Компьютерная графика

Harpie

February 2025

17.02.25

1 Собъективно-ориентированное программирование

Это парадигма программирования которая представляет отрезок вменении во время которого возникают определенные события. Программа выполняется не линейно, программа описывается как набор обработчиков/ обработчик события.

Некоторые события происходят автоматически, какие то нужно/можно инициализировать.

Примеры обработчиков:
Paint
Load
Resize

Некоторые обработчики могут быть инициализированы
Refresh

2 Полигональная КГ

Объект описывается с помощью вершин/точек, которые соединенны отрезками. Перед описанием объекта нужно знать и задать координаты точек.

3 Модели

Модель строиться набором отрезков, которые определенным образом соединенны в точках, в некоторой системе координат. **Модельная система координат, объектная система координат,локальная система координат**

Правая система координат - система повернута против часовой стрелки под 90 градусов

Система координат экрана - (нужно заполнить тут)

Правая декартовая СК (мировая система координат) - "виртуальный мир в котором существеют модели.

4 Кадрирование

Операция кадрирования - совмещенние одного кадра с другим, преобразования одного кадра

Кадр - прямоуглник, стороны которого парарлельны осям координат и у которого есть параметры

Размер по горизонтале - Vx

Размер по вертикале - Vy

Координты нижного угла - Vcx, Vcy

Исходный кадр обозоначается как - Wx, Wy, Wcx, Wcy

Безразмерные координаты :

$x_1 = x - V_c x$ $\quad y_1 = y - V_c y$

$x_2 = \frac{x - V_c x}{V_x}$ $\quad y_2 = \frac{y - V_c y}{V_y}$

$x_3 = \frac{x - V_c x}{V_x} * W_x$ $\quad y_3 = \frac{y - V_c y}{V_y} * W_y$

$x' = \frac{x - V_c x}{V_x} * W_x + W_c x$ $\quad y' = \frac{y - V_c y}{V_y} * W_y + W_c y$

Если умножить на - 1 еденицу, то картинка перевернется (касается перевернутых картинок)

$x_4 = \frac{x - V_c x}{V_x} * W_x + W_c x$ $\quad y_4 = \frac{y - V_c y}{V_y} * W_y - W_c y$

$x_5 = \frac{x - V_c x}{V_x} * W_x + W_c x$ $\quad y_5 = \frac{y - V_c y}{V_y} * W_x + 2W_c y$

$x' = \frac{x - C_c x}{V_x} * W_x + W_c x$ $\quad y' = W_c y - \frac{x - C_c x}{V_x} * W_y$

24.02.25

Продолжение

$x' = \frac{x - V_{cx}}{V_x} * W_x + W_c$

$y' = W_y - \frac{y - V_{cy}}{V_y} * W_y$

5 Преобразование изображения

5.1 Элементарные преобразования

1. Перенос/Сдвиг

Смысл преобразования: объект в одной системе координат нужно сдвинуть в "другую систнму координат".

Пример:

$x_1 \implies x'_1$ между ними расстояние T_x $\qquad y_1 \implies y'_1$ между ними расстояние T_y

$x_2 \implies x'_2$ между ними расстояние V_x $y_2 \implies y'_2$ между ними расстояние V_y

Итог: $x' = x + T_x$ $y' = y + T_y$

2. Масштабирвоание относительно начала координат

Это озночает, все точки преобразуются за исключением начальной точки. На месте остается только точка начала координат.

Пример:

$3 \implies 4.5$ и т.д

Итог: $x' = x * S_x$ $y' = y * S_y$

3. Поворот относительно начала координат против часовой стрелки на угол ϑ

Нужно соеденить лучом точку с началом координат. Преоброзовать точку означает, повернуть этот отрезок на заданный угол ϑ. В результате полчвется новая точка.

$x' = rcos(\alpha + \vartheta)^x = rcos\alpha cos\vartheta - rsin\alpha sin\Theta = xcos\vartheta - ysin\vartheta$

$y' = rsin(\alpha + \vartheta)^x = rsin\alpha cos\vartheta + rsin\vartheta cos\alpha = ycos\vartheta + xsin\vartheta$

Итог : $x' = xcos\vartheta - ysin\vartheta$ $y' = ycos\vartheta + xsin\vartheta$

4. Зеракальное отражение. Частный случай масштабирвоание

$x' = x * -1$ $y' = y * -1$

6 Совмещенние преобразований

Пример: поворот на угол ϑ против часовой стрелки относительно т. $A(x_a, y_a)$

$x^{(1)} = x - x_a$ $y^{(1)} = y - y_a$

$x^{(2)} = x^{(1)}cos\vartheta - y^{(2)}sin\vartheta$ $y^{(2)} = x^{(1)}csin\vartheta + y^{(2)}cos\vartheta$

$x' = x^{(2)} + x_a = (x - x_a)cos\vartheta - (y - y_a)sin\vartheta + x_a$

$y' = y^{(2)} + y_a = (x - x_a)csin\vartheta + (y - y_a)cos\vartheta + y_a$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

6.1 Однородные координаты

6.1.1 Евклидовые координаты

В однородных координатах координаты объекта задается с точностью какого то множетеля. Например, для прямой вида $A_x + B_y + C = 0$. Можно скзаать, что координаты прямой задаются тройкой (A, B, C). В таком случае, если домножить эту тройку, то она все еще будет указывать на иходную прямую.

Для евклидовой точки (x, y) выберем некоторую произвольную

Переход от однородных в евклидовые координаты

Предположим были однородные координаты (χ, γ, α) и нам нужно получить $(\chi', \gamma', \alpha')$

6.2 Однородные преобразоавания

Формула для масштабирвоание

$\chi' = \chi S_\chi$

$\gamma' = \gamma S_\gamma$

$\alpha' = \alpha$

Матрица масштабирвоания

$$\begin{bmatrix} \chi' \\ \gamma' \\ \alpha' \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \chi \\ \gamma \\ \alpha \end{bmatrix}$$

Матрица поворота

$$\begin{bmatrix} \chi' \\ \gamma' \\ \alpha' \end{bmatrix} = \begin{bmatrix} cos\vartheta & -sin\vartheta & 0 \\ sin\vartheta & cos\vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \chi \\ \gamma \\ \alpha \end{bmatrix}$$

$\chi' = \alpha x' = \alpha x + Y_x \alpha$

$\gamma' = \gamma x' = \gamma x + Y_x \alpha$

$\alpha' = \alpha$

Матрица для переноса

$$\begin{bmatrix} \chi' \\ \gamma' \\ \alpha' \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \chi \\ \gamma \\ \alpha \end{bmatrix}$$

Универсальная формула

$$\begin{bmatrix} \chi' \\ \gamma' \\ \alpha' \end{bmatrix} = \begin{bmatrix} cos\vartheta & -sin\vartheta & 0 \\ sin\vartheta & cos\vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \chi \\ \gamma \\ \alpha \end{bmatrix}$$

03.03.25

$P' = MP$ В таком виде записывается матричное преообразование

P - столбец

M - матрица

P$= \begin{bmatrix} \chi \\ \gamma \\ \alpha \end{bmatrix}$ P$= \begin{bmatrix} \chi' \\ \gamma' \\ \alpha' \end{bmatrix}$

M' - обратная матрица

$P = M'(MP)$

$Rightarrow M' = M^-1$

6.3 Преобразование перенос

Translate $(T_x, T_y) = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -T_x \\ 0 & 1 & -T_y \\ 0 & 0 & 1 \end{bmatrix}$

T(T_x, T_y)

6.4 Преобразование масштабирвоание

Scale (S_x, S_y) $\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/S_x & 0 & 0 \\ 0 & 1/S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$

S(S_x, S_y)

6.5 Преобразование поворот

Rotate$(\vartheta) = \begin{bmatrix} cos\vartheta & -sin\vartheta & 0 \\ sin\vartheta & cos\vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\vartheta & sin\vartheta & 0 \\ -sin\vartheta & cos\vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

7 Совмещенное преобразование

Смысл:

$P' = (M_2 ... M_3, M_2, M_1)P$

$E = M'(M_3, M_2, M_1)$

$M' = M'(M_\Delta^{-1}, M_2^{-1}, M_3^{-1})$

8 Принцип двойтсвенности

Двойственное преобразование

У нас есть матрица

$\begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix}$

Масштабирвоание равно преобразованию....

Напоминание: Операция кордирования $x' = \frac{x - V_{cx}}{V_x} W_x + W_{cx}$

$y' = \frac{y - V_{cy}}{V_y} W_y + W_{cy}$

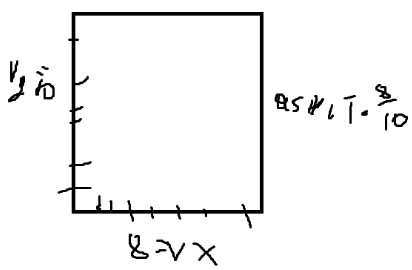
$\frac{W_x}{V_x} < \frac{W_y}{V_y}$

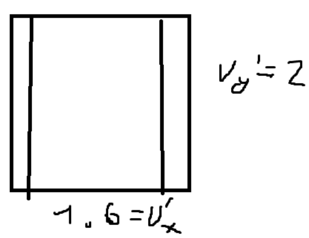
$\frac{2}{V_x} < \frac{2}{V_y}$

$1 = \frac{2}{2} < \frac{V_x}{V_y}$ - aspect - соотношение сторон

Если соотношение сторон больше 1 масштабируем по x, если же наоборот то по y. (Для вписания картинки в квадрат)

$\frac{W_x}{V_x}$ или $\frac{W_y}{V_y}$
$\frac{2}{V_x}$ или $\frac{2}{V_y}$
$$\begin{cases} V_x^{'} \\ \\ V_y^{'} \end{cases} \quad \text{Размеры модели после вписания в квадрат 2x2}$$

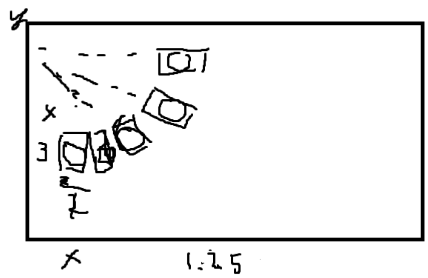




Команды:
translate x y
rotate ϑ
scale S
figure
Вспомогательные команды
popTransform - забираем из стка
pushTransform - добавляем в стек

scale 1.25
pushTransform
translate 0 -R
translate x y

translate -x -y
rotate 22.5
translate x y



Сокращение картинки:
scale 1.25
translate 0 -R
pushTransform
translate x y
figure
popTransform
rotate 22.5
pushTransform
translate x Y
figure

$$\left\{\begin{array}{l} pushTransform \\ rotate 22.5 \\ pushTransform \\ translatexy \\ figure \end{array}\right.$$

popTransform
10.03.25

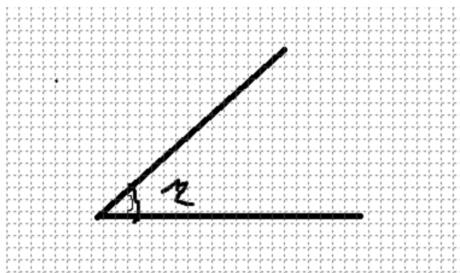
9 Операции над векторами

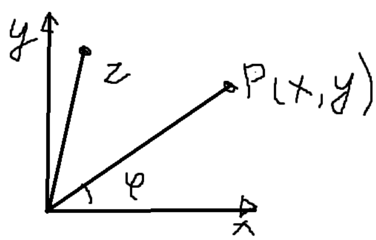
9.1 Двумерные вектора

У веторов две характиристики:
длина
$+$
направлнение

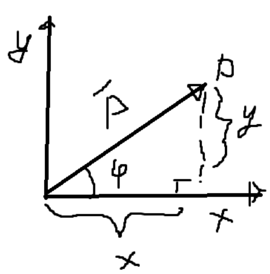
Подобные вектора называется свободными векторами если, есть точка из которой он начинается, то он называется связанным

Угол вектора (r, φ)
Угол ветора против часовой стрелки



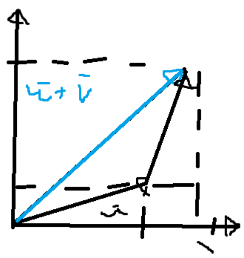


Радиус вектор точка P
$\bar{P}(x, \varphi)$
$x = rcos\varphi$
$y = rsin\varphi$

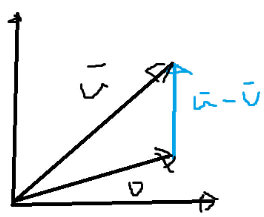


Сложение векторов(сумма)
$\bar{u} u \bar{v}$
$\bar{u} + \bar{v}$
$\bar{u} = (u_x, u_y)$

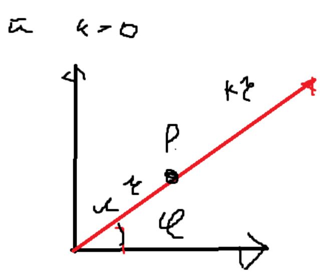
$\bar{v} = (v_x, v_y)$
$\bar{u} + \bar{v} = (u_x + v_x, u_y + v_y)$



Разность векторов $\bar{u}u\bar{v}$
$\bar{u} - \bar{v}$
$\bar{u} - \bar{v} = (u_x - v_x, u_y - v_y)$

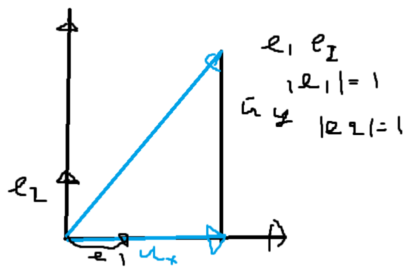


пупуп

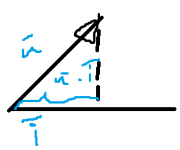


$k\bar{u} = (ku_x, ku_y)$
Скалярное произведение векторов

$\bar{u}\bar{v}$
$\bar{u} * \bar{v} = |\bar{u}||\bar{v}|cos\widehat{\bar{u}\bar{v}}$
$\bar{u} = (r, \varphi)$
$|\bar{u}| = r$
$\bar{u} = u_x \bar{+} \bar{u_y}$
$\bar{u} * \bar{v} = (u_x \bar{+} \bar{u_y}) * (\bar{v_x} + \bar{v_y}) = (|u_x|e_1 + |u_y|e_2) * (|v_x|e_1 * |v_x|e_1)$
$\bar{u} * \bar{v} = |\bar{u_x}||\bar{v_x}||\bar{u_x} + |\bar{v_y}|$ - основная формула

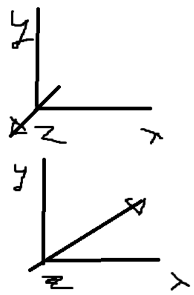


$\bar{u}\bar{i}$
$\bar{u} * \bar{i} = |\bar{u}||\bar{i}|cos\widehat{\bar{u}\bar{i}} = |\bar{u}|cos\widehat{\bar{u}\bar{i}}$

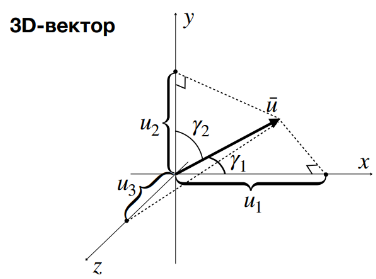


dot-prodect - Скалярное произведение
Псевдоскалярное произведение
$\bar{u}\bar{v}$
$\bar{u}x\bar{v} = |\bar{u}|\bar{v}|sin < \bar{u}\bar{v}$
$\bar{u}x\bar{v} = -\bar{u}x\bar{v}$
$e_1xe_1 = 0e_1xe_2 = 1e_2xe_1 = -1e_2xe_2 = 0 = u_xv_y - u_yv_x > \bar{u}x\bar{v} = \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix}$
cross-prodect - Псевдоскалярное произведение

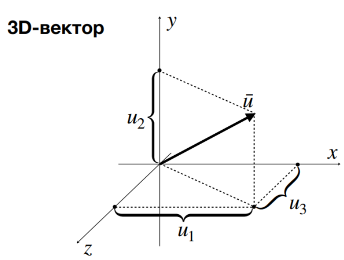
9.2 Трехмерные вектора



Левосторонняя и правостороняя

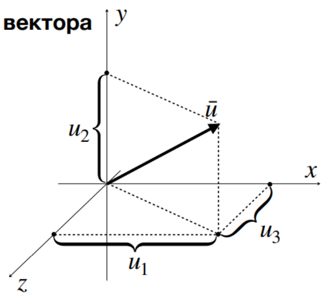


$$u_1 = |\bar{u}|\cos\gamma_1$$
$$u_2 = |\bar{u}|\cos\gamma_2$$
$$u_3 = |\bar{u}|\cos\gamma_3$$



$$\bar{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = [u_1 \quad u_2 \quad u_3]^T = (u_1, u_2, u_3)$$

Длина вектора

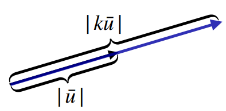


$$|\bar{u}| = \sqrt{u_1^2 + u_2^2 + u_3^2}$$

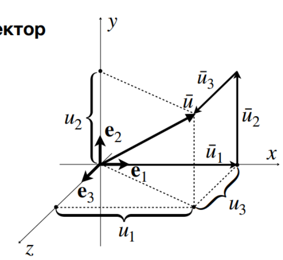
Умножение вектора на скаляр

$$k\bar{u} = \begin{bmatrix} ku_1 \\ ku_2 \\ ku_3 \end{bmatrix} = \begin{bmatrix} ku_1 & ku_2 & ku_3 \end{bmatrix}^T = (ku_1, ku_2, ku_3)$$

$$|k\bar{u}| = \sqrt{(ku_1)^2 + (ku_2)^2 + (ku_3)^2} = k\,|\bar{u}|$$



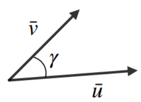
3D-вектор



$$\bar{u}_1 = u_1\mathbf{e}_1 \qquad \bar{u} = \bar{u}_1 + \bar{u}_2 + \bar{u}_3$$
$$\bar{u}_2 = u_2\mathbf{e}_2$$
$$\bar{u}_3 = u_3\mathbf{e}_3 \qquad \bar{u} = u_1\mathbf{e}_1 + u_2\mathbf{e}_2 + u_3\mathbf{e}_3$$

Сложение вектора по координатам

!Скалярное произведение аналогично двухмерному



$$\bar{u} \cdot \bar{v} = |\bar{u}||\bar{v}|\cos\gamma$$

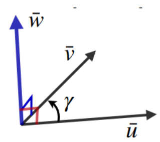
$$\bar{u} \cdot \bar{v} = u_1 v_1 + u_2 v_2 + u_3 v_3$$

$$\bar{u} \cdot \bar{v} = [u_1 \quad u_2 \quad u_3] \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \bar{u}^T \bar{v}$$

$$\bar{u} \cdot \bar{v} = [v_1 \quad v_2 \quad v_3] \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \bar{v}^T \bar{u}$$

Через матрицу

Векторное произведение



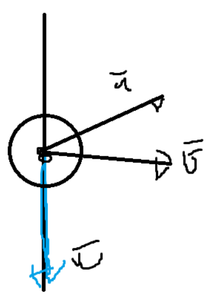
$$\bar{w} = \bar{u} \times \bar{v}$$

$$|\bar{u} \times \bar{v}| = |\bar{u}||\bar{v}|\sin\angle\gamma$$

$$\bar{w} \perp \bar{u} \qquad \bar{w} \cdot \bar{u} = 0$$

$$\bar{w} \perp \bar{v} \qquad \bar{w} \cdot \bar{v} = 0$$

Угол может быть как положительным, так и отрицательным
Не важно в какую сторону отмереяться угол
Если положительный результат, то вверх, отрицательный же вниз



$$\bar{u} \times \bar{v} = M\bar{v} = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

$$[\bar{u}]_\times = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}$$

$$[\bar{u}]_\times \bar{v} = \bar{u} \times \bar{v}$$

Матрица векторного произведения

17.03.25

10 Одномерноые координаты

$e = (A, B, C)(0, 0, 3)$

$A_x + B_y + C = 0$

$P\ (\chi, \gamma, \alpha) = (\frac{\chi}{\alpha}, \frac{\gamma}{\alpha})$

$(2, 3, 0)$

$A_\chi + B_\gamma + C + \alpha = 0$

$(A, B, C)(\chi, \gamma, \alpha) = 0$

$P_1(\chi_1, \gamma_1, \alpha_1)$ и $P_2(\chi_2, \gamma_2, \alpha_2)$

$P_1 x P_2 = e$

$e_1 x e_2 = P$

Одномерноые координаты не будут использоваться в этом курсе, но пусть будут как факт

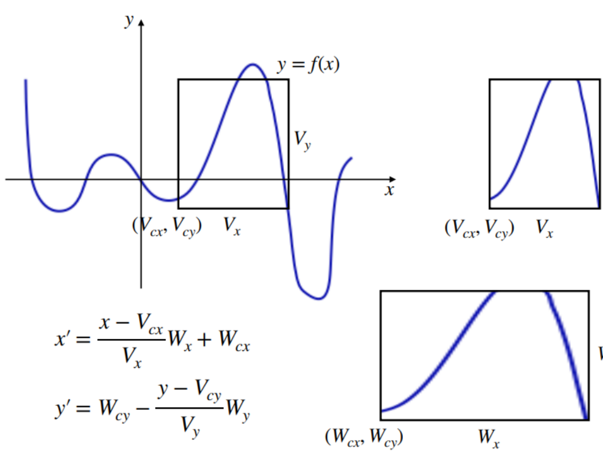
Плюкерова координата - $\beta = LP$

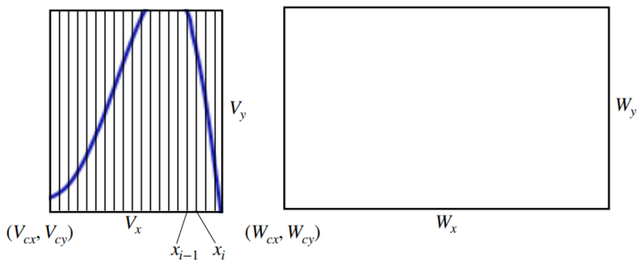
L - представление как матрицы 4x4

Все выше перечисленное не бует на экзамене, это чисто для общего развитися

11 Построение 2D - графика

Задана некая функция, нужно изобразить на экране. Функция бесконечная, как и графика. Нужно определить кадр, выделить его. Здаем кадр на экране.





$x_i = x_{i-1} + \Delta_x$

$$x' = \frac{x - V_{cx}}{V_x} W_x + W_{cx}$$

$$y' = W_{cy} - \frac{y - V_{cy}}{V_y} W_y$$

Δ - шаг

Что получается

$x_0 = V_{cx} \longrightarrow x_0^{'} = W_{cx}$ - кадрирование

$y_0 = f(x_0) \longrightarrow y_0^{'}$ - кадрирование

$x_i = x_{i-1} + x \longrightarrow x_0^{'}$ - кадрирование

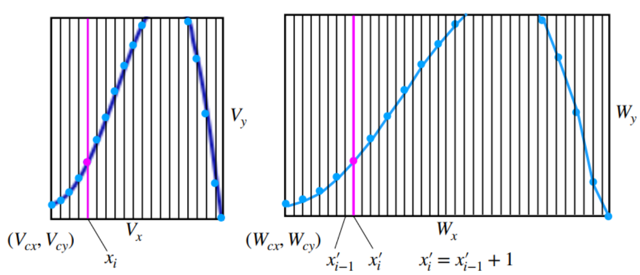
$y_i = f(x_0) + x \longrightarrow y_0^{'}$ - кадрирование

отрезок от $x_{i-1}^{'}, y_{i-1}^{'}$ до $x_i^{'}, y_i^{'}$

цикл $x_i < W_{cx} + W_x$

Как выбрать Δ_x?

Нужно отталкиваться от экрана, в зависимсмоти от этого и выбиратьеся Δ_x. Экран растровое пространство. По сути это сетка - точка растра.



$$x' \rightarrow x = \frac{x' - W_{cx}}{W_x} V_x + V_{cx}$$

$$x' = \frac{x - V_{cx}}{V_x} W_x + W_{cx}$$

$$y' = W_{cy} - \frac{y - V_{cy}}{V_y} W_y$$

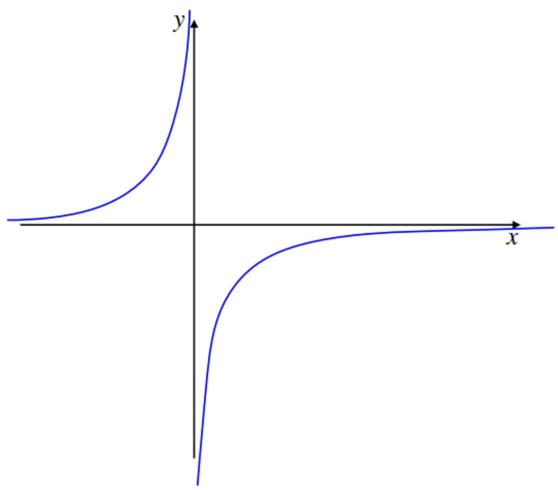
$$x \rightarrow y = f(x)$$

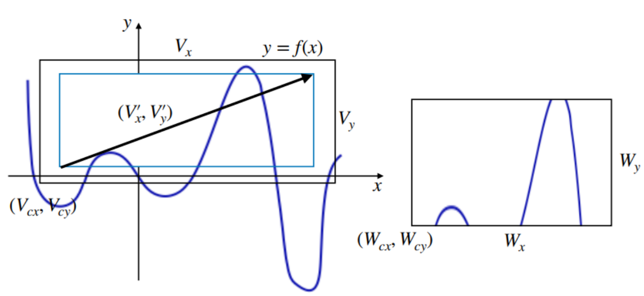
$$y' = W_{cy} - \frac{y - V_{cy}}{V_y} W_y$$

Цикл отриосовки:

пока $x_i^{'} <= W_{cx} + W_x$
$x_i^{'} = x_{i-1}^{'} + 1 \longrightarrow x_i$ кадр
$y_i^{'} \longleftarrow y_i(f_i)$ кадр
обработка от $(x_{i-1}^{'}, y_{i-1}^{'})$ до $(x_i^{'}, y_i^{'})$
$x_i = x_{i-1} + \frac{V_x}{W_x}$

Все усложняется, когда появляется разрыв...





Увеличиваются V_x и/или V_y — график сжимается по оси Ox и/или Oy;
Увеличиваются V_{cx} и/или V_{cy} — график сдвигается влево и/или вниз.

Чтобы растянуть график относительно центра кадра в S_x и S_y раз по осям Ox и Oy, нужно масштабировать окно относительно центра кадра, т. е. применить операции

- масштабирования точки (V_{cx}, V_{cy}) относительно центра окна;
- масштабирования величин V_x, V_y (вектора (V_x, V_y)) в S_x и S_y соответственно.

$$Translate(T_x, T_y) = \begin{pmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{pmatrix} \quad Scale(S_x, S_y) = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$T = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} S_x^\star & 0 & T_x^\star \\ 0 & S_y^\star & T_y^\star \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{bmatrix} V'_{cx} \\ V'_{cy} \\ 1 \end{bmatrix} = \begin{pmatrix} S_x^\star & 0 & T_x^\star \\ 0 & S_y^\star & T_y^\star \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} V_{cx} \\ V_{cy} \\ 1 \end{bmatrix} \quad \begin{bmatrix} V'_x \\ V'_y \end{bmatrix} = \begin{pmatrix} S_x^\star & 0 \\ 0 & S_y^\star \end{pmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix}$$

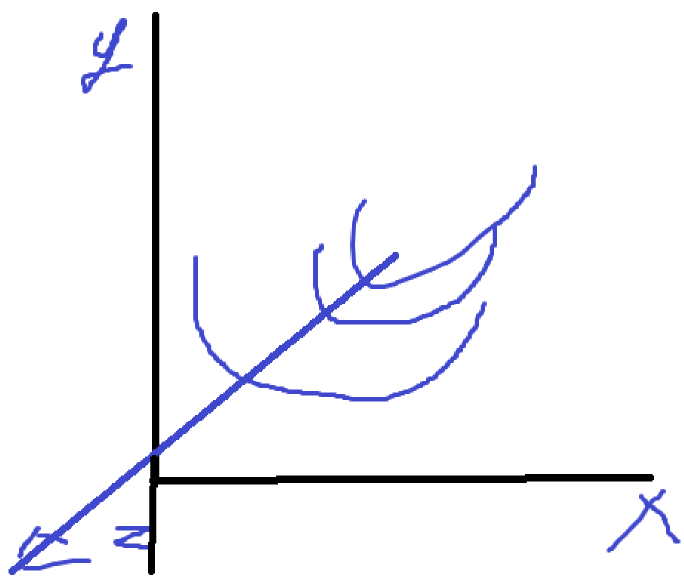
Так будет масштобироваться в окне

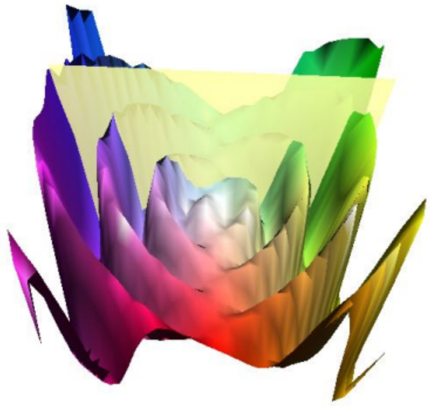
12 Построение 3D-граифка

Трехмерный график формирктся из двумерных графиков

Пусть:

Бужем обозначать $y = f(x, z)$. Теперь график зависит еще и от z

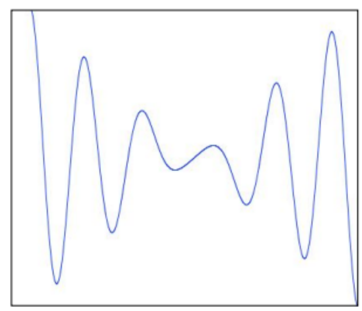




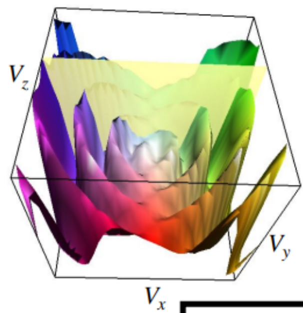
$$y = -x\sin\sqrt{x^2+z^2}$$
$$z = -5$$

$$\Downarrow$$

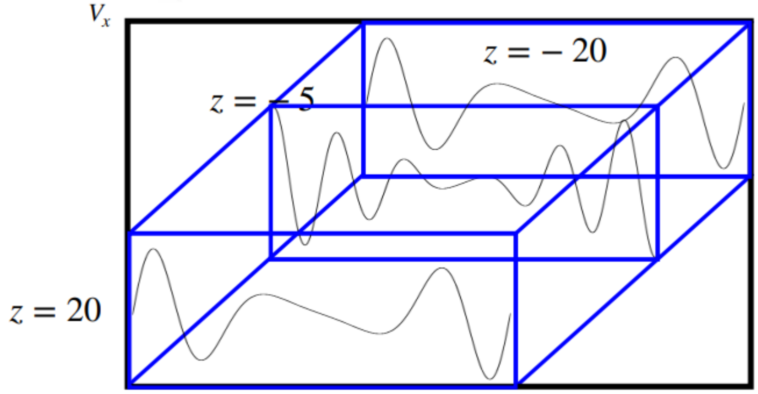
$$y = -x\sin\sqrt{x^2+25}$$



Для рисования такого используется "Алгоритм художника". Вначале начинаем с заднегго плана и от него идет вплоть до самого ближнего плана

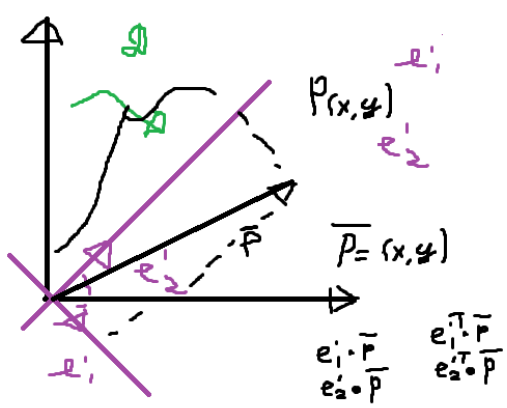


$$y = -x \sin\sqrt{x^2 + z^2}$$

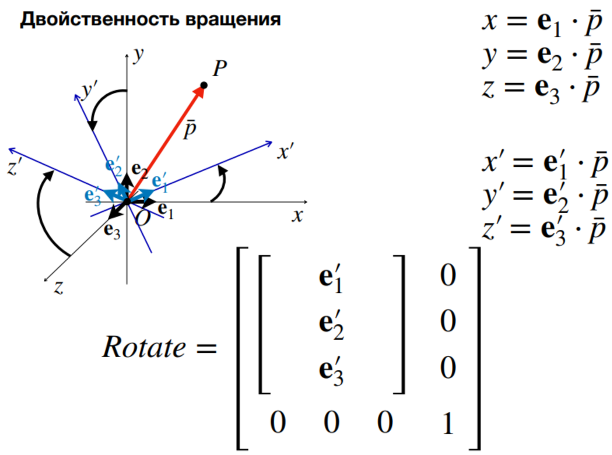


24.03.25
Пропуск лекции из-за досдачи
31.03.25

12.1 3D - преобразоавания



Матричная форма $\bar{p'} = \begin{bmatrix} e_1'^T \\ e_2'^T \end{bmatrix} \bar{p}$



13 Система координат наблюдателя

Замечание

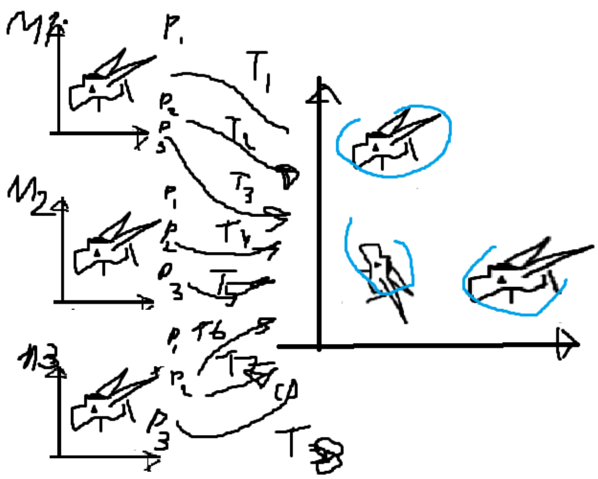
Модель это набор отрезков, на данный момент.

Вершины заданы в какой-то система координат.

Преобразование накапливается.

Для каждого набора точек, своя модель преобразоавания.

Есть неизменяемые вершинные данные, изменяются только преобразования.



Точка наблюдене - точка зрения наблюдателя(camera).

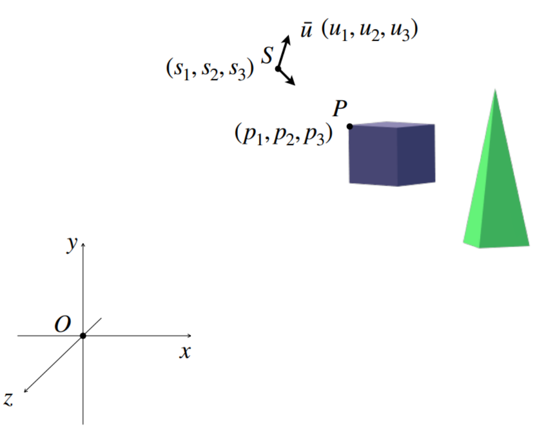
Вектор наблюдения - вектор направленый от взляда наблюдателя в точку наблюдения.

Система координат - правая декартовая система координат, лежащая от глаз наблюдателя

Ось x направленна влевео вниз от наблюдателя

Ось y направленна вверх от наблюдателя

Ось z направленна вправо от наблюдателя

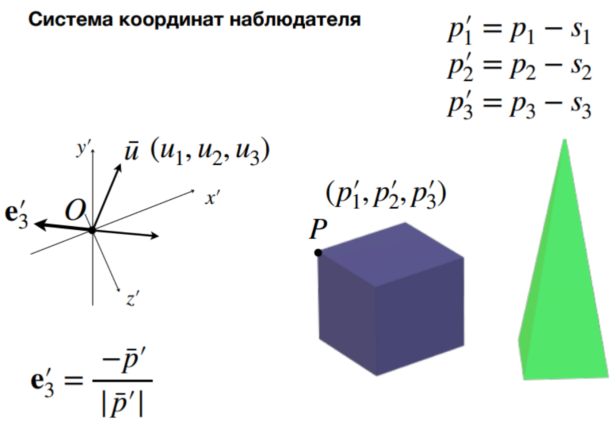


Преобразование:

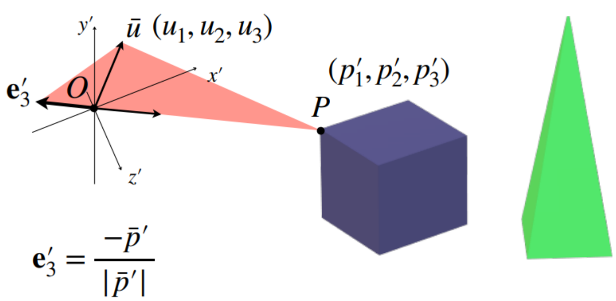
1. Поворот

$$\begin{bmatrix} 0 & 0 & 0 & -S_1 \\ 0 & 1 & 0 & -S_2 \\ 0 & 0 & 1 & -S_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Вращение. Нужно определить матрицу вращения



Определить первый базисный вектор
Определить второй базисный вектор
Определить третий базисный вектор



После всех превращений нужно...

LookAt - получет матрицу вращений. Преобразование системы переходы из мировой системы координат в ситсему координат наблюдателя

Преобразование *LookAt*

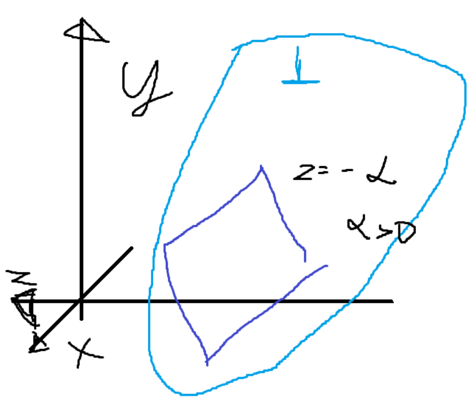
$$\mathbf{e}_3' = \frac{-\bar{p}'}{|\bar{p}'|} \qquad \mathbf{e}_1' = \frac{\bar{u} \times \mathbf{e}_3'}{|\bar{u} \times \mathbf{e}_3'|} \qquad \mathbf{e}_2' = \frac{\mathbf{e}_3' \times \mathbf{e}_1'}{|\mathbf{e}_3' \times \mathbf{e}_1'|}$$

$$LookAt(S, P, \bar{u}) = \begin{bmatrix} \begin{bmatrix} & \mathbf{e}_1' & \\ & \mathbf{e}_2' & \\ & \mathbf{e}_3' & \end{bmatrix} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -s_1 \\ 0 & 1 & 0 & -s_2 \\ 1 & 0 & 0 & -s_3 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Вверху на кортинке опечатка: лишняя единица и на глваной диагонали нехваетает единицы

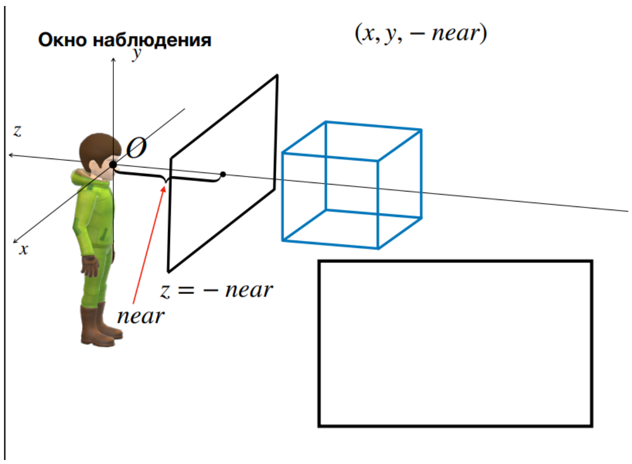
$$\begin{bmatrix} 0 & 0 & 0 & -S_1 \\ 0 & 1 & 0 & -S_2 \\ 0 & 0 & 1 & -S_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Правильная матрица
Плоскость наблюдения (круг).
Окно наблюдения(квадрат) кадр

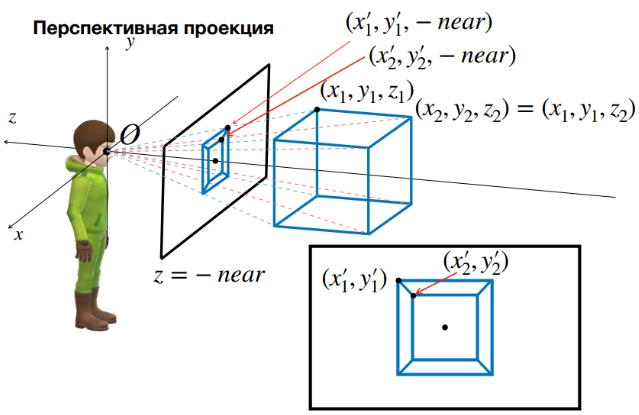


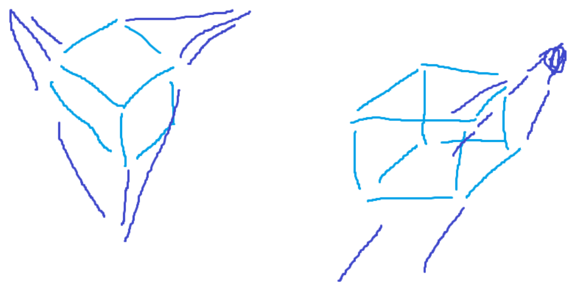
near - расстояние от наблюдателя до окна наблюдения

14 Трехмерные преобазования



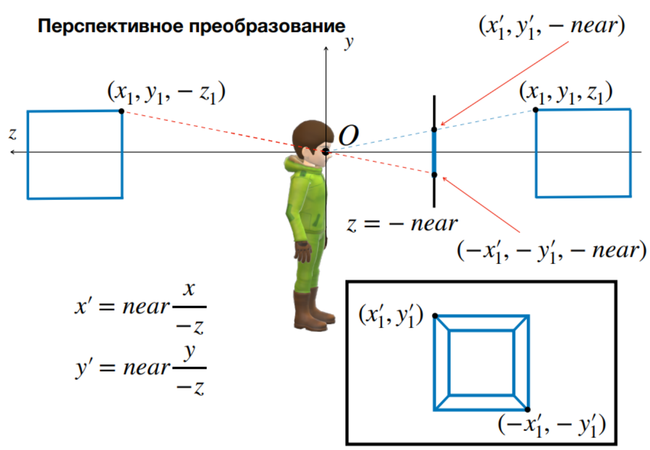
Перспектива - это намерное искажения изображения с целью приданию глубины





14.1 Перспекитвное преобразование

тут должно быть изображение с изображением того что на доске (смотреть в телефоне)



07.04.25

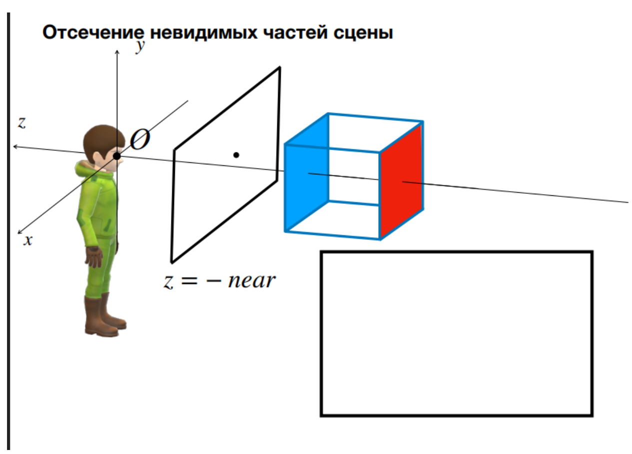
14.2 Отсечение невидмых частей сцены

1) **Проволоченое** изрбражение - это объект заданный отрезками, здесь можно переспективное преобзразование применить, или прямоугольная проекция.

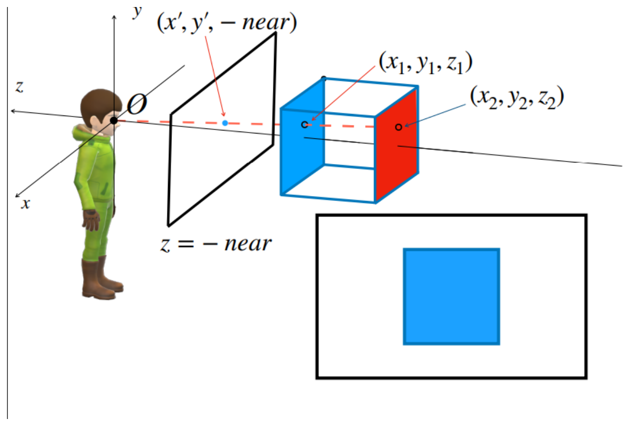
2) Когда изображение задананно гранями оно называется **контурным**

Можно описать грани как: последовательность отрезков, получается мы описываем их как набор многоугольников.

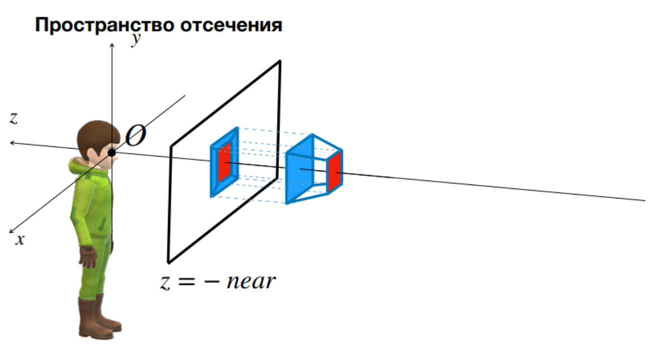
3) **Полутонове** изображение у каждой грани есть атрибуты.



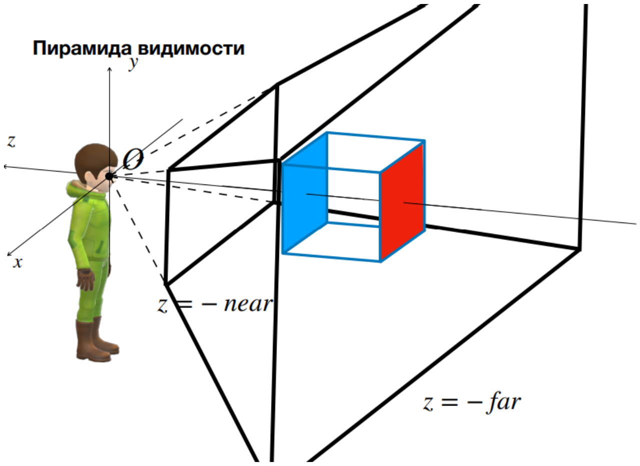
Отсечение



Пространство отсечения



Пирамида видимости



Пространство отсечения

Система координат ростравнсвто отсечения - промежуточная система координат, ... прямоугольная проекция.

Главный вопрос насколько мы далеко "отодвинули "раздвинула"

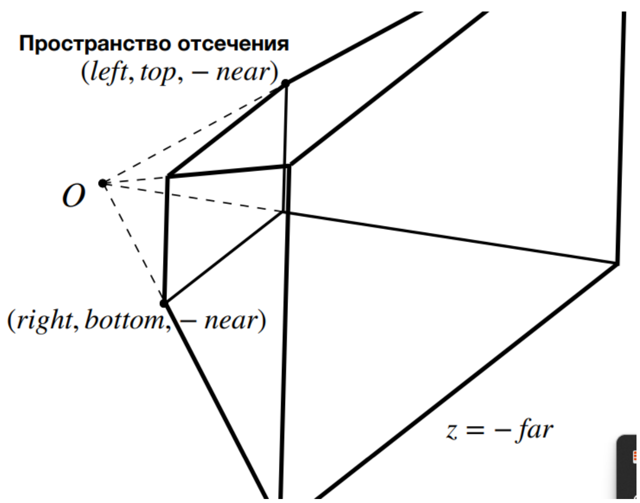
В этом преобзравние оси x и z меняются местами. Система координат левая.

Будем говорить, что наблюдатель видит все что до far, за far его область видимости пропадает.

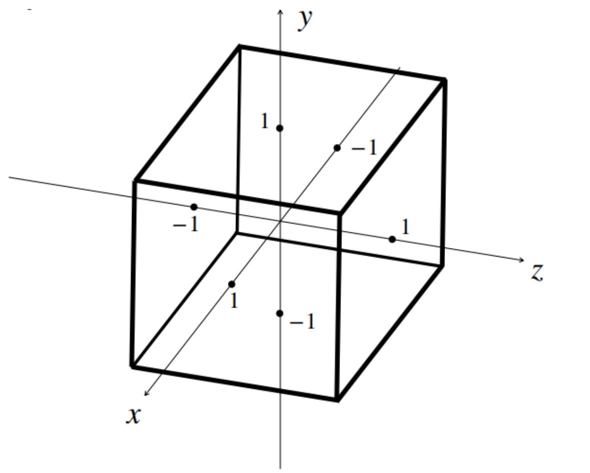
far - расстояние

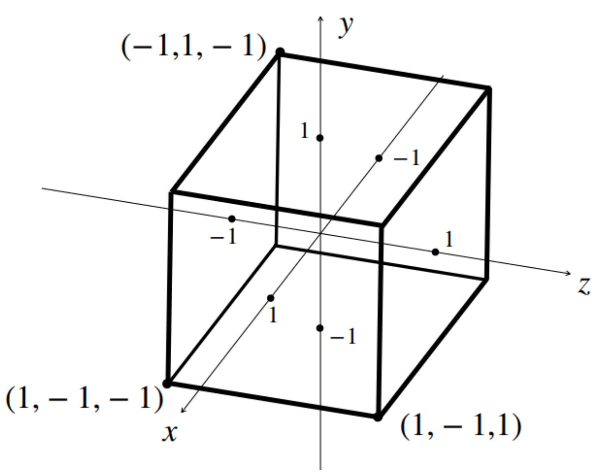
-far - координата

FRUSTUM - усеченная пирамида, по русски это называется пирамида видимости. Она ограничина плоскостью горизонта, областью видимости наблюдателя.



Задаем окно наблюдения - оно ограниченно
left,rirgt - x - параметры наблюдения, координаты
top, bottom - y - параметры наблюдения, координаты
Вся плоскость наблюдения ограниченна -near
near, far - расстояния





Переход в пространтсво отсечения

Если взять окно наблюдения, то оно передет в одну из сторон куба, left,top,-near перейдет в (-1,1,-1), а right,bottom,-near перейдет в (1,-1,-1), z=-far перейдет в z=1

В начале нужно провести оперцию кодрирования

Это переспективное преобразование

$Vcx = left$ $\qquad Wcx = -1$

$Vcy = bottom$ $\qquad Wyx = -1$

$Vx = right - lefft$ $\qquad Wy = 2$

$Vy = top - bottom$ $\qquad Wy = 2$

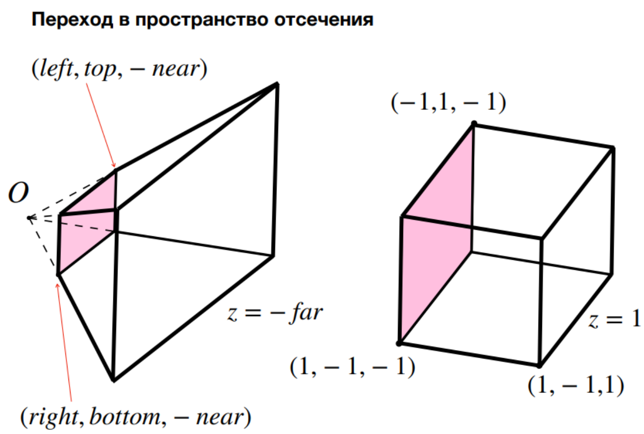
$x - \frac{x' right - lefft}{}$ это недописано

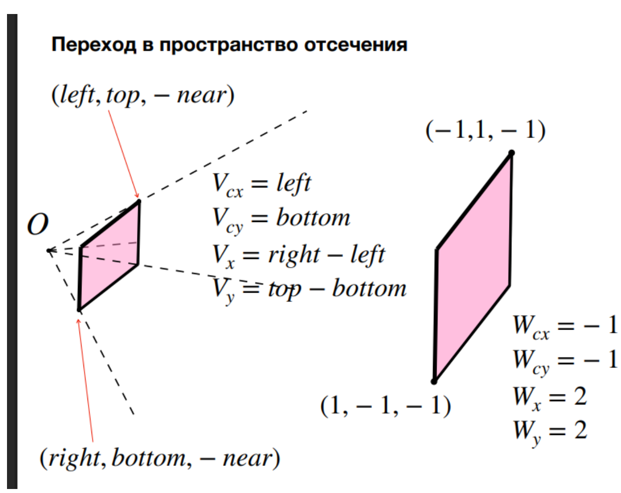
$y - \frac{y' right - lefft}{}$ это недописано

А это переспективное проекция

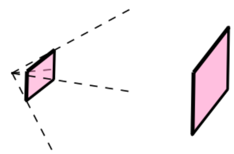
$x - \frac{\frac{nearx}{-z} - left}{right - left} * 2 - 1$

$y - \frac{\frac{neary}{-z} - bottom}{top - bottom} * 2 - 1$





$V_{cx} = left$
$V_{cy} = bottom$
$V_x = right - left$
$V_y = top - bottom$



$W_{cx} = -1$
$W_{cy} = -1$
$W_x = 2$
$W_y = 2$

$$x'' = \frac{x' - left}{right - left} \cdot 2 - 1 \qquad x' = near\frac{x}{-z}$$

$$y'' = \frac{y' - bottom}{top - bottom} \cdot 2 - 1 \qquad y' = near\frac{y}{-z}$$

А тперь преобразование

$$x - \frac{\frac{nearx+zleft}{-z}}{right-left} * 2 - 1$$

$$x - \frac{\frac{nearx+zlef}{right-left}}{-z} * 2 - 1$$

$$x - \frac{\frac{nearx+zlef}{right-left}}{-z} * 2 - 1$$

$$x'' = \frac{near\frac{x}{-z} - left}{right - left} \cdot 2 - 1$$

$$x'' = \frac{2 \cdot near \cdot x + 2 \cdot z \cdot left + z \cdot right - z \cdot left}{-z(right - left)}$$

$$x'' = \frac{x\frac{2 \cdot near}{right - left} + z\frac{right + left}{right - left}}{-z}$$

14.3 Матрица перспективного преобразования

$$x'' = \frac{x\frac{2 \cdot near}{right - left} + z\frac{right + left}{right - left}}{-z}$$

$$y'' = \frac{y\frac{2 \cdot near}{top - bottom} + z\frac{top + bottom}{top - bottom}}{-z}$$

$$\begin{bmatrix} \chi'' \\ \gamma'' \\ \zeta'' \\ \alpha'' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$\alpha - -z$

$$x'' = \frac{x\frac{2 \cdot near}{right - left} + z\frac{right + left}{right - left}}{-z} \qquad \alpha'' = -z$$

$$y'' = \frac{y\frac{2 \cdot near}{top - bottom} + z\frac{top + bottom}{top - bottom}}{-z}$$

$$\begin{bmatrix} \chi'' \\ \gamma'' \\ \zeta'' \\ \alpha'' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\chi'' = x\frac{2 \cdot near}{right - left} + z\frac{right + left}{right - left} \qquad \alpha'' = -z$$

$$\gamma'' = y\frac{2 \cdot near}{top - bottom} + z\frac{top + bottom}{top - bottom}$$

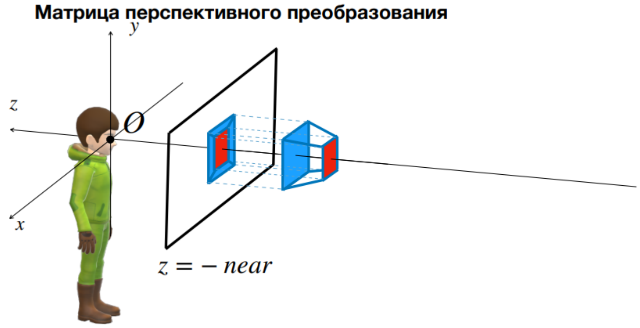
$$\begin{bmatrix} \chi'' \\ \gamma'' \\ \zeta'' \\ \alpha'' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\chi'' = x\frac{2 \cdot near}{right - left} + z\frac{right + left}{right - left} \qquad \alpha'' = -z$$

$$\gamma'' = y\frac{2 \cdot near}{top - bottom} + z\frac{top + bottom}{top - bottom}$$

$$\begin{bmatrix} \chi'' \\ \gamma'' \\ \zeta'' \\ \alpha'' \end{bmatrix} = \begin{bmatrix} \frac{2 \cdot near}{right - left} & 0 & \frac{right + left}{right - left} & 0 \\ 0 & \frac{2 \cdot near}{top - bottom} & \frac{top + bottom}{top - bottom} & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Матрица перспективного преобразования



$$\chi'' = x\frac{2 \cdot near}{right - left} + z\frac{right + left}{right - left} \qquad \alpha'' = -z$$

$$\gamma'' = y\frac{2 \cdot near}{top - bottom} + z\frac{top + bottom}{top - bottom}$$

$$\begin{bmatrix} \chi'' \\ \gamma'' \\ \zeta'' \\ \alpha'' \end{bmatrix} = \begin{bmatrix} \frac{2 \cdot near}{right - left} & 0 & \frac{right + left}{right - left} & 0 \\ 0 & \frac{2 \cdot near}{top - bottom} & \frac{top + bottom}{top - bottom} & 0 \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \chi'' \\ \gamma'' \\ \zeta'' \\ \alpha'' \end{bmatrix} = \begin{bmatrix} \frac{2 \cdot near}{right - left} & 0 & \frac{right + left}{right - left} & 0 \\ 0 & \frac{2 \cdot near}{top - bottom} & \frac{top + bottom}{top - bottom} & 0 \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\zeta'' = a_{33}z + a_{34} \qquad z'' = -a_{33} - \frac{a_{34}}{z}$$
$$\alpha'' = -z$$

$$\zeta'' = a_{33}z + a_{34} \qquad z'' = -a_{33} - \frac{a_{34}}{z}$$
$$\alpha'' = -z$$

$$\begin{aligned} z &= -near \\ z &= -far \end{aligned} \quad \Longmapsto \quad \begin{aligned} z'' &= -1 \\ z'' &= 1 \end{aligned}$$

$$-a_{33} - \frac{a_{34}}{-near} = -1$$
$$-a_{33} - \frac{a_{34}}{-far} = 1$$

$$\frac{a_{34}}{-far} - \frac{a_{34}}{-near} = -2$$

Матрица перспективного преобразования

$$-a_{33} - \frac{a_{34}}{-near} = -1 \qquad -a_{33} - \frac{\frac{-2 \cdot far \cdot near}{far - near}}{-far} = 1$$

$$-a_{33} - \frac{a_{34}}{-far} = 1 \qquad -a_{33} - \frac{2 \cdot near}{far - near} = 1$$

$$\frac{a_{34}}{-near} - \frac{a_{34}}{-far} = 2 \qquad a_{33} = -\frac{2 \cdot near}{far - near} - 1$$

$$a_{34}\frac{far - near}{-far \cdot near} = 2 \qquad a_{33} = -\frac{2 \cdot near + far - near}{far - near}$$

$$a_{34} = \frac{-2 \cdot far \cdot near}{far - near} \qquad a_{33} = -\frac{far + near}{far - near}$$

$$\begin{bmatrix} \chi'' \\ \gamma'' \\ \zeta'' \\ \alpha'' \end{bmatrix} = \begin{bmatrix} \frac{2 \cdot near}{right - left} & 0 & \frac{right + left}{right - left} & 0 \\ 0 & \frac{2 \cdot near}{top - bottom} & \frac{top + bottom}{top - bottom} & 0 \\ 0 & 0 & -\frac{far + near}{far - near} & \frac{-2 \cdot far \cdot near}{far - near} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$Frustum(left, right, bottom, top, near, far)$$

Эта матрица зависит от этих параметров

тут можно еще можно дополнить (произошел спидран лекции), вот час побудет подробнее

14.04.25

Модельная система коордниат(МСК) $\Longrightarrow^{modelveiew}$ Мировая система координат $\Longrightarrow^{LookAt}$ СКН $\Longrightarrow^{FRUSTUM}$ СКПО

far y field f view Oy

aspect aspect ratio

near

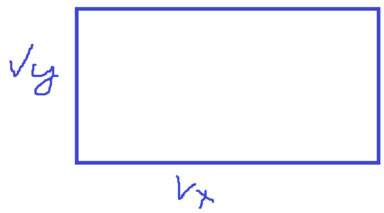
far

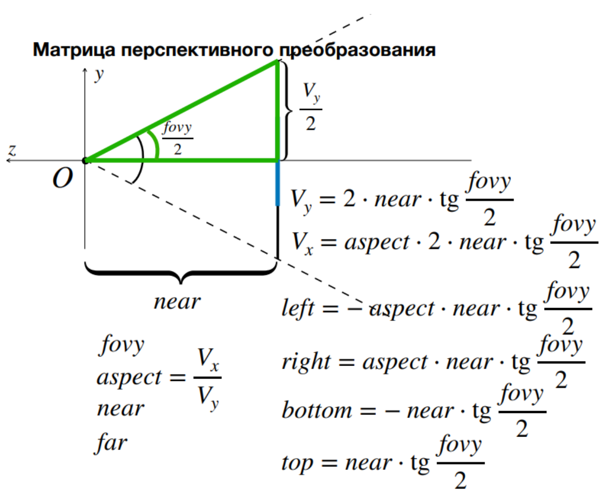
Частный случай:

left right

bottop top

near far



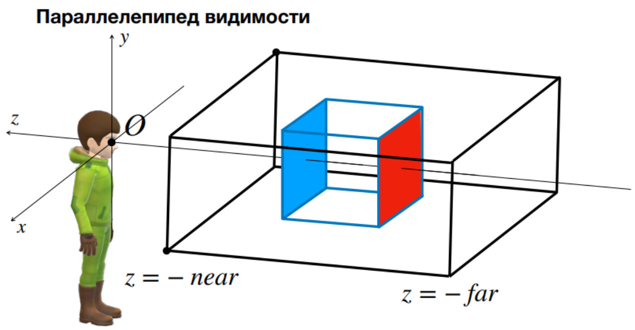


Матрица частного слуя преобразоавания

$$\begin{bmatrix} \chi \\ \gamma \\ \zeta \\ \alpha \end{bmatrix} = \begin{bmatrix} \frac{1}{aspect} ctg\frac{fovy}{2} & 0 & 0 & 0 \\ 0 & ctg\frac{fovy}{2} & 0 & 0 \\ 0 & 0 & -\frac{far+near}{far-near} & \frac{-2far*near}{far-ner} \\ 0 & 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Perspective(fovy,aspect,near,far)

Прямоугольная проекция



$x' = \frac{x-Vcx}{Vx}Wx + Wcx$ Vx = right-left

$y' = \frac{y-Vcy}{Vy}Wy + Wcy$ Vy = top-bottom

$z' = Wcx - \frac{z-Vcz}{Vz}Wz$ Vz = far-near

(-1,-1,-1) Wcx=Wcy+Wcz

$x' = \frac{x-left}{righht-left}2 - 1 = \frac{2x}{right-left} + \frac{-2left-right+left}{right-left}$

$y' = \frac{y-Vcy}{top-bottom}2 - 1y = \frac{2y}{top-botton} + \frac{-2bottom-top+bottom}{top-bottom}$

$z' = -1 - \frac{z+near}{far-near}2 = \frac{-}{2z}far - near + \frac{-2near-far+near}{far-near}$

Матрица преобразование Ortho. В основном используется в OpenGL

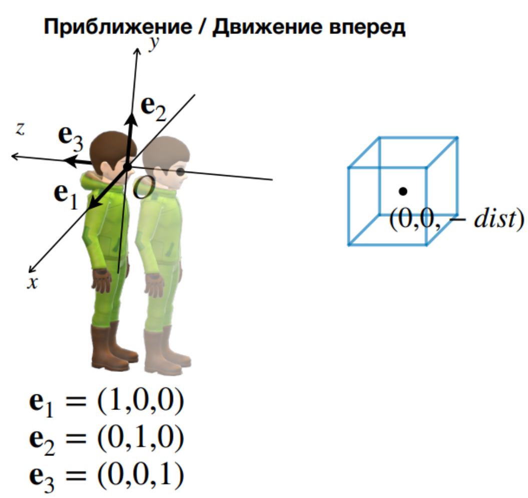
$$\begin{bmatrix} \chi' \\ \gamma' \\ \zeta' \\ \alpha' \end{bmatrix} = \begin{bmatrix} \frac{2}{right-left} & 0 & 0 & -\frac{right+left}{right-left} \\ 0 & \frac{2}{top-bottom} & 0 & -\frac{top+bottom}{top-bottom} \\ 0 & 0 & \frac{-2}{far-near} & -\frac{far+near}{far-near} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$x' = x\frac{2}{right-left} - \frac{right+left}{right-left}$$

$$y' = y\frac{2}{top-bottom} - \frac{top+bottom}{top-bottom}$$

$$z' = z\frac{-2}{far-near} - \frac{far+near}{far-near}$$

15 Организация движения в трехмерном пространтсве



Чтобы применить преобразование LookAt нужно:

1. В которую переходит глаз наблюдателя
2. Точка в которую смотрит наблюдатель
3. Вектор направление вверх

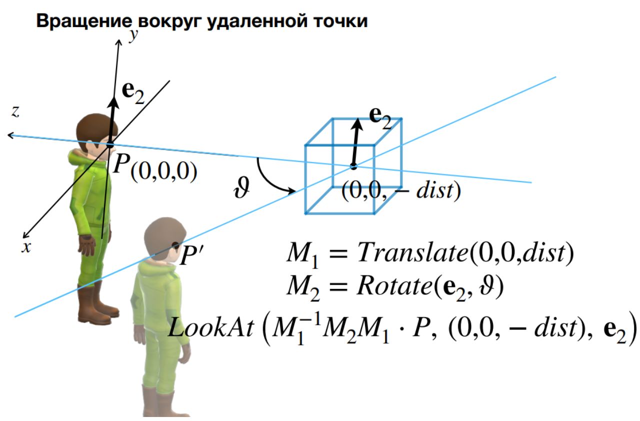
LookAt(S,P,$\bar{u}$)
LookAt((0,0,-1),(0,0,-2),(0,1,0))
LookAt((0,0,1),(0,0,0),(0,1,0))
и т.д

Подобное происходит и напрвалением вверх/вниз
LookAt((0,1,0),(0,1,-1),(0,1,0))
и т.д

21.04.25

Для двежиения достаточно умнажать обшую матрицу на матрицу Tr, P и т.д $(0(T_r * Rot * T_r * L * M) * P)$ это преобразование LookAt.

Совмещенное преобразование

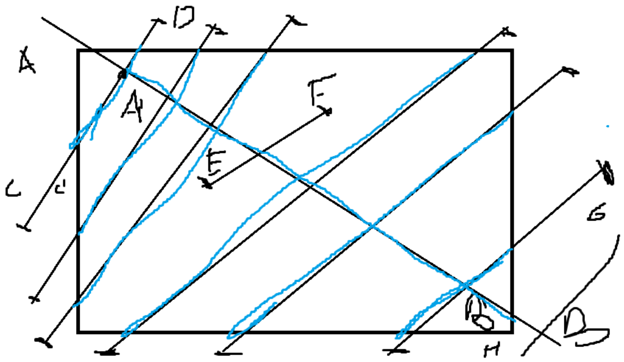


$$\begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & T \end{bmatrix}$$

$P = M^{-1}MP$

16 Алгоритм Коэна-Сазерленда

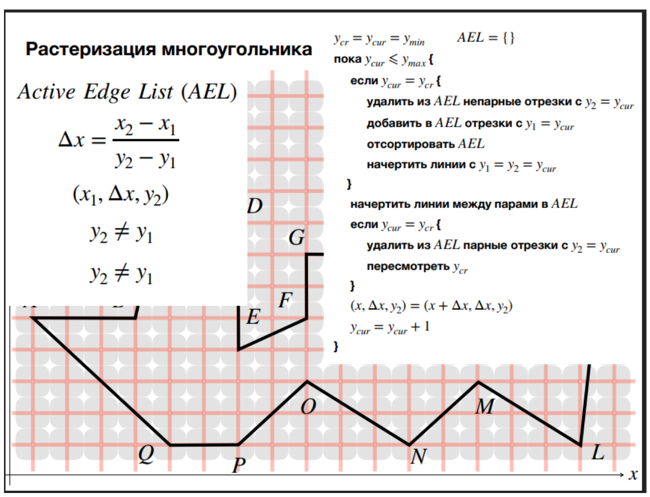
Алгоритм обязательно знать, без него не сдать экзамен!

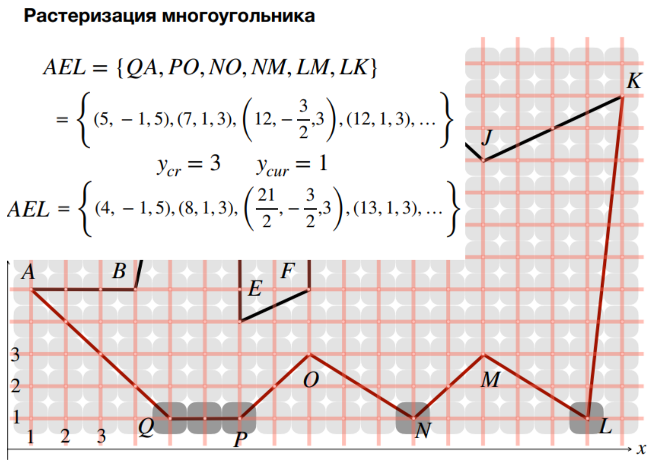


Пропус

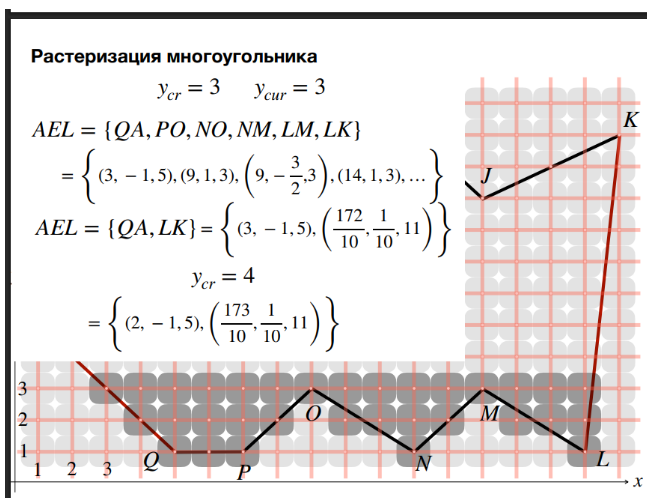
05.05.25

17 Растеризация





Для каждого из остальных ребер формируем тройку, где первый элемент равен X_1, второй (это не точно, не расслышала про второй элемент) Z_1, третий Y_1.



$x_1 0$
(r_1, g_1, b_1)
$x_2 - x_1 + 1$

x_2
(r_2, g_2, b_2)
По сути это цвтеовой шаг. $x_2 - x_1 + 1$ - отрезок

Растеризация многоугольника

$AB \quad (x_1, y_1, z_1) \; I_1 \qquad\qquad (x_2, y_2, z_2) \; I_2$

$I_1 = (r_1, g_1, b_1) \qquad\qquad I_2 = (r_2, g_2, b_2)$

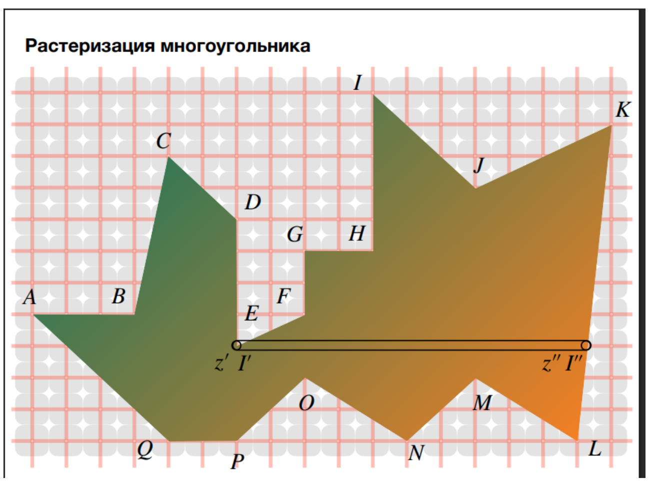
$$(x, \Delta x, y_2) \quad x = x_1 \quad \Delta x = \frac{x_2 - x_1}{y_2 - y_1}$$

$$\Delta z = \frac{z_2 - z_1}{y_2 - y_1}$$

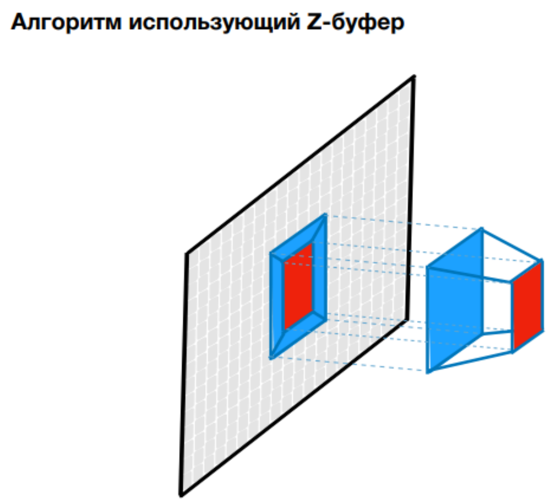
$$\Delta I = (\Delta r, \Delta g, \Delta b) = \left(\frac{r_2 - r_1}{y_2 - y_1}, \frac{g_2 - g_1}{y_2 - y_1}, \frac{b_2 - b_1}{y_2 - y_1}\right)$$

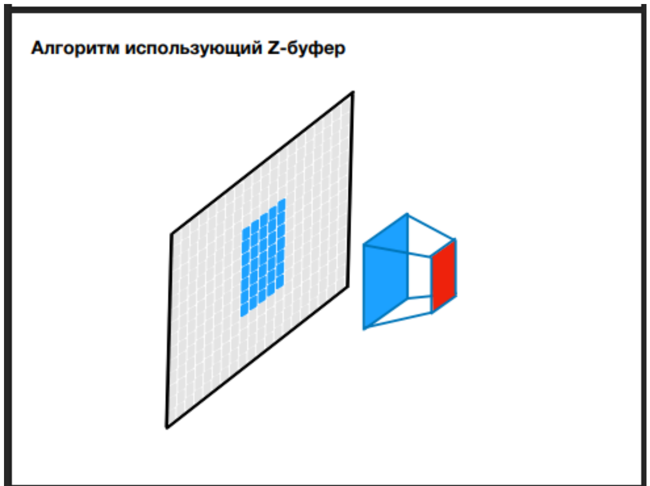
$$(x, \Delta x, y_2, z, \Delta z, I, \Delta I) \quad z = z_1 \quad I = I_1$$

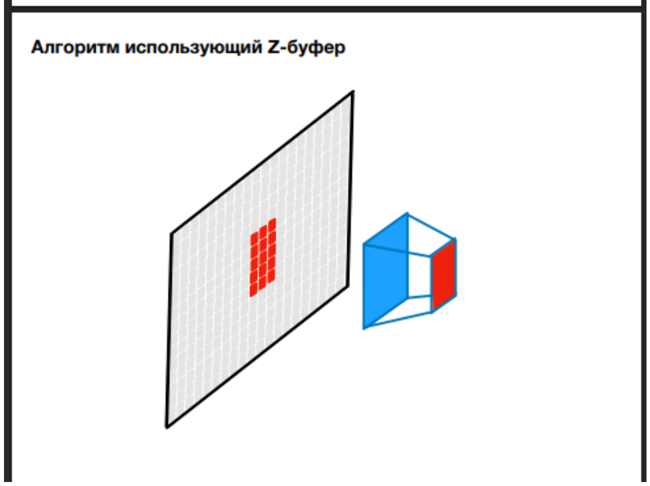
Если мы предполагем, что у нас есть z, то она будет, также изменяться как и x и y.



17.1 Алгоритм использующий Z-буфер







Z-буфер это двумерный массив, в котором столько элементов сколько столбцов и сколько строк растра.

18 Графический конвейр в OpenGL

Построение трехмерного прообраза

modelView - Модельное преобзравние (переход в мировую систему координат).

cameraView - Переход в систему координат наблюдателя.

OpenGL - это НЕ библиотека, это прежде всего спецификация, которая должна удевлитворяеть графическая карта, которую мы покупаем.

Спецификация - (техническое задание) предполагает

Шейдер - программа, которая выполняет определенную функцию на графической карте.

Вершинный шейдер - используются для веришины. Получает данные на входе, а на выходе получить координты вершины в системе координат отсечения. Выполняется на процессере сохраняется на граф карте. Отвечает за шаг

GLSE - язык встроенный в OpenGL, на нем пишутся шейдеры.

12.05.25

Первый шаг графического конвеера получение вершин.

Выходные пармаетры - out

Глобальная переменная для шейдеронрой программы это OpenGLPosition

Алгоритм z буфера работает сам

Шейдерная программа это программа для работы с шейдарами. Она линкуется(соеденется)

18.1 Вектор нормали

19.05.25

19 Освещение

Модели освещения можно разедлить на две части. Обычно в модели 256 цветов.

Источник света - точеченый

Быввает следущие модели: Моедль RGB - Моедль CMYK -