#include<ctype.h>

#include<time.h>

#include<conio.h>

#include<dos.h>

#include<stdlib.h>

#include<windows.h>

#include<stdio.h>

#include<string>

#include<iostream>

#include<math.h>

#include<cstdlib>

#include<GL/glut.h>

using namespace std;

#define glClearColor3v(f) glClearColor((f)[0], (f)[1], (f)[2], 1.0)

void welcome\_screen();

void main\_menu();

void timer(int value);

void keyboard(unsigned char key, int x, int y);

void options(int n);

void exit\_screen();

int validate\_alarm\_time(string s);

int start=1;

int stopwatch=0,alarm=0,a\_clock=1;

int stopwatch\_started=0,stopwatch\_stopped=0;

int alarm\_started=0;

char alarm\_confirm='n';

string alarm\_time;

time\_t start\_t, end\_t;

double diff\_t=0.0;

int trigger\_alarm=0;

int trigger\_color=0;

int validate\_alarm\_time(string s)

{

int flag=1,a,b;

char validate\_alarm\_time[100];

int TempNumOne=s.size();

int min,hour,sec;

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

validate\_alarm\_time[a]=s[a];

if(strlen(validate\_alarm\_time)!=8)

flag=0;

if(validate\_alarm\_time[2]!=':')

flag=0;

if(validate\_alarm\_time[5]!=':')

flag=0;

a = validate\_alarm\_time[6] - '0';

b = validate\_alarm\_time[7] - '0';

sec = (10\*a)+b;

a = validate\_alarm\_time[3] - '0';

b = validate\_alarm\_time[4] - '0';

min = (10\*a)+b;

a = validate\_alarm\_time[0] - '0';

b = validate\_alarm\_time[1] - '0';

hour = (10\*a)+b;

if(sec>=60)

flag=0;

if(min>=60)

flag=0;

if(hour>=24)

flag=0;

if(flag==1)

return 1;

else

return 0;

}

void exit\_screen()

{

glClearColor(0.0,0.0,0.0,1.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.0, 0.0, 1.0);

char text1[]={"C R E A T E D B Y :"};

char text21[100]={"C . A D I T Y A"};

glRasterPos2i(120,280);

for(int i = 0; text1[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, text1[i]);

glColor3f(1.0, 0.0, 0.0);

glRasterPos2i(150,220);

for(int i = 0; text21[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, text21[i]);

glFlush();

glutSwapBuffers();

Sleep(3000);

exit(0);

}

void options(int n)

{

switch(n)

{

case 1: stopwatch=1;

alarm=0;

a\_clock=0;

break;

case 2: alarm=1;

stopwatch=0;

a\_clock=0;

break;

case 3: a\_clock=1;

alarm=0;

stopwatch=0;

break;

case 4: exit\_screen();

break;

}

}

void keyboard(unsigned char key, int x, int y)

{

string temp,temp1;

if(trigger\_alarm==1)

{

switch(key)

{

case 27:trigger\_alarm=0;

a\_clock=1;

alarm=0;

alarm\_started=0;

break;

default:break;

}

}

if(stopwatch==1)

{

switch (key)

{

case 's': time(&start\_t);

stopwatch\_started=1;

break;

case 'a': time(&end\_t);

stopwatch\_stopped=1;

diff\_t = difftime(end\_t, start\_t);

break;

case 'd':diff\_t=0.0;

stopwatch\_started=0;

stopwatch\_stopped=0;

break;

case 'S': time(&start\_t);

stopwatch\_started=1;

break;

case 'A': time(&end\_t);

stopwatch\_stopped=1;

diff\_t = difftime(end\_t, start\_t);

break;

case 'D':diff\_t=0.0;

stopwatch\_started=0;

stopwatch\_stopped=0;

break;

default: break;

}

}

if(alarm==1)

{

int res;

switch(key)

{

case 'y': b: system("cls");

printf("\n Kindly Enter The Time For The Alarm In The Following Format (HH:MM:SS) :\n ");

cin >> temp;

getchar();

res=validate\_alarm\_time(temp);

if(res==1)

{

printf("\n\n\n Are You Sure ? (y/n) : \n ");

alarm\_confirm=getchar();

if(alarm\_confirm=='y' || alarm\_confirm=='Y')

{

alarm\_started=1;

alarm\_time=temp;

printf("\n\n\n ALARM SET !!!");

}

else

{

if(alarm\_started==1)

{

printf("\n\n\n ALARM UNCHANGED !!!");

alarm\_started=1;

}

else

{

printf("\n\n\n ALARM NOT SET !!!");

alarm\_started=0;

}

}

Sleep(2000);

system("cls");

}

if(res==0)

{

printf("\n\n INVAILD TIME. KINDLY ENSURE YOU FOLLOW THE CORRECT FORMAT AS DISPLAYED");

printf("\n\n Press any key to continue......");

getch();

goto b;

}

a\_clock=1;

break;

case 'n':a\_clock=1;

break;

case 'Y':c: system("cls");

printf("\n Kindly Enter The Time For The Alarm In The Following Format (HH:MM:SS) :\n ");

cin >> temp;

getchar();

res=validate\_alarm\_time(temp);

if(res==1)

{

printf("\n\n\n Are You Sure ? (y/n) : \n ");

alarm\_confirm=getchar();

if(alarm\_confirm=='y' || alarm\_confirm=='Y')

{

alarm\_started=1;

alarm\_time=temp;

printf("\n\n\n ALARM SET !!!");

}

else

{

if(alarm\_started==1)

{

printf("\n\n\n ALARM UNCHANGED !!!");

alarm\_started=1;

}

else

{

printf("\n\n\n ALARM NOT SET !!!");

alarm\_started=0;

}

}

Sleep(2000);

system("cls");

}

if(res==0)

{

printf("\n\n INVAILD TIME. KINDLY ENSURE YOU FOLLOW THE CORRECT FORMAT AS DISPLAYED");

printf("\n\n Press any key to continue......");

getch();

goto c;

}

a\_clock=1;

break;

default:break;

}

}

}

void timer(int value)

{

if(trigger\_alarm==1)

{

alarm=0;

stopwatch=0;

a\_clock=0;

GLfloat color[8][3]={{0.0,0.0,0.0},{0.0,0.0,1.0},{0.0,1.0,0.0},{0.0,1.0,1.0},{1.0,0.0,0.0},{1.0,0.0,1.0},{1.0,1.0,0.0},{1.0,1.0,1.0}};

char text1[]={"P R E S S [ E S C ] T O S T O P A L A R M"};

char text22[100]={" A L A R M ! ! ! ! ! !"};

glClearColor3v(color[trigger\_color]);

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.5,0.5,0.5);

trigger\_color++;

if(trigger\_color==8)

trigger\_color=0;

glRasterPos2i(140,350);

for(int i = 0; text22[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, text22[i]);

glRasterPos2i(20,200);

for(int i = 0; text1[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, text1[i]);

Beep(523,500);

glFlush();

glutSwapBuffers();

}

else

{

if(alarm\_started==1)

{

time\_t curtime;

time(&curtime);

char date\_and\_time[100];

char date\_time[9];

strcpy(date\_and\_time, ctime(&curtime));

date\_time[0]= date\_and\_time[11];

date\_time[1]= date\_and\_time[12];

date\_time[2]= ':';

date\_time[3]= date\_and\_time[14];

date\_time[4]= date\_and\_time[15];

date\_time[5]= ':';

date\_time[6]= date\_and\_time[17];

date\_time[7]= date\_and\_time[18];

date\_time[8]= '\0';

string present\_time=string(date\_time);

if(alarm\_time == present\_time)

{

alarm\_started=0;

trigger\_alarm=1;

alarm=0;

a\_clock=1;

}

}

if(stopwatch==1)

{

if(stopwatch\_started==1 && stopwatch\_stopped==0)

{

time(&end\_t);

diff\_t = difftime(end\_t, start\_t);

}

else if(stopwatch\_started==0 && stopwatch\_stopped==0)

diff\_t = 0.0;

}

}

glutPostRedisplay();

glutTimerFunc(100,timer,0);

}

void init()

{

glClearColor(0.0,0.0,0.0,1.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0,499.0,0.0,499.0);

glMatrixMode(GL\_MODELVIEW);

}

void main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutInitWindowPosition(750, 100);

glutCreateWindow("A & A Clocking Systems !!!");

glutDisplayFunc(welcome\_screen);

glutTimerFunc(100,timer,0);

glutKeyboardFunc(keyboard);

glutCreateMenu(options);

glutAddMenuEntry(" STOP WATCH ",1);

glutAddMenuEntry(" ALARM ",2);

glutAddMenuEntry(" CLOCK ",3);

glutAddMenuEntry(" QUIT ",4);

glutAttachMenu(GLUT\_RIGHT\_BUTTON);

init();

glutMainLoop();

}

void welcome\_screen(void)

{

if(start==1)

{

glClearColor(0.0,0.0,0.0,1.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.0, 0.0, 1.0);

start=0;

char welcome[]={"W E L C O M E ! ! !"};

char welcome1[]={"A & A C L O C K I N G"};

char welcome2[]={"S Y S T E M S"};

glColor3f(1.0, 0.0, 0.0);

glRasterPos2i(120,400);

for(int i = 0; welcome[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, welcome[i]);

glColor3f(0.0, 1.0, 0.0);

glRasterPos2i(100,220);

for(int i = 0; welcome1[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, welcome1[i]);

glRasterPos2i(160,190);

for(int i = 0; welcome2[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, welcome2[i]);

glFlush();

glutSwapBuffers();

Sleep(3000);

glColor3f(0.0, 0.0, 1.0);

main\_menu();

}

else

main\_menu();

}

void main\_menu()

{

char text111[]={" A L A R M N O T S E T !!!"};

char text222[]={" A L A R M S E T A T "};

char display\_alarm\_time[100];

int TempNumOne=alarm\_time.size();

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

display\_alarm\_time[a]=alarm\_time[a];

if(alarm\_started==0)

{

glRasterPos2i(130,10);

for(int i = 0; text111[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, text111[i]);

}

else

{

glRasterPos2i(110,10);

for(int i = 0; text222[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, text222[i]);

for(int i = 0; display\_alarm\_time[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, display\_alarm\_time[i]);

}

if(a\_clock==1)

{

stopwatch=0;

alarm=0;

a\_clock=1;

glClearColor(0.0,0.0,0.0,1.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0, 0.5, 0.0);

time\_t curtime;

time(&curtime);

char date\_and\_time[100];

char c\_min[2],c\_sec[2],c\_hour[2];

int a,b;

int min,sec,hour;

strcpy(date\_and\_time, ctime(&curtime));

c\_min[0]=date\_and\_time[14];

c\_min[1]=date\_and\_time[15];

c\_sec[0]=date\_and\_time[17];

c\_sec[1]=date\_and\_time[18];

c\_hour[0]=date\_and\_time[11];

c\_hour[1]=date\_and\_time[12];

a = c\_sec[0] - '0';

b = c\_sec[1] - '0';

sec = (10\*a)+b;

a = c\_min[0] - '0';

b = c\_min[1] - '0';

min = (10\*a)+b;

a = c\_hour[0] - '0';

b = c\_hour[1] - '0';

hour = (10\*a)+b;

glPushMatrix();

glLoadIdentity();

glRasterPos2i(130,450);

for(int i = 0; date\_and\_time[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, date\_and\_time[i]);

glRasterPos2i(80,250);

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, '9');

glRasterPos2i(240,75);

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, '6');

glRasterPos2i(410,250);

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, '3');

glRasterPos2i(240,405);

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, '1');

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, '2');

glBegin(GL\_LINE\_LOOP);

for(int ii = 0; ii < 300; ii++)

{

float theta = 2.0 \* 3.1415926 \* float(ii) / float(300);

float x = 150 \* cos(theta);

float y = 150 \* sin(theta);

glVertex2f(x + 250, y + 250);

}

glEnd();

glBegin(GL\_LINE\_LOOP);

for(int ii = 0; ii < 300; ii++)

{

float theta = 2.0 \* 3.1415926 \* float(ii) / float(300);

float x = 180 \* cos(theta);

float y = 180 \* sin(theta);

glVertex2f(x + 250, y + 250);

}

glEnd();

glPointSize(5.0);

glBegin(GL\_POINTS);

glVertex2i(250,250);

glEnd();

glPopMatrix();

glPushMatrix();

glLoadIdentity();

glTranslatef(250.0,250.0,0.0);

glRotatef(sec\*6.0, 0.0, 0.0, -1.0);

glTranslatef(-250.0,-250.0,0.0);

glBegin(GL\_LINES);

glVertex2i(250,250);

glVertex2i(250,395);

glEnd();

glPopMatrix();

glPushMatrix();

glLoadIdentity();

glTranslatef(250.0,250.0,0.0);

glRotatef((min\*6.0)+(sec\*0.1), 0.0, 0.0, -1.0);

glTranslatef(-250.0,-250.0,0.0);

glBegin(GL\_LINES);

glVertex2i(250,250);

glVertex2i(250,360);

glEnd();

glPopMatrix();

glPushMatrix();

glLoadIdentity();

glTranslatef(250.0,250.0,0.0);

glRotatef((hour\*30.0)+(min\*0.5), 0.0, 0.0, -1.0);

glTranslatef(-250.0,-250.0,0.0);

glBegin(GL\_LINES);

glVertex2i(250,250);

glVertex2i(250,320);

glEnd();

glPopMatrix();

char text111[]={" A L A R M N O T S E T !!!"};

char text222[]={" A L A R M S E T A T "};

char display\_alarm\_time[100];

int TempNumOne=alarm\_time.size();

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

display\_alarm\_time[a]=alarm\_time[a];

if(alarm\_started==0)

{

glRasterPos2i(130,10);

for(int i = 0; text111[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, text111[i]);

}

else

{

glRasterPos2i(110,10);

for(int i = 0; text222[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, text222[i]);

for(int i = 0; display\_alarm\_time[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, display\_alarm\_time[i]);

}

glFlush();

glutSwapBuffers();

}

if(stopwatch==1)

{

stopwatch=1;

alarm=0;

a\_clock=0;

glClearColor(0.0,0.0,0.0,1.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.5, 1.0, 0.0);

char text1[]={" Press 's' to start the stopwatch."};

char text23[100]={" Press 'a' to stop the stopwatch."};

char text3[]={" Press 'd' to restart the stopwatch."};

int min=0,hour=0,sec=0,rem=0;

char c\_min[3],c\_hour[3],c\_sec[3];

glColor3f(0.5, 0.5, 0.5);

glRasterPos2i(20,80);

for(int i = 0; text1[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_12, text1[i]);

glRasterPos2i(20,65);

for(int i = 0; text23[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_12, text23[i]);

glRasterPos2i(20,50);

for(int i = 0; text3[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_12, text3[i]);

glColor3f(0.5, 1.0, 0.0);

hour=diff\_t/3600;

rem=int(diff\_t)%3600;

min=rem/60;

rem=rem%60;

sec=rem;

sprintf(c\_min, "%d", min);

sprintf(c\_sec, "%d", sec);

sprintf(c\_hour, "%d", hour);

glRasterPos2i(220,270);

for(int i = 0; c\_hour[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, c\_hour[i]);

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, ':');

for(int i = 0; c\_min[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, c\_min[i]);

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, ':');

for(int i = 0; c\_sec[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, c\_sec[i]);

char text111[]={" A L A R M N O T S E T !!!"};

char text222[]={" A L A R M S E T A T "};

char display\_alarm\_time[100];

int TempNumOne=alarm\_time.size();

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

display\_alarm\_time[a]=alarm\_time[a];

if(alarm\_started==0)

{

glRasterPos2i(130,10);

for(int i = 0; text111[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, text111[i]);

}

else

{

glRasterPos2i(110,10);

for(int i = 0; text222[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, text222[i]);

for(int i = 0; display\_alarm\_time[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, display\_alarm\_time[i]);

}

glFlush();

glutSwapBuffers();

}

if(alarm==1)

{

int res;

stopwatch=0;

alarm=1;

a\_clock=0;

glClearColor(0.0,0.0,0.0,1.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.0, 0.0, 1.0);

if(alarm\_started==1)

{

char text1[]={"A L A R M A L R E A D Y S E T !!!"};

char text24[100]={"W A N T T O M O D I F Y ? ( Y / N )"};

glRasterPos2i(50,270);

for(int i = 0; text1[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, text1[i]);

glRasterPos2i(50,240);

for(int i = 0; text24[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, text24[i]);

}

else

{

a: system("cls");

printf("\n Kindly Enter The Time For The Alarm In The Following Format (HH:MM:SS) :\n ");

cin >> alarm\_time;

getchar();

res=validate\_alarm\_time(alarm\_time);

if(res==1)

{

printf("\n\n\n Are You Sure ? (y/n) : \n ");

alarm\_confirm=getchar();

if(alarm\_confirm=='y' || alarm\_confirm=='Y')

{

alarm\_started=1;

printf("\n\n\n ALARM SET !!!");

}

else

{

if(alarm\_started==1)

{

printf("\n\n\n ALARM UNCHANGED !!!");

alarm\_started=1;

}

else

{

printf("\n\n\n ALARM NOT SET !!!");

alarm\_started=0;

}

}

Sleep(2000);

system("cls");

a\_clock=1;

}

}

if(res==0)

{

printf("\n\n INVAILD TIME. KINDLY ENSURE YOU FOLLOW THE CORRECT FORMAT AS DISPLAYED");

printf("\n\n Press any key to continue......");

getch();

goto a;

}

char text111[]={" A L A R M N O T S E T !!!"};

char text222[]={" A L A R M S E T A T "};

char display\_alarm\_time[100];

int TempNumOne=alarm\_time.size();

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

display\_alarm\_time[a]=alarm\_time[a];

if(alarm\_started==0)

{

glRasterPos2i(130,10);

for(int i = 0; text111[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, text111[i]);

}

else

{

glRasterPos2i(110,10);

for(int i = 0; text222[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, text222[i]);

for(int i = 0; display\_alarm\_time[i] != '\0'; i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, display\_alarm\_time[i]);

}

glFlush();

glutSwapBuffers();

}

}