Chart3D for WPF/Silverlight

2018.04.10 更新

グレープシティ株式会社

目次

製品の概要

データを3次元でグラフ化します。**Chart3D for WPF/Silverlight** は、本格的な 3D 面グラフを作成します。オプションで等高線レベルとゾーンを使用できます。グラフを任意の角度に回転したり、グラフの凡例を表示することもできます。

> **メモ**:説明内に含まれるクラスおよびメンバーに対するリファレンスへのリンクは、原則としてWPF版のリファレンスページを参照します。Silverlight版については、目次から同名のメンバーを参照してください。

ComponentOne for WPF/Silverlight のヘルプ

はじめに

ComponentOne for WPF/Silverlight のすべてのコンポーネントで共通の使用方法については、「ComponentOne for WPF/Silverlight ユーザーガイド」を参照してください。

主な特長

Chart3D for WPF/Silverlight は、次のユニークな機能を提供します。

- **面グラフタイプ**

  Chart3D for WPF/Silverlight には、6つの面グラフタイプがあります。グラフの作成時に、ワイヤフレーム、等高線レベル、およびゾーンを含むさまざまな面グラフを選択できます。各軸に沿って等高線とメッシュを表示するかどうかをカスタマイズします。詳細については、「**グラフタイプ**」を参照してください。

- **凡例の表示**

  ゾーングラフにグラフ凡例を表示します。詳細については、「**グラフ凡例の追加**」を参照してください。

- **回転と傾きの設定**

  Azimuth プロパティと **Elevation** プロパティを設定して、グラフを任意の角度に回転させます。詳細については、「**グラフの回転と傾き**」を参照してください。

- **床面と天井面**

  グラフに指定した等高線とゾーンをプロットキューブの天井面または床面に表示できます。グラフの上部と下部に等高線とゾーンを表示することで、3D モデルに 2D 表現も付加します。例については、「**床面と天井面の追加**」を参照してください。

- **2D 投影の表示**

  ヒートマップなどのフラットな（2次元）3D グラフを簡単に作成します。詳細については、「**2次元グラフの作成**」を参照してください。

- **カスタム軸注釈の作成**

  AnnoTemplate プロパティを使用して、軸ラベルのレイアウトと値を正確にカスタマイズできます。詳細については、「**カスタムの軸注釈テンプレート**」を参照してください。

クイックスタート

このクイックスタートは、**Chart3D for WPF** を初めて使用するユーザーのために用意されています。このクイックスタートでは、標準的なグラフを作成するための基本手順について説明します。これには、プロジェクトへの 3D グラフの追加、グラフへのデータの追加、グラフタイプの選択、グラフの外観の変更、およびグラフ凡例の追加が含まれます。

このクイックスタートは、Visual Studio で使用される手順を提供します。手順が多少異なる場合がありますが、Microsoft Blend で XAML を使用することもできます。

手順 1:プロジェクトへのコントロールの追加

この手順では、最初に Visual Studio または Blend で **Chart3D for WPF/Silverlight** を使用するグラフアプリケーションを作成します。

Visual Studio プロジェクトに Chart3D for WPF/Silverlight を追加するには、次の手順に従います。

1. Visual Studio で新しい WPF/Silverlight プロジェクトを作成します。
2. ツールボックスの**C1Chart3D** コントロールをダブルクリックしてウィンドウに追加します。XAML マークアップは次のようになります。

WPF

XAML

```
<Window x:Class="WFPChart3DQuickstart.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="350" Width="525"
        xmlns:c1chart3d="http://schemas.componentone.com/xaml/c1chart">
    <Grid>
        <c1chart3d:C1Chart3D Height="150" HorizontalAlignment="Left"
         Margin="10,10,0,0" Name="c1Chart3D1" VerticalAlignment="Top"
Width="200">
            <c1chart3d:GridDataSeries ZDataString="1 1 1,2 2 2,3 3 3" />
        </c1chart3d:C1Chart3D>
    </Grid>
</Window>
```

Silverlight

XAML

```
<UserControl xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"
x:Class="MySilverlightApplication.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    d:DesignHeight="300" d:DesignWidth="400">
    <Grid x:Name="LayoutRoot" Background="White">
```

```
<c1:C1Chart3D />
    </Grid>
</UserControl>
```

次の手順では、グラフに独自のデータを追加します。

手順 2：データの追加

Chart3D for WPF/Silverlight では、最初のインデックスが X に対応し、2番目のインデックスが Y に対応する z 値の2次元配列として定義されたデータ値がサポートされます。Chart3D の最も一般的な使用方法の1つは、3D 関数のプロットです。次のように定義された関数をプロットしてみます。

z(x,y) = x*x - y*y;

変域は次のとおりです。

-1 <= x <= 1

-1 <= y <= 1

それには、次の手順に従います。

1. プロジェクトで[表示]→[コード]を選択します。
2. ページの先頭に次の文を追加します。

WPF

C#

```
using C1.WPF.C1Chart3D;
```

Silverlight

C#

```
using C1.Silverlight.Chart3D;
```

3. 次のコードを追加してデータを定義します。

C#

```
public MainWindow()
        {
            InitializeComponent();
            c1Chart3D1.Children.Clear();
            // 10 x 10 の 2D 配列を作成します
            int xlen = 10, ylen = 10;
            var zdata = new double[xlen, ylen];
            double stepx = 2.0 / (xlen - 1);
            double stepy = 2.0 / (ylen - 1);
            // 範囲内のすべてのポイントで関数を計算します
            for (int ix = 0; ix < xlen; ix++)
                for (int iy = 0; iy < ylen; iy++)
                {
                    double x = -1.0 + ix * stepx; //  -1 <= x <= 1
```

```
                    double y = -1.0 + iy * stepy; //  -1 <= x <= 1
                    zdata[ix, iy] = x * x - y * y;
                }
            // データ系列を作成します
            var ds = new GridDataSeries();
            ds.Start = new Point(-1, -1); // x, y の開始値
            ds.Step = new Point(stepx, stepy); // x, y の1目盛り
            ds.ZData = zdata; // z-values
            // 系列をグラフに追加します
            c1Chart3D1.Children.Add(ds);
        }
```

次の手順では、グラフの外観を変更します。

手順 3:グラフの外観の変更

この手順では、**SurfaceZone** へのグラフタイプの変更、グラフ床面の外観の設定、グラフの回転、およびグラフの仰角の変更を行います。

1. ウィンドウで**C1Chart3D** コントロールを選択し、[**表示**]→[**プロパティ**]ウィンドウをクリックします。
2. **ChartType** プロパティの横にあるドロップダウン矢印をクリックし、**SurfaceZone** を選択して、グラフタイプを変更します。
3. **FloorAppearance** プロパティの横にあるドロップダウン矢印をクリックし、[**ZoneContour**]を選択します。
4. **Elevation** プロパティの隣に「**170**」と入力します。
5. **Azimuth** プロパティを **20** に設定します。

   XAML マークアップは次のようになります。

WPF

XAML

```
<Window x:Class="WFPChart3DQuickstart.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="350" Width="525"
        xmlns:c1chart3d="http://schemas.componentone.com/xaml/c1chart">
    <Grid>
        <c1chart3d:C1Chart3D Height="150" HorizontalAlignment="Left"
Margin="10,10,0,0"
         Name="c1Chart3D1" VerticalAlignment="Top" Width="200"
ChartType="SurfaceZone"
         CeilAppearance="None" FloorAppearance="ZoneContour" Elevation="170"
Azimuth="20">
            <c1chart3d:GridDataSeries ZDataString="1 1 1,2 2 2,3 3 3" />
        </c1chart3d:C1Chart3D>
    </Grid>
</Window>
```

Silverlight

XAML

```
<UserControl xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"
x:Class="SilverlightApplication12.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    d:DesignHeight="300" d:DesignWidth="400">
    <Grid x:Name="LayoutRoot" Background="White">
<c1:C1Chart3D ChartType="SurfaceZone" FloorAppearance="ZoneContour" Azimuth="20"
Elevation="170">
    <c1:C1Chart3D />
  </Grid>
</UserControl>
```

次の手順では、グラフに凡例を追加します。

手順 4：凡例の追加

この手順では、1つのプロパティのみを使用してグラフの凡例を追加します。

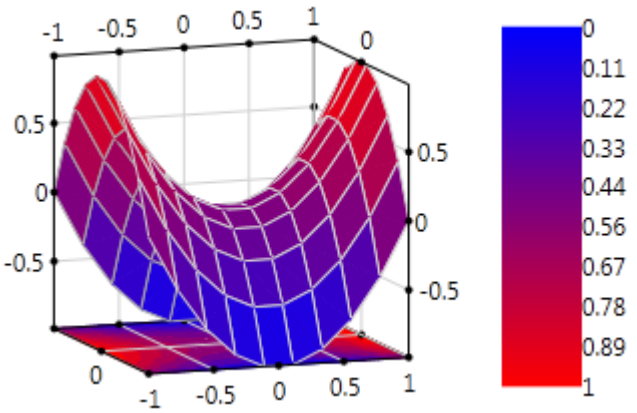
1. ウィンドウで**C1Chart3D** コントロールを選択し、[**表示**]→[**プロパティ**]ウィンドウをクリックします。
2. **Legend** プロパティの横にあるドロップダウン矢印をクリックし、**C1Chart3DLegend** を選択して、凡例を追加します。

次の手順では、プロジェクトを実行し、新しいグラフを表示します。

手順 5：プロジェクトの実行

この手順では、プロジェクトを実行してグラフを見ます。

[**スタート**]→[**デバッグ**]を選択するか、[**F5**]キーを押します。Chart3D と凡例が表示されます。



おめでとうございます。これで、**Chart3D for WPF/Silverlight** クイックスタートは終了です。

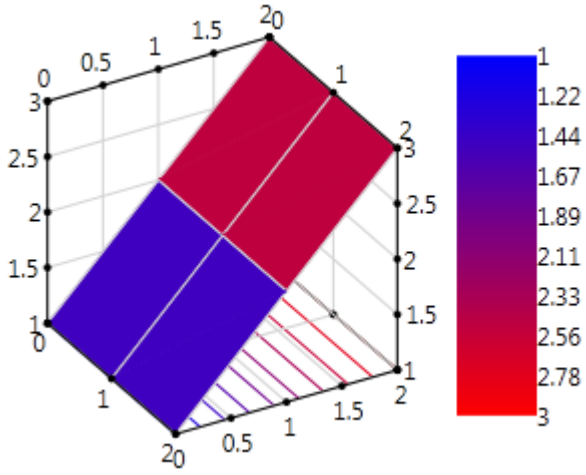
XAML クイックリファレンス

このセクションでは、XAML コードのみを使用して、WPF **C1Chart3D** コントロールを使用する方法を示すいくつかの例を提供します

例:3D 面グラフの設定

次の XAML は、**C1Chart3D** コントロールを宣言し、**ChartType**、**FloorAppearance**、**GridDataSeries** を設定し、**C1Chart3DLegend** コントロールを追加します。この XAML は、<Grid>...</Grid> タグ内に追加できます。

次のグラフは、以下の XAML コードを使用して生成されます。



XAML

```
<Grid>
    <c1chart3d:C1Chart3D Name="c1Chart3D1" ChartType="SurfaceZone"
FloorAppearance="Contour">
        <c1chart3d:C1Chart3D.Legend>
            <c1chart3d:C1Chart3DLegend />
        </c1chart3d:C1Chart3D.Legend>
        <c1chart3d:GridDataSeries ZDataString="1 1 1,2 2 2,3 3 3" />
    </c1chart3d:C1Chart3D>
</Grid>
```

DirectX レンダリング (WPF のみ)

DirectX レンダリングモードを使用すると、大規模データセットを処理する面グラフで Direct3D レンダリングの高パフォーマンスレンダリングを実現できます。DirectX レンダリングモードを実装するには、C1Chart3D.**RenderMode** プロパティを **Direct3D** に設定するだけです。

WPF アプリケーションで Direct3D レンダリングモードを使用するには、アプリケーションと共に以下のアセンブリを配布する必要があります。開発中は、これらのアセンブリを C1.WPF.C1Chart3D アセンブリと同じディレクトリに置く必要がありますが、プロジェクトから直接参照する必要はありません。

- SharpDX.D3DCompiler.dll
- SharpDX.Direct3D9.dll
- SharpDX.Direct3D10.dll
- SharpDX.dll
- SharpDX.DXGI.dll

Direct3D レンダリングには以下のシステム要件があります。

- DirectX Version 10.0 以上
- D3D9 オーバーレイのサポート

一部の仮想マシン（Hyper-V）やリモートデスクトップ接続では、必要なハードウェアレイヤがサポートされないため、Direct3D モードは動作しません。

メモ:このトピックの内容は、ComponentOne for WPF にのみ適用されます。

C1Chart3D for WPF の使い方