

User Interface Design & Implementation

Direct Manipulation

Week 3 – Lecture 6

January – May Term, 2020

Assigned Reading: Chapter 7

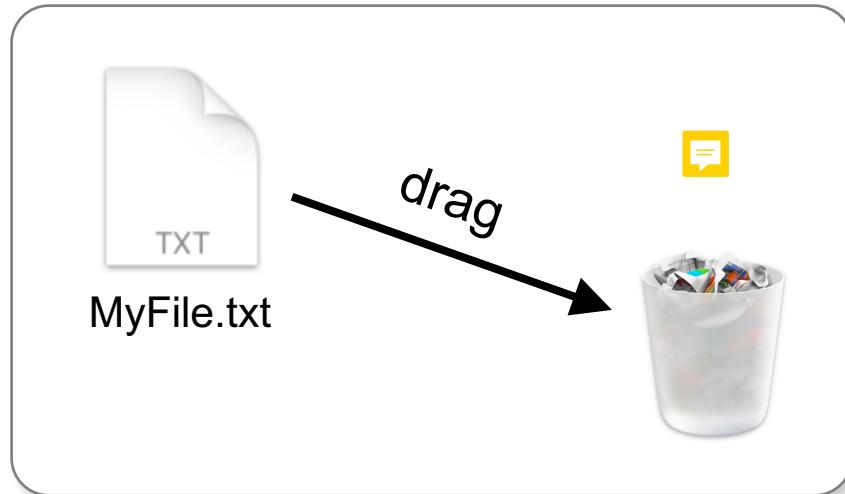
Today's Topics

- Interaction style
- What is direct manipulation?
- Object-action interface model
- Direct manipulation with multi-touch
- Examples of direct manipulation
- Discussion of direct manipulation

Interaction Style

Interaction style refers to all the ways the user can communicate or interact with the computer system

- **Recognition:** remembering with the help of a visible cue
- **Recall:** remembering with no help
- Recognition is much easier!



Interaction Style

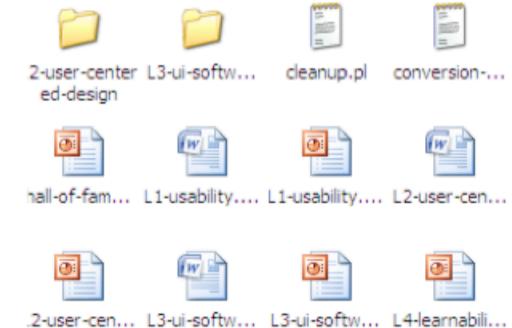
Five main types of interaction style

1. Direct Manipulation
2. Menu Selection
3. Form fill-in
4. Command Language
5. Natural Language

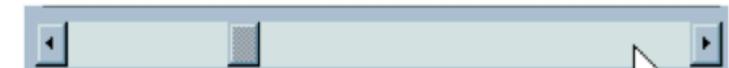


Interaction Style #1: Direct Manipulation

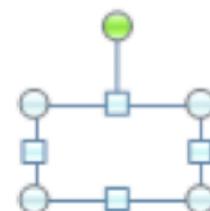
- Manipulate visual representations, e.g. Desktop metaphor, CAD, games
- Appealing to novices and easy to remember for intermittent users.
- **Pros:** fast, feedback, **easy to understand** and retain (ex. icons on your desktop), exploration encouraged, good for novices, and can be good for other classes, visual data
- **Cons:** hard to program, interaction devices are **harder to design** or modify



Files & folders on desktop



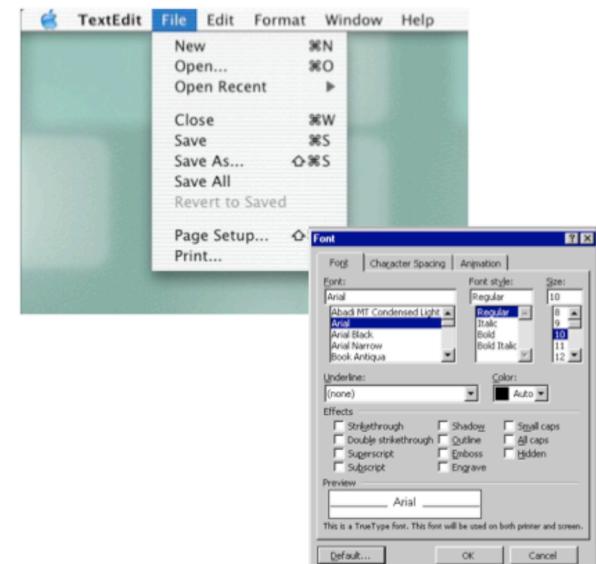
Scrollbar



Selection handles

Interaction Style #2: Menu Selection

- User reads a list of items, and selects one
- Appropriate for novice and intermittent users and can be appealing to frequent users if the display and selection mechanisms are rapid.
- **Pros:** no memorization, few actions, clear structure, tools for validity and consistency exist
- **Cons:** Make actions understandable not easy, careful task analysis



Interaction Style #3: Form Fill-in

- Data entry into fields
- Most appropriate for knowledgeable intermittent users or frequent users.
- **Pros:** rapid, tools available for forms
- **Cons:** must understand labels and request format, be able to respond to errors, training required

Required •

Email Address • Confirm Email Address •

Enter a Shipping Address

Address Nickname •
For example, Home or Work. A nickname will help you locate this address quickly in your list of addresses.

First Name • Last Name •

Care of / Company Name

Address Line 1 •

This is a P.O. Box or Military Address (APO or FPO)
[Help with APO/FPO](#)

Address Line 2

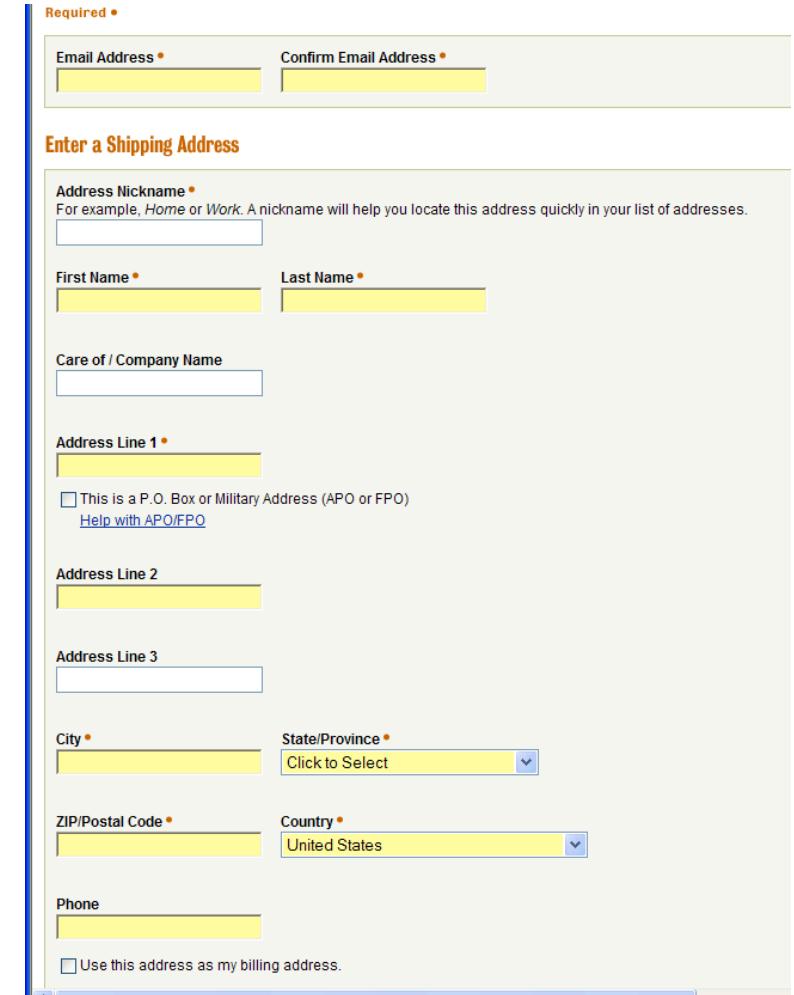
Address Line 3

City • State/Province • [Click to Select](#)

ZIP/Postal Code • Country • United States

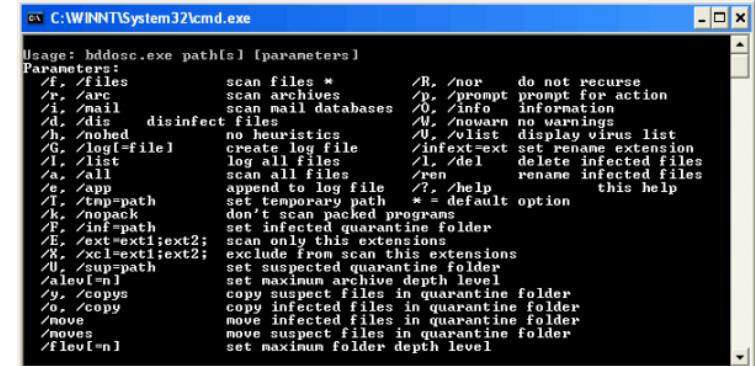
Phone

Use this address as my billing address.

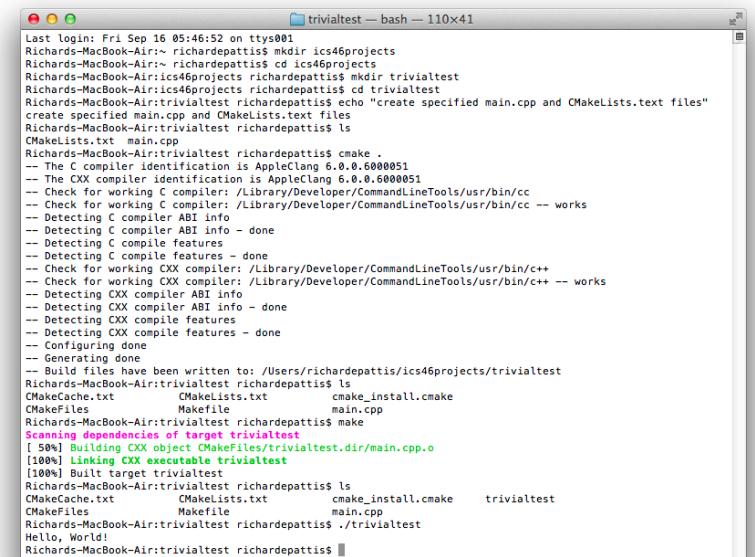


Interaction Style #4: Command Language

- Suitable for expert frequent users who derive great satisfaction from mastering a complex set of semantics and syntax.
- Pros: feeling of control, most advanced users like it, rapid, histories and macros are easy, flexibility
- Cons: high error rates, training required, poor retention rate, hard to create error messages



```
C:\WINNT\System32\cmd.exe
Usage: bddosc.exe path[!l] [parameters]
Parameters:
 /f, /files      scan files *
 /r, /arc        scan archives
 /i, /mail       scan mail databases
 /d, /dis         disinfect files
 /h, /noheur     no heuristics
 /C, /CFile[=file] create log file
 /L, /list        log all files
 /a, /all         scan all files
 /e, /app         append to log file
 /T, /Tmp=Path   set temporary path
 /k, /nopack     don't scan packed programs
 /F, /Inf=Path   set infected quarantine folder
 /E, /Ext=Ext1;Ext2; exclude from scan this extensions
 /X, /Xc1;Ext1;Ext2; exclude from scan this extensions
 /U, /Sup=Path   set suspected quarantine folder
 /aLevel=[n]     set maximum archive depth level
 /y, /CopyS      copy suspect files in quarantine folder
 /o, /CopyI      copy infected files in quarantine folder
 /move           move infected files in quarantine folder
 /moves          move suspect files in quarantine folder
 /flev[n]        set maximum folder depth level
```



```
trivialtest — bash — 110x41
Last login: Fri Sep 18 05:46:52 on ttys001
Richards-MacBook-Air: richardepattis$ mkdir ics46projects
Richards-MacBook-Air: richardepattis$ cd ics46projects
Richards-MacBook-Air: ics46projects richardepattis$ mkdir trivialtest
Richards-MacBook-Air: ics46projects richardepattis$ cd trivialtest
Richards-MacBook-Air:trivialtest richardepattis$ echo "create specified main.cpp and CMakeLists.txt files"
create specified main.cpp and CMakeLists.txt files
Richards-MacBook-Air:trivialtest richardepattis$ ls
CMakeCache.txt  main.cpp
Richards-MacBook-Air:trivialtest richardepattis$ cmake .
-- The C compiler identification is AppleClang 6.0.0_6000051
-- The CXX compiler identification is AppleClang 6.0.0_6000051
-- Check for working C compiler: /Library/Developer/CommandLineTools/usr/bin/cc
-- Check for working C compiler: /Library/Developer/CommandLineTools/usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /Library/Developer/CommandLineTools/usr/bin/c++
-- Check for working CXX compiler: /Library/Developer/CommandLineTools/usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/richardepattis/ics46projects/trivialtest
Richards-MacBook-Air:trivialtest richardepattis$ ls
CMakeCache.txt  CMakeLists.txt  main.cpp
Richards-MacBook-Air:trivialtest richardepattis$ make
Richards-MacBook-Air:trivialtest richardepattis$ make
Scanning dependencies of target trivialtest
[ 50%] Building CXX object CMakeFiles/trivialtest.dir/main.cpp.o
[100%] Linking CXX executable trivialtest
[100%] Built target trivialtest
Richards-MacBook-Air:trivialtest richardepattis$ ls
CMakeCache.txt  CMakeLists.txt  make_install.cmake  trivialtest
CMakeFiles      Makefile      main.cpp
Richards-MacBook-Air:trivialtest richardepattis$ ./trivialtest
Hello, World!
Richards-MacBook-Air:trivialtest richardepattis$
```

Interaction Style #5: Natural Language

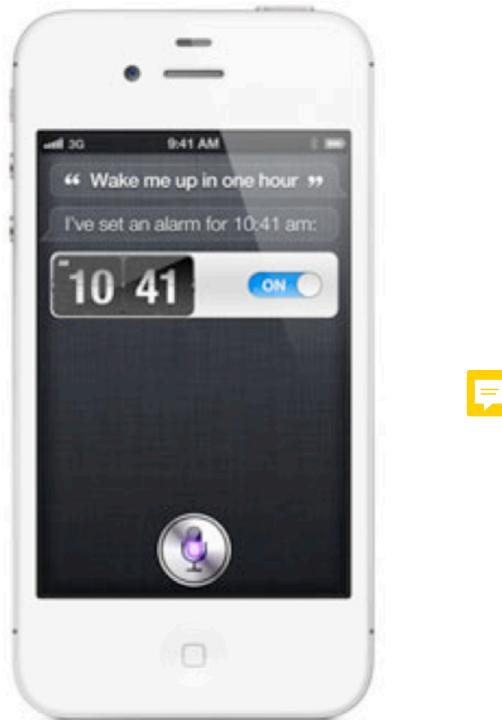
- Computers will respond properly to arbitrary natural language sentences or phrases.
- **Pros:** easy to learn
- **Cons:** unpredictable, requires clarification dialog, technology is not fully developed . . Still in research



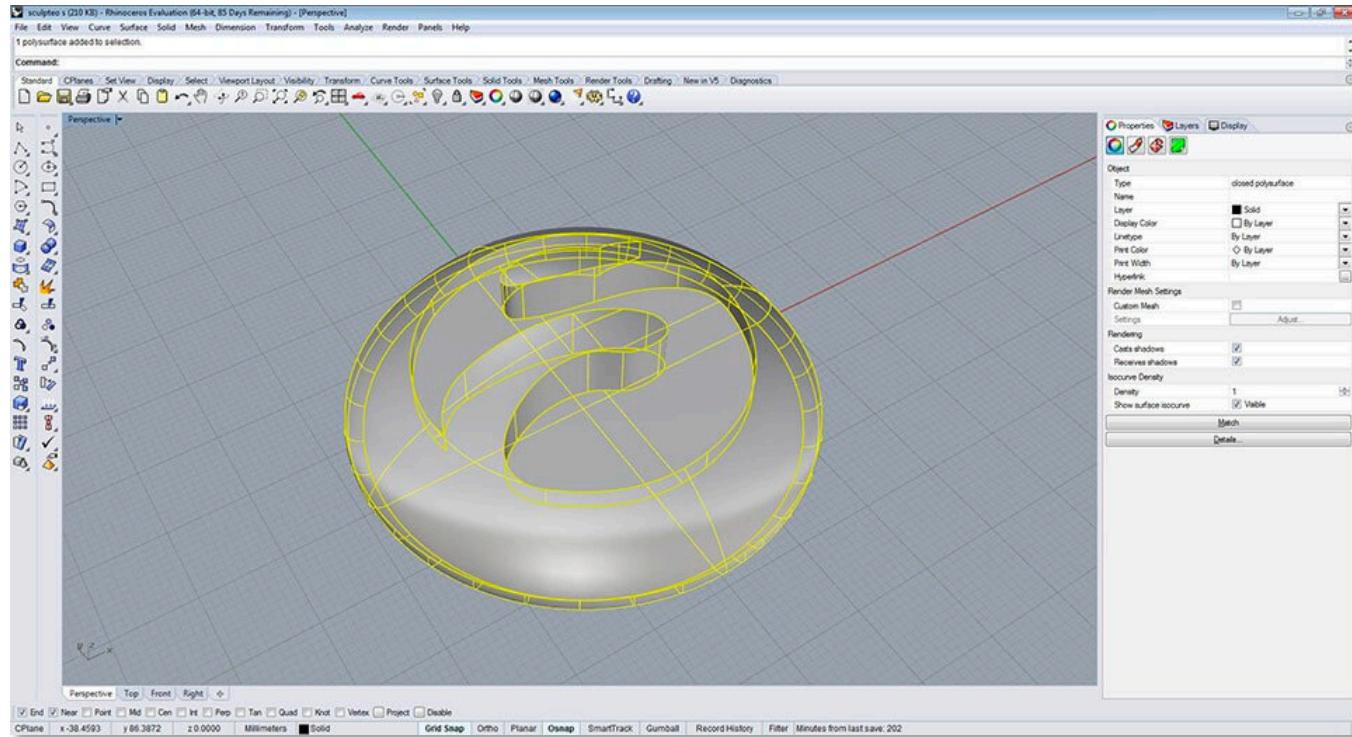
Siri - Apple

Using Multiple Interaction Styles

We can also appropriately **blend** several of them together in the same interface as well as for the same required task



iPhone Alarm



Rhinoceros 3D

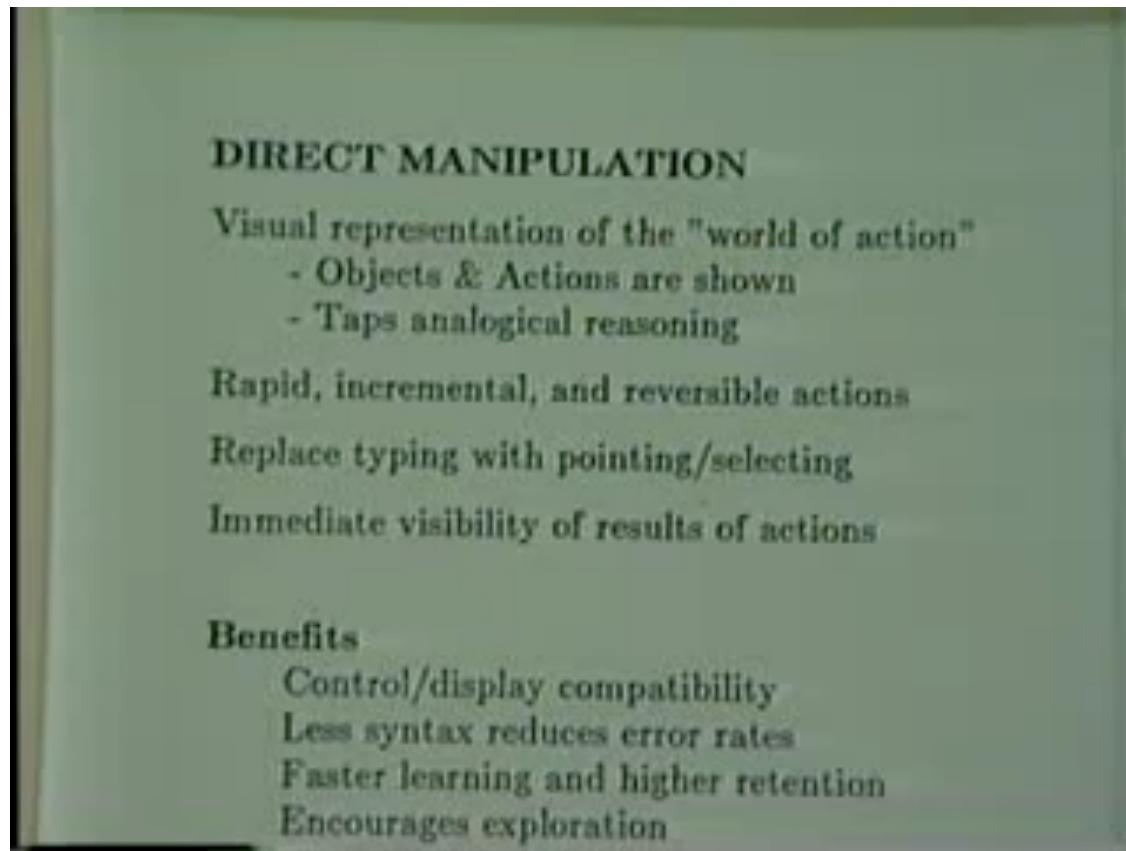
Interaction Style

Five main types of interaction style

1. **Direct Manipulation** (this lecture)
2. Menu Selection (next lecture)
3. Form fill-in (next lecture)
4. Command Language (read Chapter 9.5)
5. Natural Language (read Chapter 9.2, 9.3, 9.4)

Definition of Direct Manipulation

A user interface technique (an interaction style) that lets a user start application functions by manipulating objects.



Youtube Link:

<https://www.youtube.com/watch?v=CWgPe8VjTsM>

Classic Example of Direct Manipulation

Example: Driving a car



- Scene is directly visible outside the window
- Controls have well-known actions (brake, accelerate)
- Turn right is simply rotating the “wheel” to the right
- Turn left is simply rotating the “wheel” to the left

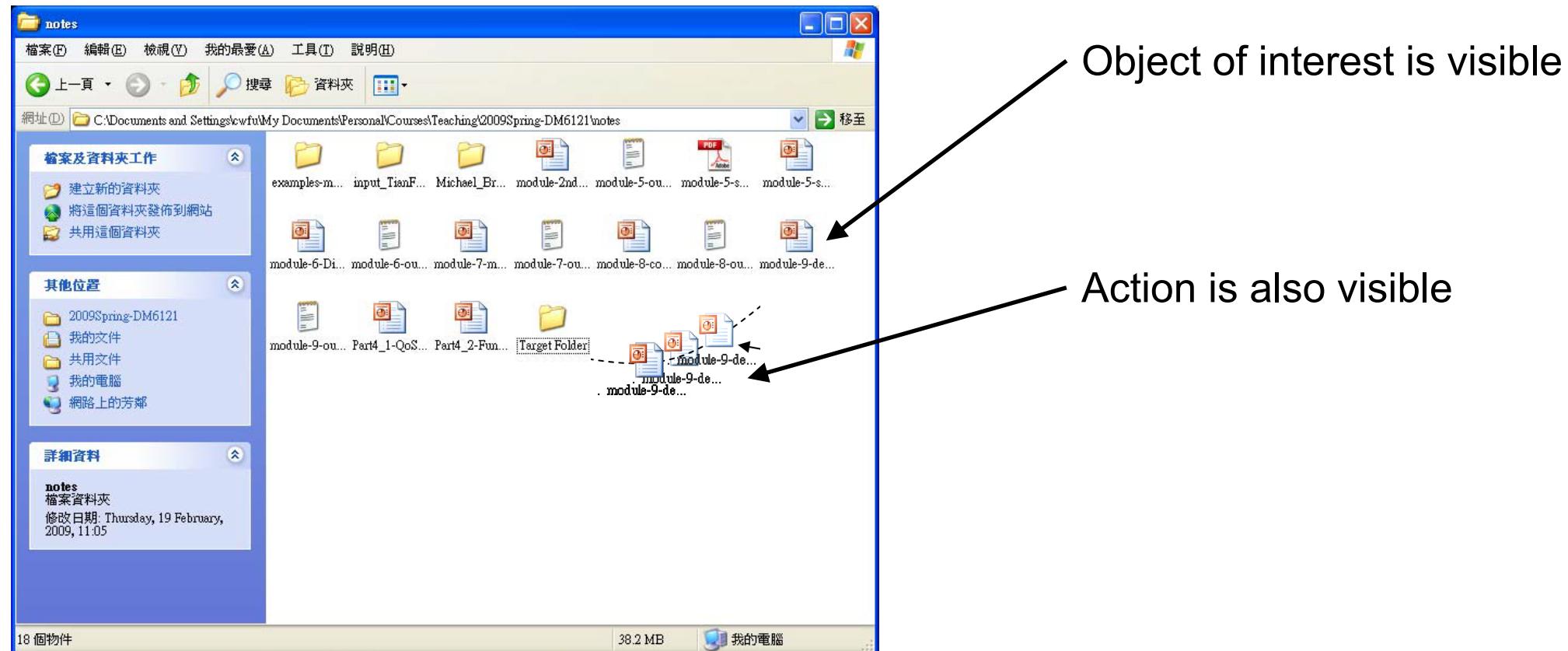
Imagine if driving had a **command line interface**, you had to type in:

command> right = 45 degree

To turn right.

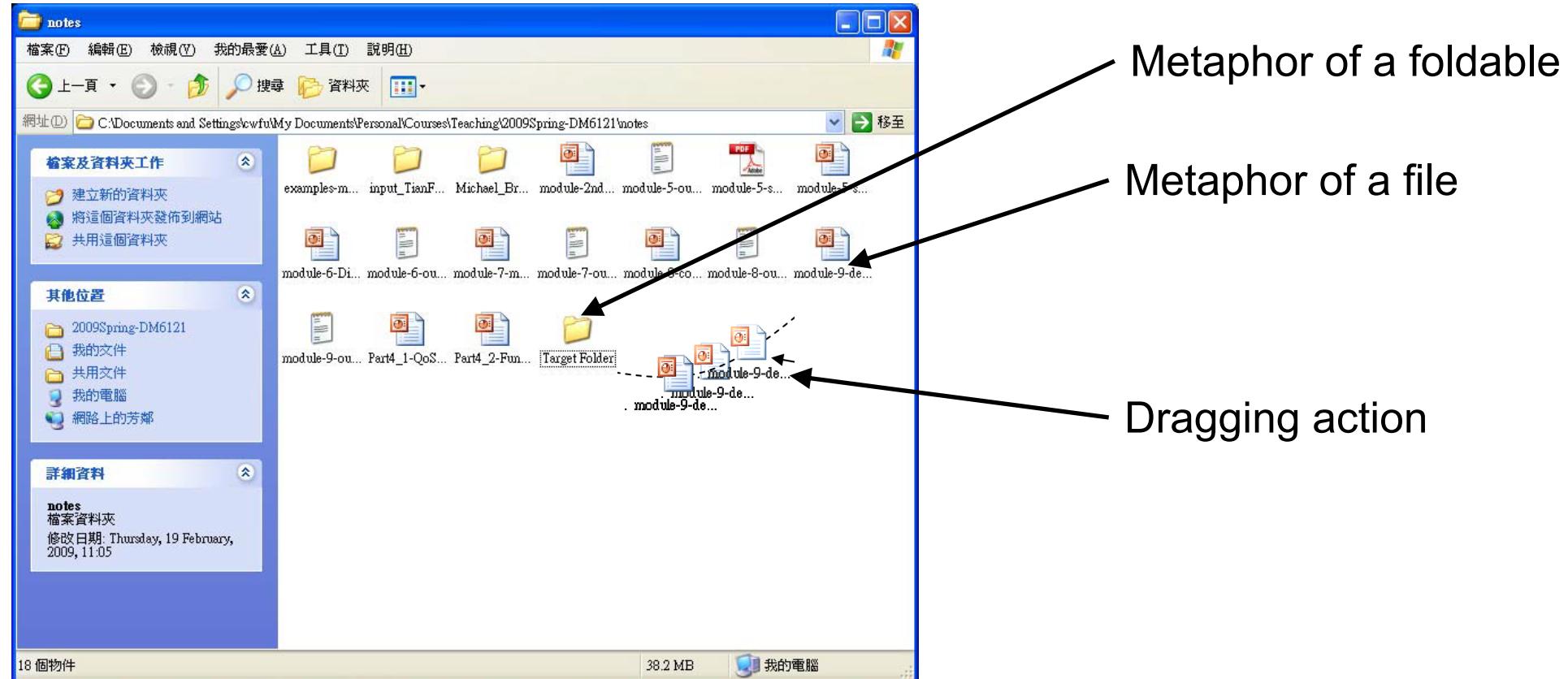
Classic Example of Direct Manipulation

Example: Moving a file from one folder to another



Classic Example of Direct Manipulation

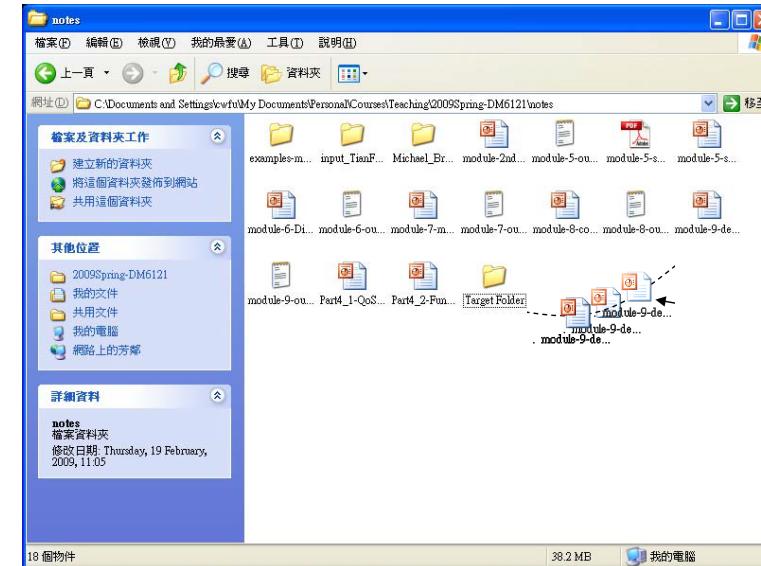
Example: Moving a file from one folder to another



Interaction Model

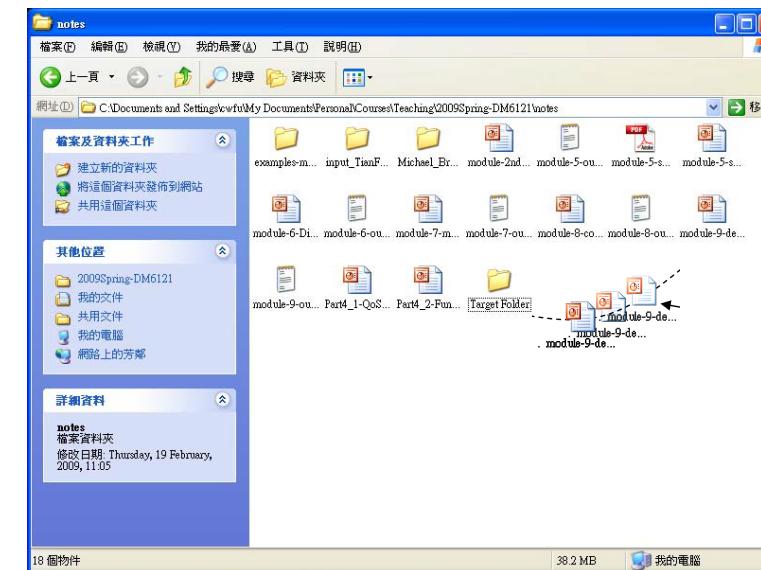
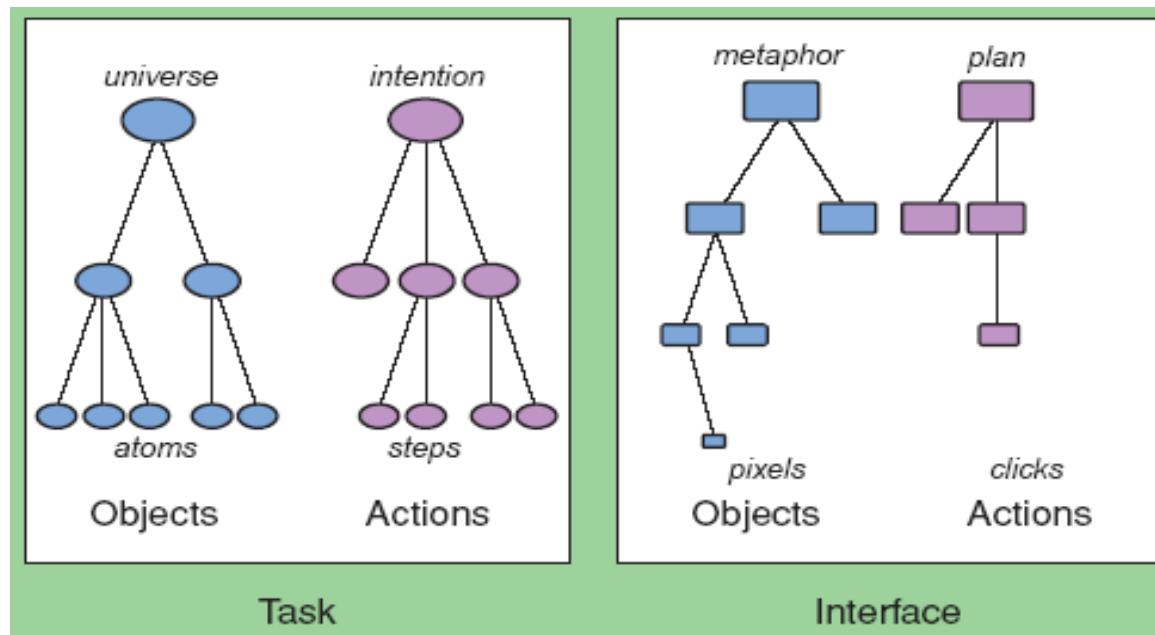
- There exist two basic interaction models for any given system:
 - **Object-action interface model:** The user first selects an object and then selects the action to be performed on the selected object
 - **Action-object interface model:** The user first selects an action to be performed and then selects the objects on which this action will be performed

```
move example.doc c:\TargetFolder
```



Object-Action Interface Model

- **Object-Action interface (OAI) model:** the user first selects an object and then selects (or performs) action on the selected object
- The OAI model is an explanatory model that focuses on **task objects** and actions, and on **interface objects** and actions



Object-Action Interface Design

- Understand the task
 - real-world objects
 - actions applied to those objects
- Create **metaphoric representations** of interface **objects** and **actions**
 - Interface objects are pixels that can be moved or copied in ways that represent real world task objects with feedback to guide users
- Make **interface actions** on the metaphors **visible** to users
 - Decompose users' plans into a series of intermediate actions, such as opening a dialog box, all the way down to a series of detailed keystrokes and clicks

Task Hierarchies of Objects and Actions

- Decomposition of real-world complex system natural
 - Human body
 - Mechanical devices
 - Buildings
 - Cities
- Tasks include **hierarchies** of objects and actions at different high and low levels
 - objects: directories, files
 - actions: save a file, apply access-control rights, assign a name

Task Hierarchies of Objects and Actions

- The following steps are recommended (by Shneiderman) in order to build **correct tasks hierarchies** for an interaction system:
 1. Know about the users and their tasks (interviewing users, reading workbooks, and taking training sessions)
 2. Generate hierarchies of tasks and objects to model the user's tasks
 3. Design interface objects and actions that metaphorically map to the real world universe

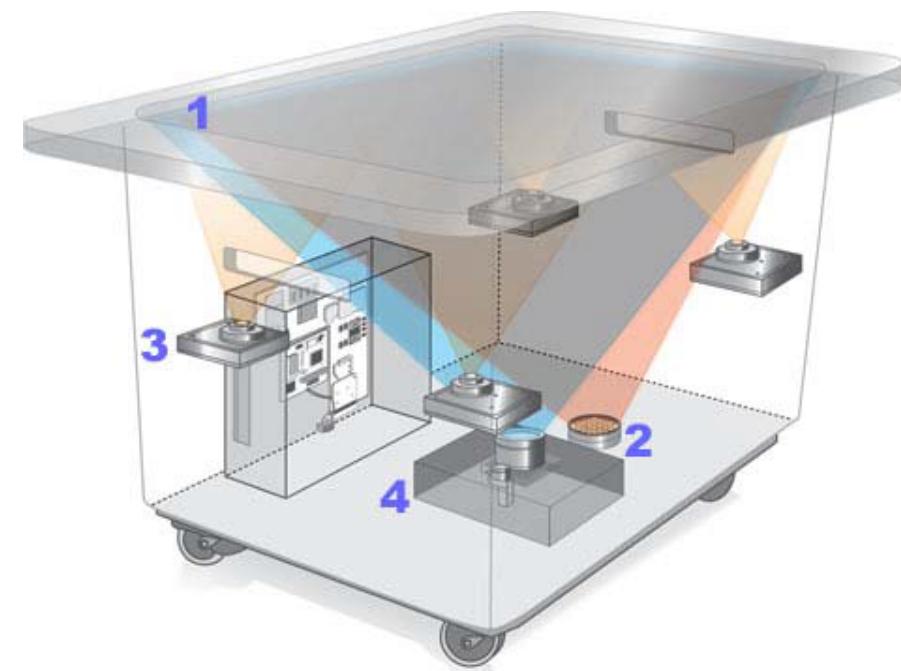
Multi-touch Interaction

A user interaction technique typically used to manipulate graphical entities with multiple fingers at the same time.



Multi-touch Interaction

- How? Two common approaches
 - Camera and image processing



Multi-touch Interaction

- How? Two common approaches
 - Camera and image processing
 - Touch-sensitive capacitance sensors



Projected capacitive touchscreen.

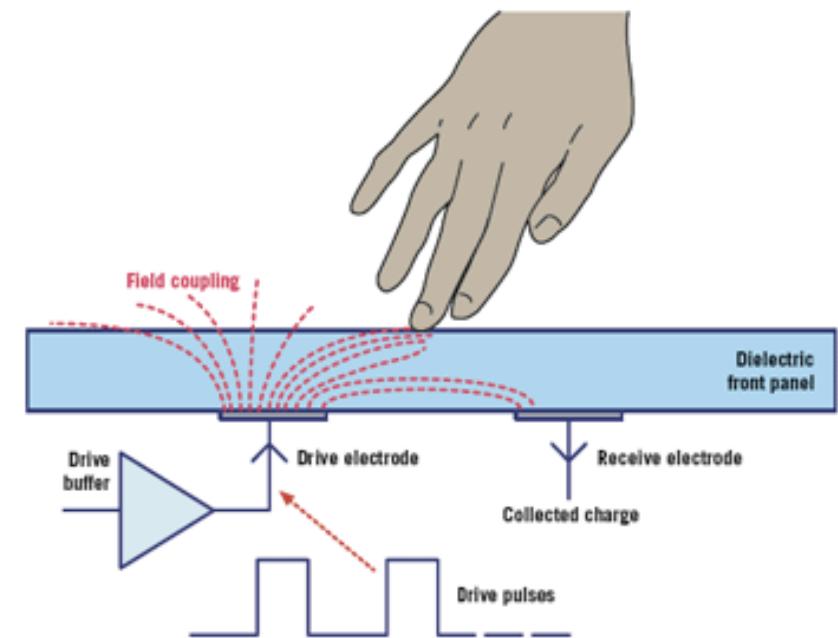
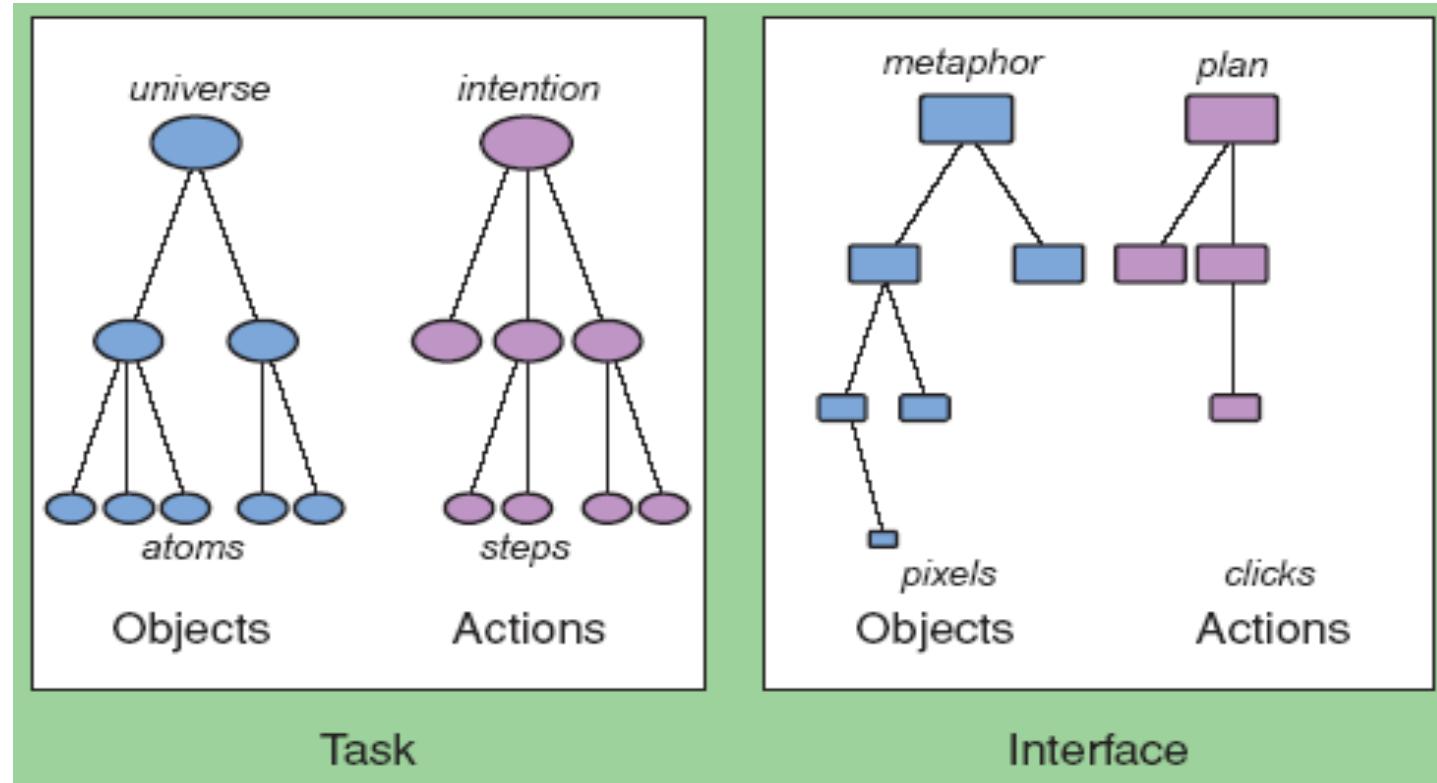


Figure 1

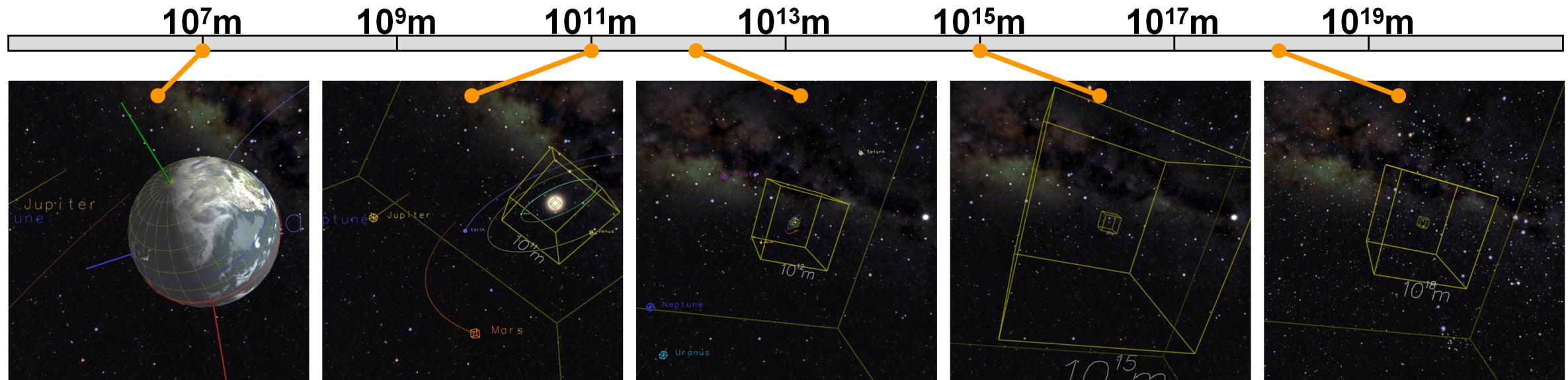
Direct Manipulation with Multi-touch

Goal: design a multi-touch interface to explore 3D astrophysical simulations guided by the Object-Action Interface Model



Direct Manipulation with Multi-touch

Example: [Multi-touch Techniques for Exploring Large-Scale 3D Astrophysical Simulations](#), CHI 2010



Hierarchy of objects: earth – solar system – the galaxy

Actions at different levels: navigate the solar system, navigate the galaxy

Direct Manipulation with Multi-touch

Goal: design a multi-touch interface to explore 3D astrophysical simulations guided by the Object-Action Interface Model

Objects in task

Earth

Mars

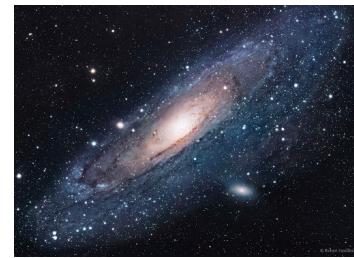
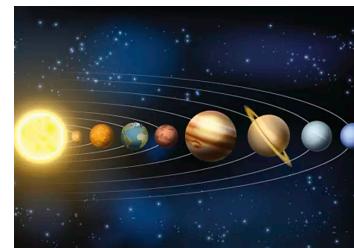
Sun

Solar system

Galaxy

...

Objects in interface



Actions in task

Select a star

Rotate around the Sun

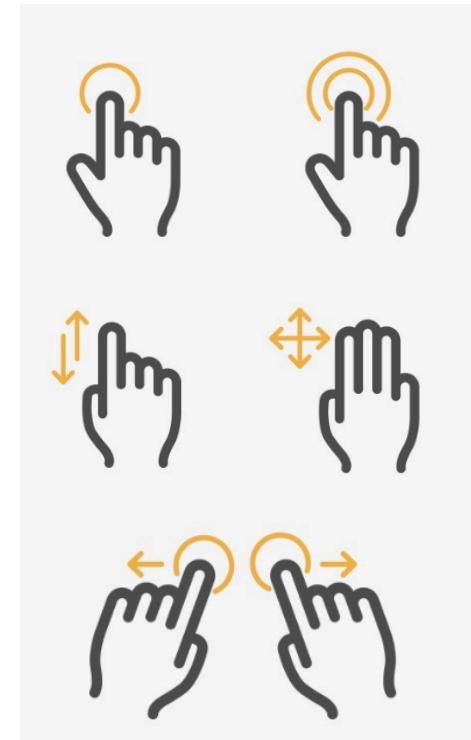
Rotate the virtual camera

Navigate the solar system

Navigate the galaxy

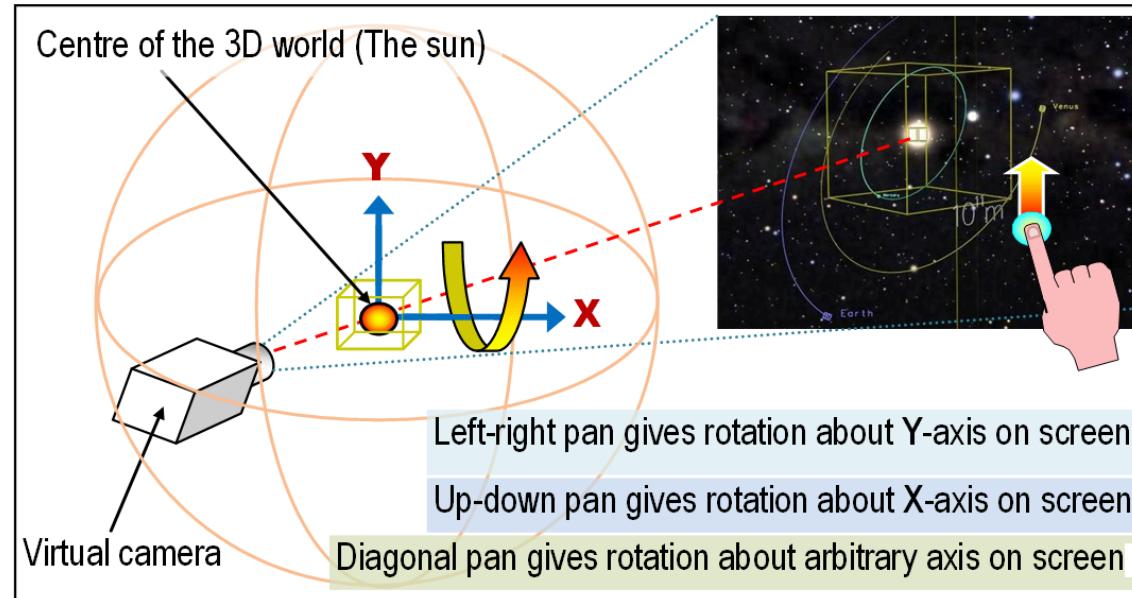
...

Actions in interface



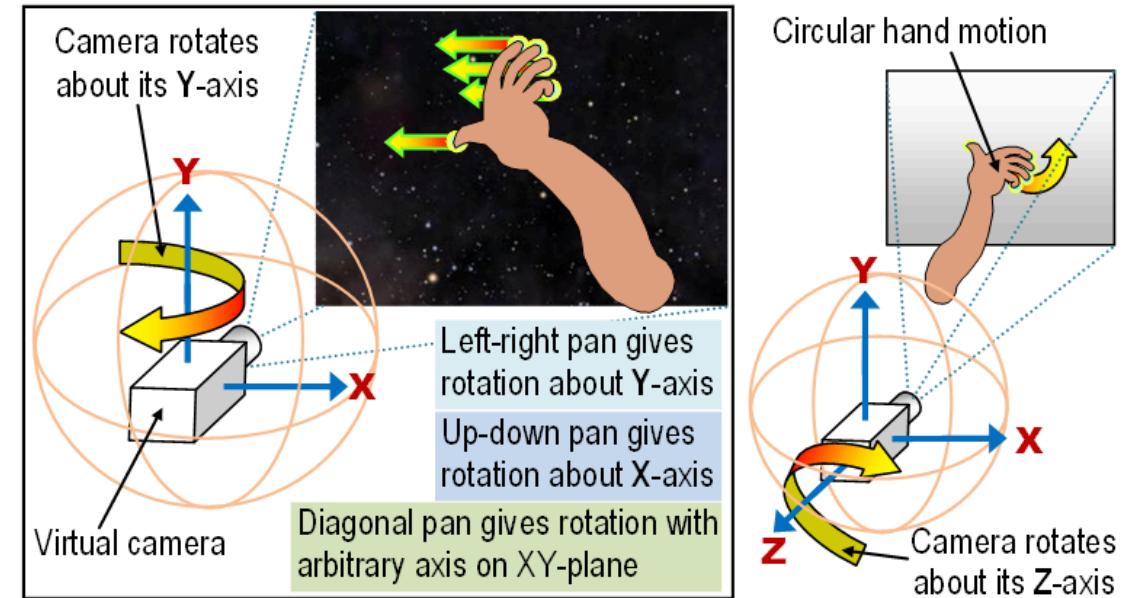
Direct Manipulation with Multi-touch

Multi-touch interaction design: explore the 3D world



Task: rotate the 3D world about its center

Action: one-finger drag gesture

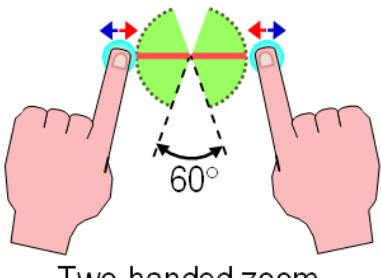


Task: rotate the virtual camera

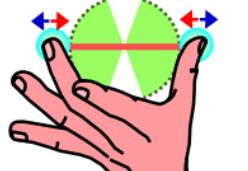
Action: five-finger pan gesture

Direct Manipulation with Multi-touch

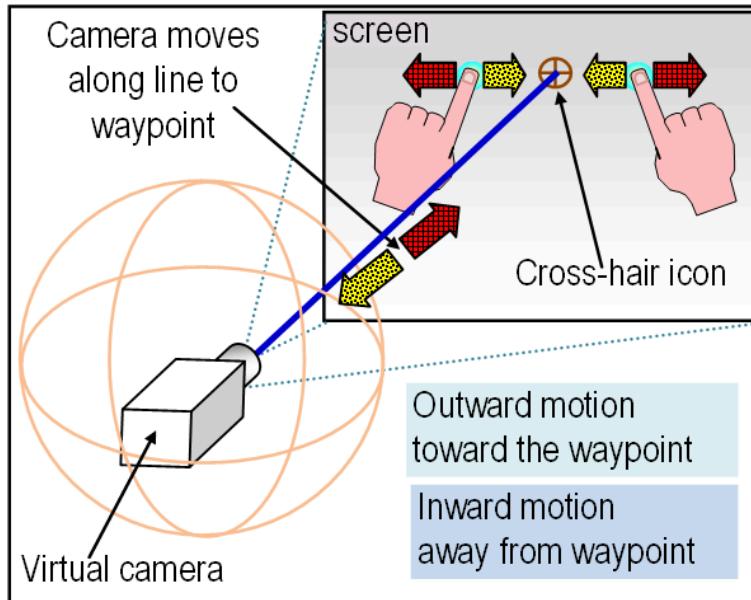
Multi-touch interaction design: navigate the solar system or the galaxy



Two-handed zoom

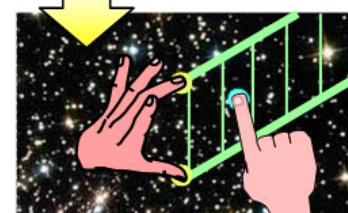
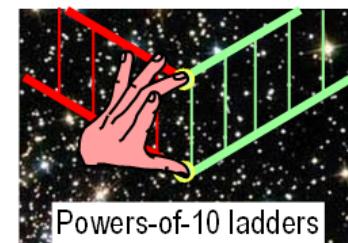


One-handed zoom

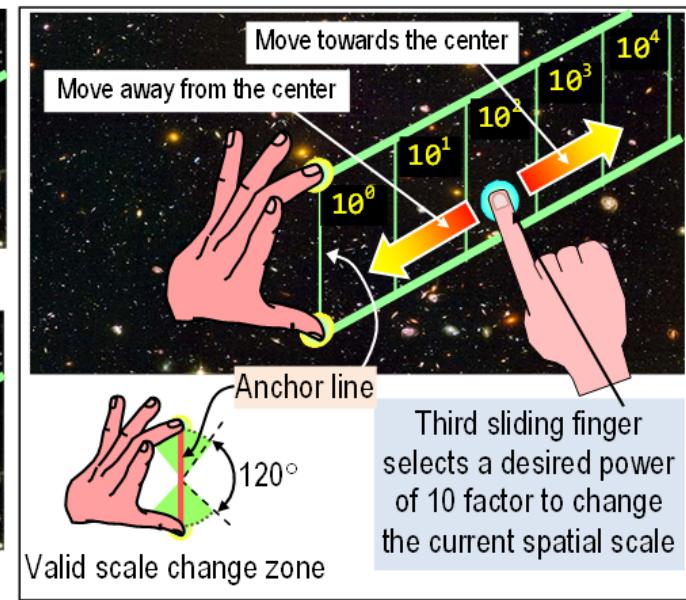


Task: navigate the solar system

Action: two-finger pinch gesture



Right-handed ladder

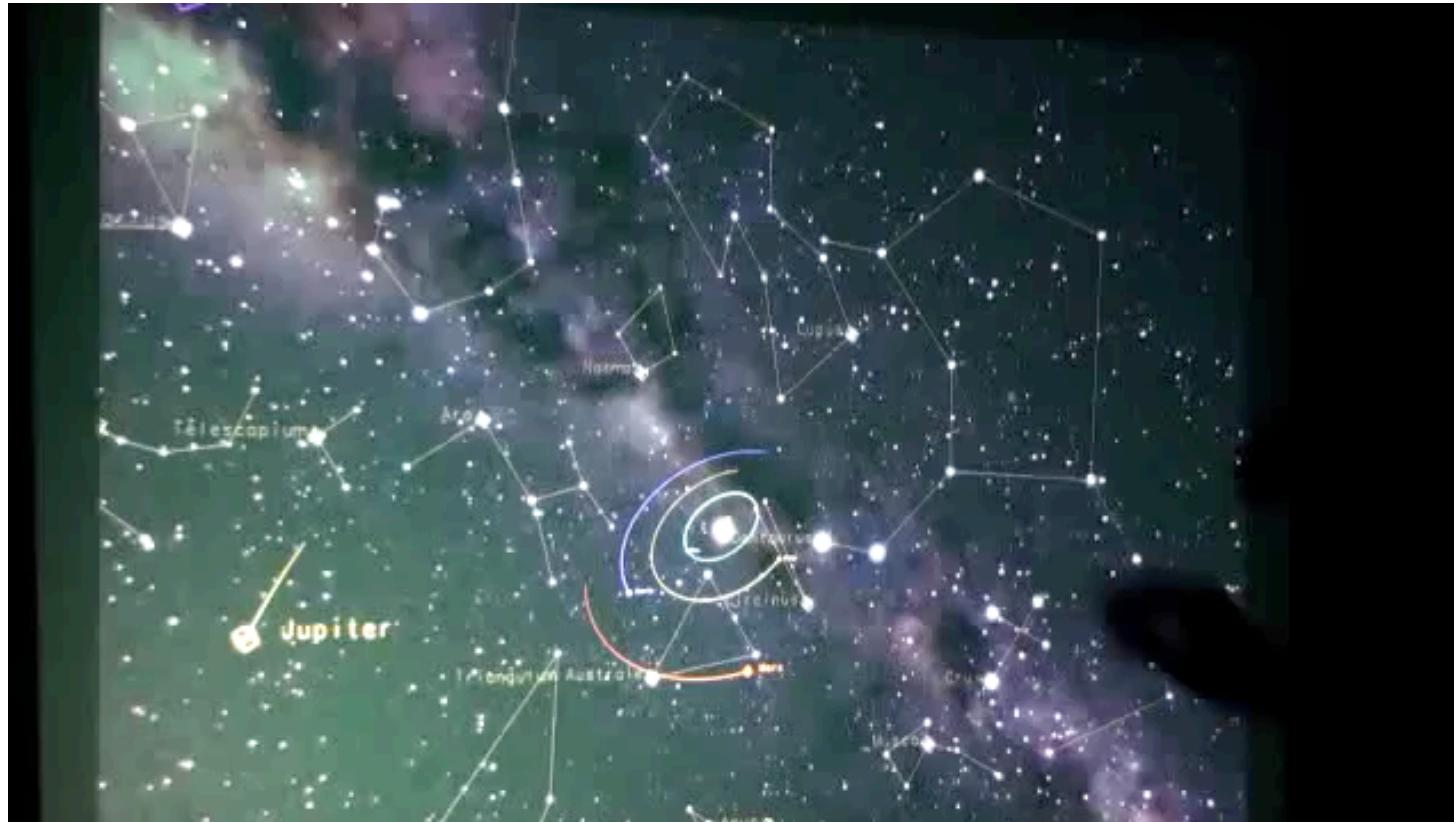


Task: navigate the galaxy

Action: one-finger drag gesture over a ladder

Direct Manipulation with Multi-touch

Example: [Multi-touch Techniques for Exploring Large-Scale 3D Astrophysical Simulations](#), CHI 2010



Youtube Link:
<https://www.youtube.com/watch?v=5J9gqHF2f7c>

Examples of Direct Manipulation

- DM changed existing applications
 - Word processing
 - Drawing Programs and CAD, etc.
- DM blossomed via Video Games
 - Console games
 - PC games
- Beyond the desktop via VR and AR

Example 1: Word Processing

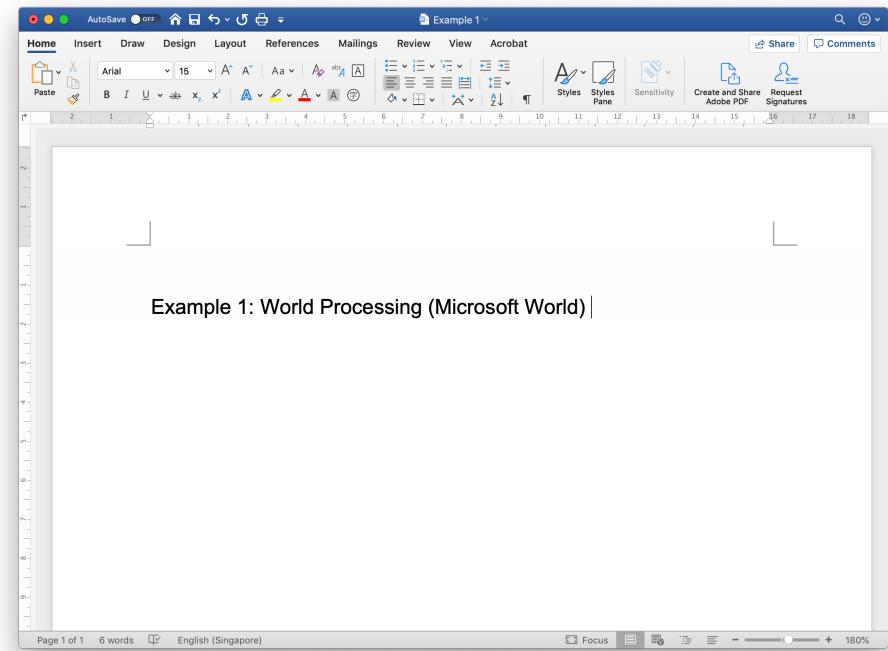
Line editor vs. WYSIWYG word processor

H:INTRO PAGE 1 LINE 9 COL 11

INSERT ON

< < <		MAIN	MENU	> > >							
--Cursor Movement--		Delete	-Miscellaneous-	-Other Menus-							
^S	char left	^D	char right	^G	char	^I	Tab	^B	Reform	(from Main only)	
^A	word left	^F	word right	DEL	chr lf	^V	INSERT ON/OFF	^J	Help	^K	Block
^E	line up	^X	line down	^T	word rt	^L	Find/Replace again	^Q	Quick	^P	Print
--Scrolling--		^Y	line	RETURN	End paragraph	^O	Onscreen				
^Z	line down	^W	line up			^N	Insert a RETURN				
^C	screen up	^R	screen down			^U	Stop a command				
-----R											
1. Introducing WordStar											
WordStar is highly flexible and very visible. Watch the screens as you give commands, and information in various parts of the screen will guide you. You won't see all the information all the time, but it will be there when you need it.											
WHERE YOU ARE											
The seven WordStar menus are your greatest aids. They are like signposts at the top of your screen, showing you where you are.											
1HELP	2INDENT	3SET	LM	4SET	RM	5UNDLIN	6BLDFCE	7BEGBLK	8ENDBLK	9BEGFIL	10ENDFIL

Wordstar Pro.



Microsoft Word for Mac

Example 1: Word Processing (line editor)

Suppose that you had a file named hello.c with the following contents:

```
int main(int argc, char **argv)
{
    printf("Hello world!\n");
    return 0;
}
```

and you wanted to change it to say "Goodbye" instead of "Hello". You might use a TECO session like this, noting that the prompt is "*" and "\$" is how ESC is echoed:

*EBhello.c\$\$	Open file for read/write with backup
*P\$\$	Read in the first page
*\$Hello\$OTT\$\$	Search for "Hello" and print the line
printf("Hello world!\n");	The line
*-5DIGoodbye\$OTT\$\$	Delete "Hello", insert "Goodbye", and print the line
printf("Goodbye world!\n");	The updated line
*EX\$\$	Copy the remainder of the file and exit

Example code

[edit]

Code sample	Explanation
ER <i>file</i> \$	open file for read access
[<i>q</i> ...] <i>q</i>	push ... pop register Q (can hold number, text, or code)
< <i>code</i> >	iteration; there are codes for <i>next</i> , <i>break</i> , <i>continue</i> , etc.
<i>n</i> " <i>X</i> <i>then-code</i> <i>else-code</i> ""	if-then-else (<i>X</i> is a test type)

[TECO \(text editor\)](#)
from MIT in 1960s

Example 1: Word Processing (line editor)

H:INTRO PAGE 1 LINE 9 COL 11

< < <	M A I N	M E N U	> > >
--Cursor Movement--		-Delete-	-Miscellaneous-
^S char left	^D char right	^G char	^I Tab
^A word left	^F word right	DEL chr lf	^B Reform
^E line up	^X line down	^T word rt	^V INSERT ON/OFF
--Scrolling--		^Y line	^L Find/Replace again
^Z line down	^W line up		RETURN End paragraph
^C screen up	^R screen down		^N Insert a RETURN
!-----!-----!-----!			
			^U Stop a command
!-----!-----!-----!			
R			

1. Introducing WordStar

WordStar is highly flexible and very visible. Watch the screens as you give commands, and information in various parts of the screen will guide you. You won't see all the information all the time, but it will be there when you need it.

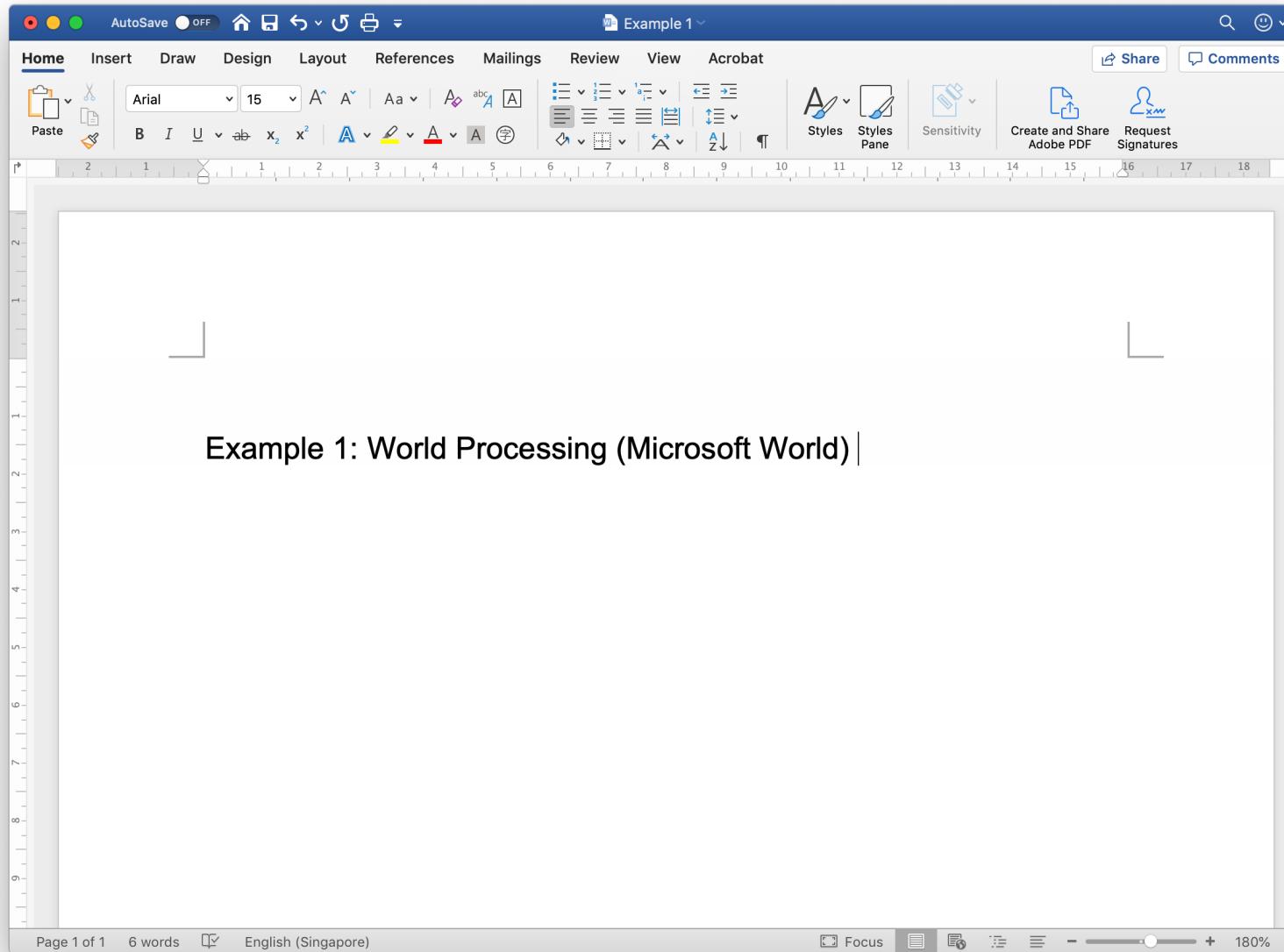
WHERE YOU ARE

The seven WordStar menus are your greatest aids. They are like signposts at the top of your screen, showing you where you are.

1HELP 2INDENT 3SET LM 4SET RM 5UNDLIN 6BLDFCE 7BEGBLK 8ENDBLK 9BEGFIL 10ENDFI

Wordstar by MicroPro in 1980s

Example 1: Word Processing (Microsoft Word)



Microsoft Word for Mac
Version 16.33

Example 1: Word Processing (LaTeX)

Computational Design of Wind-up Toys

PENG SONG, University of Science and Technology of China
XIAOFEI WANG, University of Science and Technology of China
XIAO TANG, University of Science and Technology of China
CHI-WING FU, The Chinese University of Hong Kong
HONGFEI XU, University of Science and Technology of China
LIGANG LIU*, University of Science and Technology of China
NILOY J. MITRA, University College London

Wind-up toys have a long history dating back to an early design of a wind-up lion by Leonardo da Vinci for welcoming Louis XII. Today they come as lightweight toys with compact internal mechanical assemblies that are powered by clockwork motors attached to a spring key.

Once such a key is rotated to tighten the spring and released, the stored potential energy drives the toy's internal mechanical parts, which in turn drive the end-effector parts of the toy to perform intriguing motion(s).

Examples include moving along a designated path, swinging body parts, or combinations of multiple motions over time and space.

\revised{Fig.}~\ref{fig:example}(a) shows a typical wind-up toy that moves along a wobbling path while swinging its tail.

\if0 \TODO{what are windup toys? why are they fun/challenging}

\song{Definition:
A wind-up toy is a toy powered by a spring motor.
A key, or crank is turned to tighten the spring.
When release the key, the potential energy stored in the spring powers the toy's mechanisms.}\if0

Why fun/challenging:
Since the spring motor can provide limited energy and torque, a lightweight mechanism is required to

ACM Reference Format:
Peng Song, Xiaofei Wang, Xiao Tang, Chi-Wing Fu, Hongfei Xu, Ligang Liu, and Niloy J. Mitra. 2017. Computational Design of Wind-up Toys. *ACM Trans. Graph.* 36, 6, Article 238 (November 2017), 13 pages. <https://doi.org/10.1145/3130800.3130808>

1 INTRODUCTION

Wind-up toys have a long history dating back to an early design of a wind-up lion by Leonardo da Vinci for welcoming Louis XII. Today they come as lightweight toys with compact internal mechanical assemblies that are powered by clockwork motors attached to a spring key. Once such a key is rotated to tighten the spring and released, the stored potential energy drives the toy's internal mechanical parts, which in turn drive the end-effector parts of the toy to perform intriguing motion(s). Examples include moving along a designated path, swinging body parts, or combinations of multiple motions over time and space. Fig. 2(a) shows a typical wind-up toy that moves along a wobbling path while swinging its tail.

Compared to mechanical assemblies in previous research [Bächer et al. 2015; Ceylan et al. 2013; Coros et al. 2013; Thomazewski et al. 2014; Zhu et al. 2012], wind-up toys have several distinctive characteristics. First, they are driven by a simple spring motor that can only give out limited energy and torque. Second, their parts usually perform simple periodic motions that are only roughly specified. Third, they often have a small lightweight body, particularly for those that perform locomotion. Lastly, they are usually presented as

TeXstudio

Example 1: Word Processing

- Line editors (in early file editing):
 - Users see one line at a time
 - Typed commands were needed to move the one-line window up/down
- The advances of WYSIWYG word processors:
 - Display a full page of text
 - Display of the document in the form that it will appear when the final printing is done
 - Show cursor action
 - Control cursor motion through physically obvious and intuitively natural means
 - Use of labeled icon for actions
 - Display of the results of an action immediately
 - Provide rapid response and display
 - Offer easily reversible actions
- Training times with WYSIWYG word processors are much less than line editors

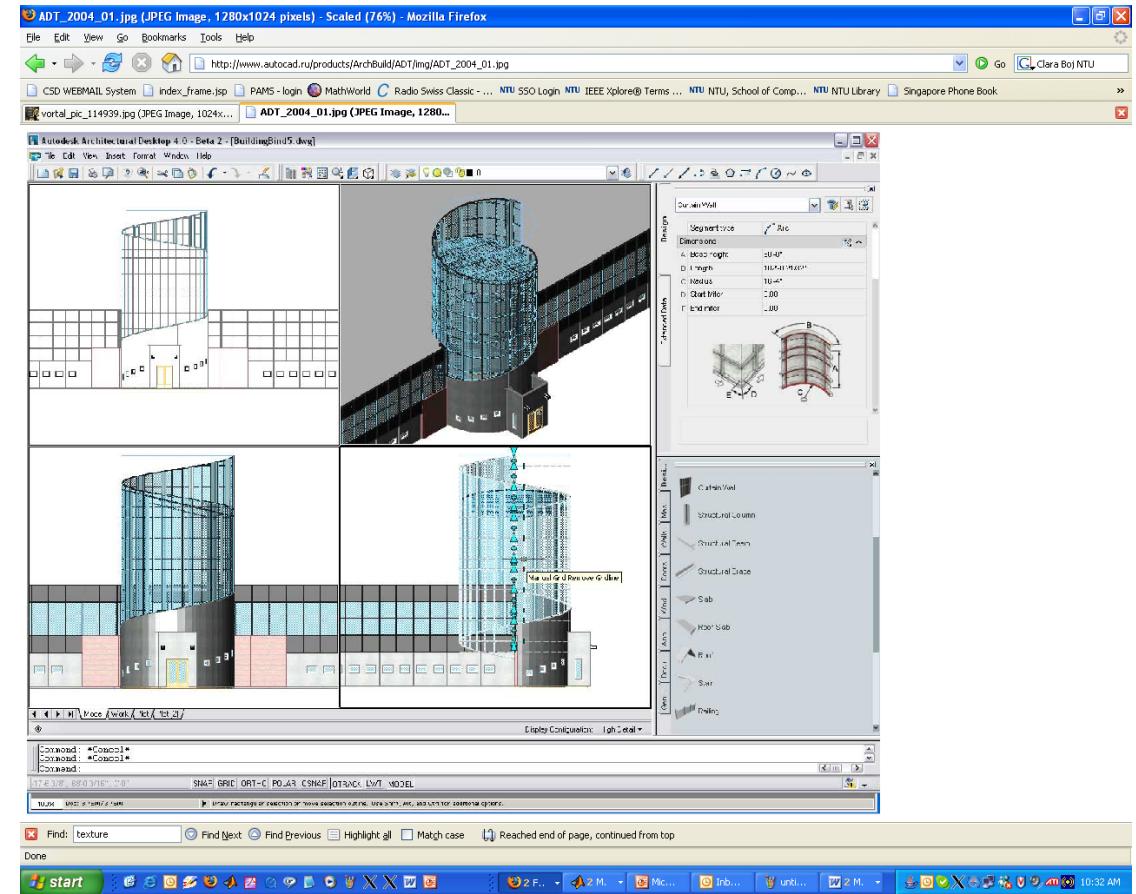
Example 1: Word Processing

Technologies derived from the word processor:

- Integration of other components, e.g., graphics, spreadsheets, etc.
- Desktop publication software, e.g., printing and choice of fonts
- Slide-presentation software
- Hypermedia environments, e.g., jump between pages and to the WWW
- Improved macro facilities
- Spell checker and thesaurus
- Grammar checkers
- Document Assemblers (template for contract, etc.)

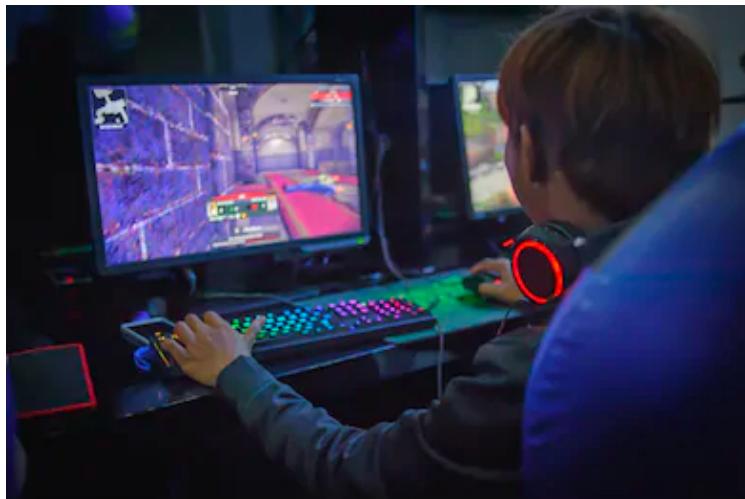
Example 2: Computer-aided design

- Computer-aided design (CAD) use direct manipulation
- Manipulate the object of interest
- Generate alternatives easily
- Explain the impact
- Problem solving by analogy to the real-world



Example 3: Video Games

- Field of action is visual and compelling
- User inputs are physical actions whose results are immediately shown on the screen
- No syntax to remember
- Most games continuously display a score (and action)



Example 3: Video Games



Need for speed



Street Fighter



Counter-Strike



Starcraft II

Example 4: Virtual Reality

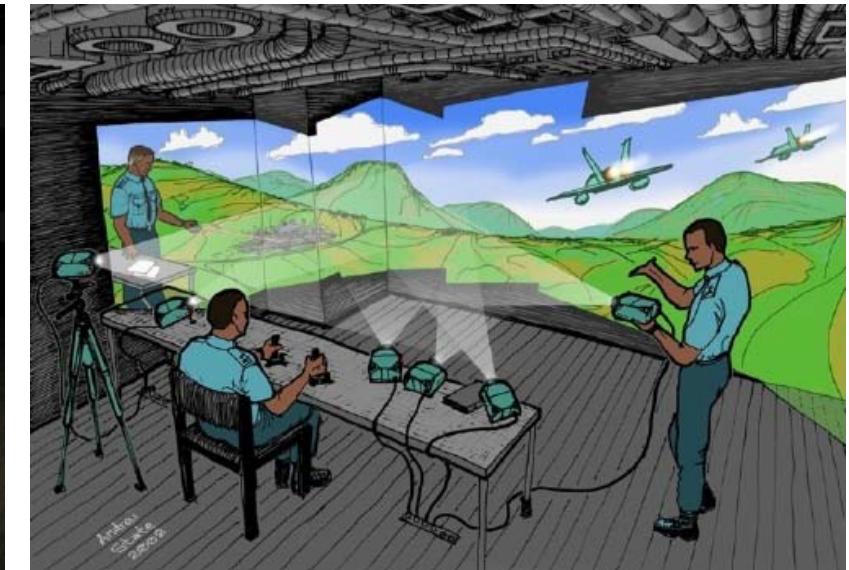
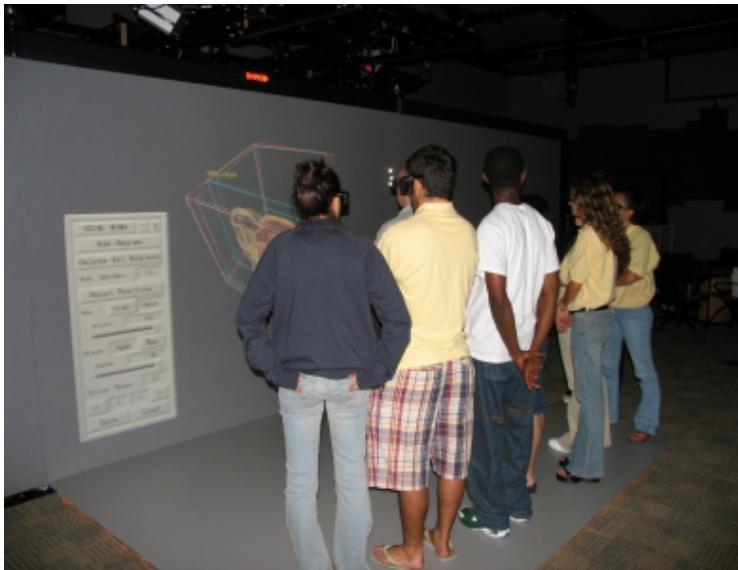
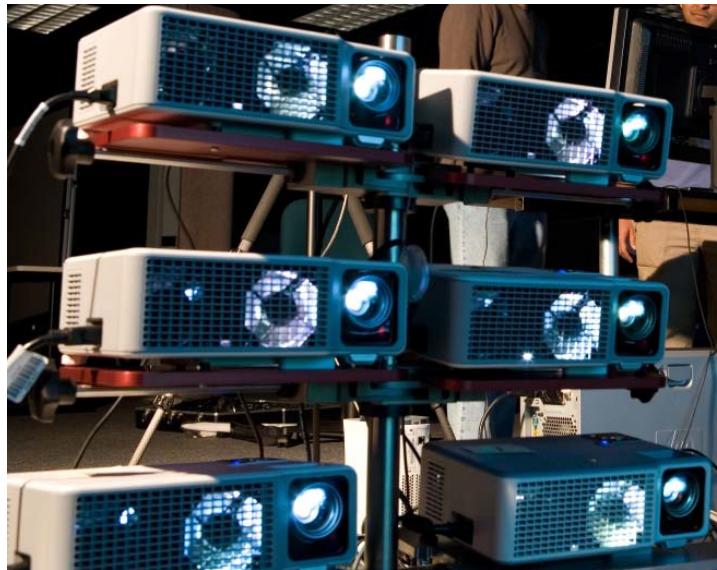
- Virtual reality breaks the physical limitations of space and allow users to act as though they were somewhere else
- Successful virtual environments depend on the smooth integration of:
 - Visual Display (usually stereo)
 - Head-position sensing
 - Force feedback
 - Sound input and output
 - Other sensations
 - Cooperative and competitive virtual reality

Example 4: Virtual Reality



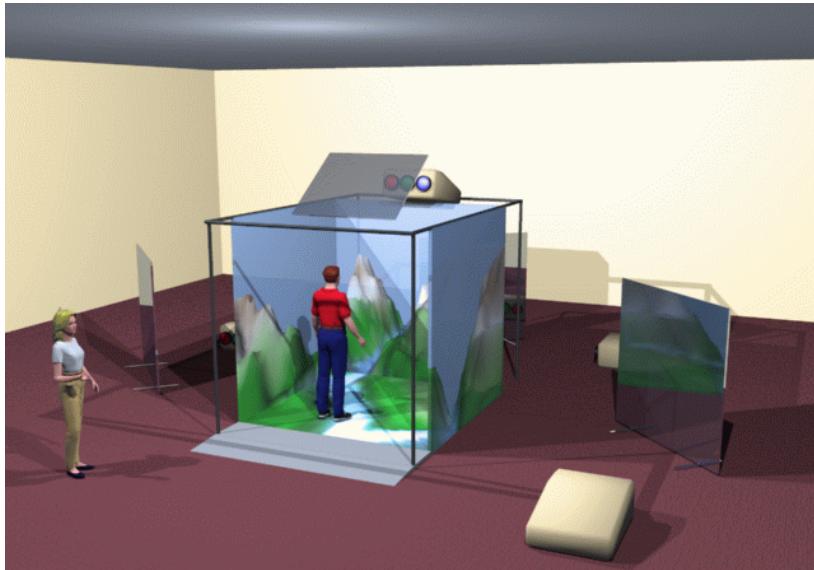
HMD (Head-mounted display)
track orientation of the user's head

Example 4: Virtual Reality



Video Wall – Multi-projector display

Example 4: Virtual Reality



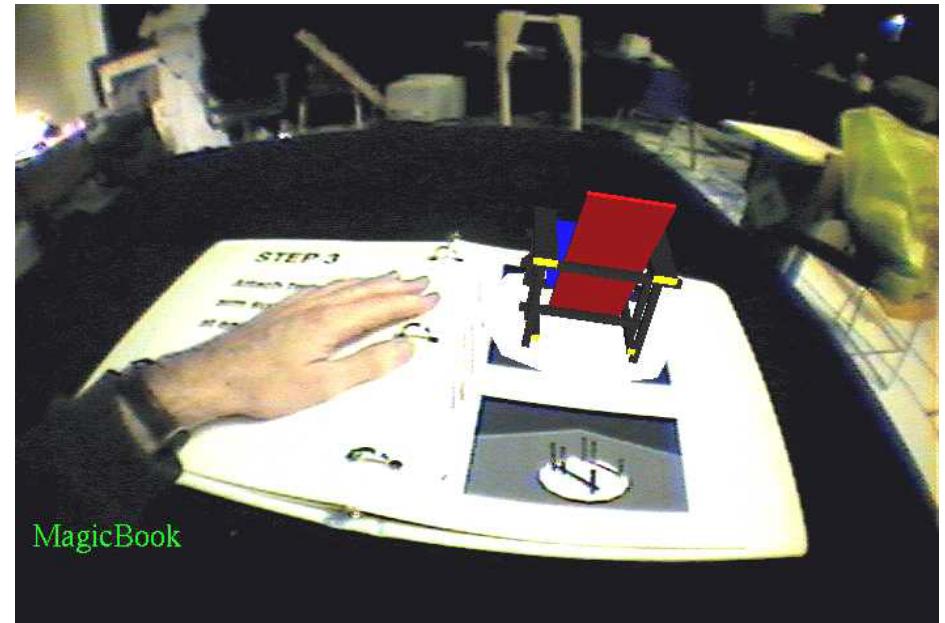
CAVE: Cave Automatic Virtual Environment
(better known by the recursive acronym CAVE)

Example 5: Augmented Reality

- Augmented reality shows the real world with an overlay of additional overlay
- Situational awareness shows information about the real world that surrounds you by tracking your movements in a computer model
- Augmented reality is an important variant
 - Enables users to see the real world with an overlay of additional interaction.



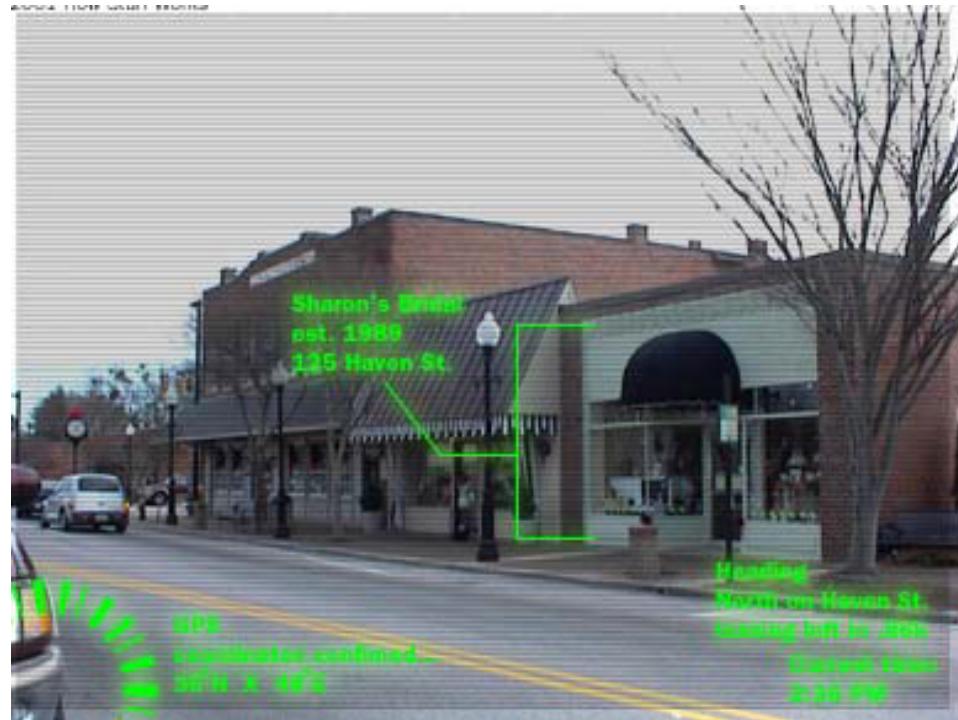
Example 5: Augmented Reality



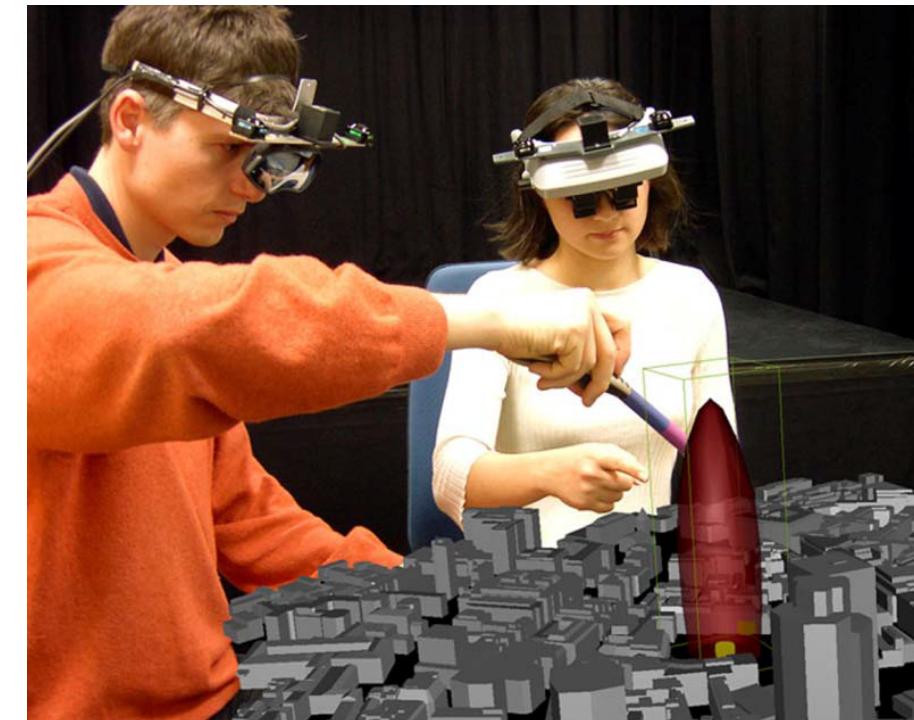
Magic book

1. Camera to track pattern on a page
2. Orientation of the user's head w.r.t. book
3. Stereo display as a overlay on real-world

Example 5: Augmented Reality



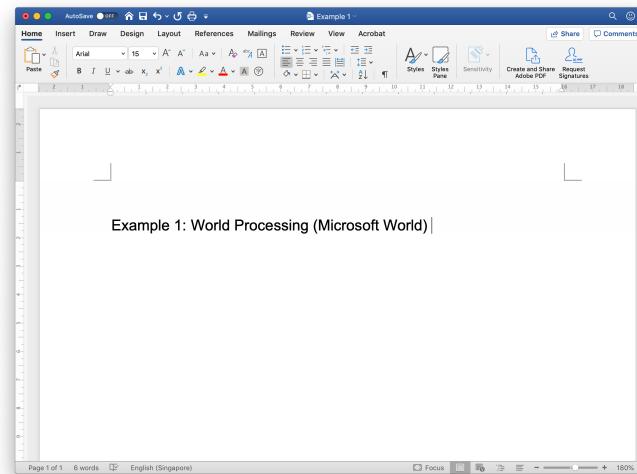
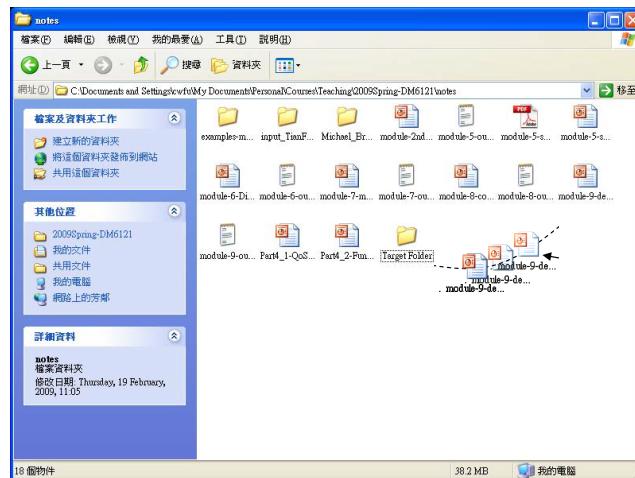
Car windshield augmentation



See-through augmented Goggles

Three Principles of Direct Manipulation

- **Principle 1:** Continuous and visible representation of the objects and actions of interest
- **Principle 2:** Rapid, incremental, and reversible actions whose effect on the object of interest is immediately visible
- **Principle 3:** Replacement of typed commands (no syntax) by a pointing action on the object of interest



Benefits of Direct Manipulation

- Novices learn quickly
- Experts (may) work rapidly
- Intermittent users can retain concepts
- Error messages are rarely needed
- Users see if their actions are furthering their goals
- Users experience less anxiety
- Users gain confidence and mastery

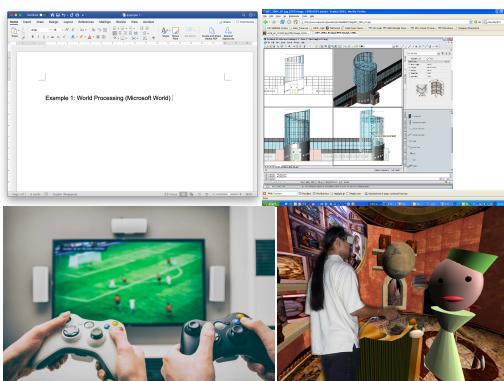
Problems of Direct Manipulation

- Spatial or visual representations can be too spread out
- High-level flowcharts and database-schema can become confusing
- Designs may force valuable information off the screen
- Users must learn the semantics behind the graphical representations
- The visual representation may be misleading (need good design of metaphors)
- Typing commands or using short-cut keys with the keyboard may be faster
 - Mouse motions take time and require a stop in typing

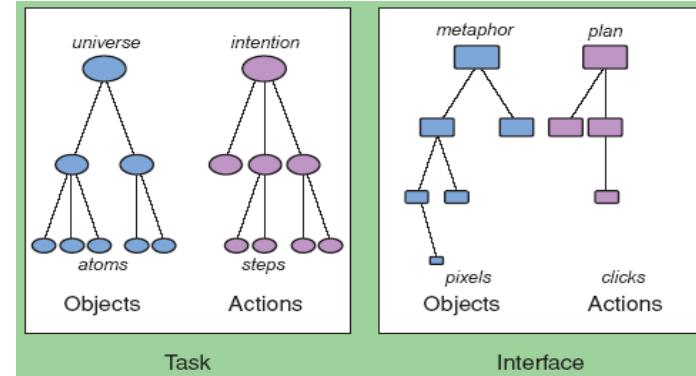
Summary



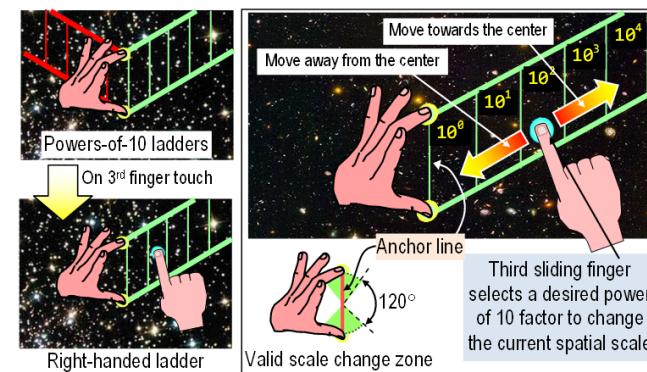
Concept of direct manipulation



Direct manipulation examples



Object-action interface model



Direct manipulation interface design

Coming Up

This Thursday: **Interaction Style II: Menus and Forms**

- Instructed by *Peng Song*
- 11:00 - 13:00 by e-learning