

Trabajo Práctico Final SPD GRUPO "E"



PIEDRA, PAPEL O TIJERA

INTEGRANTES

Mingrone Carlos Antonio - carlos_mingrone@hotmail.com

Dragonetti Maria Cecilia -mdragonetti@estudiantes.unsam.edu.ar

Fariña Espinosa Ximena -xfarinaespinosa@estudiantes.unsam.edu.ar

Pastorino Claudia -Pastorino.claudia@gmail.com

DESCRIPCIÓN DEL JUEGO

El juego consiste en vencer a la máquina seleccionando el "arma" que ganará el juego. Las reglas son las siguientes:

- Si escoges jugar con la piedra tendrás que saber que la misma aplasta a la tijera pero no así al papel.
- Si escoges la tijera, cortará el papel pero la piedra la romperá.
- Si escoges el papel, éste envuelve a la piedra pero la tijera lo corta.
- En caso de empate, el cual se da si escoges lo mismo que la máquina, el juego volverá a comenzar.

Instalación:

```
@ECHO OFF
tasm intClear.asm
tlink /t intClear.obj
intClear.com
tasm /zi tpfinal.asm
tasm /zi libtp.asm
tlink /v tpfinal.obj libtp.obj
tpfinal.exe
```

El juego se instala ejecutando el archivo comppp.bat, el cual contiene todas instrucciones necesarias para hacer la instalación de la interrupción y linkear la librería con el archivo principal

Instrucciones

Compile el programa con compp



-Presione enter para comenzar el juego

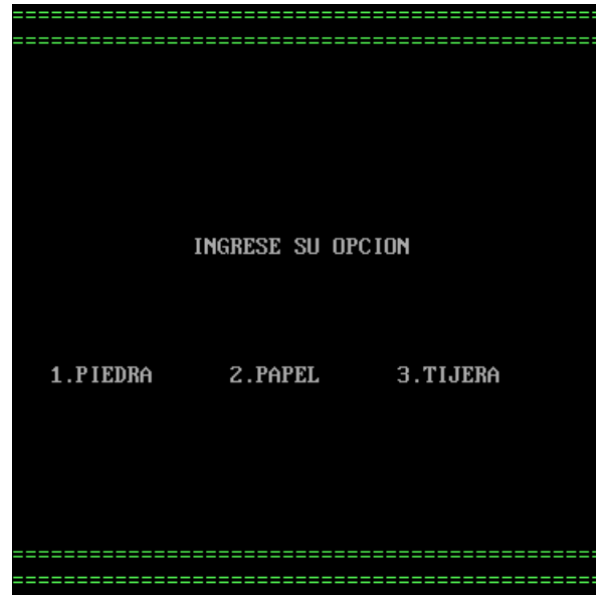


-Ingrese una opción

-1 para piedra

-2 para papel

-3 para tijera

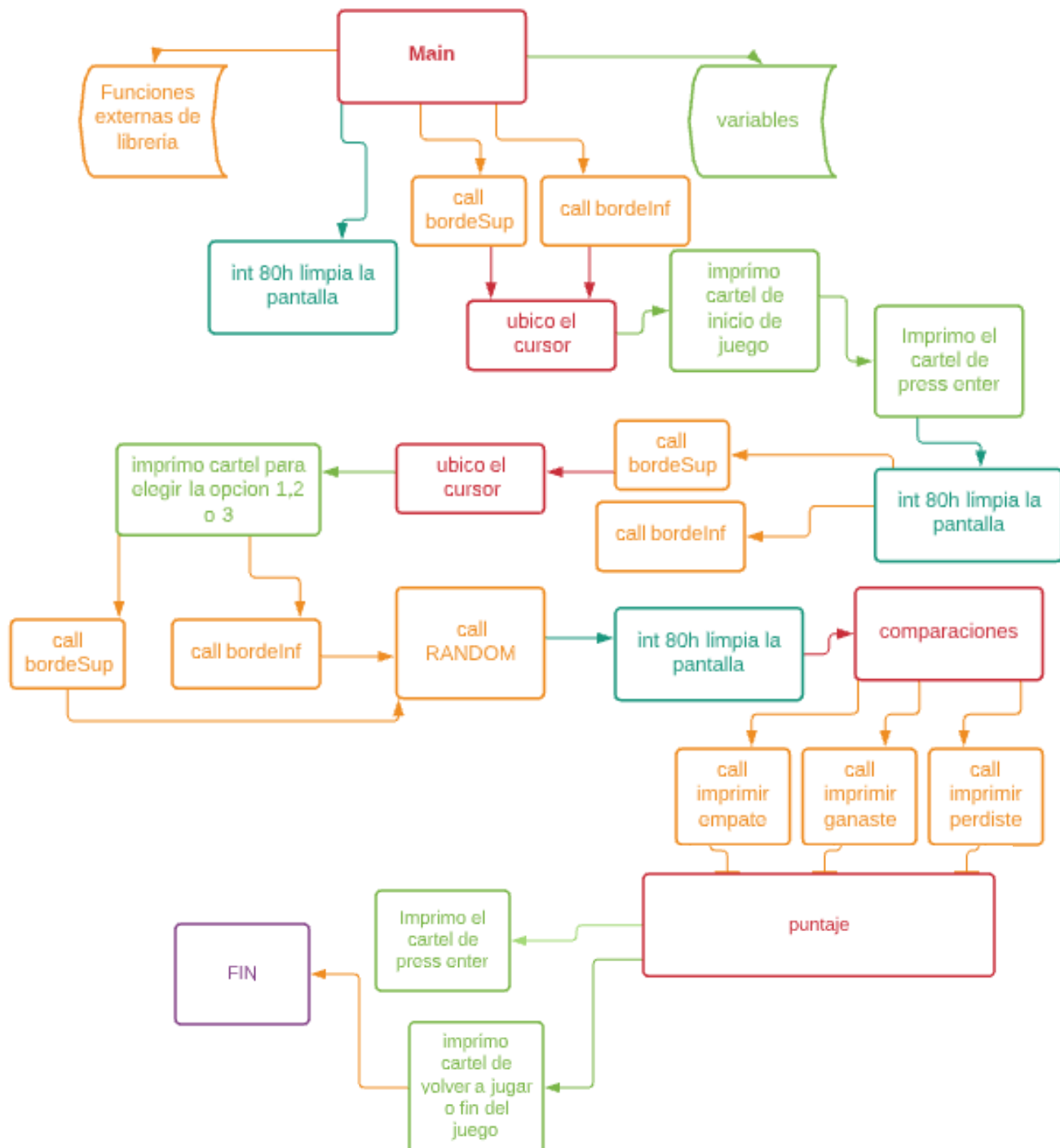


Ingrese enter para continuar

El juego termina al mejor de 3 partidas.



DIAGRAMA DE BLOQUES:





FUNCIONES IMPLEMENTADAS

```
random PROC  
  
push bx  
pushf  
    MOV     AH, 00h  
    INT     1AH  
    mov     bx, dx  
    call    CalcNew  
    xor     dx, dx  
    mov     cx, 3  
    div     cx  
    add     dl, 31h  
  
    popf  
    pop bx  
    ret  
random endp
```

FUNCIÓN RANDOM: Esta función genera un número random

que va a ser el valor que elige la máquina para poder jugar.

Devuelve un número de 1 a 3 en DL.

FUNCIÓN CALCNEW: Esta función mueve a AX un número,
multiplica bx que viene de la función random y nos da los segundos
del reloj de la computadora.

```
CalcNew PROC  
    mov     ax, 25173  
    mul     bx  
    add     ax, 13849  
    |  
    |  
    mov     bx, ax  
    SHR     AX, 5  
    ret  
CalcNew ENDP
```



FUNCIÓN BordeSup: Esta función ubica el cursor del borde superior izquierdo para dibujar la línea superior de la pantalla.

pushea los registros

xorea los registros para poder utilizarlos

llama a la interrupción 10h para imprimir el diseño en el

borde superior

llama a la interrupción 10h

llama a la interrupción 10h para imprimir el diseño en el

borde inferior

popea los registros antes pusheados

```
bordeSup PROC
    push ax
    push bx
    push dx
    push cx
    pushf

    xor ax, ax
    xor bx, bx
    xor dx, dx
    xor cx, cx

    mov ah,02h
    mov dh,2
    mov dl,0
    int 10h
    mov ah,09h
    mov al,'='
    mov bl,10
    mov cx,80
    int 10h

    mov ah,02h
    mov dh,1
    mov dl,0
    int 10h

    mov ah,09h
    mov al,'='
    mov bl,10
    mov cx,80
    int 10h
    popf
    pop cx
    pop dx
    pop bx
    pop ax
    ret
bordeSup ENDP
```

FUNCIÓN BordeInf: Esta función ubica el cursor del borde inferior para dibujar la línea inferior de la pantalla.

```
bordeInf PROC
```

```
    push ax  
    push bx  
    push dx  
    push cx  
    pushf
```

pushea los registros

```
    xor ax, ax  
    xor bx, bx  
    xor dx, dx  
    xor cx, cx
```

xorea los registros para poder utilizarlos

```
    mov ah,02h  
    mov dh,22  
    mov dl,0  
    int 10h
```

llama a la interrupción 10h para imprimir el diseño en el borde inferior

```
    mov ah,09h  
    mov al,'='  
    mov bl,10  
    mov cx,80  
    int 10h
```

```
    mov ah,02h  
    mov dh,23  
    mov dl,0  
    int 10h
```

llama a la interrupción 10h para imprimir el diseño en el borde inferior

```
    mov ah,09h  
    mov al,'='  
    mov bl,10  
    mov cx,80  
    int 10h  
    popf  
    pop cx  
    pop dx  
    pop bx  
    pop ax  
    ret
```

```
bordeInf ENDP
```



FUNCIÓN IMPRIMIR: Esta función recibe el offset de la variable en bx para poder imprimir los carteles del programa.

```
imprimir PROC
    PUSH ax
    PUSH dx
    PUSH bx
arriba:
    cmp byte ptr [bx], 24h
    je fin
    mov ah, 2
    mov dl, byte ptr [bx]
    int 21h
    inc bx
    jmp arriba

fin:
    POP bx
    POP dx
    POP ax
    ret
imprimir ENDP
```

```
ubicarCentro PROC
```

```
    push ax
    push bx
    push dx
    push cx
    pushf
```

```
    xor ax, ax
    xor bx, bx
    xor dx, dx
    xor cx, cx
```

```
    mov ah, 02h
    mov dh, 12
    mov dl, 35
    int 10h
```

```
    popf
    pop cx
    pop dx
    pop bx
    pop ax
    ret
```

```
ubicarCentro ENDP
```

FUNCIÓN UbicarCentro: esta función ubica el cursor en el centro de la pantalla para imprimir los carteles.



```
ubicarj1 PROC
    push ax
    push bx
    push dx
    push cx
    pushf

    xor ax, ax
    xor bx, bx
    xor dx, dx
    xor cx, cx

    mov ah,02h
    mov dh,9
    mov dl,25
    int 10h

    popf
    pop cx
    pop dx
    pop bx
    pop ax
    ret
ubicarj1 ENDP
```

FUNCIÓN UbicarJ1: esta función ubica el cursor en la pantalla para imprimir el cartel de la opción que eligió el usuario.

FUNCIÓN UbicarCom: esta función ubica el cursor en la pantalla para imprimir el cartel de la opción que elige la cpu.

```
ubicarCom PROC
    push ax
    push bx
    push dx
    push cx
    pushf

    xor ax, ax
    xor bx, bx
    xor dx, dx
    xor cx, cx

    mov ah,02h
    mov dh,9
    mov dl,45
    int 10h

    popf
    pop cx
    pop dx
    pop bx
    pop ax
    ret
ubicarCom ENDP
```



```
ubicarVs PROC
    push ax
    push bx
    push dx
    push cx
    pushf

    xor ax, ax
    xor bx, bx
    xor dx, dx
    xor cx, cx

    mov ah,02h
    mov dh,9
    mov dl,37
    int 10h

    popf
    pop cx
    pop dx
    pop bx
    pop ax
    ret
ubicarVs ENDP
```

FUNCIÓN UbicaVs: esta función ubica el cursor en la pantalla para imprimir el cartel VS.