Grafmin

User Guide

Dear User,

HE purpose of this document is to give you some internal information about the software you will find at Grafmin, but a few important details are secret. Almost all technical trade-offs are described here in laconic nature to help you draw your own personal conclusions quickly in order to enjoy the software fully, and with all the flaws. I wish I could do better with some technical decisions to improve things but there is no immediate plan to do so.

Some software components are carefully picked from like-minded people, the idea is to instill some kind of singular personality in the software, for better or worse. I deeply believe that beautiful software is fast, does not contain design features that are just design features, and is economical with the computer resources even if there are plenty. The economy gives a sense of accomplishment of a job well done, in the most efficient way. Thus, Grafmin does not do flashy software.

Finally, I hope to find the software useful and have a good time with it, or even better if it helps you make your own things. I would be interested to know and see different types of creative expression with the tools provided at Grafmin.

Kind Regards, Giorgos Vougioukas

Technical Specification

Markella - Rhea - Thalia - Phoebe - Euterpe - Alcyone Mnemosyne - Terpsichore - RSI - KCm - WDL-GV

> Written: C++

Language standard: ISO/IEC 14882:1998

Mature/proven standard:

Software type: Industrial-strength software

C++ Standard Template Library: No. Specialized components

> Software orientation: Longevity

> > C++ Exceptions: No

RTTI: No

RAII: Yes

Raw pointers: Yes

Pointer arithmetic: Yes

Manual memory management:

Predictable memory allocations: Yes

Any form of garbage collection:

UTF-8 (variable-length) Character encoding:

Array indexing:

32-bit signed integer

Widget toolkit:

Windows API (low-level)

Input device:

Keyboard (primary), Mouse (optional),

Touch screen (fully ignored)

Computer graphics: Raster

> 2D graphics: LICE (part of Cockos WDL)

Linux/macOS: Binary size:

SWELL (part of Cockos WDL)

Memory footprint:

Superleggera Extremely low

C runtime (CRT):

Cold startup time:

Statically linked

Instant, like your instinct

Shared library dependencies:

From none to a few

Backwards compatibility:

At least 20 years

Design heredity:

WDL (whittle)

Source model:

Open source

Sightliness:

In the eye of the beholder

- ENDS -