

GUI Creator – ein Tool zum Erstellen von graphischen Benutzeroberflächen mit sGUI

Was ist der GUI Creator?

Im Gegensatz zu einem GUI Builder mit graphischer Benutzeroberfläche beinhaltet der GUI Creator einen kleinen Editor, mit dem jedes Element (Menüeintrag, Button, Eingabefeld usw.) eines GUI durch eine Befehlszeile definiert wird. Diese enthält den Bezeichner des Elementes (d.h. den im Code zu verwendenden Variablen-Namen) und alle Parameter. Hierzu werden die für sGUI definierten Funktionen zur Erzeugung dieser Elemente verwendet (AddSimpleGadget usw.). Anschließend ergänzt der GUI Creator den noch fehlenden Code zu einem ersten lauffähigen Programm, mit dem das Design überprüft werden kann. Dieser Code muss dann noch um die gewünschten Funktionen ergänzt werden.

Mitgelieferte Dateien

Folgende Dateien sind im Downloadpaket enthalten:

- GUI Creator.bas = Sourcecode
- GUI-Creator.exe = Kompilierte Datei
- GUI-Creator.ini = Konfigurationsdatei
- Beispiel.bas = mit dem GUI Creator erzeugte Beispieldatei
- GUI Creator.png, Beispiel.png = Screenshots
- Dateien im Verzeichnis Inc: Include-Dateien für den GUI Creator
- Dateien im Verzeichnis Inc\sGUI: Include-Dateien aus dem Programmpaket sGUI

Die Dateien TUI.bas und SCROLL.bas im Verzeichnis Inc stellen eine Bibliothek für eine textbasierte Benutzeroberfläche (TUI) dar. Sie benötigen die Tastaturabfragecodes in Common.bi. Von diesen Dateien benötigt der GUI Creator nur einen Teil. Da sie jedoch bei mir querschnittlich für viele Programme verwendet werden, habe ich sie nicht gekürzt. Sie stehen dem Benutzer dieses Downloads selbstverständlich auch für andere Anwendungen zur freien Verfügung.

Die Datei Inc\GUI.bi ist eine gemeinsame Header-Datei für die die Dateien in sGUI. Ich habe die Dateien von sGUI mitgeliefert, damit man mal schnell mit dem GUI Creator etwas ausprobieren kann. Empfehlenswert ist jedoch der komplette Download (<http://www.freebasic-portal.de/downloads/sonstiges/sgui-192.html>), wegen der guten Dokumentation und der Beispielprogramme.

Konfiguration

In der Konfigurationsdatei GUI-Creator.ini befinden sich folgende Einträge:

```
C:\FreeBASIC\Prog\FB-Forum\GUI Creator\  
Inc\GUI.bi
```

Zeile 1 ist der Pfad zum Verzeichnis der FreeBASIC-Programmdateien.

Zeile 2 ist der Pfad zu GUI.bi.

Bevor der GUI Creator gestartet wird, müssen diese Einträge eventuell angepasst werden.

Arbeitsweise des GUI Creator

Die Arbeitsweise lässt sich am besten anhand des folgenden Screenshot erklären:



Nach dem Starten des Programms erscheint zunächst eine kurze Erklärung und ein Eingabefenster. In das Fenster müssen mit Hilfe der Befehle

- Menu(event; ...)
- Bezeichner = Menu(event, ..)
- DRAW STRING
- Bezeichner = AddSimpleGadget(event, ...)
- Bezeichner = AddStringGadget(event, ...)
- usw.

alle Elemente und Texte des GUI mit ihren Bezeichnern und Parametern definiert werden. Die Syntax der Befehle (Menu, AddSimpleGadget usw.) ist der Referenz zu sGUI zu entnehmen.

Wichtig:

Die Parameter müssen konkrete Werte enthalten (Zahlen bzw. Strings), keine Variablen! AddSimpleGadget(event, x, y, 80, h, txt) ist zum Beispiel nicht zulässig, sondern AddSimpleGadget(event, 20, 20, 80, 20, „Hallo!“). Denn da der GUI Creator die Klammerausdrücke nicht auswertet, würden bei der Verwendung von Variablen bei der Codeumwandlung diese nicht deklariert werden.

Ausnahmen:

AddListBox und TextViewBox. Hier muss gemäß sGUI-Referenz als letzter Parameter ein Array vom Typ STRING eingetragen werden, z.B. sa(). Beispiel:

AddListBox(event, 20, 20, 40, 10, sa()). Für diesen Parameter muss nach der Code-Konvertierung in die erzeugte FreeBASIC-Datei (siehe unten) „manuell“ eine DIM-Anweisung eingetragen werden.

Der Editor kann nur mit der Tastatur bedient werden, nicht mit der Maus. Außer den üblichen Editiertasten werden folgende Tasten unterstützt:

- | | |
|-----------------------------|---|
| - Cursorsteuerungstaste auf | zur vorigen Zeile |
| - Cursorsteuerungstaste ab | zur nächsten Zeile |
| - Pos1 | zum Zeilenanfang |
| - Ende | zum Ende des Eintrages in der aktuellen Zeile |
| - Strg Entf | aktuelle Zeile entfernen |
| - Strg Einfg | vor der aktuellen Zeile eine neue Zeile einfügen (nur wenn der Puffer des Editors (100 Einträge) noch nicht voll ist) |
| - Esc | Eingabe beenden |

Nach Beenden der Eingaben muss ein FreeBASIC-Dateiname (*.bas) eingegeben werden. Die eingegebenen Zeilen werden zu einem lauffähigen Code ergänzt welcher in dieser Datei gespeichert wird.

Wichtig:

Als erster Parameter für „Menu“ und „Add“-Funktionen ist wörtlich „event“ zu verwenden, da der Code ausschließlich „event“ als Bezeichner für den Eventhandler erzeugt.

Die (Teil)strings „Menu“, „Add“ und „event“ dürfen in den Bezeichnern nicht vorkommen.

Beispiel

Die im o.a. Screenshot eingegebenen Zeilen erzeugen den folgenden Code (die in den GUI Creator eingegebenen Zeilen sind grün hervorgehoben):

```
-----  
#INCLUDE ".Inc\GUI.bi"  
  
SCREENRES 80 * 8, 16 * 30, 32  
WIDTH 80, 30  
  
InitGFX  
  
DIM AS EventHandle PTR event  
event = CreateEventHandle  
  
DIM entry1 AS INTEGER  
DIM entry2 AS INTEGER  
DIM str1 AS Gadget PTR  
DIM button1 AS Gadget PTR  
  
Menu(event, 1, 0, 1, "Datei")  
entry1 = Menu(event, 1, 1, 1, "Oeffnen")  
entry2 = Menu(event, 1, 2, 1, "Schliessen")  
DRAW STRING (100, 100), "Geben Sie Ihren Namen ein:"  
str1 = AddStringGadget(event, 100, 120, 30, 60, "")  
button1 = AddSimpleGadget(event, 180, 150, 80, 20, "OK")  
  
MenuOn(event)  
GadgetOn(str1)  
GadgetOn(button1)  
  
DO  
  
    event -> xSleep(1)  
  
    IF event-> MENUMESSAGE THEN  
        SELECT CASE event-> MENUMESSAGE  
            CASE entry1  
                'Code ...'  
            CASE entry2  
                'Code ...'  
        END SELECT  
    END IF  
  
    IF event-> GADGETMESSAGE THEN  
        SELECT CASE event-> GADGETMESSAGE  
            CASE str1  
                'Code ...'  
            CASE button1
```

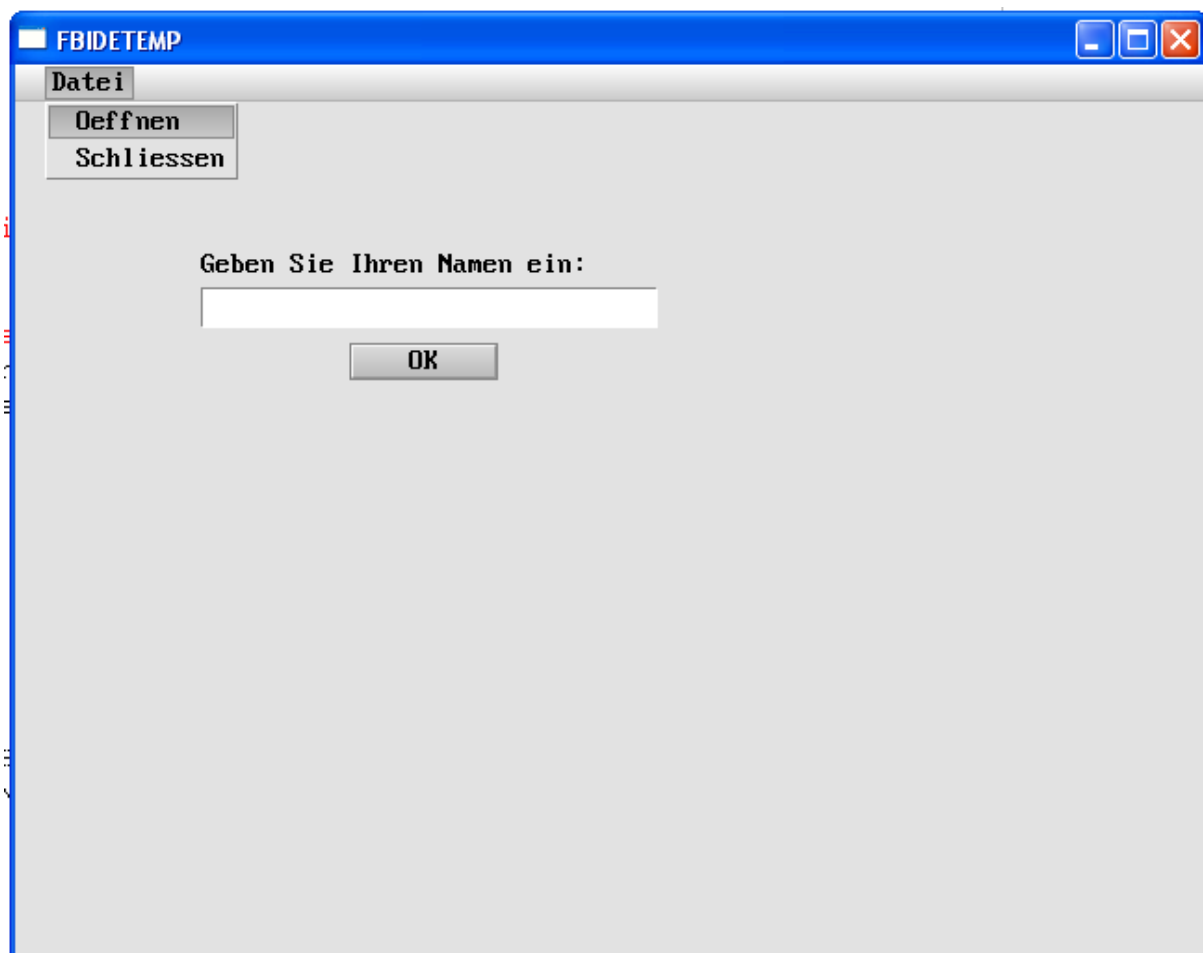
```
'Code ...'  
END SELECT  
END IF
```

```
LOOP UNTIL event-> EXITEVENT
```

```
DestroyEventHandle (event)
```

```
END
```

Startet man das Programm in der FreeBASIC-Entwicklungsumgebung, so erhält man folgendes Resultat:



Im Code lässt sich nun das Design des GUI oder die Fenstergröße ändern, und der Code kann mit den gewünschten Funktionen zum fertigen Programm ergänzt werden.

14.12.2011
Lothar Schirm