

18.11.0 Klasse DrawingArea

Dieses Steuerelement implementiert eine Zeichenfläche, auf die von der Anwendung gezeichnet werden kann. Es hat zwei Zeichen-Modi, Standard und Cached, die durch die Cached-Eigenschaft definiert werden.

Im Standard-Modus löst das DrawingArea das Draw-Ereignis jedes Mal aus, wenn es aktualisiert werden muss. Im Cached-Modus wird das Zeichnen einmal ausgeführt. Die aktuelle Zeichnung wird in einem internen Bild gespeichert, das von der DrawingArea verwendet wird, um sich selbst zu aktualisieren. Anders formuliert: Im gecacheten Modus kann man sie als Paint-Device benutzen, das heißt man kann direkt auf ihr zeichnen, wann immer es benötigt wird – so wie in diesem Beispiel:

```
Public Sub DrawingAreaCanvas_Draw()
    Paint.Begin(DrawingAreaCanvas)
        DrawBarChart()
    Paint.End()
End
```

Im nicht-gecacheten Modus dagegen entscheiden andere interne Instanzen, wann eine DrawingArea neu gezeichnet werden muss. Das ist zum Beispiel der Fall, wenn sie durch ein Fenster überdeckt war. Dann muss sie neu gezeichnet werden, sobald sie wieder sichtbar ist. In solchem Fall wird das Draw-Event der DrawingArea aufgerufen. Ein Quelltext wie der o.a. würde zu einem Fehler führen.

So erzeugen Sie ein neues DrawingArea-Steuerelement:

```
Dim hDrawingArea As DrawingArea
hDrawingArea = New DrawingArea ( Parent As Container ) [ As "EventName" ]
```

18.11.0.1 Eigenschaften

Die Klasse DrawingArea verfügt über die folgenden relevanten Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Arrangement	Integer	Gibt die gesetzte Anordnung untergeordneter Steuerelemente zurück oder legt fest, wie dieser Container seine untergeordneten Steuerelemente anordnet. Wenn die Eigenschaft auf Arrangement.None gesetzt ist, so werden die Eigenschaften Margin, Padding und Centered ignoriert.
AutoSize	Boolean	Gibt an, dass die Größe des Containers automatisch an die Größe seines Inhalts angepasst wird.
Background	Integer	Gibt die vom Steuerelement verwendete Hintergrundfarbe zurück oder setzt diese.
Border	Integer	Liefert oder setzt den Rahmentyp des Steuerelements. Eine Liste der Konstanten, die mit dieser Eigenschaft verwendet werden können, finden Sie in der Klasse 'Border'.
Cached	Boolean	Liest oder setzt, ob der Inhalt der DrawingArea in einem internen Picture-Objekt zwischengespeichert wird.
Centered	Boolean	Seit Version 3.16: Zentriert die Steuerelemente je nach Anordnung des Containers horizontal oder vertikal. Diese Eigenschaft wird nur berücksichtigt, wenn der Container eine horizontale oder vertikale Anordnung verwendet.
Enabled	Boolean	Gibt zurück oder legt fest, ob der Zeichenbereich aktiviert ist. Wenn der Zeichenbereich deaktiviert ist, wird X-Window angewiesen, keine Ereignisse an den Zeichenbereich zu senden. Verwenden Sie diese Funktion, um ein anderes X-Window-Fenster in den Zeichenbereich einzufügen.
Focus	Boolean	Gibt zurück oder legt fest, ob das Steuerelement Tastaturreignisse (Keyboard Events) und den Fokus (Focus) akzeptiert.
Indent	Boolean	Gibt zurück oder legt fest, ob die untergeordneten Steuerelemente des Containers eingerückt sind. Die Größe der Einrückung ist Desktop.Scale. Die Richtung der Einrückung entspricht der Richtung der Anordnung des Containers.
Invert	Boolean	Gibt zurück oder legt fest, ob die horizontale oder vertikale Anordnungsrichtung invertiert ist. Wenn Sie diese Eigenschaft festlegen, wird zum Beispiel aus einer

Eigenschaft	Datentyp	Beschreibung
		horizontalen Ausrichtung 'von links nach rechts' eine horizontale Ausrichtung 'von rechts nach links'.
Margin	Boolean	Gibt zurück oder legt fest, ob zwischen dem Rand des Containers und allen untergeordneten Elementen ein Rand vorhanden ist. Die Größe des Randes ist Desktop.Scale Pixel. Wenn Sie auch die Eigenschaft Padding definieren, entspricht die Größe des Randes dem Wert von Padding.
NoBackground	Boolean	Liest oder setzt, ob die DrawingArea ihren Hintergrund automatisch zeichnet. Sie können diese Eigenschaft setzen, wenn Sie wissen, dass Sie alle Pixel der DrawingArea zeichnen werden. In diesem Fall müssen Sie keinen Hintergrund zeichnen, der dann wiederhergestellt wird.
NoMouse	Boolean	Seit Version 3.19: Liest oder setzt, ob das Steuerelement für Mausereignisse transparent ist.
Padding	Integer	Liest oder setzt die Anzahl der Pixel, die als Abstand zwischen untergeordneten Steuerelementen oder als innerer Rand des Containers verwendet werden. Wenn diese Eigenschaft nicht definiert ist (d.h. gleich 0), so ist der Abstand der Wert von Desktop.Scale in Pixel.
Painted	Boolean	Seit Version 3.4: Diese Eigenschaft gilt seit Gambas 3.4 als veraltet und sollte nicht mehr verwendet werden.
Spacing	Boolean	Liest oder setzt den Abstand zwischen den Unterelementen des Containers. Der Abstand entspricht dem Wert von Desktop.Scale in Pixel. Wenn Sie auch die Eigenschaft Padding setzen, entspricht der Abstand dem Wert von Padding.
Tablet	Boolean	Seit Version 3.3: Legt fest, ob die DrawingArea Tablet-Ereignisse als MouseEreignisse empfängt, oder gibt sie zurück.

Tabelle 18.11.0.1.1 : Eigenschaften der Klasse DrawingArea

Hinweise

- In der Arrange-Klasse existiert eine Liste der Konstanten, die mit dem Attribut Arrangement verwendet werden können → <https://gambas-buch.de/doku.php?id=k20:k20.7:start>.
- Wenn die Eigenschaft 'Cached' gesetzt ist, wird der Inhalt der DrawingArea nicht gelöscht, wenn sie von einem anderen Fenster verdeckt wird. Sie wird nur durch die explizite Verwendung der Clear()-Methode geleert.
- Die Existenz der Eigenschaft 'Focus' ist ein Grund, warum die DrawingArea gern für Steuerelemente verwendet wird.
- Wenn die Eigenschaft NoMouse gesetzt ist, dann deaktiviert diese Eigenschaft die Weiterleitung von Mausereignissen an das Steuerelement und seine untergeordneten Steuerelemente. Mausereignisse werden an andere Steuerelemente weitergeleitet, als ob das Steuerelement und seine untergeordneten Steuerelemente nicht in der Hierarchie der Steuerelemente vorhanden wären. Mausclicks und andere Ereignisse werden nicht beachtet.

18.11.0.2 Methoden

Die Klasse DrawingArea verfügt nur über zwei relevante Methoden:

Methode	Beschreibung
Clear()	Löscht die Zeichenfläche. Wenn die Zeichenfläche gepuffert ist, wird ihr interner Puffer mit der Hintergrundfarbe gefüllt.
Refresh ([X As Integer, Y As Integer, Width As Integer, Height As Integer])	Löst ein Neuzeichnen des Steuerelements aus. Beachten Sie: Das Neuzeichnen wird jedoch zurückgestellt und erst beim nächsten Aufruf der Ereignisschleife ausgeführt. Wenn Sie eine sofortige Aktualisierung benötigen, so rufen Sie die Instruktion WAIT auf, nachdem Sie die Refresh()-Methode verwendet haben.

Tabelle 18.11.0.2.1 : Methode der Klasse DrawingArea

18.11.0.3 Ereignisse

Die Klasse DrawingArea verfügt über diese Ereignisse:

Ereignis	Beschreibung
Change ()	Dieses Ereignis wird ausgelöst, wenn eine der folgenden Eigenschaften geändert wurde: Application.Animations oder Application.Shadows.
Draw ()	Wird ausgelöst, wenn etwas neu gezeichnet werden soll. Vor dem Ereignis wird automatisch Draw.Begin() auf der Zeichenfläche aufgerufen und die Zeichnung auf den neu zu zeichnenden Bereich beschränkt. Draw.End() wird <u>automatisch</u> nach dem Ereignis aufgerufen.
Font ()	Dieses Ereignis wird ausgelöst, wenn sich die Schriftart der DrawingArea geändert hat.

Tabelle 18.11.0.3.1 : Ereignisse der Klasse DrawingArea

- Generell gilt: Außerhalb des Draw()-Ereignisses kann nicht in eine DrawingArea gezeichnet werden und das Draw()-Ereignis sollte nur Paint-Zeichenanweisungen ausführen.
- Gut zu wissen, dass innerhalb des Draw()-Ereignisses keine Anweisungen ausgeführt werden, welche die DrawingArea als Steuerelement modifizieren. Alles, was zu einer Modifikation der DrawingArea führt, sollten Sie mit anderen Methoden festlegen.
- Das Font()-Ereignis ist besonders dann nützlich, wenn Sie ein benutzerdefiniertes Steuerelement implementieren, da dieses Ereignis auch ausgelöst wird, wenn die Standardschriftart des Desktops global geändert wird. Es ermöglicht Ihnen, Ihr Steuerelement entsprechend der neuen Schriftart neu zu entwerfen.

In den folgenden Kapiteln werden Projekte beschrieben, die auf unterschiedliche Art die DrawingArea als Zeichenfläche nutzen.