**Chess Game**

****

Session- 2019-2020

**B.Tech(CSE), 8th Semester**

**Computer Graphics**

**Batch- B1**

**Submitted To- Submitted By-**

Dr. K. F. Rahman Parnika Agarwal(1611257)

Radha Chirania(1611280)

Saloni Jain(1611313)

**Abstract**

This chess game is just similar to games which are found under computer game section. It uses the concept of graphics to display all the objects on a single screen. The Chess game follows the basic rules of chess, and all the chess pieces only move according to valid moves for that piece. Our implementation of Chess is for two players. It is played on an 8\*8 checkered board, with a dark square in each player’s lower left corner.

**Inbuilt Functions used in the code**

1. **Line:**- It is used to draw a line from a point(x1,y1) to point(x2,y2) i.e. (x1,y1) and (x2,y2) are end points of the line.

**Declaration: -** void line(int x1, int y1, int x2, int y2);

**2. Arc:-**Arc is used to draw circular arc. It takes 5 arguments, all of the int type.

**Declaration: -**arc(int x, int y, int starting-angle,int ending-angle, int radius);

**3. Rectangle:-**Rectangle is used to draw empty rectangle.

**Declaration: -**rectangle(left, top, right, bottom);

**4. cleardevice: -** It is used to clear the screen with its background color and work like a clrscr function in c programming.

**Declaration: -**cleardevice();

**5. setcolor:-**is used set the foreground color in graphics mode.

**Declaration:-**setcolor (intcolor);

**6. outtextxy:-** It is the function used to display the given string on graphics mode at specific position .

**Declaration:-**outtextxy(int x-pos, int y-pos, char \*string);

**7. bar:-** This function is used to draw rectangular bar.

**Declaration:-** bar(int left, int top, int right, int bottom);

**8. settextstyle:-** This function is used to set text style of given string. It modify the font style, direction of string and font size. It is used with outtext or outtextxy function.

**Declaration:-** settextstyle(int fontstyle, int direction, int char-size)

**9. delay:-** It is used to suspend the execution of a program for a particular time.

**Declaration:-** void delay(unsigned int)

**SOURCE CODE**

#include <graphics.h>

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

#include <ctype.h> //For toupper

#include <dos.h> //For Delay

int board[8][8],turn=0,count1=0,count2=0;

void printboard(){

int i,j;

for(i=0;i<8;i++){

for(j=0;j<8;j++){

if((i+j)%2==0)

setcolor(WHITE);

else

setcolor(BLACK);

outtextxy(270+(j\*45),105+(i\*45),"ÛÛ");

if(board[i][j]<0)

setcolor(GREEN);

else if(board[i][j]>0)

setcolor(9);

if(abs(board[i][j])>=9 && abs(board[i][j])<=16){

outtextxy(270+(j\*45),105+(i\*45),"P");}

else if(abs(board[i][j])==1 || abs(board[i][j])==8){

outtextxy(270+(j\*45),105+(i\*45),"R");}

else if(abs(board[i][j])==2 || abs(board[i][j])==7){

outtextxy(270+(j\*45),105+(i\*45),"H");

}

else if(abs(board[i][j])==3 || abs(board[i][j])==6){

outtextxy(270+(j\*45),105+(i\*45),"B");

}

else if(abs(board[i][j])==4){

outtextxy(270+(j\*45),105+(i\*45),"Q");

}

else if(abs(board[i][j])==5){

outtextxy(270+(j\*45),105+(i\*45),"K");

}

}

}

}

void wrongStart()

{

char ch;

int i,j;

printf("\nWrong Input\nPress any key to continue...");

ch=getch();

if(ch==0){

getch();

}

for(i=1;i<=30;i++)

for(j=1;j<=4;j++)

{

gotoxy(i,j);

printf(" ");

}

gotoxy(1,1);

printf("enter initial position ¯ ");

}

/////ClearText//

void cleartext(){

int i,j;

for(i=1;i<=27;i++)

for(j=1;j<=8;j++){

gotoxy(i,j);

printf(" ");

}

gotoxy(1,1);

printf("enter initial position ¯ ");

}

/////Initialize////

void initialize(){

int i,j;

setcolor(WHITE);

rectangle(250,90,610,450);

// Board Printing

for (i=250;i<560;i+=90)

for (j=90;j<450;j+=90)

{

bar(i,j,i+45,j+45);

bar(i+45,j+45,i+90,j+90);

}

//Asigning the Values in the empty spaces

for(i=2;i<=5;i++)

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

board[i][j] = 0;

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

{

board[0][j] = j+1;

board[1][j] = j+9; //Pawns for player 1

board[7][j] = 0-(j+1);

board[6][j] = 0-(j+9); //Pawns for player 2

}

printboard();

setcolor(YELLOW);

outtextxy(265,55,"A");

outtextxy(310,55,"B");

outtextxy(355,55,"C");

outtextxy(400,55,"D");

outtextxy(445,55,"E");

outtextxy(490,55,"F");

outtextxy(535,55,"G");

outtextxy(580,55,"H");

outtextxy(220,105,"1");

outtextxy(220,150,"2");

outtextxy(220,195,"3");

outtextxy(220,240,"4");

outtextxy(220,285,"5");

outtextxy(220,330,"6");

outtextxy(220,375,"7");

outtextxy(220,420,"8");

}

////HORSE//////

int horse(int x,int y,int x1,int y1)

{

int a=0;

if((y1==y+2 && x1==x+1)||(y1==y+2 && x1==x-1)||(y1==y+1 &&

x1==x+2)||(y1==y+1 && x1==x-2)||(y1==y-1 && x1==x+2)||(y1==y-1 &&

x1==x-2)||(y1==y-2 && x1==x+1)||(y1==y-2 && x1==x-1)){

a = 1;

}

return a;

}

/////KING////

int king(int x,int y,int x1,int y1)

{

int a = 0;

if((y1==y+1 && x1==x)||(y1==y-1 && x1==x)||(y1==y+1 &&

x1==x+1)||(y1==y+1 && x1==x-1)||(y1==y-1 && x1==x+1)||(y1==y-1 &&

x1==x-1)||(y1==y && x1==x+1)||(y1==y && x1==x-1)){

a = 1;

}

/////CASTLING//////

else

if(board[y][x+1]==0&&board[y][x+2]==0&&(abs(board[y1][x1])

==abs(board[y][x+3])==1||abs(board[y1][x1])==

abs(board[y][x+3])==8)){

return 2;

}

if(a==1)

{

if(board[y][x]==5)

count1=1;

if(board[y][x]==-5)

count2=1;

}

return a;

}

//////BISHOP/////

int bishop(int x,int y,int x1,int y1)

{

int a=1,i;

if(abs(x1-x)!=abs(y1-y))

{

a=0;

}

if((x<x1)&&(y<y1))

{

for(i=1;(i+x)<x1;i++)

{

if(board[y+i][x+i]!=0)

a=0;

}

}

else if((x>x1)&&(y>y1)){

for(i=1;(x-i)>x1;i++){

if(board[y-i][x-i]!=0)

a=0;

}

}

else if((x>x1)&&(y<y1)){

for(i=1;(x-i)>x1;i++){

if(board[y+i][x-i]!=0)

a=0;

}

}

else if((x<x1)&&(y>y1)){

for(i=1;(y-i)>y1;i++){

if(board[y-i][x+i]!=0)

a=0;

}

}

return a;

}

////////QUEEN///////

int queen(int x,int y,int x1,int y1){

if(x==x1||y==y1){

//if queen moves in + direction

return rook(x,y,x1,y1);

}

else if(abs(x1-x)==abs(y-y1)){

//if queen moves in diagnoldirection

return bishop(x,y,x1,y1);

}

else

return 0;

}

///ROOK////

int rook(int x,int y,int x1,int y1){

int a=1,i,j;

if(y1==y)

{

for(i=x+1;i<x1;i++)

{

if(board[y1][i]!=0)

{

a = 0;

break;

}

}

for(i=x-1;i>x1;i--)

{

if(board[y1][i]!=0)

{

a = 0;

break;

}

}

}

else if(x1==x)

{

for(i=y+1;i<y1;i++)

{

if(board[i][x1]!=0)

{

a = 0;

break;

}

}

for(i=y-1;i>y1;i--)

{

if(board[i][x1]!=0)

{

a = 0;

break;

}

}

}

else

{

a=0;

}

return a;

}

///PAWN/////

int pawn(int x,int y,int x1,int y1){

int a=0;

if(turn==0){

if(y==1){

if(y1==(y+2) && x1==x){

if(board[y1][x1]==0&&board[y+1][x]==0){

a = 1;

}

}

}

if(y1==y+1 && x1==x){

if(board[y1][x1]==0){

a = 1;

}

}

else if(y1==(y+1) && (x1==(x+1)||x1==(x-1))){

if(board[y1][x1]<0){

a = 1;

}

}

}

else if(turn==1){

if(y==6){

if( y1==(y-2) && x1==x){

if(board[y1][x1]==0&&board[y-1][x]==0){

a = 1;

}

}

}

if( y1==(y-1) && x1==x){

if(board[y1][x1]==0){

a = 1;

}

}

else if(y1==(y-1) && (x1==(x-1)||x1==(x+1))){

if(board[y1][x1]>0){

a = 1;

}

}

}

if(a==1)

{

if(turn==0)

{

if(y1==7)

return 2;

}

else

{

if(y1==0)

return 2;

}

}

return a;

}

////////Check////////

void check()

{

int t=0,i,j,x1,y1;

if(turn==0){

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

{

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

{

if(board[i][j]==-5)

{

y1=i; x1=j;

break;

}

}

}

for(j=0;j<=7;j++){

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

if (t==1){

setcolor(9);

rectangle(20,335,100,365);

outtextxy(30,340,"Check");

break;

}

if(board[j][i]>=9 && board[j][i]<=16)

t = pawn(i,j,x1,y1);

else if(board[j][i]==2 || board[j][i]==7)

t = horse(i,j,x1,y1);

else if(board[j][i]==4)

t = queen(i,j,x1,y1);

else if(board[j][i]==1 || board[j][i]==8)

t = rook(i,j,x1,y1);

else if(board[j][i]==3 || board[j][i]==6)

t = bishop(i,j,x1,y1);

}// for

}// for

}//endif turn

else {

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

for(j=0;j<=7;j++){

if(board[i][j]==5){

y1=i; x1=j;

break;

}

}

}

for(j=0;j<=7;j++){

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

if (t==1){

setcolor(GREEN);

rectangle(20,335,100,365);

outtextxy(30,340,"Check");

break;

}

if(board[j][i]>=(-9) && board[j][i]<=(-16))

t = pawn(i,j,x1,y1);

else if(board[j][i]==(-2) || board[j][i]==(-7))

t = horse(i,j,x1,y1);

else if(board[j][i]==(-4))

t = queen(i,j,x1,y1);

else if(board[j][i]==(-1) || board[j][i]==(-8))

t = rook(i,j,x1,y1);

else if(board[j][i]==(-3) || board[j][i]==(-6))

t = bishop(i,j,x1,y1);

}// for

}// for

}// end else

}

//////GAME OVER/////

int gameover(){

int i,j,a=0,b=0;

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

{

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

{

if(board[i][j]==5)

b=1;

if(board[i][j]==-5)

a=1;

}

}

if(b==0)

{

setcolor(12);

outtextxy(30,340,"Game Over");

setcolor(GREEN);

outtextxy(30,300,"Player 1 Loses");

getch();

return 0;

}

else if(a==0)

{

setcolor(9);

outtextxy(30,340,"Game Over");

outtextxy(30,300,"Player 2 Loses");

getch();

return 0;

}

return 1;

}

//main function

void main()

{

float k;

int i,j,b,killed;

char ch;

int x=-1,y=-1,x1=-1,y1=-1,game;

//Graphics Initialization

int gdriver = DETECT;

int gmode;

initgraph(&gdriver,&gmode,"C:\\turboC3\\bgi");

//Loading Intro

outtextxy(50,120,"WELCOME TO THE CHESS");

settextstyle(1,HORIZ\_DIR,3);

getch();

cleardevice();

//Displaying Group Members

setcolor(CYAN);

outtextxy(50,220,"Designed by:");

setcolor(6);

outtextxy(50,250,"Parnika Agarwal BTBTC16185");

outtextxy(50,300,"Radha Chirania BTBTC16053");

outtextxy(50,350,"Saloni Jain BTBTC16305");

getch();

cleardevice();

cleardevice();

setcolor(6);

settextstyle(1,HORIZ\_DIR,1.5);

outtextxy(50,100,"\nPress any key to play ");

getch();

cleardevice();

cleartext();

initialize();

setcolor(9);

outtextxy(10,400,"Player 1");

for(;;)

{

game=gameover();

if(game==0)

{

break;

}

ch=getch();

if(ch==0){

ch=getch();

if(ch=='<'){

cleartext();

x=-1;

y=-1;

x1=-1;

y1=-1;

continue;

}

if(ch=='=')

{

printboard();

cleartext();

x=-1;

y=-1;

x1=-1;

y1=-1;

setcolor(0);

outtextxy(10,400,"Player Û");

if(turn!=0){

check();

setcolor(2);

outtextxy(10,400,"Player 2");

}

else{

check();

setcolor(9);

outtextxy(10,400,"Player 1");

}

continue;

}

}

if(ch==27){

break;

}

else if(ch=='u'||ch=='U'){

x=-1;

x1=-1;

y1=-1;

y=-1;

cleartext();

continue;

}

if(x==-1){

ch=toupper(ch);

if(ch>=65 && ch<=72){

printf("%c",ch);

x=ch-65;

}

}

else if(y==-1)

{

if(ch>='1' && ch<='8')

{

printf("%c",ch);

y=ch-'1';

if(board[y][x]==0){

wrongStart();

x=-1;

y=-1;

}

else

{

if(turn==0)

{

if(board[y][x]<0)

//if Player 1 has selected a piece of Player 2

{

wrongStart();

x=-1;

y=-1;

}

}

else if(turn!=0)

{

if(board[y][x]>0)

{

wrongStart();

x=-1;

y=-1;

}

}

}

}

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

printf("\nenter final position ¯ ");

}

else if(x1==-1)

{ //Converting lower to upper case

ch=toupper(ch);

if(ch>=65 && ch<=72)

{

printf("%c",ch);

x1=ch-65;

}

}

else if(y1==-1)

{

if(ch>='1' && ch<='8')

{

printf("%c",ch);

y1=ch-'1';

}

b=0,killed=0;

/////pawn/////

if(abs(board[y][x])>=9 &&

abs(board[y][x])<=16){ //if piece selscted has value greater than 8and less than 17

b = pawn(x,y,x1,y1);

}//end pawn

//////horse//////

else if(abs(board[y][x])==2 ||

abs(board[y][x])==7){ //if piece selscted has value 2 and 7

b = horse(x,y,x1,y1);

}//end horse

////queen///

else if(abs(board[y][x])==4){ ////if pieceselscted has value 4

b = queen(x,y,x1,y1);

}//end queen

///king///

else if(abs(board[y][x])==5){ //if pieceselscted has value 5

b = king(x,y,x1,y1);

}//end king

///rook///

else if(abs(board[y][x])==1 ||

abs(board[y][x])==8){ //if piece selscted has value 1 and 8

b = rook(x,y,x1,y1);

}//end rook

///bishop///

else if(abs(board[y][x])==3 ||

abs(board[y][x])==6){ //if piece selscted has value 3 and 6

b = bishop(x,y,x1,y1);

}//end bishop

if(b==2){

if(abs(board[y][x])>=9 && abs(board[y][x])<=16){

char pp;

printf("\n\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\

\bwhich piece Q,R,H,B");

pp=getch();

if(turn==0){

if(pp=='r')

board[y1][x1]=1;

else if(pp=='h')

board[y1][x1]=2;

else if(pp=='b')

board[y1][x1]=3;

else if(pp=='q')

board[y1][x1]=4;

}

else{

if(pp=='r')

board[y1][x1]=-1;

else if(pp=='h')

board[y1][x1]=-2;

else if(pp=='r')

board[y1][x1]=-3;

else if(pp=='q')

board[y1][x1]=-4;

}

board[y][x]=0;

killed =1;

}

else if(abs(board[y][x])==5){

if(board[y][x]==5){

if(count1==1);

wrongStart();}

else if(board[y][x]==-5){

if(count1==1);

wrongStart();}

else{

board[y][x+1]=board[y1][x1];

board[y][x+2]=board[y][x];

board[y1][x1]=0;

board[y][x]=0;

killed=1;

}

}

printboard();

}

else if(b==1)

{

if(turn==0)

{

if(board[y1][x1]<0)

{

board[y1][x1]=board[y][x];

board[y][x] = 0;

killed=1;

}

}

if(turn!=0)

{

if(board[y1][x1]>0)

{

board[y1][x1]=board[y][x];

board[y][x] = 0;

killed=1;

}

}

if(board[y1][x1]==0)

{

int temp=board[y][x];

board[y][x]=board[y1][x1];

board[y1][x1]=temp;

killed=1;

}

printboard();

}

x=-1;

y=-1;

x1=-1;

y1=-1;

setcolor(0);

outtextxy(30,340,"Check");

if(killed==0){

wrongStart();

continue;

}

cleartext();

setcolor(0);

outtextxy(10,400,"Player Û");

if(turn==0){

check();

turn=1;

setcolor(2);

outtextxy(10,400,"Player 2");

}

else{

check();

turn=0;

setcolor(9);

outtextxy(10,400,"Player 1");

}

}//end legal move

}//end for

}//end main

**Screenshot: -**

