**Game Coding**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

HEADER FILES USED

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<graphics.h>

#include<fstream.h>

#include<process.h>

#include<stdlib.h>

#include<string.h>

#include<stdio.h>

#include<conio.h>

#include<dos.h>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTIONS USED

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void dice(); //FUNCTION FOR ROLLING THE DICE

void draw(); //FUNCTION TO DRAW THE HEXAGON

void rules(); //FUNCTION TO DISPLAY RULES OF THE GAME

void loadin(); //FUNCTION TO DISPALY LOADING

void credits(); //FUNCTION TO DISPLAY CREDITS

void file\_del(); //FUNCTION TO DELETE FILE

void file\_list(); //FUNCTION TO SHOW SAVE FILES

void file\_save(); //FUNCTION TO SAVE THE GAME

void clr\_match(); //FUNCTION TO CHECK FOR SAME CLR

void game\_choose(); //FUNCTION TO CHOOSE THE GAME

void modi(int); //FUNCTION FOR POSTION MODIFICATION

void clr\_chs(int); //FUNCTION TO CHOOSE COLOUR FOR COINS

void hexagon(int,int); //FUNCTION TO CHANGE THE COORDINATES

void position(int\*,int\*); //FUNCTION TO PRINT THE POSITIONS

void graphics\_screen(); //FUNCTION TO INITALISE GRA[HICS

int moves(); //FUNCTION TO CHECK FOR CORRECT MOVES

int lstrw(); //FUNCTION MOVE TO LAST ROW

int capchk(); //FUNCTION TO CHECK FOR CAPTURE

int game\_cho(); //FUNCTION TO CHOOSE THE GAME IN START

int file\_read(); //FUNCTION TO READ THE SAVE FILE

int winchk(int); //FUNCTION TO CHECK FOR WIN

int poschag(int); //FUNCTION FOR POISTION CHANGING

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

GLOBAL VARIABLES

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

const int plyr[2]={1,2};

char nm[2][30],fl\_nm[30],grp[50]="c:\\turboc3\\bgi",ch;

int poly[14],rw1,cl1,rw2,cl2,ply=1,flag,cho,new\_gm=0;

int clr1=4,clr2=2,bkclr=8,count=0;

fstream file;

int p1rw[7]={0,0,4,4,8,9,10},p1cl[7]={0,4,0,9,7,0,3};

int p2rw[7]={0,1,2,6,6,10,10},p2cl[7]={2,6,0,0,9,1,5};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Representation of the inital board positions as a variable

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int pos[11][11]={

/\* 0 \*/ { 11,0,2,0,1,0, -1,-1,-1,-1,-1},

/\* 1 \*/ { 0,0,0,0,0,0,2, -1,-1,-1,-1},

/\* 2 \*/ { 2,0,0,0,0,0,0,0, -1,-1,-1},

/\* 3 \*/ { 0,0,0,0,0,0,0,0,0, -1,-1},

/\* 4 \*/ { 1,0,0,0,0,0,0,0,0,1, -1},

/\* 5 \*/ { 0,0,0,0,0,0,0,0,0,0,0 },

/\* 6 \*/ { 2,0,0,0,0,0,0,0,0,2, -1},

/\* 7 \*/ { 0,0,0,0,0,0,0,0,0, -1,-1},

/\* 8 \*/ { 0,0,0,0,0,0,0,1, -1,-1,-1},

/\* 9 \*/ { 1,0,0,0,0,0,0, -1,-1,-1,-1},

/\* 10\*/ { 0,2,0,1,0,22, -1,-1,-1,-1,-1}};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CLASS DECLRATION

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class save

{

private:

char nm[2][30];

int count,ply;

int clr1,clr2,bkclr;

int p1rw[7],p1cl[7];

int p2rw[7],p2cl[7];

int pos[11][11];

public:

save(); //constuctor

void copy\_to(); //to copy from global to class

void copy\_4m(); //to copy from class to global

} s;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CONSTRUCTOR OF CLASS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

save::save()

{

int i,j,k=5;

strcpy(nm[0],"Player 1");

strcpy(nm[1],"Player 2");

count=0;

ply=1;

clr1=4;

clr2=2;

bkclr=8;

for(i=0;i<7;i++)

p1rw[i]=p1cl[i]=p2rw[i]=p1cl[i]=0;

p1rw[2]=p1rw[3]=p1cl[1]=4;

p1rw[6]=p2rw[5]=p2rw[6]=10;

p1rw[5]=p1cl[3]=p2cl[4]=9;

p2rw[3]=p2rw[4]=p2cl[1]=6;

p2rw[2]=p2cl[0]=2;

p2rw[1]=p2cl[5]=1;

p1rw[4]=8;

p1cl[4]=7;

p1cl[6]=3;

p2cl[6]=5;

for(i=0;i<11;i++)

for(j=0;j<11;j++)

pos[i][j]=0;

pos[0][0]=11;

pos[10][5]=22;

pos[0][4]=pos[4][0]=pos[4][9]=pos[8][7]=pos[9][0]=pos[10][3]=1;

pos[0][2]=pos[1][6]=pos[2][0]=pos[6][0]=pos[6][9]=pos[10][1]=2;

for(i=0;i<5;i++)

for(j=6+i;j<11;j++)

pos[i][j]=-1;

for(j=10;j>5;j--)

{

for(i=10;i>k;i--)

pos[i][j]=-1;

cout<<endl;

k++;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CLASS FUNTION COPY\_TO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void save::copy\_to()

{

int i,j;

strcpy(nm[0],::nm[0]);

strcpy(nm[1],::nm[1]);

count=::count;

ply=::ply;

clr1=::clr1;

clr2=::clr2;

bkclr=::bkclr;

for(i=0;i<7;i++)

{

p1rw[i]=::p1rw[i];

p1cl[i]=::p1cl[i];

p2rw[i]=::p2rw[i];

p2cl[i]=::p2cl[i];

}

for(i=0;i<11;i++)

for(j=0;j<11;j++)

pos[i][j]=::pos[i][j];

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CLASS FUNTION COPY\_4M

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void save::copy\_4m()

{

int i,j;

strcpy(::nm[0],nm[0]);

strcpy(::nm[1],nm[1]);

::count=count;

::ply=ply;

::clr1=clr1;

::clr2=clr2;

::bkclr=bkclr;

for(i=0;i<7;i++)

{

::p1rw[i]=p1rw[i];

::p1cl[i]=p1cl[i];

::p2rw[i]=p2rw[i];

::p2cl[i]=p2cl[i];

}

for(i=0;i<11;i++)

for(j=0;j<11;j++)

::pos[i][j]=pos[i][j];

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO INITALISE GRAPHICS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void graphics\_screen()

{

int gd=DETECT,gm,ec;

initgraph(&gd,&gm,grp);

ec=graphresult();

if(ec!=grOk) // an error occurred

{

cout<<"\"Graphics error:\" \n"<<grapherrormsg(ec)<<endl;

cout<<"Enter ur Graphics dir:-\n";

gets(grp);

graphics\_screen();

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MAIN FUNCTION

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main()

{

clrscr();

int run=1;

int chk,mov,cap,i,poschg,gm\_ch;

char sav;

count=::count;

if(run)

loadin(); //calling of loadin

gm\_ch=game\_cho(); //calling of game choose

if(gm\_ch==1)

goto cont;

else if(gm\_ch==-1)

{

if(run)

credits(); //callin of credits

getch();

exit(0);

}

if(run)

{

rules(); //calling of rules function

clrscr();

dice(); //calling of dice function

clr\_chs(0); //calling of colour function

clr\_chs(1); //calling of colour function

clr\_match(); //calling of clr\_match

}

else

cout<<"MUKESH"<<endl; //will never get displayed

cont: //Graphics initialisation

graphics\_screen();

if(gm\_ch==1)

{

cout<<"\nplayer 1's coins are of the colour"<<endl;

setfillstyle(1,clr1);

fillellipse(300,25,10,10);

cout<<"\n\nplayer 2's coins are of the colour"<<endl;

setfillstyle(1,clr2);

fillellipse(300,70,10,10);

getch();

cleardevice();

}

draw(); //calling of draw function

modi(1); //calling of modi function

do

{

gotoxy(1,1);

for(i=0;i<28;++i)

cout<<"\t\t\t\t\t\t "<<endl;

if(count%2==0)

{

gotoxy(1,1);

cout<<"player 1"<<endl;

ply=1;

}

else

{

gotoxy(1,1);

cout<<"player 2"<<endl;

ply=2;

}

top: cout<<"enter r and c to \"move from\""<<endl;

cin>>rw1>>cl1;

//getting inputs of move from inital coordinates

if(pos[rw1][cl1]==0||pos[rw1][cl1]==(-1))

{

cout<<"entered coordinates are wrong"<<endl

<<"enter the coordinates again"<<endl;

goto top;

}

again: cout<<"enter r and c to \"move to\""<<endl;

cin>>rw2>>cl2;

//getting inputs of move to final coordinates

mov=moves(); //calling of moves function

if(mov==0)

{

cout<<"\"illegal move\" you can only move"<<endl

<<"one step at a time"<<endl;

goto again;

}

poschg=poschag(1); //calling of poschag function

if(poschg==1)

{

cout<<"illegal move"<<endl

<<"try again"<<endl;

goto top;

}

cap=capchk(); //calling of capchk function

if(cap==1||cap==3)

{

if(cap==1)

if(ply==2)

ply--;

else

ply++;

cout<<"your \"queen\" has been captured"<<endl

<<"your queen will be moved by"<<endl

<<"your \"opponent\""<<endl;

poschag(0);

}

else if(cap==2||cap==4)

{

if(cap==2)

if(ply==2)

ply--;

else

ply++;

cout<<"your \"guard\" has been captured"<<endl

<<"your guard will be moved "<<endl

<<"by \"you\""<<endl;

poschag(0);

}

chk=winchk(ply); //calling of winchk function

if(chk==1)

{

if(count%2==0)

cout<<nm[0]<<" has WON the game"<<endl;

else

cout<<nm[1]<<" has WON the game"<<endl;

getch();

delay(800);

exit(0);

}

cout<<"do u want to continue ('y')"<<endl;

cin>>ch;

if(ch=='x')

exit(0);

else if(ch=='s')

file\_save(); //saving the game

else if(ch!='y'&&ch!='Y')

game\_choose(); //main menu calling

count++;

}while(ch=='y'||ch=='Y'||ch=='s'||ch=='S');

getch();

cout<<"MUKESH"<<endl;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO DISPLAY LOADING

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void loadin()

{

clrscr();

graphics\_screen(); //initialsing graphics

outtextxy(150,325,"please wait while the game is Loading......");

settextstyle(4,0,6);

outtextxy(225,100,"AGON");

bar(45,364,601,386);

setfillstyle(1,0);

bar(46,365,600,385);

gotoxy(7,24);

for(int i=0;i<69;i++)

{

cout<<(char)176;

if(i<32)

delay(500-15\*i);

else

delay(50);

}

settextstyle(0,0,1);

outtextxy(200,425,"press any key to continue !!");

getch();

closegraph();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO DISPLAY RULES OF THE GAME

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void rules()

{

cout<<endl<<"\t\t\t\t\"RULES\""<<endl<<endl

<<"\tEach player has one Queen and six Guards"<<endl

<<"\twhich are placed on the outer ring of the board"<<endl<<endl

<<"\tThe goal of each player is to put the Queen on the center"<<endl

<<"\tand surround her with her Guards"<<endl<<endl

<<"\tThe first player being selected by the throw of a die"<<endl<<endl

<<"\tPieces can only be moved one cell forwards or sideways"<<endl

<<"\tthey cannot be moved back away from the center of the board"<<endl

<<"\tOnly a Queen may move onto the centre cell"<<endl<<endl

<<"\tIf you try to move your oppnents piece you wil loose your turn"<<endl<<endl

<<"\tIf a Guard is trapped between two opposing pieces,"<<endl

<<"\tthe owner of the captured Guard must move it"<<endl

<<"\tto any cell of his own choosing on the outermost ring"<<endl<<endl

<<"\tIf a Queen is trapped between two opposing pieces"<<endl

<<"\tthe owner of a captured Queen must move it to any cell"<<endl

<<"\tof his opponent's choosing"<<endl;

getch();

delay(800);

cout<<"MUKESH"<<endl;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION FOR ROLLING THE DICE

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void dice()

{

int i,ran[2];

randomize();

cout<<"enter the player 1's name:"<<endl;

gets(nm[0]);

cout<<"enter the player 2's name:"<<endl;

gets(nm[1]);

roll: for(i=0;i<2;++i)

{

cout<<endl<<nm[i]<<endl

<<"the dice is being rolled"<<endl

<<"and the no obtained is"<<endl;

delay(700);

ran[i]=(random(6)+1);

cout<<ran[i]<<endl;

delay(700);

}

if(ran[0]==ran[1])

{

cout<<"both players have got the \"same\""<<endl

<<"number on the dice, so \"rolling again\""<<endl;

delay(700);

goto roll; //to roll dice again

}

if(ran[0]>ran[1])

cout<<endl<<nm[0]<<" wins the \"first chance\""<<endl;

else

cout<<endl<<nm[1]<<" wins the \"first chance\""<<endl;

cout<<endl<<"You are the \"first Player\""<<endl;

getch();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO CHOOSE COLOUR FOR COINS & BOARD

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void clr\_chs(int bk)

{

int i=0;

char cho='n';

char ch[14][3]={"1","2","3","4","5","6","7","8"

,"9","10","11","12","13","14"};

clrscr();

if(!bk) //coin's colour

{

cout<<"Do you want to choose colours for your coins"<<endl

<<"by Default"<<endl

<<"Player 1 gets \"RED\" coins"<<endl

<<"Player 2 gets \"GREEN\" coins"<<endl

<<"do you want to change ?? (y\\n)"<<endl;

cin>>cho;

}

else //board's colour

{

cout<<"Do you want to choose colours for the board"<<endl

<<"by Default"<<endl

<<"board has \"GRAY\" colour"<<endl

<<"do you want to change ?? (y\\n)"<<endl;

cin>>cho;

}

if(cho!='y')

return;

graphics\_screen(); //intialise graphics

if(!bk) //colour of coins

{

cout<<"\t\t Enter the colours you want"

<<"for your coins"<<endl;

}

else //colour of board

{

cout<<"\t\tEnter the board colour you want"<<endl;

}

for(i=0;i<6;i++)

{

outtextxy(88+80\*i,185,ch[i]);

outtextxy(88+80\*i,285,ch[i+6]);

}

for(i=0;i<6;i++)

{

setcolor(i+1);

setfillstyle(1,i+1);

fillellipse(90+i\*80,160,20,20);

}

for(i=8;i<14;i++)

{

setcolor(i+1);

setfillstyle(1,i+1);

fillellipse(90+(i-8)\*80,260,20,20);

}

setcolor(15);

setfillstyle(1,15);

fillellipse(90+80\*2,360,20,20);

outtextxy(85+80\*2,385,ch[12]);

setcolor(15);

setfillstyle(1,16);

fillellipse(90+80\*3,360,19,19);

outtextxy(85+80\*3,385,ch[13]);

enter: gotoxy(1,5);

if(!bk) //colour of coins

{

cout<<nm[0]<<"-choose the colour :";

cin>>clr1;

cout<<nm[1]<<"-choose the colour :";

cin>>clr2;

if((clr1==clr2)||(clr1>14||clr2>14))

{

gotoxy(1,5);

for(i=0;i<4;i++)

cout<<"\t\t\t\t\t\t\t\t"<<endl;

gotoxy(1,4);

cout<<"enter again as chooses are wrong or same"<<endl;

goto enter;

}

if(clr1>6)

clr1+=2;

if(clr2>6)

clr2+=2;

}

else //colour of board

{

cout<<"enter the colour"<<endl;

cin>>bkclr;

if(bkclr>14)

{

gotoxy(1,5);

for(i=0;i<4;i++)

cout<<"\t\t\t\t\t\t\t\t"<<endl;

gotoxy(1,4);

cout<<"enter again as choose is wrong"<<endl;

goto enter;

}

if(bkclr>6)

bkclr+=2;

}

cout<<"MUKESH"<<endl;

closegraph();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO CHECK FOR SAME COLOUR OF COINS & BOARD

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void clr\_match()

{

if((clr1!=bkclr)&&(clr2!=bkclr))

return;

cout<<"colour of the coin and board are same"<<endl

<<"enter them again and make sure dey are diff"<<endl;

getch();

clr\_chs(0); //chosing colour of coins

clr\_chs(1); //chosing colour of board

clr\_match(); //colour match

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO CHOOSE THE GAME IN START

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int game\_cho()

{

int fr;

char ch;

gm: cho=0;

cout<<"Do you you want to :-"<<endl

<<"1.Play the new game"<<endl

<<"2.Play the saved game "<<endl

<<"3.Any other key to exit"<<endl

<<"enter ur choice:";

cin>>cho;

if(cho==1)

{

nw\_gm: clrscr();

cout<<"\n\n\n\n\n\n\n\n\n\n\n\t\t\t";

cout<<" NEW GAME STARTED";

return(0);

}

else if(cho==2)

{

fr=file\_read();

if(fr==0)

goto nw\_gm;

else

return(1);

}

else

{

cout<<"are u sure u wanna exit this amazing game??"<<endl;

cin>>ch;

if(ch!='y')

{

clrscr();

goto gm;

}

}

return(-1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO CHOOSE THE GAME

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void game\_choose()

{

int fr;

char ch12,sav,cho12;

gm: cho=0;

closegraph();

cout<<"Do you you want to :-"<<endl

<<"1.Play the new game"<<endl

<<"2.Continue current game"<<endl

<<"3.Save the current game"<<endl

<<"4.See the saved games"<<endl

<<"5.Play the saved game "<<endl

<<"6.Delete a exirsting file"<<endl

<<"7.Read the instructions"<<endl

<<"8.Choose ur coin colour"<<endl

<<"9.Choose board colour"<<endl

<<"\nAny other key to exit"<<endl

<<"\nenter ur choice:";

cin>>cho12;

cho=(char)cho12;

if(cho==49)

{

nw\_gm:

cout<<"\n\n\n\n\n\n\n\n\n\n\n\t\t\t";

cout<<" NEW GAME STARTED";

save c;

c.copy\_4m();

new\_gm=1;

count=-1;

ch='y';

}

else if(cho==50)

{

ch='y';

}

else if(cho==51)

{

file\_save();

ch='y';

}

else if(cho==52)

{

file\_list();

getch();

ch='y';

}

else if(cho==53)

{

fr=file\_read();

if(fr==0)

goto nw\_gm;

else

{

count--;

ch='y';

}

}

else if(cho==54)

{

file\_del();

clrscr();

ch='y';

}

else if(cho==55)

{

rules();

ch='y';

}

else if(cho==56)

{

clr1=4;

clr2=2;

clr\_chs(0);

clr\_match();

graphics\_screen();

draw(); //calling of draw function

modi(1); //calling of modi function

ch='y';

}

else if(cho==57)

{

bkclr=8;

clr\_chs(1);

clr\_match();

graphics\_screen();

draw(); //calling of draw function

modi(1); //calling of modi function

ch='y';

}

else

{

cout<<"are u sure u wanna exit this amazing game??"<<endl;

cin>>ch12;

if(ch12!='y')

{

ch='y';

graphics\_screen();

draw(); //calling of draw function

modi(1); //calling of modi function

return;

}

cout<<"do u want to save be4 exiting ??"<<endl;

cin>>sav;

if(sav=='y')

file\_save();

getch();

credits();

getch();

exit(0);

}

graphics\_screen();

draw(); //calling of draw function

modi(1); //calling of modi function

if(new\_gm==1)

{

fillellipse(450,65,5,5);

fillellipse(430,145,5,5);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO SHOW SAVE FILES

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void file\_list()

{

char file\_nm[30];

file.open("txtfl\_MK.txt",ios::in|ios::out);

if(!file)

{

cout<<"No Saved files found"<<endl

<<"Press any key to continue"<<endl;

getch();

return;

}

cout<<"the existing save files are:"<<endl;

while(!file.eof())

{

file.getline(file\_nm,30,'\n');

cout<<file\_nm<<endl;

}

file.close();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO SAVE THE GAME

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void file\_save()

{

char names[30],ch;

int f\_flag=0,rep=0;

file\_list();

cout<<"enter file name to save as"<<endl;

gets(fl\_nm);

strcpy(names,fl\_nm);

strcat(fl\_nm,".dat");

ofstream tochk(fl\_nm,ios::nocreate);

if(tochk)

{

cout<<"file already exists"<<endl

<<"do u want to replace old file"<<endl;

cin>>ch;

if(ch=='y')

{

f\_flag=1;

rep=1;

}

}

else

f\_flag=1;

tochk.close();

if(!f\_flag)

return;

count++;

s.copy\_to();

count--;

file.open(fl\_nm,ios::out|ios::binary);

if(!file)

{

cout<<"file not opened save"<<endl;

flag=0;

getch();

return;

}

file.write((char\*)&s,sizeof(s));

file.close();

if(!rep)

{

file.open("txtfl\_MK.txt",ios::app);

if(!file)

{

cout<<"text file not opened"<<endl;

getch();

return;

}

if(file)

{

file<<names<<"\n";

cout<<"save successful"<<endl;

}

file.close();

}

getch();

return;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO READ FROM SAVE FILE

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int file\_read()

{

file\_list();

cout<<"enter file name to read from"<<endl;

gets(fl\_nm);

strcat(fl\_nm,".dat");

file.open(fl\_nm,ios::in|ios::binary);

if(!file)

{

cout<<"file does not exist read"<<endl;

getch();

return(0);

}

file.read((char\*)&s,sizeof(s));

s.copy\_4m();

cout<<"data retreaved successfully"<<endl;

file.close();

getch();

return(1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO DELETE A FILE

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void file\_del()

{

char name[30],buff[30];

file\_list();

cout<<"enter the file name which u wanna delete"<<endl;

gets(name);

strcpy(fl\_nm,name);

strcat(name,".dat");

file.open(name,ios::in);

if(!file)

{

cout<<"file does not exist delete"<<endl;

getch();

return;

}

file.close();

ofstream temp("temp.txt",ios::out);

if(!temp)

{

cout<<"temp file does not exist"<<endl;

getch();

return;

}

file.open("txtfl\_MK.txt",ios::in);

if(!file)

{

cout<<"text file not opened"<<endl;

getch();

return;

}

while(!file.eof())

{

file.getline(buff,20,'\n');

if(strcmp(buff,fl\_nm)!=0)

temp<<buff<<"\n";

}

remove(name);

file.close();

temp.close();

remove("txtfl\_MK.txt");

rename("temp.txt","txtfl\_MK.txt");

cout<<"delete succesful"<<endl;

getch();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO DRAW THE HEXAGON

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void draw()

{

int i,j,n=6,x=460,y=10;

setbkcolor(0);

setfillstyle(1,bkclr);

for(j=0,y;j<6;++j,++n)

{

for(i=0;i<n;x+=20,++i)

{

hexagon(x,y); //calling of hexagon function

fillpoly(7,poly);

}

x=x-10-20\*n;

y+=20;

}

x+=20;

n=n-2;

for(j=0;j<5;++j,--n)

{

for(i=0;i<n;x+=20,++i)

{

hexagon(x,y); //calling of hexagon function

fillpoly(7,poly);

}

x=x+10-20\*n;

y+=20;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO CHANGE THE COORDINATES

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void hexagon (int x, int y)

{

poly[0]=x+10;

poly[1]=y;

poly[2]=x+20;

poly[3]=y+10;

poly[4]=x+20;

poly[5]=y+20;

poly[6]=x+10;

poly[7]=y+30;

poly[8]=x;

poly[9]=y+20;

poly[10]=x;

poly[11]=y+10;

poly[12]=x+10;

poly[13]=y;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO CHECK FOR CORRECT MOVES

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int moves()

{

int x=rw1,y=cl1,i,j,k;

if(x==y) //diagonal (0,0) to (4,4)

{

if(x<5)

if((rw2==(x+1)&&cl2==(y+1))||(rw2==x&&

cl2==(y+1))||(rw2==(x+1)&&cl2==y))

return(1);

}

for(i=0;i<5;i++)

{

if(rw1==5&&cl1==i) //diagonal (5,0) to (5,4)

if((rw2==4&&cl2==i)||(rw2==5&&

cl2==(i+1))||(rw2==6&&cl2==i))

return(1);

if(rw1==i&&cl1==5) //diagonal (0,5) to (4,5)

if((rw2==i&&cl2==4)||(rw2==(i+1)&&

cl2==5)||(rw2==(i+1)&&cl2==6))

return(1);

}

for(i=10,j=0;i>j;--i,++j)

{

if(rw1==i&&cl1==j) //diagonal (10,0) to (6,0)

if((rw2==(i-1)&&cl2==j)||(rw2==(i-1)&&

cl2==(j+1))||(rw2==i&&cl2==(j+1)))

return(1);

}

for(i=10;i>5;--i)

{

if(rw1==i&&cl1==5) //diagoanl (10,5) to (6,5)

if((rw2==i&&cl2==4)||(rw2==(i-1)&&

cl2==5)||(rw2==(i-1)&&cl2==6))

return(1);

if(rw1==5&&cl1==i) //diagonal (5,10) to (5,6)

if((rw2==4&&cl2==(i-1))||(rw2==5&&

cl2==i)||(rw2==6&&cl2==(i-1)))

return(1);

}

for(i=0;i<4;++i) //Triangle 1

for(j=4;j>0;--j)

{

if(rw1==j&&cl1==i)

if((rw2==(j-1)&&cl2==i)||

(rw2==j&&cl2==(i+1))||(rw2==(j+1)&&

cl2==(i+1))||rw2==(j+1)&&cl2==i)

return(1);

}

for(i=6;i<10;++i) //Triangle 2

for(k=9,j=0;k>=i;++j,--k)

{

if(rw1==i&&cl1==j)

if((rw2==(i-1)&&cl2==j)||(rw2==(i-1)&&

cl2==(j+1))||(rw2==i&&cl2==(j+1))||

(rw2==(i+1)&&cl2==j))

return(1);

}

for(i=10;i>6;--i) //Triangle 3

for(j=4,k=6;k<i;++k,--j)

{

if(rw1==i&&cl1==j)

if((rw2==i&&cl2==(j-1))||(rw2==(i-1)&&

cl2==j)||(rw2==(i-1)&&cl2==(j+1))||

(rw2==i&&cl2==(j+1)))

return(1);

}

for(i=6;i<10;++i) //Triangle 4

for(j=6,k=9;k>=i;--k,++j)

{

if(rw1==i&&cl1==j)

if((rw2==(i+1)&&cl2==(j-1))||(rw2==i&&

cl2==(j-1))||(rw2==(i-1)&&cl2==j)||

(rw2==(i-1)&&cl2==(j+1)))

return(1);

}

for(i=4;i>0;--i) //Triangle 5

for(k=0,j=6;k<i;++j,++k)

{

if(rw1==i&&cl1==j)

if((rw2==(i+1)&&cl2==(j+1))||(rw2==(i+1)&&

cl2==j)||(rw2==i&&cl2==(j-1))||

(rw2==(i-1)&&cl2==(j-1)))

return(1);

}

for(i=0;i<4;++i) //Triangle 6

for(k=4,j=4;k>i;--k,--j)

{

if(rw1==i&&cl1==j)

if((rw2==i&&cl2==(j+1))||(rw2==(i+1)&&

cl2==(j+1))||(rw2==(i+1)&&cl2==j)||

(rw2==i&&cl2==(j-1)))

return(1);

}

return(0);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION FOR POSTION MODIFICATION

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void modi(int z)

{

int prw[7],pcl[7];

for(int j=0;j<2;++j)

{

for(int k=0;k<7;++k)

if(plyr[j]==1)

{

setfillstyle(1,clr1);

prw[k]=p1rw[k];

pcl[k]=p1cl[k];

}

else

{

setfillstyle(1,clr2);

prw[k]=p2rw[k];

pcl[k]=p2cl[k];

}

if(z==1)

position(prw,pcl);//calling of position function

for(int a=0;a<=rw1;++a)

if(prw[a]==rw1&&pcl[a]==cl1)

if(j==0&&ply==1)

{

p1rw[a]=rw2;

p1cl[a]=cl2;

}

else if(j==1&&ply==2)

{

p2rw[a]=rw2;

p2cl[a]=cl2;

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO PRINT THE POSITIONS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void position(int\*prw,int\*pcl)

{

int x=460,y=10,r=5,X,Y;

for(int i=0;i<7;++i)

if(prw[i]<=5)

{

X=x+10\*(2\*pcl[i]+1)-10\*prw[i];

Y=y+15+20\*prw[i];

fillellipse(X,Y,r,r);

}

else if(prw[i]>5)

{

int re;

re=prw[i]-6;

X=x+10\*(2\*pcl[i]+1)-10\*prw[i]+20\*(re+1);

Y=y+135+20\*re;

fillellipse(X,Y,r,r);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION FOR POISTION CHANGING

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int poschag(int z)

{

int a,b,i,j,x=460,y=10,r=5,X,Y,coin=0,lst;

if(ply==1)

{

a=11;

b=1;

}

else

{

a=22;

b=2;

}

if(z==0)

{

agn: lst=lstrw();

if(lst==0)

goto agn;

}

if((pos[rw1][cl1]==a||pos[rw1][cl1]==b)&&pos[rw2][cl2]==0)

{

if(pos[rw1][cl1]==a)

{

if(ply==1)

coin=11;

else

coin=22;

}

if(pos[rw1][cl1]==b)

{

if(ply==1)

coin=1;

else

coin=2;

}

//To errase the circle

setcolor(bkclr);

setfillstyle(1,bkclr);

if(rw1<=5)

{

X=x+10\*(2\*cl1+1)-10\*rw1;

Y=y+15+20\*rw1;

fillellipse(X,Y,r,r);

}

if(rw1>5)

{

int re;

re=rw1-6;

X=x+10\*(2\*cl1+1)-10\*rw1+20\*(re+1);

Y=y+135+20\*re;

fillellipse(X,Y,r,r);

}

//To make the new circle

setcolor(15);

if(ply==1)

setfillstyle(1,clr1);

else

setfillstyle(1,clr2);

if(rw2<=5)

{

X=x+10\*(2\*cl2+1)-10\*rw2;

Y=y+15+20\*rw2;

fillellipse(X,Y,r,r);

}

if(rw2>5)

{

int re;

re=rw2-6;

X=x+10\*(2\*cl2+1)-10\*rw2+20\*(re+1);

Y=y+135+(20\*re);

fillellipse(X,Y,r,r);

}

pos[rw1][cl1]=0;

pos[rw2][cl2]=coin;

modi(0); //calling of modi function

}

else

return(1);

return(0);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO CHECK FOR CAPTURE

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int capchk()

{

int a1,a2,b1,b2,x=rw2,y=cl2,flag,flag\_cap;

if(ply==2)

{

a1=11;

b1=1;

b2=2;

}

else

{

a1=22;

b1=2;

b2=1;

}

if(x<=5)

{

if(x-1!=-1&&y-1!=-1&&y!=x+6)

/\*Centre up\*/ if((pos[x][y]==a1||pos[x][y]==b1)&&

((pos[x-1][y]==b2&&pos[x+1][y]==b2)||

(pos[x][y+1]==b2&&pos[x][y-1]==b2)||

(pos[x+1][y+1]==b2&&pos[x-1][y-1]==b2)))

{

rw1=x;

cl1=y;

if(pos[x][y]==a1)

return(1);

else if(pos[x][y]==b1)

return(2);

}

if(x-1!=-1&&x-2!=-1)

/\*Corner up 1\*/ if((pos[x][y]==b2)&&(pos[x-2][y]==b2)&&

(pos[x-1][y]==a1||pos[x-1][y]==b1))

{

rw1=x-1;

cl1=y;

if(pos[x-1][y]==a1)

return(1);

else if(pos[x-1][y]==b1)

return(2);

}

if(x-1!=-1&&y-1!=-1&&x-2!=-1&&y-2!=-1)

/\*Corner up 2\*/ if((pos[x-1][y-1]==a1||pos[x-1][y-1]==b1)&&

(pos[x][y]==b2)&&(pos[x-2][y-2]==b2))

{

rw1=x-1;

cl1=y-1;

if(pos[x-1][y-1]==a1)

return(1);

else if(pos[x-1][y-1]==b1)

return(2);

}

if(y-1!=-1&&y-2!=-1)

/\*Corner up 3\*/ if((pos[x][y-1]==a1||pos[x][y-1]==b1)&&

(pos[x][y]==b2)&&(pos[x][y-2]==b2))

{

rw1=x;

cl1=y-1;

if(pos[x][y-1]==a1)

return(1);

else if(pos[x][y-1]==b1)

return(2);

}

/\*Corner up 4\*/ if((pos[x+1][y]==a1||pos[x+1][y]==b1)&&

(pos[x][y]==b2)&&(pos[x+2][y]==b2))

{

rw1=x+1;

cl1=y;

if(pos[x+1][y]==a1)

return(1);

else if(pos[x+1][y]==b1)

return(2);

}

if(y!=x+6)

/\*Corner up 5\*/ if((pos[x+1][y+1]==a1||pos[x+1][y+1]==b1)&&

(pos[x+2][y+2]==b2)&&(pos[x][y]==b2))

{

rw1=x+1;

cl1=y+1;

if(pos[x+1][y+1]==a1)

return(1);

else if(pos[x+1][y+1]==b1)

return(2);

}

if(y!=x+6)

/\*Corner up 6\*/ if((pos[x][y+1]==a1||pos[x][y+1]==b1)&&

(pos[x][y+2]==b2)&&(pos[x][y]==b2))

{

rw1=x;

cl1=y+1;

if(pos[x][y+1]==a1)

return(1);

else if(pos[x][y+1]==b1)

return(2);

}

}

else

{

/\*Center down\*/

if((pos[x][y]==a1||pos[x][y]==b1)&&

((pos[x-1][y+1]==b2&&pos[x+1][y-1]==b2)||

(pos[x][y+1]==b2&&pos[x][y-1]==b2)||

(pos[x+1][y]==b2&&pos[x-1][y]==b2)))

{

rw1=x;

cl1=y;

if(pos[x][y]==a1)

return(1);

else if(pos[x][y]==b1)

return(2);

}

if(x-1!=-1&&x-2!=-1&&x+y!=16)

/\*Corner down1\*/if((pos[x-1][y+1]==a1||pos[x-1][y+1]==b1)&&

(pos[x-2][y+2]==b2)&&(pos[x][y]==b2))

{

rw1=x-1;

cl1=y+1;

if(pos[x-1][y+1]==a1)

return(1);

else if(pos[x-1][y+1]==b1)

return(2);

}

if(x+y!=16)

/\*Corner down2\*/if((pos[x][y+1]==a1||pos[x][y+1]==b1)&&

(pos[x][y+2]==b2)&&(pos[x][y]==b2))

{

rw1=x;

cl1=y+1;

if(pos[x][y+1]==a1)

return(1);

else if(pos[x][y+1]==b1)

return(2);

}

if(x+1!=11&&x+2!=11)

/\*Corner down3\*/if((pos[x+1][y]==a1||pos[x+1][y]==b1)&&

(pos[x+2][y]==b2)&&(pos[x][y]==b2))

{

rw1=x+1;

cl1=y;

if(pos[x+1][y]==a1)

return(1);

else if(pos[x+1][y]==b1)

return(2);

}

if(x+1!=11&&y-1!=-1&&x+2!=11&&y-2!=-1)

/\*Corner down4\*/if((pos[x+1][y-1]==a1||pos[x+1][y-1]==b1)&&

(pos[x][y]==b2)&&(pos[x+2][y-2]==b2))

{

rw1=x+1;

cl1=y-1;

if(pos[x+1][y-1]==a1)

return(1);

else if(pos[x+1][y-1]==b1)

return(2);

}

if(y-1!=-1&&y-2!=-1)

/\*Corner down5\*/if((pos[x][y-1]==a1||pos[x][y-1]==b1)&&

(pos[x][y]==b2)&&(pos[x][y-2]==b2))

{

rw1=x;

cl1=y-1;

if(pos[x][y-1]==a1)

return(1);

else if(pos[x][y-1]==b1)

return(2);

}

if(x-2!=-1&&x-1!=-1)

/\*Corner down6\*/if((pos[x-1][y]==a1||pos[x-1][y]==b1)&&

(pos[x][y]==b2)&&(pos[x-2][y]==b2))

{

rw1=x-1;

cl1=y;

if(pos[x-1][y]==a1)

return(1);

else if(pos[x-1][y]==b1)

return(1);

}

}

/\*Center up \*/

a1=11;

b1=1;

b2=2;

if(pos[x][y]==a1||pos[x][y]==b1)

{

if(x-1!=-1&&y-1!=-1)

{

if(pos[x-1][y]==b2&&pos[x+1][y]==b2)

flag\_cap=1;

if(pos[x+1][y+1]==b2&&pos[x-1][y-1]==b2)

flag\_cap=1;

if(pos[x][y+1]==b2&&pos[x][y-1]==b2)

flag\_cap=1;

}

if(flag\_cap==1)

{ cout<<"";

rw1=x;

cl1=y;

if(pos[x][y]==a1)

return(1);

else if(pos[x][y]==b1)

return(2);

}

}

/\*Center down\*/

a1=22;

b1=2;

b2=1;

if((pos[x][y]==a1||pos[x][y]==b1)&&

((pos[x-1][y+1]==b2&&pos[x+1][y-1]==b2)||

(pos[x][y+1]==b2&&pos[x][y-1]==b2)||

(pos[x+1][y]==b2&&pos[x-1][y]==b2)))

{

rw1=x;

cl1=y;

if(pos[x][y]==a1)

return(3);

else if(pos[x][y]==b1)

return(4);

}

return(0);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION MOVE TO LAST ROW

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int lstrw()

{

int i,j;

cout<<"enter r and c to move the \"captured peice\" too"<<endl

<<"make sure that the piece is "<<endl

<<"movied to the \"last ring\""<<endl;

cin>>rw2>>cl2;

if(pos[rw2][cl2]==0)

{

for(i=0;i<11;++i)

if(rw2==i&&cl2==0)//left up & dwn

return(1);

for(i=0;i<6;++i)

{

if(rw2==10&&cl2==i)//dwn

return(1);

if(rw2==0&&cl2==i)//up

return(1);

}

for(i=0,j=5;i<6;++i,++j)

if(rw2==i&&cl2==j)//right up

return(1);

for(i=5,j=10;i<11;++i,--j)

if(rw2==i&&cl2==j)//right down

return(1);

}

return(0);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO CHECK FOR WIN

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int winchk(int ply)

{

int x,y;

if(ply==1)

{

x=11;

y=1;

}

else

{

x=22;

y=2;

}

if(pos[5][5]==x&&pos[4][4]==y&&pos[4][5]==y&&

pos[5][4]==y&&pos[5][6]==y&&pos[6][4]==y&&pos[6][5]==y)

return(1);

else

return(0);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION TO DISPLAY CREDITS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void credits()

{

cout<<"\t\t\t \"THANKS FOR PLAYING\""<<endl

<<"\n\t\t\t\t \"AGON\"\n"<<endl

<<"\t\t\t\ta C++ Project"<<endl

<<"\t\t\tDone by:-"<<endl

<<"\t\t\t (Lead Programer)"<<endl

<<"\t\t\t\tA.Mukesh"<<endl

<<"\t\t\t (Programers)"<<endl

<<"\t\t\t\tP.Pravin Kumar"<<endl

<<"\t\t\t\tR.Madhumitha"<<endl

<<"\t\t\t\tV.Prabha"<<endl

<<"\n\n\t\tOur Sincere thanks to your Computer Teacher"<<endl

<<"\t\t\t \"Mr.Kamal Kishore Sharma\""<<endl

<<"\t\tfor being a spectacular support and guide"<<endl

<<"\t\tthroughout the period of our project."<<endl

<<"\t\tHe has been motivating and inspiring us."<<endl

<<"\t\tAlso, he has given great opportunities to explore"<<endl

<<"\t\t\tand instill programming skills."<<endl

<<"\t\tI also thank my friends R.Ashwin and Shagesh"<<endl

<<"\t\tfor helping us in making this project"<<endl

<<"\t\t\t\ta succesful one"<<endl

<<"\t\t\t\t thank you ALL"<<endl;

}

/\* we thank our computer teacher for being a spectacular support and guide throughout the period of our project making. He has been motivating and inspirin us. Also, he has given great opportunities to explore and instill programming skills. \*/