Компьютерная графика

silvia.lesnaia

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 параллельны 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 :

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

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

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

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

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

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

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

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

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

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

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

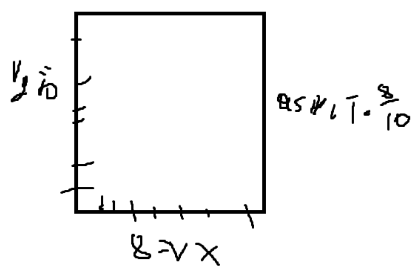
$\frac{2}{V_x} < \frac{2}{V_y}$

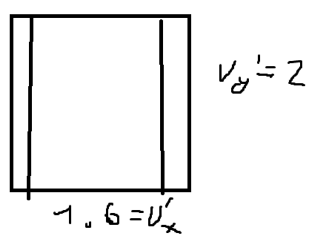
$1 = \frac{2}{2} < \frac{V_x}{V_y}$ - aspect - соотношение сторон

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 $\frac{2}{V_y}$

$$* \begin{cases} V_x^{'} & \text{Размеры модели после вписания в квадрат 2x2} \\ V_y^{'} & \end{cases} *$$

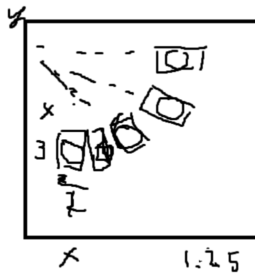




Команды:
translate x y
rotate ϑ
scale S
figure
Вспомогательные команды
popTransform - забираем из стка
pushTransform - добавляем в стек

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

$$* \begin{cases} pushTransform \\ rotate 22.5 \\ pushTransform \\ translate x y \\ figure \end{cases} *$$

popTransform