Frank Hileman en Anne Szyjan zijn ontwikkelaars bij VG.net

Vector graphics in Visual Studio .NET: VG.net

REAL-TIME GEGEVENSVISUALISATIE MET VG.NET

In dit artikel wordt beschreven hoe u met VG.net opvallende vector graphics kunt toevoegen aan een user interface. VG.net is een add-on voor vector graphics die geïntegreerd kan worden in Visual Studio .NET. De auteurs hebben een real-time gegevenssimulatie gemaakt ter illustratie van gegevensvisualisatie met vector graphics. Een serverapplicatie genereert gegevens die via TCP/IP naar een clientapplicatie worden verzonden om te worden gevisualiseerd. In dit artikel wordt de gegevensvisualisatie van dit voorbeeld uitgelicht.

Als u het voorbeeld wilt uitvoeren, kunt u het downloaden van www.microsoft.nl/netmagazine6. De lite versie van VG.net kunt u (kosteloos) downloaden van www.vgdotnet.com. Vervolgens pakt u de bestanden in VGDotNetDemoBinaries.zip uit in een map. Voer eerst Server.exe uit; dit programma moet draaien voordat u de client start. U ziet het serverformulier in afbeelding 1. Voer Client.exe een of meer keren uit. De client wordt weergegeven in afbeelding 2. Wijzig de grootte van het clientformulier en u ziet dat de objecten in de vorm van vector graphics automatisch worden aangepast. Experimenteer met de instellingen van het serverformulier. Probeer een puls te versturen naar de client. Als u de server en client op verschillende machines wilt uitvoeren, start u de server als eerste. Start Client.exe vanaf de command line en geef hierbij de naam van de servermachine op als eerste argument. De broncode voor dit voorbeeld vindt u in VGDotNetDemoSource.zip. Als u VG.net-graphics wilt bewerken in Visual Studio .NET, moet u de Lite versie van VG.net downloaden van www.vgdotnet.com en installeren.

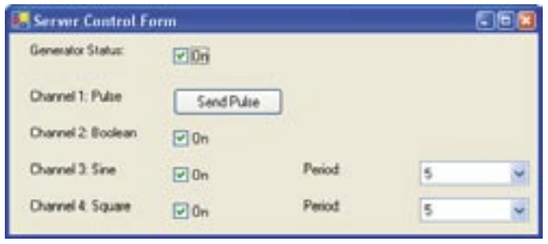
Vector graphics in het .NET Framework

Het .NET Framework omvat tekenklassen in de namespace System.Drawing. Als u de klassen in System.Drawing bekijkt, ziet u veel tekenfuncties, maar geen algemeen systeem van objecten die automatisch worden getekend en die reageren op muis-events. VG.net voorziet in dit ontbrekende grafische systeem. VG.net bestaat uit een grafische designer en een run-time animatie-engine. Met de grafische designer die deel uitmaakt van Visual Studio .NET (zie afbeelding 3) kunt u grafische onderdelen maken op dezelfde manier als u UserControls maakt: met de Toolbox en het venster Properties.

Pictures en elements

Een Picture is een .NET-klasse met elementen in de vorm van vector graphics. U kunt pictures maken met de VG.net Picture-designer die wordt geopend als u dubbelklikt op een picture in de Solution Explorer. Een picture is een onderdeel op het hoogste niveau in de VG.net-designer, vergelijkbaar met een UserControl in de UserControl-designer. Net als elke andere .NET-klasse, kan een picture eigen properties, methoden en events hebben. Element is de basisklasse voor alle grafische objecten in een picture. Elementklassen zijn onder andere:

- Shapes, met vullingen, randen en tekst. Shapeklassen zijn bijvoorbeeld Rectangle, Ellipse, Polyline, Spline, Pie, Arc en Path. Polyline en Path kunnen veelhoeken vormen.
- Image.
- Group, gemaakt door het groeperen van bestaande elementen.
- Sub Pictures.



Afbeelding 1. Serverformulier

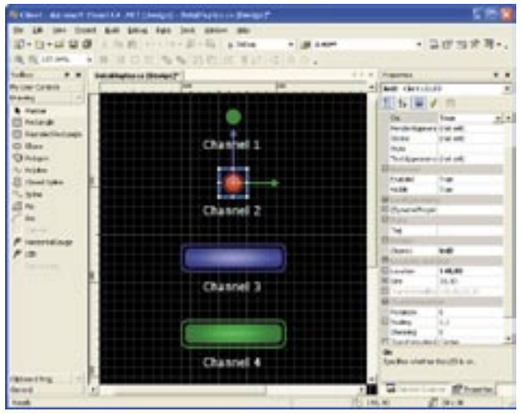
Afbeelding 2. Clientformulier

Aangezien de klasse Picture properties overneemt van Element, kan een picture andere pictures bevatten als onderliggende elementen. Onderliggende pictures worden Sub Pictures genoemd. Pictures worden niet direct in een Windows-formulier weergegeven. Pictures worden tijdens run-time weergegeven in een speciale control met de naam Canvas. Laten we beginnen met het maken van de real-time gegevenssimulatie. Maak eerst een nieuwe solution in Visual Studio .NET met de naam Sockets.

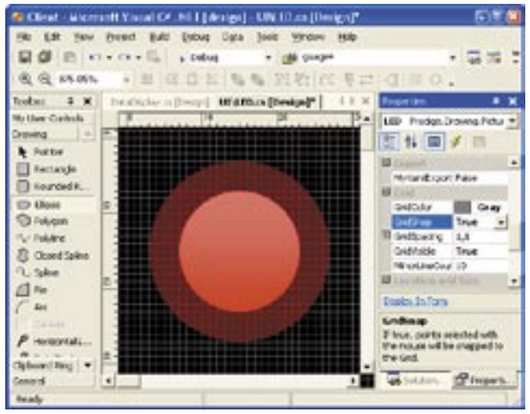
De projecten en formulieren maken

Voeg twee Windows Forms-applicatieprojecten toe aan de solution: Client en Server. De server genereert gegevens voor vier waarden van het type double, die elk staan voor een gegevenskanaal. De server genereert elk kanaal op de volgende manier:

- Kanaal 1: normaal ingesteld op 0, kan voor korte tijd worden 'gepulsed' naar 1 op de server.
- Kanaal 2: ingesteld op een constante waarde die wordt geselecteerd op de server, hetzij 0 of 1.



Afbeelding 3. Grafische designer



Afbeelding 4. Lampje bestaande uit 2 Ellipse-objecten

- Kanaal 3: een sinusgolf tussen -1 en 1, met een golfperiode die kan worden aangepast.
- Kanaal 4: een blokgolf tussen -1 en 1, met een golfperiode die kan worden aangepast.

De server verstuurt gegevens naar de client via een TCP/IP-verbinding. De client geeft de huidige waarde van elk gegevenskanaal weer. Om de gegevens op de client te visualiseren maken we een lampje (LED) en een meter (Gauge).

Er zijn twee Windows-formulieren:

- Op de client een formulier met de VG.net-picture met exemplaren van het lampje en de meter.
- Op de server een formulier met de user interface die elk gegevenskanaal bestuurt.

Maak twee Windows-formulieren, een in elk project. Het formulier op de client noemen we ClientForm en dat op de server ServerForm. We maken deze formulieren later af. Maak voor het gemak een submap met de naam UI in het project Client. In deze map slaan we de onderdelen lampje en meter op. Laten we nu de eerste VG.net-picture maken, het lampje.

Het uiterlijk van het lampje

Maak eerst een picture met de naam LED in de map UI in het project Client. Het lampje is eenvoudiger te maken als u de picture selecteert en de volgende properties instelt:

- U krijgt een beter idee van de manier waarop het lampje eruitziet in de uiteindelijke weergave als u de property BackColor instelt op Black. De property BackColor heeft alleen invloed op de weergave in de designer. Een picture heeft geen backgroundobject, tenzij u er een opgeeft.
- Aangezien het lampje maar klein is, hebt u een raster nodig om nauwkeurig te kunnen tekenen als u inzoomt. Stel de property GridSpacing in op 1 in elke dimensie en stel GridSnap en GridVisible in op True.

Het lampje is eenvoudig. Het bestaat uit twee Ellipse-objecten, zoals in afbeelding 4. De binnenste ellips is het werkelijke lampje en de buitenste de lichtkrans. U vindt het element Ellipse (Ovaal) in de Toolbox op het tabblad Drawing. Maak een ovaal met een willekeurige grootte en stel de properties in zoals in deze tabel:

Property	Waarde
Name	back
Location	(3, 3)
Size	(24, 24)
DrawAction	Fill

Tabel 1. Properties van het element Ovaal.

De ovaal is nu heel klein. Selecteer To Fit in het vak Zoom op de werkbalk Drawing om in te zoomen op de ovaal. Klik hierna op ZoomOut om de ovaal een beetje kleiner te maken. De ovaal met de naam Back vormt de lichtkrans van het lampje. We veranderen de vulkleur met het toevoegen van styles. Voeg nog een ovaal toe en stel de properties hiervan in zoals in de volgende tabel:

Property	Waarde
Name	light
Location	(7, 7)
Size	(16, 16)
DrawAction	Fill

Tabel 2. Properties van de tweede ovaal.

Hoewel u de vulling van beide ovalen kunt wijzigen door de property Fill aan te passen, gebruiken we hier in dit geval styles voor, zodat we het uiterlijk van de ovalen tijdens runtime kunnen wijzigen door de style-verwijzingen te wisselen. Selecteer de picture LED. Open de collectie Styles. U vindt deze onder de kop Collections in het venster Properties. Voeg vier styles toe, zoals in afbeelding 5. De instellingen voor de styles ziet u in deze tabel:

Naam van de style	Property Fill
OnRed	Bright red
OffRed	Dark red
BackRed	Red met Opacity van .33
OnGreen	Bright green
OffGreen	Dark green
BackGreen	Green met Opacity van .25

Tabel 3. Instellingen voor de styles.

Als u de styles hebt gemaakt, kunt u deze toewijzen aan de property Style van elke ovaal. Stel de style van de ovaal met de naam Back in op BackRed en de style van de ovaal met de naam Light in op OffRed. In het lampje dat wij hebben gemaakt, hebben we een paar kleurovergangen toegevoegd in plaats van effen kleuren bij de styles met 'On' in de naam om ze wat mooier te maken. Bekijk het voorbeeldproject voor de details. U kunt het lampje gemakkelijker plaatsen en groter of kleiner maken door een rechthoek toe te voegen op Location (0, 0) en met Size (30, 30). Geef de rechthoek de naam 'enclosingRectangle', stel de DrawAction in op Edge, en wijzig de Stroke Opacity zodat deze volledig transparant is. De

rechthoek definieert de grenzen van het lampje zonder dat hij zichtbaar is.

Het gedrag van het lampje

Het lampje kan maar twee kleuren hebben: rood of groen. We willen dat de gebruiker de kleur van het lampje kan selecteren. Hiervoor voegen we een enum toe met de naam LEDColor. Binnen de klasse LED voegen we een veld toe met de naam Color van het type LEDColor en met een property die dat veld bestuurt; zie codevoorbeeld 1. De methode UpdateStyles (zie codevoorbeeld 2) zorgt ervoor dat de stijlen daadwerkelijk worden bijgewerkt volgens de waarde van het veld Color. Bij UpdateStyles wordt gebruikgemaakt van een Booleaans veld met de naam On dat is toegevoegd aan het lampje. Dit veld geeft aan of het lampje aan of uit is. De status van het lampje wordt ingesteld met de property On; zie codevoorbeeld 3.

Het uiterlijk van de horizontale meter

Voeg een picture met de naam HorizontalGauge toe aan de map UI in het project Client. De meter is eenvoudiger te maken als u de picture selecteert en de volgende properties instelt:

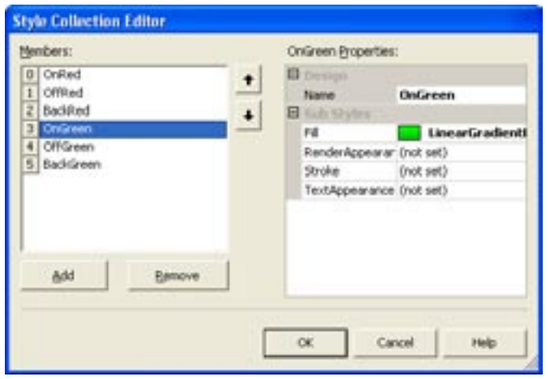
- Stel de property BackColor in op Black.
- Stel de property GridSpacing in op 2 in elke dimensie en stel GridSnap en GridVisible in op True.

Property	Waarde
Name	border
CornerRadius	7
Location	(0, 0)
Size	(104, 24)
Stroke	Color = dark blue, Width = 2
DrawAction	Edge

Tabel 4. Properties van de rechthoek.

De meter, die u ziet in afbeelding 6, bestaat uit een rechthoek die de randen vormt en een binnenste groep met drie elkaar overlappende rechthoeken met precies dezelfde omvang. De breedte van de binnenste groep verandert als een gegevenswaarde verandert. Maak eerst de rand. Maak een afgeronde rechthoek met een willekeurige grootte en stel de properties in zoals in deze tabel:

In afbeelding 7 ziet u de instelling voor Stroke in het dialoogvenster Stroke Properties. Aangezien de rechthoek met de naam Border klein is, selecteert u To Fit op de werkbalk Drawing om in te zoomen. Maak vervolgens de achtergrond van de waardebalk. Maak nog een afgeronde rechthoek en stel de properties in zoals in deze tabel:



Afbeelding 5. Toevoegen styles

Property	Waarde
Name	back
CornerRadius	5
Location	(4, 4)
Size	(96, 16)
Fill	Solid, met Color = bright blue
DrawAction	Fill

Tabel 5. Properties tweede rechthoek.

Als het goed is, ziet de rechthoek met de naam Back eruit zoals in afbeelding 8. Hier overheen voegt u een andere rechthoek toe met dezelfde afmetingen en dezelfde instelling voor CornerRadius. Selecteer de rechthoek Back en kopieer en plak deze. U hebt nu een nieuwe rechthoek die identiek is aan de eerste. Wijzig de properties van deze nieuwe rechthoek als volgt:

Property	Waarde
Name	shine
Fill	LinearGradientFill: GradientType: TwoColorBell StartColor: (not set) StartOpacity: 0 EndColor: white EndOpacity: .412 Angle: 90

Tabel 6. Properties nieuwe rechthoek.

Als het goed is, ziet de rechthoek met de naam Shine eruit zoals in afbeelding 9. Hier overheen voegt u nog een andere rechthoek toe met dezelfde afmetingen en dezelfde instelling voor CornerRadius. Selecteer de rechthoek Shine en kopieer en plak deze. Wijzig de properties van deze nieuwe rechthoek als omschreven in tabel 7.

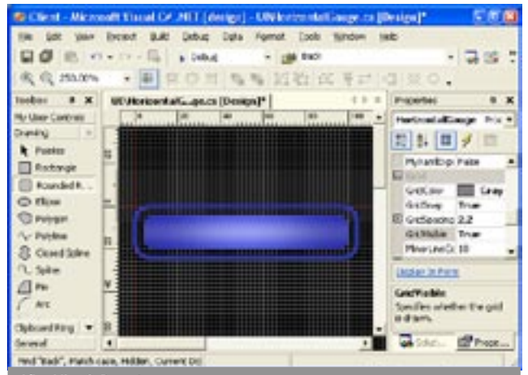
Als het goed is, ziet de rechthoek met de naam Ends eruit zoals in afbeelding 10. Tezamen vormen deze objecten een gloedeffect. Selecteer alle drie de binnenste rechthoeken, Back, Shine en Ends, en klik op de opdracht Group op de werkbalk Drawing. U kunt de drie rechthoeken eenvoudig tegelijk selecteren door er een selectievak omheen te trekken met de muis. Als u de rechthoeken hebt geselecteerd, maakt u een groep door te klikken op het pictogram Group op de werkbalk Drawing. Geef de groep de naam valueBar. Dankzij de groepering kunt u de breedte van de drie rechthoeken tijdens run-time met één bewerking aanpassen.

Property	Waarde
Name	ends
Fill	LinearGradientFill: GradientType: TwoColorBell StartColor: very dark blue StartOpacity: .502 EndColor: (not set) EndOpacity: 0 Angle: 0

Tabel 7. Wijziging properties nieuwe rechthoek.

Het gedrag van de meter

Een herbruikbaar grafisch onderdeel is flexibeler als het properties heeft waarmee het uiterlijk kan worden gewijzigd. We voegen properties toe om de kleur van de rechthoek Border en de rechthoek Back en een van de kleuren die wordt gebruikt voor de rechthoek Ends in te stellen. We doen niets met de kleuren van de rechthoek Shine omdat deze een inherent onderdeel zijn van het stralende uiterlijk en goed samengaan met alle kleuren. De property BorderColor en de bijbehorende methoden ziet u in codevoorbeeld 4.

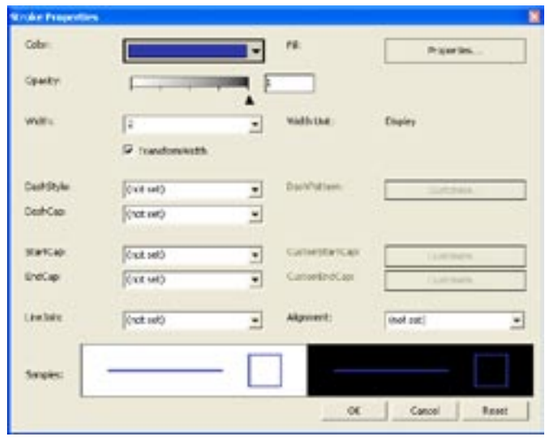


Afbeelding 6. De meter

De property BorderColor maakt gebruik van een private property, BorderFill. De property BorderFill is bedoeld voor uw gemak en omvat drie bewerkingen: het object Stroke van de Rectangle Border ophalen, het object Fill van de Stroke ophalen en Fill instellen op SolidFill. Met de property BorderColor haalt u de property Color van BorderFill op of stelt u deze in. De methoden ShouldSerializeBorderColor en ResetBorderColor maken het ontwerpen plezieriger. Meer informatie over de methoden ShouldSerialize en Reset vindt u in het gedeelte “ShouldSerialize and Reset Methods” in de .NET Framework Developer’s Guide (Engelstalig). De methode ShouldSerializeBorderColor zorgt ervoor dat de property BorderColor niet kan worden geserialiseerd, tenzij de property wordt ingesteld op een niet-standaardwaarde. Met de methode ResetBorderColor kan de gebruiker de property BorderColor opnieuw instellen op een standaardwaarde door met de rechtermuisknop te klikken in het venster Properties. Beide methoden maken gebruik van het statische veld defaultBorderColor, dat wordt geïnitialiseerd met een standaardwaarde in de constructor van HorizontalGauge. Met de properties BackColor en EndColor in HorizontalGauge kan de gebruiker respectievelijk de kleur van de rechthoek Back en de niet-transparante kleur van de rechthoek Ends wijzigen. Deze properties worden op dezelfde manier geïmplementeerd als de property BorderColor.

Naast deze properties voor het uiterlijk, heeft HorizontalGauge properties voor de weergave van een gegevenswaarde:

- Value: de huidige gegevenswaarde die is opgeslagen in het veld met de naam Current.
- Minimum: de minimumwaarde van de property Value, die is opgeslagen in het veld met de naam Minimum.
- Maximum: de maximumwaarde van de property Value, die is opgeslagen in het veld met de naam Maximum.



Afbeelding 7. Instelling voor Stroke in het dialoogvenster Stroke Properties

De implementatie van deze properties ziet u in codevoorbeeld 5. Het leeuwendeel van het werk vindt plaats in de UpdateWidth-methode, waarmee de property Width van de groep ValueBar proportioneel wordt aangepast aan het veld Current.

De picture DataDisplay maken

De picture DataDisplay bevat enkele onderdelen van het lampje, enkele onderdelen van de meter en enkele transparante rechthoeken met tekstlabels. Compileer eerst uw project, zodat de onderdelen LED en HorizontalGauge beschikbaar zijn. Voeg een picture met de naam DataDisplay toe aan het project Client. Dit gaat eenvoudiger als u de picture selecteert en de volgende properties instelt:

- Stel de property BackColor in op Black.
- Stel de properties GridSnap en GridVisible in op True.
- Maak nu twee LED-objecten met de namen Led1 en Led2. Maak het eerste LED groen door de property LEDColor te wijzigen.

Maak twee HorizontalGauge-objecten met de namen gauge3 en gauge4. Wijzig de drie properties voor kleur van gauge4 zodat u deze kunt onderscheiden van gauge3. Deze properties zijn BorderColor, BackColor en EdgeColor. Kies kleuren die u mooi vindt. Voeg tot slot labels toe. De labels zijn eigenlijk rechthoeken waarvan de property DrawAction is ingesteld op Fill. Om deze rechthoeken van binnen transparant te maken en van een witte tekst te voorzien, maakt u een style met de naam Label. Wijzig in de style met de naam Label de property Fill zodat deze volledig transparant is (Opacity 0) en wijzig de property TextAppearance zodat wit wordt gebruikt als tekstkleur. Stel hierna de property Style van elke labelrechthoek in op Label. De picture DataDisplay moet inkomende gegevens doorsturen naar het lampje en de meter. Voeg vier properties toe aan DataDisplay, een voor elk gegevenskanaal, zoals in codevoorbeeld 6. Als het goed is, ziet de picture DataDisplay eruit zoals in afbeelding 3.

Het clientformulier samenstellen

Compileer eerst het project Client zodat de picture DataDisplay beschikbaar is. Wijzig het formulier met de naam ClientForm in het project Client. In de voorbeeldcode is ClientForm lang en smal, maar u kunt elke gewenste grootte gebruiken die de afmetingsverhouding van de picture DataDisplay benadert. Terwijl het formulier is geopend in de Forms-designer, opent u het tabblad Drawing van de Toolbox. Hier ziet u de klasse Canvas. Sleep een canvas naar het formulier en stel de properties als volgt in:

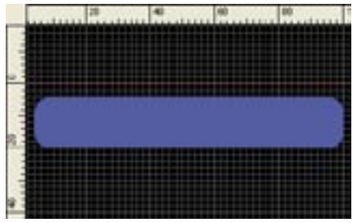
Property	Waarde
Name	canvas
Dock	Fill
Achtergrond	Black
AutoSizePicture	True
AutoSizePadding	10, 10

Tabel 8. Properties van de klasse Canvas.

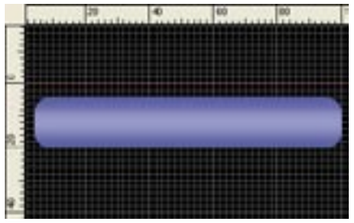
Als het goed is, is de picture DataDisplay zichtbaar als item in het gedeelte Drawing van de Toolbox. Sleep dit item naar het formulier en u ziet een DataDisplay-object in de Component Tray onder het formulier. Geef dit object de naam dataDisplay. Selecteer nu het object Canvas en wijzig de property Picture in dataDisplay. De picture wordt weergegeven op het canvas, zoals in afbeelding 11. De property AutoSizePicture zorgt ervoor dat de afmetingen van de picture worden aangepast aan het canvas en dat de picture wordt gecentreerd op het canvas. AutoSizePadding bepaalt de x- en y-ruimte tussen de randen van de picture en de randen van het canvas.

Het serverformulier samenstellen

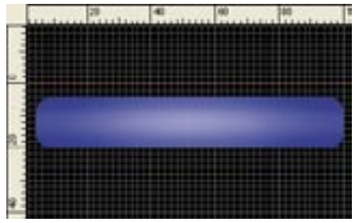
Het valt buiten het bestek van dit artikel om de samenstelling van het serverformulier in afbeelding 1 te beschrijven. Het serverformulier bevat een eenvoudige collectie standaardcontrols voor Windows-



Afbeelding 8. Rechthoek Back



Afbeelding 9. Rechthoek Shine



Afbeelding 10. Rechthoek Ends

formulieren. Raadpleeg de voorbeeldcode voor meer informatie.

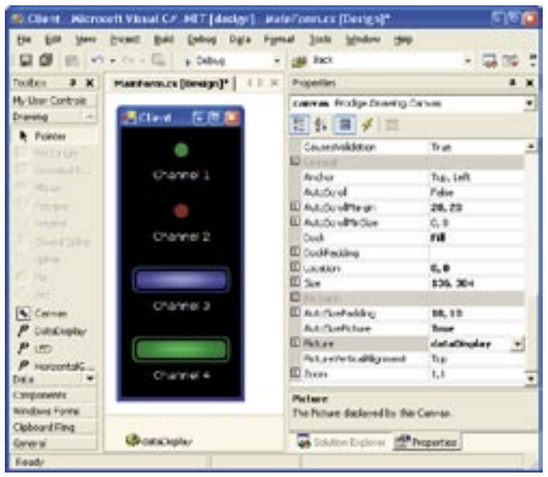
Gegevenscommunicatie met TCP/IP

In de serverapplicatie worden gegevens gegenereerd door de klasse SignalGenerator. Deze klasse maakt gebruik van een timer om gegevensgeneratie te starten met tussenpozen van 30 milliseconden. Alle gegevens worden berekend in de functie UpdateData. Waarden van sinus- en blokgolven worden gebaseerd op de verstreken tijd sinds het begin van de simulatie. SignalGenerator luistert naar verzoeken om verbinding van de client met een TcpListener-object. Wanneer een verbinding in behandeling is, accepteert SignalGenerator het verzoek en wordt het corresponderende Socket-object doorgegeven aan een nieuw ClientWriter-object. Elk ClientWriter-object dat wordt gemaakt, verzendt gegevens naar één client. In de clientapplicatie is de klasse ClientReader verantwoordelijk voor de verbinding met een server en het ontvangen van gegevens. Er wordt gebruik gemaakt van de klasse TcpClient van het .NET Framework. Wanneer gegevens worden ontvangen, veroorzaakt de ClientReader een event en roept de methode DataReceived aan in het clientformulier. DataReceived vindt u in codevoorbeeld 5. De methode DataReceived wijzigt de properties in de klasse DataDisplay en het scherm wordt bijgewerkt bij de volgende weergave.

Meer tips

In dit artikel hebben we real-time gegevensvisualisatie met VG.net beschreven. Dit systeem voor vector graphics kent vele andere toepassingen. Aangezien elk grafisch object muis-events ondersteunt, net als een Control-object, kunt u onderdelen van user interfaces zoals knoppen en schuifregelaars, maken met VG.net. De installatiemap van VG.net bevat voorbeelden van user interfaceobjecten. U vindt meer tips voor VG.net op www.vgdotnet.com en het VG.net development blog (http://weblogs.asp.net/frank_hileman).

Frank Hileman is lead developer van VG.net. Hij bouwt al sinds 1986 objectgeoriënteerde vector graphics systems. Anne Szyjan is een senior software engineer. Zij werkt aan VG.net en heeft meegewerkt aan veel verschillende softwareapplicaties op het gebied van business, robotics en process control.



Afbeelding 11. Picture op het canvas

```
Codevoorbeelden:

public LEDColor Color
{
    get { return color; }
    set
    {
        if (color == value)
            return;
        color = value;
        UpdateStyles();
    }
}
```

Codevoorbeeld 1.
Toevoegen van LEDColor

```
private void UpdateStyles()
{
    if (color == LEDColor.Red)
    {
        back.Style = "BackRed";
        if (on)
            light.Style = "OnRed";
        else
            light.Style = "OffRed";
    }
    if (color == LEDColor.Green)
    {
        back.Style = "BackGreen";
        if (on)
            light.Style = "OnGreen";
        else
            light.Style = "OffGreen";
    }
    back.Visible = on;
}
```

Codevoorbeeld 2.
De functie UpdateStyles

```
public bool On
{
   get { return color; }
   set
   {
      if (on == value)
         return;
      on = value;
      UpdateStyles();
   }
}
```

Codevoorbeeld 3.
De status van het lampje wordt ingesteld met de property On

```
public Color BorderColor
{
    get { return BorderFill.Color; }
    set { BorderFill.Color = value; }
}

private bool ShouldSerializeBorderColor()
{
    return BorderColor != defaultBorderColor;
}

private void ResetBorderColor()
{
    BorderColor = defaultBorderColor;
}

private SolidFill BorderFill
{
    get { return (SolidFill)border.Stroke.Fill; }
}
```

Codevoorbeeld 4.
De property BorderColor van HorizontalGauge en ondersteunde methoden.

```
public float Minimum
{
    get { return minimum; }
    set
    {
        if (minimum == value)
            return;
        minimum = value;
        UpdateWidth();
    }
}

public float Maximum
{
    get { return maximum; }
    set
    {
        if (maximum == value)
            return;
        maximum = value;
        UpdateWidth();
    }
}

public float Value
{
    get { return current; }
    set
    {
        if (current == value)
            return;
        current = value;
        UpdateWidth();
    }
}

private void UpdateWidth()
{
    // update the width of the valueBar to reflect
    // the current value
    float range = maximum - minimum;
    if (range == 0)
        return;
    float value = current;
    if (value > maximum)
        value = maximum;
    if (value < minimum)
        value = minimum;
    float width =
        ((value - minimum) / range) * maxBarWidth;
    if (width < 1)
        width = 1;
    valueBar.Width = width;
}
```

Codevoorbeeld 5.
Properties van HorizontalGauge die betrekking hebben op gegevensweergave

```
public double Value1
{
    set { led1.On = value > 0; }
}

public double Value2
{
    set { led2.On = value > 0; }
}

public double Value3
{
    set { gauge3.Value = (float)value; }
}

public double Value4
{
    set { gauge4.Value = (float)value; }
}
```

Codevoorbeeld 6.
Properties van DataDisplay die gegevens doorgeeft aan LED en HorizontalGauge.

```
private void DataReceived(double value1, double value2,
    double value3, double value4)
{
    dataDisplay.Value1 = value1;
    dataDisplay.Value2 = value2;
    dataDisplay.Value3 = value3;
    dataDisplay.Value4 = value4;
}
```

Codevoorbeeld 7.
De methode DataReceived in ClientForm.