Предварительные замечания:

1. Описание алгоритма состоит из описания четырех нижеприведенных процедур (четвертая процедура — дополнительная, на которую нет прямой ссылки в алгоритме).

2. Предполагается, что к началу исполнения алгоритма был произведен переход к экранной системе координат и проведена операция кадрирования, в ходе которой координата z каждой точки (координата в экранной системе координат) не отбрасывается, а остается неизменной. Т. е. после перехода к экранной системе координат выполняется преобразование:

$$\begin{bmatrix} \chi' \\ \gamma' \\ \zeta' \\ \alpha' \end{bmatrix} = \begin{bmatrix} W_x/2 & 0 & 0 & W_{cx} + W_x/2 \\ 0 & -W_y/2 & 0 & W_{cy} - W_y/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \chi \\ \gamma \\ \zeta \\ \alpha \end{bmatrix}$$

3. В описании алгоритма подразумевается, что область рисования имеет координаты по x от W_{cx} до $W_{cx}+W_x$, по y от W_{cy} до $W_{cy}+W_y$, где W_{cx}, W_x, W_{cy}, W_y — неотрицательны.

4. При сравнении значений координаты y на равенство следует округлять эти значения до целого числа.

5. Предполагается, что каждый многоугольник P_i в списке $\mathscr{P}$ представлен семеркой (L, n, C, a, b, c, d), где L — список вершин многоугольника при обходе в некотором порядке, n — количество вершин, C — цвет многоугольника, a, b, c, d — коэффициенты уравнения несущей плоскости (для их вычисления можно использовать приведенный ниже третий алгоритм).

Алгоритм 1: Алгоритм отсечения невидимых граней Ray Tracking

Вход: $\mathscr{P}$ — список многоугольников трехмерной сцены.

начало алгоритма

цикл для x *от* W_{cx} *до* $W_{cx} + W_x$ **выполнять**

цикл для y *от* W_{cy} *до* $W_{cy} + W_y$ **выполнять**

· $Color =$ цвет фона; $Z_b = -1$;

цикл для каждого $P = (L, n, C, a, b, c, d)$ *в* $\mathscr{P}$ **выполнить**

· **если** SurroundTest(x, y, P) **то**

Вычислить $z =$ Depth(x, y, P);

· **если** $Z_b \leqslant z \leqslant 1$ **то** Присвоить $Z_b = z$, $Color = C$;

· Закрасить точку (x, y) цветом $Color$;

конец алгоритма

Алгоритм 2: SurroundTest Проверка на принадлежность точки многоугольнику

Вход: x, y — координаты точки, P — список вершин выпуклого многоугольника.

Выход: $True$ — Если точка с координатами x и y лежит внутри многоугольника.

начало алгоритма

Присвоить переменной n количество элементов в списке P, $i = 1$;

$(x_1, y_1, z_1) = P[n]$, $Sgn_1 = 0$;

· **цикл пока** $Sgn_1 = 0$ **выполнять**

$(x_2, y_2, z_2) = P[i]$;

Вычислить $Sgn_1 = signum((x_2 - x_1)(y - y_1) - (y_2 - y_1)(x - x_1))$;

Присвоить $x_1 = x_2$, $y_1 = y_2$, $i = i + 1$;

· **цикл пока** $i \leqslant n$ **выполнять**

$(x_2, y_2, z_2) = P[i]$;

Вычислить $Sgn_2 = signum((x_2 - x_1)(y - y_1) - (y_2 - y_1)(x - x_1))$;

· **если** $Sgn_2 \neq 0$ *и* $Sgn_2 \neq Sgn_1$ **то** вернуть $False$; **закончить алгоритм**;

Присвоить $x_1 = x_2$, $y_1 = y_2$, $i = i + 1$;

Вернуть $True$;

конец алгоритма

Алгоритм 3: Depth Вычисление глубины точки на несущей плоскости

Вход: x, y — координаты точки, P — идентификатор многоугольника.

Выход: z — координата z точки на несущей плоскости многоугольника P, имеющая координаты x и y.

начало алгоритма

$$z = -\frac{ax + by + d}{c}$$

конец алгоритма

Алгоритм 4: InitializePolygon Вычисление коэффициентов уравнения плоскости

Вход: P — список вершин многоугольника в трехмерном пространстве

Выход: a, b, c, d — коэффициенты уравнения несущей плоскости

начало алгоритма

Пусть (x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3) — первые три вершины в списке P. Предполагаем, что эти вершины не лежат на одной прямой. Тогда

$$a = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix}; \quad b = \begin{vmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{vmatrix}; \quad c = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}; \quad d = -\begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}.$$

конец алгоритма