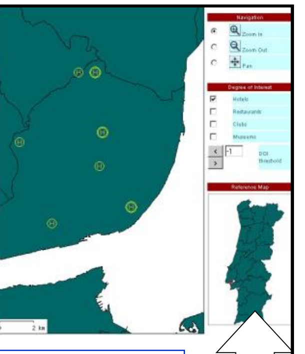
# Transformação de coordenadas

As aplicações de Computação Gráfica envolvem, em geral, **transformações de um sistema de coordenadas para outro** ao longo do *pipeline* de produção das imagens



# Transformação de coordenadas



## Exemplo 1

Mapa de Portugal construído tendo por unidade o km

Transformação de coordenadas

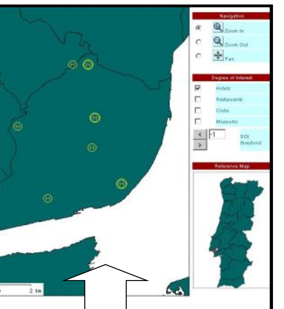
Desenhar o mapa de Portugal no canto inferior direito do ecrã

Sistema de coordenadas do utilizador (ou do mundo) *World coordinates*

Sistema de coordenadas da área de desenho - *Device coordinates*

# Transformação de coordenadas

Exemplo 2



Mapa de Portugal construído tendo por unidade o km

Escolher zona a visualizar

zona de Lisboa

Transformação de coordenadas

Desenhar o mapa de Lisboa na área à esquerda do ecrã

Sistema de coordenadas do utilizador

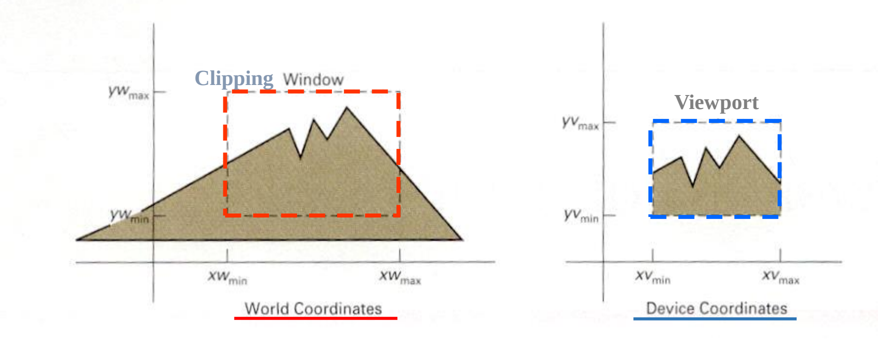
Sistema de coordenadas da área de desenho



# Transformação
## *Clipping window – viewport transformation*

*(ou Viewport transformation ou Window-Viewport transformation)*

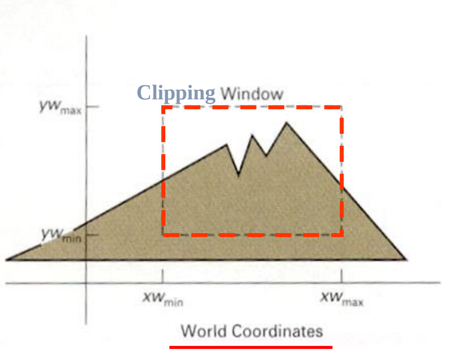
(em PT: Transformação janela de recorte – enquadramento)




Espaço do utilizador

Espaço da área de desenho

## Transformação
## *clipping window – viewport*




A ***clipping window*** (janela do utilizador)
é habitualmente:
• uma área rectangular com lados paralelos aos eixos principais
• que contém todos os dados que o utilizador quer ver representados na imagem.

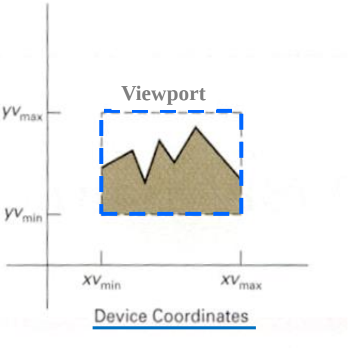
Os dados exteriores a esta área são recortados (*clipped).*



## Transformação
## *clipping window – viewport*

O *viewport* (enquadramento)
é habitualmente:
• uma área rectangular, com lados paralelos aos eixos principais
• que contém a imagem correspondente aos dados do utilizador contidos na *clipping window*.

O *viewport* está contido numa *display window* (área de desenho).

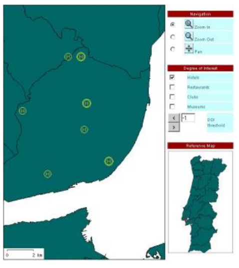


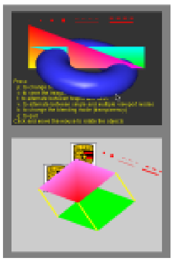

Espaço da área de desenho (*display window*)

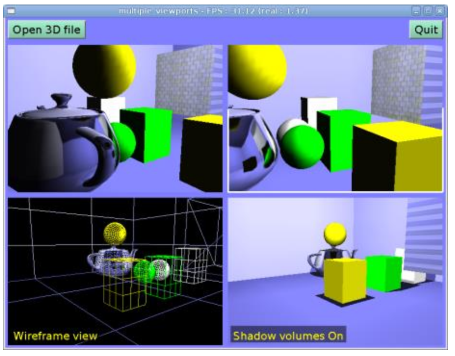
# Transformação
## *clipping window – viewport*

Uma mesma *display window* (área de desenho) pode conter vários *viewports* (enquadramentos).

**Exemplos:**

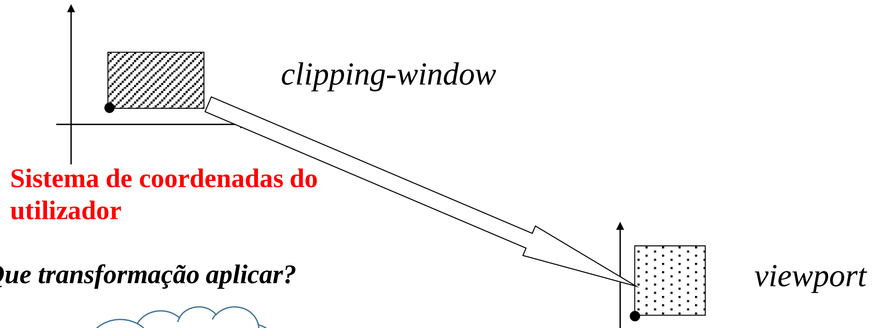








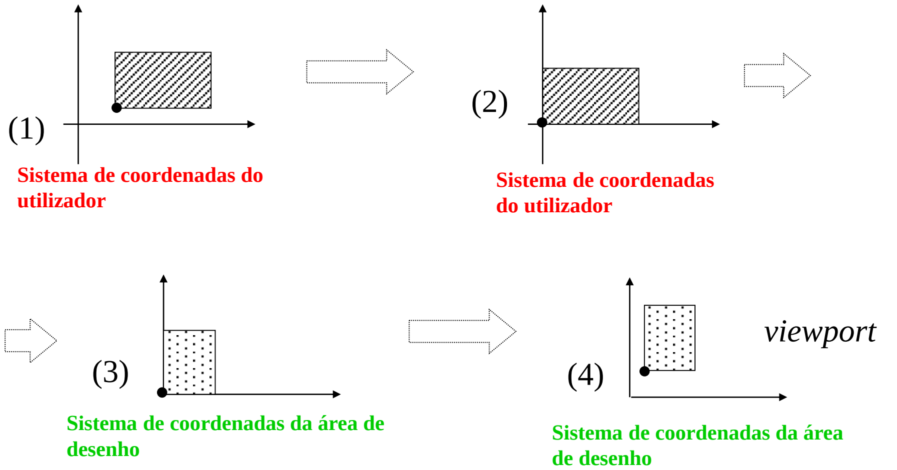

# Transformação
## *clipping window – viewport*




***Que transformação aplicar?***


***Mas S(sx,sy) tem um ponto invariante...***

## Transformação *clipping window – viewport*




**(1) ⟶ (2) Translação para a origem do referencial**
**(2) ⟶ (3) Mudança de Escala**
**(3) ⟶ (4) Translação para o canto inferior esquerdo do *viewport***



## Transformação *clipping window – viewport*

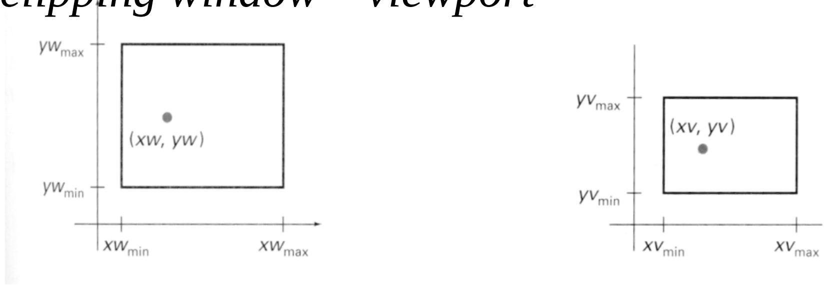

**Sistema de coordenadas do utilizador**

**Sistema de coordenadas da área de desenho**

$$\boxed{T(xv_{min}, yv_{min}) . S(s_x, s_y) . T(-xw_{min}, -yw_{min})}$$

***Transformação clipping window – viewport***

# Transformação
# *clipping window – viewport*




$$s_x = \frac{\text{Largura do } \textit{viewport}}{\text{Largura da } \textit{clipping window}} = \frac{xv_{max} - xv_{min}}{xw_{max} - xw_{min}}$$

$$s_y = \frac{\text{Altura do } \textit{viewport}}{\text{Altura da } \textit{clipping window}} = \frac{yv_{max} - yv_{min}}{yw_{max} - yw_{min}}$$



# Transformação
# *clipping window – viewportc*

Imagens **sem deformação** ⇔ a razão de aspecto(*) do *viewport* = razão de aspecto da *clipping window* ⇔ $\mathbf{s_x = s_y}$

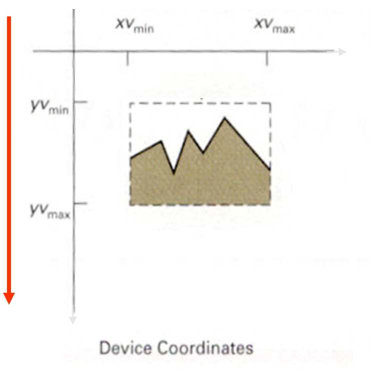
**Mudança de escala Uniforme**

**(*) Razão de aspecto (ou Relação de aspeto) = largura/altura**

# Transformação
## *clipping window – viewport*

Nalguns casos o sistema de coordenadas da área de desenho tem a sua **origem no canto superior esquerdo**.

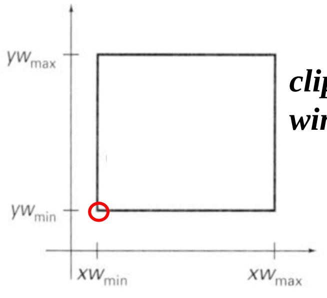
Coordenadas y crescem neste sentido



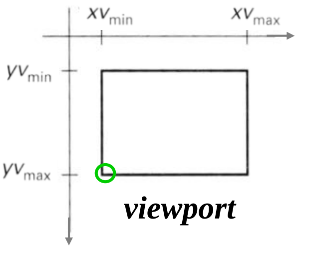

É preciso garantir que a imagem que obtemos não fica invertida.



# Transformação
## *clipping window – viewport*




***clipping window***




***viewport***

**Sistema de coordenadas do utilizador**

**Sistema de coordenadas da área de desenho**

$$\boxed{T(xv_{min}, yv_{max}) . S(s_x, -s_y) . T(-xw_{min}, -yw_{min})}$$

**Simetria em relação ao eixo dos xx**

**Transformação *clipping window – viewport***

## Exercício:

Suponha:

- uma primeira *clipping window* (janela do utilizador), rectangular, com vértices opostos em (1,3) e (3,9);
- uma segunda *clipping window* (janela do utilizador), quadrada, de lado unitário e canto inferior esquerdo em (2,4);
- uma área de desenho de 100 x 150 (largura x altura) com origem no canto inferior esquerdo.

Calcule:

a) A matriz de transformação *clipping window – viewport* que permite obter uma imagem do conteúdo da **primeira *clipping window*** na **metade esquerda** da área de desenho. Esta imagem deve ser **tão grande quanto possível** e **sem deformação**.

b) A matriz de transformação *clipping window – viewport* que permite obter uma imagem do conteúdo da **segunda *clipping window*** na **metade direita** da área de desenho. Esta imagem deve ser **tão grande quanto possível**, **sem deformação** e **centrada**.

c) A circunferência de centro (2.5,4.5) e raio 0.2 está contida nas duas *clipping windows.* Assim, terá duas representações na área de desenho, uma em cada metade desta. Caracterize estas duas representações.



Resultados finais:

a) $S(25,25).\ T(-1, -3) = \begin{bmatrix} 25 & 0 & -25 \\ 0 & 25 & -75 \\ 0 & 0 & 1 \end{bmatrix}$

b) $T(50,50)\ .\ S(50,50)\ .\ T(-2, -4) = \begin{bmatrix} 50 & 0 & -50 \\ 0 & 50 & -150 \\ 0 & 0 & 1 \end{bmatrix}$

c) Na transformação da alínea a) a imagem da circunferência é uma circunferência de centro (37.5, 37.5) e raio 0,2 x 25 = 5

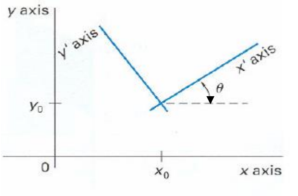
Na transformação da alínea b) a imagem da circunferência é uma circunferência de centro (75, 75) e raio 0,2 x 50 = 10

# Transformações entre sistemas de coordenadas 2D cartesianas

Exemplo: visualizar um mapa orientado de acordo com a direcção e sentido do movimento

Consideremos um sistema de coordenadas cartesianas ***x' y'*** com a origem no ponto ***(xo, yo)*** e um ângulo θ em relação ao sistema ***xy***

Para transformar descrições de objectos do sistema ***xy*** para ***x'y'***, determinamos uma transformação que sobreponha os eixos ***x'y'*** sobre os eixos ***xy*** :




1- Aplicar translação para colocar a origem de ***x'y', (xo, yo),*** na origem de ***xy, (0,0)***

2- Rodar o eixo ***x'*** por forma a coincidir com ***x***



# Outra notação Matricial

pontos podem ser representados por vetores linha

- Na maior parte dos livros e dos softwares gráficos (por exemplo, OpenGL, WebGL), **os pontos são representados por vectores coluna, como vamos uasr ao longo do semester nesta disciplina.**
- **Mas existem livros que usam outra notação matricial em que os pontos são representados** por vetores linha. Neste caso, a notação matricial é diferente da que usamos nesta disciplina.

  **Do ponto de vista matemático as duas notações são perfeitamente equivalentes!**

## Outra notação Matricial

pontos representados por vetores linha

Vejamos a relação entre as duas notações:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}^T = [\, x' \;\; y' \;\; 1 \,] \;=\; \left( M \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \right)^T$$

$$\underset{(*)}{=} [\, x \;\; y \;\; 1 \,] . M^T$$

(*) Lembrar que:

$(A . B)^T = B^T . A^T$



## Outra notação Matricial

pontos representados por vetores linha

- nesta outra notação matricial temos as seguintes matrizes:

Translação

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

Mudança de Escala

$$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotação

$$\begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

## Outra notação Matricial

pontos representados por vetores linha

Na concatenação, se aplicarmos **$M_2$ após $M_1$**:

$$[\, x' \;\; y' \;\; 1 \,] = [\, x \;\; y \;\; 1 \,] . M_1 . M_2$$

As matrizes das transformações elementares são as do slide anterior.

**Do ponto de vista matemático as duas notações são perfeitamente equivalentes!**