# Тюльников Михаил ПИН-21М ДЗ-3

# Вариант 18.

**Вопрос 4: Как создается матрица ортографического проецирования на видовую плоскость.**

**Ответ:** Матрица ортографического проецирования на видовую плоскость строится путем перехода от мировых координат к видовым. Этот переход описывается с помощью матрицы, которая применяется к мировым координатам объекта. В результате ортографического проецирования точка в мировых координатах преобразуется в точку в видовых координатах. Соответствующая матрица проецирования и видового преобразования комбинируется, что позволяет выполнять преобразование точек сцены из 3D в 2D.

double Xe(double x, double y) {

return -sf \* x + cf \* y;

}

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

return -ct \* cf \* x - ct \* sf \* y + st \* z;

}

**Вопрос 11: Какова схема реализации метода Z-буфера**

**Ответ:** Метод Z-буфера используется для удаления невидимых элементов в 3D-сцене, чтобы ближние объекты заслоняли дальние. Для каждого пикселя в Z-буфере сохраняется значение глубины z (расстояние от камеры до объекта). Когда объект отрисовывается, проверяется его глубина: если она меньше текущей записанной в Z-буфере, то обновляются как глубина, так и цвет пикселя. Иначе, этот объект игнорируется, поскольку он находится за уже отрисованными объектами.

struct ZbuffS {

double z;

ColorS c;

};

if (zb[p].z < ze) {

zb[p].z = ze;

zb[p].c.R = r;

zb[p].c.G = g;

zb[p].c.B = b;

}

**Вопрос 18: Как выглядит код функций, осуществляющих центральное проецирование.**

**Ответ:** Центральное проецирование отличается тем, что точки на 3D-сцене проецируются с учетом перспективы. Это означает, что более дальние объекты кажутся меньше, чем близкие. Преобразование включает деление координат на параметр u, который зависит от расстояния от камеры до объекта.

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;

}

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;

}

**Вопрос 25: Как можно изменить цвета надписей на диалоговых окнах.**

**Ответ:** Цвета надписей в диалоговых окнах можно изменить через API-функции WinAPI, которые позволяют управлять элементами управления (controls) и их отображением. Например, при создании диалогового окна можно задать соответствующие свойства текста.

В коде функции обработки сообщений от диалоговых окон для изменения текста может использоваться функция SetWindowText, где можно задать текст или цвет надписей через параметры окна.

SetWindowText(GetDlgItem(hwndDlg, IDC\_EDIT1), "New Text");

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

**Ответ:** Для построения линии поля, выходящей из заданной точки, можно использовать следующую функцию, которая вычисляет векторы магнитного поля и шаг за шагом строит линию.

void LineField(HDC hdc, POINT3 PointB) {

VECTORS vect;

vect.x = PointB.x;

vect.y = PointB.y;

vect.z = PointB.z;

double dt = 0.01;

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

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;

vect.x += vect.dx \* dt;

vect.y += vect.dy \* dt;

vect.z += vect.dz \* dt;

// Рисование линии

MoveToEx(hdc, x1, y1, NULL);

LineTo(hdc, x2, y2);

} while (в пределах области);

}