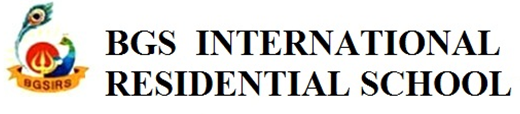
****

**THE CHESS GAME**

BY:

**ABHISHEK KUMAR (Reg. No. -4622)**

**&**

**PRADUM KUMAR (Reg. No. – 4622540)**

**Under the guidance of**

**Mrs. Divya C.**

**BGSIRS**

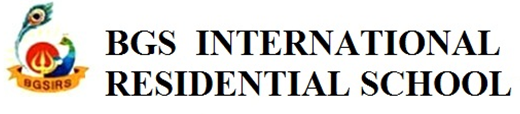
A detail report submitted in partial fulfillment

Of the requirements of the project

Of 12th CBSE Board (computer science)

Class XII

(2016-17)



**CERTIFICATE**

This is to certify that the report titled ***THE CHESS GAME*** is a bonafide Record of work done by ***Abhishek Kumar*** and ***Pradum Kumar*** in partial fulfillment of requirements of Class XII during the year 2016-17

**Subject Teacher Signature Signature of the Principal**

**Signature of Director**

**Valued-by:**

**Name:**

1. **Register Number :**

**Examination Centre:**

**2) Date of Exam:**

**ACKNOWLEDGEMENT**

We would like to extend our sincere gratitude to our respected Director **Mrs.Geetha Basavaraj** and Principal **Mr.J.B.Murthy** for his valued motivation and assistance and for providing us this platform to complete our higher secondary education successfully.

We would like to thank our computer science facilitator, **Mrs. Divya C** for all her effort motivation and support for enabling this project successfully.

We would like to thank our parents, teachers and friends for their everlasting support and guidance.

We would also like to extend our gratitude to the management of BGSIRS for providing us with an excellent infrastructure and faculty.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SERIAL**  **NO.** | **CONTENTS** | **PAGE NO**. |
| 1 | Overview of C++ | 5 |
| 2 | Abstract | 6 |
| 3 | Requirements | 7 |
| 4 | Source code | 8-32 |
| 5 | Shortcomings | 33 |
| 6 | Output | 34-38 |
| 7 | Header files used | 39 |
| 7 | Bibliography | 40 |

**OVERVIEW OF C++**

C++ is a statically typed complied, general – purpose, case-sensitive, free-form programming language that supports procedural, object- oriented, and generic programming.

C++ is regarded as a middle-level language, as it comprises a combination of both High- level and low – level language features.

C++ was developed by **Bjarne Stroustrup** starting in 1979 at Bell Labs in Murray Hill, New Jersey, as an enhancement to the C language and originally named C with Classes but later it was renamed C++ in 1983.

C++ is a superset of C, and that virtually and legal C program is a legal C++ program.

1. Stronger type checking

2. Support for data abstraction

3. Support for object-oriented programming

4. Support for generic programming.

**ABSTRACTION**

Chess is a game played by two players. This game requires strategy, patience and high thinking capacity. In our project we have created a chess board and the suitable pieces in C++ graphics. We have also given different logics for the movement of the chess pieces. This version of chess game requires two players, both humans.

This program includes a main menu from which all parts of the game can be accessed. This game is played using the co-ordinates given on screen along the chess board. The game starts with player 1(PL1) which is white and then player 2 (PL2) which is black. Improvised version will be coming soon.

**REQUIREMENTS**

* 1. **SYSTEM REQUIREMENTS**
* Intel Core i3 , 16 bit operating system
* Microsoft Windows Server 2003 (SP1) Microsoft Windows 7

Windows 2000 Professional (SP4) Windows 2000 Server (SP4)

* 2.00 GB RAM
* CD-ROM or DVD-ROM drive
* SVGA or higher-resolution monitors (XGA recommended)
  1. **SOFTWARE REQUIREMENTS**
* Turbo C++ 3.0/C-free-3.0/Code::bocks or any other compiler capable of working with C++ programming language.

**SOURCE CODE**

**THE MAIN FUNCTION OF A PROGRAM**

**#include<iostream.h>**

**#include<conio.h>**

**#include<fstream.h>**

**#include<graphics.h>**

**#include<stdio.h>**

**#include<stdlib.h>**

**#define Exit 27**

**#define Enter 13**

**int w,l,u,v,p,q,x,k,c=0,i,j,con,gt,go,the=4,found=0;**

**char a[8][8],b[8][8],f[2],t[2],ch,save;**

**class brd**

**{ void pawn(int x,int y,int cl)**

**{w=(y\*60)+95; l=(x\*50)+60;**

**line(w,l,w+30,l); line(w+5,l-5,w+25,l-5);**

**arc(w-38,l-33,328,356,50); arc(w+68,l-33,185,214,50);**

**setcolor(8); circle(w+15,l-33,6);**

**setfillstyle(SOLID\_FILL,cl); floodfill(w+15,l-33,8);**

**bar(w,l-4,w+30,l-1); rectangle(w,l,w+30,l-5);}**

**void rook(int x,int y,int cl)**

**{ w=(y\*60)+95; l=(x\*50)+60;**

**line(w,l,w+30,l); line(w+5,l-5,w+25,l-5);**

**line(w+5,l-40,w+8,l-37); line(w+8,l-37,w+8,l-32);**

**line(w+8,l-32,w+12,l-32); line(w+12,l-32,w+12,l-37);**

**line(w+12,l-37,w+18,l-37); line(w+18,l-37,w+18,l-32);**

**line(w+18,l-32,w+22,l-32); line(w+22,l-32,w+22,l-37);**

**line(w+25,l-40,w+22,l-37); line(w+5,l-40,w+5,l-24);**

**line(w+25,l-40,w+25,l-24); line(w+5,l-24,w+25,l-24);**

**arc(w-42,l-23,340,0,50); arc(w+72,l-23,180,200,50);**

**arc(w+5,l,90,180,5); arc(w+25,l,0,90,5);**

**setfillstyle(SOLID\_FILL,cl); bar(w,l-4,w+30,l-1);**

**bar(w+6,l-31,w+24,l-25); rectangle(w,l,w+30,l-5);}**

**void bishop(int x,int y,int cl)**

**{ int w=(y\*60)+95; int l=(x\*50)+60;**

**line(w,l,w+30,l); line(w+5,l-5,w+25,l-5);**

**arc(w-38,l-33,328,356,50);**

**arc(w+68,l-33,185,214,50); arc(w+15,l-33,135,237,6);**

**arc(w+15,l-33,315,91,6); setcolor(8);**

**circle(w+15,l-33,6); setfillstyle(SOLID\_FILL,cl);**

**floodfill(w+15,l-33,8); bar(w,l-4,w+30,l-1);**

**rectangle(w,l,w+30,l-5); setcolor(15);**

**line(w+10,l-37,w+15,l-32); line(w+12,l-38,w+18,l-32);**

**setcolor(8);}**

**void king(int x,int y,int cl)**

**{ int w=(y\*60)+95; int l=(x\*50)+60; line(w,l,w+30,l);**

**line(w+5,l-5,w+25,l-5); line(w+14,l-30,w+14,l-32);**

**line(w+8,l-29,w+22,l-29); line(w+8,l-29,w+6,l-27);**

**line(w+22,l-29,w+24,l-27); line(w+16,l-30,w+16,l-32);**

**line(w+14,l-32,w+10,l-32); line(w+16,l-32,w+20,l-32);**

**line(w+16,l-35,w+20,l-35); line(w+14,l-35,w+10,l-35);**

**line(w+10,l-32,w+10,l-35); line(w+20,l-32,w+20,l-35);**

**line(w+14,l-35,w+14,l-42); line(w+14,l-42,w+16,l-42);**

**line(w+16,l-42,w+16,l-35); rectangle(w+6,l-26,w+24,l-24); arc(w-42,l-23,340,0,50);**

**arc(w+72,l-23,180,200,50); arc(w+5,l,90,180,5); arc(w+25,l,0,90,5);**

**setfillstyle(SOLID\_FILL,cl); bar(w,l-4,w+30,l-1); bar(w+11,l-34,w+19,l-33);**

**bar(w+15,l-41,w+15,l-30); bar(w+7,l-25,w+23,l-25); rectangle(w,l,w+30,l-5);}**

**void queen(int x,int y,int cl)**

**{ int w=(y\*60)+95; int l=(x\*50)+60;**

**line(w+8,l-29,w+22,l-29); line(w+8,l-29,w+6,l-27); line(w+22,l-29,w+24,l-27);**

**line(w+8,l-29,w+4,l-38); line(w+4,l-38,w+12,l-32); line(w+12,l-32,w+15,l-42);**

**line(w+15,l-42,w+18,l-32); line(w+18,l-32,w+26,l-38); line(w+26,l-38,w+22,l-29);**

**rectangle(w+6,l-26,w+24,l-24); arc(w-42,l-23,340,0,50); arc(w+72,l-23,180,200,50);**

**arc(w+5,l,90,180,5); arc(w+25,l,0,90,5); setfillstyle(SOLID\_FILL,cl);**

**bar(w,l-4,w+30,l-1); rectangle(w,l,w+30,l-5); bar(w+7,l-25,w+23,l-25);}**

**void horse(int x,int y,int cl)**

**{ int w=(y\*60)+95; int l=(x\*50)+60;**

**line(w,l,w+30,l); line(w+5,l-5,w+25,l-5);**

**arc(w+5,l,90,180,5); arc(w+25,l,0,90,5); arc(w-38,l-33,328,349,50);**

**arc(w+8,l-15,335,85,20); arc(w+15,l-16,100,170,20); arc(w-2,l-18,150,350,3);**

**arc(w-5,l-35,290,320,20); setcolor(8); circle(w+5,l-28,1);**

**setfillstyle(1,cl); floodfill(w+5,l-28,8);**

**arc(w-10,l-33,5,20,20); arc(w-5,l-33,5,25,20); arc(w+25,l-30,145,165,20);**

**arc(w+30,l-32,150,160,20); arc(w-15,l-30,320,327,20);**

**setfillstyle(SOLID\_FILL,cl); bar(w,l-4,w+30,l-1); rectangle(w,l,w+30,l-5);}**

**public:**

**void chess\_board()**

**{**

**//------------------------------CHESS BOARD----------------------------------**

**cleardevice();**

**setbkcolor(the);**

**setfillstyle(SOLID\_FILL,6);**

**bar(80,15,140,65); bar(200,15,260,65); bar(320,15,380,65);**

**bar(440,15,500,65); bar(80,115,140,165); bar(80,215,140,265);**

**bar(80,315,140,365); bar(140,65,200,115); bar(140,165,200,215);**

**bar(140,265,200,315); bar(140,365,200,415); bar(200,115,260,165);**

**bar(200,215,260,265); bar(200,315,260,365); bar(260,65,320,115);**

**bar(260,165,320,215); bar(260,265,320,315); bar(260,365,320,415);**

**bar(320,115,380,165); bar(320,215,380,265); bar(320,315,380,365);**

**bar(380,65,440,115); bar(380,165,440,215); bar(380,265,440,315);**

**bar(380,365,440,415); bar(440,115,500,165); bar(440,215,500,265);**

**bar(440,315,500,365); bar(500,65,560,115); bar(500,165,560,215);**

**bar(500,265,560,315); bar(500,365,560,415); setfillstyle(SOLID\_FILL,15);**

**bar(80,65,140,115); bar(80,165,140,215); bar(80,265,140,315);**

**bar(80,365,140,415); bar(140,15,200,65); bar(140,115,200,165);**

**bar(140,215,200,265); bar(140,315,200,365); bar(200,65,260,115);**

**bar(200,165,260,215); bar(200,265,260,315);**

**bar(200,365,260,415); bar(260,15,320,65); bar(260,115,320,165);**

**bar(260,215,320,265); bar(260,315,320,365);**

**bar(380,15,440,65); bar(380,115,440,165);**

**bar(380,215,440,265); bar(380,315,440,365);**

**bar(500,15,560,65); bar(500,115,560,165); bar(500,215,560,265);**

**bar(500,315,560,365); bar(320,65,380,115); bar(320,165,380,215);**

**bar(320,265,380,315); bar(320,365,380,415); bar(440,65,500,115);**

**bar(440,165,500,215); bar(440,265,500,315); bar(440,365,500,415);**

**setcolor(8); rectangle(80,15,560,415); line(80,65,560,65);**

**line(80,115,560,115); line(80,165,560,165); line(80,215,560,215);**

**line(80,265,560,265); line(80,315,560,315); line(80,365,560,365);**

**line(140,15,140,415); line(200,15,200,415); line(260,15,260,415);**

**line(320,15,320,415); line(380,15,380,415); line(440,15,440,415);**

**line(500,15,500,415); setcolor(15); settextstyle(2,0,4);**

**outtextxy(108,2,"A"); outtextxy(168,2,"B"); outtextxy(228,2,"C");**

**outtextxy(288,2,"D"); outtextxy(348,2,"E"); outtextxy(408,2,"F");**

**outtextxy(468,2,"G"); outtextxy(528,2,"H"); outtextxy(72,35,"1");**

**outtextxy(72,85,"2"); outtextxy(72,135,"3"); outtextxy(72,185,"4");**

**outtextxy(72,235,"5"); outtextxy(72,285,"6"); outtextxy(72,335,"7");**

**outtextxy(72,385,"8"); settextstyle(14,0,1); setcolor(14);**

**outtextxy(234,460," Press Esc to EXIT"); setcolor(8);**

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

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

**{if(a[i][j]=='R')**

**rook(i,j,b[i][j]);**

**else if(a[i][j]=='H')**

**horse(i,j,b[i][j]);**

**else if(a[i][j]=='B')**

**bishop(i,j,b[i][j]);**

**else if(a[i][j]=='K')**

**king(i,j,b[i][j]);**

**else if(a[i][j]=='Q')**

**queen(i,j,b[i][j]);**

**else if(a[i][j]=='P')**

**pawn(i,j,b[i][j]);}} }**

**//----------------------------CHESS BOARD ENDS-------------------------------**

**}ob;**

**class Newg**

**{ public:**

**char p; int col; int i,j;}obj,obj1;**

**class scr**

**{void exd()**

**{setcolor(14); setfillstyle(1,12); bar(200,200,430,300);**

**rectangle(200,200,430,300); rectangle(205,205,425,295); settextstyle(7,0,1);**

**outtextxy(250,220,"Are you sure"); outtextxy(260,250,"YES NO");**

**ch=getch();**

**if(ch=='y'||ch=='Y')**

**{closegraph();**

**exit(1);}**

**else**

**mainmenu(); }**

**void clrch()**

**{ cleardevice(); setbkcolor(the); setcolor(YELLOW);**

**line(50,850,50,0); line(54,850,54,0); line(580,850,580,0);**

**line(584,850,584,0); line(0,50,850,50); line(0,54,854,54);**

**line(0,425,800,425); line(0,429,800,429); settextstyle(7,0,4);**

**setcolor(WHITE); outtextxy(130,160," THEME ");**

**setcolor(YELLOW); outtextxy(300,160," 1");**

**setcolor(WHITE); outtextxy(130,240," THEME");**

**setcolor(YELLOW); outtextxy(300,240," 2");**

**settextstyle(14,0,1); setcolor(14);**

**outtextxy(238,450," Press Esc to go Back"); ch=getch();**

**if(ch=='1')**

**{the=4;**

**clrch();}**

**else if(ch=='2')**

**{ the=0;**

**clrch();}**

**else if(ch==Exit)**

**optscr();**

**else**

**clrch();}**

**void help()**

**{cleardevice(); setcolor(14); line(50,850,50,0);**

**line(54,850,54,0); line(580,850,580,0); line(584,850,584,0);**

**line(0,50,850,50); line(0,54,854,54); line(0,425,800,425);**

**line(0,429,800,429); settextstyle(4,0,4); setcolor(WHITE);**

**outtextxy(190,55," HOW TO PLAY"); line(440,90,190,90);**

**settextstyle(14,0,2); setcolor(YELLOW);**

**outtextxy(80,130,"The Chess Game is played between 2 players with 8 pawns,");**

**outtextxy(80,150,"2 bishops, 2 rooks, 2 horses, 1 King and 1 Queen in each");**

**outtextxy(80,170,"teams.The player should enter the coordinates of the current");**

**outtextxy(80,190,"position and the coordinates of position where the player ");**

**outtextxy(80,210,"wants to move the coin.White player goes first then players ");**

**outtextxy(80,230,"alternate turns.");**

**settextstyle(14,0,1); setcolor(14);**

**outtextxy(238,450," Press Esc to go Back"); ch=getch();**

**if(ch==Exit)**

**{ optscr();**

**}**

**else**

**help();**

**}**

**void optscr()**

**{cleardevice();**

**settextstyle(4,0,5); setbkcolor(the); setcolor(14);**

**outtextxy(173,170," C"); setcolor(15); outtextxy(200,170," hange Colour");**

**setcolor(14); outtextxy(269,230," H");**

**setcolor(15); outtextxy(300,230," elp"); setcolor(14);**

**line(50,850,50,0); line(54,850,54,0); line(580,850,580,0);**

**line(584,850,584,0); line(0,50,850,50); line(0,54,854,54);**

**line(0,425,800,425); line(0,429,800,429); settextstyle(14,0,1);**

**setcolor(14); outtextxy(238,450," Press Esc to go Back");**

**ch=getch();**

**if(ch=='c'||ch=='C')**

**{ clrch();}**

**else if(ch=='h'||ch=='H')**

**{ help();**

**}**

**else if(ch==Exit)**

**mainmenu();**

**else**

**optscr();**

**}**

**void newgame()**

**{**

**ifstream f("NEW.dat",ios::in|ios::binary);**

**f.seekg(ios::beg);**

**while(!f.eof())**

**{**

**f.read((char\*)&obj,sizeof(obj));**

**a[obj.i][obj.j]=obj.p;**

**b[obj.i][obj.j]=obj.col;**

**}**

**f.close();**

**}**

**void contgame()**

**{**

**fstream fc("Saved\_Game",ios::in|ios::binary);**

**fc.seekg(0);**

**while(!fc.eof())**

**{**

**fc.read((char\*)&obj1,sizeof(obj1));**

**a[obj1.i][obj1.j]=obj1.p;**

**b[obj1.i][obj1.j]=obj1.col;**

**}**

**fc.close();**

**}**

**public:**

**void scr1()**

**{**

**int gd=DETECT,gm;**

**initgraph(&gd,&gm,"C:\\turboc3\\bgi");**

**setcolor(YELLOW); settextstyle(2,0,10);**

**outtextxy(145,50,"WELCOME TO THE."); delay(500);**

**outtextxy(495,50,"."); delay(500);**

**outtextxy(509,50,"."); delay(400);**

**settextstyle(4,0,20); outtextxy(110,111,"CHESS");**

**settextstyle(4,0,10); outtextxy(100,230,"GAME...");**

**delay(1500);**

**}**

**void mainmenu()**

**{ cleardevice(); setbkcolor(the); setcolor(14);**

**settextstyle(4,0,9); outtextxy(170,15,"CHESS");**

**line(800,120,0,120); line(470,130,180,130);**

**line(800,37,0,37); line(470,27,180,27);**

**settextstyle(7,0,5); setcolor(14); delay(10);**

**outtextxy(140,180,"C"); setcolor(15);**

**outtextxy(165,180,"ONTINUE GAME"); delay(5); setcolor(14);**

**outtextxy(202,260,"N"); setcolor(15);**

**outtextxy(230,260,"EW GAME"); delay(5); setcolor(14);**

**outtextxy(220,340,"O"); setcolor(15);**

**outtextxy(242,340,"PTIONS"); settextstyle(14,0,1); setcolor(14);**

**outtextxy(234,450," Press Esc to EXIT");**

**gt:**

**ch=getch();**

**if(ch=='n'||ch=='N')**

**{**

**newgame();**

**ob.chess\_board();**

**}**

**else if(ch=='c'||ch=='C')**

**{**

**contgame();**

**ob.chess\_board();**

**}**

**else if(ch=='o'||ch=='O')**

**{**

**optscr();**

**}**

**else if(ch==Exit)**

**{**

**exd(); goto gt;**

**}**

**else**

**mainmenu();**

**}**

**}ob1;**

**void sel()**

**{**

**if(f[0]=='A'||f[0]=='a')**

**u=0;**

**else if(f[0]=='B'||f[0]=='b')**

**u=1;**

**else if(f[0]=='C'||f[0]=='c')**

**u=2;**

**else if(f[0]=='D'||f[0]=='d')**

**u=3;**

**else if(f[0]=='E'||f[0]=='e')**

**u=4;**

**else if(f[0]=='F'||f[0]=='f')**

**u=5;**

**else if(f[0]=='G'||f[0]=='g')**

**u=6;**

**else if(f[0]=='H'||f[0]=='h')**

**u=7;**

**if(f[1]=='1')**

**p=0;**

**else if(f[1]=='2')**

**p=1;**

**else if(f[1]=='3')**

**p=2;**

**else if(f[1]=='4')**

**p=3;**

**else if(f[1]=='5')**

**p=4;**

**else if(f[1]=='6')**

**p=5;**

**else if(f[1]=='7')**

**p=6;**

**else if(f[1]=='8')**

**p=7;**

**if(t[0]=='A'||t[0]=='a')**

**v=0;**

**else if(t[0]=='B'||t[0]=='b')**

**v=1;**

**else if(t[0]=='C'||t[0]=='c')**

**v=2;**

**else if(t[0]=='D'||t[0]=='d')**

**v=3;**

**else if(t[0]=='E'||t[0]=='e')**

**v=4;**

**else if(t[0]=='F'||t[0]=='f')**

**v=5;**

**else if(t[0]=='G'||t[0]=='g')**

**v=6;**

**else if(t[0]=='H'||t[0]=='h')**

**v=7;**

**if(t[1]=='1')**

**q=0;**

**else if(t[1]=='2')**

**q=1;**

**else if(t[1]=='3')**

**q=2;**

**else if(t[1]=='4')**

**q=3;**

**else if(t[1]=='5')**

**q=4;**

**else if(t[1]=='6')**

**q=5;**

**else if(t[1]=='7')**

**q=6;**

**else if(t[1]=='8')**

**q=7;**

**}**

**void horse()**

**{**

**if(q-p==2||p-q==2)**

**{**

**if(v-u==1||u-v==1)**

**{**

**if(a[q][v]=='\0'||b[q][v]!=b[p][u])**

**k=1;**

**else**

**k=0;**

**}**

**else**

**k=0;**

**}**

**else if(q-p==1||p-q==1)**

**{**

**if(u-v==2||v-u==2)**

**{**

**if(a[q][v]=='\0'||b[q][v]!=b[p][u])**

**k=1;**

**else**

**k=0;**

**}**

**else**

**k=0;**

**}**

**else**

**k=0;**

**}**

**void bishop()**

**{**

**if(a[q][v]=='\0'||b[q][v]!=b[p][u])**

**{**

**if(p+u==q+v)**

**{**

**if(p>q)**

**{**

**k=1;**

**for(i=p-1,j=u+1;i>q,j<v;i--,j++)**

**{**

**if(a[i][j]!='\0')**

**{**

**k=0;**

**break;**

**}**

**}**

**}**

**else if(q>p)**

**{**

**k=1;**

**for(i=p+1,j=u-1;i<q,j>v;i++,j--)**

**{**

**if(a[i][j]!='\0')**

**{**

**k=0;**

**break;**

**}**

**}**

**}**

**else**

**k=0;**

**}**

**else if(p-u==q-v)**

**{ if(p>q)**

**{ k=1; for(i=p-1,j=u-1;i>q,j>v;i--,j--)**

**{ if(a[i][j]!='\0')**

**{ k=0; break;**

**} } } else if(q>p)**

**{ k=1;**

**for(i=p+1,j=u+1;i<q,j<v;i++,j++)**

**{ if(a[i][j]!='\0')**

**{ k=0;**

**break;**

**} } } else**

**k=0;**

**} else**

**k=0;**

**} else**

**k=0;**

**}**

**void king()**

**{ if(a[q][v]=='\0'||b[q][v]!=b[p][u])**

**{ if(u==v)**

**{ if(q==p+1||q==p-1)**

**k=1;**

**else**

**k=0;**

**} else if(p==q)**

**{ if(v==u+1||v==u-1)**

**k=1;**

**else**

**k=0;**

**} else if((p-q==1)&&(u-v==1))**

**{**

**k=1;**

**} else if((q-p==1)&&(u-v==1))**

**{**

**k=1;**

**} else if((p-q==1)&&(v-u==1))**

**{**

**k=1;**

**} else if((q-p==1)&&(v-u==1))**

**{**

**k=1;**

**} else**

**k=0;**

**} else**

**k=0;**

**}**

**void rook()**

**{ if(u==v||p==q)**

**{ if((b[p][u]!=b[q][v])||(a[q][v]=='\0'))**

**{ if(u>v)**

**{ k=1; for(i=u-1;i>v;i--)**

**{ if(a[q][i]!='\0')**

**k=0;**

**} }**

**else if(v>u)**

**{ k=1;**

**for(i=u+1;i<v;i++)**

**{ if(a[q][i]!='\0')**

**k=0;**

**} }**

**else if(q>p)**

**{ k=1;**

**for(i=p+1;i<q;i++)**

**{ if(a[i][u]!='\0')**

**k=0;**

**} }**

**else if(p>q)**

**{ k=1;**

**for(i=p-1;i>q;i++)**

**{ if(a[i][u]!='\0')**

**k=0;**

**} } else**

**k=0;**

**} else**

**k=0;**

**} else**

**k=0;**

**}**

**void bpawn()**

**{ if(u==v)**

**{ if((q==4)&&(p==6))**

**{ if((a[q][v]=='\0')&&(a[q+1][v]=='\0'))**

**k=1;**

**else**

**k=0;**

**}**

**else if(p-q==1)**

**{ if(a[q][v]=='\0')**

**k=1;**

**else**

**k=0;**

**} else**

**k=0;**

**} else if(u-v==1||v-u==1)**

**{ if(p-q==1)**

**{ if(b[q][v]==7)**

**k=1;**

**} else**

**k=0;**

**} else**

**k=0;**

**} else**

**k=0;**

**}**

**void wpawn()**

**{**

**if(u==v)**

**{ if(q==3&&p==1)**

**{ if((a[q][v]=='\0')&&(a[p+1][v]=='\0'))**

**k=1;**

**else**

**k=0;**

**}**

**else if(q-p==1)**

**{**

**if(a[q][v]=='\0')**

**k=1;**

**else**

**k=0;**

**}**

**else**

**k=0;**

**}**

**else if(u-v==1||v-u==1)**

**{**

**if(q-p==1)**

**{**

**if(b[q][v]==8)**

**k=1;**

**else**

**k=0;**

**}**

**else**

**k=0;**

**}**

**else**

**k=0;**

**}**

**void move();**

**void rep()**

**{**

**if(k==1)**

**{**

**a[q][v]=a[p][u]; b[q][v]=b[p][u]; a[p][u]=b[p][u]='\0';**

**c++;**

**}**

**else**

**{**

**ob.chess\_board();**

**settextstyle(14,0,1);**

**setcolor(14);**

**outtextxy(70,445,"Enter a valid move");**

**move();**

**}**

**}**

**void move()**

**{**

**ofstream fout("Saved\_Game",ios::out|ios::app|ios::binary);**

**con=getch();**

**while(con!=Exit)**

**{**

**k=0;**

**f[0]=t[0]=f[1]=t[1]='\0';**

**setcolor(14);**

**settextstyle(17,0,1);**

**if(c%2==0)**

**outtextxy(70,430,"PL1");**

**else**

**outtextxy(70,430,"PL2");**

**outtextxy(100,430,"Enter your move : From :");**

**for(i=0;i<2;i++)**

**{**

**f[i]=getch();**

**outtextxy(300,430,f);**

**}**

**outtextxy(330,430,"To:");**

**for(i=0;i<2;i++)**

**{**

**t[i]=getch();**

**outtextxy(360,430,t);**

**}**

**if(c%2==0)**

**{**

**sel();**

**if(b[p][u]==7)**

**{**

**if(a[p][u]=='P')**

**wpawn();**

**else if(a[p][u]=='R')**

**rook();**

**else if(a[p][u]=='H')**

**horse();**

**else if(a[p][u]=='B')**

**bishop();**

**else if(a[p][u]=='K')**

**king();**

**else if(a[p][u]=='Q')**

**{**

**rook();**

**if(k==0)**

**bishop();**

**}**

**}**

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

**{**

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

**{**

**if((a[i][j]=='K')&&(b[i][j]==8))**

**found=1;**

**}**

**}**

**if(found==0)**

**{setcolor(14); setfillstyle(1,12);**

**bar(200,200,430,300); rectangle(200,200,430,300); rectangle(205,205,425,295);**

**settextstyle(7,0,1); outtextxy(240,220,"Congratulations");**

**outtextxy(280,260,"PL1 Won"); getch(); ob1.mainmenu();**

**} }**

**else**

**{ sel();**

**if(b[p][u]==8)**

**{if(a[p][u]=='P')**

**bpawn();**

**else if(a[p][u]=='R')**

**rook();**

**else if(a[p][u]=='H')**

**horse();**

**else if(a[p][u]=='B')**

**bishop();**

**else if(a[p][u]=='K')**

**king();**

**else if(a[p][u]=='Q')**

**{rook();**

**if(k==0)**

**bishop();**

**} }**

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

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

**{ if((a[i][j]=='K')&&(b[i][j]==7))**

**found=1;**

**} } if(found==0)**

**{ setcolor(14); setfillstyle(1,12);**

**bar(200,200,430,300); rectangle(200,200,430,300); rectangle(205,205,425,295);**

**settextstyle(7,0,1); outtextxy(240,220,"Congratulations"); outtextxy(280,260,"PL2 Won");**

**getch(); ob1.mainmenu();**

**}} rep();ob.chess\_board();**

**}**

**if(con==Exit)**

**{ go:**

**setcolor(14);setfillstyle(1,12); bar(200,200,430,300);**

**rectangle(200,200,430,300);rectangle(205,205,425,295);**

**settextstyle(7,0,1);**

**outtextxy(215,220,"Do you want to Save"); outtextxy(260,250,"YES NO"); save=getch();**

**if(save=='y'||save=='Y')**

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

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

**{obj1.p=a[i][j];obj1.col=b[i][j];obj1.i=i;obj1.j=j;**

**fout.write((char\*)&obj1,sizeof(obj1));**

**} } }**

**else if(save=='n'||save=='N')**

**ob1.mainmenu();**

**else**

**goto go;**

**}}**

**void main()**

**{**

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

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

**a[i][j]=b[i][j]='\0';**

**} ob1.scr1();**

**ob1.mainmenu();**

**move();**

**closegraph();**

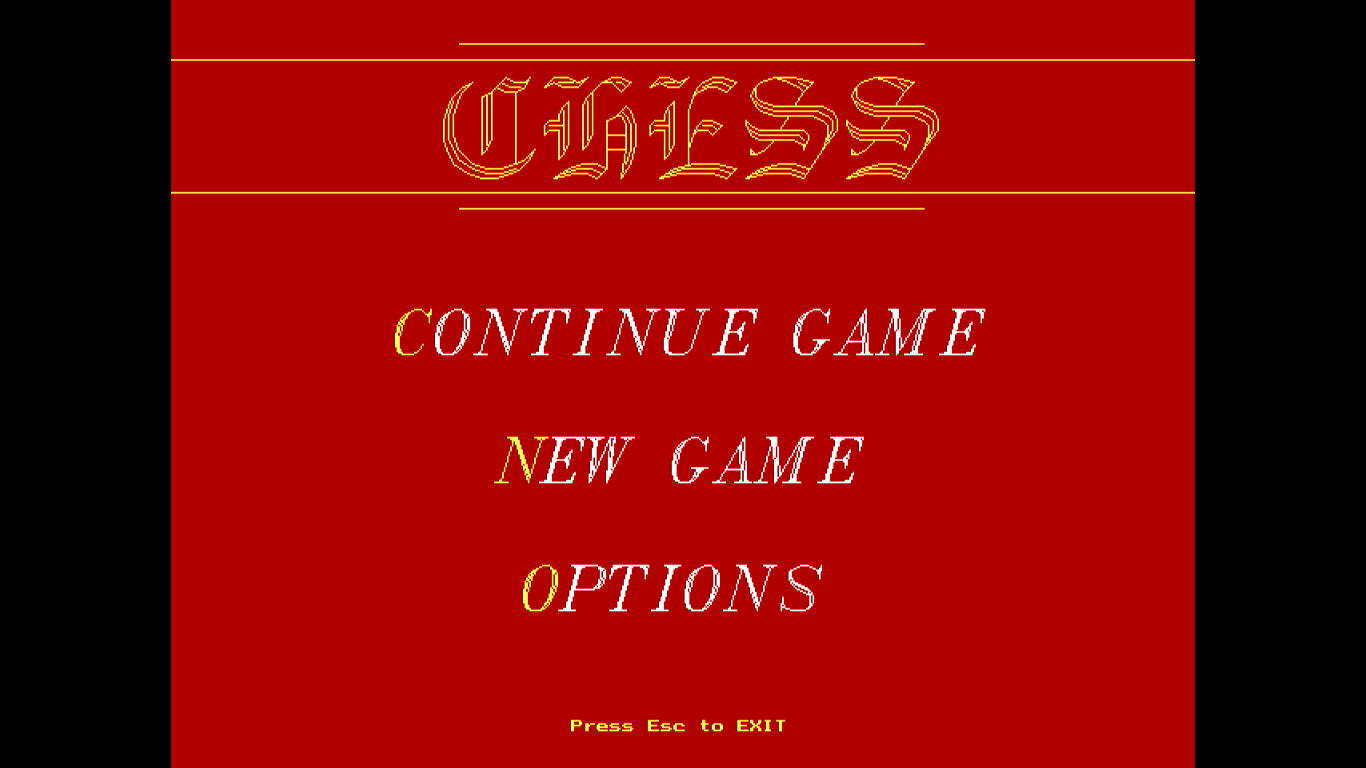
**}**

**SHORT COMINGS**

* Computer v/s Human game play.
* 3-D Chess coins.
* Mouse Controls.
* Check and Checkmate Shown by the program
* Castling
* En Passant
* Pawn Promotion

**OUTPUT**



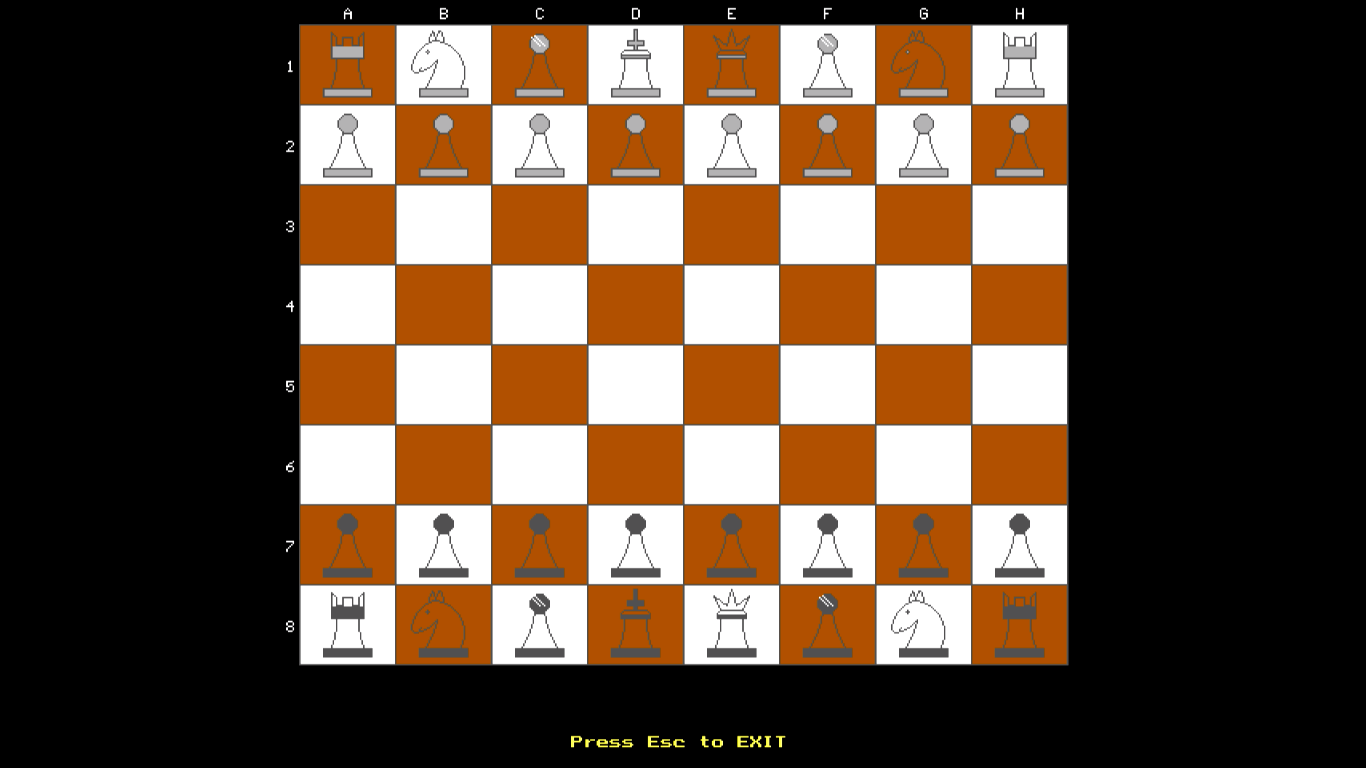


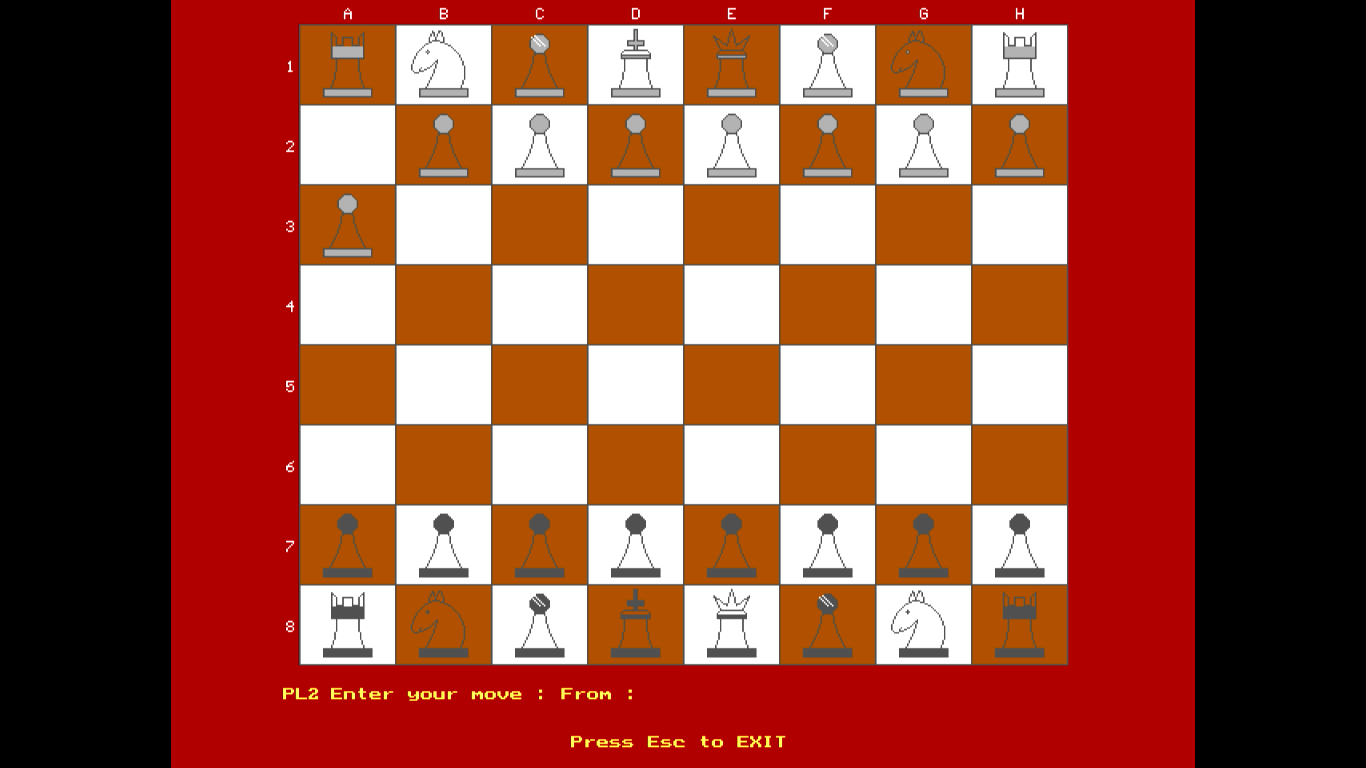


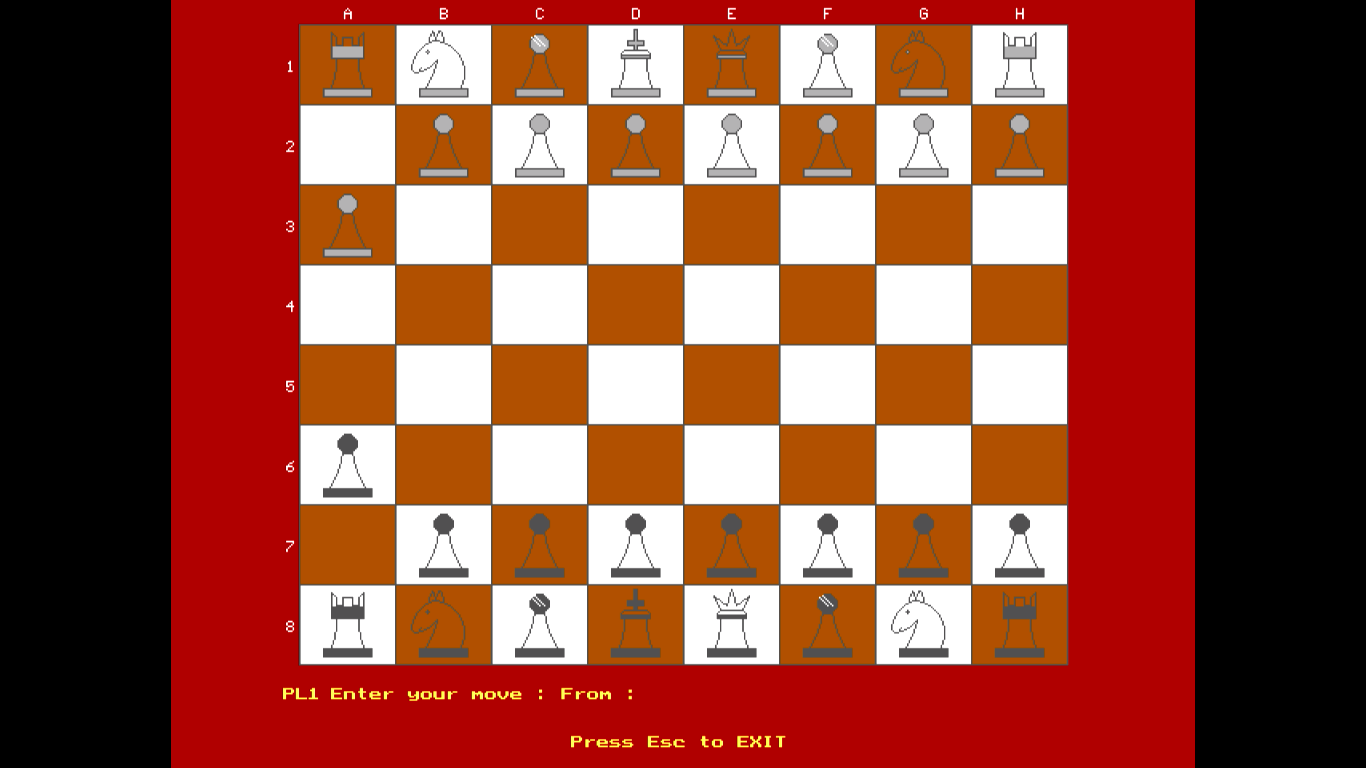














**HEADER FILES USED**

* **#include<iostream.h>**
* **#include<conio.h>**
* **#include<fstream.h>**
* **#include<graphics.h>**
* **#include<stdio.h>**
* **#include<stdlib.h>**

**BIBLIOGRAPHY**

* [www.google.com](http://www.google.com)
* [www.gradestack.com](http://www.gradestack.com)
* [www.youtube.com](http://www.youtube.com)
* [www.thecrazyprogrammer.com](http://www.thecrazyprogrammer.com)

**BOOKS REFFERED**

* SUMITHA ARORA, COMPUTER SCIENCE for class 12th WITH C++, 8TH EDITION