#include <stdio.h>

#include <stdlib.h>

#include <GL/glut.h>

#include <math.h>

#include <string.h>

#define ESCAPE 27

GLint window;

GLint window2;

GLint Xsize=1000;

GLint Ysize=800;

float i,theta;

GLint nml=0,day=1;

char name3[]="PROJECT: 3D CAR ANIMATION";

GLfloat xt=0.0,yt=0.0,zt=0.0,xw=0.0;

GLfloat xs=1.0,ys=1.0,zs=1.0;

GLfloat xangle=0.0,yangle=0.0,zangle=0.0,angle=0.0;

GLfloat r=0,g=0,b=1;

GLint light=1;

int count=1,flg=1;

int view=0;

int flag1=0,aflag=1; //to switch car driving mode

int flag2=0,wheelflag=0; //to switch fog effect

GLUquadricObj \*t;

static void SpecialKeyFunc( int Key, int x, int y );

GLvoid Transform(GLfloat Width, GLfloat Height)

{

glViewport(0, 0, Width, Height);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(45.0,Width/Height,0.1,100.0);

glMatrixMode(GL\_MODELVIEW);

}

GLvoid InitGL(GLfloat Width, GLfloat Height)

{

glClearColor(1.0, 1.0, 1.0, 1.0);

glLineWidth(2.0); /\* Add line width, ditto \*/

Transform( Width, Height ); /\* Perform the transformation \*/

t=gluNewQuadric();

gluQuadricDrawStyle(t, GLU\_FILL);

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

GLfloat ambientLight[] = { 0.2f, 0.2f, 0.2f, 1.0f };

GLfloat diffuseLight[] = { 0.8f, 0.8f, 0.8, 1.0f };

GLfloat specularLight[] = { 0.5f, 0.5f, 0.5f, 1.0f };

GLfloat position[] = { 1.5f, 1.0f, 4.0f, 1.0f };

glLightfv(GL\_LIGHT0, GL\_AMBIENT, ambientLight);

glLightfv(GL\_LIGHT0, GL\_DIFFUSE, diffuseLight);

glLightfv(GL\_LIGHT0, GL\_SPECULAR, specularLight);

glLightfv(GL\_LIGHT0, GL\_POSITION, position);

}

void init()

{

glClearColor(0,0,0,0);

glPointSize(5.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0.0,900.0,0.0,600.0,50.0,-50.0);

glutPostRedisplay();

}

void display\_string(int x, int y, char \*string, int font)

{

int len,i;

glColor3f(0.8,0.52,1.0);

glRasterPos2f(x, y);

len = (int) strlen(string);

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

if(font==1)

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

if(font==2)

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

if(font==3)

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

if(font==4)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_10,string[i]);

}

}

void display1(void)

{

glClearColor(1.0 ,1.0 ,1.0,1.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

display\_string(190,540,"Amruta institute of engineering & management sciences",1);

display\_string(225,500,name3,1);

display\_string(390+10,470,"HELP",2);

display\_string(10,450,"MOUSE",2);

display\_string(10,410,"PRESS RIGHT BUTTON FOR MENU",3);

display\_string(10,370,"KEYBOARD",2);

display\_string(10,340,"X-Y-Z KEYS FOR CORRESPONDING ROTATION",3);

display\_string(10,280+30,"U-F FOR CAMERA VIEW SETTINGS",3);

display\_string(10,250+30,"USE LEFT ARROW(<-) AND RIGHT ARROW(->) TO MOVE CAR",3);

display\_string(10,220+30,"ESCAPE TO EXIT",3);

display\_string(250,150+30,"PRESS SPACE BAR TO ENTER",2);

glutPostRedisplay();

glutSwapBuffers();

}

GLvoid DrawGLScene()

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

if(view==0)

{

init();

display1();

}

else

{

if(count==1)

InitGL(Xsize,Ysize);

if(aflag==1)/\* Initialize our window. \*/

glClearColor(1,1,1,1);

else

glClearColor(0.1,0.1,0.1,0);

glPushMatrix();

glLoadIdentity();

glTranslatef(-1.0,0.0,-3.5);

glRotatef(xangle,1.0,0.0,0.0);

glRotatef(yangle,0.0,1.0,0.0);

glRotatef(zangle,0.0,0.0,1.0);

glTranslatef(xt,yt,zt);

glScalef(xs,ys,zs);

glEnable(GL\_COLOR\_MATERIAL);

if(flag2==1)

{

GLfloat fogcolour[4]={1.0,1.0,1.0,1.0};

glFogfv(GL\_FOG\_COLOR,fogcolour);

glFogf(GL\_FOG\_DENSITY,0.1);

glFogi(GL\_FOG\_MODE,GL\_EXP);

glFogf(GL\_FOG\_START,3.0);

glFogf(GL\_FOG\_END,100.0);

glHint(GL\_FOG\_HINT, GL\_FASTEST);

glEnable(GL\_FOG);

}

if(flag2==0)

{

glDisable(GL\_FOG);

}

if(!aflag){

glBegin(GL\_POINTS);

glColor3f(1,1,1);

glPointSize(200.0);

int ccount=0;

float x=10,y=10;

while(ccount<20)

{

glVertex2f(x,y);

x+=10;

y+=10;

if(y>Ysize) y-=10;

if(x>Xsize) x-=10;

ccount++;

}

glEnd();}

glColor3f(1.0,.75,0.0);

glPointSize(30.0);

glBegin(GL\_POINTS);

glVertex3f(0.2,0.3,0.3);

glVertex3f(0.2,0.3,0.5);

glEnd();

glPointSize(200.0);

glBegin(GL\_QUADS); /\* OBJECT MODULE\*/

/\* top of cube\*/

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FRONT BODY\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

glColor3f(r,g,b);

glVertex3f( 0.2, 0.4,0.6);

glVertex3f(0.6, 0.5,0.6);

glVertex3f(0.6, 0.5,0.2);

glVertex3f( 0.2,0.4,0.2);

/\* bottom of cube\*/

glVertex3f( 0.2,0.4,0.6);

glVertex3f(0.6,0.2,0.6);

glVertex3f(0.6,0.2,0.2);

glVertex3f( 0.2,0.2,0.2);

/\* front of cube\*/

glVertex3f( 0.2,0.2,0.6);

glVertex3f(0.2, 0.4,0.6);

glVertex3f(0.2,0.4,0.2);

glVertex3f( 0.2,0.2,0.2);

/\* back of cube.\*/

glVertex3f(0.6,0.2,0.6);

glVertex3f(0.6,0.5,0.6);

glVertex3f(0.6,0.5,0.2);

glVertex3f( 0.6,0.2,0.2);

/\* left of cube\*/

glVertex3f(0.2,0.2,0.6);

glVertex3f(0.6,0.2,0.6);

glVertex3f(0.6,0.5,0.6);

glVertex3f(0.2,0.4,0.6);

/\* Right of cube \*/

glVertex3f(0.2,0.2,0.2);

glVertex3f( 0.6,0.2,0.2);

glVertex3f( 0.6,0.5,0.2);

glVertex3f( 0.2,0.4,0.2);

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

glVertex3f(0.7,0.65,0.6);

glVertex3f(0.7,0.65,0.2);

glVertex3f(1.7,0.65,0.2); //top cover

glVertex3f(1.7,0.65,0.6);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*back guard\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

glColor3f(r,g,b); /\* Set The Color To Blue\*/

glVertex3f( 1.8, 0.5,0.6);

glVertex3f(1.8, 0.5,0.2);

glVertex3f(2.1, 0.4, 0.2);

glVertex3f(2.1,0.4,0.6);

/\* bottom of cube\*/

glVertex3f( 2.1,0.2,0.6);

glVertex3f(2.1,0.2,0.2);

glVertex3f(1.8,0.2,0.6);

glVertex3f( 1.8,0.2,0.6);

/\* back of cube.\*/

glVertex3f(2.1,0.4,0.6);

glVertex3f(2.1,0.4,0.2);

glVertex3f(2.1,0.2,0.2);

glVertex3f(2.1,0.2,0.6);

/\* left of cube\*/

glVertex3f(1.8,0.2,0.2);

glVertex3f(1.8,0.5,0.2);

glVertex3f(2.1,0.4,0.2);

glVertex3f(2.1,0.2,0.2);

/\* Right of cube \*/

glVertex3f(1.8,0.2,0.6);

glVertex3f(1.8,0.5,0.6);

glVertex3f(2.1,0.4,0.6);

glVertex3f(2.1,0.2,0.6);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MIDDLE BODY\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

glVertex3f( 0.6, 0.5,0.6);

glVertex3f(0.6, 0.2,0.6);

glVertex3f(1.8, 0.2, 0.6);

glVertex3f(1.8,0.5,0.6);

/\* bottom of cube\*/

glVertex3f( 0.6,0.2,0.6);

glVertex3f(0.6,0.2,0.2);

glVertex3f(1.8,0.2,0.2);

glVertex3f( 1.8,0.2,0.6);

/\* back of cube.\*/

glVertex3f(0.6,0.5,0.2);

glVertex3f(0.6,0.2,0.2);

glVertex3f(1.8,0.2,0.2);

glVertex3f(1.8,0.5,0.2);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ENTER WINDOW\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

glColor3f(0.3,0.3,0.3);

glVertex3f( 0.77, 0.63,0.2);

glVertex3f(0.75, 0.5,0.2); //quad front window

glVertex3f(1.2, 0.5, 0.2);

glVertex3f( 1.22,0.63,0.2);

glVertex3f(1.27,0.63,.2);

glVertex3f(1.25,0.5,0.2); //quad back window

glVertex3f(1.65,0.5,0.2);

glVertex3f(1.67,0.63,0.2);

glColor3f(r,g,b);

glVertex3f(0.7,0.65,0.2);

glVertex3f(0.7,0.5,.2); //first separation

glVertex3f(0.75,0.5,0.2);

glVertex3f(0.77,0.65,0.2);

glVertex3f(1.2,0.65,0.2);

glVertex3f(1.2,0.5,.2); //second separation

glVertex3f(1.25,0.5,0.2);

glVertex3f(1.27,0.65,0.2);

glVertex3f(1.65,0.65,0.2);

glVertex3f(1.65,0.5,.2); //3d separation

glVertex3f(1.7,0.5,0.2);

glVertex3f(1.7,0.65,0.2);

glVertex3f( 0.75, 0.65,0.2);

glVertex3f(0.75, 0.63,0.2); //line strip

glVertex3f(1.7, 0.63, 0.2);

glVertex3f( 1.7,0.65,0.2);

glVertex3f( 0.75, 0.65,0.6);

glVertex3f(0.75, 0.63,0.6); //line strip

glVertex3f(1.7, 0.63, 0.6);

glVertex3f( 1.7,0.65,0.6);

glColor3f(0.3,0.3,0.3);

glVertex3f( 0.77, 0.63,0.6);

glVertex3f(0.75, 0.5,0.6); //quad front window

glVertex3f(1.2, 0.5, 0.6);

glVertex3f( 1.22,0.63,0.6);

glVertex3f(1.27,0.63,.6);

glVertex3f(1.25,0.5,0.6); //quad back window

glVertex3f(1.65,0.5,0.6);

glVertex3f(1.67,0.63,0.6);

glColor3f(r,g,b);

glVertex3f(0.7,0.65,0.6);

glVertex3f(0.7,0.5,.6); //first separation

glVertex3f(0.75,0.5,0.6);

glVertex3f(0.77,0.65,0.6);

glVertex3f(1.2,0.65,0.6);

glVertex3f(1.2,0.5,.6); //second separation

glVertex3f(1.25,0.5,0.6);

glVertex3f(1.27,0.65,0.6);

glVertex3f(1.65,0.65,0.6);

glVertex3f(1.65,0.5,.6);

glVertex3f(1.7,0.5,0.6);

glVertex3f(1.7,0.65,0.6);

glEnd();

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

glBegin(GL\_QUADS);

/\* top of cube\*/

glColor3f(0.3,0.3,0.3);

glVertex3f( 0.6, 0.5,0.6);

glVertex3f(0.6, 0.5,0.2); //quad front window

glVertex3f(0.7, 0.65, 0.2);

glVertex3f( 0.7,0.65,0.6);

glVertex3f(1.7,0.65,.6);

glVertex3f(1.7,0.65,0.2); //quad back window

glVertex3f(1.8,0.5,0.2);

glVertex3f(1.8,0.5,0.6);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*road and surrounding development\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if(flag1)

{

glPushMatrix();

glTranslatef(xw,0,0);

glColor3f(0,1,0);

glVertex3f(-100,0.1,-100);

glVertex3f(-100,0.1,0); //a green surroundings

glVertex3f(100,0.1,0);

glVertex3f(100,0.1,-100);

glColor3f(0.7,0.7,0.7);

glVertex3f(-100,0.1,0);

glVertex3f(-100,0.1,0.45); //a long road

glVertex3f(100,0.1,0.45);

glVertex3f(100,0.1,0);

glColor3f(1.0,0.75,0.0);

glVertex3f(-100,0.1,0.45); //a median

glVertex3f(-100,0.1,0.55);

glVertex3f(100,0.1,0.55);

glVertex3f(100,0.1,0.45);

glColor3f(0.7,0.7,0.7);

glVertex3f(-100,0.1,0.55);

glVertex3f(-100,0.1,1); //a long road

glVertex3f(100,0.1,1);

glVertex3f(100,0.1,0.55);

glColor3f(0,1,0);

glVertex3f(-100,0.1,1);

glVertex3f(-100,0.1,100); //a green surroundings

glVertex3f(100,0.1,100);

glVertex3f(100,0.1,1);

glPopMatrix();

}

glEnd();

if(wheelflag)

{

glPushMatrix();

glTranslatef(xw,0,0);

glColor3f(0.5,.2,0.3);

glBegin(GL\_QUADS);

for(i=0;i<200;i+=0.2)

{

glVertex3f(-100+i,0,1);

glVertex3f(-99.9+i,0,1);

glVertex3f(-99.9+i,0.2,1);

glVertex3f(-100+i,0.2,1);

i+=0.5;

}

for(i=0;i<200;i+=0.2)

{

glVertex3f(-100+i,0,0);

glVertex3f(-99.9+i,0,0);

glVertex3f(-99.9+i,0.2,0);

glVertex3f(-100+i,0.2,0);

i+=0.5;

}

glEnd();

glPopMatrix();

}

//

glBegin(GL\_TRIANGLES); /\* start drawing the cube.\*/

/\* top of cube\*/

glColor3f(0.3,0.3,0.3);

glVertex3f( 0.6, 0.5,0.6);

glVertex3f( 0.7,0.65,0.6); //tri front window

glVertex3f(0.7,0.5,0.6);

glVertex3f( 0.6, 0.5,0.2);

glVertex3f( 0.7,0.65,0.2); //tri front window

glVertex3f(0.7,0.5,0.2);

glVertex3f( 1.7, 0.65,0.2);

glVertex3f( 1.8,0.5,0.2); //tri back window

glVertex3f( 1.7,0.5,0.2);

glVertex3f( 1.7, 0.65,0.6);

glVertex3f( 1.8,0.5,0.6); //tri back window

glVertex3f(1.7,0.5,0.6);

glEnd();

//\*\*\*\*\*\*\*\*\*\*\*\*IGNITION SYSTEM

glPushMatrix();

glColor3f(0.3,0.3,0.7);

glTranslatef(1.65,0.2,0.3);

glRotatef(90.0,0,1,0);

gluCylinder(t,0.02,0.03,.5,10,10);

glPopMatrix();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WHEEL

glColor3f(0.7,0.7,0.7);

glPushMatrix();

glBegin(GL\_LINE\_STRIP);

for(theta=0;theta<360;theta=theta+40)

{

glVertex3f(0.6,0.2,0.62);

glVertex3f(0.6+(0.08\*(cos(((theta+angle)\*3.14)/180))),0.2+(0.08\*(sin(((theta+angle)\*3.14)/180))),0.62);

}

glEnd();

glBegin(GL\_LINE\_STRIP);

for(theta=0;theta<360;theta=theta+40)

{

glVertex3f(0.6,0.2,0.18);

glVertex3f(0.6+(0.08\*(cos(((theta+angle)\*3.14)/180))),0.2+(0.08\*(sin(((theta+angle)\*3.14)/180))),0.18);

}

glEnd();

glBegin(GL\_LINE\_STRIP);

for(theta=0;theta<360;theta=theta+40)

{

glVertex3f(1.7,0.2,0.18);

glVertex3f(1.7+(0.08\*(cos(((theta+angle)\*3.14)/180))),0.2+(0.08\*(sin(((theta+angle)\*3.14)/180))),0.18);

}

glEnd();

glBegin(GL\_LINE\_STRIP);

for(theta=0;theta<360;theta=theta+40)

{

glVertex3f(1.7,0.2,0.62);

glVertex3f(1.7+(0.08\*(cos(((theta+angle)\*3.14)/180))),0.2+(0.08\*(sin(((theta+angle)\*3.14)/180))),0.62);

}

glEnd();

glTranslatef(0.6,0.2,0.6);

glColor3f(0,0,0);

glutSolidTorus(0.025,0.07,10,25);

glTranslatef(0,0,-0.4);

glutSolidTorus(0.025,0.07,10,25);

glTranslatef(1.1,0,0);

glutSolidTorus(0.025,0.07,10,25);

glTranslatef(0,0,0.4);

glutSolidTorus(0.025,0.07,10,25);

glPopMatrix();

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

glPopMatrix();

glEnable(GL\_DEPTH\_TEST);

glutPostRedisplay();

glutSwapBuffers();

}

}

void NormalKey(GLubyte key, GLint x, GLint y)

{

switch ( key ) {

case ESCAPE : printf("escape pressed. exit.\n");

glutDestroyWindow(window);

exit(0);

break;

case ' ':view=1;

DrawGLScene();

break;

case 'x': xangle += 5.0;

glutPostRedisplay();

break;

case 'X':xangle -= 5.0;

glutPostRedisplay();

break;

case 'y':

yangle += 5.0;

glutPostRedisplay();

break;

case 'Y':

yangle -= 5.0;

glutPostRedisplay();

break;

case 'z':

zangle += 5.0;

glutPostRedisplay();

break;

case 'Z':

zangle -= 5.0;

glutPostRedisplay();

break;

case 'u': /\* Move up \*/

yt += 0.2;

glutPostRedisplay();

break;

case 'U':

yt -= 0.2; /\* Move down \*/

glutPostRedisplay();

break;

case 'f': /\* Move forward \*/

zt += 0.2;

glutPostRedisplay();

break;

case 'F':

zt -= 0.2; /\* Move away \*/

glutPostRedisplay();

break;

default:

break;

}

}

static void SpecialKeyFunc( int Key, int x, int y )

{

switch ( Key ) {

case GLUT\_KEY\_RIGHT:

if(!wheelflag)

xt += 0.2;

if(wheelflag)

{

angle+=5;

xw+=0.2;

}

glutPostRedisplay();

break;

case GLUT\_KEY\_LEFT:

if(!wheelflag)

xt -= 0.2;

if(wheelflag)

{

angle+=5;

xw-=0.2;

}

glutPostRedisplay();

break;

}

}

void myMenu(int id)

{

if (id==1)

{

flag1=0;

wheelflag=0;

glutPostRedisplay();

}

if(id ==2)

{

flag1=1;

flag2=0;

wheelflag=0;

xangle += 5.0;

glutPostRedisplay();

}

if (id==4)

{

wheelflag=1;

glutPostRedisplay();

}

if(id==12)

{

aflag=1;

day=1;

glClearColor(1,1,1,1);

glDisable(GL\_FOG);

glutPostRedisplay();

}

if(id==13)

{

aflag=0;

day=0;

flag2=2;

glClearColor(0.1,0.1,0.1,0);

GLfloat fogcolour[4]={0.0,0.0,0.0,1.0};

glFogfv(GL\_FOG\_COLOR,fogcolour);

glFogf(GL\_FOG\_DENSITY,0.5);

glFogi(GL\_FOG\_MODE,GL\_EXP);

glHint(GL\_FOG\_HINT, GL\_FASTEST);

glEnable(GL\_FOG);

glutPostRedisplay();

}

}

void colorMenu(int id)

{

if (id==6)

{

r=g=0;

b=1;

glutPostRedisplay();

}

if(id ==7)

{

r=0.8;

b=g=0;

glutPostRedisplay();

}

if(id==8)

{

g=1;

r=b=0;

glutPostRedisplay();

}

if (id==9)

{

r=b=g=0;

glutPostRedisplay();

}

if(id==10)

{

b=0;

r=g=1;

glutPostRedisplay();

}

if(id==11)

{

b=r=g=.7;

glutPostRedisplay();

}

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_RGBA | GLUT\_DOUBLE|GLUT\_DEPTH);

glutInitWindowSize(Xsize,Ysize);

glutInitWindowPosition(50,50);

glutCreateWindow("3D CAR ANIMATION");

glutDisplayFunc(DrawGLScene);

glutKeyboardFunc(NormalKey);

glutSpecialFunc( SpecialKeyFunc );

InitGL(Xsize,Ysize);

int submenu=glutCreateMenu(colorMenu);

glutAddMenuEntry("blue", 6);

glutAddMenuEntry("red", 7);

glutAddMenuEntry("green",8);

glutAddMenuEntry("black",9);

glutAddMenuEntry("yellow",10);

glutAddMenuEntry("grey",11);

glutCreateMenu(myMenu);

glutAddMenuEntry("car model mode", 1);

glutAddMenuEntry("car driving mode", 2);

glutAddMenuEntry("wheel effect",4);

glutAddSubMenu("car colors",submenu);

glutAddMenuEntry("daymode",12);

glutAddMenuEntry("Night mode",13);

glutAttachMenu(GLUT\_RIGHT\_BUTTON);

glutMainLoop();

return 1;

}