

What the Digerati Know

INFO/CSE 100, Autumn 2004
Fluency in Information Technology

<http://www.cs.washington.edu/100>

Readings and References

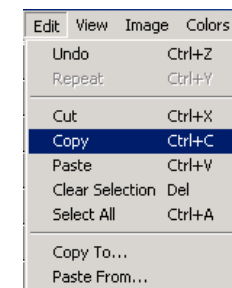
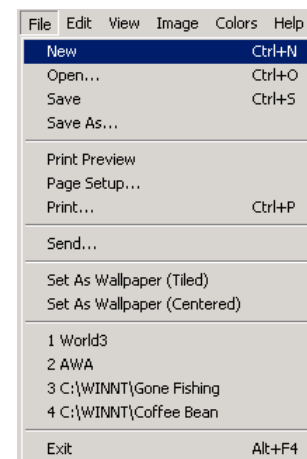
- Reading
 - » *Fluency with Information Technology*
 - Chapter 2, What the Digerati Know

Learning New Tools

- How do we learn to use new tools?
 - Be taught their use by someone else -- car, bicycle
 - Reading the owner's manual -- chain saw
 - Figure them out ourselves -- CD player
- Software designers wanting you to learn their tool ASAP, so they try for 'intuitive' usage
 - Consistent Interfaces -- build on experience
 - Suggestive icons -- bypass terminology
 - Metaphors -- exploit analogous reasoning

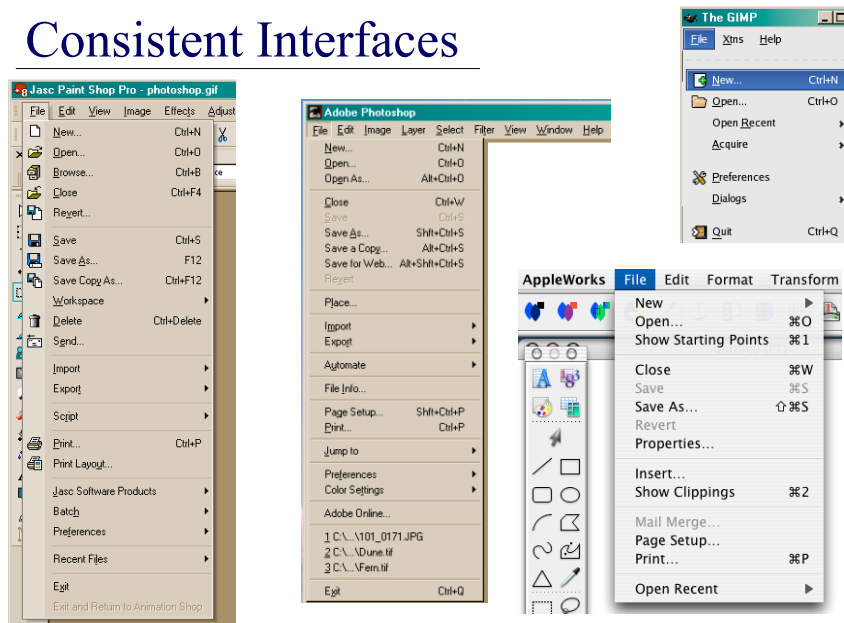
Consistent Interfaces

Most modern applications have File and Edit menus with standard commands



MS Paint

Consistent Interfaces



What does 'New' Mean?

- Most applications have the concept of a *document* with:
 - » Attributes: date created, date modified, creator, ...
 - » Content: image, text, sound, ...
- 'New' means create a 'blank instance' of a document for this application
 - » A document has attributes as well as content
 - all stored in one file with a place for everything
 - A 'blank instance' is simply the structure with some of the attributes filled in but without any of the content

4-Oct-2004

cse100-03-digerati © 2004 University of Washington

6

“Click Around”

- Software designers use standard ideas to make applications intuitive
- To learn a new application, check it out by clicking around
 - » Take a minute to ...
 - Look under all menus to see operations
 - Follow the “...” for menu operations
 - Try to recognize what the icons mean

Clicking around is exploration

4-Oct-2004

cse100-03-digerati © 2004 University of Washington

7

A New Application

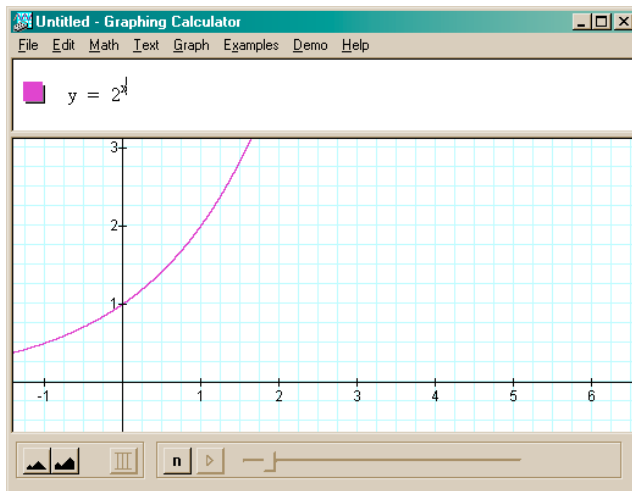


4-Oct-2004

cse100-03-digerati © 2004 University of Washington

8

A New Application



4-Oct-2004

cse100-03-digerati © 2004 University of Washington

9

“Blazing Away”

- Learn an application fast by trying it
- Beginning with a new instance, assertively try menu items
 - » *Expect to fail* and make a mess
 - » Exit the application, and if you are asked “Save?” reply “No”
 - » Try repeatedly until becoming familiar

If you are trying to achieve some goal, keep your eyes on the prize

4-Oct-2004

cse100-03-digerati © 2004 University of Washington

10

To Learn A New Tool

- Software systems build on a consistent interface, standard metaphors, etc.
 - » Expect to teach yourself applications
 - » Do so by familiarizing yourself with the features by “Clicking Around”
 - » Assertively try out the features, “Blaze Away,” watching what they do
 - Be efficient -- stay focused, don't type a lot when you expect to exit

You also might want to skim the the user manual!

4-Oct-2004

cse100-03-digerati © 2004 University of Washington

11

Differences & Similarities

- Different vendors will produce similar software for the same task
 - Superficially, the GUIs use similar features
 - Fundamentally, the task largely determines how the software must work ... they *must* be similar
- Implications ...
 - Know one word processor, learn others fast
 - Software differences: mostly glitz, convenience
 - Don't accept lousy tools ... switch to other applications

4-Oct-2004

cse100-03-digerati © 2004 University of Washington

12

Another Implication

If applications are similar at their core, techniques can be learned without learning a specific vendor's software



4-Oct-2004

cse100-03-digerati © 2004 University of Washington

13

Mac or PC???

- Arguments about which is better, Mac or PC, create only heat, no light
 - » They are more alike than different
 - » Any Fluent person can use both
- I first bought a PC, then a PowerMac, then a PC, then a Windows laptop, then an eMac, then a PC
 - » and I've got 4 Linux machines from Boeing surplus acting as file servers sometimes

4-Oct-2004

cse100-03-digerati © 2004 University of Washington

14

Summarizing

- Humans must learn to use tools
 - Software designers want you to learn easily
 - Good software uses consistent interface, metaphors, ...
- Teach yourself applications by “Clicking Around” and “Blaze Away”
 - » I don't know all the features, you don't know all the features, and that's okay!
- Software for a task must share core features
- Learn applications independent of vendor

4-Oct-2004

cse100-03-digerati © 2004 University of Washington

15