The History of Interaction

A Brief History of Computers
Early User Interfaces
Command-line Interfaces
Graphical User Interfaces and the WIMP paradigm

UI CS 349

May 08

A Brief History of Computers

UI CS 349

What is a computer?

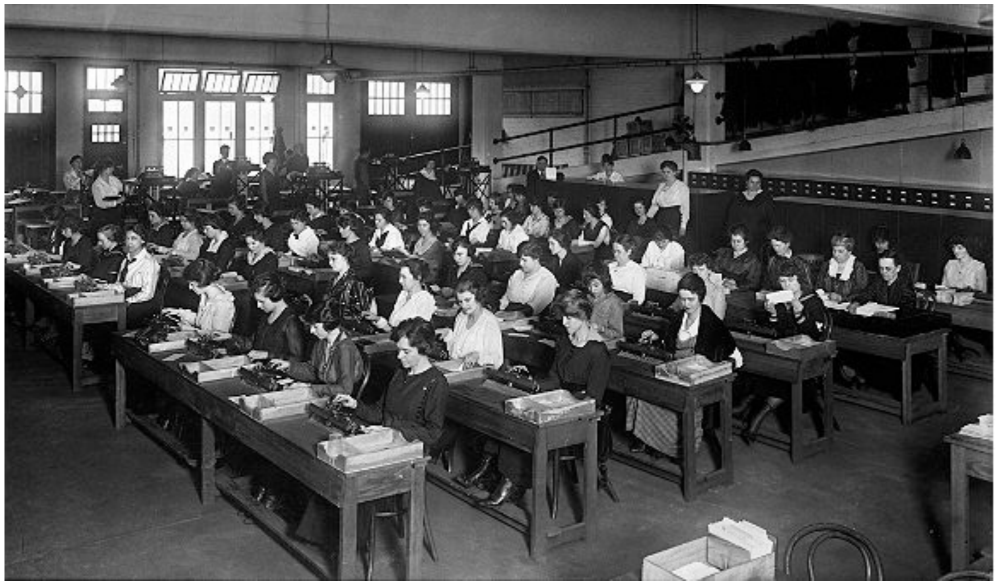
[English] Compute: calculate

[French] computer: drawing calendars according to astronomical data

[Latin] com puto: arranging together

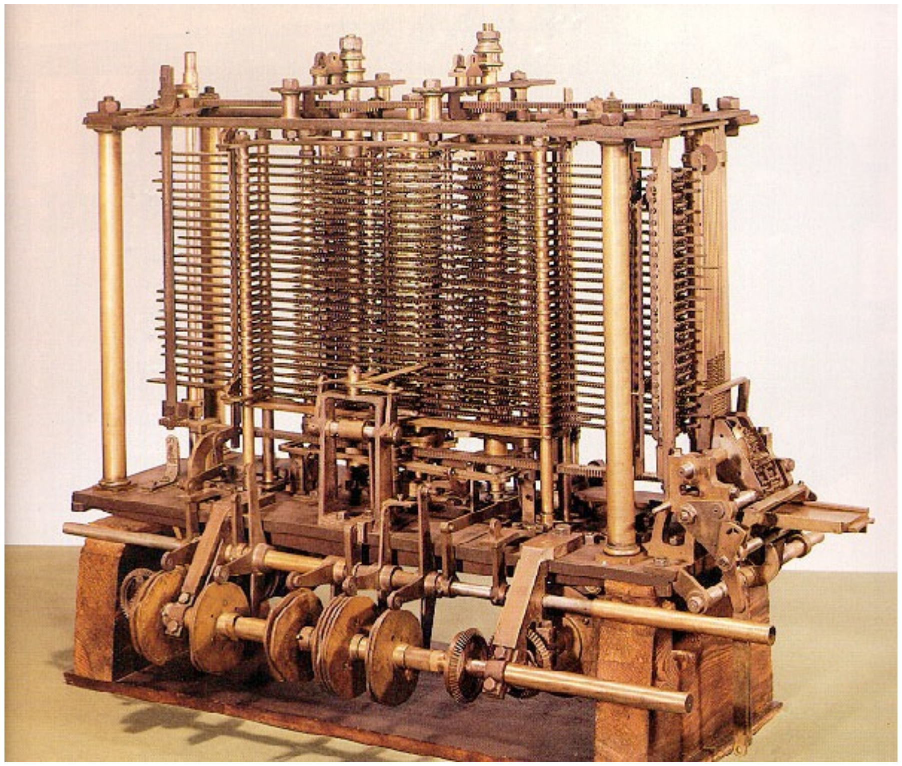
“I have read the truest computer of times…”
— Richard Brathwait, The Yong Mans Gleanings, 1613

Until the late 1800s, a computer was strictly a person, not a device.



Mechanical vs. Electro-mechanical

Analytical Engine by Charles Babbage, 1837, never built

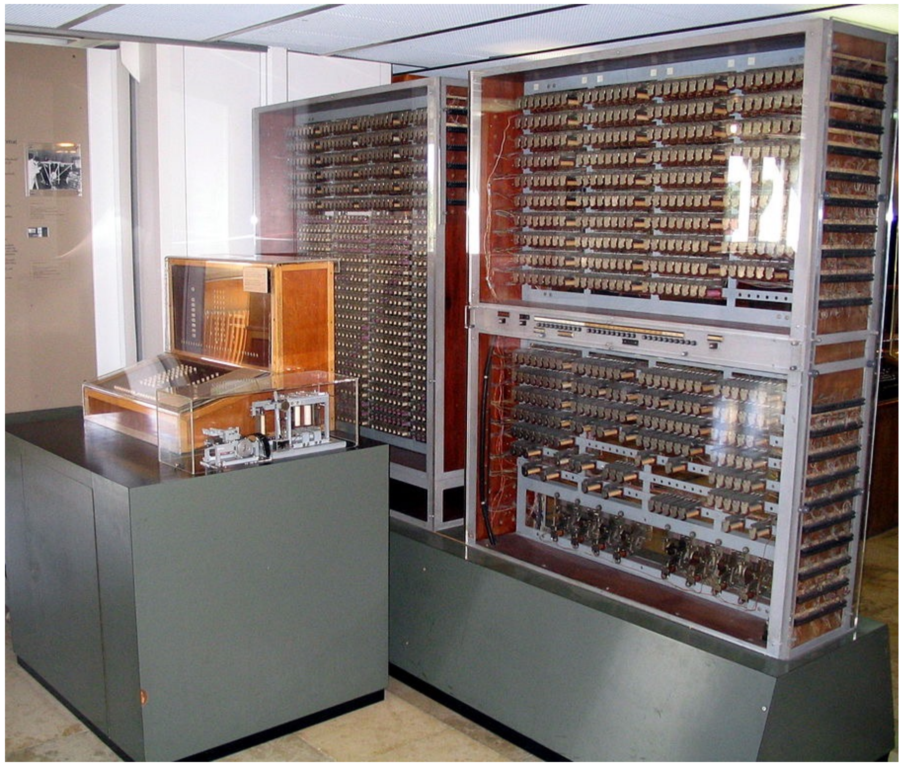


Tabulating Machine by Herman Hollerith, 1890

Analog vs. Digital

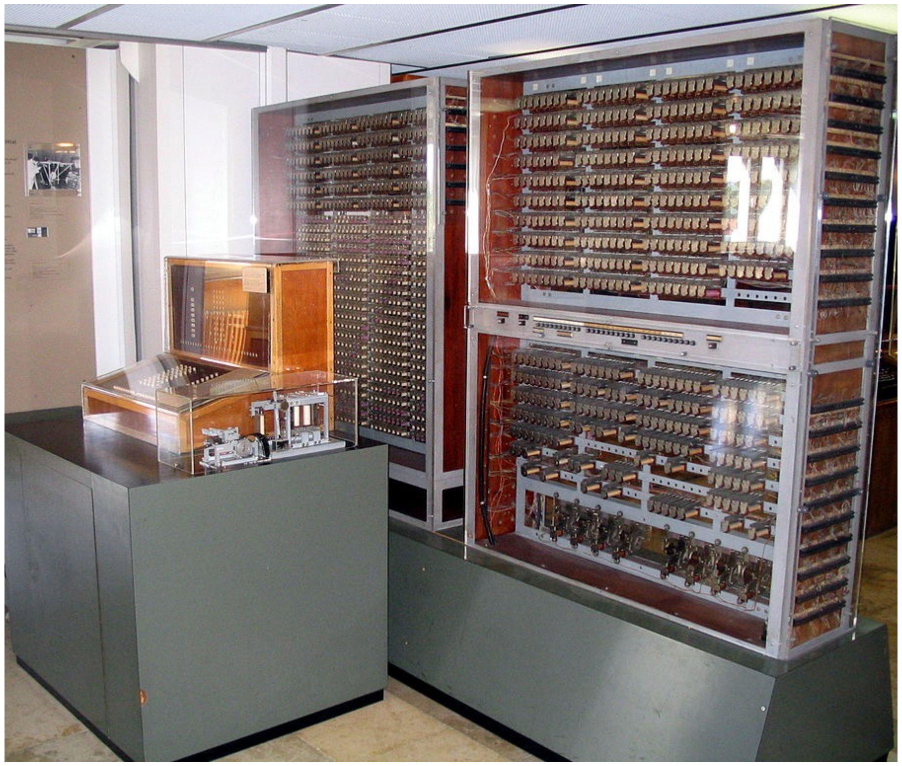
Tabulating Machine by Herman Hollerith, 1890

Z3 by Konrad Zuse, 1941

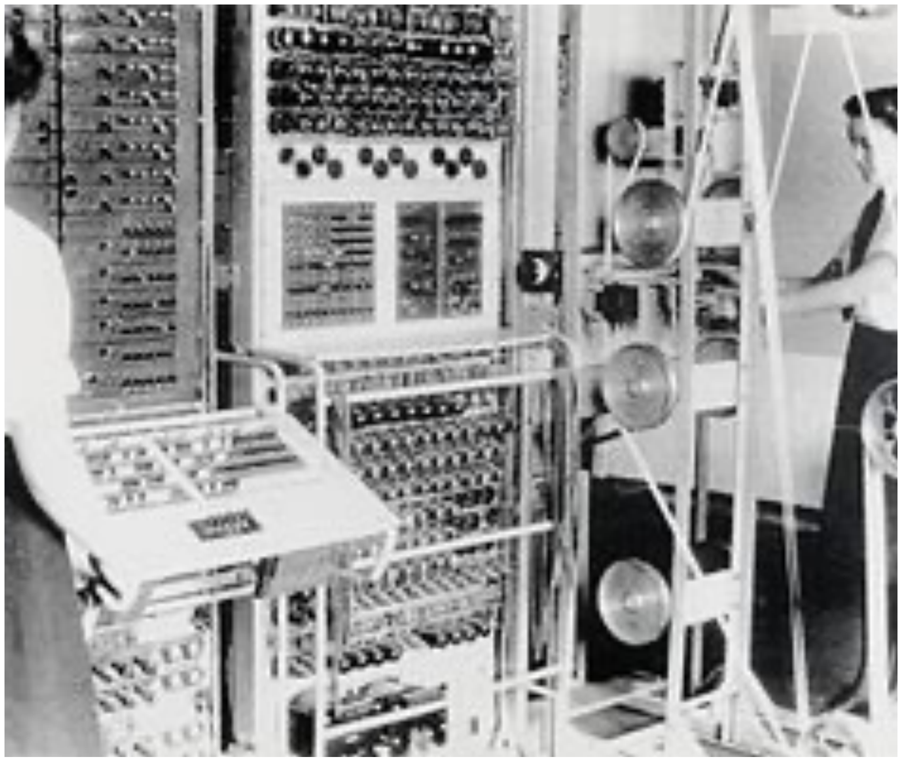


Electro-mechanical vs. Electrical

Z3 by Konrad Zuse, 1941

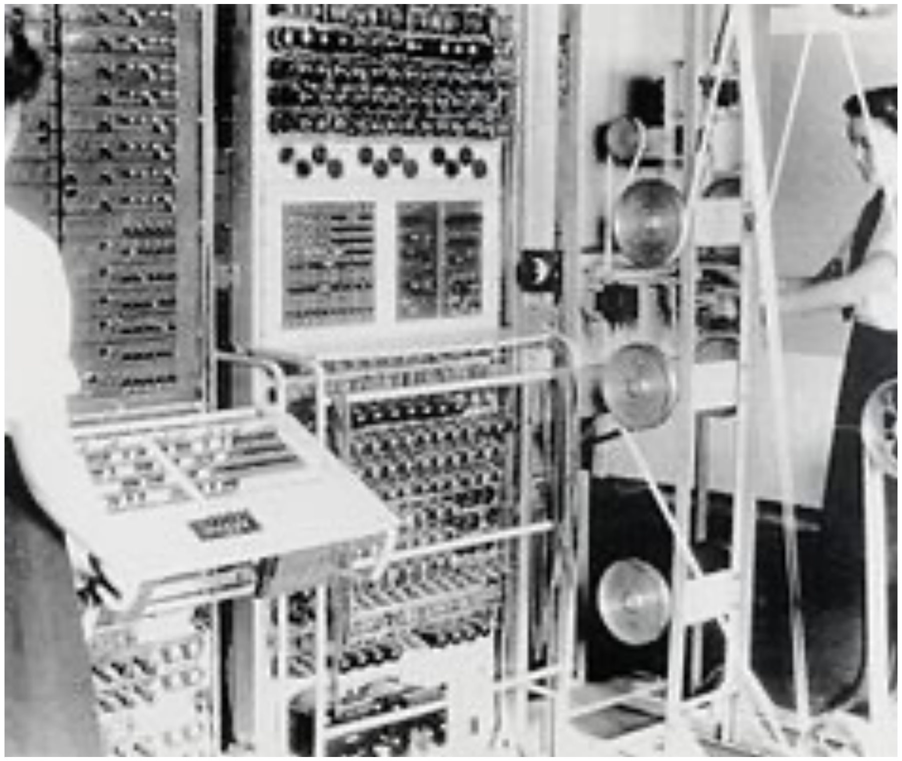


Colossus by Alan Turing, 1943



Electrical vs. Integrated Circuits

Colossus by Alan Turing, 1943

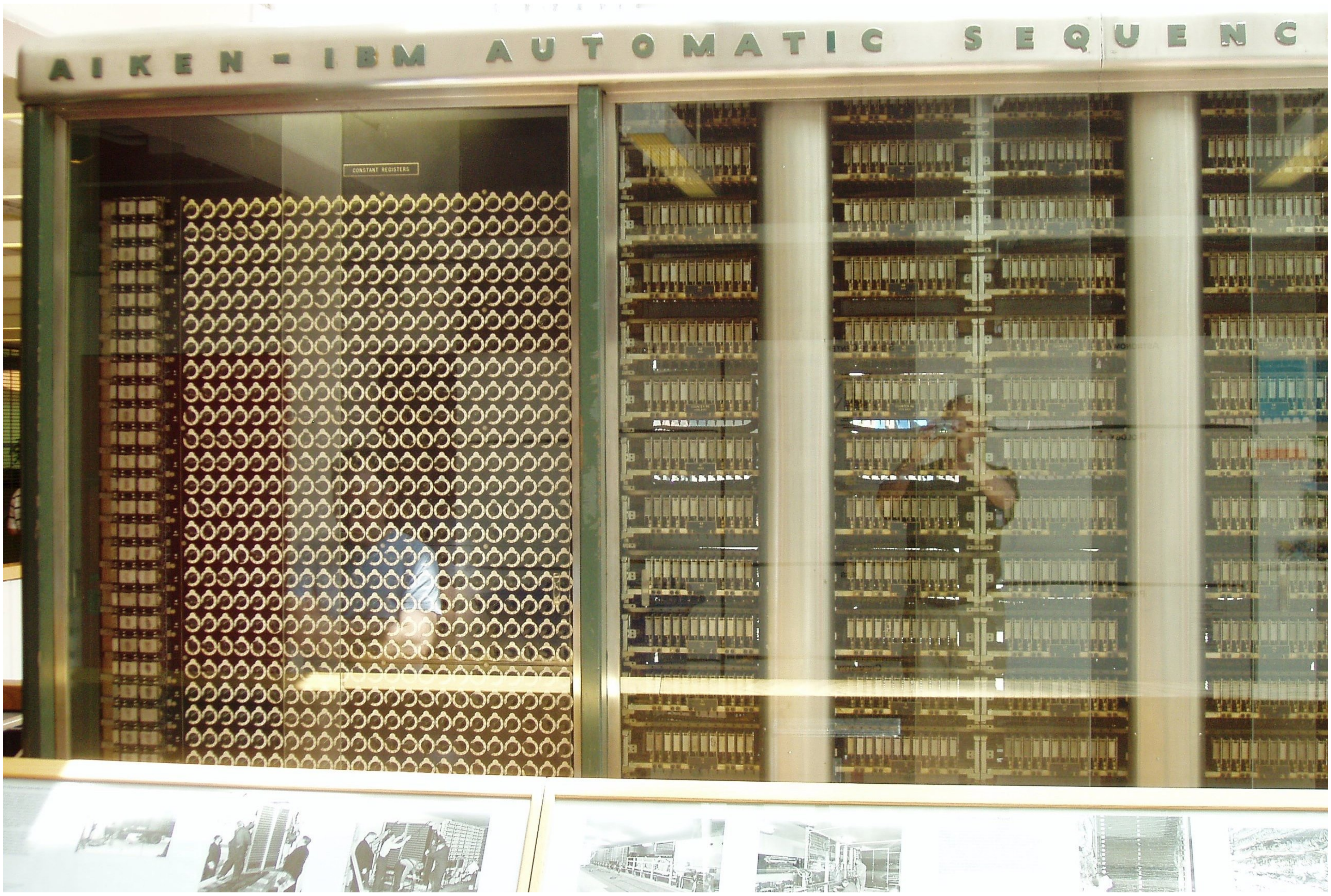


7070 by IBM, 1958

Early User Interfaces

UI CS 349

Dials, Knobs, and Lights (until 1940s)



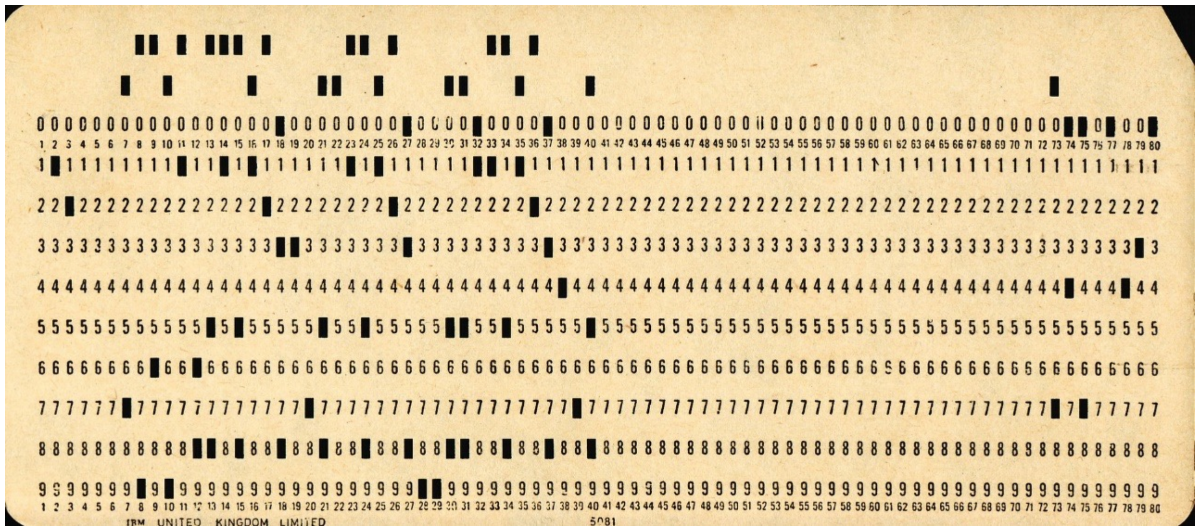
Punch Cards and Batch Interface (1945 – 1965+)

Interaction style

- Set of prepared instructions fed to computer via punch cards, paper tape, or magnetic tape
- Response typically received at the end via paper printout
- No real interaction possible while system executes instructions
- Responses received in hours or days

Users

- Highly trained individuals



Command-Line Interfaces (CLIs)

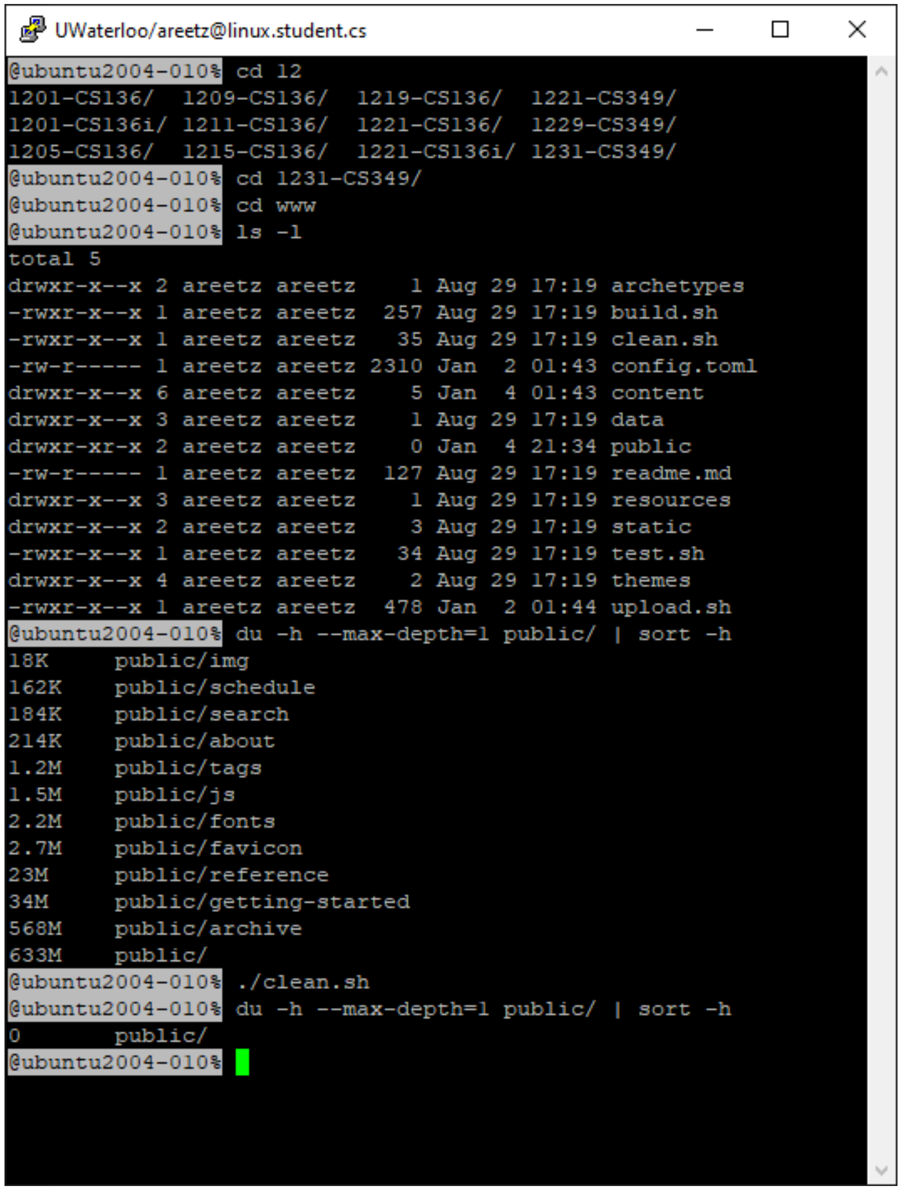
Command-Line Interface (1965 – 1985+)

Interaction style

- Commands are typed out via keyboard
- Feedback via screen, oftentimes given during execution
- Feedback received within seconds or minutes

Users

- Trained experts


UWaterloo/areetz@linux.student.cs

```
@ubuntu2004-010% cd 12
1201-CS136/  1209-CS136/  1219-CS136/  1221-CS349/
1201-CS136i/ 1211-CS136/  1221-CS136/  1229-CS349/
1205-CS136/  1215-CS136/  1221-CS136i/ 1231-CS349/
@ubuntu2004-010% cd 1231-CS349/
@ubuntu2004-010% cd www
@ubuntu2004-010% ls -l
total 5
drwxr-x--x 2 areetz areetz    1 Aug 29 17:19 archetypes
-rwxr-x--x 1 areetz areetz  257 Aug 29 17:19 build.sh
-rwxr-x--x 1 areetz areetz   35 Aug 29 17:19 clean.sh
-rw-r----- 1 areetz areetz 2310 Jan  2 01:43 config.toml
drwxr-x--x 6 areetz areetz    5 Jan  4 01:43 content
drwxr-x--x 3 areetz areetz    1 Aug 29 17:19 data
drwxr-xr-x 2 areetz areetz    0 Jan  4 21:34 public
-rw-r----- 1 areetz areetz  127 Aug 29 17:19 readme.md
drwxr-x--x 3 areetz areetz    1 Aug 29 17:19 resources
drwxr-x--x 2 areetz areetz    3 Aug 29 17:19 static
-rwxr-x--x 1 areetz areetz   34 Aug 29 17:19 test.sh
drwxr-x--x 4 areetz areetz    2 Aug 29 17:19 themes
-rwxr-x--x 1 areetz areetz  478 Jan  2 01:44 upload.sh
@ubuntu2004-010% du -h --max-depth=1 public/ | sort -h
18K     public/img
162K    public/schedule
184K    public/search
214K    public/about
1.2M    public/tags
1.5M    public/js
2.2M    public/fonts
2.7M    public/favicon
23M     public/reference
34M     public/getting-started
568M    public/archive
633M    public/
@ubuntu2004-010% ./clean.sh
@ubuntu2004-010% du -h --max-depth=1 public/ | sort -h
0       public/
@ubuntu2004-010% 
```

Command-Line Interface – Advantages

Powerful and highly flexible

- Many combinations of options: `chmod –[aAbBcCdDfFgG...]`
- Piping from output to input: `ls -a -l | more`
- Batching, macroing: `#!/usr/bin/env bash`

Built-in documentation / man-pages: `man ls`

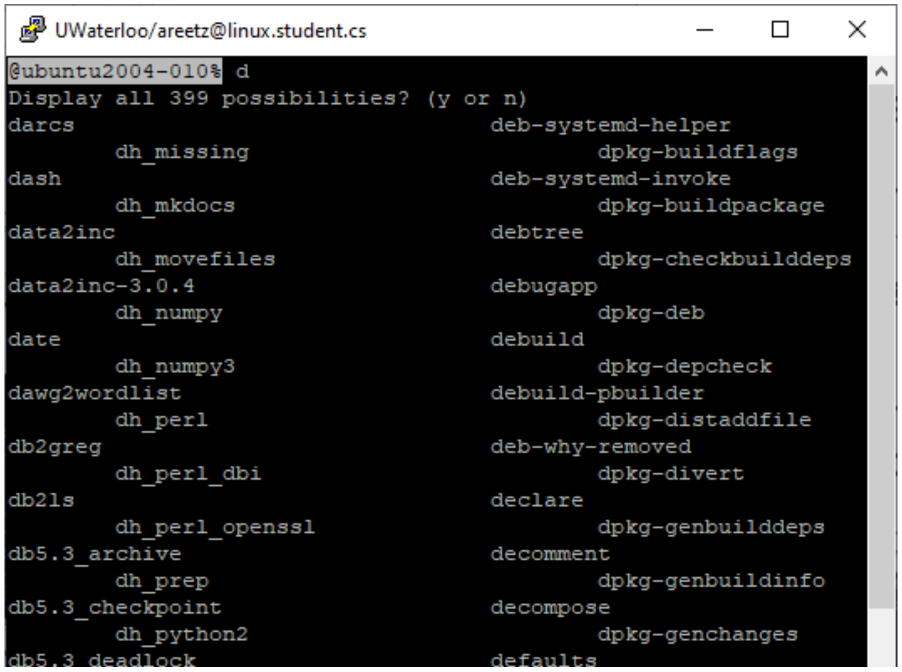
In the original Unix tradition, command-line options are single letters preceded by a single hyphen…The original Unix style evolved on slow ASR-33 teletypes that made terseness a virtue; thus the single-letter options.

— Eric Steven Raymond, The Art of Unix Programming

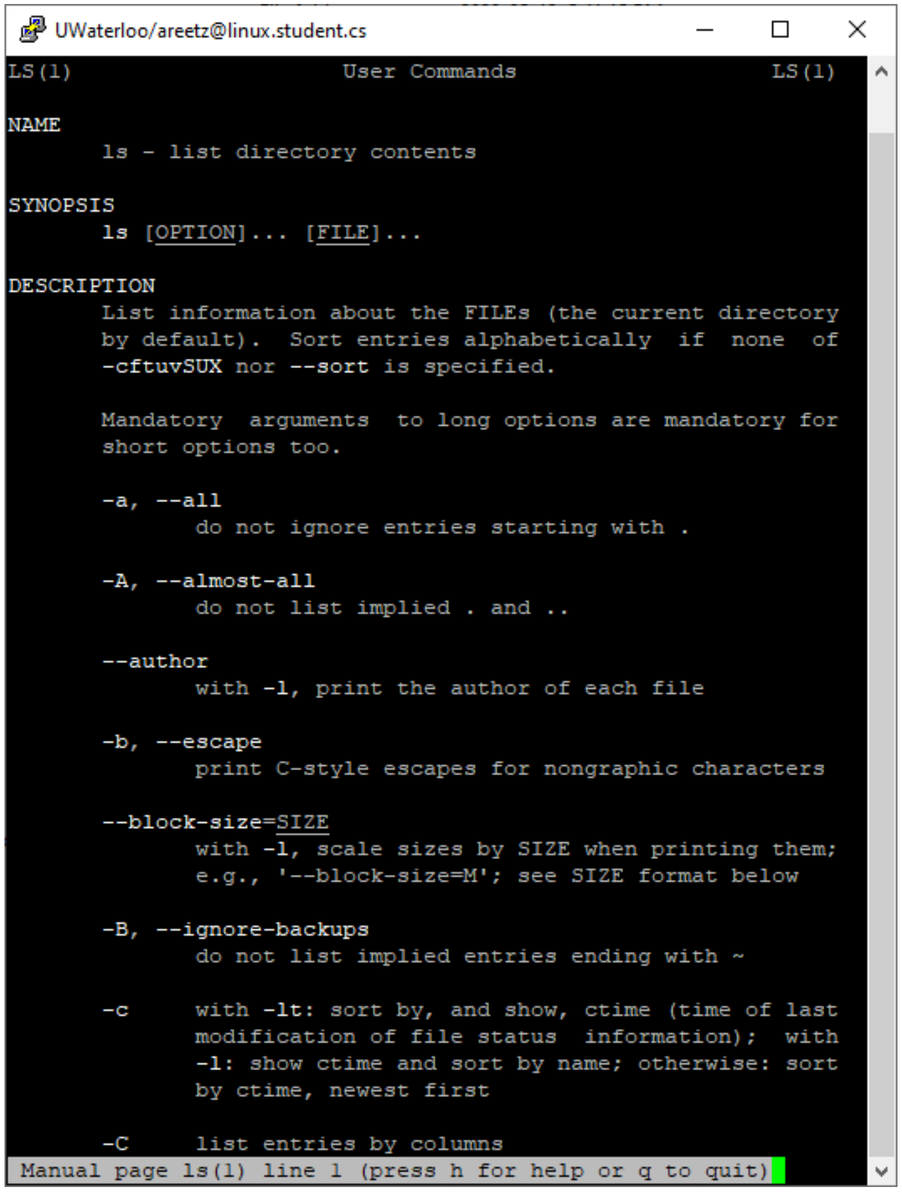
Command-Line Interface – Disadvantages

Command names and their syntax is difficult to learn and need to be **memorized.**

Command-line interfaces are **not explorable.**

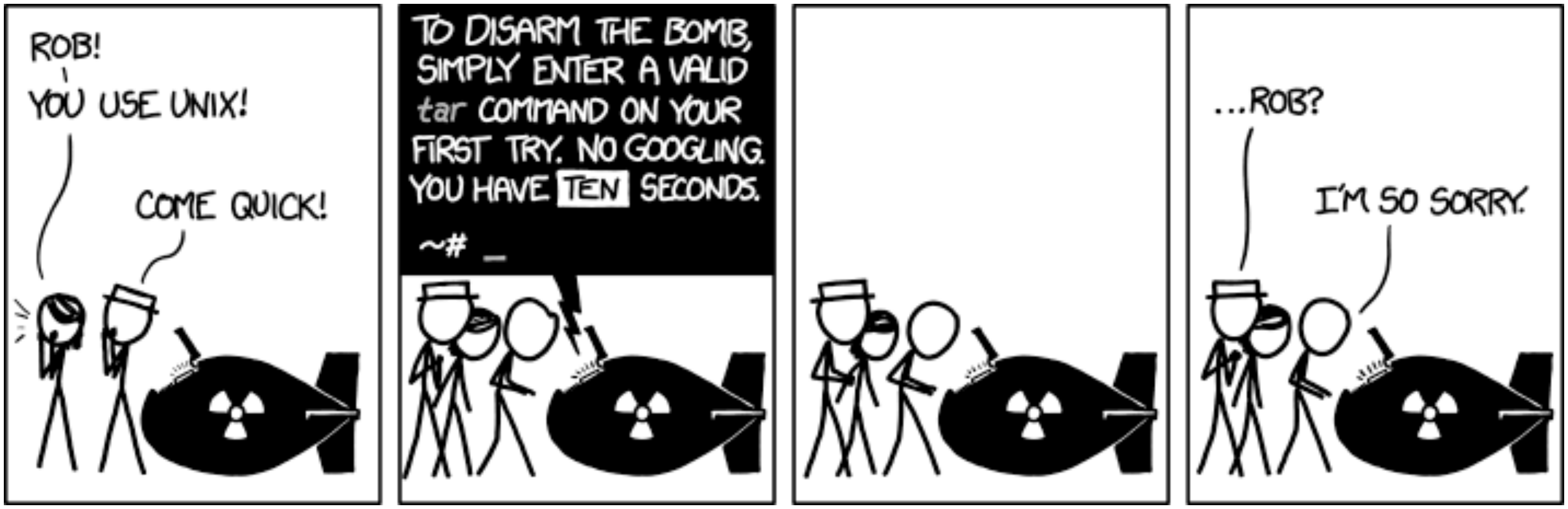

UWaterloo/areetz@linux.student.cs

```
@ubuntu2004-010% d
Display all 399 possibilities? (y or n)
darcs                                   deb-systemd-helper
        dh_missing                              dpkg-buildflags
dash                                    deb-systemd-invoke
        dh_mkdocs                               dpkg-buildpackage
data2inc                                debtree
        dh_movefiles                            dpkg-checkbuilddeps
data2inc-3.0.4                          debugapp
        dh_numpy                                dpkg-deb
date                                    debuild
        dh_numpy3                               dpkg-depcheck
dawg2wordlist                           debuild-pbuilder
        dh_perl                                 dpkg-distaddfile
db2greg                                 deb-why-removed
        dh_perl_dbi                             dpkg-divert
db2ls                                   declare
        dh_perl_openssl                         dpkg-genbuilddeps
db5.3_archive                           decomment
        dh_prep                                 dpkg-genbuildinfo
db5.3_checkpoint                        decompose
        dh_python2                              dpkg-genchanges
db5.3_deadlock                          defaults
```


UWaterloo/areetz@linux.student.cs

```
LS(1)                        User Commands                       LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List information about the FILEs (the current directory
       by default).  Sort entries alphabetically  if  none  of
       -cftuvSUX nor --sort is specified.

       Mandatory  arguments  to long options are mandatory for
       short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              with -l, scale sizes by SIZE when printing them;
              e.g., '--block-size=M'; see SIZE format below

       -B, --ignore-backups
              do not list implied entries ending with ~

       -c     with -lt: sort by, and show, ctime (time of last
              modification of file status  information);  with
              -l: show ctime and sort by name; otherwise: sort
              by ctime, newest first

       -C     list entries by columns
 Manual page ls(1) line 1 (press h for help or q to quit)
```

Command-Line Interface – Disadvantages


ROB!
YOU USE UNIX!
COME QUICK!
TO DISARM THE BOMB, SIMPLY ENTER A VALID tar COMMAND ON YOUR FIRST TRY. NO GOOGLING. YOU HAVE TEN SECONDS.
~# _
...ROB?
I'M SO SORRY.

tar by XKCD, https://xkcd.com/1168/

Command-Line Interface – Conclusion

CLIs can be highly efficient for trained users but at the cost of

- being difficult to learn to use and
- being almost completely non-explorable.

They, however,

- Are biased towards expert users, intimidating for non-expert use
- Require recall of commands rather than recognition of capabilities

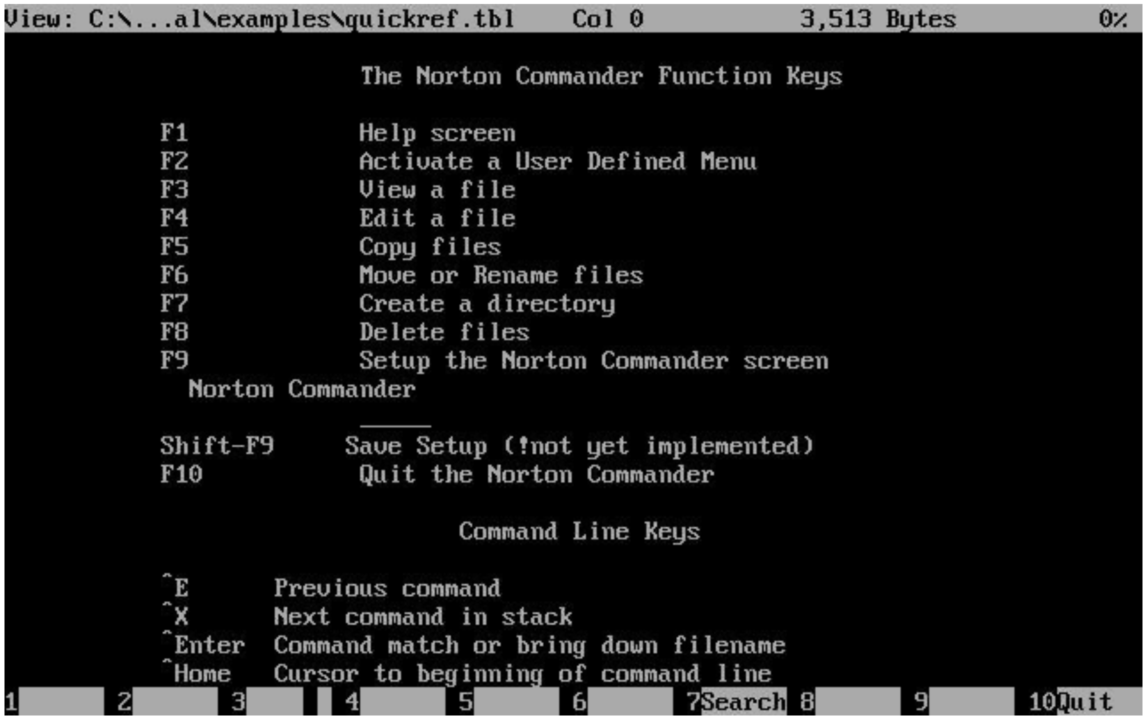
Text-based User Interface

Interaction style

- Commands are issued via keyboard shortcuts or arrow keys

Users

- Trained experts

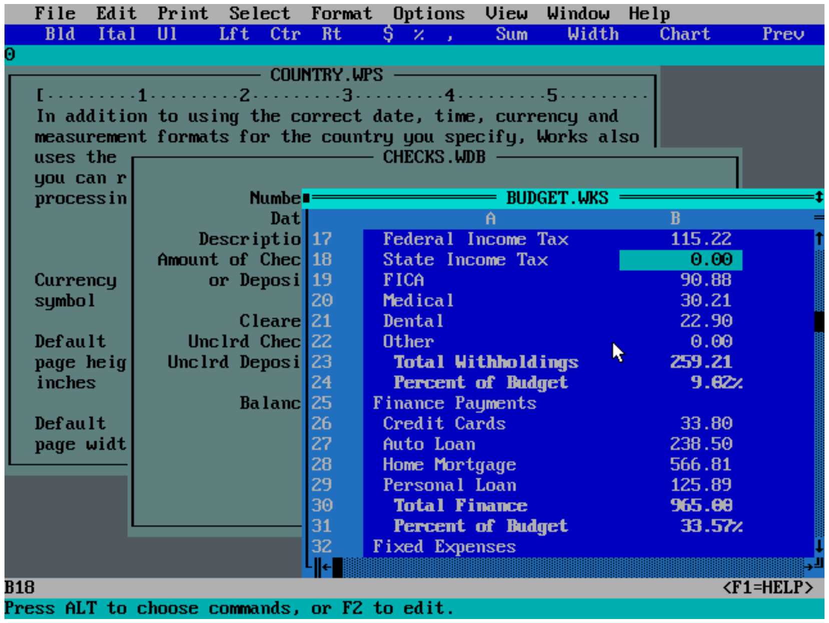


```
View: C:\...al\examples\quickref.tbl     Col 0           3,513 Bytes          0%

                         The Norton Commander Function Keys

           F1            Help screen
           F2            Activate a User Defined Menu
           F3            View a file
           F4            Edit a file
           F5            Copy files
           F6            Move or Rename files
           F7            Create a directory
           F8            Delete files
           F9            Setup the Norton Commander screen
             Norton Commander
                         _____
           Shift-F9     Save Setup (!not yet implemented)
           F10           Quit the Norton Commander

                                Command Line Keys

           ^E      Previous command
           ^X      Next command in stack
           ^Enter  Command match or bring down filename
           ^Home   Cursor to beginning of command line
1       2       3       4       5       6       7Search 8       9       10Quit
```

Norton Commander 1.0 (1986)

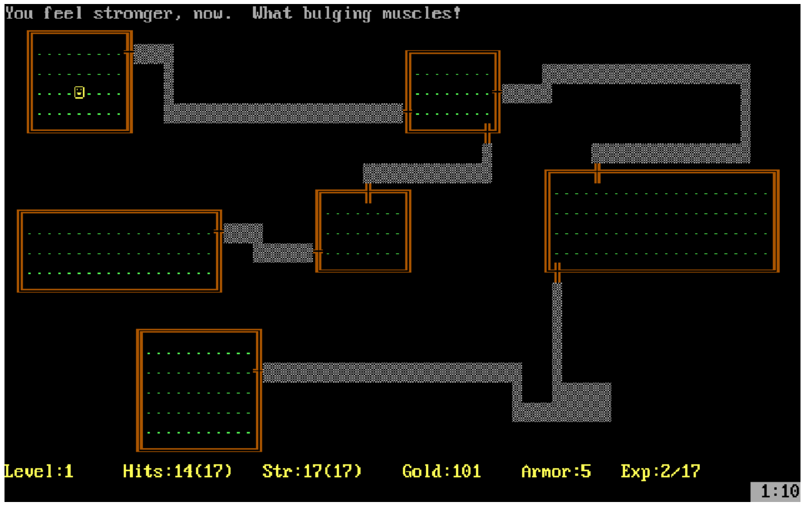
Text-based User Interface – Advantages

- Explorable
- Possible to provide graphical appearance to a text-based application



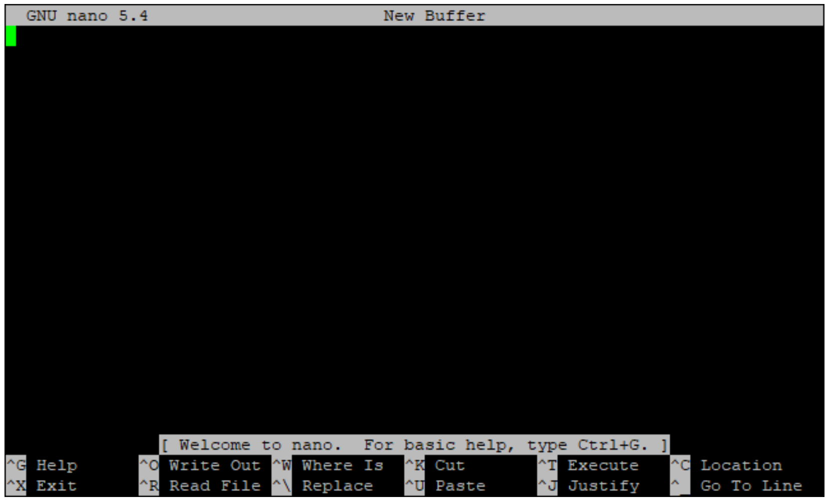
```
 File  Edit  Print  Select  Format  Options  View  Window  Help
  Bld  Ital  Ul    Lft  Ctr  Rt    $  %  ,    Sum    Width    Chart      Prev
0
                                COUNTRY.WPS
 [········1·········2·········3·········4·········5·········
 In addition to using the correct date, time, currency and
 measurement formats for the country you specify, Works also
 uses the                                  CHECKS.WDB
 you can r
 processin                 Numbe                BUDGET.WKS
                             Dat                     A                          B
                     Descriptio 17   Federal Income Tax              115.22
                  Amount of Chec 18   State Income Tax                  0.00
                     or Deposi 19   FICA                             90.88
 Currency                      20   Medical                          30.21
 symbol                        21   Dental                           22.90
                         Cleare 22   Other                             0.00
 Default            Unclrd Chec 23     Total Withholdings            259.21
 page heig         Unclrd Deposi 24     Percent of Budget              9.02%
 inches                        25  Finance Payments
                         Balanc 26   Credit Cards                     33.80
 Default                       27   Auto Loan                       238.50
 page widt                     28   Home Mortgage                   566.81
                               29   Personal Loan                   125.89
                               30     Total Finance                 965.00
                               31     Percent of Budget             33.57%
                               32  Fixed Expenses
B18                                                                  <F1=HELP>
Press ALT to choose commands, or F2 to edit.
```

Word 3.0 (1986)



```
You feel stronger, now.  What bulging muscles!

Level:1     Hits:14(17)   Str:17(17)    Gold:101     Armor:5    Exp:2/17
                                                                    1:10
```

Rogue (1980)



```
  GNU nano 5.4                      New Buffer

                  [ Welcome to nano.  For basic help, type Ctrl+G. ]
^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^_ Go To Line
```

GNU nano 5.4 (2022: 7.1)

Graphical User Interfaces (GUIs) and the WIMP paradigm

UI CS 349

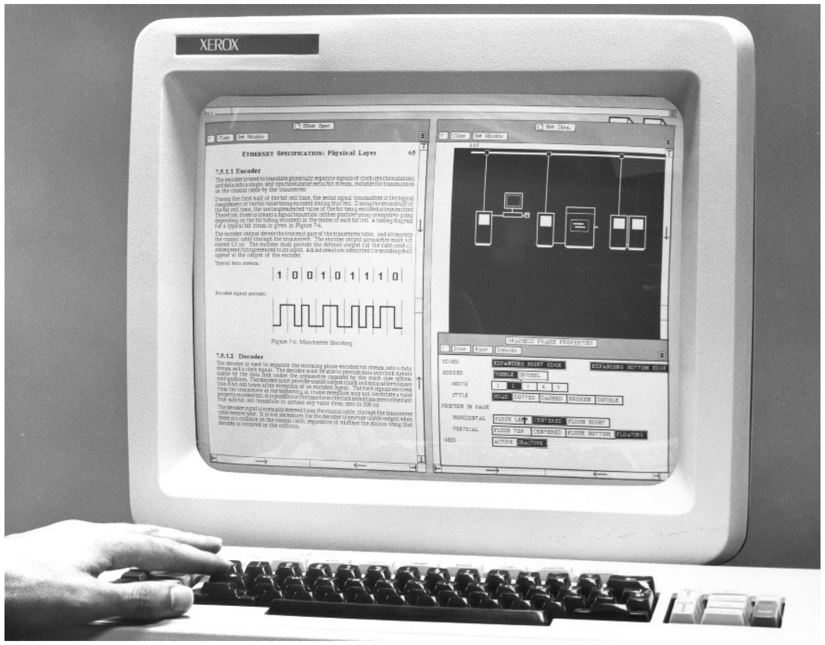
Graphical User Interfaces

Xerox Alto (1973): first computer with a GUI

- Actions are input via keyboard and mouse
- GUI consisted of windows, icons and menus; files and folders; thus introducing the “desktop” metaphor

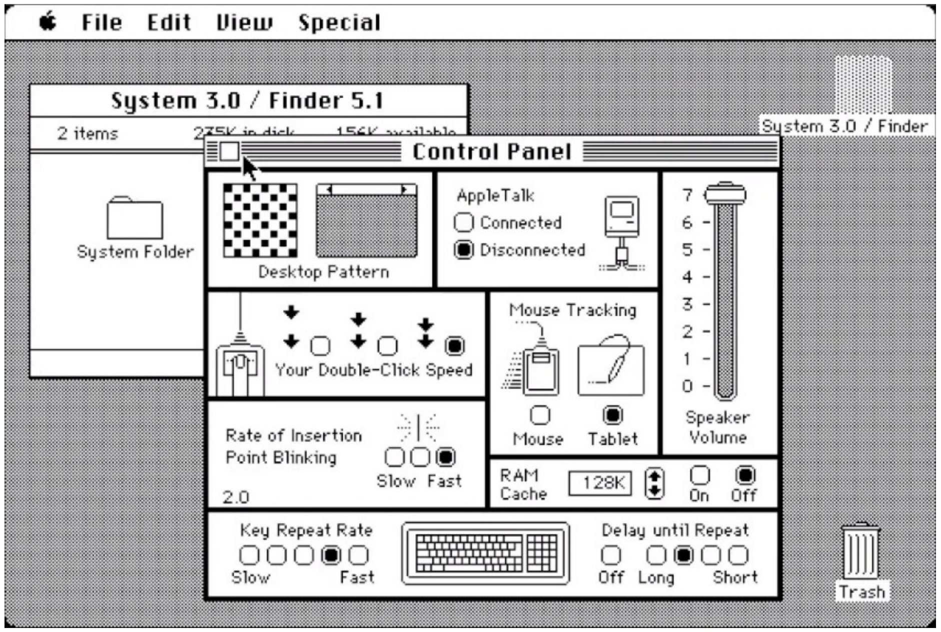
Graphical User Interfaces

Xerox 8010 Information System (1981): second computer with a GUI



Graphical User Interfaces

Apple Macintosh (1984): first commercially successful computer with a GUI



System 3.0 / Finder 5.1

2 items 275K in disk 156K available

Control Panel

Desktop Pattern

AppleTalk ○ Connected [✓ ● Disconnected]

Speaker Volume: 7 6 5 4 3 2 1 0

Your Double-Click Speed

Mouse Tracking: ○ Mouse [✓ ● Tablet]

Rate of Insertion Point Blinking: Slow Fast

2.0

RAM Cache 128K: ○ On [✓ ● Off]

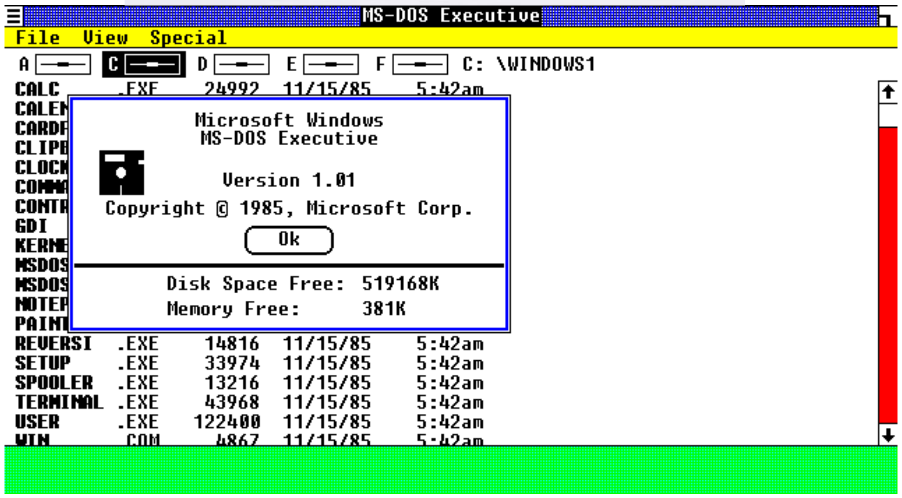
Key Repeat Rate: Slow Fast

Delay until Repeat: Off Long Short

Apple System 3.0 (1986)

Graphical User Interfaces

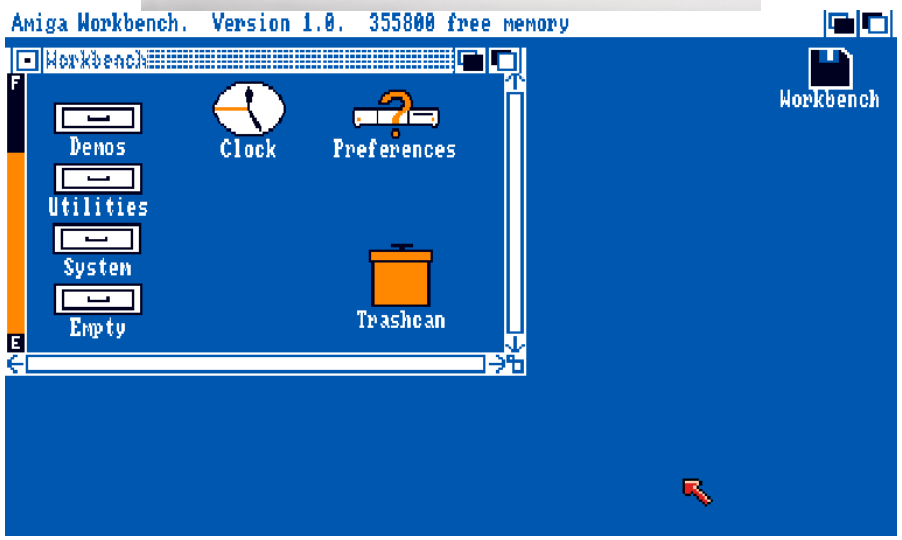
Microsoft Windows on IBM PC compatible (here: HP150II, 1985)



```
MS-DOS Executive
File  View  Special
A  C  D  E  F  C: \WINDOWS1
CALC      .EXE    24992  11/15/85    5:42am
CALEN
CARDF
CLIPB
CLOCK
COMMA
CONTR
GDI
KERNE
MSDOS
MSDOS
NOTEP
PAINT
REVERSI   .EXE    14816  11/15/85    5:42am
SETUP     .EXE    33974  11/15/85    5:42am
SPOOLER   .EXE    13216  11/15/85    5:42am
TERMINAL  .EXE    43968  11/15/85    5:42am
USER      .EXE   122400  11/15/85    5:42am
WIN       .COM     4867  11/15/85    5:42am
```

```
Microsoft Windows
MS-DOS Executive

Version 1.01
Copyright © 1985, Microsoft Corp.
Ok

Disk Space Free:  519168K
Memory Free:         381K
```

AmigaOS on the Amiga 1000 (1985)



Amiga Workbench. Version 1.0. 355800 free memory

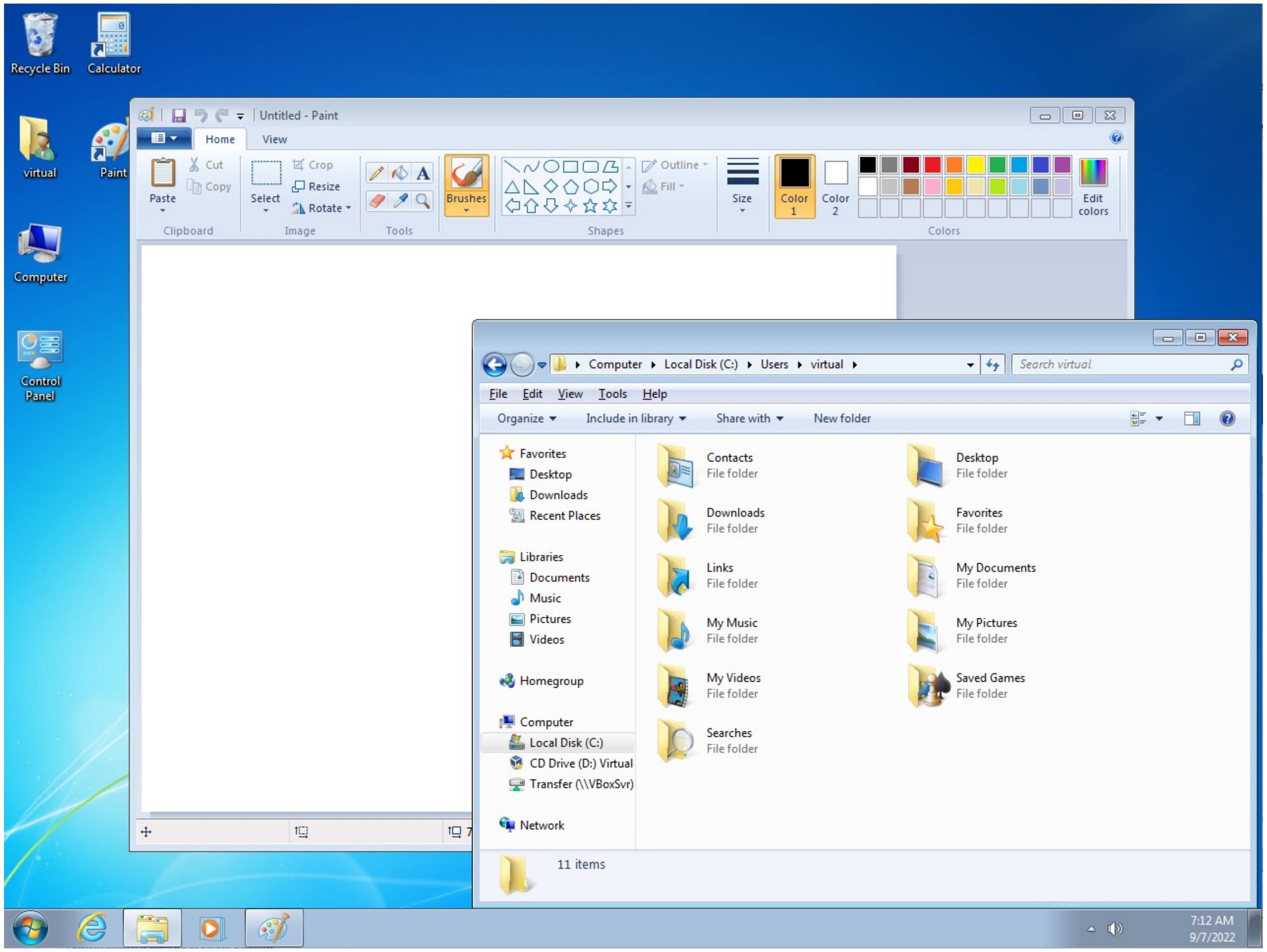
Graphical User Interfaces – Requirements

Today, we would consider the following as requirements for a GUI:

- Screen capable of graphics output
- Keyboard (mechanical, touchscreen, etc.)
- Pointing device (mouse, touchpad, graphics tablet, etc.)

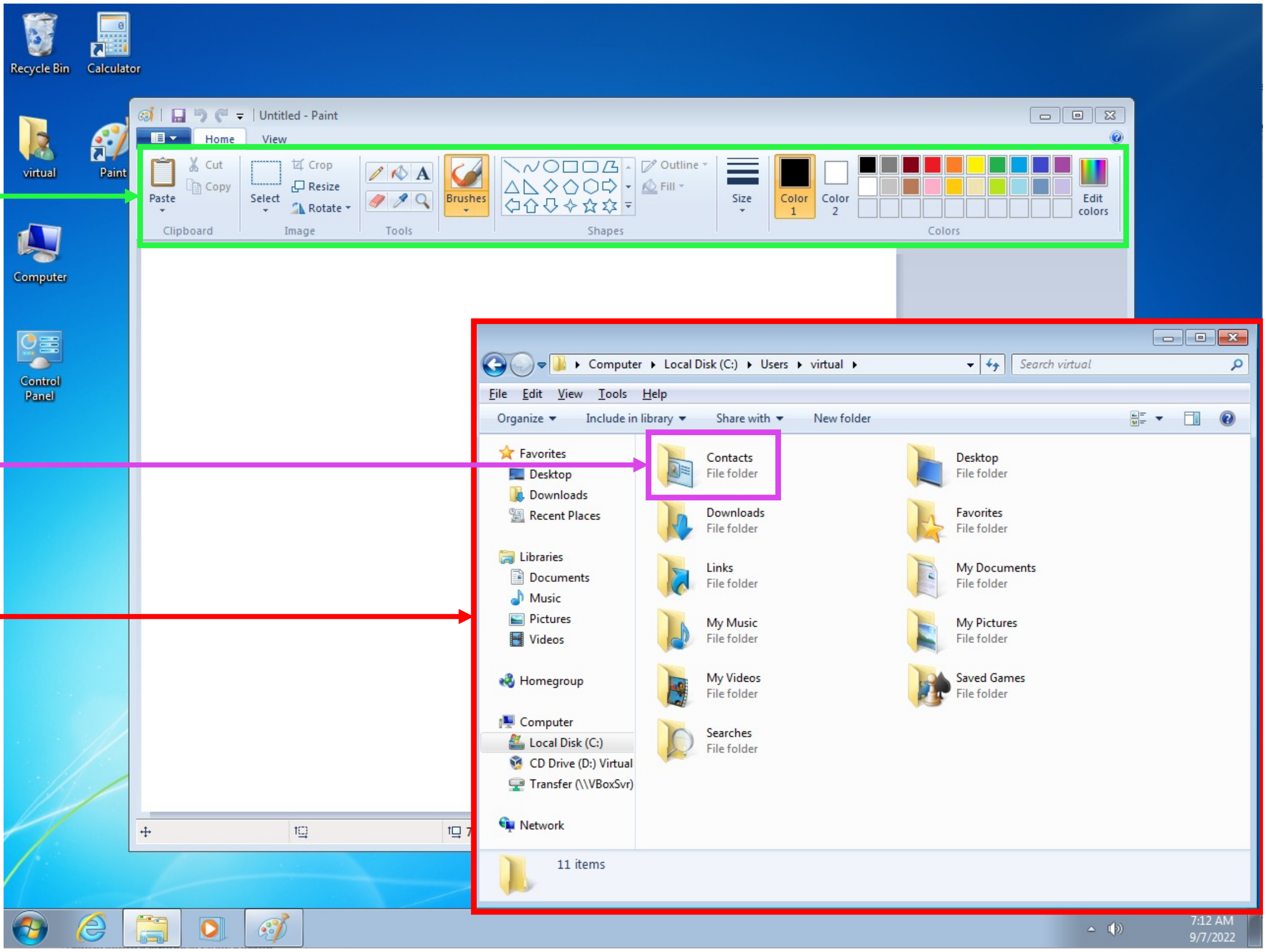
The WIMP paradigm

Almost all current GUIs follow the “WIMP”-paradigm (Windows, Icons, Menus, Pointers).



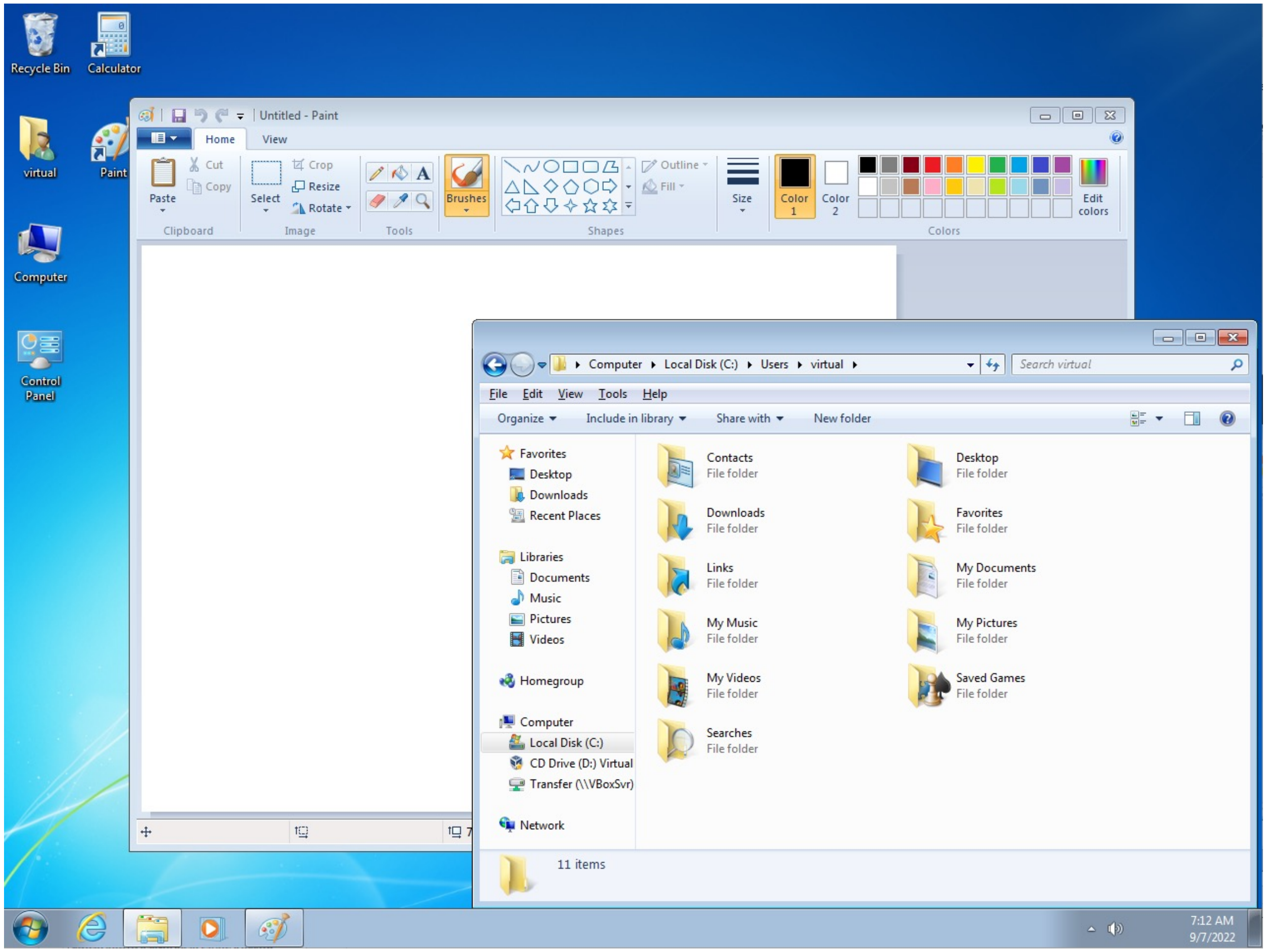
The WIMP paradigm

- Windows
- Icons
- Menus
- Pointers



The WIMP paradigm

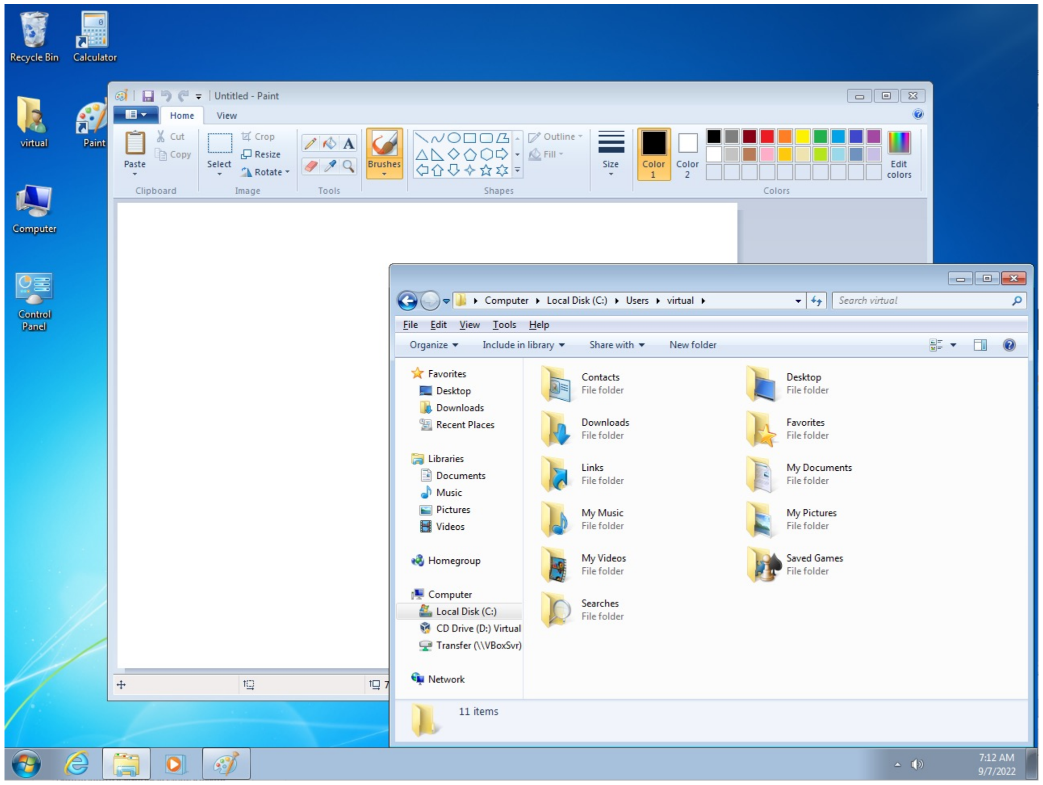
It is usually associated with the “desktop” metaphor and often includes a “Desktop” (or background).



The WIMP paradigm

In addition to standard GUI capabilities:

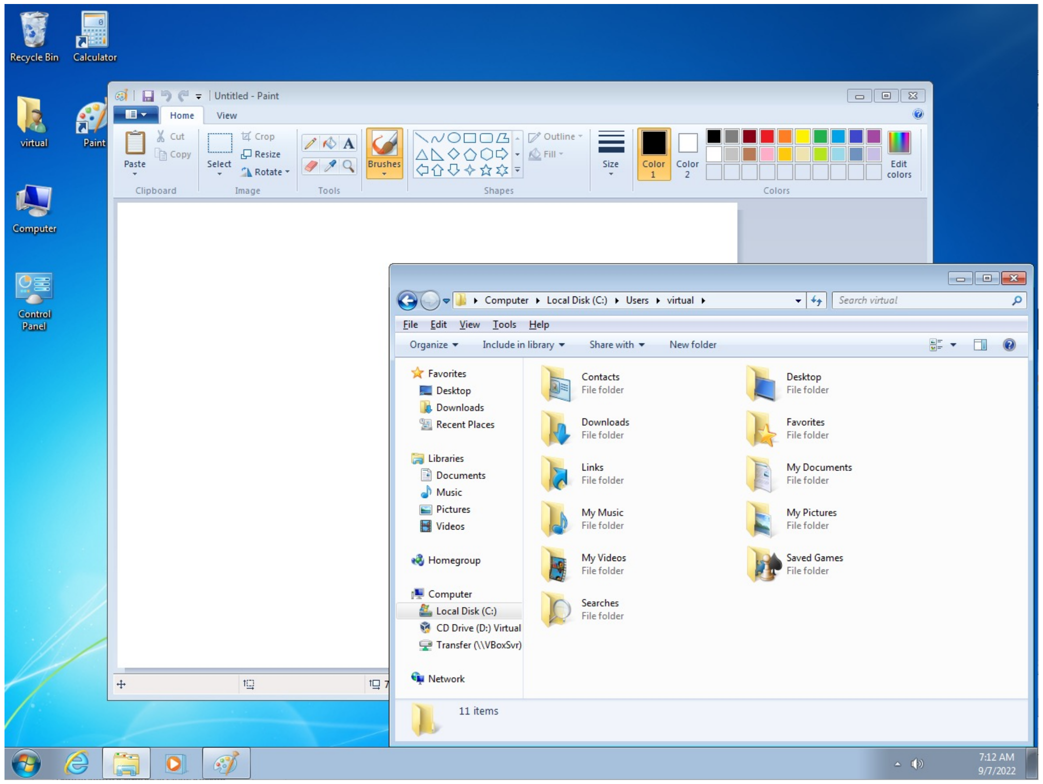
- Each application is isolated within its own window(s).
- System has methods to move, resize, re-order, re-draw windows.
- System supports common presentation of applications / elements.
- Provide common GUI elements for building apps (e.g., buttons).
- Emphasizes recognition of interface features over recall of commands.



The WIMP paradigm

Windows are independent of one another:

- They do not need to know where they are located on the screen or what other apps are running
- They can be spatially re-arranged to facilitate viewing and manipulating data from multiple sources
- Input and output is directed to a specific window

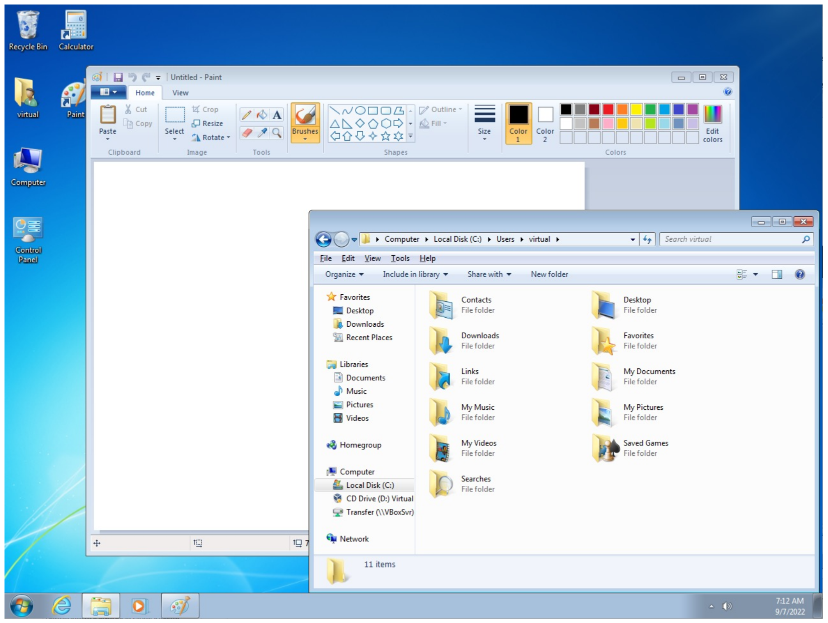


GUI Interfaces

For the remainder of the course, we will focus on building GUI interfaces.

Characteristics and principles underlying them are applicable to any OS with a graphical interface.

- Windows
- macOS
- Linux



We apply these principles to build desktop interfaces (point-click with a mouse). We will also discuss how to modify and apply our approach to building mobile interfaces (touch to interact).

Next class: we'll talk about Kotlin.

End of the Chapter

Main take-aways:

- Understand the advantages and disadvantages of CLIs and GUIs.
- Remember the components of WIMP.

Any further questions?