

RDM5: MOVEM.L D1-D7/A0-A6, -(SP)

BSR VQMOUSE

BEQ GEU24

MOVEM D1/D2, (A3)

BTST #1, D0

BEQ GEU25

← page 110 i

GEU25: MOVEQ #-1, D0

GEU26: DBRA D0, GEU26

BSR VQMOUSE

MOVE.L (A3)+, D3

M

MOVEM D1/D2, (A3)

CMP.L (A3), D3

BNE GEU28

TST D0

BNE GEU25

GEU27: LEA TCURSA, A0

↓ bouton gauche relâché de suite

MOVEQ #1, D0

CMP (A0)+, D3

BGE GEU23

MOVE 4(A0), D0

teusec ↓ dans le menu

ADDQ #1, D0

MULU #10, D1

DIVU D0, D1

∈ [0, 10[

MULU #10, D3

MOVE #200, D0

ADD D1, D0

ADD D3, D0

BRA GEU24

GEU28: MOVE D2, D3

TST D0

BEQ GEU27

TST.B KEYRECT

BNE GEU25

CMP TCURSA, D2

BLT GEU25

} ⊗ pas de bloc commençant sur le dernier

GEU20: MOVEQ #-1, D0

GEU22: DBRA D0, GEU22

BSR VQMOUSE

BNE GEU20

CMP (A3)+, D1

BNE GEU24

CMP (A3)+, D2

BNE GEU24

CMP TCURSA, D2

BCS GEU24

MOVEQ #2, D0

} ⊗

GEU23: SWAP D0

LEA MOUSEXY, A3

MOVEP 1(A3), D0

GEU24: MOVEM.L (SP)+, D1-D7/A0-A6

RTS

1

```

add #12, a3
dr.l (a3)+
dr.l (a3)+

```

vide le rectangle  
"vidéo inverse"

```

BSR HIDEW
BSR XCURROFF
BSR SHOWM

```



```
MOVEM (TCURSV, D0-D2
```

```
MOVEM. D0/D2, TCURSM ) comme curseur
```

```

gen29: bsr HIDEW
LEA MOUSEXY, A3

```

```
MOVEM (A3), D0-D3
```

```
MOVE TCURSA, D4
```

normalise

```

CMP D4, D1
BCC gen30
MOVE D4, D1

```

```

gen30: CMP D4, D3
BCC gen31
move d4, d3

```

```
gen31: movem d0-d3, (A3)
```

```

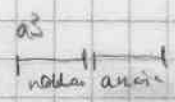
CMP D0, D2
BCC gen32
EXG D0, D2

```

```

gen32: CMP D1, D3
BCC gen33
EXG D1, D3

```



```

gen33: addq #1, d2
      addq #1, d3

```

```
MOVEM d0-d3, (a3)
```

} nouveau rectangle

```
x BSR differ
```

} réécrit les cases du 2<sup>ème</sup> rectangle qui ne sont pas dans le 1<sup>er</sup> (conservé a3)

```
BSR videoi vidéo inverse
```

```
MOVEM.L (a3), d0-d3
```

```
exg d0-d2
```

```
exg d1-d3
```

```
movem.l d0-d3, (a3)
```

} échange les rectangles et réécrit 2<sup>ème</sup> rect

```
x BSR differg
```

```
BSR video^n
```

vidéo normale

```
BSR SHOWM
```

souris

```

geu34:BSR  VQMOUSE
      BNE  geu35
      lea mousexy+16,a3
      dr.l -(a3)
      dr.l -(a3)

```

} vide 1er rectangle

```

BSR  HIDEW
BSR  differ
BSR  SHOWM
BSR  XCRCL
BSR  XCURION

```

} remet en video normale  
remet le curseur

```

moveq #1, do
lea mousexy+16, a3
move.l (a3)+, d1
sub.l #10001, (a3)
cmp.l (a3), d1
beq geu23

```

→ rectangle vide

```

moveq.l -3(a3), do
bset #31, do

```

⊗

```

MOVEQ #3, DO
SWAP DO
ASL #8, D1
MOVE D1, DO
MOVE.B 3(A3), DO
BRA GEU24

```





```

1
gen35: addq #4, a3
      movem (a3), d3/d4    valeurs précédentes
      movem d1/d2, (a3)
      cmp  d1, d3
      bne  gen29
      cmp  d2, d4
      bne  gen29
      movq #-1, do
gen36: dbra do, gen36
      bra  gen34

```