

16.19 Klasse Spinner

Die Klasse Spinner (gb.form) implementiert ein Steuerelement, das eine Animation in zwei Varianten zeigt:



Abbildung 16.19.1: Spinner-Variante mit aktivierter Fortschrittsanzeige (Gambas 3.7.1 und Gambas > 3.8.0)

16.19.1 Eigenschaften

Die Klasse *Spinner* verfügt u.a. über diese relevanten Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Border	Boolean	Gibt an, ob ein Rahmen gesetzt werden soll oder nicht. Verwenden Sie die Border-Klasse für eine Liste von Konstanten.
Enabled	Boolean	Zeigt an, ob das Steuerelement aktiviert ist oder nicht.
Label	Boolean	Gibt an, ob ein Fortschrittswert angezeigt werden soll oder nicht.
Value	Float	Gibt einen Fortschrittswert zwischen 0 (0%) und 1.0 (100%) zurück. Sie können den Wert auch innerhalb des Wertebereichs [0 1] setzen. Der Fortschrittswert wird nur dann angezeigt, wenn die Label-Eigenschaft auf True gesetzt ist.

Tabelle 16.19.1.1 : Eigenschaften der Klasse Spinner

16.19.2 Methoden

Für den Einsatz der Klasse Spinner sind vor allem die ersten beiden Methoden bedeutsam:

Methode	Beschreibung
Start	Startet die Animation.
Stop	Stoppt die Animation.
Wait	Ruft die Event-Schleife auf und stellt sicher, dass die Event-Schleife nicht öfter als alle 200 Millisekunden aufgerufen wird.

Tabelle 16.19.2.1 : Methoden der Klasse Spinner

16.19.3 Projekt

Das Projekt *ArchivDownloadGIT* übernimmt die Programm-Idee aus dem Projekt *FileDownload* im → Kapitel 17.1, bei dem die Archivdatei (Gambas-Quelltexte) in einer bestimmten Version von einem Server in ein bestimmtes Verzeichnis geladen und anschließend entpackt wird. Zur Anzeige des Download-Fortschritts der Archivdatei auf <https://gitlab.com/gambas/gambas/repository/stable/archive.tar.bz2> wird die *Progressbar* durch einen *Spinner* ersetzt.



Abbildung 16.19.3.1: Download der Archivdatei vom Server (GB 3.10.0)

Für den Download der Archivdatei wird das Programm *wget* eingesetzt. Die Ausgaben des Programms *wget* werden über eine geeignete Prozess-Variable *hWGETProcess As Process* ausgewertet.

```
Public Sub btnGetArchivFile_Click()

    CreateDirectory(txbArchivDirectory.Text)

    If txbCodeSourceURL.Text Begins "https://" Then
        sArchivFileName = File.Name(Replace(txbCodeSourceURL.Text, "https://", ""))
    Else
        sArchivFileName = File.Name(txbCodeSourceURL.Text)
    Endif

    txaMessages.Clear()
    txaMessages.Insert(gb.NewLine)
    txaMessages.Insert("Quelle: " & gb.Tab & txbCodeSourceURL.Text & gb.NewLine)
    txaMessages.Insert("Ziel: " & gb.Tab & txbArchivDirectory.Text & / sArchivFileName & gb.NewLine)
    txaMessages.Insert(gb.NewLine)

    GetArchiv(txbCodeSourceURL.Text, txbArchivDirectory.Text & / sArchivFileName)

    txaMessages.Insert(Subst("&l '&2' &3", ("The archive"), sArchivFileName, ("will be unpacked!")) & "\n")
    Wait 2
    UnPackArchiv(txbArchivDirectory.Text & / sArchivFileName)

End ' btnGetFile_Click()
```

Diese Ausgaben liefern im Error-Ereignis *hWGETProcess_Error(sOutput As String)* neben Beginn und Ende des Downloads auch den Download-Fortschritt sowie die Größe des Archivs auf dem Server, denn alle Ausgaben von Programm *wget* kommen über die Standard-Fehler-Ausgabe:

```
Public Sub hWGETProcess_Error(sOutput As String)

    Dim aMatrix, aListe As String[]
    Dim sElement As String
    Dim iSizeMB As Float

    aMatrix = Split(sOutput, gb.newline)

    For Each sElement In aMatrix
        If sElement Ends "200 OK" Then
            txaMessages.Insert(("The connection to the server was successfully established. "))
            txaMessages.Insert(gb.NewLine)
        Endif ' sElement Ends "200 OK" ?
        If sElement Begins "Länge:" Or sElement Begins "Length:" Then
            aListe = Split(sElement, " ")
            iArchivSize = CInteger(aListe[1])
            iSizeMB = Round(iArchivSize / 1000000) * 0.953674316, -1)
            txaMessages.Insert(("Archive size = ") & Str(iArchivSize) & " Byte » (" & Str(iSizeMB) & " MiB )")
            txaMessages.Insert(gb.NewLine & "Download aktiv ..." & gb.NewLine)
            SizeTimer.Start()
            Spinner1.Label = True
            Spinner1.Start()
        Endif ' sElement Begins "Länge:" ?

        If sElement Like "*gespeichert*" Or sElement Like "*saved*" Then
            txaMessages.Insert(("Download finished!") & gb.NewLine)
            SizeTimer.Stop()
            Spinner1.Stop()
            Spinner1.Visible = False
        Endif ' LIKE ?
    Next ' sElement

End ' hWGETProcess_Error(..)
```

Während des *Downloads der Archivdatei* soll im Spinner der Download-Fortschritt (in Prozent) angezeigt werden. Dazu benötigen Sie einerseits – vor dem Download – die Größe der Archivdatei auf dem Server und andererseits Angaben, wie viel Prozent der Archivdatei bereits geladen worden sind oder die aktuelle Größe der lokalen Archivdatei. Im Projekt wird die 2. Variante mit einem Timer umgesetzt:

```
Public Sub SizeTimer_Timer()

    Dim iCurrentFileSize As Integer

    iCurrentFileSize = Stat(txbArchivDirectory.Text & / sArchivFileName).Size
    Spinner1.Value = iCurrentFileSize / iArchivSize

End ' Timer1_Timer()
```

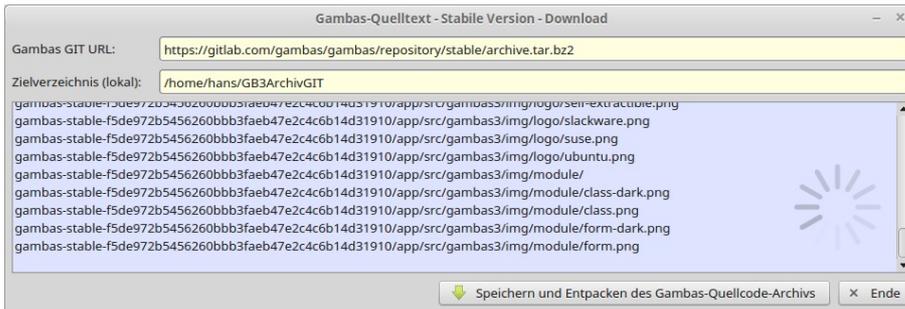


Abbildung 16.19.3.2: Entpacken der Archivdatei ohne Fortschrittsanzeige im Spinner

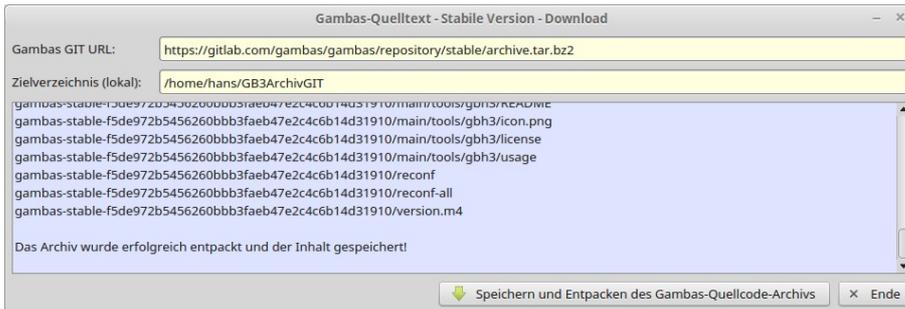


Abbildung 16.19.3.3: Ziel erreicht: Das Archiv wurde entpackt und gespeichert