

### 21.3.6 Projekt – GUI für das Konsolen-Programm GnuPlot

Es wird eine grafische Benutzeroberfläche (GUI) für das exzellente Konsolen-Programm 'gnuplot' im Zusammenhang mit der Verwendung der Instruktionen SHELL und EXEC vorgestellt.

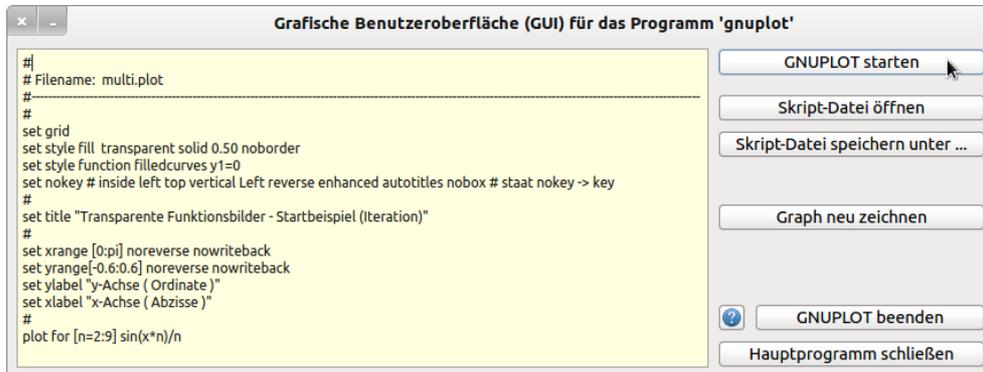


Abbildung 21.3.6.1: GUI für das Konsolen-Programm 'gnuplot'

Das Projekt ist vollständig ausgebaut und orientiert sich an der Lösung der Aufgabe, wie man eine grafische Benutzeroberfläche für ein (Konsolen-)Programm in Gambas entwickelt und kodiert.

Folgende Anforderungen sind umgesetzt:

- Entwicklung einer einfachen und übersichtlichen Programm-Oberfläche, die auf mehrjährigen Erfahrungen mit dem Programm 'gnuplot' basiert.
- Installation des zu verwendenden (Konsolen-)Programms 'gnuplot' unter Root-Rechten, wenn es nicht installiert ist.
- Aufruf einer elementaren Hilfe zum Programm 'gnuplot'.
- Sammlung von Skripten, die viele Bereich des Anwendungsspektrums von 'gnuplot' abdecken.

Wenn das Konsolen-Programm 'gnuplot' installiert ist, dann wird nach einem Klick auf den Button 'GNU PLOT starten' ein Prozess gestartet und das Start-Skript abgearbeitet. Es ergibt sich dann diese grafische Darstellung für das Start-Beispiel:

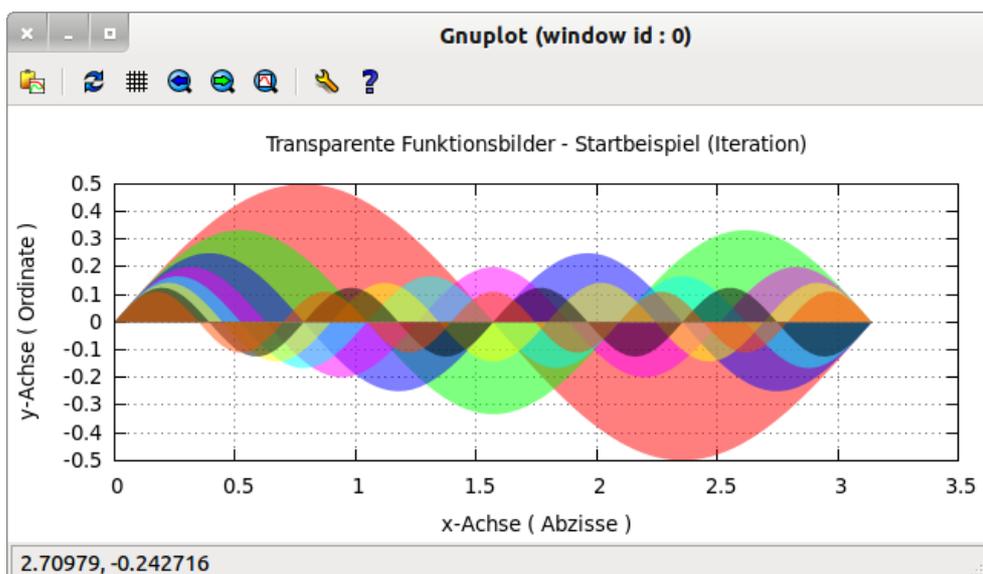


Abbildung 21.3.6.2: Konsolen-Programm 'gnuplot' in einem eigenen Fenster

Sie können anschließend Ihre eigenen GnuPlot-Skripte schreiben und abspeichern oder gespeicherte Skripte editieren und verändern. Schauen Sie auch in die mitgelieferte Sammlung von GnuPlot-Skripten im GnuPlot-Projekt nach, um sich Anregungen für eigene GnuPlot-Grafiken zu holen.

Interessant ist die Lösung der Teilaufgabe zur Installation des Konsolen-Programms 'gnuplot', das nicht zu den Standard-Programmen jeder Linux-Distribution gehört. Die Grundidee der Lösung ist der Start eines Programms das testet, ob das Programm 'gnuplot' installiert ist. Fällt der Test positiv aus, dann wird das Startprogramm beendet und es startet sofort die grafische Benutzeroberfläche (GUI). Im anderen Fall wird u.a. angeboten, das fehlende Programm 'gnuplot' zu installieren:

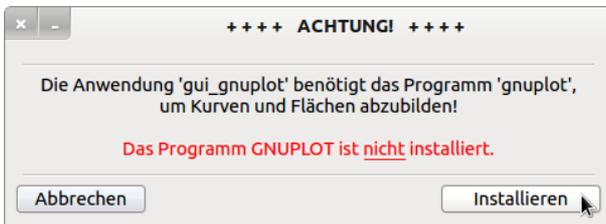


Abbildung 21.3.6.3: Option: Installation von 'gnuplot'

Entscheiden Sie sich für die Installation von 'gnuplot', dann müssen Sie noch das geforderte Passwort eingeben und die Installation wird angeschoben:

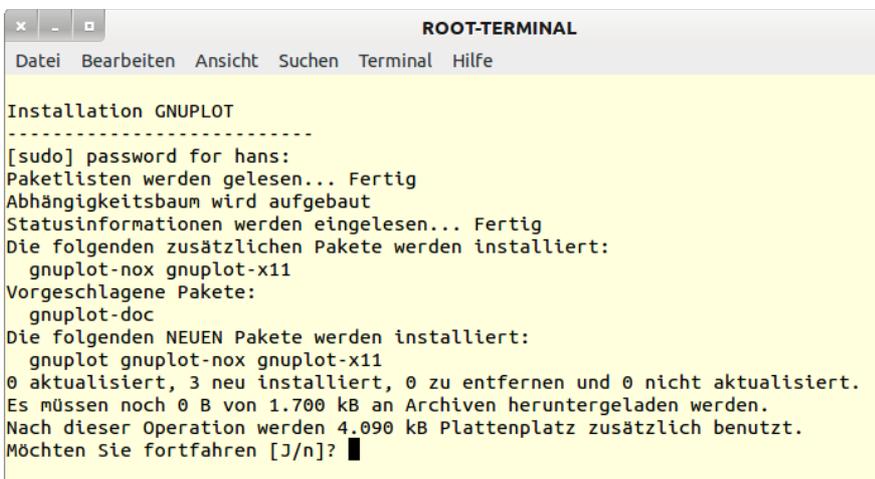


Abbildung 21.3.6.4: Installation von 'gnuplot'

Nach erfolgreicher Installation öffnet sich die GUI → Abbildung 21.3.6.1 und Sie können arbeiten. Der Quelltext für das Start-Programm wird vollständig angegeben und anschließend kurz kommentiert:

```
[1] ' Gambas class file
[2]
[3] Private sProgrammName As String
[4]
[5] Public Sub Form_Open()
[6]     Dim sText As String
[7]
[8]     FStart.Center
[9]     FStart.Resizable = False
[10]    FStart.Caption = "+ + + +  ACHTUNG!  + + + +"
[11]
[12] ' sPaketName = Lower("gnuplot") ' Option
[13] sProgrammName = Lower("gnuplot")
[14]
[15] If CheckProgram(sProgrammName) = True Then
[16]     Start()
[17] Else
[18]     sText = "<center>Die Anwendung '" & Application.Name & "' benötigt das Programm '" & \
[19]         sProgrammName & "',<br>"
[20]     sText &= "um Kurven und Flächen abzubilden!<br><br>"
[21]     sText &= "<font color='red'>"
[22]     sText &= "Das Programm '" & sProgrammName & "' ist <u>nicht</u> installiert."
[23]     sText &= "</font></center>"
[24]     txlHinweis.Text = sText
[25] Endif ' CheckProgram(sProgrammName) = True
[26]
[27] End ' Form_Open()
[28]
[29] Public Sub btnInstall_Click()
[30]     Dim sShellCommand, sMessage As String
```

```

[31]
[32] If GetTerminalList().Count > 0 Then
[33]     sShellCommand = "echo 'Installation " & sProgrammName & " ';"
[34]     sShellCommand &= "echo '-----';"
[35]     sShellCommand &= "sudo apt-get install " & sProgrammName
[36]     GoToTerminal(GetTerminalList()[0], sShellCommand)
[37]     Start()
[38] Else
[39]     sMessage = "Es wurde kein Terminal-Programm gefunden!<br>"
[40]     sMessage &= "Die Abbildung von Kurven und Flächen ist nicht möglich!<br><br>"
[41]     sMessage &= "Das Programm " & Application.Name & " wird beendet!"
[42]     Message.Warning(sMessage)
[43]     FStart.Close
[44] Endif ' GetTerminalList().Count > 0
[45] End ' btnInstall_Click()
[46]
[47] Public Sub btnCancel_Click()
[48]     Dim sMessage As String
[49]
[50]     sMessage = "<font color='red'>"
[51]     sMessage &= "Die Anwendung '" & Application.Name & "' wird beendet!"
[52]     sMessage &= "</font></center>"
[53]     Message.Warning(sMessage)
[54]     FStart.Close
[55] End ' btnCancel_Click()
[56] '*****
[57] Public Sub Start()
[58]     FStart.Delete ' Start-Programm
[59]     FMain.Show ' GUI
[60] End ' Start()
[61]
[62] Private Function GetTerminalList() As String[]
[63]     Dim hWhich As Process
[64]     Dim aCurrentTerminalList, aTerminals As String[]
[65]     Dim sTerminal As String
[66]
[67]     aCurrentTerminalList = New String[]
[68]     '          Ubuntu          Kubuntu          Xubuntu          Lubuntu          XSystem
[69]     aTerminals = ["gnome-terminal", "konsole", "xfce4-terminal", "lxterminal", "xterm"]
[70]     For Each sTerminal In aTerminals
[71]         hWhich = Exec ["which", sTerminal] Wait For Read
[72]         If hWhich.Value = 0 Then ' Terminal-Programm gefunden
[73]             aCurrentTerminalList.Add(sTerminal)
[74]         Endif ' hWhich.Value?
[75]     Next ' sTerminal
[76]
[77]     Return aCurrentTerminalList
[78]
[79] End ' GetTerminalList()
[80]
[81] Private Function CheckProgram(sProgram As String) As Boolean
[82]     Dim hWhich As Process
[83]
[84]     sProgrammName = Lower(sProgram)
[85]     hWhich = Exec ["which", sProgrammName] Wait For Read
[86]     If hWhich.Value = 0 Then Return True ' 0 = Das angegebene Programm ist installiert
[87]
[88] End ' CheckProgram(..)
[89]
[90] Public Sub GoToTerminal(sTerminal As String, sCommandLine As String)
[91]     Dim sShellCommand As String
[92]
[93]     sShellCommand = "echo; sudo -k; " ' Passworteingabe wird erzwungen!
[94]     sShellCommand &= sCommandLine
[95]     sShellCommand &= "; echo; echo 'Weiter mit ENTER oder CTRL+C !'; read dummy"
[96]     File.Save("/tmp/command.sh", sShellCommand)
[97]     Exec ["chmod", "+x", "/tmp/command.sh"]
[98]
[99]     Select Case Lower(sTerminal)
[100]         Case "gnome-terminal"
[101]             Shell sTerminal & " --geometry 77x12+100+100 -t 'ROOT-TERMINAL' -e " & "'/tmp/command.sh'" Wait
[102]         Case "konsole"
[103]             Shell sTerminal & " --geometry 75x10+100+100 -t 'ROOT-TERMINAL' -e " & "'/tmp/command.sh'" Wait
[104]         Case "xfce4-terminal"
[105]             Shell sTerminal & " --geometry 75x10+100+100 -T 'ROOT-TERMINAL' -e " & "'/tmp/command.sh'" Wait
[106]         Case "lxterminal"
[107]             Shell sTerminal & " --geometry 75x10+100+100 -T 'ROOT-TERMINAL' -e " & "'/tmp/command.sh'" Wait
[108]         Case "xterm"
[109]             Shell sTerminal & " -bg blue -fg white -cr gray -b 15 -bc -u8 -geometry 75x11+100+100 \
[110]                 -title 'ROOT-TERMINAL' -e " & "'/tmp/command.sh'" Wait
[111]         End Select ' sTerminal
[112]
[113] End ' GoToTerminal(sCommandLine AS String)

```

Kommentare:

- In den Zeilen 15 bis 25 wird geprüft, ob das Konsolen-Programm 'gnuplot' installiert ist. Entweder startet die GUI oder Sie sehen die Oberfläche des Start-Programms mit dem Angebot, das benötigte (Konsolen-)Programm zu installieren oder das Startprogramm zu beenden.
- Die Hauptlast der Installation von 'gnuplot' liegt im Quelltext in der Zeile 36.
- Die Prozedur *GoToTerminal(GetTerminalList()[0], sShellCommand)* in dieser Zeile startet die erste Konsole aus der generierten Liste der auf dem System gefundenen (Standard-)Terminalprogramme und übergibt dieser Konsole ein (temporäres) Shell-Skript.
- Das Shell-Skript enthält die Aufforderung das Benutzer-Passwort einzugeben und eine Folge von Kommandos, die danach unter Root-Rechten ausgeführt werden.

Die Verwendung eines Start-Programms im Sinne eines Vorschalt-Programms vor der GUI in der dargestellten Weise bewährt sich genau dann, wenn Sie aus einem Gambas-Programm heraus ein Programm unter Root-Rechten installieren wollen.

Die grafische Benutzeroberfläche startet – wie alle Beispiel-Programme in diesem Kapitel 21 – einen Prozess, auf den lesend und schreibend zugegriffen wird. Im folgenden Quelltext-Ausschnitt wird das deutlich. Eine Besonderheit kennzeichnet den hier gestarteten Prozess *hGnuPlot*, da alle vom Prozess ausgegebenen Daten (Daten oder Fehlermeldungen) nur über den Kanal 2 gelesen werden können:

```
Public hGnuPlot As Process ' hGnuPlot ist eine Prozess-Variabale (für das u.a. SHELL-Kommando)
..
Private Sub GnuPlotStart()
    GnuPlotStop
    hGnuPlot = Shell "gnuplot" For Read Write As "hGnuPlot"

    If Not IsNull(hGnuPlot) Then
        If hGnuPlot.State = hGnuPlot.Running Then
            Print #hGnuPlot, txaPlotCode.Text
            btnGnuPlotStop.Enabled = True
            btnRedrawGraph.Enabled = True
            btnOpenScriptFile.Enabled = True
        Endif ' hGnuPlot.State = 1?
    Endif ' NOT IsNull(hGnuPlot)?
End ' GnuPlotStart

Private Sub GnuPlotStop()
    If Not IsNull(hGnuPlot) Then
        If hGnuPlot.State = hGnuPlot.Running Then
            Print #hGnuPlot, "q" ' q = quit
            btnGnuPlotStop.Enabled = False
            btnRedrawGraph.Enabled = False
        Endif ' hGnuPlot.State = hGnuPlot.Running?
    Endif ' NOT IsNull(hGnuPlot)?

' Alternative: System-Aufruf:
' IF NOT IsNull(hGnuPlot) THEN
'     IF hGnuPlot.State = 1 THEN SHELL "kill -s 9 " & hGnuPlot.Id
' ENDIF
End ' GnuPlotStop()

Public Sub hGnuPlot_Error(sError_OR_Result As String)

    txaPlotCode.Height = 160
    txaErrorAndFit.Background = Color.RGB(255, 223, 223)
    txaErrorAndFit.Visible = True
    txaErrorAndFit.SetFocus
' Über den Kanal 2 kommen *alle* Ausgaben von GnuPlot!
    If InStr(sError_OR_Result, "Iteration 0") <> 0 Then
        sError_OR_Result = gb.NewLine & "|==== RESULTATE =====|" & \
            sError_OR_Result
        btnResultSave.Visible = True
    Endif ' InStr(sError_OR_Result, "Iteration 0") <> 0?
    txaErrorAndFit.Text &= sError_OR_Result ' Alternative: txaErrorAndFit.Insert(sError_OR_Result)
    Error.Clear
End ' hGnuPlot_Error

Public Sub hGnuPlot_Read()
' Hier kommen bei gnuplot *keine* Daten an.
' Nur über den Kanal 2 (Standard-Fehlerausgabe) kommen *alle* Ausgaben von GnuPlot!
End '

```