

Four on-screen keyboards compared

Citation for published version (APA):

Smeets, S. (1996). *Four on-screen keyboards compared: does keyboard-layout matter?* (IPO rapport; Vol. 1105). Instituut voor Perceptie Onderzoek (IPO).

Document status and date:

Published: 23/05/1996

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

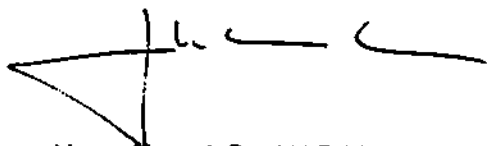
openaccess@tue.nl

providing details and we will investigate your claim.

Rapport no. 1105

**Four On-Screen
Keyboards Compared
Does Keyboard-layout matter?**

Sjef Smeets

A handwritten signature in black ink, consisting of a stylized 'J' followed by several loops and a long horizontal stroke.

Voor akkoord: Dr. J.H.D.M. Westerink

FOUR

On-screen Keyboards

Compared

*Does
Keyboard-layout
Matter ?*

Author Sief Smeets
Interaction Design
Utrecht School of the Arts
Faculty of Art, Media & Technology

Supervisor David Ziedman
Philips Media Benelux

Supervisor Joyce Westerink
Institute for Perception Research

Summary

This report deals with a study on the 'lanthe-Keyboard', an on-screen keyboard-design newly developed for the CD-i environment. It is contrasted to three commonly used keyboard-designs in the fields of layout, use of restricted areas and magnetic feel. The 'lanthe-Keyboard' is designed according to the guidelines described in an earlier report¹ and forms a basis for further research and usability testing. Furthermore, some additional ideas on design and functionality are described in the appendices.

¹ 'Guidelines for Designing On-Screen Keyboards' by Sjet Smeets, IPO report no. 1076

Contents

Summary	1
Contents	3
1. Introduction	5
2. Task Analysis	7
2.1 Overview of User Tasks	8
2.2 Information-Search	10
3. Design Specifications	13
3.1 Basic Functionality	14
3.2 A Proposed Design: the lanthe-Keyboard	18
3.3 Restricted Cursor Movement	19
4. Testing Variables	21
4.1 The Layouts	17
4.2 Magnetism	20
4.3 Restricted Cursor Movement	21
4.4 Types of Feedback	22
5. Expected Results	29
5.1 What Users Should Prefer... ..	30
5.2 The Test-Probabilities	31
5.3 What Users Do Prefer... ..	34
6. Appendices	37
6.A Additional Functionality	39
6.B Additional Keyboard-Designs	43
6.C Additional Possibilities	47
6.D The Pictures	49
7. References	61

1. Introduction

This report is dedicated to the design and testing of on-screen keyboards and to one keyboard in particular: the 'lanthe-keyboard'². From all the possible uses for keyboards it is mainly directed to the more 'serious' use as for Encyclopaediae, Dictionaries, Thesauri et cetera, which means in repetitive used situations, requiring ease of use and facilitation of speed.

The report therefore is written with serious use in mind, and in regard of the CD-I title³ in which it was intended to be tested. I tried to maintain consistency within the overall interface of the application, and especially with the other keyboards to which the lanthe-keyboard will be compared.

Chapter 2. 'Task Analysis' as well as Chapter 3. "Design Specifications' are written with this in mind. Chapter 3 deals with the specific design-factors of the new layout, while important factors for testing the lanthe-keyboard against other keyboards are described in Chapter 4. 'Testing Variables'. Chapter 5. 'Expected Results' provides my predictions for the outcome of the tests.

Several Appendici are included, the first dealing with some, to my knowledge yet-unused, extended functionalities of the keyboard (see Appendix A.: 'Additional Functionalities'), in addition to the lanthe-keyboard some –maybe less serious but not necessarily less usable– design-ideas (see Appendix B.: 'Additional Keyboard-Designs'), a plea concerning the use of games (see Appendix C.: 'Additional Possibilities') and finally some large-scale pictures of the designs mentioned in this report (Appendix D.: 'The Pictures').

★ ★ ★

² This report 'Four On-Screen Keyboards Compared', in fact is a successor to my report 'Guidelines for Designing On-Screen Keyboards', the complete References for both reports are to be found at the end of the second report.

³ The application which was aimed at for testing was the 'Philips Media Interactieve Encyclopedie', an electronic encyclopaedia in Dutch, in development for the Dutch and the Belgian (Flemish) market.

2. Task Analysis

2.1 Overview of User Tasks	8
2.1.1 Information-Search	8
2.1.2 Marking of Found Information	9
2.1.3 Exploring ('Browsing')	9
2.1.4 'Showing Off'	9
2.1.5 'Playing of a Game'	9
2.2 Information-Search	10
2.2.1 Task Analysis on Title-Search	10

2. Task Analysis

2.1 Overview of User Tasks

The following overview is based on use of encyclopaedia-like media, mostly directed towards the use of an electronic interactive encyclopaedia in the CD-i environment. Most tasks nevertheless do have their paper counterpart as we often observe in the names and icons for certain tasks (for instance: 'bookmarks')

There probably are quite a lot of target users with sometimes very different goals, but a lot of these goals will overlap. A list is provided of the main goals, each with a sub-section of the different goals within each heading, and each followed by a result (indicated by '->' and printed in *italics*). Since this task analysis is directed towards the use of a keyboard, I did not elaborate tasks that would not primarily involve the use of a keyboard. Keyboard-related tasks are marked with two bullets (••).

The most important task would probably be 'Information Search'. This can be divided into several types, depending on the user, the user's goal, and the possibilities of the application. More tasks could be imaginable as for instance several types of games, setting up a kind of user-defined slide-shows, et cetera, but these are not within the scope of the application used for testing and furthermore they are not primarily keyboard-related. (see also Appendix A: 'Additional Functionality')

Use of Encyclopaedia

2.1.1 Information-Search

- 1. Search by a Keyword
 - > *(List of) Article(s)*
- 2. Search by a Subject *(This should be one of the keywords, of course)*
 - > *(List of) Article(s)*
- 3. Search by a Title
 - > *(List of) Article(s)*
- 4. Search by a Name (a Person, an Object, a ...)
 - > *(List of) Article(s), List of Names (Name-index)*
- 5. Search by a Date (a Year, a Period)²
 - > *(List of) Article(s), List of Dates (Date-index)*
- 6. Search for a Synonym 'Thesaurus-function'

¹ Here the user is provided with a list of articles that have the required word in them. The order of the articles could be based on the number of times the word is used in the article. See also Appendix A: 'Additional Functionality'.

² This is not available in the CD-i title used for testing.

- > *(List of) Synonym(s)*
- 7. Search for an Explanation 'Dictionary-function'
 - > *(List of) Article(s)*
- 8. Search for previously found information ('Data-recovery')
 - 'Bookmark-function'
 - > *(List of) Title(s), Picture(s), Sound(s), Mark(s)*

2.1.2 Marking of Found Information³

- 1. Compose set(s) of Marks.
 - > *'a kind of Slide-Show'*
- 2. Name a set of Marks.⁴
 - > *List of Sets*

2.1.3 Exploring ('Browsing')

- 1. Hyper-linking (et cetera)
 - > *Probably no use of a Keyboard*
- 2. Marking of found information. 'Bookmark-function'

2.1.4 'Showing Off'

- 1. - *Keyboard-use as in 'Information-Search'*

2.1.5 'Playing of a Game'⁵

- 1 - *Probably no use of a Keyboard.*

(This, of course, would depend on the type of game. A game for instance could be a 'Trivial Pursuit-like' game which would not need a keyboard but could be a multiple-choice game⁶; a Synonym-type of game could probably better be played with a keyboard, in order to provide a word-input.)

(See also Appendix C. 'Additional Possibilities')

- 2. Assistance to a game
 - *High Score name entry.*
 - *Determination of a 'random' character*

* * *

³ This could be of great use for educational purposes. In this case it should be possible to compose marks to a set with a user-defined order. See also Appendix A: 'Additional Functionality'.

⁴ This only is necessary when it is possible to compose more than two sets.

⁵ This is not (yet) available in the CD-i title used for testing.

⁶ A Keyboard could be used for choices 'A', 'B' or 'C', but it is preferable to select the choice directly.

2.2 Information-Search

Since this appears to be the main goal in general (in 'normal use'), and totally within the scope of the application, this task is elaborated. Another main reason of course is the involvement of the keyboard in this task.

Although 'Information-search' can be divided into several types, the task-analysis –as far as the use of a keyboard is concerned– would be quite the same for each of these. Therefore only one task, a 'Search for a Subject' ('Title-Search'), is elaborated. The task analysis is given from the point when 'A tot Z' is selected either from the main screen, or through selecting 'Return'⁷.

2.1.2 Task Analysis on Title-Search

0. Go to 'Alphabetical-Search'
 - 0.1 Select the screen to put in a text-entry
 - 0.1.1 Move the cursor to the wanted Button (select 'A tot Z')⁸
 - 0.1.2 Press Button One⁹
1. Go to Character 'X' ('X' running from 1 to 10, for instance)
 - 1.1 Select the character from the keyboard
 - 1.1.1 Move the cursor to the wanted character
 - 1.1.2 Press Button One⁸
2. Check 'Contents' or Go to Character 'X' (repeat 1.)
 - 2.1 Check the whole entry.
 - 2.1.1 Look at the entry whether it meets the expectations
 - 2.2 Check the last Character of the entry.
 - 2.2.1 Look at the last Character of the entry whether it meets the expectations
 - 2.3 Check the list.
 - 2.3.1 Look at the list whether it meets the expectations
 - 2.4 Delete a (part of) the entry
 - 2.4.1 Select 'Backspace'
 - 2.4.2 Select 'Delete'
 - 2.5 Go to the Main screen
3. Go to the article
 - 3.1 Check the list for (a) wanted article(s)
 - 3.1.1 Scroll through the list
 - 3.2 Select an article
 - 3.2.1 Move the cursor to the wanted article

⁷ The analysis assumes that, when entering this screen, the textfield is blank, otherwise each time the task would start with either a Delete or a Backspace-action. The actual situation will depend on the conventions of the CD-i title.

⁸ Conventions of the original CD-i title, it also may be executed by pressing 'Return' in case the user already had chosen 'A tot Z' before.

⁹ Conventions of the CD-i title used in the tests.

3.2.2 Press Button One¹⁰.

¹⁰ Conventions of the CD-i title used in the tests.

3. Design Specifications

3.1 Basic Functionality	14
3.1.1 Use of the Keyboard	14
1. Selecting a character	14
2. The Backspace-Button	15
3. The Delete-Button	16
3.1.2 Use of the List	17
1. Selecting a Subject from the List	17
2. Scrolling Up or Down	17
3.2 A Proposed Design: the lanthe-Keyboard .	18
3.3 Restricted Cursor Movement	19

3. Design Specifications

3.1 Basic Functionality

The design-specifications that will be described are based according to the guidelines in the report 'Guidelines for Designing on-Screen Keyboards'¹.

When the user has chosen 'A tot Z' in the Encyclopaedia CD-i title (see also Appendix D. 'The Pictures') he or she is shown a screen with a keyboard for input, a textfield to show the typed entry (if any), a list showing the titles (the subjects) with the accompanying scroll-buttons and, at the bottom, –eventually– a row of buttons for the general functions (see also Appendix D. 'The Pictures').

3.1.1 Use of the Keyboard

The keyboard is shown after selecting 'A tot Z'. It is used to put in a text-entry, to get a list of subjects to choose from. This can be done by selecting the characters, which make up the name of a subject, one after another.

1. Selecting a character

- 1.1 Move the cursor to the keyboard
 - 1.1.1 The cursor changes to a cross²
- 1.2 Move the cursor to the wanted key
 - 1.2.1 The key the cursor is 'on' changes in colour. (lighter than 'normal')
 - 1.2.2 The key shifts up and to the left (1 or 2 pixels)
 - 1.2.3 The key is surrounded with a rectangle. Top and left are light-, bottom and right are dark-coloured. (thickness 2 or 3 pixels)
- 1.3 'Click' (Press Button One)³
 - 1.3.1 The key the cursor is 'on' changes in colour (darker than 'normal')
 - 1.3.2 A 'Click-sound' is audible
 - 1.3.3 The key shifts down and to the right (2 or 4 pixels)
 - 1.3.4 The key is surrounded with a rectangle. Top and left are dark-, bottom and right are light-coloured. (thickness 2 or 3 pixels)
 - 1.3.5 The character appears in the Textfield
 - 1.3.6 The List is changed (or: 'appears'⁴) starting with the entry typed so far

1.4 Release Button One

¹ IPO-report no. 1076

² This only is necessary by absence of a 'Magnetic Feel'.

³ Button Two provides the user with the 'General Functions', conventions of the original CD-i title.

⁴ According to the conventions of the CD-i title used for testing the list is 'blank' on the start.

- 1.4.1 The key the cursor is 'on' changes in colour (lighter than 'normal')
 - 1.4.2 The key shifts up and to the left (2 or 4 pixels)
 - 1.4.3 The key is surrounded with a rectangle. Top and left are light-, bottom and right are dark-coloured. (thickness 2 or 3 pixels)
- *When Button One is not released within 'about a third of a second', it should be regarded as a next pressing of Button One⁵, this includes all steps. (Also a next click-sound)*
 - *To put a space in the entry the user can select the 'Space-Key' and click (press Button One). The same 'Click-sound' is used here.*
 - *The list shows all subjects that are available in the part of the list that starts with the entry typed so far. The subjects are shown in an alphabetical order.*
 - *When the user selects a character-combination that does not match any entry, a list will still be shown. This will be the last possible list of selectable subjects⁶ but in a different colour, to provide feedback to the user that no matching items are found, however the shown items still may be selected.*

★ ★ ★

2 The Backspace-Button

If an entry is misspelled or not available, either the last typed character(s) can be deleted, or the whole entry. To delete the last character(s) the Backspace-Button ('<=&') is used.

- 2.1 Move the cursor to the keyboard
(Assumed the cursor is not yet on the keyboard)
 - 2.1.1 The cursor changes to a cross⁷
- 2.2 Move the cursor to the 'Backspace-Key'⁸
(to delete the last character(s))
 - 2.2.1 The Backspace-Key changes in colour, lighter than 'normal'
 - 2.2.2 The Backspace-Key shifts up and to the left (1 or 2 pixels)
 - 2.2.3 The Backspace-Key is surrounded with a rectangle. Top and left are light-, bottom and right are dark-coloured. (thickness 2 or 3 pixels)
- 2.3 'Click' (Press Button One)
 - 2.3.1 The Backspace-Key changes in colour, darker than 'normal'
 - 2.3.2 A 'Backspace-sound' is audible
 - 2.3.3 The Backspace-Key shifts down and to the right. (2 or 4 pixels)
 - 2.3.4 The Backspace-Key is surrounded with a rectangle. Top and left are dark-, bottom and right are light-coloured. (thickness 2 or 3 pixels)
 - 2.3.5 The last character disappears from the Textfield
 - 2.3.6 The List is changed starting with the now available entry
- 2.4 Release Button One
 - 2.4.1 The Backspace-Key changes in colour (lighter than 'normal')
 - 2.4.2 The Backspace-Key shifts up and to the left (2 or 4 pixels)
 - 2.4.3 The Backspace-Key is surrounded with a rectangle. Top and left are light-, bottom and right are dark-coloured. (thickness 2 or 3 pixels)

⁵ This should be done according to the conventions of the original CD-i title.

⁶ This would probably be the list based on the current entry, minus the last selected character(s)

⁷ This only is necessary by absence of 'Magnetism'.

⁸ Extra feedback could be provided by showing the name of the function in its selectable state.

- *When Button One is not released within 'about a third of a second', it should be regarded as a next pressing of Button One⁹, this includes all steps. (Also a next Backspace-sound)*
- *When the textfield is blank through deleting the first character the list also is 'blank'*

★ ★ ★

3. The Delete-Button

If an entry is misspelled or not available, either the last typed character(s) can be deleted, or the whole entry. To delete the whole entry at once, the Delete-Button ('X') is used.

- 3.1 Move the cursor to the keyboard
(Assumed the cursor is not yet on the keyboard)
 - 3.1.1 The cursor changes to a cross¹⁰
- 3.2 Move the cursor to the 'Delete-Key' (to delete the whole entry at once)
 - 3.2.1 The Delete-Key changes in colour¹¹, lighter than 'normal'
 - 3.2.2 The Delete-Key shifts up and to the left (1 or 2 pixels)
 - 3.2.3 The Delete-Key is surrounded with a rectangle. Top and left are light-, bottom and right are dark-coloured. (thickness 2 or 3 pixels)
- 3.3 'Click' (Press Button One)
 - 3.3.1 The Delete-Key changes in colour, darker than 'normal'
 - 3.3.2 A 'Delete-sound' is audible
 - 3.3.3 The Delete-Key shifts down and to the right. (2 or 4 pixels)
 - 3.3.4 The Delete-Key is surrounded with a rectangle. Top and left are dark-, bottom and right are light-coloured. (thickness 2 or 3 pixels)
 - 3.3.5 All characters disappear from the Textfield (made 'blank')
 - 3.3.6 The List is changed in colour ('greyed') starting with the last available entry
- 3.4 Release Button One
 - 3.4.1 The Delete-Key changes in colour¹², lighter than 'normal'
 - 3.4.2 The Delete-Key shifts up and to the left (2 or 4 pixels)
 - 3.4.3 The Delete-Key is surrounded with a rectangle. Top and left are light-, bottom and right are dark-coloured. (thickness 2 or 3 pixels)

- *When Button One is not released within 'about a third of a second', it should be regarded as a next pressing of Button One⁹, this includes all steps. (Also a next click-sound)*
- *There is an important difference between the Backspace-Key and the Delete-Key. When the textfield is blank through use of the Backspace-Key the list also is 'blank'. When the Delete-Key is used¹² the list is 'greyed' but still visible and available, in case of accidental use. The user could either 'undo' his or her action, or still select an item from the list.*

★ ★ ★

⁹ This should be done according to the conventions of the original CD-i title.

¹⁰ This only is necessary by absence of 'Magnetism'.

¹¹ Extra feedback could be provided by showing the name of the function in its selectable state.

¹² This could be extended by –temporarily– changing it into an 'Undo-Delete Button'. (see also Appendix A. 'Additional Functionality')

3.1.2 Use of the List

1. Selecting a Subject from the List

Each subject is displayed by its name. Assumed is that the name of the wanted subject is visible in the list, otherwise see: 'Scrolling Up or Down'.

1.1 Move the cursor to the List

1.1.1 The cursor changes (back) to an arrow¹³

1.2 Move the cursor to the line of the wanted subject

1.2.1 The name of the subject changes in colour¹⁴

1.3 'Click' (Press Button One)

1.3.1 The name of the subject changes in colour¹⁴ (inverted ?)

1.3.2 A 'Click-sound' is audible

1.4 Release Button One

1.4.1 The cursor changes to a watch¹⁵

1.4.2 Feedback is provided that the application is 'working'

1.4.3 Within ± 2 seconds a new screen is shown with the wanted article

- *When a list is 'greyed' because of a typed entry that is not available, this has to be considered as a visible notice only. All functions regarding the list, as scrolling and selecting, do still function.*

* * *

2. Scrolling Up or Down

In the list, each subject is displayed by its name¹⁶. Assumed is that a list already is available.

2.1 Move the cursor to one of the Scroll-Buttons

2.1.1 The cursor changes (back) to an arrow¹³

2.1.2 The Scroll-Button changes in colour, lighter than 'normal'

2.1.3 The Scroll-Button shifts up and to the left (1 or 2 pixels)

2.1.4 The Scroll-Button is surrounded with a rectangle. Top and left are light-, bottom and right are dark-coloured. (thickness 2 or 3 pixels)

2.2 'Click' (Press Button One) to scroll by line

2.2.1 The Scroll-Button changes in colour, darker than 'normal'

2.2.2 A 'Click-sound' is audible (?)

2.2.3 The Scroll-Button shifts down and to the right (2 or 4 pixels)

¹³ The shape of the cursor is according to the conventions of the original CD-i title.

¹⁴ This is according to the conventions of the original CD-i title, other possibilities include change of background-colour, a rectangle surrounding the selectable line or a combination of these (see also Appendix A: 'Additional Functionality').

¹⁵ This should be done according to the conventions of the original CD-i title.

¹⁶ This could be done together with (possible) additional information, depending on the conventions of the program. Additional information may be the presence of picture(s), sound(s), movie(s), et cetera (see also Appendix A: 'Additional Functionality').

2.2.4 The Scroll-Button is surrounded with a rectangle. Top and left are dark-, bottom and right are light-coloured. (thickness 2 or 3 pixels)

2.2.5 The List is changed (moved Up or Down, according to the chosen Scroll-Button)

- *If the Down-Button is chosen the list is moved up one line. If the Up-Button is chosen the list is moved down one line.*

2.3 Release Button One

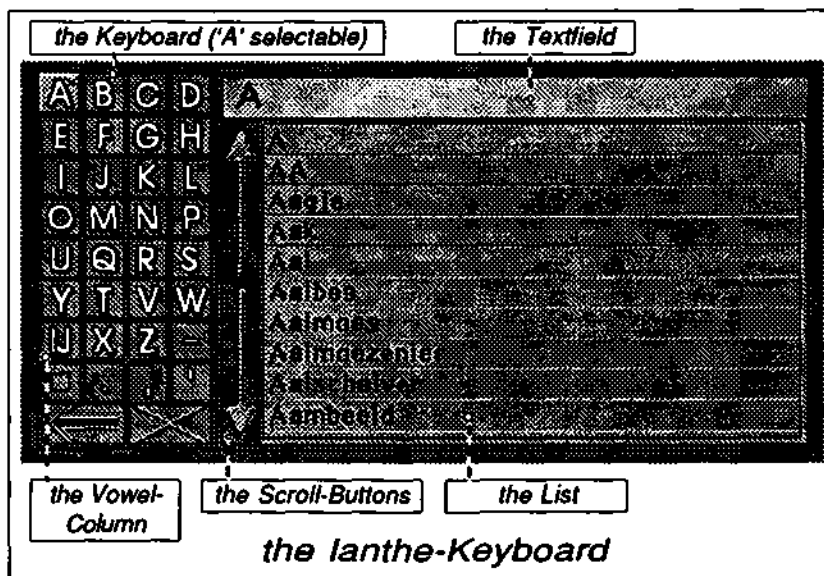
- 2.3.1 The Scroll-Button changes in colour, lighter than 'normal'
- 2.3.2 The Scroll-Button shifts up and to the left (2 or 4 pixels)
- 2.3.3 The Scroll-Button is surrounded with a rectangle. Top and left are light-, bottom and right are dark-coloured. (thickness 2 or 3 pixels)

- When Button One is not released within 'about a third of a second', it should be regarded as a next pressing of Button One¹⁷, this includes all steps.
- When Button One is being held down longer than 'a certain amount of time¹⁸', the scrolling should speed up.
- When Button Two is pressed, the list should scroll by 'Page minus one', meaning that the last visible item of the list becomes the first (when scrolling Up) and vice versa (when scrolling Down)

* * *

3.2 A Proposed Design: the lanthe-Keyboard

The 'lanthe-keyboard' is a 4-Column keyboard (or a 7-line keyboard, as far as the characters are concerned) in which the characters are arranged in such a way that the vowels all are in one column (on the far left). This means they can profit from the restricted area on the left side. This restriction should be extended to the top of the keyboard.



In the picture of the 'lanthe-Keyboard shown here (which is slightly different from the layout as used for testing), the font I propose to use on the keyboard is 'Avant Garde'. The font I propose to use in the list and the textfield is 'Geneva-Bold'.

It will depend on the availability of time and money if these will be used in the tests, because it may mean a lot of changes in the original interface.

(The layout shown here differs from the one used for testing in having a key with the cha-

¹⁷ This should be done according to the conventions of the original CD-i title.

¹⁸ The exact amount of time should be the outcome from user-testing.

racter 'IJ', which is an independent character in the Dutch language. Since this keyboard-layout originally was meant to be used in an Encyclopaedia for the Dutch market, I proposed to provide the character with its own key. For testing purposes the layout has been changed to match the original design in order to create an equal test-environment. This picture is from a prototype without any real database.)

* * *

3.3 Restricted Cursor Movement

This is the most important for the 'lanthe-Keyboard' since the use of restricted areas was one of the principles that led to this layout. It could be useful for the 4-column keyboard too, but –like the 2-line keyboard– this could possibly benefit more from a certain level of 'Magnetism' especially in creating a restricted feel.

- The movement of the cursor, as far as the keyboard is concerned is restricted at the left-side of the 'lanthe-Keyboard', and the top of it, meaning that on those sides the cursor cannot be moved outside these areas.
- This restriction in some way can be simulated by use of a restricted feel, meaning that the keyboard is surrounded by a level of magnetism, making it harder for the cursor to 'break out' involuntarily (or unintentional).
- In case of use of a Magnetic Feel, the 'Magnetic Center' of each key on the keyboard is just below the selectable character, so preventing the cursor from blocking the view on the selectable character as well as the character(s) below.

* * *

4. Testing Variables

4.1 The Layouts	21
1 The Ianthe-Keyboard	22
2 The 2-Line Keyboards	22
3 The 4-Column Alphabetical Keyboard	23
4 The QWERTY-Keyboard	23
5 Positioning the Scroll-Buttons	23
4.2 Magnetism	24
1. No Magnetic Feel	24
1.1 With Change of Cursor	24
1.2 No Change of Cursor	24
2. A Light Magnetic Feel	24
3. A Strong ('Heavy') Magnetic Feel	25
4.3 Restricted Cursor Movement	25
1. No Restricted Areas	25
2. Restricted Feel (Restricted Surroundings)	26
3. Restricted Keyboard	26
4.4 Types of Feedback	27
1. Simple Feedback	28
2. Extended or Multiple Feedback	28

4. Testing Variables

As I stated before, several aspects will have to be tested. First of all several layouts will have to be compared, so the newly developed lanthe-Keyboard will be tested against several commonly used keyboard-designs. The differences in the layout are small between the two 4-Column Keyboards, being one having an Alphabetical layout, the other (the 'lanthe-Keyboard') having a Vowel-Column. In regard to the Alphabetical order the 2-Line keyboard and the –regular– 4-Column keyboard are the same. An exception is the QWERTY-Keyboard.

Some other aspects in connection with these layouts will have to be taken into account. 'Traditionally' Scroll-Buttons are positioned at the right side of a list. Since I propose to position them between the keyboard and the list, they are moved to the left side of the list in case of the 4-Column Keyboards. In case of the 2-Line keyboard as well as the QWERTY-keyboard this means the scroll-buttons are horizontal. For testing purposes horizontal Scroll-Buttons should also be compared with vertical Scroll-Buttons within the same keyboard-layout. This is tested with the 2-Line Keyboard.

Furthermore, one of the most important aspects of design may be the use of restricted areas (a restriction in cursor-movement) so this should also be put to the test. Another enhancement could well be the adding of a certain level of 'Magnetic Feel', this should be tested too.

* * *

4.1 The Layouts

The primary tests concern the comparing of the lanthe-Keyboard with a number of different layouts. (For the general layout see: Chapter 3. 'Design Specifications'. For the full pictures see Appendix D: 'The Pictures')

The layouts to be tested are:

1. The lanthe-Keyboard

This layout is newly developed for use in the CD-i environment. Among other factors it is based on the use of restricted areas.

2. The 2-Line Keyboards:

- = The 2-Line Keyboard with Horizontal Scroll-Buttons**
- = The 2-Line Keyboard with Vertical Scroll-Buttons**

This layout seems the most commonly used in the CD-i environment.

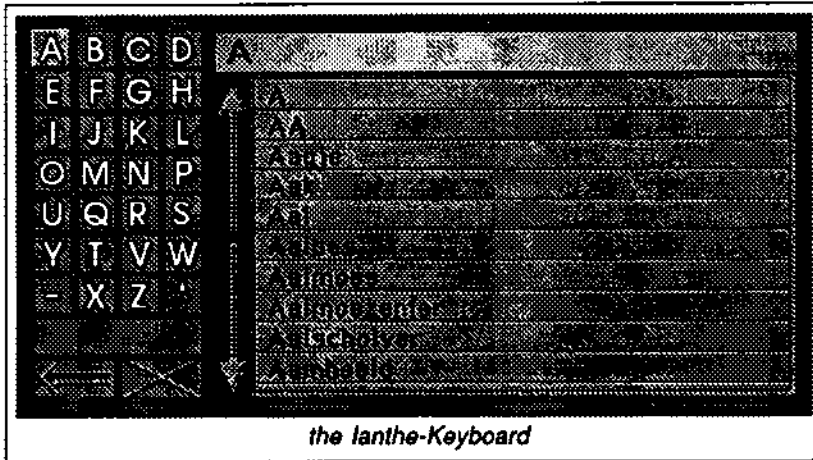
3. The 4-Column Alphabetical Keyboard

Probably the second commonly used layout in the CD-i environment.

4. The QWERTY-Keyboard

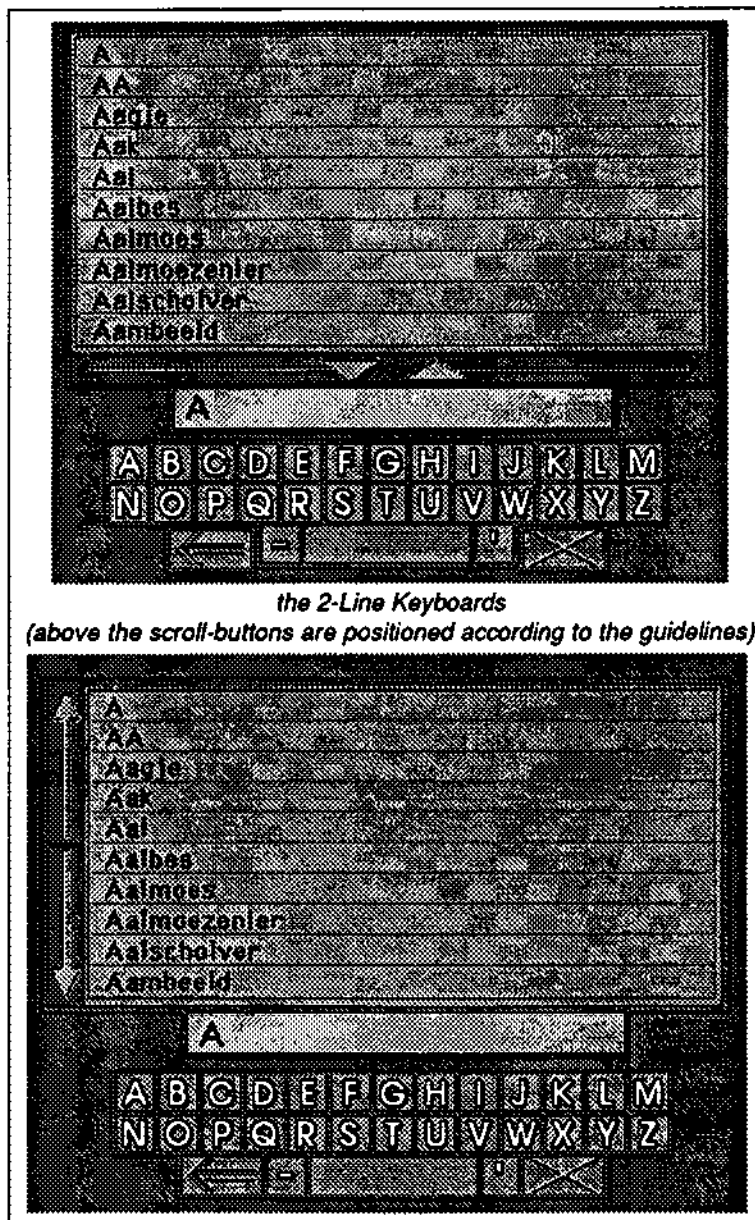
The commonly used layout for ten-finger boards.

1. The 'lanthe-Keyboard'



The 'lanthe-Keyboard' is a 4-Column keyboard (or a 7-Line as far as the characters are concerned) in which the characters are arranged in such a way that the vowels all are in one column (on the far left). This means they can profit from the restricted area (a restriction in cursor-movement) on the left side. This restriction should be extended to the top of the keyboard as well.

2. The '2-Line Keyboards'



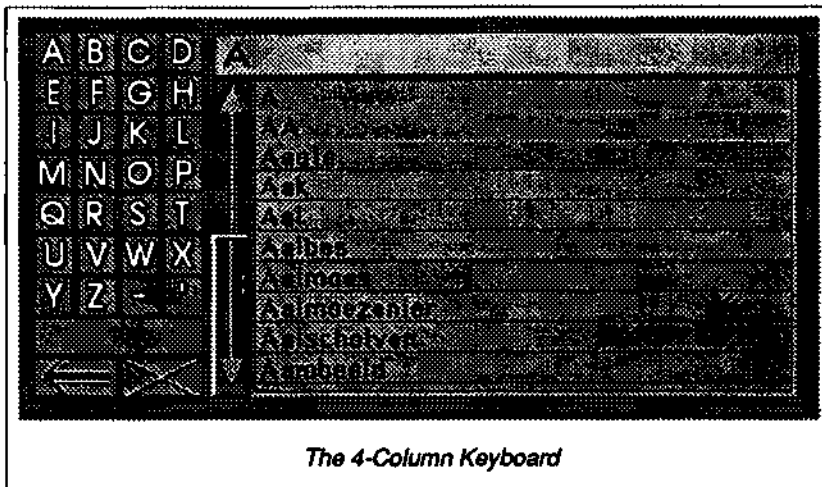
The '2-line Keyboard' is quite commonly used within the CD-i environment. This type of keyboard can easily be given some enhancement by adding some level of 'Magnetism' in order to provide the user with a 'Magnetic' or 'Sticky' keyboard.

Furthermore, in relation to 'Real World Events', it may feel 'natural' to have the text-entry just above the keyboard.

The two versions shown here differ in respect to the positioning of the scroll-buttons. In the upper version the Scroll-Buttons are positioned between the Keyboard and the List, (according to my report 'Guidelines for Designing On-Screen Keyboards').

Usually the scroll-buttons are positioned at the right side of the list (similar to the Macintosh and the Windows interfaces) essential in this matter however is the positioning at the *side* of the list. Therefore the second picture shows a version where the Scroll-Buttons are positioned at the left side of the list, similar to the layout of the 4-Column Keyboard as well as the lanthe-Keyboard.

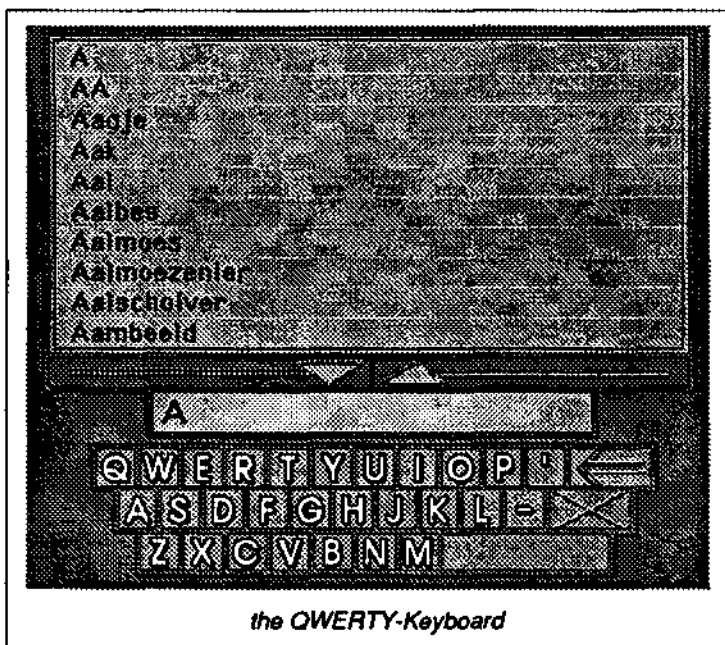
3. The '4-Column Keyboard'



This is the 'traditional' 4-Column Alphabetical Keyboard (or a 7-Line as far as the characters are concerned). This keyboard-layout can be encountered gradually more often in the newer CD-i titles. The characters are arranged in the regular alphabetical order.

The 4-Column Keyboard-layout may gain from the use of 'Magnetism' as well as from the use of restricted areas.

4. The 'QWERTY-Keyboard'



This is a 3-line Keyboard-layout (although often a 4-line layout is used, in which the Spacebar is an independent line of its own) which you still may encounter in some CD-i titles. The characters are arranged in an order which is (was) suitable for mechanical type-writers and has been used ever since.

Despite some later improvements in layouts, nowadays it has become a world-wide standard.

Since this design is based on a ten-finger use, this layout is not a suitable keyboard for the CD-i environment.

★ ★ ★

5. Positioning the Scroll-Buttons

The position of the Scroll-Buttons are tested within the 2-Line Layout. Two different positions are compared:

5.1 Positioning as in the 4-Column Keyboards. (at the left side of the list)

This is done for two reasons:

1. to get all circumstances even (the only difference in the design is the layout of the key-

board)

2. This often is more or less the position where you find these.

5.2 Positioning according to the Guidelines-report

This has to be tested because it is supposed to be a disadvantage for an keyboard-interface not to have the Scroll-Buttons positioned in between the keyboard and the list.

★ ★ ★

4.2 Magnetism

The different layouts will probably benefit from different levels of Magnetism each. Where the 'lanthe-Keyboard' will probably benefit more from the use of restricted areas (being one of its design-basics), the 2-Line Keyboard will probably benefit most from a magnetic feel, since the magnetic feel tends to make the user go in straight lines instead of crossing the keyboard. Where 4-Column Keyboards offer the possibility to cross the keyboard this hardly is the case with a 2-Line Keyboard.

In general I propose the cursor, when 'on' the keyboard, to be changed into a less 'heavy' or –preferably– a more 'transparent' form, in order to prevent it from blocking the view on the selectable character. In case of the use of a Magnetic Feel, the 'magnetic centre' of each key should be positioned just below the character to prevent the cursor from blocking the view.

1. No Magnetic Feel

The Magnetic Feel is tested on three levels, one being Level Zero.

1.1 With Change of Cursor

Since users tend to position the cursor in the center of the keys of the keyboard, often the cursor blocks the view on the selectable character. So, when on the keyboard, the cursor should change into a more transparent form, for instance into a cross.

1.2 No Change of Cursor

Since there seems to be no consensus whether users like the cursor to be changed in appropriate situations, it is wise to put it to a test.

★ ★ ★

2. A Light Magnetic Feel

The Magnetic Feel in this case should hardly be notable on a conscious level, just assisting the user, without attracting attention .

Since users tend to position the cursor in the center of the key they want to select, the 'Magnetic Center' (and therefore the position of the cursor) should be defined with great care, in such a way that the cursor does not block the view on the selectable character. A preferable positioning—assuming the cursor is an arrow, pointing up—should be as much near the center of the key to select as possible, just below the character, so preventing the cursor from blocking the view on the selectable character as well as any other.

This level could be helpful on all four keyboards.

* * *

3. A Strong ('Heavy') Magnetic Feel

The Magnetic Feel in this case is very prominent, and it is clear for the user that he or she is assisted' by the application. It should be worthwhile to consider the possibility for the user to determine either the level or the presence/absence of a Magnetic Feel.

The same remarks regarding the positioning of the 'Magnetic Centre' as stated above should be applied to this level.

* * *

4.3 Restricted Cursor Movement

Restricted areas have different benefits in relation to the different Interfaces. It depends on the position as well as the layout of the keyboard if it will profit, and how much. Since the layout of the lanthe-Keyboard is based on the use of restricted areas it should benefit from it, so would probably the 4-column keyboards in general. A 2-Line keyboard or layouts like 'QWERTY' (which in fact is a 3-Line keyboard) would probably benefit more from a restricted feel.

A 2-Line keyboard could benefit from a restricted area, depending on its position in the interface. Positioned at or near the bottom of the screen—as is the case in the CD-i title used for testing—the downside could be restricted, unless some functionality is found underneath the keyboard. It still is possible to use restrictions in case the functionality underneath is not available. In the application used for testing in general it is not, unless the user 'calls' for it by pressing Button Two of the remote. This would mean presence of a restricted area by absence of the general functions and absence of a restricted area by presence of the general functions... We should not do that to our users.

1. No Restricted Areas

The cursor is free to move anywhere. In case of use of a magnetic feel (on the keyboard) this is used on the keyboard only, and has an equal level for the whole keyboard.

* * *

2. Restricted Feel (Restricted Surroundings)

This is an adaption of the use of 'Magnetic feel'. Instead of a real restricted area the cursor is free to move anywhere, but the surroundings of the keyboard provide a certain amount of Magnetic Feel, making it harder for the cursor to 'break out' involuntarily. In case of use of a magnetic feel on the keyboard, this is not 'equally spread' but more heavy on the outline of the keyboard.

★ ★ ★

3. Restricted Keyboard ('Real' Restricted Areas)

The cursor is free to move, except in those directions that would be of no practical use. This is in fact a way to prevent the user from 'falling off the keyboard', so the outsides of the keyboard, beyond which no functionality is found, can be restricted. As I stated before it is possible to make this dependable on the actual situation in the interface, but I propose not to do so, since it does make the interface less transparent to the user than it should be, thereby making it less usable.

The 'lanthe-keyboard' takes its advantage from the restricted area on the left-side of the keyboard, providing the user with easy and fast access to the vowels.

★ ★ ★

4.4 Types of Feedback

Feedback has many faces, one of them being sounds. In general I propose feedback to be as extended as possible, combinations of several visible forms with audible feedback. In general the minimum feedback should be:

- visible feedback when something is *selectable* (a key of the keyboard for instance)
- audible when something is *selected*

Audible feedback should be used with caution, it is the one form in which too much feedback can do harm. Minimum audio-feedback should be provided to let the user know 'something has happened'. This can be extended so the user may know what has happened. This could be done best in categories. Selecting a character should be accompanied by a sound. Unless it is something like a telephone-interface a simple click-sound should be sufficient. A different sound should be used for deleting, to which I propose that the sound for Backspace again should differ from Delete. It is questionable whether a sound should be used for scrolling, especially if different types of scrolling are provided (scrolling by line as well as scrolling by page). In the case sounds are provided for these functions the possibility should be added to turn them off. Selecting a subject from the list could use the same click-sound as selecting a character from the keyboard.

Visible feedback should at least be given before each select-action, whether it is selecting a character, a button or a subject. This means that the most important feedback is on selectability rather than selection: the user has to know what he or she can select at this position of the cursor. This nearly always is extended because each action will have a result. When a character is selected it appears in the text-field, probably the list is changed or appears. To prevent the attention from going back and forth be-

tween for instance the keyboard and the textfield, the visible feedback can be extended with providing a feedback on selectability and a different feedback on the actual selection.

1. Simple Feedback

The most simple form of feedback is provided, Visible feedback when selectable, Aural feedback when selected.

- A. A selectable button or key is changed in colour. (colour-change of the key/button only)
A selectable subject from the list is changed in colour. (colour-change of the text)*
- B. Selection of a key or a button activates aural feedback. Usually this is a Click-sound, exceptions are the Backspace- and the Delete-Key, which have sounds that differ.
Scroll-buttons are not provided with aural feedback.*

* * *

2. Extended or Multiple Feedback

The extensions concern visible feedback only. Different feedback is provided in relation to the three phases. The results are three variations in key-colour:

- *Non-active* (the key is not selectable)
 - *Selectable* (the cursor is 'on' the character)
 - *Selected* (the character is selected and appears in the Textfield).
-
- A. A selectable button or key is changed in colour. (colour-change of the key/button only)
A selectable button or key is provided with a two-coloured rectangle around it.
A selectable button or key is shifted in position. (Up and to the Left)
A selectable subject from the list is changed in colour. (colour-change of the text)*
 - B. Selection of a key or a button activates aural feedback. Usually this is a Click-sound, exceptions are the Backspace- and the Delete-Key, they have sounds that differ.
Scroll-buttons are not provided with aural feedback.
While being selected a button or key is (again) changed in colour. (colour-change of the key or button only)
While being selected a button or key is provided with an inverted two-coloured rectangle around it.
While being selected a button or key is shifted in position. (Down and to the Right)
While being selected a subject from the list is (again) changed in colour. (colour-change of the text)*

Remark: after selecting, the character should return to phase A. (selectable) until the cursor has been moved away from the character.

* * *

5. Expected Results

5.1 What Users Should Prefer...	30
5.2 The Test-Probabilities	31
5.2.1 The Layouts Compared	31
5.2.2 The Scroll-Buttons	32
5.2.3 A Magnetic Feel	32
5.2.4 The Use of Restrictions	32
5.2.5 Feedback	33
5.3 What Users Do Prefer...	34
5.3.1 General Conclusions on Expected Results	34
5.3.2 Overall Conclusion	34
5.3.3 A Final Word	35

5. Expected Results

5.1 What Users Should Prefer...

Unfortunately no real testing could –yet– take place, due to circumstances beyond our control, so what I have to offer is mere philosophising about the possible outcomes of the tests (which still will be performed but the results will be published in a next report which will not be written by me).

Above all, I have to start with saying that Usability Testing always is necessary. Primarily because of the differences between what *is* easy to use and what people *think* is easy to use and, for that reason, do prefer. As a designer I may think that the lanthe-Keyboard may be a clever solution to some difficulties regarding On-screen Keyboards. But I cannot know if users will appreciate it and if they would want to use this type of keyboard, even although I believe that the concept is sound. The sole fact that a restricted area may improve performance does not mean that users will like it that way.

The same problem arises with the use of extended feedback. As a designer, again I may think that I did develop a smart concept in which the change in key-colour as well as a rectangle around the 'key' as well as a shift in position all come together in a natural way, but it still is a rational solution. I still cannot predict that users will like it only because I am so impertinent to think that 'it is best for them'... What we do need is Usability Testing, because even users are not the logical people that we think we are. Besides, we all like to kid ourselves in some ways. To prove this I share with you an example that I find very illuminating¹.

From all devices, developed to make a computer more user-friendly, one of the most important inventions probably was The Mouse. In fact, the mouse is so user-friendly that quite a lot of experienced computer-users look upon it as a device for the computer-illiterate. And since computer-experts are not, they generally prefer key-commands. Not for that reason only of course, keyboard-commands are just faster to use.

Unfortunately, they are wrong. All research in this field and all tests do show that *in all cases the mouse at least is as fast in use as keyboard-commands are*². So, why do all those logically thinking people not use the mouse?

The answer is most disillusioning about people being logical: it just does not *feel* that way. Especially expert-users of computers do not experience the mouse as being faster than keyboard-commands. *They experience the keyboard-commands as being faster*, and not without reason, of course. It is only that

¹ The example is described in 'Tog on Interface', Chapter 6: 'Command Keys vs. the Mouse', page 26, and Chapter 22: 'The Holy Interface, or Command Keys Revisited', page 179 to 182; written by Bruce Tognazzini.

² This is a simplification of the truth, as 'Tog' says it: "(Quote) ...it takes just as long to decide upon a command key as it does to access the mouse. The difference is that the command-key decision is a high-level cognitive function of which there remains no long-term memory. Therefore, subjectively, keys seem faster when in fact they usually take just as long to use.

Since Mouse acquisition is a low-level cognitive function, the user need not abandon cognitive process on the primary task during the acquisition period. Therefore, the mouse acquirer achieves greater productivity..." (from 'Tog on Interface', chapter 22 'The holy interface, or command keys revisited', page 181)

the real reason is that the mouse can be used 'without –conscious– thinking'. Of course you do have to think, about where to go, the position of the mouse, et cetera, but this level of thinking is so low it leaves you plenty of time to notice other things. Not so with the keyboard-commands. It requires a much higher level of brain-activity to remember what keys to use. So high in fact, it takes all your brain-activity for that moment *and you do not experience time*. Very much unlike using the mouse, you indeed did experience time there. Measured, the mouse-time may be considerably shorter than the key-command time, but since you do not *experience* that last time, it simply does not feel that way.

Well, since we are assumed to be so logical, I wonder what will you prefer from now on.....?³

★ ★ ★

5.2 The Test-Probabilities

5.2.1 The Layouts Compared

My expectations are, that between the keyboard-layouts solely (the layouts are compared 'sec', without any other attachments even if they are a part of the design), the QWERTY-keyboard will be the less satisfying. I expect it to be slowest (measured as well as perceived) in time and the least appreciated. The others will end up neck to neck with possible a slight preference for the 2-Line Keyboard (especially among the experienced CD-i users, since they may be used to it).

Since the Ianthe-Keyboard is designed for use with a restricted area and this is left out for testing purposes, it probably will not be preferred in this stage of a test.

As far as the QWERTY-layout is concerned, I may refer to the example in 5.1 'What Users Should Prefer'. Experienced QWERTY-users (e.g. typists among others) still may prefer this layout because they know it well (or at least they think they do). They could prefer it because they expect it to be faster for them. Even after testing they do not necessarily change their mind, due to these differences between measurement and perception.

★ ★ ★

³ And to illuminate this even more: Yes, I do use a mouse –after all, I like my Mac– but also I do use keyboard-commands... I am so rational, don't you think?

5.2.2 The Scroll-Buttons

My expectations are that Scroll-Buttons above anything else should be practical and therefore the positioning has its influence on the appreciation. Since Scroll-Buttons most times are needed when the user (the cursor) leaves the keyboard and is moving towards the list, they should be positioned between the keyboard and the list. Therefore I expect that positioning to be preferred.

Important may be the level of magnetism that may be used on the scroll-buttons, it should be zero to 'very light', because the Scroll-Buttons must not 'get in the way' when the users only wants to go to the list directly. 'Getting stuck' is not helping in that case.

* * *

5.2.3 A Magnetic Feel

My expectations are that the appreciation of magnetism partly will depend on the layout of the keyboard. Any keyboard that uses more than two lines may benefit from the possibility to cross the keyboard. As the magnetic level increases it becomes more difficult to do so. So where the 2-line keyboard may benefit from a stronger magnetic feel, the lanthe-Keyboard and the 4-Column Keyboard could benefit from the lightest level or even absence of magnetic feel.

As I stated before (4.2 'Magnetism'): a point of consideration is either the positioning of the cursor, or the shape of the cursor. When magnetism is used it is important that the 'Magnetic Centre' of each key is chosen in such a way, that the cursor does not block the view on either the selectable character or the characters that surround it. So the Magnetic Centre should be chosen with care, the best solution probably being just underneath the selectable character.

By absence of a magnetic feel (and possibly also with a very light feel) I would prefer to change the cursor to a more 'transparent' shape. Instead of a 'massive' arrow it could be shaped like an open arrow, a cross, maybe even a kind of 'Ghost-Arrow', being still visible but also having the background showing through.

* * *

5.2.4 The Use of Restrictions

My expectations are that it opens up possibilities for several keyboard-layouts, but it seems reasonable to presume that keyboards that are designed with specific restrictions in mind should benefit the most and therefore may be preferred. This presumption puts the lanthe-Keyboard in the lead as far as this aspect is concerned.

The key-factor of course is to find the golden mean, where as much characters as possible may benefit

from the restrictions and still enough freedom is left to be pleasant.

As far as a restricted feel is concerned it seems reasonable to assume that those keyboards that benefit from any magnetic feel should benefit from a restricted feel as well. In case of the lanthe-Keyboard as well as the 4-Column Keyboard this level should be well-chosen, meaning not 'too heavy magnetic', because, since the Scroll-Buttons are positioned next to the keyboard itself, 'bursting out of the keyboard' could make the cursor jump over the Scroll-Buttons, even when not intentioned. This is not the case with the 2-Line Keyboard nor the QWERTY-layout since these have a textfield between the keyboard and the Scroll-Buttons.

In fact the use of restricted Cursor Movement is quite common in the CD-i environment. It seems reasonable to assume that it is wise to embed this particular use (concerning the keyboard) in a range of layouts. When the outer left of the screen already is a restricted area, the lanthe-Keyboard as well as the 4-Column Keyboard should be positioned in such a way that the users do not experience this restriction as an extra. Coming from the lanthe-Keyboard the top is a restricted area, but coming from the other side (above the list, and approaching the keyboard) the cursor should not be 'blocked' but automatically jump to the top-line of the keyboard instead (combining the restriction in movement with a certain kind of magnetism, so probably preventing the user from feeling restricted...) Although another solution of course could be to restrict the whole top-part of the interface.

★ ★ ★

5.2.5 Feedback

My expectations are that extended (multiple) feedback is preferred above simple (single) feedback. Here the crux could be to find the right concept, maybe the right metaphor to provide this extended feedback with a 'natural' feel. It is not the problem of merely adding bells and whistles, road-signs and flashlights...

Most people (among which are a lot of users) do not prefer to be treated as 'stupid', so we should look for ways to make this 'Redundant Feedback' feel natural. One possibility, already commonly used, is combining visual feedback with aural feedback, but future designers could extend this on the visual side. I merely showed just one way to do this, now it is up to the real designers either to make it look good or find an own -better looking- way to do this.

Apart from the fact that a user or a designer may despise a 3-dimensional look, it still is a solution that brings together three different aspects of multiple feedback. And above that it leaves room for a designer to make things look good. (I am not so pigheaded as to think that my designs look beautiful, but I am so pigheaded to think they are sound concepts. And also, I think I know when to leave things to the real arts and crafts.)

★ ★ ★

5.3 What Users Do Prefer...

5.3.1 General Conclusions on Expected Results

My expectations are that different aspects work for different layouts in different ways. At first sight the lanthe-Keyboard will be handicapped, because it does not resemble (how should it do that) a 'known' keyboard. On the other hand, many 'known' keyboards, as is the QWERTY-board, not always work so very good in the on-screen environment. (On the contrary, in the CD-i environment 'QWERTY', for one, is a clumsy contraption of characters⁴.)

To summarise the –expected– conclusions:

- The positioning of the scroll-buttons: they should be between the keyboard and the list (although I am afraid it may sometimes mean a harder job for the graphical designer).
- Usability of magnetism: it differs for the various keyboard-layouts. It is hardly useful for the lanthe-Keyboard, except in 'keeping the user on board'. It does however quite a good job in case of the 2-Line Keyboard.
- Since magnetism seems to force the user to move in straight lines (and most users do not like to *feel* being forced), it is best to use a rather light magnetic feel since this is the least experienced consciously.
- In case of absence of a magnetic feel, the cursor, while 'on' the keyboard, should be changed to a more 'transparent' shape.
- The use of restricted cursor movement (since it is quite common in the CD-i environment) should, if possible, be embedded in a whole of arrangements.
- Feedback in general should be clear as well as redundant, meaning the use of Multiple Feedback.

★ ★ ★

5.3.2 Overall Conclusion

Based on my expectations, the easiest way seems to be to tell you simply to use the lanthe-Keyboard with the applied 3D-feel, but I am afraid it is a bit more complicated. The layout may work well, the restrictions make it very useful, the positioning of the Scroll-Buttons goes without saying, the Multiple Feedback provides ease of use, et cetera, but still the overall conclusion must be that it should be offered as a *default*.

The logical consequence to this conclusion has to be that we have to provide options for the user. This opens a wide range of possibilities, as we could provide the users with, for instance, different keyboards, each embedded in its own 'set' of optimum conditions.

The Alphabetical 4-Column Keyboard, for one, could be included (although it is not that much diffe-

⁴ This is a matter of Performance versus Preference (or rather: Perception), of which the latter mainly concerns the 'safety' of known and familiar circumstances.

rent from the lanthe-Keyboard), 'defaulted' with another set (a certain level of magnetism, no 'hard' restricted areas but a restricted feel, et cetera). A more likely option however could be the addition of the 2-Line Keyboard (although this could mean quite a lot of differences in the whole layout of the interface), which also should be defaulted with a certain level of magnetism; and maybe even the QWERTY-layout (not because it works so well for CD-i, but simply because it is a known layout and users may like it for that reason, although personally I do not believe in it).

★ ★ ★

5.3.3 A Final Word

I do believe in guidance for the user, but guidance with freedom, the freedom to choose. It means the freedom to choose for the unexpected, the freedom to choose for the laborious, but, above all, the freedom to choose the preferred, for whatever reason this preference may be. Because that is exactly what is really important for people: not *having* to choose, but *having the possibility* to choose.

Our task is to provide the optimum possibilities to choose from, which means a lot of hard work. For instance, it is hard work to decide which options will be in the hands of the user, since we should not let the user choose his keyboard from 10 or more defaulted choices (as a 2-Line Keyboard with heavy magnetism, a 2-Line Keyboard with light magnetism, and so on), neither should the user probably have to decide what level of magnetism to apply to what keyboard, but the same user could easily choose between some well-chosen sets. (And then again there is some more hard work in doing usability-tests to see if our decisions were right...)

So lets provide the user with choices. Provide him or her with some well-chosen defaults and leave it to the user (preferably to every individual user) to choose his or her own defaults⁵.

★ ★ ★

⁵ Games are an example where users commonly use the same disk, but with –for instance– different remotes: each player uses the one that he or she prefers. The same should be made possible as far as keyboards are concerned: provide the possibility for each user (player) to assign his or her own favourite one. In the case of feedback it may even make an extension, providing the players with additional –redundant– information whose turn it is.

Appendices

A. Additional Functionality	39
1. Concerning Keyboard-functions	40
2. Concerning Search-functions	34
3. Concerning List-functions	35
4. Concerning Marker-functions	36
B. Additional Keyboard-Designs	43
1. The Cube-Keyboard	44
2. The Ball-Keyboard	45
3. The Wheel-keyboard	46
C. Additional Possibilities	47
1. The Use of Games	42
D. The Pictures	49
1. The Keyboard-Layouts	50
2. The Keyboard Behaviour	55
3. Selecting an Item from the List	57
4. The Additional Pictures	58

Appendix A

Additional Functionality

1. Concerning the Keyboard-functions	40
The Delete-Function	40
2. Concerning the Search-functions	40
The Word Search-Function	40
The Multiple Search-Function	41
The Media Based Search-Function	41
3. Concerning the List-functions	42
The Added Information-Function	42
4. Concerning the Marker-functions	42
The Composing of a 'Set of Marks'	42

Appendix A

Additional Functionality

1. Concerning the Keyboard-functions

– *The Delete-function*

Since the Delete-button and the Backspace-button are placed next to another on most keyboards, users either have to be very careful in selecting the right button, or accept the fact that sometimes they accidentally hit the Delete-button, while the intention was only to delete the last letter of the entry. Neither of these will improve speed or ease of use (or a feeling of satisfaction). A possible solution could be to change the Delete-button –after using it– temporarily to an 'Undo-Delete button', just until the user either selects this function (meaning he or she intended probably to use the Backspace-button) or starts to select characters for another entry.

This added functionality does have its consequences for the list-functions. When a delete-action can be 'undone', then the list, the one present before the delete-action, should not be cleared, but only be changed at the moment the user starts another entry. Until then the list should still be visible and available (It would still be possible to select an item from the list). Eventually the list could be changed in colour ('greyed' or suchlike).

In most interfaces the Delete-button as well as the Backspace-button are represented by highly abstract symbols. A lot of users could be done a favour by adding text to these buttons, for instance in the Selectable State, so the user could know unmistakably what he or she is able to select.

* * *

2. Concerning the Search-functions

– *The Word Search-function*

Most search will probably be done on a 'Keyword' or a 'Subject', from where cross-references can be made through hyper-linked words. Sometimes it could be very useful to get all articles that refer to a certain subject, or for instance all articles in which a certain item ('Keyword') is named (Commonly called

'Full-Text Retrieval').

The result could either be a list of titles of articles to select from (that would be my personal choice) or it could be the row of articles directly, through which the user may scroll his or her way.

There are not that many differences between 'Search by Title' and 'Search by Keyword(s)', apart from the different headings. Differences may be found in the list, which could (in case of the latter) show more additional information, as for each article the number of pages, for each entry-word the number of articles, et cetera (This is not shown in any picture).

★ ★ ★

– *The Multiple Search-function*

A useful extension of the 'Word Search-function' could be this function, which allows multiple words to be searched within all available articles. A point of attention might be that most users would like to be informed beforehand how many articles are/will be found in which the searched item is mentioned. If this should exceed a certain number (say about 200), then a narrowed search could be useful. For use in the CD-i environment it would probably be sufficient to have an 'Included Search¹'. Such a Multiple Search-function should probably offer the possibility of a combination of three items at most.

★ ★ ★

– *The Media Based Search-function*

Sometimes it could be very helpful to have a search narrowed beforehand. When a user only wants to browse the pictures, or sounds only, or just wants to see the movies or the slide-shows. This could for instance be of great use in the field of education, since particularly teachers would be very interested to know –preferably in a simple way– for instance which pictures are available.

The narrowing would even not have to be done beforehand. We could provide buttons (underneath the list for instance) for a more specific search (Categories could for instance be: 'Articles', 'Movies' (Video), 'Pictures', and 'Sounds'). Then we could even offer the possibility to toggle between views on categories at any time. As a default 'Articles' should be selected.

These categories could also be shown as icons but when selectable, then it would help most users to have the function-name shown over the icon. Icons can be great for a fast search (very depending on the number of icons!) but in the selectable state (textual) confirmation is even better.

★ ★ ★

¹ This means that the user is not confronted with choices for different types of search as for 'item A and item B', 'item A or item B', 'item A with exclusion of item B' et cetera.

3. Concerning the List-functions

– *The Added Information-function*

When a list of titles is displayed, it is possible to add extra information so the user would know beforehand what he or she could expect when a title is selected. This could be plain text or one or more pictures could be included, or movies, or sounds et cetera. This could be shown by icons. Most interfaces offer these information only in the article, sometimes up front sometimes scattered, sometimes even only at the end of the article so a user most times probably would have to scroll to find out....

A sensible extra addendum at the end of an article could be something like 'Recommended References'. This would be in the spirit of "If you are interested in this subject then we can recommend these subject too". This would have to be apart from the already present hyper-linking of words.

★ ★ ★

4. Concerning the Marker-function

– *The Composing of a 'Set of Marks'*

This could be of great help for use in the fields of education. Imagine this to be something as a kind of slide-show, in which the 'slides' may be pictures, sounds, movies, text et cetera. For use in the field of education, for instance by a teacher preparing some lessons, it would have to be possible to arrange the order of the 'slides', by defining the order of the marks. Having composed such a 'Set of Marks', this set would have to be named, in order to be able to reuse the work. Several sets could be composed, each with its own –user defined– name.

To get full advantage of this system the usual functionality of the markers (often called 'bookmarks') would have to be changed. 'Normally' the user goes to the list of markers and selects a marker. After looking in on that subject the user may return to the list or select a hyper-linked word or suchlike. Within the proposed use, it would be more useful within a 'Set of Marks' to be able to go back and forth (within the 'slide-show') without having to return to the usual Main List of Bookmarks.

★ ★ ★

Appendix B

Additional Keyboard-designs

1. The Cube-Keyboard	44
2. The Ball-Keyboard	45
3. The Wheel-keyboard	46

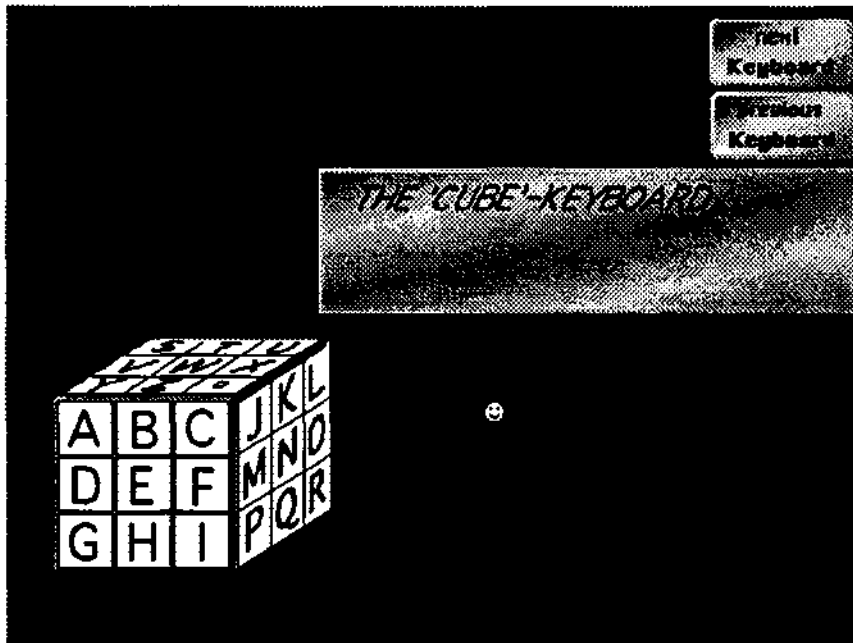
Appendix B

Additional Keyboard-designs

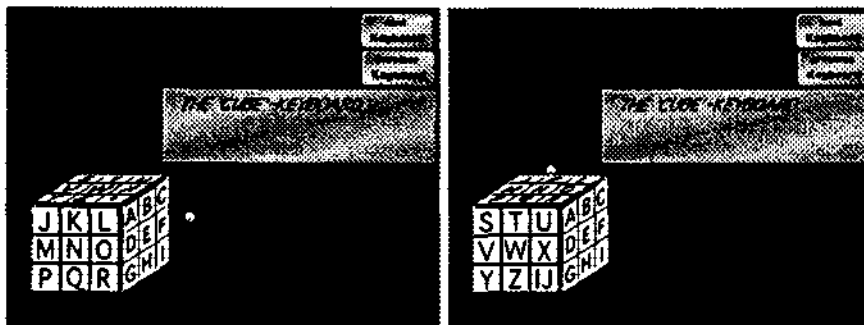
Concerning 'playful Keyboards'

1. The Cube-keyboard

Maybe this is not the most practical design but it offers a full alphabet visually with a relatively limited use of space.



The 'Cube-Keyboard' (in 3 different stages)



The 'Cube'-keyboard looks like a 'Rubik's Cube' with letters on it. Each side has nine positions and three sides are visible permanently, making a total of 27 positions. This provides a vast range of possibilities in arranging the alphabet.

In the 'Cube' as shown in the Picture above, the alphabet is arranged along the three visible sides of the Cube. With characters on each side, the front (in 'starting position') could be 'A' to 'I', on the right-side 'J' to 'R' and on the top-side 'S' to 'Z' along with either a 'space-key' or a 'delete-key'. The user can either choose one of the nine front-characters or replace them with another set (either the 'top-set' or the 'side-set').

This arrangement limits the number of possible movements to the nine characters in front (these are the

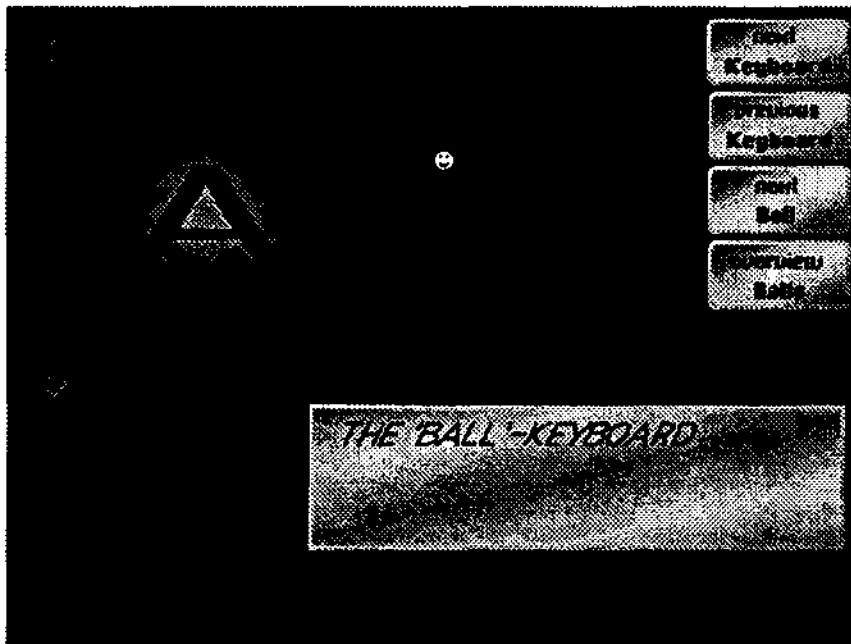
ones that are 'active') along with the two movements which bring the left- or the top-side in front. But it provides the user with a view on the whole alphabet at any time with only a very limited use of space. The 'Cube' should even work better with use of restricted areas, preferably at the left and at the bottom.

Another arrangement could be the one as used in the 'Ball'-keyboard (see hereafter). This opens up a new range of possibilities but it also will make it more difficult for the user to comprehend.

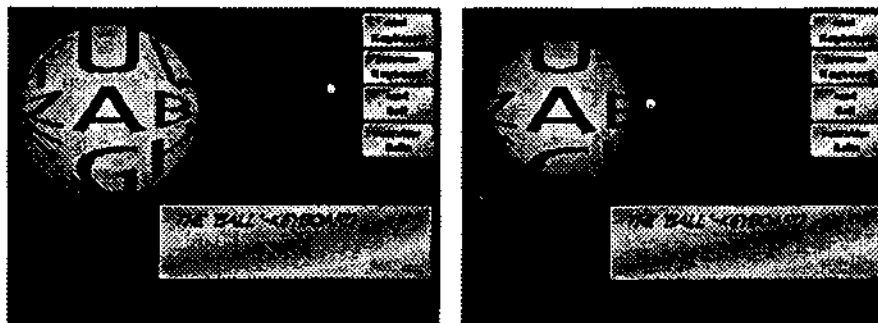
★ ★ ★

2. The Ball-keyboard

Although this is designed as a keyboard, there are additional uses possible, it could be a ball with icons for different purposes, et cetera.



The 'Ball-Keyboards' (in 3 different shades)



The 'Ball'-keyboard looks like a ball with letters on it. The 'Ball' is positioned in such a way that there are three rows of letters visible, with one character clearly in the middle. These rows are in alphabetical order from left to right, but the rows are shifted in respect to each other.

The user can 'push' (the cursor may for instance change to an open hand) the Ball forwards (and go from 'A' to 'B', et cetera) or backwards (and go from 'A' to 'Z' to 'Y', et cetera). But furthermore, the user can make 'jumps' in the alphabet by pushing the Ball downwards (and go from 'A' to 'G' to 'M', et cetera) or upwards (and go from 'A' to 'U' to 'O', et cetera).

The 'Ball'-keyboard provides a way to 'jump' from one part of the alphabet towards another. It makes

use of shifted positions in lines of characters which are in alphabetical order themselves. Only one character is 'really' showing, the others which surround it are less visible (smaller and 'rounded away'). For example: 'A' is accompanied with the 'B' on the right and the 'Z' on the left, but above the 'A' is the 'U' with the 'T' on the left and the 'V' on the right, while underneath the 'A' is the 'H' with respectively 'G' and 'I' on the left and right sides. The ball can be 'rolled' in either direction and so facilitates jumps.

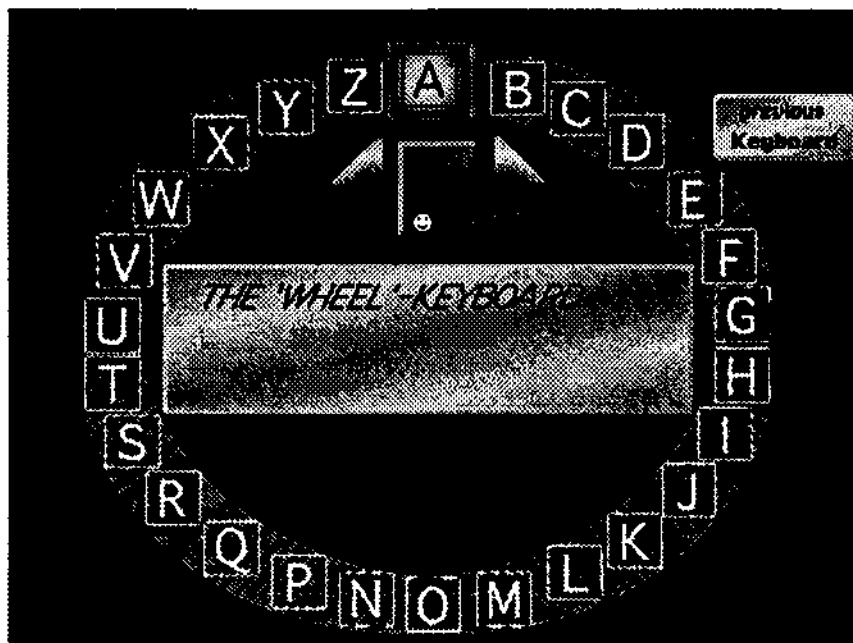
¹ It may come to mind that the cube could actually 'roll' on the screen as a kind of virtual dice. I choose for holding the cube in exact the same position, thus leaving the user with more 'work' (on entering a character from a non-selected side), since the user has to move to a side, click, and move back, but users seem to prefer this simpler mental model.

In the picture there are two buttons provided with the Ball, one for space, another for deleting a character. These are of course options, but should not be placed on the 'Ball'-keyboard itself.

A possible adaptation of the 'Ball'-keyboard could be a 'Cylinder'- or 'Roll'-keyboard. In this case it is possible to 'roll' the cylinder to the next character or shift either the cylinder or the visible piece to another part (This could be done by using the cursor as a kind of 'spot-light'). The Cylinder could be placed vertical as well as horizontal.

— The Wheel-keyboard

In the concern for mental load this could be an improvement on the Letter-strip.



In this design the alphabet is simply placed in a circle which can rotate, clockwise as well as counterclockwise. The controls could simply consist of two buttons: 'to the right' and 'to the left', but more likely, in a 'text-mode', moving the trackball or the joy-stick to the left or to the right could by itself activate the rotating of the wheel.

In the picture, I did provide two buttons for rotating, and a separate button in the middle for actually 'typing' the chosen character. The text-field was placed on the inside of the circle.

A disadvantage of this design is the amount of space it takes on the screen. (Although with more time and more careful planning and different 'letter-blocks'—smaller ones, circles for instance- this could be decreased a lot.)

The main advantages of this design are that it shows the whole alphabet, it is shown in the right order, and the user only needs two controls (a left/right control, preferably the joy-stick) and one button to 'click' the chosen character. This could make the design very suitable for low-level use.

Some adaptation of the 'Wheel' could be a single bar, not shown in silhouette but from above (or in front, whatever you prefer). An advantage would be the space it leaves for other parts, a great disadvantage would be the mental load on the user, who would have to 'link' the outsides of the alphabet in his or her head.

Appendix C

Additional Possibilities

1. The Use of Games	48
---------------------------	----

Appendix C

Additional Possibilities

1. The Use of Games

Games are very common in the CD-i environment. Not only the Titles that are just games, but also a lot of 'Edutainment' and 'Infotainment'. I propose the addition of 'a Game' in as many titles as possible, not only including the 'serious' titles, but rather just with emphasis on the serious titles.

Several reasons apply to this matter.

1. Most 'Serious Titles' tend to teach or explain something, with the intention to facilitate or assist the process of memorising. Employment of the right game may be a perfect solution for that target.
2. Most people like to 'break out' every once and a while. With a nice game on board, the possibility is offered to break out, without having to switch discs or even –depending on the disc– having to leave the application.
3. The user is offered the possibility to start with the game and from there on obtain information, explanation et cetera. So the user is offered the possibility to finally end up with the application the disc was made for in the first place.

★ ★ ★

Appendix D

The Pictures

1. The Keyboard-Layouts	50
The 2-Line Keyboards	50
The 4-Column Keyboard	52
The Ianthe-Keyboard	53
The QWERTY-Keyboard	54
2. The Keyboard Behaviour	55
3. Selecting an Item from the List	57
4. The Additional Pictures	58

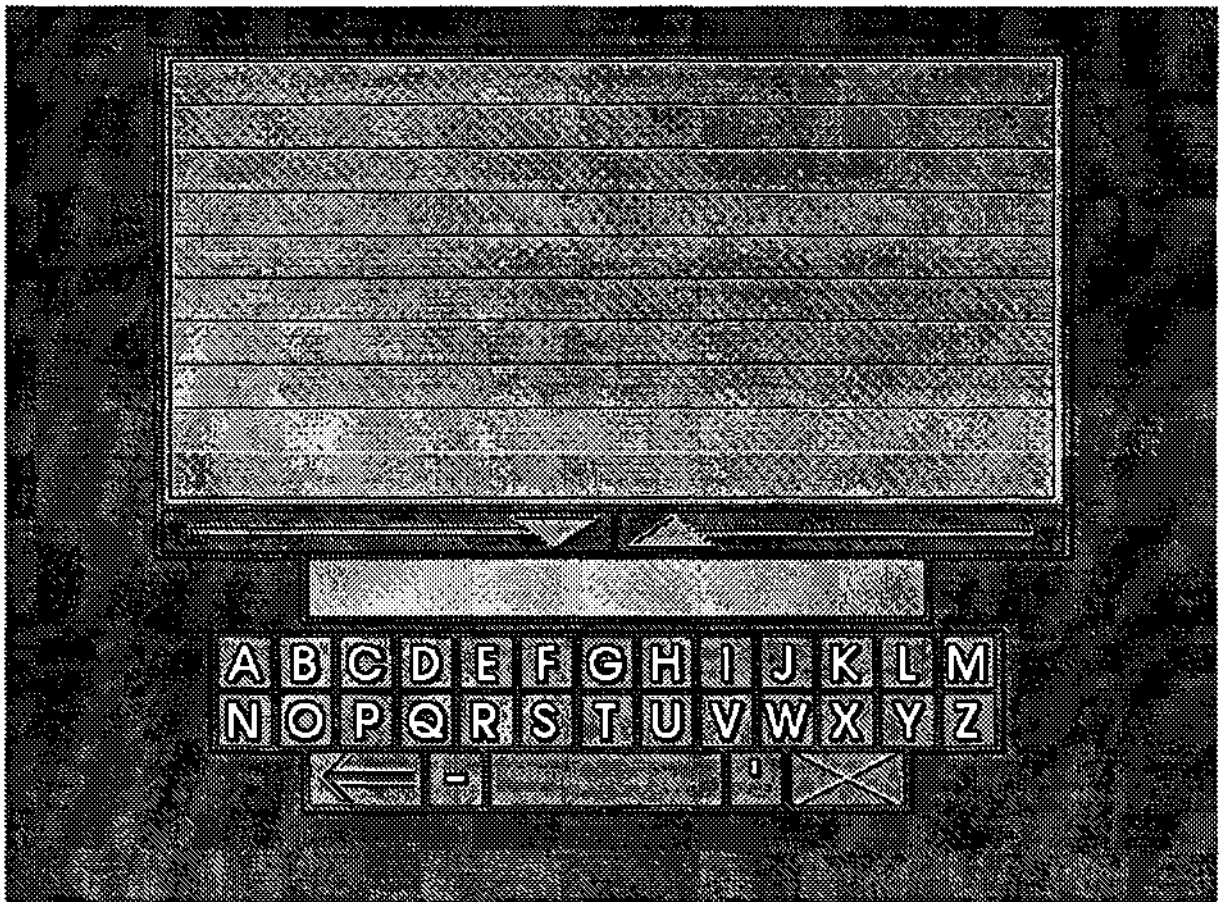
Appendix D

The Pictures

1. The Keyboard-Layouts

Four different layouts are used, two 'traditional' 2-Line Alphabetical Keyboards, one 4-Column (horizontal) Alphabetical keyboard and a newly developed design, the 'lanthe-Keyboard'.

The following pictures show the different layouts.



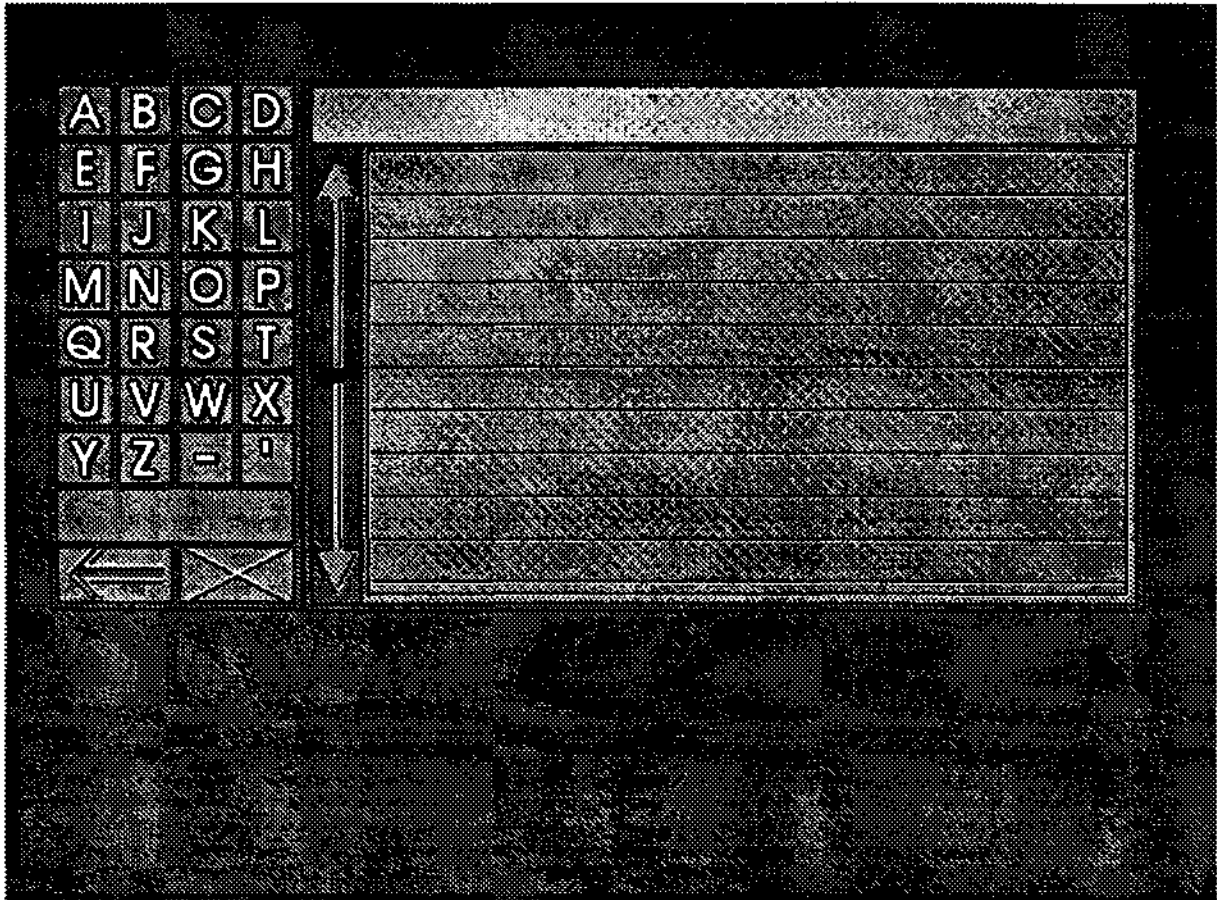
The 2-Line Keyboard 'Preferred'

This Keyboard-Layout differs from the other 2-Line keyboard in the position of the Scroll-buttons. In this Layout the 'preferred' position is used.



The 2-Line Keyboard 'Traditional'

This Keyboard-Layout differs from the other 2-Line keyboard in the position of the Scroll-buttons. In this Layout the 'traditional' position is used.

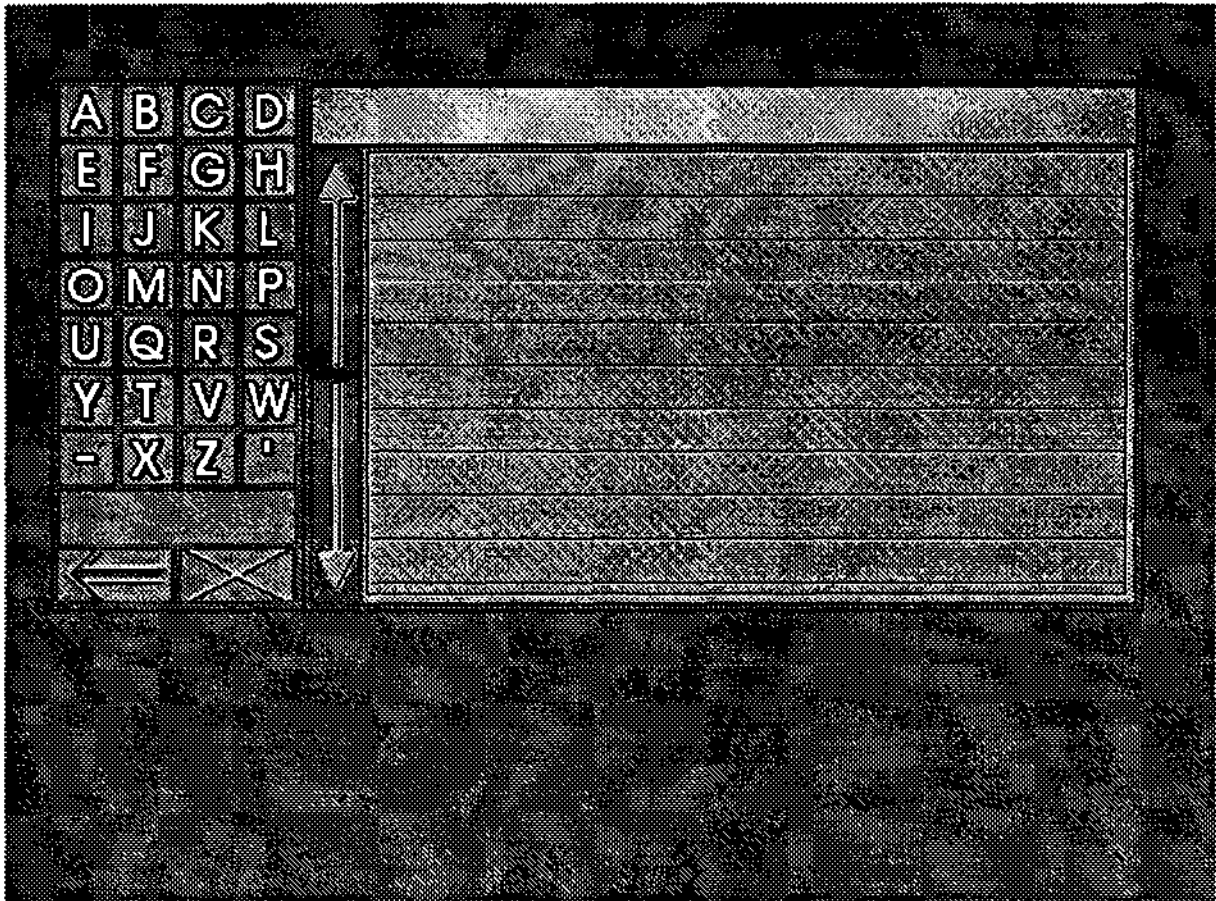


The 4-Column Keyboard

This Keyboard-Layout differs from the preferred 2-Line keyboard in the position of the Scroll-buttons.

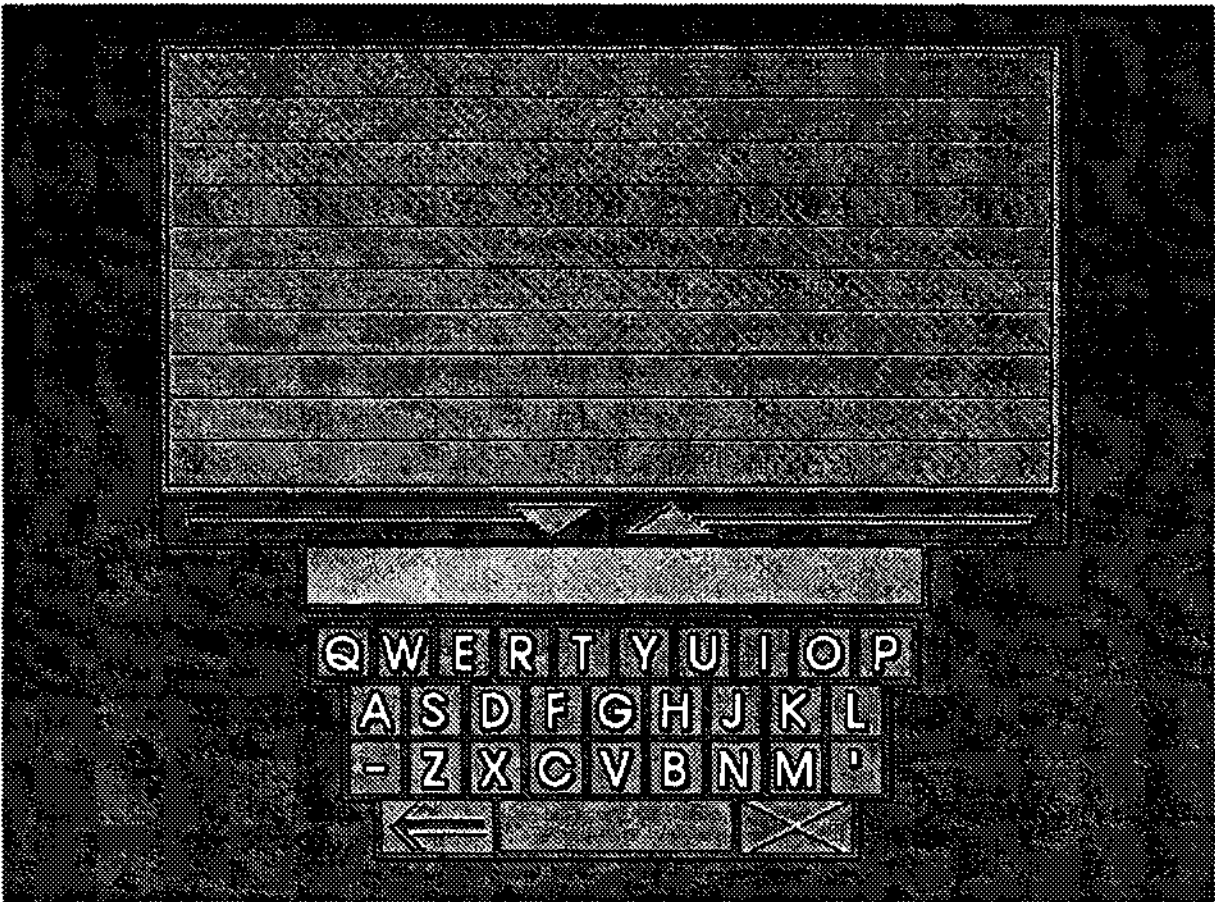
In this Layout the 'traditional' position is used, but here it is similar to the preferred position.

The layout differs from the 'lanthe-Keyboard' (which is also a 4-Column Keyboard) in the arrangement of the vowels.



The 'lanthe-keyboard'

This Keyboard-Layout differs from the traditional 4-Column keyboard in the arrangement of the vowels. In this Layout these are positioned in such a way they form a straight column in order to profit from the restricted area as much as possible. The position of the Scroll-buttons is similar to the traditional 4-Column keyboard as well as the traditional position within the 2-Line Layout.



The QWERTY-Keyboard

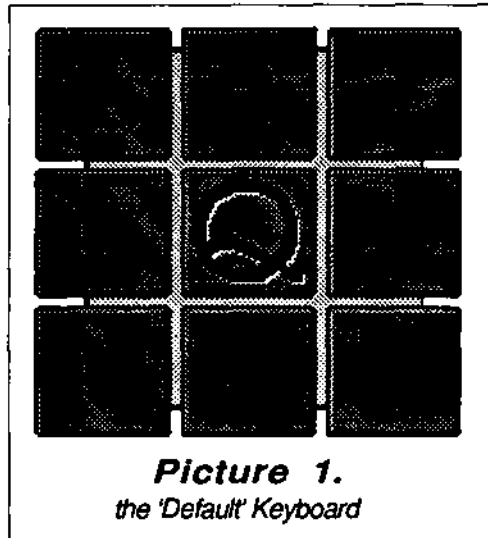
This Keyboard-Layout is the traditional layout for ten-finger use. It is not very recommendable for the CD-I environment but since it is still widely used it seemed wise to include this layout in the tests.

★ ★ ★

2. The Keyboard Behaviour.

Above all this is a visualisation of the (extended) feedback as described in Chapter 3 'Design Specifications'.

Pictures 1. to 3. show the different stages in selecting a character. In the pictures all characters are left out, except the character to select, being the 'Q'.

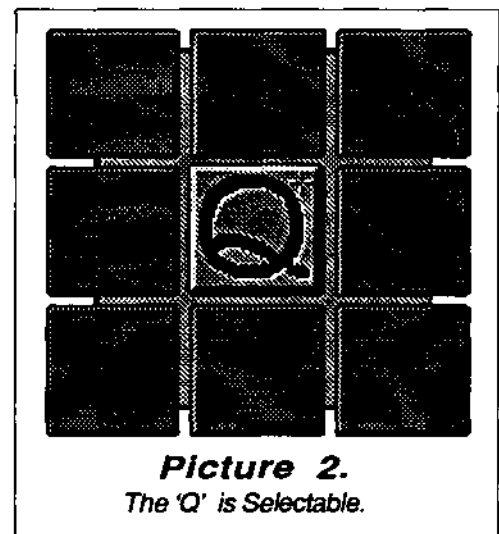


Picture 1.:
the 'Default' Keyboard.

The cursor (an arrow) is not visible.

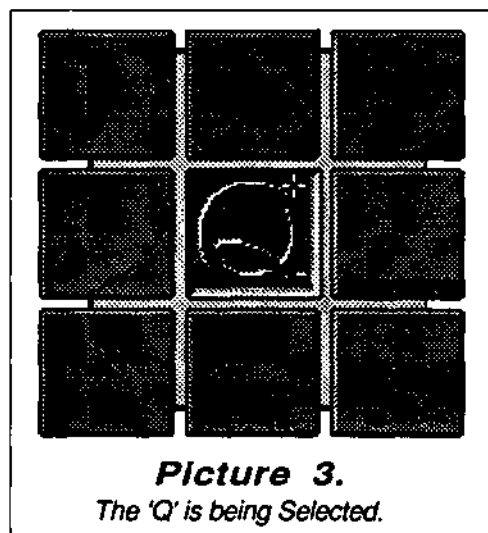
Picture 2.:
the 'Q' is *selectable*.

This is shown by a colour-change (lighter) and a 'rise' of the key (consisting of a shift in position combined with a two-coloured rectangle around the key). The cursor is changed to a Cross.

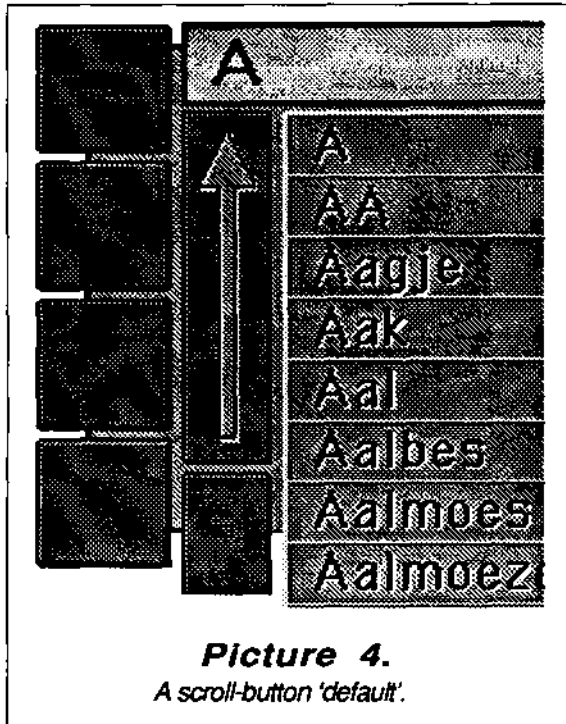


Picture 3.:
the 'Q' is *being selected*.

This is shown by a new colour-change (darker) and a 'sinking' ('pressing') of the key (consisting of a shift in position combined with a two-coloured—reversed—rectangle around the key). (Cursor is still changed to a Cross.)



Pictures 4., 5. and 6. show the same type of feedback but then regarding the Scroll-Buttons (in the pictures only 'Scroll-Up' or 'Go-Down', whatever you prefer). Here the cursor is changed into a cross also, but in fact this depends on the level of magnetism being used. The cursor-change only is necessary by absence of a magnetic feel, but not necessarily also on the scroll-buttons.



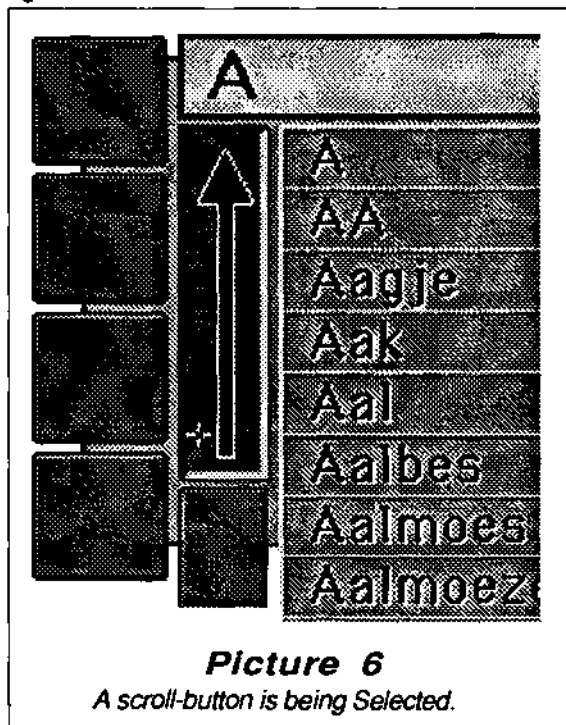
Picture 4.

A scroll-button 'default'.

Picture 5.:

The Scroll-Button is *selectable*.

This is shown by a colour-change (lighter) and a 'rise' of the key (consisting of a shift in position combined with a two-coloured rectangle around the key). The cursor is changed to a Cross. One more change in visual appearance is added, the 'depth' of the arrow on the button also is changed, from 'bas-relief' to 'high-relief'.



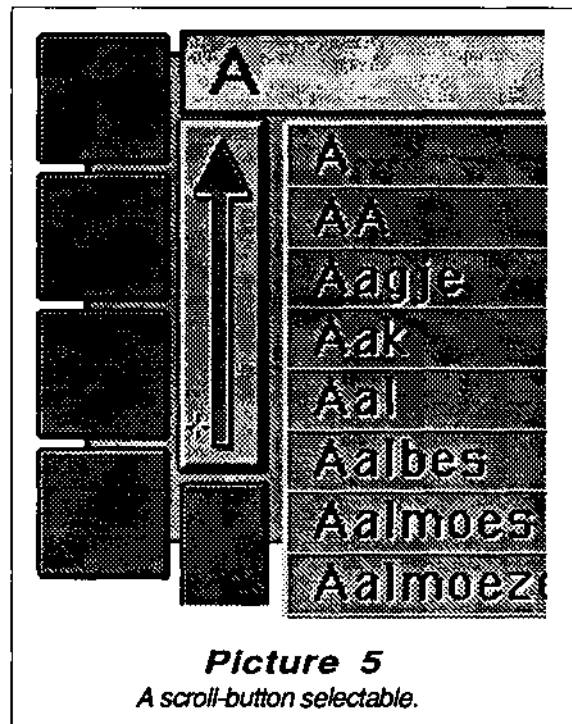
Picture 6

A scroll-button is being Selected.

Picture 4.:

The Scroll-Button 'Default'.

The cursor is not visible.



Picture 5

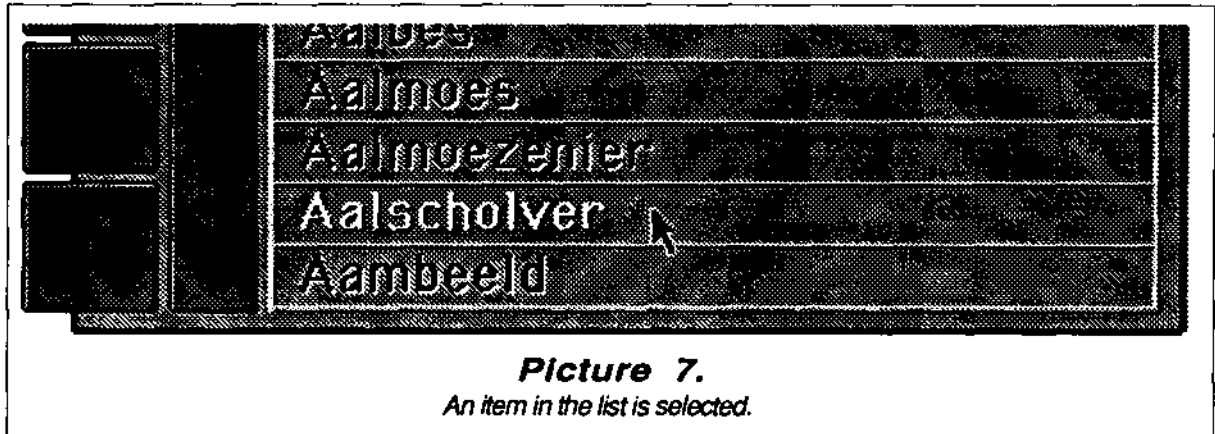
A scroll-button selectable.

Picture 6.: The Scroll-Button is being selected.

This is shown by a new colour-change (darker) and a 'sinking' ('pressing') of the key (consisting of a shift in position combined with a two-coloured -reversed- rectangle around the key). (The cursor is a Cross.) The arrow still is in 'high-relief'.

3. Selecting an Item from the List

Picture 7. shows the selecting of an item from the list. It is apparent that the whole line is selectable, the selectability showing by a colour-change of the selectable item.



Picture 7.:

One of the items in the list is *selectable*.

The cursor is changed back into an arrow (in case of absence of a magnetic feel, otherwise it should not have to be changed in a more transparent form).

4. The Additional Pictures.

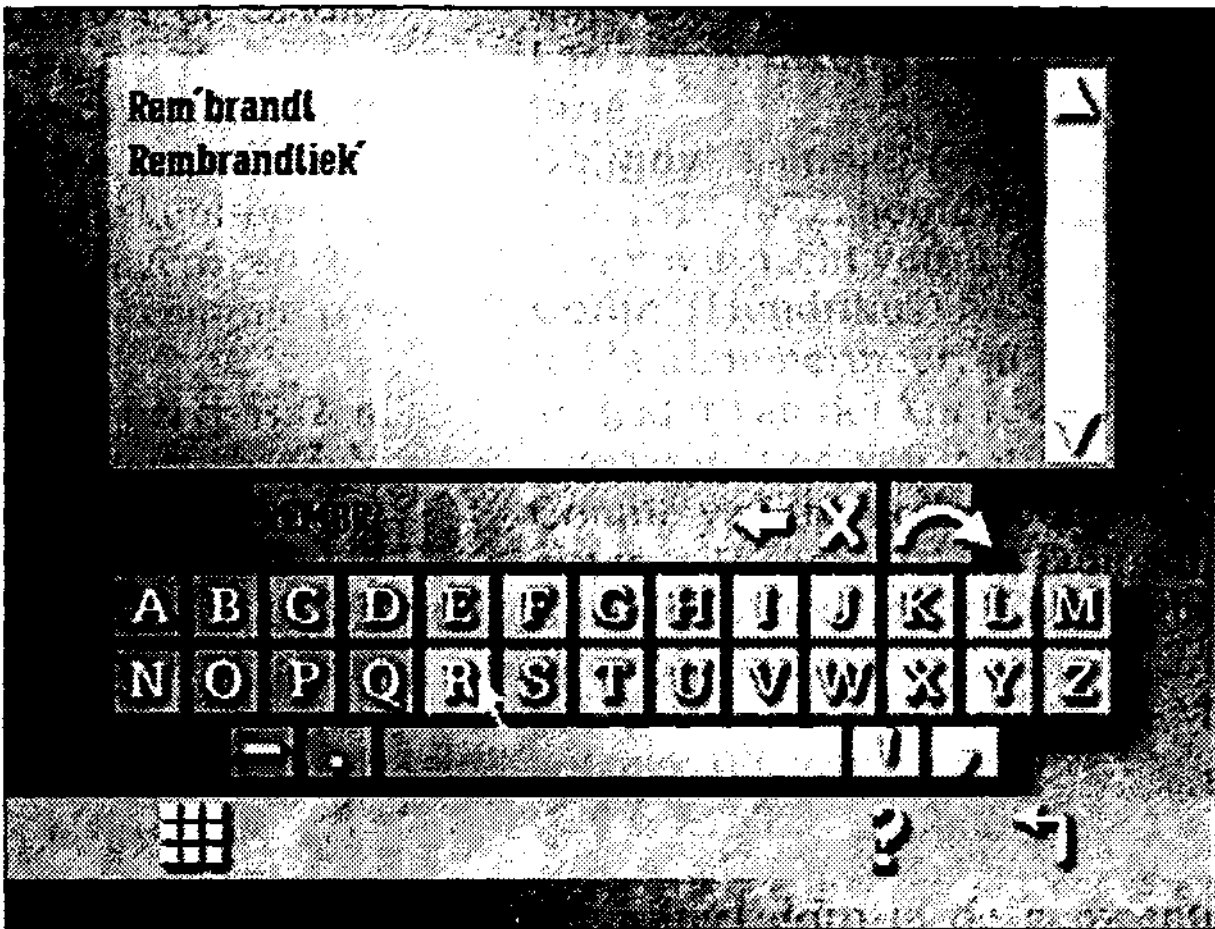
The tests are to be performed on the basis of a CD-i, titled the 'Philips Media Interactieve Encyclopedie', at the moment of this report released in the Dutch and Belgian markets. The following are pictures from the original interface (not as used in the tests).



The 'Philips Media Interactieve Encyclopedie'

The Main-Screen, the keyboards are selected by choosing 'A tot Z' (see picture underneath).





The 'Philips Media Interactieve Encyclopedie'

The original 2-Line Keyboard, the first of the two keyboards available in this title.



The 'Philips Media Interactieve Encyclopedie'

The Letter-Strip (1-Line) Keyboard, the second of the two original keyboards available in this title.

7. References

- The Psychology of Everyday Things

Norman, Donald; 1988
Basic Books, Inc., New York, N.Y., U.S.A.
(Translation in Dutch: 'Dictatuur van het Design', 1990)
(A.W. Bruna Uitgevers B.V., Utrecht)

This is not a specific book on the subject of keyboards or keyboard-design (except for a part of chapter 6), but an important book in general. In fact this book is often looked at as a 'popular' book on designing, just because it is very readable. Don't let yourself be kidded by the fact that it is possible to like a book on the subject. For me this was the book that made pieces of the puzzle began to fall in place. It made me understand that you (and I) could actually do something about it.

- Tog on Interface

Tognazzini, Bruce 'Tog'; 1991
Addison-Wesley Publishing Company, Inc.

This book is based on a series of columns ('Apple Direct'), with excerpts from papers, correspondence and never before published writings. Although Tog's official title at Apple seems to be 'Human Interface Evangelist', it could even be of more interest for non-Macintosh users. It is as good to read as it is to hear him talk, but now you have the chance to go back and reread!

- Why alphabetic keyboards are not easy to use:

Keyboard Layout doesn't much matter

Norman, D.A. & Fisher, D.; 1984
in: 'Human Factors', p. 509-519

This is the original article which 'Psychology of Everyday Things' refers to. In fact this is one of the few researches on the subject of Keyboard-design with some relation to our subject of On-Screen Keyboards.

- Assessment of trends in the technology and techniques of human-computer interaction

Marshall, Chris; Christie, Bruce & Gardiner, Margaret M.;
in: 'Applying Cognitive Psychology to User Interface Design';
ed.: Gardner, M. & Christie, B.

This chapter describes the input and output techniques and mention some research on keyboard-layouts, comparisons among others of a 'Standard' (probably QWERTY) and an Alphabetic keyboard, concluding that –for ten-finger use– there is no real improvement. This led to the remark (and I quote) 'All this implies that excessive concentration of effort on optimising keyboard designs is perhaps unwarranted in the light of the more global problems in human-machine interaction.'

The following List 'Literature' primarily consists of books and articles which either are about designs for ten-finger use as the 'Dvorak-design' or the various 'Chord-keyboards' (for instance the 'Velotype', the Bulgarian 'Isot-keyboard' or 'Stenokey' and the American Chord-keyboards), or one-hand keyboards as used in fighter jet-planes, which all require more or less 'heavy' training and go beyond the scope of this report, or either are about (Interaction) Designing and Usability Testing in general.

Literature

- The Timeless Way of Building
Alexander, Christopher; 1979
Oxford University Press
- The User Interface: Concepts & Design
Barfield, Lon; 1993
Addison-Wesley Publishing Company, Inc.
- Ergonomic Criteria for the Evaluation of Human-Computer Interfaces.
(*Critères Ergonomiques pour l'Evaluation d'Interfaces Utilisateurs*)
Bastien, J. M. Christian; Scapin, Dominique L.; 1993
Technical report No. 156, INRIA, Rocquencourt.
- Evaluation of Alternative Alphanumeric Keying Logics
Butterbaugh, Larry C.; 1982
in: 'Human Factors', p. 521-533
- User Interface Design
Cox, Kevin & Walker, David; 1993
Prentice Hall
- Task Analysis for Human-Computer Interaction
Diaper, Dan; 1989
Ellis Horwood Ltd., John Wiley & Sons
- Human-Computer Interaction
Dix, Alan a.o.; 1993
Prentice Hall
- The Chord Keyboard Report -
Writing Clear Text with the Speed of Speech
Dopping, Olof; 1988
Handikappinstitutet, Bromma, Sweden (*The Swedish Institute for the Handicapped*)
- Exploiting a New Home Market: Games on CD-i
Garcia, John; 1991
The 2nd Multimedia Conference on Interactive CD, Speaker's Papers
- Handbook of Human-Computer Interaction
Helander, Martin; 1988
North-Holland, Amsterdam
- Developing User Interfaces
Hix, Deborah a.o.; 1993
John Wiley & Sons Inc.
- The Art of Human-Computer Interface Design
Laurel, Brenda (ed.); 1990
Addison-Wesley Publishing Company, Inc.
- Computers as Theatre
Laurel, Brenda; 1993
Addison-Wesley Publishing Company, Inc.
- Taking the British Golf Museum Home
Lewis, Peter; 1991

The 2nd Multimedia Conference on Interactive CD, Speaker's Papers

- **Usability Engineering**
Nielsen, Jakob; 1993
Academic Press, Inc.; Harcourt Brace & Company, Publishers
- **Things that makes us Smart**
Norman, Donald A.; 1993
Addison-Wesley Publishing Company, Inc.
- **A Provisional Evaluation of a New Chord Keyboard, the Velotype**
Noorden, Leon P. A. S.;
in: 'Human-Computer Interaction Psychonomic Aspects',
ed.: Veen, G. C. v.d.; Mulder, G.;
- **Ergonomics and Information Technology**
Osborne, David J.;
in: 'Computers in the Human Context',
ed.: Forrester, T.
- **Children's Musical Theatre & Other CD-i Titles**
Ramo, John; 1991
The 2nd Multimedia Conference on Interactive CD, Speaker's Papers
- **Evaluating Usability of Human-Computer Interfaces:**
A Practical Method
Ravden, Susannah J.; Johnson, Graham I. ; 1993
Ellis Horwood Ltd., John Wiley & Sons.

*** * ***