

h	th	b	f	d	h, kh	ḍ	hw, khw
ḥ	nt	ḅ	mp	ḏ	nk, nc	ḥ	nkʷ, nqu, ñq (etc.)
ṁ	n	Ṃ	m	Ṃ	ñ	Ṃ	nw
ṛ	r	Ṛ	v	Ṛ	y	Ṛ	w
ʀ	r	ʀ	rd	ʀ	l	ʀ	ld
ᶑ	s	ᶑ	s	ᶑ	ss	ᶑ	ss
λ	h	ḏ		λ	-i	o	-u
l	h	l		l			

How does the program know if the 'n' your typing is a solitary 'n' or part of the cluster 'nt'? As soon as you type a 'n', a ṁ (númen) will appear on the screen. If a 't' follows, the númen is replaced with a ḥ (anto). The same thing happens when you type an 'r' or an 's'. First an ṛ (óre) or ᶑ (silme) appears, but if you follow with a vowel, the óre or silme will be replaced with a ʀ (rómen) or ʀ (silme nuquerna) and the appropriate tehta will be placed above it.

To type a 'ñ', just precede the 'n' with a '~' (tilde).

Aha, hyarmen and halla

When you type a 'h', normally a ḏ (aha) will appear. A λ (hyarmen) will be used at the beginning of a word, and a l (halla) will only be used if a word starts with 'hl' or 'hr'. (Note: unfortunately, the program can only recognize the beginning of a word if you just typed a space. So if it doesn't work properly, type a space and delete it later.)

Palatalized consonants and anna

If a consonant is followed by a 'y' (a so-called palatalized consonant), two dots will be placed under the consonant. Thus 'Quenya' results in ḑṛṛṁ. 'y' alone gives Ṛ.

Nwalme

'nw' will only be combined to Ṃ (nwalme) at the beginning of a word, otherwise the combination ṁṂ will be used.

Double consonants

If you type the same vowel twice, a bar will appear under the tengwa to indicate that this consonant is prolonged.

Vowels

Below is a table of the tehtar and how to type them:

ḏ	a	ḏ	e	ḏ	i	ḏ	o	ḏ	u
---	---	---	---	---	---	---	---	---	---

The grey tengwar characters are for demonstration purposes only. If you type a vowel, the tehta will appear above the preceding consonant. If there is no preceding consonant, the tehta will be placed over a short carrier. So, 'alda' will become ḏḏ.

Long vowels

Long vowel can be typed in two ways:

- by repeating the same vowel twice
- by preceding the vowel with a ' (the single quotation mark).

Thus the word *únótimē* can be typed as 'un'otime or as *uunootime*, which will both result in ᵎᵎᵎᵎ.

Diphthongues

The six Quenya diphthongues can be gotten simply by typing 'ai', 'oi', 'ui', 'au', 'eu' or 'iu'.

Other sequences of vowels

If you type a vowel sequence that is not a diphthongue, they will be treated as separate vowels. There is no type the *Umlaut*, which is just some (redundant) information for the English reader. So, the word 'laurië' can be typed as 'laurie', which will give τᵎᵎᵎᵎ.

Special stuff

Hook-s

If you use the 's' after another consonant, a hook will be used instead. This will only happen when this is the last tengwa in a word (so when there's nothing after the 's', or only a short vowel). Thus, 'otsos' becomes íᵎᵎᵎᵎ. The hook-s appears the moment you type a space or some interpunction character to signal the end of the word.

Interpunction

The following table shows the interpunction symbols and how to type them:

:	. or ; or ;	· ,	~ ...	§ !	ß ?
---	-------------	-----	-------	-----	-----

Numbers

If you type a number, you get the appropriate tengwar character. Advanced stuff like the duodecimal system, or reversing the digits, are not supported (yet).

Capitals

If you type a capital, nothing special happens. That means, it is treated as a normal, lowercase character.

Separating characters

If you do not wish to combine several latin characters into one tengwa, you can separate them by typing a '-' (dash) between them. So, if for some stange reason you want to write ᵎᵎᵎ rather than ᵎᵎᵎ, you can do so by typing 'n-k'.

You can also use the dash to force tehtar to use a carrier, to avoid the hook-s etc. It is as if you had typed a space, but no space will appear.

Unsupported characters

(This section will only make sense if you understand Dan Smith's font mapping.)

If you type a character that the program doesn't know what to do with, it will beep. If you want to type a special Tengwar character that the program doesn't support, you have to know where it is in Dan Smith's font mapping. To tell the program you want to input a character using Dan Smith's font mapping, you have to type a '\' (backslash)

first. So, if you want the ! , you have to type '\!' to get it. The '\' causes the program to return the next character without interpreting it.

(Note for LaTeX-fans: if you type '\\\' you get the ~, but this is pure coincidence.)

Bugs and comments

I am no expert on Quenya. If you find any bugs, please contact me.

The source code for this keyboard is in a file *something.kmn*, where *something* is the name of this keyboard. If you have any questions about the source, please contact me. You are free to elaborate on my program, but I would not like to see two incompatible implementations of the same mode, so I would appreciate to be notified. If you've written a new mode (and I like it), I will be happy to point to it on my website. The source code for the Keyman program itself is not available, unfortunately.

I am no native speaker of English. Unless you want to offer to rewrite this entire document, please don't bother me about it.

Contact

My name is Liesbeth Flobbe. You can reach me through email at flobbe@ai.rug.nl. The website where I put this package is www.ai.rug.nl/~flobbe/tengwar.

This document was written on December 9, 2001.