Hacettepe University
Computer Science and Engineering Department

Lab Report

Name and Surname	Furkan ARSLAN
Identity Number	21026686
Course	Bil-137
Experiment	EXPERIMENT 2
Subject	Functions and Arrays
Date Due	26.11.2010- 14.12.2010
Advisors	Sevil ŞEN, Erkut ERDEM, Safa SOFUOĞLU
e-mail	furkan14_bjk @hotmail.com
Main Program	C programming language

2. Software Using Documentation

2.1. Software Usage

The program starts and wait an input from the user. Entered variable must be conform chess table which formed 8 line and 8 colomn. Line variables must be between 1-8, column variables must be between a-h and stone kinds determined as “k, q, b, r, n, p ”.

A)If user enter place: The program wait the user to enter a location and kind of stone. If entered location and kind of stone, conform line and column requirements and if the location is empty, program will appoint the stone to that location.

B)If user enter move: The program wait the user to enter first location of stone and new location. Then stone is appointed to new location

C)If user enter showmoves: The program wait the user to enter a location for stone which situated that location, shows where it can move. User enter column and line variable.Then program list where the stone can move.

D)If user enter print: The program show chess stone on chess table at time.

E)If user enter clear: The program clear chess Stone on chess table.Then user will has empty chess table.

F)If user enter exit: program will be close.

2.2. Provided Possibilities

There are no extra possibilities in my program. The all operation that you read in software usage will be done.

2.3 Error Messages

My program will show error message which is “FAILED” at every error. Some reason to error:

1)If user enters line numbers that are higher than 8 or less than 0.

2)If user enters column letters that are except between a-h.

3)In place ıf user enter location is not empty.

4)In showmoves and In move ıf user enter empty location

5)In move ıf user enter wrong location where stone can not go.

6)If user enters wrong kind of piece.

3.Software Design Notes

3.1. Desctiption of the program ..

3.1.1. Problem

In this experiment, I must develop a chess application. This application comform rules of chees.

My program include place and move stone on the chess board, and show the board’s layout, and clean the board.

If user want to place piece on the board. The program can firstly check at the location and then will be can assign piece.

If user want to move a piece to another place. The program should firstly check the target location whether it is empty or there is a rival piece, then it should assign piece to the location and should clean the former location. If there is a friendly piece, the program should determinent and warn the user.

There is a different to real chess game. White pawns will only be able to move upwards and black pawns will only be able to move downwards. And pawn can not move two-square.

3.1.2.Solution

In this experiment, I developed chess application by using C programming language.

I solve the problem with one method. The method is that the main function would determine what the program will do, by using user input. The main function call other functions to do user’s request.I use string structure for calling other functions. The other function were determined as 5 parts.

1)place: Main function call the place function if user enter place. the purpose of place is that assigning a piece to a target location.I make the program to check variable then assign piece.

2)move: Main function call the place function if user enter move.the purpose of move is that move a piece to another location.For this purpose i call the function which were determined according to kind of stone. Then i make the program to determine where piece go. After that process I make the program to check variable then it assign piece to target location and clean former location.

3)showmoves: Main function call the place function if user enter showmoves. The purpose of showmoves is that show where a piece which enter by user can go. For this problem ı use same way to move. But only different is that showmoves can not assign but it can show screen where piece go.

4)print: Print function only show chess table at screen. For this process ı use loops.

5)clear: Clear function have one process. The function cleans chess table.

User also can enter ‘exit’ in main function. If user enter this, the program were adjust to close program.

3.3. Main Data Structures

The main data in my system is arrays.

White pieces=white[6];

Black pieces=black[6];

All pieces= stones[12];

between a-h letter=letter[8];

chess[55];

chessboard [8] [8];

I use these arrays almost every function and i save the almost every data in these arrays. I use chess[] array to save where a piece can go.

3.4. Algorithm

#include<stdio.h>

#include<string.h>

1.entered place

1.1 determine variable

1.2 Check variable

1.3 assign a piece to a location

1.4 show 'OK' or 'FAILED'

2.entered move

2.1 determine variable

2.2 check variable and determine kind of piece

2.3 Calling function according to kind of piece

2.3.1 determine where the piece can go

2.3.2 sort determined variable

2.3.3 check the entered location in terms of a possible place that the piece can go

2.4 move to the target location

2.5 clean former location

2.6 show 'OK' or 'FAILED'

3.entered showmoves

3.1 determine variable

3.2 check variable

3.3. Calling function according to kind of piece

3.3.1 Determine where the piece can go

3.3.2 Sort determined variable

3.4 show where a piece can go

4 entered print

4.1 show chess table

5 entered clear

5.1 clean chess table

6 entered exit

6.1 close the program

3.5. Special Design Properties

I have no new approaches to the problem.

3.6. Execution Flow Between Subprograms

int main ()

*it calls place() or move() or showmoves() or print() or clear()

Void place(chessboard[][])

*this function assign to the piece to the location

*it is called from the main()

* it is a void function (return nothing

Void move(chessboard[][])

*this function give new place to the piece in rules of chess.

* it is called from the main()

* it is a void function (return nothing)

*it is call king() or queen() or rook() or knight() or bishop or pawn()

Void showmoves(chessboard[][])

*this function show where the piece can go

* it is a void function (return nothing)

* it is called from the main()

* it is call king() or queen() or rook() or knight() or bishop or pawn()

Void print(chessboard[][])

*this function show chess table on screen

* it is a void function (return nothing)

* it is called from the main()

Void clear(chessboard[][])

*this function clean chess table

* it is a void function (return nothing)

* it is called from the main()

Void king()

*this function determine movement of king

* it is a void function (return nothing)

* it is called from the showmoves() or move()

*it calls sort() and test_stone()

Void queen()

*this function determine movement of queen

* it is a void function (return nothing)

* it is called from the showmoves() or move()

*it calls bishop() and rook()

Int rook()

*this function determine movement of rook

* it is a int function (return in that place of chess array)

* it is called from the showmoves() or move() or queen()

*it calls sort() and test_stone()

Void bishop()

*this function determine movement of bishop

* it is a void function (return nothing)

* it is called from the showmoves() or move() or queen()

*it calls sort() and test_stone()

Void knight()

*this function determine movement of knight

* it is a void function (return nothing)

* it is called from the showmoves() or move()

*it calls sort() and test_stone()

Void pawn()

*this function determine movement of pawn

* it is a void function (return nothing)

* it is called from the showmoves() or move()

*it calls sort() and test_stone() and pawn_control()

Void pawn_control()

*this function determine movement of pawn

* it is a void function (return nothing)

*it is called from pawn

*it calls nothing

İnt test_stone()

*this function control whether exist rival piece or not the location

*it is return ‘n’ variable (if exist rival piece return (n+2) else return (n))

*it is called from king()or rook() or knight() or bishop or pawn()

*it calls nothing

Void sort()

*this function sort determined variable of movement of the piece

*it is called from king() or rook() or knight() or bishop or pawn()

* it is a void function (return nothing)

*it calls nothing

4. SOFTWARE TESTING NOTES :

4.1. Bugs and Software Reliability

My program Works correctly. I did not find any mistake.

4.2. Software Extendibility and Upgradability

In my program you can add any features. Because i solve the problem in my opinion and this program work correctly now but if you can add any extra features this program allow this. And you do not have to write all program. It is enough to write only what you want to and add it to my program.

There are 500 – 600 lines in my program.

REFERENCES

Program solving and program design in C   Fifth Edition

Programlamaya giriş C dersi (Gül Tokdemir,Çiğdem Turhan,C.Fügen Selbes / 2010)