|  |  |
| --- | --- |
| Name | Rai Ali Yar |
| Roll No | 22F – 3361 |
| Team Name | DUMB Developer |

[org 0x0100]

jmp start

;;;;;;;;;;;;;;;;;;;;;;;; DATA ;;;;;;;;;;;;;;;;;;;;;;;;;

HEIGHT: dw 24

WIDTH: dw 50

basketx: dw 38

baskety: dw 22

basket: dw 0x42db

fruit: dw 0x4401

score: dw 0

score\_str: db 'SCORE'

timer\_str: db 'TIMER'

timer\_loc: dw 1610

score\_loc: dw 970

timer: dw 0

timer\_counter: dw 0

x: db '---------------------------'

game\_name: db '| FRUIT CATCHER | '

team\_name: db 'By DUMB<Unexpected> Developer'

to\_play: db 'PRESS ANY KEY TO PLAY THE GAME'

over: db 'GAME OVER'

your\_score: db 'YOUR SCORE: '

myname: db ' | RAI ALI YAR |'

roll\_number: db '22F-3361'

credit: db 'CREDITS'

again: db 'PRESS ANY KEY TO PLAY AGAIN'

escape: db 'PRESS ESC TO EXIT'

team: db ' DUMB Developer '

instructions: db 'CATCH AS MANY FRUITS AS YOU CAN IN 60 SECONDS.'

instructions2: db 'GET READY TO SCORE BIG!'

;;;;;;;;;;;;;;;;;;;;;; START ;;;;;;;;;;;;;;;;;;;;;;;;;

start:

call print\_title

mov ah, 0

int 16h

call clrscr

call Board

push word[basket]

call draw\_basket

mov si, score\_str ; "score"

mov cx, 5

mov di, 810

call printstr

mov si, timer\_str ; "TIMER"

mov cx, 5

mov di, 1450

call printstr

mov cx, 0xffff

game\_loop:

call move\_Fruits

call Board

call genFruit

push word[score]

push word[score\_loc]

call printnum

inc word[timer\_counter]

cmp word[timer\_counter], 50

jb continue\_loop

mov word[timer\_counter], 0

inc word[timer]

push word[timer]

push word[timer\_loc]

call printnum

cmp word[timer], 60

je game\_over

continue\_loop:

mov ah, 0x01

int 0x16

jz NO\_Input

mov ah, 0x00

int 0x16

call check\_Input

NO\_Input:

call delay

jmp game\_loop

;;;;;;;;;;;;;;;;;;;;; END ;;;;;;;;;;;;;;;;;;;;;;

game\_over:

call colour\_screen

call gameOver

mov ah, 0

int 0x16

mov ah, 01

int 0x16

cmp al, 0x1B

je byebye

mov word[score], 0

mov word[timer], 0

jmp start

byebye:

call credits

;;;;;;;;;;;;;;;;;;;;; EXIT ;;;;;;;;;;;;;;;;;;;;;;

exit:

mov ax, 0x4c00

int 0x21

;;;;;;;;;;;;;;;;;;;;;; NUMBER PRINTER ;;;;;;;;;;;;;;;;;;;;;;;;;;;;

printnum:

push bp

mov bp, sp

push es

push ax

push bx

push cx

push dx

push di

mov ax, 0xb800

mov es, ax

mov ax, [bp+6]

mov bx, 10

mov cx, 0

nextdigit:

mov dx, 0

div bx

add dl, 0x30

push dx

inc cx

cmp ax, 0

jnz nextdigit

mov di, [bp+4]

nextpos:

pop dx

mov dh, 0x34

mov [es:di], dx

add di, 2

loop nextpos

pop di

pop dx

pop cx

pop bx

pop ax

pop es

pop bp

ret 4

;;;;;;;;;;;;;;;;;;;;;;;;; STRING PRINTER ;;;;;;;;;;;;;;;;;;;;;;;;

printstr:

push es

push ax

mov ax, 0xb800

mov es, ax

mov ax, 0x3420

psloop1:

lodsb

stosw

loop psloop1

pop ax

pop es

ret

;;;;;;;;;;;;;;;;;;;;; Title Page ;;;;;;;;;;;;;;;;;;;;;;;;;;

print\_title:

push si

push cx

push di

call clrscr

mov si, x

mov cx, 27

mov di, 844

call printstr

mov si, game\_name

mov cx, 17

mov di, 1176

call printstr

mov si, x

mov cx, 27

mov di, 1484

call printstr

mov si, team\_name

mov cx, 29

mov di, 1812

call printstr

mov si, instructions

mov cx, 46

mov di, 2430

call printstr

mov si, instructions2

mov cx, 23

mov di, 2606

call printstr

mov si, to\_play

mov cx, 30

mov di, 3080

call printstr

mov si, x

mov cx, 27

mov di, 3404

call printstr

pop di

pop cx

pop si

ret

;;;;;;;;;;;;;;;;;;;;; CLEAR SCREEN ;;;;;;;;;;;;;;;;;;;;;;;;;;

clrscr:

push es

push ax

push cx

push di

mov ax, 0xb800

mov es, ax

xor di, di

mov ax, 0x3020 ; 3 is for the background color

mov cx, 2000

cld

rep stosw

pop di

pop cx

pop ax

pop es

ret

;;;;;;;;;;;;;;;;;;;;; DRAW BOARD ;;;;;;;;;;;;;;;;;;;;;;;;;;;;

Board:

push ax

push bx

push cx

push es

push di

mov ax, 0xb800

mov es, ax

mov di, 32 ; border starting CORDINATES

mov cx, [WIDTH]

dec cx

mov ax, 0x32cd ; above line above =

mov word[es:di-2], 0x31c9 ; left above corner

cld

rep stosw

mov word[es:di-2], 0x31bb ; right above corner

mov di, 30

mov bx, [WIDTH]

dec bx

shl bx, 1

mov cx, [HEIGHT]

mov ax, 0x3220

side\_board:

add di, 160

mov word[es:di], 0x32ba

mov word[es:di+bx], 0x32ba

loop side\_board

mov cx, [WIDTH]

dec cx

mov ax, 0x32cd

mov word[es:di], 0x31c8 ; bottom left corner

add di, 2

cld

rep stosw

mov word[es:di-2], 0x31bc ; bottom right corner

pop di

pop es

pop cx

pop bx

pop ax

ret

;;;;;;;;;;;;;;;;;;;; DRAW BASKET ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

draw\_basket:

push bp

mov bp,sp

push ax

push cx

push di

push es

mov ax,0xb800

mov es,ax

mov ax,80

mul byte[baskety]

add ax,[basketx]

shl ax,1

mov di,ax

sub di, 4

mov ax,[bp + 4]

mov ah,0x31 ;if want to change color of basket change the second attribute first is for the blue background

;also change in mfskip cmp statement

mov word[es:di], ax

mov cx, 11

push di

rep stosw

pop di ; prev value of di position ki save kr lety hain

add di,164

mov cx, 7

push di

rep stosw

pop di

; if want to increse size of basket use 3 in cx and decrease y cord by 1 to 21

pop es

pop di

pop cx

pop ax

pop bp

ret 2

;;;;;;;;;;;;;;;;;;;;;;;;; GENERATE RANDOM ;;;;;;;;;;;;;;;;;;;;;;;;;;;

gen\_random:

push bx

push dx

mov bx, ax

rdtsc

div bx

mov ax, dx

pop dx

pop bx

ret

;;;;;;;;;;;;;;;;;;; GENERATE FRUITS ;;;;;;;;;;;;;;;;;;;;;;;;;

genFruit:

pushad

mov ax, 0xb800

mov es, ax

mov ax, 30

call gen\_random

cmp ax, 0

jne gFexit

mov ax, 2

call gen\_random

mov cx, ax

cmp cx, 0

je gFexit

generateAgain:

mov ax, [WIDTH]

sub ax, 6

call gen\_random

add ax, 18

mov di, ax

shl di, 1

add di, 160

mov word[es:di], 0x4401

loop generateAgain

gFexit:

popad

ret

;;;;;;;;;;;;;;;;;;;;;;; CHECK INPUT ;;;;;;;;;;;;;;;;;;;;;;;;;;;

check\_Input:

pushad

cmp al, 'a'

jne chiskip1

cmp word[basketx], 18

je chiskip1

mov ax, 0x0020

push ax

call draw\_basket

dec word[basketx]

push word[basket]

call draw\_basket

jmp chiexit

chiskip1:

cmp al, 'd'

jne chiskip2

cmp word[basketx], 55 ; border

je chiskip2

mov ax, 0x0020

push ax

call draw\_basket

inc word [basketx]

push word[basket]

call draw\_basket

jmp chiexit

chiskip2:

cmp al, 0x1B ; ASCII of "esc" key

je game\_over

chiexit:

popad

ret

;;;;;;;;;;;;;;;;;;;;;;;;; DELAY loop ;;;;;;;;;;;;;;;;;;;;;;;;;

delay:

push cx

mov cx, 0xffff

dlloop2:

loop dlloop2

pop cx

ret

;;;;;;;;;;;;;;;;;;;;;; MOVE FRUITS ;;;;;;;;;;;;;;;;;;;;;;;;;;;

move\_Fruits:

push ax

push es

push di

push cx

push si

mov ax, 0xb800

mov es, ax

mov cx, 2000

mov di, 0

mov si, 4000

mov ax, [fruit]

mfloop1:

cmp word[es:di], 0x44f8

jne mfskip

mov word[es:di], 0x3020

cmp word[es:di-160], ax

jne mfskip12

inc word[score]

mov word[es:di-160], 0x3020

jmp mfskip

mfskip12:

mov word[es:di-160], 0x44f8

mfskip:

cmp word[es:si], ax

jne mfskip2

cmp word[es:si+160], 0x31db

jne mfskip21

inc word[score]

mov word[es:si], 0x3020

jmp mfskip2

mfskip21:

mov word[es:si], 0x3020

mov word[es:si+160], ax

mfskip2:

sub si, 2

add di, 2

loop mfloop1

mfexit:

pop si

pop cx

pop di

pop es

pop ax

ret

;;;;;;;;;;;;;;;;;;;;; GameOver Page ;;;;;;;;;;;;;;;;;;;;;;;;;;

gameOver:

push si

push cx

push di

call clrscr

mov si, over

mov cx, 9

mov di, 1670

call printstr

mov si, your\_score

mov cx, 12

mov di, 1986

call printstr

push word[score]

mov ax, 2010

push ax

call printnum

mov si, again

mov cx, 27

mov di, 2292

call printstr

mov si, escape

mov cx, 17

mov di, 2622

call printstr

mov si, team

mov cx, 17

mov di, 3902

call printstr

pop di

pop cx

pop si

ret

;;;;;;;;;;;;;;;;;;;;; Credits Page ;;;;;;;;;;;;;;;;;;;;;;;;;;

credits:

push si

push cx

push di

call clrscr

mov si, x

mov cx, 27

mov di, 1012

call printstr

mov si, credit

mov cx, 7

mov di, 1352

call printstr

mov si, x

mov cx, 27

mov di, 1652

call printstr

mov si, myname

mov cx, 21

mov di, 1980

call printstr

mov si, roll\_number

mov cx, 8

mov di, 2152

call printstr

mov si, team

mov cx, 17

mov di, 3902

call printstr

pop di

pop cx

pop si

ret

;;;;;;;;;;;;;;;;;;;;;;;; COLOUR ;;;;;;;;;;;;;;;;;;;;;;;;;;

colour:

push bp

mov bp, sp

push es

push ax

mov ax, 0xb800

mov es, ax

mov ax, [bp+4]

mov di, 0

mov cx, 4000

psloop:

mov [es:di], ax

add di, 2

dec cx

jnz psloop

pop ax

pop es

pop bp

ret 2

colour\_screen:

push dx

push cx

call clrscr

mov dx, 0x2020

push dx

call colour

mov cx, 5

ag:

call delay

loop ag

mov dx, 0x4020

push dx

call colour

mov cx, 5

ag1:

call delay

loop ag1

mov dx, 0x1020

push dx

call colour

mov cx, 5

ag2:

call delay

loop ag2

mov dx, 0x6020

push dx

call colour

mov cx, 5

ag3:

call delay

loop ag3

pop cx

pop dx

ret

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;; ;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;; THAT'S ALL FOLKS 😁 😎 😏 🤐 ;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;; ;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;; ;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;; Izat Zilat meray Allah ky hath mai hy ;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;; ;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;; ;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;; و َتُعِزُّ مَنۡ تَشَآءُ وَتُذِلُّ مَنۡ تَشَآءُ ;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;; ;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;; ;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;; كُلُّ نَفْسٍ ذَآئِقَةُ الْمَوْت ;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;; ;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;