****

Computer Science Project

Group Members:

K Prahlad Narasimhan

Roshan S Kaundinya

Aravind Panchanathan

Nanda H Krishna

**ACKNOWLEDGEMENTS**

We would like to thank the CS Department for all the help and guidance they gave us for this project. We would like to thank our teachers, who assisted us and helped us complete the project and also gave us valuable suggestions and ideas. We would like to thank everyone who supported us in the successful completion of this Project. Finally we would like to thank the school for providing to us all the facilities required and making the experience of the Project wonderful and something we would cherish forever.

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Topic** | **Page** |
| 1 | Salient Features |  |
| 2 | Project Analysis |  |
| 3 | Project Listing |  |
| 4 | Sample Output |  |

**SALIENT FEATURES**

The project is a collection of 8 applications, each with a unique function and design. The whole integrated project is a ‘simulation of an operating system’. The various applications contained in the project are:

* Apps:

1. Calendar
2. Clock
3. Calculator
4. Notepad

* Games:

1. Connect 4
2. Rall-E
3. Mastermind
4. Space Wars

A detailed explanation of each is given in the project analysis.

The applications are accessible through dedicated homescreens, one for the 4 Apps and one for the 4 Games. These homescreens are customizable – the user can choose from a set of wallpapers included in the package. The loading screen for the applications can also be chosen from a set of included loading screens.

The project is thus a near imitation or simulation of the user interface of a basic operating system, with utilitarian Apps and fun to play and interesting Games.

**PROJECT ANALYSIS**

**Apps**

**Calendar**

The calendar application is a functional basic application that allows the user to view the date, day and year. It displays in month-wise format. The current day is highlighted so that it is noticeable. One can navigate between months and years efficiently by clicking on the arrows on-screen. To go to a desired year, an ‘Input Year’ text button is also given. Important dates in the current month can be accessed through the ‘Important Dates’ text button. The calendar works based on Zeller’s Rule, a famous mathematical rule used to calculate the day based on the date, by plugging in certain ‘key’ numbers in a formula. The application uses graphics, time functions, files and many other C++ concepts.

**Clock**

The clock application is a functional basic application that allows the user to view the current system time. It is an analog clock with a digital bar also present for the user’s convenience. The clock displays the current year on the top and also shows the current date and month in short form, and the current day. The hands of the clock are displayed based on mathematical formulae related to circles, modified for proper viewing on the graphical screen. The application uses graphics, time functions and other C++ concepts.

**Calculator**

The calculator is a functional application allowing the user to perform basic and some advanced mathematical calculations. It is operated using the mouse, by clicking the necessary on-screen buttons. When a button is clicked, the colour of the text changes indicating that a click was registered. A large font display is also present for easy viewing. The calculator returns errors in cases of incorrect mathematical syntax or if the result exceeds the calculator’s range limit. Functions include square root, square, cosine, sine, arithmetic operations, etc. The user can store a certain result by clicking a dedicated button and the same can be pasted on the calculator screen for further use using another dedicated button. Other dedicated buttons include those for accessing previous calculations and rules respectively. The calculator uses graphics, stack and infix-postfix concepts, files, mathematics and many others.

**Notepad**

The notepad is a functional application that allows the user to perform basic document related activities such as opening a file, creating a new file, typing and inserting data into a file, saving a file, etc. The user is allowed to create a new file or open and edit an existing file. Once the user is done, the file can be saved or not saved based on the user’s choice. Navigation of the cursor is done with the help of the arrow keys which employs the BIOSKEY concept. The notepad also uses files, linked list (dynamic pointers), graphics and other concepts.

**Games**

**Connect 4**

Connect 4 is a popular two-player connection game in which the players first choose a colour and then take turns dropping coloured discs from the top into a seven-column, six-row vertically suspended grid. The pieces fall straight down, occupying the next available space within the column. The objective of the game is to be the first to form a horizontal, vertical, or diagonal line of four of one's own discs. Connect Four is a [solved](https://en.wikipedia.org/wiki/Solved_game) game. On-screen, an arrow marker indicating the current player’s colour is used to select the column and the arrow keys are used to move the marker. The space bar is used to drop a coin. The two players can enter their names and play, and continue playing if they wish to do so. A tally of wins is maintained. The game employs graphics, array concepts and many others.

**Rall-E**

Rall-E is a simple game of Table Tennis where the user is ‘given’ a racket and must play against the computer machine. The objective of the game is to keep rallying the ball and not miss. The user racket is moved using up and down arrow keys in an arc and the user ball is thus directed towards the machine if hit. Points are awarded based on the colour of the ball. Black ‘gravity’ balls must not be hit and are automatically recalled by the machine if they are avoided. The game is never ending, that is, until the user misses, the game will continue forever. The game employs graphics, inheritance, files for high-scores, line equations of mathematics and other concepts.

**Mastermind**

Mastermind is a popular brain game. This version of Mastermind is also called Cows and Bulls. It is a number guessing game where the user can choose to play against another person or against the computer. If playing against another person, the other person must enter a four digit number with no repeated digits. If playing against the computer, it automatically generates a proper random number. The user must guess the number within 10 turns. The clues given are the number of Cows and Bulls. If a digit exists in the right place, it is a Cow. If a digit is in the wrong place, it is a Bull. Preference is given to Cow for calculation of the number of Cows and Bulls. Once 4 Cows are obtained, the game ends. The game employs graphics, arrays and other concepts.

**Space Wars**

Space Wars is an interesting action game. It involves three levels. The user is ‘given’ a UFO in space that has the power to shoot one bullet at a time. The user has three lives. In each level, the user must shoot and eliminate the foe spaceships that are constantly shooting bullets which must be avoided. If the user UFO is hit, a life is lost. If the user hits the enemy spaceships once, they are eliminated. The third level has an extra powerful ‘Boss’ with three lives and must thus be shot thrice. Progression from one level to another is possible only when all enemies of the current level are eliminated. The game uses graphics, files for high-scores and other concepts.

**PROJECT LISTING**

**BROSAPPS**

#include<stdio.h>

#include<conio.h>

#include<bios.h>

#include<fstream.h>

#include<graphics.h>

#include<math.h>

#include<ctype.h>

#include<string.h>

#include<stdlib.h>

#include<dos.h>

#define PI 3.141593

#define R 170

#define sc setcolor

#include<process.h>

#include"mousesrc.cpp"

#include"brosnloa.cpp"

#include"broscloc.cpp"

#include"broscalc.cpp"

#include"broscal.cpp"

#include"ftest.cpp"

#include"brosload.cpp"

#include"brloads.cpp"

#include"calcapps.cpp"

#include"pacman.cpp"

char \*brOS[] = { "Prahlad", "Roshan", "Aravind", "Nanda" };

void main()

{ int gd = DETECT, gm;

initgraph(&gd,&gm,"C:\\TC\\BGI");

cleardevice(); brosn();

delay(2000);

cleardevice(); screen();

initmouse(); i.x.ax=2;

setlinestyle(1,0,1);

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

{ setfillstyle(SOLID\_FILL,BLACK);

bar(0,a,getmaxx(),a+getmaxy()/30);

delay(5);

}

delay(100);

cleardevice(); pac();

delay(1000);

setbkcolor(BLACK);

cleardevice(); sc(WHITE);

settextstyle(2,0,10);

for(int e = -1, ex, ey = getmaxy()/2-145; e < 4; e++, ey+=60)

{ if(e==-1)

{ ex = getmaxx()/2 - 4\*11;

outtextxy(ex,ey,"brOS");

line(ex-3,ey+39,ex+90,ey+39);

}

else

{

ex = getmaxx()/2 - 11\*strlen(brOS[e]);

outtextxy(ex,ey,brOS[e]);

}

}

delay(3000);

cleardevice();

closegraph();

}

**BROSGAME**

#include<stdio.h>

#include<conio.h>

#include<bios.h>

#include<fstream.h>

#include<graphics.h>

#include<math.h>

#include<ctype.h>

#include<string.h>

#include<stdlib.h>

#include<dos.h>

#define PI 3.141593

#define R 170

#define sc setcolor

#include<process.h>

#include"mousesrc.cpp"

#include"brosnloa.cpp"

#include"ralle.cpp"

#include"connect4.cpp"

#include"fspace.cpp"

#include"cowsbull.cpp"

#include"brosload.cpp"

#include"brloads.cpp"

#include"calcgames.cpp"

#include"pacman.cpp"

char \*brOS[] = { "Prahlad", "Roshan", "Aravind", "Nanda" };

void main()

{ int gd = DETECT, gm;

initgraph(&gd,&gm,"C:\\TC\\BGI");

cleardevice();

brosn();

delay(2000);

cleardevice(); screen();

initmouse(); i.x.ax=2;

setlinestyle(1,0,1);

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

{ setfillstyle(SOLID\_FILL,BLACK);

bar(0,a,getmaxx(),a+getmaxy()/30);

delay(5);

}

delay(100);

cleardevice(); pac();

delay(1000);

setbkcolor(BLACK);

cleardevice();

sc(WHITE);

settextstyle(2,0,10);

for(int e = -1, ex, ey = getmaxy()/2-145; e < 4; e++, ey+=60)

{ if(e==-1)

{ ex = getmaxx()/2 - 4\*11;

outtextxy(ex,ey,"brOS");

line(ex-3,ey+39,ex+90,ey+39);

}

else

{

ex = getmaxx()/2 - 11\*strlen(brOS[e]);

outtextxy(ex,ey,brOS[e]);

}

}

delay(3000);

cleardevice();

closegraph();

}

**BRLOADS**

void orbit()

{

initmouse();

i.x.ax=2;

void orbits();

float xx,yy; float aa = 40; float bb = 85; float a = 50; float b = 12; float A = 135;

float B = 90; float Ra = 5; float x,y,X,Y,ex,sq;

int r = 0;

setcolor(14);

for(float i=0,j=0;i<20;j++)

{

x=a\*cos(i+10)-a\*sin(i+10)+15;

y=b\*sin(i+10)+a\*cos(i+10);

X=A\*cos(i+20)-B\*sin(i+20+90)+15;

Y=B\*sin(i+20)+B\*cos(i+20+90);

xx=aa\*cos(i)+15;

yy=bb\*sin(i);

setcolor(14);

setfillstyle(1,14);

circle(x+300,y+200,Ra);

floodfill(x+300,y+200,14);

setcolor(14);

setfillstyle(1,14);

circle(xx+300,yy+200,Ra);

floodfill(xx+300,yy+200,14);

setcolor(14);

setfillstyle(1,14);

circle(X+300,Y+200,Ra);

floodfill(X+300,Y+200,14);

delay(70);

setcolor(0);

setfillstyle(1,0);

circle(x+300,y+200,Ra);

floodfill(x+300,y+200,0);

setcolor(0);

setfillstyle(1,0);

circle(xx+300,yy+200,Ra);

floodfill(xx+300,yy+200,0);

setcolor(0);

setfillstyle(1,0);

circle(X+300,Y+200,Ra);

floodfill(X+300,Y+200,0);

orbits();

i+=0.2;

}

do

{

setcolor(14);

setfillstyle(1,14);

circle(x+300,y+200,Ra);

floodfill(x+300,y+200,14);

setcolor(14);

setfillstyle(1,14);

circle(xx+300,yy+200,Ra);

floodfill(xx+300,yy+200,14);

setcolor(14);

setfillstyle(1,14);

circle(X+300,Y+200,Ra);

floodfill(X+300,Y+200,14);

delay(500);

bar(int(r),getmaxy()-40,int(r+getmaxx()/4+4),getmaxy());

r=r+getmaxx()/4;

}while(r<int(getmaxx()));

}

void orbits()

{

for(float i=0;i<=60;i+=1)

{

float xx,yy,aa=45,bb=80,x,y,X,Y,a=50,b=9,A=145,B=100,ex,sq;

x=a\*cos(i)-a\*sin(i)+15;

y=b\*sin(i)+a\*cos(i);

X=A\*cos(i)-B\*sin(i+90)+15;

Y=B\*sin(i)+B\*cos(i+90);

xx=aa\*cos(i)+15;

yy=bb\*sin(i);

putpixel(x+300,y+200,14);

putpixel(X+300,Y+200,14);

putpixel(xx+300,yy+200,14);

}

}

void curvegra()

{

initmouse(); i.x.ax = 2;

setbkcolor(BLACK);

setcolor(YELLOW);

setlinestyle(0,0,1);

for(int y1=35,x1=325,x2=325;y1<=225;y1+=5,x1+=4,x2-=4)

{

line(325,y1,x1,240);

line(325,450-y1,x2,240);

//delay(200);

line(325,y1,x2,240);

line(325,450-y1,x1,240);

delay(300);

}

int r=0;

setcolor(RED);

for(x1=220,y1=120;r<=50;r+=3)

{

circle(x1,y1,r);

delay(30);

}

delay(100);

r=0;

setcolor(LIGHTRED);

for(x1=420,y1=370;r<=50;r+=3)

{

circle(x1,y1,r);

delay(30);

}

delay(100);

r=0;

setcolor(BLUE);

for(x1=420,y1=120;r<=50;r+=3)

{

circle(x1,y1,r);

delay(30);

}

delay(100);

r=0;

setcolor(LIGHTBLUE);

for(x1=220,y1=370;r<=50;r+=3)

{

circle(x1,y1,r);

delay(30);

}

r=0;

}

void discoc()

{

initmouse(); i.x.ax = 2; cleardevice();

int x,y;

x=getmaxx()/2;

y=getmaxy()/2;

randomize();

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

{

setcolor(random(15)+1);

for(int i=0;i<50;i++)

{

circle(x,y,i) ;delay(5);

}

setcolor(random(15)+1);

for(int j=70;j<120;j++){circle(x,y,j);delay(5);}

setcolor(random(15)+1);

for(int k=140;k<=190;k++){circle(x,y,k);delay(5);}

setcolor(random(15)+1);

for(int l=210;l<=230;l++)

{ circle(x,y,l);delay(5); }

delay(200);

}

setcolor(BLACK);

for(a=0;a<232;a++)

{

circle(x,y,a); delay(5);

}

}

void polka()

{ initmouse(); i.x.ax = 2;

int mx=getmaxx()/2,my=getmaxy()/2;

setcolor(YELLOW);

setfillstyle(1,YELLOW);

pieslice(mx-100,my,0,360,40);

delay(250);

setcolor(WHITE);

setfillstyle(1,WHITE);

pieslice(mx-100,my-100,0,360,40);

delay(250);

setcolor(RED);

setfillstyle(1,RED);

pieslice(mx,my-100,0,360,40);

delay(250);

setcolor(CYAN);

setfillstyle(1,CYAN);

pieslice(mx+100,my-100,0,360,40);

delay(250);

setcolor(GREEN);

setfillstyle(1,GREEN);

pieslice(mx+100,my,0,360,40);

delay(250);

setcolor(MAGENTA);

setfillstyle(1,MAGENTA);

pieslice(mx+100,my+100,0,360,40);

delay(250);

setcolor(DARKGRAY);

setfillstyle(1,DARKGRAY);

pieslice(mx,my+100,0,360,40);

delay(250);

setcolor(BROWN);

setfillstyle(1,BROWN);

pieslice(mx-100,my+100,0,360,40);

delay(250);

setcolor(BLUE);

setfillstyle(1,BLUE);

pieslice(mx,my,0,360,30);

delay(400);

setcolor(BLACK);

setfillstyle(1,BLACK);

pieslice(mx-100,my,0,360,40);

delay(250);

pieslice(mx-100,my-100,0,360,40);

delay(250);

pieslice(mx,my-100,0,360,40);

delay(250);

pieslice(mx+100,my-100,0,360,40);

delay(250);

pieslice(mx+100,my,0,360,40);

delay(250);

pieslice(mx+100,my+100,0,360,40);

delay(250);

pieslice(mx,my+100,0,360,40);

delay(250);

pieslice(mx-100,my+100,0,360,40);

delay(250);

setcolor(BLUE);

setfillstyle(1,BLUE);

pieslice(mx,my,0,360,40);

delay(500);

}

void goku()

{ cleardevice();

initmouse(); i.x.ax = 2;

int points[]={70,140,85,190,110,205,120,205,145,190,160,140,156,148,150,154,153,146,156,127,146,142,131,154,134,144,135,131,122,147,119,159,119,166,107,152,106,137,111,124,91,141,83,153,85,130,101,111,77,139,75,153,70,140},points1[]={85,140,80,150,75,145,110,170,110,165,85,140},points2[]={120,170,120,165,145,140,150,150,155,145,120,170};

int points3[]={89,154,86,159,90,171,100,174,109,169,89,154},points4[]={141,154,144,159,140,171,130,174,121,169,141,154};

int p[]={118,177,118,185,119,184,118,177};

int mx=getmaxx(),my=getmaxy();

setcolor(WHITE);

randomize();

for(int i=0;i<1000;i++)

putpixel(random(mx),random(my-200)+208,WHITE);

settextstyle(2,0,6);

outtextxy(getmaxx()-120,getmaxy()-25,"ROLDASH INC.");

drawpoly(27,points);

drawpoly(6,points3);

drawpoly(6,points4);

fillpoly(6,points1);

fillpoly(6,points2);

circle(132,163,3);

circle(98,163,3);

floodfill(132,163,WHITE);

floodfill(98,163,WHITE);

line(111,190,115,191);

line(115,191,119,190);

line(119,190,123,189);

line(123,189,128,186);

fillpoly(4,p);

int a[]={75,153,68,150,59,162,65,175,75,185,82,182,75,153},b[]={155,153,162,150,171,162,163,179,155,185,147,182,155,153};

drawpoly(7,a);

drawpoly(7,b);

int hair[]={65,175,48,162,35,157,45,153,53,155,24,131,7,125,38,122,51,127,22,96,1,91,36,77,70,80,54,60,30,51,54,60,70,80,75,49,63,19,42,8,67,8,105,28,120,48,125,62,135,35,131,11,148,47,140,69,152,71,160,74,168,79,175,86,177,94,183,105,213,106,235,111,205,119,188,129,210,126,223,127,200,147,181,159,198,155,182,172,163,179,171,162,162,150,155,153,156,148,150,154,153,146,156,127,146,142,131,154,134,144,135,131,122,147,119,159,119,166,107,152,106,137,111,124,91,141,83,153,85,130,101,111,77,139,75,153,68,150,59,162,65,175};

drawpoly(116,hair);

int ear1[]={65,159,68,157,75,172,74,179,70,176,70,168,73,167,70,168,70,176,74,179,75,172,68,157,65,159};

int ear2[]={165,159,162,157,155,172,156,179,160,176,160,168,157,167,160,168,160,176,156,179,155,172,162,157,165,159};

drawpoly(13,ear1);

drawpoly(13,ear2);

line(68,159,67,164);

line(162,159,163,164);

int m1[]={105,190,125,190,125,199,105,199,105,190},m2[]={105,190,125,190,125,194,105,194,105,190};

gotoxy(38,8);

setcolor(WHITE);

delay(500);

for(int r=0;r<=50;r++)

{

setcolor(CYAN);

circle(100,338,r);

setcolor(WHITE);

if(r==10){ cout<<"KAA...";

setcolor(BLACK);

line(111,190,115,191);

line(115,191,119,190);

line(119,190,123,189);

line(123,189,128,186);

setcolor(WHITE);

drawpoly(5,m1);

delay(400);

}

else if(r==19){ setcolor(BLACK);

drawpoly(5,m1);

setcolor(WHITE);

line(105,190,125,190);

}

else if(r==20){ cout<<"ME...";

drawpoly(5,m2);

delay(400);

}

else if(r==29){ setcolor(BLACK);

drawpoly(5,m2);

setcolor(WHITE);

line(105,190,125,190);

}

else if(r==30){ cout<<"HA...";

drawpoly(5,m1);

delay(400);

}

else if(r==39){ setcolor(BLACK);

drawpoly(5,m1);

setcolor(WHITE);

line(105,190,125,190);

}

else if(r==40){ cout<<"ME...";

drawpoly(5,m2);

delay(400);

}

else if(r==49){ setcolor(BLACK);

drawpoly(5,m2);

setcolor(WHITE);

line(105,190,125,190);

}

else if(r==50){ cout<<"HA!!!!!";

drawpoly(5,m1);

}

delay(100);

}

setfillstyle(1,CYAN);

for(int s=100;s<=600;s++)

{ for(int t=313;t<=363;t++)

{

bar(s,t,s+1,t+1);

}

delay(5);

}

delay(500);

setcolor(BLACK);

drawpoly(5,m1);

setcolor(WHITE);

line(111,190,115,191);

line(115,191,119,190);

line(119,190,123,189);

line(123,189,128,186);

delay(1000);

}

void coolcirc()

{

initmouse(); i.x.ax = 2; setcolor(YELLOW);

int maxX=getmaxx()/2,j=1;

int maxY=getmaxy()/2;

for(float i=0,x1,y1;i<=360;i+=0.1,j++)

{

x1=150\*cos(i\*3.14159265/180);

y1=150\*sin(i\*3.14159265/180);

putpixel(x1+maxX,y1+maxY,YELLOW);

if(j%20==0)

{

setcolor(BLUE);

circle(x1+maxX,y1+maxY,50);

}

if(j%15==0)

{

setcolor(BROWN);

line(maxX,maxY,x1+maxX,y1+maxY);

}

delay(2);

}

delay(500);

}

int wallno,loadno;

char wallpapers[7][15]={"DEFAULT","MINION","BATMAN","CIVIL WAR","VM","SPACE","CHENNAI"},loadings[7][15]={"DEFAULT","CIRCULAR","GRAPHICAL","GOKU","POLKA","DISCO","ORBIT"};

void loads()

{ cleardevice();

switch(loadno)

{ case 0: brosload(); break;

case 1: coolcirc(); break;

case 2: curvegra(); break;

case 3: goku(); break;

case 4: polka(); break;

case 5: discoc(); break;

case 6: orbit(); break;

}

}

void refresh()

{

initmouse(); i.x.ax = 2; cleardevice();

int r1 = random(14) + 1; int r2;

do

{ r2 = random(15);

}while(r2==r1);

setbkcolor(BLACK); setlinestyle(1,0,1); setfillstyle(SOLID\_FILL,r1);

setcolor(r1);

for(int r = 0; r <= getmaxy()/2-100; r+=3)

{ pieslice(getmaxx()/2,getmaxy()/2,0,360,r);

delay(0.1);

}

setcolor(r2); settextstyle(3,0,1);

outtextxy(getmaxx()/2-18,getmaxy()/2-10,"brOS");

delay(1000);

setfillstyle(SOLID\_FILL,BLACK);

}

**BROSCAL**

#include"mouse-1.cpp"

struct dosdate\_t d,s;

int year,mon;

void findday();

void cal\_display();

void publicholiday();

void checkpublic\_holiday();

void input\_year();

void cal();

void calendar()

{

cleardevice();

\_dos\_getdate(&d);

year=int(d.year);

mon=int(d.month);

findday();

cal\_display();

mouse.init\_mouse();

mouse.hide\_mouse();

}

void findday()

{

int Century,Decade,Final\_Value,Month\_january=11,total=0,Remaining\_Days,First\_day\_Month,x1=125,y1=150,a,b,c;

\_dos\_getdate(&s);

a=int(s.year);

b=int(s.month);

c=int(s.day);

int s=0,d;

char st2[3],st4[5];

int days[]={31,28,31,30,31,30,31,31,30,31,30,31};

char month[12][20] ={"JANUARY","FEBRUARY","MARCH","APRIL","MAY","JUNE","JULY","AUGUST","SEPTEMBER","OCTOBER","NOVEMBER","DECEMBER"};

year--;

Century=year/100;

Decade=year%100;

Final\_Value=(1+((13\*Month\_january-1)/5)+Decade+(Decade/4)+(Century/4)-(2\*Century))%7;// Zellar Formula Which Says k+[13\*m-1/5] + D + D/4 + C/4 -2\*C

if(Final\_Value <0)Final\_Value+=7;

year++;

if(year%4==0&&year%100==0&&year%400==0)

days[1]=29;

else if(year%4==0&&year%100!=0)

days[1]=29;

for(int r=0;r<mon-1;r++)

{

total+=days[r];

}

Remaining\_Days=total%7;

First\_day\_Month =(Final\_Value + Remaining\_Days);

if(First\_day\_Month>=7)First\_day\_Month=(First\_day\_Month)%7;

for( r=1;r<=First\_day\_Month;r++) {x1+=62;s++;}

cal();

itoa(year,st4,10);

setcolor(15);

settextstyle(2,0,8);

outtextxy(getmaxx()/2-strlen(month[mon-1])\*7,80,month[mon-1]);

outtextxy(getmaxx()/2-strlen(st4)\*7,50,st4);

settextstyle(1,0,1);

for( r=1;r<=days[mon-1];r++)

{

itoa(r,st2,10);

if(x1>=83&&x1<=147) setcolor(RED);

else setcolor(BLACK);

if(year==a && mon==b && r==c) setcolor(GREEN);

if(r/10==0)

{

char t=st2[0];

st2[1]=t;

st2[0]='0';

st2[2]='\0';

}

if(s>=6)

{

if(year==a && mon==b && r==c) setcolor(GREEN);

else setcolor(BLACK);

outtextxy(x1,y1,st2);

y1=y1+30;

x1=x1-360;

}

if(s!=6)

{ outtextxy(x1,y1,st2);

x1+=60;

s++;

}

else s = 0;

}

}

void cal\_display()

{

mouse.init\_mouse();

mouse.show\_mouse();

int x=1;

while(x)

{

mouse.get\_status();

if(mouse.cx>=getmaxx()/2-222&&mouse.cy>=55&&mouse.cx<=getmaxx()/2-172&&mouse.cy<=100)

{

year--;

findday();

}

else if(mouse.cx>=(getmaxx()/2)-162 && mouse.cy>=55 && mouse.cx<=(getmaxx()/2)-122 && mouse.cy<=100)

{

mon--;

if(mon<=0)

{

mon+=12;

year--;

}

findday();

}

else if(mouse.cx>=getmaxx()/2+122 && mouse.cy>=55 && mouse.cx<=getmaxx()/2+162 && mouse.cy<=100)

{

mon++;

if(mon>12)

{

mon=mon-12;

year++;

}

findday();

}

else if(mouse.cx >=getmaxx()/2+172 && mouse.cy>=55 && mouse.cx<=getmaxx()/2+222 && mouse.cy<=100)

{

year++;

findday();

}

else if(mouse.cx>=70 && mouse.cy>=390 && mouse.cx<=165 && mouse.cy<=420)

{

input\_year();

}

else if(mouse.cx>=250 && mouse.cy>=390 && mouse.cx<=400 && mouse.cy<=420)

{

publicholiday();

}

else if(mouse.cx>=520 && mouse.cy>=390 && mouse.cx<=580 && mouse.cy<=420)

{

x=0;

}

delay(200);

}

}

void cal()

{

cleardevice();

int x=120,y=117;

char \*day[]={"SUN","MON","TUE","WED","THR","FRI","SAT"};

settextstyle(3,0,1);

setcolor(BLACK);

setfillstyle(1,0);

bar(70,390,185,420);

bar(250,390,400,420);

bar(520,390,580,420);

setcolor(CYAN);

outtextxy(70,390,"INPUT YEAR");

outtextxy(260,390,"IMPORTANT DATES");

outtextxy(520,390,"EXIT");

setbkcolor(0); settextstyle(10,0,3); setcolor(BLACK); setfillstyle(1,7);

bar(getmaxx()/2-222,105,getmaxx()/2+222,335);

setfillstyle(1,8);

bar(getmaxx()/2-222,55,getmaxx()/2-172,100); //MATHUDA

bar(getmaxx()/2-162,55,getmaxx()/2-122,100);

bar(getmaxx()/2+122,55,getmaxx()/2+162,100);

bar(getmaxx()/2+172,55,getmaxx()/2+222,100);

setcolor(WHITE);

outtextxy(100,50,"<<");

outtextxy(505,50,">>");

settextstyle(10,0,3);

outtextxy(170,50,"<");

outtextxy(455,50,">");

setcolor(BLACK);

setfillstyle(1,7);

for(int x1=30;x1<=390;x1+=60)

{

for(int y1=20;y1<=170;y1+=30)

{

rectangle(85+x1,95+y1,135+x1,155+y1);

}

}

settextstyle(1,0,2);

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

{

if(g==0) setcolor(4);

else setcolor(0);

outtextxy(x,y,day[g]);

x+=60;

}

}

void input\_year()

{

mouse.init\_mouse(); mouse.hide\_mouse();

cleardevice();

char a[20], yr[10], yrout[2];

mouse.init\_mouse(); cleardevice();

settextstyle(1,0,2); setcolor(WHITE);

setfillstyle(SOLID\_FILL,BLACK);

outtextxy(getmaxx()/2-100,getmaxy()/2-20,"Enter Year:");

for(int i=0,f=1,x=0 ;i<4&&f==1;)

{ char al = getch();

if(al>='0'&&al<='9'){ yr[i++] = al; yrout[0] = al; yrout[1] = '\0'; outtextxy( 340+x, 220,yrout); x+=20;}

else if(al==13) f=0;

else if(al==8&&i!=0){i--; bar(320+x,210,getmaxx(),getmaxy()); x-=20;}

}

yr[i] = '\0'; year=atoi(yr); delay(100);

cleardevice();

findday();

mouse.init\_mouse(); mouse.show\_mouse();

}

void publicholiday()

{

int y=80;char line[200];

mouse.init\_mouse();

mouse.hide\_mouse();

cleardevice();

char file[12][20] ={"JAN","FEB","MARCH","APRIL","MAY","JUNE","JULY","AUG","SEP","OCT","NOV","DEC"};

char month[12][20]={"JANUARY","FEBRUARY","MARCH","APRIL","MAY","JUNE","JULY","AUGUST","SEPTEMBER","OCTOBER","NOVEMBER","DECEMBER"};

setbkcolor(BLACK); setcolor(WHITE); settextstyle(1,0,4);

outtextxy(getmaxx()/2-strlen(month[mon-1])\*10,30,month[mon-1]);

settextstyle(1,0,1); setcolor(WHITE);

strcat(file[mon-1],".txt");

ifstream fin(file[mon-1],ios::in);

fin.getline(line,80,'\n');

while(!fin.eof())

{

outtextxy(getmaxx()/2-300,y,line);

fin.getline(line,100,'\n');

y+=20;

}

fin.close();

checkpublic\_holiday();

findday(); mouse.init\_mouse(); mouse.show\_mouse();

}

void checkpublic\_holiday()

{

setcolor(WHITE);

settextstyle(3,0,1);

outtextxy(getmaxx()/2-100,getmaxy()-30,"Press ENTER to return");

int kb;

do

{ kb = getch();

}while(kb!=13);

}

**BROSCALC**

int count,maxx,maxy,flag,k,y,z,j,n,rad=1,help=-1,pre=-1,h;

char calc[13],mem[13],inexp[30],postexp[30],stack[15],degrad[5]="RAD";

char name[26][5]={"=",".","DEL","CLR","x",{251},"1","2","3","+","^","4","5","6","-",{'x',253},"7","8","9","X","COS","SIN","0","!","/","\0"};

char guide[10][200]={"1. The Calc takes care of syntax.","2. Press the black button to switch between Deg. and Rad.","3. The maximum number of characters that can be entered is 10.","4. M-SAVE saves the current result/expession.","5. M-CDIS clears screen and copies saved result/expression to it." ,"6. Click the HELP button again to hide/show this list.","7. Click the PRE button to view last 5 calculations/return to Calc.","8. Click the = button to calculate.","9. If there is any runtime error the Calc resets to 0.","10. Click the X to return to brOS Home."};

class store

{

public: char exp[20];

char res[20];

store();

}data[10];

store::store()

{

for(int i=0;i<20;i++) exp[i]=res[i]='\0';

}

void button(int x,int y, int z=15)

{

if(count!=4)

{

setfillstyle(SOLID\_FILL,7);

bar(x+4,y+4,x+49,y+59);

setcolor(z);

settextstyle(1,0,2);

outtextxy(x+26-strlen(name[count])\*5.6,y+20,name[count]);

}

else

{

setfillstyle(SOLID\_FILL,RED);

bar(x+4,y+4,x+49,y+59);

setcolor(15);

settextstyle(1,0,2);

outtextxy(x+21,y+17,name[count]);

}

}

void draw()

{

count=0;

int maxx=getmaxx(),maxy=getmaxy();

::maxx=maxx;

::maxy=maxy;

setfillstyle(SOLID\_FILL,8);

bar(maxx/2-152,1,maxx/2+152,maxy);

setfillstyle(SOLID\_FILL,LIGHTGRAY);

bar(30,100,120,130);

bar(maxx-120,100,maxx-30,130);

bar(30,190,120,220);

bar(maxx-120,190,maxx-30,220);

setcolor(WHITE);

setfillstyle(SOLID\_FILL,BLACK);

settextstyle(1,0,1);

outtextxy(55,102,"HELP");

outtextxy(43,192,"M-SAVE");

outtextxy(maxx-90,102,"PRE");

outtextxy(maxx-107,192,"M-CDIS");

bar(maxx/2-140,40,maxx/2+140,115);

setfillstyle(SOLID\_FILL,7);

bar(maxx/2-145,5,maxx/2+145,25);

bar(maxx/2-145,maxy-10,maxx/2+145,maxy);

setfillstyle(SOLID\_FILL,BLACK);

bar(maxx/2+125,8,maxx/2+140,22);

setfillstyle(SOLID\_FILL,WHITE);

bar(maxx/2+130,12,maxx/2+135,18);

settextstyle(3,0,1);

outtextxy(maxx/2-20,1,"brOS");

settextstyle(2,0,4);

setfillstyle(SOLID\_FILL,DARKGRAY);

for(int y=125;y+68<maxy;)

{

for(int x=maxx/2-145;x<maxx/2+145;)

{

button(x,y);

x+=59;

count++;

}

y+=68;

}

}

int input()

{

int xpos=6,ypos=6;

i.x.ax=3;

int86(0x33,&i,&o);

int cx=o.x.cx,dx=o.x.dx,bx=o.x.bx;

if(bx==1)

{

if(cx>maxx/2-145&&cx<maxx/2-92) xpos=0;

else if(cx>maxx/2-86&&cx<maxx/2-33) xpos=1;

else if(cx>maxx/2-27&&cx<maxx/2+26) xpos=2;

else if(cx>maxx/2+32&&cx<maxx/2+85) xpos=3;

else if(cx>maxx/2+91&&cx<maxx/2+144) xpos=4;

if(dx>130&&dx<195) ypos=0;

else if(dx>200&&dx<265) ypos=1;

else if(dx>270&&dx<335) ypos=2;

else if(dx>340&&dx<405) ypos=3;

else if(dx>410&&dx<475) ypos=4;

if(cx>maxx/2+125&&cx<maxx/2+140&&dx>8&&dx<22)

{

xpos=0;

ypos=10;

}

if(cx>30&&cx<120)

{ if(dx>100&&dx<130)

{

xpos=25;

ypos=10;

}

else if(dx>190&&dx<220)

{

xpos=5;

ypos=8;

}

}

else if(cx>maxx-120&&cx<maxx-30)

{

if(dx>100&&dx<130)

{

xpos=20;

ypos=10;

}

else if(dx>190&&dx<220)

{

xpos=10;

ypos=10;

}

}

}

if(xpos!=6&&ypos!=6) flag=1;

else flag=0;

return xpos+5\*ypos;

}

int con0(char x)

{

if(x=='.') return 0;

if(isdigit(x)==0) return 1;

return 0;

}

void checksymbol()

{

if(k==0)

{

if(count==0||count==1||count==2||count==9||count==10||count==15||count==19||count==23||count==24)

count=26;

}

if(count==0)

{

if(calc[k-1]=='.'||calc[k-1]==char(251)||calc[k-1]=='+'||calc[k-1]=='^'||calc[k-1]=='-'||calc[k-1]=='X'||calc[k-1]=='C'||calc[k-1]=='S'||calc[k-1]=='/')

count=27;

}

if(count==1)

{

int flag1=0,flag =0,i=k-1;

while(flag==0&&flag1==0&&i>=0)

{

if(i>=1) flag1=con0(calc[i-1]);

else flag1=1;

if(calc[i]=='.') flag=1;

i--;

}

if(flag==1) count=26;

}

if((count==5||count==20||count==21)&&k!=0)

{

if(isdigit(calc[k-1])!=0&&calc[k-1]!='\0')

calc[k++]='X';

if(calc[k-1]=='.')

calc[k-1]='X';

}

if((count==5||count==6||count==7|| count==8||count==11||count==12||count==13||count==16||count==17||count==18||count==20||count==21||count==22)&&k!=0)

{

int flag=0;

while(flag==0)

if(calc[k-1]=='!'||calc[k-1]==char(253))

{

calc[k]='X';

k++;

}

else flag=1;

}

if(count==5||count==15||count==20||count==21)

{

int flag=0;

while(!flag)

{

if(calc[k-1]=='C'||calc[k-1]=='S'||calc[k-1]==char(253)||calc[k-1]==char(251)||calc[k-1]=='^')

calc[k--]='\0';

else flag=1;

}

}

if(count==1||count==15||count==23)

{

int flag=0;

while(!flag)

{

if(isdigit(calc[k-1])==0&&k>0)

calc[k--]='\0';

else flag=1;

}

if(k==0)

{

checksymbol();

if(count==26) calc[k]='\0';

}

}

if(count==14)

{

int flag=0;

while(flag==0)

{

if(calc[k-1]=='+'||calc[k-1]=='.'||calc[k-1]==char(251))

{

calc[k]='\0';

k--;

}

else if(calc[k-1]=='-')

{

calc[k]='\0';

calc[--k]='+';

count=9;

}

else flag=1;

}

}

if(count==1||count==9||count==10||count==15||count==19||count==23||count==24)

{

int flag=0;

while(flag==0&&k>0)

if(calc[k-1]=='+'||calc[k-1]=='-'||calc[k-1]=='X'||calc[k-1]=='/'||calc[k-1]=='C'||calc[k-1]=='S'||calc[k-1]==char(251)||calc[k-1]=='^'||calc[k-1]=='.')

{

calc[k]='\0';

k--;

}

else flag=1;

if(k==0)

{

checksymbol();

if(count==26) calc[k]='\0';

}

}

}

int display(int pos)

{

count=pos;

if(k>=11)

{

strcpy(calc,"ERROR!");

calc[6]='\0';

setfillstyle(SOLID\_FILL,BLACK);

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

{

bar(maxx/2-140,40,maxx/2+140,115);

delay(300);

setcolor(WHITE);

settextstyle(0,0,3);

outtextxy(maxx/2+140-24\*6,80,calc);

delay(300);

}

return 1;

}

if(pos==45)

{

for(int i=0;i<13;i++) mem[i]='\0';

strcpy(mem,calc);

delay(200);

return 4;

}

if(pos==50)

{

if(rad==1) strcpy(degrad,"DEG");

else strcpy(degrad,"RAD");

rad\*=-1;

setfillstyle(SOLID\_FILL,BLACK);

bar(maxx/2+110,40,maxx/2+140,60);

settextstyle(2,0,4);

outtextxy(maxx/2+120,40,degrad);

delay(200);

return 4;

}

if(pos==60)

{

for(int i=0;i<13;i++) calc[i]='\0';

strcpy(calc,mem);

k=strlen(calc);

setfillstyle(SOLID\_FILL,BLACK);

bar(maxx/2-140,40,maxx/2+140,115);

setcolor(WHITE);

settextstyle(0,0,3);

outtextxy(maxx/2+140-24\*k,80,calc);

settextstyle(2,0,4);

outtextxy(maxx/2+120,40,degrad);

delay(200);

return 4;

}

if(pos==70)

{

if(pre==-1)

{

cleardevice();

setfillstyle(SOLID\_FILL,LIGHTGRAY);

bar(maxx-120,100,maxx-30,130);

settextstyle(1,0,1);

outtextxy(maxx-90,102,"PRE");

outtextxy(5,5,"Previous Calculations: ");

settextstyle(3,0,1);

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

{

outtextxy(5,40+70\*i,"Expression: ");

outtextxy(120,40+70\*i,data[i].exp);

outtextxy(5,70+70\*i,"Result: ");

outtextxy(120,70+70\*i,data[i].res);

}

}

else

{

cleardevice();

draw();

setcolor(WHITE);

settextstyle(0,0,3);

outtextxy(maxx/2+140-24\*k,80,calc);

settextstyle(2,0,4);

outtextxy(maxx/2+120,40,degrad);

}

pre\*=-1;

delay(200);

return 4;

}

if(pos==75)

{

if(help==-1)

{

cleardevice();

setfillstyle(SOLID\_FILL,LIGHTGRAY);

bar(30,100,120,130);

settextstyle(1,0,1);

outtextxy(55,102,"HELP");

settextstyle(3,0,1);

for(int i=0;i<10;i++)

outtextxy(5,140+30\*i,guide[i]);

}

else

{

cleardevice();

draw();

setcolor(WHITE);

settextstyle(0,0,3);

outtextxy(maxx/2+140-24\*k,80,calc);

settextstyle(2,0,4);

outtextxy(maxx/2+120,40,degrad);

}

help\*=-1;

delay(200);

return 4;

}

if(pos==100)

{

strcpy(calc,"brOS");

calc[4]='\0';

setfillstyle(SOLID\_FILL,BLACK);

bar(maxx/2-140,40,maxx/2+140,115);

setcolor(15);

settextstyle(0,0,4);

outtextxy(maxx/2-55,65,calc);

settextstyle(2,0,4);

outtextxy(maxx/2+120,40,degrad);

calc[0]=calc[1]=calc[2]=calc[3]='\0';

return 4;

}

if(pos==4)

{

button(maxx/2-145+pos%5\*59,125+68\*(pos/5),0);

delay(200);

button(maxx/2-145+pos%5\*59,125+68\*(pos/5));

return 0;

}

if(pos==3)

{

button(maxx/2-145+pos%5\*59,125+68\*(pos/5),0);

delay(200);

button(maxx/2-145+pos%5\*59,125+68\*(pos/5));

return 1;

}

if(pos==2&&k!=0)

{

button(maxx/2-145+pos%5\*59,125+68\*(pos/5),0);

delay(200);

button(maxx/2-145+pos%5\*59,125+68\*(pos/5));

return 2;

}

if(pos==0)

{

checksymbol();

if(count!=26&&count!=27)

{

button(maxx/2-145+pos%5\*59,125+68\*(pos/5),0);

delay(200);

button(maxx/2-145+pos%5\*59,125+68\*(pos/5));

return 5;

}

else if(count==26) return 1;

else return 2;

}

if(flag==1)

{

checksymbol();

pos=count;

if(pos!=15&&pos!=26) calc[k++]=name[pos][0];

else if(pos==15) calc[k++]=name[pos][1];

setfillstyle(SOLID\_FILL,BLACK);

bar(maxx/2-140,40,maxx/2+140,115);

setcolor(WHITE);

settextstyle(0,0,3);

outtextxy(maxx/2+140-24\*k,80,calc);

settextstyle(2,0,4);

outtextxy(maxx/2+120,40,degrad);

if(pos!=26)

{

button(maxx/2-145+pos%5\*59,125+68\*(pos/5),0);

delay(200);

button(maxx/2-145+pos%5\*59,125+68\*(pos/5));

}

return 4;

}

return 100;

}

int value(char x,char y=' ')

{

if(x=='+') return 1;

if(x=='-'&&(y=='^'||y=='X'||y=='C'||y=='S'||y=='/')) return 5;

else if(x=='-') return 1;

if(x=='X'||x=='/') return 2;

if(x=='^'||x==char(253)) return 3;

if(x=='C'||x=='S'||x=='!'||x==char(251)) return 4;

if(x=='\_') return 5;

if(x=='\0') return 0;

return 100;

}

int con3(char x)

{

if(x=='.') return 1;

if(x==' ') return 1;

if(isdigit(x)!=0) return 1;

return 0;

}

int con2(char x, char y)

{

if(x=='.') return 0;

if(isdigit(x)==0)

if(isdigit(y)!=0) return 1;

return 0;

}

int con1(char x,char y)

{

if(isdigit(x)!=0)

{

if(y=='.') return 0;

if(isdigit(y)==0) return 1;

}

return 0;

}

void postfix()

{

if(h==5)

{

for(int i=1;i<5;i++)

{

strcpy(data[i-1].exp,data[i].exp);

strcpy(data[i-1].res,data[i].res);

}

h--;

}

strcpy(data[h].exp,calc);

k++;

for(int i=0;i<30;i++) inexp[i]=postexp[i]='\0';

y=j=z=0;

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

{

if(i>0)

{

int b=con1(calc[i],calc[i-1]), c=con2(calc[i],calc[i-1]);

if(b) inexp[j++]=' ';

else if(c) inexp[j++]=' ';

}

else inexp[j++]=' ';

inexp[j++]=calc[i];

}

inexp[j]=';';

inexp[j+1]='~';

for(i=0;inexp[i]!='~';i++)

{

int d=con3(inexp[i]);

if(d)

{

postexp[z++]=inexp[i];

}

else if(inexp[i]!=';')

{

int p,q;

if(i>1)p=value(inexp[i],inexp[i-1]);

else if(i==1&&inexp[i]=='-') p=5;

else p=4;

if(p==5&&inexp[i]=='-') inexp[i]='\_';

if(y==0) q=0;

else q=value(stack[y-1]);

if(p>q)

{

stack[y++]=inexp[i];

}

else

{

while(y>0&&p<=q)

{

postexp[z++]=stack[y-1];

y--;

stack[y]='\0';

if(y==0) q=0;

else q=value(stack[y-1]);

}

stack[y++]=inexp[i];

}

}

else

{

for(;y>0;y--)

{

postexp[z++]=stack[y-1];

stack[y-1]='\0';

}

}

}

}

float \_(float n)

{

return -n;

}

float S(float n)

{

if(rad==-1) n=n\*3.1415/180;

return sin(n);

}

float C(float n)

{

if(rad==-1) n=n\*3.1415/180;

return cos(n);

}

float SQRT(float n)

{

return sqrt(n);

}

int fact(int n)

{

int s=1;

for(;n>0;n--)

s\*=n;

return s;

}

float power(float n1,float n2)

{

return pow(n1,n2);

}

float sum(float n1,float n2)

{

return n1+n2;

}

float dif(float n1,float n2)

{

return n1-n2;

}

float mul(float n1,float n2)

{

return n1\*n2;

}

float div(float n1,float n2)

{

if(n2!=0) return n1/n2;

else return 0;

}

void calculate()

{

float result[10]={0};

n=0;

char temp[15]={'\0'};

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

{

int x=0;

if(isdigit(postexp[i])!=0)

{

while(isspace(postexp[i])==0)

{

temp[x]=postexp[i];

x++;

i++;

}

result[n]=strtod(temp,NULL);

n++;

for(x=0;x<15;x++)

temp[x]='\0';

}

else if(postexp[i]!=' ')

{

switch(postexp[i])

{

case '\_':result[n-1]=\_(result[n-1]);break;

case char(251):result[n-1]=sqrt(result[n-1]);break;

case 'C':result[n-1]=C(result[n-1]);break;

case 'S':result[n-1]=S(result[n-1]);break;

case '!':result[n-1]=fact(result[n-1]);break;

case char(253):result[n-1]=power(result[n-1],2);break;

case '^':result[n-2]=power(result[n-2],result[n-1]);result[n-1]=0;n--;break;

case '+':result[n-2]=sum(result[n-2],result[n-1]);result[n-1]=0;n--;break;

case '-':result[n-2]=dif(result[n-2],result[n-1]);result[n-1]=0;n--;break;

case 'X':result[n-2]=mul(result[n-2],result[n-1]);result[n-1]=0;n--;break;

case '/':result[n-2]=div(result[n-2],result[n-1]);result[n-1]=0;n--;break;

}

}

}

for(k=0;k<13;k++)

calc[k]=temp[k]='\0';

double m=result[0];

if(m<0)

{

calc[0]='-';

m\*=-1;

}

m\*=1000;

if(m-int(m)>4/9.0)m++;

m/=1000;

ltoa(m,temp,10);

strcat(calc,temp);

if(strlen(calc)<6)

{

for(k=0;k<13;k++) temp[k]='\0';

strcat(calc,".");

ltoa((m-int(m))\*100,temp,10);

int s=(m-int(m))\*100;

for(k=0;s!=0;s/=10) k++;

for(;k<2;k++) strcat(calc,"0");

strcat(calc,temp);

k=strlen(calc);

}

else for(i=0;i<13;i++) calc[i]='\0';

strcpy(data[h].res,calc);

if(h!=5) h++;

}

void broscalc()

{

setbkcolor(BLACK);

int x = 1;

cleardevice();

draw();

count=0;

i.x.ax=0;

int86(0x33,&i,&o);

i.x.ax=1;

int86(0x33,&i,&o);

draw();

display(100);

while(x)

{

x=display(input());

delay(80);

if(x==1)

{

for(k=0;k<13;k++) calc[k]='\0';

k=0;

display(25);

k--;

}

if(x==2)

{

k--;

display(25);

k--;

}

if(x==5)

{

postfix();

calculate();

setfillstyle(SOLID\_FILL,BLACK);

bar(maxx/2-140,40,maxx/2+140,115);

setcolor(WHITE);

settextstyle(0,0,3);

outtextxy(maxx/2+140-24\*k,80,calc);

settextstyle(2,0,4);

outtextxy(maxx/2+120,40,degrad);

}

}

}

**BROSCLOC**

char \*month[]={"Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"};

void Box(int x,int y,int x1,int y1,int width,int color,int color1)

{ int i;

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

{ sc(color);line(x+i,y+i,x+i,y1-i);line(x+i,y+i,x1-i,y+i);

sc(color1);line(x+i,y1-i,x1-i,y1-i);line(x1-i,y+i,x1-i,y1-i);

}

}

int quit()

{ int x, y, b;

click(&x,&y,&b);

if((b&1)==1&&x<=30&&y<=30)

{ delay(50);

cleardevice();

return 1;

}

else return 0;

}

void clockone(int &midx, int &midy, int &i, int &P, int &Q, char ss[])

{

midx = getmaxx()/2;

initmouse();

setfillstyle(SOLID\_FILL,RED);

bar(0,0,30,30);

sc(WHITE);

settextstyle(3,0,1);

outtextxy(11,1,"x");

midy = getmaxy()/2;

setbkcolor(BLUE);

setfillstyle(SOLID\_FILL,1);

bar(0,280,2\*midx,2\*midy);

sc(8);

setfillstyle(1,8);

fillellipse(midx, midy, 207, 207);

setfillstyle(1,14);

fillellipse(midx, midy, 200, 200);

for(i=0;i<12;i++)

{

pieslice(midx, midy, i\*30, (i+1)\*30, 200);

}

sc(14);

fillellipse( midx , midy , 185 , 185 );

settextstyle(SANS\_SERIF\_FONT, HORIZ\_DIR, 3);

for(i=1;i<=12;i++)

{

sc(1); //sprintf(ss,"%d",i);

itoa(i, ss, 10);

if(i==5)

{ P = 25; Q = -11; }

else if(i==1)

{ P = 2; Q = -8; }

else if(i==2)

{ P = 10; Q = -15; }

else if(i==3)

{ P = 15; Q = -15; }

else if(i==4)

{ P = 14; Q = -18; }

else if(i==6)

{ P = 27; Q = -3; }

else if(i==10)

{ P = 10; Q = 5; }

else if(i==12)

{ P = 3; Q = -14; }

else if (i==11)

{ P = 3; Q = -5; }

else if(i==7)

{ P=25; Q=0; }

else if(i==9)

{ P = 14; Q = 8; }

else if(i==8)

{ P = 16; Q = 6; }

outtextxy(midx+185\*sin(PI\*30\*i/180)+Q, midy-185\*cos(PI\*30\*i/180)-P, ss);

}

delay(10);

}

void clock()

{ cleardevice();

int midx, midy, i, P, Q;

char ss[90];

clockone(midx, midy, i, P, Q, ss);

showmouseptr();

while(!quit())

{

settextstyle(SANS\_SERIF\_FONT, HORIZ\_DIR, 1);

sc(8);

outtextxy(midx-16, midy-100, "brOS");

textbackground(14);

struct dosdate\_t date;

\_dos\_getdate(&date);

itoa(int(date.year),ss,10);

sc(14); outtextxy( midx-22 , 5 , ss );

settextstyle(3, 0, 1); sc(8);

itoa(int(date.day),ss,10);

if((int(date.day))/10==0)

{ ss[1] = '\0';

for(int i = 1; i>=0; i--)

{ ss[i+1] = ss[i];

}

ss[0] = '0';

}

outtextxy(293,midy-151,ss);

outtextxy(323,midy-151,month[date.month-1]);

int d;

d = date.dayofweek + 1;

char dayweek[10]={'\0'};

if(d==1) strcpy(dayweek,"Sunday");

else if(d==2) strcpy(dayweek,"Monday");

else if(d==3) strcpy(dayweek,"Tuesday");

else if(d==4) strcpy(dayweek,"Wednesday");

else if(d==5) strcpy(dayweek,"Thursday");

else if(d==6) strcpy(dayweek,"Friday");

else if(d==7) strcpy(dayweek,"Saturday");

outtextxy(midx-10\*strlen(dayweek)/2, 320, dayweek);

Box(midx-34, midy-150, midx+42, midy-125, 1, 8, 1);

struct dostime\_t t; \_dos\_gettime(&t);

setfillstyle(1,14); bar(midx+120, midy-9, midx+142, midy+10);

if(t.minute==0&&t.hour==0&&t.second==0) bar(midx-34, midy-150, midx+42, midy-125);

if(t.second==0) bar(midx+90, midy-9, midx+112, midy+10);

if(t.second==0&&t.minute==0) bar(midx+60, midy-9, midx+82, midy+9);

sc(8);

itoa(int(t.hour),ss,10);

if(int(t.hour)/10==0)

{ for(int i = strlen(ss), l = strlen(ss); i >=0; i--)

ss[i] = ss[i-1];

ss[0]='0';

ss[l+1] = '\0';

}

ss[strlen(ss)+1]='\0';

ss[strlen(ss)]=':';

outtextxy(midx+60, midy-12, ss);

itoa(int(t.minute),ss,10);

if(int(t.minute)/10==0)

{ for(int i = strlen(ss), l = strlen(ss); i >=0; i--)

ss[i] = ss[i-1];

ss[0]='0';

ss[l+1] = '\0';

}

ss[strlen(ss)+1]='\0';

ss[strlen(ss)]=':';

outtextxy(midx+90,midy-12,ss);

itoa(int(t.second),ss,10);

if(int(t.second)/10==0)

{ for(int i = strlen(ss), l = strlen(ss); i >=0; i--)

ss[i] = ss[i-1];

ss[0]='0';

ss[l+1] = '\0';

}

ss[strlen(ss)]='\0';

outtextxy(midx+120,midy-12,ss);

Box(midx+58, midy-10, midx+145, midy+11, 1, 8, 1);

sc(14);

line( midx ,midy , midx+sin(PI\*6\*(t.second-1)/180)\*R , midy-cos(PI\*6\*(t.second-1)/180)\*R );

if(t.second%10==0){ for(i=-3;i<=3;i++)

line(midx+i, midy+i, midx+sin(PI\*(6\*t.minute+(t.second-10)/10)/180)\*(R-20), midy-cos(PI\*(6\*t.minute+(t.second-10)/10)/180)\*(R-20));

}

if(t.second==0)

{ for(i=-4;i<5;i++)

line(midx+i, midy+i, midx+sin(PI\*(30\*t.hour+(t.minute-2)/2)/180)\*(R-70), midy-cos(PI\*(30\*t.hour+(t.minute-2)/2)/180)\*(R-70));

}

sc(4);

line(midx, midy, midx+sin(PI\*6\*t.second/180)\*R, midy-cos(PI\*6\*t.second/180)\*R);

sc(8);

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

line(midx+i, midy+i, midx+sin(PI\*(6\*t.minute+t.second/10)/180)\*(R-20), midy-cos(PI\*(6\*t.minute+t.second/10)/180)\*(R-20));

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

line(midx+i, midy+i, midx+sin(PI\*(30\*t.hour+t.minute/2)/180)\*(R-70), midy-cos(PI\*(30\*t.hour+t.minute/2)/180)\*(R-70));

setfillstyle(1,4);fillellipse( midx , midy , 5 , 5 );

delay(100);

}

initmouse();

}

**FTEST**

#define MAX\_INPUT\_LEN 79

char inputbuf[MAX\_INPUT\_LEN+1];

int x=3,tr=20,input\_pos=0,bmk=0,x\_pos=0;

char na[40],t;

const int MAX\_BUF\_LEN = 79;

int MAX\_LINES\_IN\_EDIT = 100;

int sent (int);

class CNode

{

char m\_buffer[MAX\_BUF\_LEN + 1];

CNode \* m\_next;

public:

CNode (char\*);

void set (char\*);

int setEx (char\*);

void get (char\*);

int setnext (CNode\*);

void deleteNext ();

void append (CNode\*);

void print (int bRecursive);

int length ();

CNode \* getnode (int i);

void insert (CNode\*);

CNode \* read\_into\_file (char\*);

void write\_into\_file (char\*);

void display (int);

void erase();

};

CNode \*open\_file(char \*);

CNode \* pGhead= NULL;

void CNode:: get (char\* p)

{

strcpy (p,m\_buffer);

}

int CNode:: length ()

{

int i = 1;

CNode\* cur\_node = m\_next;

while (cur\_node != NULL)

{

cur\_node=cur\_node -> m\_next;

i++;

}

return i;

}

CNode \* CNode:: getnode (int i)

{

CNode \* cur\_node=this;

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

{

if(cur\_node==NULL) return cur\_node;

cur\_node=cur\_node -> m\_next;

}

return cur\_node;

}

CNode:: CNode (char \* a)

{

set (a);

m\_next=NULL;

}

void CNode:: append (CNode\* pNode)

{

CNode \* pTail=this;

while(pTail->m\_next)

{

pTail=pTail->m\_next;

}

pTail->m\_next = pNode;

}

int CNode:: setEx (char\* a)

{

int flag=0;

for (int i = MAX\_BUF\_LEN-1; i >= 0 && flag==0; i--)

{

if (m\_buffer[i] == ' ')

m\_buffer [i] = 0;

else

flag=1;

}

for (int k = MAX\_BUF\_LEN-1; k >= 0 && flag==1; k--)

{

if (a[i] == ' ')

a [i] = 0;

else

flag=0;

}

if ((strlen (m\_buffer) + strlen (a) <= MAX\_BUF\_LEN))

{

\_fstrcat (m\_buffer, a);

return i+2;

}

else

{

int kl = strlen (m\_buffer);

strncpy(m\_buffer+kl,a,MAX\_BUF\_LEN-kl);

m\_buffer[MAX\_BUF\_LEN] = '\0';

return kl;

}

}

void CNode:: set (char\* a)

{

for (int i = 0; i < MAX\_BUF\_LEN; i++)

m\_buffer[i] = ' ';

m\_buffer[MAX\_BUF\_LEN] = '\0';

if (a && strlen (a) <= MAX\_BUF\_LEN)

{

strcpy (m\_buffer, a);

m\_buffer[strlen(m\_buffer)]=' ';

}

}

int CNode:: setnext (CNode\* k)

{

if (!m\_next)

{

m\_next = k;

return 1;

}

return 0;

}

void CNode:: print (int bRecursive)

{

cout << m\_buffer << endl;

if (m\_next && bRecursive)

m\_next->print (bRecursive);

}

void CNode:: insert (CNode \* P)

{

P->m\_next=m\_next;

m\_next=P;

}

void CNode:: deleteNext ()

{

CNode\* P = m\_next;

m\_next = m\_next->m\_next;

delete P;

}

void CNode:: erase()

{

CNode \* pHead = this;

while(pHead!=NULL)

{

CNode \* P = pHead;

pHead=pHead->m\_next;

delete P;

}

}

CNode \* CNode:: read\_into\_file (char \* na)

{

char ch[99];

CNode\* pHead;

pHead=NULL;

fstream Note(na,ios::in);

while(!Note.eof())

{

Note.getline(ch,98);

ch[MAX\_INPUT\_LEN - 1]=0;

CNode\* pNode = new CNode (ch);

if(!pHead)

pHead = pNode;

else

pHead->append (pNode);

}

Note.close();

return pHead;

}

void CNode:: write\_into\_file (char \* x)

{

CNode\* pTail;

pTail = this;

fstream Note(x,ios::out);

while(pTail!=NULL)

{

Note<<pTail->m\_buffer<<'\n';

pTail = pTail-> m\_next;

}

Note.close();

}

void heading (char \* k)

{

setfillstyle(SOLID\_FILL,7);

bar(0,0,640,20);

setcolor(0);

rectangle(0,0,640,20);

rectangle(1,1,639,19);

settextstyle(DEFAULT\_FONT,0,0);

if(strlen(k)==0)

{

outtextxy(10,5,"brOS");

}

else

{

outtextxy(10,5,k);

}

setfillstyle(SOLID\_FILL,15);

bar(0,20,640,480);

setfillstyle(SOLID\_FILL,8);

bar(0,20,3,480);

}

void CNode:: display (int flag)

{

setfillstyle(SOLID\_FILL,15);

bar(0,20,640,480);

setfillstyle(SOLID\_FILL,8);

bar(0,20,3,480);

if (flag)

line(1+3,tr+8,4+3,tr+8);

CNode \* pTail = this;

int y1 = 20;

while(pTail!=NULL)

{

setcolor(0);

outtextxy(x,y1,pTail->m\_buffer);

pTail = pTail->m\_next;

y1+=8;

}

if (flag)

tr=20;

}

int first\_page ()

{

int flag=0;

setfillstyle(SOLID\_FILL,8);

bar(0,0,640,840);

settextstyle(SANS\_SERIF\_FONT,0,4);

setcolor(15);

outtextxy(250,40,"Notepad");

settextstyle(SANS\_SERIF\_FONT,0,1);

outtextxy(80,150,"If you want to use the application enter 1.");

outtextxy(80,170,"If you want to exit notepad enter 2.");

outtextxy(80,400,"Press enter key once the option is given.");

outtextxy(80,420,"To modify your option use backspace.");

while(!flag)

{

outtextxy(100,240,"Enter the option of your choice: ");

char ch[1];

ch[0]= getch();

ch[1]='\0';

outtextxy(390,240,ch);

if(getch()!=13)

{

setcolor(8);

outtextxy(390,240,ch);

setcolor(15);

}

else if(ch[0]=='2')

{

return 0;

}

else if(ch[0]=='1')

{

delay(800);

int k=1;

outtextxy(100,270,"Enter file name:") ;

outtextxy(100,290,"If you want a new file just press Enter key.");

do

{

int check=1;

for(int b=0;check;)

{

char buf=getch();

switch(buf)

{

case 8: if(b)

{

setcolor(8);

outtextxy(390,270,na);

na[--b]=0;

setcolor(15);

outtextxy(390,270,na);

}break;

case 13: check=0;break;

default: if(b<40-1 && buf>=' ' && buf<='~')

{

na[b]=buf;

na[++b]=0;

outtextxy(390,270,na);

}

}

}

na[b]='\0';

pGhead = open\_file(na);

if(pGhead==NULL)

{

outtextxy(100,320,"The file given doesn't exist.");

outtextxy(100,340,"Please enter new file name.");

delay(100);

setcolor(8);

outtextxy(390,270,na);

setcolor(15);

}

else

{

setcolor(8);

outtextxy(100,320,"The file given doesn't exist");

outtextxy(100,340,"Please enter new file name");

k=0;

setcolor(15);

outtextxy(100,320,"Enjoy Editing!");

outtextxy(100,340,"NOTE: To exit edit mode press ESCAPE KEY.");

delay(4000);

return 1;

}

}while(k);

delay(600);

flag=1;

}

else

{

setcolor(8);

outtextxy(390,240,ch);

setcolor(15);

}

}

setcolor(0);

return 0;

}

CNode\* open\_file (char\* na)

{

// If a new file then create an empty Linked List

if (na == NULL || strlen (na) == 0)

{

CNode\* pHead = new CNode ("");

for (int i = 0; i < MAX\_LINES\_IN\_EDIT; i++)

{

CNode\* N = new CNode ("");

pHead->append (N);

}

return pHead;

}

else

{

{

fstream Note(na,ios::in);

if(!Note)

return NULL;//Invalid File

}

// Coast is clear. Read in the file into the Link List...

CNode C("");

CNode \*pHead= C.read\_into\_file(na);

if (pHead)

{

int len = pHead->length ();

if (len < MAX\_LINES\_IN\_EDIT)

{ for(int i=len;i < MAX\_LINES\_IN\_EDIT;i++)

{

CNode \* e= new CNode("");

pHead->append (e);

}

}

}

return pHead;

}

}

int savefile()

{

int flag=0;

while(!flag)

{

outtextxy(80,230,"Do you want to save your file? (Y/N): ");

char ch[1];

ch[0]= getch();

ch[1]='\0';

outtextxy(540,240,ch);

if(getch()!=13)

{

setcolor(8);

outtextxy(540,240,ch);

setcolor(15);

}

else if(ch[0]=='Y'|| ch[0]=='y')

{

return 1;

}

else if(ch[0]=='N'|| ch[0]=='n')

{

return 0;

}

}

return 0;

}

int initeditor ()

{

initmouse();

i.x.ax=2;

setcolor(BLACK);

int exit\_check=first\_page();

// pGHead is Initialized

if(exit\_check)

{

heading(na);

pGhead->display (1);

while(sent(bioskey(0)));

}

else

{

return 0;

}

setfillstyle(1,8);

bar(0,0,640,480);

setcolor(15);

settextstyle(1,0,2);

int jeck=savefile();

if((na==NULL || strlen(na)==0) && jeck)

{

setfillstyle(1,8);

bar(0,0,640,480);

setcolor(15);

settextstyle(1,0,2);

outtextxy(140,200,"Save file as:");

settextstyle(1,0,2);

outtextxy(240,300,"- Notepad -");

int b = 0;

bmk=1;

do

{

int check=1;

char buf=getch();

switch(buf)

{

case 8: if(b)

{

setcolor(8);

outtextxy(300,200,na);

na[--b]=0;

setcolor(15);

outtextxy(300,200,na);

}break;

case 13: check=0;break;

default: if(b<40-1 && buf>=' ' && buf<='~')

{

na[b]=buf;

na[++b]=0;

outtextxy(300,200,na);

}

}

na[b]='\0';

if(strlen(na)==0)

{

setcolor(15);

outtextxy(200,260,"Please give a name to your file.");

delay(100);

bmk=1;

b=0;

check = 1;

}

else

{

if (!check)

{

setcolor(8);

outtextxy(200,260,"Please give a name to your file.");

setcolor(15);

outtextxy(200,260,"Your file has been saved.");

check = 1;

bmk = 0;

}

}

} while(bmk);

}

delay(150);

if(jeck)

{

pGhead->write\_into\_file(na);

}

pGhead->erase();

pGhead=NULL;

return 0;

}

int gettextwidth (char \* buff , int pos)

{

char t\_buff[MAX\_INPUT\_LEN+1];

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

{

t\_buff[i]=buff[i];

}

t\_buff[i]='\0';

return textwidth(t\_buff);

}

void moveright ()

{

if(input\_pos<78)

{

input\_pos++;

setcolor(15);

line(bmk+1+3,tr+8,bmk+4+3,tr+8);

bmk = gettextwidth(inputbuf,input\_pos);

}

}

void moveleft ()

{

if(input\_pos>0)

{

input\_pos--;

setcolor(15);

line(bmk+1+3,tr+8,bmk+4+3,tr+8);

bmk = gettextwidth(inputbuf,input\_pos);

}

}

void moveup (int &p,int &q)

{

if(tr>20)

{

setcolor(15);

line(p+1+3,q+8,p+4+3,q+8);

q-=8;

}

}

void movedown (int &p,int &q)

{

if(tr<460)

{

setcolor(15);

line(p+1+3,q,p+4+3,q);

q+=8;

}

}

int sent(int c)

{

CNode \* N= pGhead->getnode((tr-20)/8);

N->get(inputbuf);

setcolor(15);line(bmk+1+3,tr+8,bmk+4+3,tr+8);

setcolor(0);

bmk=gettextwidth(inputbuf,input\_pos);

if(char(c)==8)

{

if(input\_pos-1!=-1)

{

setcolor(15);

line(bmk+1+3,tr+8,bmk+4+3,tr+8);

outtextxy(x,tr,inputbuf);

inputbuf[--input\_pos] = 0;

setcolor(0);

bmk=textwidth(inputbuf);

inputbuf[input\_pos]=' ';

inputbuf[MAX\_INPUT\_LEN-1]='\0';

for(int i=input\_pos ; i<MAX\_INPUT\_LEN-2;i++)

inputbuf[i]=inputbuf[i+1];

outtextxy(x,tr,inputbuf);

CNode \* N= pGhead->getnode((tr-20)/8);

N->set(inputbuf);

inputbuf[input\_pos]='\0';

}

else

{

if(tr > 20)

{

tr-=8;

CNode \* N= pGhead->getnode((tr-20)/8);

input\_pos=N->setEx (inputbuf);

N->deleteNext ();

pGhead->display (0);

N->get (inputbuf);

if(input\_pos)

inputbuf[input\_pos]=0;

else

{

for(int o=MAX\_INPUT\_LEN-1;inputbuf[o]==' ';o--);

input\_pos = o+1;

}

bmk=textwidth(inputbuf);

}

}

}

else if(char(c)==13)

{

if(input\_pos>0)

{

CNode \* P=new CNode(inputbuf+input\_pos);

N->insert(P);

inputbuf[input\_pos]=0;

N->set(inputbuf);

}

tr+=8;

setcolor(15);

line(bmk+1+3,tr,bmk+4+3,tr);

input\_pos=0;

bmk=0;

pGhead -> display(0);

}

else if(char(c)==27)

{

inputbuf[0]=0;

return 0;

}

else if(c==18432) {outtextxy(x,tr,inputbuf);moveup(bmk,tr);}

else if(c==20480) {outtextxy(x,tr,inputbuf);movedown(bmk,tr);}

else if(c==19200) {outtextxy(x,tr,inputbuf);moveleft();}

else if(c==19712) {outtextxy(x,tr,inputbuf);moveright();}

else

{

char cf = c;

if ( cf >= ' ' && cf <= '~')

{

if(input\_pos<=MAX\_INPUT\_LEN-2)

{

setcolor(15);

line(bmk+1+3,tr+8,bmk+4+3,tr+8);

outtextxy(x,tr,inputbuf); setcolor(0);

char e=inputbuf[input\_pos];

inputbuf[input\_pos]=cf;

input\_pos++;

char t=inputbuf[input\_pos];

inputbuf[input\_pos]=0;bmk=textwidth(inputbuf);

inputbuf[input\_pos]=e;

inputbuf[MAX\_INPUT\_LEN-1]='\0';

if (input\_pos+1 <= MAX\_INPUT\_LEN-2)

{

for(int i=MAX\_INPUT\_LEN-2 ; i>input\_pos+1;i--)

inputbuf[i]=inputbuf[i-1];

inputbuf[input\_pos+1]=t;

}

outtextxy(x,tr,inputbuf);

CNode \* N= pGhead->getnode((tr-20)/8);

N->set(inputbuf);

inputbuf[input\_pos]=0;

}

else

{

tr+=8;input\_pos=0;

bmk = 0;

CNode \* N= pGhead->getnode((tr-20)/8);

N->get(inputbuf);

setcolor(15);

outtextxy(x,tr,inputbuf);

char e=inputbuf[input\_pos];

inputbuf[input\_pos]=cf;

input\_pos++;

char t=inputbuf[input\_pos];

inputbuf[input\_pos]=0;bmk=textwidth(inputbuf);

inputbuf[input\_pos]=e;

inputbuf[MAX\_INPUT\_LEN-1]='\0';

if (input\_pos+1 <= MAX\_INPUT\_LEN-2)

{

for(int i=MAX\_INPUT\_LEN-2 ; i>input\_pos+1;i--)

inputbuf[i]=inputbuf[i-1];

inputbuf[input\_pos+1]=t;

}

setcolor(0);

outtextxy(x,tr,inputbuf);

N= pGhead->getnode((tr-20)/8);

N->set(inputbuf);

inputbuf[input\_pos]=0;

}

}

}

setcolor(0);

line(bmk+1+3,tr+8,bmk+4+3,tr+8);

return 1;

}

**MOUSESRC**

int initmouse();

void showmouseptr();

void click(int \*, int \*, int \*);

union REGS i, o;

int initmouse()

{

i.x.ax = 0;

int86(0X33,&i,&o);

return ( o.x.ax );

}

void showmouseptr()

{

i.x.ax = 1;

int86(0X33,&i,&o);

}

void click(int \*x, int \*y, int \*button)

{

i.x.ax = 3;

int86(0X33,&i,&o);

\*button = o.x.bx;

\*x = o.x.cx;

\*y = o.x.dx;

}

**BROSNLOA**

void dot()

{

setcolor(BLUE);

setfillstyle(SOLID\_FILL,BLUE);

}

void dott()

{

setcolor(WHITE);

setfillstyle(SOLID\_FILL,WHITE);

}

void adot(int x, int y)

{

dot();

pieslice(x,y,0,360,3);

delay(80);

dott();

pieslice(x+1,y,0,360,1);

delay(110);

}

void b()

{ int xc = 150;

for(int i = 175; i < 291; i+=10)

adot(xc,i);

int x = xc+8, y = i-10;

adot(x,y);

adot(x+=9,y-=5);

adot(x+=8,y-=6);

adot(x+=7,y-=7);

adot(x+=6,y-=8);

adot(x, y-=9);

adot(x-=6,y-=8);

adot(x-=7,y-=7);

adot(x-=8,y-=6);

adot(x-=9,y-=4);

}

void r()

{ int xc = 210, yc = 233;

adot(xc,yc);

adot(xc+=7,yc+=7);

int retx = xc, rety = yc;

adot(xc+=6,yc+=7);

for(yc += 10; yc < 291; yc += 9)

adot(xc,yc);

adot(retx+=14,rety);

adot(retx+=8,rety-=6);

adot(retx+=10,rety);

adot(retx+=10,rety);

}

void O()

{ setcolor(YELLOW);

setfillstyle(SOLID\_FILL,YELLOW);

for(int i = 0; i < 60; i+=2)

{ pieslice(345,228,0,360,i);

delay(90);

}

setcolor(BLACK);

setfillstyle(SOLID\_FILL,BLACK);

pieslice(323,218,0,360,11);

bar(327,217,363,219);

pieslice(367,218,0,360,11);

bar(325,257,365,259);

}

void S()

{ int xc = 495, yc = 187;

adot(xc,yc);

adot(xc-=7,yc-=7);

adot(xc-=8,yc-=6);

adot(xc-=9,yc-=5);

adot(xc-=9,yc);

adot(xc-=9,yc+=5);

adot(xc-=8,yc+=6);

adot(xc-=7,yc+=7);

adot(xc-=5,yc+=8);

for(;xc<485;xc++) adot(xc+=6,yc+=9);

adot(xc-=7,yc+=7);

adot(xc-=8,yc+=6);

adot(xc-=9,yc+=5);

adot(xc-=9,yc);

adot(xc-=9,yc-=5);

adot(xc-=8,yc-=6);

adot(xc-=7,yc-=7);

adot(xc-=5,yc-=8);

}

void rks()

{ b();

r();

S();

O();

}

void brosn()

{

delay(1000);

rks();

}

**BROSLOAD**

void brosload()

{

initmouse();

i.x.ax = 2;

int maxx,maxy;

setbkcolor(LIGHTBLUE); setcolor(RED);

setfillstyle(SOLID\_FILL,RED);

circle(30,30,15);

floodfill(30,30,RED); //Get the image of red circle

long int s;

void \*p; s=imagesize(86,86,116,116);

p=(char \*)malloc(s);

setfillstyle(SOLID\_FILL,GREEN);

getimage(12,13,47,45,p);

cleardevice();

maxx=getmaxx();

maxy=getmaxy();

bar(1,398,maxx,maxy);

setfillstyle(SOLID\_FILL,BROWN);

bar(maxx/2-20,maxy/2-10,maxx/2+21,maxy/2+30); //Draw the Ground.

setfillstyle(SOLID\_FILL,YELLOW);

bar(maxx/2-10,maxy/2,maxx/2+10,maxy/2+20);

setfillstyle(LINE\_FILL,BROWN); //Draw the brick thing

bar(maxx/2-10,maxy/2,maxx/2+10,maxy/2+20);

setfillstyle(SOLID\_FILL,BLACK);

for(int x=1;x<maxx;x+=2)

{

setcolor(BLACK);

setfillstyle(SOLID\_FILL,BLACK);

bar(x-2,365,x+33,397);

putimage(x,364,p,1); //Make the ball roll

if(x==maxx/2-18)

{

for(int i=1;i<4;i++)

{

for(int y=365;y>maxy/2+32;y-=3) //If below the brick, Jump 3 times.

{

putimage(x,y,p,1); delay(10);

bar(x-1,y-2,x+33,y+32);

}

if(i==3)

{ setcolor(YELLOW);

setfillstyle(SOLID\_FILL,YELLOW);

pieslice(maxx/2,maxy/2-60,0,360,25);

}

setcolor(RED);

if(i==3)

{ settextstyle(0,0,1);

outtextxy(maxx/2-15,maxy/2-63,"100%");

}

setfillstyle(SOLID\_FILL,BLACK);

for(y=maxy/2+33;y<366;y+=3) //If below the brick, Jump 3 times.

{

putimage(x,y,p,1); delay(10);

bar(x-1,y-2,x+33,y+32);

}

delay(100);

}

} delay(15);

} setbkcolor(0);

}

**PACMAN**

void pac()

{

int size1,size2,size3;

int y=getmaxy()/2;

cleardevice();

settextstyle(0,0,4);

void \*p1,\*p2,\*p3;

setcolor(YELLOW);

setfillstyle(1,YELLOW);

pieslice(getmaxx()/2,getmaxy()/2,40,320,50);

size1=imagesize(getmaxx()/2-51,getmaxy()/2-51,getmaxx()/2+51,getmaxy()/2+51);

p1=malloc(size1);

getimage(getmaxx()/2-51,getmaxy()/2-51,getmaxx()/2+51,getmaxy()/2+51,p1);

cleardevice();

pieslice(getmaxx()/2,getmaxy()/2,0,360,50);

size2=imagesize(getmaxx()/2-51,getmaxy()/2-51,getmaxx()/2+51,getmaxy()/2+51);

p2=malloc(size2);

getimage(getmaxx()/2-51,getmaxy()/2-51,getmaxx()/2+51,getmaxy()/2+51,p2);

cleardevice();

setcolor(WHITE);

outtextxy(getmaxx()/2+40,getmaxy()/2-10,"++");

setcolor(BLACK);

setfillstyle(SOLID\_FILL,BLACK);

for(int x=2;x<=getmaxx()/2+55;x+=2)

{

pieslice(x-2,y,0,360,50);

if(x%20<10)putimage(x-51,y-51,p1,XOR\_PUT);

else putimage(x-51,y-51,p2,XOR\_PUT);

if(x==300) delay(500);

delay(20);

}

settextstyle(0,0,10);

setcolor(WHITE);

outtextxy(105,210,"b");

outtextxy(200,210,"r");

outtextxy(465,210,"S");

delay(500);

setcolor(BLACK);

for(;x!=getmaxx()/2-1;x+=2)

{

pieslice(x-2,getmaxy()/2,0,360,50);

if(x==getmaxx()-1) x=48;

if(x%20<10)putimage(x-51,y-51,p1,XOR\_PUT);

else putimage(x-51,y-51,p2,XOR\_PUT);

delay(20);

}

for(int i=0;i<52;i++)

{

setcolor(BLACK);

setfillstyle(SOLID\_FILL,BLACK);

pieslice(x-2,getmaxy()/2,0,360,i);

delay(15);

}

}

**CALCAPPS**

void printname()

{

int xcoord=350,ycoord=120,size=40,radius=18;

outtextxy(xcoord-9,ycoord+size+5,"CALCULATOR");

xcoord=268;

ycoord=140;

outtextxy(xcoord-13,ycoord+radius+7,"CLOCK");

xcoord=150;

ycoord=120;

outtextxy(xcoord-3,ycoord+size+5,"CALENDAR");

xcoord=450;

ycoord=120;

outtextxy(xcoord-1,ycoord+size+5,"NOTEPAD");

settextstyle(2,0,4);

outtextxy(170,230,"TO VIEW OUR GAME COLLECTION, HEAD OVER TO brOSGAME!");

}

void batman()

{

setfillstyle(1,0);

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

setcolor(YELLOW);

setfillstyle(SOLID\_FILL,YELLOW);

fillellipse(getmaxx()/2,getmaxy()/2-15,200,125);

int x=getmaxx()/2+40,y=getmaxy()/2;

setcolor(BLACK);

line(x-150,y-50,x-70,y-50);

arc(x-170,y-20,330,60,35);

arc(x-112,y+65,10,110,75);

arc(x+35,y+65,70,170,75);

arc(x+90,y-20,120,210,35);

line(x+70,y-50,x-10,y-50);

arc(x-55,y-50,180,270,15);

arc(x-25,y-50,270,0,15);

line(x-55,y-35,x-54,y-45);

line(x-25,y-35,x-26,y-45);

line(x-54,y-45,x-52,y-40);

line(x-26,y-45,x-28,y-40);

line(x-52,y-40,x-28,y-40);

setfillstyle(SOLID\_FILL,BLACK);

floodfill(x-52,y-30,BLACK);

setfillstyle(1,DARKGRAY);

setcolor(WHITE);

bar(0,0,getmaxx(),50);

settextstyle(6,0,4);

outtextxy(290,0,"brOS");

setcolor(RED);

settextstyle(2,0,4);

printname();

}

void chennai()

{

ellipse(getmaxx()/2,getmaxy()/2,0,360,300,100);

setfillstyle(SOLID\_FILL,BLUE);

fillellipse(getmaxx()/2,getmaxy()/2,300,100);

char che[7][2]={"C","H","E","N","N","A","I"};

settextstyle(0,0,13);

setcolor(YELLOW);

outtextxy(getmaxx()/2-250,190,che[0]);

for(int i=1,x=120;i<6;i++,x+=70)

{

settextstyle(0,0,7);

setcolor(YELLOW);

outtextxy(getmaxx()/2-250+x,230,che[i]);

}

outtextxy(getmaxx()/270+x+60,230,che[6]);

setfillstyle(1,14);

bar(getmaxx()/2-250+120,getmaxy()/2-40,getmaxx()/2-250+x,getmaxy()/2-30);

outtextxy(getmaxx()/270+x+60,163,".");

setfillstyle(1,DARKGRAY);

setcolor(WHITE);

bar(0,0,getmaxx(),50);

settextstyle(6,0,4);

outtextxy(290,0,"brOS");

setcolor(BLACK);

settextstyle(2,0,4);

printname();

}

void captainamerica()

{

setcolor(RED);

setfillstyle(INTERLEAVE\_FILL,RED);

pieslice(getmaxx()/2,getmaxy()/2,0,180,120);

pieslice(getmaxx()/2,getmaxy()/2,180,360,120);

setcolor(WHITE);

setfillstyle(SOLID\_FILL,WHITE);

pieslice(getmaxx()/2,getmaxy()/2,0,360,105);

setcolor(RED);

setfillstyle(INTERLEAVE\_FILL,RED);

pieslice(getmaxx()/2,getmaxy()/2,0,360,80);

pieslice(getmaxx()/2,getmaxy()/2,180,360,80);

setcolor(BLUE);

setfillstyle(SOLID\_FILL,BLUE);

pieslice(getmaxx()/2,getmaxy()/2,0,360,60);

int cx=getmaxx()/2,cy=getmaxy()/2;

int star[22]={cx,cy-60,cx+15,cy-20,cx+55,cy-20,cx+25,cy+5,cx+40,cy+45,cx,cy+20,cx-40,cy+45,cx-25,cy+5,cx-55,cy-20,cx-15,cy-20,cx,cy-60};

setcolor(BLACK);

setfillstyle(SOLID\_FILL,WHITE);

fillpoly(11,star);

setfillstyle(1,RED);

setcolor(WHITE);

bar(0,0,getmaxx(),50);

settextstyle(6,0,4);

outtextxy(290,0,"brOS");

setcolor(LIGHTGRAY);

settextstyle(2,0,4);

printname();

}

void minion()

{

int mx=getmaxx(),my=getmaxy();

setcolor(YELLOW);

setfillstyle(1,YELLOW);

bar(mx/2-80,160,mx/2+80,my-120);

pieslice(mx/2,160,0,180,81);

setcolor(BLUE);

setfillstyle(INTERLEAVE\_FILL,BLUE);

pieslice(mx/2,my-120,180,360,81);

bar(mx/2-40,my-170,mx/2+40,my-120);

bar(mx/2-80,my-180,mx/2+80,my-160);

setfillstyle(1,BLACK);

bar(mx/2-80,180,mx/2+80,190);

setcolor(LIGHTGRAY);

setfillstyle(1,LIGHTGRAY);

pieslice(mx/2,185,0,360,40);

setcolor(WHITE);

setfillstyle(1,WHITE);

pieslice(mx/2,185,0,360,30);

setcolor(BROWN);

setfillstyle(1,BROWN);

pieslice(mx/2,185,0,360,15);

setcolor(BLACK);

setfillstyle(1,BLACK);

pieslice(mx/2,185,0,360,9);

setcolor(WHITE);

setfillstyle(1,WHITE);

pieslice(mx/2-8,180,0,360,2);

setfillstyle(1,9);

setcolor(WHITE);

bar(0,0,getmaxx(),50);

settextstyle(6,0,4);

outtextxy(290,0,"brOS");

setcolor(RED);

settextstyle(2,0,4);

printname();

}

void space()

{

int mx=getmaxx(),my=getmaxy();

setcolor(WHITE);

randomize();

for(int i=0;i<1000;i++)

putpixel(random(mx),random(my),WHITE);

setfillstyle(1,GREEN);

setcolor(BLACK);

bar(0,0,getmaxx(),50);

settextstyle(6,0,4);

outtextxy(290,0,"brOS");

setcolor(WHITE);

settextstyle(2,0,4);

printname();

}

void vm()

{

int mx=getmaxx(),my=getmaxy();

setcolor(WHITE);

setfillstyle(1,WHITE);

pieslice(mx/2,my,0,180,90);

setcolor(GREEN);

setfillstyle(INTERLEAVE\_FILL,GREEN);

pieslice(mx/2,my,0,180,60);

setcolor(WHITE);

settextstyle(0,0,10);

outtextxy(120,80,"VIDYA");

setcolor(GREEN);

outtextxy(80,180,"MANDIR");

setcolor(BLACK);

settextstyle(0,0,3);

outtextxy(mx/2-22,my-30,"VM");

setfillstyle(1,WHITE);

setcolor(BLACK);

bar(0,0,getmaxx(),50);

settextstyle(6,0,4);

outtextxy(290,0,"brOS");

setcolor(WHITE);

settextstyle(2,0,4);

printname();

}

void normal()

{

setfillstyle(SOLID\_FILL,BLUE);

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

setfillstyle(1,9);

bar(0,0,getmaxx(),50);

settextstyle(6,0,4);

setcolor(BLACK);

outtextxy(290,0,"brOS");

setcolor(WHITE);

settextstyle(2,0,4);

printname();

}

void calcicon()

{

int xcoord,ycoord,size;

xcoord=350;

ycoord=120;

size=40;

setfillstyle(SOLID\_FILL,DARKGRAY);

bar(xcoord,ycoord,xcoord+size,ycoord+size);

setlinestyle(SOLID\_LINE,0,3);

setcolor(BLACK);

setfillstyle(SOLID\_FILL,RED);

bar(xcoord+size/2,ycoord+size/2,xcoord+size,ycoord+size);

line(xcoord+size/2,ycoord,xcoord+size/2,ycoord+size);

line(xcoord,ycoord+size/2,xcoord+size,ycoord+size/2);

settextstyle(6,0,1);

setcolor(LIGHTGRAY);

outtextxy(xcoord+4,ycoord-7,"+");

outtextxy(xcoord+size/2+5,ycoord-7,"-");

outtextxy(xcoord+6,ycoord+size/2-6,"x");

setcolor(WHITE);

outtextxy(xcoord+size/2+5,ycoord+size/2-5,"=");

}

void clockicon()

{

int xcoord,ycoord,radius;

xcoord=268;

ycoord=140;

radius=18;

setlinestyle(1,0,3);

setcolor(BLACK);

setfillstyle(1,BLACK);

bar(xcoord-radius-2,ycoord-20,xcoord+radius+2,ycoord+20);

setcolor(YELLOW);

setfillstyle(SOLID\_FILL,YELLOW);

pieslice(xcoord,ycoord,0,360,radius);

setlinestyle(SOLID\_LINE,0,3);

setcolor(BLACK);

line(xcoord,ycoord,xcoord+radius/1.7,ycoord+5);

line(xcoord,ycoord,xcoord-radius/1.5,ycoord+10);

setcolor(RED);

setlinestyle(SOLID\_LINE,0,1);

line(xcoord,ycoord,xcoord-5,ycoord-radius+3);

}

void mmicon()

{

int xcoord,ycoord,size;

//CHANGE THE FOLLOWING

xcoord=350;

ycoord=220;

size=40;

setfillstyle(SOLID\_FILL,BROWN);

bar(xcoord,ycoord,xcoord+size,ycoord+size);

setcolor(YELLOW);

settextstyle(2,0,6);

outtextxy(xcoord+2,ycoord,"????");

setcolor(BLUE);

setlinestyle(SOLID\_LINE,0,3);

int m[10]={xcoord+10,ycoord+size-3,xcoord+10,ycoord+size/2-5,xcoord+size/2,ycoord+size/2+10,xcoord+size-10,ycoord+size/2-5,xcoord+size-10,ycoord+size-3};

drawpoly(5,m);

}

void calenicon()

{

int xcoord,ycoord,size;

xcoord=150;

ycoord=120;

size=40;

setfillstyle(SOLID\_FILL,WHITE);

bar(xcoord,ycoord,xcoord+size,ycoord+size);

setfillstyle(SOLID\_FILL,RED);

bar(xcoord,ycoord,xcoord+size,ycoord+size/3);

setcolor(BLACK);

setlinestyle(0,0,1);

rectangle(xcoord,ycoord,xcoord+size,ycoord+size/3);

settextstyle(2,0,4);

setcolor(WHITE);

struct dosdate\_t d;

char date[3];

char month[12][4] = { "JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP","OCT","NOV","DEC" };

\_dos\_getdate(&d);

outtextxy(xcoord+size/2-7,ycoord,month[d.month-1]);

setcolor(BLACK);

settextstyle(2,0,8);

if(int(d.day)/10==0)

{ date[0] = '0';

date[1] = int(d.day) + 48;

date[2] = '\0';

}

else itoa(int(d.day),date,10);

outtextxy(xcoord+size/2-13,ycoord+size/2-7,date);

}

void evicon()

{

int xcoord,ycoord,size;

xcoord=450;

ycoord=220;

size=40;

setfillstyle(SOLID\_FILL,BLACK);

bar(xcoord,ycoord,xcoord+size,ycoord+size);

for(int i=0;i<50;i++)

putpixel(random(size-1)+xcoord+1,random(size-1)+ycoord+1,WHITE);

setfillstyle(SOLID\_FILL,GREEN);

setlinestyle(SOLID\_LINE,0,1);

setcolor(WHITE);

pieslice(xcoord+size/2,ycoord+size/2,0,180,size/4);

int body[10]={xcoord+size/4-5,ycoord+size/2,xcoord+size\*3/4+5,ycoord+size/2,xcoord+size\*3/4+8,ycoord+size/2+5,xcoord+size/4-8,ycoord+size/2+5,xcoord+size/4-5,ycoord+size/2};

setfillstyle(1,DARKGRAY);

fillpoly(5,body);

settextstyle(2,0,4);

outtextxy(xcoord+5,ycoord+size-10,"SPACE");

}

void c4icon()

{

int xcoord,ycoord,size;

xcoord=150;

ycoord=220;

size=40;

setfillstyle(SOLID\_FILL,CYAN);

bar(xcoord,ycoord,xcoord+size,ycoord+size);

settextstyle(9,0,2);

outtextxy(xcoord+5,ycoord-5,"C");

setcolor(BLACK);

settextstyle(2,0,10);

outtextxy(xcoord+size/2,ycoord,"4");

}

void noteicon()

{

int xcoord,ycoord,size;

xcoord=450;

ycoord=120;

size=40;

setfillstyle(SOLID\_FILL,LIGHTGRAY);

bar(xcoord,ycoord,xcoord+size,ycoord+size);

setfillstyle(SOLID\_FILL,YELLOW);

bar(xcoord,ycoord,xcoord+size,ycoord+size/3-3);

setlinestyle(0,0,1);

setcolor(RED);

line(xcoord+5,ycoord,xcoord+5,ycoord+size);

line(xcoord,ycoord+size/3-3,xcoord+size,ycoord+size/3-3);

setcolor(CYAN);

for(int i=0;i<5;i++) line(xcoord,ycoord+size/3+2+5\*i,xcoord+size,ycoord+size/3+2+5\*i);

}

void ralleicon()

{

int xcoord,ycoord,size;

xcoord=250;

ycoord=220;

size=40;

setfillstyle(INTERLEAVE\_FILL,GREEN);

bar(xcoord,ycoord,xcoord+size,ycoord+size);

int handle[10]={xcoord+size/2-5,ycoord+size/2+12,xcoord+2\*size/3-2,ycoord+2\*size/5,xcoord+2\*size/3+2,ycoord+2\*size/5,xcoord+size/2-3,ycoord+size/2+15,xcoord+size/2-5,ycoord+size/2+12};

setfillstyle(SOLID\_FILL,BROWN);

setcolor(BROWN);

fillpoly(5,handle);

setcolor(BLACK);

drawpoly(5,handle);

circle(xcoord+size/3,ycoord+size/4,4);

setfillstyle(SOLID\_FILL,RED);

fillellipse(xcoord+2\*size/3,ycoord+2\*size/5,10,12);

ellipse(xcoord+2\*size/3,ycoord+2\*size/5,0,360,10,12);

setfillstyle(SOLID\_FILL,WHITE);

setcolor(WHITE);

pieslice(xcoord+size/3,ycoord+size/4,0,360,4);

setcolor(BLACK);

circle(xcoord+size/3,ycoord+size/4,4);

}

void lucky()

{

int xcoord,ycoord,size;

xcoord=getmaxx()/2-80;

ycoord=312;

setfillstyle(SOLID\_FILL,WHITE);

settextstyle(2,0,6);

bar(xcoord,ycoord,xcoord+160,ycoord+40);

setcolor(BLACK);

outtextxy(xcoord+13,ycoord+9,"FEELING LUCKY?");

}

void setting()

{

int xcoord,ycoord,size;

xcoord=getmaxx()/2-60;

ycoord=372;

setfillstyle(SOLID\_FILL,DARKGRAY);

settextstyle(2,0,6);

bar(xcoord,ycoord,xcoord+120,ycoord+40);

setcolor(BLACK);

outtextxy(xcoord+22,ycoord+9,"SETTINGS");

}

void officon()

{

int xcoord,ycoord,size;

xcoord=getmaxx()/2-30;

ycoord=getmaxy()-50;

setfillstyle(SOLID\_FILL,RED);

bar(xcoord,ycoord,xcoord+60,ycoord+30);

setcolor(WHITE);

settextstyle(2,0,6);

outtextxy(xcoord+17,ycoord+5,"OFF");

}

int setting\_menu()

{

i.x.ax=3;

int86(0x33,&i,&o);

int cx=o.x.cx,dx=o.x.dx,bx=o.x.bx;

if(bx==1)

{

if(cx>getmaxx()/2-120&&dx>160&&cx<getmaxx()/2-90&&dx<210) return 1;

if(cx>getmaxx()/2+90&&dx>160&&cx<getmaxx()/2+120&&dx<210) return 2;

if(cx>getmaxx()/2-120&&dx>320&&cx<getmaxx()/2-90&&dx<370) return 3;

if(cx>getmaxx()/2+90&&dx>320&&cx<getmaxx()/2+120&&dx<370) return 4;

if(cx>getmaxx()/2-80&&dx>getmaxy()-50&&cx<getmaxx()/2+80&&dx<getmaxy()-20) return 0;

}

return 6;

}

int input\_menu()

{

randomize();

i.x.ax=3;

int86(0x33,&i,&o);

int cx=o.x.cx,dx=o.x.dx,bx=o.x.bx;

if(bx==1)

{

if(dx>120&&dx<160)

{

if(cx>150&&cx<190) return 1;

else if(cx>250&&cx<290) return 2;

else if(cx>350&&cx<390) return 3;

else if(cx>450&&cx<490) return 4;

}

else if(dx>220&&dx<260)

{

if(cx>150&&cx<190) return 5;

else if(cx>250&&cx<290) return 6;

else if(cx>350&&cx<390) return 7;

else if(cx>450&&cx<490) return 8;

}

if(cx>290&&cx<350&&dx>430&&dx<460) return 0;

if(cx>260&&cx<380&&dx>372&&dx<412) return 9;

if(cx>240&&cx<400&&dx>312&&dx<342) return random(1000)%4+1;

}

return 24;

}

void settingmenu()

{

i.x.ax=2;

int86(0x33,&i,&o);

setfillstyle(SOLID\_FILL,DARKGRAY);

for(int a=0; a<=getmaxy(); a+=2)

{

bar(0,a,getmaxx(),a+getmaxy()/30);

delay(6);

}

delay(100);

setcolor(BLACK);

settextstyle(2,0,8);

setfillstyle(SOLID\_FILL,WHITE);

bar(0,0,getmaxx(),40);

outtextxy(getmaxx()/2-58,6,"SETTINGS");

outtextxy(getmaxx()/2-72,85,"WALLPAPERS");

setlinestyle(0,0,3);

line(getmaxx()/2-150,115,getmaxx()/2+150,115);

setfillstyle(SOLID\_FILL,LIGHTGRAY);

bar(getmaxx()/2-80,160,getmaxx()/2+80,210);

bar(getmaxx()/2-120,160,getmaxx()/2-90,210);

bar(getmaxx()/2+90,160,getmaxx()/2+120,210);

setcolor(WHITE);

settextstyle(2,0,9);

outtextxy(getmaxx()/2-110,166,"<");

outtextxy(getmaxx()/2+100,166,">");

setcolor(BLACK);

settextstyle(2,0,8);

outtextxy(getmaxx()/2-110,245,"LOADING SCREENS");

setlinestyle(0,0,3);

line(getmaxx()/2-150,275,getmaxx()/2+150,275);

bar(getmaxx()/2-80,320,getmaxx()/2+80,370);

bar(getmaxx()/2-120,320,getmaxx()/2-90,370);

bar(getmaxx()/2+90,320,getmaxx()/2+120,370);

setcolor(WHITE);

settextstyle(2,0,9);

outtextxy(getmaxx()/2-110,326,"<");

outtextxy(getmaxx()/2+100,326,">");

setfillstyle(SOLID\_FILL,RED);

bar(getmaxx()/2-80,getmaxy()-50,getmaxx()/2+80,getmaxy()-20);

setcolor(WHITE);

settextstyle(2,0,6);

outtextxy(getmaxx()/2-62,getmaxy()-45,"SAVE AND EXIT");

settextstyle(2,0,8);

outtextxy(getmaxx()/2-strlen(wallpapers[wallno])\*7,170,wallpapers[wallno]);

outtextxy(getmaxx()/2-strlen(loadings[loadno])\*7,330,loadings[loadno]);

i.x.ax=1;

int86(0x33,&i,&o);

int x=6;

while(x)

{

x=setting\_menu();

switch(x)

{

case 1: if(wallno!=0) wallno--;

else wallno=6;

setfillstyle(1,LIGHTGRAY);

setcolor(WHITE);

settextstyle(2,0,8);

bar(getmaxx()/2-80,160,getmaxx()/2+80,210);

outtextxy(getmaxx()/2-strlen(wallpapers[wallno])\*7,170,wallpapers[wallno]);

delay(250);

break;

case 2: if(wallno!=6) wallno++;

else wallno=0;

setfillstyle(1,LIGHTGRAY);

setcolor(WHITE);

settextstyle(2,0,8);

bar(getmaxx()/2-80,160,getmaxx()/2+80,210);

outtextxy(getmaxx()/2-strlen(wallpapers[wallno])\*7,170,wallpapers[wallno]);

delay(250);

break;

case 3: if(loadno!=0) loadno--;

else loadno=6;

setfillstyle(1,LIGHTGRAY);

setcolor(WHITE);

settextstyle(2,0,8);

bar(getmaxx()/2-80,320,getmaxx()/2+80,370);

outtextxy(getmaxx()/2-strlen(loadings[loadno])\*7,330,loadings[loadno]);

delay(250);

break;

case 4: if(loadno!=6) loadno++;

else loadno=0;

setfillstyle(1,LIGHTGRAY);

setcolor(WHITE);

settextstyle(2,0,8);

bar(getmaxx()/2-80,320,getmaxx()/2+80,370);

outtextxy(getmaxx()/2-strlen(loadings[loadno])\*7,330,loadings[loadno]);

delay(250);

break;

case 0: delay(250);break;

}

}

}

void printh()

{

i.x.ax = 0;

int86(0x33,&i,&o);

i.x.ax = 2;

cleardevice();

switch(wallno)

{

case 1:minion();break;

case 2:batman();break;

case 3:captainamerica();break;

case 4:vm();break;

case 5:space();break;

case 6:chennai();break;

default:normal();break;

}

calcicon();

clockicon();

calenicon();

noteicon();

officon();

lucky();

setting();

delay(100);

i.x.ax=0;

int86(0x33,&i,&o);

i.x.ax=1;

int86(0x33,&i,&o);

}

void screen()

{

int app;

printh();

do{

app = input\_menu();

switch(app)

{ case 1: loads(); calendar(); refresh(); break;

case 2: loads(); clock(); refresh(); break;

case 3: loads(); broscalc(); refresh(); break;

case 4: loads(); initeditor(); refresh(); break;

case 9: settingmenu(); printh(); break;

}

if((app>=1&&app<=4)||app==9)

{ printh();

printh();

}

}while(app);

}

**CALCGAME**

void printname()

{

int xcoord=350,ycoord=120,size=40,radius=18;

outtextxy(xcoord-8,ycoord+size+5,"MASTERMIND");

xcoord=450;

ycoord=120;

outtextxy(xcoord-9,ycoord+size+5,"SPACE WARS");

xcoord=150;

ycoord=120;

outtextxy(xcoord-7,ycoord+size+5,"CONNECT-4");

xcoord=250;

ycoord=120;

outtextxy(xcoord+2,ycoord+size+5,"RALL-E");

outtextxy(172,230,"TO VIEW OUR APP COLLECTION, HEAD OVER TO brOSAPPS!");

}

void batman()

{

setfillstyle(1,0);

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

setcolor(YELLOW);

setfillstyle(SOLID\_FILL,YELLOW);

fillellipse(getmaxx()/2,getmaxy()/2-15,200,125);

int x=getmaxx()/2+40,y=getmaxy()/2;

setcolor(BLACK);

line(x-150,y-50,x-70,y-50);

arc(x-170,y-20,330,60,35);

arc(x-112,y+65,10,110,75);

arc(x+35,y+65,70,170,75);

arc(x+90,y-20,120,210,35);

line(x+70,y-50,x-10,y-50);

arc(x-55,y-50,180,270,15);

arc(x-25,y-50,270,0,15);

line(x-55,y-35,x-54,y-45);

line(x-25,y-35,x-26,y-45);

line(x-54,y-45,x-52,y-40);

line(x-26,y-45,x-28,y-40);

line(x-52,y-40,x-28,y-40);

setfillstyle(SOLID\_FILL,BLACK);

floodfill(x-52,y-30,BLACK);

setfillstyle(1,DARKGRAY);

setcolor(WHITE);

bar(0,0,getmaxx(),50);

settextstyle(6,0,4);

outtextxy(290,0,"brOS");

setcolor(RED);

settextstyle(2,0,4);

printname();

}

void chennai()

{

ellipse(getmaxx()/2,getmaxy()/2,0,360,300,100);

setfillstyle(SOLID\_FILL,BLUE);

fillellipse(getmaxx()/2,getmaxy()/2,300,100);

char che[7][2]={"C","H","E","N","N","A","I"};

settextstyle(0,0,13);

setcolor(YELLOW);

outtextxy(getmaxx()/2-250,190,che[0]);

for(int i=1,x=120;i<6;i++,x+=70)

{

settextstyle(0,0,7);

setcolor(YELLOW);

outtextxy(getmaxx()/2-250+x,230,che[i]);

}

outtextxy(getmaxx()/270+x+60,230,che[6]);

setfillstyle(1,14);

bar(getmaxx()/2-250+120,getmaxy()/2-40,getmaxx()/2-250+x,getmaxy()/2-30);

outtextxy(getmaxx()/270+x+60,163,".");

setfillstyle(1,DARKGRAY);

setcolor(WHITE);

bar(0,0,getmaxx(),50);

settextstyle(6,0,4);

outtextxy(290,0,"brOS");

setcolor(BLACK);

settextstyle(2,0,4);

printname();

}

void captainamerica()

{

setcolor(RED);

setfillstyle(INTERLEAVE\_FILL,RED);

pieslice(getmaxx()/2,getmaxy()/2,0,180,120);

pieslice(getmaxx()/2,getmaxy()/2,180,360,120);

setcolor(WHITE);

setfillstyle(SOLID\_FILL,WHITE);

pieslice(getmaxx()/2,getmaxy()/2,0,360,105);

setcolor(RED);

setfillstyle(INTERLEAVE\_FILL,RED);

pieslice(getmaxx()/2,getmaxy()/2,0,360,80);

pieslice(getmaxx()/2,getmaxy()/2,180,360,80);

setcolor(BLUE);

setfillstyle(SOLID\_FILL,BLUE);

pieslice(getmaxx()/2,getmaxy()/2,0,360,60);

int cx=getmaxx()/2,cy=getmaxy()/2;

int star[22]={cx,cy-60,cx+15,cy-20,cx+55,cy-20,cx+25,cy+5,cx+40,cy+45,cx,cy+20,cx-40,cy+45,cx-25,cy+5,cx-55,cy-20,cx-15,cy-20,cx,cy-60};

setcolor(BLACK);

setfillstyle(SOLID\_FILL,WHITE);

fillpoly(11,star);

setfillstyle(1,RED);

setcolor(WHITE);

bar(0,0,getmaxx(),50);

settextstyle(6,0,4);

outtextxy(290,0,"brOS");

setcolor(LIGHTGRAY);

settextstyle(2,0,4);

printname();

}

void minion()

{

int mx=getmaxx(),my=getmaxy();

setcolor(YELLOW);

setfillstyle(1,YELLOW);

bar(mx/2-80,160,mx/2+80,my-120);

pieslice(mx/2,160,0,180,81);

setcolor(BLUE);

setfillstyle(INTERLEAVE\_FILL,BLUE);

pieslice(mx/2,my-120,180,360,81);

bar(mx/2-40,my-170,mx/2+40,my-120);

bar(mx/2-80,my-180,mx/2+80,my-160);

setfillstyle(1,BLACK);

bar(mx/2-80,180,mx/2+80,190);

setcolor(LIGHTGRAY);

setfillstyle(1,LIGHTGRAY);

pieslice(mx/2,185,0,360,40);

setcolor(WHITE);

setfillstyle(1,WHITE);

pieslice(mx/2,185,0,360,30);

setcolor(BROWN);

setfillstyle(1,BROWN);

pieslice(mx/2,185,0,360,15);

setcolor(BLACK);

setfillstyle(1,BLACK);

pieslice(mx/2,185,0,360,9);

setcolor(WHITE);

setfillstyle(1,WHITE);

pieslice(mx/2-8,180,0,360,2);

setfillstyle(1,9);

setcolor(WHITE);

bar(0,0,getmaxx(),50);

settextstyle(6,0,4);

outtextxy(290,0,"brOS");

setcolor(RED);

settextstyle(2,0,4);

printname();

}

void space()

{

int mx=getmaxx(),my=getmaxy();

setcolor(WHITE);

randomize();

for(int i=0;i<1000;i++)

putpixel(random(mx),random(my),WHITE);

setfillstyle(1,GREEN);

setcolor(BLACK);

bar(0,0,getmaxx(),50);

settextstyle(6,0,4);

outtextxy(290,0,"brOS");

setcolor(WHITE);

settextstyle(2,0,4);

printname();

}

void vm()

{

int mx=getmaxx(),my=getmaxy();

setcolor(WHITE);

setfillstyle(1,WHITE);

pieslice(mx/2,my,0,180,90);

setcolor(GREEN);

setfillstyle(INTERLEAVE\_FILL,GREEN);

pieslice(mx/2,my,0,180,60);

setcolor(WHITE);

settextstyle(0,0,10);

outtextxy(120,80,"VIDYA");

setcolor(GREEN);

outtextxy(80,180,"MANDIR");

setcolor(BLACK);

settextstyle(0,0,3);

outtextxy(mx/2-22,my-30,"VM");

setfillstyle(1,WHITE);

setcolor(BLACK);

bar(0,0,getmaxx(),50);

settextstyle(6,0,4);

outtextxy(290,0,"brOS");

setcolor(WHITE);

settextstyle(2,0,4);

printname();

}

void normal()

{

setfillstyle(SOLID\_FILL,BLUE);

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

setfillstyle(1,9);

bar(0,0,getmaxx(),50);

settextstyle(6,0,4);

setcolor(BLACK);

outtextxy(290,0,"brOS");

setcolor(WHITE);

settextstyle(2,0,4);

printname();

}

void mmicon()

{

int xcoord,ycoord,size;

//CHANGE THE FOLLOWING

xcoord=350;

ycoord=120;

size=40;

setfillstyle(SOLID\_FILL,BROWN);

bar(xcoord,ycoord,xcoord+size,ycoord+size);

setcolor(YELLOW);

settextstyle(2,0,6);

outtextxy(xcoord+2,ycoord,"????");

setcolor(BLUE);

setlinestyle(SOLID\_LINE,0,3);

int m[10]={xcoord+10,ycoord+size-3,xcoord+10,ycoord+size/2-5,xcoord+size/2,ycoord+size/2+10,xcoord+size-10,ycoord+size/2-5,xcoord+size-10,ycoord+size-3};

drawpoly(5,m);

}

void evicon()

{

int xcoord,ycoord,size;

//CHANGE THE FOLLOWING IF REQUIRED

xcoord=450;

ycoord=120;

size=40;

setfillstyle(SOLID\_FILL,BLACK);

bar(xcoord,ycoord,xcoord+size,ycoord+size);

for(int i=0;i<50;i++)

putpixel(random(size-1)+xcoord+1,random(size-1)+ycoord+1,WHITE);

setfillstyle(SOLID\_FILL,GREEN);

setlinestyle(SOLID\_LINE,0,1);

setcolor(WHITE);

pieslice(xcoord+size/2,ycoord+size/2,0,180,size/4);

int body[10]={xcoord+size/4-5,ycoord+size/2,xcoord+size\*3/4+5,ycoord+size/2,xcoord+size\*3/4+8,ycoord+size/2+5,xcoord+size/4-8,ycoord+size/2+5,xcoord+size/4-5,ycoord+size/2};

setfillstyle(1,DARKGRAY);

fillpoly(5,body);

settextstyle(2,0,4);

outtextxy(xcoord+5,ycoord+size-10,"SPACE");

}

void c4icon()

{

int xcoord,ycoord,size;

//CHANGE THE FOLLOWING TO READJUST POSITION AND SIZE

xcoord=150;

ycoord=120;

size=40;

setfillstyle(SOLID\_FILL,CYAN);

bar(xcoord,ycoord,xcoord+size,ycoord+size);

settextstyle(9,0,2);

outtextxy(xcoord+5,ycoord-5,"C");

setcolor(BLACK);

settextstyle(2,0,10);

outtextxy(xcoord+size/2,ycoord,"4");

}

void ralleicon()

{

int xcoord,ycoord,size;

//CHANGE THE FOLLOWING TO READJUST POSITION AND SIZE

xcoord=250;

ycoord=120;

size=40;

setfillstyle(INTERLEAVE\_FILL,GREEN);

bar(xcoord,ycoord,xcoord+size,ycoord+size);

int handle[10]={xcoord+size/2-5,ycoord+size/2+12,xcoord+2\*size/3-2,ycoord+2\*size/5,xcoord+2\*size/3+2,ycoord+2\*size/5,xcoord+size/2-3,ycoord+size/2+15,xcoord+size/2-5,ycoord+size/2+12};

setfillstyle(SOLID\_FILL,BROWN);

setcolor(BROWN);

fillpoly(5,handle);

setcolor(BLACK);

drawpoly(5,handle);

circle(xcoord+size/3,ycoord+size/4,4);

setfillstyle(SOLID\_FILL,RED);

fillellipse(xcoord+2\*size/3,ycoord+2\*size/5,10,12);

ellipse(xcoord+2\*size/3,ycoord+2\*size/5,0,360,10,12);

setfillstyle(SOLID\_FILL,WHITE);

setcolor(WHITE);

pieslice(xcoord+size/3,ycoord+size/4,0,360,4);

setcolor(BLACK);

circle(xcoord+size/3,ycoord+size/4,4);

}

void lucky()

{

int xcoord,ycoord,size;

//CHANGE THE FOLLOWING TO READJUST POSITION AND SIZE

xcoord=getmaxx()/2-80;

ycoord=312;

setfillstyle(SOLID\_FILL,WHITE);

settextstyle(2,0,6);

bar(xcoord,ycoord,xcoord+160,ycoord+40);

setcolor(BLACK);

outtextxy(xcoord+13,ycoord+9,"FEELING LUCKY?");

}

void setting()

{

int xcoord,ycoord,size;

//CHANGE THE FOLLOWING TO READJUST POSITION AND SIZE

xcoord=getmaxx()/2-60;

ycoord=372;

setfillstyle(SOLID\_FILL,DARKGRAY);

settextstyle(2,0,6);

bar(xcoord,ycoord,xcoord+120,ycoord+40);

setcolor(BLACK);

outtextxy(xcoord+22,ycoord+9,"SETTINGS");

}

void officon()

{

int xcoord,ycoord,size;

//CHANGE THE FOLLOWING TO READJUST POSITION AND SIZE

xcoord=getmaxx()/2-30;

ycoord=getmaxy()-50;

setfillstyle(SOLID\_FILL,RED);

bar(xcoord,ycoord,xcoord+60,ycoord+30);

setcolor(WHITE);

settextstyle(2,0,6);

outtextxy(xcoord+17,ycoord+5,"OFF");

}

int setting\_menu()

{

i.x.ax=3;

int86(0x33,&i,&o);

int cx=o.x.cx,dx=o.x.dx,bx=o.x.bx;

if(bx==1)

{

if(cx>getmaxx()/2-120&&dx>160&&cx<getmaxx()/2-90&&dx<210) return 1;

if(cx>getmaxx()/2+90&&dx>160&&cx<getmaxx()/2+120&&dx<210) return 2;

if(cx>getmaxx()/2-120&&dx>320&&cx<getmaxx()/2-90&&dx<370) return 3;

if(cx>getmaxx()/2+90&&dx>320&&cx<getmaxx()/2+120&&dx<370) return 4;

if(cx>getmaxx()/2-80&&dx>getmaxy()-50&&cx<getmaxx()/2+80&&dx<getmaxy()-20) return 0;

}

return 6;

}

int input\_menu()

{

randomize();

i.x.ax=3;

int86(0x33,&i,&o);

int cx=o.x.cx,dx=o.x.dx,bx=o.x.bx;

if(bx==1)

{

if(dx>120&&dx<160)

{

if(cx>150&&cx<190) return 1;

else if(cx>250&&cx<290) return 2;

else if(cx>350&&cx<390) return 3;

else if(cx>450&&cx<490) return 4;

}

else if(dx>220&&dx<260)

{

if(cx>150&&cx<190) return 5;

else if(cx>250&&cx<290) return 6;

else if(cx>350&&cx<390) return 7;

else if(cx>450&&cx<490) return 8;

}

if(cx>290&&cx<350&&dx>430&&dx<460) return 0;

if(cx>260&&cx<380&&dx>372&&dx<412) return 9;

if(cx>240&&cx<400&&dx>312&&dx<342) return random(1000)%4+1;

}

return 24;

}

void settingmenu()

{

i.x.ax=2;

int86(0x33,&i,&o);

setfillstyle(SOLID\_FILL,DARKGRAY);

for(int a=0; a<=getmaxy(); a+=2)

{

bar(0,a,getmaxx(),a+getmaxy()/30);

delay(6);

}

delay(100);

setcolor(BLACK);

settextstyle(2,0,8);

setfillstyle(SOLID\_FILL,WHITE);

bar(0,0,getmaxx(),40);

outtextxy(getmaxx()/2-58,6,"SETTINGS");

outtextxy(getmaxx()/2-72,85,"WALLPAPERS");

setlinestyle(0,0,3);

line(getmaxx()/2-150,115,getmaxx()/2+150,115);

setfillstyle(SOLID\_FILL,LIGHTGRAY);

bar(getmaxx()/2-80,160,getmaxx()/2+80,210);

bar(getmaxx()/2-120,160,getmaxx()/2-90,210);

bar(getmaxx()/2+90,160,getmaxx()/2+120,210);

setcolor(WHITE);

settextstyle(2,0,9);

outtextxy(getmaxx()/2-110,166,"<");

outtextxy(getmaxx()/2+100,166,">");

setcolor(BLACK);

settextstyle(2,0,8);

outtextxy(getmaxx()/2-110,245,"LOADING SCREENS");

setlinestyle(0,0,3);

line(getmaxx()/2-150,275,getmaxx()/2+150,275);

bar(getmaxx()/2-80,320,getmaxx()/2+80,370);

bar(getmaxx()/2-120,320,getmaxx()/2-90,370);

bar(getmaxx()/2+90,320,getmaxx()/2+120,370);

setcolor(WHITE);

settextstyle(2,0,9);

outtextxy(getmaxx()/2-110,326,"<");

outtextxy(getmaxx()/2+100,326,">");

setfillstyle(SOLID\_FILL,RED);

bar(getmaxx()/2-80,getmaxy()-50,getmaxx()/2+80,getmaxy()-20);

setcolor(WHITE);

settextstyle(2,0,6);

outtextxy(getmaxx()/2-62,getmaxy()-45,"SAVE AND EXIT");

settextstyle(2,0,8);

outtextxy(getmaxx()/2-strlen(wallpapers[wallno])\*7,170,wallpapers[wallno]);

outtextxy(getmaxx()/2-strlen(loadings[loadno])\*7,330,loadings[loadno]);

i.x.ax=1;

int86(0x33,&i,&o);

int x=6;

while(x)

{

x=setting\_menu();

switch(x)

{

case 1: if(wallno!=0) wallno--;

else wallno=6;

setfillstyle(1,LIGHTGRAY);

setcolor(WHITE);

settextstyle(2,0,8);

bar(getmaxx()/2-80,160,getmaxx()/2+80,210);

outtextxy(getmaxx()/2-strlen(wallpapers[wallno])\*7,170,wallpapers[wallno]);

delay(250);

break;

case 2: if(wallno!=6) wallno++;

else wallno=0;

setfillstyle(1,LIGHTGRAY);

setcolor(WHITE);

settextstyle(2,0,8);

bar(getmaxx()/2-80,160,getmaxx()/2+80,210);

outtextxy(getmaxx()/2-strlen(wallpapers[wallno])\*7,170,wallpapers[wallno]);

delay(250);

break;

case 3: if(loadno!=0) loadno--;

else loadno=6;

setfillstyle(1,LIGHTGRAY);

setcolor(WHITE);

settextstyle(2,0,8);

bar(getmaxx()/2-80,320,getmaxx()/2+80,370);

outtextxy(getmaxx()/2-strlen(loadings[loadno])\*7,330,loadings[loadno]);

delay(250);

break;

case 4: if(loadno!=6) loadno++;

else loadno=0;

setfillstyle(1,LIGHTGRAY);

setcolor(WHITE);

settextstyle(2,0,8);

bar(getmaxx()/2-80,320,getmaxx()/2+80,370);

outtextxy(getmaxx()/2-strlen(loadings[loadno])\*7,330,loadings[loadno]);

delay(250);

break;

case 0: delay(250);break;

}

}

}

void printh()

{

i.x.ax = 0;

int86(0x33,&i,&o);

i.x.ax = 2;

cleardevice();

switch(wallno)

{

case 1:minion();break;

case 2:batman();break;

case 3:captainamerica();break;

case 4:vm();break;

case 5:space();break;

case 6:chennai();break;

default:normal();break;

}

mmicon();

evicon();

c4icon();

ralleicon();

officon();

lucky();

setting();

delay(100);

i.x.ax=0;

int86(0x33,&i,&o);

i.x.ax=1;

int86(0x33,&i,&o);

}

void screen()

{

int app; printh();

do{

app = input\_menu();

switch(app)

{

case 1: loads(); c4(); refresh(); break;

case 2: loads(); ralle(); refresh(); break;

case 3: loads(); cowsbulls(); refresh(); break;

case 4: loads(); spacew(); refresh(); break;

case 9: settingmenu(); printh(); break;

}

if((app>=1&&app<=4)||app==9)

{ printh();

printh();

}

}while(app);

}

**CONNECT 4**

int maxx1,maxy1,xc4=4,xpos,ypos;

int game[7][7],turn=1,win1,win2;

char player1[20],player2[20],guides[10][100]={"1. Enter the name of the players who are playing.","2. The computer randomly chooses the player who goes first.","3. Your turn is highlighted by the color of the marker.","4. Use the arrow keys to move the marker.","5. Use the spacebar to drop the coin.","6. You win if 4 continuous holes are filled by your coins.","7. If all the holes are filled, then the game is a draw.","8. Choose the play again option to play with the same players.","9. Press any key to go back to the main menu."};

void menu();

void drawmenu()

{

setfillstyle(SOLID\_FILL,BLUE);

bar(0,0,maxx1,maxy1);

setcolor(WHITE);

settextstyle(9,0,4);

outtextxy(maxx1/2-143,80,"CONNECT 4");

setcolor(WHITE);

setlinestyle(SOLID\_LINE,0,3);

line(maxx1/2-143,150,maxx1/2+143,150);

setfillstyle(SOLID\_FILL,CYAN);

settextstyle(9,0,3);

setcolor(WHITE);

bar(maxx1/2-125,200,maxx1/2+125,260);

outtextxy(maxx1/2-46,202,"PLAY");

bar(maxx1/2-125,280,maxx1/2+125,340);

outtextxy(maxx1/2-56,282,"RULES");

setfillstyle(SOLID\_FILL,RED);

bar(maxx1/2-100,360,maxx1/2+100,420);

outtextxy(maxx1/2-32,362,"EXIT");

}

int input2()

{

i.x.ax=3;

int86(0x33,&i,&o);

int cx=o.x.cx,dx=o.x.dx,bx=o.x.bx;

if(bx==1)

{

if(cx>maxx1/2-125&&cx<maxx1/2+125)

if(dx>170&&dx<230) return 1;

else if(dx>270&&dx<330) return 2;

else if(dx>370&&dx<430) return 0;

}

else return 4;

return 4;

}

void playmenu()

{

for(int l=0;l<10;l++)player1[l]=player2[l]='\0';

i.x.ax=2;

int86(0x33,&i,&o);

setfillstyle(SOLID\_FILL,BLACK);

bar(0,0,maxx1,maxy1);

delay(300);

setcolor(WHITE);

settextstyle(9,0,1);

outtextxy(233,10,"CONNECT 4");

setlinestyle(0,0,3);

line(233,50,403,50);

outtextxy(219,100,"Who's playing?");

setfillstyle(SOLID\_FILL,CYAN);

settextstyle(9,0,3);

bar(maxx1/2-125,170,maxx1/2+125,230);

bar(maxx1/2-125,270,maxx1/2+125,330);

setfillstyle(SOLID\_FILL,RED);

bar(maxx1/2-100,370,maxx1/2+100,430);

outtextxy(maxx1/2-50,372,"PLAY!");

i.x.ax=0;

int86(0x33,&i,&o);

i.x.ax=1;

int86(0x33,&i,&o);

int flag=1;

while(flag)

{

flag=input2();

if(flag==1)

{

for(int l=0;l<10;l++)player1[l]='\0';

setfillstyle(SOLID\_FILL,CYAN);

bar(maxx1/2-125,170,maxx1/2+125,230);

i.x.ax=2;

int86(0x33,&i,&o);

settextstyle(9,0,2);

char ch=1;

for(int i=0;i<10&&ch!=13;i++)

{

setcolor(WHITE);

setlinestyle(0,0,3);

rectangle(maxx1/2-125,170,maxx1/2+125,230);

ch=getch();

if(ch!=13&&ch!=8) player1[i]=ch;

else if(ch==8&&i!=0)

{

player1[--i]='\0';

setfillstyle(SOLID\_FILL,CYAN);

settextstyle(9,0,2);

bar(maxx1/2-125,170,maxx1/2+125,230);

i--;

}

strupr(player1);

outtextxy(maxx1/2-120,175,player1);

delay(200);

}

}

if(flag==1)

{

setcolor(BLACK);

setlinestyle(0,0,1);

rectangle(maxx1/2-125,170,maxx1/2+125,230);

i.x.ax=0;

int86(0x33,&i,&o);

i.x.ax=1;

int86(0x33,&i,&o);

}

if(flag==2)

{

for(int l=0;l<10;l++)player2[l]='\0';

setfillstyle(SOLID\_FILL,CYAN);

bar(maxx1/2-125,270,maxx1/2+125,330);

i.x.ax=2;

int86(0x33,&i,&o);

settextstyle(9,0,2);

char ch=1;

for(int i=0;i<10&&ch!=13;i++)

{

setcolor(WHITE);

setlinestyle(0,0,3);

rectangle(maxx1/2-125,270,maxx1/2+125,330);

ch=getch();

if(ch!=13&&ch!=8) player2[i]=ch;

else if(ch==8&&i!=0)

{

player2[--i]='\0';

setfillstyle(SOLID\_FILL,CYAN);

settextstyle(9,0,2);

bar(maxx1/2-125,270,maxx1/2+125,330);

i--;

}

strupr(player2);

outtextxy(maxx1/2-120,275,player2);

delay(200);

}

}

if(flag==2)

{

setcolor(BLACK);

setlinestyle(0,0,1);

rectangle(maxx1/2-125,270,maxx1/2+125,330);

i.x.ax=0;

int86(0x33,&i,&o);

i.x.ax=1;

int86(0x33,&i,&o);

}

}

randomize();

if(random(1000)%2)

{

char temp[15];

strcpy(temp,player1);

strcpy(player1,player2);

strcpy(player2,temp);

}

if(strlen(player1)==0) strcpy(player1,"PLAYER-1");

if(strlen(player2)==0) strcpy(player2,"PLAYER-2");

}

int win()

{

int flag=0,j;

for(int i=0;flag<4&&i<=7;i++)

{

if(game[i][ypos]==turn) flag++;

else flag=0;

}

if(flag==4) return 1;

flag=0;

for(i=0;flag<4&&i<=6;i++)

{

if(game[xpos][i]==turn) flag++;

else flag=0;

}

if(flag==4) return 1;

flag=0;

i=xpos+ypos-5;

j=xpos+ypos;

if(j>5) j=6;

if(i<=0) i=0;

for(;i<=j&&flag<4;i++)

{

if(game[i][xpos+ypos-i]==turn) flag++;

else flag=0;

}

if(flag==4) return 1;

flag=0;

i=xpos-ypos;

j=5+i;

if(j>6) j=6;

if(i<0) i=0;

for(;i<=j&&flag<4;i++)

{

if(game[i][i-(xpos-ypos)]==turn) flag++;

else flag=0;

}

if(flag==4) return 1;

flag=0;

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

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

if(game[i][j]==0) flag=1;

if(flag==0) return 2;

else return 0;

}

int input3()

{

i.x.ax=3;

int86(0x33,&i,&o);

int cx=o.x.cx,dx=o.x.dx,bx=o.x.bx;

if(bx==1)

{

if(dx>maxy1/2+20&&dx<maxy1/2+80)

if(cx>maxx1/2-150&&cx<maxx1/2-25) return 1;

else if(cx>maxx1/2+25&&cx<maxx1/2+150) return 0;

}

else return 4;

return 4;

}

void drawhole(int x, int y, int col)

{

setcolor(col);

setfillstyle(SOLID\_FILL,col);

pieslice(x,y,0,360,25);

setcolor(BLACK);

setlinestyle(0,0,1);

circle(x,y,23);

settextstyle(4,0,4);

outtextxy(x-7,y-23,"b");

}

void insert()

{

char ch=1;

int yay=0;

xc4=4;

while(xc4==4)

{

setfillstyle(1,BLACK);

bar(90+74\*xpos,5,125+74\*xpos,25);

if(ch==77)

{

if(xpos==6) xpos=0;

else xpos++;

}

if(ch==75)

{

if(xpos==0) xpos=6;

else xpos--;

}

if(ch==32)

{

for(ypos=0;ypos<6&&game[xpos][ypos]==0;ypos++);

ypos--;

if(ypos!=-1)

{

if(turn==1) drawhole(100+74\*xpos,62+61\*ypos,4);

else drawhole(100+74\*xpos,62+61\*ypos,14);

game[xpos][ypos]=turn;

yay=win();

turn\*=-1;

}

}

if(yay==0)

{

if(turn==1)

{

setfillstyle(1,BLACK);

setcolor(BLACK);

setlinestyle(SOLID\_LINE,0,3);

rectangle(maxx1-222,418,maxx1-48,462);

setcolor(4);

rectangle(48,418,222,462);

}

else

{

setfillstyle(1,BLACK);

setcolor(BLACK);

setlinestyle(SOLID\_LINE,0,3);

rectangle(48,418,222,462);

setcolor(14);

rectangle(maxx1-222,418,maxx1-48,462);

}

settextstyle(0,0,3);

char arrow[2]="\0\0";

arrow[0]=char(31);

outtextxy(92+74\*xpos,5,arrow);

if(ch==32) delay(300);

ch=getch();

}

else

{

delay(200);

setfillstyle(1,CYAN);

setcolor(WHITE);

settextstyle(9,0,1);

for(int g=-100;g<100;g+=5)

{

bar(maxx1/2-175,maxy1/2-100,maxx1/2+175,maxy1/2+g);

delay(50);

}

if(yay==2)

{

setcolor(BLUE);

char temp[25]=" It is a draw!";

outtextxy(maxx1/2-strlen(temp)\*7,maxy1/2-70,temp);

}

else

{

if(turn==1)

{

setcolor(YELLOW);

char temp[25];

strcpy(temp,player2);

strcat(temp," has won!");

outtextxy(maxx1/2-strlen(temp)\*7,maxy1/2-70,temp);

win2++;

}

else

{

setcolor(RED);

char temp[25];

strcpy(temp,player1);

strcat(temp," has won!");

outtextxy(maxx1/2-strlen(temp)\*8,maxy1/2-70,temp);

win1++;

}

}

setfillstyle(1,BLUE);

setlinestyle(0,0,3);

setcolor(BLACK);

bar(maxx1/2-150,maxy1/2+20,maxx1/2-25,maxy1/2+80);

rectangle(maxx1/2-150,maxy1/2+20,maxx1/2-25,maxy1/2+80);

bar(maxx1/2+25,maxy1/2+20,maxx1/2+150,maxy1/2+80);

rectangle(maxx1/2+25,maxy1/2+20,maxx1/2+150,maxy1/2+80);

setcolor(WHITE);

outtextxy(maxx1/2-138,maxy1/2+27,"AGAIN?");

outtextxy(maxx1/2+60,maxy1/2+27,"EXIT");

delay(200);

i.x.ax=0;

int86(0x33,&i,&o);

i.x.ax=1;

int86(0x33,&i,&o);

xc4=4;

while(xc4==4)

{

xc4=input3();

}

}

}

}

void drawboard()

{

i.x.ax=2;

int86(0x33,&i,&o);

setfillstyle(SOLID\_FILL,BLACK);

bar(0,0,maxx1,maxy1);

setfillstyle(SOLID\_FILL,BLUE);

delay(300);

bar(50,30,maxx1-50,400);

setcolor(BLACK);

setlinestyle(0,0,3);

line(60,37,60,392);

line(maxx1-60,37,maxx1-60,392);

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

{

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

{

drawhole(98+74\*k,62+61\*j,0);

setlinestyle(1,0,3);

if(k!=0) line(60+74\*k,40+61\*j,60+74\*k,80+61\*j);

}

}

setfillstyle(SOLID\_FILL,BLUE);

bar(50,420,220,460);

bar(maxx1-220,420,maxx1-50,460);

settextstyle(0,0,2);

setcolor(WHITE);

outtextxy(135-strlen(player1)\*8,433,player1);

outtextxy(maxx1-135-strlen(player2)\*8,433,player2);

settextstyle(9,0,1);

outtextxy(maxx1/2-20,420,"V/S");

char w1[3],w2[3];

itoa(win1,w1,10);

itoa(win2,w2,10);

outtextxy(maxx1/2-40-strlen(w1)\*7,420,w1);

outtextxy(maxx1/2+40,420,w2);

insert();

i.x.ax=1;

int86(0x33,&i,&o);

}

void play()

{

win1=win2=0;

i.x.ax=2;

int86(0x33,&i,&o);

cleardevice();

playmenu();

i.x.ax=2;

int86(0x33,&i,&o);

cleardevice();

xc4=4;

while(xc4)

{

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

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

game[i][j]=0;

drawboard();

}

i.x.ax=1;

int86(0x33,&i,&o);

i.x.ax=2;

int86(0x33,&i,&o);

cleardevice(); drawmenu();

delay(500);

xc4=4;

i.x.ax=1;

int86(0x33,&i,&o);

}

void rulesc4()

{

i.x.ax=2; int86(0x33,&i,&o);

cleardevice(); settextstyle(3,0,3);

outtextxy(5,80,"RULES");

setlinestyle(0,0,3);

line(5,110,65,110);

settextstyle(3,0,1);

for(int j=0;j<10;j++)

outtextxy(5,140+30\*j,guides[j]);

i.x.ax=1; int86(0x33,&i,&o);

getch(); i.x.ax=2;

int86(0x33,&i,&o);

cleardevice(); drawmenu();

xc4=4;

delay(500);

i.x.ax=1; int86(0x33,&i,&o);

}

int inputc4()

{

i.x.ax=3; int86(0x33,&i,&o);

int cx=o.x.cx,dx=o.x.dx,bx=o.x.bx;

if(bx==1)

{

if(cx>maxx1/2-125&&cx<maxx1/2+125)

if(dx>200&&dx<260) return 1;

else if(dx>280&&dx<340) return 2;

else if(dx>360&&dx<420) return 0;

}

else return 4;

return 4;

}

void menu()

{

xc4=4;

drawmenu();

i.x.ax=0;

int86(0x33,&i,&o);

i.x.ax=1;

int86(0x33,&i,&o);

while(xc4)

{

xc4=inputc4();

if(xc4==1) play();

else if(xc4==2) rulesc4();

}

}

void c4()

{

maxx1=getmaxx();

maxy1=getmaxy();

menu();

}

**RALLE**

int ttf = 1, wl = 1, ran, score; float m, c;

void ttmenu(); void ttdraw(); void menup();

class ballpar

{ protected: int x;

int y;

int dir;

public: void drawball(int);

};

class Ball:public ballpar

{ public: int paths[7][2];

void moveball(int&,int&,int,int,int);

void path()

{ paths[0][0] = getmaxx()/2-212; paths[0][1] = getmaxy()/2-126;

paths[1][0] = getmaxx()/2-222; paths[1][1] = getmaxy()/2-93;

paths[2][0] = getmaxx()/2-232; paths[2][1] = getmaxy()/2-45;

paths[3][0] = getmaxx()/2-234; paths[3][1] = getmaxy()/2;

paths[4][0] = getmaxx()/2-232; paths[4][1] = getmaxy()/2+45;

paths[5][0] = getmaxx()/2-222; paths[5][1] = getmaxy()/2+93;

paths[6][0] = getmaxx()/2-212; paths[6][1] = getmaxy()/2+126;

}

}ball;

void ballpar::drawball(int r)

{ pieslice(x,y,0,360,r);

}

void Ball::moveball(int &d, int &colb, int xp = -1, int yp = -1, int rb = 3)

{ char b[20];

delay(1);//2or3

if(xp!=-1) x = xp;

else x = getmaxx()/2 + 94;

if(yp!=-1) y = yp;

else y = getmaxy()/2;

dir = d; drawball(rb);

if((getpixel(x-5,y)==RED&&dir==1)||(colb==0&&x<=ball.paths[ran][0])){ dir = d = -1; if(colb==0&&getpixel(x-5,y)==RED) wl = 0; else if(colb==1) score++; else if(colb==2) score+=2; }

if(getpixel(x+5,y)==RED&&dir==-1){ dir = d = 1; colb = random(1000)%3; }

if(x<ball.paths[ran][0]-10&&wl>=1){ wl--; sound(1000); delay(10); nosound(); }

setfillstyle(SOLID\_FILL,YELLOW);

bar(getmaxx()-90,30,getmaxx(),60);

setcolor(BLACK); itoa(score,b,10);

outtextxy(getmaxx()-45-strlen(b)\*7,30,b);

}

void racket(int r, int a, int col)

{ setcolor(col);

for(int ra = r; ra<=r+3; ra++)

{ arc(getmaxx()/2+130,getmaxy()/2,a,a+4,ra);

}

if(col==4) setcolor(BROWN);

else setcolor(GREEN);

for(ra = r+1; ra<=r+2; ra++) arc(getmaxx()/2+130,getmaxy()/2,a+4,a+6,ra);

}

void ttdraw()

{ setfillstyle(SOLID\_FILL,GREEN);

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

setfillstyle(SOLID\_FILL,BLUE);

bar(getmaxx()/2-190,getmaxy()/2-145,getmaxx()/2+190,getmaxy()/2+145);

setcolor(WHITE); setlinestyle(0,0,3);

rectangle(getmaxx()/2-188,getmaxy()/2-143,getmaxx()/2+189,getmaxy()/2+144);

setcolor(LIGHTGRAY);

line(getmaxx()/2,getmaxy()/2-145,getmaxx()/2,getmaxy()/2+145);

setfillstyle(SOLID\_FILL,YELLOW);

bar(getmaxx()-90,0,getmaxx(),60);

setcolor(BLACK); settextstyle(2,0,6);

outtextxy(getmaxx()-70,2,"Score");

setfillstyle(SOLID\_FILL,RED); setcolor(RED);

pieslice(getmaxx()/2+130,getmaxy()/2,120,240,30);

setfillstyle(SOLID\_FILL,BLACK); setcolor(BLACK);

pieslice(getmaxx()/2+130,getmaxy()/2,0,360,20);

setcolor(YELLOW);

pieslice(getmaxx()/2+130,getmaxy()/2,0,360,5);

}

class hs

{ public: char n[11];

int s;

}ralh[6];

void highscore()

{

fstream f;

f.open("HSRAL",ios::in|ios::binary);

hs r;

f.read((char\*)&r,sizeof(r)); int i = 0;

while(!f.eof()&&i<5)

{ ralh[i] = r;

i++;

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

}

f.close();

hs t;

for(int j = 1; j < 6; j++)

{ t = ralh[j];

int k = j - 1;

while(k>=0&&t.s>ralh[k].s)

{ ralh[k+1] = ralh[k]; k--;

}

ralh[k+1] = t;

}

f.open("HSRAL",ios::out|ios::binary);

f.write((char\*)&ralh[0],(5)\*sizeof(ralh[0]));

f.close();

}

void hscores()

{ initmouse(); i.x.ax = 2;

cleardevice();

fstream f;

f.open("HSRAL",ios::in);

int i = 0, x = getmaxx()/2-143, y = getmaxy()/2-40;

hs a;

char sl[10];

settextstyle(3,0,1);

setcolor(GREEN);

outtextxy(x,y-20,"NAME");

setcolor(RED);

outtextxy(x+220,y-20,"SCORE");

setcolor(WHITE);

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

while(i<5)

{ outtextxy(x, y, a.n);

outtextxy(x+220, y, itoa(a.s,sl,10));

y+=20;

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

i++;

}

f.close();

settextstyle(2,0,5);

outtextxy(getmaxx()/2-80,getmaxy()-40,"Press ENTER to return");

do

{ i = getch();

}while(i!=13);

}

void rules()

{

initmouse();

i.x.ax = 2;

char rule[6][100] = { "1. Press enter to begin/end playing.", "2. Use up and down arrows to move the racket.", "3. Hit white balls to increase score by 1.", "4. Hit pink balls to increase score by 2.", "5. Do not hit the black 'gravity' balls.", "6. If you miss, you're out!"};

setfillstyle(SOLID\_FILL,BLACK);

setcolor(WHITE);

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

settextstyle(3,0,1);

outtextxy(30,40,"Rules");

line(29,61,65,61);

for(int i = 0, y = 66; i < 6; i++)

{ outtextxy(30, y,rule[i]);

y+=25;

}

int kb;

do

{ kb = getch();

}while(kb!=13);

}

void racketmove()

{

wl = 1;

int i, am = 180, db = 1, xb, yb, colb=1;

int colorb[3] = { 0, 15, 13};

xb = getmaxx()/2+94;

yb = getmaxy()/2;

racket(370,180,4);

int kb;

do

{ kb = getch();

}while(kb!=13);

while(wl!=0)

{

setcolor(WHITE);

setlinestyle(0,0,3);

rectangle(getmaxx()/2-188,getmaxy()/2-143,getmaxx()/2+189,getmaxy()/2+144);

setcolor(LIGHTGRAY);

line(getmaxx()/2,getmaxy()/2-145,getmaxx()/2,getmaxy()/2+145);

if(kbhit()) i = getch();

else i = 0;

if(i==72&&am>=156)

{ racket(370,am,2);

racket(370, am-2,4);

am-=2;

}

else if(i==80&&am<=200)

{ racket(370,am,2);

racket(370,am+2,4);

am+=2;

}

if((xb==-1&&yb==-1&&db==1))

{ setfillstyle(SOLID\_FILL,colorb[colb]); setcolor(colorb[colb]);

ball.moveball(db,colb);

ran = random(7);

}

else

{

switch(ran)

{ case 0: m = 0.412; c = 69.43; break;

case 1: m = 0.294; c = 118.28; break;

case 2: m = 0.138; c = 182.87; break;

case 3: m = 0.000; c = 240.00; break;

case 4: m = -0.138; c = 115+182.87; break;

case 5: m = -0.294; c = 244+118.28; break;

case 6: m = -0.412; c = 344+69.43; break;

}

xb = xb - 2\*db;

yb = m\*xb + c;

setfillstyle(SOLID\_FILL,colorb[colb]); setcolor(colorb[colb]);

ball.moveball(db,colb,xb,yb);

if(xb>=getmaxx()/2-188)

{ setfillstyle(SOLID\_FILL,BLUE); setcolor(BLUE);

}

else

{ setcolor(GREEN); setfillstyle(SOLID\_FILL,GREEN);

}

xb+=2\*db;

yb = m\*xb + c;

ball.moveball(db,colb,xb,yb);

racket(370,am,4);

if(xb<=getmaxx()/2-185)

{ setfillstyle(SOLID\_FILL,GREEN);

bar(getmaxx()/2-194,getmaxy()/2-150,getmaxx()/2-189,getmaxy());

setfillstyle(SOLID\_FILL,BLUE);

bar(getmaxx()/2-186,getmaxy()/2-140,getmaxx()/2-179,getmaxy()/2+143);

}

else if(xb>=getmaxx()/2+90)

{ setfillstyle(SOLID\_FILL,RED);

setcolor(RED);

pieslice(getmaxx()/2+130,getmaxy()/2,120,240,30);

setfillstyle(SOLID\_FILL,BLACK);

setcolor(BLACK);

pieslice(getmaxx()/2+130,getmaxy()/2,0,360,20);

setcolor(YELLOW);

pieslice(getmaxx()/2+130,getmaxy()/2,0,360,5);

}

setfillstyle(SOLID\_FILL,BLUE);

bar(getmaxx()/2+83,getmaxy()/2-10,getmaxx()/2+99,getmaxy()/2+10);

xb-=2\*db;

yb = m\*xb + c;

setfillstyle(SOLID\_FILL,colorb[colb]); setcolor(colorb[colb]);

ball.moveball(db,colb,xb,yb);

if(xb>=getmaxx()/2+90&&db==-1)

{ xb = getmaxx()/2+94;

yb = getmaxy()/2;

ran = random(7);

}

}

}

delay(1000);

settextstyle(9,0,1); setcolor(RED); setfillstyle(SOLID\_FILL,BLACK);

int lax;

do

{ lax = getch();

}while(lax!=13);

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

outtextxy(getmaxx()/2-94,getmaxy()/2-100,"GAME OVER");

ifstream f;

f.open("HSRAL",ios::in|ios::binary);

hs r;

f.read((char\*)&r,sizeof(r)); int fi = 0;

while(!f.eof()&&fi<5)

{ ralh[fi] = r;

fi++;

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

}

settextstyle(2,0,6);

if(ralh[4].s<score)

{ outtextxy(getmaxx()/2-200,getmaxy()/2+100,"HIGH SCORE! Enter your name:");

char a; int c = 0;

do

{ a = getch();

if((a>='a'&&a<='z')||(a>='A'&&a<='Z'))

{ if(c!=10){

ralh[5].n[c] = a;

c++;

ralh[5].n[c] = '\0';

outtextxy(getmaxx()/2+100,getmaxy()/2+100,ralh[5].n);

}

}

else if(a==8&&c!=0)

{ c--;

ralh[5].n[c] = '\0';

bar(getmaxx()/2+98,getmaxy()/2+98,getmaxx(),getmaxy()/2+140);

outtextxy(getmaxx()/2+100,getmaxy()/2+100,ralh[5].n);

}

}while(a!=13);

}

else

{

strcpy(ralh[5].n," ");

outtextxy(getmaxx()/2-58,getmaxy()/2+50,"Press ENTER");

int a;

do

{ a = getch();

}while(a!=13);

}

ralh[5].s = score; highscore();

}

void playtt()

{ delay(100);

ttdraw();

score = 0;

racketmove();

menup();

initmouse(); showmouseptr();

}

void menup()

{

setfillstyle(SOLID\_FILL,BLUE);

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

setfillstyle(SOLID\_FILL,GREEN);

bar(getmaxx()/2-130,getmaxy()/2-90,getmaxx()/2+110,getmaxy()/2-40);

setfillstyle(SOLID\_FILL,CYAN);

bar(getmaxx()/2-130,getmaxy()/2-10,getmaxx()/2+110,getmaxy()/2+40);

setfillstyle(SOLID\_FILL,MAGENTA);

bar(getmaxx()/2-130,getmaxy()/2+70,getmaxx()/2+110,getmaxy()/2+120);

setfillstyle(SOLID\_FILL,RED);

bar(getmaxx()/2-130,getmaxy()/2+150,getmaxx()/2+110,getmaxy()/2+200);

setcolor(YELLOW);

settextstyle(10,0,6);

outtextxy(getmaxx()/2-130,getmaxy()/2-227,"Rall-E");

settextstyle(9,0,2);

outtextxy(getmaxx()/2-45,getmaxy()/2-6,"Rules");

outtextxy(getmaxx()/2-40,getmaxy()/2-88,"Play");

outtextxy(getmaxx()/2-100,getmaxy()/2+74,"High Scores");

outtextxy(getmaxx()/2-41,getmaxy()/2+152,"Exit");

}

void ttmenu()

{ menup(); showmouseptr();

int tmx, tmy, tmb;

do

{ click(&tmx,&tmy,&tmb);

}while((tmb&1)!=1);

if((tmb&1)==1)

{ if(tmx>=getmaxx()/2-120&&tmx<=getmaxx()/2+120)

if(tmy>=getmaxy()/2+150&&tmy<=getmaxy()/2+200) ttf = 0;

else if(tmy>=getmaxy()/2-90&&tmy<=getmaxy()/2-40){initmouse(); i.x.ax = 2; playtt();}

else if(tmy>=getmaxy()/2+70&&tmy<=getmaxy()/2+120) hscores();

else if(tmy>=getmaxy()/2-10&&tmy<=getmaxy()/2+40) rules();

}

}

void ralle()

{

score = 0;

cleardevice(); randomize();

ball.path(); menup();

initmouse(); menup();

while(ttf!=0)

{

ttmenu();

}

ttf = 1; wl = 1;

}

**COWSBULL**

const int total=10;

char rulest[10][100]={"1. You have to find the 4 digit number with distinct digits.","2. You can play against the computer or another player.", "3. There are ten chances to guess the number.","4. A cow is when a digit is guessed and in the right place.","5. Bull is when a digit exists but in a different place.","6. The objective is to get 4 cows.","7. Press ENTER to begin playing!"};

int option, gout;

int norep(int a, int il, int number[])

{

int flag = 0;

for(int i = il-1; i >= 0&&flag==0; i--)

{ if(a==number[i]) flag = 1;

}

if(flag==1) return 0;

else return 1;

}

void gamecb()

{

int x, y, b;

cleardevice();

settextstyle(3,0,1);

setbkcolor(BLUE);

setcolor(YELLOW);

line(0,35,getmaxx(),35);

outtextxy(180,40,"COWS AND BULLS - MASTERMIND"); //BUNCH OF RULES.

line(0,70,getmaxx(),70);

for(int opt=10;opt!=1&&opt!=2&&opt!=3&&opt!=4;) //VS 2nd PLAYER? OR COMP?

{

outtextxy(28,210,"Choose an option: 1. vs Human 2. vs Computer 3. Rules 4. Exit");

outtextxy(270,300,"Choice: ");

opt=0;

int ent = 0;

do

{ char ic;

ic = getch();

setcolor(BLUE);

setfillstyle(SOLID\_FILL,BLUE);

if(ic>=49&&ic<=52)

{ opt = ic-48;

char a[2]; a[0] = char(ic); a[1] = '\0';

setcolor(YELLOW);

outtextxy(340,300,a);

ent = getch()-12;

if(ent==-4) bar(337,300,370,320);

else if(ent!=1)

do

{ ent = getch()-12;

if(ent==-4) bar(337,300,370,320);

}while(ent!=1&&ent!=-4);

}

}while((opt!=1&&opt!=2&&opt!=3&&opt!=4)||ent!=1);

}

int start=1;

option = opt;

if(opt!=3&&opt!=4)

{ delay(500);

cleardevice(); setcolor(YELLOW);

outtextxy(165,getmaxy()/2-16,"Press ENTER when you are ready!");

for(;start!=13;start=getch());

}

cleardevice();

int number[4];

if(opt==1) //VS 2nd PLAYER.

{

outtextxy(180,getmaxy()/2-50,"Player 2, enter your number:");

int sp = 0, ncheck = 0;

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

{

int a=getch()-48;

if(a>=0&&a<=9)

{ if(!(i==0&&a==0))

if(norep(a,i,number))

{

outtextxy(287+sp,getmaxy()/2+25,"\*");

number[i] = a;

i++;

ncheck++;

sp = sp + 20;

}

}

else if(a==-40)

{ if(i!=0) i--;

bar(287+sp-20,getmaxy()/2+20,287+sp,getmaxy()/2+50);

sp -=20;

}

}

delay(600);

}

else if(opt==2) //VS COMP.

{

int z[10]={0,1,2,3,4,5,6,7,8,9},y;

for(int x=0;x<4;x++)

{

randomize();

y=random(10-x);

number[x]=z[y];

if(number[0]==0) x--;

else

{

for(int j=y;j<9-x;j++)

z[j]=z[j+1];

}

}

}

cleardevice();

if(opt!=3&&opt!=4){

setcolor(YELLOW);

line(0,35,getmaxx(),35);

line(0,70,getmaxx(),70);

outtextxy(180,40,"COWS AND BULLS - MASTERMIND");

settextstyle(3,1,15);

setcolor(RED);

outtextxy(getmaxx()-200,95,"????");

}

settextstyle(3,0,1);

setcolor(YELLOW);

int i, yds = 0;

for(int guessno=1,guess[4],cow=0,bull=0;guessno<=total&&cow<4&&opt!=4&&opt!=3;guessno++) //"MAIN LOOP"

{

cow=bull=0;

char bl[4] = {'\0'};

outtextxy(20,171,"Guess Number Cows Bulls ");

line(19,192,71,192);

line(114,192,178,192);

line(225,192,270,192);

line(313,192,351,192);

for(int flag=0,error=1;flag==0;)

{

if(guessno!=1)

{ char d;

do

{ d = getch();

}while(d!=13);

}

bar(0,85,getmaxx()-210,163);

outtextxy(20,88,"Guess ");

outtextxy(75,88,itoa(guessno,bl,10));

gout = guessno;

if(guessno!=10) line(18,111,87,111);

else line(18,111,99,111);

int xs = 0;

for(i=0;i<4;)

{

int l = getch()-48; //EASY WAY TO ACCEPT SINGLE DIGIT NO.

if(l>=0&&l<=9)

{ char a[3];

itoa(l,a,10);

outtextxy(120+xs,88,a);

guess[i] = l;

i++; xs+=20;

}

else if(i!=0&&l==-40)

{ i--;

bar(98+xs,85,getmaxx()-210,110);

xs-=20;

}

}

if(guess[0]==0) outtextxy(20,130,"OOPS! Enter a 4 digit number!");

else{outtextxy(20,130,"Number accepted! See below!"); flag=1;}

}

if(error==1)

{

i=0;

int guessuse[4];

for(int y=0;y<4;y++) guessuse[y]=guess[y];

for(int j;i<4;i++) //IMP., COMPARING \*NUMBER\* WITH \*GUESS\*

{ //OUTER LOOP FOR NUMBER,INNER FOR GUESS.i->NUMBER. j-> GUESS.

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

{

if(guessuse[j]==number[i])

{

if(j==i)

{

cow++;

for(int y=0;y<4;y++)

if(guessuse[j]==guessuse[y]&&y!=j) guessuse[y]=-1;

}

}

}

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

{

if(guessuse[j]==number[i])

{

bull++;

for(int y=0;y<4;y++)

if(guessuse[j]==guessuse[y]&&y!=j) guessuse[y]=-1;

}

}

}

}

bull-=cow;

char bb[2]; bb[0] = bull + 48; bb[1] = '\0';

outtextxy(20,198+yds,itoa(guessno,bl,10));

for(int fda=0;fda<4;fda++) bl[fda]=guess[fda]+48;

bl[4]='\0';

outtextxy(120,198+yds,bl);

itoa(cow,bl,10);

outtextxy(245,198+yds,bl);

outtextxy(330,198+yds,bb);

yds+=23;

}

char out[5];

out[0] = gout+48; out[1] = '\0';

if(opt!=3&&opt!=4)

{

if(cow==4)

{ if(gout!=10){ outtextxy(20,getmaxy()-33,"Right Answer! You found the no. in move(s)!");

outtextxy(335,getmaxy()-33,out);

}

else outtextxy(20,getmaxy()-33,"Right Answer! You found the no. in 10 move(s)!");

}

else

{

outtextxy(20,getmaxy()-33,"Sorry! All chances are over! The answer is ");

for(int li=0;li<4;li++)

{ out[li] = number[li] + 48;

}

out[4] = '\0';

outtextxy(410,getmaxy()-33,out);

}

}

if(opt==3)

{ cleardevice();

settextstyle(3,0,1);

outtextxy(20,40,"Rules");

line(20,65,60,65);

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

outtextxy(20,80+30\*i,rulest[i]);

char lll;

do

{ lll=getch();

}while(lll!=13);

}

}

void mastermind()

{

cleardevice();

char o='y', op[2], ch;

while(option!=4)

{ if(o=='Y'||o=='y')

{ gamecb();

if(option!=4&&option!=3)

{ do

{ delay(3000);

cleardevice();

outtextxy(getmaxx()/2-100,getmaxy()/2-15,"Continue(Y/N):");

int ent = 0;

do

{ if(o==8) bar(getmaxx()/2+48,getmaxy()/2-20,getmaxx(),getmaxy());

if(ent != 13 && op[0]!='y'&&op[0]!='Y'&&op[0]!='N'&&op[0]!='n') o = getch();

op[0] = o; op[1] = '\0';

strupr(op);

if(o=='y'||o=='Y'||o=='n'||o=='N'){

outtextxy(getmaxx()/2+50,getmaxy()/2-15,op); ent = getch();

}

if(ent==8) o = 8;

}while((o!='Y'&&o!='y'&&o!='n'&&o!='N')||ent!=13);

delay(70);

}while(o!='Y'&&o!='y'&&o!='n'&&o!='N');

if(o=='N'||o=='n') option = 4;

}

}

}

option = 0;

}

void cowsbulls()

{

cleardevice();

mastermind();

}

#define KILL dead1,dead2,dead3

class high

{

public: char hey[10];

int hey\_score;

void waf();

void sort();

void war();

high()

{

strcpy(hey,"NULL");

hey\_score=0;

}

void pri();

}hio[100],ob;

void high::waf()

{

fstream f("highne.txt",ios::binary|ios::out);

for(int i=0;i<100;i++)

{

f.write((char\*)&hio[i],sizeof(hio[i]));

}

f.close();

}

void high::war()

{

fstream f("highne.txt",ios::binary|ios::in);

int i=0;

while(f)

{

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

hio[i].hey\_score=ob.hey\_score;

strcpy(hio[i].hey,ob.hey);

i++;

} f.close();

}

void high::sort()

{

for(int i=0;i<100;i++)

for(int j=0;j<100-1-i;j++)

if(hio[j].hey\_score < hio[j+1].hey\_score)

{

high t=hio[j];

hio[j]=hio[j+1];

hio[j+1]=t;

}

}

void high::pri()

{

hio[0].sort(); int op= 150;char dummy1[20],dummy2[20];

outtextxy(80+100,125,"NAME");

outtextxy(80+200+100,125,"SCORE");

for(int i=0;i<10 && hio[i].hey\_score!=0;i++)

{

setcolor(15);

itoa(i+1,dummy1,10);

outtextxy(80,op,dummy1);

outtextxy(80+30,op,".");

outtextxy(80+100,op,hio[i].hey);

itoa(hio[i].hey\_score,dummy2,10);

outtextxy(80+200+100,op,dummy2);

op+=20;

}

outtextxy(80,op+50,"To revert back press any key");

}

void usercraft();

void enemycraft\_1\_kill(int,int);

void enemycraft\_2\_kill(int ,int );

void enemycraft\_3\_kill(int ,int );

void scoreboard();

void boss();

int curx\_1=305,cury\_1=360+40,curx\_2=255,cury\_2=400+1,curx\_3=355,cury\_3=405+1;

int dead1=1,dead2=1,dead3=1;

int y1=200,y2=180;

int hell;int kell;

int xell[1000],yell[1000];

int scorefs=0;

int e1\_x=120,e2\_x=635/2,e3\_x=540-30,e1\_y=100+18,e2\_y=100+18,e3\_y=100+18;

int e1b1\_y=146,e2b2\_y=152,e3b3\_y=146,e1b1\_x=270,e2b2\_x=318,e3b3\_x=365;

int bad1=120,bad2=635/2,bad3=540-30;

int flagell=1,lol;

int shift1,shift2,shift3;

int checklev=1;

int oell=3;

int iell=1;

int lo=1;

int ha=3;

int check1=0;

int check3=0;

char Jeev[4]={3,3,3};

char Jeev1[4]={3,3,3};

char ch[4]={3,3,3,0};

char ch1[4]={3,3,3,0};

char d\_score[20];

char ka[1];

char nag[20];

void black()

{

setfillstyle(1,0);

setcolor(0);

pieslice(curx\_1,cury\_1,0,360,30);

bar(curx\_1-4,cury\_1-40,curx\_1+4,cury\_1-30);

}

void shoot(int k)

{

setcolor(0);

line(k,y1,k,y2);

setcolor(15);

y1-=40;

y2-=40;

line(k,y1,k,y2);

if(checklev==1)

{

enemycraft\_1\_kill(k,y2);

}

else if(checklev==2)

{

enemycraft\_2\_kill(k,y2);

}

else if(checklev==3)

{

enemycraft\_3\_kill(k,y2);

}

delay(200);

}

void modr\_usercraft()

{

black();

curx\_1+=40;

setfillstyle(1,8);

setcolor(8);

pieslice(curx\_1,cury\_1,0,360,30);

setfillstyle(1,GREEN);

setcolor(GREEN);

pieslice(curx\_1,cury\_1,0,360,15);

setcolor(15);

circle(curx\_1,cury\_1,15);

setfillstyle(1,RED);

bar(curx\_1-4,cury\_1-40,curx\_1+4,cury\_1-30);

}

void modl\_usercraft()

{

black();

curx\_1-=40;

setfillstyle(1,8);

setcolor(8);

pieslice(curx\_1,cury\_1,0,360,30);

setfillstyle(1,GREEN);

setcolor(GREEN);

pieslice(curx\_1,cury\_1,0,360,15);

setcolor(15);

circle(curx\_1,cury\_1,15);

setfillstyle(1,RED);

bar(curx\_1-4,cury\_1-40,curx\_1+4,cury\_1-30);

}

void enemycraft\_1()

{

setfillstyle(1,RED);

setcolor(RED);

pieslice(120,100,0,360,15);

setfillstyle(1,GREEN);

setcolor(GREEN);

pieslice(120,100,0,360,4);

setcolor(15);

circle(120,100,4);

setfillstyle(1,RED);

setcolor(RED);

pieslice(635/2,100,0,360,15);

setfillstyle(1,GREEN);

setcolor(GREEN);

pieslice(635/2,100,0,360,4);

setcolor(15);

circle(635/2,100,4);

setfillstyle(1,RED);

setcolor(RED);

pieslice(540-30,100,0,360,15);

setfillstyle(1,GREEN);

setcolor(GREEN);

pieslice(540-30,100,0,360,4);

setcolor(15);

circle(540-30,100,4);

setfillstyle(1,8);

bar(120-2,100+10+5,120+2,100+2+10+5);

bar(635/2-2,100+10+5,635/2+2,100+2+10+5);

bar(510-2,100+10+5,510+2,100+2+10+5);

}

void enemycraft\_1\_kill(int hell,int kell)

{

setfillstyle(1,0);

setcolor(0);

if((kell<=110 && kell>=90))

{

if((hell>=100 && hell<=150 && dead1==1))

{

setcolor(0);

pieslice(120,100,0,360,15);

bar(120-2,100+10+5,120+2,100+2+10+5);

scorefs+=30;dead1=0;

}

else if((hell>=635/2-10-10 && hell<=635/2+10+10 && dead2==1))

{

pieslice(635/2,100,0,360,15);

bar(635/2-2,100+10+5,635/2+2,100+2+10+5);

scorefs+=30;dead2=0;

}

else if((hell>=540-30-10-10) && (hell<=540-30+10+10) && dead3==1)

{

pieslice(540-30,100,0,360,15);

bar(510-2,100+10+5,510+2,100+2+10+5);

scorefs+=30;dead3=0;

}

}

}

void usercraft\_hit(int j,int kell, int l)

{

if((pow((j-curx\_1),2)+pow((kell-cury\_1),2)<=30\*30)||(pow((j-curx\_1),2)+pow((l-cury\_1),2)<=30\*30))

{

black();

delay(1000);

if(j<=curx\_1)

modr\_usercraft();

if(j>=curx\_1)

modl\_usercraft();

ch[--oell]='\0';

}

scoreboard();

}

void enemycraft\_1\_shoot(int dead1=1,int dead2=1,int dead3=1)

{

//Always shooting segment.

{

setcolor(0);

line(e1\_x,e1\_y,e1\_x,e1\_y+40); e1\_y+=20;

if(dead1)

{

setcolor(15); line(e1\_x,e1\_y,e1\_x,e1\_y+40);

if(e1\_y>=490)

e1\_y=100+18;

usercraft\_hit(e1\_x,e1\_y,e1\_y+40);

}

}

if(iell%2==0)

{

setcolor(0);

line(e2\_x,e2\_y,e2\_x,e2\_y+40); e2\_y+=20;

if(dead2)

{

setcolor(15); line(e2\_x,e2\_y,e2\_x,e2\_y+40);

if(e2\_y>=490)

e2\_y=100+18;

usercraft\_hit(e2\_x,e2\_y,e2\_y+40);

}

}

if(iell%3==0)

{

setcolor(0);

line(e3\_x,e3\_y,e3\_x,e3\_y+40); e3\_y+=20;

if(dead3)

{

setcolor(15); line(e3\_x,e3\_y,e3\_x,e3\_y+40);

if(e3\_y>=490)

e3\_y=100+18;

usercraft\_hit(e3\_x,e2\_y,e3\_y+40);

}

}

iell++;

}

void scoreboard()

{

setcolor(15);

rectangle(0,0,50,50);

rectangle(0,0,50,20);

outtextxy(0,5," SCORE");

setcolor(0);

ch[3]='\0';

outtextxy(20,30,d\_score);

setcolor(15);

itoa(scorefs,d\_score,10);

outtextxy(20,30,d\_score);

if(oell!=3){ setcolor(0); outtextxy(1,54,ch1);}

setcolor(RED);

outtextxy(1,54,ch);

setcolor(15);

}

void background()

{

for(int iell=0;iell<1000;iell++)

{

hell=random(640);kell=random(480);

putpixel(hell,kell,15);

xell[iell]=hell;

yell[iell]=kell;

}

usercraft();

enemycraft\_1();

scoreboard();

}

void level1()

{

do

{

for(int iell=0;iell<1000;iell++)

{

putpixel(xell[iell],yell[iell],BLACK);

delay(100);

putpixel(xell[iell],yell[iell],15);

enemycraft\_1\_shoot(KILL);

if(!strlen(ch))

{

flagell=0;iell=1001;

}

if(kbhit())

{

char opt=getch();

switch(opt)

{

case 77:{

if(curx\_1<540)

{

modr\_usercraft();

kell=380+random(50);

putpixel(random(640),kell,15);

}

}break;

case 75:{

if(curx\_1>90)

{

modl\_usercraft();

kell=380+random(50);

putpixel(random(640),kell,15);

}

}break;

case 32:

y1=350;y2=340;int l=curx\_1;

for(int i=0;i<12;i++)

{

enemycraft\_1\_shoot(KILL);

if(!strlen(ch))

{ flagell=0; i=1001;}

shoot(l);

if(kbhit())

{

char opt=getch();

switch(opt)

{

case 77:{

if(curx\_1<540)

{

modr\_usercraft();

kell=380+random(50);

putpixel(random(640),kell,15);

}

}break;

case 75: if(curx\_1>90)

{

modl\_usercraft();

kell=380+random(50);

putpixel(random(640),kell,15);

}break;

}

} setcolor(15);scoreboard();setcolor(0);

}break;

}

if(dead1==0 && dead2==0 && dead3==0)

{

iell=1001;

flagell=0;

}

}

}

}while(flagell);

checklev=2;

}

void usercraft()

{

setfillstyle(1,8);

setcolor(8);

pieslice(curx\_1,cury\_1,0,360,30);

setfillstyle(1,GREEN);

setcolor(GREEN);

pieslice(curx\_1,cury\_1,0,360,15);

setcolor(15);

circle(curx\_1,cury\_1,15);

setfillstyle(1,RED);

bar(curx\_1-4,cury\_1-40,curx\_1+4,cury\_1-30);

}

void black1()

{

setfillstyle(1,0);

setcolor(0);

pieslice(bad1,100-30-20,0,360,15);

bar(bad1-2,100+10+5-30-20,bad1+2,100+2+10+5-30-20);

}

void black2()

{

setfillstyle(1,0);

setcolor(0);

pieslice(bad2,100,0,360,15);

bar(bad2-2,100+10+5,bad2+2,100+2+10+5);

}

void black3()

{

setfillstyle(1,0);

setcolor(0);

pieslice(bad3,100+30+20,0,360,15);

bar(bad3-2,100+10+5+30+20,bad3+2,100+2+10+5+30+20);

}

void display2()

{

setfillstyle(1,RED);

setcolor(RED);

pieslice(bad1,100-30-20,0,360,15);

setfillstyle(1,GREEN);

setcolor(GREEN);

pieslice(bad1,100-30-20,0,360,4);

setcolor(15);

circle(bad1,100-30-20,4);

setfillstyle(1,RED);

setcolor(RED);

pieslice(bad2,100,0,360,15);

setfillstyle(1,GREEN);

setcolor(GREEN);

pieslice(bad2,100,0,360,4);

setcolor(15);

circle(bad2,100,4);

setfillstyle(1,RED);

setcolor(RED);

pieslice(bad3,100+30+20,0,360,15);

setfillstyle(1,GREEN);

setcolor(GREEN);

pieslice(bad3,100+30+20,0,360,4);

setcolor(15);

circle(bad3,100+30+20,4);

setfillstyle(1,8);

bar(bad1-2,100+10+5-30-20,bad1+2,100+2+10+5-30-20);

bar(bad2-2,100+10+5,bad2+2,100+2+10+5);

bar(bad3-2,100+10+5+30+20,bad3+2,100+2+10+5+30+20);

}

void bad1\_shoot(int dead1)

{

setcolor(0);

line(e1\_x,e1\_y,e1\_x,e1\_y+40); e1\_y+=20;

if(dead1)

{

setcolor(15); line(e1\_x,e1\_y,e1\_x,e1\_y+40);

if(e1\_y>=490)

{ e1\_y=88; bad1+=shift1;check1=1;}

usercraft\_hit(e1\_x,e1\_y,e1\_y+40);

}

}

void bad2\_shoot(int dead2)

{

setcolor(0);

line(e2\_x,e2\_y,e2\_x,e2\_y+40); e2\_y+=20;

if(lo%3==0)

{

if(dead2)

{

setcolor(15); line(e2\_x,e2\_y,e2\_x,e2\_y+40);

if(e2\_y>=490)

e2\_y=100+18;

usercraft\_hit(e2\_x,e2\_y,e2\_y+40);

}

}

lo++;

}

void bad3\_shoot(int dead3)

{

setcolor(0);

line(e3\_x,e3\_y,e3\_x,e3\_y+40); e3\_y+=20;

if(dead3)

{

setcolor(15); line(e3\_x,e3\_y,e3\_x,e3\_y+40);

if(e3\_y>=490)

{e3\_y=148+20; bad3+=shift3; check3=1;}

usercraft\_hit(e3\_x,e3\_y,e3\_y+40);

}

}

void enemycraft\_2\_shoot()

{

setfillstyle(1,RED);

if(dead1)

{

if(bad1==120)

{

e1\_x=120;

shift1=5;

bad1\_shoot(dead1);

}

else if(bad1==510)

{

e1\_x=510;

bad1\_shoot(dead1);

shift1=-5;

}

else

{

if(bad1==125 && check1==1)

{

bad1-=5;

}

if(bad1==505 && check1==1)

{

bad1+=5;

}

check1=0;

black1();

bad1+=shift1;

setfillstyle(1,RED);

setcolor(RED);

pieslice(bad1,100-30-20,0,360,15);

setfillstyle(1,GREEN);

setcolor(GREEN);

pieslice(bad1,100-30-20,0,360,4);

setcolor(15);

circle(bad1,100-30-20,4);

setfillstyle(1,8);

bar(bad1-2,100+10+5-30-20,bad1+2,100+2+10+5-30-20);

}

}

kell=bad1+random(50);

putpixel(random(640),kell,15);

bad2\_shoot(dead2);

if(dead3)

{

if(bad3==120)

{

e3\_x=120;

bad3\_shoot(dead3);

shift3=5;

}

else if(bad3==540-30)

{

e3\_x=510;

bad3\_shoot(dead3);

shift3=-5;

}

else

{

if(bad3==125 && check3==1)

{

bad3-=5;

}

if(bad3==505 && check3==1)

{

bad3+=5;

}

check3=0;

black3();

bad3+=shift3;

setfillstyle(1,RED);

setcolor(RED);

pieslice(bad3,100+30+20,0,360,15);

setfillstyle(1,GREEN);

setcolor(GREEN);

pieslice(bad3,100+30+20,0,360,4);

setcolor(15);

circle(bad3,100+30+20,4);

setfillstyle(1,8);

bar(bad3-2,100+10+5+30+20,bad3+2,100+2+10+5+30+20);

}

}

if(dead1==0)

{

bad1\_shoot(dead1);

}

if(dead3==0)

{

bad3\_shoot(dead3);

}

kell=bad3+random(50);

putpixel(random(640),kell,15);

}

void enemycraft\_2\_kill(int hell,int kell)

{

if((pow(hell-bad1,2)+pow(kell-50,2))<=225 && dead1)

{

black1(); dead1=0;scorefs+=30;

}

if((pow(hell-bad2,2)+pow(kell-(100),2))<=225 && dead2)

{

black2(); dead2=0;scorefs+=30;

}

if((pow(hell-bad3,2)+pow(kell-(100+30+20),2))<=225 && dead3)

{

black3(); dead3=0; scorefs+=30;

}

}

void level2()

{

dead1=1;dead2=1;dead3=1;

e1\_y=88;e2\_y=100+18;e3\_y=100+18+30+20;flagell=1;

do

{

for(int i=0;i<1000;i++)

{

putpixel(xell[i],yell[i],BLACK);

delay(100);

putpixel(xell[i],yell[i],15);

enemycraft\_2\_shoot();

if(!strlen(ch))

flagell=0,i=1001;

if(kbhit())

{

char opt=getch();

switch(opt)

{

case 77:{

if(curx\_1<540)

{

modr\_usercraft();

kell=380+random(50);

putpixel(random(640),kell,15);

}

}break;

case 75:{

if(curx\_1>90)

{

modl\_usercraft();

kell=380+random(50);

putpixel(random(640),kell,15);

}

}break;

case 32: y1=350;y2=340;int l=curx\_1;

for(int i=0;i<12;i++)

{

enemycraft\_2\_shoot();

if(!strlen(ch)) { flagell=0; i=1001;}

shoot(l);

if(kbhit())

{

char opt=getch();

switch(opt)

{

case 77:{

if(curx\_1<540)

{

modr\_usercraft();

kell=380+random(50);

putpixel(random(640),kell,15);

}

}break;

case 75: if(curx\_1>90)

{

modl\_usercraft();

kell=380+random(50);

putpixel(random(640),kell,15);

}break;

}

}

setcolor(15);scoreboard();setcolor(0);

}break;

}

}

if(dead1==0 && dead2==0 && dead3==0)

{ flagell=0;

i=1001;

}

}

}while(flagell); checklev=3;

}

void boss\_life()

{

Jeev[ha]='\0';

if(ha!=3)

{setcolor(0);outtextxy(635/2-10,35,Jeev1);}

setcolor(15); outtextxy(635/2-10,35,Jeev);

}

void enemycraft\_3\_shoot()

{

setcolor(0);

if(iell%2==0)

{

line(e1\_x,e1\_y,e1\_x,e1\_y+40); e1\_y+=20;

if(dead1)

{

setcolor(15); line(e1\_x,e1\_y,e1\_x,e1\_y+40);

if(e1\_y>=490)

e1\_y=100+18;

usercraft\_hit(e1\_x,e1\_y,e1\_y+40);

}

}

//Always shooting segment.

{

setcolor(0);

line(e1b1\_x,e1b1\_y,e1b1\_x,e1b1\_y+40); e1b1\_y+=20;

if(dead2)

{

setcolor(15); line(e1b1\_x,e1b1\_y,e1b1\_x,e1b1\_y+40);

if(e1b1\_y>=490)

e1b1\_y=146;

usercraft\_hit(e1b1\_x,e1b1\_y,e1b1\_y+40);

}

}

//Always shooting segment.

{

setcolor(0);

line(e3b3\_x,e3b3\_y,e3b3\_x,e3b3\_y+40); e3b3\_y+=20;

if(dead2)

{

setcolor(15); line(e3b3\_x,e3b3\_y,e3b3\_x,e3b3\_y+40);

if(e3b3\_y>=490)

e3b3\_y=146;

usercraft\_hit(e3b3\_x,e3b3\_y,e3b3\_y+40);

}

}

if(iell%2==0)

{

setcolor(0);

line(e2b2\_x,e2b2\_y,e2b2\_x,e2b2\_y+40); e2b2\_y+=20;

if(dead2)

{

setcolor(15); line(e2b2\_x,e2b2\_y,e2b2\_x,e2b2\_y+40);

if(e2b2\_y>=490)

e2b2\_y=152;

usercraft\_hit(e2b2\_x,e2b2\_y,e2b2\_y+40);

}

}

if(iell%3==0)

{

setcolor(0);

line(e3\_x,e3\_y,e3\_x,e3\_y+40); e3\_y+=20;

if(dead3)

{

setcolor(15); line(e3\_x,e3\_y,e3\_x,e3\_y+40);

if(e3\_y>=490)

e3\_y=100+18;

usercraft\_hit(e3\_x,e2\_y,e3\_y+40);

}

}

iell++;

}

void enemycraft\_3\_kill(int hell,int kell)

{

setfillstyle(1,0);

setcolor(0);

// To kill small aliens.

if((kell<=110 && kell>=90))

{

if((hell>=100 && hell<=150 && dead1))

{

setcolor(0);

pieslice(120,100,0,360,15);

bar(120-2,100+10+5,120+2,100+2+10+5);

scorefs+=30;dead1=0;

}

else if((hell>=540-30-10-10) && (hell<=540-30+10+10) && dead3)

{

pieslice(540-30,100,0,360,15);

bar(510-2,100+10+5,510+2,100+2+10+5);

scorefs+=30;dead3=0;

}

}

//To kill boss

if(pow(hell-635/2,2)+pow(kell-100,2)<=13\*13 && dead2)

{

ha--;

boss\_life();

scorefs+=50;

if(Jeev[0]=='\0')

{

dead2=0;

setfillstyle(1,0);

setcolor(0);

fillellipse(635/2,100,100,50);

bar(635/2-30,100+10+5+34,635/2+30,100+2+10+5+40);

bar(635/2-30-20,100+10+5+34,635/2-30-15,100+5+40);

bar(635/2+30+15,100+10+5+34,635/2+30+20,100+5+40);

}

else

{

boss();

}

}

scoreboard();

}

void boss()

{

setfillstyle(1,RED);

setcolor(RED);

fillellipse(635/2,100,100,50);

setfillstyle(1,GREEN);

setcolor(15);

fillellipse(635/2,100,70,30);

setfillstyle(1,RED);

setcolor(15);

fillellipse(635/2,100,50,20);

setfillstyle(1,8);

setcolor(8);

pieslice(635/2,100,0,360,20);

setcolor(15);

circle(635/2,100,20);

bar(635/2-30,100+10+5+34,635/2+30,100+2+10+5+40);

bar(635/2-30-20,100+10+5+34,635/2-30-15,100+5+40);

bar(635/2+30+15,100+10+5+34,635/2+30+20,100+5+40);

}

void display\_3()

{

setfillstyle(1,RED);

setcolor(RED);

fillellipse(635/2,100,100,50);

setfillstyle(1,GREEN);

setcolor(15);

fillellipse(635/2,100,70,30);

setfillstyle(1,RED);

setcolor(15);

fillellipse(635/2,100,50,20);

setfillstyle(1,8);

setcolor(8);

pieslice(635/2,100,0,360,20);

setcolor(15);

circle(635/2,100,20);

setfillstyle(1,RED);

setcolor(RED);

pieslice(120,100,0,360,15);

setfillstyle(1,GREEN);

setcolor(GREEN);

pieslice(120,100,0,360,4);

setcolor(15);

circle(120,100,4);

setfillstyle(1,RED);

setcolor(RED);

pieslice(540-30,100,0,360,15);

setfillstyle(1,GREEN);

setcolor(GREEN);

pieslice(540-30,100,0,360,4);

setcolor(15);

circle(540-30,100,4);

setfillstyle(1,8);

bar(120-2,100+10+5,120+2,100+2+10+5);

bar(635/2-30,100+10+5+34,635/2+30,100+2+10+5+40);

bar(635/2-30-20,100+10+5+34,635/2-30-15,100+5+40);

bar(635/2+30+15,100+10+5+34,635/2+30+20,100+5+40);

bar(510-2,100+10+5,510+2,100+2+10+5);

}

void level3()

{

boss\_life();

e1\_x=120;e3\_x=510;e1\_y=100+18;e3\_y=100+18;

dead1=1;dead2=1;dead3=1; iell=0;flagell=1;

do

{

for(int i=0;i<1000;i++)

{

putpixel(xell[i],yell[i],BLACK);

delay(100);

putpixel(xell[i],yell[i],15);

if(dead2)

boss();

enemycraft\_3\_shoot();

if(!strlen(ch))

flagell=0,i=1001;

if(kbhit())

{

char opt=getch();

switch(opt)

{

case 77:{

if(curx\_1<540)

{

modr\_usercraft();

kell=380+random(50);

putpixel(random(640),kell,15);

}

}break;

case 75:{

if(curx\_1>90)

{

modl\_usercraft();

kell=380+random(50);

putpixel(random(640),kell,15);

}

}break;

case 32:y1=350;y2=340;int l=curx\_1;

for(int i=0;i<12;i++)

{

enemycraft\_3\_shoot();

if(!strlen(ch))

{ flagell=0; i=1001;}

shoot(l);

if(dead2)

boss();

if(kbhit())

{

char opt=getch();

switch(opt)

{

case 77:{

if(curx\_1<540)

{

modr\_usercraft();

kell=380+random(50);

putpixel(random(640),kell,15);

}

}break;

case 75: if(curx\_1>90)

{

modl\_usercraft();

kell=380+random(50);

putpixel(random(640),kell,15);

}break;

}

}

setcolor(15);

scoreboard();

setcolor(0);

if(dead2)

boss();

}break;

}

}

if(dead1==0 && dead2==0 && dead3==0)

{

i=1001;

flagell=0;

}

}

}while(flagell); checklev=3;

}

void gameover()

{

setcolor(14);

settextstyle(3,0,6);

outtextxy(180,200,"GAME OVER");

outtextxy(180,250,"SCORE:");

outtextxy(380,250,d\_score);

delay(3000);

setcolor(15);

getch();

}

void start\_level1()

{

setcolor(15);

settextstyle(3,0,6);

outtextxy(180,200,"LEVEL 1");

delay(1200);

setcolor(0);

outtextxy(180,200,"LEVEL 1");

settextstyle(DEFAULT\_FONT,0,1);

}

void start\_level2()

{

setcolor(15);

settextstyle(3,0,6);

outtextxy(180,200,"LEVEL 2");

delay(1200);

setcolor(0);

outtextxy(180,200,"LEVEL 2");

settextstyle(DEFAULT\_FONT,0,1);

}

void rules(int kell)

{

setfillstyle(1,0);

bar(0,0,640,480);

setcolor(kell);

settextstyle(3,0,1);

outtextxy(5,100,"Rules :");

outtextxy(5,110,"----");

outtextxy(5,140,"1.To move your spacecraft press left and right arrow keys.");

outtextxy(5,160,"2.To shoot tap the space bar.");

outtextxy(5,180,"3.You are only allowed one shot at a time.");

outtextxy(5,200,"4.Three lives are with you at the start of the game.");

outtextxy(5,220,"5.Once three lives are lost you'll return to the homescreen.");

outtextxy(5,240,"6.If an alien is hit you gain 30 points.");

outtextxy(5,260,"7.The scoreboard is displayed at the left corner of your screen.");

outtextxy(5,280,"8.To return to home page press any key");

getch();

}

void homescreen()

{

setfillstyle(1,0);

bar(0,0,640,480);

for(int i=0;i<1000;i++)

{

hell=random(640);kell=random(480);

putpixel(hell,kell,kell);

xell[i]=hell;

yell[i]=kell;

}

setcolor(15);

settextstyle(3,0,6);

outtextxy(150,50,"SPACE WARS");

settextstyle(3,0,3);

outtextxy(80,130,"Enter 1 to PLAY");

outtextxy(80,160,"Enter 2 to RULES");

outtextxy(80,190,"Enter 3 for HIGH SCORES");

outtextxy(80,220,"Enter 4 for EXIT");

outtextxy(80,300,"Enter your option:");

int kell=1;

do

{

char b=getch();

ka[1]='\0';

switch(b)

{

case 8:setcolor(0);outtextxy(300,300,ka);ka[0]='\0';break;

case 13:if(ka[0]!='\0')

{

kell=0;

}break;

default:setcolor(0); outtextxy(300,300,ka);ka[0]=b; setcolor(15); outtextxy(300,300,ka);break;

}

}while(kell);

settextstyle(DEFAULT\_FONT,0,1);

}

void clrhome()

{

setcolor(0);

settextstyle(3,0,6);

outtextxy(150,50,"SPACE WARS");

settextstyle(3,0,3);

outtextxy(80,130,"Enter 1 to PLAY");

outtextxy(80,160,"Enter 2 to RULES");

outtextxy(80,190,"Enter 3 for HIGH SCORES");

outtextxy(80,220,"Enter 4 for EXIT");

outtextxy(80,300,"Enter your option:");

outtextxy(300,300,ka);

setcolor(15);

settextstyle(DEFAULT\_FONT,0,1);

}

void start\_level3()

{

setcolor(15);

settextstyle(3,0,6);

outtextxy(180,200,"LEVEL 3");

delay(1200);

setcolor(0);

outtextxy(180,200,"LEVEL 3");

settextstyle(DEFAULT\_FONT,0,1);

}

void winner()

{

setcolor(14);

settextstyle(3,0,6);

outtextxy(180,200,"YOU WON");

outtextxy(180,250,"SCORE:");

outtextxy(380,250,d\_score);

delay(400);

setcolor(15);

delay(3000);

getch();

}

void high\_score(int co)

{ setfillstyle(1,0);

bar(0,0,640,480);

setcolor(co);

settextstyle(3,0,1);

outtextxy(5,100,"Highscores:");

hio[0].pri();

getch();

}

void final()

{

char opt[1];

do

{

hio[0].war();

homescreen();

if(ka[0]=='1')

{

clrhome();

//Reinitializes everything!

curx\_1=305,cury\_1=360+40,curx\_2=255,cury\_2=400+1,curx\_3=355,cury\_3=405+1;

dead1=1,dead2=1,dead3=1;

y1=200;y2=180;

scorefs=0;

e1\_x=120,e2\_x=635/2,e3\_x=540-30,e1\_y=100+18,e2\_y=100+18,e3\_y=100+18;

e1b1\_y=146,e2b2\_y=152,e3b3\_y=146,e1b1\_x=270,e2b2\_x=318,e3b3\_x=365;

bad1=120,bad2=635/2,bad3=540-30;

flagell=1;

checklev=1;

oell=3;

iell=1;

lo=1;

ha=3;

check1=0;

check3=0;

strcpy(ch,ch1);

strcpy(Jeev,Jeev1);

setcolor(15);

settextstyle(3,0,3);

outtextxy(80,280,"Enter your name:");

int bv = 0; int kell = 1;

do

{ char buf=getch();

switch(buf)

{

case 8: if(bv)

{

setcolor(0);

outtextxy(290,280,nag);

nag[--bv]=0;

setcolor(15);

outtextxy(290,280,nag);

}break;

case 13: kell=0;break;

default: if(bv<40-1 && buf>=' ' && buf<='~')

{

nag[bv]=buf;

nag[++bv]=0;

outtextxy(290,280,nag);

}

}

nag[bv]='\0';

if(bv==9) kell = 0;

} while(kell);

strcpy(hio[99].hey,nag);

delay(600);

setcolor(0); settextstyle(3,0,3);

outtextxy(80,280,"Enter your name:");

outtextxy(290,280,nag);

settextstyle(DEFAULT\_FONT,0,1);

start\_level1();

background();

level1();

if(!strlen(ch))

{

hio[99].hey\_score=scorefs;

gameover();

hio[0].sort();

hio[0].waf();

}

else

{

start\_level2();

display2();

level2();

if(!strlen(ch))

{

hio[99].hey\_score=scorefs;

gameover();

hio[0].sort();

hio[0].waf();

}

else

{

start\_level3();

display\_3();

level3();

if(!strlen(ch))

{

hio[99].hey\_score=scorefs;

gameover();

hio[0].sort();

hio[0].waf();

}

else

{ hio[99].hey\_score=scorefs;

hio[0].sort();

hio[0].waf();

winner();

}

}

}

}

else if(ka[0]=='4')

{

return ;

}

else if(ka[0]=='2')

{

clrhome();

rules(15);

}

else if(ka[0]=='3')

{

clrhome();

high\_score(15);

}

setcolor(0);

}while(1);

}

void spacew()

{ final();

}

**MOUSE-1**

class mouse

{

REGS regs; int no\_buttons;

public: int cx,cy,q;

void get\_status();

void init\_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);

}

}mouse;

void mouse::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;

}

**BIBLIOGRAPHY**

* Sumita Arora Class 11 and 12
* CS Notebooks
* CrazyProgrammer
* TutorialsPoint