

**Coal Project Description**

Project Name:

Draughts

Team members:

Fraz Zaheer 19F-0818

Rafaqat Ali 19F-0808

**Description:**

It is a strategy game for two players in which players can move their pieces diagonaly to capture the piece of other player. This game is for player to player not for computer to player.

Game view looks like this



**Features:**

1. Only two players can play this game
2. Each player has 12 pieces
3. Palyer can move piece only diagonally
4. Player can kill the piece of opponent
5. The main purpose of this game to capture all the pieces of his opponent.

**Code:**

include Irvine32.inc

.data

cB DB ' ', ' ', ' ', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-'

DB '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|'

DB ' ', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'X', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'X', ' ', ' '

DB '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'X', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'X', ' ', ' ', '|'

DB ' ', ' ', ' ', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-'

DB '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|'

DB ' ', ' ', ' ', '|', ' ', ' ', 'X', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'X', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' '

DB '|', ' ', ' ', 'X', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'X', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|'

DB ' ', ' ', ' ', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-'

DB '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|'

DB ' ', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'X', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'X', ' ', ' '

DB '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'X', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'X', ' ', ' ', '|'

DB ' ', ' ', ' ', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-'

DB '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|'

DB ' ', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'O', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|'

DB ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|'

DB ' ', ' ', ' ', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-'

DB '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|'

DB ' ', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|'

DB ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|'

DB ' ', ' ', ' ', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-'

DB '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|'

DB ' ', ' ', ' ', '|', ' ', ' ', 'O', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'O', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|'

DB ' ', ' ', 'O', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'O', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|'

DB ' ', ' ', ' ', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-'

DB '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|'

DB ' ', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'O', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'O', ' ', ' ', '|'

DB ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'O', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'O', ' ', ' ', '|'

DB ' ', ' ', ' ', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-'

DB '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|'

DB ' ', ' ', ' ', '|', ' ', ' ', 'O', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'O', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|'

DB ' ', ' ', 'O', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|', ' ', ' ', 'O', ' ', ' ', '|', ' ', ' ', ' ', ' ', ' ', '|'

DB ' ', ' ', ' ', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-'

DB '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|', '-', '-', '-', '-', '-', '|'

flag DB 0

str1 DB "Player 1 Piece: ", 0

str2 DB "Player 2 Piece: ", 0

str3 DB "Player1 Move Piece to :", 0

str4 DB "Player2 Move Piece to :", 0

userInput DB ?

.code

pieceAccess1 proc

mov ECX,EAX

LLL1:

add ESI,6

loop LLL1

MOV ECX,EBX

LLL2 :

add ESI, 52

loop LLL2

ret

pieceAccess1 endp

pieceAccess proc USES EBX EAX

mov ESI, 0

CMP EAX,65

JL JJ

JNL JJN

JJ:

CMP EAX,56

JG JJ1

JNG JJN1

JJ1:

SUB EAX,56

mov EBX,15

call pieceAccess1

JMP drawJump1

JJN1:

CMP EAX, 48

JG JJ2

JNG JJN2

JJ2 :

SUB EAX, 48

mov EBX, 13

call pieceAccess1

JMP drawJump1

JJN2:

CMP EAX, 40

JG JJ3

JNG JJN3

JJ3 :

SUB EAX, 40

mov EBX, 11

call pieceAccess1

JMP drawJump1

JJN3 :

CMP EAX, 32

JG JJ4

JNG JJN4

JJ4 :

SUB EAX, 32

mov EBX, 9

call pieceAccess1

JMP drawJump1

JJN4 :

CMP EAX, 24

JG JJ5

JNG JJN5

JJ5 :

SUB EAX, 24

mov EBX, 7

call pieceAccess1

JMP drawJump1

JJN5 :

CMP EAX, 16

JG JJ6

JNG JJN6

JJ6 :

SUB EAX, 16

mov EBX, 5

call pieceAccess1

JMP drawJump1

JJN6 :

CMP EAX, 8

JG JJ7

JNG JJN7

JJ7 :

SUB EAX, 8

mov EBX, 3

call pieceAccess1

JMP drawJump1

JJN7 :

CMP EAX, 0

JG JJ8

JNG JJN8

JJ8 :

SUB EAX, 0

mov EBX, 1

call pieceAccess1

JMP drawJump1

JJN:

call pieceAccess

JJN8 :

drawJump1:

ret

pieceAccess endp

checkerBoard proc

mov EBX, offset cB

mov ESI, 0

call crlf

mov ECX, 17

L1:

PUSH ECX

mov ECX, 52

L2 :

mov AL, [EBX+ESI]

call writechar

inc ESI

loop L2

POP ECX

call crlf

loop L1

ret

checkerBoard endp

Player1 proc

call crlf

LEA EDX, str1

call writestring

mov EBX, offset cB

call readdec

call pieceAccess

mov DL, 'X'

CMP DL, [EBX + ESI]

JZ checkinputJump1

JNZ checkinputJump2

checkinputJump1 :

ADD EAX,9

call pieceAccess

call crlf

LEA EDX, str3

call writestring

PUSH EAX

call readdec

mov EDX,EAX

POP EAX

CMP EDX, EAX

JZ checkinputJump3

JNZ checkinputJump4

; Right move

checkinputJump3:

mov DL, ' '

CMP[EBX + ESI], DL

JZ moveJump1

JNZ moveJump2

;single move

moveJump1:

mov DL, 'X'

mov[EBX + ESI], DL

SUB EAX, 9

call pieceAccess

mov DL, ' '

mov[EBX + ESI], DL

JMP exeptionJump1

;Double move

moveJump2:

mov DL, 'O'

CMP [EBX+ESI],DL

JZ moveJump3

JNZ moveJump4

moveJump3:

PUSH EAX

add EAX, 9

mov DL, ' '

call pieceAccess

CMP [EBX + ESI], DL

JZ moveJump4

JNZ moveJump5

moveJump4:

mov DL, 'X'

CALL pieceAccess

mov[EBX + ESI], DL

POP EAX

mov DL, ' '

CALL pieceAccess

mov[EBX + ESI], DL

SUB EAX,9

CALL pieceAccess

mov[EBX + ESI], DL

JMP exeptionJump1

;left move

checkinputJump4:

sub EAX, 2

CMP EDX, EAX

JZ moveJump6

JNZ exeptionJump1

moveJump6:

mov DL, ' '

CALL pieceAccess

CMP [EBX + ESI], DL

JZ moveJump7

JNZ moveJump8

;single move

moveJump7:

mov DL, 'X'

mov [EBX + ESI], DL

SUB EAX, 7

mov DL, ' '

CALL pieceAccess

mov [EBX + ESI], DL

JMP exeptionJump1

;double move

moveJump8:

mov DL, 'O'

CMP [EBX+ESI],DL

JZ moveJump9

JNZ moveJump5

moveJump9:

PUSH EAX

add EAX, 7

mov DL, ' '

call pieceAccess

CMP [EBX+ESI],DL

JZ moveJump10

JNZ moveJump5

moveJump10:

mov DL, 'X'

call pieceAccess

mov[EBX + ESI], DL

POP EAX

mov DL, ' '

call pieceAccess

mov[EBX + ESI], DL

SUB EAX,7

call pieceAccess

mov[EBX + ESI], DL

JMP exeptionJump1

moveJump5:

POP EAX

exeptionJump1 :

call checkerBoard

checkinputJump2:

call Player2

ret

Player1 endp

Player2 proc

call crlf

LEA EDX, str2

call writestring

mov EBX, offset cB

call readdec

mov ESI, EAX

mov AL, 'O'

CMP AL, [EBX + ESI]

JZ checkinputJump5

JNZ exeptionJump2

checkinputJump5:

SUB ESI, 7

call crlf

LEA EDX, str4

call writestring

call readdec

CMP ESI, EAX

JZ moveJump20

JNZ moveJump21

; Right Side

moveJump20 :

mov AL, ' '

CMP [EBX + ESI], AL

JZ moveJump22

JNZ moveJump23

moveJump22 :

mov AL, 'O'

mov[EBX + ESI], AL

ADD ESI, 7

mov AL, ' '

mov[EBX + ESI], AL

JMP exeptionJump2

;double move

moveJump23 :

mov AL, 'X'

CMP[EBX + ESI], AL

JZ moveJump24

JMP exeptionJump2

moveJump24:

PUSH ESI

SUB ESI, 7

mov AL, ' '

CMP [EBX + ESI], AL

JZ moveJump25

JNZ moveJump26

moveJump25 :

mov AL, 'O'

mov[EBX + ESI], AL

POP ESI

; for removing middle opposite piece

mov AL, ' '

mov[EBX + ESI], AL

ADD ESI, 7

mov[EBX + ESI], AL

JMP exeptionJump2

;left Side

moveJump21:

mov EAX, ' '

SUB ESI,2

CMP [EBX + ESI], AL

JZ moveJump27

JNZ moveJump28

moveJump27 :

mov AL, 'O'

mov[EBX + ESI], AL

ADD ESI, 9

mov AL, ' '

mov[EBX + ESI], AL

JMP exeptionJump2

moveJump28 :

mov AL, 'X'

CMP[EBX + ESI], AL

JZ moveJump29

JMP exeptionJump2

moveJump29:

PUSH ESI

SUB ESI, 9

mov AL, ' '

CMP[EBX + ESI], AL

JZ moveJump30

JNZ moveJump26

moveJump30 :

mov AL, 'O'

mov[EBX + ESI], AL

POP ESI

; for removing middle opposite piece

mov AL, ' '

mov[EBX + ESI], AL

ADD ESI, 9

mov[EBX + ESI], AL

JNZ exeptionJump2

; As I PUSH once either checkJump4 or checkJump3

moveJump26:

POP ESI

exeptionJump2 :

call checkerBoard

call Player1

ret

Player2 endp

main proc

call checkerBoard

call Player1

call waitmsg

exit

main endp

end main

**Screen Shots:**





