**Coal Lab**



**PROJECT : Shotting Game**

**Group Members**

|  |  |  |
| --- | --- | --- |
| **Name: Hassan Maqbool** | **18F-0281** | **Sec : C** |
| **Name: Aamir sohail** | **18F-0215** | **Sec : C** |
| **Name: Zaid UL Hassan** | **18F-0168** | **Sec : A** |

**Detail**

**This game is also called shooting game as you have to shoot the thing in order to win the game The shooting speed is decided in the game But If you shoot once you have to wait for next shoot until the enemy or wall of that angle is not distrroyed**

**The object "#" are use in this game as the oponent if they reach toward you and are not completely distroyed than you loss**

**If you destroy all # object in the game you will win**

**code**

include Irvine32.inc

.data ;data segment

cnt dd 0

str1 db 70 Dup ("\_"),0

str2 db "|",0

str3 db " ",0

crtl db "(=====)",0

crtl2 db " ",0

hash db 65 DUP ("#"),0

row db 29

col db 31

fire db "^^^",0

efir db " ",0

startindex byte 3

endindex byte 19

win db "You win",0

loss db "You loss",0

var1 db 0

.code ;code segment

;//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

main proc ;//////

call board ;printing the board ;//////

call drawtarget

call racket ;printing the racket ;/////

call movRacket ;moving the racket ;/////

exit ;/////

main endp ;/////

;//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

drawtarget proc uses eax ecx edx

mov dl,2

mov dh,1

call gotoxy

mov edx,offset hash

call writestring

ret ;returning

drawtarget endp

dropdown proc

dec startindex

mov ecx, 66

mov dl,2

mov dh,startindex

call gotoxy

mov al, ' '

l1:

call writechar

loop l1

inc startindex

mov ecx, 66

mov dl,2

mov dh,startindex

call gotoxy

mov al, '@'

l2:

call writechar

loop l2

mov al, startindex

cmp al, 27

je e1

inc startindex

e1:

ret

dropdown endp

movRacket proc uses eax ecx edx ;function to move racket \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

call readchar

cmp al,97

je moveLeft

cmp al,100

je moveright

cmp al,32

je movefire

ret

movracket endp ;Ending of the mov racket ||||||||||||||||||||

movefire proc ;function to fire\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

call dropdown

mov dl,row

mov var1,dl

mov ecx,10

f1:

mov dh,var1

dec dh

mov dl,col

add dl,2

call gotoxy

mov edx,offset fire

call writestring

mov eax,100

call delay

mov dh,var1

dec dh

mov dl,col

add dl,2

call gotoxy

mov edx,offset efir

call writestring

sub var1,3

loop f1

jmp movRacket

ret

movefire endp ;ending fire function|||||||||||||||||||||||||

moveright proc uses eax ecx edx ;moving the Right Racket \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

cmp col,62

je limitbs1

mov dh,row

mov dl,col

call gotoxy

mov edx,offset str3

call writestring

inc col

mov dh,row

mov dl,col

call gotoxy

mov edx,offset crtl

call writestring

jmp movracket

ret

moveright endp ;ending the Right Racket ||||||||||||||||||||||||||||

moveLeft proc uses eax ecx edx ;moving the left \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

cmp col,1

je limitbs

mov dh,row

mov dl,col

call gotoxy

mov edx,offset crtl2

call writestring

dec col

mov dh,row

mov dl,col

call gotoxy

mov edx,offset crtl

call writestring

jmp movRacket

ret

moveLeft endp ;ending the left |||||||||||||||||

racket proc uses eax ecx edx ;function of Racket \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

mov dh,row

mov dl,col

call gotoxy

mov edx,offset crtl

call writestring

ret

racket endp ;ending of racket||||||||||||||||||||

limitbs proc uses eax ecx edx ;checking the limit of Racket\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

mov dh,row

mov dl,col

call gotoxy

mov edx,offset crtl2

call writestring

mov col,62

jmp moveleft

ret

limitbs endp ;ending the limit of racket||||||||||||||||

limitbs1 proc uses eax ecx edx ;cheking the 2nd limit\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

mov dh,row

mov dl,col

call gotoxy

mov edx,offset crtl2

call writestring

mov col,1

jmp moveright

ret

limitbs1 endp ; ;ending the limit of racket||||||||||||||||

board PROC uses eax ecx edx ;printing the board\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

mov edx,offset str1

call writestring

call crlf

mov ecx,29

l1:

mov edx,offset str2

call writestring

mov cnt ,ecx

mov ecx,68

l2:

mov edx,offset str3

call writestring

loop l2

mov ecx,cnt

mov edx,offset str2

call writestring

call crlf

loop l1

mov edx,offset str1

call writestring

ret

board endp

end main ;ending the racket |||||||||||||||||||\\

screen shoot



