

### 23.4.7 Komponente ScrollArea (gb.qt4)

Wenn Sie in einem Gambas-Programm Bilder anzeigen wollen, dann werden Sie bestrebt sein, eine möglichst große Anzeigefläche zur Verfügung zu stellen. Ist das nicht möglich, dann kann der Einsatz der Komponente *ScrollArea* dieses Manko mildern, weil Sie sich dann das Bild in Ausschnitten ansehen können. Über die Methode *Image.Stretch(..)* integrieren Sie komfortabel eine Zoom-Funktion für das anzuzeigende Bild.

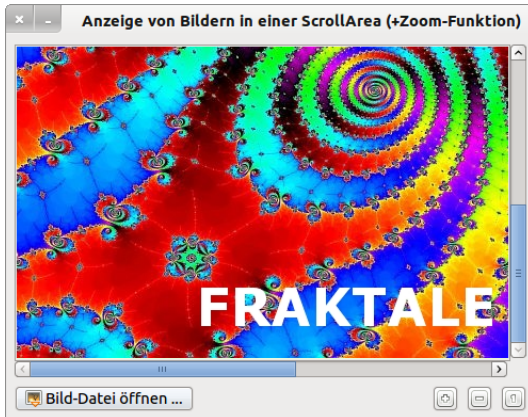


Abbildung 23.4.7.1: Bildanzeige in einer ScrollArea mit Zoomfunktion

Sowohl für den horizontalen und als auch für den vertikalen Scroll-Balken einer ScrollArea existieren Kontextmenüs, die vor allem im Zusammenhang mit der Eigenschaft *ScrollArea.Tablet* interessant werden können:

Hierher scrollen	Hierher scrollen
Linker Rand	Anfang
Rechter Rand	Ende
Eine Seite nach links	Eine Seite nach oben
Eine Seite nach rechts	Eine Seite nach unten
Nach links scrollen	Nach oben scrollen
Nach rechts scrollen	Nach unten scrollen

Abbildung 23.4.7.2: Kontext-Menü für die Scroll-Balken einer ScrollArea

#### 23.4.7.1 Eigenschaften, Methoden und Ereignisse der Komponente ScrollArea

Die Komponente *ScrollArea* (gb.qt4) ist eine *DrawingArea* mit Scroll-Balken. Um dieses Steuerelement zu verwenden, müssen Sie die Größe des Inhalts (Bild-Größe) mit der Methode *.ResizeContents(..)* definieren und im Draw-Event-Handler den Inhalt zeichnen.

Im nächsten Abschnitt werden nur ausgewählte Eigenschaften, Methoden und Ereignisse (Events) der Komponente vorgestellt und beschrieben.

Eigenschaft	Type	Beschreibung
<i>.ScrollBar</i>	Integer	Über die Konstanten None (0), Horizontal (1), Vertical (2) und Both (3) geben Sie an, ob keine, je eine oder je nach Bedarf eine oder zwei Scroll-Balken angezeigt werden sollen. Sie können die Anzahl und Art der angezeigten Scroll-Balken auch auslesen.
<i>.Tablet</i>	Boolean	Mit True wird festgelegt, dass Tablet-Events wie entsprechende Maus-Events behandelt werden (Ab Version 3.4.0) oder ermittelt, ob diese Einstellung aktuell gilt.
<i>.ScrollX</i>	Integer	Liest die horizontale Scroll-Position aus oder setzt sie.
<i>.ScrollY</i>	Integer	Liest die vertikale Scroll-Position aus oder setzt sie.

Tabelle 23.4.7.1.1: Ausgewählte Eigenschaften – ScrollArea

Methode	Beschreibung
Sub ResizeContents ( W As Integer, H As Integer )	Definiert die Größe des Inhalts der ScrollArea
Sub Scroll ( X As Integer, Y As Integer )	Scrollt den Inhalt an die gegebene Position

Tabelle 23.4.7.1.2: Ausgewählte Methoden – ScrollArea

Ereignisse	Beschreibung
Event Scroll ( )	Das Ereignis wird ausgelöst, wenn ein Scroll-Balken der ScrollArea aktiviert wurde.
Event Draw ( )	Das Ereignis wird ausgelöst, wenn etwas neu gezeichnet werden muss.

Tabelle 23.4.7.1.3: Ausgewählte Ereignisse – ScrollArea

### 23.4.7.2 Projekt

In Projekt werden die ausgewählten Eigenschaften, Methoden und Ereignisse eingesetzt und eine Zoom-Funktion integriert. Als Inhalt können nur Bilder geladen werden und angezeigt werden.

Der Quelltext wird vollständig angegeben:

```
[1] ' Gambas class file
[2]
[3] Private imgUploaded As Image
[4] Private imgDisplay As Image
[5] Private fImageZoom As Float
[6]
[7] Public Sub Form_Open()
[8]
[9]     FMain.Center
[10]    FMain.Resizable = False
[11]    lblImageResolution.AutoSize = True
[12]    lblImageResolution.Expand = False
[13]    lblImageResolution.Visible = False
[14]    pSpace.Expand = True ' variabler Zwischenraum
[15]    imgUploaded = Image.Load("Images/fraktal.jpg") ' Lokales Startbild
[16]    imgDisplay = imgUploaded
[17]    ScrollAreal.ResizeContents(imgDisplay.W, imgDisplay.H)
[18]    ScrollAreal.Scroll(0, ScrollAreal.H + 8)
[19]
[20]    fImageZoom = 1
[21]
[22] End ' Form_Open()
[23]
[24] Public Sub btnOpenImage_Click()
[25]
[26]    Dialog.Title = "Wählen Sie ein Bild aus..."
[27]    Dialog.Filter = ["*.png;*.jpg;*.jpeg;*.gif;*.xpm", "Bild-Dateien: "]
[28]    If Dialog.OpenFile() Then Return
[29]
[30]    Try imgUploaded = Image.Load(Dialog.Path)
[31]    If Error Then
[32]        Message.Error("Fehler beim Öffnen der Bild-Datei!")
[33]        Return
[34]    Endif ' ERROR ?
[35]    ScrollAreal.Scroll(0, 0)
[36]    imgDisplay = imgUploaded
[37]    ScrollAreal.ResizeContents(imgDisplay.W, imgDisplay.H)
[38]    ScrollAreal.Refresh
[39]    Wait
[40]
[41]    lblImageResolution.Visible = True
[42]    lblImageResolution.Text = "Bildgröße: " & imgUploaded.W & " x " & imgUploaded.H & " Pixel"
[43]
[44] End ' btnOpenImage_Click()
[45]
[46] Public Sub ScrollAreal_Draw()
[47]     If imgDisplay Then
[48]         Draw.Image(imgDisplay, - ScrollAreal.ScrollX, - ScrollAreal.ScrollY)
[49]     Endif ' imgDisplay ?
[50] End ' ScrollAreal_Draw()
[51]
[52] Public Sub btnZoomUp_Click()
```

```

[53] If fImageZoom < 3.01 Then
[54]     fImageZoom = fImageZoom + 0.05
[55] Endif ' fImageZoom < 3.01 ?
[56] Display()
[57] End ' btnZoomUp_Click()
[58]
[59] Public Sub btnZoomDown_Click()
[60] If fImageZoom > 0.09 Then
[61]     fImageZoom = fImageZoom - 0.05
[62] Endif ' fImageZoom > 0.09 ?
[63] Display()
[64] End ' btnZoomDown_Click()
[65]
[66] Public Sub btnOriginal_Click()
[67]     fImageZoom = 1
[68] Display()
[69] End ' btnOriginal_Click()
[70]
[71] Private Sub Display()
[72]     imgDisplay = imgUploaded.Stretch(Round(imgUploaded.W * fImageZoom, 0), Round(imgUploaded.H *
fImageZoom, 0))
[73]     lblImageResolution.Text = "Bildgröße: " & Round(imgUploaded.W * fImageZoom, 0) & " x " &
Round(imgUploaded.H * fImageZoom, 0) & " Pixel"
[74]     ScrollAreal.ResizeContents(imgDisplay.W, imgDisplay.H)
[75]     ScrollAreal.Refresh
[76]     Wait
[77] End ' Display()

```

## Hinweise:

- Zeile 13 – Die Anzeige der Bildauflösung zum beim Startbild unterdrückt.
- Die Größe des Inhalts der ScrollArea wird auf die Bildgröße des geladenen Bildes gesetzt.
- In einem Datei-Öffnen-Dialog – Zeilen 24 bis 44 – kann ein Bild in vorgegebenen Formaten ausgewählt und geladen werden. Eine Fehlerbehandlung schließt sich an, wenn ein Fehler beim Laden des Bildes auftrat und abgefangen wurde.
- Die Zoom-Faktoren werden in den Zeilen 53-55, 60-62 auf hinreichend praktikable Werte eingeschränkt und in der Zeile auf Originalgröße gesetzt.
- Die Hauptlast trägt die Prozedur in den Zeilen von 46 bis 50. Es wird der gewählte *Bildausschnitt* stets neu gezeichnet, wenn ein Scroll-Balken bewegt wurde. Der Koordinatenursprung O(xo,yo) verschiebt sich nach links oben; dadurch sehen Sie im *Bildausschnitt* mehr vom rechten unteren Bild.
- Die Prozedur Display() ab Zeile 71 verändert die Bildgröße mit dem aktuellen Zoom-Faktor, zeigt die Auflösung an und ändert die Größe des Inhalts der ScrollArea auf die gezoomte Bildgröße. Der Einsatz der Methode *ScrollArea.Refresh* erfordert das Wait-Kommando, wenn die Änderungen in der Ansicht sofort ausgeführt werden sollen.