**COAL Project**

**Made By:**  
  
22K-4492 Syed Muhammad Bilal

22K-4143 Aadil Raja

**Code:**

INCLUDE Irvine32.inc

.data

strScore BYTE "Your final score is: ",0

score BYTE 0

loading BYTE "-------------------------------------Game is Loading! Please Wait!------------------------------------------------------",0

introduction BYTE" ",0Ah

BYTE" ",0Ah

BYTE " 1) Welcome to our snake game! ",0Ah

BYTE " 2) Navigate the snake using the W,A,S,D",0Ah

BYTE " 3) Eat the Food to grow longer and earn points",0Ah

BYTE " 4) Avoid Obstacles in your Path ",0Ah,0

strTryAgain BYTE "To play again press 1, to quit press 0: ", 0

strHandleDeath BYTE "Game Over! Better luck next time!", 0

strNames BYTE "BY Aadil Raja and S.Bilal",0

strPoints BYTE " Total point(s)",0

blank BYTE " ",0

snake BYTE "S", 50 DUP("s")

xPos BYTE 50,49,48,47,46, 50 DUP(?)

yPos BYTE 18,18,18,18,18, 50 DUP(?)

ObstaclePos BYTE 40, 13, 30, 9, 45, 12, 50, 10,60,11,65,17,70,18,80,19,85,20

xWall BYTE 15,15,90,90

yWall BYTE 5,24,5,24

obsx BYTE 20,50

obsy BYTE 6,15

xfoodPos BYTE ?

yfoodPos BYTE ?

input BYTE "+"

lastInput BYTE ?

strSpeed BYTE "Please Choose Speed (1-2x, 2-1.5x, 3-1x): ",0

speed DWORD 0

factor DWORD 70

SCOREBOARD\_WIDTH equ 30

SCOREBOARD\_HEIGHT equ 3

SCOREBOARD\_TOP equ 1

SCOREBOARD\_LEFT equ 1

prompt BYTE "Snake Game",0

.code

main PROC

mov eax,(yellow\*16)+blue

call settextcolor

mov dh,60

mov dl,127

call gotoxy

mov edx,OFFSET introduction

call writestring

call crlf

call crlf

mov edx,offset loading

call writestring

mov eax,5000

call delay

call clrscr

call DrawWall

call DrawObstacles

call GetSpeed

mov dh,0

mov dl,3

call gotoxy

mov edx,offset prompt

call writestring

mov dh,25

mov dl,30

call gotoxy

mov edx,offset strNames

call writestring

call DrawScoreboardBorder

mov dh, SCOREBOARD\_TOP + 1 ;

mov dl, SCOREBOARD\_LEFT + 1 ;

call Gotoxy

mov edx, OFFSET strScore

call WriteString

mov esi,0

mov ecx,5

drawSnake:

call DrawingSnake

inc esi

loop drawSnake

call Randomize

call CreateRandomfood

call Drawfood

snakegameloop:

mov dl,100

mov dh,0

call Gotoxy

call ReadKey

jz NoKey

GetInput:

mov bl, input

mov lastInput, bl

mov input,al

NoKey:

cmp input,"w"

je upward

cmp input,"s"

je downward

cmp input,"x"

je exitgame

cmp input,"a"

je left

cmp input,"d"

je right

jmp snakegameloop

downward:

cmp lastInput, "w"

je dontChgDirection

mov cl, yWall[1]

dec cl

push ecx

call CheckObstacleCollision

pop ecx

cmp yPos[0],cl

jl moveDown

je died

left:

cmp lastInput, "d"

je dontChgDirection

mov cl, xWall[0]

inc cl

push ecx

call CheckObstacleCollision

pop ecx

cmp xPos[0],cl

jg moveLeft

je died

right:

cmp lastInput, "a"

je dontChgDirection

mov cl, xWall[2]

dec cl

push ecx

call CheckObstacleCollision

pop ecx

cmp xPos[0],cl

jl moveRight

je died

upward:

cmp lastInput, "s"

je dontChgDirection

mov cl, yWall[0]

inc cl

push ecx

call CheckObstacleCollision

pop ecx

cmp yPos,cl

jl moveUp

je died

moveUp:

mov eax, speed

call delay

mov esi, 0

call UpdateSnake

mov ah, yPos[esi]

mov al, xPos[esi]

dec yPos[esi]

call DrawingSnake

call DrawingBody

call Check\_Snake

moveDown:

mov eax, speed

call delay

mov esi, 0

call UpdateSnake

mov ah, yPos[esi]

mov al, xPos[esi]

inc yPos[esi]

call DrawingSnake

call DrawingBody

call Check\_Snake

moveLeft:

mov eax, speed

call delay

mov esi, 0

call UpdateSnake

mov ah, yPos[esi]

mov al, xPos[esi]

dec xPos[esi]

call DrawingSnake

call DrawingBody

call Check\_Snake

moveRight:

mov eax, speed

call delay

mov esi, 0

call UpdateSnake

mov ah, yPos[esi]

mov al, xPos[esi]

inc xPos[esi]

call DrawingSnake

call DrawingBody

call Check\_Snake

checkfood::

mov esi,0

mov bl,xPos[0]

cmp bl,xfoodPos

jne snakegameloop

mov bl,yPos[0]

cmp bl,yfoodPos

jne snakegameloop

call Eatingfood

jmp snakegameloop

dontChgDirection:

mov input, bl

jmp NoKey

dontGoLeft:

mov input, "+"

jmp snakegameloop

died::

call HandleDeath

playagain::

call RestartGame

exitgame::

exit

main ENDP

DrawHorizontalLine PROC

pusha

mov ecx, SCOREBOARD\_WIDTH

mov al, "-"

DrawLine:

call WriteChar

loop DrawLine

popa

ret

DrawHorizontalLine ENDP

DrawObstacles PROC

mov ecx, lengthof ObstaclePos / 2

mov ebx, 0

DrawObstacleLoop:

mov dl, ObstaclePos[ebx]

mov dh, ObstaclePos[ebx + 1]

mov al, '|'

call Gotoxy

call WriteChar

add ebx, 2

loop DrawObstacleLoop

ret

DrawObstacles ENDP

DrawScoreboardBorder PROC

mov dh, SCOREBOARD\_TOP

mov dl, SCOREBOARD\_LEFT

call Gotoxy

call DrawHorizontalLine

mov dh, SCOREBOARD\_TOP + SCOREBOARD\_HEIGHT

mov dl, SCOREBOARD\_LEFT

call Gotoxy

call DrawHorizontalLine

ret

DrawScoreboardBorder ENDP

CheckObstacleCollision PROC

mov esi, 0

movzx eax, xPos[0]

movzx ebx, yPos[0]

CheckCollisionLoop:

cmp esi, (lengthof ObstaclePos)

jae NoCollision

mov cl, ObstaclePos[esi]

cmp cl, al

jne NextObstacle

mov cl, ObstaclePos[esi+1]

cmp cl, bl

je CollisionDetected

NextObstacle:

add esi, 2

jmp CheckCollisionLoop

CollisionDetected:

call HandleDeath

ret

NoCollision:

ret

CheckObstacleCollision ENDP

DrawWall PROC

mov eax, "#"

mov dh, yWall[0]

mov dl, xWall[0]

TopWallLoop:

call Gotoxy

call WriteChar

inc dl

cmp dl, xWall[2]

jl TopWallLoop

mov dh, yWall[1]

mov dl, xWall[0]

BottomWallLoop:

call Gotoxy

call WriteChar

inc dl

cmp dl, xWall[2]

jl BottomWallLoop

mov dl, xWall[0]

mov dh, yWall[0]

LeftWallLoop:

call Gotoxy

call WriteChar

inc dh

cmp dh, yWall[1]

jl LeftWallLoop

mov dl, xWall[2]

mov dh, yWall[0]

RightWallLoop:

call Gotoxy

call WriteChar

inc dh

cmp dh, yWall[1]

jl RightWallLoop

ret

DrawWall ENDP

GetSpeed PROC

mov edx,0

mov dl,45

mov dh,1

call Gotoxy

mov edx,OFFSET strSpeed

call WriteString

mov eax,0

call readInt

mul factor

mov speed, eax

ret

GetSpeed ENDP

DrawingSnake PROC

mov dl,xPos[esi]

mov dh,yPos[esi]

call Gotoxy

mov dl, al

mov al, snake[esi]

call WriteChar

mov al, dl

ret

DrawingSnake ENDP

UpdateSnake PROC

mov dl, xPos[esi]

mov dh,yPos[esi]

call Gotoxy

mov dl, al

mov al, " "

call WriteChar

mov al, dl

ret

UpdateSnake ENDP

Drawfood PROC

mov dl,xfoodPos

mov dh,yfoodPos

call Gotoxy

mov al,"F"

call WriteChar

ret

Drawfood ENDP

CreateRandomfood PROC

mov eax, 74

call RandomRange

add eax, 16

mov xfoodPos, al

mov eax, 17

call RandomRange

add eax, 6

mov yfoodPos, al

GeneratefoodLoop:

movzx ecx, score

add ecx, 5

mov esi, 0

CheckfoodPosition:

movzx eax, xfoodPos

cmp al, xPos[esi]

jne NoMatchX

movzx eax, yfoodPos

cmp al, yPos[esi]

je Regeneratefood

NoMatchX:

inc esi

cmp esi, ecx

jb CheckfoodPosition

mov ecx, lengthof ObstaclePos / 2

mov esi, 0

CheckObstaclePosition:

mov al, ObstaclePos[esi\*2]

cmp al, xfoodPos

jne NoMatchObstacleX

mov al, ObstaclePos[esi\*2 + 1]

cmp al, yfoodPos

je Regeneratefood

NoMatchObstacleX:

inc esi

cmp esi, ecx

jb CheckObstaclePosition

ret

Regeneratefood:

jmp GeneratefoodLoop

CreateRandomfood ENDP

Check\_Snake PROC

mov al, xPos[0]

mov ah, yPos[0]

mov esi,4

mov ecx,1

add cl,score

checkXposition:

cmp xPos[esi], al

je XposSame

contloop:

inc esi

loop checkXposition

jmp checkfood

XposSame:

cmp yPos[esi], ah

je died

jmp contloop

Check\_Snake ENDP

DrawingBody PROC

mov ecx, 4

add cl, score

printbodyloop:

inc esi

call UpdateSnake

mov dl, xPos[esi]

mov dh, yPos[esi]

mov yPos[esi], ah

mov xPos[esi], al

mov al, dl

mov ah,dh

call DrawingSnake

cmp esi, ecx

jl printbodyloop

ret

DrawingBody ENDP

Eatingfood PROC

inc score

mov ebx, 4

add bl, score

mov esi, ebx

mov ah, yPos[esi-1]

mov al, xPos[esi-1]

mov xPos[esi], al

mov yPos[esi], ah

cmp xPos[esi-2], al

jne yPosAdjustment

cmp yPos[esi-2], ah

jl increaseYPosition

jg decreaseYPosition

increaseYPosition:

inc yPos[esi]

jmp updateGame

decreaseYPosition:

dec yPos[esi]

jmp updateGame

yPosAdjustment:

cmp yPos[esi-2], ah

jl increaseXPosition

jg decreaseXPosition

increaseXPosition:

inc xPos[esi]

jmp updateGame

decreaseXPosition:

dec xPos[esi]

updateGame:

call DrawingSnake

call CreateRandomfood

call Drawfood

mov dl, 30

mov dh, 2

call Gotoxy

mov al, score

call WriteInt

push eax

mov eax,(blue\*16)+yellow

pop eax

ret

Eatingfood ENDP

HandleDeath PROC

mov eax, 1000

call delay

Call ClrScr

mov dl, 57

mov dh, 12

call Gotoxy

mov edx, OFFSET strHandleDeath

call WriteString

mov dl, 56

mov dh, 14

call Gotoxy

movzx eax, score

call WriteInt

mov edx, OFFSET strPoints

call WriteString

mov dl, 50

mov dh, 18

call Gotoxy

mov edx, OFFSET strTryAgain

call WriteString

retry:

mov dh, 19

mov dl, 56

call Gotoxy

call ReadInt

cmp al, 1

je playagain

cmp al, 0

je exitgame

HandleDeath ENDP

RestartGame PROC

mov xPos[0], 50

mov xPos[1], 49

mov xPos[2], 48

mov xPos[3], 47

mov xPos[4], 46

mov yPos[0], 18

mov yPos[1], 18

mov yPos[2], 18

mov yPos[3], 18

mov yPos[4], 18

mov score,0

mov input, "+"

Call ClrScr

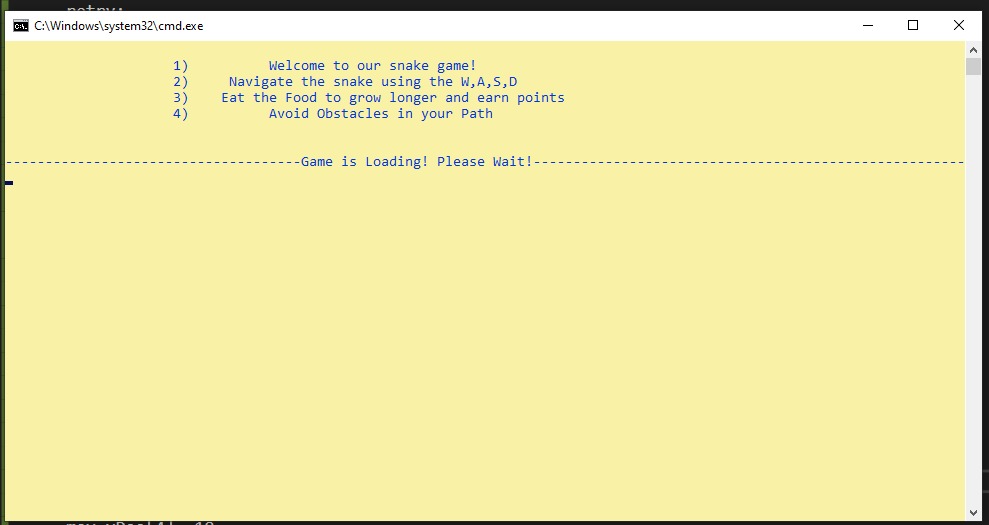
jmp main

RestartGame ENDP

END main

**Output:**

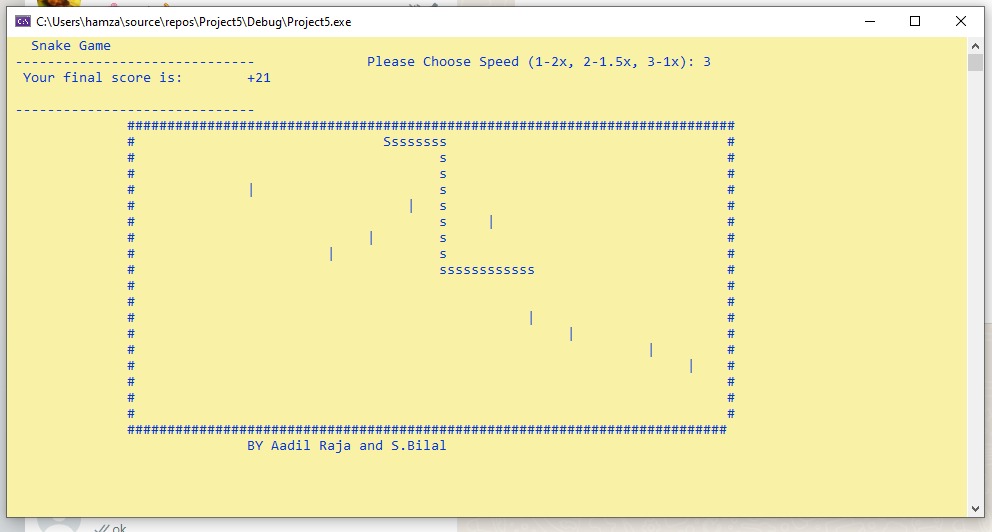
Welcome screen



Choosing speed



Playing game



Died

