

25.1.10 Zeichnen mit Cairo

25.1.10.1 Zeichenflächen

Auf diese vier Zeichenflächen können Sie mit Cairo zeichnen:

- Image
- CairoPdfSurface
- CairoPsSurface
- CairoSvgSurface.

25.1.10.2 Geometrische Formen

Die folgenden geometrischen Formen (Linien oder Flächen oder Text) können Sie auf die Zeichenfläche zeichnen:

1. Punkt
2. Linie – Strecke als kürzeste Verbindung zweier Punkte
3. Rechteck (Spezialfall Quadrat),
4. Kreis – Kreis-Bogen – Kreis-Sektor / Ellipse – Ellipsen-Bogen – Ellipsen-Sektor
5. Bézier-Kurven 3. Grades als krummlinige Verbindung von zwei Punkten
6. Text

Um Text zu zeichnen, müssen Sie den Text, die Informationen zum Font, zur Schriftfarbe sowie den (Start-)Punkt angeben, an dem die Schrift nach dem Aufruf spezieller Methoden in die Zeichenfläche eingefügt wird.

25.1.10.3 Cairo-Koordinatensystem

Alle Zeichenflächen besitzen ein (virtuelles) Koordinatensystem, auf das sich alle Angaben von Koordinaten in den Eigenschaften oder in den Argumenten der Methoden der Klasse Cairo beziehen. Beachten Sie bitte diese Hinweise:

- Der Koordinaten-Ursprung $O(0|0)$ des Cairo-Koordinatensystems liegt in der *linken oberen Ecke* der Cairo-Zeichenfläche.
- Die positive x-Achse zeigt nach rechts. Die positive y-Achse zeigt – im Gegensatz zum gewohnten kartesischen Koordinatensystem – nach unten!
- Koordinaten in Cairo sind immer vom Datentyp Float.
- Jeder Punkt auf der Zeichenfläche wird durch ein Koordinaten-Paar $P(x|y)$ festgelegt.

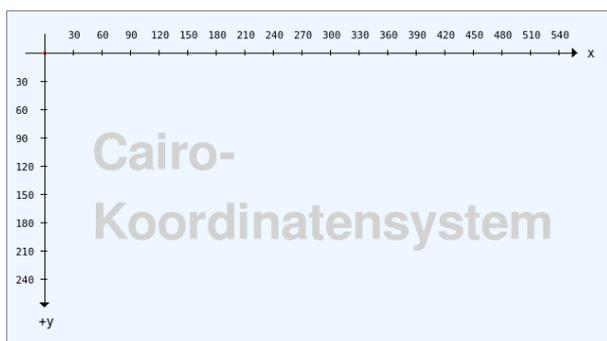


Abbildung 25.1.10.3.1: Cairo-Koordinatensystem mit verschobenem Koordinatenursprung

Um die Bezeichnungen der beiden Achsen und die Werte *bei Bedarf* gut *ablesbar* einzuzeichnen, können Sie den Koordinatenursprung des Koordinatensystem mit der Translate-Methode der Cairo-Klasse (permanent) verschieben. Diese Verschiebung ändert jedoch die *Richtung* der beiden Koordinatenachsen nicht!

```
Cairo.Begin(hImage)
    Cairo.Translate(40, 45) ' Verschiebung um +40.0 in x-Richtung und um +45.0 in y-Richtung
    ...
Cairo.End
```

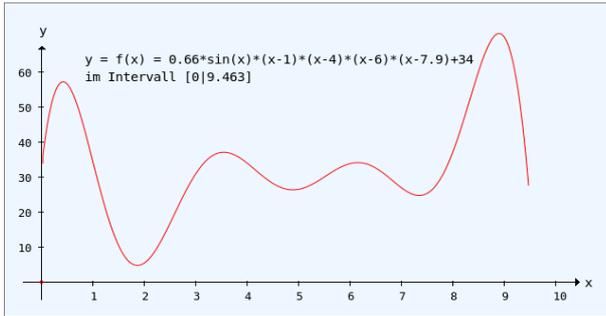


Abbildung 25.1.10.3.2: Bild einer Funktion $y = f(x)$

Um im Gegensatz zur → Abbildung 25.1.10.3.1 die gewohnte Sicht zu realisieren, können Sie das Koordinatensystem so *skalieren*, dass die positive y-Achse nach oben zeigt. Eine Parallelverschiebung des Koordinatenursprungs $O(0|0)$, verbunden mit einer Spiegelung der y-Achse an der Abszisse, realisiert durch den Einsatz der Scale-Methode, setzen Sie so um:

```
Cairo.Begin(hImage)
  Cairo.Translate(40, 45) ' Verschiebung um +40.0 in x-Richtung und um +45.0 in y-Richtung
  Cairo.Scale(1, -1) ' Die -1 bewirkt eine Invertierung der Richtung der y-Achse → +y ▲
  ...
Cairo.End
```

25.1.10.4 Zeichnen mit Methoden der Klasse Cairo

Das Konzept zum Zeichnen geometrischer Formen mit Cairo ist ähnlich zum Zeichnen mit Paint. Unterschiede werden im Absatz → 25.1.10.7 beschrieben. Eine geometrische Form wird stets *in drei Schritten* auf die benutzte Zeichenfläche gezeichnet:

1. Zuerst weisen Sie dem Pinsel über *Cairo.Source* geeignete Eigenschaften wie zum Beispiel Füll-Farbe oder Linien-Farbe zu. Die Linienbreite legen Sie bei Bedarf separat fest.
2. Dann legen Sie den so genannten *Zeichenpfad* fest, der durch einen Start-Punkt und die Angabe einer geometrischen Form beschrieben wird. Einen Start-Punkt legen Sie über die Methode *Cairo.MoveTo(..)* fest oder Sie nutzen den Endpunkt einer abgeschlossenen Zeichen-Aktion. Die Form definieren Sie mit Hilfe ausgewählter Zeichen-Methoden wie zum Beispiel *Cairo.LineTo(..)*, *Cairo.Rectangle(..)* oder *Cairo.Arc(..)* mit definierten Argumenten. Unterscheiden Sie bei *Flächen* stets zwischen dem Rand (Form Linie) und der Fläche innerhalb der Umrandung.
3. Abschließend lassen Sie den Zeichenpfad mit dem definierten Pinsel auf der Zeichenfläche *nachzeichnen*, wobei Sie für eine festgelegte Linie die Methode *Cairo.Stroke(A)* verwenden und für eine Fläche die Methode *Cairo.Fill(A)*. Im Normalfall wird der zuletzt benutzte Zeichenpfad nach (einmaligem) Nachzeichnen gelöscht. Setzen Sie das optionale Argument A auf den Wert True, dann bleibt der Zeichenpfad erhalten und kann weiter genutzt werden.

25.1.10.5 Bild zeichnen, anzeigen und speichern

Beispiel: Eine Punktwolke wird auf ein Image(-Objekt) als Zeichenfläche gezeichnet:

```
Public Sub CairoScriptPointCloud()
  Dim i As Integer

  GenerateNewImage()
  SetImageBorder()
  Cairo.Begin(hImage)
  Cairo.Translate(xTranslate, yTranslate)
  Cairo.Scale(xScale, yScale) ' +y ▲
  Cairo.AntiAlias = False
  DrawCoordinateSystem()
  For i = 1 To 50000
    ' Punkte mit zufälligen Koordinaten und einer Zufallsfarbe setzen
    SetPoint(Rnd(10, 550 - 5), Rnd(10, 260), Color.RGB(Rnd(0, 255), Rnd(0, 255), Rnd(0, 255)))
  Next
  Cairo.AntiAlias = True
  Cairo.End
End ' CairoScriptPointCloud()
```

Um sich vom Ergebnis des Zeichnens zu überzeugen, wird das Image – das sich bisher nur im Speicher befindet – in eine DrawingArea (daCanvas) gezeichnet und somit erst sichtbar:

```
Public Sub daCanvas_Draw()
    If hImage Then Paint.DrawImage(hImage, 0, 0, hImage.W, hImage.H)
End ' daCanvas_Draw()
```

Diese flexible Art und Weise zu zeichnen hat sich bewährt. Wenn Sie das gezeichnete Bild sichern und weiterverarbeiten wollen, um es zum Beispiel in einem Text zu präsentieren, so speichern Sie das Image in einer Bild-Datei ab:

```
Public Sub btnSaveCurImage_Click()
    hImage.Save(sImagePath & "/" & Lower(cmbImages.Text & ".png"))
End ' btnSaveCurImage_Click()
```

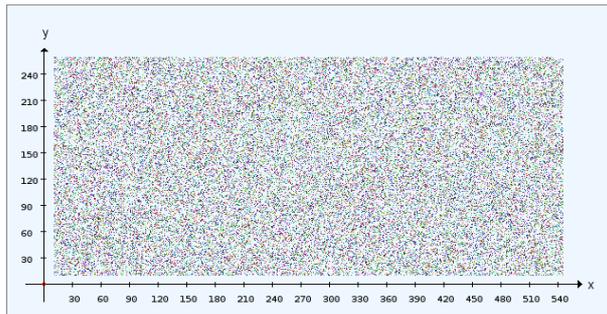


Abbildung 25.1.10.5.1: Punktwolke

Im Kapitel → 25.1.11 CairoProjekte werden Ihnen weitere Bild-Beispiele vorgestellt und ausführlich kommentiert.

25.1.10.6 Standardwerte

Für das Zeichnen mit Cairo gibt es festgelegte (Standard-)Startwerte:

- Die Linien- oder Füll-Farbe oder die Schriftfarbe ist voll-deckend schwarz (&H000000) – ohne Alpha-Kanal in der Farbe.
- Die Linienstärke beträgt 2.
- Die Kantenglättung (AntiAlias) ist eingeschaltet.
- In der Cairo-Dokumentation → <http://cairographics.org/manual/cairo-text.html> steht, dass der Standard-Font *plattform-spezifisch* ist und der Quasi-Standard 'sans-serif' ist. Den Font-Namen des verwendeten *Standard-Fonts* können Sie in Gambas nicht auslesen. Das gelingt erst dann, wenn Sie einen eigenen Font über die Eigenschaft *Cairo.Font.Name* als aktuellen Font zugewiesen haben.

Bei einem Test ergeben sich diese Werte:

```
Print Cairo.LineWidth      ' 2
Print Cairo.AntiAlias      ' 0
Print Cairo.Font.Size      ' 10
```

25.1.10.7 Unterschiede beim Zeichnen mit Paint und Cairo

Sie sollten sich die Dokumentationen zu den Eigenschaften und Methoden der Klasse Cairo der Komponente Cairo und der Klasse Paint (gb.qt4) sehr genau ansehen, da einige Methoden zwar gleiche Namen haben, jedoch eine unterschiedliche Syntax aufweisen oder zum Beispiel gleiche Eigenschaften unterschiedliche Wertebereiche besitzen. Besondere Aufmerksamkeit ist bei folgenden Themen geboten:

- Zeichen-Flächen
- Font-Zuweisungen
- Farbverteilungen in einem Muster bei LinearGradient und RadialGradient
- Kreise und Kreisteile
- Anti-Aliasing