

Компьютерная графика

Harpie

February 2025

17.02.25

1 Субъективно-ориентированное программирование

Это парадигма программирования которая представляет отрезок времени 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 соединены отрезками. Перед описанием объекта нужно знать и задать координаты точек.

3 Модели

Модель строится набором отрезков, которые определенным образом соединены 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 декартова СК (мировая система координат) - "виртуальный мир в котором существуют модели.

4 Кадрирование

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

Кадр - прямоугольник, стороны которого параллельны осям координат и у которого есть параметры

Размер по горизонтали - V_x

Размер по вертикали - V_y

Координаты нижнего угла - V_{cx} , V_{cy}

Исходный кадр обозначается как - W_x , W_y , W_{cx} , W_{cy}

Безразмерные координаты :

$$\begin{aligned}x_1 &= x - V_{cx} & y_1 &= y - V_{cy} \\x_2 &= \frac{x - V_{cx}}{V_x} & y_2 &= \frac{y - V_{cy}}{V_y} \\x_3 &= \frac{x - V_{cx}}{V_x} * W_x & y_3 &= \frac{y - V_{cy}}{V_y} * W_y \\x' &= \frac{x - V_{cx}}{V_x} * W_x + W_{cx} & y' &= \frac{y - V_{cy}}{V_y} * W_y + W_{cy}\end{aligned}$$

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

$$\begin{aligned}x_4 &= \frac{x - V_{cx}}{V_x} * W_x + W_{cx} & y_4 &= \frac{y - V_{cy}}{V_y} * W_y - W_{cy} \\x_5 &= \frac{x - V_{cx}}{V_x} * W_x + W_{cx} & y_5 &= \frac{y - V_{cy}}{V_y} * W_x + 2W_{cy} \\x' &= \frac{x - V_{cx}}{V_x} * W_x + W_{cx} & y' &= W_{cy} - \frac{x - V_{cx}}{V_x} * W_y\end{aligned}$$

24.02.25

Продолжение

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

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

5 Преобразование изображения

5.1 Элементарные преобразования

1. Перенос/Сдвиг

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

Пример:

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

Итого: $x' = x + T_x$ $y' = y + T_y$

2. Масштабирование относительно начала координат

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

Пример:

$3 \implies 4.5$ и т.д

Итого: $x' = x * S_x$ $y' = y * S_y$

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

Нужно соединить лучом точку с началом координат. Преобразовать точку означает, повернуть этот отрезок на заданный угол ϑ . В результате получается новая точка.

$$x' = r \cos(\alpha + \vartheta)^x = r \cos \alpha \cos \vartheta - r \sin \alpha \sin \vartheta = x \cos \vartheta - y \sin \vartheta$$

$$y' = r \sin(\alpha + \vartheta)^x = r \sin \alpha \cos \vartheta + r \cos \alpha \sin \vartheta = y \cos \vartheta + x \sin \vartheta$$

Итого: $x' = x \cos \vartheta - y \sin \vartheta$ $y' = y \cos \vartheta + x \sin \vartheta$

4. Зеркальное отражение. Частный случай масштабирования

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

6 Совмещение преобразований

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

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

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

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

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

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

6.1 Однородные координаты

6.1.1 Евклидовые координаты

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

Формула для масштабирования

$$\chi' = \chi S_\chi$$

$$\gamma' = \gamma S_\gamma$$

$$\alpha' = \alpha$$

Матрица масштабирования

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

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 преобразование

P - столбец

M - матрица

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

M' - обратная матрица

$$P = M'(MP)$$

$$Rightarrow M' = M^{-1}$$

6.3 Преобразование перенос

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

$$T(T_x, T_y)$$

6.4 Преобразование масштабирование

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

$$S(S_x, S_y)$$

6.5 Преобразование поворот

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

7 Совмещенное преобразование

Смысл:

$$P' = (M_2 \dots M_3, M_2, M_1)P$$

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

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

8 Принцип двойственности

Двойственное преобразование

У нас есть матрица

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

Масштабирование равно преобразованию....

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

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

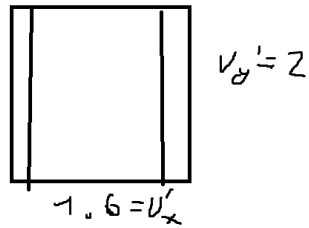
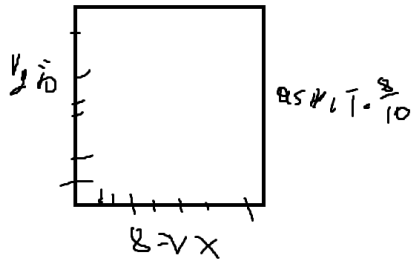
$$\frac{W_x}{V_x} < \frac{W_y}{V_y}$$

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

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

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

$$\begin{cases} \frac{W_x}{V_x} \text{ или } \frac{W_y}{V_y} \\ \frac{2}{V_x} \text{ или } \frac{2}{V_y} \\ \begin{cases} V'_x \\ V'_y \end{cases} \end{cases} \quad \text{Размеры модели после вписания в квадрат } 2 \times 2$$



Команды:

translate x y

rotate ϑ

scale S

figure

Вспомогательные команды

popTransform - забираем из стека

pushTransform - добавляем в стек

scale 1.25

pushTransform

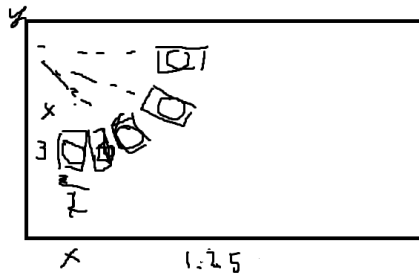
translate 0 -R

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
{
  pushTransform
  rotate22.5
  pushTransform
  translatexy
  figure
}
popTransform
10.03.25

```

9 Операции над векторами

9.1 Двумерные вектора

У веторов две хакартиристики:

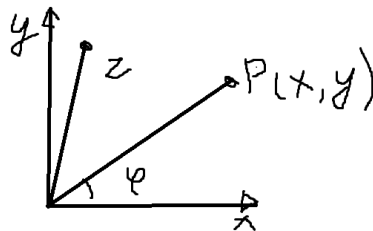
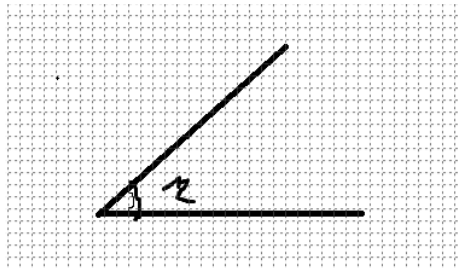
длина

+

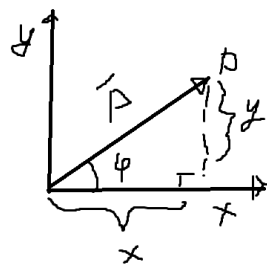
направление

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

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



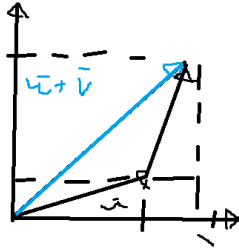
Радиус вектор точки P
 $\vec{P}(x, \varphi)$
 $x = r \cos \varphi$
 $y = r \sin \varphi$



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

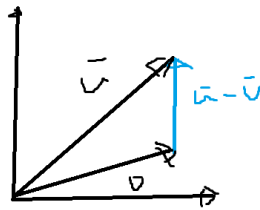
$$\vec{v} = (v_x, v_y)$$

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



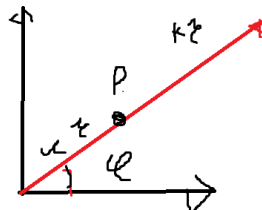
Разность векторов $\vec{u} - \vec{v}$

$$\vec{u} - \vec{v} = (u_x - v_x, u_y - v_y)$$



пууп

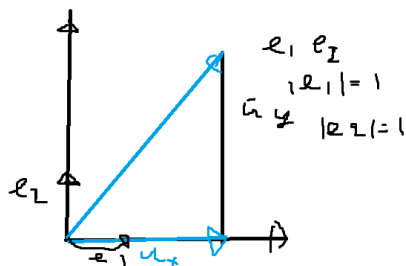
$$\vec{u} \quad k > 0$$



$$k\vec{u} = (ku_x, ku_y)$$

Скалярное произведение векторов

$$\begin{aligned} \vec{u} \vec{v} \\ \vec{u} * \vec{v} &= |\vec{u}| |\vec{v}| \cos \widehat{\vec{u} \vec{v}} \\ \vec{u} &= (r, \varphi) \\ |\vec{u}| &= r \\ \vec{u} &= u_x \vec{e}_1 + u_y \vec{e}_2 \\ \vec{u} * \vec{v} &= (u_x \vec{e}_1 + u_y \vec{e}_2) * (v_x \vec{e}_1 + v_y \vec{e}_2) = (|u_x| e_1 + |u_y| e_2) * (|v_x| e_1 + |v_y| e_2) \\ \vec{u} * \vec{v} &= |\vec{u}_x| |\vec{v}_x| \vec{e}_1 + |\vec{u}_y| |\vec{v}_y| \vec{e}_2 - \text{основная формула} \end{aligned}$$



$$\begin{aligned} \vec{u} \vec{i} \\ \vec{u} * \vec{i} &= |\vec{u}| |\vec{i}| \cos \widehat{\vec{u} \vec{i}} = |\vec{u}| \cos \widehat{\vec{u} \vec{i}} \end{aligned}$$



dot-product - Скалярное произведение

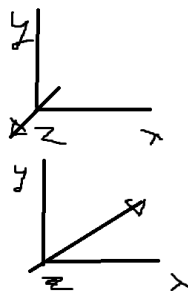
Псевдоскалярное произведение

$$\begin{aligned} \vec{u} \vec{v} \\ \vec{u} x \vec{v} &= |\vec{u}| |\vec{v}| \sin \widehat{\vec{u} \vec{v}} \\ \vec{u} x \vec{v} &= -\vec{u} x \vec{v} \end{aligned}$$

$$e_1 x e_1 = 0, e_1 x e_2 = 1, e_2 x e_1 = -1, e_2 x e_2 = 0 = u_x v_y - u_y v_x > \vec{u} x \vec{v} = \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix}$$

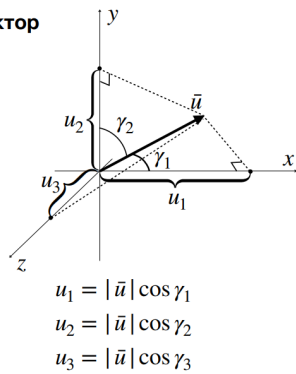
cross-product - Псевдоскалярное произведение

9.2 Трёхмерные вектора

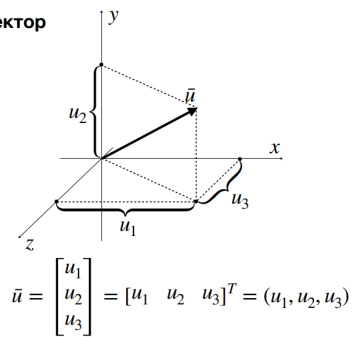


Левосторонняя и правосторонняя

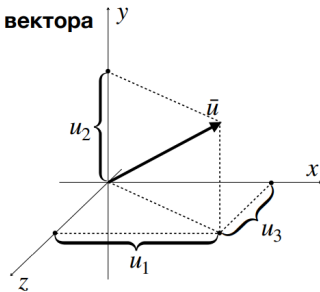
3D-вектор



3D-вектор



Длина вектора

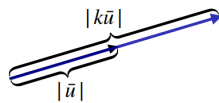


$$|\vec{u}| = \sqrt{u_1^2 + u_2^2 + u_3^2}$$

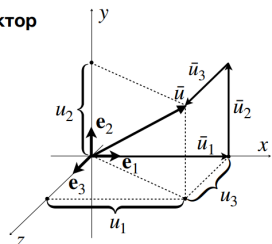
Умножение вектора на скаляр

$$k\vec{u} = \begin{bmatrix} ku_1 \\ ku_2 \\ ku_3 \end{bmatrix} = [ku_1 \quad ku_2 \quad ku_3]^T = (ku_1, ku_2, ku_3)$$

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



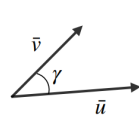
3D-вектор



$$\begin{aligned} \vec{u}_1 &= u_1 \mathbf{e}_1 & \vec{u} &= \vec{u}_1 + \vec{u}_2 + \vec{u}_3 \\ \vec{u}_2 &= u_2 \mathbf{e}_2 & \vec{u} &= u_1 \mathbf{e}_1 + u_2 \mathbf{e}_2 + u_3 \mathbf{e}_3 \\ \vec{u}_3 &= u_3 \mathbf{e}_3 \end{aligned}$$

Сложение вектора по координатам

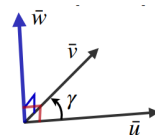
!Скалярное произведение аналогично двухмерному



$$\begin{aligned}\vec{u} \cdot \vec{v} &= |\vec{u}| |\vec{v}| \cos \gamma \\ \vec{u} \cdot \vec{v} &= u_1 v_1 + u_2 v_2 + u_3 v_3 \\ \vec{u} \cdot \vec{v} &= \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \vec{u}^T \vec{v} \\ \vec{u} \cdot \vec{v} &= \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \vec{v}^T \vec{u}\end{aligned}$$

Через матрицу

Векторное произведение

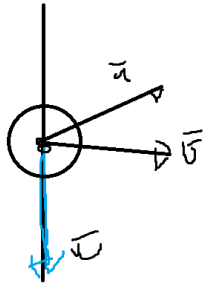


$$\begin{aligned}\vec{w} &= \vec{u} \times \vec{v} \\ |\vec{u} \times \vec{v}| &= |\vec{u}| |\vec{v}| \sin \angle \gamma \\ \vec{w} &\perp \vec{u} & \vec{w} \cdot \vec{u} &= 0 \\ \vec{w} &\perp \vec{v} & \vec{w} \cdot \vec{v} &= 0\end{aligned}$$

Угол может быть как положительным, так и отрицательным

Не важно в какую сторону отмеряться угол

Если положительный результат, то вверх, отрицательный же вниз



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

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

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

Матрица векторного произведения

17.03.25

10 Одномерные координаты

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

$$A_x + B_y + C = 0$$

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

$$(2, 3, 0)$$

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

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

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

$$P_1 x P_2 = e$$

$$e_1 x e_2 = P$$

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

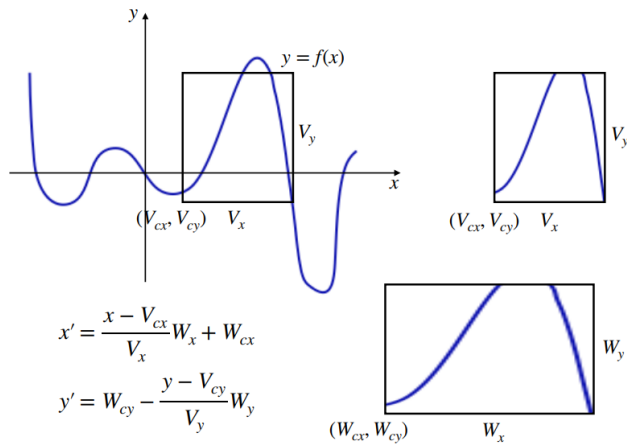
Плюкорова координата - $\beta = LP$

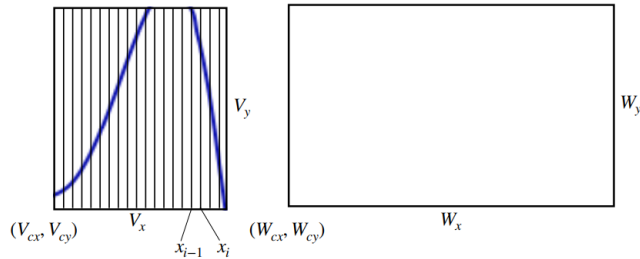
L - представление как матрицы 4x4

Все выше перечисленное не будет на экзамене, это чисто для общего развития

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

$x_0 = V_{cx} \rightarrow x'_0 = W_{cx}$ - кадрирование

$y_0 = f(x_0) \rightarrow y'_0$ - кадрирование

$x_i = x_{i-1} + \Delta \rightarrow x'_i$ - кадрирование

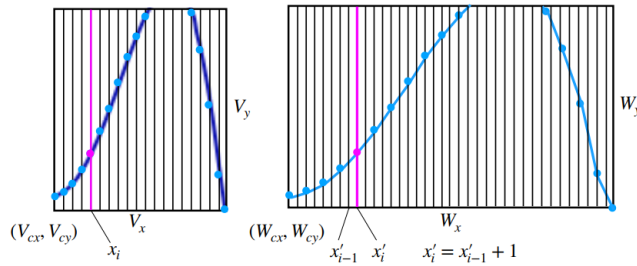
$y_i = f(x_i) \rightarrow y'_i$ - кадрирование

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 зависимости от этого и выбираться Δ_x . Экран растровое пространство. По сути это сетка - точка раstra.



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

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

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

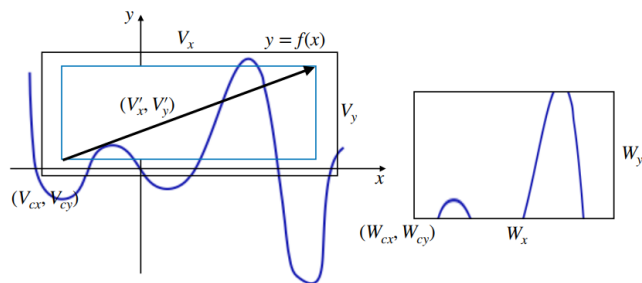
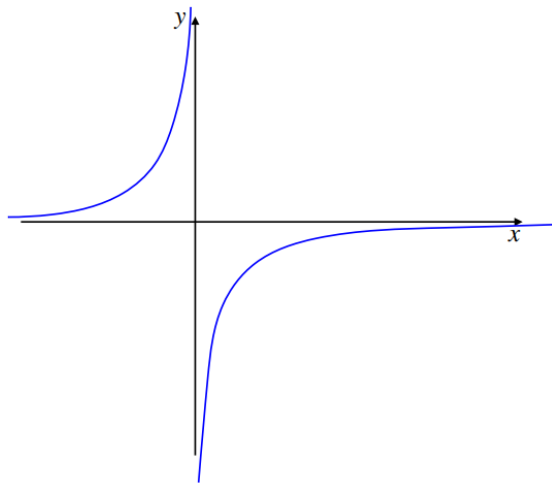
$$y' = W_{cy} - \frac{y - V_{cy}}{V_y} W_y$$

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

Цикл отрисовки:

пока $x'_i \leq W_{cx} + W_x$
 $x'_i = x'_{i-1} + 1 \rightarrow x_i$ кадр
 $y'_i \leftarrow y_i(f_i)$ 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.

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

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

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

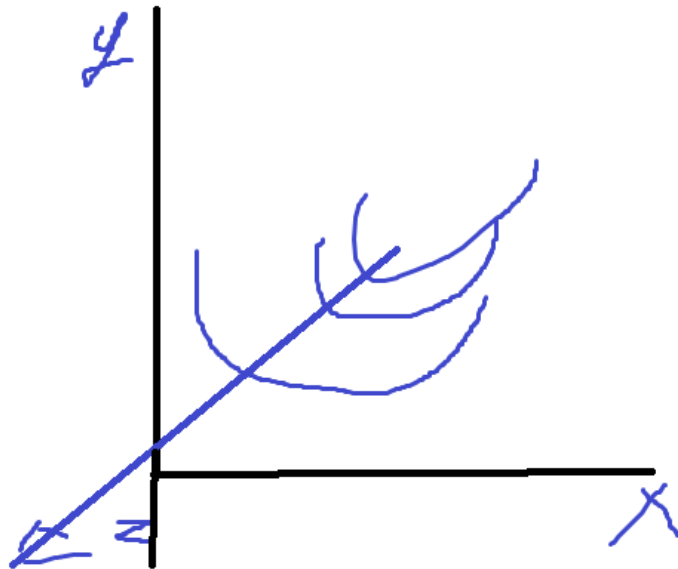
Так будет масштабироваться в окне

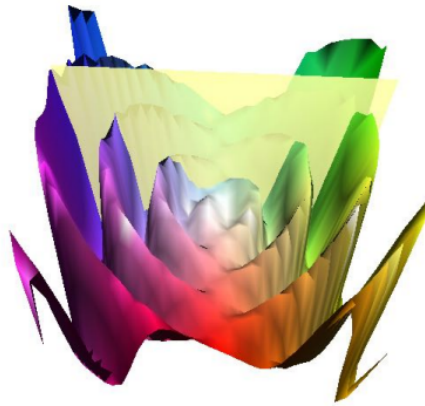
12 Построение 3D-графика

Трёхмерный график формируется из двумерных графиков

Пусть:

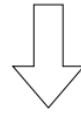
Будем обозначать $y = f(x, z)$. Теперь график зависит ещё и от z



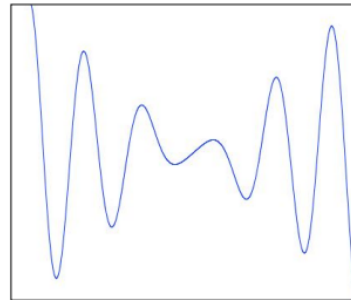


$$y = -x \sin \sqrt{x^2 + z^2}$$

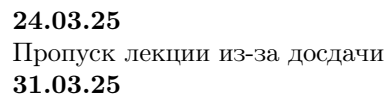
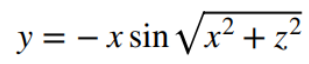
$$z = -5$$



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

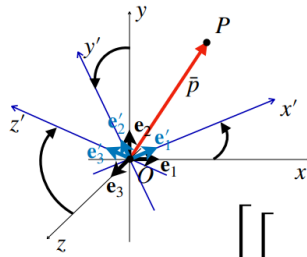


Для рисования такого используется "Алгоритм художника". Вначале начинаем с заднего плана и от него идет вплоть до самого ближнего плана



Матричная форма $\vec{p}' = \begin{bmatrix} e_1'^T \\ e_2'^T \\ e_3'^T \end{bmatrix} \vec{p}$

Двойственность вращения



$$\begin{aligned} x &= \mathbf{e}_1 \cdot \vec{p} \\ y &= \mathbf{e}_2 \cdot \vec{p} \\ z &= \mathbf{e}_3 \cdot \vec{p} \end{aligned}$$

$$\begin{aligned} x' &= \mathbf{e}_1' \cdot \vec{p} \\ y' &= \mathbf{e}_2' \cdot \vec{p} \\ z' &= \mathbf{e}_3' \cdot \vec{p} \end{aligned}$$

$$Rotate = \begin{bmatrix} \begin{bmatrix} \mathbf{e}_1' \\ \mathbf{e}_2' \\ \mathbf{e}_3' \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} & 1 \end{bmatrix}$$

13 Система координат наблюдателя

Замечание

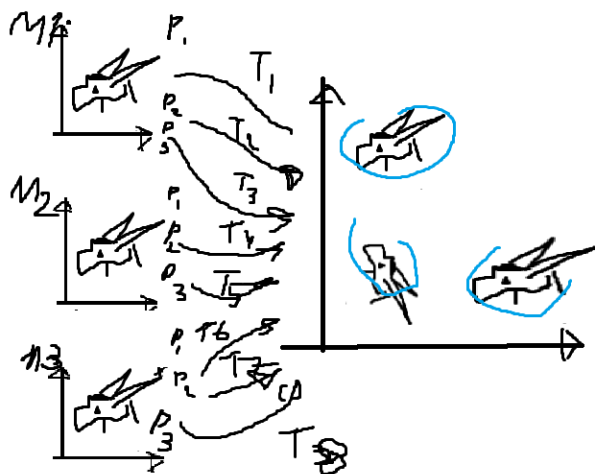
Модель это набор отрезков, на данный момент.

Вершины заданы в какой-то система координат.

Преобразование накапливается.

Для каждого набора точек, своя модель преобразования.

Есть неизменяемые вершинные данные, изменяются только преобразования.



Точка наблюдене - точка зрения наблюдателя(самега).

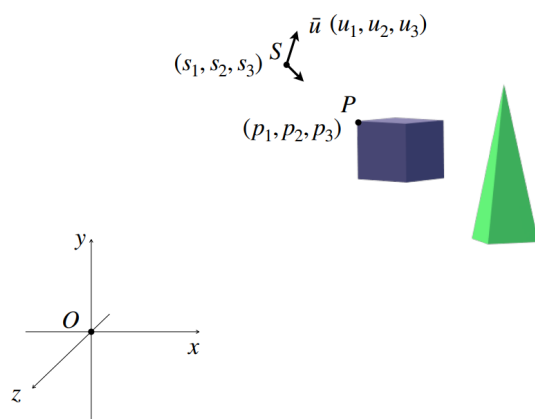
Вектор наблюдения - вектор направленный от взгляда наблюдателя в точку наблюдения.

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

Ось x направленна влево вниз от наблюдателя

Ось y направленна вверх от наблюдателя

Ось z направленна вправо от наблюдателя



Преобразование:

1. Поворот

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

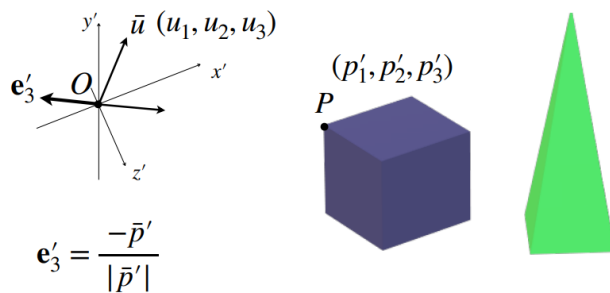
2. Вращение. Нужно определить матрицу вращения

Система координат наблюдателя

$$p'_1 = p_1 - s_1$$

$$p'_2 = p_2 - s_2$$

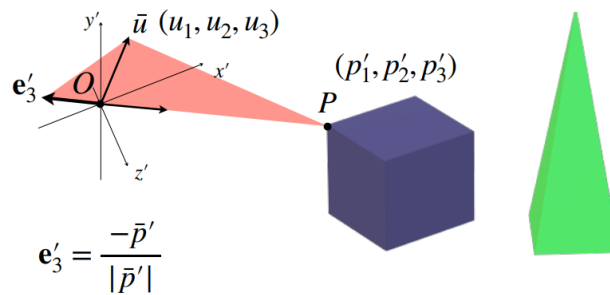
$$p'_3 = p_3 - s_3$$



Определить первый базисный вектор

Определить второй базисный вектор

Определить третий базисный вектор



После всех превращений нужно...

LookAt - получит матрицу вращений. Преобразование системы переходы из мировой системы координат в систему координат наблюдателя

Преобразование *LookAt*

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

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

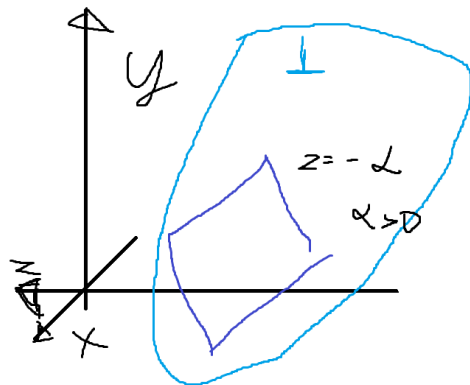
Вверху на картинке опечатка: лишняя единица и на главной диагонали не хватает единицы

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

Правильная матрица

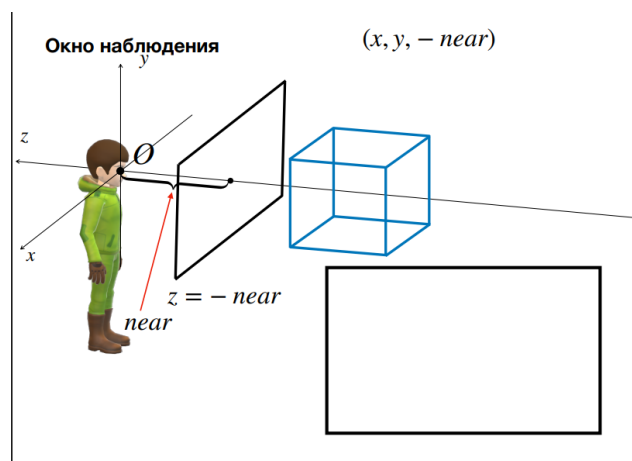
Плоскость наблюдения (круг).

Окно наблюдения(квадрат) кадр

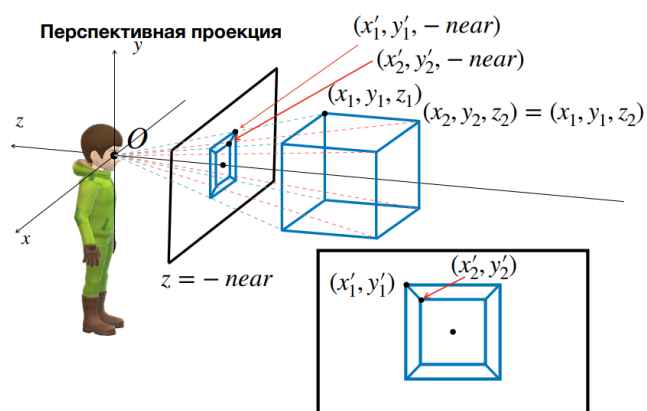


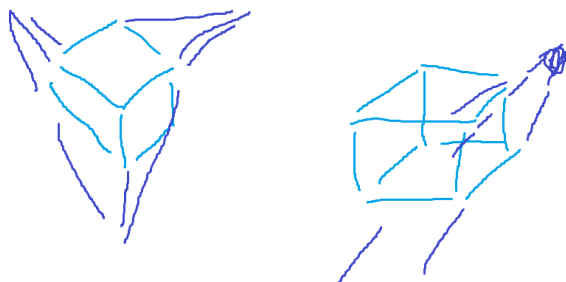
near - расстояние от наблюдателя до окна наблюдения

14 Трехмерные преобразования



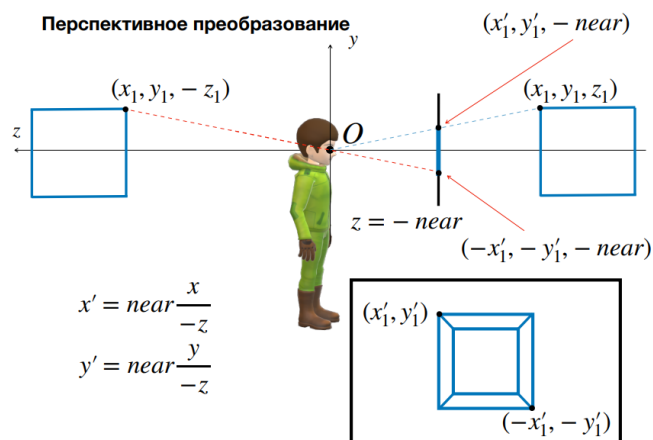
Перспектива - это намеренное искажения изображения с целью приданию глубины





14.1 Перспективное преобразование

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



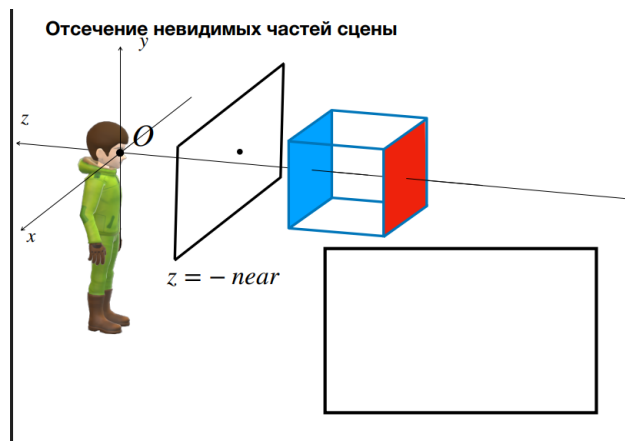
07.04.25

14.2 Отсечение невидимых частей сцены

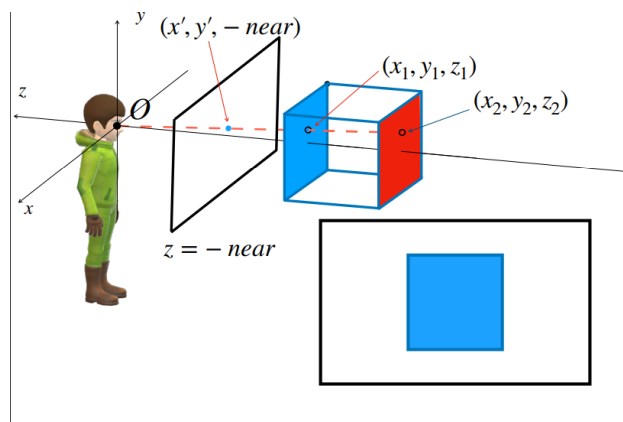
1) **Проволочное** изображение - это объект заданный отрезками, здесь можно перспективное преобразование применить, или прямоугольная проекция.

2) Когда изображение заданно гранями оно называется **контурным**
Можно описать грани как: последовательность отрезков, получается мы описываем их как набор многоугольников.

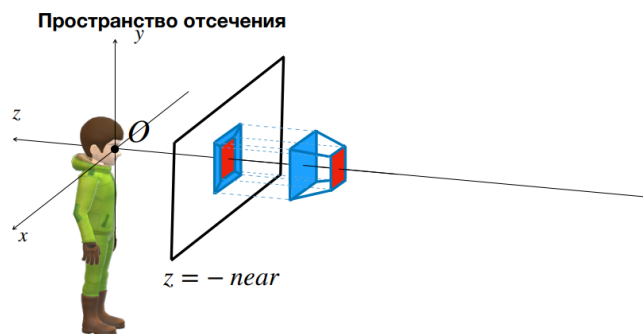
3) **Полутонное** изображение у каждой грани есть атрибуты.



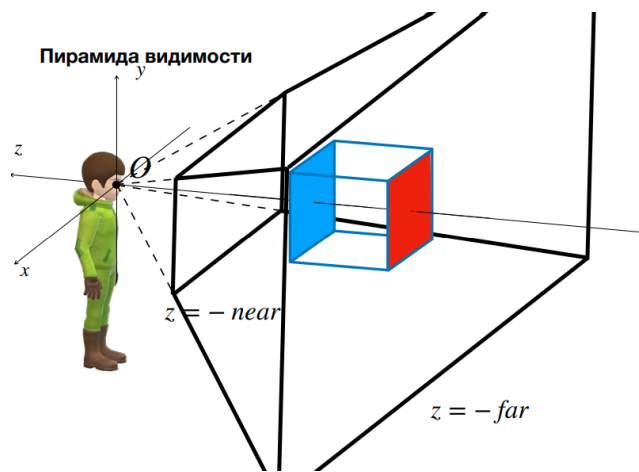
Отсечение



Пространство отсечения



Пирамида видимости



Пространство отсечения

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

Главный вопрос насколько мы далеко "отодвинули" "раздвинула"

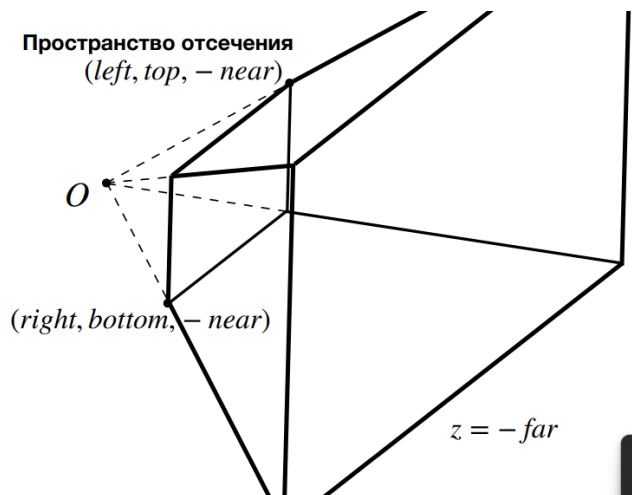
В этом преобразовании оси x и z меняются местами. Система координат левая.

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

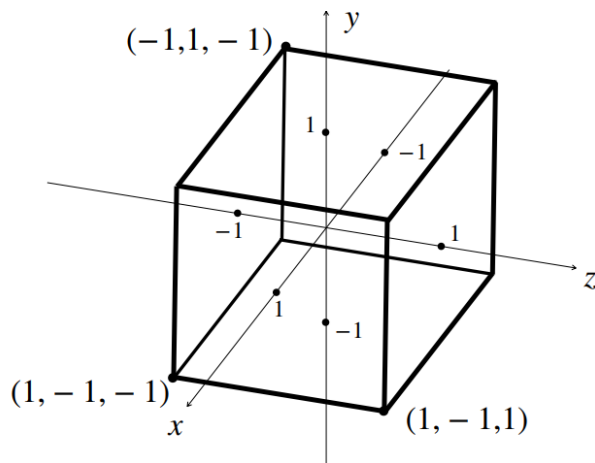
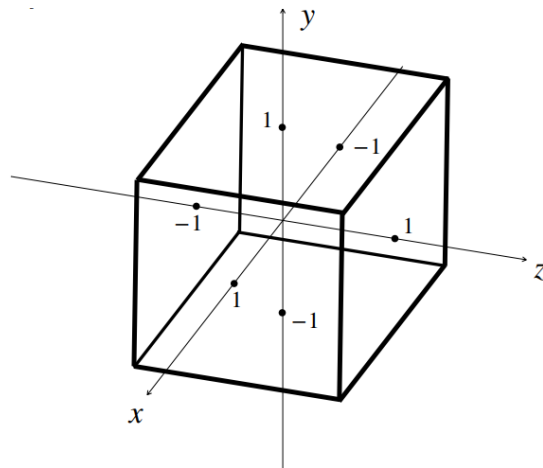
far - расстояние

$-far$ - координата

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



Задаем окно наблюдения - оно ограничено
 left, right - x - параметры наблюдения, координаты
 top, bottom - y - параметры наблюдения, координаты
 Вся плоскость наблюдения ограничена -near
 near, far - расстояния



Переход в пространство отсечения

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

В начале нужно провести операцию кодирования

Это переспективное преобразование

$$\begin{aligned} V_{cx} &= left & W_{cx} &= -1 \\ V_{cy} &= bottom & W_{yx} &= -1 \\ V_x &= right - left & W_y &= 2 \\ V_y &= top - bottom & W_y &= 2 \end{aligned}$$

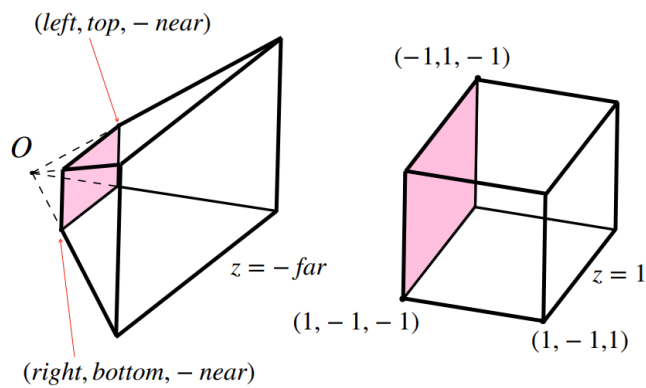
$x - \frac{x'_{right-left}}{z}$ это недописано

$y - \frac{y'_{right-left}}{z}$ это недописано

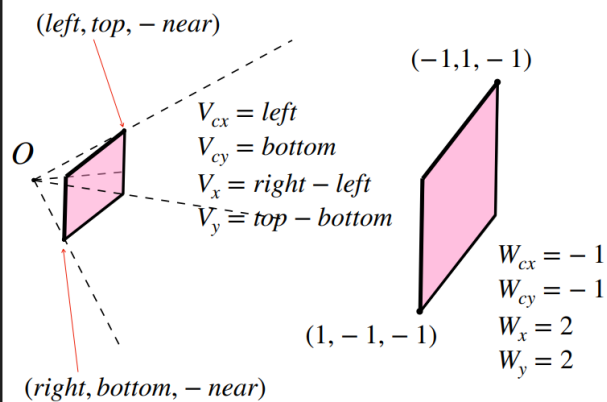
А это переспективное проекция

$$\begin{aligned} x &= \frac{\frac{nearx-left}{-z} * 2 - 1}{\frac{right-left}{-z} * 2 - 1} \\ y &= \frac{\frac{neary-bottom}{-z} * 2 - 1}{\frac{top-bottom}{-z} * 2 - 1} \end{aligned}$$

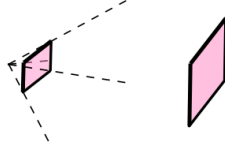
Переход в пространство отсечения



Переход в пространство отсечения



$$\begin{aligned}
V_{cx} &= left \\
V_{cy} &= bottom \\
V_x &= right - left \\
V_y &= top - bottom
\end{aligned}$$



$$\begin{aligned}
W_{cx} &= -1 & x'' &= \frac{x' - left}{right - left} \cdot 2 - 1 & x' &= near \frac{x}{-z} \\
W_{cy} &= -1 & y'' &= \frac{y' - bottom}{top - bottom} \cdot 2 - 1 & y' &= near \frac{y}{-z} \\
W_x &= 2 & & & & \\
W_y &= 2 & & & &
\end{aligned}$$

А теперь преобразование

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

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

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

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

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

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

14.3 Матрица перспективного преобразования

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

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

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

$$\alpha = -z$$

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

$$\alpha'' = -z$$

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

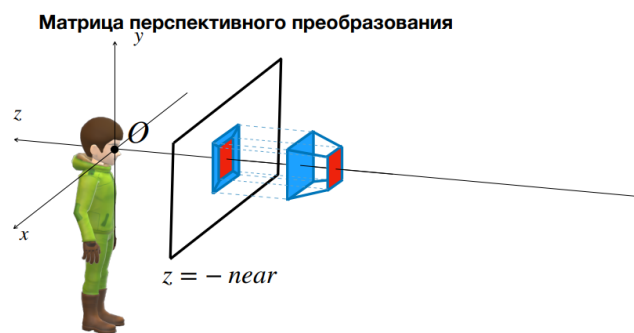
$$\begin{bmatrix} x'' \\ y'' \\ z'' \\ \alpha'' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

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

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

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

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



$$\begin{aligned}\chi'' &= x \frac{2 \cdot \text{near}}{\text{right} - \text{left}} + z \frac{\text{right} + \text{left}}{\text{right} - \text{left}} \\ \gamma'' &= y \frac{2 \cdot \text{near}}{\text{top} - \text{bottom}} + z \frac{\text{top} + \text{bottom}}{\text{top} - \text{bottom}} \\ \alpha'' &= -z\end{aligned}$$

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

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

$$\begin{aligned}\zeta'' &= a_{33}z + a_{34} \\ \alpha'' &= -z\end{aligned} \quad \zeta'' = -a_{33} - \frac{a_{34}}{z}$$

$$\begin{aligned}\zeta'' &= a_{33}z + a_{34} \\ \alpha'' &= -z\end{aligned} \quad \zeta'' = -a_{33} - \frac{a_{34}}{z}$$

$$\begin{array}{l} z = -\text{near} \\ z = -\text{far} \end{array} \quad \Rightarrow \quad \begin{array}{l} z'' = -1 \\ z'' = 1 \end{array}$$

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

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

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

Матрица перспективного преобразования

$$\begin{aligned}
 -a_{33} - \frac{a_{34}}{-near} &= -1 & -a_{33} - \frac{\frac{-2 \cdot far \cdot near}{far - near}}{-far} &= 1 \\
 -a_{33} - \frac{a_{34}}{-far} &= 1 & -a_{33} - \frac{2 \cdot near}{far - near} &= 1 \\
 \frac{a_{34}}{-near} - \frac{a_{34}}{-far} &= 2 & a_{33} &= -\frac{2 \cdot near}{far - near} - 1 \\
 a_{34} \frac{far - near}{-far \cdot near} &= 2 & a_{33} &= -\frac{2 \cdot near + far - near}{far - near} \\
 a_{34} &= \frac{-2 \cdot far \cdot near}{far - near} & a_{33} &= -\frac{far + near}{far - near}
 \end{aligned}$$

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

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

Эта матрица зависит от этих параметров

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

14.04.25

Модельная система координат(МСК) $\Rightarrow^{modelview}$ Мировая система координат \Rightarrow^{LookAt} СКН $\Rightarrow^{FRUSTUM}$ СКПО

far y field f view Oy

aspect aspect ratio

near

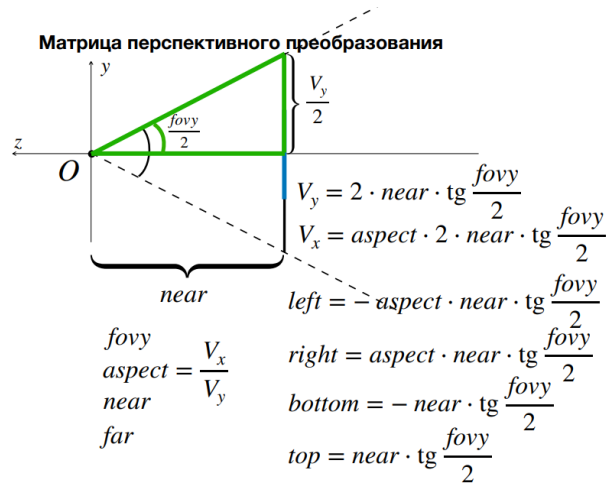
far

Частный случай:

left right

bottom top

near far

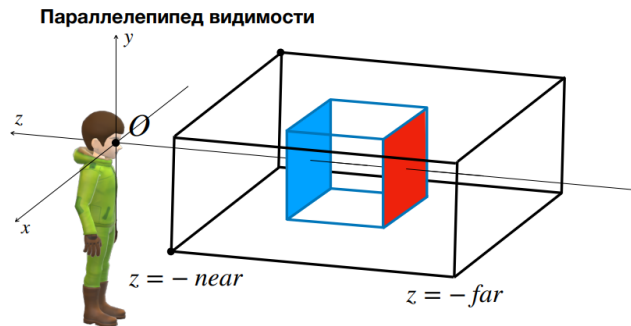


Матрица частного случая преобразования

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

Perspective(fovy,aspect,near,far)

Прямоугольная проекция



$$x' = \frac{x - V_{cx}}{V_x} W_x + W_{cx} \quad V_x = \text{right-left}$$

$$y' = \frac{y - V_{cy}}{V_y} W_y + W_{cy} \quad V_y = \text{top-bottom}$$

$$z' = W_{cz} - \frac{z - V_{cz}}{V_z} W_z \quad V_z = \text{far-near}$$

$$(-1, -1, -1) \quad W_{cx} = W_{cy} + W_{cz}$$

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

$$y' = \frac{y - V_{cy}}{\text{top-bottom}} 2 - 1 = \frac{2y}{\text{top-bottom}} + \frac{-2\text{bottom} - \text{top} + \text{bottom}}{\text{top-bottom}}$$

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

Матрица преобразование Ortho. В основном используется в OpenGL

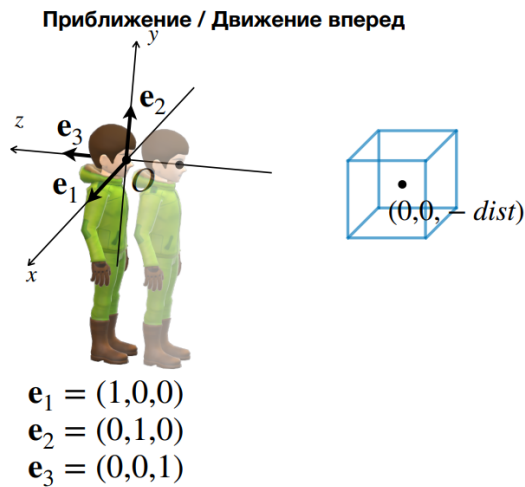
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{2}{right-left} & 0 & 0 & -\frac{right+left}{right-left} \\ 0 & \frac{2}{top-bottom} & 0 & -\frac{top+bottom}{top-bottom} \\ 0 & 0 & \frac{-2}{far-near} & -\frac{far+near}{far-near} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix}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

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

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

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

15 Организация движения в трехмерном пространстве



Чтобы применить преобразование LookAt нужно:

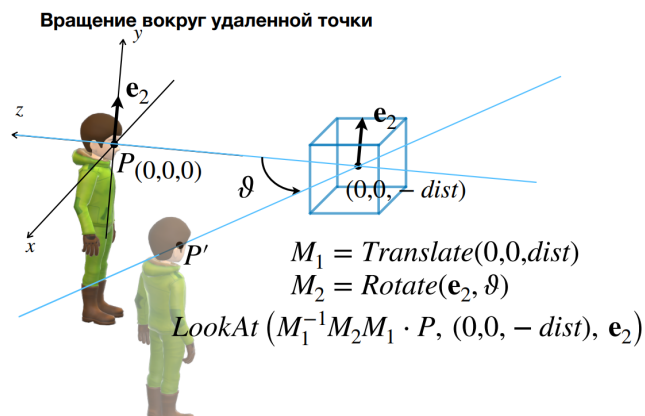
1. В которую переходит глаз наблюдателя
 2. Точка в которую смотрит наблюдатель
 3. Вектор направление вверх
- LookAt(S,P, \bar{u})
 LookAt((0,0,-1),(0,0,-2),(0,1,0))
 LookAt((0,0,1),(0,0,0),(0,1,0))
 и т.д

Подобное происходит и направлением вверх/вниз
 $\text{LookAt}((0,1,0),(0,1,-1),(0,1,0))$
 и т.д

21.04.25

Для движения достаточно умножать общую матрицу на матрицу T_r , P
 и т.д $(0(T_r * Rot * T_r * L * M) * P)$ это преобразование LookAt.

Совмещенное преобразование

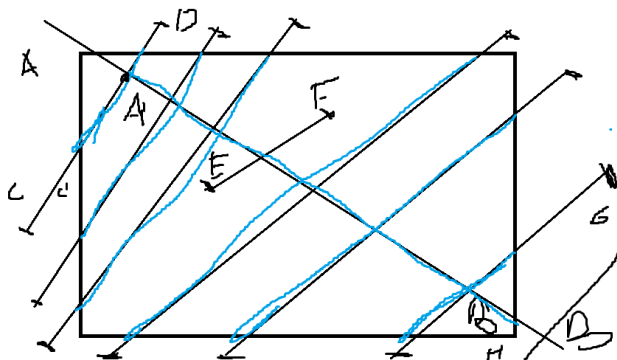


$$\begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & T \end{bmatrix}$$

$$P = M^{-1}MP$$

16 Алгоритм Коэна-Сазерленда

Алгоритм обязательно знать, без него не сдать экзамен!



Пропус

05.05.25

17 Растеризация

Растеризация многоугольника

Active Edge List (AEL)

$$\Delta x = \frac{x_2 - x_1}{y_2 - y_1}$$

$(x_1, \Delta x, y_2)$

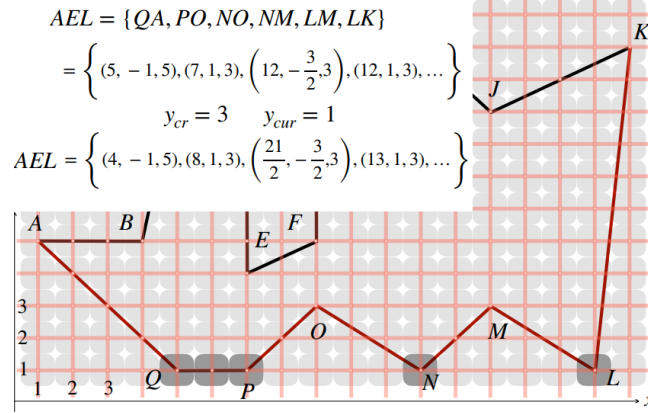
$y_2 \neq y_1$

$y_2 \neq y_1$

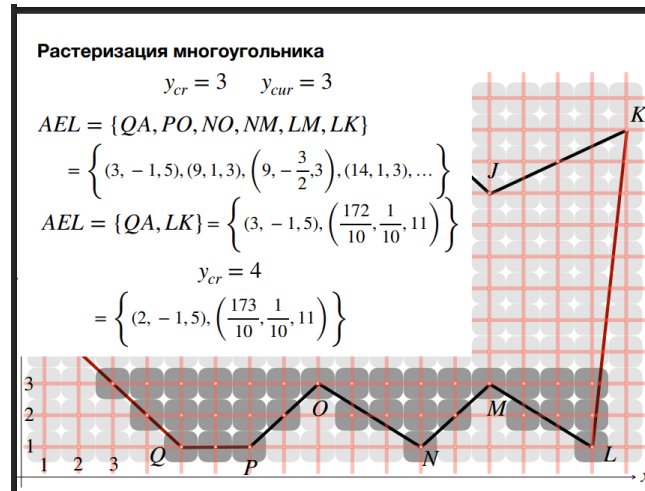
```

ycr = ycur = ymin    AEL = {}
пока ycur ≤ ymax {
    если ycur = ycr {
        удалить из AEL непарные отрезки с y2 = ycur
        добавить в AEL отрезки с y1 = ycur
        отсортировать AEL
        начертить линии с y1 = y2 = ycur
    }
    начертить линии между парами в AEL
    если ycur = ycr {
        удалить из AEL парные отрезки с y2 = ycur
        пересмотреть ycr
    }
    (x, Δx, y2) = (x + Δx, Δx, y2)
    ycur = ycur + 1
}
        
```


Растеризация многоугольника



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 цветовой шаг. $x_2 - x_1 + 1$ - отрезок

Растеризация многоугольника

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

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

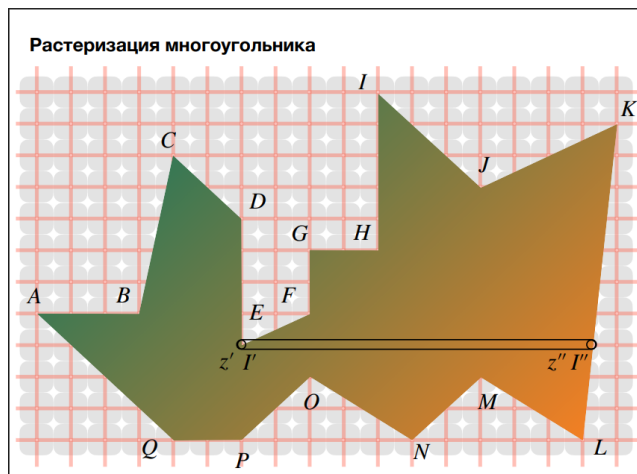
$$(x, \Delta x, y_2) \quad x = x_1 \quad \Delta x = \frac{x_2 - x_1}{y_2 - y_1}$$

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

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

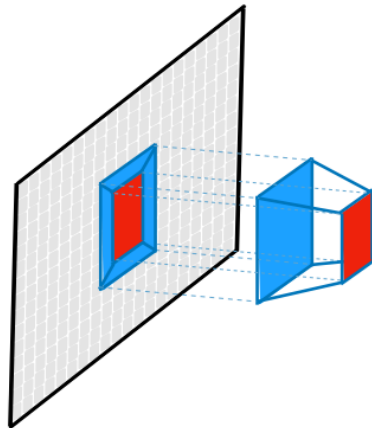
$$(x, \Delta x, y_2, z, \Delta z, I, \Delta I) \quad z = z_1 \quad I = I_1$$

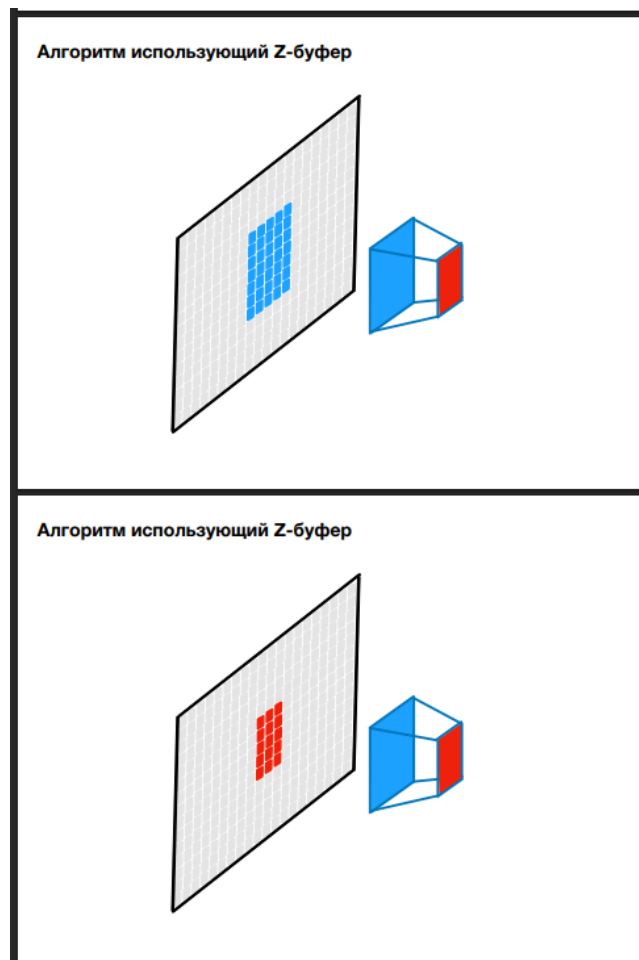
Если мы предполагаем, что у нас есть z, то она будет, также изменяться как и x и y.



17.1 Алгоритм использующий Z-буфер

Алгоритм использующий Z-буфер





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

18 Графический конвейр в OpenGL

Построение трехмерного прообраза

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

cameraView - Переход в систему координат наблюдателя.

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

Спецификация - (техническое задание) предполагает