SCIENCES SORBONNE UNIVERSITÉ

Event-based programming, an example

Fabrice.Kordon@lip6.fr

LIP6

Graphical interfaces

Strongly rely on event-programming

- Origin, X
  - Xerox PARC laboratories (Palo Alto Research Center)
  - In the 1970's

Massively adopted in the 1980's

- Unix
- Highlighted by the LISA (1983)
  … and mostly the Macintosh (1984)

Typical start of such programs

Configure «handlers»

```
begin
 enable_event (SIG_MOV, @upd_pointer)
 enable_event (SIG_L, @draw_menu)
 wait_event
end
```

Is that all?

Yes, the event manager handles everything

Saving CPU, saving energy, etc.

Managing the mouse pointer

Available elements

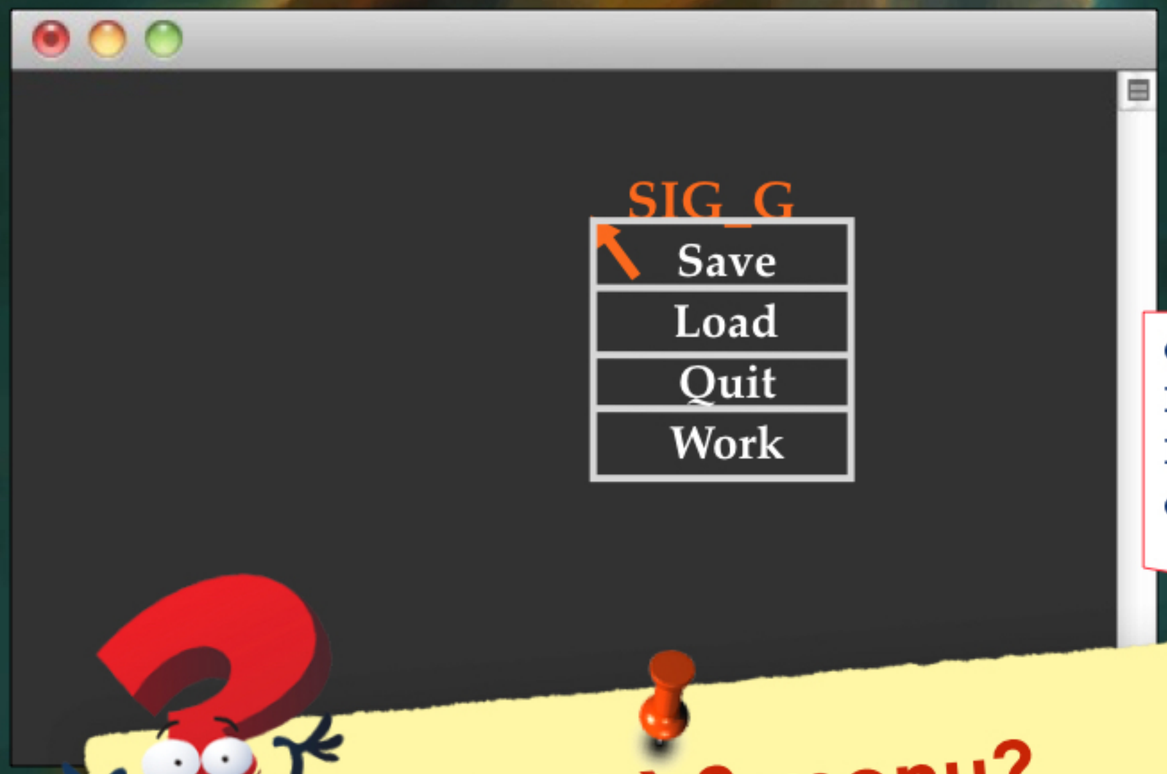
- The absolute position of the pointer
  - **Global variables H & V**
- Changes of position since the last event-catch
  - **Global variables deltaH & deltaV**
- The `delete_pointer` primitive to delete the pointer
  - **Position of the pointeur passed as arguments**
- The `draw_pointer` primitive to draw the pointer
  - **Position of the pointeur passed as arguments**
- The draw_menu primitive to draw the menu
  - **Position of the menu passed as arguments**

The upd_pointer and draw_menu handlers


```
begin
 delete_pointer (H, V)
 PMV := V
 PMH := H
 draw_menu (PMH, PMV)
 draw_pointer (PMH, PMV)
 enable_event (SIG_M, @active_item)
 enable_event (SIG_L, @efface_menu)
 wait_event
end
```

```
begin
    delete_pointer (H, V)
    H := H + deltaH
    V := V + deltaV
    draw_pointer (H, V)
    wait_event
end
```

Execution (preemptive events)



Pointer(s) ? menu?

Are you at the right position?

My program

SIG_G

```
delete_pointer (H, V)
PMV := V
PMH := H
draw_menu (PMH, PMV)
```

SIG_DEPL

```
delete_pointer (H, V)
H := H + deltaH
V := V + deltaV
draw_pointer (H, V)
wait_event
```

```
draw_pointer (PMH, PMV)
enable_event (SIG_M, @active_item)
enable_event (SIG_L, @efface_menu)
wait_event
```

Synchronize to avoid conflits

How?

- Disarming events = delay events
- Use synchronization mechanisms

```
begin
    disarm_event
    efface_curseur (H, V)
    PMV := V
    PMH := H
    affiche_menu (PMH, PMV)
    dessine_curseur (PMH, PMV)
    enable_event (SIG_M, @active_item)
    enable_event (SIG_L, @efface_menu)
    rearm_event
    wait_event
end
```

```
begin
    disarm_event
    efface_curseur (H, V)
    H := H + deltaH
    V := V + deltaV
    dessine_curseur (H, V)
    rearm_event
    wait_event
end
```

As a conclusion

These are basic principles

- Reality is more complex

Good example

- Common mechanisms
- Common problems

You will face such problems

- At a higher level
- Solutions based on synchronizations too
  - **Often handled in complex frameworks**