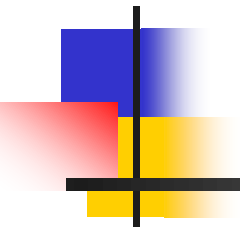


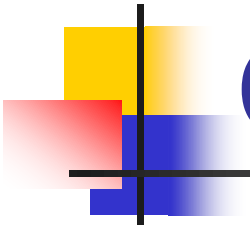
Graphical User Interface Design and Programming

Week 7

Graphical User Interface Components and Development Tools

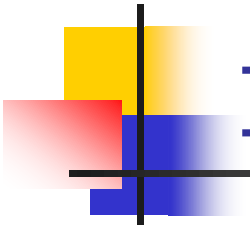
Recap of last lecture

- Interaction styles
  - direct manipulation, menus, forms, command line, natural language ... termasuk juga beberapa "exotic styles" yang sulit untuk diklasifikasikan.
- Banyak interfaces yang menggabungkan beberapa styles sebagai alternatif guna melayani "special tasks and users".
- Graphical User Interfaces (GUI) adalah salah satu contoh dari combined style.



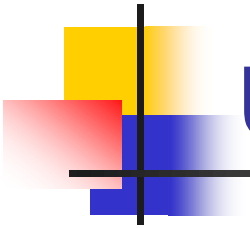
Overview

- User interface development
- Events overview
- Windowing systems
- Event dispatching and handling
- GUI development tools



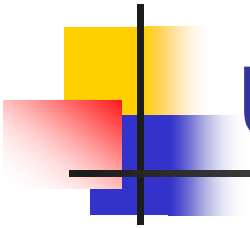
Interaction techniques

- GUI mengandung sekumpulan obyek-obyek (windows, widgets) yang di aplikasikan dalam banyak interaction techniques (styles).
  - Direct manipulation (buttons, resize window, scrollbars).
  - Menus (pull down, pop ups).
  - Dialog boxes, Text windows (form-fill in).



User interface development

- Interfaces →easier to use → harder to create.
- User interface software biasanya
  - Large
  - Complex
  - Difficult to implement
  - Debug
  - Modify.
- 50% dari listing code digunakan untuk membuat user interface system.



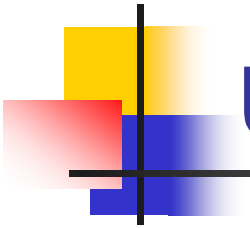
User interface development

- Waterfall model tidak lagi mencukupi kebutuhan untuk user interface development. Dibutuhkan pengunaan “iterative and reactive development methods”.
- User interface adalah salah satu bagian penting dari keseluruhan system

“deals with the user”

is more important than

“the functional core”

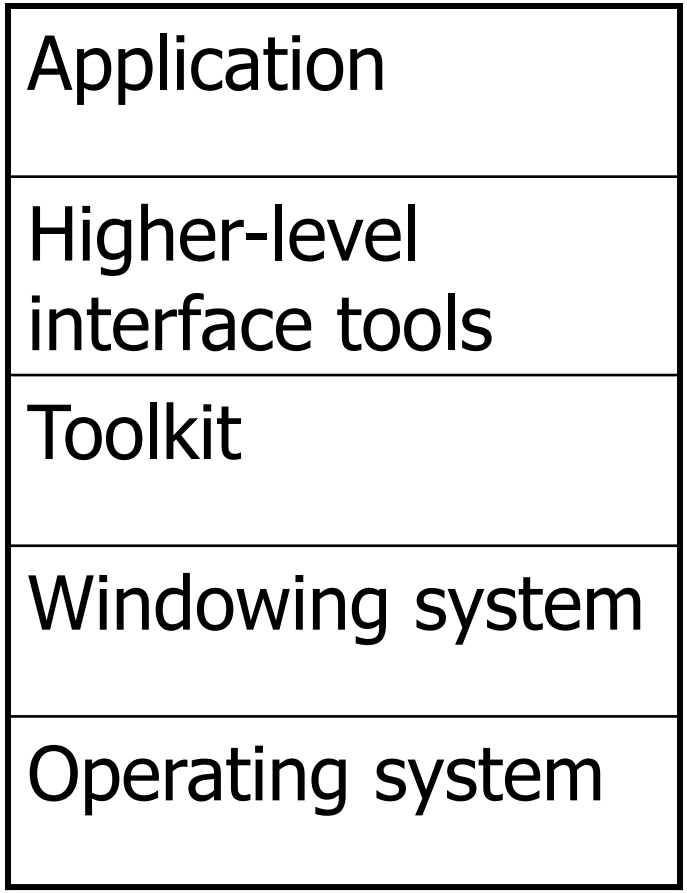


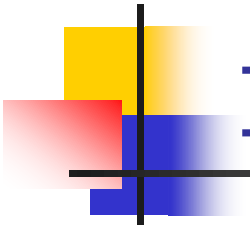
User interface : tools

- Bersamaan dengan dikembangkannya user interfaces dikembangkan pula software systems dan tools khusus untuk mendisain implementasi GUI.
- Banyak tools dikembangkan dan dijadikan produk komersial guna membantu software developers menghasilkan GUI yang baik
- Saat ini kebanyakan user interface software diciptakan dengan menggunakan tools.

User interface software

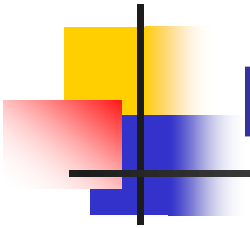
- User interface software dapat dibagi dalam beberapa tingkatan :
  - higher-level tools,
  - toolkits,
  - windowing systems.





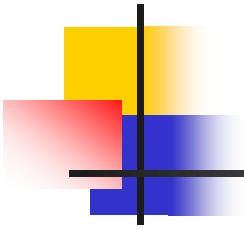
Interface technologies

- Event-based languages
- Windowing systems
- Toolkits
- Interactive graphical interface builders
- Component systems – Java Beans
- Scripting languages – Javascript, Perl
- Hypertext – web based
- Object oriented programming



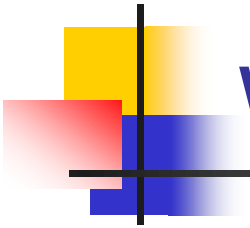
Event-driven programming

- Sequential (standard) program vs. event-driven program.
- Sequential Programs:
  - Program menggunakan control dan prompts untuk input command → command-line prompts (DOS, UNIX).
  - User menunggu program.
  - Program mengatakan kepada user bahwa siap untuk input lebih lanjut → User memasukkan input selanjutnya.



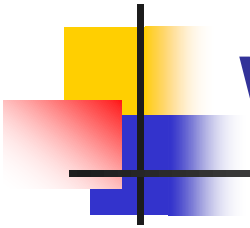
Event-driven programming

- Use tidak hanya menunggu program untuk siap melakukan task berikutnya, akan tetapi program juga menunggu user untuk juga dapat melakukan input task berikutnya.
- Seluruh komunikasi dari user ke komputer dilakukan lewat “events”.
- "*event*“ adalah sesuatu “of interest” yang terjadi dalam system:
  - Mouse button dapat digerakkan naik-turun
  - Item dapat di-drag and drop
  - Keyboard button ditekan



Windowing systems

- Kebanyakan GUI menggunakan windowing techniques – area segi empat (screens) yang mengandung aplikasi system.
- Windowing systems diberikan oleh software libraries yang dapat membuat windows dan mengimplementasikan elemen-elemen interaktifnya.
- Memberikan standardisasi dalam aplikasi.



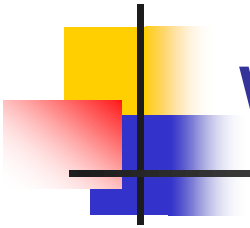
Windowing systems: advantages

- Memberikan pembagian logical dan physical dari multiple tasks.
- Menolong resources
  - screen dan human perceptual (visual field) dan cognitive resources.
- Bentuk yang rectangular memudahkan penyajian data bagi user.
- Overlapping menolong memory untuk memudahkan mengingat command / syntax.

Windowing systems

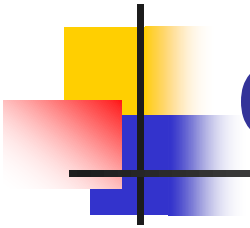
- Dalam Unix Window manager digunakan secara flexible:
  - Ada banyak window managers untuk X
  - Linux mengijinkan toggling antara dua Windows managers yang berbeda
- Dalam kebanyakan komersial system hanya ada satu pilihan saja.

Originally in Microsoft
Operating System = Windowing System.



Windowing systems

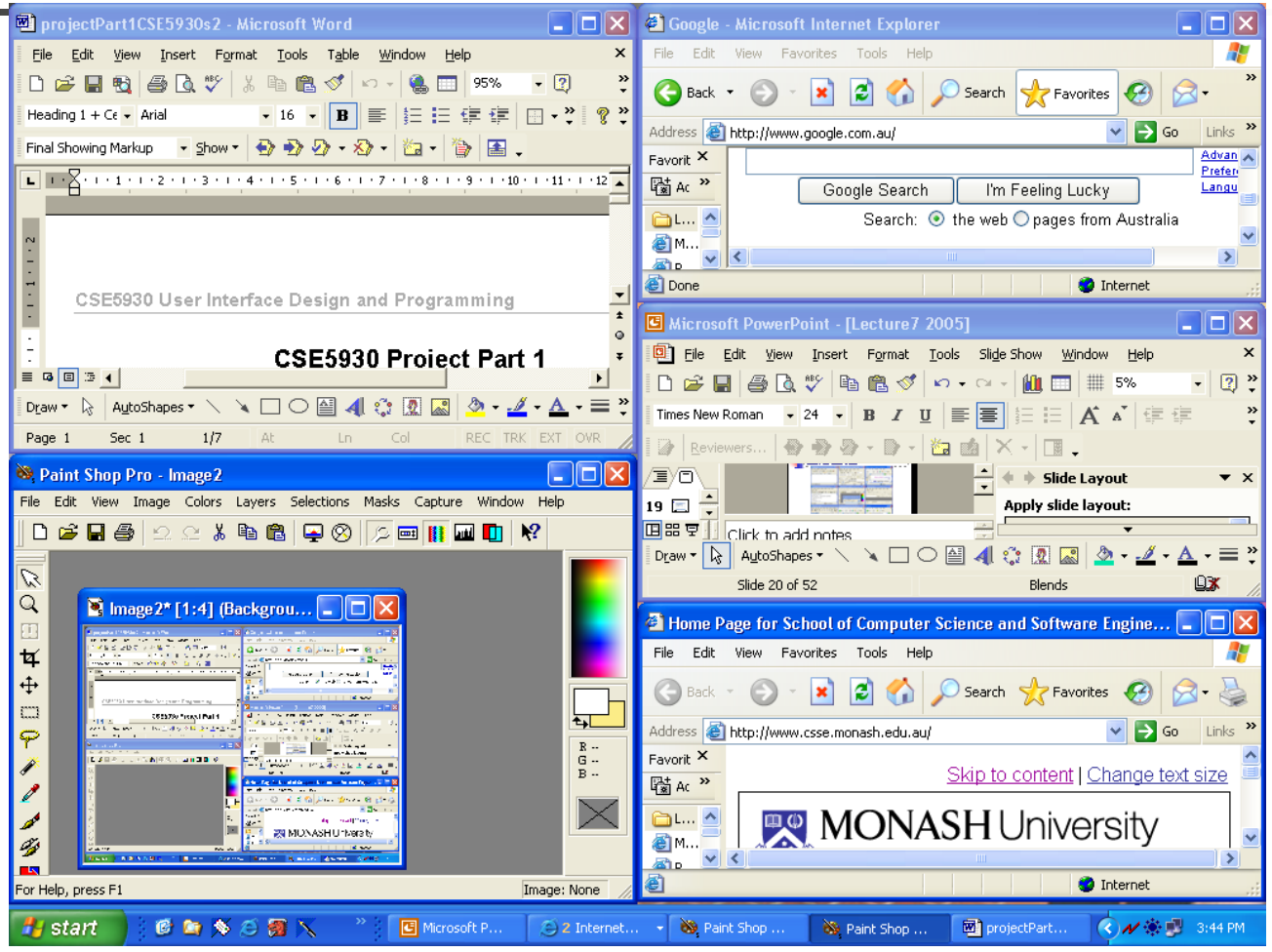
- Dalam Window system :
  - Output ke windows
  - Input dari user di lakukan dalam windows
  - User dapat menggerakkan mouse dan bekerja seluruh windows
  - User dapat men-display title lines, borders, dan icons diseluruh windows.



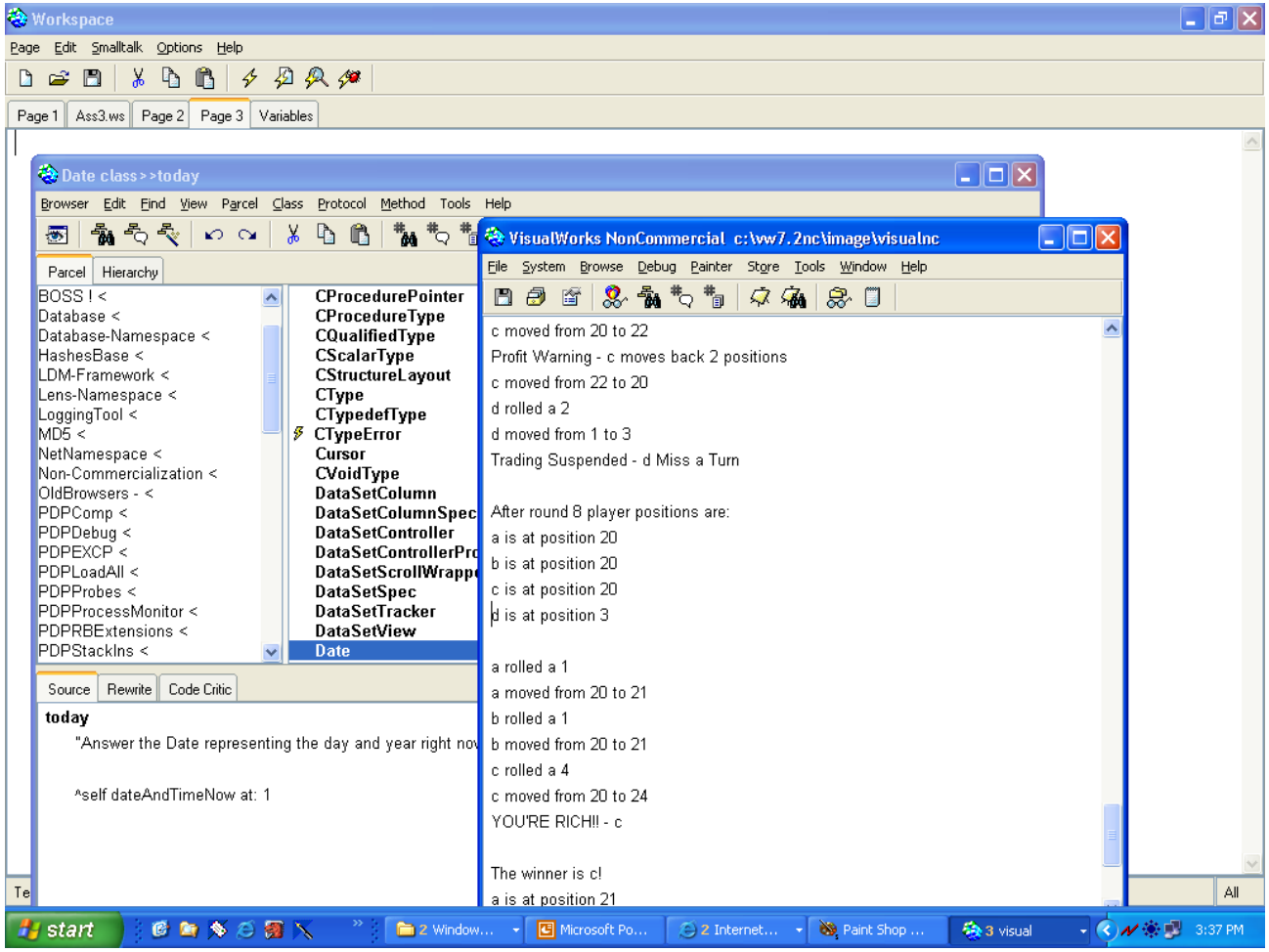
Overlapping and tiling

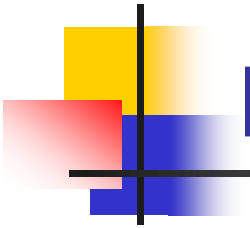
- Overlapping windows:
  - Menggunakan metaphor of overlapping sheets di desktop.
  - Overlapping windows tidak cukup baik untuk skala windows dalam multiple applications.
  - Ada bahaya ‘losing of applications’.
- Tiled windows:
  - Membagi-bagi screen dalam beberapa windows aplikasi.
  - Navigasi antara windows akan lebih mudah dengan tiling
  - Dibutuhkan ‘screen space’ yang besar.

Tiled windows



Overlapping windows

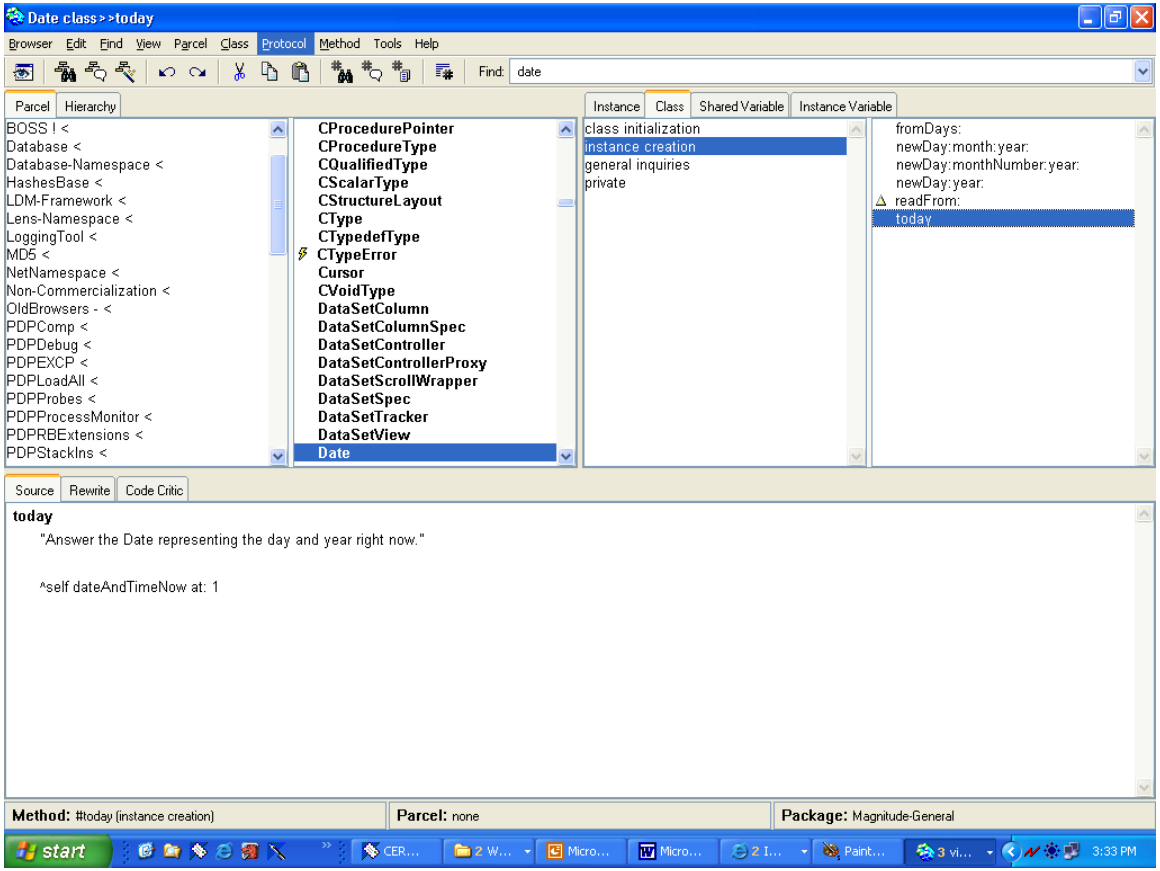


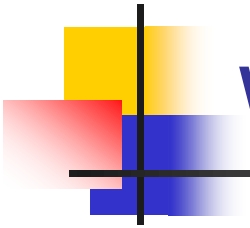


Multipaned windows

- Dalam satu windows terdapat beberapa ‘Independent windows’.
  - Dipisahkan dengan ‘splitters’ garis pembagi.
  - Informasi yang independen tapi berhubungan dapat disimpan dalam satu screen – mengurangi navigasi windows.

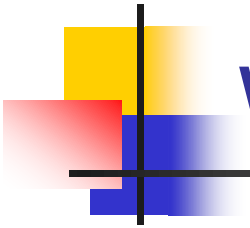
Multipaned windows





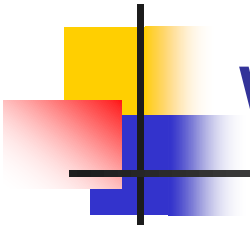
Windowing systems

- Windowing system menjaga agar user tetap fokus pada interface yang sedang dijalankan.
- Top level windows (main or root window)
  - Memberikan pemisahan aplikasi
  - Semua ‘descendant windows’ adalah satu dan sama dalam aplikasi



Windowing systems

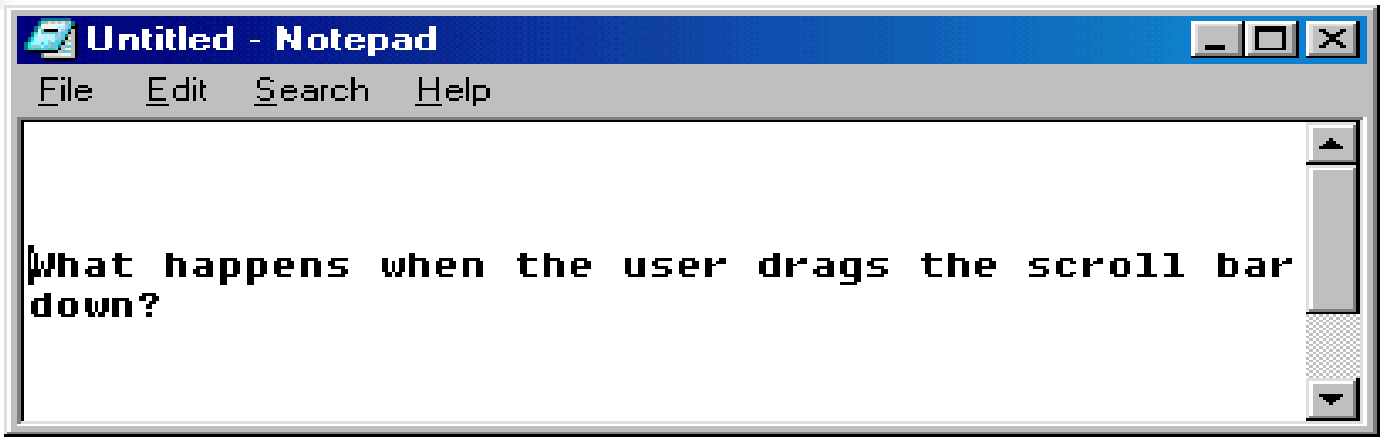
- Windowing system software :
  - Sekumpulan resources yang dihubungkan dengan setiap widget class
    - color, label, language
  - Selalu membuat dan menghapus ‘instance’ apa yang sudah dilakukan oleh widget.
  - Mendukung untuk ‘callbacks to run’ saat terjadi events pada instance widget.



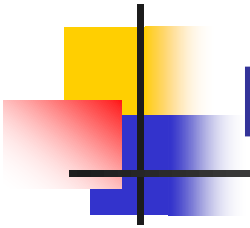
Window events

- User berinteraksi dengan input device:
  - action yang ditejemahkan kedalam software events
  - Harus ditempatkan dalam windows yang sesuai
- Events mengandung informasi:
  - Tipe-tipe input devices dan actions
  - mouse position atau character key
  - Windows dan event selalu berhubungan

Window events



- Windowing software harus :
  - Langsung mengena pada mouse events ke scroll bar
  - update scroll bar display selama nge-drag obyek
  - Mencatat text editing window jika membutuhkan scroll sehingga text memperlihatkan apa yang akan dipindah



Event focus

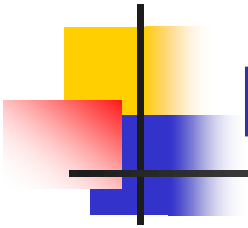
Windowing system mungkin digunkan dalam bernagai macam cara untuk menentukan fokus dari :

- Click-to-focus
  - User harus menggunakan mouse click dalam windows untuk secara langsung meng-inputkan data ke dalam window tersebut. Seluruh keyboard events langsung ditujukan ke dalam windows dimana mouse-click terjadi.
- Mouse-to-focus
  - Input dapat dimasukkan secara langsung kemana saja dalam seluruh screen. Keyboard events dijadikan satu dengan mouse position, diperlakukan secara sama seperti dalam mouse events.

Basic code structure for an interactive program

```
Initialize;
    Repeat
        Get next event
        Dispatch event
        If something_has_changed Then
          Redraw_All;
    Until time_to_exit;
```

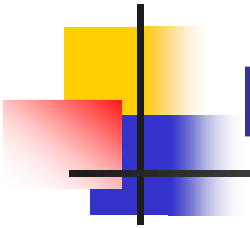
"Event/Redraw Loop" ini banyak digunakan dalam interactive systems.



Event queues

- Input events ditempatkan secara berurutan.
  - Pastikan bahwa events akan dilakukan sesuai dengan urutan
- Main event loop mengubah input events dari antrian (get-next-event) dan digabungkan untuk diproses.

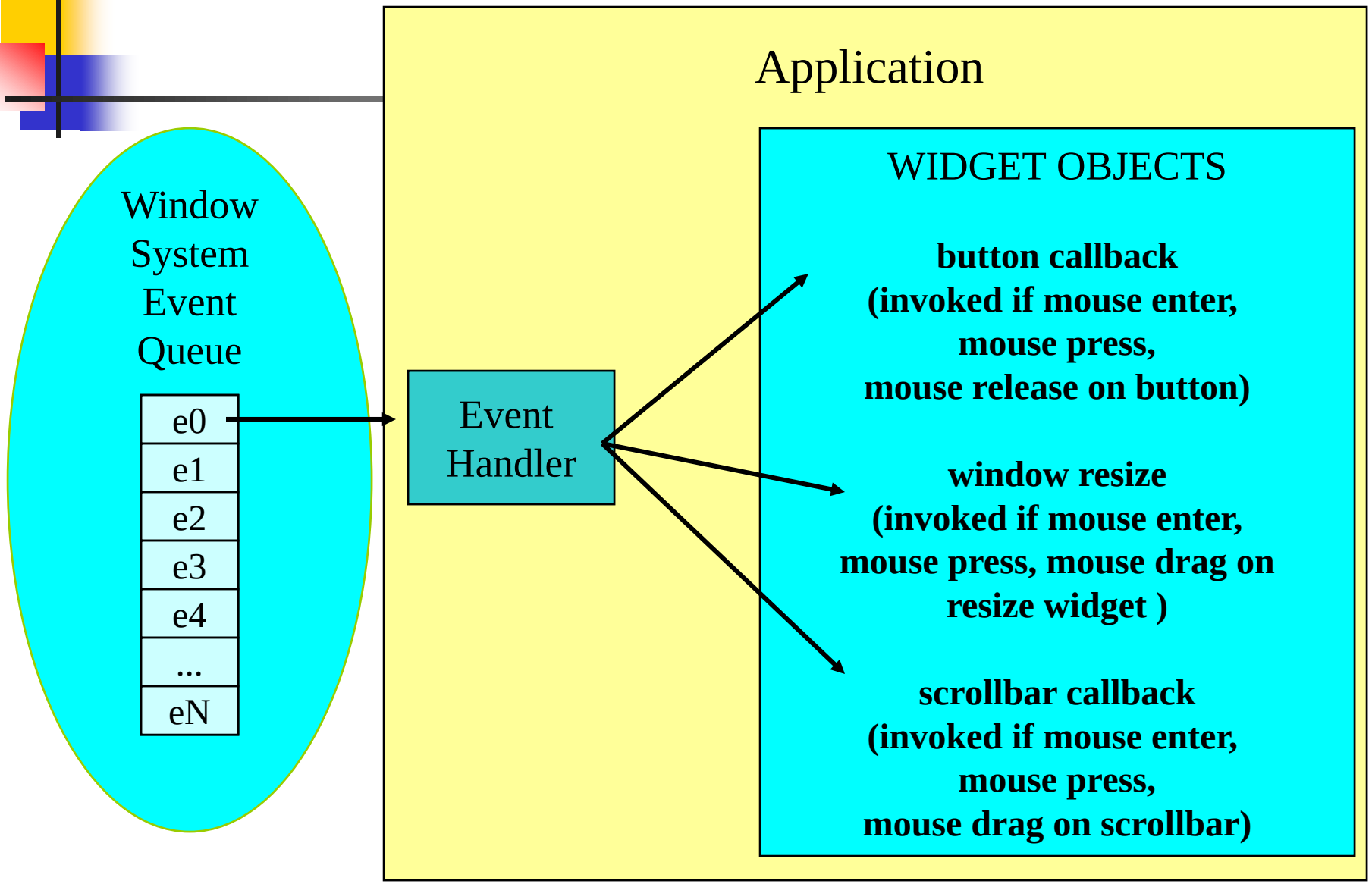
```
Mouse move (22, 33)
Mouse move (40, 30)
Mouse down left (45, 34)
Mouse up left (46, 35)
```

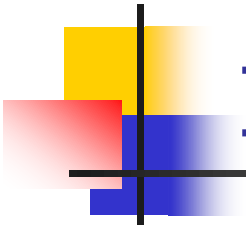


Event queues

- Dapat mengabaikan events yang tidak relevan dengan aplikasi :
  - Mouse movement kemungkinana dapat diabaikan dalam forms-based application. Hanya enter/exit events yang diproses.
  - Namun, dalam ‘drawing program’, kita ingin mengetahui gerakan ‘track mouse’.

Event dispatching





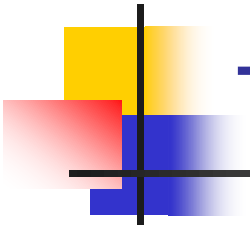
Interactor trees

Untuk mengatur interactive objects:

- Menbangun kembali interactive objects ke dalam ‘tree’:
  - Didasarkan pada screen geometry dari objects.
  - User butuh rectangles.
- Digunakan untuk menjalankan events:
  - Events dikirim didasarkan pada kode yang dihubungkan dengan widget
  - Kode tersebut menjadi respon dari events

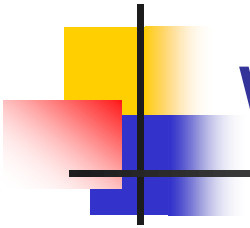
Interactor tree

- Display window[grey]
  - Title bar window[*blue*]
    - Result Win [white]
      - Result String
    - Keypads [*grey*]
      - = button
      - + button
      - - button
      - **0** button



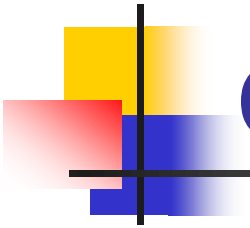
Toolkits

- Toolkit adalah library of “widgets” (juga disebut controls, interactors) yang dapat dijalankan dengan aplikasi program tertentu.
- Biasanya mengandung widgets yang sudah dikenal seperti menus, buttons, scroll bars, dan text input fields.
- Membuat interface menggunakan toolkit dengan tanpa bimbingan dari interface.



Widgets

- Banyak widgets mempunyai ‘particular appearance and behavior’ yang telah menjadi standard.
- Widgets mempunyai keterbatasan dalam interaksi dan juga butuh mendesain untuk spesialisasi / special tasks.

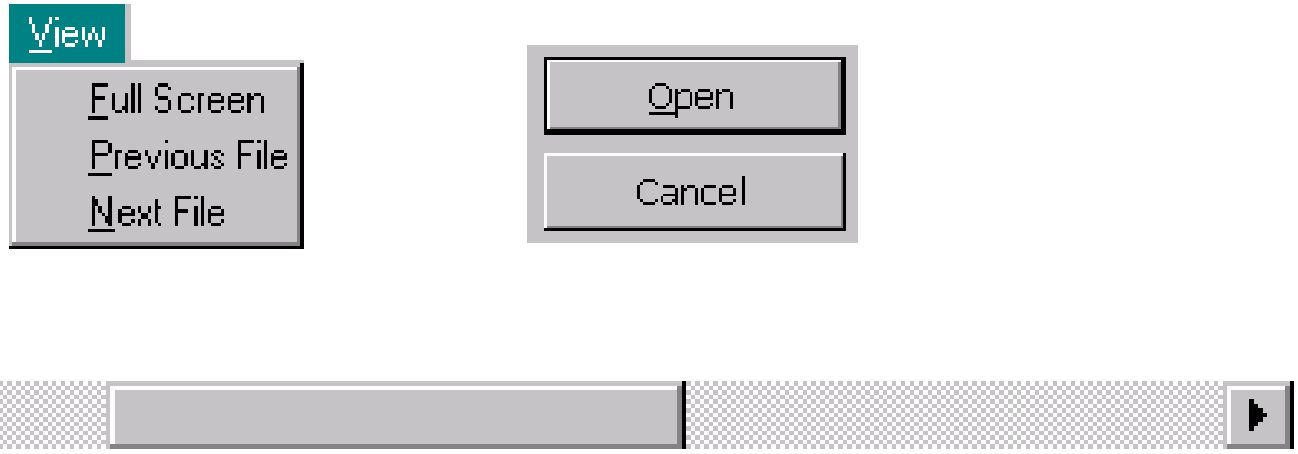


Original widgets

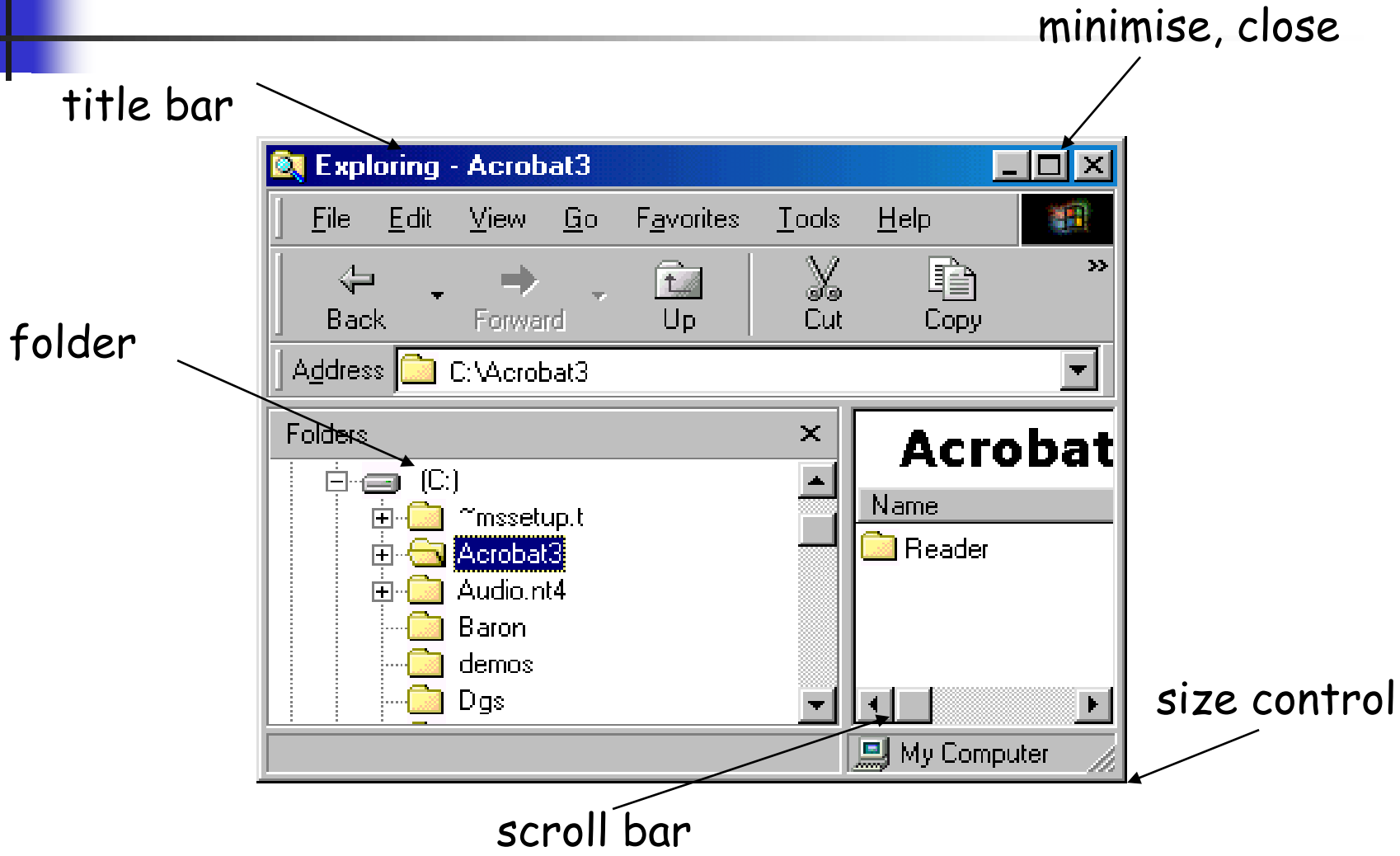
- Macintosh (1984) adalah implement GUI yang pertama dan sukses :
  - Secara umum, datang dgn 7 interaktor yang dibangun demi into toolbox.
  - Button, slider, pulldown menu, check boxes, radio buttons, text entry / edit fields, file management widget.

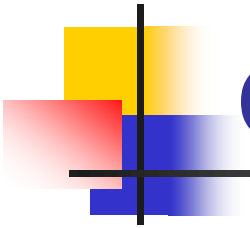
Widgets

- Button mempunyai “button properties” dan juga “state”.



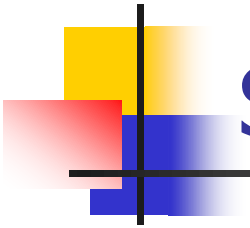
Widgets can register specific kinds of events





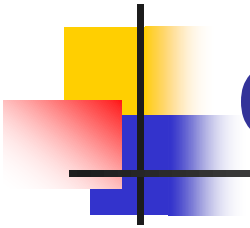
CallBacks

- Setiap widget mempunyai sekumpulan callbacks. Callback akan berjalan saat widget menerima perintah event.
- Hanya dengan memberikan sesuatu yang khusus tentang apa yang akan dilakukan jika terjadi event occurs pada widget.
- Menggunakan object-oriented model.



Scrolling text window: callback

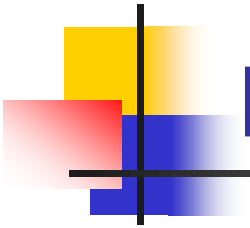
- Membuat ‘vertical scroll bar widget’.
- Menulis prosedur callback yang mempunyai kode untuk menunjuk / notify text windows pada posisi yang baru.
- Register callback sebagai suatu program untuk dijalankan saat scroll bar digerakan.
- Register text window sebagai data dari callback, maka system mengetahui window yang mana yang akan di-scroll.
- Mouse focus - tetaplah memfokuskan pada scrollbar widget sampai mouse button dilepaskan.



Constraints

relationships that are declared once and then maintained automatically by the system.

- Sebagai contoh desainer dapat menentukan warna dari kotak yang di constraint menjadi bernilai slider – maka system akan secara otomatis mengupdate kotak jika user menggerakkan slider

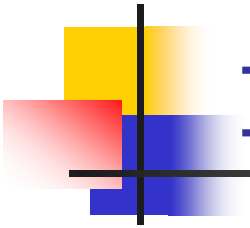


Higher level tools

- Karena programming pada level toolkit cukup sulit maka perhatian diarahkan pada level tools yang lebih tinggi.
- Kebanyakan digunakan dalam interface builder yang mengijinkan ‘interactive construction of interfaces’ dilakukan oleh widgets.
- High level tools yang lain adalah :
  - special-purpose languages.
  - component architectures.
  - automatically generation of interface from high-level model or specification.

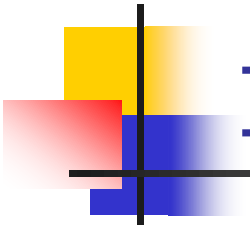
Interface tools : evaluation

- *Depth and Breadth*.
  - Berapa banyak cakupan interface tools dalam system ?
  - Apakah system menggunakan interface styles yang berbeda?
  - Dapatkah interaction techniques dan widgets yang baru ditambahkan?
- *Portability*.
  - Apakah user interface dapat bekerja dalam multiple platforms?
- *Ease of Use* dan *Efficiency* dari tools?



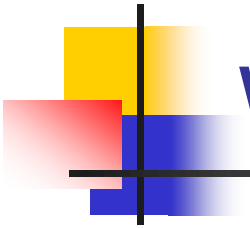
Interface tools : evaluation

- Bagaimana evaluasi tentang kualitas dan kuantitas dari system interfaces.
- Beberapa tools membutuhkan libraries yang amat besar dalam memori saat run-time.
- Bagaimana kehandalan dan ketahanan system interface
- Support – karena banyak tools cukup sulit untuk dioperasikan maka training dan after sales service menjadi penting bagi user.



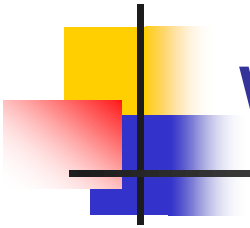
Interface builders

- Mempunyai palette atau menu widgets yang dapat dipilih dan diletakkan dengan mouse dalam sebuah form/window.
- Mempunyai sekumpulan properties.
- Menghubungkan call-backs dengan setiap widget.
- Membuat listing code secara langsung (C, PHP dll.)
- Mdah untuk digunakan.



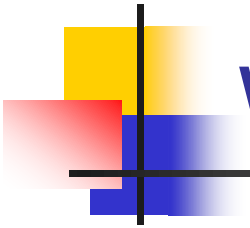
Why use interface builder?

- Desain berbeda dengan programming.
  - Orang yang tidak mengenal programming dapat mendesain dan mengimplementasikan user interfaces.
  - Kolaborasi antara desainer, user dan manager dalam proses pengembangan user interface.
  - User interface code menjadi lebih mudah dan lebih ekonomis karena adanya tools.



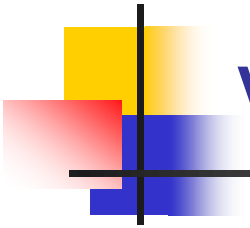
Why use interface builder?

- Memberikan sekumpulan standard user interface komponen.
- Dapat bekerja di multiple-platform.
- Fungsi-fungsi yang sulit / kompleks langsung dapat diotomatisasi
  - Validasi user inputs
  - Penanganan user errors, undo, field scrolling dan editing.
  - Menawarkan uji constraint dan consistency.



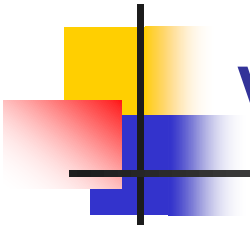
Why use interface builder?

- Kualitas interfaces dapat lebih baik karena :
  - Desain dapat dengan cepat di-prototypekan dan diimplementasikan.
  - Kemungkinan menjadi lebih mudah untuk menggabungkan perubahan yang ditemukan berdasar pada user testing.
  - Ada banyak user interfaces untuk aplikasi yang sama.
  - Ada kemungkinan untuk mengembangkan ‘sophisticated tools’ bagi penggunaaan aplikasi yang berbeda bagi setiap user.



VB.NET

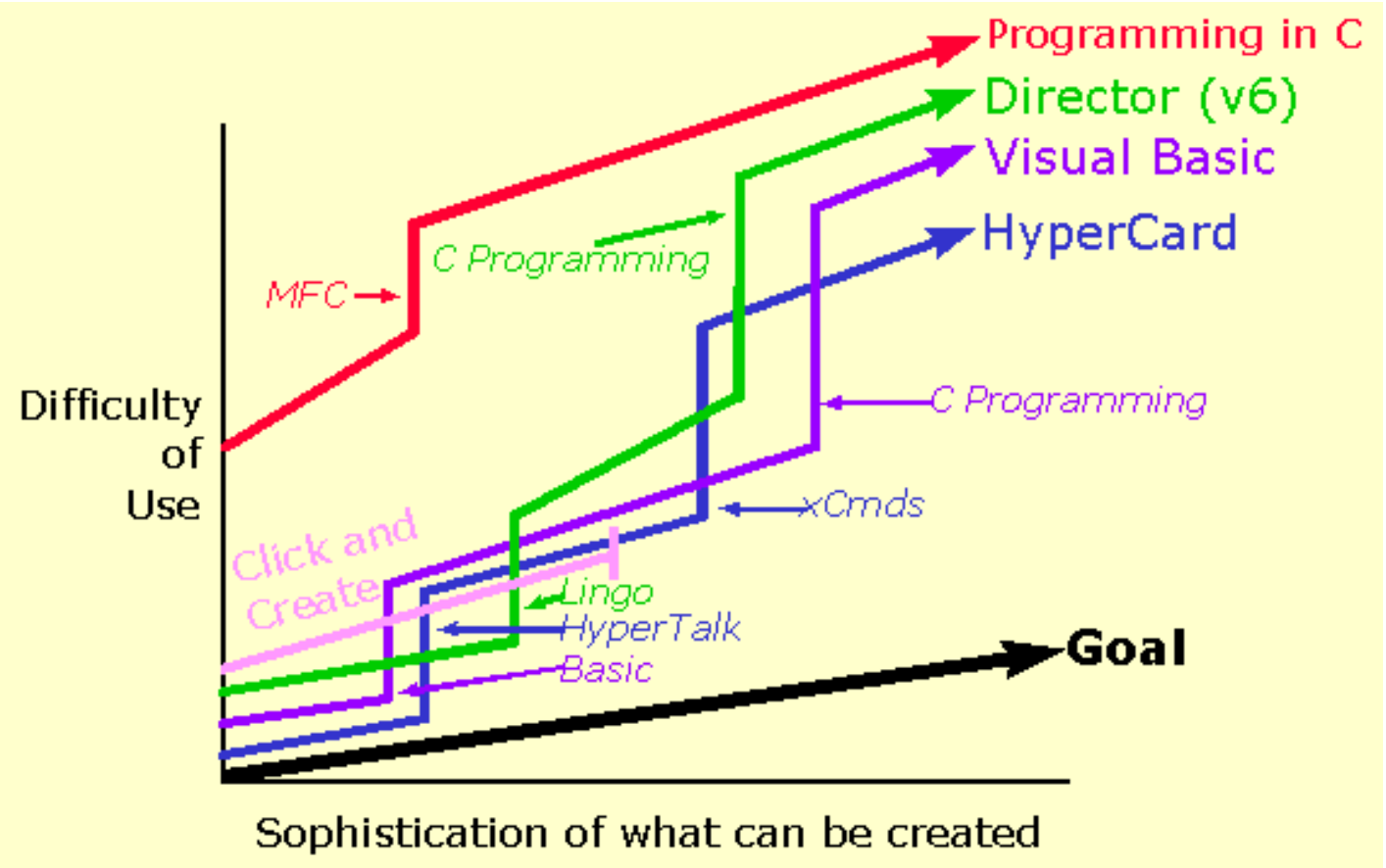
- Visual Basic.NET, termasuk dalam Microsoft Visual Studio .NET sebagai salah satu GUI programming languages yang banyak digunakan.
- VB.NET menggunakan ‘direct manipulation’ dan mengandung banyak komponen pre-written dan fungsi-fungsi otomatisasi.



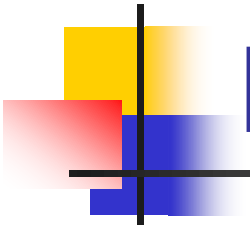
VB.NET

- VB and VS.NET membuat prototyping dan development user interface secara cepat dengan hanya drag and drop objects ke dalam forms.
- Objects kemudian dapat dimodifikasi dengan menggunakan Properties window.

Ease of use of tool vs. sophistication of creation



(Myers, 1998)



References

- Carroll, J. M. (2002). Human-Computer Interaction in the New Millenium. New York, New York, USA. (Chapter 10)
- Cooper, A., & Reimann, R. (2003). About Face 2.0: The Essentials of Interaction Design. Indianapolis, Indiana, USA: Wiley Publishing, Inc. (Chapter 25)
- Myers, B., 1998, Natural programming: Project proposal and overview, (http://www.cs.cmu.edu/~bam)
- Myers, B. A. & Rosson, M. B., (1992), Surveu on user interface programming, CHI '92. p.196-202.
- Shneiderman, B., & Plaisant, C. (2005). Designing the User Interface: Strategies for Effective Human-Computer Interaction
- Stone, D., Jarrett, C., Woodroffe, M., & Minocha, S. (2005). User Interface Design and Evaluation.