboard should provide functionality to turn character prediction on and off.

17.

Background:

Comment:

* “Self-learning word prediction? – iso doc part 2”

Context:

* Part 2 of the iso virtual keyboard document has an empty section for self-learning word prediction (6.3.7.3).

Location:

* 6.2

Significance:

* Current design only includes guidance on doing word prediction based on frequency and adding words to a dictionary.
* There could be additional guidance indicating that adding words to the dictionary could be done either by the user or by the system.

Solution/Improvement:

* Updated the ability to add to dictionary guideline to include “If word prediction is provided, the virtual keyboard may provide a method for the dictionary to learn words the user uses”

18.

Background:

Comment:

* “How many predictions is appropriate?”
* “How many is too many?”

Context:

* These comments fall under guideline 6.2.4 use appropriate number of predictions.

Location:

* 6.2.4

Significance:

* The reader may not have the knowledge needed to make an educated decision about how many words should be predicted at a time. Instead of forcing the user to make this decision the document can give a suggestion about how many predictions to include.

Solution/Improvement:

* Went back to my research to determine how many words tend to be predicted and how what affects the ability to use them. From this I determined that the most that should be included is around 7. I have updated the guidance in this section to reflect this.

19.

Background:

Comment:

* “Is this something that a user should be able to turn off if they wish?”

Context:

* This comment is on a section providing guidance on adding space after selected prediction.

Location:

* 6.2.5

Significance:

* Not every use might be able to understand how the automatic space after a word works, or they may wish to have the option of not using that space. This means that there should be a way from them to disable this feature.

Solution/Improvement:

* Broke 6.2.5 down into multiple pieces of guidance to include, “the virtual keyboard should provide setting allowing use to turn off the space after the is predicted”

## Part C Proof by Examples

For creating examples, I went through each guideline and tried to come up with one or more ways that the guideline could be implemented within the virtual keyboard. For the guidelines that were not strictly implementation guides, I tried to come up with an example that would illustrate how the guideline would assist the virtual keyboard in helping single-switch users accomplish tasks with the virtual keyboard.

 As I created examples I made note of any problems that could be created by the examples or thoughts that were raised while creating them. I included these in the document for part C (CMPT-480RevisedAnalysisandDesignPartC.docx). Examples have been added in green, comments on examples are in red.

 Using these comments and the format for presenting comments from previous sections I worked through each question raised, determining what a good solution to this would be, the changes that were made to the document to reflect these decisions, and a new example to prove the changes assist in resolving the issue.

 Updates to the document that resulted from the discussion on the comments on examples are made in blue.

### Discussion of Examples

1.

Background:

Comment:

* “Are there other ways that this layout change could be done? Should there be a guideline about not making the user cycle through layouts?”

Context:

* Comment on an example of the ways a user could change the layout of the keyboard.

Location:

* 4.2.2

Significance:

* Having to cycle through every layout attempting to get to the one that the user wants would be very time consuming, and could fatigue the user. To prevent this the key provided to switch between layouts should not just go to the next layout.

Solution/Improvement:

* Created another part to the guideline 4.2.2 to take into account that the user should never have to one, by, one move through layouts.

2.

Background:

Comment:

* “This space should be removed if there is punctuation added after.”

Context:

* Comment on the virtual keyboard always adding a space after the selection of a predicted word.

Location:

* 6.2.5

Significance:

* The addition of a space when the user wishes to place punctuation next would result in the user having to erase the space first and then add the punctuation. This slows the efficiency of the user down and could cause fatigue faster.

Solution/Improvement:

* Add another guideline to the subsection indicating that if the next item entered is a piece of punctuation the previously added space is removed.

3.

Background:

Comment:

* *“*Help should provide instructions on how to navigate to ensure that the keyboard can be used to navigate.”

Context:

* The example said, “User can use the scanning on the virtual keyboard to input the keys required to navigate through the help sections”

Location:

* 7.1.2

Significance:

* If the user does not know how to navigate the help sections, then what use will the help be. In order to make the help accessible the help will need to provide some sort of guide to inform users on what keys are used to navigate.

Solution/Improvement:

* Add an additional guideline to ensure the help will provide guidance on how to navigate help.

The creation of examples did not raise as many questions or issues as I had expected. This section of the evaluation is therefore fairly short. The majority of the work in this part of the evaluation is documented in the revised anaylsis and design document itself, where all the small examples were written down alongside the guideline they corresponded to.

# Bibliography

|  |  |
| --- | --- |
| [1]  | H. H. Koester and R. C. Simpson, "Method for enhancing text entry rate with single-switch scanning," *JRRD,* vol. 51, no. 6, pp. 995-1012, 2014.  |
| [2]  | C. E. Steriadis and P. Constantinou, "Designing human-computer interfaces for quadriplegic people," *ACM Transactions on Computer-Human Interaction (TOCHI) ,* vol. 10, no. 2, pp. 87-118, 2003.  |
| [3]  | P. E. Jones, "Virtual keyboard with scanning and augmented by prediction," *The 2nd European Conference on Disability, Virtual Reality and Associated Technologies, Proceedings,* vol. 1, pp. 45-51, 1998.  |
| [4]  | I. Fishman, Electonic Communication Aids, Boston: College-Hill Press, 1987.  |
| [5]  | J. Treviranus and R. Tannock, "A Scanning Computer Access System for Children With Severe Physical Disabilities," *American Journal of Occupational Therapy,* vol. 41, no. 11, pp. 733-738, 1987.  |
| [6]  | S. Trewin, "Automating Accessibility: The Dynamic Keyboard," in *The Sixth International ACM SIGACCESS Conference on Computers and Accessibility*, New York, The Association for Computing Machinery, 2004, pp. 71-78. |
| [7]  | Lake Software, "Click-N-Type," Lake Software, 19 March 2013. [Online]. Available: http://cnt.lakefolks.com/. [Accessed 15 October 2016]. |
| [8]  | Strategic Transitions, "WiViK (Windows Virtual Keyboard)," [Online]. Available: http://www.goqsoftware.com/wivik.php. [Accessed 1 November 2016]. |
| [9]  | *DiscoverPro Overview Video.* [Film]. Madentec Limited, 2007.  |
| [10]  | G. F. Dalrymple, "My Head Controls My Computer: Computer Access With Cerebral Palsy," *IEEE Technology and Society Magazine,* vol. 4, no. 3, pp. 7-10, 1985.  |
| [11]  | K. S. Lee and D. J. Thomas, Control of Computer-Based Technology for People with Physical Disabilities: An Assessment Manual, Toronto: University of Toronto Press, 1990.  |
| [12]  | G. K. Thompson, "Alternative Access Methods For Accessing AAC Systems For Individuals With Severe Physical Impairment," University of Texas at El Paso, El Paso, 2007. |
| [13]  | J. Angelo, "Factors affecting the use of a single switch with assistive technology devices," *Journal of Rehabilitation Research and Development,* vol. 37, no. 6, pp. 591-598, 2000.  |
| [14]  | S. A. López, F. Corno and L. D. Russis, "Playable one-switch video games for children with severe motor disabilities based on GNomon," in *2015 7th International Conference on Intelligent Technologies for Interactive Entertainment (INTETAIN)*, 2015.  |
| [15]  | T. Broderick and D. J. C. MacKay, "Fast and Flexible Selection with a Single Switch," *PLoS One,* vol. 4, no. 10, 2009.  |
| [16]  | J. J. Lazzaro, Adaptive Technologies for Learning & Work Environments, Chicago: American Library Association, 2001.  |
| [17]  | MiguelC, "Make popup of the key pressed in a customized keyboard," 18 September 2014. [Online]. Available: http://stackoverflow.com/questions/25911409/make-popup-of-the-key-pressed-in-a-customized-keyboard. [Accessed 29 October 2016]. |