

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS Y SISTEMAS

ARQUITECTURA DE COMPUTADORES Y ENSAMBLADORES 1

SECCIÓN “N”

VACACIONES DEL PRIMER SEMESTRE 2023



MANUAL TÉCNICO

PRÁCTICA 2

Diego Andrés Huite Alvarez

202003585

03/07/2023

Índice

INTRODUCCIÓN	3
REQUERIMIENTOS	4
ARCHIVOS DE MASM	5
 DIRECTIVA .DATA	6
CARGAR EL NIVEL	8
CAMBIAR CONTROLES	9
INTERRUPCIONES USADAS	11

INTRODUCCIÓN

Bienvenido al manual de usuario del proyecto 2 de arqui1, un juego que imita al famoso juego SOKOBAN.

REQUERIMIENTOS

Software:

- Sistema operativo windows, linux o macOS
- DOSbox instalado
- MASM611

Hardware:

- Mouse
- Teclado
- Monitor
- 2gb de ram

ARCHIVOS DE MASM

- **Main.asm:** Contiene el flujo principal del programa, así como las verificaciones de las físicas, del juego, contadores y demás.
- **Sprites.asm:** Contiene los sprites usados en el juego para su correcto funcionamiento.
- **Utils.asm:** Contiene funciones breves para imprimir mensajes, recibir inputs de usuarios, etc.

DIRECTIVA .DATA

```
.DATA
getSprites
iniciar_juego db "INICIAR JUEGO$"
cargar_nivel db "CARGAR NIVEL$"
configuracion db "CONFIGURACION$"
puntajes db "PUNTAJES ALTOS$"
salir db "SALIR$"
iniciales db "Diego Andres Huite Alvarez",0A,"          202003585","$"
inicialesGame db "DDAHA - 202003585","$"
controlesActuales db "CONTROLES ACTUALES","$"
newLine db 0A, "$"
;; JUEGO
xJugador db 0
yJugador db 0
puntos dw 0
;; MENÚS
```

Contiene variables, prompts y buffers que serán de utilidad a lo largo del flujo del programa.

```
.CODE
.STARTUP
inicio:
    ;; MODO VIDEO ;;
    mov AH, 00
    mov AL, 13
    int 10
    ;;

    ;;;; POSICIONARNOS EN LA FILA 08 COLUMNA 05
    mov DL, 05
    mov DH, 08
    mov BH, 00
    mov AH, 02
    int 10
    ;; IMPRIMIMOS MENSAJE

    printString iniciales
    ; haciendo un for para un delay algo largo
    xor si, si

    cfor:
        cmp si, 0A
```

el flujo inicial del programa comienza seteando el modo video e imprime el mensaje de bienvenida

```
exitfor:
mov [contadorMovs], 0000
mov [contadorLevel], 0000
mov nivel_x[04], "0"
mov nivel_x[05], "0"
mov di, offset mapa
mov cx, 3e8
mov al, 0000
call memset
```

```
displayPrincipalMenu:
call clear_pantalla
call menu_principal
mov AL, [opcion]
cmp AL, 0
je cargar_un_nivel

cmp AL, 01
je loadLevel

cmp AL, 03
je menuconfig
```

modificamos siempre la variable nivel_x para asegurarnos de que ingrese primeramente al nivel 00. Y en base al valor de la variable opción, vamos a cada uno de los distintos menús.

CARGAR EL NIVEL

```
loadLevel:
call clear_pantalla
mov DL, 05
mov DH, 08
mov BH, 00
mov AH, 02
int 10

printString promptpath
saveBufferedInput inputBuffer
printString newLine
bufferPrinter inputBuffer
```

Se hizo uso de las macros en utils.asm para poder pedirle un input al usuario y buscar el nivel para jugar.

CAMBIAR CONTROLES

```
changeControls:
    mov DL, 05 ; columna 12
    mov DH, 01 ;fila 1
    mov BH, 00
    mov AH, 02
    int 10
    ;; <<-- posicionar el cursor

    call clear_pantalla
    printString promptDown
    mov ah, 00
    int 16
    mov controlDown, ah
    mov stringcontrolDown[0], al

    mov DL, 05 ; columna 12
    mov DH, 01 ;fila 1
    mov BH, 00
    mov AH, 02
    int 10
    ;; <<-- posicionar el cursor
```

Se hizo uso de la interrupción 16 subcodigo 00 para poder guardar el scancode de las teclas para cambiar la configuración y se reemplaza el valor en la variable.

La lógica para pasar de niveles se hace en base a una variable

```
pasarSiguienteLevel:
    call clear_pantalla
    cmp customLevel, 1
    je displayPrincipalMenu

    inc contadorLevel
    cmp contadorLevel, 01
    je gotoSecond
    cmp contadorLevel, 02
    je gotoThird
    jmp restart

gotoSecond:
    mov nivel_x[04], "0"
    mov nivel_x[05], "1"

    mov di, offset mapa
    mov cx, 3e8
    mov al, 0000
    call memset
    jmp cargar_un_nivel

gotoThird:
    mov nivel_x[04], "1"
    mov nivel_x[05], "0"
    mov di, offset mapa
```

INTERRUPCIONES USADAS

1. INT 10H
2. INT 21H
3. INT 16H