

### 17.17 ScrollBar – Regler

Die ScrollBar-Komponente bietet einen vertikalen oder horizontalen Regler. Der Regler ist vertikal ausgerichtet, wenn seine Höhe größer ist als seine Breite; ansonsten ist er horizontal. Mit einer ScrollBar können Sie Zahlenwerte (Typ Integer) in einem bestimmten Definitionsbereich [  $Z_{min}$  |  $Z_{max}$  ] generieren.

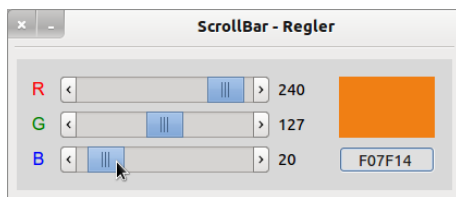


Abbildung 17.17.1: Auswahl der Farbanteile (RGB) mit einer ScrollBar

Der minimale Wert steht stets links bei einem horizontalen Regler und unten bei einer vertikalen Version. Eine ScrollBar besitzt keine Skale und die aktuellen Werte werden nicht angezeigt. Diese Aufgabe muss der Entwickler in geeigneter Weise bearbeiten.

#### 17.17.1 Eigenschaften ScrollBar

Eigenschaften der Komponente ScrollBar werden in der folgenden Tabelle beschrieben:

ScrollBar	Datentyp	Default	Beschreibung
MinValue	Integer	1	Legt den minimalen Wert fest, der in der ScrollBar erzeugt wird oder gibt diesen Wert zurück.
MaxValue	Integer	100	Legt den maximalen Wert fest, der in der ScrollBar erzeugt wird oder gibt diesen Wert zurück.
Step	Integer	1	Ändert den Wert der erzeugten ganzen Zahl (→ ScrollBar.Value) um den Wert von Step, solange die ScrollBar den Fokus besitzt.
PageStep	Integer	10	Ändert den Wert der erzeugten ganzen Zahl (→ ScrollBar.Value) um den Wert von PageStep.
Value	Integer	-	Legt den Reglerwert fest oder gibt die generierte ganze Zahl zurück.
Tracking	Boolean	False	Zeigt an, dass die ScrollBar das Change-Event kontinuierlich emittiert, wenn der Reglerknopf bewegt wird.

Tabelle 17.17.1.1: ScrollBar-Eigenschaften

#### 17.17.2 Ereignisse ScrollBar

Die Komponente *ScrollBar* hat nur ein *spezifisches* Ereignis: Change.

- Das Change-Ereignis wird jedes Mal ausgelöst, wenn sich der Wert der ScrollBar ändert.
- Der Wert ändert sich, wenn Sie auf die 2 (aktiven) Pfeiltasten klicken (Step), auf die Reglerbahn klicken (PageStep) oder wenn Sie den Reglerknopf an seine neue Position ziehen. Ein schöner Effekt stellt sich ein, wenn Sie auf die Reglerbahn klicken und dabei die Maus gedrückt halten. Mit dem Mausrad ändern Sie den Wert mit der (internen) Schrittweite 3.

#### 17.17.3 Projekt ScrollBar

Im folgenden Abschnitt wird ein Projekt vorgestellt, das die generierten Werte von drei ScrollBars in einem Farbwert als Kombination speichert und mit diesem Farbwert ein Panel einfärbt. Die Farbanteile RGB werden dezimal angezeigt und die Farbe hexadezimal. Sie können auch die Farbe als hexadezimalen Farbwert eingeben und mit Enter die Farbe übernehmen. Bei fehlerhaften Farbwerten wird *keine* Fehlermeldung ausgegeben, jedoch die Farbe intern auf *rot* gesetzt.

Eine Besonderheit finden Sie nicht im Quelltext, weil die Festlegung der Eigenschaft *Group* für alle drei Scrollbars nur in der IDE vorgenommen werden kann! Da sich die Farbe für das Panel aus den drei Farbanteilen (rot, grün und blau (RGB)) zusammensetzt, bot sich die Festlegung einer Farbgruppe an, so dass nur *eine Ereignisbehandlungsroutine* (RGB\_Change()) geschrieben werden musste, wenn sich bei einer der drei ScrollBars der Wert ändert. Die Abbildung 17.17.1 zeigt das Programmfenster für das vorgestellte Projekt, dessen Quelltext vollständig angegeben wird:

```
' Gambas class file

Public Sub _new()

    scbColorRed.MinValue = 0
    scbColorRed.MaxValue = 255
    scbColorRed.PageStep = 10
    scbColorRed.Step = 1
    scbColorRed.Value = 127

    scbColorGreen.MinValue = 0
    scbColorGreen.MaxValue = 255
    scbColorGreen.PageStep = 10
    scbColorGreen.Step = 1
    scbColorGreen.Value = 127

    scbColorBlue.MinValue = 0
    scbColorBlue.MaxValue = 255
    scbColorBlue.PageStep = 10
    scbColorBlue.Step = 1
    scbColorBlue.Value = 127

End ' _new()

Public Sub Form_Open()

    FMain.Center
    FMain.Resizable = False
    txbColorHex.MaxLength = 6
    RGB_Change()

End ' Form_Open()

Public Sub RGB_Change()
    SetPanelColor()
    SetRGBLabel()
    txbColorHex.Text = Hex$(Color.RGB(scbColorRed.Value, scbColorGreen.Value, scbColorBlue.Value), 6)
End ' RGB_Change()

Public Sub txbColorHex_Activate()
    Try panColor.Background = Val("&H" & txbColorHex.Text & "&")
    If Error Then
        panColor.Background = Color.Red
        txbColorHex.Text = Hex$(Color.Red, 6)
        SetRGB(panColor.Background)
    Else
        SetRGB(panColor.Background)
    Endif ' ERROR ?
End ' txbColorHex_Activate()

Public Sub SetPanelColor()
    panColor.Background = Color.RGB(scbColorRed.Value, scbColorGreen.Value, scbColorBlue.Value)
    panColor.Refresh
    Wait
End ' SetColor()

Private Sub SetRGB(iColor As Integer)
    Object.Lock(scbColorRed)
    scbColorRed.Value = Color[iColor].Red
    Object.Unlock(scbColorRed)
    Object.Lock(scbColorGreen)
    scbColorGreen.Value = Color[iColor].Green
    Object.Unlock(scbColorGreen)
    Object.Lock(scbColorBlue)
    scbColorBlue.Value = Color[iColor].Blue
    Object.Unlock(scbColorBlue)
    SetRGBLabel()
End ' SetRGB(aColor As Integer)

Public Sub SetRGBLabel()
    lblValueRed.Text = Str(scbColorRed.Value)
    lblValueGreen.Text = Str(scbColorGreen.Value)
    lblValueBlue.Text = Str(scbColorBlue.Value)
End ' SetRGBLabel()

Public Sub btnClose_Click()
    FMain.Close
End ' btnClose_Click()
```