Visual Studio 2019

Zuletzt verwendetes öffnen

Diese Woche

MeineHakse.sln 30. 05. 2022 19:51
C:\Users\plane\OneDrive\Desktop\MeineHakse

ewigeGesundheit.sln 30. 05. 2022 05:13
C:\Users\plane\source\repos\ewigeGesundheit

unitedartist.sln 30. 05. 2022 04:55
C:\Users\plane\source\repos\unitedartist

Neues Projekt erstellen
Wählen Sie zu Beginn eine Projektvorlage mit Codegerüstbau.

Los geht's

Repository klonen
Ruft Code aus einem Onlinerepository (z. B. GitHub oder Azure DevOps) ab.

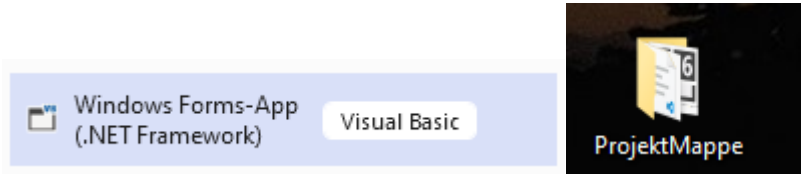
Projekt oder Projektmappe öffnen
Hiermit öffnen Sie ein lokales Visual Studio-Projekt oder eine SLN-Datei.

Lokalen Ordner öffnen
Navigieren und Bearbeiten von Code in beliebigem Ordner

Neues Projekt erstellen
Wählen Sie zu Beginn eine Projektvorlage mit Codegerüstbau.

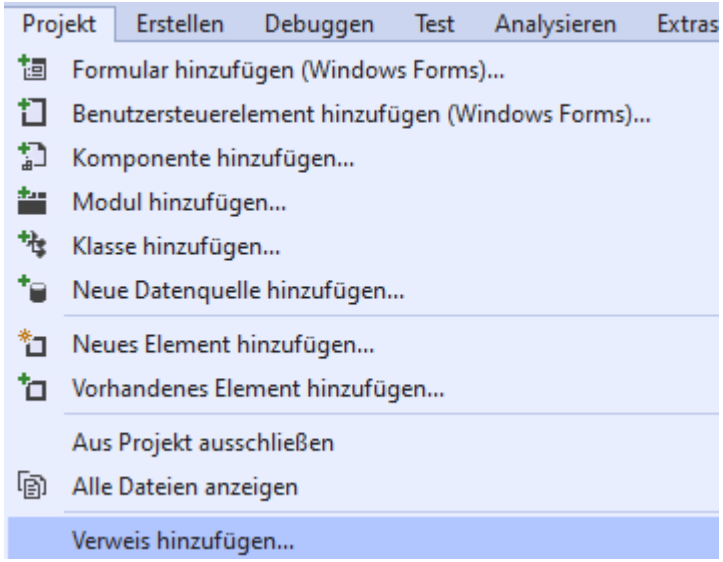
Ohne Code fortfahren →

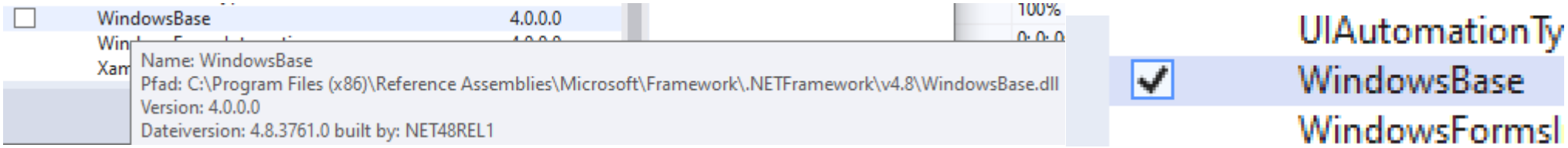
Neues Projekt erstellen
Wählen Sie zu Beginn eine Projektvorlage mit Codegerüstbau.

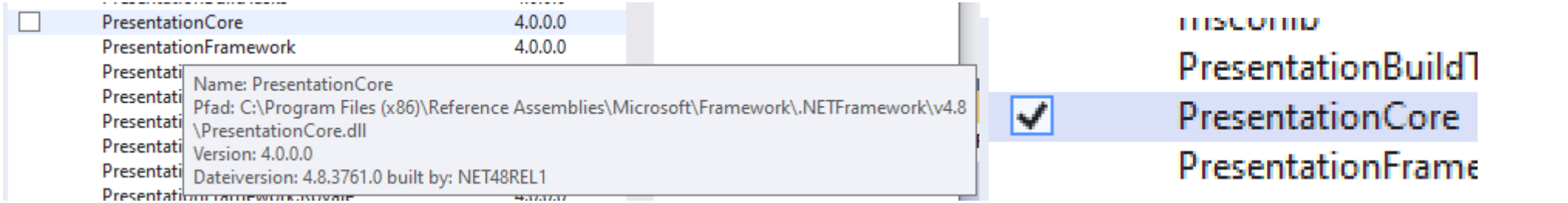


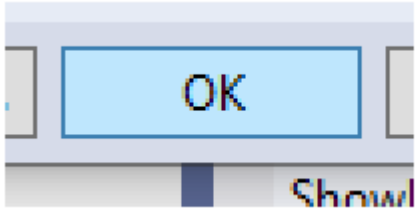
Of course, you want to put the code of the AnimatedGifEncoder in it's own DLL since it need a reference to "PresentationCore" and "WindowsBase" and you don't want to expose those namespace in your application.

Ist nicht notwendig, weil die Verschlüsselungsebenen Sicher genug sein sollten.



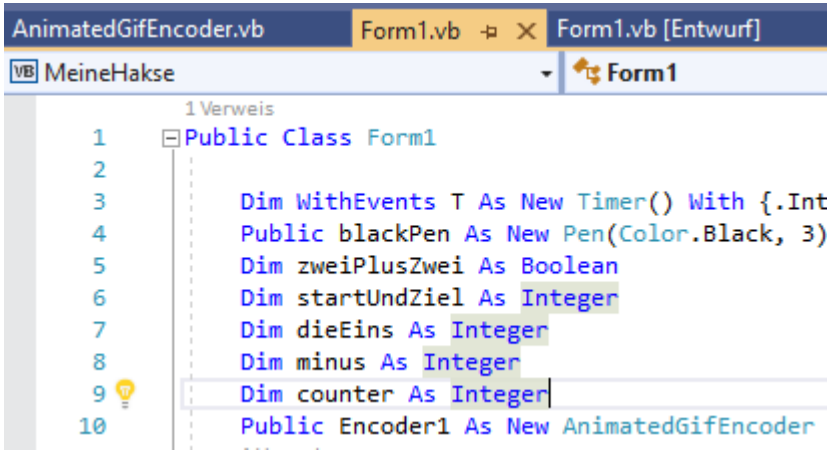


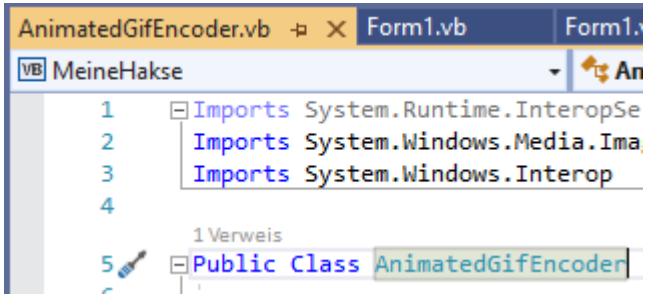




?

Also you need to import "System.Windows.Media.Imaging" and "System.Windows.Interop" in the AnimatedGifEncoder class






```
Imports System.Runtime.InteropServices
Imports System.Windows.Media.Imaging
Imports System.Windows.Interop


Public Class AnimatedGifEncoder


    Private GifEncoder As New
System.Windows.Media.Imaging.GifBitmapEncoder


    ''' <summary>
    ''' Return the GIF specification version. This always
returns "GIF89a"
```

```
    ''' </summary>
    Public ReadOnly Property EncoderVersion As String
        Get
            Return "GIF89a"
        End Get
    End Property

    ''' <summary>
    ''' Get or set a value that indicate if the GIF will repeat the animation after the last frame is shown. The default value is True
    ''' </summary>
    Public Property Repeat As Boolean = True

    ''' <summary>
```

```
    ''' Get or set a collection of metadata string to be
embedded in the GIF file. Each string has a max length of 254
    ''' characters (Any character above this limit will be
truncated). The string will be encoded UTF-7.
    ''' </summary>
    Public Property MetadataString As List(Of String) = New
List(Of String)


    ''' <summary>
    '''  Get or set the amount of time each frame will be shown
(in milliseconds). The default value is 200ms
    ''' </summary>
    Public Property FrameRate As Integer = 200


    ''' <summary>
```

```
    ''' Add a frame to the encoder frame collection
    ''' </summary>
    ''' <param name="Frame">The bitmap to be added</param>
    Public Sub AddFrame(Frame As Bitmap)
        If Frame IsNot Nothing Then
            If Not (Frame.Width = 0) And Not (Frame.Height = 0)
Then
                Dim bmpSource =
Imaging.CreateBitmapSourceFromHBitmap(Frame.GetHbitmap,
                                      IntPtr.Zero,
                                      Windows.Int32Rect.Empty,

BitmapSizeOptions.FromEmptyOptions)

GifEncoder.Frames.Add(BitmapFrame.Create(bmpSource))
```

```
            Else
                Throw New ArgumentException("Argument Frame,
The bitmap size cannot be zero")
            End If
        Else
            Throw New ArgumentException("Argument Frame cannot
be nothing")
        End If
    End Sub

    ''' <summary>
    ''' Writes the animated GIF binary to a specified IO.Stream
    ''' </summary>
```

```
    ''' <param name="Stream">The stream where the binary is to
be output. Can be any object type that derives from
IO.Stream</param>
    Public Sub Save(Stream As IO.Stream)


        Dim Data() As Byte
        If Not GifEncoder.Frames.Count = 0 Then
            'Get the raw binary
            Using MStream As New IO.MemoryStream
                GifEncoder.Save(MStream)
                Data = MStream.ToArray
            End Using
        Else
            Throw New Exception("Cannot encode the Gif. The
frame collection is empty.")
```

```
            '   Only documented exception is if Frames.count=0
        End If


        'Locate the right location where to insert the metadata
in the binary
        'This will be just before the first label &H0021F9
(Graphic Control Extension)
        Dim MetadataPTR As Integer = -1
        Dim flag As Integer = 0
        Do
            MetadataPTR += 1
            If Data(MetadataPTR) = 0 Then
                If Data(MetadataPTR + 1) = &H21 Then
                    If Data(MetadataPTR + 2) = &HF9 Then
                        flag = 1
```

```
                    End If
                End If
            End If
        Loop While flag = 0

        'SET METADATA Repeat
        'This add an Application Extension Netscape2.0
        If Repeat Then
            Dim Temp(CInt(Data.Length) - 1 + 19) As Byte
            'label: &H21, &HFF + one byte: length(&HB) +
NETSCAPE2.0 + one byte: Datalength(&H3) + {1, 0, 0} + Block
terminator, 1 byte, &H00
            Dim ApplicationExtension() As Byte = {&H21, &HFF,
&HB, &H4E, &H45, &H54, &H53, &H43, &H41, &H50, &H45, &H32,
&H2E, &H30, &H3, &H1, &H0, &H0, &H0}
```

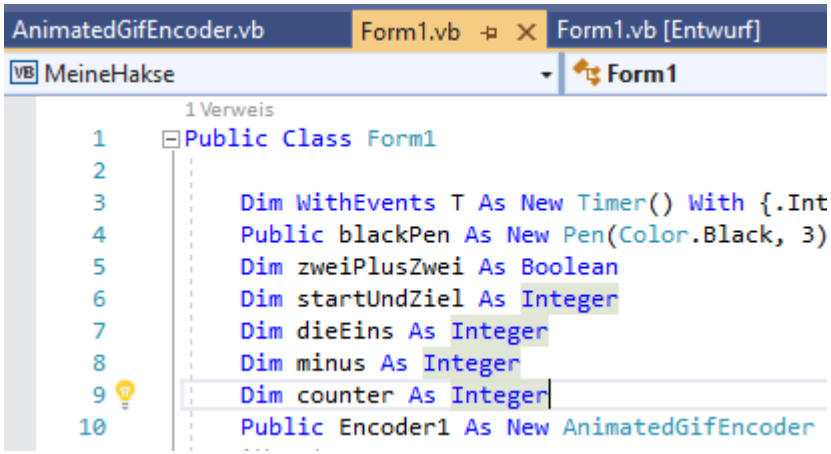
```
            Array.Copy(Data, Temp, MetadataPTR)
            Array.Copy(ApplicationExtension, 0, Temp,
MetadataPTR + 1, 19)
            Array.Copy(Data, MetadataPTR + 1, Temp, MetadataPTR
+ 20, Data.Length - MetadataPTR - 1)
            Data = Temp
        End If

        'SET METADATA Comments
        'This add a Comment Extension for each string
        If MetadataString.Count > 0 Then
            For Each Comment As String In MetadataString
                If Not String.IsNullOrEmpty(Comment) Then
                    Dim TheComment As String
                    If Comment.Length > 254 Then
```

```
                    TheComment = Comment.Substring(0, 254)
                Else
                    TheComment = Comment
                End If
                Dim CommentStringBytes() As Byte =
System.Text.UTF7Encoding.UTF7.GetBytes(TheComment)
                Dim DataString() As Byte = New Byte()
{&H21, &HFE, CByte(CommentStringBytes.Length)}
                DataString =
DataString.Concat(CommentStringBytes).Concat(New Byte()
{&H0}).ToArray
                Dim Temp(Data.Length - 1 +
DataString.Length) As Byte
                Array.Copy(Data, Temp, MetadataPTR)
                Array.Copy(DataString, 0, Temp, MetadataPTR
+ 1, DataString.Length)
```

```
                    Array.Copy(Data, MetadataPTR + 1, Temp,
MetadataPTR + DataString.Length + 1, Data.Length - MetadataPTR
- 1)
                    Data = Temp
                End If
            Next
        End If

        'SET METADATA frameRate
        'Sets the third and fourth byte of each Graphic Control
Extension (5 bytes from each label 0x0021F9)
        For x As Integer = 0 To Data.Count - 1
            If Data(x) = 0 Then
                If Data(x + 1) = &H21 Then
                    If Data(x + 2) = &HF9 Then
```

```
                        If Data(x + 3) = 4 Then
                            'word, little endian, the
hundredths of second to show this frame
                            Dim Bte() As Byte =
BitConverter.GetBytes(FrameRate \ 10)
                            Data(x + 5) = Bte(0)
                            Data(x + 6) = Bte(1)
                        End If
                    End If
                End If
            End If
        Next
        Stream.Write(Data, 0, Data.Length)
    End Sub
```

End Class



Public Class Form1

```
    Dim WithEvents T As New Timer() With {.Interval = 50}
    Public blackPen As New Pen(Color.Black, 3)
    Dim zweiPlusZwei As Boolean
    Dim startUndZiel As Integer
    Dim dieEins As Integer
    Dim minus As Integer
```

```
    Dim counter As Integer
    Public Encoder1 As New AnimatedGifEncoder
    Public Property MetadataComment As List(Of String) = New
List(Of String)


    Private Sub Form1_Load(sender As Object, e As
System.EventArgs) Handles Me.Load
        Me.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.None
        CenterToScreen()
        t.Start()
        DoubleBuffered = True
        zweiPlusZwei = True
        startUndZiel = 10
        dieEins = 1
```

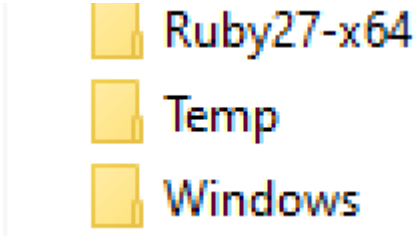
```
      minus = 200
   End Sub


   Private Sub Form1_Paint(sender As Object, e As
System.Windows.Forms.PaintEventArgs) Handles Me.Paint
      DrawRectangleRectangle(e)
   End Sub


   Private Sub Timer_Tick(sender As Object, e As EventArgs)
Handles T.Tick
      counter += 1


      If zweiPlusZwei = True Then
         If startUndZiel <= 128 Then
            startUndZiel += dieEins
```

```
      minus -= dieEins
   Else
      zweiPlusZwei = False
   End If
End If

If zweiPlusZwei = False Then
   If startUndZiel > 10 Then
      startUndZiel -= dieEins
      minus += dieEins
   Else
      zweiPlusZwei = True
   End If
End If
```

```
    If counter = 121 Then
       T.Stop()
    End If

    Refresh()
    CastleGrayscul(e)
End Sub


Public Sub SaveImage(filename As String)
    Dim FS As New IO.FileStream(filename, IO.FileMode.Create)
    Encoder1.FrameRate = t.Interval
    Encoder1.MetadataString = MetadataComment
```

```
      Encoder1.Repeat = True
      Encoder1.Save(FS)
      FS.Flush()
      FS.Close()
   End Sub

   Public Sub DrawRectangleRectangle(ByVal e As PaintEventArgs)
      e.Graphics.DrawRectangle(blackPen, 20, startUndZiel, 100,
minus)
   End Sub



   Sub CastleGrayscul(ByVal e As EventArgs) 'castleGrayscul
versus Monkeyisland. . .
      Dim bmp As New Bitmap(Width - 16, Height - 39)
```

```
        DrawToBitmap(bmp, New Rectangle(0, 0, bmp.Width,
bmp.Height))
        Encoder1.AddFrame(bmp)
        bmp.Dispose()
    End Sub


    Private Sub Form1_FormClosed(sender As Object, e As
FormClosedEventArgs) Handles MyBase.FormClosed
```




```
        SaveImage("C:\Temp\Probst.gif")
    End Sub
End Class
```