23.3.5.4 Bild-Lupe

Das Projekt *Bild-Lupe* entstand aus dem adaptierten Quelltext von *Fabien Bodard* für sein Beispiel-Projekt MapView, den Sie im Zusammenhang mit der Komponente gb.map im → Kapitel 24.8.0 'Komponente Map' nachlesen können. Der Quelltext für das Projekt Bild-Lupe wurde in der Erprobungsphase von Fabien Bodard optimiert.

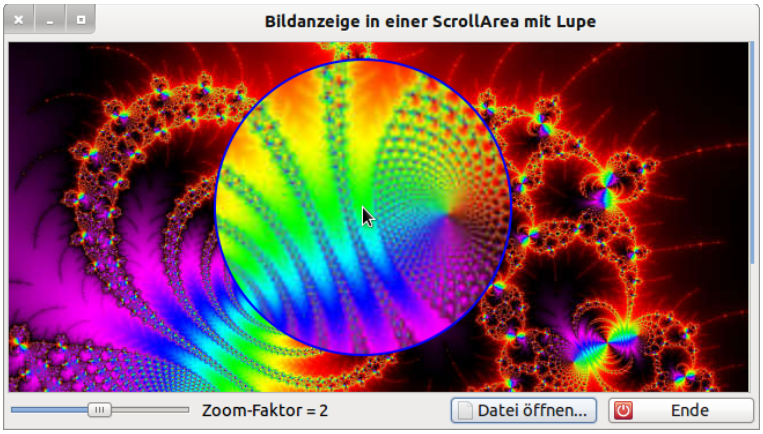

Abbildung 23.3.5.4.1: Bild-Lupe als beeindruckender Effekt

Im Unterschied zum Beispiel-Projekt MapView ist die Bild-Lupe permanent eingeschaltet. Der Zoom kann vom Wert 1 (Bild:Original=1) bis zum Wert 3 über einen Schieberegler kontinuierlich geändert werden. Die bemerkenswerteste Änderung gegenüber dem Projekt MapView ist die Verwendung eines Timers, mit dem das Bild in kleinen Zeitintervallen stets neu gezeichnet wird.

Quelltext:

```
[1]  ' Gambas class file
[2]
[3]  Private $sFile As String
[4]  Private $hImage As Image
[5]  Private $himgZoom As Image
[6]  Private $iLensSize As Integer = 256
[7]  Private $fZoom As Float = 2
[8]  Private $bMouseDown As Boolean
[9]  Private $MX As Integer
[10] Private $MY As Integer
[11]
[12] ' Special technic to delay and cumulate drawing events
[13] ' In some case of too numerous call this allow to reduce it to one call by event loop
[14] ' Durch den Timer werden viele - sonst notwendige - Auswertungen der Event-Schleife auf
[15] ' einen Aufruf reduziert: $TimerDraw.Trigger
[16]
[17] Private $TimerDraw As Timer
[18]
[19] Public Sub _new()
[20]   $TimerDraw = New Timer As "TimerDraw"
[21]   $TimerDraw.Delay = 5
[22] End ' _new()
[23]
[24] Public Sub Form_Open()
[25]   FMain.H = 636
[26]   FMain.W = 806
[27]   FMain.Center
[28]   $sFile = "fractal.jpg"
[29]   Try $hImage = Image.Load($sFile)
[30]   If Error Then
[31]      Message(Error.Text)
[32]      Return
[33]   Endif
[34]   Slider1.Value = 200
[35]   lblDisplayZoom.Text = ("Zoom-Factor = ") & Str(Slider1.Value / 100)
[36]   ScrollArea1.ResizeContents($hImage.Width, $hImage.Height)
[37]   $TimerDraw.Trigger ' Call refreshing
[38] End ' Form_Open()
[39]
[40] Public Sub btnFileOpen_Click()
[41]   Dialog.Path = User.Home
[42]   Dialog.Title = ("Select an image ...")
[43]   Dialog.Filter = (["*.png;*.jpg", ("Images")])
```

```
[44]   If Not Dialog.OpenFile() Then
[45]      $sFile = Dialog.Path
[46]      Try $hImage = Image.Load($sFile)
[47]      If Error Then
[48]         Message(Error.Text)
[49]         Return
[50]      Endif
[51]      ScrollArea1.ResizeContents($hImage.Width, $hImage.Height)
[52]      $TimerDraw.Trigger
[53]   Endif
[54] End ' btnFileOpen_Click()
[55]
[56] Public Sub ScrollArea1_Draw()
[57]
[58]   If Not $hImage Then Return
[59]
[60]   Paint.DrawImage($hImage, 0 - ScrollArea1.ScrollX, 0 - ScrollArea1.ScrollY, $hImage.Width,
                    $hImage.Height)
[61]   If $bMouseDown Then
[62]      Paint.LineWidth = 2
[63]      Paint.Brush = Paint.Image($himgZoom, $mx - $iLensSize / 2, $my - $iLensSize / 2)
[64]      Paint.Arc($MX, $MY, $iLensSize / 2)
[65]      Paint.Fill(True)
[66]      Paint.Brush = Paint.Color(Color.Blue)
[67]      Paint.Stroke
[68]   Endif
[69]
[70] End ' ScrollArea1_Draw()
[71]
[72] Public Sub ScrollArea1_MouseDown()
[73]   ScrollArea1_MouseMove
[74] End ' ScrollArea1_MouseDown()
[75]
[76] Public Sub ScrollArea1_MouseUp()
[77]   $bMouseDown = False
[78]   $TimerDraw.Trigger
[79] End ' ScrollArea1_MouseUp()
[80]
[81] Public Sub ScrollArea1_MouseMove()
[82]   Dim iMx, iMy As Integer
[83]   Dim fSquareSize As Float
[84]   Dim tmpImg, tmpImg2 As Image
[85]
[86]   If Not $hImage Then Return
[87]
[88]   $MX = Mouse.X
[89]   $MY = Mouse.y
[90]   If Mouse.Left Then
[91]   ' Real position of the mouse
[92]     iMx = Mouse.x + ScrollArea1.ScrollX
[93]     iMy = Mouse.y + ScrollArea1.ScrollY
[94]   ' Copy the part below divide by the zoom level
[95]     fSquareSize = $iLensSize / $fZoom
[96]     tmpImg = New Image(fSquareSize, fSquareSize, Color.DarkGray)
[97]     tmpImg2 = $hImage.Copy(iMx - fSquareSize / 2, iMy - fSquareSize / 2, fSquareSize, fSquareSize)
[98]
[99]     Paint.Begin(tmpImg)
[100]        If iMy > $hImage.Height / 2 Then
[101]           If iMx > $hImage.Width / 2 Then
[102]            ' Bottom/Right
[103]              Paint.DrawImage(tmpImg2, 0, 0)
[104]           Else
[105]            ' Bottom/Left
[106]              Paint.DrawImage(tmpImg2, Paint.Width - tmpImg2.Width, 0)
[107]           Endif
[108]        Else
[109]           If iMx > $hImage.Width / 2 Then
[110]            ' Top/Right
[111]              Paint.DrawImage(tmpImg2, 0, Paint.Height - tmpImg2.Height)
[112]           Else
[113]            ' Top/Left
[114]              Paint.DrawImage(tmpImg2, Paint.Width - tmpImg2.Width, Paint.Height - tmpImg2.Height)
[115]           Endif
[116]        Endif
[117]      Paint.End
[118]
[119]      $himgZoom = tmpImg.Stretch($iLensSize, $iLensSize)
[120]      $bMouseDown = True
[121]      $MX = Mouse.X
[122]      $MY = Mouse.y
[123]      $TimerDraw.Trigger
[124]    Endif
[125]
[126]  End ' ScrollArea1_MouseMove()
```

```
[127]
[128]  Public Sub TimerDraw_Timer()
[129]    ScrollArea1.Refresh
[130]  End ' TimerDraw_Timer()
[131]
[132]  Public Sub Slider1_Change()
[133]    $fZoom = Round(Slider1.Value / 100, -2)
[134]    lblDisplayZoom.Text = ("Zoom-Factor = ") & $fZoom
[135]    $TimerDraw.Trigger
[136]  End ' Slider1_Change()
[137]
[138]  Public Sub btnClose_Click()
[139]    FMain.Close
[140]  End ' btnClose_Click()
```