

5.3.1 Gambas – Konsole

Diesem Kapitel werden die folgenden Überlegungen vorangestellt:

- Das Gambas-Projekt 'gui_gnuplot' wurde in der IDE von Gambas angelegt, entwickelt und getestet.
- Danach übernahm ein anderer Entwickler das Muster-Projekt 'gui_gnuplot'. Der Projekt-Ordner ist `$HOME/gui_gnuplot`.
- Nach den ersten Tests wurde der Quelltext in einem geeigneten Editor wie zum Beispiel 'Blue-Fish' punktuell geändert.
- Anschließend wird das geänderte Projekt – ohne die Nutzung der Gambas-IDE – so lange getestet, bis alle Anforderungen aus dem Pflichtenheft hinreichend erfüllt werden.

Nach jeder Änderung am Quelltext ist für einen Programm-Test zuerst das Programm neu zu kompilieren, dann eine ausführbare Programm-Datei zu generieren und abschließend das Programm neu zu starten. Die notwendigen Programme Compiler, Archiver und Interpreter sind bereits in den Kapiteln 5.0.1 und 5.0.2 vorgestellt worden.

5.3.1.1 Einsatz Compiler

Zum Kompilieren des Muster-Projekts werden alternativ folgende Befehle in der Konsole ausgeführt:

```
Syntax:    gbc3 [optionen] [<projekt-verzeichnis>]
```

```
hans@linux:~$ gbc3 $HOME/gui_gnuplot
OK
hans@linux:~$ gbc3 -all $HOME/gui_gnuplot
OK
hans@linux:~$ gbc3 -ag $HOME/gui_gnuplot
OK

hans@linux:~$ cd $HOME/gui_gnuplot
hans@linux:~/gui_gnuplot$ gbc3 --all
OK

hans@linux:~$ gbc3 -agv $HOME/gui_gnuplot # Umfangreiche Kommentare und Informationen
OK
hans@linux:~$ gbc3 -agv $HOME/gui_gnuplot > gui_gnuplot.log
```

Wenn das Kompilieren erfolgreich war, dann wird als Quittung `OK` ausgegeben und es befindet sich ein neuer, versteckter Ordner `.gambas` im Projekt-Ordner oder der Inhalt wird überschrieben. In diesem Ordner liegen Dateien mit den Dateinamen der Dateien im Ordner `.src` – hier `FMAIN` und `MTERMINAL` (in Großbuchstaben).

Wenn beim Kompilieren ein Fehler auftritt, wird ein Hinweis angezeigt mit einem Verweis auf die Zeile, in welcher der Fehler auftrat:

```
hans@linux:~$ gbc3 -ag $HOME/gui_gnuplot
FMain.class:58: error: Unknown identifier: En
hans@linux:~$
```

Hier war es ein simpler Schreibfehler: `En dif` statt `Endif`. Auch im Fehlerfall wird im Projektordner ein versteckter – aber leerer – Ordner `.gambas` angelegt. Das ist unkritisch, da der Inhalt des Ordners bei einem weiteren, fehlerfreien Durchlauf beim Kompilieren überschrieben wird.

5.3.1.2 Einsatz Archiver

Nur wenn das Projekt erfolgreich kompiliert wurde, wird in einem weiteren Schritt durch das Programm *Archiver* im Projekt-Ordner die ausführbare Datei `gui_gnuplot.gambas` erzeugt oder ein Update.

```
Syntax:    gba3 [optionen] [<projekt-verzeichnis>]
```

Es wird *keine* Quittung ausgegeben, wenn das Generieren der ausführbaren Datei (*Gambas executable file*) erfolgreich war. Die ausführbare Datei (Archiv-Datei) erhält als Datei-Namen den *Projektname*

men (Standard) und als Datei-Extension *gambas*. Mit diesen Befehlen generieren Sie eine ausführbare Datei:

```
hans@linux:~$ gba3 $HOME/gui_gnuplot
hans@linux:~$ cd $HOME/gui_gnuplot # Alternative
hans@linux:~/gui_gnuplot$ gba3
```

Wenn Sie der ausführbaren Datei einen *anderen* Dateinamen geben wollen, dann müssen Sie folgende Syntax benutzen:

```
hans@linux:~$ gba --output=ZIEL QUELLE
hans@linux:~$ gba3 --output=$HOME/gui_gnuplot/gnuplot44.gambas $HOME/gui_gnuplot
hans@linux:~$ gba3 -o $HOME/gui_gnuplot/gnuplot44.gambas $HOME/gui_gnuplot
hans@linux:~$ cd $HOME/gui_gnuplot # Alternative
hans@linux:~/gui_gnuplot$ gba3 -o $HOME/gui_gnuplot/gnuplot44.gambas
```

Exkurs

Setzen Sie die Option *-x*, dann können Sie sich den Inhalt ausgewählter Dateien im Gambas-Archiv anzeigen lassen – sofern die Datei im Gambas-Archiv vorhanden ist:

```
Syntax: gba3 -x <archiv-pfad> <datei>
```

Dieser Aufruf war erfolgreich:

```
hans@linux:~$ gba3 -x $HOME/gui_gnuplot/gui_gnuplot.gambas .project
# Gambas Project File 3.0
# Compiled with Gambas 3.3.4
Title=GnuPlot & Process
Startup=FMmain
Icon=Symbols/projekt_icon.png
Version=0.0.78
Component=gb.image
..
Authors="Dr. Hans Lehmann ~ Osterburg ~ 2012"
TabSize=2
SourcePath=/home/hans/Desktop
hans@linux:~$
```

Auch in diesem Fall ist das Ergebnis eindeutig:

```
hans@linux:~$ gba3 -x $HOME/gui_gnuplot/gui_gnuplot.gambas .settings
gba: file not found in archive
hans@linux:~$
```

5.3.1.3 Einsatz Interpreter

Zum Ausführen der Datei *projektname.gambas* oder *neu_name.gambas* benötigen Sie notwendigerweise den Interpreter *gbx3*, der auf dem System installiert sein muss. Die Archiv-Datei ist als *Skript* mit dem 'Magic Header' *#!/usr/bin/gbr3* markiert und zeigt Linux an, wie das Skript ausgeführt werden soll. Linux führt das Skript aus, indem der im Header angegebene Interpreter aufgerufen wird:

```
Syntax: gbx3 [optionen] [<projekt-datei>] [--<argumentliste: arg_1 arg_2 ... arg_k>]
```

Es wird standardmäßig nach *gui_gnuplot.gambas* gesucht und diese – falls vorhanden – durch den Interpreter gestartet; sonst muss der Pfad zur Archiv-Datei angegeben werden und der symbolische Link auf den Interpreter verwendet werden:

```
hans@linux:~$ gbx3 $HOME/gui_gnuplot ' Standard projektname.gambas
hans@linux:~$ gbr3 $HOME/gui_gnuplot/gnuplot44.gambas ' Geänderter Name der Archiv-Datei
```

Dem Start von Gambas-Programmen mit *Parametern* ist ein eigenes Kapitel: 5.3.3 'Programme mit Parametern' gewidmet.

5.3.1.4 Einsatz Compiler, Archiver und Interpreter

Hier finden Sie die Aktionen Kompilieren, Archivieren und Interpretieren für das Muster-Projekt im Verzeichnis `$HOME/gui_gnuplot` in einer erprobten Kurzfassung:

```
hans@linux:~$ gbc3 -ag $HOME/gui_gnuplot
OK
hans@linux:~$ gba3 $HOME/gui_gnuplot
hans@linux:~$ gbx3 $HOME/gui_gnuplot
```

Das Ergebnis kann sich sehen lassen:

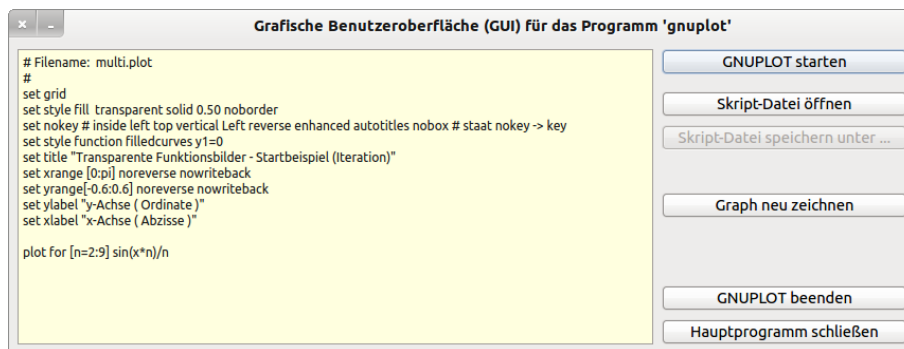


Abbildung 5.3.1.4.1: GUI für das Konsolen-Programm 'gnuplot'

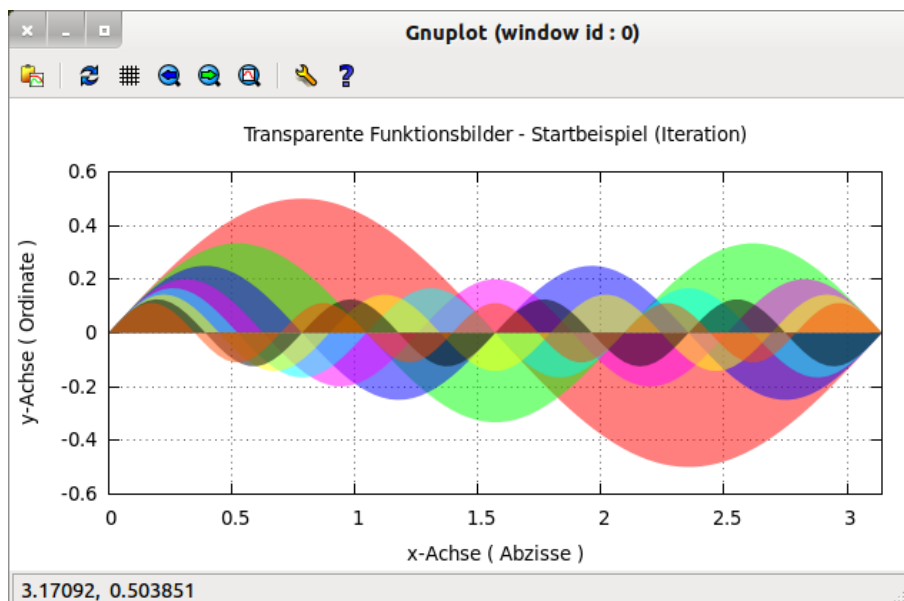


Abbildung 5.3.1.4.2: Das Konsolen-Programm 'gnuplot' in Aktion