

# **INTERFETE OM-MASINA**

## **- CURS 2**

O mașină care nu oferă suport pentru multitasking la un moment dat poate rula o singură aplicație,  
Mașinile și sistemele de operare care asigură suport pentru multitasking mai multe aplicații pot fi rulate simultan.

În acest din urmă caz apar următoarele probleme: Care aplicație va afișa rezultatele pe monitorul calculatorului?  
Când și cum poate utilizatorul controla acest lucru?

Sistemul de gestiune a ferestrelor - SGF


Sistemul de gestiune a ferestrelor (SGF) este cel care asigură organizarea ecranului în regiuni dreptunghiulare distincte, numite ferestre. El nu este altceva decât un soft ce-l ajută pe utilizator să monitorizeze și să controleze diferite contexte separându-le fizic, afișându-le pe unul sau mai multe ecrane sau ferestre.



Funcționalitatea ferestrei este împărțită pe două nivele, identificând cele două componente ce formează un SGF:

1. sistemul de ferestre (nivelul de bază) care asigură interfața către programator pe baza modelelor de intrări și ieșiri - asigură funcționalități de bază ca afișarea/desenarea graficii ecran (modelul de ieșiri) și preluarea datelor de la utilizator (modelul de intrări)

2. managerul de fereastră (nivelul interfeței utilizator) care se ocupă cu prezentarea datelor și identificarea comenzilor - controlează toate aspectele vizibile utilizatorului. Aceasta componenta are la randul ei, doua sub-componente:

- a. Subcomponenta de prezentare (a datelor de ieșire)
  - b. Subcomponenta de identificare a comenzilor (pe baza datelor de intrare).
- 

# MODELUL DE IEȘIRI

Modelul de ieșiri este constituit din setul de proceduri pe care o aplicație le poate folosi pentru a afișa imagini pe ecran. Este important ca toate ieșirile să fie direcționate către sistemul de ferestre astfel încât primitivele grafice să poată "decupa" (clipping) imaginea ce va fi afișată într-o fereastră. De exemplu dacă un program trasează un segment ale cărui capete ar trece dincolo de marginile ferestrei, trebuie avut grijă ca la afișarea acestuia în fereastra țintă conținutul celorlalte ferestre independente, active pe ecran, să nu fie alterat. Majoritatea SGF-urilor oferă căi speciale ce permit programelor să scrie direct pe ecran, fără a folosi clipping-ul din sistem. Desigur că aceste operații pot fi mult mai rapide dar în același timp și foarte periculoase, fapt ce atrage după sine folosirea sporadică acestei alternative. Mai mult, noile plăci grafice ce apar pe piața încep să fie optimizate special pentru manipularea eficientă a sistemului de ferestre.



La primele SGF-uri, ca Smalltalk, Blit și Sapphire, principala operație de ieșire avea la bază rasterizarea și se numea BitBlt. Aceste sisteme ofereau suport pentru monitoare monocrom în care fiecare pixel putea fi de culoare alb sau negru, iar o operația BitBlt se rezuma la a prelua un dreptunghi (hartă de biți) de pe ecran și muta în altă zonă tot pe ecran, între sursă și destinație putându-se preciza o operație booleană de combinare a pixelilor. Cu ajutorul BitBlt se puteau desena dreptunghiuri solide în alb/negru, afișa text, ferestre de defilare (scrolling) și se puteau realiza câteva efecte speciale. Singura operație suplimentară oferită de aceste sisteme era desenarea segmentelor drepte.

```
BOOL BitBlt(  
    HDC hdc - handle to the destination device context.,  
    int x,  
    int y,  
    int cx,  
    int cy,  
    HDC hdcSrc - handle to the source device context.,  
    int x1,  
    int y1,  
    DWORD rop ); raster-operation code
```

Câteva sisteme au ales modelul Postscript pentru crearea de imagini ecran. Acest model permite folosirea unor coordonate sistem independente de periferic și asigură tratarea unitară a operațiilor de rotire și scalare pentru toate tipurile de obiecte, inclusiv text. Alt avantaj este dat de folosirea aceluiași limbaj pentru a imprima pe hârtie conținutul ferestrelor, majoritatea imprimantelor implementând limbajul Postscript. Sun a fost prima firmă care a folosit acest model pentru SGF-ul NeWS. Ulterior creatorul postscript-ului, firma Adobe, a lansat o versiune oficială de "Display Postscript" care a fost preluată SGF-ul NeXT-ului și folosită ca extensie a X-ului de mai mulți producători, inclusiv IBM și DEC.



## Modelul de intrări

Modelele de intrări din primele standarde grafice, nu asigurau suportul existent astăzi în sistemele moderne bazate pe manipularea directă a interfeței. Programatorul trebuia să apeleze o rutină ce solicita utilizatorului valori pentru identificarea unității virtuale, care putea fi: locator - poziția cursorului indicată de mouse, string - câmpuri de tip text editabil, choice - opțiunea unui meniu sau pick - obiectul grafic selectat, și apoi să aștepte ca utilizatorul să îndeplinească acțiunea. Acest model a fost abandonat pentru că utilizatorii cereau o interfață generală, fără moduri de lucru specifice unui anumit tip de acțiune, în care să poată decide oricând și din orice parte a programului, dacă vor să selecteze o opțiune din meniu, un obiect grafic sau doar să tasteze ceva.



## Comunicarea

Un element important într-un SGF este calea folosită pentru comunicare între sistemul de ferestre și aplicație. În X de exemplu s-a decis folosirea comunicației între procese peste un protocol de rețea. Acest lucru înseamnă că programele aplicației pot rula pe un alt calculator decât cel care gestionează ferestrele aplicației. În toate celelalte SGF-uri, operațiile sunt implementate direct prin apelul unor proceduri ale managerului de fereastră sau prin intermediul unor facilități speciale ale sistemului de operare. Avantajul principal al mecanismului oferit de X este următorul: mai multe calculatoare pot rula programele aceleiași aplicații iar ferestrele acestora sunt gestionate (și afișate) într-un singur loc, pe un alt calculator.



## Managerul de fereastră

Interfața utilizator a unui SGF reprezentată prin managerul de fereastră are rolul de a-i permite utilizatorului acestuia controlul asupra ferestrelor. În X, utilizatorul poate comuta ușor între mai multe interfețe utilizator. Pur și simplu utilizatorul oprește execuția (kill, uciderea) unui anumit manager de fereastră și lansează altul în execuție. Existența unui protocol standard prin care programele comunică cu nivelul de bază, face posibil ca programele să-și continue rularea în timp ce utilizatorul comută pe alt manager de fereastră. Este posibilă astfel rularea unor aplicații ce folosesc widgeturi Motif într-o fereastră controlată de managerul de fereastră OpenLook.



## Prezentarea

Prezentarea ferestrelor definește modul în care va arăta ecranul. Este interesant de subliniat că la acest capitol există o diferențiere clară între managerele de fereastră ce permit suprapunerea ferestrelor pe ecran și cele care nu permit acest lucru. Cele care permit suprapunerea se spune că respectă metafora "biroului de lucru", făcându-se o paralelă între o fereastră și o foaie de hârtie de pe "birou", foaie care poate acoperi/descoperi alte foi aflate pe același "birou". Deși studii efectuate în timp au arătat că managerele de fereastră care nu acceptă suprapunerea ferestrelor sunt mai eficiente, "evoluția" impusă de cerințele utilizatorilor a urmat calea alternativă, majoritatea acceptând astăzi metafora biroului de lucru.



Alt aspect important al prezentării ferestrelor este utilizarea iconurilor. Acestea sunt imagini în miniatură ce identifică ferestre sau câteodată fișiere și sunt utilizate pentru a rezolva problema minimizării ferestrelor în cazul în care numărul acestora depășește numărul care poate fi gestionat rezonabil pe ecran. Alte aspecte ale prezentării se referă la:

*existența/absența "etichetei" de titlu a unei ferestre,*

*afișarea fundalului (culoare, pattern) și reîmprospătarea zonelor ce nu mai sunt acoperite cu ferestre*

*proprietăți ce pot fi atribuite unor elemente din fereastră (bara de titlu, marginile ferestrei) prin care să se permită executarea unor operații cu ferestre (redimensionare, mutare, minimizare, maximizare, etc).*




## Comenzile

Din moment ce un calculator dispune de o singură tastatură și de un singur mouse a trebuit să se rezolve problema preluării controlului de către utilizator asupra unei ferestre anume din multitudinea celor ce pot fi afișate la un moment dat pe ecran. Fereastra în cauză, în momentul în care utilizatorul preia controlul asupra ei, este numită **fereastră de intrare (focus input)**, sau "**fereastră focalizată**". Câteva sisteme o numesc fereastră activă, sau fereastră curentă însă acești termeni nu mai sunt valabili în cazul unui sistem multiprocesor, când mai multe ferestre pot fi active, afișând simultan diverse rezultate.

Managerul de fereastră are rolul de a oferi modalități prin care să se poată specifica și evidenția care fereastră este focalizată. Opțiunile uzuale existente la acest capitol sunt:

*click-to-type, folosită de Macintosh, care înseamnă că utilizatorul trebuie să facă un clic cu mouse-ul în interiorul ferestrei pentru ca fereastra respectivă să poată prelua intrări de la tastatură;*



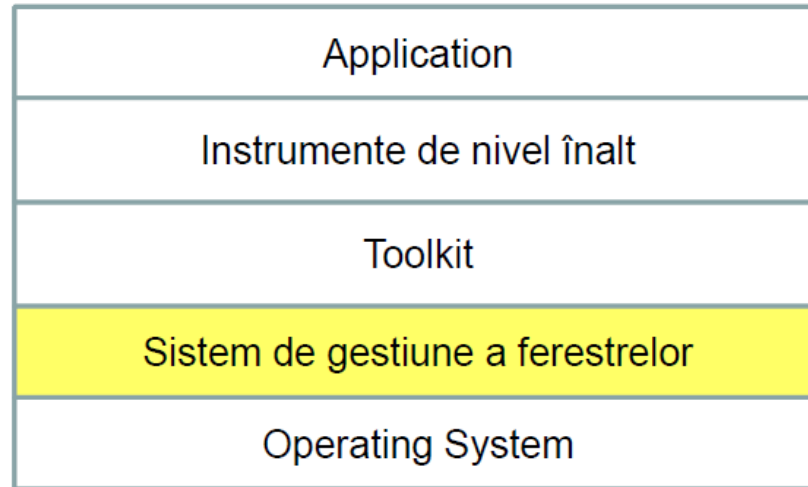
*move-to-type, care înseamnă că mouse-ul trebuie doar poziționat deasupra unei ferestre pentru ca aceasta să devină focalizată. Deși asigură o viteză sporită această alternativă poate genera ușor greșeli dacă ne gândim cât de frecvent se întâmplă să se atingă involuntar mouse-ul.*

Toate sistemele ce permit suprapunerea ferestrelor permit utilizatorului să stabilească care este fereastra din vârf (neacoperită deloc, din top) și care este fereastra acoperită de toate celelalte ferestre (bottom). Alte comenzi permit modificarea dimensiunii, mutarea, crearea sau distrugerea unei ferestre.



# Instrumente UI

- Componente ale unui sistem UI:



- Sistemul de gestiune a ferestrelor:
  - Pachete software ce ajută utilizatorii să monitorizeze și controleze diverse contexte prin separarea lor fizică în părți ale ecranului

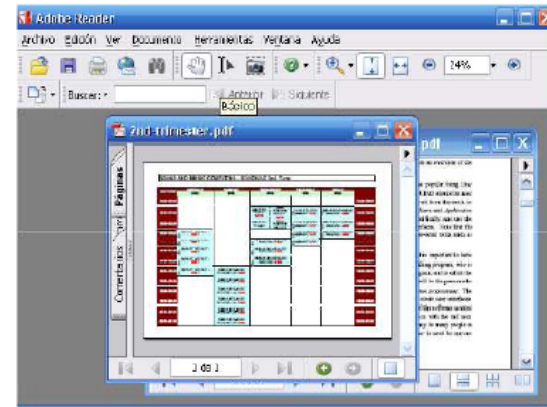
# Sistemul de gestiune a ferestrelor



- Sistemul de ferestre: rutine de bază
  - Afișează grafica (ieșire)
  - Acces la dispozitivele de intrare
- Managerul de ferestre: aspectele vizibile pentru utilizator
  - Prezentare: stilul ferestrelor
  - Comenzi: cum manipulează utilizatorul ferestrele

# Exemple gestione ferestre:

- Overlapping

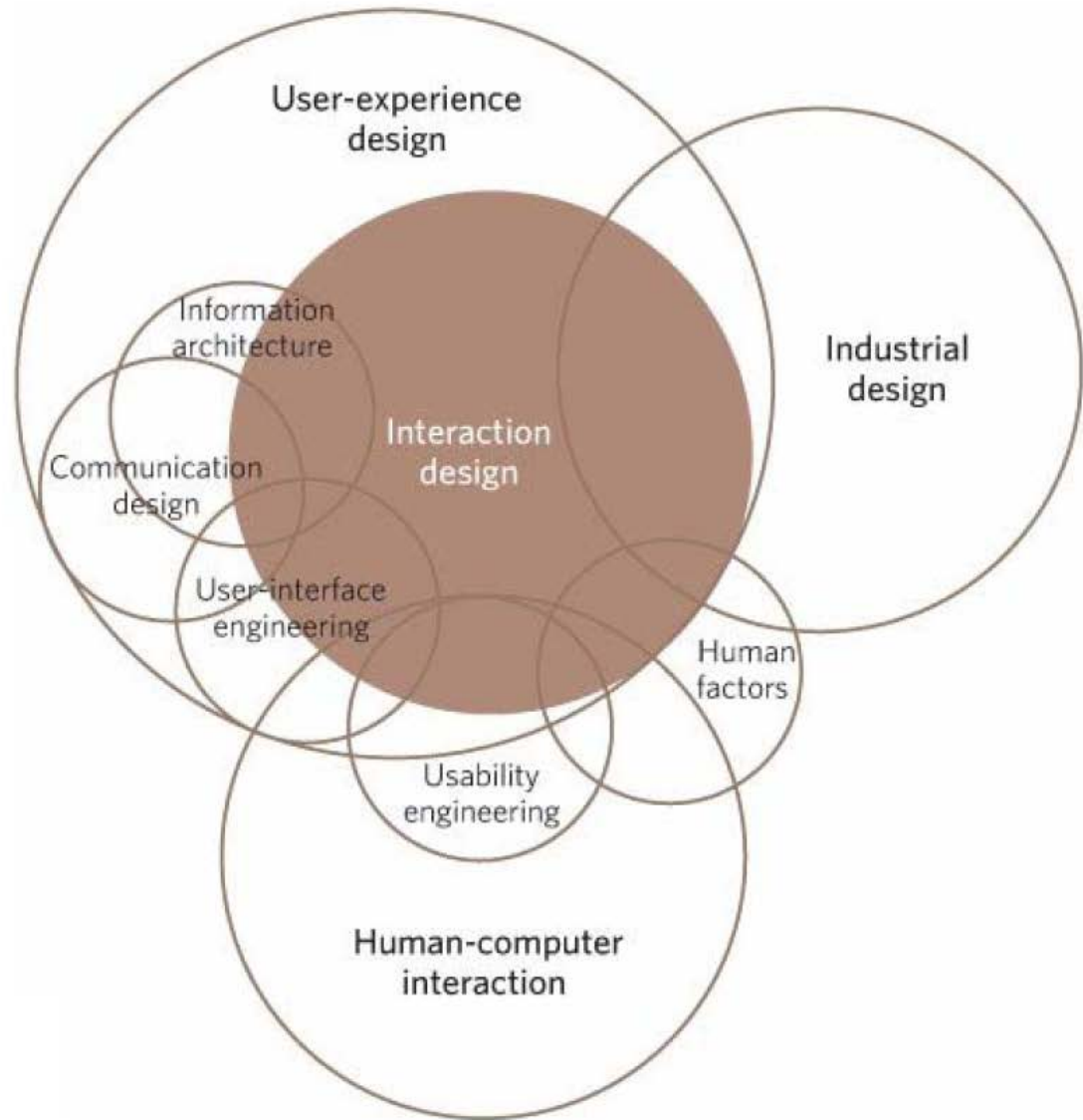


- Tiles

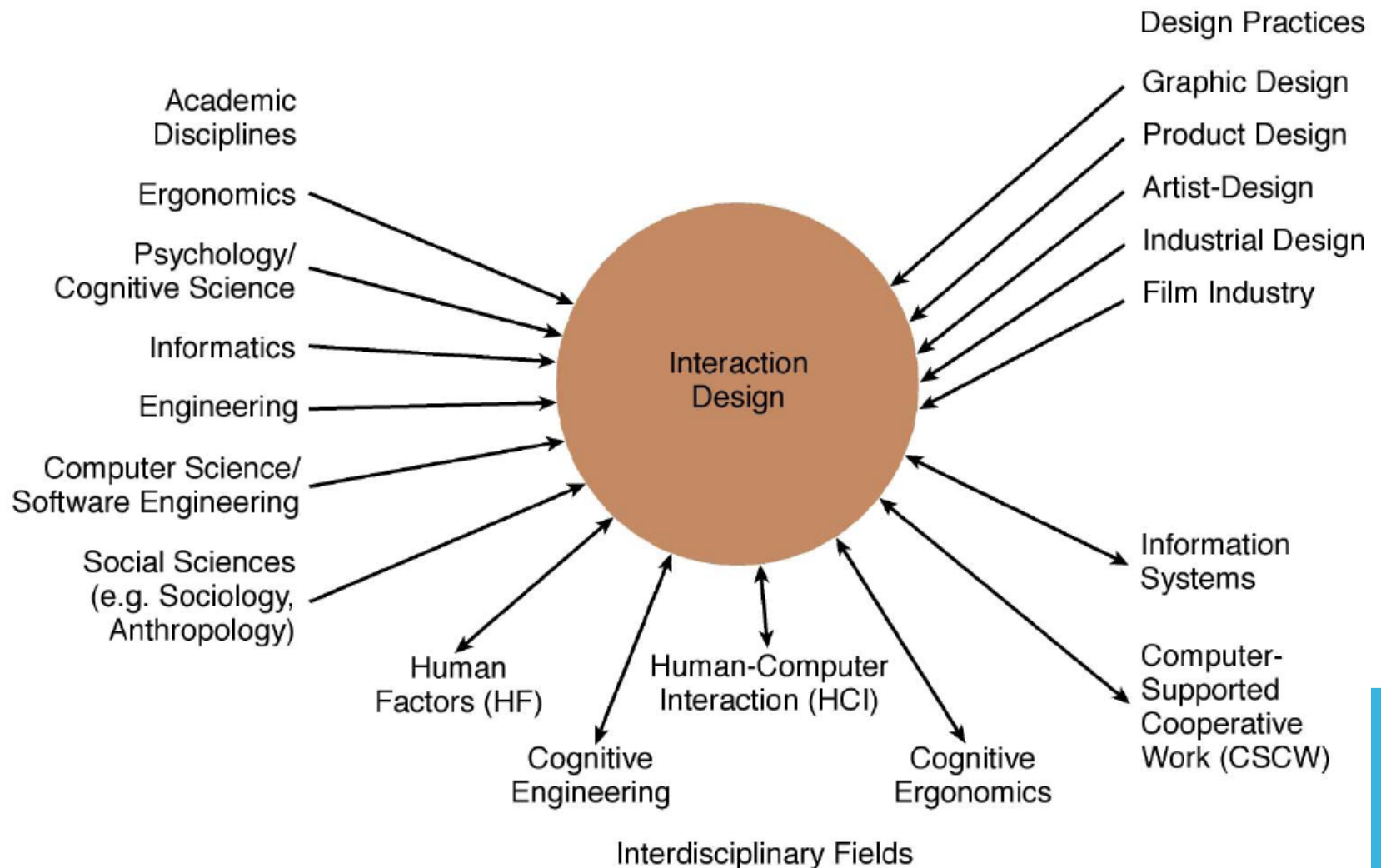




- Discipline  
conexe  
(Dan Saffer,  
2006)



# Domenii (Yvonne Rogers, 2007)



# Cum se construiește interfața?

- Proiectare precară *versus* proiectare bună
  - Ușurința manipulării
  - Aranjament logic & cromatic al butoanelor
  - Ușurința localizării butoanelor (grupate conform funcționalității)



# Scurt istoric

- **Douglas Engelbart** (Stanford, 1968) – proiectul **Augment**: interfața grafică & ferestre pe ecran, hipertext, procesor de texte, *e-mail*, teleconferințe, *script*-uri, mouse
- **Ken Thompson, Dennis Ritchie** (1969) – **UNIX** (AT&T Bell Labs.) – interfața text
- **Xerox Alto** (1973) – primul calculator cu interfață grafică: ferestre suprapuse, elemente de interfață, utilizarea mouse-ului
- 8008 Intel (1975) – calculator “personal”: Commodore, Amiga, Spectrum
- **IBM PC** (1981) cu sistemul de operare linie de comandă: DOS (Microsoft) inspirat din CP/M, apoi din UNIX
- **Xerox STAR** (1981) – interfață intuitivă, pictograme, *waste-basket*, ferestre de dialog, rezoluție 1024×768

# NLS – ONLINE SYSTEM

(60s)



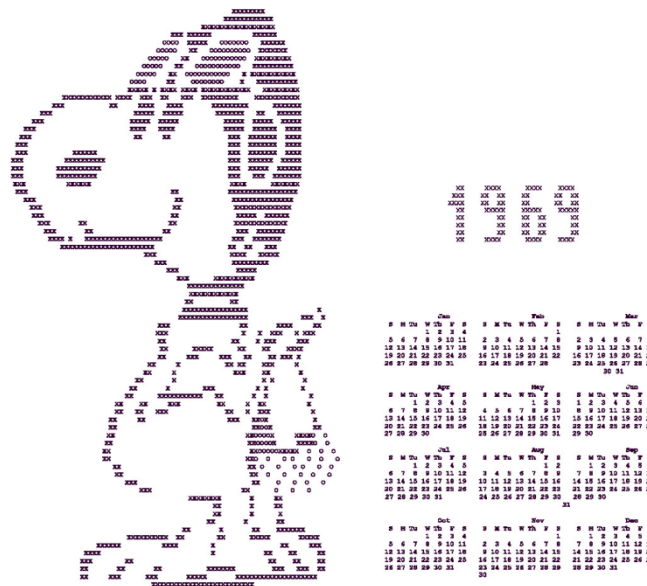
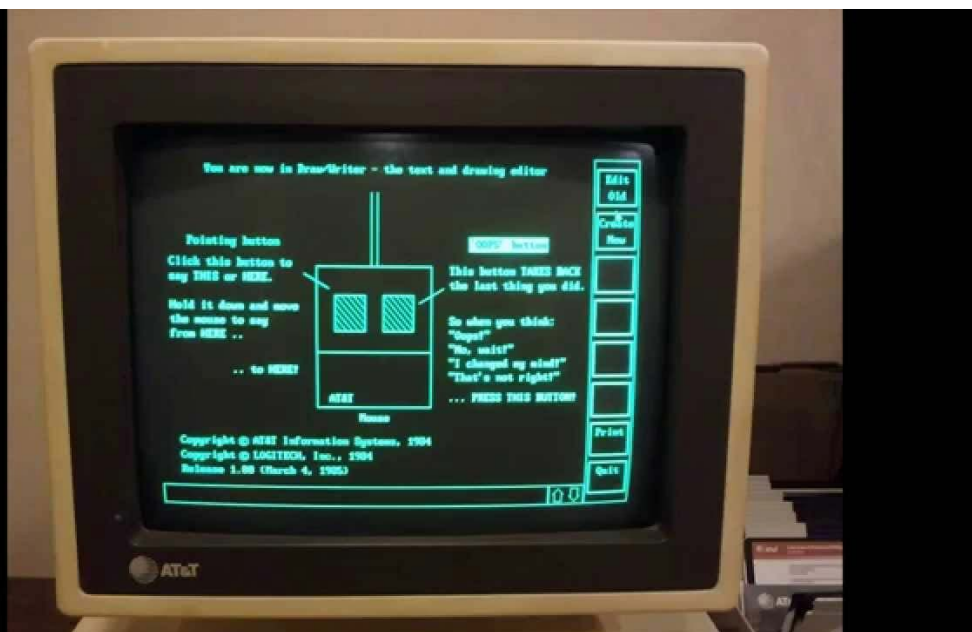
<http://www.dougenelbart.org/>

18

**Engelbart – Augment project**







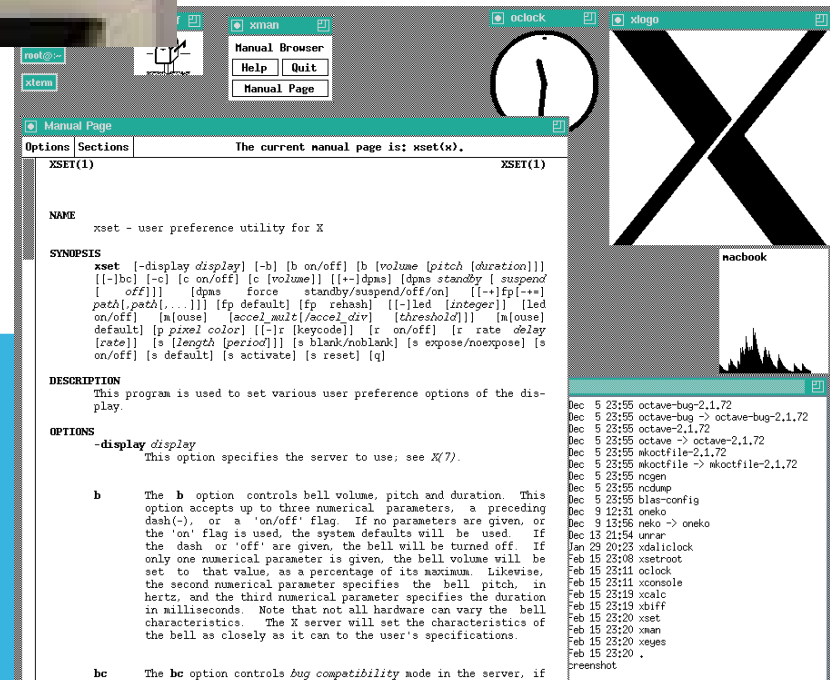
# XEROX ALTO



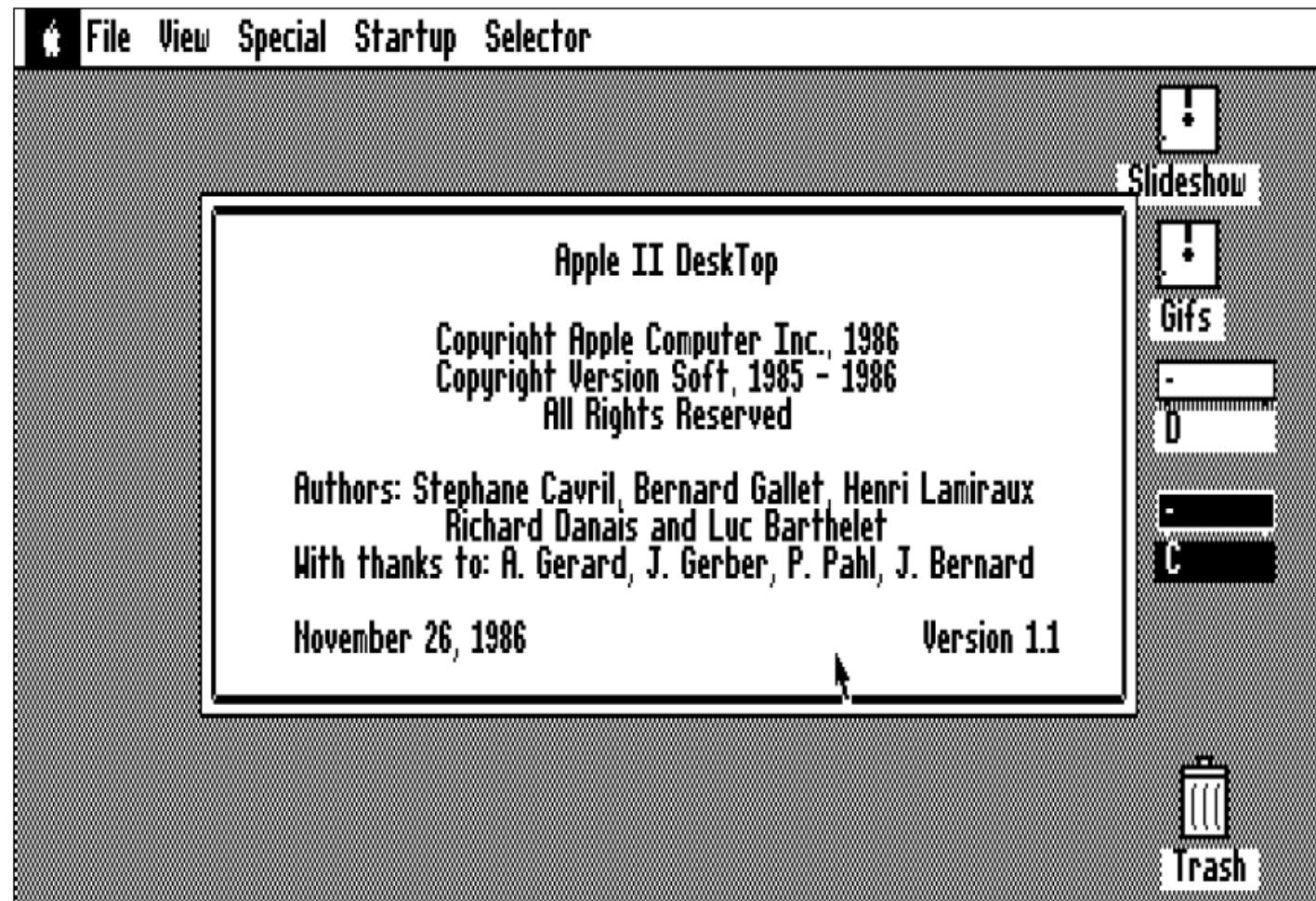
- Interfața Spectrum (1983)



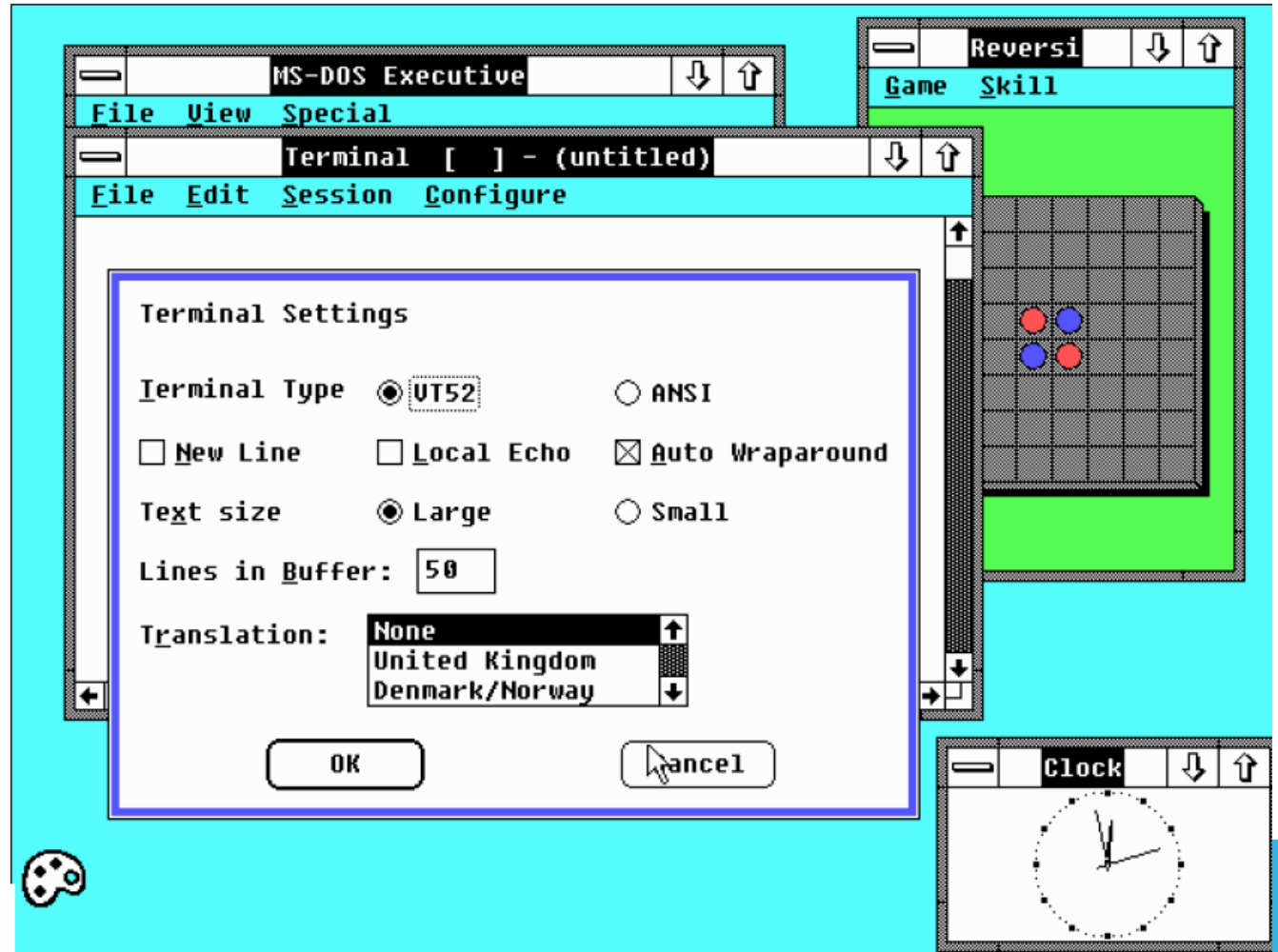




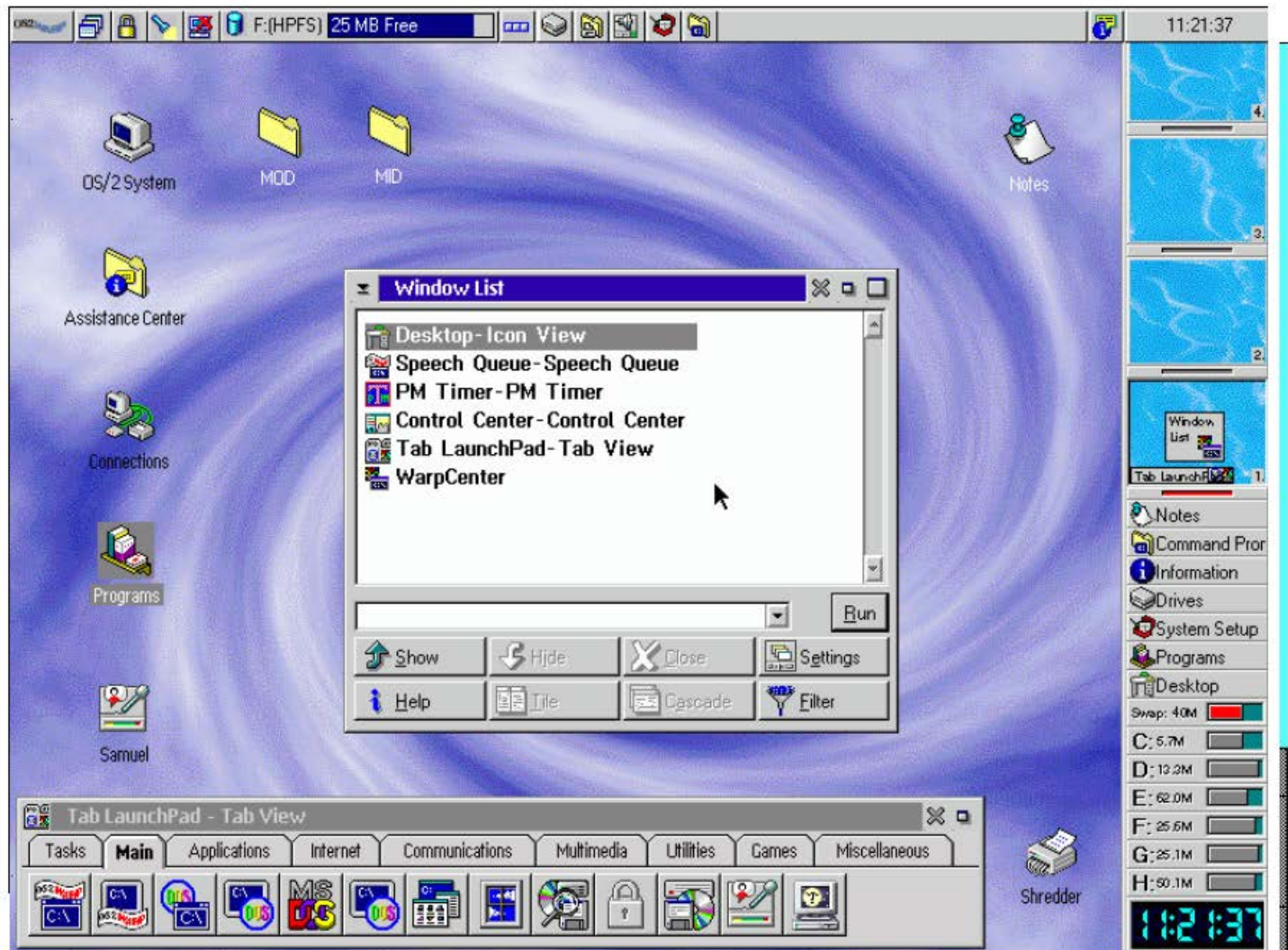
- Interfața  
Apple II  
(1986)



- Interfața  
Win 2.03  
(1987)



- Interfața OS/2 (1997)

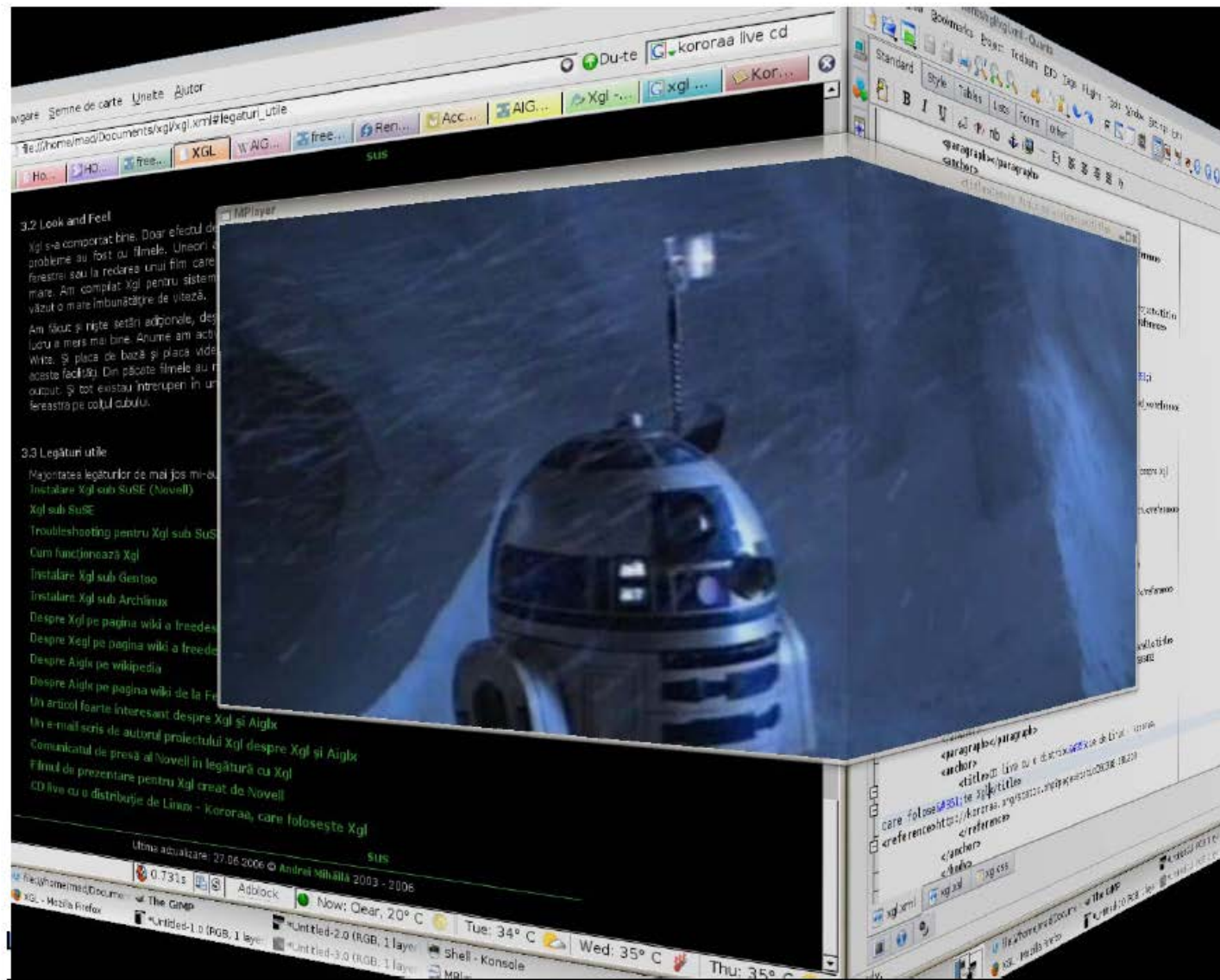




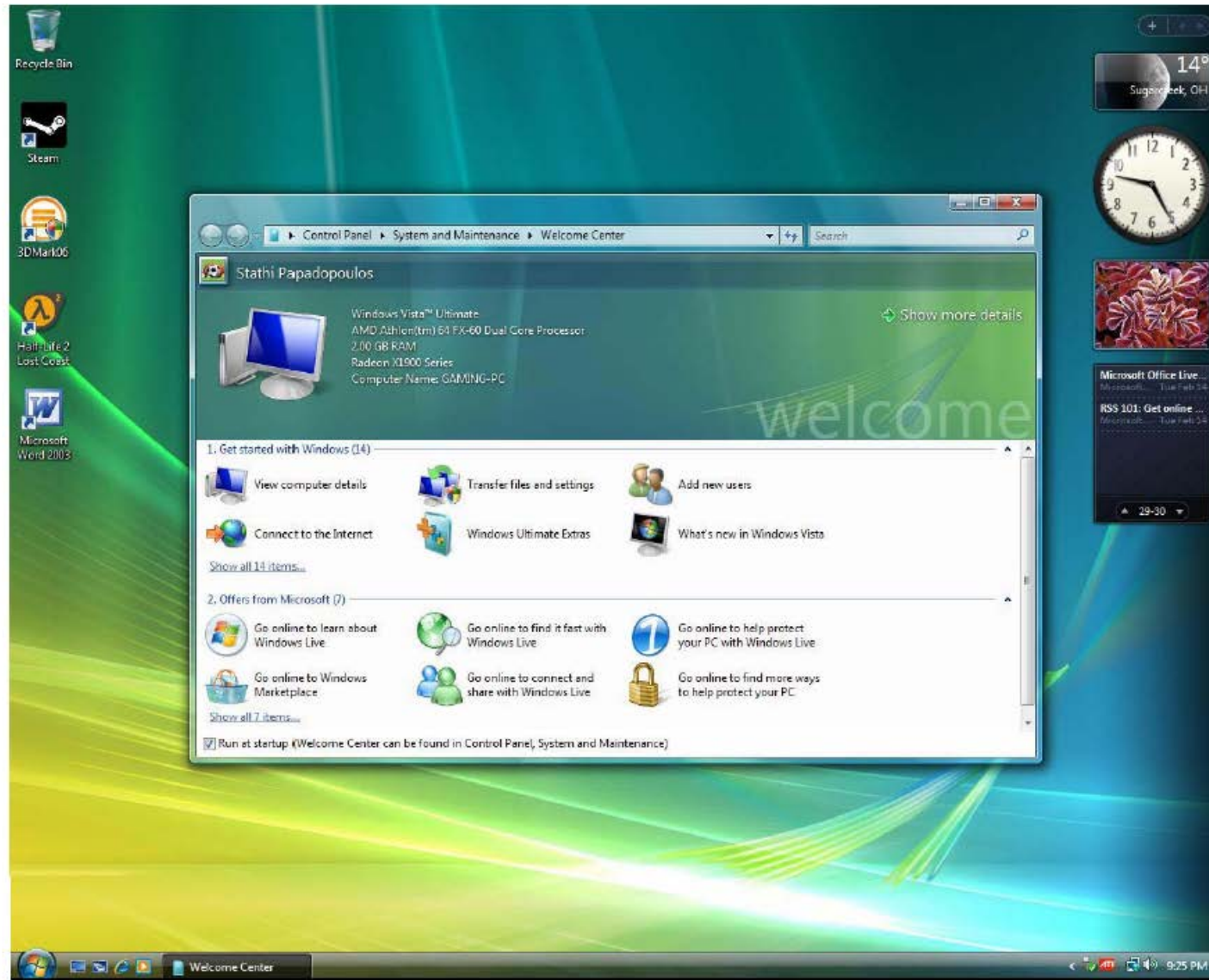
# • Interfața BeOS (2000)



- Interfața  
XWindow  
prin XGL  
(2005)



- Interfața  
Windows  
Vista  
(2006)





# VIITOR?



In cadrul acestei prezentari au fost folosite informati si imagini din documentele elaborate de dl Sabin Corneliu Burada , suport de curs  
<http://www.aie.ugal.ro/sica/curs> , diverse resurse web