Using GUI interfaces

Indirect vs. Direct Manipulation
Instrumental Interaction

CS 349

June 19

Reminder – GUI Interaction

For desktop-based GUI interaction, we can assume the presence of a

- Screen capable of “high-resolution” graphics output
- Text entry (e.g., keyboard), and
- Pointing device (e.g., mouse, touchpad)

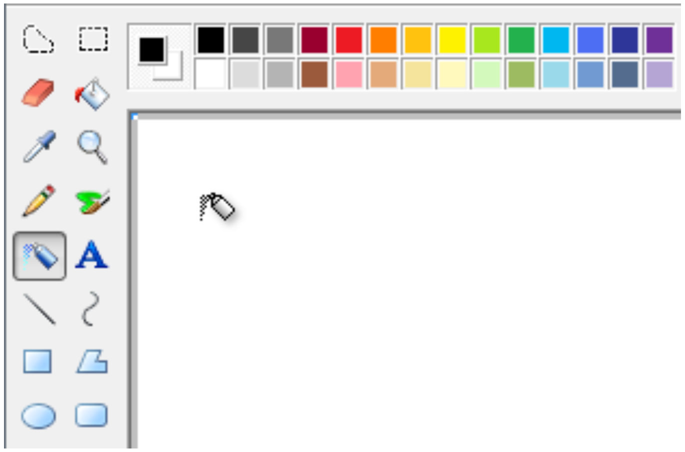
Reminder – WIMP

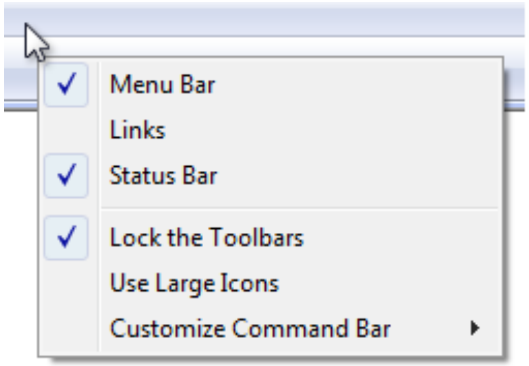
Interfaces consists of Windows, Icons, Menus, and Pointers

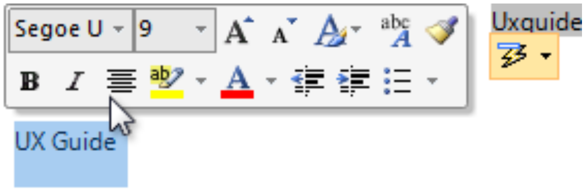
It is usually associated with the “desktop” metaphor.

Users interact by pointing + clicking

- Point at the object of interest (e.g., widget, image, text)
- Click to select, Drag to move, Double-click to activate
- “Real-time” interaction and feedback, using on-screen animation







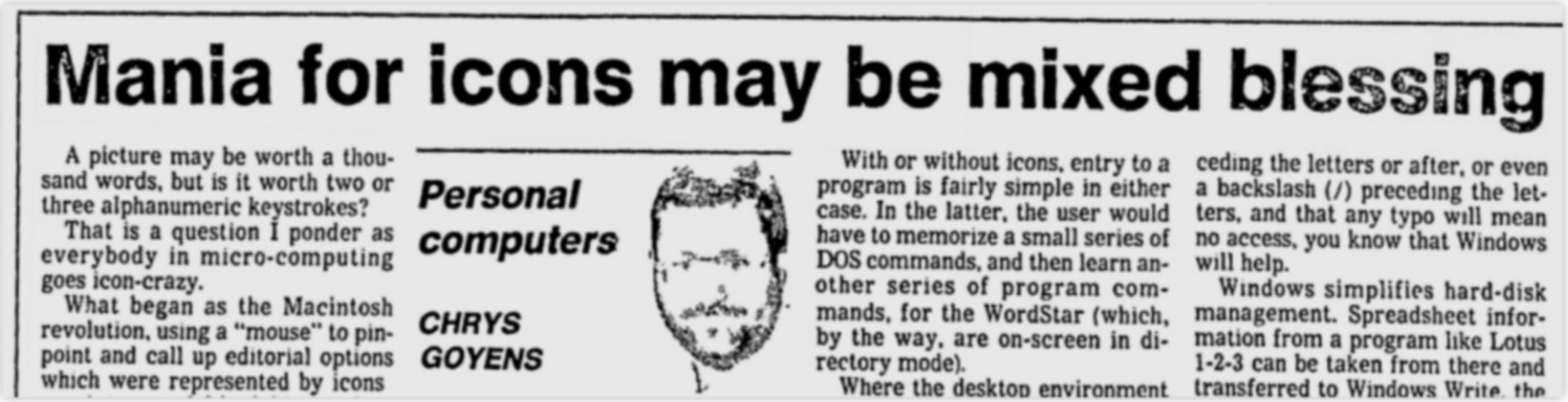
GUI Interaction – Advantages

The user always remains in control

- The system waits for input, then responds
- Unlike batch where the user waits for the program to complete

The design emphasizes **recognition** of interface features over **recall** of complex commands.

- Utilizes familiar graphical elements across different applications.
- Enables discovery of options and experimentation



Mania for icons may be mixed blessing

Personal computers

CHRYS GOYENS

A picture may be worth a thousand words, but is it worth two or three alphanumeric keystrokes?

That is a question I ponder as everybody in micro-computing goes icon-crazy.

What began as the Macintosh revolution, using a "mouse" to pinpoint and call up editorial options which were represented by icons

With or without icons, entry to a program is fairly simple in either case. In the latter, the user would have to memorize a small series of DOS commands, and then learn another series of program commands, for the WordStar (which, by the way, are on-screen in directory mode).

Where the desktop environment

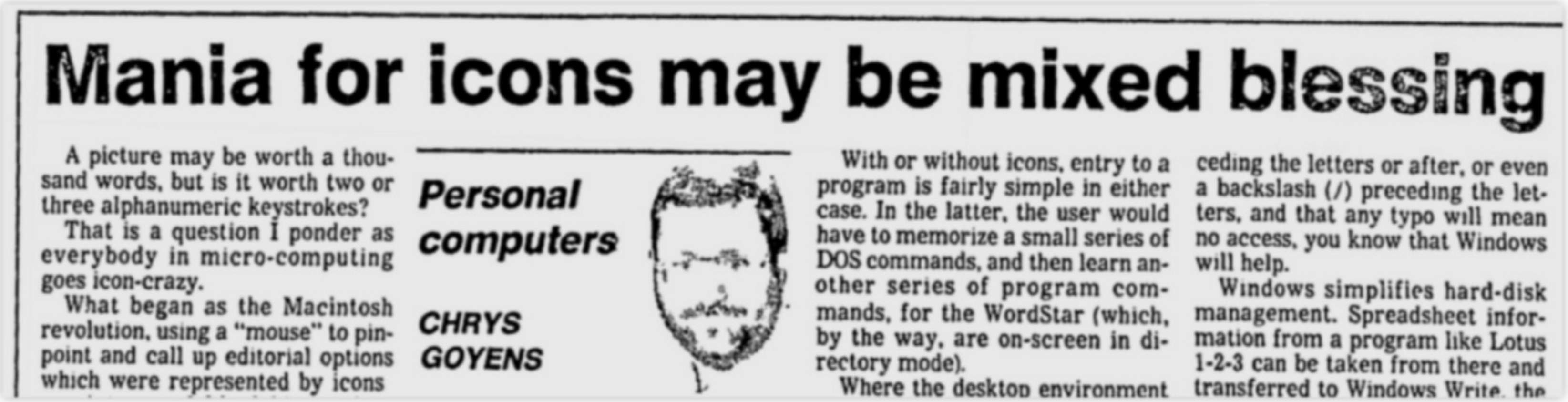
ceding the letters or after, or even a backslash (/) preceding the letters, and that any typo will mean no access, you know that Windows will help.

Windows simplifies hard-disk management. Spreadsheet information from a program like Lotus 1-2-3 can be taken from there and transferred to Windows Write, the

GUI Interaction – Advantages

It uses metaphor to make the interface more familiar

- Graphical objects results in an interaction language that is closer to users' own language, and closer to the task domain
- Examples include: "desktop", "folder", and "drag-and-drop"



Mania for icons may be mixed blessing

Personal computers

CHRYS GOYENS

A picture may be worth a thousand words, but is it worth two or three alphanumeric keystrokes?

That is a question I ponder as everybody in micro-computing goes icon-crazy.

What began as the Macintosh revolution, using a "mouse" to pinpoint and call up editorial options which were represented by icons

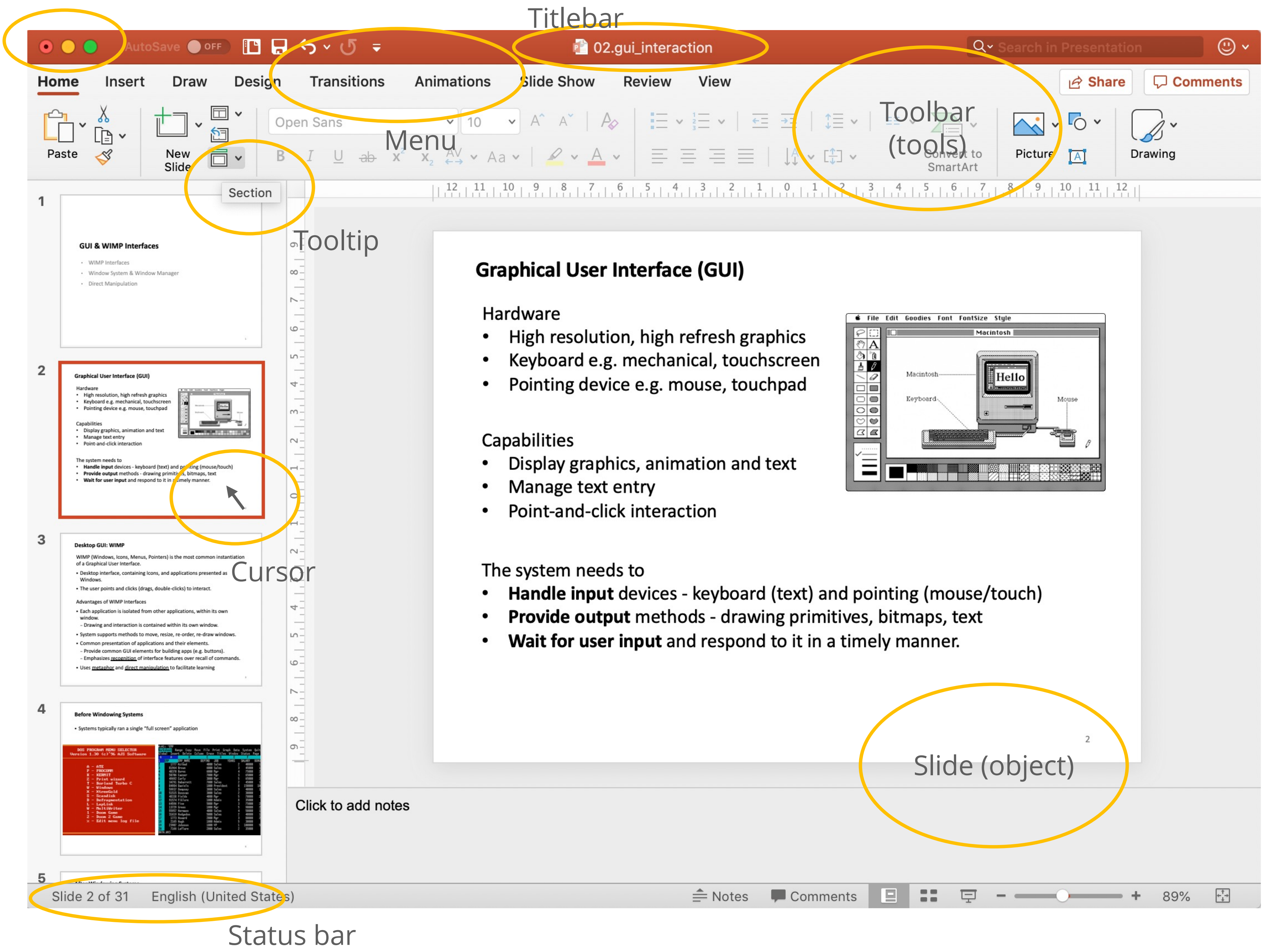
With or without icons, entry to a program is fairly simple in either case. In the latter, the user would have to memorize a small series of DOS commands, and then learn another series of program commands, for the WordStar (which, by the way, are on-screen in directory mode).

Where the desktop environment

ceding the letters or after, or even a backslash (/) preceding the letters, and that any typo will mean no access, you know that Windows will help.

Windows simplifies hard-disk management. Spreadsheet information from a program like Lotus 1-2-3 can be taken from there and transferred to Windows Write, the

Consistent interface elements also help to make a GUI explorable and predictable for users.



Graphical User Interface (GUI)

Hardware

- High resolution, high refresh graphics
- Keyboard e.g. mechanical, touchscreen
- Pointing device e.g. mouse, touchpad

Capabilities

- Display graphics, animation and text
- Manage text entry
- Point-and-click interaction

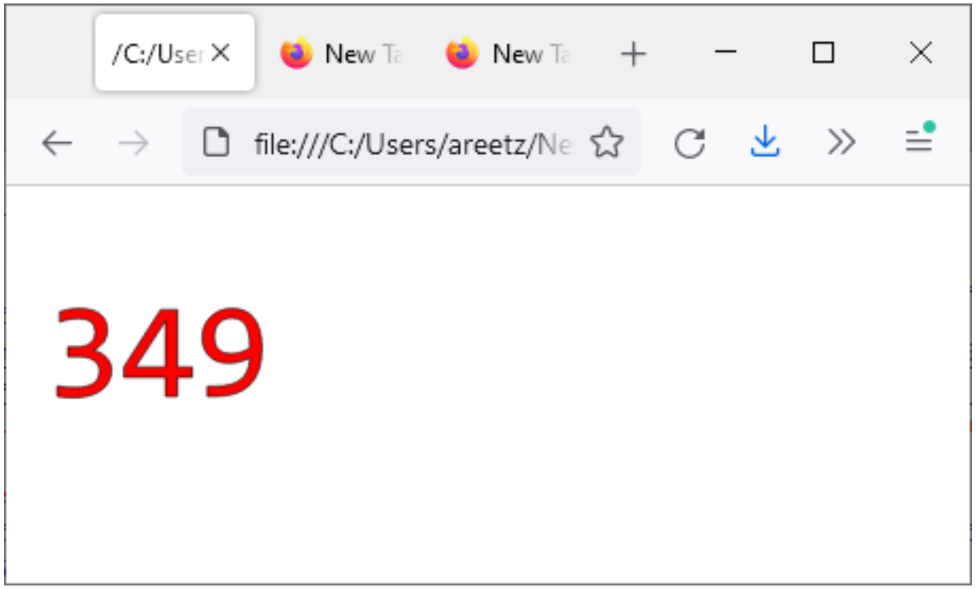
The system needs to

- **Handle input** devices - keyboard (text) and pointing (mouse/touch)
- **Provide output** methods - drawing primitives, bitmaps, text
- **Wait for user input** and respond to it in a timely manner.

Indirect vs. Direct Manipulation

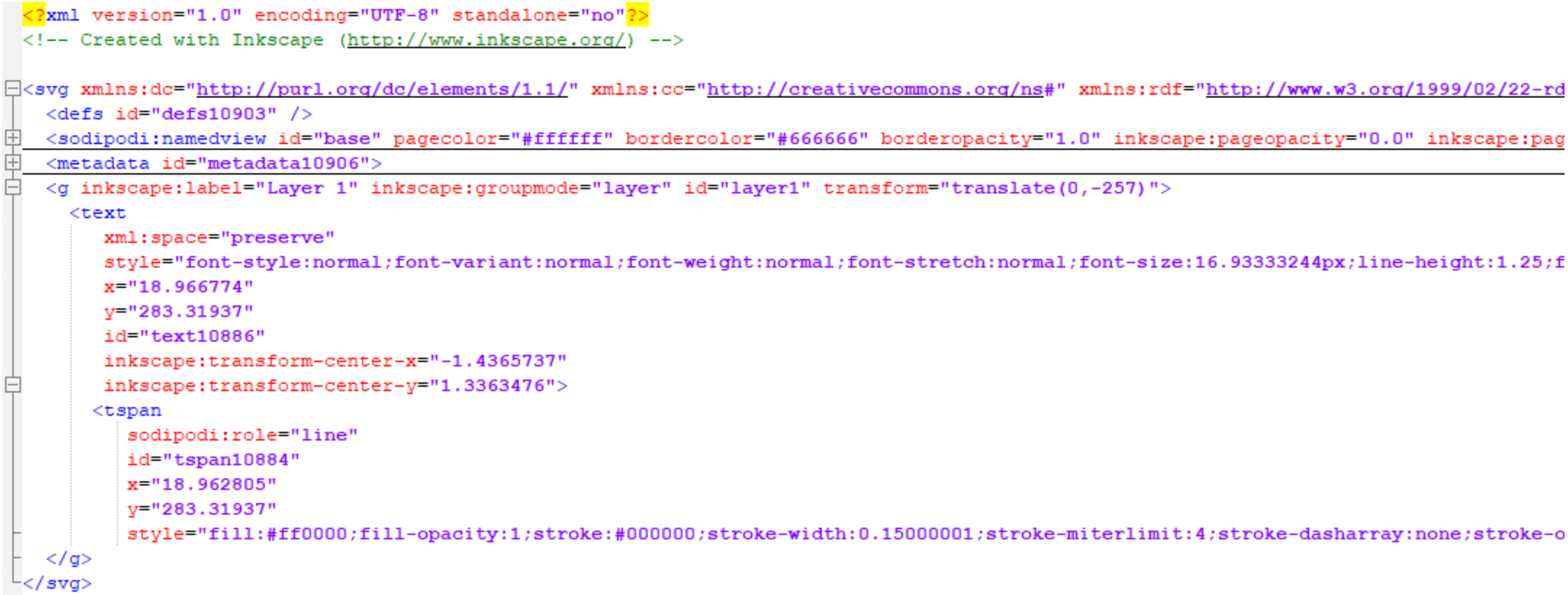
Indirect vs. Direct Manipulation

Our goal is modifying this SVG-file (Scalable Vector Graphic), here shown in Firefox. We want to move the text “**349**” to the right:



Indirect Manipulation

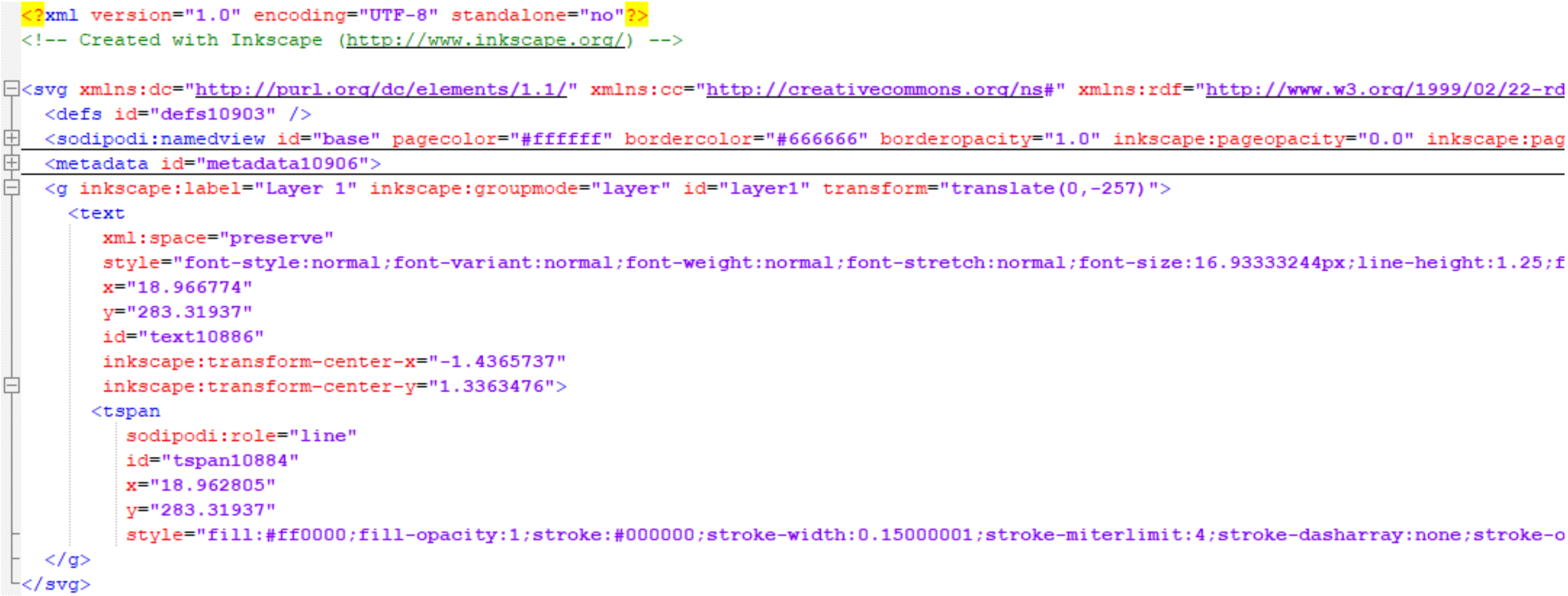
One way to achieve this goal is by modifying the SVG-file:



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- Created with Inkscape (http://www.inkscape.org/) -->

<svg xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:cc="http://creativecommons.org/ns#" xmlns:rdf="http://www.w3.org/1999/02/22-rd
  <defs id="defs10903" />
  <sodipodi:namedview id="base" pagecolor="#ffffff" bordercolor="#666666" borderopacity="1.0" inkscape:pageopacity="0.0" inkscape:pag
  <metadata id="metadata10906">
  <g inkscape:label="Layer 1" inkscape:groupmode="layer" id="layer1" transform="translate(0,-257)">
    <text
       xml:space="preserve"
       style="font-style:normal;font-variant:normal;font-weight:normal;font-stretch:normal;font-size:16.93333244px;line-height:1.25;f
       x="18.966774"
       y="283.31937"
       id="text10886"
       inkscape:transform-center-x="-1.4365737"
       inkscape:transform-center-y="1.3363476">
      <tspan
         sodipodi:role="line"
         id="tspan10884"
         x="18.962805"
         y="283.31937"
         style="fill:#ff0000;fill-opacity:1;stroke:#000000;stroke-width:0.15000001;stroke-miterlimit:4;stroke-dasharray:none;stroke-o
  </g>
</svg>
```

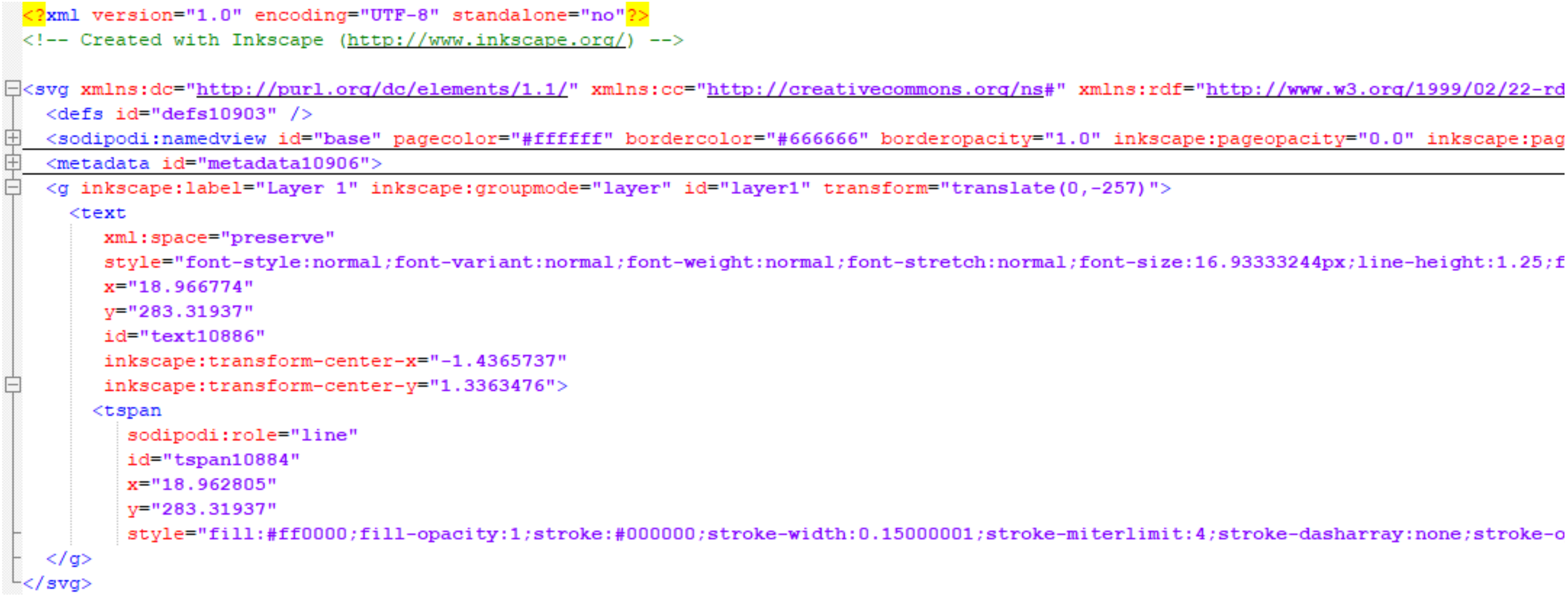
Indirect Manipulation

Is this good or bad (in terms of interaction design)?



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- Created with Inkscape (http://www.inkscape.org/) -->

<svg xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:cc="http://creativecommons.org/ns#" xmlns:rdf="http://www.w3.org/1999/02/22-rd
  <defs id="defs10903" />
  <sodipodi:namedview id="base" pagecolor="#ffffff" bordercolor="#666666" borderopacity="1.0" inkscape:pageopacity="0.0" inkscape:pag
  <metadata id="metadata10906">
  <g inkscape:label="Layer 1" inkscape:groupmode="layer" id="layer1" transform="translate(0,-257)">
    <text
       xml:space="preserve"
       style="font-style:normal;font-variant:normal;font-weight:normal;font-stretch:normal;font-size:16.93333244px;line-height:1.25;f
       x="18.966774"
       y="283.31937"
       id="text10886"
       inkscape:transform-center-x="-1.4365737"
       inkscape:transform-center-y="1.3363476">
      <tspan
         sodipodi:role="line"
         id="tspan10884"
         x="18.962805"
         y="283.31937"
         style="fill:#ff0000;fill-opacity:1;stroke:#000000;stroke-width:0.15000001;stroke-miterlimit:4;stroke-dasharray:none;stroke-o
  </g>
</svg>
```

Indirect Manipulation



```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- Created with Inkscape (http://www.inkscape.org/) -->

<svg xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:cc="http://creativecommons.org/ns#" xmlns:rdf="http://www.w3.org/1999/02/22-rd
  <defs id="defs10903" />
  <sodipodi:namedview id="base" pagecolor="#ffffff" bordercolor="#666666" borderopacity="1.0" inkscape:pageopacity="0.0" inkscape:pag
  <metadata id="metadata10906">
  <g inkscape:label="Layer 1" inkscape:groupmode="layer" id="layer1" transform="translate(0,-257)">
    <text
       xml:space="preserve"
       style="font-style:normal;font-variant:normal;font-weight:normal;font-stretch:normal;font-size:16.93333244px;line-height:1.25;f
       x="18.966774"
       y="283.31937"
       id="text10886"
       inkscape:transform-center-x="-1.4365737"
       inkscape:transform-center-y="1.3363476">
      <tspan
         sodipodi:role="line"
         id="tspan10884"
         x="18.962805"
         y="283.31937"
         style="fill:#ff0000;fill-opacity:1;stroke:#000000;stroke-width:0.15000001;stroke-miterlimit:4;stroke-dasharray:none;stroke-o
  </g>
</svg>
```

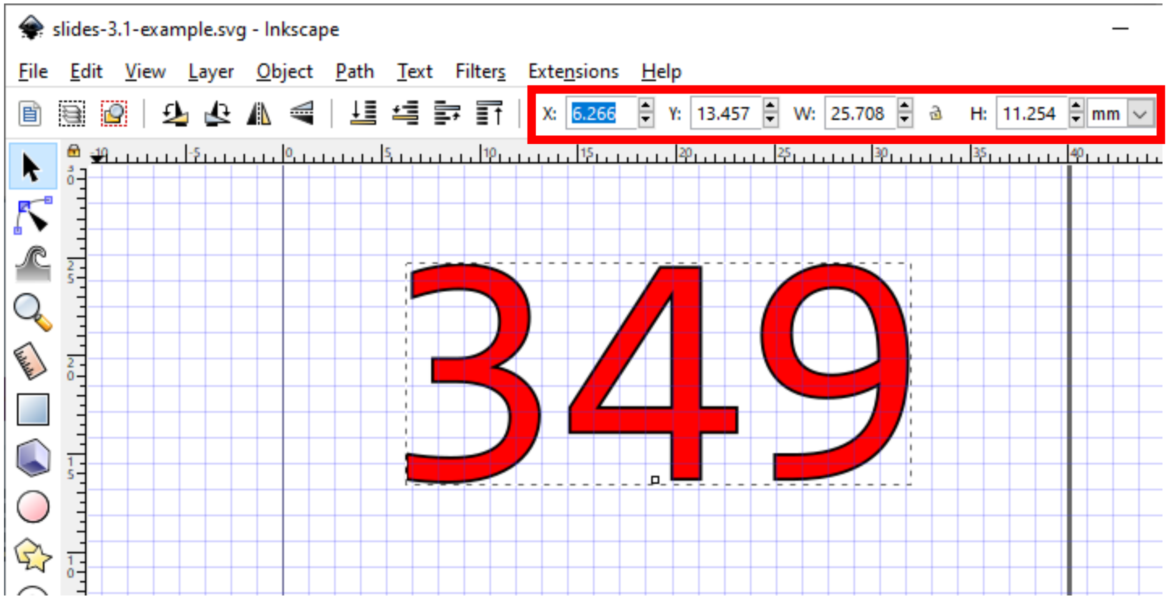
Advantages:

- Can be perfectly precise
- Can be automated (e.g., when batch-processing multiple files)

Disadvantages:

- Unclear which value to modify
- Unclear what the values mean
- Can be time-consuming on complex files
- No visual feedback of the result of the modification

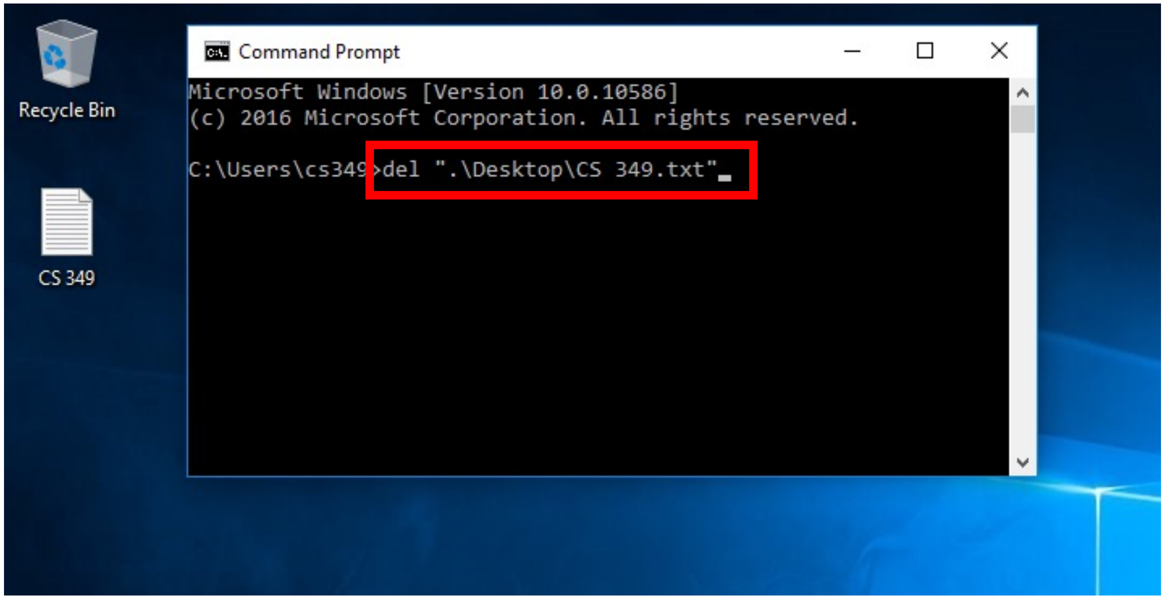
Indirect Manipulation – Other Examples


slides-3.1-example.svg - Inkscape
File Edit View Layer Object Path Text Filters Extensions Help
349

X: 6.266 Y: 13.457 W: 25.708 H: 11.254 mm

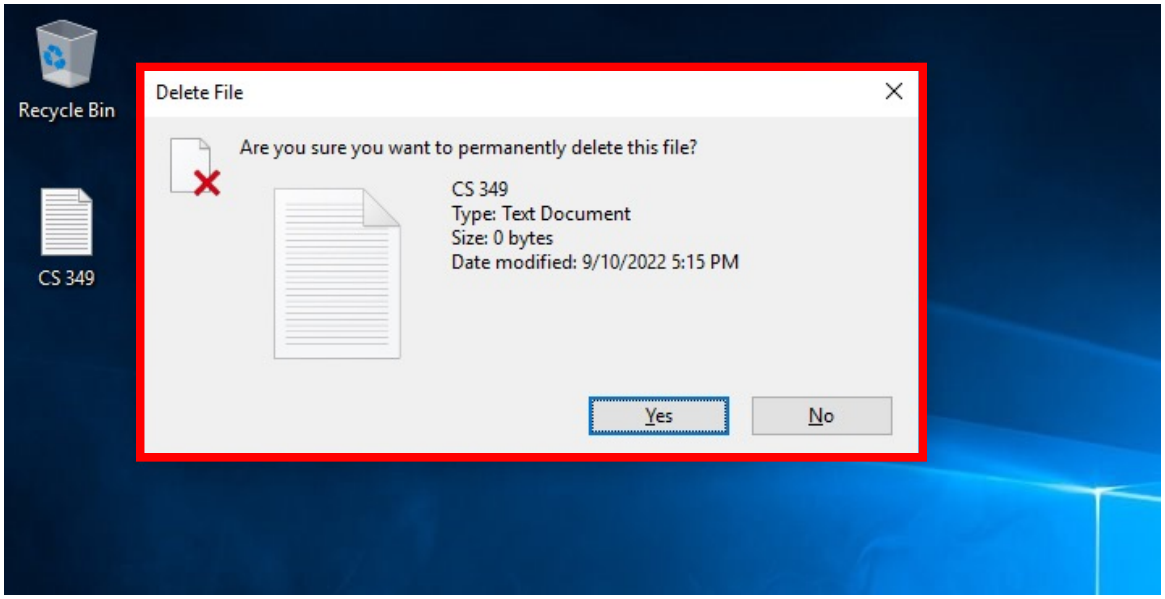
Other disadvantages

- Oftentimes no “Undo”



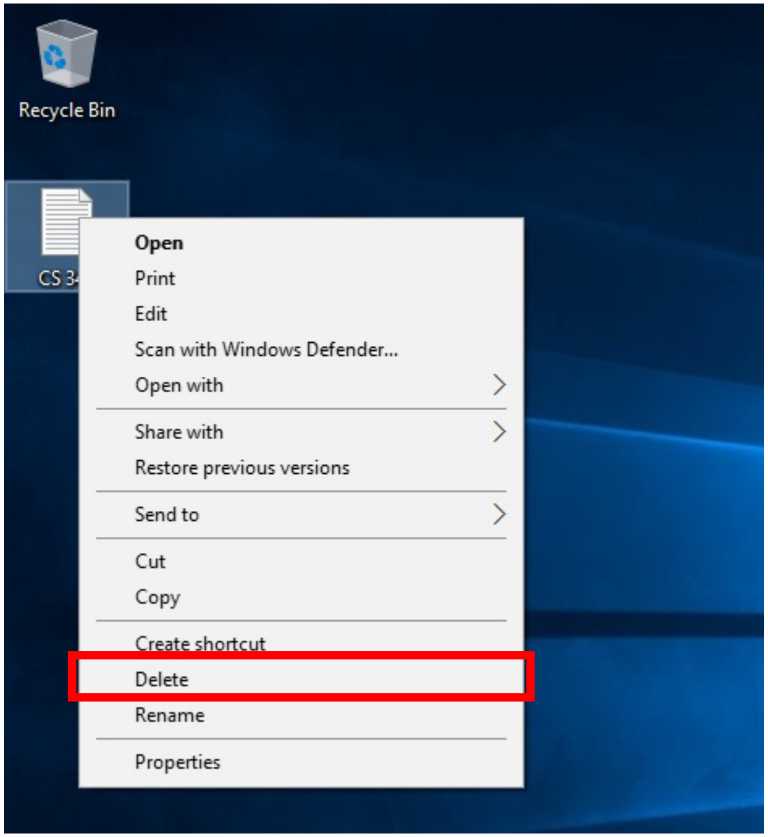
```
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\cs349>del ".\Desktop\CS 349.txt"_
```

Indirect Manipulation – Other Examples



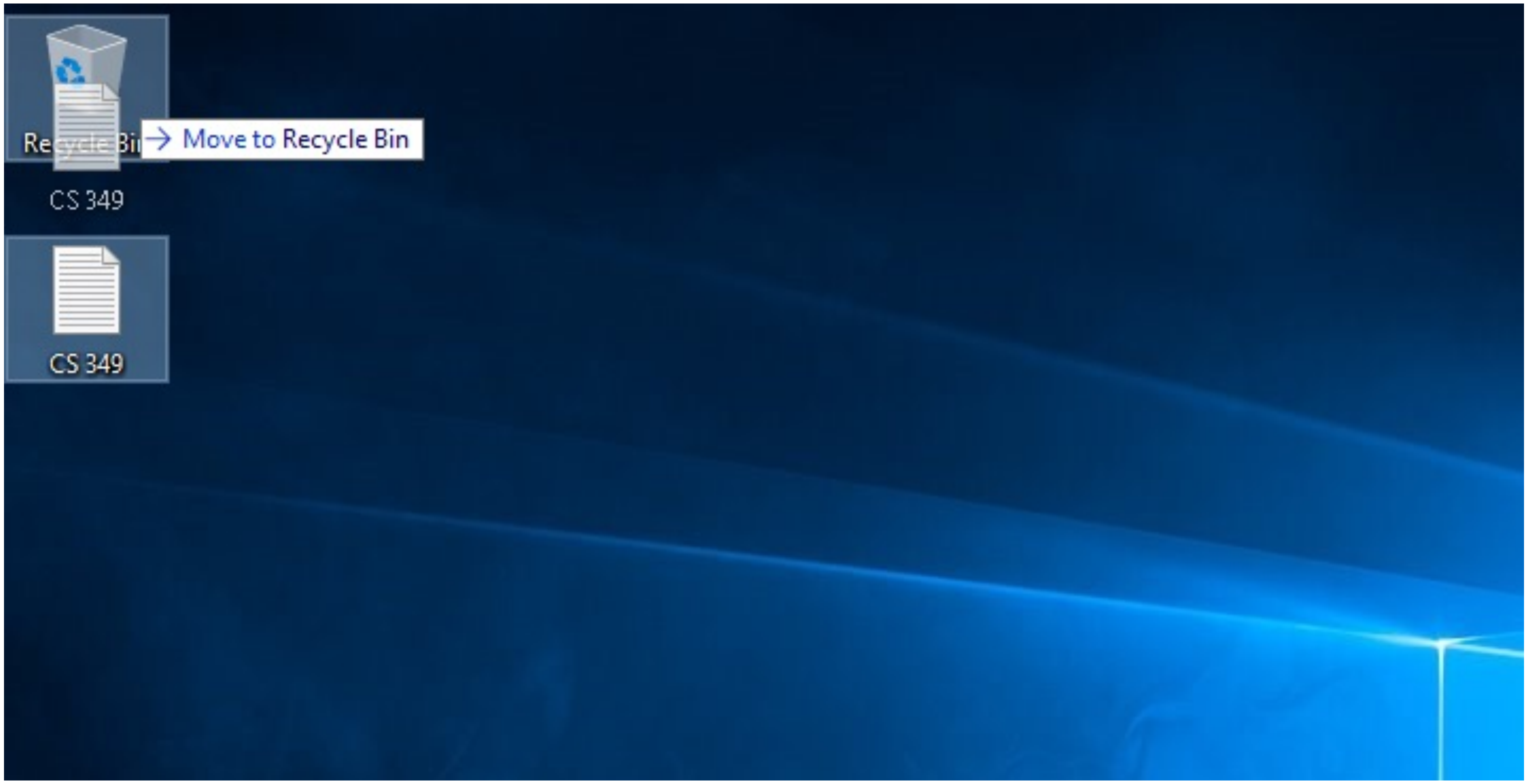
Are you sure you want to permanently delete this file?

CS 349
Type: Text Document
Size: 0 bytes
Date modified: 9/10/2022 5:15 PM



Direct Manipulation

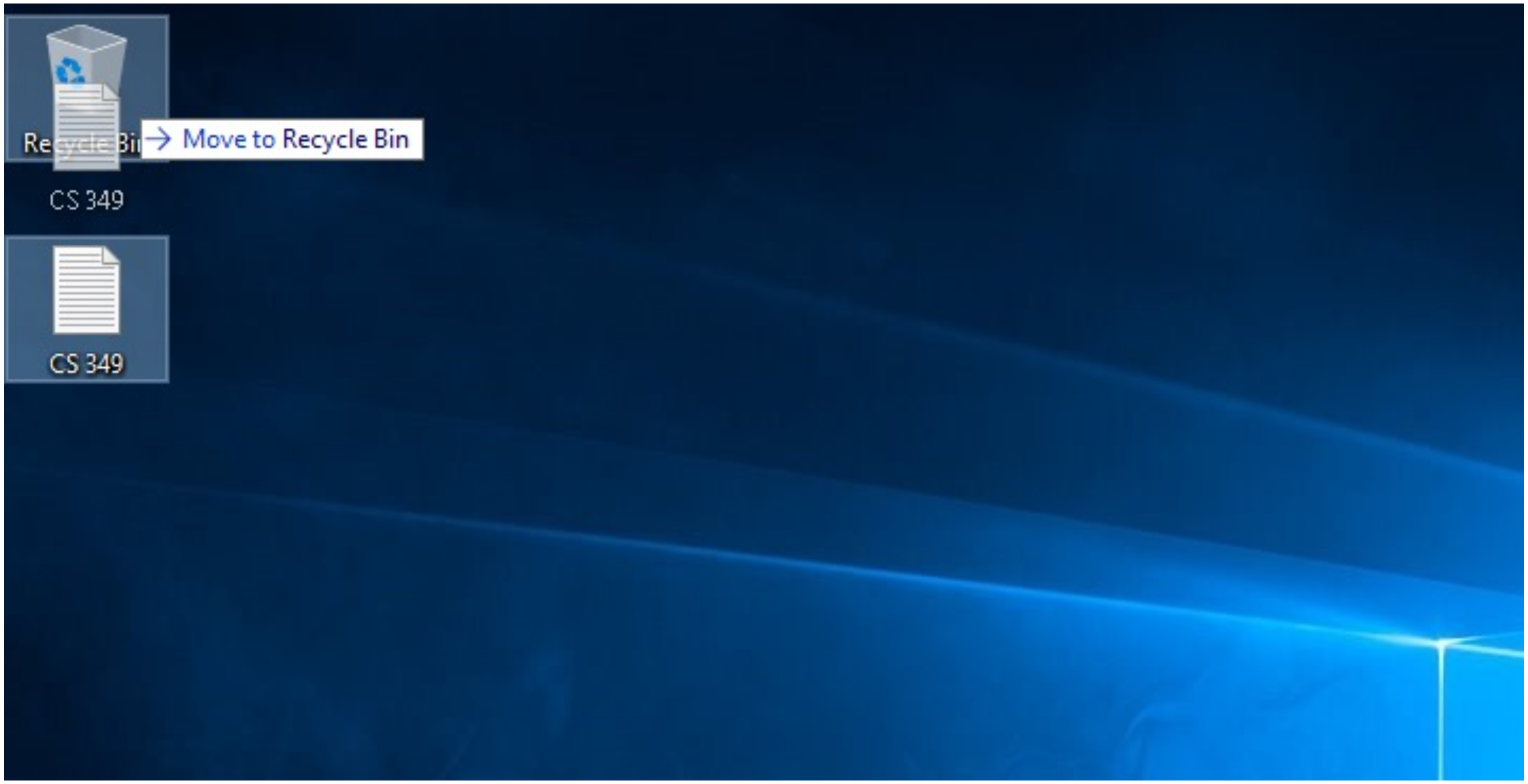
- Continuous representation of the object of interest
- Physical actions instead of complex syntax
- Continuous feedback and reversible, incremental actions
- Rapid, self-revealing approach to learning



Direct Manipulation

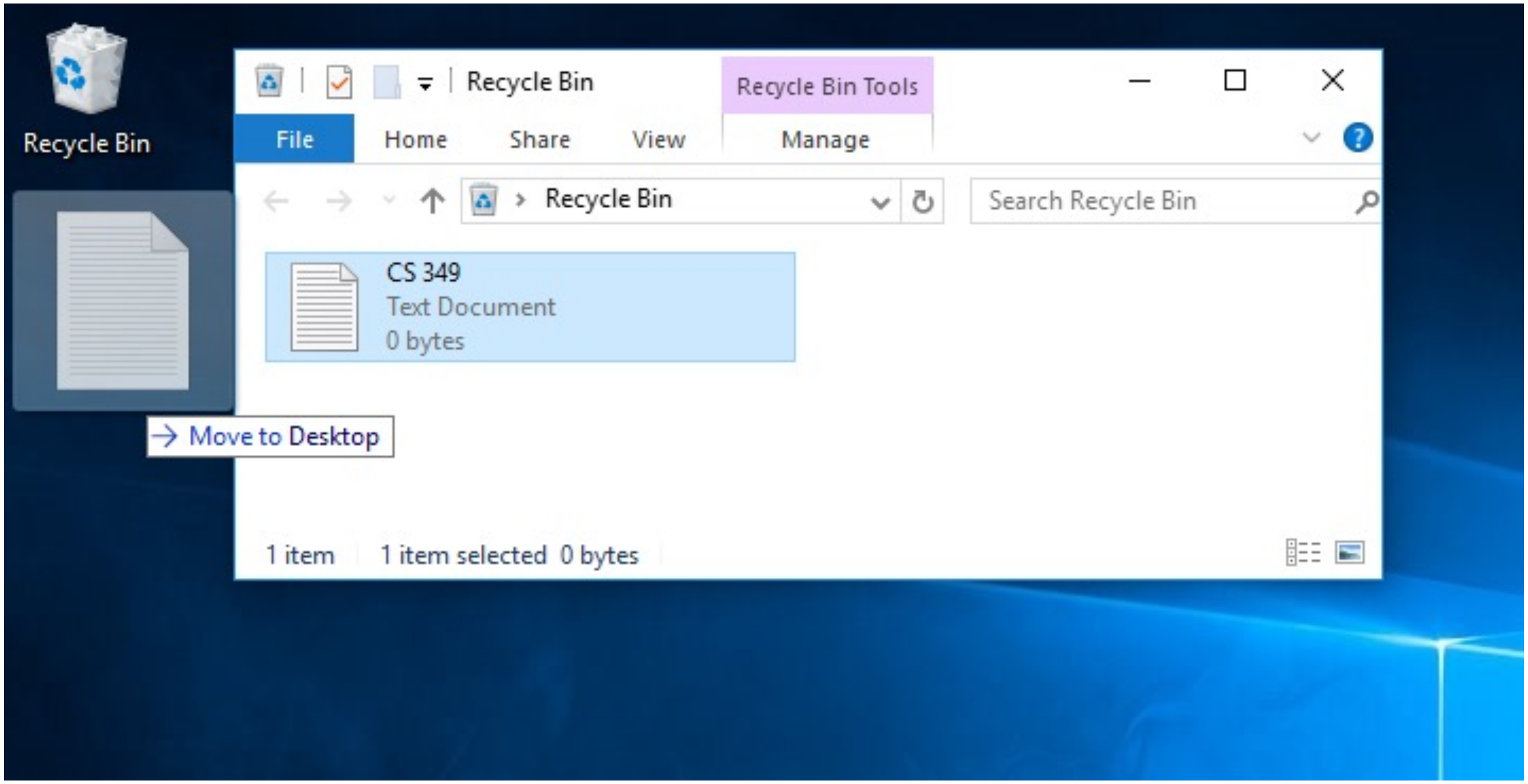
Direct manipulation is when a virtual representation of data (object of interest) is manipulated in a similar way to a real-world object.

Direct manipulation is meant to make the interaction feel as if the user was manipulating a real-world object instead of working through an intermediary.



Direct Manipulation

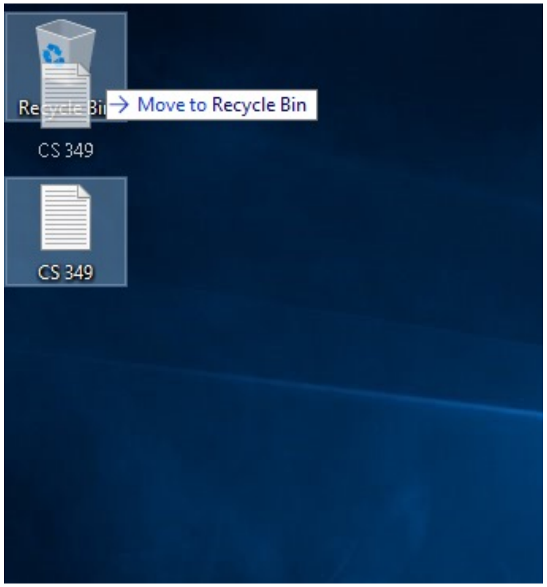
- Continuous representation of the object of interest
- Physical actions instead of complex syntax
- Continuous feedback and reversible, incremental actions
- Rapid, self-revealing approach to learning

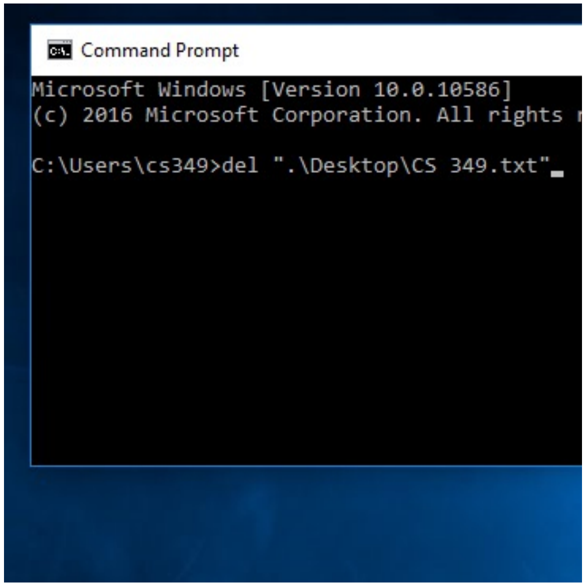


Direct Manipulation – Analogy to the Real World

While interacting with direct manipulation interfaces, users feel as if they are interacting with the domain rather than with the interface, so they focus on the task rather than on the technology.

There is a feeling of direct involvement with a world of objects of interest rather than communication with an intermediary.





```
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights r

C:\Users\cs349>del ".\Desktop\CS 349.txt"_
```

Direct Manipulation – Analogy to the Real World

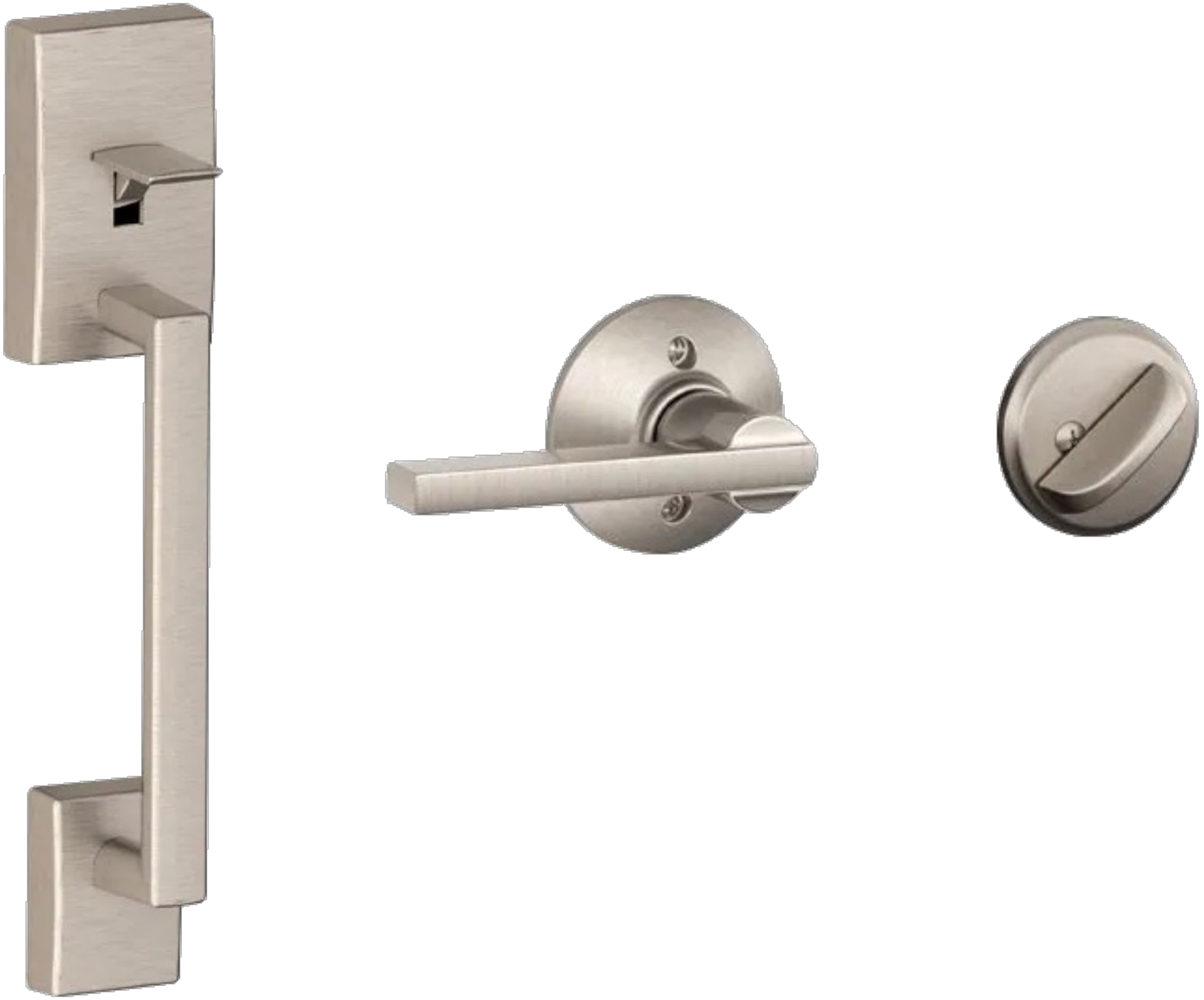
Real World Interaction:

- Object to be discarded
  - Move hand to object
  - Pick up object with hand
- Waste basket
  - Move hand with object to waste basket
  - Release object from hand

DM Interface:

- Icon of object of interest (OOI) to be discarded
  - Move pointer to OOI
  - Click button to select OOI
- Waste basket icon
  - Drag icon of OOI to waste basket icon
  - Release button to deselect OOI

Direct Manipulation – Affordance



Direct Manipulation – Affordance

Perceived affordance: what the user believes that they can do with an object, based on its appearance.

Actions that are suggested by the object.

Affordances in the interface are like affordances for analogous actions in the real world: they should build on existing experiences and intuitions to aid learning.

Direct Manipulation – Adapted to the Desktop

How does this apply to WIMP interfaces?

- Create objects of interests and supporting tools – *more in a few slides*
- Add visible cues that suggest actions to users e.g. window corner
- Allow for physical actions where possible in the interface.

Where do we see Direct Manipulation?

- Working surface / Desktop
- Grab and move / Click and drag to move elements (e.g., icons, files)
- Grab to engage / Click to activate (e.g., buttons, toolbars, other objects)

Desktop metaphor is prevalent/dominant

- It does not consistently use DM / there is a degree of “DM-ness”

BumpTop - A Multi Touch 3D Physics Desktop

https://github.com/bumptop/BumpTop/wiki

https://www.youtube.com/watch?v=eqcmPJ-oVL0

Direct Manipulation – Adaption to the Desktop

Modern GUIs do not always use direct manipulation (Desktop interfaces, frankly, do a poor job of modeling DM):

- Due to the complexity / possibilities of the digital world, many objects of interest have properties that cannot be easily represented using DM

- Many commands are invoked indirectly
  - Menus, dialog boxes, toolbars are not direct manipulation … they are “tools” that pull users away from the target object

- Many objects in the interface are not manipulable
  - Toolbar palettes

Direct Manipulation – Adaption to the Desktop

Are there times to not use Direct Manipulation?

There are also very good reasons to deliberately break away from DM:

- Visually impaired users cannot see the graphics; no linear flow for screen readers; physically impaired may have difficulty with required movements
- Consumes valuable screen space, forcing valuable information off-screen
- Switching between keyboard and pointer is time consuming
- Analogies may not be clear
  - Users need to learn meaning of visual representations
  - Visual representations may be misleading

Instrumental Interaction

Interaction Model

"An interaction model is a set of principles, rules, and properties that guide the design of an interface. It describes **how to combine interaction techniques in a meaningful and consistent way** and defines the look and feel of the interaction from the user's perspective. Properties of the interaction model can be used to evaluate specific interaction designs."

— Michel Beaudouin-Lafon. 2000. *Instrumental interaction: an interaction model for designing post-WIMP user interfaces*. In *Proc of the CHI '00*, 446–453. DOI: 10.1145/332040.332473

Instrumental Interaction

"The instrumental interaction model is based on how we naturally use tools (or instruments) to manipulate objects of in the physical world. Objects of interest are called *domain objects* and are manipulated with computer artifacts called *interaction instruments*."

— Michel Beaudouin-Lafon. 2000. *Instrumental interaction: an interaction model for designing post-WIMP user interfaces*. In *Proc of the CHI '00*, 446–453. DOI: 10.1145/332040.332473

With instrumental interaction, interfaces have *domain objects* and *interaction instruments*

- **Domain objects**: the thing of interest, data and associated attributes, which is manipulated using an interaction instrument
- **Interaction instrument**: a necessary mediator between the user and domain objects

Instrumental Interaction

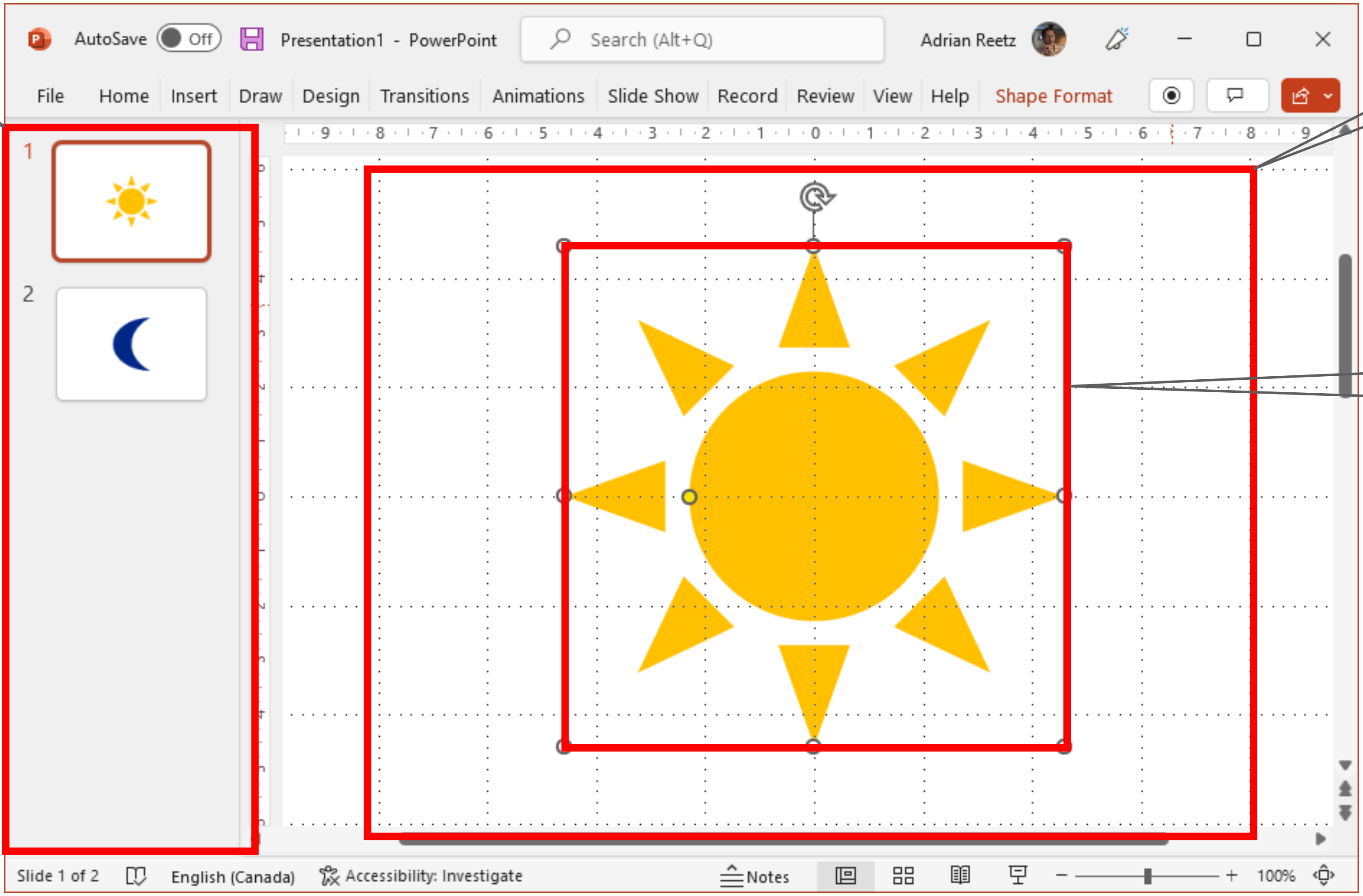
With instrumental interaction, interfaces have *domain objects* and *interaction instruments*

Goal:

- **Domain objects**: the thing of interest, data and associated attributes, which is manipulated using an interaction instrument

- **Interaction instrument**: a necessary mediator between the user and domain objects

Instrumental Interaction – Domain Objects

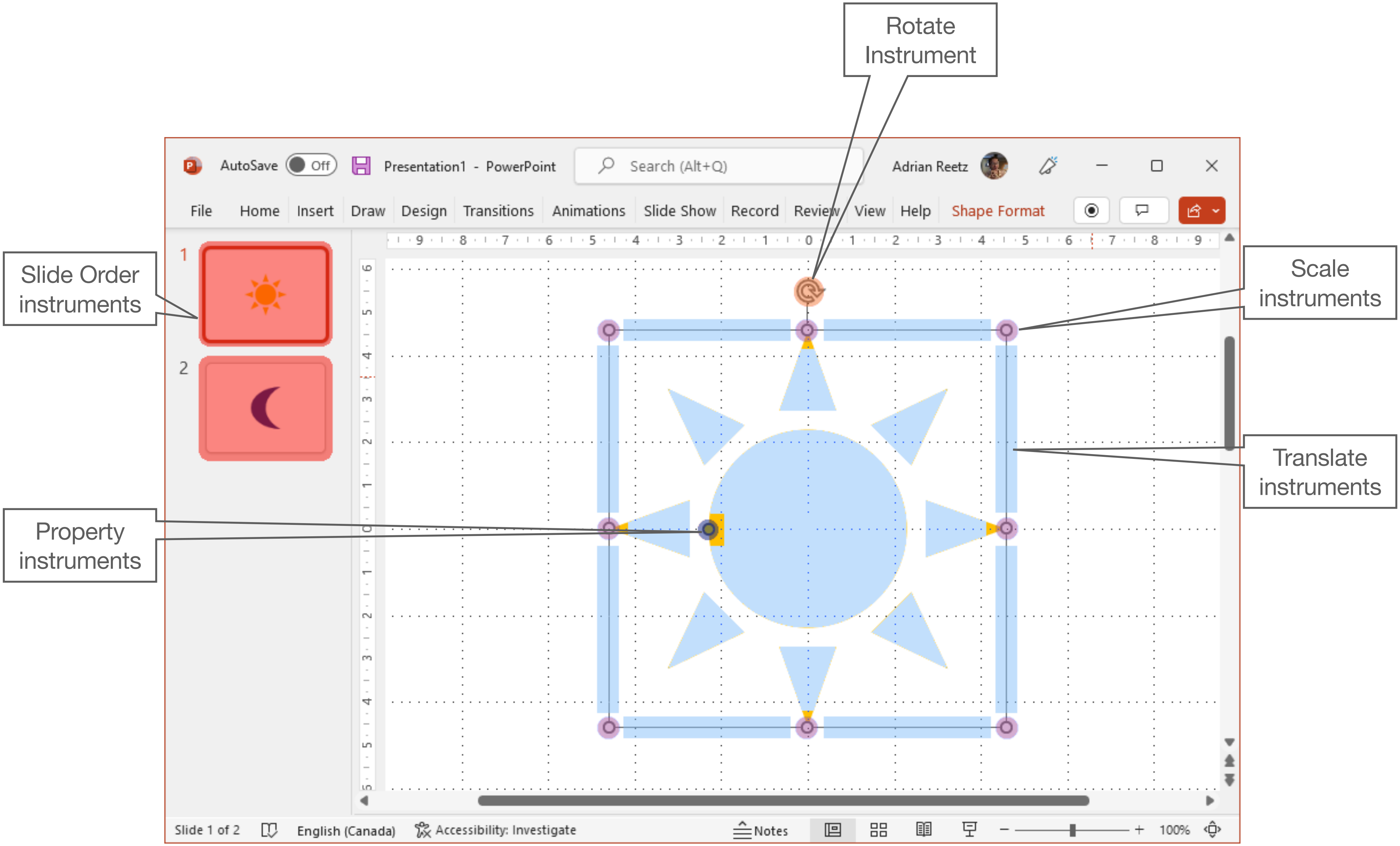


Presentation Object

Slide Object

Drawing Object

Instrumental Interaction – Interaction Instruments



Rotate Instrument

Slide Order instruments

Scale instruments

Translate instruments

Property instruments

Instrumental Interaction – Activation

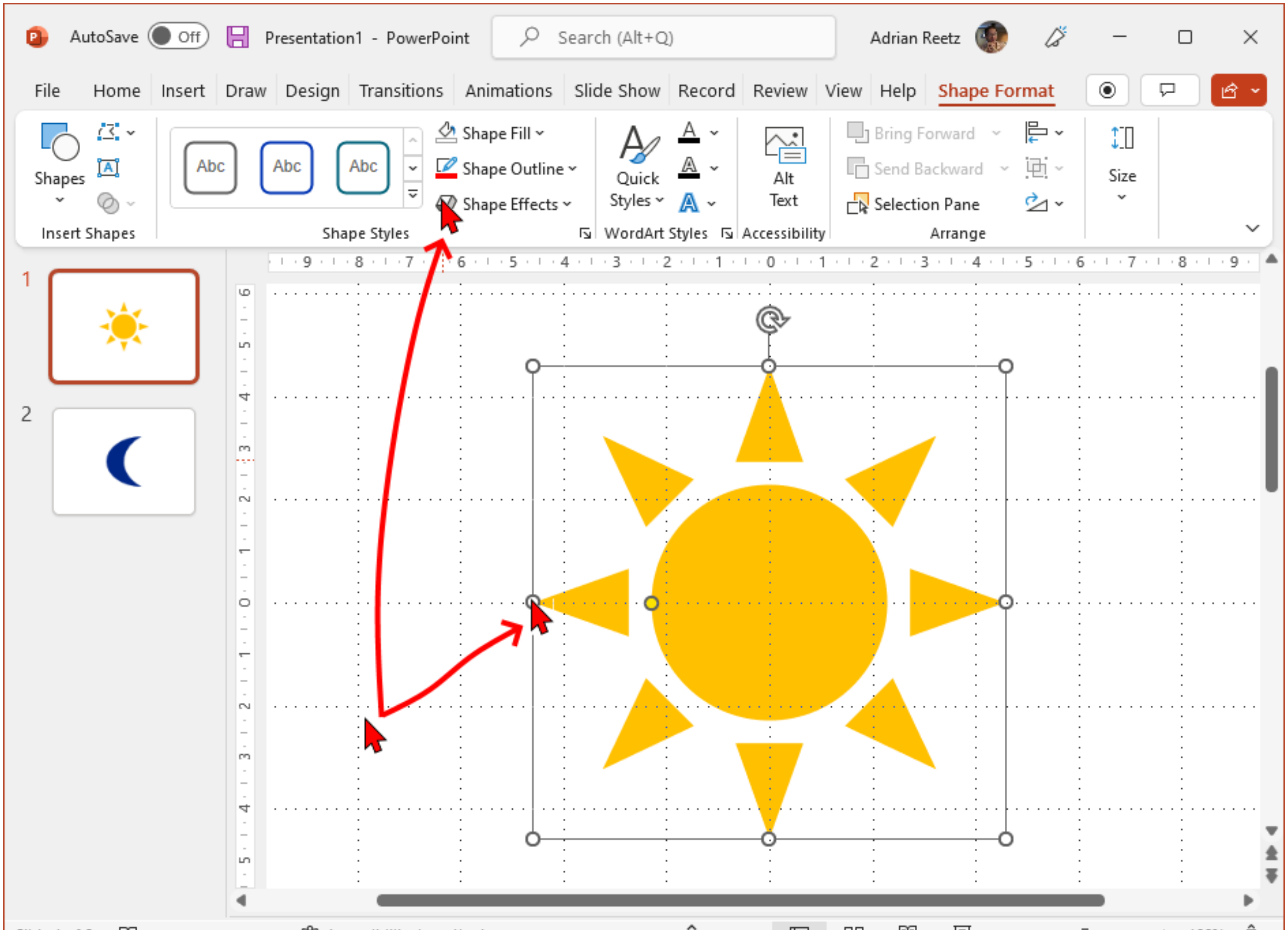
Activation = triggering functionality

Interaction instruments can be activated **spatially** and **temporally**

- Spatial activation has a **movement cost**
- Temporal activation has a **time cost**

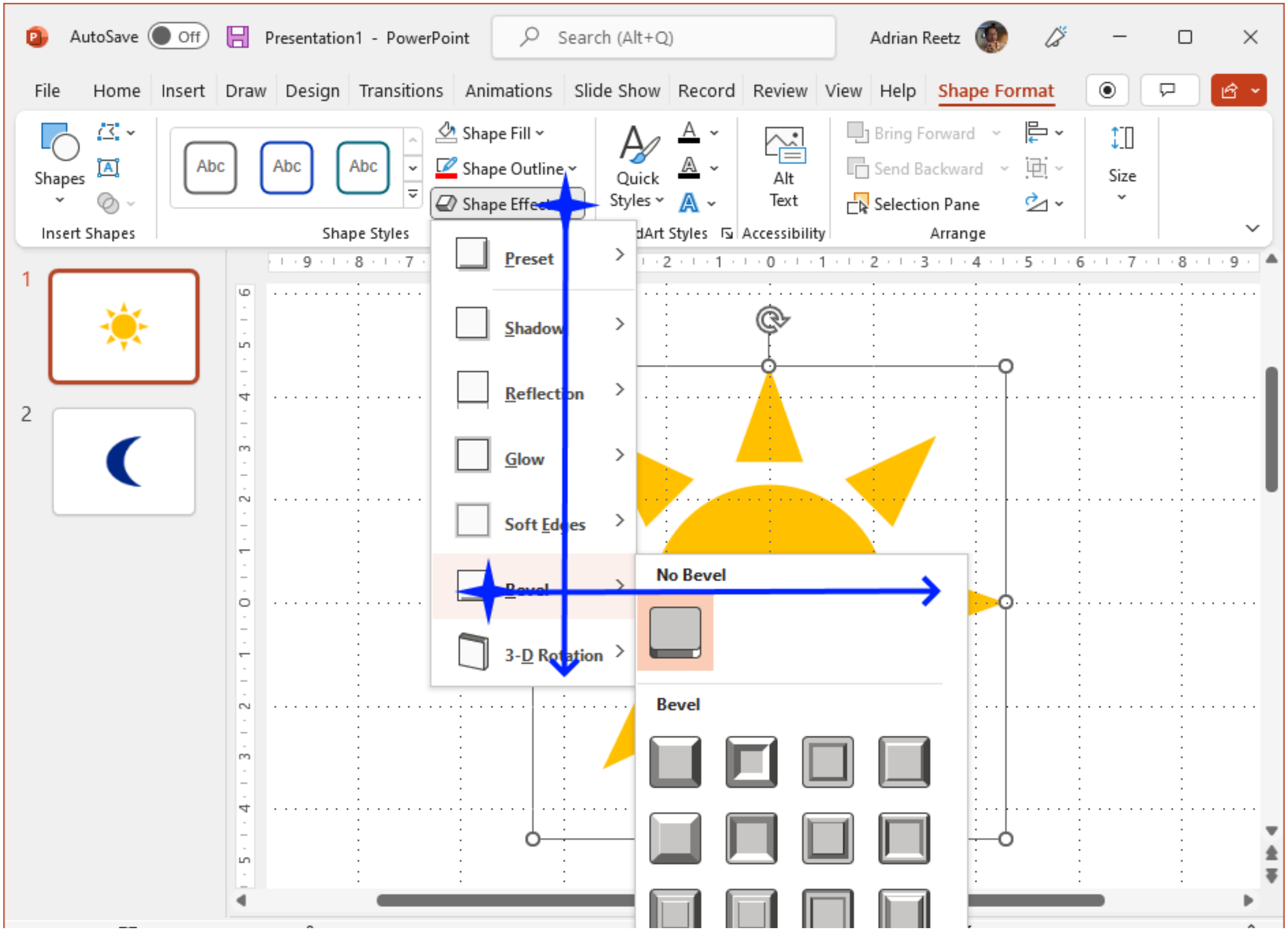
Instrumental Interaction – Activation

Spatial activation has a **movement cost**.



Instrumental Interaction – Activation

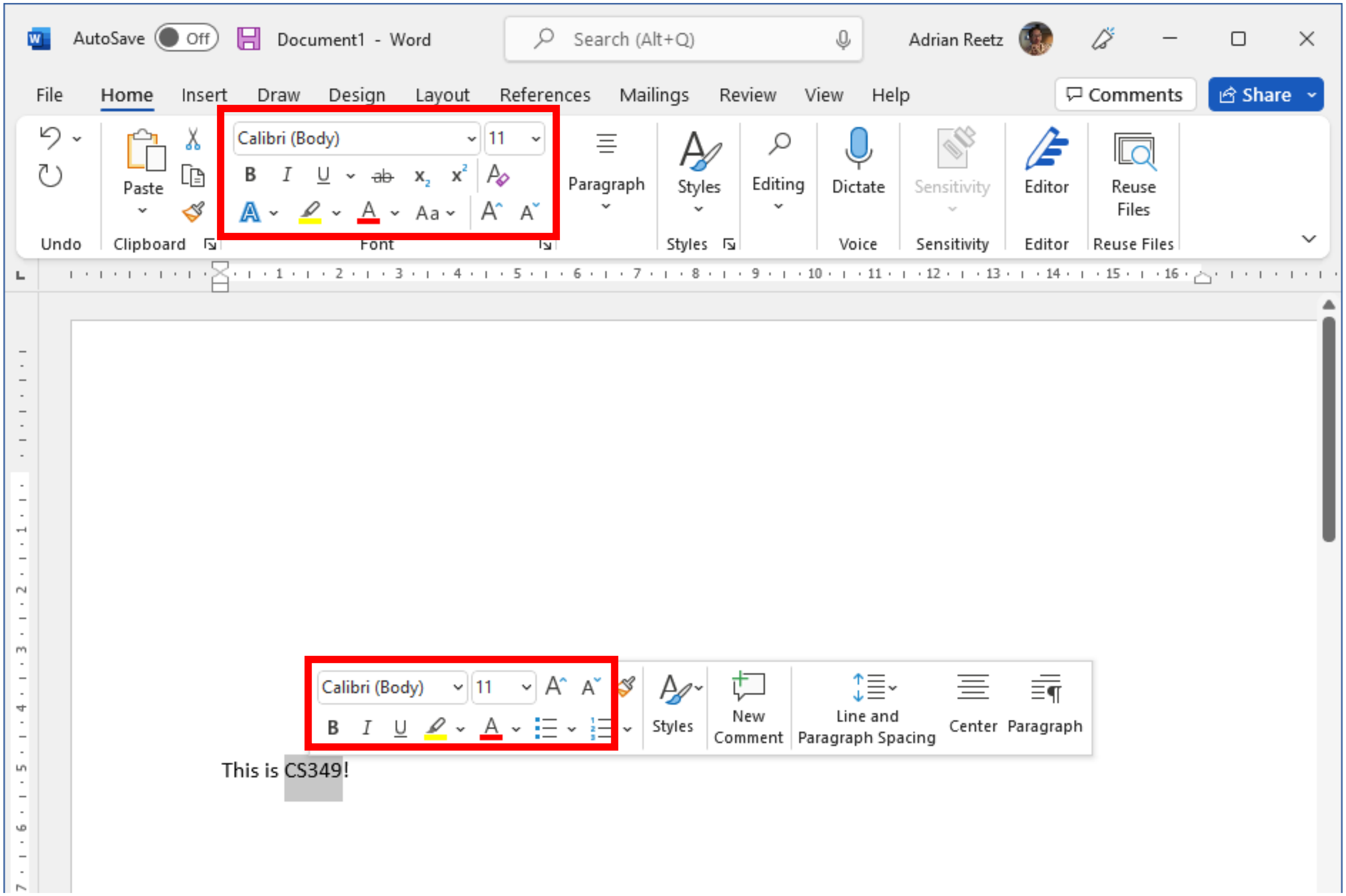
Temporal activation has a **time cost**.



Instrumental Interaction – Activation

UI layout and design is concerned with the tradeoff of these costs.

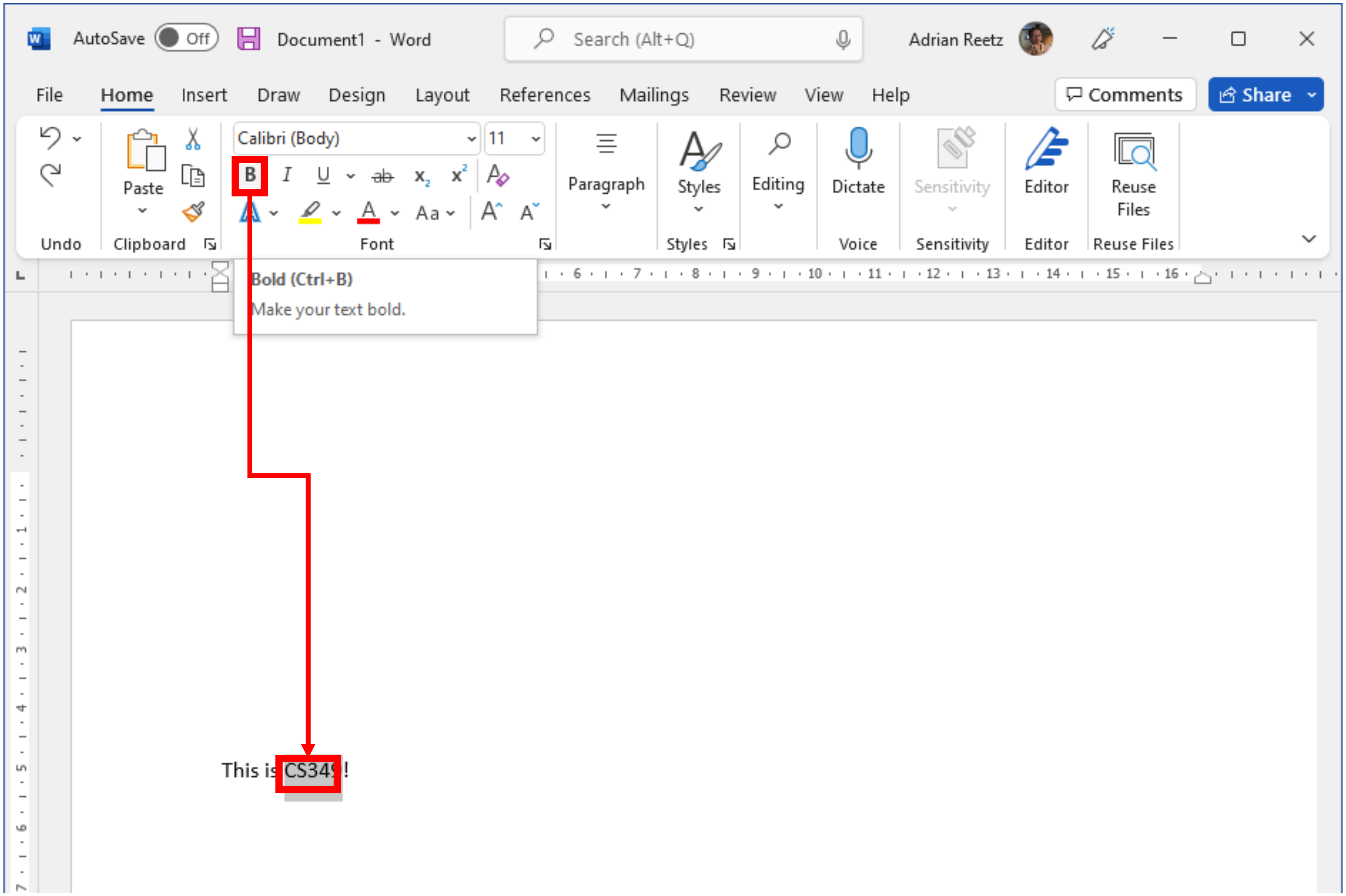
The more frequently an instrument is used, the lower its activation cost should be.



This is CS349!

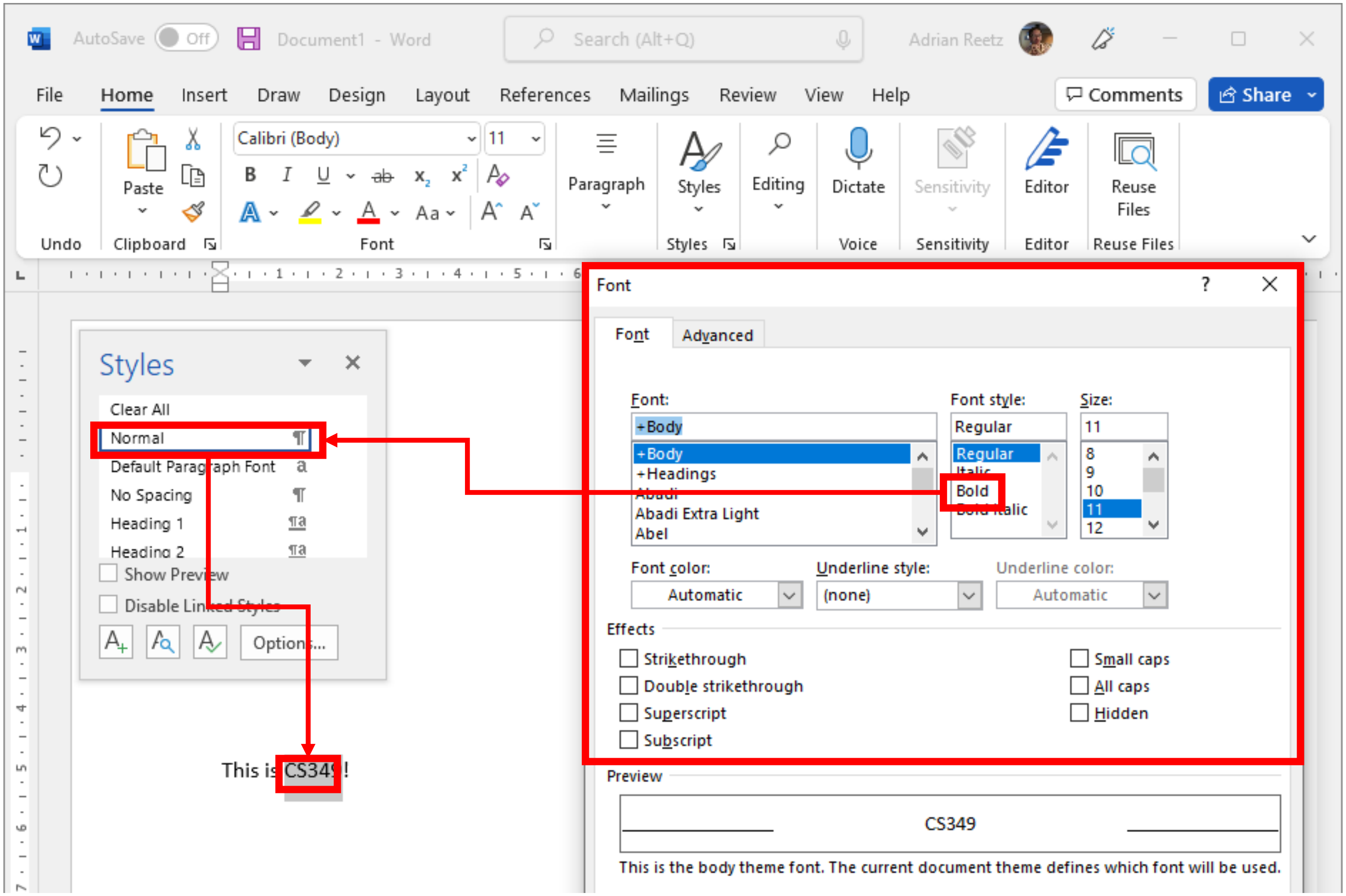
Instrumental Interaction – Reification

Below, a property (font weight) of a domain object (“CS349”) is changed (using the interaction instrument “[B]old” button).



Instrumental Interaction – Reification

Below, the property of an interaction instrument (Style “Normal”) is changed (using the interaction instrument “Font” window).



This is CS349!

Font

Font | Advanced

Font: +Body

+Body
+Headings
Abadi
Abadi Extra Light
Abel

Font style: Regular

Regular
Italic
Bold
Bold Italic

Size: 11

8
9
10
11
12

Font color: Automatic

Underline style: (none)

Underline color: Automatic

Effects

□ Strikethrough
□ Double strikethrough
□ Superscript
□ Subscript
□ Small caps
□ All caps
□ Hidden

Preview

CS349

This is the body theme font. The current document theme defines which font will be used.

Instrumental Interaction – Reification

Reification: turning concepts into something concrete.

For example, the single concept of “text style” is a reification of multiple concepts, such as, font type, font weight, font size, alignment, baseline spacing, letter spacing, word spacing, kerning, shifting, direction, orientation, …

This reification can become a domain object in itself!

Domain object “CS349” ↦ interaction instrument “Style”,

Domain object “Style ‘Normal’” ↦ interaction instrument “’Font style’ selection box”.

Instrumental Interaction – Meta-Instruments

Meta-instrument: an instrument that acts on reified domain object.

Domain object “CS349” ↦ interaction instrument “Style”,

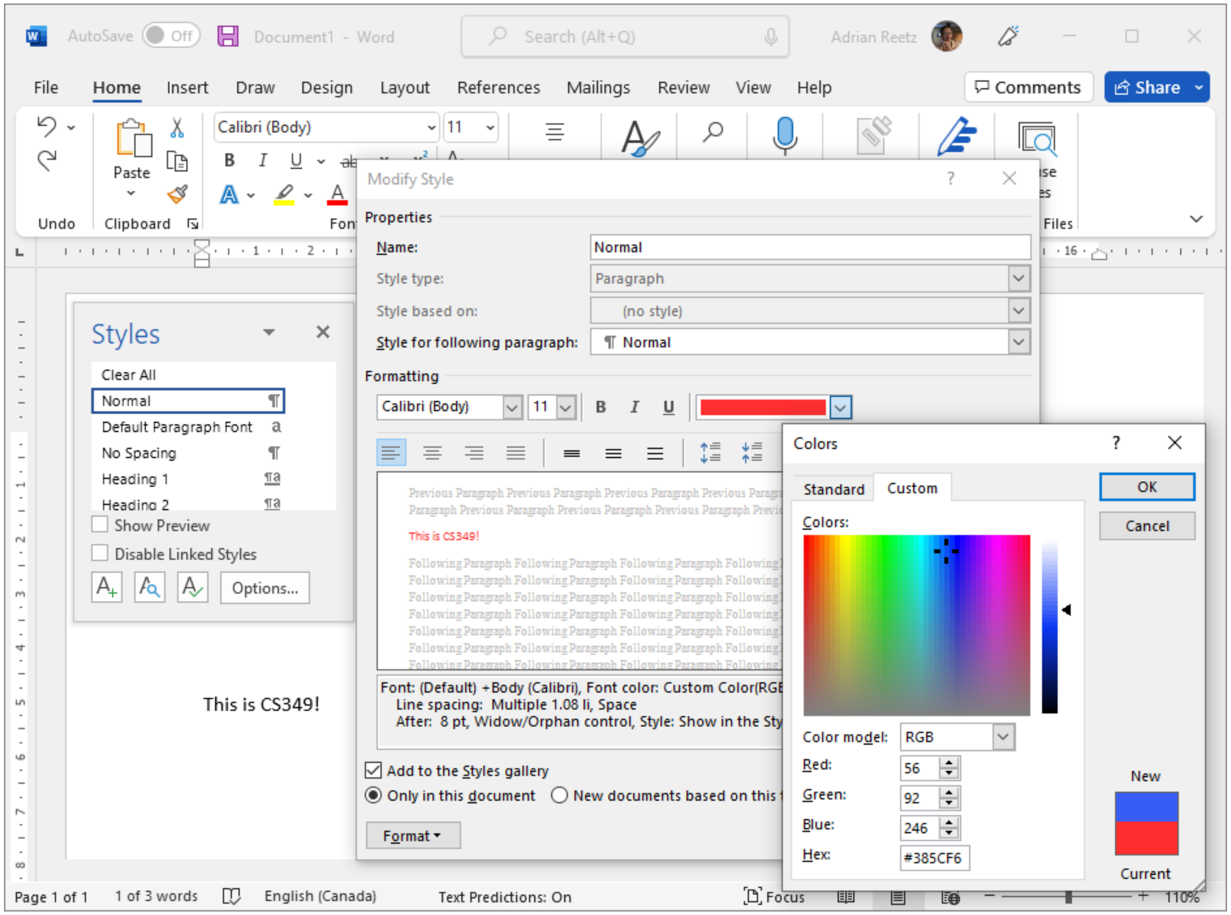
Domain object “Style ‘Normal’” ↦ **meta** interaction instrument “’Font style’ selection box”.

Instrumental Interaction – Meta-Instruments

Domain object “CS349” ↦ interaction instrument “Style”,

Domain object “Style ‘Normal’” ↦ **meta** interaction instrument “’Color’ selection box”.

Domain object “Color ‘Red’” ↦ **meta** interaction instrument “Color picker”



Modify Style

Properties

Name: Normal

Style type: Paragraph

Style based on: (no style)

Style for following paragraph: ¶ Normal

Formatting

Calibri (Body) 11 B I U 🟥

This is CS349!

Font: (Default) +Body (Calibri), Font color: Custom Color(RGE
Line spacing: Multiple 1.08 li, Space
After: 8 pt, Widow/Orphan control, Style: Show in the Sty

☑ Add to the Styles gallery

● Only in this document ○ New documents based on this t

Colors

Color model: RGB

Red: 56

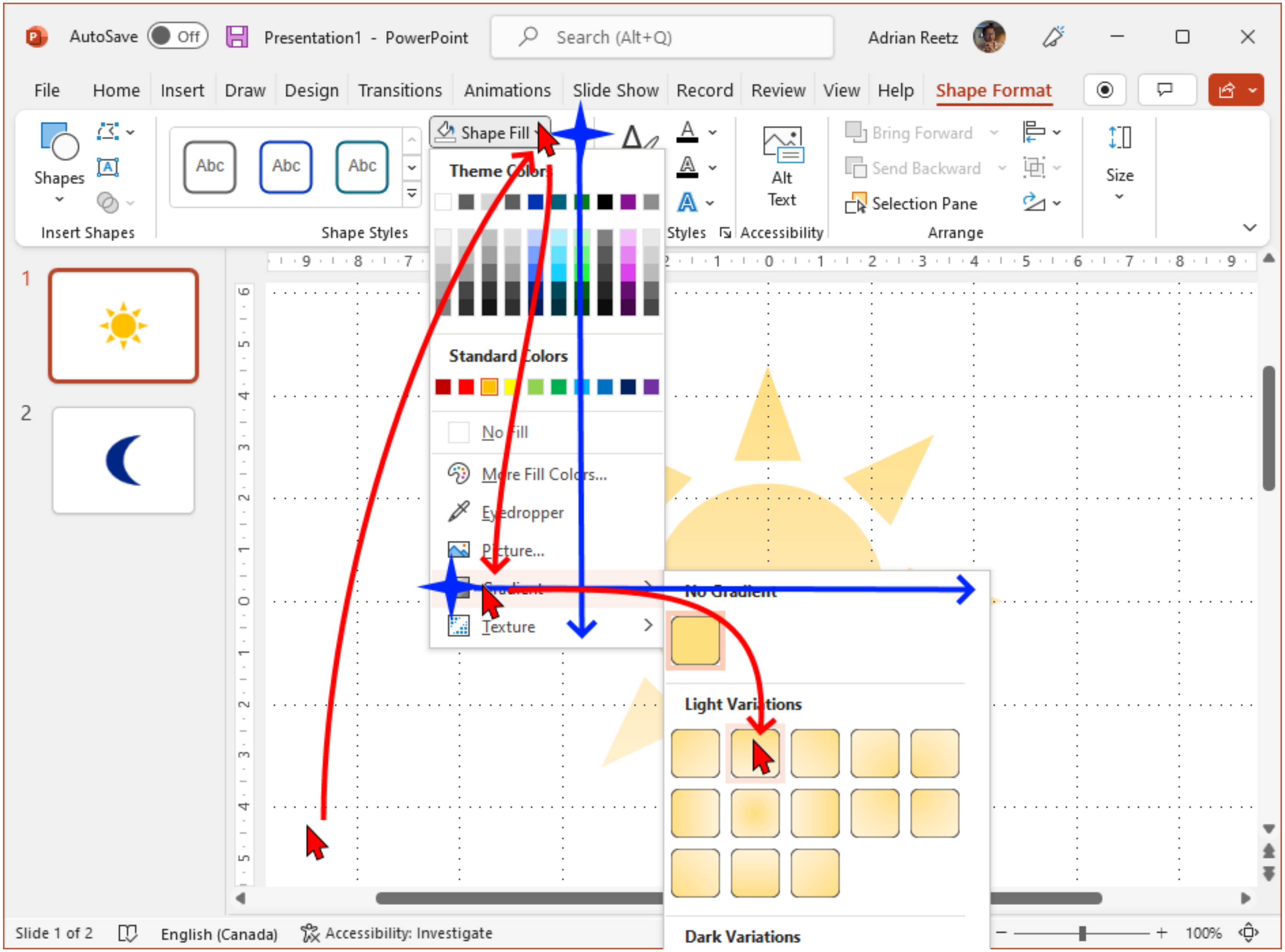
Green: 92

Blue: 246

Hex: #385CF6

This is CS349!

Instrumental Interaction – Evaluation?



Instrumental Interaction – Evaluation

How do we describe instruments?

How do we evaluate their effectiveness?

Degree of Indirection

- Spatial / temporal offset between instrument and action on object

Degree of Integration

- Suitability of input device for manipulating instrument
- (Ratio of degrees of freedom of instrument to degrees of freedom of input device)

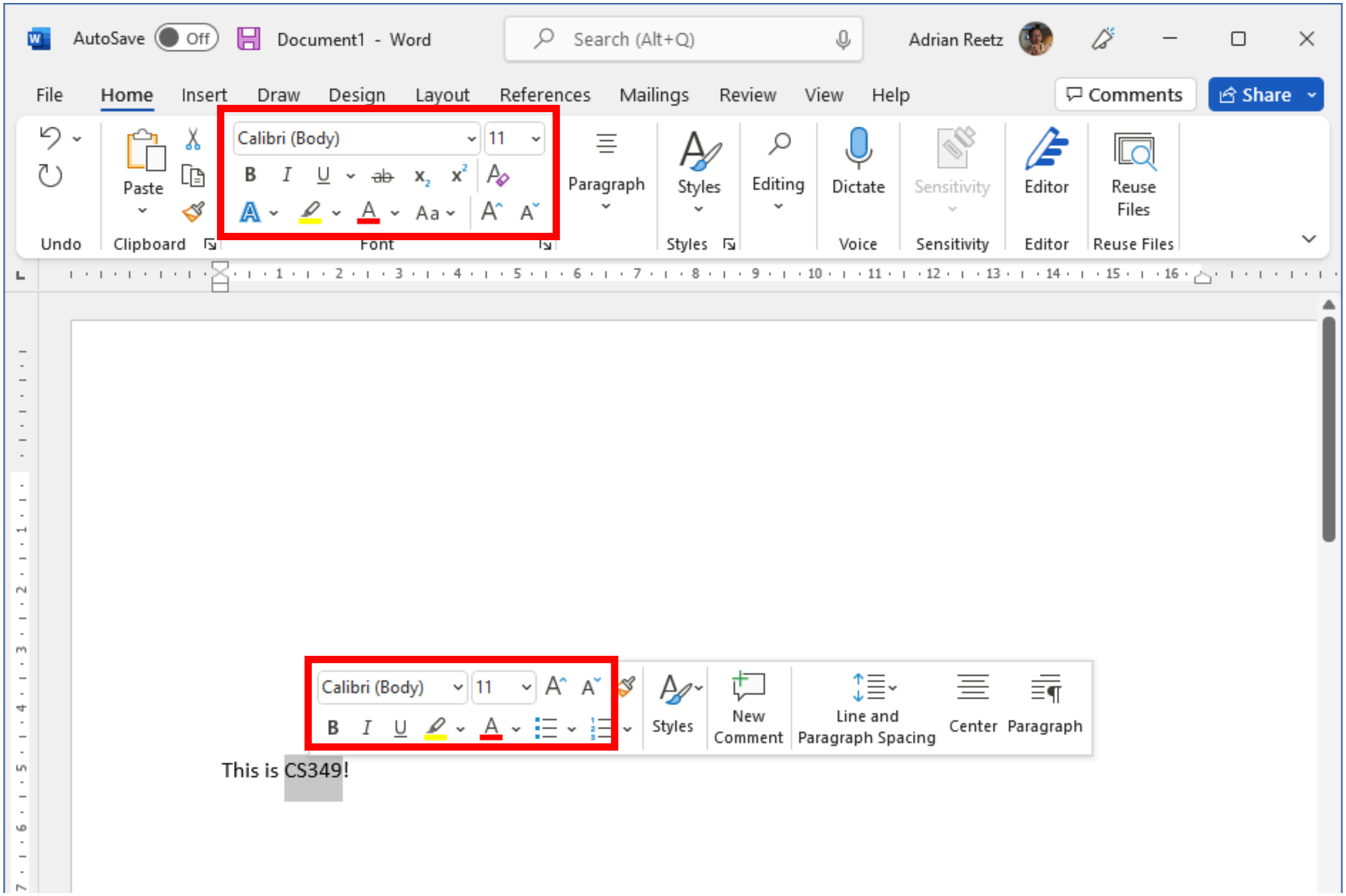
Degree of Compatibility

- Similarity of action on control device/instrument to action on object

Instrumental Interaction – Degree of Indirection

Near, e.g., drag to translate, handles on to resize

Far, e.g., menu bar, scroll bar

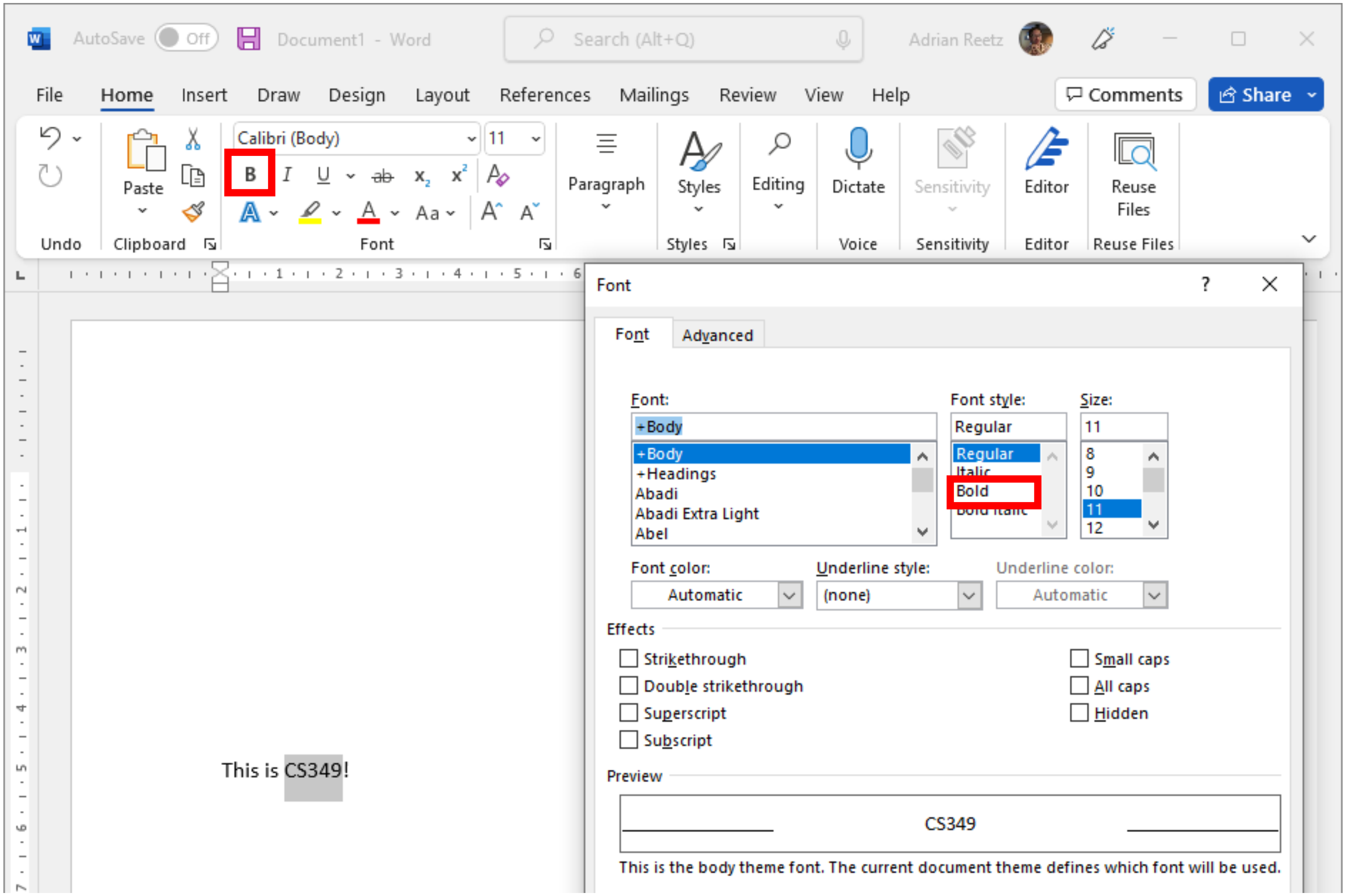


This is CS349!

Instrumental Interaction – Degree of Indirection

Short, e.g., directly accessible buttons; less information to process

Long, e.g., menu activation, animation; more information to process



This is CS349!

Font

Font | Advanced

Font: +Body

+Body
+Headings
Abadi
Abadi Extra Light
Abel

Font style: Regular

Regular
Italic
Bold
Bold Italic

Size: 11

8
9
10
11
12

Font color: Automatic

Underline style: (none)

Underline color: Automatic

Effects

☐ Strikethrough
☐ Double strikethrough
☐ Superscript
☐ Subscript
☐ Small caps
☐ All caps
☐ Hidden

Preview

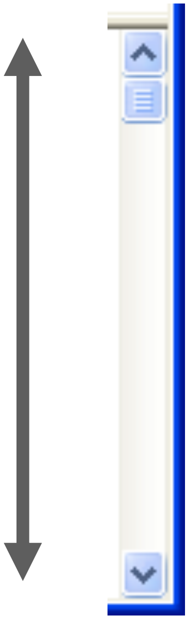
CS349

This is the body theme font. The current document theme defines which font will be used.

Instrumental Interaction – Degree of Integration

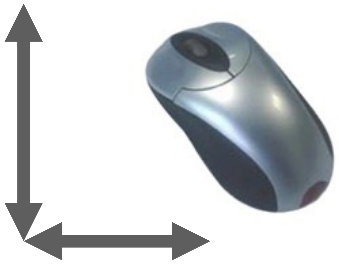
Degree of integration is the ratio between the number of degrees of freedom (DOF) of the instrument and the DOF captured by input device (reflects suitability)

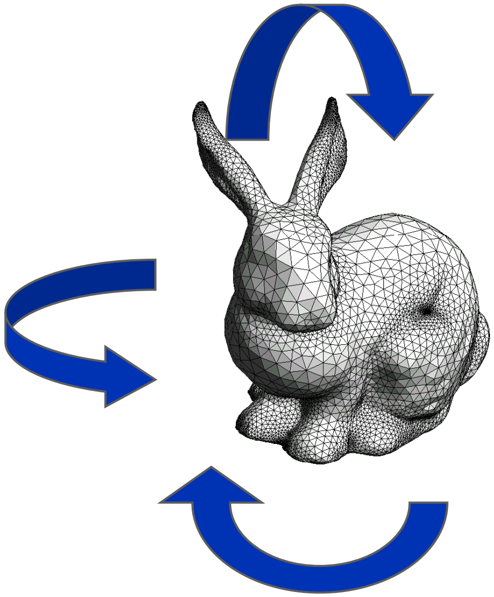
1 DOF



1/2 = 0.5

2 DOF

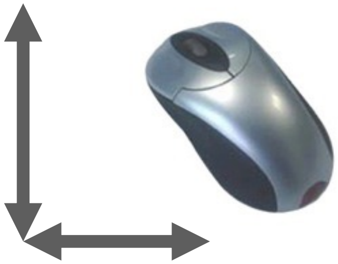




3 DOF

3/2 = 1.5

2 DOF



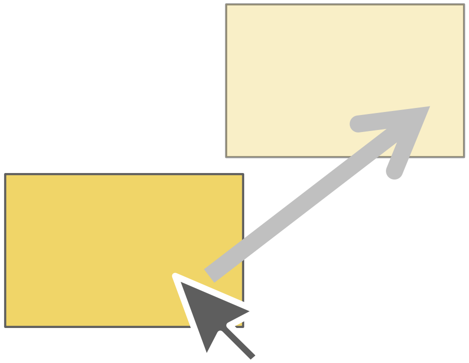
Instrumental Interaction – Degree of Compatibility

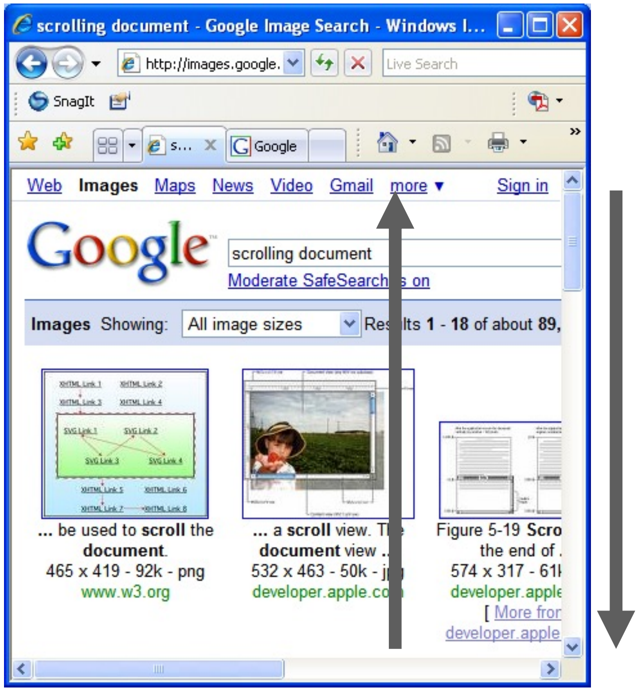
Degree of Compatibility is the similarity between the physical action on the instrument and the response of the object (similarity makes actions feel natural or intuitive).

Dragging = high

Scrolling = medium

Dialog = low





http://images.google.

scrolling document

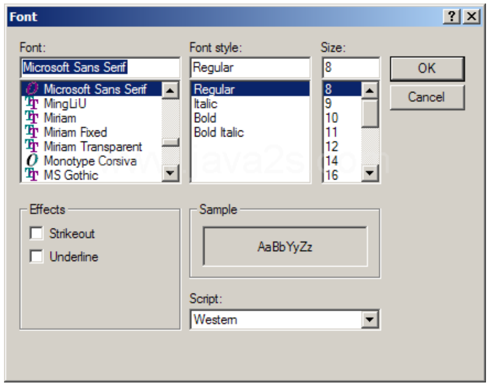
Moderate SafeSearch is on

Images Showing: All image sizes Results 1 - 18 of about 89,

... be used to **scroll** the **document**.
465 x 419 - 92k - png
www.w3.org

... a **scroll** view. The **document** view ...
532 x 463 - 50k - jpg
developer.apple.com

Figure 5-19 **Scro**... the end of ...
574 x 317 - 61k
developer.apple...
[More from developer.apple...



Font: Microsoft Sans Serif

- Microsoft Sans Serif
- MingLiU
- Miriam
- Miriam Fixed
- Miriam Transparent
- Monotype Corsiva
- MS Gothic

Font style: Regular

- Regular
- Italic
- Bold
- Bold Italic

Size: 8

- 8
- 9
- 10
- 11
- 12
- 14
- 16

□ Strikeout

□ Underline

Script: Western

Direct Manipulation & Instrumental Interaction

A direct manipulation interface allows a user to directly act on a set of objects in the interface.

- Low degree of indirection (i.e., low spatial and temporal offsets)
- High degree of integration (1:1 correspondence)
- High degree of compatibility (similarity of action and effect)

Direct means instruments are visually indistinguishable from objects they control

- The actions on instrument / object entities are analogous to actions on similar objects in the real world.
- The actions on instrument/object entities preserve the conceptual linkage between instrument and object.

Bret Victor, Inventing on Principle (talk from CUSEC 2012)

https://www.student.cs.uwaterloo.ca/~cs349/videos/Bret Victor - Inventing on Principle-HD.mp4

End of the Chapter

- Direct Manipulation
  - what it is
  - its components
  - its advantages and disadvantages
- Instrumental Interaction
  - what it is
  - its components
  - its advantages and disadvantages