|  |
| --- |
| 快速开始 |

本快速入门教程介绍了一些 Win2D 的基本功能。您将学习如何:

* 将 Win2D 添加到 XAML VB Windows 10 项目
* 绘制文本和几何
* 应用滤镜效果
* 对您的 Win2D 内容进行动画处理
* 按照 Win2D 的最佳做法

### 安装 Visual Studio

* 如果你没有安装 Visual Studio 的支持的版本，请按照此处的步骤:[获取设置](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FIntroduction.htm%23GettingSetUp).

### 使用 Win2D 创建一个新的项目

1. 启动 Visual Studio 中，并创建一个新的项目:**文件 → 新项目**.
2. 如果你针对 Windows 10，选择**安装 → 模板 → Visual Basic Windows 通用 → 空白应用程序 (通用 Windows)**.
3. 为该项目名称，选择一个位置，并创建它。
4. Win2D 释放作为一个 Nuget.org 包，需要安装之前您可以使用它。有两个包版本，本教程使用 [Windows 10](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fwww.nuget.org%2Fpackages%2FWin2D.uwp%2F)。通过**工具 → NuGet 程序包管理器 → 管理 NuGet 程序包的解决方案**去启动 NuGet 程序包管理器.
5. 搜索"Win2D"并选择相应的包为 Windows 10 的目标版本。

选择**Win2D.uwp**.

最后，单击**安装**。如果系统提示您若要查看更改，单击**确定**。如果你介绍与 Win2D 许可条款，请单击**我接受**.

Win2D 现在已安装在您的项目。

### 向您的应用程序 XAML 添加 Win2D CanvasControl

1. 为了使用 Win2D，您需要绘制图形的地方。在 XAML 应用程序，这样做的最简单方法是[CanvasControl](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FT_Microsoft_Graphics_Canvas_UI_Xaml_CanvasControl.htm" \t "_top)添加到 XAML 页面。

Win2D 在 c++ 中实现，因此使用 Win2D 的项目需要指向一个特定的 CPU 体系结构。

1. 导航到**MainPage.xaml**您的项目中通过在**解决方案资源管理器**中双击它。这将打开该文件。为方便起见，您可以双击**XAML**按钮在**设计器**选项卡这将隐藏可视设计器中，并保留所有的空间的代码视图。
2. 你添加的控件之前，你首先要告诉 XAML CanvasControl定义的位置。要做到这一点，转到的页面元素，定义并添加此语句: xmlns:canvas="using:Xaml"。您的 XAML 现在应该像这样:[副本](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FQuickStart.htm%23)

<Page

...

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:canvas="using:Xaml"

mc:Ignorable="d">

1. 现在，将新的画布: CanvasControl作为子元素添加到根网格元素。为该控件指定一个名称，例如"画布"。您的 XAML 现在应该像这样:[副本](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FQuickStart.htm%23)

<Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">

<canvas:CanvasControl x:Name="canvas"/>

</Grid>

1. 接下来，定义[绘制](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FE_Microsoft_Graphics_Canvas_UI_Xaml_CanvasControl_Draw.htm" \t "_top)事件的事件处理程序。CanvasControl提出了绘制每当您的应用程序需要绘制或重新绘制其内容。最简单的方法是让视觉工作室自动完成协助你。在CanvasControl定义中，开始键入一个新的属性绘制事件处理程序:[副本](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FQuickStart.htm%23)

<canvas:CanvasControl x:Name="canvas" Draw="canvas\_Draw" />

一旦您输入了在绘制 ="，Visual Studio 应该弹出一个框提示您让它自动填写正确的定义事件处理程序。按**TAB**可接受 Visual Studio 默认事件处理程序。这将也自动格式正确的事件处理程序方法在代码中添加背后 (**MainPage.xaml.vb**)。不要担心，如果你没有使用自动完成在下一步中，您可以手动添加事件处理程序方法。

### 在 Win2D 中绘制你第一次的文本

1. 现在，让我们去后面的 VB 代码。从**解决方案资源管理器**打开**MainPage.xaml.vb**.
2. 顶部的 VB 文件是不同的命名空间定义。添加以下命名空间:[副本](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FQuickStart.htm%23)

Imports Microsoft.Graphics.Canvas

Imports Microsoft.Graphics.Canvas.Effects

Imports Microsoft.Graphics.Canvas.UI

Imports Microsoft.Graphics.Canvas.UI.Xaml

Imports Windows.UI

接下来，您应该看到下面的空事件处理程序被插入的自动完成:

Private Sub canvas\_Draw(sender As CanvasControl, args As CanvasDrawEventArgs)

End Sub

1. (如果你没有使用自动完成上一步中，添加此代码现在)。
2. [CanvasDrawEventArgs](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FT_Microsoft_Graphics_Canvas_UI_Xaml_CanvasDrawEventArgs.htm)参数公开一名成员， DrawingSession，这是[CanvasDrawingSession](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FT_Microsoft_Graphics_Canvas_CanvasDrawingSession.htm" \t "_top)类型。此类提供最基本的绘图功能，在 Win2D: 它具有的方法如[CanvasDrawingSession. DrawRectangle](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FOverload_Microsoft_Graphics_Canvas_CanvasDrawingSession_DrawRectangle.htm), [CanvasDrawingSession. DrawImage](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FOverload_Microsoft_Graphics_Canvas_CanvasDrawingSession_DrawImage.htm)和你需要绘制文本的方法[CanvasDrawingSession. DrawText](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FOverload_Microsoft_Graphics_Canvas_CanvasDrawingSession_DrawText.htm" \t "_top).

将下面的代码添加到canvas\_Draw方法:

args.DrawingSession.DrawText("Hello, World!", 100, 100, Colors.Black)

第一个参数，"**你好，世界!**"，是你想要 Win2D 要显示的字符串。两个"**100**"s 告诉 Win2D 来抵消此文本由 100 蘸 (独立于设备的像素) 向右和向下。最后，"**Colors.Black**"定义了文本的颜色。

1. 你现在准备好运行您第一次的 Win2D 应用程序按**F5**键编译并启动。您应该看到一个空窗口与"你好，世界!"在黑色。

### 正确处置 Win2D 资源

1. 在继续之前上绘制其他种类的内容，你首先应该添加一些代码来确保您的应用程序可以避免内存泄漏。大多数的 Win2D 应用程序写在.NET 语言和使用 Win2D 控制像[CanvasControl](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FT_Microsoft_Graphics_Canvas_UI_Xaml_CanvasControl.htm" \t "_top)需要遵循以下步骤。严格说来，你简单的"Hello，world"应用程序不受影响，但这是一个好的习惯，一般遵循。

有关详细信息，请参阅[避免内存泄漏](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FRefCycles.htm).

1. 打开**MainPage.xaml** ，找到包含您的**CanvasControl**的**页**XAML 元素。它应该是在该文件中的第一个元素。
2. 添加**卸载**事件的处理程序。你的 XAML 应该如下所示:[副本](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FQuickStart.htm%23)

<Page

...

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:canvas="using:Xaml"

mc:Ignorable="d"

Unloaded="Page\_Unloaded">

1. 去**MainPage.xaml.vb** ，找到**Page\_Unloaded**事件处理程序。添加以下代码:[副本](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FQuickStart.htm%23)

Sub Page\_Unloaded(sender As Object, e As RoutedEventArgs)

Me.canvas.RemoveFromVisualTree()

Me.canvas = Nothing

End Sub

1. 如果您的应用程序包含多个 Win2D 控件，然后你需要为每个包含一个 Win2D 控件的 XAML 页重复上述步骤。您的应用程序，目前仅有单一的**CanvasControl**所以你所有的事情。

### 画一些图形

1. 它是一样容易将二维几何添加到您的应用程序添加下面的代码到canvas\_Draw的末尾: [副本](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FQuickStart.htm%23)

args.DrawingSession.DrawCircle(125, 125, 100, Colors.Green)

args.DrawingSession.DrawLine(0, 0, 50, 200, Colors.Red)

这两种方法的参数是类似于DrawText。一圈是由中心点 (**125、 125**)，半径 (**100**) 和一种颜色 (**绿色**) 定义的。一条线是由 (**0，0**) 的开始、 结束 (**50、 200**) 和一种颜色 (**红色**定义的。).

1. 现在，按**f5 键**运行应用程序。您应该看到"你好，世界!"以及一个绿色的圆和红线。

你可能想知道如何控制更先进的绘图选项，如线条的粗细和短划线，或更复杂的填充选项，如使用画笔。Win2D 提供的所有这些选项和更多，和容易地使用它们的时候你想要。所有的画 [......]的方法提供许多重载方法可以接受额外的参数，如[CanvasTextFormat](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FT_Microsoft_Graphics_Canvas_Text_CanvasTextFormat.htm" \t "_top) (字体、 大小等) 和[CanvasStrokeStyle](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FT_Microsoft_Graphics_Canvas_Geometry_CanvasStrokeStyle.htm" \t "_top) (短划线、 圆点、 末端等)。感到自由探索 API 表面，以了解有关这些选项的详细信息。

### 动态生成的绘图参数

1. 现在，让我们添加一些品种通过绘制一束的形状和文本与随机颜色。

将下面的代码添加到您的主页类的顶部。这是生成的随机值，绘制时将使用的帮助器功能:

Dim rnd As New Random

Private Function RndPosition() As Vector2

Dim x = rnd.NextDouble() \* 500.0F

Dim y = rnd.NextDouble() \* 500.0F

Return New Vector2(x, y)

End Function

Private Function RndRadius() As Single

Return rnd.NextDouble() \* 150.0F

End Function

Private Function RndByte() As Byte

Return rnd.Next(256)

End Function

1. 修改您的canvas\_Draw方法来使用这些随机参数绘制:[副本](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FQuickStart.htm%23)

Private Sub canvas\_Draw(sender As CanvasControl, args As CanvasDrawEventArgs)

args.DrawingSession.DrawText("Hello, World!", RndPosition(), Color.FromArgb(255, RndByte(), RndByte(), RndByte()))

args.DrawingSession.DrawCircle(RndPosition(), RndRadius(), Color.FromArgb(255, RndByte(), RndByte(), RndByte()))

args.DrawingSession.DrawLine(RndPosition(), RndPosition(), Color.FromArgb(255, RndByte(), RndByte(), RndByte()))

End Sub

让我们打破DrawText有怎样的变化。"**你好，世界!**"仍然和以前一样。X 和 y 偏移量的参数已经被取代单一的[System.Numerics.Vector2](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=https%3A%2F%2Fmsdn.microsoft.com%2Flibrary%2Fwindows%2Fapps%2FSystem.Numerics.Vector2" \t "_top)是由RndPosition产生。最后，而不是使用预定义的颜色， Color.FromArgb允许您定义一种颜色使用 A、 R、 G 和 B 值。A 是 alpha 通道或不透明度级别在这种情况下，你总是想要完全不透明 (**255**).

画圆和DrawLine同样经营往返DrawText.

1. 最后，将您的绘图代码包装在for循环。你应该会得到下面的canvas\_Draw代码:[副本](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FQuickStart.htm%23)

For i = 0 To 99

args.DrawingSession.DrawText("Hello, World!", RndPosition(), Color.FromArgb(255, RndByte(), RndByte(), RndByte()))

args.DrawingSession.DrawCircle(RndPosition(), RndRadius(), Color.FromArgb(255, RndByte(), RndByte(), RndByte()))

args.DrawingSession.DrawLine(RndPosition(), RndPosition(), Color.FromArgb(255, RndByte(), RndByte(), RndByte()))

Next

再次运行该应用程序。您应该看到一大堆的文本、 线条和圆圈与随机位置和大小。

### 将图像效果应用到您的内容

1. 图像效果，也称为滤镜效果，是应用于像素数据的图形转换。饱和度、 色相旋转和高斯模糊是一些常见的图像效果。图像效果可以链接在一起，生产精致的视觉外观为最小的努力。

通过提供一个源图像 (你开始使用的内容)、 创建效果如[GaussianBlurEffect](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FT_Microsoft_Graphics_Canvas_Effects_GaussianBlurEffect.htm)，设置属性，如[BlurAmount](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FP_Microsoft_Graphics_Canvas_Effects_GaussianBlurEffect_BlurAmount.htm" \t "_top)，然后绘制效果的输出与DrawImage使用图像效果.

要应用到您的文本和形状的图像效果，您需要首先将该内容呈现到[CanvasCommandList](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FT_Microsoft_Graphics_Canvas_CanvasCommandList.htm)。此对象是用作输入到你的效果。

1. 更改您的canvas\_Draw方法来使用下面的代码:[副本](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FQuickStart.htm%23)

Dim cl As New CanvasCommandList(sender)

Using clds = cl.CreateDrawingSession()

For i = 0 To 99

clds.DrawText("Hello, World!", RndPosition(), Color.FromArgb(255, RndByte(), RndByte(), RndByte()))

clds.DrawCircle(RndPosition(), RndRadius(), Color.FromArgb(255, RndByte(), RndByte(), RndByte()))

clds.DrawLine(RndPosition(), RndPosition(), Color.FromArgb(255, RndByte(), RndByte(), RndByte()))

Next

End Using

就像如何从CanvasDrawEventArgs您可以绘制与获得的CanvasDrawingSession ，你可以从CanvasCommandList创建的CanvasDrawingSession 。唯一的区别是，当你画到命令列表绘图会话 (clds)，你不直接渲染到CanvasControl。相反，命令列表是渲染的一个中间对象来存储供以后使用的结果。

你可能已经注意到包装命令列表绘图会话使用的块。绘图会话实现[idisposable 可以根据](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmsdn.microsoft.com%2Flibrary%2Fsystem.idisposable)和必须释放时，你做的渲染 ( using块这样做)。对你，但你必须处置您显式创建的任何绘图会话被闭合的你自动获取从CanvasDrawEventArgs CanvasDrawingSession.

1. 最后，定义GaussianBlurEffect通过向canvas\_Draw方法的末尾，添加以下代码:[副本](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FQuickStart.htm%23)

Dim blur As New GaussianBlurEffect

blur.Source = cl

blur.BlurAmount = 10.0F

args.DrawingSession.DrawImage(blur)

此代码创建一个高斯模糊，设置其源，你只是提请， CanvasCommandList将其模糊半径设置为**10**，以及最后呈现的输出到原始绘图会话模糊 (参数。DrawingSession).

1. 再次运行该应用程序。您应该看到你的线条、 文本和出现模糊的圆。

### 对您的应用程序与 CanvasAnimatedControl 进行动画处理

1. Win2D 使您能够更新和实时，您的内容进行动画处理，例如可通过改变每一帧与高斯模糊的模糊半径。要做到这一点，您将使用[CanvasAnimatedControl](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FT_Microsoft_Graphics_Canvas_UI_Xaml_CanvasAnimatedControl.htm).

CanvasControl是最适合主要是静态图形内容 — — 它只绘制时引发事件您的内容需要更新或重新绘制。如果你有不断变化的内容然后你应该考虑改用CanvasAnimatedControl 。这两个控件操作非常类似，除了CanvasAnimatedControl引发绘制事件定期默认情况下，它被称为每秒 60 次。

1. 若要切换到CanvasAnimatedControl，请转到**MainPage.xaml**，删除CanvasControl线，并替换下面的 XAML:[副本](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FQuickStart.htm%23)

<Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">

<canvas:CanvasAnimatedControl x:Name="canvas" Draw="canvas\_DrawAnimated" CreateResources="canvas\_CreateResources"/>

</Grid>

就像CanvasControl，让自动完成创建为你绘制事件处理程序。默认情况下，Visual Studio 将名称此处理程序canvas\_Draw\_1因为canvas\_Draw已存在在这里，我们已经重命名方法canvas\_AnimatedDraw要弄清楚，这是一个不同的事件。

此外，您也正在处理一个新的事件， [CreateResources](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FE_Microsoft_Graphics_Canvas_UI_Xaml_CanvasAnimatedControl_CreateResources.htm)。再次，让自动完成创建处理程序。

现在，您的应用程序将在每秒 60 帧重绘，是更有效地创建您的 Win2D 视觉资源一次并重新使用它们的每一帧。它是低效创建的CanvasCommandList和 300 元素引入 60 次每秒当内容仍然是静态的。CreateResources是只有 Win2D 确定您需要重新创建您视觉资源，例如当加载页面时，将触发一个事件。

1. 切换到**MainPage.xaml.vb**。找到你的canvas\_Draw方法，应该看起来像这样:[副本](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FQuickStart.htm%23)

Private Sub canvas\_Draw(sender As CanvasControl, args As CanvasDrawEventArgs)

Dim cl = New CanvasCommandList(sender)

Using clds = cl.CreateDrawingSession()

For i = 0 To 99

clds.DrawText("Hello, World!", RndPosition(), Color.FromArgb(255, RndByte(), RndByte(), RndByte()))

clds.DrawCircle(RndPosition(), RndRadius(), Color.FromArgb(255, RndByte(), RndByte(), RndByte()))

clds.DrawLine(RndPosition(), RndPosition(), Color.FromArgb(255, RndByte(), RndByte(), RndByte()))

Next

End Using

Dim blur = New GaussianBlurEffect()

blur.Source = cl

blur.BlurAmount = 10.0F

args.DrawingSession.DrawImage(blur)

End Sub

绝大多数的此现有的绘制代码不需要执行与每一帧: 包含文本、 线条和圆圈的命令列表保持不变，每一帧，和唯一的变化是模糊半径。因此，你可以将这个"静态"的代码移动到CreateResources.

做到这一点，首先剪切 (**CTRL + X**) canvas\_Draw，除了最后一行的全部内容 (参数。DrawingSession.DrawImage(blur))。你现在可以删除剩余的canvas\_Draw ，因为它不再需要: 记得CanvasAnimatedControl有其自己独特绘制事件。

1. 找到自动生成的canvas\_CreateResources的方法:[副本](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FQuickStart.htm%23)

Private Sub canvas\_CreateResources(sender As CanvasAnimatedControl, args As CanvasCreateResourcesEventArgs)

End Sub

1. (**CTRL + V**) 你原先被削减的代码粘贴到此方法。接下来，移动宣言 》 的GaussianBlurEffect方法体的外部，所以该变量成为主页类的一个成员。现在，您的代码应该类似于以下内容:[副本](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FQuickStart.htm%23)

Dim blur As GaussianBlurEffect

Private Sub canvas\_CreateResources(sender As CanvasAnimatedControl, args As CanvasCreateResourcesEventArgs)

Dim cl = New CanvasCommandList(sender)

Using clds = cl.CreateDrawingSession()

For i = 0 To 99

clds.DrawText("Hello, World!", RndPosition(), Color.FromArgb(255, RndByte(), RndByte(), RndByte()))

clds.DrawCircle(RndPosition(), RndRadius(), Color.FromArgb(255, RndByte(), RndByte(), RndByte()))

clds.DrawLine(RndPosition(), RndPosition(), Color.FromArgb(255, RndByte(), RndByte(), RndByte()))

Next

End Using

Dim blur = New GaussianBlurEffect() With {.Source = cl, .BlurAmount = 10.0F}

End Sub

1. 现在您可以创建动画的高斯模糊。查找canvas\_AnimatedDraw方法并添加下面的代码:[副本](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FQuickStart.htm%23)

Private Sub canvas\_AnimatedDraw(sender As ICanvasAnimatedControl, args As CanvasAnimatedDrawEventArgs)

Dim radius = (1 + Math.Sin(args.Timing.TotalTime.TotalSeconds)) \* 10.0F

blur.BlurAmount = radius

args.DrawingSession.DrawImage(blur)

End Sub

这读取由[CanvasAnimatedDrawEventArgs](http://www.microsofttranslator.com/bv.aspx?from=en&to=zh-CHS&a=http%3A%2F%2Fmicrosoft.github.io%2FWin2D%2Fhtml%2FT_Microsoft_Graphics_Canvas_UI_Xaml_CanvasAnimatedDrawEventArgs.htm" \t "_top)提供的总运行时间并使用这来计算所需的模糊程度正弦函数提供了有趣的变化，随着时间的推移。最后， GaussianBlurEffect被呈现。

1. 运行应用程序看到随着时间的推移，模糊的含量变化。

祝贺您完成这个快速开始教程