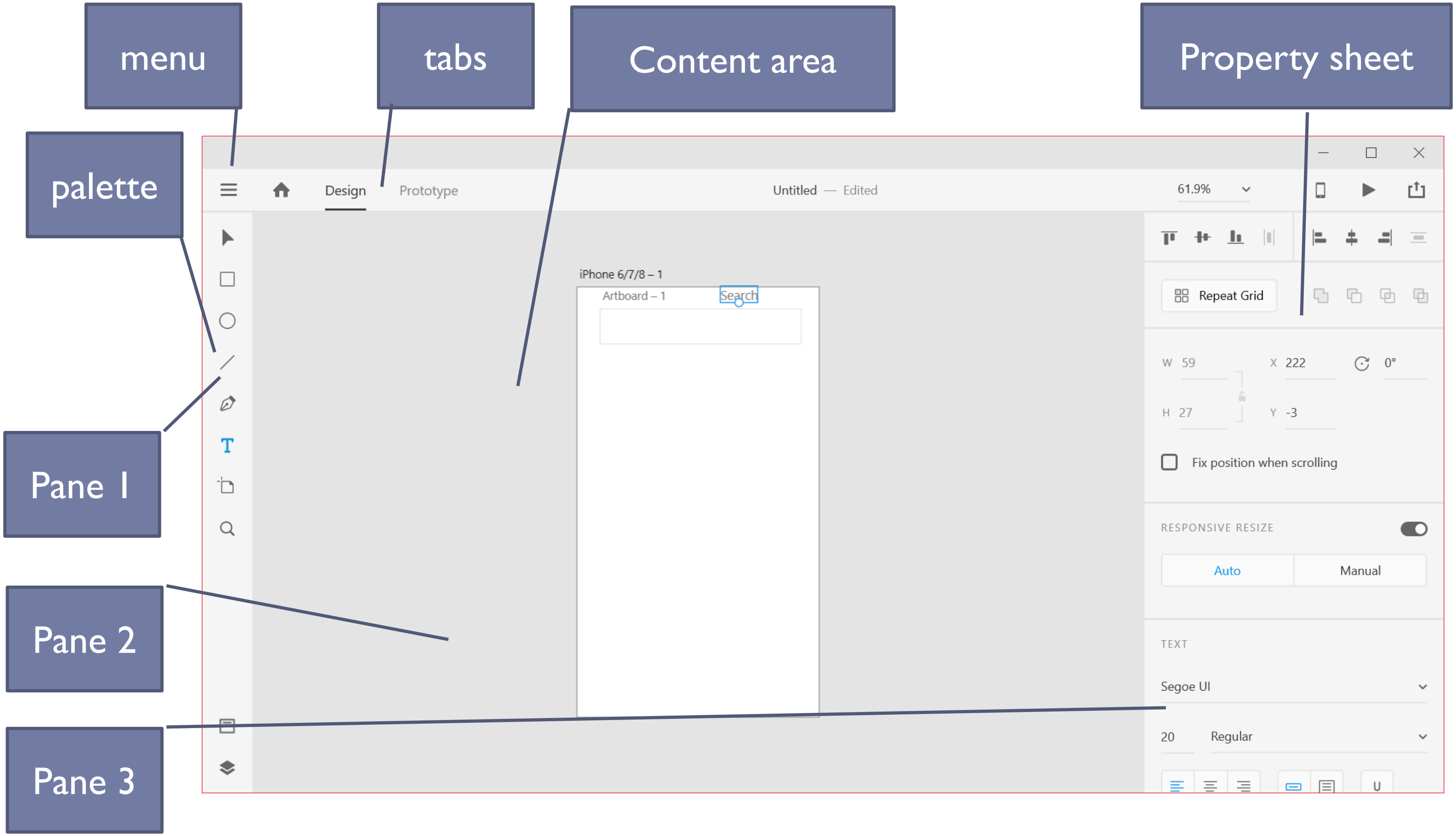
# Principles of GUI Design and Programming

Interaction Elements

# Contents

- Anatomy of a desktop application
- widgets
-

# Anatomy of a desktop application



menu

tabs

Content area

Property sheet

palette

Pane 1

Pane 2

Pane 3

W 59

X 222

0°

H 27

Y -3

□ Fix position when scrolling

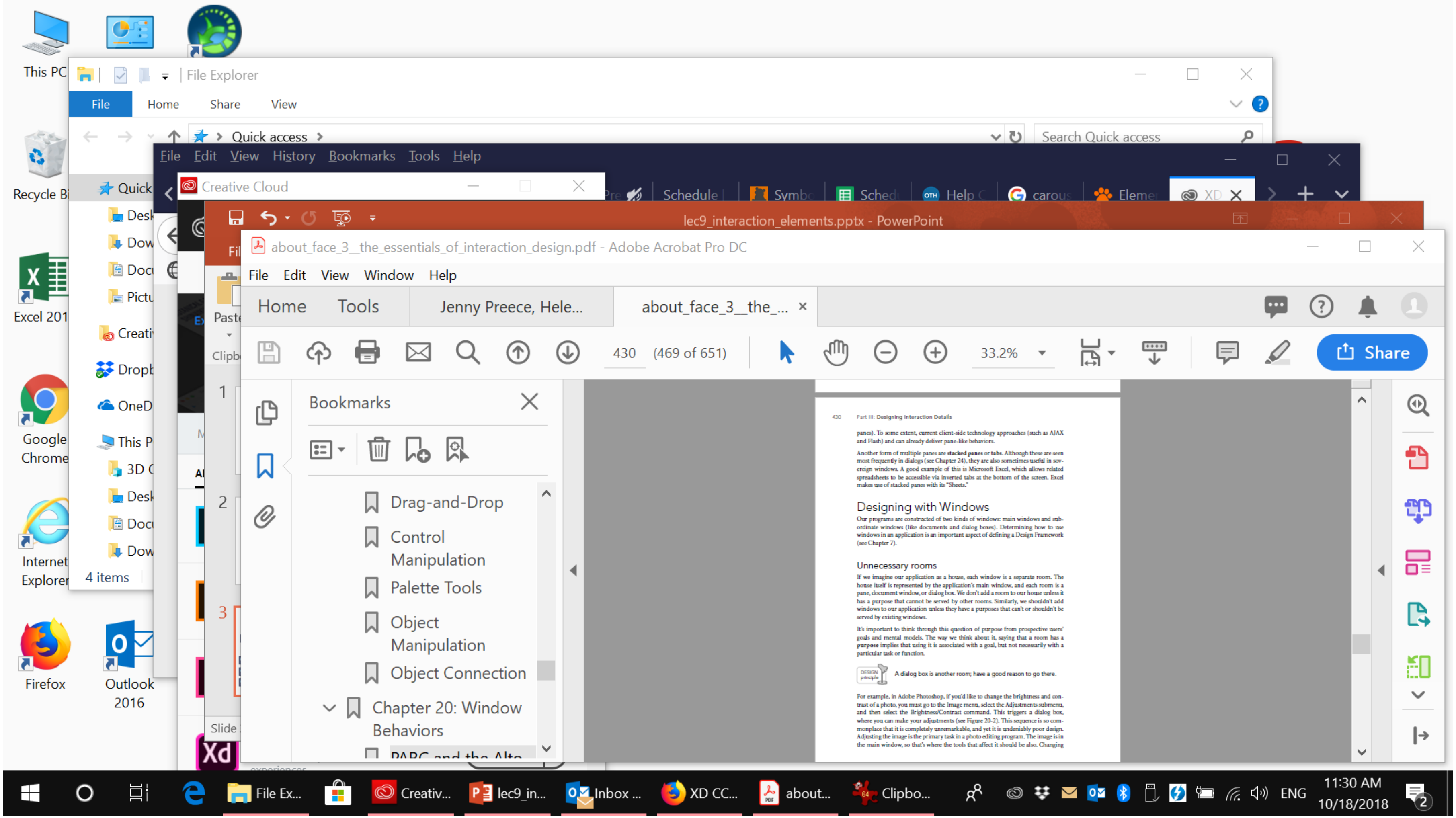
Segoe UI

20 Regular

# Window Managers

- The window manager
  - Lays out applications on the desktop
  - Determines the size of each application
  - Determines if applications overlap or not
  - Determines which window has the focus (follow or click)
- X windows has separate window managers which can be switched any time
- MS Windows and Mac have window managers as part of the windowing system with some configurability

# MS Windows – Cascading windows



430 Part III: Designing Interaction Details

panes). To some extent, current client-side technology approaches (such as AJAX and Flash) and can already deliver pane-like behaviors.

Another form of multiple panes are **stacked panes** or **tabs**. Although these are seen most frequently in dialogs (see Chapter 24), they are also sometimes useful in sovereign windows. A good example of this is Microsoft Excel, which allows related spreadsheets to be accessible via inverted tabs at the bottom of the screen. Excel makes use of stacked panes with its "Sheets."

Designing with Windows

Our programs are constructed of two kinds of windows: main windows and subordinate windows (like documents and dialog boxes). Determining how to use windows in an application is an important aspect of defining a Design Framework (see Chapter 7).

Unnecessary rooms

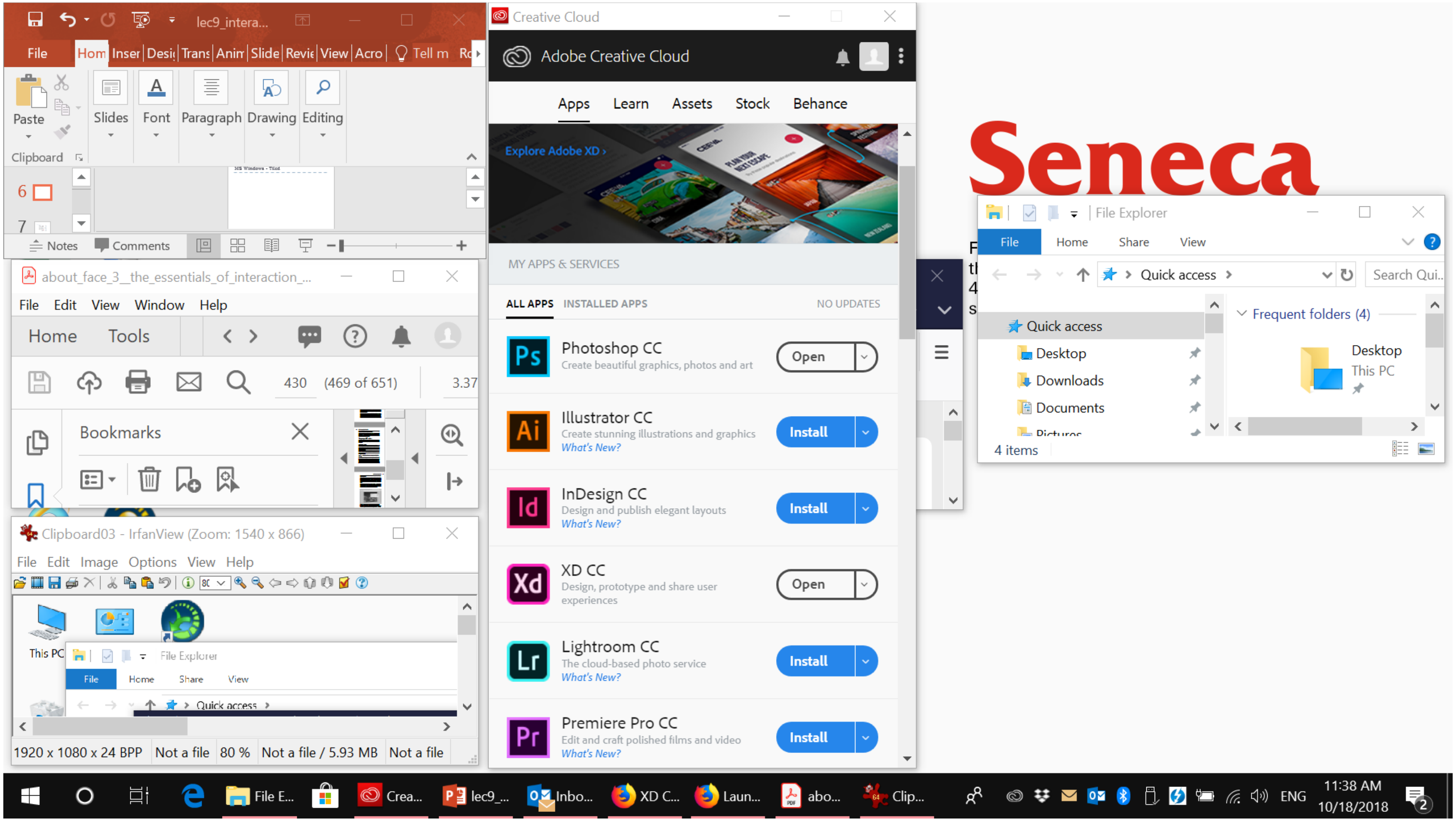
If we imagine our application as a house, each window is a separate room. The house itself is represented by the application's main window, and each room is a pane, document window, or dialog box. We don't add a room to our house unless it has a purpose that cannot be served by other rooms. Similarly, we shouldn't add windows to our application unless they have a purposes that can't or shouldn't be served by existing windows.

It's important to think through this question of purpose from prospective users' goals and mental models. The way we think about it, saying that a room has a **purpose** implies that using it is associated with a goal, but not necessarily with a particular task or function.

DESIGN principle A dialog box is another room; have a good reason to go there.

For example, in Adobe Photoshop, if you'd like to change the brightness and contrast of a photo, you must go to the Image menu, select the Adjustments submenu, and then select the Brightness/Contrast command. This triggers a dialog box, where you can make your adjustments (see Figure 20-2). This sequence is so commonplace that it is completely unremarkable, and yet it is undeniably poor design. Adjusting the image is the primary task in a photo editing program. The image is in the main window, so that's where the tools that affect it should be also. Changing

# MS Windows - Tiled



Photoshop CC — Create beautiful graphics, photos and art — Open

Illustrator CC — Create stunning illustrations and graphics — What's New? — Install

InDesign CC — Design and publish elegant layouts — What's New? — Install

XD CC — Design, prototype and share user experiences — Open

Lightroom CC — The cloud-based photo service — What's New? — Install

Premiere Pro CC — Edit and craft polished films and video — What's New? — Install

# Window states

- Fullscreen
  - Occupies the full screen
- Minimized
  - In the task bar
- Resumed
  - Shown at some size but allowing other windows to be shown on the screen at the same time

# Full-screen applications

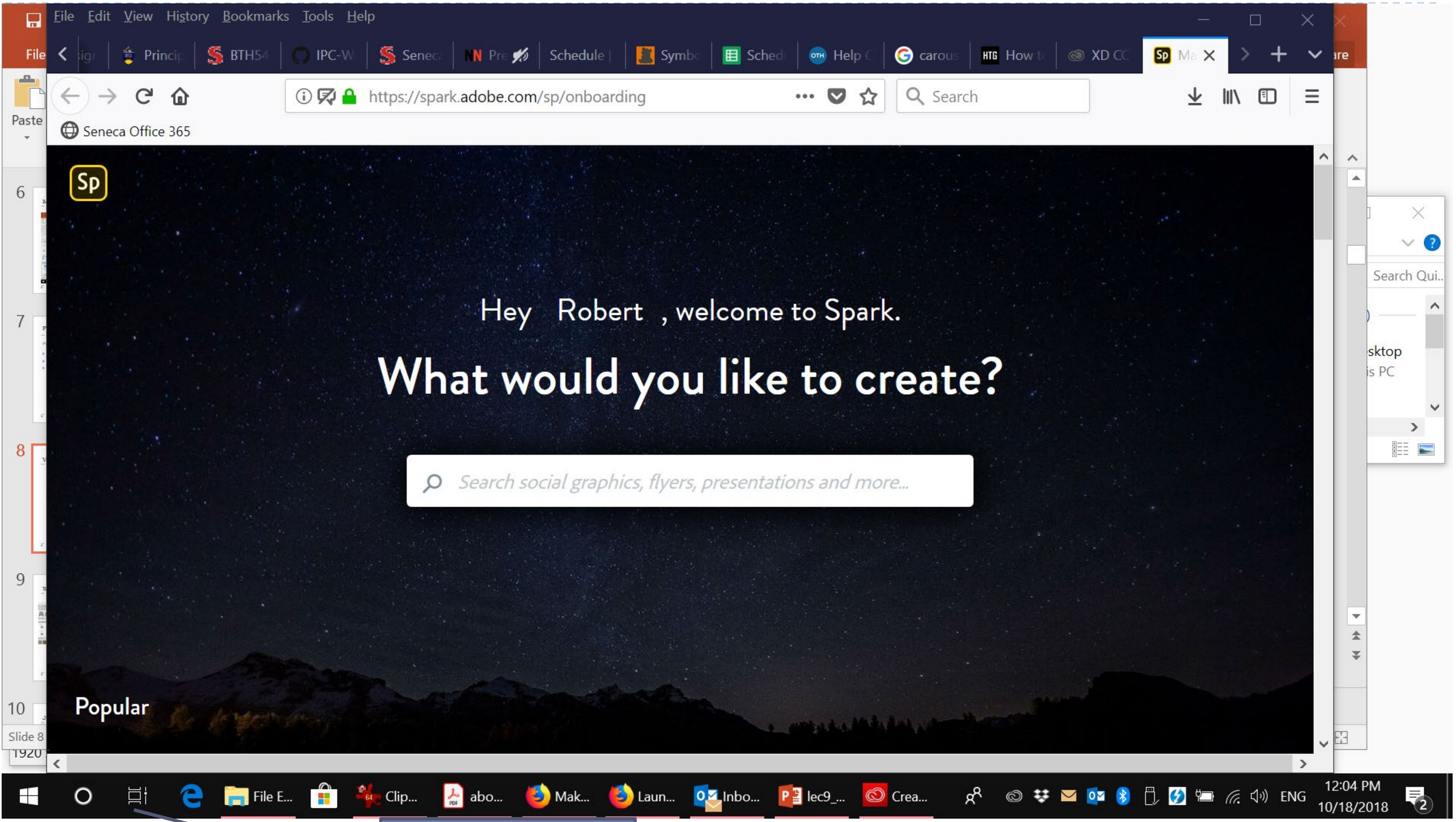
## Pros

- See everything big
- Good for small screens
- Concentrate on one thing without distractions

## Cons

- Difficult to work with two applications at once
- Need a taskbar to allow switching applications
- Unaware of what is happening in other applications
- Wastes space on large screens

# Virtual Desktops



Hey Robert , welcome to Spark.

What would you like to create?

Search social graphics, flyers, presentations and more...

Popular

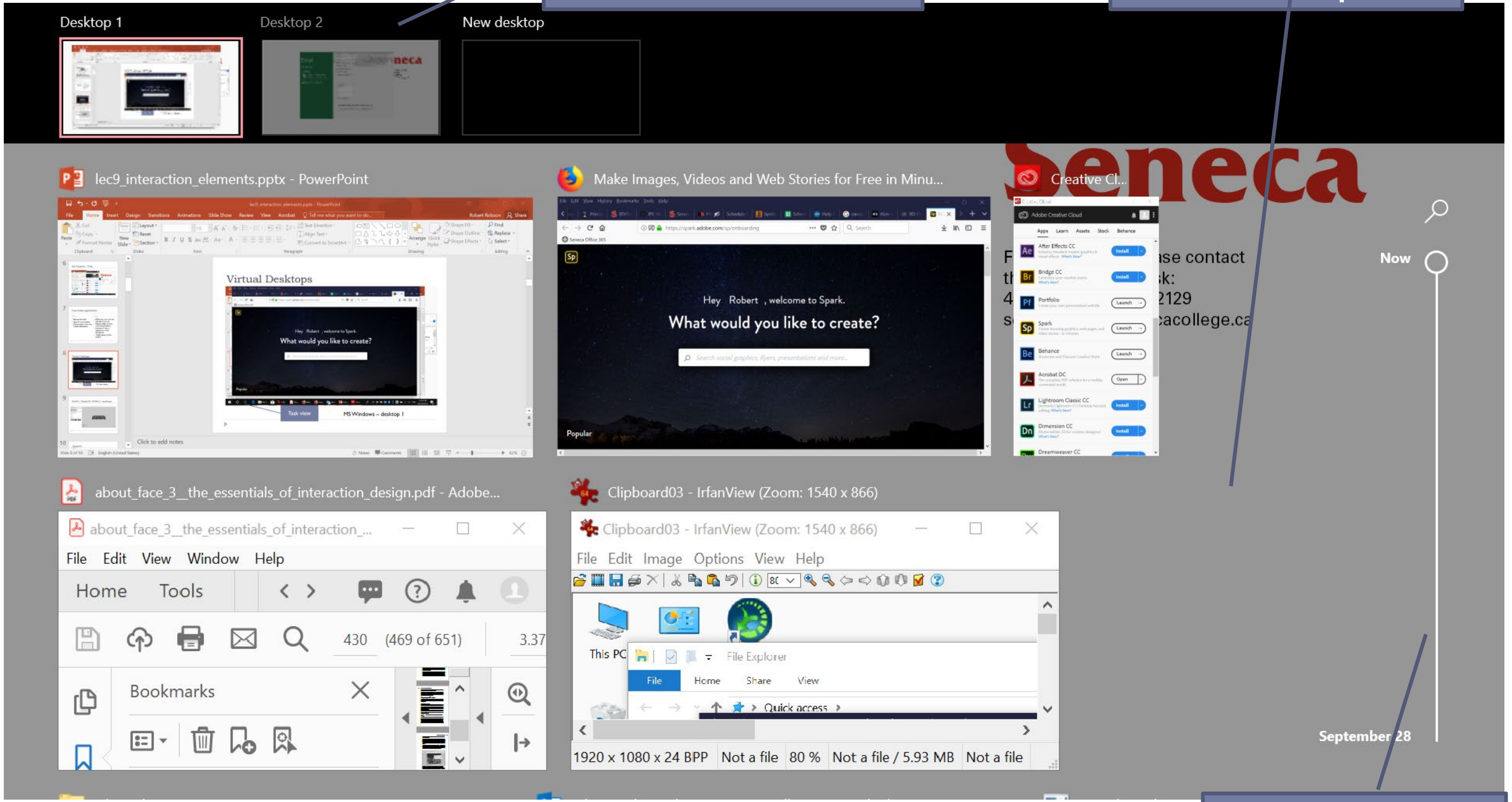
Task view

MS Windows – desktop 1

# Virtual Desktops

Desktop switcher

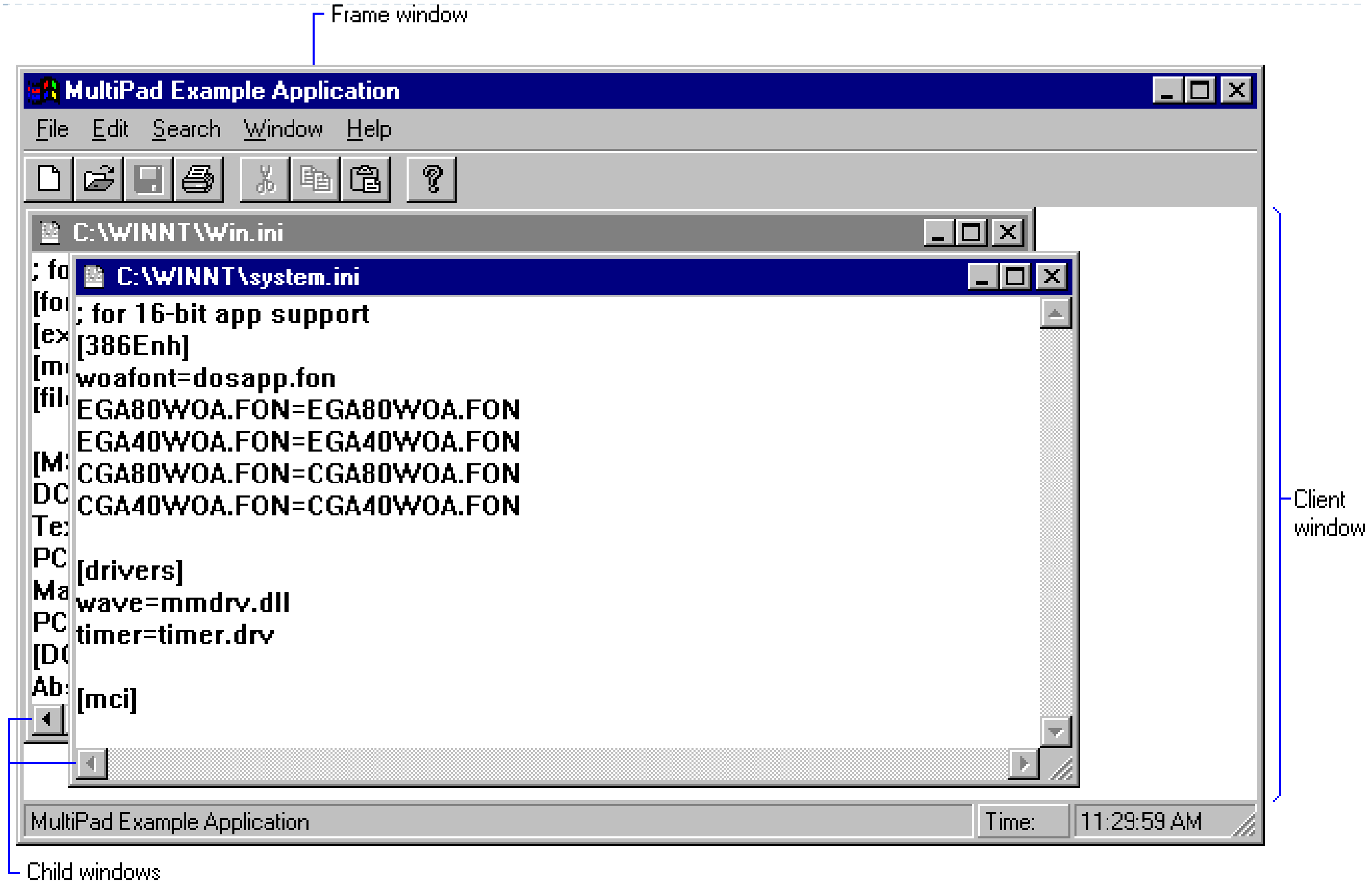
Apps on current desktop


Desktop 1
Desktop 2
New desktop
lec9_interaction_elements.pptx - PowerPoint
Make Images, Videos and Web Stories for Free in Minu...
Creative Cl...
Hey Robert , welcome to Spark.
What would you like to create?
Now
about_face_3__the_essentials_of_interaction_design.pdf - Adobe...
File Edit View Window Help
Home Tools
430 (469 of 651)
Bookmarks
Clipboard03 - IrfanView (Zoom: 1540 x 866)
File Edit Image Options View Help
This PC
File Explorer
File Home Share View
Quick access
1920 x 1080 x 24 BPP Not a file 80 % Not a file / 5.93 MB Not a file
September 28

MS Windows Task View

timeline

# Multi document interface



```
; for 16-bit app support
[386Enh]
woafont=dosapp.fon
EGA80WOA.FON=EGA80WOA.FON
EGA40WOA.FON=EGA40WOA.FON
CGA80WOA.FON=CGA80WOA.FON
CGA40WOA.FON=CGA40WOA.FON

[drivers]
wave=mmdrv.dll
timer=timer.drv

[mci]
```

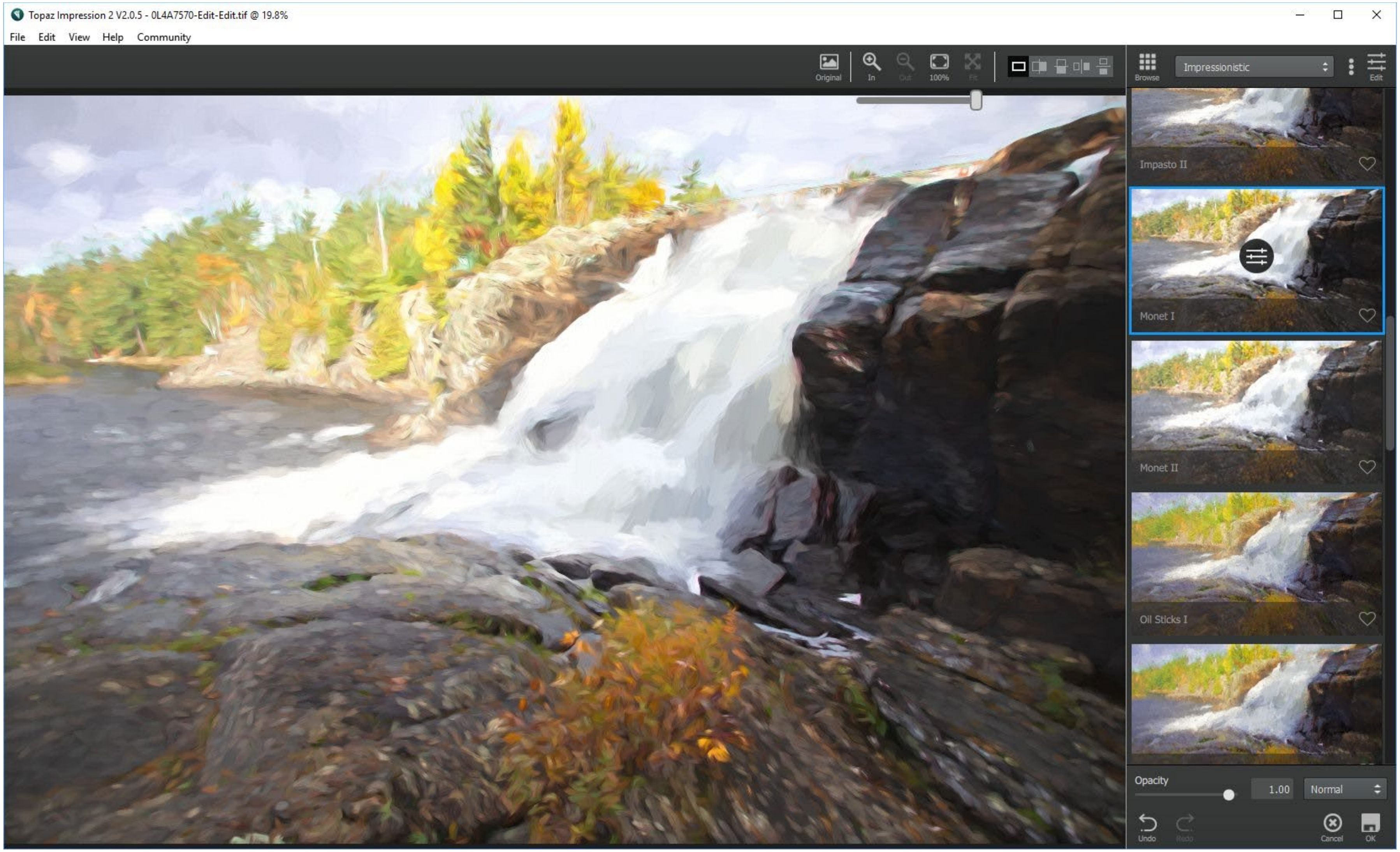
# Unnecessary Rooms

- We can think of an application like a house with rooms
  - Every room has a purpose that lets us do something unrelated to the rest of the house
  - When we go into a room we lose the ability to do things in the other rooms
  - We do not add a room to a house unless we need it
- The equivalent in an application is
  - Every secondary window (outside the main window) is the equivalent of a new room
  - The most common form of secondary window is the dialog box
  - These prevent the user from interacting with the main application

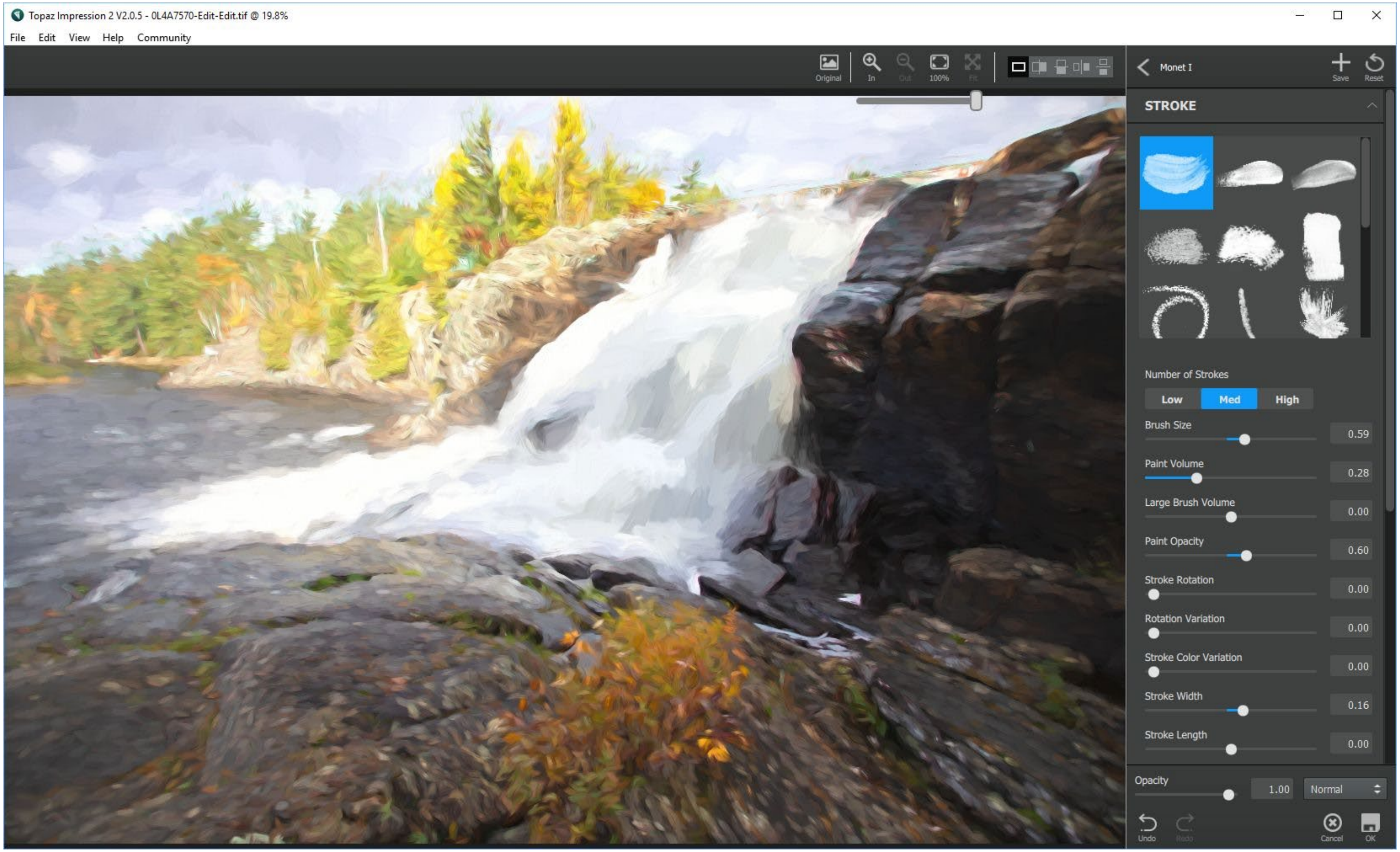
# Unnecessary rooms

- Many applications make frequent use of dialogs
- This restricts how you can interact with the application
- In many cases, this is poor design
- You could achieve the same thing with
  - A property sheet
  - A popup control

# Unnecessary rooms



# Unnecessary rooms



Topaz Impression 2 V2.0.5 - 0L4A7570-Edit-Edit.tif @ 19.8%

Monet I

STROKE

Number of Strokes: Low [✓ Med] High

Brush Size 0.59

Paint Volume 0.28

Large Brush Volume 0.00

Paint Opacity 0.60

Stroke Rotation 0.00

Rotation Variation 0.00

Stroke Color Variation 0.00

Stroke Width 0.16

Stroke Length 0.00

Opacity 1.00 Normal

# Direct Manipulation

- Direct manipulation was coined by Ben Schneiderman in 1974
- It is using some type of pointing device to directly manipulate the representation of some real-world object or concept in the virtual world of the screen
- In many cases, direct manipulation is intuitive
- However,
    - Sometimes people need a demo of how it works
    - It often requires dexterity
    - Some pointing devices are better than others

# Controls / Widgets

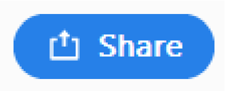
- Controls or widgets are the graphical objects which allow you to interact with your application
- There are several types
  - Imperative controls
    - Used to initiate a function
  - Selection controls
    - Used to select options or data
  - Entry controls
    - For data entry
  - Display controls
    - Used to directly manipulate the program

# Imperative Controls

- These initiate an immediate action
- Often associated with a verb indicating the action they perform

# Buttons

- Often have a 3D appearance
- Provides visual feedback to look like it is pushed in when clicked
- Can use text, icons or both
- Are always visible, as opposed to menu items
- Pure icons are difficult to interpret, particularly for novices and can benefit from tooltips



# Hyperlinks

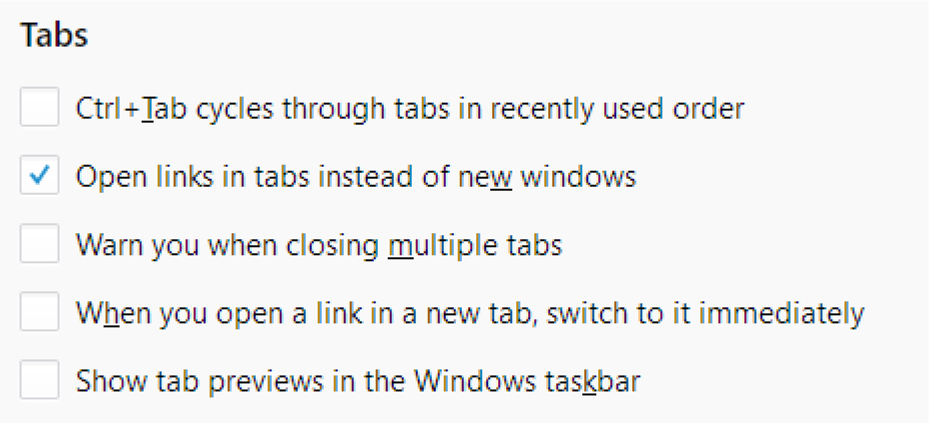
- These are links which will navigate to a different part of the data, application or even to another application
- Due to the ubiquity of links on the web, some designers are using them to perform actions rather than navigation
- Buttons are for actions and links for navigation

# Selection Controls

- Selection controls allow the user to select something (usually a noun) from some type of list of choices
- They usually do not initiate an action but provide data to an action or modify an action
- In this way they act as an adjective or adverb
- Some selections, like picking a font size, might initiate the action of changing the font size.

# Checkboxes

- Presents single, binary choice
- Usually text but can use an icon
- This can be an icon only that changes its appearance and becomes a *butcon*



Tabs

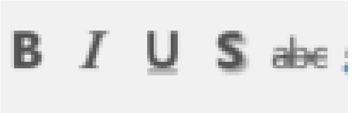
□ Ctrl+Tab cycles through tabs in recently used order

☑ Open links in tabs instead of new windows

□ Warn you when closing multiple tabs

□ When you open a link in a new tab, switch to it immediately

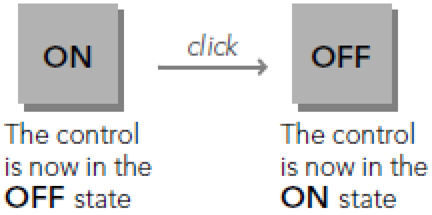
□ Show tab previews in the Windows taskbar



Butcons to select font styles

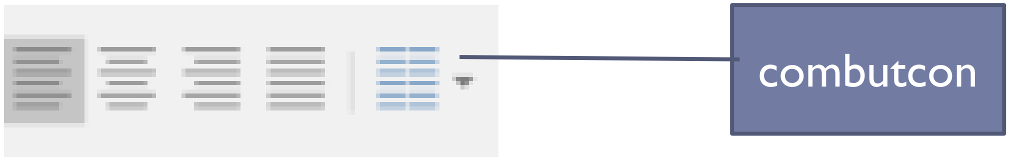
# Flip flop buttons

- These are buttons that change text or icon when clicked
- They can usually go between two states
- A good example is the play/pause button on an audio player
- This makes good use of screen real estate
- This uses the button to perform an action and to indicate state
- This confuses some users and should be replaced with 2 controls, one for the action and one for state


ON
click
OFF
The control is now in the OFF state
The control is now in the ON state

# Radio Buttons

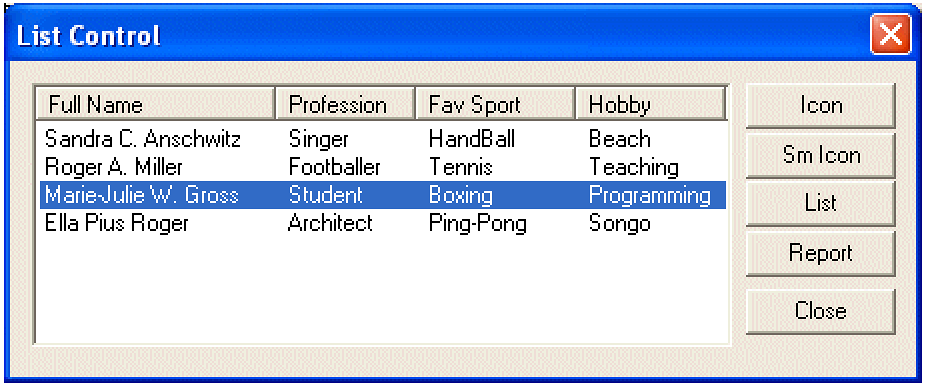
- Radio buttons select one of several mutually exclusive choices so that when one is selected all others are deselected
- Can consume more real estate than check boxes
- Often implemented as butcons to make better use of real estate
- A variant is the *combutcon* that shows a drop down list of choices



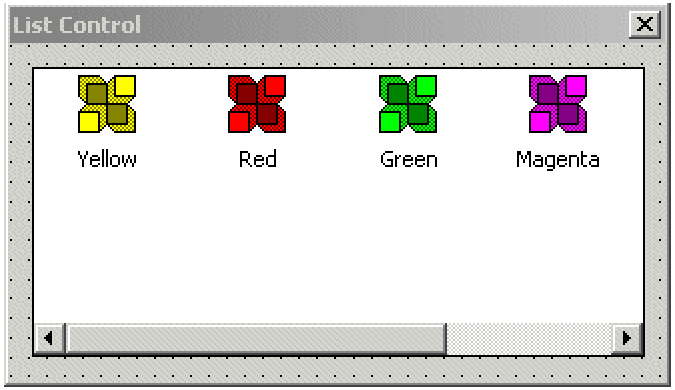
combutcon

# Lists

- These allow you to select text from a list
- It can either be a scrolling list or a dropdown list
- Lists can be configured to allow either single of multiple selections
- Problematic if it has to scroll to perform multiple selections
- Can use icons rather than text

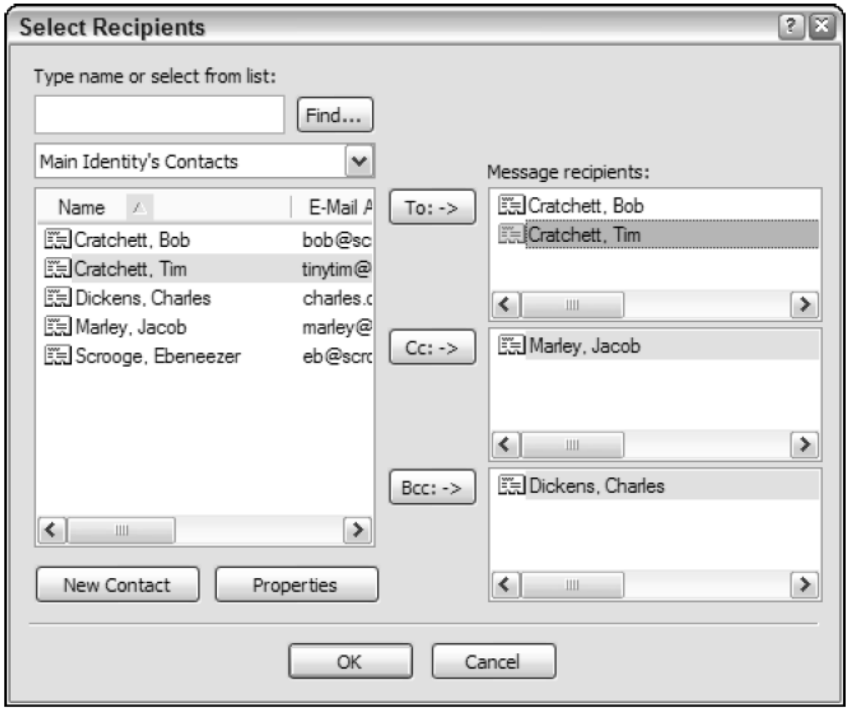


| Full Name | Profession | Fav Sport | Hobby |
|---|---|---|---|
| Sandra C. Anschwitz | Singer | HandBall | Beach |
| Roger A. Miller | Footballer | Tennis | Teaching |
| Marie-Julie W. Gross | Student | Boxing | Programming |
| Ella Pius Roger | Architect | Ping-Pong | Songo |



# Drag and Drop from lists

- It is handy to be able to drag and drop from a list to a destination
- This avoids the need for the move controls in the dialog to the right



| Name | E-Mail A |
|---|---|
| Cratchett, Bob | bob@sc |
| Cratchett, Tim | tinytim@ |
| Dickens, Charles | charles.c |
| Marley, Jacob | marley@ |
| Scrooge, Ebeneezer | eb@scrc |

To: Cratchett, Bob<br>Cratchett, Tim

Cc: Marley, Jacob

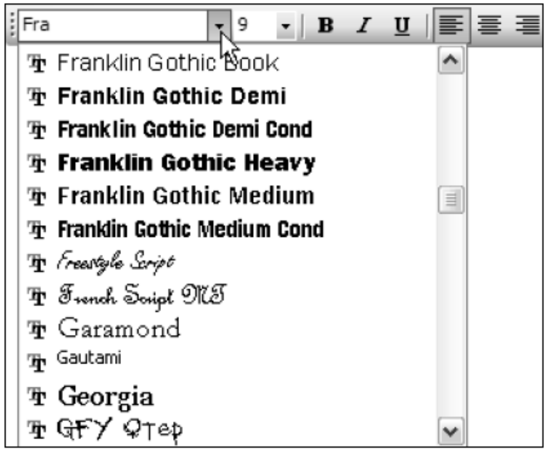
Bcc: Dickens, Charles

# Ordering lists

- Ordering the values in the lists makes it easier to find and select items
- Column headers can be used to reverse the sort order
- You might want to implement algorithms to sort by
  - Frequency of access
  - Some weighting factor not shown on screen
- You also might want to allow people to drag to reposition the items in the list
- Lists can be made to scroll either vertically or horizontally
- Horizontal scrolling of text is usually a bad idea since some of the text is always hidden

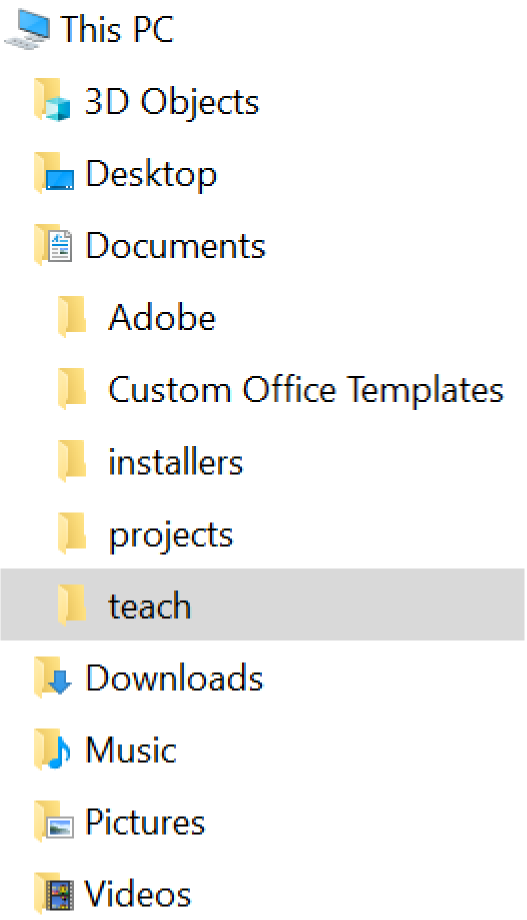
# Combo Boxes

- This is a drop down list where the user can select the value displayed from the list
- Some allow new values to be added or the selected value edited
- Combo boxes make good use of screen real estate
- You can implement drag and drop for them as well as with lists



# Tree Controls

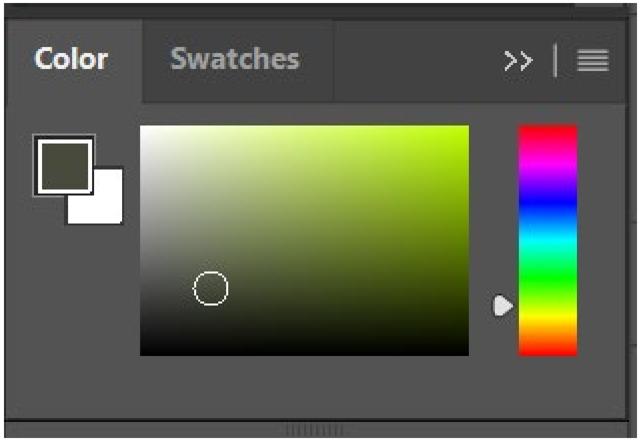
- Tree controls display a hierarchy, such as the file system
- While this control is intuitive for programmers, many users have difficulty in thinking in terms of hierarchical structures
- This means it is best to only use it when everyone views the content as a tree as they would with a family tree

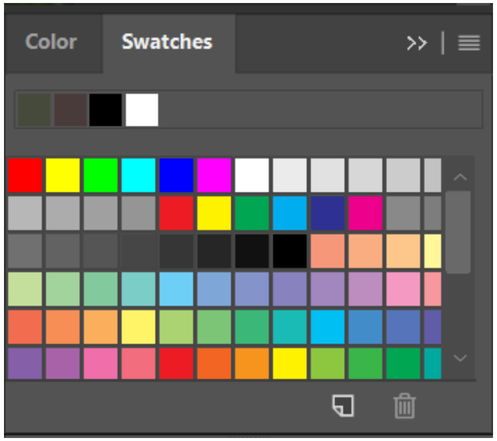


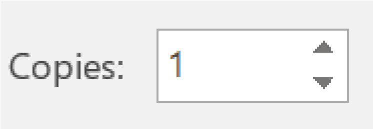
# Entry Controls

- These allow the entry of new information into an application rather than just selecting information from a list
- A bounded entry control is one which restricts the values which can be entered. These should be used whenever the data has natural limits

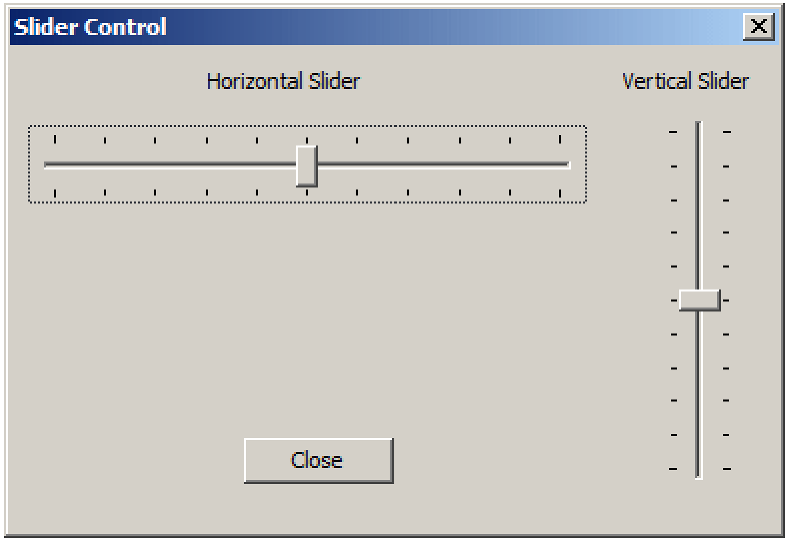
# Bounded Entry Controls

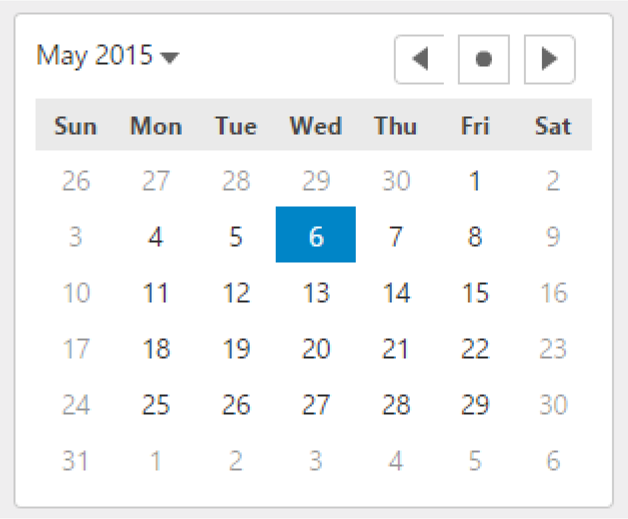






Copies: 1



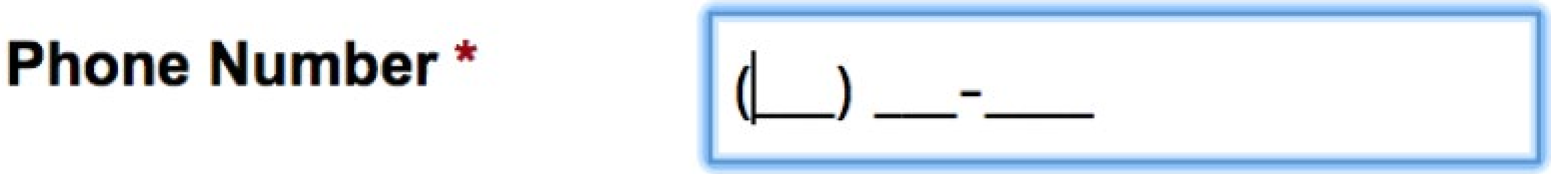


May 2015

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|---|---|---|---|---|---|---|
| 26 | 27 | 28 | 29 | 30 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 1 | 2 | 3 | 4 | 5 | 6 |

# Unbounded entry controls

- The text box is an unbounded entry control since the user can enter anything
- When you want to restrict what can be entered
  - Passively validate the input and show an error dialog (bad)
  - Actively validate so that incorrect key strokes are discarded
  - Provide a template of what they entry should look like (phone number and date)



Phone Number *

(__) ___-___

# Unbounded entry controls

- Real-time validation can introduce performance issues
  - You can set a timer on every keystroke and, if the user pauses for more than half a second, start validation on the assumption the user has paused to think
- To provide feedback the control can change background colour
  - Red is a weak password and green is a strong password
  - Field turns red if invalid character is entered
- A cluebox, like a tooltip, can be shown when the user hovers over a control. This will display acceptable values or an example of input

# Measurement

- When the input is measurements, there are different units which could be used
  - Allow the user to provide a unit suffix
    - 5.3 in OR 4.6 cm OR 72pt
  - This will let the user express it in the most familiar units and the program will perform any conversions
- Programs will often convert the input into the default measurements that have been configured for the application

# Output with entry controls

- While you can use the text box to show output
  - This is usually a bad idea as people expect it to be for input, not output
  - You might write output to an input control if you expect the user to edit the output before it is finalized

# Display controls

- These are intended to manage the layout of the controls and content of the application as well as display information
- Examples include
  - Scrollbars
  - Screen splitters
  - Grids
  - Group boxes
  - Labels

# Text controls

- These are common and used to display
  - Labels
  - Textual output
- Sometimes, these are used in dialogs to gather input when a WYSIWYG display of formatted text could be used more effectively

# Scrollbars

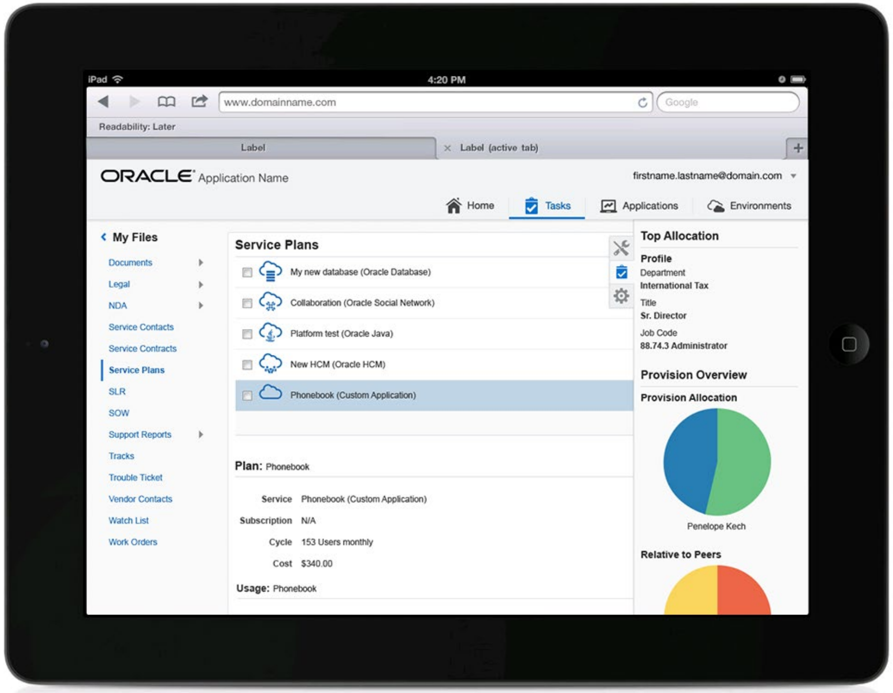
- Scrollbars
  - Let a small window view a larger amount of information
  - Give feedback to the user as to where in the document they are positioned
  - Some vary the size of the thumb to show what fraction of the document is visible in the window
  - Some show a preview of the text under the thumb as it is scrolled before being released
- Downsides include
  - Precision with the mouse
  - Does not work well with unlimited values such as time

# Splitters

- This is a handy way of splitting a sovereign application into parts
- The splitters are usually visible and
  - Can be dragged to resize portions of the application
  - Can be moved to an edge to remove one view that was being split by the application

# Drawers

- Drawers contain content but appear and disappear as needed
- They are similar to dialogs but are part of the main application window



**Service Plans**

- □ My new database (Oracle Database)
- □ Collaboration (Oracle Social Network)
- □ Platform test (Oracle Java)
- □ New HCM (Oracle HCM)
- □ Phonebook (Custom Application)

**Plan:** Phonebook

| | |
|---|---|
| Service | Phonebook (Custom Application) |
| Subscription | N/A |
| Cycle | 153 Users monthly |
| Cost | $340.00 |

**Usage:** Phonebook

**Top Allocation**

**Profile**
Department
International Tax
Title
Sr. Director
Job Code
88.74.3 Administrator

**Provision Overview**

**Provision Allocation**

**Relative to Peers**

# Dialogs

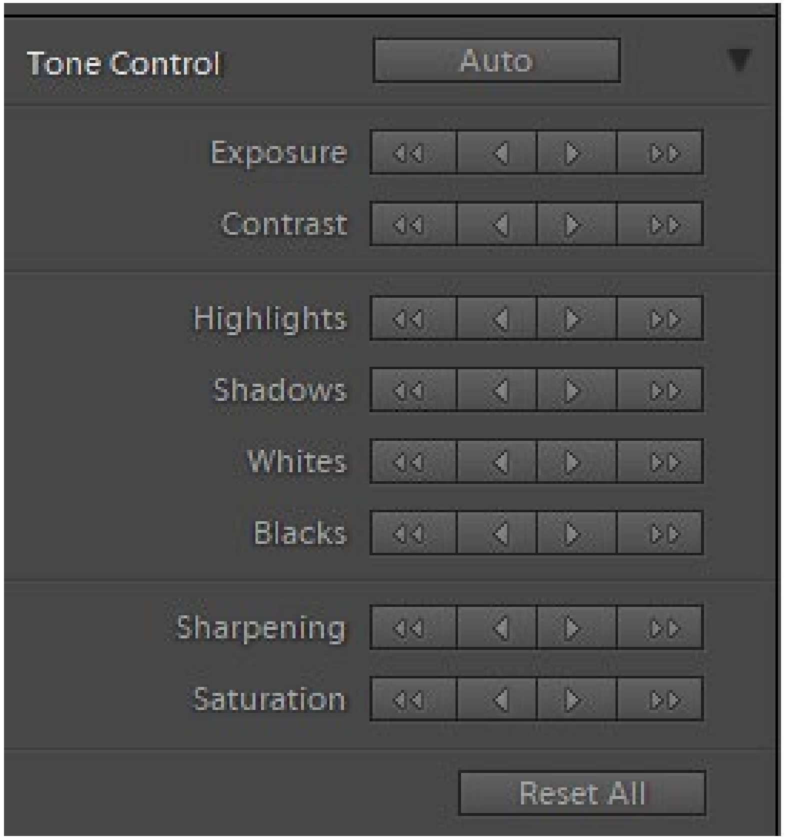
- Dialogs are usually
  - Poorly written,
  - Unhelpful,
  - Rude,
  - Prevent the user from interacting with the program,
  - Too late to prevent the problem
- Dialogs often indicate the failure of the designer to prevent the problem from happening
- Dialogs should be replaced with
  - Drawers or property sheets
  - Error prevention

# Flow

- Flow is when
  - You are totally concentrated on your task
  - You are unaware of your surroundings
  - You are highly productive

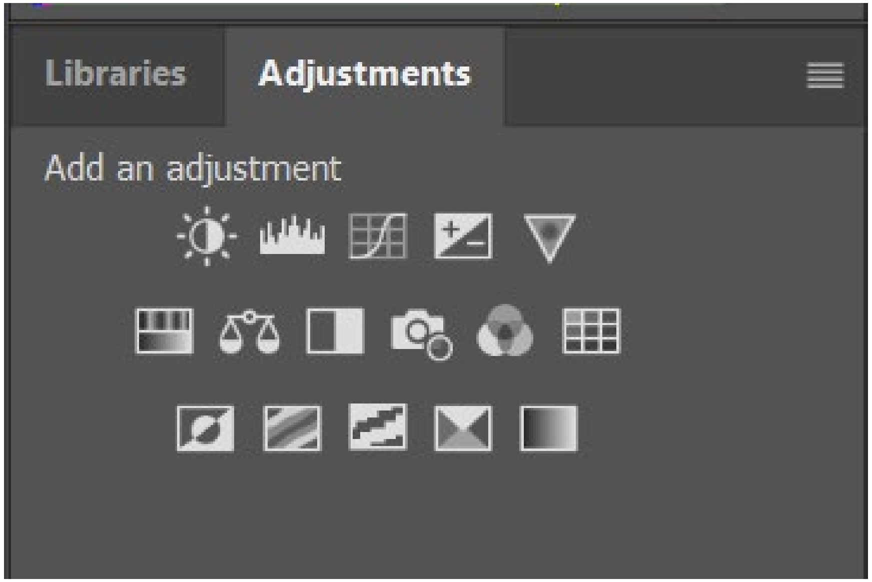
# Maintaining flow – mental models

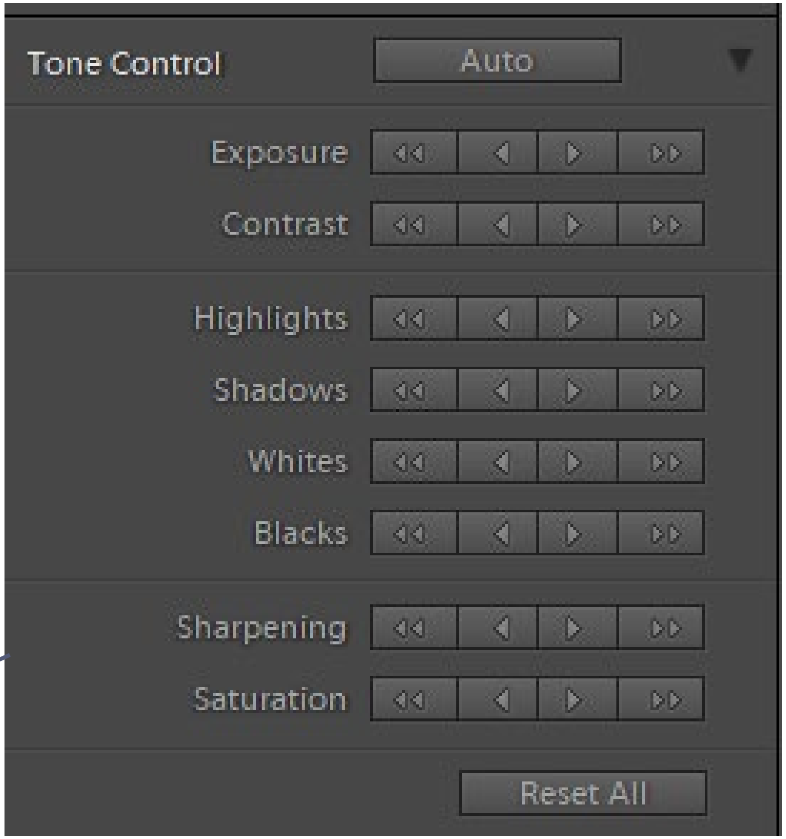
Have the interface reflect the mental model the user has of the problem space.



Terms are familiar and provide simple controls for the most common operations

# Maintaining flow – less is more



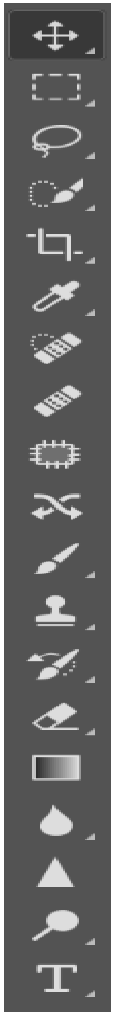


Photoshop image adjustments – complex operations that are powerful

Lightroom – simpler operations to get the job done

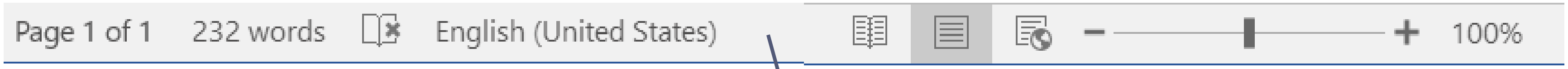
# Maintaining flow – direct, not discuss

Toolbars or palettes let the user direct the order of operations



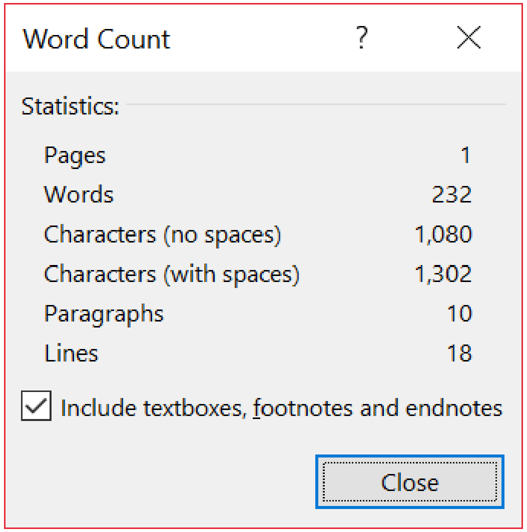
Do not build interfaces where you ask a series of questions and force the user to perform operations in an order determined by the application.

# Maintaining flow – modeless feedback



MS Word provides feedback at the bottom of the page without the user having to ask for it and it is provided continuously without having to go into a special mode as you would need with a dialog box.

Dialog box puts app into a new mode restricting interaction



Statistics:

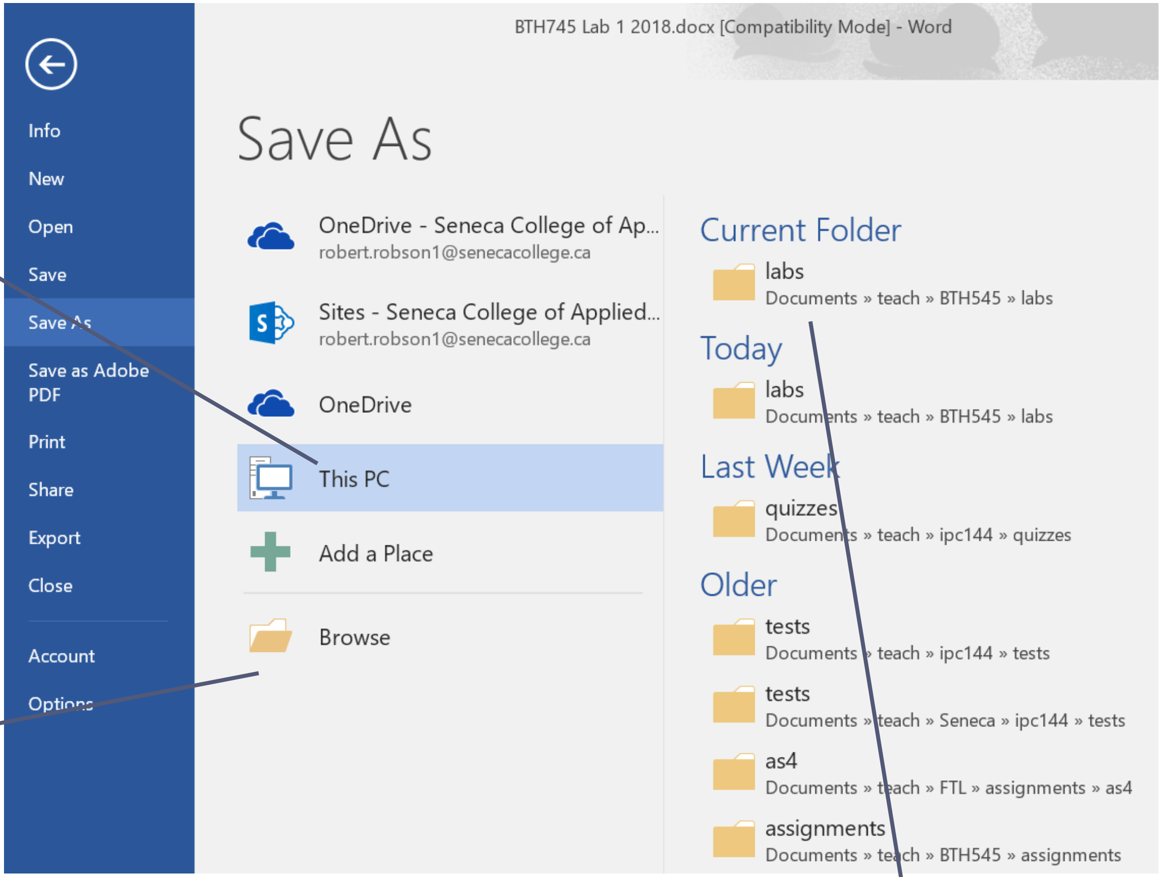
| | |
|---|---|
| Pages | 1 |
| Words | 232 |
| Characters (no spaces) | 1,080 |
| Characters (with spaces) | 1,302 |
| Paragraphs | 10 |
| Lines | 18 |

☑ Include textboxes, footnotes and endnotes

# Maintaining flow – design for probable but anticipate possible

Shows last place you saved

Other possible locations



BTH745 Lab 1 2018.docx [Compatibility Mode] - Word

Save As

OneDrive - Seneca College of Ap...
robert.robson1@senecacollege.ca

Sites - Seneca College of Applied...
robert.robson1@senecacollege.ca

OneDrive

This PC

Add a Place

Browse

Current Folder

labs
Documents » teach » BTH545 » labs

Today

labs
Documents » teach » BTH545 » labs

Last Week

quizzes
Documents » teach » ipc144 » quizzes

Older

tests
Documents » teach » ipc144 » tests

tests
Documents » teach » Seneca » ipc144 » tests

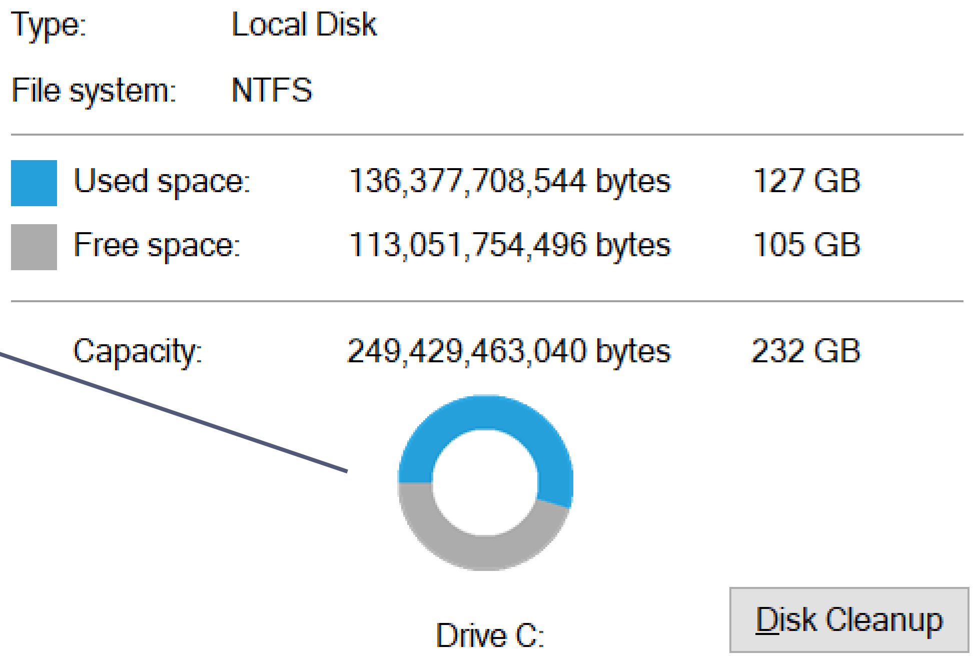
as4
Documents » teach » FTL » assignments » as4

assignments
Documents » teach » BTH545 » assignments

Most likely location

# Maintaining flow – contextualize information

When you want to see how full your disk is you want an approximate answer and the graph is better than the numbers



| | | | |
|---|---|---|---|
| Type: | Local Disk | | |
| File system: | NTFS | | |
| 🟦 Used space: | 136,377,708,544 bytes | 127 GB | |
| ⬜ Free space: | 113,051,754,496 bytes | 105 GB | |
| Capacity: | 249,429,463,040 bytes | 232 GB | |

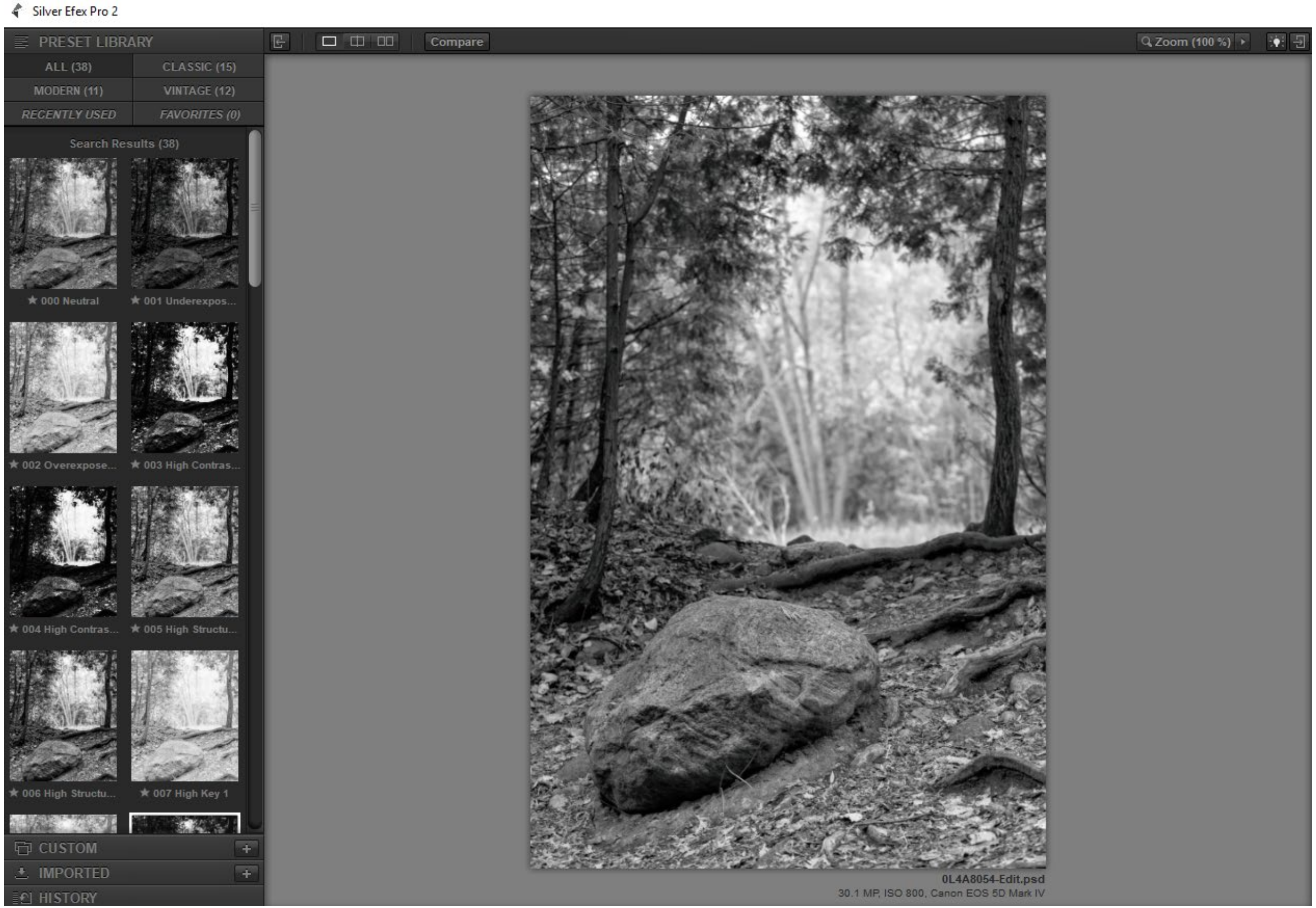
# Maintaining flow – unnecessary reporting

In UNIX commands that work print nothing.
Command which fail print messages.

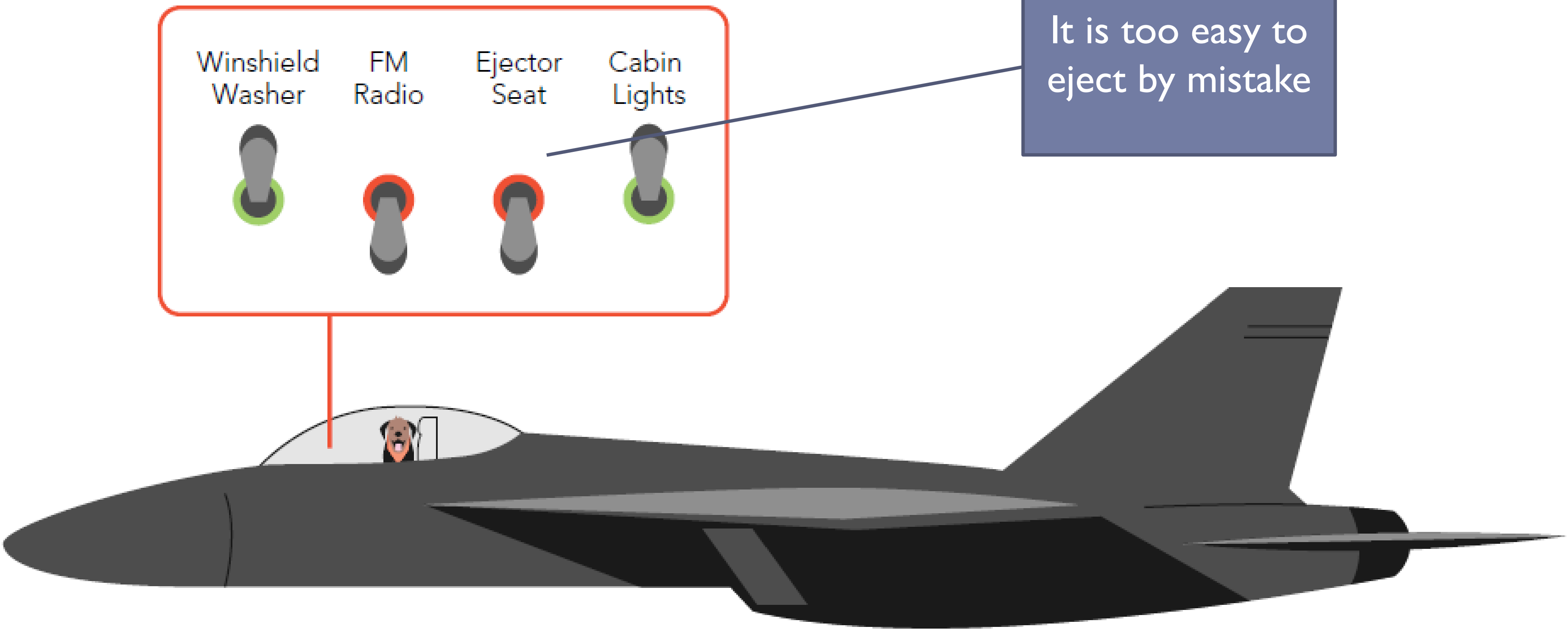
Do not report normalcy.

# Maintain flow – avoid blank slates

Application provides visual presets to avoid the "what do I do now"? phenomenon.



# Maintaining flow – hide the ejector seat button



It is too easy to eject by mistake

# Maintaining flow – optimize for responsiveness, accommodate latency

Assume operations are quick but if they are not provide the user feedback on how long they will take

# Animation

- Animation can be used to
  - Show the next steps in a workflow
  - Focus the user on one part of the interface
  - Show that a task has been completed
  - Help me to understand the meaning of data

* Alla Kudin

# Transitions

- Transitions show
- Parts of the interface moving on or off screen
- The availability or removal of controls
- The completion of a step in the workflow
- The availability of a new step in the workflow

Hey man! Check out this weather let's go for a walk?

* Tubik Studio