

### 12.2.3 Form – Transparente Fenster

Im Zusammenhang mit den Themen 'Transparente Fenster' und 'Fenster mit transparentem Hintergrund' liefert dieser Hinweis von Minisini wertvolle Informationen:

*... OK, it was a bit difficult, but I succeeded: in revision #6308, you get a new Window.Transparent property. By setting it, you make your window (or form) transparent, and then you can use a RGBA background color and/or a background picture with an alpha channel. Of course, that works only if you have a compositing window manager. Regards, Benoît Minisini*

Compositing-Fenster-Manager sind Fenster-Manager, die spezielle Effekte wie die Transparenz von Fenstern unterstützen. Die Antwort auf die Frage, ob ein Compositing-Fenster-Manager aktiv ist, liefert diese Zeile in einer Konsole:

```
xprop -id $(xprop -root -notype | awk '$1=="_NET_SUPPORTING_WM_CHECK:"{print $5}') -notype \
-f _NET_WM_NAME 8t | grep _NET_WM_NAME | sed 's/.*"(.*)" /\1/'
```

und zeigt unter Mint 17.3 LTS Chinnamon den Fenster-Manager an:

```
Mutter (Muffin)
```

Unter Ubuntu 12.04 ergab der folgende Befehl (→ <http://askubuntu.com/questions/72549/how-to-determine-which-window-manager-is-running>):

```
hans@linux:~$ wmcctl -m
Name: Metacity Class: N/A PID: N/A
Window manager's "showing the desktop" mode: OFF
```

auch eine eindeutige Antwort. Der Fenster-Manager ist Metacity, der jedoch nicht aktiviert war. Abhilfe schaffte das Aktivieren mit Hilfe des Konfigurationseditors 'gconf-editor' in den Einstellungen 'metacity' > general> *compositing-manager* = on → <https://wiki.ubuntuusers.de/metacity#Effekte>. Für Mint 17.3 muss das Programm 'wmcctl' (WindowsManagerControl) nach-installiert werden.

Die folgenden Projekte erzeugen jeweils ein voll transparentes Fenster oder ein Fenster mit transparentem Hintergrund. Ein voll transparentes Fenster ist ein Fenster mit den Eigenschaften Form.Transparent = True und Form.Border = False. Das bewirkt, dass nur noch der rahmenlose, transparente Hintergrund das aktive Fenster repräsentiert. Es bedarf eines alternativen Konzeptes, um voll transparente Fenster verschieben und schließen zu können. Ein Fenster mit einem transparenten Hintergrund und sichtbarer Titel-Zeile (Titel und Schaltflächen) wird erzeugt, wenn die Eigenschaften Form.Transparent = True und Form.Border = True gesetzt werden.

Mit diesem Quelltext sollte ein Fenster mit transparentem Fenster-Hintergrund in der (Hintergrund-) Farbe hell-grün erzeugt werden:

```
Public Sub Form_Open()
    FMain.Center()
    FMain.Transparent = True
    FMain.Border = True
    FMain.Background = &H0091B75B&
End
```

Hier das Ergebnis:



Abbildung 12.2.3.1: Alpha-Kanal 00 (0<sub>dez</sub>)

Transparenz sieht anders aus! Wie Sie sehen, hat der RGBA-Farb-Wert *&H0091B75B&* im o.a. Quelltext-Ausschnitt einen Alpha-Kanal von 0, was in Gambas als 'Null Transparenz' oder 'vollkommen undurchsichtig' zu interpretieren ist.

Änderungen des Wertes für den Alpha-Kanal ergaben diese Ergebnisse:

```

1 - Public Sub Form_Open()
2   FMain.Center()
3   FMain.Transparent = True
4   FMain.Border = True
5   FMain.Background = &H3F91B75B&
6   FMain.Text = "Transparenter Fenster-Hintergrund"
7 End ' Form_Open()
    
```

Abbildung 12.2.3.2: Alpha-Kanal 3F (63dez)

```

4 - Public Sub Form_Open()
5   FMain.Center()
6   FMain.Transparent = True
7   FMain.Border = True
8   FMain.Background = &HB091B75B&
9   FMain.Text = "Transparenter Fenster-Hintergrund"
10 End ' Form_Open()
11
    
```

Abbildung 12.2.3.3: Alpha-Kanal B0 (176dez)

```

4 - Public Sub Form_Open()
5   FMain.Center()
6   FMain.Transparent = True
7   FMain.Border = True
8   FMain.Background = &HFF91B75B&
9   FMain.Text = "Transparenter Fenster-Hintergrund"
10 End ' Form_Open()
11
    
```

Abbildung 12.2.3.4: Alpha-Kanal FF (255dez)

### 12.2.3.0.1 Projekt 1

Im Projekt SimpleClock wird der aktuelle Zeit-Text in einem einfachen Label auf voll transparentem Fenster-Hintergrund angezeigt. Die Entscheidung, den Fenster-Rahmen anzuzeigen, hat einzig praktische Gründe, weil das Fenster so ohne Aufwand verschoben und geschlossen werden kann:

```

21
22
23 - Public Sub Timer1_Timer()
24   lblShowTime.Text = Format(Now, "hh:mm:ss")
25 End ' Timer1_Timer()
26
27
28
    
```

Abbildung 12.2.3.5: Fenster mit voll transparentem Hintergrund

Dann reicht dieser Quelltext, um die aktuelle Zeit auf transparentem Fenster-Hintergrund anzuzeigen:

```

' Gambas class file

Public Sub Form_Open()

  FMain.Transparent = True
  FMain.Border = True
  FMain.Resizable = False
  FMain.Margin = True
  FMain.Arrangement = Arrange.Fill
  FMain.Center()

  lblShowTime.Alignment = Align.Center
  lblShowTime.Font = Font["FreeMono, Bold, +16"]

  Timer1.Enabled = True
  Timer1.Delay = 1000
  Timer1.Trigger()

End ' Form_Open()

Public Sub Timer1_Timer()
  lblShowTime.Text = Format(Now, "hh:mm:ss")
End ' Timer1_Timer()
    
```

Das Projekt SimpleClock finden Sie zusammen mit den anderen Projekten im Download-Bereich.

### 12.2.3.0.2 Projekt 2

Für das Projekt 'TransparentClock\_Digital' als Mini-Programm für den Desktop (Widget, Desklet oder GadGets) ist dieser Hinweis notwendig: Der Aufwand, den Zeit-Text mit Paint-Methoden zu zeichnen, ist nicht notwendig. Ursprünglich sollte das Layout für diese Digital-Uhr einen Rahmen mit abgerundeten Ecken erhalten. Daher fiel die Entscheidung für die gezeichnete Variante mit den Paint-Methoden.

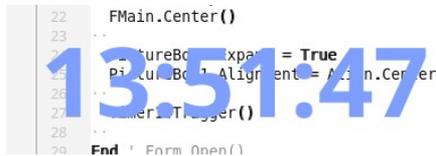


Abbildung 12.2.3.6: Transparentes Fenster

Interessanter als die Fragen: Wie verschiebt man ein voll transparentes Fenster auf dem Desktop? Wie beendet man ein transparentes Fenster? Wie informiert man den Nutzer über diese Möglichkeiten der Interaktion? sind die Antworten. Der Quelltext wird vollständig angegeben und anschließend kommentiert:

```
[1] ' Gambas class file
[2]
[3] Private $fFontSize As Float
[4] Private $MX As Integer
[5] Private $MY As Integer
[6]
[7] Public Sub Form_Open()
[8]
[9]     $fFontSize = 64
[10]
[11]     FMain.Font = Font[$fFontSize]
[12]     FMain.Arrangement = Arrange.Fill
[13]     FMain.Margin = True
[14]     FMain.Padding = 21
[15]     FMain.Resize(FMain.Font.TextWidth("00:00:00") + 64, FMain.Font.Height + 16)
[16]     FMain.Resizable = False
[17]     FMain.Text = "Was - schon so spät ?"
[18]     FMain.Background = &HFFFFFFF
[19]     FMain.Opacity = 100
[20]     FMain.Border = False
[21]     FMain.Transparent = True
[22]     FMain.Tooltip = "Verschieben mit gedrückter LMT\nSchließen über Kontextmenü (RMT)"
[23]     FMain.Center()
[24]
[25]     PictureBox1.Expand = True
[26]     PictureBox1.Alignment = Align.Center
[27]
[28]     Timer1.Trigger()
[29]
[30] End ' Form_Open()
[31]
[32] Public Sub Timer1_Timer()
[33]     PaintDigitalClock($fFontSize)
[34] End ' Timer1_Timer()
[35]
[36] Private Sub PaintDigitalClock(FontSize As Float)
[37]
[38]     Dim hPicture As Picture
[39]     Dim sText As String
[40]
[41]     sText = Str(Time)
[42]
[43] ' Das Picture hat die transparente Maske, wenn der optionale Parameter 'Transparent' den Wert True hat
[44] ' LINK: https://sourceforge.net/p/gambas/mailman/message/32766778/
[45]     hPicture = New Picture(FMain.Width, FMain.Height, True)
[46]
[47]     Paint.Begin(hPicture)
[48]         Paint.Font.Size = FontSize
[49]         Paint.Font.Bold = True
[50]         Paint.Brush = Paint.Color(&H7F9FFF)
[51]         Paint.DrawText(sText, 0, 0, hPicture.W, hPicture.H, Align.Center)
[52]     Paint.End()
[53]
[54]     FMain.PictureBox1.Picture = hPicture
[55]
[56] End ' PaintDigitalClock()
```

```

[57]
[58] Public Sub PictureBox1_DblClick()
[59]     FMain.Close()
[60] End ' PictureBox1_DblClick()
[61]
[62] Public Sub PictureBox1_MouseDown()
[63]
[64]     Dim hPopupMenu, hMenuItem As Menu
[65]
[66]     $MX = Mouse.ScreenX - Me.X
[67]     $MY = Mouse.ScreenY - Me.Y
[68]     FMain.Background = &H00E2E2E2
[69]
[70]     If Mouse.Right = True Then
[71]         ' Ein PictureBox-Kontextmenü erzeugen
[72]         hPopupMenu = New Menu(Me) As "mnuPopupMenu"
[73]         hMenuItem = New Menu(hPopupMenu) As "mnuExit"
[74]         hMenuItem.Text = "Exit"
[75]         hMenuItem.Picture = Picture["icon:/16/quit"]
[76]         hPopupMenu.Popup()
[77]     Endif
[78]
[79] End ' PictureBox1_MouseDown()
[80]
[81] Public Sub PictureBox1_Mouseup()
[82]     FMain.Background = &HEEEEEFFF
[83] End ' PictureBox1_Mouseup()
[84]
[85] Public Sub PictureBox1_MouseMove()
[86]     If Mouse.Left Then Me.Move(Mouse.ScreenX - $MX, Mouse.ScreenY - $MY)
[87] End ' PictureBox1_MouseMove()
[88]
[89] Public Sub mnuExit_Click()
[90]     Me.Close()
[91] End ' mnuExit_Click()
[92]
[93] ' Public Sub Form_MouseDown()
[94] '     $MX = Mouse.ScreenX - Me.X
[95] '     $MY = Mouse.ScreenY - Me.Y
[96] '     FMain.Background = &H00C3DDFF
[97] ' End
[98]
[99] ' Public Sub Form_MouseUp()
[100] '     FMain.Background = &HEEEEEFFF
[101] ' End
[102]
[103] ' Public Sub Form_MouseMove()
[104] '     If Mouse.Left Then Me.Move(Mouse.ScreenX - $MX, Mouse.ScreenY - $MY)
[105] ' End

```

Kommentar:

- In den Zeilen 20 und 21 wird ein voll transparentes Programm-Fenster deklariert.
- Mit dem Tooltip in der folgenden Zeile wird der Nutzer informiert, wie man das Fenster verschiebt und wie es beendet werden kann.
- In den Zeilen 81 bis 87 wird die Verschiebung des voll transparentes Programm-Fensters realisiert.
- Während der Verschiebung wird der Fenster-Hintergrund hellgrau eingefärbt, um die Fenster-Position temporär anzuzeigen.
- Eine Alternative zur realisierten Verschiebung finden Sie ab Zeile 93.
- Das Schließen des Fensters wird über einen Doppelklick (Zeilen 58 bis 60) oder über das Kontext-Menü der fenster-füllenden *PictureBox* in den Zeilen 62 bis 77 und 89 bis 91 realisiert.



Abbildung 12.2.3.7: Markiertes transparentes Fenster mit Kontext-Menü

### 12.2.3.0.3 Projekt 3

Das Projekt *TransparentClock\_Analog* von Claus Dietrich zeigt eine analoge Uhr, deren Design Sie über ein Kontext-Menü verändern können. Das Menü enthält auch den Menü-Eintrag 'Schließen'.



Abbildung 12.2.3.8: Analoge Uhr mit Kontext-Menü

Sie können das voll transparente Programmfenster mit der Maus auf dem Desktop *frei* verschieben:



Abbildung 12.2.3.9: Analoge Uhr als Desklet

Die Einstellung für das verwendete *Design* wird beim Schließen des transparenten Programmfensters in einer Settings-Datei gespeichert und steht bei einem weiteren Start als Vorgabewert für das Design zur Verfügung.

In der readme-Datei zu diesem Desklet finden Sie neben dem Hinweis, wie Sie für dieses Programm einen Starter auf dem Desktop anlegen können auch die Information, dass die Zifferblätter der Uhren Modifikationen von freien Grafiken sind, deren Quellen angegeben werden.