Multimodal Interfaces with Microsoft.Ink

Joan Pastor Pellicer
María José Castro

jpastor@dsic.upv.es, mcastro@dsic.upv.es

Universitat Politècnica de València
Departament de Sistemes Informàtics i Computació

Table of Contents

Online Handwriting Recognition Systems

- We know (or just have a little idea) how a handwritten recognition system works.

- Online Handwriting Recognition Systems:
  - we know the strokes written by the user
  - we know the direction of the stroke

- Given a sequence of points (strokes) try to recognize the written text.

Online Handwriting Recognition Systems

How can I use it in my application?

What can I do if I want to add a new input source to my application?

Multimodal Systems

Multimodal interaction provides the user with multiple modes of interfacing with a system.

Example:

- When you call to any call center: use numbers or voice.
- Smartphones (touch)
- Kinnect
- Siri, Google Now
- Disabled people

Microsoft Handwriting Recognition with Digital Ink

- Microsoft provides a set of handwriting recognition tools in different versions of windows.
- Provides components and controls for .NET applications.
- Tablet PC: Digital Ink

Microsoft Windows XP Tablet PC Edition

- Add pen based capabilities
- Add "digital ink" to a full range of Windows applications.
- The digitized handwriting can be converted to standard text through handwriting recognition, or it can remain as handwritten text.

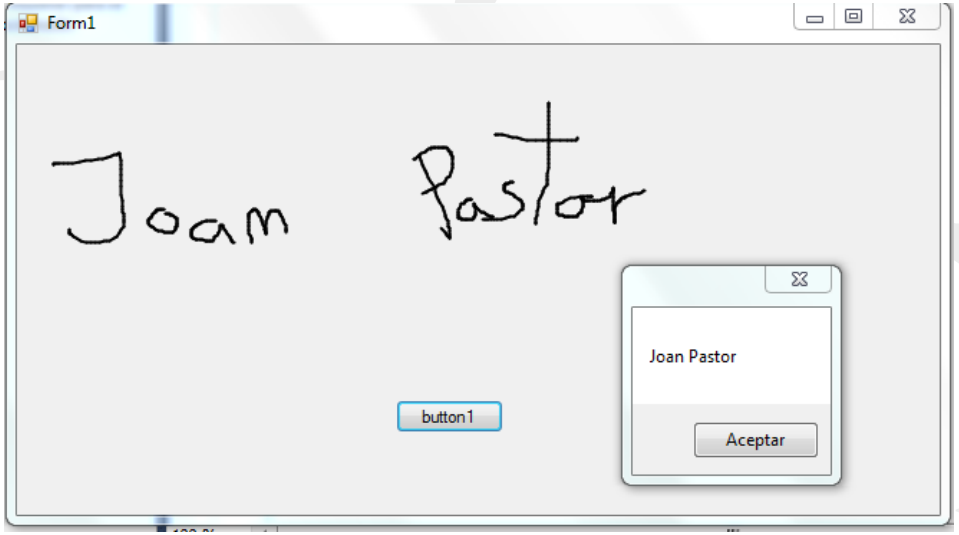
Microsoft.Ink Library

Class Name	Description
Divider	Analyzes ink to distinguish text from pictures.
DrawingAttributes	Controls appearance of ink, such as color, line width, and so on.
Gesture	Ink interpreted as a command.
Ink	The main container to hold ink. Holds the collection of strokes that make up ink input. Supports moving ink between memory and disk. Supports clipboard actions on ink.
InkCollector	Snap-on support for creating a basic ink-enabled window.
InkOverlay	Snap-on support for adding ink-enabled support to existing application windows.
PenInputPanel	An in-place input window for adding in-place ink input for existing controls.
Recognizer	Provides language-specific conversion of ink strokes to text and the system dictionary with common words in a given language.
RecognizerContext	Organizes the input elements needed for ink-to-text conversion: a recognizer, hints about data type (factoid), and an application-specific dictionary (word list).
Renderer	Draws ink onto a drawing surface.
Stroke	A set of (x,y) point values generated by a pen. A stroke starts when a pen makes contact with the drawing surface, and includes the locations traversed by the pen until the pen is lifted from the drawing surface.
Strokes collection	Holds a set of strokes.

InkOverlay (Windows Form)

Can be attached to a application window (or control) and be used as ink input control.

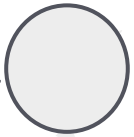
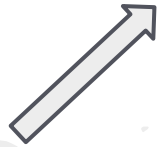
(http://pastebin.com/AzsYK666)



Canvas (WPF)

Gestures

A gesture is a special movement with the pen:

- Draw some form 
- A direction movement:  
- A tap (like a mouse click or double)

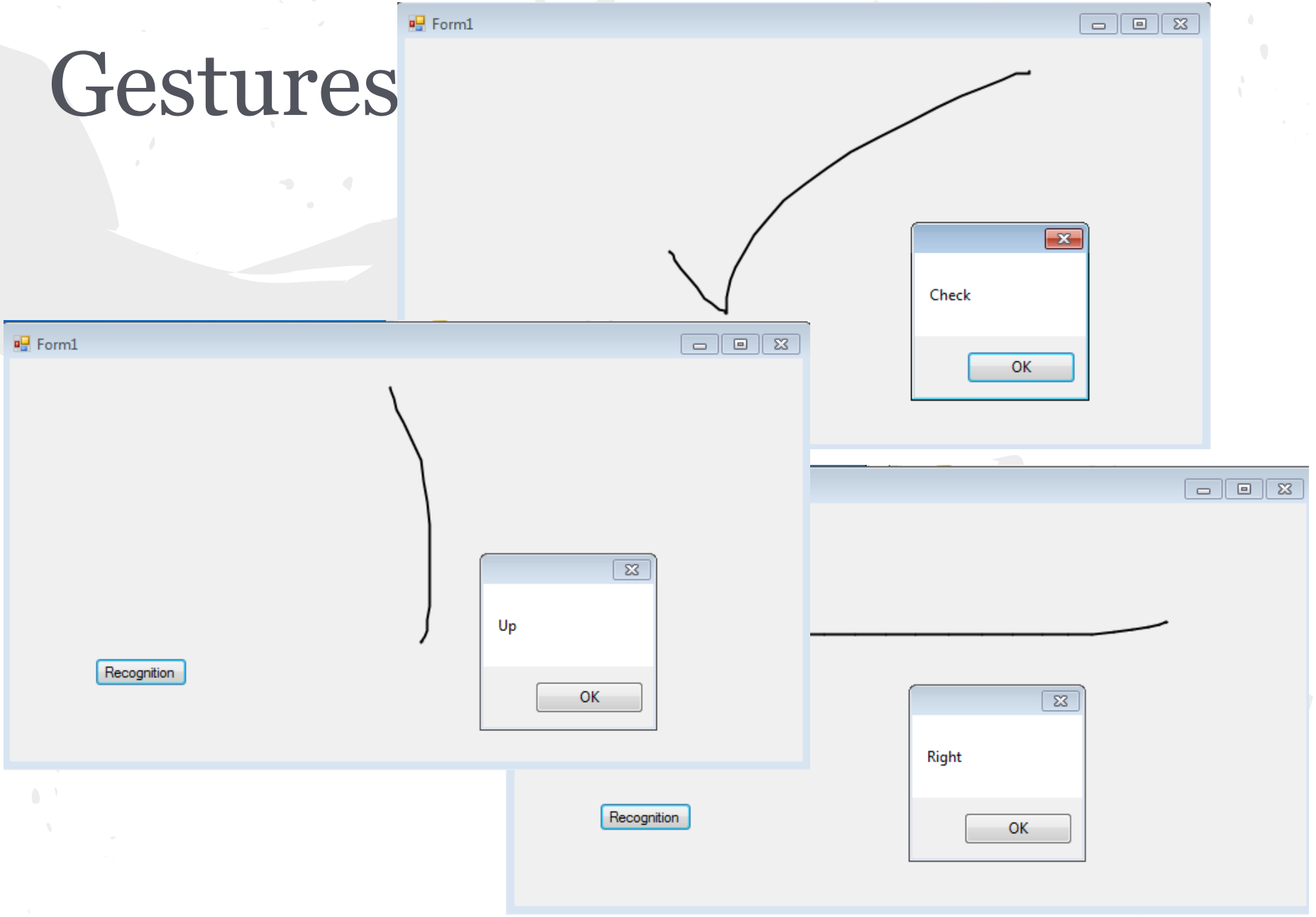
Gesture

Gesture Event:

- Add a InkCollectorGestureEventHandler for the Event Gesture.
- Activate Ink and Gesture Mode.
- Set the status of the recognize Gestures.

(http://pastebin.com/c12wbhe0)

Gestures


Form1
Check
OK
Form1
Recognition
Up
OK
Recognition
Right
OK

Stroke and Strokes

- Stroke is a temporal sequence of points.
- Numbers and letters written by more than one stroke: "f" or "t"
- In Handwriting, a word can be only one stroke
- General drawing or gestures

Strokes and Stroke

- Strokes is a collection of strokes.
- Individual strokes can be accessed by index.

- Stroke is an array of points.
- Individual points can be accessed by index.

- You can modify (add, delete, update) by code these attributes. (http://pastebin.com/0MDE2hJf)
- You can check the intersections between strokes.

Recognizing Strokes

- Convert these strokes into text requires a Recognizer.
  Language and locale properties of the recognizer.

  Recognizer object with a set of installed recognizers.
  It is needed to select a recognizer (usually the first) and create a context in order to start the recognition.

Creating Strokes from the scratch

- Convert a set of coordinates in Strokes.
- Set of points detected by Kinnect.

(http://pastebin.com/makQDFtd)

Recognizing Strokes

- Factoid: To give an intuition (or a hint) about the text to recognize.
  - DIGIT
  - HIRAGANA/KATAKANA
  - ONECHAR
  - TIME
  - ...

- WordList:
  - You can add your own list of words

Recognizer and hypothesis

- Use the RecognizerContext to perform the Strokes recognition.
- The result is a list of n-best hypothesis with their confidences.

(http://pastebin.com/SB13uU5h)

Events and Recognition instance

We can Recognize at any time (invoking recognize method):

- Pushing a button (click event)
- Losing focus (leave out the control)
- Pen up Event
  - Problem: More than one stroke
  - Solution: Use a timer

(http://pastebin.com/wtjTAqbr)

InkEdit Control

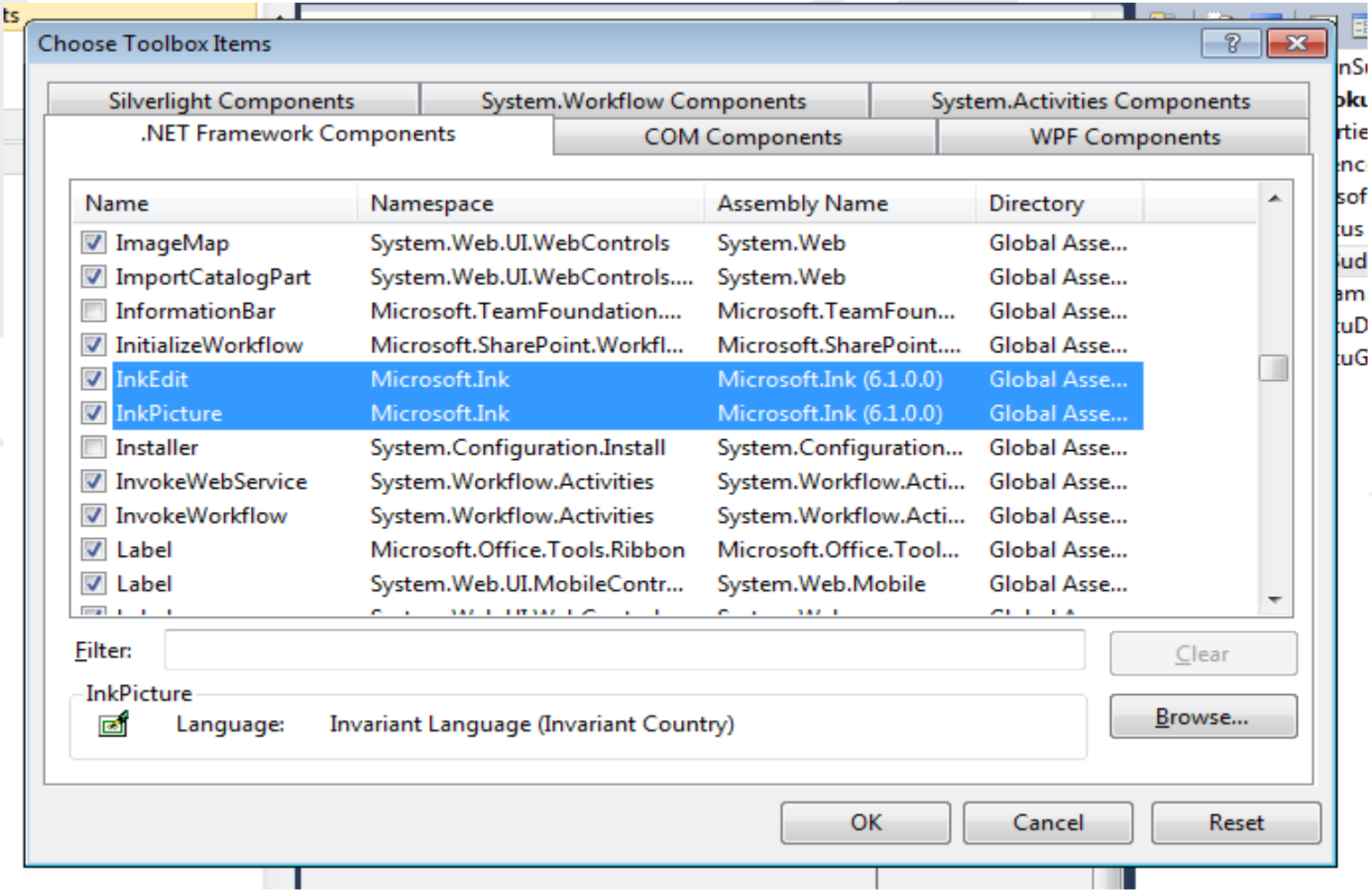
- Control derived from RichTextBox

1. Captures Ink
2. Removes Ink
3. Displays the recognized text within the textbox.

Adding InkEdit to Visual Studio

1. Right-click on Toolbox
2. Select **Customize Toolbox**
3. Select the .NET Framework Components tab
4. Check **InkEdit** and click **OK**

Adding InkEdit to Visual Studio



Controlling Recognition

RecoTimeOut: 2000 ms by default

- Events:
  - Gesture
  - Recognition

(http://pastebin.com/L1TLsqb8)

Thank you

Grazie!