**HULIZ**

**Project done by**

***N. Mohamed Riaz***

***Anshumaan Shankar***

***M.Badri Narayanan***

***S.Rahul***

***S.Anoosh***

**CONTENTS**

* Acknowledgment
* Salient Features
* Project Analysis
* Program Listing
* Sample Output

**Acknowledgment**

We would like to thank our Computer Science teachers Mrs. Harifa Begum and Mrs. Nirmala Devi for guiding us in making this program. Their valuable inputs and timely help have enabled us to make this program efficient and user friendly.

**SALIENT FEATURES**

Topics covered in this project are :

* Othello Game
* Ticketing System For Movies
* Correction Factor For Defective Eyes
* Nuclear Fusion Reaction In The Sun

**PROJECT ANALYSIS**

***OTHELLO*** is a popular board game played between two players - one player choosing the BLUE colour and the other playing choosing the RED colour. The game is played an 8 x 8 square grid.

***Playing the Game :***

Two BLUE and RED coins are placed on the central four squares in the grid. The player who chose BLUE starts the game. He should place another BLUE coin in a new square. His positioning of the coin is valid if there is at least one of the opponent's coins next to his coin in at least one of the eight directions available and if there is one of his own coins placed behind the opponent’s coin. If the player's positioning is valid then all his opponent's coins that are placed in the above manner will be converted into his coins. Once this is done, the turn shifts to the opponent who makes the moves using his coin.

***End of Game :***

The Game is over in the following cases :

* When there are no coins belonging to the opponent on the grid, then the player who has all his coins on the grid wins.
* When every square in the grid is filled with coins, then the player with greater number of coins in the grid wins.

**The Main Logic :**

We have applied a simple logic of searching in 8 directions to create the game. When any player clicks in a place in the grid, we analyze all 8 squares around the 'clicked' square. If there is a coin of the opposite colour in that position, then we run the search in that direction. The compiler searches in that direction till a coin of the same colour is reached, without any empty square in between. Then all opposite colour coins in between are converted to coins of the same colour. This method is carried out for all 8 directions. Now, if the person clicks at a correct square, then he will convert some opposite colour coins. Based on this idea, we put the condition that if no coin of the opposite colour is around or there is no coin of the same colour after opposite colour coins in a series, then no move will be possible and the computer flashes error statement making the player to attempt again. Once a valid move is made, the chance shifts to the opponent to play in a similar way. Each time a particular player changes the coins his score is incremented with the number of coins he had done and the opponents coins which are changed are decremented from the opponent's score. This game continues till all the positions in the grid are filled with coins. Then we count the number of coins belonging to each player and announce the winner. If however, no move is available for a player, then the game ends. Here we have given the following options to user.

**New Game :**

This option allows them to start a new Game. Game asks the name of the

two users and set their score to 0. Here, the user can save the game after and continue later.

**Load Game :**

This allows the user to input the file name of already saved game and continue from that position onwards.

**Print Rules :**

This prints the basic rules of the Game.

**TICKET BOOKING**

The main outcome of the program is to simulate a website in order to facilitate the booking of movie tickets. The following program contains the following parts

1. The layout of the website where the ticket booking takes place. This part of the program provides the outlook as to the design of the website and is the first screen encountered by the user upon running the program.
2. The second screen is a transitionary one that gives an appearance as though the next screen is loaded, for the time period that the second screen is displayed.
3. This screen shows us a detailed layout of the cinema hall with the title of the movie, the time of screening, number of seats in the hall(unavailable, available and selected), the direction in which the screen is present, number of tickets booked etc.
4. The next step is seen in the same screen, with the actual selection of tickets. The available tickets, when clicked upon turn blue in color, highlighting the user’s selection. Once booked, a confirm button which when clicked confirms the selections made by the user. In case the user wishes to go back to the previous screen, the back button can be used to do so.
5. Once the booking has been confirmed, the final screen thanking the user for using the website is displayed.

***Proton–Proton Chain Reaction***

In [nuclear physics](https://en.wikipedia.org/wiki/Nuclear_physics), nuclear fusion is a [reaction](https://en.wikipedia.org/wiki/Nuclear_reaction) in which two or more [atomic nuclei](https://en.wikipedia.org/wiki/Atomic_nucleus) come close enough to form one or more different atomic nuclei and subatomic particles (neutrons or protons).

The proton–proton chain reaction is one of the two (known) sets of [fusion](https://en.wikipedia.org/wiki/Nuclear_fusion) [reactions](https://en.wikipedia.org/wiki/Nuclear_reaction) by which [stars](https://en.wikipedia.org/wiki/Star) convert [hydrogen](https://en.wikipedia.org/wiki/Hydrogen) to [helium](https://en.wikipedia.org/wiki/Helium). It dominates in stars the size of the [Sun](https://en.wikipedia.org/wiki/Sun) or smaller. This code explains the step wise reaction that happens in this chain reaction

**Step 1 :**

The first step involves the fusion of two 1 H  nuclei ([protons](https://en.wikipedia.org/wiki/Proton)) into [deuterium](https://en.wikipedia.org/wiki/Deuterium), releasing a [positron](https://en.wikipedia.org/wiki/Positron) and a [neutrino](https://en.wikipedia.org/wiki/Neutrino) as one proton changes into a [neutron](https://en.wikipedia.org/wiki/Neutron). It is a two-stage process; first, two protons fuse to form a [diproton](https://en.wikipedia.org/wiki/Diproton" \o "Diproton) e+ = Positron [The positron or antielectron is the [antiparticle](https://en.wikipedia.org/wiki/Antiparticle) of the [electron](https://en.wikipedia.org/wiki/Electron). The positron has an [electric charge](https://en.wikipedia.org/wiki/Electric_charge) of +1 [*e*](https://en.wikipedia.org/wiki/Elementary_charge), a [spin](https://en.wikipedia.org/wiki/Spin_(physics)) of 1/2 (same as electron), and has the same [mass as an electron](https://en.wikipedia.org/wiki/Electron_rest_mass)].

v= Neutrino [A neutrino is a [fermion](https://en.wikipedia.org/wiki/Fermion" \o "Fermion) (an [elementary particle](https://en.wikipedia.org/wiki/Elementary_particle) with [half-integer spin](https://en.wikipedia.org/wiki/Spin-1/2)) that interacts only via the [weak subatomic force](https://en.wikipedia.org/wiki/Weak_interaction) and [gravity](https://en.wikipedia.org/wiki/Gravity) ]

http://burro.cwru.edu/academics/Astr221/StarPhys/pp1.gif

**Step 2 :**

The deuterium produced in the first stage can fuse with another proton to produce the light [isotope](https://en.wikipedia.org/wiki/Isotope) of [helium](https://en.wikipedia.org/wiki/Helium), [3](https://en.wikipedia.org/wiki/Helium-3" \o "Helium-3) [He](https://en.wikipedia.org/wiki/Helium-3" \o "Helium-3)

γ = Gamma Rays [Gamma rays are penetrating [electromagnetic radiation](https://en.wikipedia.org/wiki/Electromagnetic_radiation) of a kind arising from the [radioactive decay](https://en.wikipedia.org/wiki/Radioactive_decay) of [atomic nuclei](https://en.wikipedia.org/wiki/Atomic_nucleus). It consists of [photons](https://en.wikipedia.org/wiki/Photon) in the highest observed range of [photon energy](https://en.wikipedia.org/wiki/Photon_energy)]

http://burro.cwru.edu/academics/Astr221/StarPhys/pp2.gif

**Step 3 :**

Two  3 He atoms from step two undergo fusion to give a 4 He atom and two

1 H atoms

http://burro.cwru.edu/academics/Astr221/StarPhys/pp3.gif

***LENS***

A corrective lens is a lens typically worn in front of the eye to improve vision. The most common use is to treat refractive errors: myopia, hypermetropia, astigmatism, and presbyopia. Glasses or "spectacles" are worn on the face a short distance in front of the eye. Contact lenses are worn directly on the surface of the eye. Intraocular lenses are surgically implanted most commonly after cataract removal, but can be used for purely refractive purposes.  
**MYOPIA**

Near-sightedness, also known as short-sightedness and **myopia**, is a condition of the eye where light focuses in front of, instead of on, the retina. This causes distant objects to be blurry while close objects appear normal. Other symptoms may include headaches and eye strain.

**HYPERMETROPIA**

Far-sightedness, also known as hyperopia, is a condition of the eye in which light is focused behind, instead of on, the retina. This results in close objects appearing blurry, while far objects may appear normal. As the condition worsens, objects at all distances may be blurry. Other symptoms may include headaches and eye .

***ASTIGMATISM***

**Astigmatism** is a common vision condition that causes blurred vision. It occurs when the cornea (the clear front cover of the eye) is irregularly shaped or sometimes because of the curvature of the lens inside the eye.

***PRESBYOPIA***

**Presbyopia** is a natural part of the aging process. It is due to hardening of the lens of the eye causing the eye to focus light behind rather than on the retina when looking at close objects. It is a type of refractive error along with nearsightedness, farsightedness, and astigmatism

**PROJECT LISTING**

**TICKET.CPP**

#include<iostream.h>

#include<dos.h>

#include<conio.h>

#include<string.h>

#include<fstream.h>

#include<graphics.h>

#include<stdlib.h>

#include"mouse-1.cpp"

#include"fusion.cpp"

#include"othello.cpp"

#include"eye-lens.cpp"

struct tick

{

int x,y, pos,col;

char row;

long ph\_no;

}t\_ob[9][24]= {0};

int seat\_N=0;

void box(int ltx,int lty,int rbx,int rby,int colr)

{

setcolor(colr);

line(ltx,lty,rbx,lty);//-------

line(rbx,lty,rbx,rby);//right line

line(rbx,rby,ltx,rby);//-------

line(ltx,rby,ltx,lty);//left line

}

void layoutdetails()

{

void seat\_print(int,int,int);

setfillstyle(1,9);

setcolor(15);

seat\_print(90,95,8);

outtextxy(105,95,"Available Seats");

seat\_print(270,95,BLUE);

outtextxy(290,95,"Selected Seats");

//\*\*\*screen

setcolor(10);

ellipse(330,420,185,357,250,30);

outtextxy(262,430,"SCREEN THIS WAY");

setlinestyle(0,0,3);

for(int arrow=230;arrow<430;arrow+=16)

{

line(arrow,425,arrow,436);

putpixel(arrow,437,10);

putpixel(arrow,439,10);

if(arrow==246)arrow+=136;

}

setlinestyle(0,0,1);

outtextxy(227,432,"v v");

outtextxy(395,432,"v v");

//\*\*\*confirm button

setfillstyle(1,11);

bar(560,447,630,467);

setcolor(9);

outtextxy(568,453,"CONFIRM");

//\*\*\*\*\*\*\*\* layout over

}

void websiteoutline()

{

// \*\*\*\* Website outline

setcolor(8);

//setrgbpalette(12,0,250,50);

setfillstyle(1,CYAN);

bar(0,0,getmaxx(),getmaxy());//setbkcolor(3);

setlinestyle(0,0,3);

rectangle(0,0,639,28);

setfillstyle(1,8);

floodfill(5,5,1);

setfillstyle(1,15);

bar(70,3,600,25); //bg-wh , unav-red , avail-grey , sele-bl

//\*\*\*lhs buttons

setcolor(3);

outtextxy(10,10,"<");

//outtextxy(42,10,">");

setlinestyle(0,0,2);

line(10,13,20,13);

//line(36,13,46,13);

setlinestyle(0,0,2);

box(6,5,24,21,15);

//box(32,5,50,21,15);

//\*\*close button

setcolor(3);

setlinestyle(0,0,1);

line(615,8,625,20);

line(615,20,625,8);

setlinestyle(0,0,2);

box(610,5,630,23,15);

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

}

void loadscreen(int numofload=4,int redirect=0)

{

cleardevice();

setcolor(8);

setlinestyle(0,0,3);

delay(150);

if(redirect==0)outtextxy(355,240,"Loading");

else outtextxy(355,240,"Redirecting");

delay(300);

arc(320,240,5,85,25);

delay(300);

settextstyle(0,0,1);

outtextxy(280,280,"Please Wait");

arc(320,240,277,355,25);

delay(300);

switch(numofload)

{

case 4: arc(320,240,187,263,25);

delay(300);

arc(320,240,95,173,25);

delay(300);

break;

case 3: arc(320,240,187,263,25);

delay(300);

}

setlinestyle(0,0,1);

delay(1000);

cleardevice();

}

void create\_seats()

{

char k = 'A';

int m, n = 140;

for (int i = 0; i <9; i++, n+= 25)

{

m = 60;

for (int j = 0; j < 24; j++,m+= 20)

{

t\_ob[i][j].x = m;

t\_ob[i][j].y= n;

t\_ob[i][j].pos=0;

t\_ob[i][j].col = 8;

t\_ob[i][j].row=k;

t\_ob[i][j].ph\_no=0;

if (j==6||j==18) m= m+13;

}

k++;

}

}

void seat\_print(int st, int rows, int colr)

{

int end = st+13;

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

for(int j=0;j<14;j++)

putpixel(st+j,rows+i,colr);

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

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

putpixel(st+3+j,rows-5+i,colr);

for(i=0;i<3;i++)

for(j=0;j<14;j++)

putpixel(st+j,rows-8+i,colr);

for(i=0;i<3;i++)

for(j=0;j<3;j++)

{

putpixel(st+j,rows-11+i,colr);

putpixel(end-j,rows-11+i,colr);

}

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

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

putpixel(st+4+j,rows-11+i,colr);

for(i=0;i<3;i++)

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

{

putpixel(st-2+j,rows-4+i,colr);

putpixel(end-1+j,rows-4+i,colr);

}

for(i=0;i<3;i++)

{

putpixel(st-2,rows-1+i,colr);

putpixel(end+2,rows-1+i,colr);

}

}

void seat()

{

create\_seats();

setcolor(10);

setlinestyle(0,0,2);

setfillstyle(1,2);

char k[2]="\0";

k[0]='A';

for (int i = 0; i <9; i++,k[0]++)

{

outtextxy(t\_ob[i][0].x-50,t\_ob[i][0].y-10,k);

for (int j = 0; j <24; j++)

{

seat\_print(t\_ob[i][j].x, t\_ob[i][j].y,8);

}

}

}

int gx(int cx)

{

for(int x=60;x<=cx;x+=21)

{

if(x==186||x==463)x+=25;

}

if(x!=60)return x-21; else return x;

}

int gy(int cy)

{

for(int y=140;y<=cy;y+=25);//strt ycor-space,xcor-space,seat

return y-5;//y+20-25 160-25 , 60-21 ,6-12-6 ; formation

}

int exitpage()

{

websiteoutline();

setcolor(7);

outtextxy(78,10,"Secure:https://www.huliz.com/exit-page");

setcolor(12);

settextstyle(1,0,1);

outtextxy(135,165,"Thank You for choosing the huliz website.");

delay(2000);

exit(0);

}

int check\_home\_selection(int x,int y)

{

int ret=0;

if(x>=55&&x<=160&&y>=145&&y<=170)ret=1;

else if(x>=610&&x<=630&&y>=5&&y<=23)ret=1;

return ret;

}

void screen\_write()

{

ofstream f("screen.txt");

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

{ for(int j=0;j<getmaxy();j++)

{ int p=getpixel(i,j);

f<<p<<" ";

}

}

f.close();

}

void load\_game()

{

ifstream f("screen.txt");

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

{ for(int j=0;j<getmaxy();j++)

{ int p;

f>>p;

putpixel(i,j,p);

}

}

f.close();

}

void file\_write(char fname[])

{

ofstream f(fname,ios::binary);

for (int i = 0; i <9; i++)

{

for (int j = 0; j <24; j++)

{

if (t\_ob[i][j].pos==1) t\_ob[i][j].pos = 2;

f.write((char\*)&t\_ob[i][j],sizeof(t\_ob[i][j]));

}

}

f.close();

}

void file\_read(char fname[])

{

ifstream f(fname,ios::binary);

for (int i = 0; i <9; i++)

{

for (int j = 0; j <24; j++)

f.read((char\*)&t\_ob[i][j],sizeof(t\_ob[i][j]));

}

f.close();

}

void check(int cx, int cy, int mob, int num)

{

int flag=0;

layoutdetails();

for (int i = 0; i <9&&flag==0; i++)

for (int j = 0; j <24&&flag==0; j++)

{

int x, y;

x = t\_ob[i][j].x; y = t\_ob[i][j].y;

if((cx>=x-2 && cx<=x+15) && (cy >= y-15 && cy <=y+5))

{

if (t\_ob[i][j].pos!=2)

{

if (t\_ob[i][j].pos==0 && seat\_N <num)

{

t\_ob[i][j].col = BLUE;

t\_ob[i][j].pos=1; seat\_N++;

t\_ob[i][j].ph\_no=mob;

}

else if (t\_ob[i][j].pos==1 && seat\_N<=num)

{

t\_ob[i][j].col = 8;

t\_ob[i][j].pos=0; seat\_N--;

t\_ob[i][j].ph\_no=0;

}

seat\_print(x,y,t\_ob[i][j].col);

flag=1;

}

}

}

}

void moviebook()

{

restorecrtmode();

int nofseats,zero=0,date,mn;

long mobno;

char m\_name[5][60]= {"SPIDER MAN","ROBO","FIST OF FURY",

"IRON MAN","ROCKY III"};

cout << endl << endl;

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

{

cout << "\n"<<i+1 <<" : " << m\_name[i]<< endl;

}

do

{

cout << "\n\nEnter the number : " ;

cin >> mn;

} while(mn<1 || mn > 5);

char st[10];

itoa(mn-1,st,10);

char fname[20];

strcpy(fname, m\_name[mn-1]);

fname[3]='\0';

strcat(fname,st);

strcat(fname,".dat");

cout << fname;

seat\_N=0;

do

{

clrscr();gotoxy(2,3);

cout<<"Enter No. of tickets(1-10) : "; cin>>nofseats;

}while(nofseats<1||nofseats>10);

gotoxy(2,6);

cout <<"Enter Contact Phone number : ";

cin>>mobno;

gotoxy(2,8);

cout << "Cost of each Ticket : Rs 200" ;

gotoxy(2,10);

cout << "Total cost to be paid : Rs " << 200\*nofseats;

gotoxy(2,12);

long card,code;

cout <<"Enter the card Number : "; cin >> card;

gotoxy(2,14);

cout << "Enter the Password : "; cin >> code;

int gdriver=9,gmode=2;

initgraph(&gdriver,&gmode,"f:\\tc\\BGI");

int flag;

cleardevice();

ifstream f(fname,ios::binary);

if (!f)

{

loadscreen(2);

delay(200);

websiteoutline();

setcolor(7);

outtextxy(78,10,"Secure:https://www.huliz.com/movie-ticket");

layoutdetails();

seat();

}

else

{

setcolor(7);

cleardevice();

load\_game();

file\_read(fname);

}

int fl=0;

while(seat\_N<=nofseats && fl==0)

{

m.show\_mouse();

m.get\_status();

delay(300);

if (cx >=560 && cx <= 630 && cy>=447 && cy <=467 &&

seat\_N==nofseats) fl=1;

m.hide\_mouse();

if (fl==0) check(cx,cy,mobno,nofseats);

delay(300);

}

m.hide\_mouse();

screen\_write();

cleardevice();

websiteoutline();

setcolor(7);

outtextxy(78,10,"Secure:https://www.huliz.com/ticket-booked");

settextstyle(0,0,2);

setcolor(15);

outtextxy(50,65,"Thank You for using Huliz.");

settextstyle(0,0,1);

outtextxy(50,90,"You have sucessfully booked your ticket(s).");

outtextxy(50,115,"Do visit us again.");

delay(1000);

file\_write(fname);

}

void homepage()

{

//\*\*\*\* Website outline

websiteoutline();

setcolor(7);

outtextxy(78,10,"Secure:https://www.huliz.com/home-page");

setcolor(12);

settextstyle(0,0,2);

outtextxy(135,65,"Welcome to huliz website");

settextstyle(0,0,1);

setcolor(15);

box(55,145,160,170,15);

outtextxy(60,150,"Movie Ticket");

outtextxy(78,161,"Booking");

box(470,145,585,170,15);

outtextxy(500,155,"OTHELLO");

box(470,320,585,345,15);

outtextxy(500,330,"FUSIONS");

box(60,320,170,345,15);

outtextxy(71,330,"EYE - OPTICS");

}

void main()

{

int gdriver=9,gmode=2,res;

mouse ch;

do

{

initgraph(&gdriver,&gmode,"f:\\tc\\BGI");

homepage();

ch.show\_mouse();

ch.get\_status();

delay(100);

if(cx>= 55 && cx <= 160&& cy >= 145 && cy <= 170) moviebook();

else if(cx>= 470 && cx <= 585&& cy >= 145 && cy <= 170)othello();

else if(cx>= 470 && cx <= 585&& cy >= 320 && cy <= 345)fusion();

else if(cx>= 60 && cx <= 170&& cy >= 320 && cy <= 345)

{

ch.hide\_mouse();

cleardevice();

eye\_optics();

}

else if(cx>= 610 && cx <= 630&& cy >= 5 && cy <= 23)

{

ch.show\_mouse();

cleardevice();

exitpage();

}

}while(7);

}

**OTHELLO.CPP**

class square

{ int c\_x,c\_y;

int col;

public: void fill();

void plot();

void play(int game=3);

void draw\_circle(int ,int , int );

int check\_YELLOW();

int check(int, int, int, int);

int check\_chance(int ,int );

int load\_game(char \*);

void load\_fill();

void save\_game(int);

void write\_score(int , int );

void save\_option(char \* st);

void print\_rules();

char \* get\_input(int , int , int , int ,int = 30,int=80);

void Menu();

} ob[8][8];

mouse m;

int player[2]={2,2};

int pos\_i, pos\_j,player\_check;

// Filling the array ob with center x,y coodinates and YELLOW colour

void square ::fill()

{ int x=125,color;

int y=100;

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

{ x=125;

for(int j=0;j<8;j++,x = x+50)

{ ob[i][j].c\_x=x;

ob[i][j].c\_y=y;

ob[i][j].col = YELLOW;

}

y+=40;

}

// Filling the the center 4 boxes with BLUE and RED colour

ob[3][3].col= BLUE; ob[3][4].col= RED;

ob[4][3].col= BLUE; ob[4][4].col= RED;

for (i = 3; i<=4; i++)

{ for (int j = 3; j<=4; j++)

{

setcolor(WHITE);

setfillstyle(1,ob[i][j].col);

circle(ob[i][j].c\_x, ob[i][j].c\_y,15);

floodfill(ob[i][j].c\_x, ob[i][j].c\_y,WHITE);

}

}

}

void square ::load\_fill()

{ int x=125,color;

int y=100;

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

{ x=125;

for(int j=0;j<8;j++,x = x+50)

{ ob[i][j].c\_x=x;

ob[i][j].c\_y=y;

ob[i][j].col = getpixel(x,y);

}

y+=40;

}

}

void square::draw\_circle(int i, int j, int d\_c)

{

int x,y;

m.hide\_mouse();

x = 100+j\*50; y = 80+i\*40;

setfillstyle(1,YELLOW);

bar(x+2,y+2,x+48,y+38);

setcolor(WHITE);

circle(x+25, y+20,15);

setfillstyle(1,d\_c);

floodfill(x+25,y+20,WHITE);

m.show\_mouse();

}

int square::check(int x,int y, int c, int oc)

{

int flag,s,p,i,j;

i = pos\_i; j = pos\_j;

flag = 0; s= 0;

do

{

i= i+x; j = j+y;

if ((i<0 || i>7)|| ( j <0 || j >7)) flag= 1;

else

{

p = ob[i][j].col;

if (p==c || p==YELLOW) flag=1;

else if(p==oc) s++;

}

} while (flag==0);

if (s !=0 && p==c)

{

i = pos\_i; j = pos\_j; player\_check=1;

draw\_circle(i,j,c);

ob[i][j].col=c;

for (int k = 1; k <= s; k++)

{

i = i+x; j = j+y;

ob[i][j].col=c;

draw\_circle(i,j,c);

}

return s;

}

else return 0;

}

int square::check\_chance(int color,int op\_c)

{

int flag=0;

for (int i=0; i<8&&flag==0;i++)

{

for (int j=0; j<8&&flag==0;j++)

{

if (ob[i][j].col==YELLOW)

{

for (int k = -1,m=1; m<=9&&flag==0; m++)

{ if (m%3==0) k++;

if (ob[i+k][j+k].col==op\_c)flag=1;

}

}

}

}

return flag;

}

int square::check\_YELLOW()

{

int s = 0;

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

{

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

if (ob[i][j].col==YELLOW) s++;

}

return s;

}

// load already saved game back

int square::load\_game(char fname[])

{

int player\_no;

ifstream f(fname);

f>>player[0]>>player[1]>>player\_no;

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

{ for(int j=0;j<getmaxy();j++)

{ int p;

f>>p;

putpixel(i,j,p);

}

}

f.close();

return player\_no;

}

// write the score n of the player in the position x,210

void square::write\_score(int x, int n)

{

char st[10];

setfillstyle(1,0);

bar(x-10,200,x+30,250);

settextstyle(3,0,3);

setcolor(WHITE);

itoa(n, st,10);

outtextxy(x,210,st);

}

// displaying the options save

void square::save\_option(char \* st)

{

setfillstyle(1,BLUE);

bar(10,420,100,450);

setcolor(WHITE);

outtextxy(20,430,st);

}

// Get input from user in x,y coordinate

char \* square::get\_input(int x, int y, int bc, int col,int inc,int barcord)

{

char string[100]="\0",ch;

int i =0;

ch = getch();

while(ch!='\r')

{

if (ch!='\b') string[i]= ch;

else i=i-2;

string[i+1]= '\0';

setfillstyle(1,bc);

bar(x,y,x+barcord,y+inc);

setcolor(col);

outtextxy(x,y+10,string);

i++;

ch = getch();

}

return string;

}

// Write the game pixel by pixel in the hard disk

void file\_write(char f\_name[],int player\_no)

{

ofstream f(f\_name);

f <<player[0]<<" " << player[1]<< " "

<<player\_no<<" ";

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

{ for(int j=0;j<getmaxy();j++)

{ int p=getpixel(i,j);

f<<p<<" ";

}

}

f.close();

}

// Printing the Rules

void square::print\_rules()

{

char st[100];

int y=10;

ifstream f("rules.txt");

settextstyle(3,0,2);

while(f.getline(st,100,'\n'))

{ outtextxy(10,y,st);

y=y+40;

if(y>=440)

{ getch();

cleardevice();

y=10;

}

}

}

// displaying the option Main Menu

void return\_option()

{

setfillstyle(1,BLUE);

bar(300,420,380,450);

setcolor(WHITE);

outtextxy(303,430,"MAIN MENU");

}

// displaying the option Quit

void quit\_option()

{

setfillstyle(1,BLUE);

bar(580,420,640,450);

setcolor(WHITE);

outtextxy(590,430,"QUIT");

}

// displaying the Save

void square::save\_game(int player\_no)

{

char f\_name[20];

setfillstyle(1,BLUE);

bar(10,420,200,450);

setcolor(WHITE);

settextstyle(0,0,1);

outtextxy(10,430,"File Name : ");

strcpy(f\_name,get\_input(100,420,BLUE,WHITE));

settextstyle(0,0,1);

setfillstyle(1,BLUE);

bar(10,420,200,450);

save\_option("SAVING GAME...");

delay(1000);

setfillstyle(1,BLACK);

bar(10,420,200,450);

save\_option("SAVE GAME");

file\_write(f\_name,player\_no);

}

// Play the Game

void square::play(int game)

{

int color,opp\_col,points,pl\_chance,pl,k;

char st[10];

char name[2][50];

if(game==3)

{

settextstyle(3,0,3);

outtextxy(10,100,"Enter Player Name 1 : ");

strcpy(name[0],get\_input(270,90,0,WHITE,60,400));

outtextxy(10,200,"Enter Player Name 2 : ");

strcpy(name[1],get\_input(270,190,0,WHITE,60,400));

cleardevice();

setcolor(LIGHTCYAN);

settextstyle(3,0,4);

outtextxy(200,20,"O T H E L L O");

setfillstyle(1,YELLOW);

setcolor(GREEN);

setfillstyle(1,YELLOW);

setcolor(YELLOW);

bar(100,80,8\*50+100,80+8\*40);

setcolor(GREEN);

for(int x=100;x<=500;x+=50)

line(x,80,x,400);

for( int y=80;y<=400;y+=40)

line(100,y,500,y);

setcolor(4);

settextstyle(0,0,1);

fill();

save\_option("SAVE GAME");

return\_option();

quit\_option();

settextstyle(1,0,2);

setcolor(LIGHTBLUE);

outtextxy(10,100,name[0]);

setcolor(LIGHTRED);

outtextxy(575,100,name[1]);

write\_score(10,player[0]);

write\_score(550,player[1]);

game = 0;

}

m.show\_mouse();

int flag = 1,pos;

for (k = game; flag!=0; k++)

{ flag = check\_YELLOW();

if (flag==0) continue;

switch(k%2)

{

case 0 : color = BLUE;

opp\_col= RED;pos = 5;

pl=1;break;

case 1 : color = RED;

opp\_col= BLUE;pos = 550;

pl=0;

break;

}

pl\_chance= check\_chance(color,opp\_col);

if (pl\_chance==0)

{

cout<<"no chance"; flag=0;

player[pl]+= (64-k+1);

continue;

}

settextstyle(1,0,2);

setcolor(color);

outtextxy(pos,250,"Your");

outtextxy(pos,280,"chance");

m.get\_status(); delay(500);

pos\_j = (cx-100)/50;

pos\_i = (cy-80)/40;

if ((cx >= 580 && cx <=640) && (cy >= 420 && cy <= 450) ) exit(0);

if ((cx >= 10 && cx <=200) && (cy >= 420 && cy <= 450) )

{

m.hide\_mouse();

save\_game(k%2);

m.show\_mouse();k--;

setfillstyle(1,0);

bar(pos,250,pos+90,300);

continue;

}

if ((cx >= 300 && cx <=380) && (cy >= 420 && cy <= 450) )

{

flag=0;

continue;

}

if (ob[pos\_i][pos\_j].col==YELLOW)

{

int sum = 0,kk; kk = 0;

player\_check = 0;

for (int pos =1; pos <=8; pos++)

{

switch (pos)

{

case 1 : points= check(0,1,color,opp\_col); break;

case 2 : points= check(-1,1,color,opp\_col); break;

case 3 : points= check(-1,0,color,opp\_col); break;

case 4 : points= check(-1,-1,color,opp\_col);break;

case 5 : points= check(0,-1,color,opp\_col); break;

case 6 : points= check(1,-1,color,opp\_col); break;

case 7 : points= check(1,0,color,opp\_col); break;

case 8 : points= check(1,1,color,opp\_col); break;

}

if (points >0) kk++;

sum = sum+points;

}

if (kk>0) kk=1; else kk = 0;

if (kk==1)

{

switch(k%2&& sum!=0)

{

case 0 : player[0] += sum+kk; player[1]-= sum;break;

case 1 : player[1] += sum+kk; player[0]-= sum;break;

}

write\_score(10,player[0]);

write\_score(560,player[1]);

}

else k--;

}

delay(100);

setfillstyle(1,0);

bar(pos,250,pos+90,300);

}

//setcolor();

write\_score(10,player[0]);

write\_score(600,player[1]);

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

{

outtextxy(420,400,"Winner ");

outtextxy(400,250,name[0]);

}

else

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

{

outtextxy(420,400,"Winner ");

outtextxy(400,250,name[1]);

}

else outtextxy(400,400,"TIE ");

}

void square::Menu()

{

char st[][20] = {"1.New Game", "2.Load Game", "3.Help",

"4.Back to Main"};

char f\_name[20];

int a,b,k;

char ch[5]="\0";

do

{

a = 9; b = 2;

initgraph(&a,&b,"f:\\TC\\bgi");

setfillstyle(1,BLUE);

bar(0,0,getmaxx(),getmaxy());

setcolor(WHITE);

settextstyle(4,0,3);

b = 100;

for (int i = 0; i <=3;i++)

{

outtextxy(250,b,st[i]);

b = b+50;

}

outtextxy(200,b,"Enter Option : ");

ch[0] = getche();

//getch();

ch[1] = '\0';

outtextxy(360,b,ch);

//delay(1000);

getch();

cleardevice();

switch(ch[0])

{

case '1' :play(); break;

case '2' :outtextxy(10,100,"Enter File Name : ");

strcpy(f\_name,get\_input(270,90,0,WHITE));

k = load\_game(f\_name);load\_fill();

play(k);break;

case '3' :print\_rules();break;

case '4' :;//exit(0);

}

//getch();

closegraph();

}while(ch[0]!='4');

}

void othello()

{

square x;

x.Menu();

}

**FUSION.CPP**

void circle1(int a , int b, int c)

{

circle(a,b,20);

setfillstyle(1,c);

floodfill(a,b,WHITE);

setcolor(WHITE);

}

void disp(char a[],char b[], int c ,int d)

{

settextstyle(3,0,2);

outtextxy(c,d,a);

settextstyle(3,0,1);

outtextxy(c-10,d-10,b);

delay(500);

}

void disp2(int b , int c , char a[])

{

settextstyle(3,0,1);

outtextxy(b,c,a);

delay(500);

}

void disp3(int b , int c , char a[])

{

settextstyle(0,0,1);

outtextxy(b,c,a);

delay(500);

}

void circle2(int a,int b,int c)

{

circle(a,b,10);

setfillstyle(1,c);

floodfill(a,b,WHITE);

setcolor(WHITE);

}

void fusion()

{

int a=9,b=2;

initgraph(&a,&b,"c:\\turboc3\\bgi");

circle1(30,30,RED);// Hydrogen 1 //

disp(" H "," 1 ",55,30);

circle1(530,30,RED);

disp(" H "," 1 ",565,30);// Hydrogen 4 //

line(30,50,50,110);

delay(500);

line(530,50,490,110);

circle1(100,30,RED);//Hydrogen 2 //

disp(" H "," 1 ",125,30);

circle1(470,30,RED);//Hydrogen 3 //

disp(" H "," 1 ",425,30);

line(100,50,50,110);

delay(500);

line(470,50,490,110);

delay(500);

line(50,110,10,170);

delay(500);

line(490,110,360,153);

delay(500);

circle2(10,180,LIGHTCYAN); // Positron 1 //

disp2(0,200," Positron ");

circle2(350,160,LIGHTCYAN);// Positron 2 //

disp2(300,110," Positron ");

line(50,110,70,175);

delay(500);

circle1(50,180,GREEN);// H2 //

circle1(90,180,RED);

disp(" H "," 2 ",115,180);

line(490,110,550,173);

delay(500);

circle1(530,180,GREEN);// H2 //

circle1(570,180,RED);

disp(" H "," 2 ",595,180);

line(50,110,160,140);

delay(500);

disp3(160,140," Neutrino ");

line(490,110,590,140);

delay(500);

disp3(560,140," Neutrino ");

settextstyle(3,0,2);

outtextxy(140,170," + ");

delay(500);

settextstyle(3,0,2);

outtextxy(470,170," + ");

delay(500);

circle1(200,180,RED);// H1 //

disp(" H "," 1 ",235,180);

circle1(410,180,RED);// H1 //

disp(" H "," 1 ",440,180);

line(70,180,90,250);

delay(500);

line(550,173,540,250);

delay(500);

line(185,195,90,250);

delay(500);

line(420,200,540,250);

delay(500);

line(90,250,50,300);

delay(500);

line(540,250,600,300);

delay(500);

disp3(20,300," Gamma Rays ");

disp3(560,300," Gamma Rays ");

line(90,250,130,300);

delay(500);

line(540,250,460,300);

delay(500);

circle1(140,320,RED);// Helium 3 //

circle1(120,355,GREEN);

circle1(160,355,RED);

disp(" He "," 3 ",70,355);

circle1(450,320,RED); // Helium 3 //

circle1(430,355,GREEN);

circle1(470,355,RED);

setfillstyle(1,RED);

disp(" He "," 3 ",495,355);

line(180,355,290,375);

delay(500);

line(290,375,410,350);

delay(500);

line(290,375,330,395);

delay(500);

circle1(350,400,GREEN);// Helium 4 //

circle1(390,400,RED);

circle1(350,440,GREEN);

circle1(390,440,RED);

disp(" He "," 4 ",420,440);

line(290,375,290,435);

delay(500);

circle1(290,455,RED);

disp(" H "," 1 ",315,455);

line(290,375,200,425);

delay(500);

circle1(190,445,RED);

disp(" H "," 1 ",215,445);

getch();

}

**EYE**-**LENS.CPP**

void myopia()

{

outtextxy(320,30,"MYOPIA");

arc(450,100,270,450,75);

arc(500,100,150,210,75);

arc(370,100,330,390,75);

outtextxy(340,110,"F");

outtextxy(240,110,"2F");

outtextxy(350,310,"F");

outtextxy(250,310,"2F");

outtextxy(410,340,"EYE LENS");

outtextxy(290,370,"CORRECTIVE LENS");

outtextxy(410,140,"EYE LENS");

line(200,100,480,100);//center pt

line(200,65,435,65);//far pt to top of lens

line(435,65,480,100);//top left connecting line

line(200,135,435,135);

line(435,135,480,100);

arc(450,300,270,450,75);

arc(500,300,150,210,75);

arc(370,300,330,390,75);

arc(475,300,150,210,110);

arc(225,300,330,390,110);

line(200,300,525,300);

line(350,280,435,265);

line(200,280,350,280);

line(350,320,435,335);

line(200,320,350,320);

line(435,335,525,300);

line(435,265,525,300);

line(320,245,380,245);

line(320,355,380,355);

}

void normal()

{

outtextxy(340,260,"F");

outtextxy(240,260,"2F");

outtextxy(300,130,"NORMAL EYE");

outtextxy(410,290,"EYE LENS");

arc(450,250,270,450,75); arc(500,250,150,210,75);

arc(370,250,330,390,75); line(200,250,525,250);

line(200,225,435,225); line(435,225,525,250);

line(200,275,435,275); line(435,275,525,250);

}

void metropia()

{

outtextxy(320,30,"HYPERMETROPIA");

outtextxy(340,110,"F"); outtextxy(240,110,"2F");

outtextxy(350,310,"F"); outtextxy(250,310,"2F");

outtextxy(410,340,"EYE LENS");

outtextxy(290,370,"CORRECTIVE LENS");

outtextxy(410,140,"EYE LENS");

arc(450,100,270,450,75); arc(500,100,150,210,75);

arc(370,100,330,390,75); arc(450,300,270,450,75);

arc(500,300,150,210,75); arc(370,300,330,390,75);

line(200,100,555,100); line(200,75,435,75);

line(435,75,555,100); line(200,125,435,125);

line(435,125,555,100); arc(450,300,150,210,110);

arc(260,300,330,390,110); line(200,300,525,300);

line(355,255,435,265); line(355,345,435,335);

line(435,335,525,300); line(435,265,525,300);

line(200,255,355,255); line(200,345,355,345);

}

void eye\_lens()

{

setcolor(7);

myopia();

delay(500);

cleardevice();

normal();

delay(500);

cleardevice();

metropia();

getch();

}

**MOUSE-1.CPP**

int cx,cy,q;

class mouse

{

public: REGS regs;

int no\_buttons;

public:

mouse()

{ regs.x.ax=0;

int86(0x33,&regs,&regs);

if(regs.x.ax==0xfff)no\_buttons=regs.x.bx;

}

void show\_mouse()

{ regs.x.ax=0x01;

int86(0x33,&regs,&regs);

}

void hide\_mouse()

{

regs.x.ax=0x02;

int86(0x33,&regs,&regs);

}

void get\_status()

{ regs.x.ax=0x03;

int86(0x33,&regs,&regs);

while(q!=1)

{

regs.x.ax=0x03;

int86(0x33,&regs,&regs);

q=regs.x.bx;

cx=regs.x.cx;

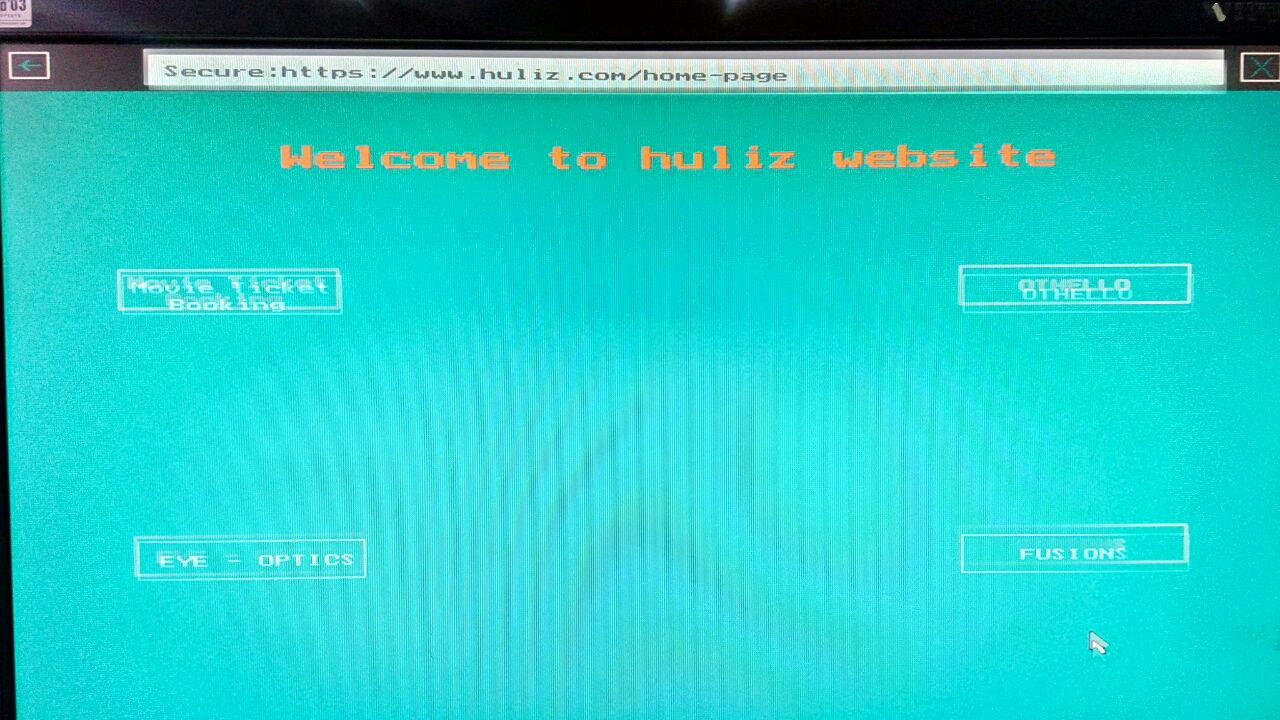
cy=regs.x.dx;

}

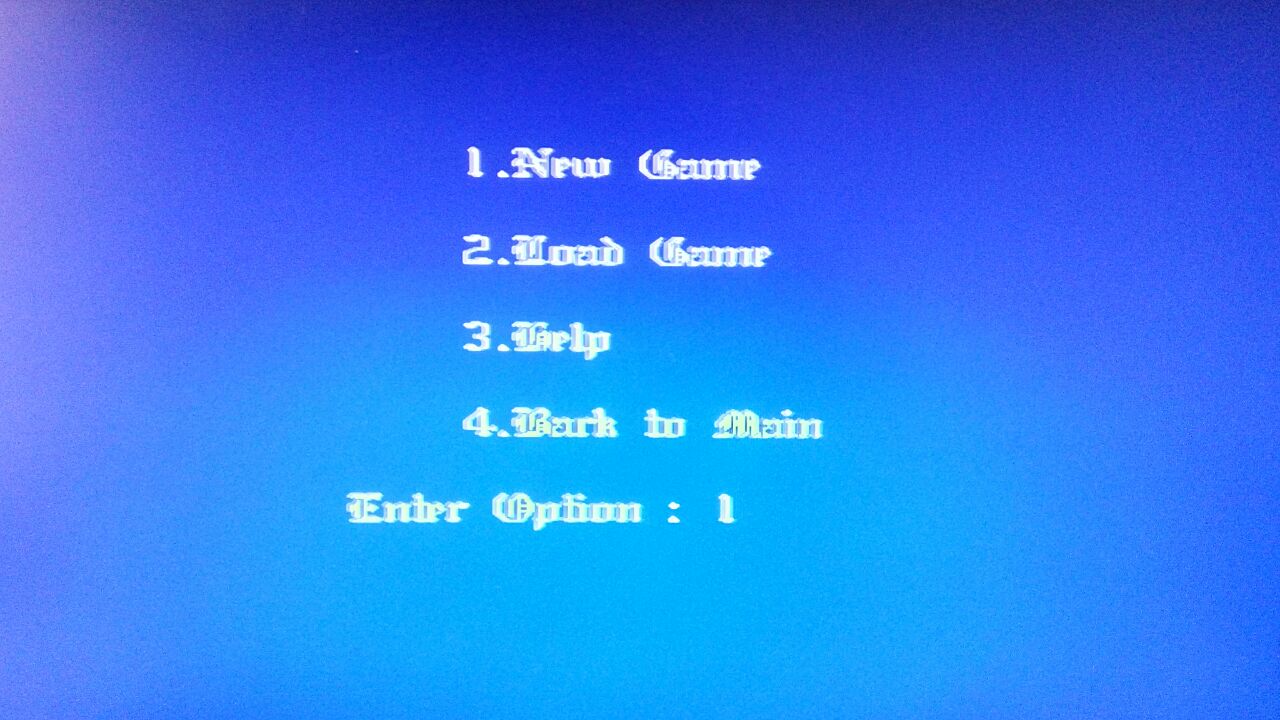
q=0;

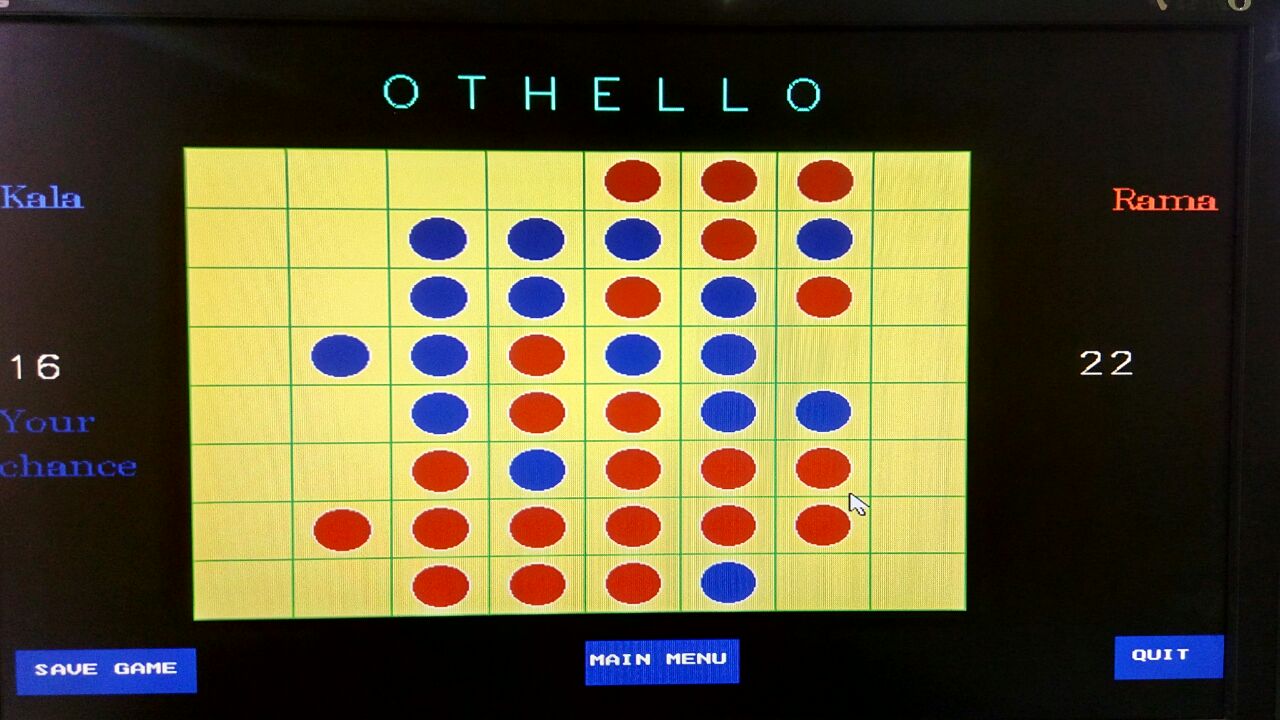
}

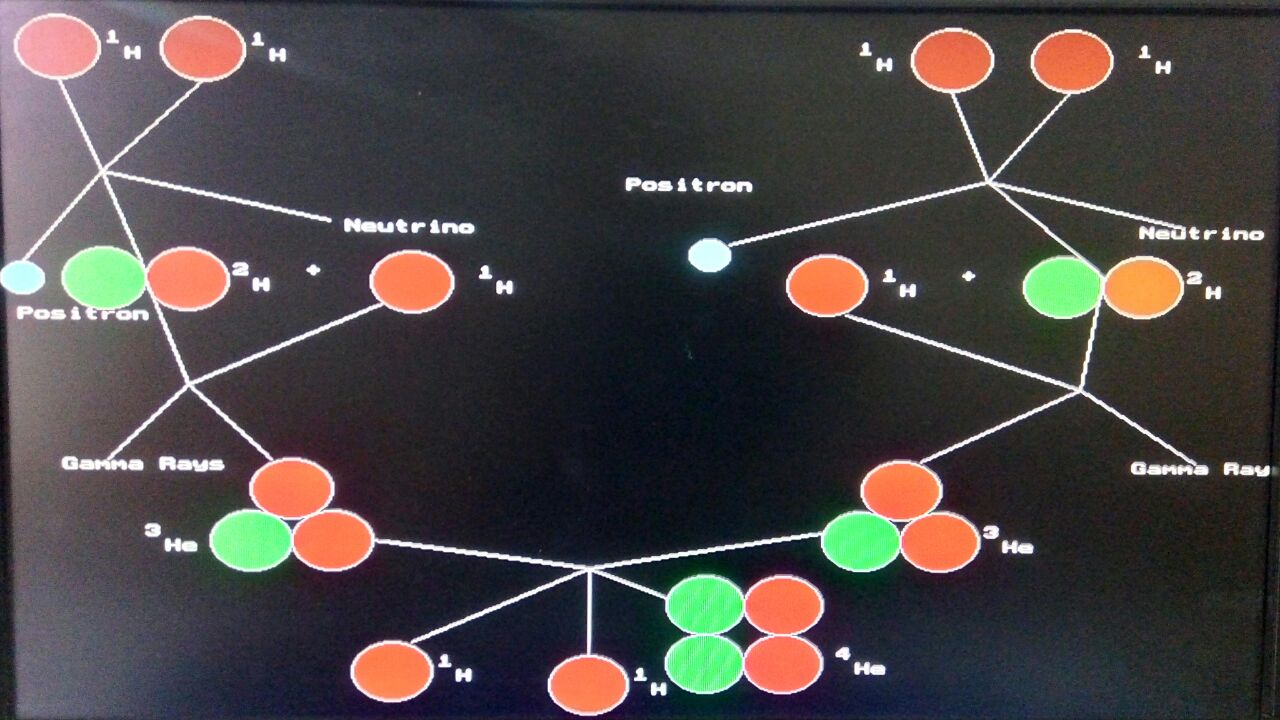
};

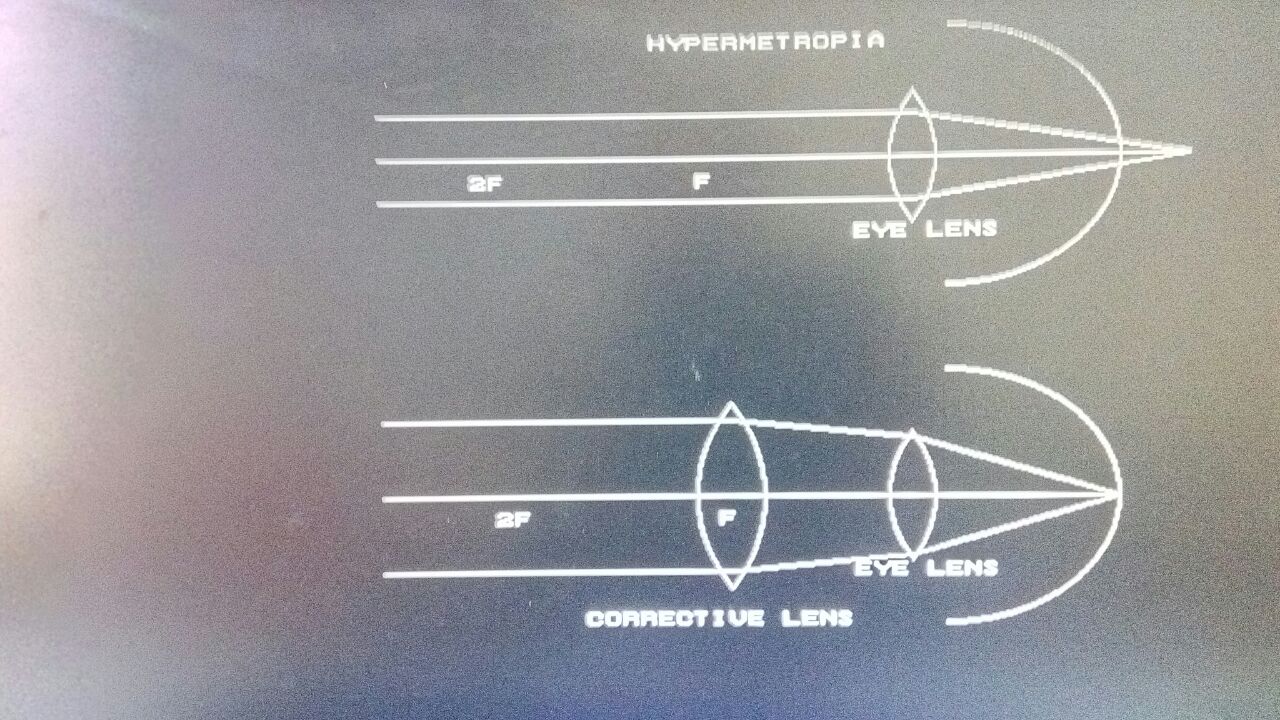
****

****

****

****

****

****