



Tablet PC applicaties ontwikkelen

DE UNIEKE MOGELIJKHEDEN VAN DE TABLET PC IN JE EIGEN APPLICATIES

“Leuk zo’n Tablet PC. Maar wat kan ik er als ontwikkelaar mee?” In dit artikel laat auteur Gerrit Jan Straatsma zien hoe eenvoudig de unieke mogelijkheden van de Tablet PC, zoals peninvoer en handschriftherkenning, zijn toe te voegen aan je eigen applicaties. Hij besteedt veel aandacht aan de platform binary executables die als basis dienen voor de ontwikkeling van applicaties die van de digitale inktmogelijkheden van de Tablet PC gebruik maken.

Laat ik voor de duidelijkheid beginnen met uit te leggen wanneer een PC een Tablet PC is. Een Tablet PC is vergelijkbaar met een normaal notebook waarbij een Tablet PC bovendien de mogelijkheid biedt de PC te bedienen met een speciale pen (een stylus) en het scherm. Naast alle standaardmogelijkheden die Microsoft Windows XP Pro biedt, heeft een Tablet PC ook de mogelijkheid voor het gebruik van digitale inkt die door handschriftherkenning kan worden omgezet naar tekst.

Ter ondersteuning van de Tablet PC heeft Microsoft een speciale versie van Windows XP ontwikkeld met specifieke op de Tablet PC toegesneden functionaliteit. Naast alle functionaliteit van de Windows XP Pro editie bevat de Windows XP Tablet PC Edition de volgende set aan softwarefeatures:

- Tablet PC specifieke applicaties: Windows Journal voor het maken van handgeschreven notities, Sticky Notes voor het maken van handgeschreven geeltjes en het spel InkBall voor de nodige ontspanning.
- Het Tablet Input Panel biedt tekstinvoermogelijkheden (ook voor bestaande applicaties) wanneer het toetsenbord niet wordt gebruikt.
- Platform binary executables dienen als basis voor het ontwikkelen van appli-

caties die gebruik maken van de digitale inktmogelijkheden van de Tablet PC.

Inkt in plaats van tekst

In tegenstelling tot de voorlopers van de Tablet PC behandelt de Tablet PC inkt ook als inkt en niet als tussenstap naar tekst. Ingevoerde notities, opmerkingen en tekeningen blijven bewaard op de manier waarop deze zijn ingevoerd. Hierdoor ontstaat een natuurlijk gebruik van de Tablet PC dat vergelijkbaar is met een schrijfblok. Op het moment dat geschreven tekst ook als tekst moet worden behandeld, kan deze door middel van handschriftherkenning worden geconverteerd. Dit gebruik van digitale inkt is mogelijk doordat de inkt gezien wordt als een nieuw datatype.

Het gebruik van inkt in een Tablet PC applicatie kent een drietal verschijningsvormen:

- *Free-Form Ink*. Deze vorm komt het meest overeen met het gebruik van gewoon papier. De gebruiker is volledig vrij in het invoeren van inkt in het document. Het maakt niet uit of het tekeningen, tekst of aantekeningen zijn.
- *Annotation Ink*. Hierbij kan de gebruiker aantekeningen maken in bestaande documenten. Vergelijkbaar met het maken van aantekeningen in een dictaat.

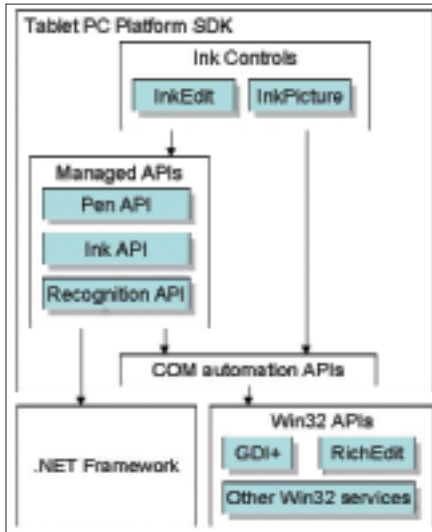
- *Inline Ink*. De laatste mogelijkheid voor het gebruik van inkt in een applicatie is door het te behandelen als andere tekstele elementen. Door de inkt tussen te voegen, of in de normale flow van een document te plaatsen, ontstaat normale tekstinvoer. Deze mogelijkheid kan bijvoorbeeld gebruikt worden voor het invoeren van gegevens op een invulformulier.

Tablet PC SDK

Met de Tablet PC SDK kunnen we de digitale inktmogelijkheden van de Tablet PC toevoegen aan onze eigen applicaties. Voor het gebruik van deze SDK is geen Tablet PC noodzakelijk. Alle inktinteracties zijn ook mogelijk op andere Windows-systemen. Voor het gebruik van de handschriftherkenning is echter wel de Windows XP Tablet Edition vereist.

Afbeelding 1 geeft schematisch de architectuur van de Platform SDK weer. Naast de Managed Application Programming Interface (API) biedt de SDK, Ink Controls, helpfiles, voorbeeldcode en ook een COM API. De managed API’s zijn onderverdeeld in drie onderdelen die in het vervolg van dit artikel verder zullen worden besproken:

- *Tablet Input API (Pen API)*. Deze API



Afbeelding 1. Architectuurschema Platform SDK

verzorgt de penspecifieke mogelijkheden, zoals de knoppen op de pen, het verzamelen van digitale inkt en *gestures* door de bewegingen van de pen.

- *Ink Data Management API (Ink API)*. Wanneer de inkt verzameld is via de Pen API kan deze gemanipuleerd en bewaard worden met de Ink API.
- *Ink Recognition API (Recognition API)*. Gespecialiseerde API voor het intelligent

interpreteren van inkt door het groeperen en herkennen van geschreven inkt.

De Pen API

Het InkCollector-object dient als basis voor de Pen API en ondersteunt in het verzamelen en weergeven van digitale inkt wanneer deze wordt ingevoerd door de gebruiker. De InkCollector gebruikt een Windows 'Form-based' window als inktcanvas. Dit is een rechthoekige *region* waarbinnen de inktinvoer verzameld wordt. Dit window wordt ook wel aangeduid als de InkCollector's host window. Het InkOverlay-object en voegt extra functionaliteit toe voor het selecteren en wissen van de verzamelde inkt. Voor de meeste applicaties zal het InkOverlay-object de beste keuze zijn. Codevoorbeeld 1 laat zien hoe eenvoudig een Windows Form geschikt gemaakt kan worden voor pen-input. Door de *handle* van het window te koppelen aan een instantie van een InkCollector accepteert het window de pen-input. Vergeet niet om een referentie te leggen naar de 'Microsoft Tablet PC API' (Microsoft.Ink.dll). Met de

DrawingAttributes-class is het mogelijk de weergave van de digitale inkt aan te passen. Hiermee is het onder andere mogelijk om kleur, hoogte, breedte en transparantie van de inkt te bepalen. Door de DefaultDrawingAttributes van de InkCollector of InkOverlay aan te passen, worden alle nieuwe pennenstreken weergegeven op basis van deze instellingen. Codevoorbeeld 2 laat zien hoe de kleur van de inkt kan worden aangepast. Naast de twee genoemde objecten bestaat de Pen API ook uit de volgende hardwaregerelateerde objecten: Tablet, Tablets, Cursor, CursorButton en Cursors waarmee de eigenschappen van de verschillende hardwarecomponenten te benaderen zijn.

De Ink API

De Ink API biedt toegang tot de meeste inktgerelateerde functionaliteit van de Tablet PC. Nadat de ingevoerde inkt door InkCollector of InkOverlay verzameld is, kunnen we de inkt verder bewerken. Afbeelding 2 geeft schematisch de relatie weer tussen de verschillende onderdelen van de Ink API. Zoals in de vorige paragraaf beschreven verzorgt een InkOverlay de invoer van de inkt. De digitale inkt in het InkOverlay is door middel van het Ink-object te benaderen. In een Ink-object is elke *pennenhaal* opgeslagen in een Stroke-object. Een Stroke-object bevat een serie van x/y-coördinaten die overeenkomt met de inputgegevens van de pen. De Strokes-class is een verzameling van pennenhalen. Voor de duidelijkheid spreken we in het vervolg over een Strokes Collection wanneer het een verzameling van pennenhalen betreft. Als laatste zien we de mogelijkheid om de geselecteerde Stroke-objecten van een InkOverlay op te vragen. Het resultaat van deze functie is een Strokes Collection. De Stroke-objecten in de selectie zijn dus een subset van de Strokes Collection van het Ink-object. Nu we weten hoe we de digitale inkt kunnen benaderen, kunnen we aan de slag met de ingevoerde inkt. Doordat de inkt niet geconverteerd wordt naar een ander dataformaat blijven alle eigenschappen bewaard. Het is bijvoorbeeld mogelijk om achteraf de inkt te schalen, van kleur te laten veranderen en de dikte te variëren. Door bijvoorbeeld stukken inkt

```

'Importeer de juiste namespace
Imports Microsoft.Ink

Public Class Form1
    Inherits System.Windows.Forms.Form
    '...
    'InkOverlay voor het verzamelen van inkt
    Dim myInkOverlay As InkOverlay
    '...
    Private Sub Form1_Load(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles MyBase.Load

        'Koppel de handle van panell aan het InkOverlay
        myInkOverlay = New InkOverlay(Panell.Handle)

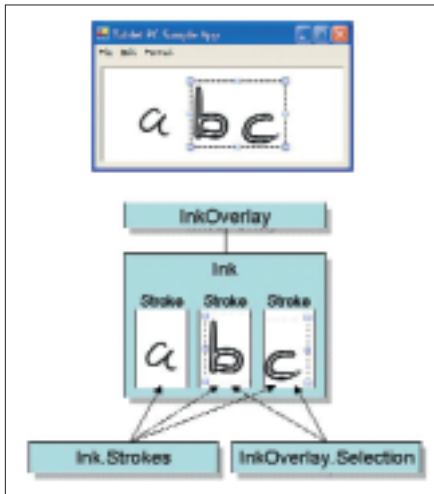
        'Enable de inkt invoer
        myInkOverlay.Enabled = True
    End Sub
    '...
End Class
  
```

Codevoorbeeld 1. Windows Form geschikt gemaakt voor pen-input

```

Dim clrDlg As New ColorDialog()
If clrDlg.ShowDialog = DialogResult.OK Then
    myInkOverlay.DefaultDrawingAttributes.Color = clrDlg.Color
End If
  
```

Codevoorbeeld 2. Aanpassen kleur inkt



Afbeelding 2. Schematische relatie tussen de verschillende onderdelen van de Ink API

te selecteren kunnen we een gedeelte van de inkt van kleur laten veranderen; zie voorbeeldcode 3. Door het wijzigen van de `DrawingAttributes` van de geselecteerde `Stroke`-objecten veranderen deze in de geselecteerde kleur.

Verwijderen van inkt

De `InkOverlay` biedt standaard de mogelijkheid om inkt te selecteren en te verwijderen. Door het wijzigen van de `EditMode` van de `InkOverlay` kan inkt worden verwijderd met de pen (`myInkOverlay.EditingMode = InkOverlayEditMode.Delete`). Er zijn twee manieren om inkt op deze manier te verwijderen: `PointErase` en `StrokeErase`. Bij de eerste gedraagt de pen zich als een gummetje en wordt de inkt onder de pen verwijderd. Bij de laatste wordt de hele pennenstreek verwijderd wanneer de pen over de pennenstreek beweegt. Inkt kan ook programmatisch verwijderd worden. Het `Ink`-object in de `InkOverlay` biedt de mogelijkheid een gegeven `Strokes`-collectie te verwijderen. Door hier de geselecteerde pennenstreken voor te gebruiken worden de pennenstreken verwijderd (`myInkOverlay.Ink.DeleteStrokes(myInkOverlay.Selection)`).

Één van de krachtigste mogelijkheden is het 'gummen' met de achterkant van de stylus. Dit wordt overigens niet ondersteund door alle typen pennen. Codevoorbeeld 4 laat zien hoe dit gebruik van de stylus wordt geïmplementeerd.

Inktopslag

Natuurlijk willen we niet dat de ingevoer-

```
If myInkOverlay.Selection.Count > 0 Then
    myInkOverlay.Selection.ModifyDrawingAttributes _
        (myInkOverlay.DefaultDrawingAttributes)
End If
```

Codevoorbeeld 3. Inktkleur van selectie veranderen

```
Private previousMode As InkOverlayEditMode

'Afhandeling CursorInRange event van het InkOverlay
Private Sub myInkOverlay_CursorInRange(ByVal sender As
System.Object, _
    ByVal e As InkCollectorCursorInRangeEventArgs)

    If e.Cursor.Inverted Then
        'Erase tip/achterkant stylus. Schakel over op erase mode
        If Not myInkOverlay.EditingMode = InkOverlayEditMode.Delete Then
            'Bewaars huidige editing mode
            previousMode = myInkOverlay.EditingMode
            'Schakel over naar erase mode
            myInkOverlay.EditingMode = InkOverlayEditMode.Delete
        End If
    Else
        'Standaard invoer. Schakel indien nodig over naar vorige mode
        If myInkOverlay.EditingMode = InkOverlayEditMode.Delete & _
            Not previousMode = InkOverlayEditMode.Delete Then
            myInkOverlay.EditingMode = previousMode
        End If
    End If
End Sub
```

Codevoorbeeld 4. gebruik stylus implementeren

```
Dim myStream As Stream
Dim isf As Byte()
Dim saveDialog As New SaveFileDialog()

saveDialog.Filter = "Ink Data (*.isf)|*.isf"

If saveDialog.ShowDialog = DialogResult.OK Then
    'Creëer een stream door het openen van de file
    myStream = saveDialog.OpenFile()
    If Not myStream Is Nothing Then
        'Serialiseer de inkt input
        isf = myInkOverlay.Ink.Save( _
            PersistenceFormat.InkSerializedFormat)
        'Schrijf de byte array naar disk
        myStream.Write(isf, 0, isf.Length)
        'Sluit de stream
        myStream.Close()
    End If
End If
```

Codevoorbeeld 5. Opslag inktgegevens

de inkt verdwijnt zodra we een applicatie afsluiten. De Ink API biedt ook hier uitkomst. Door middel van de `Save-` en

`Load-`functies van het `Ink`-object worden de inktdata geserialiseerd en gedeserialiseerd. De feitelijke opslag van de inkt-

```

Dim recos As New Recognizers()
Dim reco As Recognizer
Dim recoContext As RecognizerContext
Dim status As RecognitionStatus
Dim result As RecognitionResult

If recos.Count > 0 Then
    'gebruik de default recognizer
    reco = recos.GetDefaultRecognizer()
    'creëer een recognizer context
    recoContext = reco.CreateRecognizerContext()

    'handschriftherkenning op selectie of op alle inkt
    If myInkOverlay.Selection.Count > 0 Then
        recoContext.Strokes = myInkOverlay.Selection
    Else
        recoContext.Strokes = myInkOverlay.Ink.Strokes
    End If

    'stop de invoer van inkt
    recoContext.EndInkInput()
    'voer de handschriftherkenning uit
    result = recoContext.Recognize(status)

    If status = RecognitionStatus.NoError Then
        'geef het resultaat weer
        MessageBox.Show(result.ToString())
    End If
Else
    MessageBox.Show("Er zijn geen recognizers beschikbaar")
End If

```

Codevoorbeeld 6. Recognizer-class gebruiken

gegevens komt dus overeen met het opslaan van een byte array; zie de codevoorbeeld 5.

De Recognition API

Het laatste onderdeel van de API is de Ink Recognition API waarmee de handschriftherkenning van de Tablet PC beschikbaar is voor zelf ontwikkelde applicaties. Ik zal hier niet ingaan op alle mogelijkheden van de handschriftherkenning, maar de belangrijkste mogelijkheden zullen wel aan bod komen.

De eenvoudigste oplossing is het gebruik van de standaard ToString-functie. Bij een Strokes-collection wordt handschriftherkenning toegepast en is het resultaat de tekstinterpretatie van de inkt. Deze methode is echter zeer beperkt toepasbaar. Doordat de inkt pas geïnterpreteerd wordt wanneer de ToString-functie wordt aangeroepen, zal het relatief lang duren

voor grote hoeveelheden inkt zijn herkend. Daarnaast geeft de ToString-functie niet aan hoe betrouwbaar de herkenning is, wordt de standaardtaal gebruikt, is er geen overzicht van mogelijke alternatieven en is de handschriftherkenning niet te beïnvloeden.

Om meer invloed te hebben op de handschriftherkenning kunnen we gebruik maken van de Recognizer-class. Deze class is de verpersoonlijking van een op het systeem geïnstalleerde *=recognizer=*. De attributen van de class geven onder andere aan welke talen worden ondersteund en welke mogelijkheden de recognizer biedt. De Recognizers-collection is een verzameling van Recognizer-objecten, en de inhoud komt overeen met alle op het systeem geïnstalleerde recognizers. Het gebruik van de Recognizer-class is vrij eenvoudig en bestaat uit vier stappen: (1) referentie opzetten naar de te

gebruiken recognizer, (2) een recognition-session starten, (3) een Strokes-collection definiëren en (4) het resultaat opvragen; zie voorbeeldcode 6. Deze methode maakt echter, net als de ToString functie, nog steeds gebruik van synchrone handschriftherkenning. Door de recognizer echter gedurende het schrijven te voorzien van de nieuwe penstreken, kan de uiteindelijke tijd voor de handschriftherkenning na het schrijven verkort worden.

De effectiefste manier van handschriftherkenning is asynchrone handschriftherkenning. Hierbij wordt tijdens het schrijven - in een andere thread - de inkt direct geconverteerd naar tekst. Het resultaat van de recognizer bij handschriftherkenning bestaat uit meer dan alleen een tekststring. Naast deze tekst geeft het resultaat ook aan hoe zeker de recognizer is van de handschriftherkenning via het 'confidence level' en zijn mogelijke alternatieven beschikbaar.

Ink Controls

De Tablet PC Platform SDK bevat tevens een tweetal Windows-controls waarmee zeer eenvoudig pen- en inktfunctionaliteit kan worden toegevoegd aan nieuwe en bestaande Windows-applicaties. Dit alles door de controls met een drag-drop vanuit de toolbox op een Form te plaatsen. Het InkEdit-control is een uitbreiding op de RichEdit-control. Het verzamelt de ingevoerde inkt en converteert deze naar tekst. Dit control is met name toepasbaar wanneer rechtstreekse conversie van inkt naar tekst gewenst is. Het InkPicture-control is bedoeld voor het verzamelen en weergeven van inkt. Één van de belangrijkste mogelijkheden is het weergeven van een afbeelding achter de inkt. Uit dit artikel blijkt dat de unieke mogelijkheden van de Tablet PC eenvoudig zijn toe te voegen aan je eigen applicaties en dat je slechts beperkt wordt door je eigen fantasie!

Nuttige internetadressen

- <http://www.microsoft.com/windowsxp/tabletpc>
- <http://www.microsoft.com/windowsxp/expertzone/focuson/tabletpc.asp>
- <http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28001295>
- <http://www.tabletdeveloper.com>