

## 23.3.5.5 Bild-Schatten

Im Zentrum dieses Projektes steht die Anzeige eines Bildes, dem ein Bild-Schatten hinzugefügt wird. Das Original-Bild sollte bestimmte Voraussetzungen – wie einen transparenten Hintergrund – erfüllen, damit die Zugabe des Bildschattens einen guten Effekt liefert.



Abbildung 23.3.5.5.1: Original-Bild



Abbildung 23.3.5.5.2: Original-Bild mit Schatten

Die Schatten-Wirkung zeigt sich am deutlichsten, wenn Sie das Projekt starten und sich beide Bilder nacheinander ansehen.

Das erste Projekt verwendet das (Teil-)Programm 'convert' aus der Programm-Sammlung 'ImageMagick', das daher notwendigerweise auf Ihrem System installiert sein muss. Das Programm und dessen Dokumentation können Sie unter Ubuntu über diese Anweisung installieren:

```
sudo apt-get install imagemagick imagemagick-doc
```

Wollen Sie das externe Programm nicht einsetzen, dann existiert im zweiten Abschnitt noch ein weiteres Projekt, das nur Methoden von Gambas verwendet.

Quelltext 1:

```
[1] ' Gambas class file
[2]
[3] Public Sub Form_Open()
[4]     Me.Center
[5]     Me.Resizable = False
[6]     picBox.Picture = Picture[Application.Path & "/logo.png"]
[7]     If Exist(Application.Path & "/logos.png") Then
[8]         Kill Application.Path & "/logos.png"
[9]     Endif
[10]    btnGenerateShadow.Text = "Schatten ein ..."
[11] End ' Form_Open()
[12]
[13] Public Sub btnGenerateShadow_Click()
[14]     Dim sCommand As String
[15]
[16]     ' Das Programm 'convert' muss auf dem System installiert sein
[17]     sCommand = "convert -background gray -shadow 80x3 " & Application.path & "/logo.png "
[18]         & Application.path & "/logos.png"
[19]     Shell sCommand Wait
[20]     MakeShadow
[21] End ' btnGenerateShadow_Click()
[22]
[23] Public Sub MakeShadow()
```

```
[24] Dim Foreground, Shadow, hImage As Image
[25]
[26] Shadow = Image.Load(Application.path & "/logos.png")
[27] Foreground = Image.Load(Application.path & "/logo.png")
[28]
[29] If btnGenerateShadow.Text = "Schatten ein ..." Then
[30]     hImage = Shadow.Copy()
[31]     hImage.PaintImage(Foreground, 0, 0)
[32]     btnGenerateShadow.Text = "Schatten aus ..."
[33] Else
[34]     hImage = Foreground.Copy()
[35]     btnGenerateShadow.Text = "Schatten ein ..."
[36] Endif
[37]
[38] pictureBox.Picture = hImage.Picture
[39]
[40] End ' MakeShadow()
```

Kommentar:

- Zum Programm-Start wird das Original-Bild in einer PictureBox angezeigt und ein existierendes Schatten-Bild gelöscht, wenn es im angegebenen Pfad existiert → Zeilen 6 bis 9.
- In den Zeilen 17 und 18 wird vom Original-Bild mit Hilfe des Programms 'convert' ein Schattenbild erzeugt und im Pfad der Original-Bild-Datei in einer Bild-Datei 'logos.png' abgespeichert.
- In der Prozedur 'MakeShadow' wird in Abhängigkeit vom Text auf dem Button *btnGenerateShadow* entweder das Original-Bild angezeigt oder ein drittes temporäres Bild (hImage vom Typ Image) aus Original-Bild und Schattenbild erzeugt und dann angezeigt.
- Das temporäre Bild entsteht, indem (Zeile 31) auf das Schatten-Bild das Original-Bild gezeichnet wird, wie man den Zeilen 26 bis 32 entnehmen kann.

In der folgenden Variante von Tobias Boege und Ingo Beckert kommen nur Methoden von Gambas zum Einsatz, um den gleichen Schatten-Effekt zu erzeugen.



Abbildungen 23.3.5.5.3: Original-Bild und Bild mit Schatten

Auch der Quelltext 2 für das *alternative Projekt* wird hier vollständig angegeben:

```
[1] ' Gambas class file
[2]
[3] Public Sub Form_Open()
[4]     Me.Center
[5]     Me.Resizable = False
[6]     pictureBox.Picture = Picture[Application.Path & "/logo.png"]
[7]     btnGenerateShadow.Text = "Schatten ein ..."
[8] End ' Form_Open()
[9]
[10] Public Sub btnGenerateShadow_Click()
[11]     MakeShadow
[12] End ' btnGenerateShadow_Click()
[13]
[14] Public Sub MakeShadow()
[15]     Dim Foreground, hImage As Image
[16]
[17]     Foreground = Image.Load(Application.path & "/logo.png")
[18]
[19]     If btnGenerateShadow.Text = "Schatten ein ..." Then
[20]         hImage = Foreground.Copy()
[21]         hImage.PaintImage(Foreground, 0, 0)
[22]         pictureBox.Picture = AddShadow(pictureBox.Picture.Image).Picture
[23]         btnGenerateShadow.Text = "Schatten aus ..."
[24]     Else
[25]         hImage = Foreground.Copy()
```

```
[26]     picBox.Picture = hImage.Picture
[27]     btnGenerateShadow.Text = "Schatten ein ..."
[28] Endif
[29]
[30] End ' MakeShadow()
[31]
[32] Public Function AddShadow(hImage As Image) As Image
[33]     Dim hNewImage As New Image(hImage.W, hImage.H)
[34]
[35]     Paint.Begin(hNewImage)
[36]     Paint.DrawImage(Shadow(hImage), 4, 4)
[37]     Paint.DrawImage(hImage, 0, 0)
[38]     Paint.End()
[39]     Return hNewImage
[40]
[41] End ' AddShadow(hImage As Image) As Image
[42]
[43] Public Function Shadow(hImage As Image) As Image
[44]     Dim hShadow As Image = hImage.Copy()
[45]
[46]     hShadow.Fuzzy(3)
[47]     hShadow.Colorize(Color.Gray)
[48]     hShadow.Opacity(0.5)
[49]     Return hShadow
[50]
[51] End ' Shadow(hImage As Image) As Image
```