Building Tablet PC Applications

Ananda Gunawardena
School of Computer Science
Carnegie Mellon University
Ubiquitous Computing

ubiquitous : a vision of people and environments augmented with computational resources that provide information and services when and where desired [Mark Weiser 1991]

• Ubiquitous computing inspires application development that is “off the desktop.”
• Humans speak, gesture, and use writing utensils to communicate with other humans. They can do the same with machines.
• Enable it by supplementing or replacing elements of the GUI interaction paradigm with computer interfaces that support more natural human forms of communication (e.g., handwriting, speech, touch and gestures)
Natural Interaction

• Humans express their ideas and interact through sketches
• Sketches are ubiquitous.
• Sketches are dead
• Sketching Enabling Hardware: PDAs, digital tablets, laptops, and wall-sized electronic whiteboards (smart boards)
Sketches are Ubiquitous

The height of the right
is 0, you can’t distribute.

The height of the right
is 1, you can distribute.
Sketches are Dead

HW2 Group Sketch (Adib, Ameer, Anirban)

Main

Add Node

Sub Node

Mult Node

Div Node

Float Node

Operator Node

Expression

Morph Exception

Who throws Morph Exception?
Natural Interaction

• Natural interaction is knowledge based
  – AI problem

• Natural interaction need to be enabled through novel interfaces
  – HCI problem

• We are working at the intersection
  – Pragma, Adaptive Book projects at CMU
  – Goal: Student behavior modeling
Free Form Interaction

• Pen-based or free-form interaction
  – Resurgence after the failure of the first generation of pen computing.

• Require Natural Data Types
  – audio, video, ink, and sensor input

• Much of the interest in pen-based computing has focused on recognition techniques to convert the “ink” from pen input to text
  – Least interesting (except in some domains)

• Manipulating the freeform ink data type within programs
  – How can we recognize ink sketches, interpret and manipulate them
Natural Interaction Requires Natural Interfaces
Typical setting for today’s interface

- Fixed stable environment, with a keyboard,
- Indirect interaction,
- High precision, stable pointing system
Typical Tablet PC use

- Portable, unstable environment, without a keyboard
- Direct interaction,
- Low precision aiming
Designing for a Great Mobile Experience
Things to Consider

• Be easy to use and fully functional in the absence of a keyboard and mouse
• Present information in a way that users can quickly digest it. Avoid congested screen designs
• Be able to launch and run in full screen mode
• Provide exit control when in full screen mode
• Be able to run in portrait and landscape modes
• Ensure that all user interface elements are visible and unclipped at 800x480
Things to Consider

• Ensure that dialog box height is less than 480 pixels
• Ensure that UI elements scale to accommodate range of display sizes and typical display resolutions
• Design the UI to support content consumption, not content creation
Optimizing Touch command and Control

• Make touch target at least 1 cm x 1 cm
• Space targets to sufficiently ensure an accurate interaction layer
• Ensure that graphics in the target area maintains legibility
• Ensure that targets are not obscured by hand during interaction
• Provide larger targets if your application is used in a highly mobile situation
• Group targets by functionality
• Avoid placing targets at the edges of the display
Designing for keyboard Free experience

• Provide on-screen alternatives to commonly used keyboard shortcuts such as copy, paste, undo, save and open
• Provide an alternative to ctrl-alt-del
• Where text is needed offer auto complete or quick text
• Explore ways to use alternate input modalities such as speech
Designing for mouse free scenario

• Touch screen devices typically do not provide hover-like features
  – Do not rely on hover like features for critical tasks
  – Consider providing hover feedback with object focus after an item is tapped
  – If pop-ups are provided for hover feedback, then close them immediately
Replacing Right Click and more

• Minimize reliance on right click functionality
• Provide alternative ways to access a functionality that is typically right click
• Use “tapping” as a way to open context menus
• Use alternative methods to scrolling
• Ensure that drag-and-drop features are clear, quick and simple and requires no fine motor skills
Enabling Natural Interaction - Research Challenges
Research Challenges

- **(HCI) Natural interfaces** that facilitate a richer variety of communications capabilities between humans and computation.
  - speech, pen and gesture input
- **(AI) Context-aware applications**
  - sensor fusion and activity recognition.
  - Identity, Position, shape, time, history etc..
  - 5 W’s (who, what, where, when and why)
    - Context Toolkit (Saber 1999) – simplifies the design
- **Capturing** live experiences and indexing
Defining new Data Types

• What kinds of operations should be supported for a natural data types such as ink? what about gesturing? audio?
  – Ink data type
    • merge independent strokes together as they form letters, words, and other segments of language.
    • Strokes forming objects that are recognizable in the context

• What can we learn from handwriting recognition?
We can Learn from Handwriting Recognition

- Handwriting recognition has come a long way
- Windows Journal has a handwriting recognition around 90%
- Focus on improving the recognition accuracy.
- But it is tough
- Even humans make mistakes when dealing with these same forms of communication.
- Even the most expert handwriting recognizers (humans) can have a recognition accuracy as low as 54% [Schomaker 1994].
- Human accuracy increases to 88% for cursive handwriting [Schomaker 1994] 96.8% for printed handwriting
Windows Hand Writing Recognition System

- Training with Neural Nets
- Using back-propagation training
- Microsoft collects millions of words of ink data from thousands of writers
  - Young and old, male and female, left handed and right handed
  - Natural text, newspaper text, URLs, email addresses, street addresses
- Microsoft collects in nearly two dozen languages around the world
- Training on such large databases takes weeks
- Microsoft Researchers study about how well the data reflect their customers
  - Their writing styles
  - Their text content
Languages

- Microsoft Support Handwriting Recognition in:
  - English (US), English (UK), French, German, Spanish, Italian
- They are working on:
  - Dutch, Portuguese, Swedish, Danish, Norwegian, Finnish
  - Haven’t released yet
- Microsoft is starting research in other languages
- Microsoft also ship now:
  - Japanese, Chinese (Simplified), Chinese (Traditional), Korean
Handwriting Recognizer Architecture

CMU-15-397
Fall 07
Research Challenges
Pen Based Computing

- Microsoft Focus Group
- Conferences
  - ACM Symposium on User Interface Software and Technology (UIST)
  - AAAI Workshop on Making Pen-Based Interaction Intelligent and Natural
  - the International Conference on Multimodal Interfaces (ICMI)

- Group identifies several research challenges
  - “Pen” interfaces and toward “Multimodal”
  - Taking Advantage of Multiple Input Modalities
  - Recognition/Analysis of Non-textual Information
  - Interfaces that Adapt
  - Design Methods for Intelligent Interfaces
  - New Interface Metaphors
  - Merging Ink and Existing Applications
Ink computing and Pedagogy

• **Tablet as a Vehicle for Teaching Methods**
  – *Live annotations*
  – *Real time annotated feedback*
  – *Linking informal annotations and formal resources*

• **Tablet as Student Multimedia Authoring Device**
  – Students expressing learning using multimodal inputs

• **Bridging the Formal and Informal learning**
  – *Students’ thought and learning proceeds from informal stages to more formal stages*
  – *Pen computing supports informal expression well through sketching augmented with more formal expressions*
Pen Based Computing

- Disciplines that can benefit
  - Music (Composition tools - Grove City College)
  - Chemistry (Molecular Structures - Mt Lebanon)
  - Design
  - Architecture
  - Any engineering discipline (Circuit Designers)
  - Computer Science
    - Graphics
    - Data Structures (Pragma Research)
  - Business (Process Flow)
  - Special Education (Disable people)
  - Math (Equation Manipulators)
  - Doctors/Nurses making rounds (Cancer Center)
Ink on Browsers

- Ink on browsers is now possible
- ASP Net 2.0 server controls, onClientClick simplifies the process of moving ink from webpage to webserver
  - Can tie a client side event to a server side event
- AJAX allows ink to remain in the page during a refresh event
- Steps of Building an ink enabled application on browser
  - Add Microsoft.Ink API into project
  - Add inkOverLay component to control
  - Add color buttons and other controls and enable them by writing a click method for each
  - Add public methods so that web pages client-side code can interact with the controls
    - Returns current Color Name, loading data into control, returns ink data as a base-64 encoded string
  - Create a webpage to host the ink controls
    - Enable partial postbacks with ASP.net AJAX updates
    - Send the ink control data to an image file
    - Retrieve ink data from an image file and load it into the ink control for additional drawings
  - Get the page to interact with the controls
    - Communicating with control to send/receive ink data
    - Java script can enable this
  - Get the ink data to the server side
    - Client side – get the ink data from control
    - Client Side – store it in a hidden control that is accessible from the server side
    - Server Side – Access the data in the hidden control
Silverlight

- **About Microsoft Silverlight**
  - cross-browser
  - cross-platform
  - plug-in for delivering the next generation of media experiences and rich interactive applications (RIAs)
  - Incorporate video, animation, interactivity, and stunning user interfaces.

- **Silverlight** (previously codenamed "WPF/E") is a lightweight subset of XAML for building rich media experiences on the web.
Silverlight

- Silverlight is the only Web technology to support collection and display of high-quality pen input from Tablet PC devices
Ink in Silverlight

- The ink in Silverlight is made up of a StrokeCollection object.
- StrokeCollection object is made up of individual Stroke objects.
- A Stroke can be a dot, a straight line, or a curving line.
- Each Stroke is made up of a StylusPointCollection object, which, in turn, is made up of individual StylusPoint objects.
- StylusPoint objects are collected when the pen moves while it is in contact with the digitizer. The ink also contains characteristics such as color, width, and outline color, all of which are contained in the DrawingAttributes class.
Ink Presenter

- The **InkPresenter** element is a Canvas that displays ink.
- **InkPresenter** is generally the same as the **Canvas** element, but also has a **StrokeCollection**
- You add **Stroke** objects to the **StrokeCollection**, the **InkPresenter** automatically renders them.
Collecting ink

• Collect ink by retrieving StylusPoint objects when the pen moves while in contact with the screen.

• Then add a Stroke object, which consists of the collected StylusPoint objects, to the StrokeCollection that is associated with the InkPresenter.

• Example