**Table Tennis:**

The Game is quite close to Table tennis but it is in 2D

To score a point you have to hit the ball in a way that it touches the vertical wall side of the opponent.

Whoever scores six points first wins the match.

Two consecutive serves are given to each player.

The Game consists of two modes.

1- Player vs Bot mod.

You can select your desired difficulty level

2- Player vs Player

Two humans play against each other

**Rest of the Instructions are in the game.**

**Group Members:**

**1-Abdullah Saleem 18F-0232**

**2-Asad Mehmood 18F-0339**

**Code:**

include irvine32.inc

.data

startspace byte 20 dup(' '),0

horizontal byte 81 dup('-'),0

VERTICAL BYTE '|',0

MIDSPACE BYTE 39 DUP (' '),0

ball byte ("@"),0

ballcoordinateX Byte 23

ballcoordinateY byte 5

difficulty byte ?

rnum byte ?

racket1x byte 22

racket1y byte 5

racket2x byte 98

racket2y byte 11

serve byte 0

scorep1 byte 0

scorep2 byte 0

checkM byte 0

checkfu byte 0

checkfd byte 0

checkbu byte 0

checkbd byte 0

speedslower byte 0

speedloop byte 0

p1wins byte " Player 1 Wins!",0

p2wins byte " Player 2 Wins! ",0

cmpwins byte " BOT Wins ",0

msg byte " PING PONG ",0

msg0 byte " MENU ",0

msg1 byte " 1- Start Game ",0

msg2 byte " 2- Instructions ",0

msg3 byte " 3- Exit ",0

mod0 byte " select Mode ",0

mod1 byte "1- Player VS BOT ",0

mod2 byte "2- Player vs Player ",0

msg1sub byte " Select difficulty ",0

msg1sub1 byte "1- Easy ",0

msg1sub2 byte "2- Medium ",0

msg1sub3 byte "3- Hard ",0

modvar byte 0

ins1 byte "Instructions ",0

ins2 byte "1-Avoid the ball from hitting the vertical walls of the table by ",0

ins3 byte " using the rackets. ",0

ins4 byte "2-player1: ",0

ins5 byte " Use character 'w' and 's' for moving up and down respectively ",0

ins6 byte "3-player2: ",0

ins7 byte " Use character 'i' and 'k' for moving up and down respectively ",0

ins8 byte " Press 0 to return to menu ",0

ins9 byte "4- Long Press space to serve ",0

.code

main proc

call menu

exit

main endp

menubox proc uses eax ebx ecx

mov eax, blue +(lightgray\*16)

call settextcolor

mov dl, 20

mov dh, 2

call gotoxy

mov ecx, 70

l1:

mov al, '-'

call writechar

loop l1

mov dl, 20

mov dh, 3

call gotoxy

mov ecx,20

l2:

mov al,'|'

call writechar

inc dh

call gotoxy

loop l2

mov dl, 20

mov dh, 23

call gotoxy

mov ecx, 70

l3:

mov al, '-'

call writechar

loop l3

mov dl, 89

mov dh, 3

call gotoxy

mov ecx,20

l4:

mov al,'|'

call writechar

inc dh

call gotoxy

loop l4

mov eax, white +(black\*16)

call settextcolor

ret

menubox endp

menu proc uses eax edx ebx ecx

start:

call menubox

mov eax, white +(black\*16)

call settextcolor

mov dl, 50

mov dh, 4

call gotoxy

mov edx,offset msg

call writestring

mov dl, 52

mov dh, 7

call gotoxy

mov edx,offset msg0

call writestring

mov dl, 23

mov dh, 10

call gotoxy

mov edx,offset msg1

call writestring

mov dl, 23

mov dh, 12

call gotoxy

mov edx,offset msg2

call writestring

mov dl, 23

mov dh, 14

call gotoxy

mov edx,offset msg3

call writestring

call crlf

call crlf

label1:

mov eax, white +(black\*16)

call settextcolor

call readchar

cmp al, 49

jz game1

cmp al,50

jz instruction1

cmp al,51

jz exit1

jmp label1

game1:

call clrscr

call modepage

call clrscr

call play

call clrscr

jmp start

instruction1:

call clrscr

call instructions

rchar:

call readchar

cmp al,'0'

je seg1

JMP rchar

seg1:

call clrscr

jmp start

jmp exit1

exit1:

mov eax, white +(black\*16)

call settextcolor

ret

menu endp

modepage proc

start1:

call menubox

mov dl, 50

mov dh, 4

call gotoxy

mov edx,offset msg

call writestring

mov dl, 52

mov dh, 7

call gotoxy

mov edx,offset msg0

call writestring

mov dl, 22

mov dh, 10

call gotoxy

mov edx ,offset mod0

call writestring

mov dl, 22

mov dh, 12

call gotoxy

mov edx ,offset mod1

call writestring

mov dl, 22

mov dh, 13

call gotoxy

mov edx ,offset mod2

call writestring

call readchar

cmp al, '1'

je seg1

cmp al, '2'

je seg2

jmp start1

seg1:

call clrscr

mov modvar,0

call selectdifficulty

jmp exit1

seg2:

mov modvar,1

exit1:

ret

modepage endp

selectdifficulty proc

start1:

call menubox

mov dl, 50

mov dh, 4

call gotoxy

mov edx,offset msg

call writestring

mov dl, 52

mov dh, 7

call gotoxy

mov edx,offset msg0

call writestring

mov dl, 22

mov dh, 10

call gotoxy

mov edx ,offset msg1sub

call writestring

mov dl, 22

mov dh, 12

call gotoxy

mov edx ,offset msg1sub1

call writestring

mov dl, 22

mov dh, 13

call gotoxy

mov edx ,offset msg1sub2

call writestring

mov dl, 22

mov dh, 14

call gotoxy

mov edx ,offset msg1sub3

call writestring

call readchar

cmp al, '1'

je set1

cmp al ,'2'

je set2

cmp al, '3'

je set3

jmp start1

set1:

mov difficulty,1

jmp exit1

set2:

mov difficulty,2

jmp exit1

set3:

mov difficulty,3

exit1:

ret

selectdifficulty endp

instructions proc

call menubox

mov dl, 22

mov dh, 4

call gotoxy

mov edx ,offset ins1

call writestring

mov dl, 22

mov dh, 6

call gotoxy

mov edx ,offset ins2

call writestring

mov dl, 22

mov dh, 7

call gotoxy

mov edx ,offset ins3

call writestring

mov dl, 22

mov dh, 9

call gotoxy

mov edx ,offset ins4

call writestring

mov dl, 22

mov dh, 10

call gotoxy

mov edx ,offset ins5

call writestring

mov dl, 22

mov dh, 11

call gotoxy

mov edx ,offset ins6

call writestring

mov dl, 22

mov dh, 12

call gotoxy

mov edx ,offset ins7

call writestring

mov dl, 22

mov dh, 17

call gotoxy

mov edx ,offset ins8

call writestring

mov dl, 22

mov dh, 14

call gotoxy

mov edx ,offset ins9

call writestring

ret

instructions endp

play proc

mov scorep1,0

mov scorep2,0

mov serve,0

call table

call drawrackets

l1:

mov speedloop,0

call randomgen

call score

cmp serve,2

jae servep2

servep1:

mov eax, 10

call delay

call turn1

mov eax, 20

call delay

call readkey

cmp al,'0'

je exit1

cmp al,' '

jne servep1

mov al,racket1y

inc al

mov ballcoordinatex,23

mov ballcoordinatey,al

jmp seg1

servep2:

mov eax, 10

call delay

call turn2

mov eax, 20

call delay

call readkey

cmp al,'0'

je exit1

cmp al, ' '

jne servep2

mov al,racket2y

inc al

mov ballcoordinatex,95

mov ballcoordinatey,al

cmp serve,3

je set0

seg1:

call ballmovements

jmp scorecheck

set0:

call ballmovements

mov serve,0

scorecheck:

cmp scorep1,6

jz exit1

cmp scorep2,6

jz exit1

jmp l1

exit1:

call score

mov dl, 55

mov dh, 25

call gotoxy

cmp scorep1,6

je win1

cmp scorep2,6

je win2

jmp exitfinal

win1:

mov edx, offset p1wins

call writestring

jmp exitfinal

win2:

cmp modvar,0

je s1

mov edx, offset p2wins

jmp s2

s1:

mov edx,offset cmpwins

s2:

call writestring

exitfinal:

mov eax,2000

call delay

ret

play endp

Table proc uses edx ecx

mov edx, offset startspace

call writestring

mov eax, white +(blue\*16)

call settextcolor

mov edx, offset horizontal

call writestring

call crlf

mov ecx ,20

vert:

mov eax, white +(black\*16)

call settextcolor

mov edx, offset startspace

call writestring

mov eax, white +(blue\*16)

call settextcolor

mov edx,offset vertical

call writestring

mov edx, offset midspace

call writestring

mov edx,offset vertical

call writestring

mov edx, offset midspace

call writestring

mov edx,offset vertical

call writestring

call crlf

loop vert

mov eax, white +(black\*16)

call settextcolor

mov edx, offset startspace

call writestring

mov eax, white +(blue\*16)

call settextcolor

mov edx, offset horizontal

call writestring

mov eax, white +(black\*16)

call settextcolor

ret

Table endp

DrawRackets proc uses edx ecx

mov eax, red +(blue\*16)

call settextcolor

mov dl,racket1x

mov dh, racket1y

mov ecx,3

l1:

call gotoxy

mov al ,']'

call writechar

INC dh

loop L1

mov dl, racket2x

mov dh, racket2y

mov ecx,3

l2:

call gotoxy

mov al ,'['

call writechar

INC dh

loop L2

mov eax, white +(black\*16)

call settextcolor

ret

DrawRackets endp

turn1 proc uses eax ecx

mov eax, white +(blue\*16)

call settextcolor

call readkey

cmp al , 77h

JE moveup

cmp al , 73h

JE movedown

cmp al , 57h

JE moveup

cmp al ,53h

JE movedown

jmp exit1

moveup:

cmp racket1y,2

jge segu

jmp end1

segu:

mov dl,racket1x

mov dh, racket1y

mov ecx,3

l1:

call gotoxy

mov al ,' '

call writechar

INC dh

loop L1

sub racket1y,1

call drawrackets

end1:

jmp exit1

movedown:

cmp racket1y,17

jle segd

jmp end2

segd:

mov dl,racket1x

mov dh, racket1y

mov ecx,3

l2:

call gotoxy

mov al ,' '

call writechar

INC dh

loop L2

add racket1y,1

call drawrackets

end2:

exit1:

mov eax, white +(black\*16)

call settextcolor

ret

turn1 endp

turn2 proc uses eax ecx

mov eax, white +(blue\*16)

call settextcolor

call readkey

cmp al , 49h

JE moveup

cmp al ,4Bh

JE movedown

cmp al , 69h

JE moveup

cmp al ,6Bh

JE movedown

jmp exit1

moveup:

cmp racket2y,2

jge segu

jmp end1

segu:

mov dl,racket2x

mov dh, racket2y

mov ecx,3

l1:

call gotoxy

mov al ,' '

call writechar

INC dh

loop L1

sub racket2y,1

call drawrackets

end1:

jmp exit1

movedown:

cmp racket2y,17

jle segd

jmp end2

segd:

mov dl,racket2x

mov dh, racket2y

mov ecx,3

l2:

call gotoxy

mov al ,' '

call writechar

INC dh

loop L2

add racket2y,1

call drawrackets

end2:

exit1:

mov eax, white +(black\*16)

call settextcolor

ret

turn2 endp

randomGEN proc

call randomize

mov eax,2

call randomrange

mov rnum,al

cmp difficulty,1

je exit1

cmp difficulty,2

je add2

add rnum,3

jmp exit1

add2:

add rnum,2

exit1:

ret

randomGen endp

comp proc uses eax ecx

mov eax, white +(blue\*16)

call settextcolor

cmp ballcoordinatex,50

jb exit1

mov al,speedloop

cmp al,rnum

jle speed1

speed2:

cmp speedslower,2

jl exit1

jmp segment1

speed1:

cmp speedslower,1

jl exit1

segment1:

mov speedslower, 0

mov al, racket2y

inc al

cmp al , ballcoordinatey

ja moveup

jb movedown

je exit1

moveup:

cmp racket2y,2

jge segu

jmp end1

segu:

mov dl,racket2x

mov dh, racket2y

mov ecx,3

l1:

call gotoxy

mov al ,' '

call writechar

INC dh

loop L1

sub racket2y,1

call drawrackets

end1:

jmp exit1

movedown:

cmp racket2y,17

jle segd

jmp end2

segd:

mov dl,racket2x

mov dh, racket2y

mov ecx,3

l2:

call gotoxy

mov al ,' '

call writechar

INC dh

loop L2

add racket2y,1

call drawrackets

end2:

exit1:

inc speedslower

mov eax, white +(black\*16)

call settextcolor

ret

comp endp

score proc uses edx ecx

mov dl, 55

mov dh, 22

call gotoxy

movzx eax,scorep1

call writedec

mov ecx ,9

l1:

mov al,' '

call writechar

loop l1

movzx eax,scorep2

call writedec

ret

score endp

ballmovements proc uses eax ebx ecx edx

inc serve

cmp serve,2

jle movstraightf

ja movstraightb

movstraightF:

l1:

mov dl,ballcoordinateX

mov dh,ballcoordinateY

call gotoxy

mov eax, yellow +(blue\*16)

call settextcolor

mov edx, offset ball

call writestring

mov eax, 20

call delay

cmp modvar,0

je setcomp1

call turn2

jmp over1

setcomp1:

call comp

over1:

mov dl,ballcoordinateX

mov dh,ballcoordinateY

call gotoxy

mov eax, white +(blue\*16)

call settextcolor

mov al , ' '

call writechar

cmp ballcoordinatex,60

jne pass1

mov dl,ballcoordinatex

mov dh, ballcoordinatey

call gotoxy

mov al, '|'

call writechar

pass1:

add ballcoordinatex ,1

mov al, ballcoordinatex

mov bl, racket2x

dec bl

cmp al,bl

JL L1

;checking racket2

mov al,ballcoordinatey

mov bl,racket2y

cmp al, bl

je movbackup

inc bl

cmp al, bl

je movstraightb

inc bl

cmp al,bl

je movbackdown

inc scorep1

jmp exit1

movstraightB:

inc speedloop

l2:

mov dl,ballcoordinateX

mov dh,ballcoordinateY

call gotoxy

mov eax, yellow +(blue\*16)

call settextcolor

mov edx, offset ball

call writestring

mov eax, 20

call delay

call turn1

mov dl,ballcoordinateX

mov dh,ballcoordinateY

call gotoxy

mov eax, white +(blue\*16)

call settextcolor

mov al , ' '

call writechar

cmp ballcoordinatex,60

jne pass2

mov dl,ballcoordinatex

mov dh, ballcoordinatey

call gotoxy

mov al, '|'

call writechar

pass2:

sub ballcoordinatex ,1

mov al, ballcoordinatex

mov bl, racket1x

inc bl

cmp al,bl

Ja L2

;checking racket1

mov al,ballcoordinatey

mov bl,racket1y

cmp al,bl

je movforup

inc bl

cmp al,bl

je movstraightF

inc bl

cmp al,bl

je movfordown

inc scorep2

jmp exit1

movforup:

mov checkfu,0

l3:

mov dl,ballcoordinateX

mov dh,ballcoordinateY

call gotoxy

mov eax, yellow +(blue\*16)

call settextcolor

mov edx, offset ball

call writestring

mov eax, 50

call delay

cmp modvar,0

je setcomp2

call turn2

jmp over2

setcomp2:

call comp

over2:

mov dl,ballcoordinateX

mov dh,ballcoordinateY

call gotoxy

mov eax, white +(blue\*16)

call settextcolor

mov al , ' '

call writechar

cmp ballcoordinatex,60

jne pass3

mov dl,ballcoordinatex

mov dh, ballcoordinatey

call gotoxy

mov al, '|'

call writechar

pass3:

add ballcoordinatex ,1

cmp checkfu ,0

je upmov3

jne downmov3

upmov3:

cmp ballcoordinatey,2

jae dec3

jmp incval3

dec3:

sub ballcoordinatey,1

jmp exit3

incval3:

inc checkfu

jmp exit3

downmov3:

cmp ballcoordinatey,19

jle inc3

jmp decval3

inc3:

add ballcoordinatey,1

jmp exit3

decval3:

dec checkfu

exit3:

mov al, ballcoordinatex

mov bl, racket2x

dec bl

cmp al,bl

JL L3

;checking racket2

mov al,ballcoordinatey

mov bl,racket2y

cmp al, bl

je movbackup

inc bl

cmp al, bl

je movstraightb

inc bl

cmp al,bl

je movbackdown

inc scorep1

mov checkfu,0

jmp exit1

movbackup:

mov checkbu,0

inc speedloop

l4:

mov dl,ballcoordinateX

mov dh,ballcoordinateY

call gotoxy

mov eax, yellow +(blue\*16)

call settextcolor

mov edx, offset ball

call writestring

mov eax, 50

call delay

call turn1

mov dl,ballcoordinateX

mov dh,ballcoordinateY

call gotoxy

mov eax, white +(blue\*16)

call settextcolor

mov al , ' '

call writechar

cmp ballcoordinatex,60

jne pass4

mov dl,ballcoordinatex

mov dh, ballcoordinatey

call gotoxy

mov al, '|'

call writechar

pass4:

sub ballcoordinatex ,1

cmp checkbu ,0

je upmov4

jne downmov4

upmov4:

cmp ballcoordinatey,2

jae dec4

jmp incval4

dec4:

sub ballcoordinatey,1

jmp exit4

incval4:

inc checkbu

jmp exit4

downmov4:

cmp ballcoordinatey,19

jle inc4

jmp decval4

inc4:

add ballcoordinatey,1

jmp exit4

decval4:

dec checkbu

exit4:

mov al, ballcoordinatex

mov bl, racket1x

inc bl

cmp al,bl

Ja L4

;checking racket1

mov al,ballcoordinatey

mov bl,racket1y

cmp al,bl

je movforup

inc bl

cmp al,bl

je movstraightF

inc bl

cmp al,bl

je movfordown

inc scorep2

mov checkbu,0

jmp exit1

movfordown:

mov checkfd,0

l5:

mov dl,ballcoordinateX

mov dh,ballcoordinateY

call gotoxy

mov eax, yellow +(blue\*16)

call settextcolor

mov edx, offset ball

call writestring

mov eax, 50

call delay

cmp modvar,0

je setcomp3

call turn2

jmp over3

setcomp3:

call comp

over3:

mov dl,ballcoordinateX

mov dh,ballcoordinateY

call gotoxy

mov eax, white +(blue\*16)

call settextcolor

mov al , ' '

call writechar

cmp ballcoordinatex,60

jne pass5

mov dl,ballcoordinatex

mov dh, ballcoordinatey

call gotoxy

mov al, '|'

call writechar

pass5:

add ballcoordinatex ,1

cmp checkfd ,1

je upmov5

jne downmov5

upmov5:

cmp ballcoordinatey,2

jae dec5

jmp decval5

dec5:

sub ballcoordinatey,1

jmp exit5

decval5:

dec checkfd

jmp exit5

downmov5:

cmp ballcoordinatey,19

jle inc5

jmp incval5

inc5:

add ballcoordinatey,1

jmp exit5

incval5:

inc checkfd

exit5:

mov al, ballcoordinatex

mov bl, racket2x

dec bl

cmp al,bl

JL L5

;checking racket2

mov al,ballcoordinatey

mov bl,racket2y

cmp al, bl

je movbackup

inc bl

cmp al, bl

je movstraightb

inc bl

cmp al,bl

je movbackdown

inc scorep1

mov checkfd,0

jmp exit1

movbackdown:

mov checkbd,0

inc speedloop

l6:

mov dl,ballcoordinateX

mov dh,ballcoordinateY

call gotoxy

mov eax, yellow + (blue\*16)

call settextcolor

mov edx, offset ball

call writestring

mov eax, 50

call delay

call turn1

mov dl,ballcoordinateX

mov dh,ballcoordinateY

call gotoxy

mov eax, white +(blue\*16)

call settextcolor

mov al , ' '

call writechar

cmp ballcoordinatex,60

jne pass6

mov dl,ballcoordinatex

mov dh, ballcoordinatey

call gotoxy

mov al, '|'

call writechar

pass6:

sub ballcoordinatex ,1

cmp checkbd ,1

je upmov6

jne downmov6

upmov6:

cmp ballcoordinatey,2

jae dec6

jmp decval6

dec6:

sub ballcoordinatey,1

jmp exit6

decval6:

dec checkbd

jmp exit6

downmov6:

cmp ballcoordinatey,19

jle inc6

jmp incval6

inc6:

add ballcoordinatey,1

jmp exit6

incval6:

inc checkbd

exit6:

mov al, ballcoordinatex

mov bl, racket1x

inc bl

cmp al,bl

Ja L6

;checking racket2

mov al,ballcoordinatey

mov bl,racket1y

cmp al,bl

je movforup

inc bl

cmp al,bl

je movstraightF

inc bl

cmp al,bl

je movfordown

inc scorep2

mov checkbd,0

exit1:

mov eax, white +(black\*16)

call settextcolor

ret

ballmovements endp

end main

**Screenshots:**











