Вариант 10.

Вопросы - 3, 10, 17, 27, 23

3.Показать код построения линии поля, выходящей из заданной точки пространства.

struct POINT3 {

double x, y, z;

};

struct VECTORS {

double x, y, z;

double dx, dy, dz;

};

//длины координатных осей (в мировых координатах)

double xmax, ymax, zmax;

//рисует одну линию поля из начальной точки PointB

void LineField(HDC hdc,POINT3 PointB)

{

VECTORS vect;

vect.x = PointB.x;

vect.y = PointB.y;

vect.z = PointB.z;

//видовые координаты проецируемой точки

double xe, ye;

//координаты пикселов

int x1,y1,x2,y2;

double dt = 0.01;

double x, y, z, Hx, Hy, Hz, Ha;

VECMAG mag;

double xt1,yt1,zt1,xt2,yt2,zt2;

do

{

x = vect.x;

y = vect.y;

z = vect.z;

mag = magn(x,y,z);

Hx = mag.hx;

Hy = mag.hy;

Hz = mag.hz;

Ha = sqrt(Hx\*Hx + Hy\*Hy + Hz\*Hz);

vect.dx = Hx/Ha;

vect.dy = Hy/Ha;

vect.dz = Hz/Ha;

xt1 = vect.x; yt1 = vect.y; zt1 = vect.z;

xt2 = xt1 + vect.dx\*dt;

yt2 = yt1 + vect.dy\*dt;

zt2 = zt1 + vect.dz\*dt;

xe=Xe(xt1,yt1);

ye=Ye(xt1,yt1,zt1);

x1=xn(xe);

y1=ym(ye);

xe=Xe(xt2,yt2);

ye=Ye(xt2,yt2,zt2);

x2=xn(xe);

y2=ym(ye);

MoveToEx(hdc,x1,y1,NULL);

LineTo(hdc,x2,y2);

vect.x = xt2;

vect.y = yt2;

vect.z = zt2;

}while((x>-xmax)&&(x<xmax)&&(y>-ymax)&&(y<ymax)&&(z>-zmax)&&(z<zmax));

}

10.Привести код рисования стрелок на линиях поля.

Приводится код функции arrowVector, рисующий стрелку в виде уголка, на заданном отрезке.

void arrowVector(HDC hdc, int x1, int y1, int x2, int y2, COLORREF rgb)

{

int k = 5; //размеры стрелки в пикселях

double d = sqrt((x2-x1)\*(x2-x1) + (y2-y1)\*(y2-y1));

double nx = (x2-x1)/d, ny = (y2-y1)/d;

double mx = -ny, my = nx;

int x3, y3, x4, y4, x0, y0;

x0 = int(k\*nx) + x1;

y0 = int(k\*ny) + y1;

x3 = int(k\*mx) + x1;

y3 = int(k\*my) + y1;

x4 = -int(k\*mx) + x1;

y4 = -int(k\*my) + y1;

HPEN hPen = CreatePen(PS\_SOLID,2,rgb);

HPEN hPenOld = (HPEN)SelectObject(hdc,hPen);

MoveToEx(hdc,x3,y3,0);

LineTo(hdc,x0,y0);

LineTo(hdc,x4,y4);

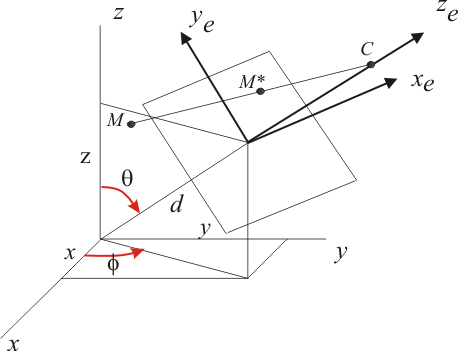
SelectObject(hdc,hPenOld);

DeleteObject(hPen);

}

17.Как создается матрица центрального проецирования на видовую плоскость.

Показан механизм центрального проецирования.



Мировые координаты  и видовые координаты Центральная проекция точки *M* на видовую плоскость  в точку *M*\*. Точка *C* центр центральной проекции.

Матрицу видового преобразования  теперь надо слева умножить на матрицу центрального проецирования.



Полученные с помощью матрицы  координаты центральной проекции в конце необходимо разделить на следующий параметр *u*.



Поэтому преобразование координат точки *M* в координаты точки *M*\*, осуществляется следующими формулами.



Ниже приводится код функций , осуществляющих переход от мировых координат к видовым координатам и выполняющих центральное проецирование на видовую плоскость.

//переход из мировой системы координат в видовую

//центральное проектирование

inline double Xe(double x,double y,double z)

{

double u =(dCentral + cCentral -(st\*cf\*x+st\*sf\*y+ct\*z))/cCentral;

return (-sf\*x+cf\*y)/u;

}

//переход из мировой системы координат в видовую

// центральное проектирование

inline double Ye(double x,double y,double z)

{

double u =(dCentral + cCentral -(st\*cf\*x+st\*sf\*y+ct\*z))/cCentral;

return (-ct\*cf\*x-ct\*sf\*y+st\*z)/u;

}

23.Как выглядит код ограничения выхода линии электрического поля за границы куба видимости.

//рисует одну линию поля из начальной точки PointB

void LineField(HDC hdc,POINT3 PointB)

{

VECTORS vect;

vect.x = PointB.x;

vect.y = PointB.y;

vect.z = PointB.z;

//видовые координаты проецируемой точки

double xe, ye;

//координаты пикселов

int x1,y1,x2,y2;

double dt = 0.01;

double x, y, z, Hx, Hy, Hz, Ha;

VECMAG mag;

double xt1,yt1,zt1,xt2,yt2,zt2;

do

{

x = vect.x;

y = vect.y;

z = vect.z;

mag = magn(x,y,z);

Hx = mag.hx;

Hy = mag.hy;

Hz = mag.hz;

Ha = sqrt(Hx\*Hx + Hy\*Hy + Hz\*Hz);

vect.dx = Hx/Ha;

vect.dy = Hy/Ha;

vect.dz = Hz/Ha;

xt1 = vect.x; yt1 = vect.y; zt1 = vect.z;

xt2 = xt1 + vect.dx\*dt;

yt2 = yt1 + vect.dy\*dt;

zt2 = zt1 + vect.dz\*dt;

xe=Xe(xt1,yt1);

ye=Ye(xt1,yt1,zt1);

x1=xn(xe);

y1=ym(ye);

xe=Xe(xt2,yt2);

ye=Ye(xt2,yt2,zt2);

x2=xn(xe);

y2=ym(ye);

MoveToEx(hdc,x1,y1,NULL);

LineTo(hdc,x2,y2);

vect.x = xt2;

vect.y = yt2;

vect.z = zt2;

}while((x>-xmax)&&(x<xmax)&&(y>-ymax)&&(y<ymax)&&(z>-zmax)&&(z<zmax));

}

В данном случае, величина куба определяется величинами xmax, ymax и zmax. В приведенном коде выполнение отрисовки прекращается при приближении значения одной из координатных компонент к границе куба.

27.В каком файле находятся идентификаторы элементов диалогового окна.

Ответ: Идентификаторы элементов диалогового окна находятся в файле resource.h