**COMPUTER ORGANIZATION & ASSEMBLY LANGUAGE LAB**

**EL-229**



**MAZE PROJECT**

**Group Members**

|  |  |  |
| --- | --- | --- |
| **Name** | **Roll #** | **Section** |
| **M. Zain ul Abidin** | **18F-0114** | **C** |
| **Abdullah Ubaid** | **18F-0303** | **C** |
| **Rameez Ahmed** | **18F-0323** | **C** |

# DESCRIPTION:

It’s a maze game that features one sprite for a player. The player has to find way from one side to the other and the player must collect all the keys otherwise the door won’t open and the player will not be able to finish the game. There are certain obstacles/hurdles (walls/blockages) in the game. The keys are randomly placed at locations on the map. Moreover there is a time calculator that calculated the time elapsed to finish the game and a steps calculator that calculates the steps taken by the player.

# CODE:

include Irvine32.inc

.data

board db 42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42

db 42,00,00,00,42,00,00,00,00,00,00,00,00,00,00,00,00,00,00,42,42,42,42,42,00,00,00,00,42,42,42,42,00,00,00,42,00,42,42,42

db 42,00,42,00,42,00,42,42,42,42,42,42,42,42,42,42,00,42,00,42,00,42,00,42,00,42,42,00,00,00,00,42,00,42,00,42,00,42,00,':'

db 42,00,42,00,42,00,42,00,00,00,00,00,00,00,00,00,00,42,00,42,00,42,00,42,00,42,42,42,42,42,00,42,00,42,00,42,00,40,00,42

db 42,00,42,42,42,00,42,00,42,00,42,42,42,42,00,42,00,42,00,42,00,42,00,42,00,42,00,00,00,42,00,42,00,42,00,42,00,42,42,42

db 42,00,00,42,42,00,42,42,42,00,00,42,42,42,00,42,00,42,00,42,00,42,00,42,00,00,00,42,00,42,00,42,00,42,00,42,00,42,42,42

db 00,00,00,42,42,00,42,00,00,00,00,42,00,42,00,42,00,42,00,42,00,42,00,42,42,42,42,42,00,42,00,42,00,42,00,42,00,00,42,42

db 42,42,00,42,42,00,42,00,00,42,00,42,00,42,00,42,00,42,00,42,00,42,00,42,42,42,42,42,00,42,00,00,00,42,00,42,42,00,42,42

db 42,00,00,42,42,00,42,00,00,42,00,42,00,00,00,42,00,42,42,42,00,42,00,00,00,00,00,00,00,42,42,42,42,42,00,42,42,00,42,42

db 42,00,42,42,42,00,42,00,42,42,00,42,42,42,42,42,00,42,42,42,00,42,00,42,42,42,42,42,00,00,00,00,00,42,00,42,42,00,42,42

db 42,00,00,00,42,00,42,00,42,42,42,42,42,42,42,42,42,42,42,42,00,42,00,42,00,00,00,00,00,42,42,42,42,42,00,00,42,00,00,42

db 42,00,42,42,00,00,42,00,00,00,00,00,00,00,00,00,00,00,42,42,00,42,00,42,00,42,00,42,00,42,42,42,42,42,42,00,42,42,00,42

db 42,00,42,42,42,00,42,42,42,42,42,42,42,42,42,00,42,42,42,42,42,42,00,42,00,42,00,42,00,42,42,00,42,00,42,00,42,42,00,42

db 42,00,42,42,42,00,42,00,00,00,00,00,00,00,00,00,42,42,42,42,42,42,00,42,00,42,00,42,00,42,42,00,00,00,42,00,00,00,00,42

db 42,00,42,42,42,00,42,00,42,42,42,42,42,42,42,42,42,00,00,00,00,42,00,42,00,42,00,00,00,00,42,00,00,00,42,00,42,42,42,42

db 42,00,00,42,42,00,42,00,42,42,42,42,42,42,42,42,42,00,42,42,00,42,42,42,00,42,42,00,42,00,42,00,00,00,00,00,42,42,42,42

db 42,00,00,42,42,00,42,00,00,00,00,00,00,00,00,42,00,00,42,00,00,42,00,00,00,42,42,00,42,00,42,42,42,42,42,00,42,42,42,42

db 42,42,00,00,00,00,42,00,42,42,42,42,42,42,42,00,42,00,42,42,00,42,00,42,00,42,42,00,42,00,42,42,42,42,00,00,00,00,00,42

db 42,00,00,42,42,42,42,00,00,00,00,00,00,00,00,00,00,00,00,42,00,00,00,42,42,42,42,00,00,00,42,42,42,42,42,00,42,42,42,42

db 42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42,42

menuStr db " \_\_\_ \_\_\_ \_\_\_\_\_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_",0dh,0ah

rowMenu = $-menuStr

db " | \ / | | \_\_\_\_| | \ | | | | | |",0dh,0ah

db " | \ / | | |\_\_\_\_ | \ | | | | | |",0dh,0ah

db " | |\\ //| | | \_\_\_\_| | |\\ | | | | | |",0dh,0ah

db " | | \\ // | | | |\_\_\_\_ | | \\| | | |\_\_\_\_| |",0dh,0ah

db " |\_\_\_| \\_/ |\_\_\_| |\_\_\_\_\_\_\_| |\_\_\_| \\_\_\_\_| |\_\_\_\_\_\_\_\_\_\_\_\_|",0

str2 db " \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_",0dh,0ah

db " \ \ / \ / / | | | \ | |",0dh,0ah

db " \ \ / \ / / | | | \ | |",0dh,0ah

db " \ \/ / \ \/ / | | | |\\\_| |",0dh,0ah

db " \\_\_\_\_\_\_/ \\_\_\_\_\_/ |\_\_\_\_| |\_\_\_\_| \\_\_\_\_\_\_|",0

count db 0

playerRow db 6

playerColumn db 0

;playerRow db 3

;playerColumn db 36

chkKEY WORD 0

CPR db 6

CPC db 0

RowSize db 40

str1 db "Keys collected : ",0

str3 db "Time Used in seconds : ", 0

str4 db "Steps Taken : ",0

keyScore db 0

cond db 0

winRow db 2

winCol db 39

win db 0

divisor DD ?

TimeTaken DD ?

startTime DD ?

Steps Word 0

Space WORD 20

CHR WORD 40

;totalKeysTobeCollected db 1

totalKeysTobeCollected db 13

doorRow db 3

doorColumn db 37

msg db "LOADING...", 0

caption db "All keys not collected", 0

str5 BYTE "The door is not unlocked, collect all the keys to unlock it.", 0dh,0ah

BYTE "Click OK to continue the game...", 0

theDoorIsOpenCondition db 0

menu1 db "1) Start the game.",0

menu2 db "2) Exit the game.",0

menu3 db "Your choice: ",0

.code

main proc

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

backToStart:

mov eax,BLUE+(white\*16)

call SetTextColor

call mainMenu

cmp eax, 2

je timeToEnd

mov eax, 0

mov ebx, offset board

mov cl, playerRow

mov al, RowSize

mul cl

add ebx, eax

movzx esi, playerColumn

mov al, 64

mov [ebx+esi], al

INVOKE GetTickCount ; Moves Current Time in eax.

mov startTime,eax

call randomKeys

mov edi,offset board

add edi, 120

mov bl, win

back:

call displayBoard

call playerMove

call youWon

call clrscr

mov al, totalKeysToBeCollected

cmp al, 0

jne doorOpen

mov cl, theDoorIsOpenCondition

cmp cl ,0

jne dontCallOpenDoor

call openTheDoors

doorOpen:

dontCallOpenDoor:

mov bl, win

cmp bl, 0

je back

mov edx, offset str2

call writestring

mov eax, 3000

call delay

jmp backToStart

timeToEnd:

exit

main endp

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

mainMenu proc uses ecx edx

itsNotfine:

call clrscr

;1\_

mov edx, offset menuStr

call writestring

call crlf

call crlf

call crlf

call crlf

call crlf

mov edx, offset menu1

call writestring

call crlf

call crlf

call crlf

mov edx, offset menu2

call writestring

call crlf

call crlf

call crlf

mov edx, offset menu3

call writestring

call readdec

cmp eax,1

je itsFine

cmp eax,2

je itsFine

jne itsNotfine

itsFine:

ret

mainMenu endp

randomKeys proc uses ecx ebx esi eax edx

mov edx, offset msg

call writeString

movzx ecx, totalKeysTobeCollected

mov edx, 65

mov ebx, offset board

randomK:

mov eax, 20

call randomize

call randomRange

mul rowSize

add ebx, eax

;mov eax, 500

;call delay

mov eax, 40

call randomize

call randomRange

;mul rowSize

mov esi, eax

mov al, [ebx+esi]

cmp al, 00

jne notFruitful

mov [ebx+esi],dl

inc edx

dec ecx

notFruitful:

mov ebx, offset board

cmp ecx, 0

jg randomK

call clrscr

ret

randomKeys endp

displayBoard proc uses esi ecx eax edx ebx

mov eax,0

MOV EDX,20

mov ebx, offset board

mov esi, 0

call crlf

call crlf

call crlf

display1:

mov ecx,40

mov al,0

L1:

call writeChar

Loop L1

mov ecx, 40

L212:

mov al,[ebx+esi]

call TokenCheck

cmp chkkey, 0

je COLR2

cmp al,40h

je COLR1

jmp COLR0

COLR1:

mov eax,Red+(white\*16)

call SetTextColor

jmp nxt

COLR2:

mov ah,0

mov eax,magenta+(white\*16)

call SetTextColor

jmp nxt

COLR0:

mov eax,BLUE+(white\*16)

call SetTextColor

nxt:

mov eax,0

mov al,[ebx+esi]

call writechar

inc esi

LOOP L212

add ebx, 40

call crlf

mov esi, 0

dec dx

cmp dx, 0

jne display1

mov edx, offset str1

call writeString

mov al, keyScore

call writedec

call crlf

call TimeUsed

call crlf

mov edx,offset str4

call writeString

movzx eax,steps

call WriteDec

call crlf

ret

displayBoard endp

TokenCheck Proc uses eax ebx

cmp al, 40

je L1

cmp al, ' '

je L1

cmp al, '('

je L1

cmp al, '\*'

je L1

cmp al, ':'

je L1

jmp L2

L1:

mov bx, 0

mov chkKey, bx

jmp ext

L2:

mov bx, 1

mov chkKey, bx

ext:

ret

TokenCheck endp

playerMove proc uses eax ecx esi ebx edi

call StepsTakenInc

mov al,playerRow

mov CPR, al

mov al, playerColumn

mov CPC, al

mov eax, 0

mov ebx, offset board

movzx eax, playerRow

mov cl, rowSize

mul cl

add ebx, eax

mov al, playerColumn

movzx esi, al

call readchar

cmp ax, 4d00h

je moveRight

cmp ax, 4b00h

je moveLeft

cmp ax, 4800h

je moveUp

cmp ax, 5000h

je moveDown

jmp noUse

moveRight:

add cond,1

inc esi

mov al, [ebx+esi]

call keyBonus

cmp al, 42

je noUse

cmp al, 40

je theDoor

dec esi

mov al, [ebx+esi]

mov cl,0

mov [ebx+esi], cl

inc esi

mov [ebx+esi],al

inc playerColumn

sub cond,1

jmp noUse

moveLeft:

add cond,2

dec esi

mov al, [ebx+esi]

call keyBonus

cmp al, 42

je noUse

inc esi

mov al, [ebx+esi]

mov cl,0

mov [ebx+esi], cl

dec esi

mov [ebx+esi],al

dec playerColumn

sub cond,2

jmp noUse

moveUp:

add cond,3

mov edi, ebx

mov ebx, offset board

mov cl, rowSize

movzx eax, playerRow

dec eax

mul cl

add ebx, eax

mov al, [ebx+esi]

call keyBonus

cmp al, 42

je noUse

xchg edi, ebx

mov al, [ebx+esi]

mov cl,0

mov [ebx+esi], cl

xchg edi, ebx

mov [ebx+esi],al

dec playerRow

sub cond,3

jmp noUse

moveDown:

add cond, 4

mov edi, ebx

mov ebx, offset board

mov cl, rowSize

movzx eax, playerRow

inc eax

mul cl

add ebx, eax

mov al, [ebx+esi]

call keyBonus

cmp al, 42

je noUse

xchg edi, ebx

mov al, [ebx+esi]

mov cl,0

mov [ebx+esi], cl

xchg edi, ebx

mov [ebx+esi],al

inc playerRow

sub cond, 4

jmp noUse

theDoor:

mov ebx,OFFSET caption

mov edx,OFFSET str5

call MsgBox

noUse:

ret

playerMove endp

keyBonus proc uses eax

mov cl, 65

back1:

cmp al, cl

je collected

inc cl

cmp cl, 77

jbe back1

jmp toTheEnd

collected:

inc keyScore

dec totalKeysTobeCollected

toTheEnd:

ret

keyBonus endp

TimeUsed Proc uses eax edx

INVOKE GetTickCount ;Moves current time in eax register

sub eax, startTime

mov divisor, 1000

mov edx, Offset str3

call WriteString

mov edx,0

mov timeTaken, eax

mov ebx, divisor

div ebx

call WriteDec

ret

TimeUsed endp

StepsTakenInc proc uses eax ebx

mov al,playerColumn

cmp al,CPC

JZ C1

jmp C2

C1:

mov bl,playerRow

cmp bl,CPR

JZ N

C2:

mov ax, steps

inc ax

mov steps,ax

N:

ret

StepsTakenInc endp

youWon Proc uses eax ecx

mov al, winRow

mov cl, winCol

cmp al, playerRow

je check1

jne endall

check1:

cmp cl, playerColumn

je check2

jne endall

check2:

mov win, 1

endall:

ret

youwon Endp

openTheDoors proc uses eax ecx

mov al, totalKeysTobeCollected

cmp al, 0

jne false1

mov ebx, offset board

movzx eax, rowsize

mov cl, doorRow

mul cl

add ebx, eax

movzx esi, doorColumn

mov al, 00

mov [ebx+esi], al

mov theDoorIsOpenCondition, 1

false1:

ret

openTheDoors endp

end main

# SCREENSHOT/s:







