

```

INIT_LINEA = $A000 ; Line A opcodes
DRAW_SPRITE = $A00D
ERASE_SPRITE = $A00C
TOP = 0 ; Co-ordinates of screen corners
BOTTOM = 383 ; Adjust to suit resolution you
LEFT = 0 ; are working in.
RIGHT = 623

.TEXT ; Text segment
.GLOBAL _MAIN ; Program code starts here!
_MAIN
MOVE.W #0,X_POS ; Initialise x,y co-ordinates
MOVE.W #0,Y_POS
MOVE.W #2,X_INC ; Initialise x,y increments
MOVE.W #1,Y_INC
.WORD INIT_LINEA ; Initialise Line A for usage

DRAW_LOOP ; Draw sprite
MOVE.W Y_POS,D1 ; At X_POS,YPOS
MOVE.W X_POS,DO
MOVE.L #SPRITE_DEF,A0 ; Shape in SPRITE_DEF
MOVE.L #SPRITE_SAVE,A2
.WORD DRAW_SPRITE ; Using Line A opcode $A00D

JSR DELAY ; Small delay

MOVE.W X_POS,DO ; Increment x co-ordinate
ADD.W X_INC,DO
MOVE.W DO,X_POS
CMP.W #LEFT,DO ; Past left edge?
BLT BOUNCE_X ; Yes - correct it
CMP.W #RIGHT,X_POS ; Past right edge
BGT BOUNCE_X ; Yes - correct it

X_POS_OK ; X co-ordinate ok
MOVE.W Y_POS,DO ; Now increment y co-ordinate
ADD.W Y_INC,DO
MOVE.W DO,Y_POS
CMP.W #TOP,Y_POS ; Past top edge?
BLT BOUNCE_Y ; Yes - correct it
CMP.W #BOTTOM,Y_POS ; Below bottom?
BGT BOUNCE_Y ; Yes - correct it

Y_POS_OK ; Both x+y co-ordinates ok.
MOVE.L #SPRITE_SAVE,A2 ; Restore the background
.WORD ERASE_SPRITE

SCAN_KBD ; Scan the keyboard
MOVE.W #$FF,-(SP)
MOVE.W #6,-(SP) ; Using GEMDOS function RAWCONIO
TRAP #1
ADDQ.L #4,SP
CMP.W #0,DO ; Any keys pressed?
BEQ DRAW_LOOP ; No then re-draw sprite

CLR.W -(SP) ; Key pressed - exit to desktop
TRAP #1

BOUNCE_X ; Let X_INC = -X_INC
NEG.W X_INC
MOVE.W X_POS,DO
ADD.W X_INC,DO ; Add 2*X_INC to X_POS
ADD.W X_INC,DO
MOVE.W DO,X_POS
BRA X_POS_OK ; Now X_POS is ok.

BOUNCE_Y ; Let Y_INC = -Y_INC
NEG.W Y_INC
MOVE.W Y_POS,DO
ADD.W Y_INC,DO ; Add 2*Y_INC to Y_POS
ADD.W Y_INC,DO
MOVE.W DO,Y_POS
BRA Y_POS_OK ; Now Y_POS is ok.

; *****
; * SUBROUTINES *
; *****
DELAY ; Short delay - adjust the length and
MOVE.W #5800,DO ; observe the effect on the sprite speed.
LOOP
SUB.W #1,DO
BNE LOOP
RTS

.BSS ; Variable Storage Segment
X_POS .SPACE 2 ; All 1 word long
Y_POS .SPACE 2
X_INC .SPACE 2
Y_INC .SPACE 2
SPRITE_SAVE .SPACE 266 ; Reserve 266 bytes for background save

.DATA ; Data Storage Segment
SPRITE_DEF .WORD 0 ; Sprite Definition Block
.WORD 0 ; Alter at your own risk!
.WORD 0
.WORD 1

.WORD $FFFF,$FFFF,$FFFF,$B001,$FFFF,$B001,$FFFF,$B001
.WORD $FOOF,$B001,$FOOF,$B001,$FOOF,$B001,$FOOF,$B001 ; Sprite
.WORD $FOOF,$B001,$FOOF,$B001,$FOOF,$B001,$FOOF,$B001 ; data
.WORD $FFFF,$B001,$FFFF,$B001,$FFFF,$B001,$FFFF,$FFFF

```