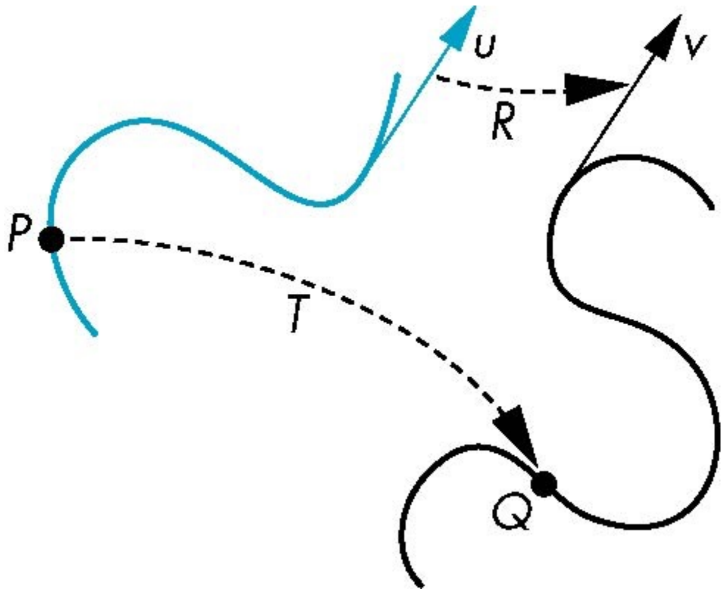
Transformações Geométricas

Objetivos

- Introduzir as transformações geométricas simples:
  - Translação
  - Rotação
  - Mudança de Escala
  - Deformação Transversa
- Derivar as respetivas matrizes de transformação com coordenadas homogéneas
- Aprender a usar a composição de transformações simples para deduzir transformações geométricas arbitrárias

Transformações Geométricas Genéricas

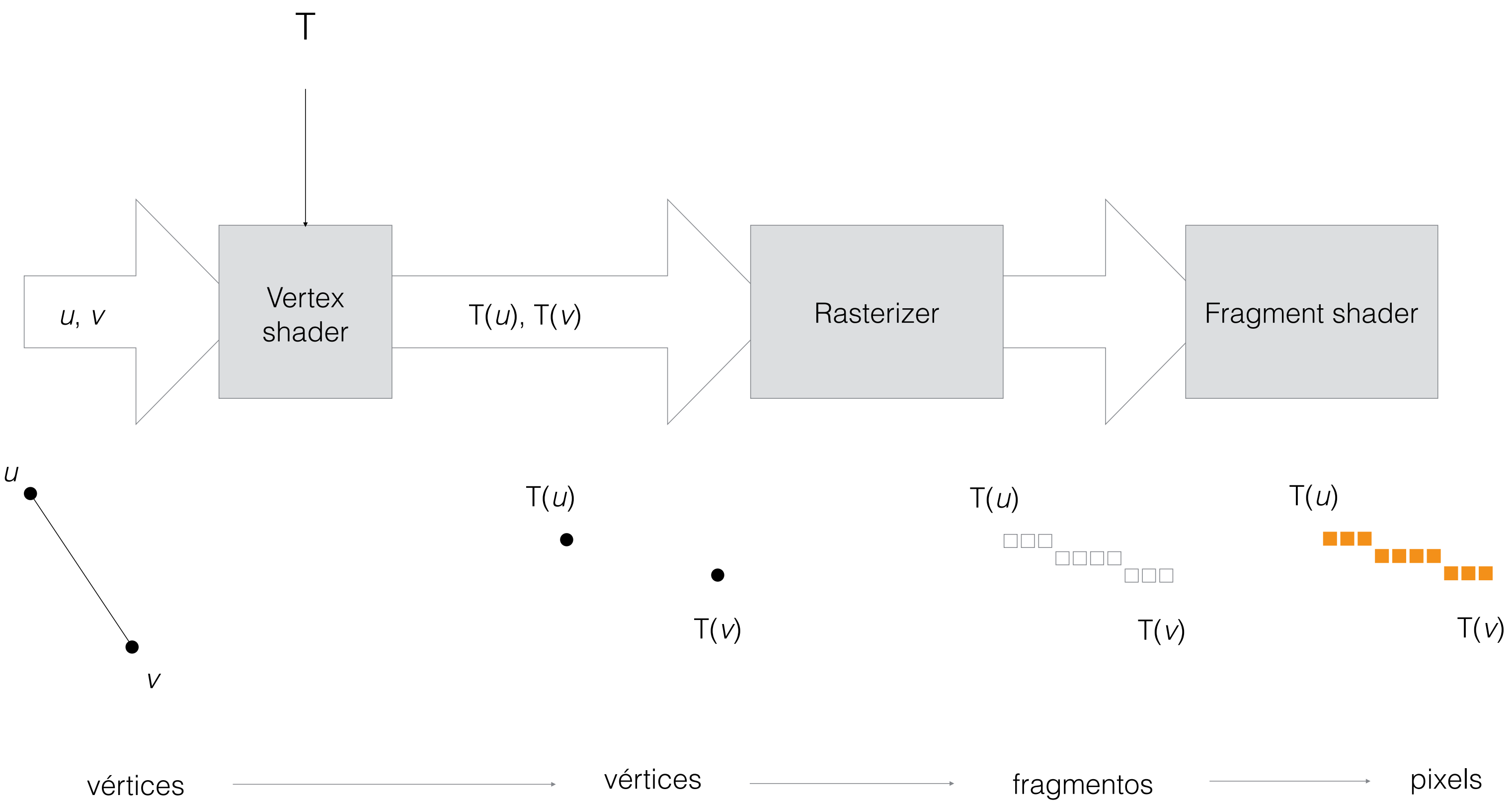
- Uma transformação geométrica qualquer transforma pontos e/ou vetores em outros pontos/vetores.



Transformações Afins

- Preservam as linhas
- As transformações mais importantes são transformações afins:
  - Transformações dos corpos rígidos (translação, rotação)
  - Mudança de escala e deformação transversa (shear)
- **Importância nos sistemas gráficos**:
  - Basta transformar os extremos duma linha (ou os vértices dum polígono) e unirmos os pontos transformados com uma linha (ou um polígono)

Tratamento no Pipeline

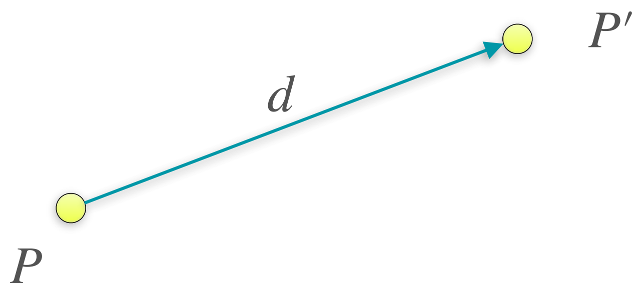


Notação

- P, Q, R: pontos num espaço afim
- u, v, w: vetores num espaço afim
- α, β, γ: escalares
- $\mathbf{p}, \mathbf{q}, \mathbf{r}$: representações de pontos - array com 4 escalares em coordenadas homogéneas
- $\mathbf{u}, \mathbf{v}, \mathbf{w}$: representações de vetores - array com 4 escalares em coordenadas homogéneas

Translação

- Mover (transladar, deslocar) um ponto para uma nova localização.



$$P' = P + d$$

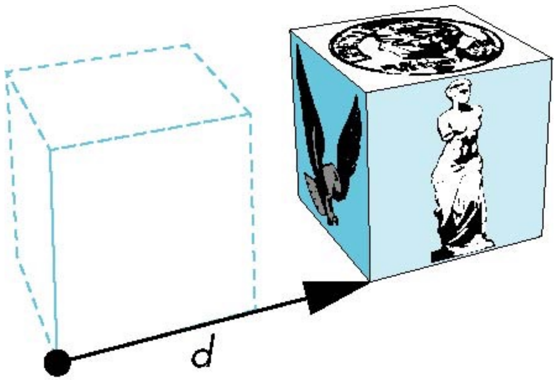
- Deslocamento determinado por um vetor d

- 3 graus de liberdade

Translação

- Embora um ponto se possa mover para outra localização mais do que de uma forma, para um conjunto de pontos existe normalmente apenas uma forma.

objeto



translação: todos os pontos deslocados dum mesmo vetor

Translação

- Usando a representação em coordenadas homogéneas num determinado referencial:

$$\mathbf{p} = \begin{bmatrix} p_x & p_y & p_z & 1 \end{bmatrix}^T$$
$$\mathbf{p}' = \begin{bmatrix} p'_x & p'_y & p'_z & 1 \end{bmatrix}^T$$
$$\mathbf{d} = \begin{bmatrix} d_x & d_y & d_z & 0 \end{bmatrix}^T$$

Componente w

Esta expressão é em 4D e exprime o conceito: ponto = ponto + vetor

$$\begin{aligned} p'_x &= p_x + d_x \\ p'_y &= p_y + d_y \\ p'_z &= p_z + d_z \\ p'_w &= 1 \end{aligned} \quad \Rightarrow \quad \mathbf{p}' = \mathbf{p} + \mathbf{d}$$

Translação (Matriz de Transformação)

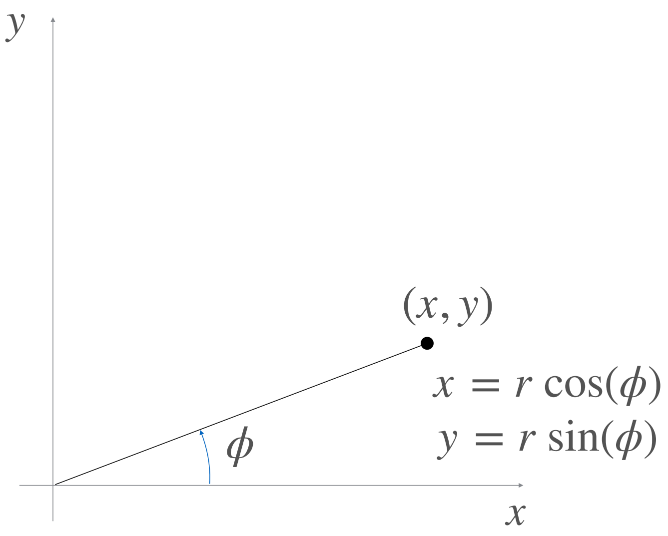
- Também se pode representar uma translação usando uma matriz $\mathbf{M}$, de 4x4, usando coordenadas homogéneas, de tal modo que:

$$\mathbf{p}' = \mathbf{M}\mathbf{p} \qquad \mathbf{M} = T(d_x, d_y, d_z) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- A vantagem desta abordagem é que todas as transformações afins se podem exprimir desta forma, podendo várias transformações ser concatenadas e guardadas numa única matriz.

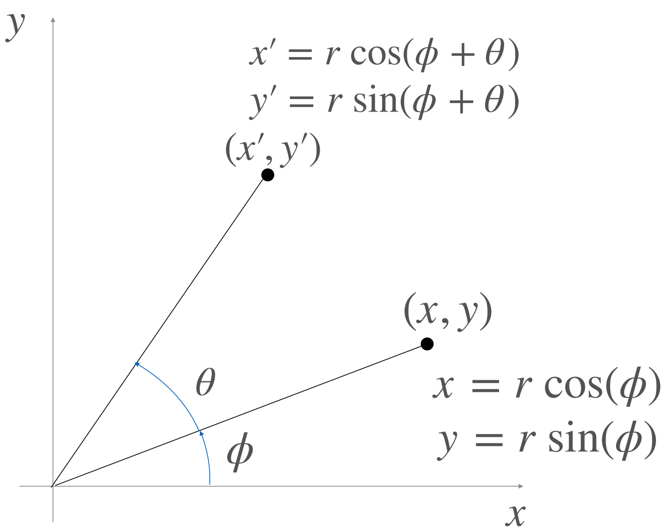
Rotação (2D)

- Considerando coordenadas polares de um ponto $P(x, y) = (r\cos(\phi), r\sin(\phi))$, na rotação em torno da origem de um ângulo θ, o raio mantém-se constante, aumentando o ângulo inicial em θ:



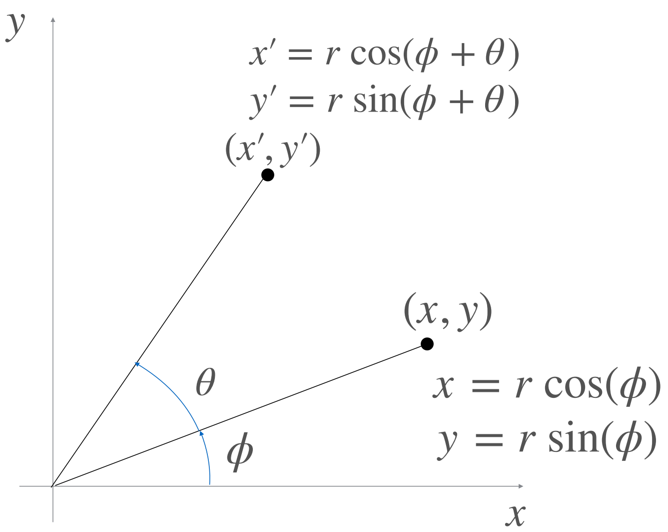
Rotação (2D)

- Considerando coordenadas polares de um ponto $P(x, y) = (r\cos(\phi), r\sin(\phi))$, na rotação em torno da origem de um ângulo θ, o raio mantém-se constante, aumentando o ângulo inicial em θ:



Rotação (2D)

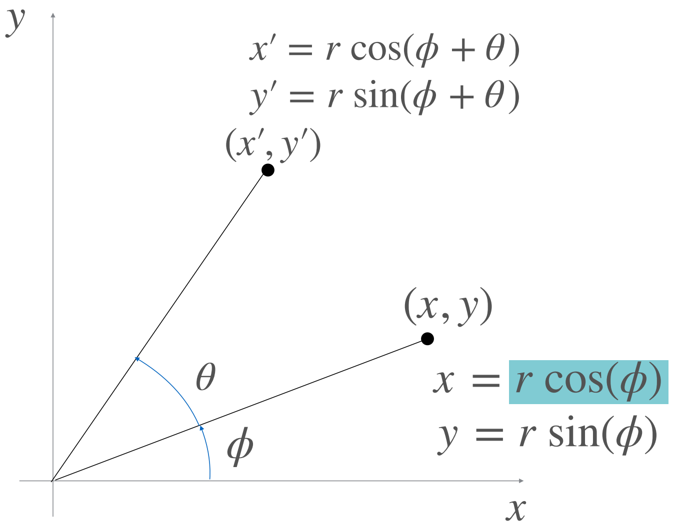
- Considerando coordenadas polares de um ponto $P(x, y) = (r\cos(\phi), r\sin(\phi))$, na rotação em torno da origem de um ângulo θ, o raio mantém-se constante, aumentando o ângulo inicial em θ:



$$x' = r\cos(\phi)\cos(\theta) - r\sin(\phi)\sin(\theta)$$
$$y' = r\sin(\phi)\cos(\theta) + r\cos(\phi)\sin(\theta)$$

Rotação (2D)

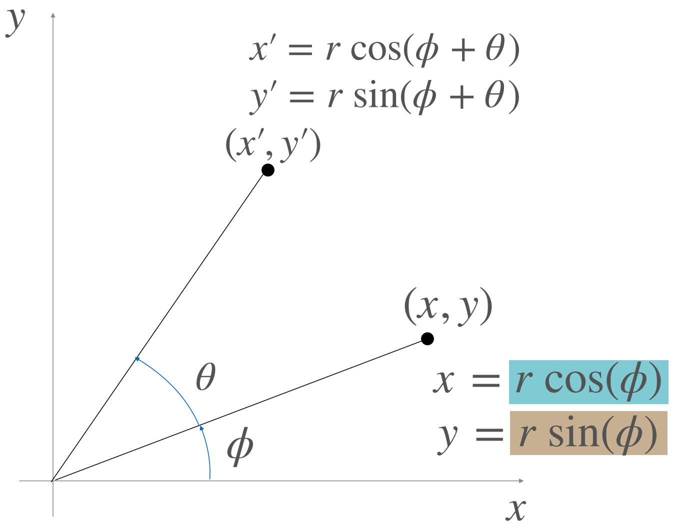
- Considerando coordenadas polares de um ponto $P(x, y) = (r\cos(\phi), r\sin(\phi))$, na rotação em torno da origem de um ângulo θ, o raio mantém-se constante, aumentando o ângulo inicial em θ:



$$x' = r\cos(\phi)\cos(\theta) - r\sin(\phi)\sin(\theta)$$
$$y' = r\sin(\phi)\cos(\theta) + r\cos(\phi)\sin(\theta)$$

Rotação (2D)

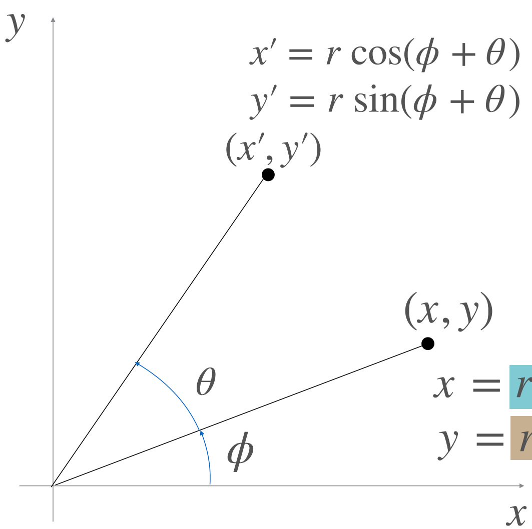
- Considerando coordenadas polares de um ponto $P(x, y) = (r\cos(\phi), r\sin(\phi))$, na rotação em torno da origem de um ângulo θ, o raio mantém-se constante, aumentando o ângulo inicial em θ:



$$x' = r\cos(\phi)\cos(\theta) - r\sin(\phi)\sin(\theta)$$
$$y' = r\sin(\phi)\cos(\theta) + r\cos(\phi)\sin(\theta)$$

Rotação (2D)

- Considerando coordenadas polares de um ponto $P(x, y) = (r\cos(\phi), r\sin(\phi))$, na rotação em torno da origem de um ângulo θ, o raio mantém-se constante, aumentando o ângulo inicial em θ:



$$x' = r\cos(\phi + \theta)$$
$$y' = r\sin(\phi + \theta)$$

$$x = r\cos(\phi)$$
$$y = r\sin(\phi)$$

$$x' = r\cos(\phi)\cos(\theta) - r\sin(\phi)\sin(\theta)$$
$$y' = r\sin(\phi)\cos(\theta) + r\cos(\phi)\sin(\theta)$$

$$x' = x\cos(\theta) - y\sin(\theta)$$
$$y' = x\sin(\theta) + y\cos(\theta)$$

Rotação em torno de Z (3D)

- Em 3D, a rotação em torno de z deixa todos os pontos com o valor da coordenada z inalterada.
- É equivalente à rotação 2D em planos de z constante:

$$
\begin{aligned}
x' &= x\cos(\theta) - y\sin(\theta) \\
y' &= x\sin(\theta) + y\cos(\theta) \\
z' &= z
\end{aligned}
$$

- Em coordenadas homogéneas:

$$\mathbf{p}' = \mathbf{R}_z(\theta)\mathbf{p} \qquad \mathbf{R}_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matrizes de rotação em 3D

- De forma análoga:

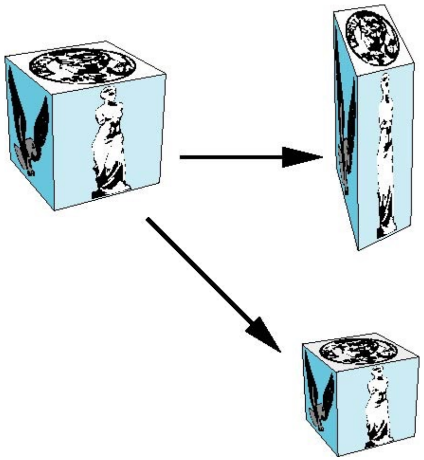
  Rotação em torno de x não altera a coordenada x

  Rotação em torno de y não altera a coordenada y

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Mudança de escala

- Expansão ou contração ao longo de cada um dos eixos, com o ponto fixo na origem

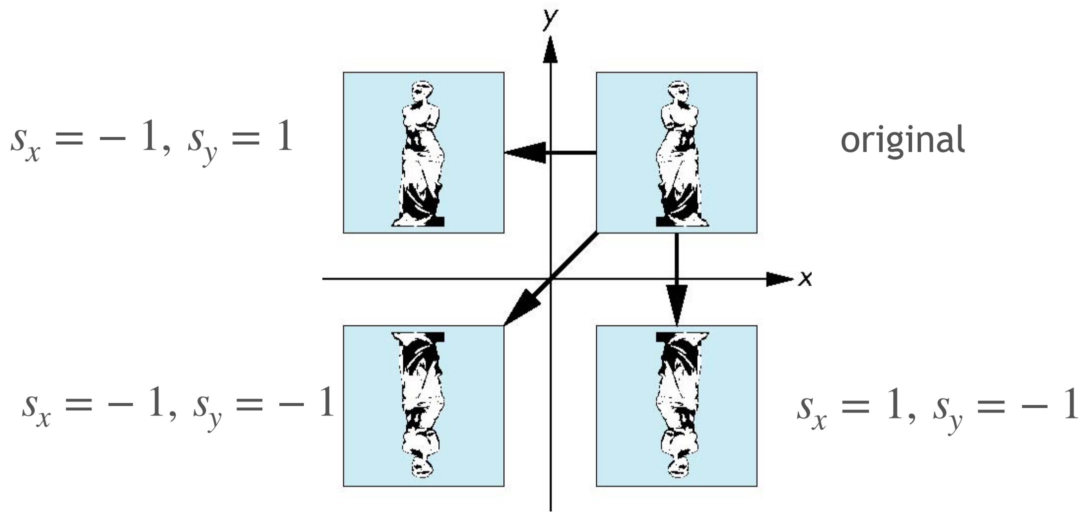


$$\mathbf{p}' = \mathbf{S}(s_x, s_y, s_z)\mathbf{p}$$

$$\mathbf{S}(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Reflexões

- São um caso particular da mudança de escala (com fatores unitários negativos)



Transformações inversas

- As transformações inversas das transformações simples podem ser obtidas sem recorrer a fórmulas gerais:
  - Translação: $\mathbf{T}^{-1}(d_x, d_y, d_z) = \mathbf{T}(-d_x, -d_y, -d_z)$
  - Rotação: $\mathbf{R}_\omega^{-1}(\theta) = \mathbf{R}_\omega(-\theta) = (\mathbf{R}_\omega(\theta))^T \qquad \omega \in \{x, y, z\}$
  - Escala: $\mathbf{S}^{-1}(s_x, s_y, s_z) = \mathbf{S}(1/s_x, 1/s_y, 1/s_z)$

Composição de transformações geométricas

- Podem formar-se transformações afins arbitrárias por composição das transformações simples: rotações, translação e mudança de escala
- Supondo que temos um conjunto grande de vértices $\mathbf{p}_i$, para transformar com uma composição de transformações simples $\mathbf{M}_n \cdot \cdots \cdot \mathbf{M}_2 \cdot \mathbf{M}_1$, o custo do cálculo de $\mathbf{M} = \mathbf{M}_n \cdot \cdots \cdot \mathbf{M}_2 \cdot \mathbf{M}_1$, é reduzido quando comparado com o custo de $\mathbf{M} \cdot \mathbf{p}_f$, para um grande número de vértices.
- O desafio é encontrar a composição certa para fazer o que se pretende numa dada aplicação.

Ordem de aplicação

- O produto de matrizes (a composição de transformações) é associativo, mas não é comutativo!

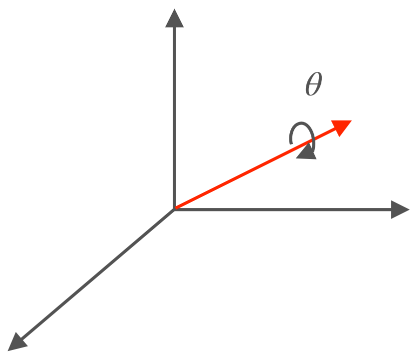
$$\mathbf{p}' = \mathbf{ABCp} = \mathbf{A}(\mathbf{B}(\mathbf{Cp}))$$

- Se se usassem vetores linha para representar os pontos, a transformação acima seria:

$$\mathbf{p}' = \mathbf{pC}^T\mathbf{B}^T\mathbf{A}^T = ((\mathbf{pC}^T)\mathbf{B}^T)\mathbf{A}^T$$

Rotação em torno de eixo arbitrário (que cruza a origem)

- Transformação complexa que pode ser decomposta em transformações simples
- Uma rotação de θ em torno dum eixo arbitrário, v, pode ser decomposta em rotações em torno dos eixos x, y e z



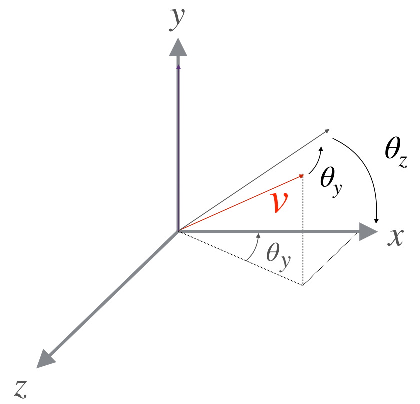
$$\mathbf{R}_{\mathbf{v}}(\theta) = \mathbf{R}_x(\theta_x)\mathbf{R}_y(\theta_y)\mathbf{R}_z(\theta_z)$$

- Os ângulos θ_x, θ_y e θ_z denominam-se de ângulos de Euler.
- Embora as rotações não se possam trocar de ordem, é possível encontrar 3 outros ângulos de Euler, para outra ordem de aplicação das transformações, que produzem o mesmo efeito.

Rotação em torno de eixo arbitrário (que cruza a origem)

- Alternativamente, poder-se-ia deduzir uma expressão que faz a mesma transformação usando os seguintes passos:

1. Rodar θ_y, segundo y, o vetor v que passa pelo eixo de rotação, por forma a colocá-lo no plano xy
2. Rodar θ_z, segundo z, por forma a que o vetor de 1 fique coincidente com o eixo x
3. Rodar θ, em torno de x
4. Desfazer as rotações dos passos 1 e 2, pela ordem inversa



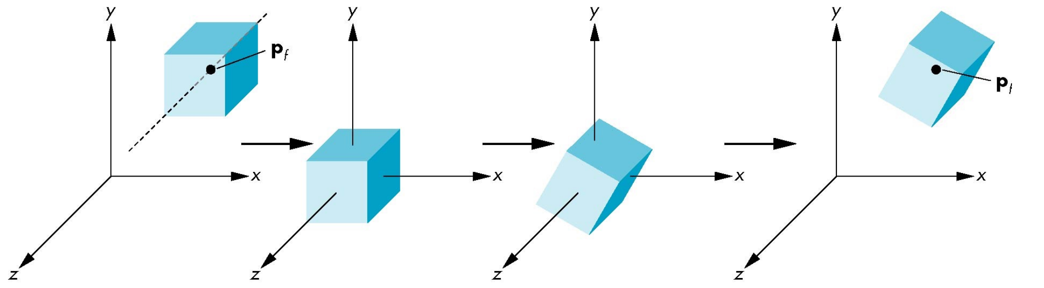
$$\mathbf{R_v}(\theta) = \mathbf{R}_y(-\theta_y)\mathbf{R}_z(-\theta_z)\mathbf{R}_x(\theta)\mathbf{R}_z(\theta_z)\mathbf{R}_y(\theta_y)$$

$$\theta_y = \tan^{-1}\left(\frac{v_z}{v_x}\right)$$

$$\theta_z = -\tan^{-1}\left(\frac{v_y}{\sqrt{v_x^2 + v_z^2}}\right)$$

Rotação em torno dum eixo que não passa na origem

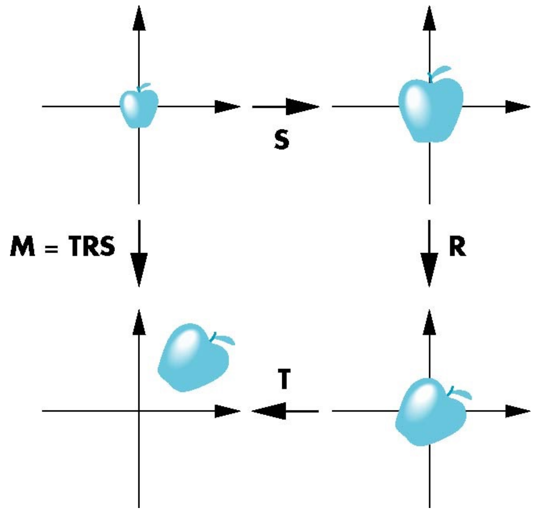
- Usa-se o resultado anterior, mas com a seguinte alteração:
  1. Mover um ponto $\mathbf{p}_f$, pertencente ao eixo de rotação, para a origem
  2. Rodar $\mathbf{R}_\mathbf{v}(\theta)$
  3. Mover o ponto $\mathbf{p}_f$ de volta para a sua posição inicial



$$\mathbf{M} = \mathbf{T}(\mathbf{p}_f)\mathbf{R}_\mathbf{v}(\theta)\mathbf{T}(-\mathbf{p}_f)$$

Transformações de instanciação

- Em modelação, os objetos primitivos estão normalmente centrados na origem, orientados com os eixos principais e com um determinado tamanho
- Aos vértices desses objetos “primitivos” aplicam-se transformações que seguem muitas vezes o padrão:
    - Escala
    - Orientação
    - Posicionamento



Transformação de Instanciação Comum

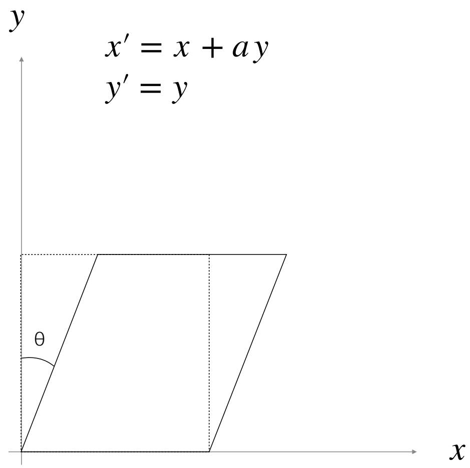
Transformações de instanciação

- Mas também é possível aplicar uma sequência arbitrária de transformações geométricas (acumulação), sem nenhuma ordem especial que não a definida pelo modelador.

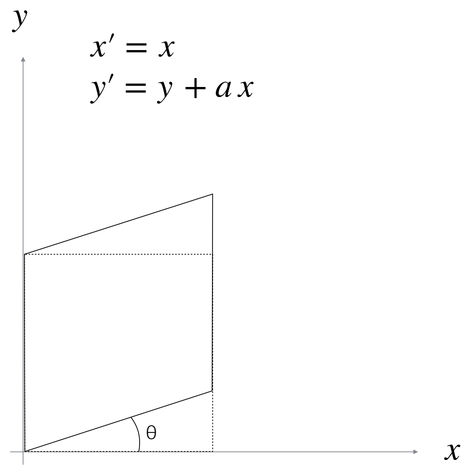
- A maior parte do software adopta a ordem fixa $\mathbf{T} \cdot \mathbf{R} \cdot \mathbf{S}$:
  - Blender
  - Unity
  - Maya
  - etc.

- Mas em alguns casos também se permite alterar essa ordem ou efetuar composições arbitrárias de transformações

Deformação Transversa em 2D



$$a = \tan(\theta)$$



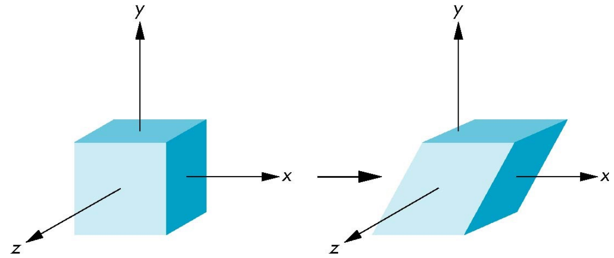
$$\mathbf{SH}_y(\theta) = \begin{bmatrix} 1 & \tan(\theta) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{SH}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ \tan(\theta) & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Nota sobre a notação: Estamos a usar como índice o eixo que não sofre qualquer alteração

Deformação Transversa

- Exemplo dum caso particular em 3D



- A matriz de deformação transversa, para o caso geral, é dada por:

$$\begin{bmatrix} 1 & a & b & 0 \\ c & 1 & d & 0 \\ e & f & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x' = x + ay + bz$$
$$y' = y + cx + dz$$
$$z' = z + ex + fy$$

Operações comutativas

- Apesar de, no caso geral, a ordem das transformações ser importante, em alguns casos pode trocar-se a ordem das operações visto serem comutativas:
  - $\mathbf{S}(a,b,c)\mathbf{S}(d,e,f) = \mathbf{S}(d,e,f)\mathbf{S}(a,b,c) = \mathbf{S}(ad,be,cf)$
  - $\mathbf{T}(a,b,c)\mathbf{T}(d,e,f) = \mathbf{T}(d,e,f)\mathbf{T}(a,b,c) = \mathbf{T}(a+d,b+e,c+f)$
  - $\mathbf{R}_\omega(\theta)\mathbf{R}_\omega(\phi) = \mathbf{R}_\omega(\phi)\mathbf{R}_\omega(\theta) = \mathbf{R}_\omega(\theta+\phi), \omega \in \{x,y,z\}$
  - $\mathbf{R}_x(\theta)\mathbf{S}(a,k,k) = \mathbf{S}(a,k,k)\mathbf{R}_x(\theta)$
  - $\mathbf{R}_y(\theta)\mathbf{S}(k,a,k) = \mathbf{S}(k,a,k)\mathbf{R}_y(\theta)$
  - $\mathbf{R}_z(\theta)\mathbf{S}(k,k,a) = \mathbf{S}(k,k,a)\mathbf{R}_z(\theta)$

Escala uniforme em dois dos eixos e rotação no 3º eixo

$$\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}s & 0\\ 0 & s\end{bmatrix} = \begin{bmatrix}s & 0\\ 0 & s\end{bmatrix}\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}$$

Transformação de enquadramento

Transformação de Enquadramento (2D)

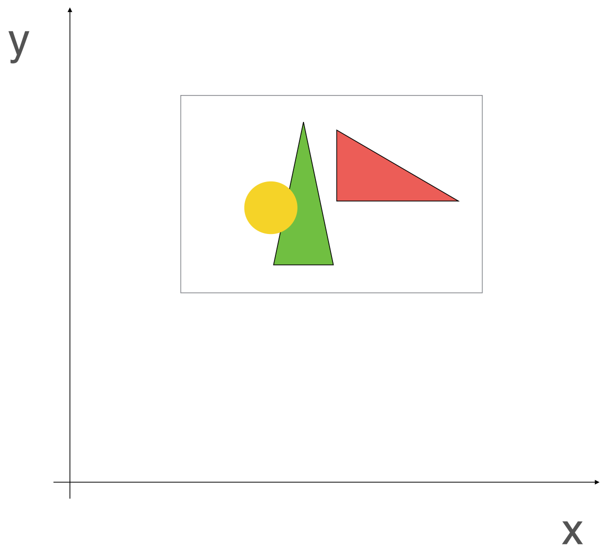
- Um problema recorrente na visualização 2D, é o da passagem do referencial do mundo (**W**orld **C**oordinates), para o referencial do dispositivo (**D**evice **C**oordinates) onde se vai proceder à visualização.
- Como a superfície de visualização (dispositivo) possui uma área limitada, com forma retangular, o problema pode ser colocado como uma transformação duma área retangular definida em **WC**, designada por **janela** (**window**), numa outra, também retangular, definida em **DC** e designada por **visor** (**viewport**)

Transformação de Enquadramento (2D)

- Dada uma janela definida em WC, bem como um visor, definido em DC (Device coordinates), determinar a transformação geométrica que transforma pontos do modelo (WC) para as suas posições no dispositivo (DC).
- A definição, quer da janela, quer do visor, é feita através dos seus limites.
- Distinguir 2 casos:
  A. Dispositivo com origem no canto inferior esquerdo;
  B. Dispositivo com origem no canto superior esquerdo.

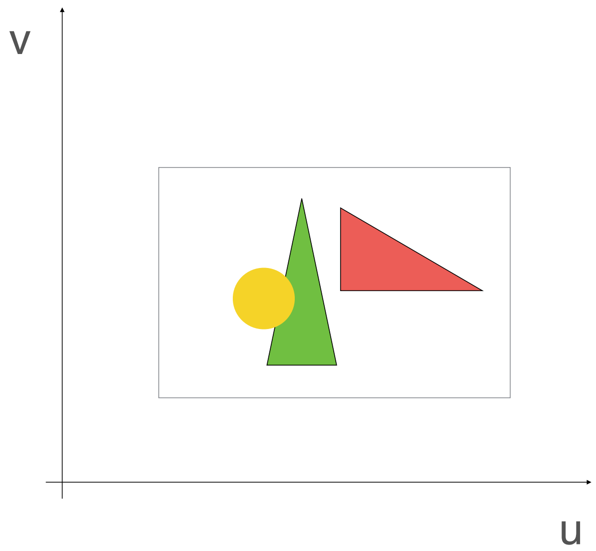
Enquadramento Janela-Visor (Caso A)

WC - World Coordinates



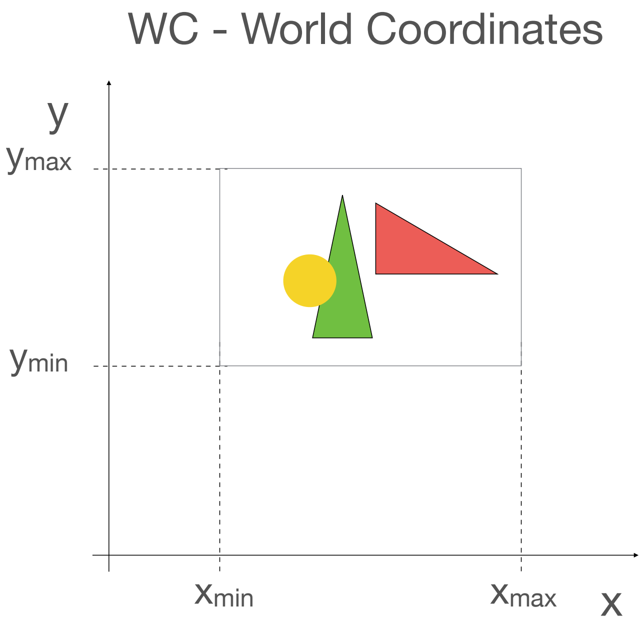
Referencial associado ao modelo/cena que se pretende visualizar. As coordenadas são dependentes do problema e podem ser dadas em metros, centímetros, unidades astronómicas, anos luz, microns, etc.

DC - Device Coordinates

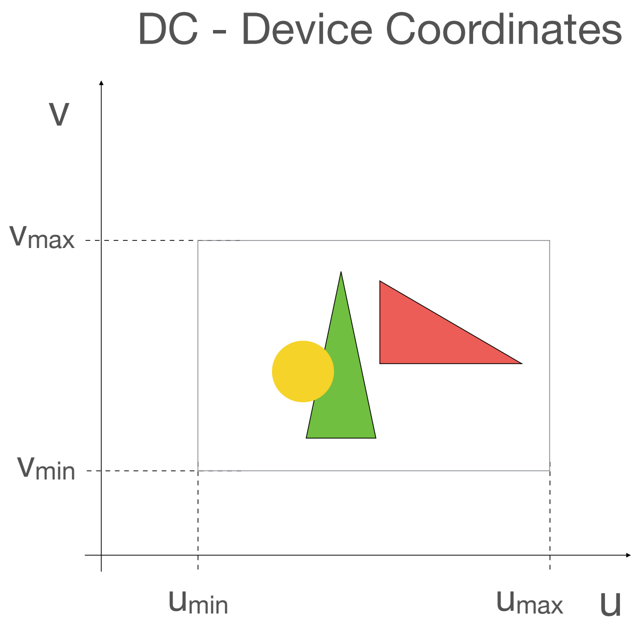


Referencial associado ao dispositivo ou a uma área disponível do mesmo. Exemplos: ecrã, janela duma aplicação no ecrã, canvas numa página HTML, página A4 numa impressora, etc.
As coordenadas são as do dispositivo.

Enquadramento Janela-Visor (Caso A)



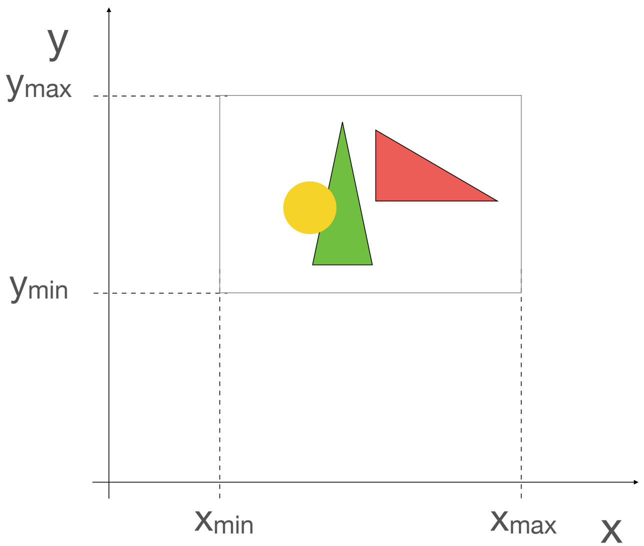
Limites da janela



Limites do visor

Enquadramento Janela-Visor (Caso A)

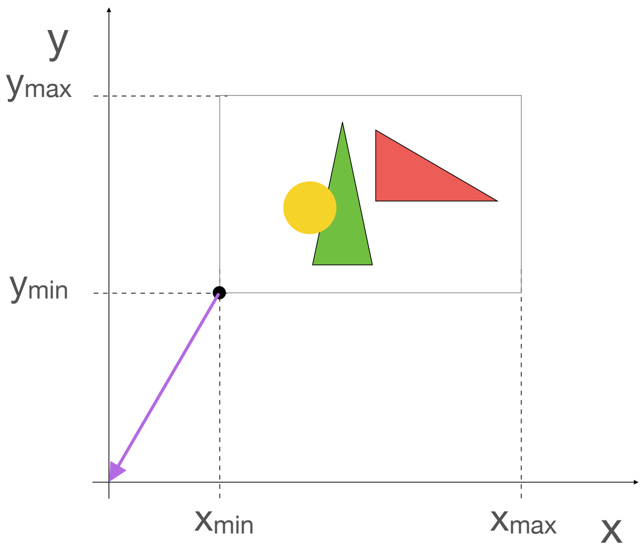
WC - World Coordinates



Escolher um **ponto de referência** (por exemplo, o canto inferior esquerdo) e **aplicar uma translação** para o colocar na **origem** do referencial

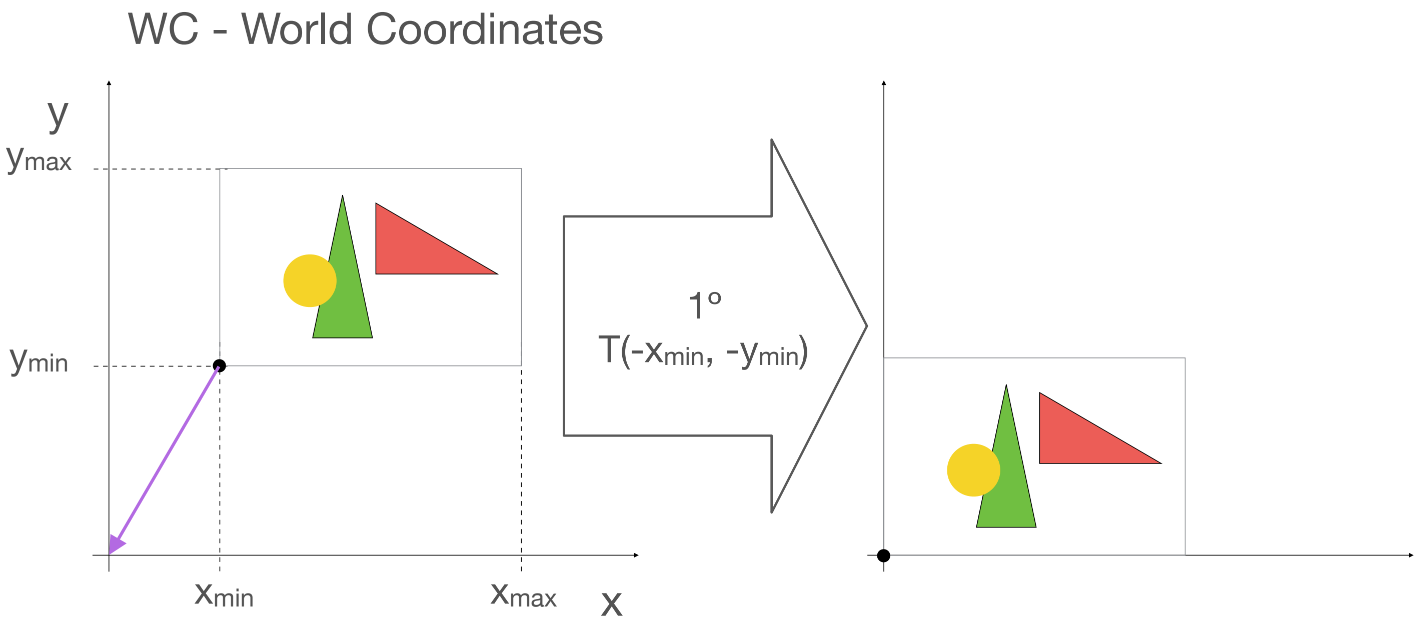
Enquadramento Janela-Visor (Caso A)

WC - World Coordinates



Escolher um **ponto de referência** (por exemplo, o canto inferior esquerdo) e **aplicar uma translação** para o colocar na **origem** do referencial

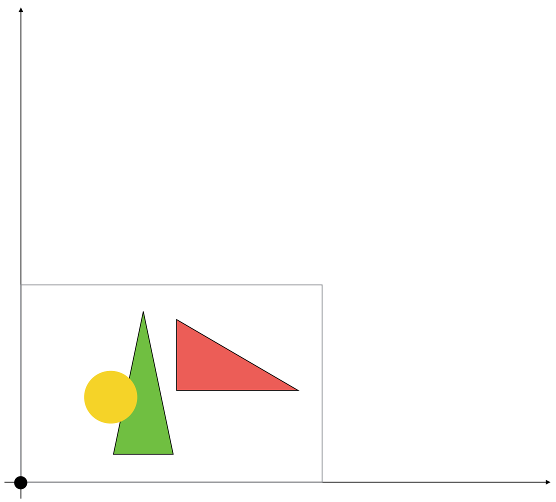
Enquadramento Janela-Visor (Caso A)



Escolher um **ponto de referência** (por exemplo, o canto inferior esquerdo) e **aplicar uma translação** para o colocar na **origem** do referencial

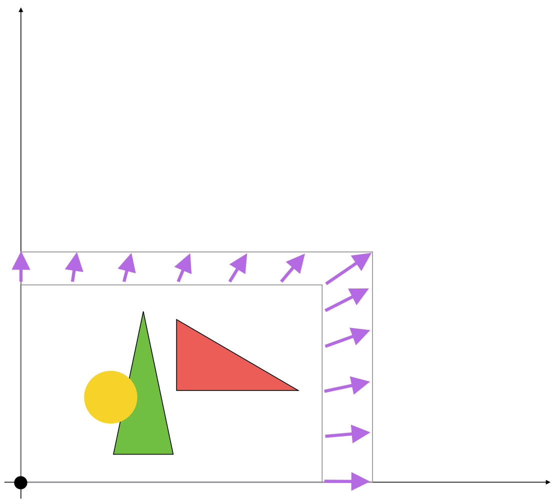
O nosso retângulo inicial está agora, com as **mesmas dimensões** iniciais, mas com o **ponto de referência** na **origem** do referencial

Enquadramento Janela-Visor (Caso A)



Aplicar uma mudança de escala para transformar
as dimensões do retângulo inicial nas dimensões
do retângulo final

Enquadramento Janela-Visor (Caso A)

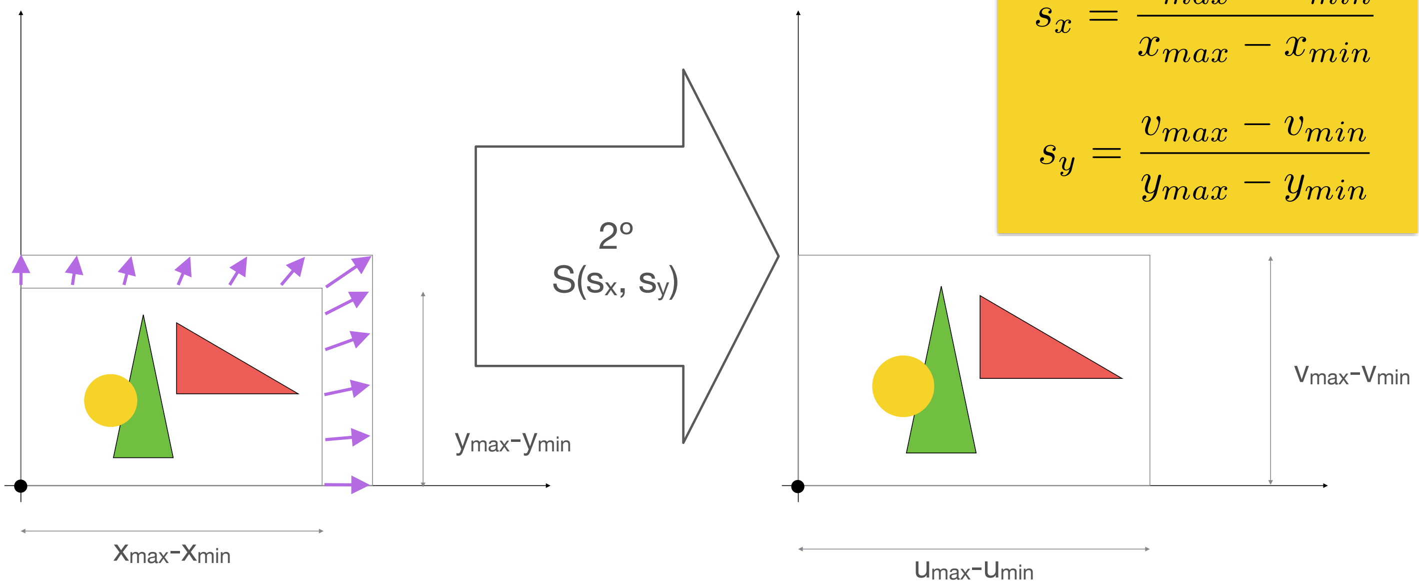


Aplicar uma mudança de escala para **transformar as dimensões** do retângulo inicial nas dimensões do retângulo final

Enquadramento Janela-Visor (Caso A)

$$s_x = \frac{u_{max} - u_{min}}{x_{max} - x_{min}}$$

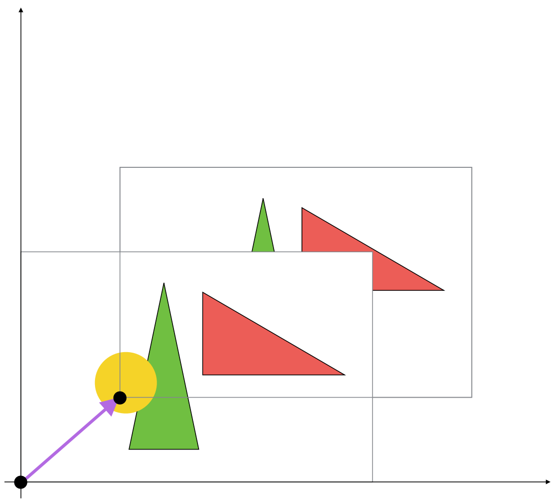
$$s_y = \frac{v_{max} - v_{min}}{y_{max} - y_{min}}$$



Aplicar uma mudança de escala para transformar as dimensões do retângulo inicial nas dimensões do retângulo final

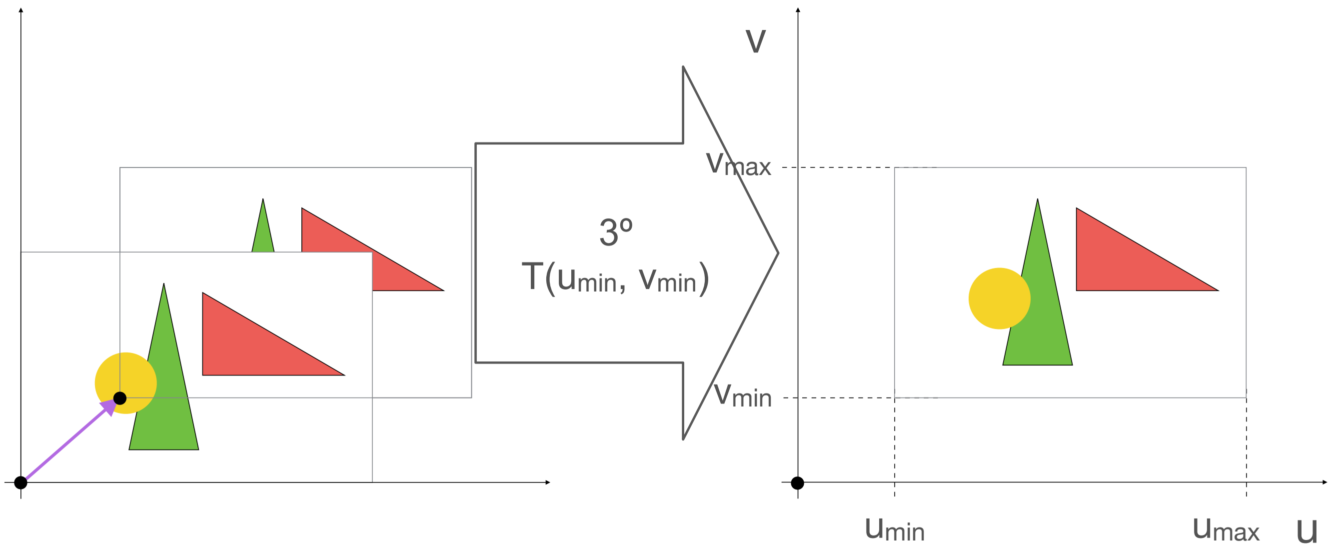
O retângulo continua com o ponto de referência na origem, mas já possui as dimensões finais pretendidas

Enquadramento Janela-Visor (Caso A)



Basta agora deslocar o retângulo para a sua posição final. O ponto de referência, agora na origem, deve deslocar-se para (u_{min},v_{min})

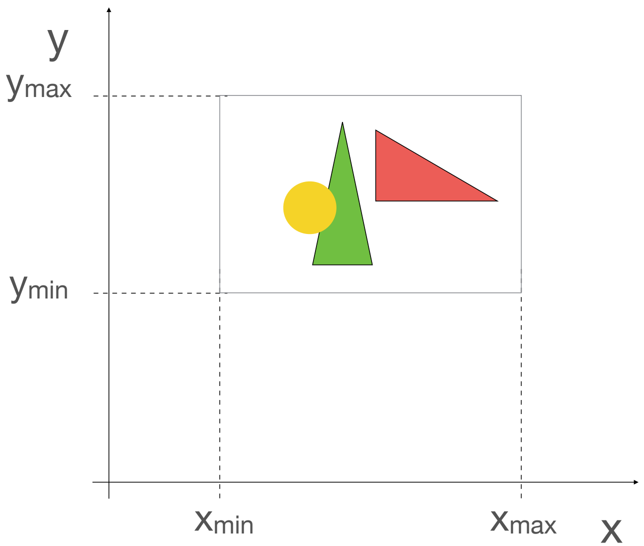
Enquadramento Janela-Visor (Caso A)



Basta agora deslocar o retângulo para a sua posição final. O ponto de referência, agora na origem, deve deslocar-se para (u_{min},v_{min})

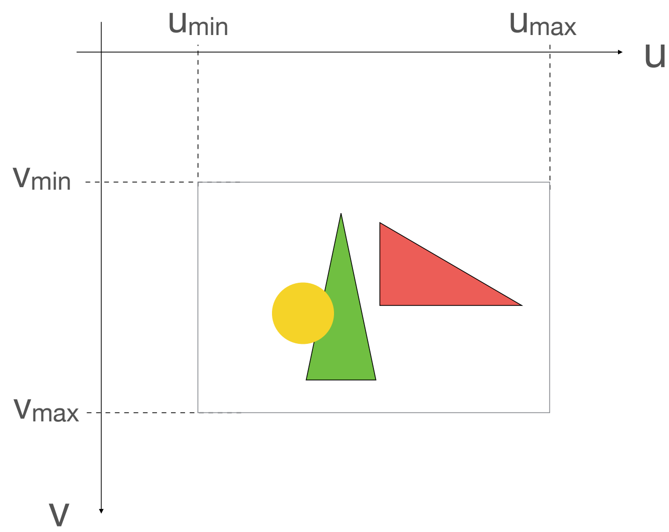
Enquadramento Janela-Visor (Caso B)

WC - World Coordinates



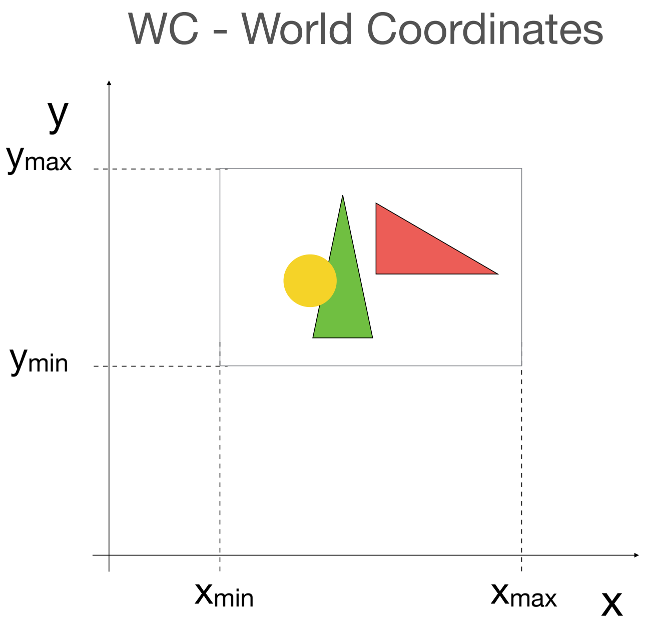
Referencial associado ao modelo/cena que se pretende visualizar. As coordenadas são dependentes do problema e podem ser dadas em metros, centímetros, unidades astronómicas, anos luz, microns, etc.

DC - Device Coordinates



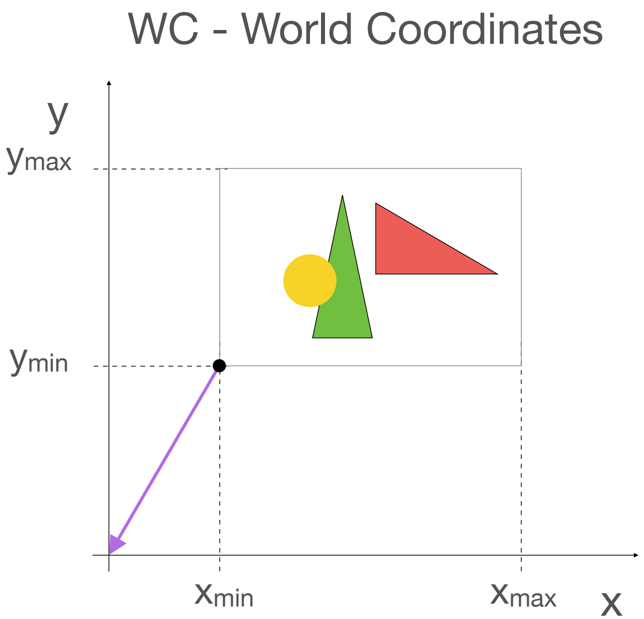
Referencial associado ao dispositivo ou a uma área disponível do mesmo. Exemplos: ecrã, janela duma aplicação no ecrã, canvas numa página HTML, página A4 numa impressora, etc.
As coordenadas são as do dispositivo.

Enquadramento Janela-Visor (Caso B)



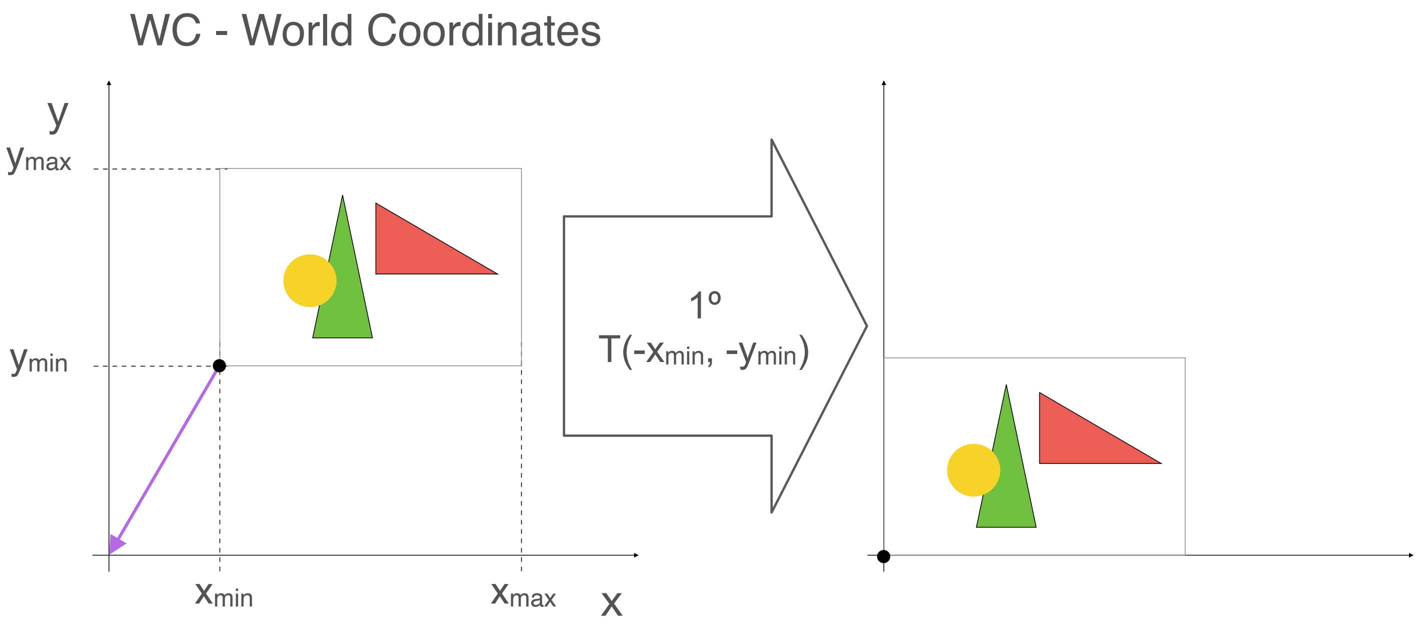
Escolher um **ponto de referência** (por exemplo, o canto inferior esquerdo) e **aplicar uma translação** para o colocar na **origem** do referencial

Enquadramento Janela-Visor (Caso B)



Escolher um **ponto de referência** (por exemplo, o canto inferior esquerdo) e **aplicar uma translação** para o colocar na **origem** do referencial

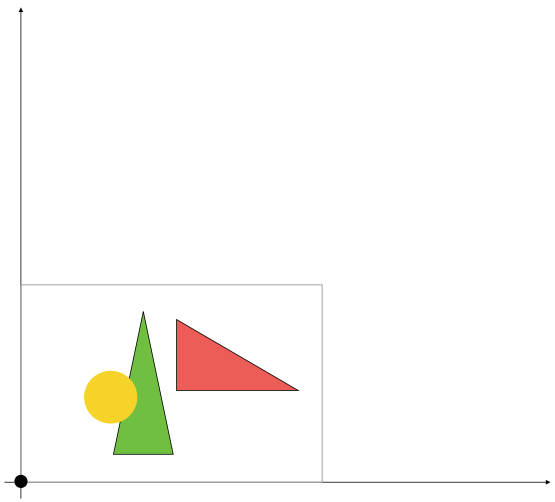
Enquadramento Janela-Visor (Caso B)



Escolher um **ponto de referência** (por exemplo, o canto inferior esquerdo) e **aplicar uma translação** para o colocar na **origem** do referencial

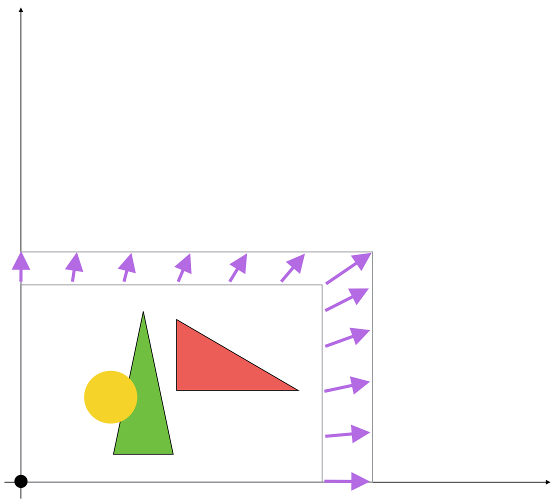
O nosso retângulo inicial está agora, com as **mesmas dimensões** iniciais, mas com o **ponto de referência** na **origem** do referencial

Enquadramento Janela-Visor (Caso B)



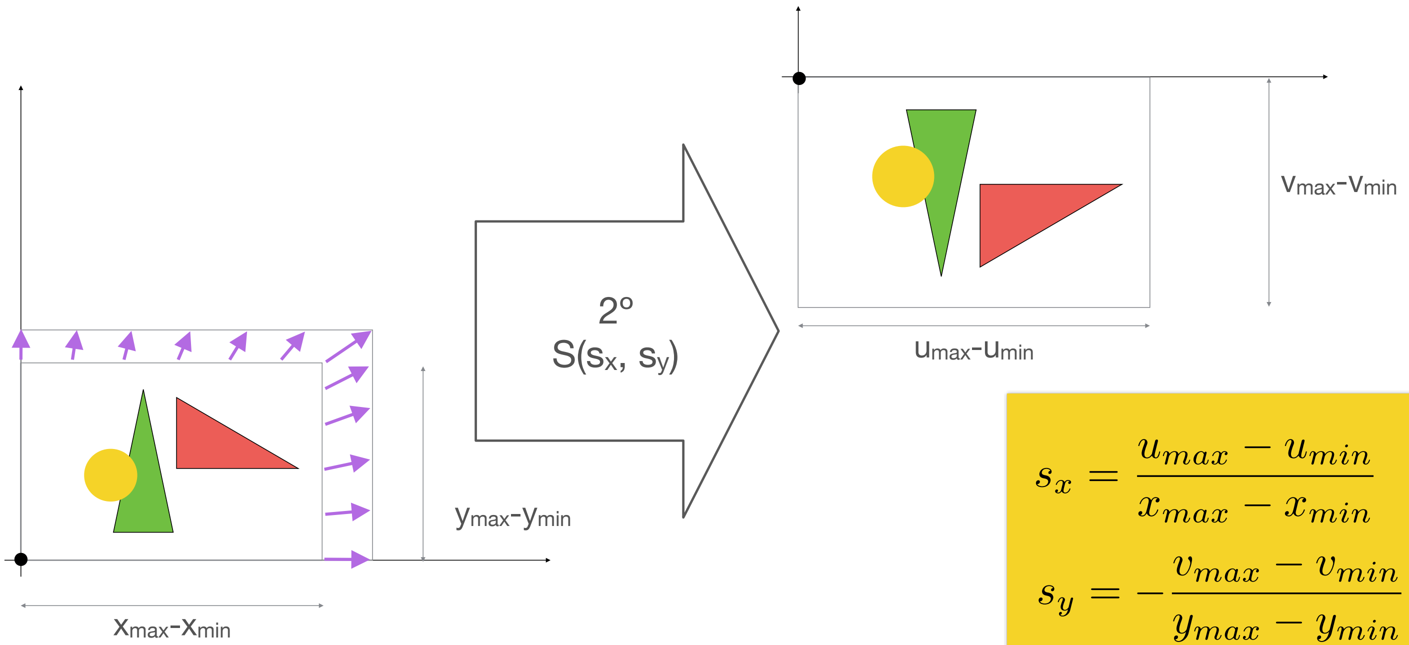
Aplicar uma mudança de escala para **transformar as dimensões** do retângulo inicial nas dimensões do retângulo final

Enquadramento Janela-Visor (Caso B)



Aplicar uma mudança de escala para **transformar as dimensões** do retângulo inicial nas dimensões do retângulo final

Enquadramento Janela-Visor (Caso B)



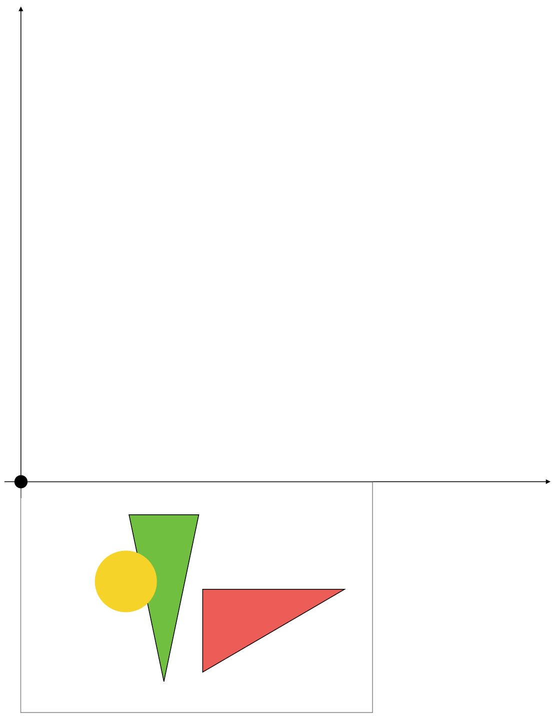
$$s_x = \frac{u_{max} - u_{min}}{x_{max} - x_{min}}$$

$$s_y = -\frac{v_{max} - v_{min}}{y_{max} - y_{min}}$$

Aplicar uma mudança de escala para **transformar as dimensões** do retângulo inicial nas dimensões do retângulo final

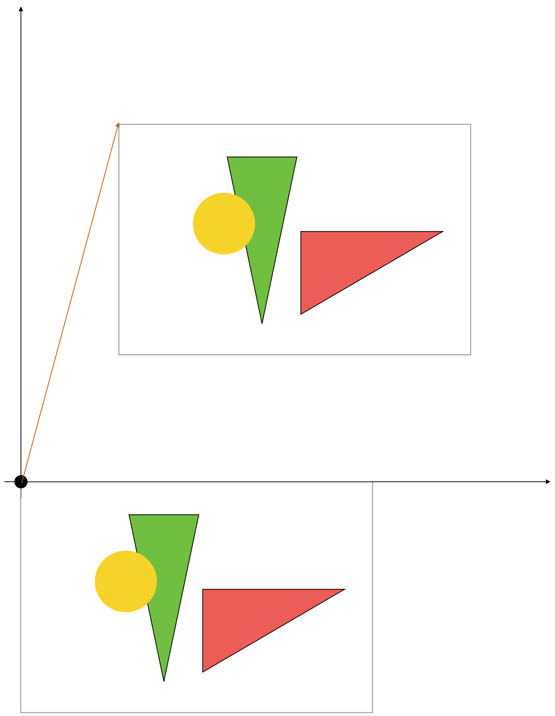
Mas com um **fator negativo no eixo y**…

Enquadramento Janela-Visor (Caso B)



O ponto de referência está na origem, mas a sua posição final deveria ser (u_{min}, v_{max})…

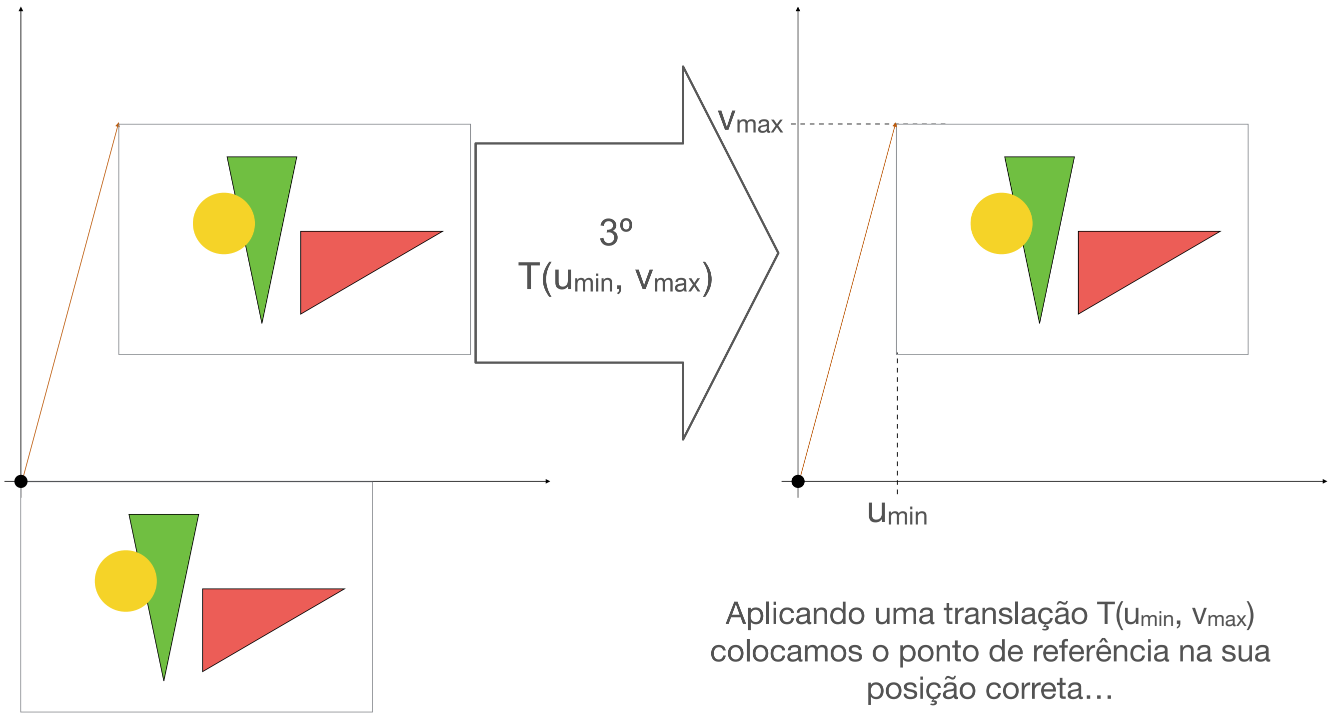
Enquadramento Janela-Visor (Caso B)



O ponto de referência está na origem, mas a sua posição final deveria ser (u_{min}, v_{max})…

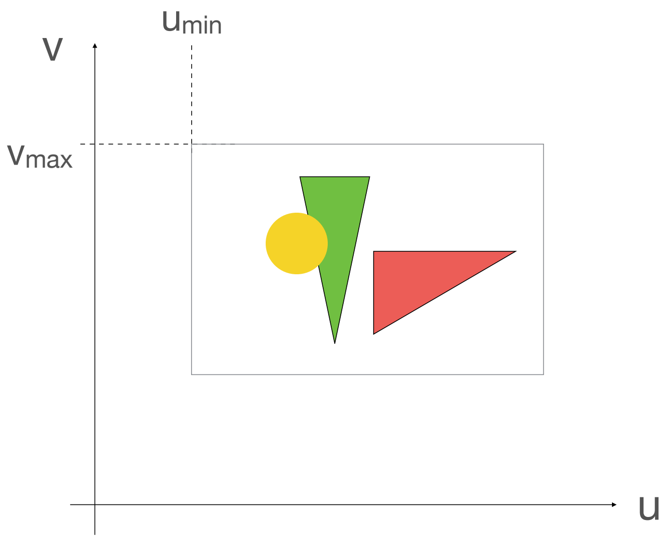
Aplicando uma translação T(u_{min}, v_{max}) colocamos o ponto de referência na sua posição correta…

Enquadramento Janela-Visor (Caso B)

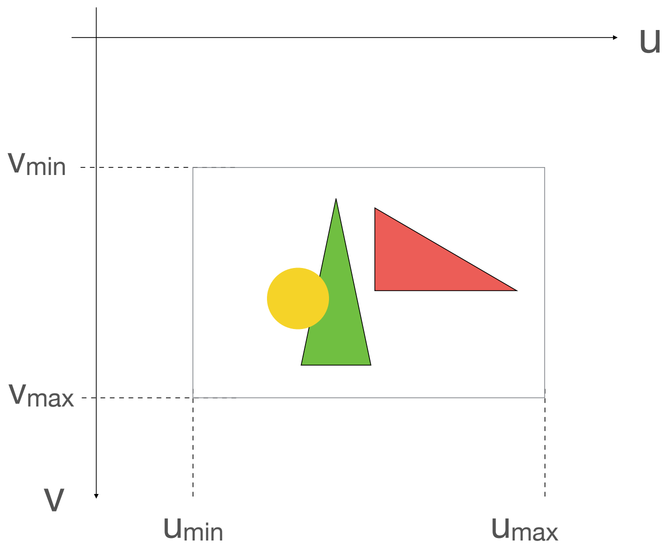


Aplicando uma translação $T(u_{min}, v_{max})$ colocamos o ponto de referência na sua posição correta…

Enquadramento Janela-Visor (Caso B)

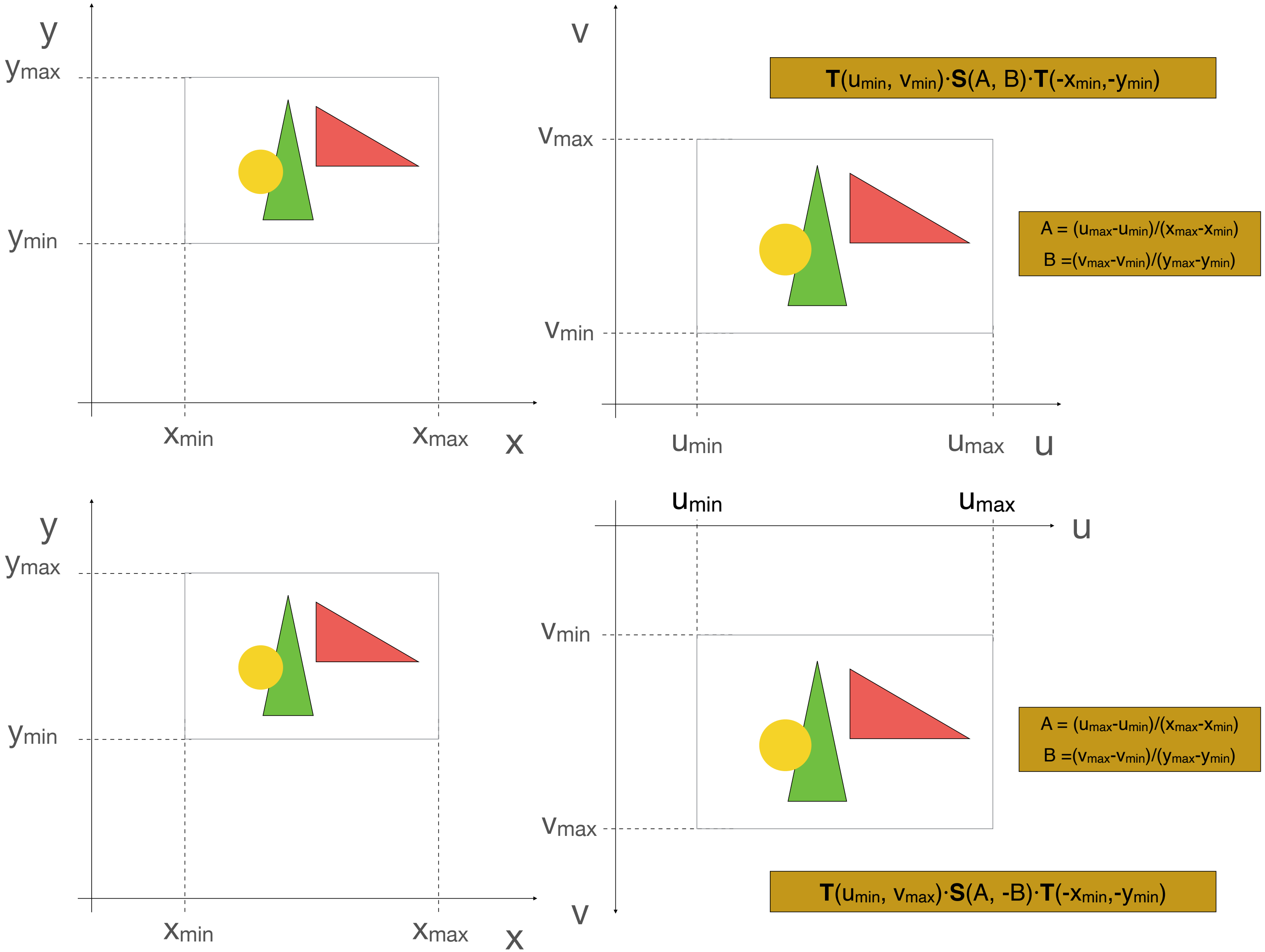


A nossa imagem está neste momento invertida…



Felizmente o dispositivo, como tem o eixo y invertido, irá colocar a imagem com a orientação correcta!

Resumo


y
y_{max}
y_{min}
x_{min}
x_{max}
x
v
v_{max}
v_{min}
u_{min}
u_{max}
u
$\mathbf{T}(u_{min}, v_{min}) \cdot \mathbf{S}(A, B) \cdot \mathbf{T}(-x_{min}, -y_{min})$
$A = (u_{max} - u_{min})/(x_{max} - x_{min})$
$B = (v_{max} - v_{min})/(y_{max} - y_{min})$
y
y_{max}
y_{min}
x_{min}
x_{max}
x
u_{min}
u_{max}
u
v_{min}
v_{max}
v
$A = (u_{max} - u_{min})/(x_{max} - x_{min})$
$B = (v_{max} - v_{min})/(y_{max} - y_{min})$
$\mathbf{T}(u_{min}, v_{max}) \cdot \mathbf{S}(A, -B) \cdot \mathbf{T}(-x_{min}, -y_{min})$

Conclusão

- No mapeamento Janela➔Visor pode usar-se qualquer ponto para referência, não necessitando ser forçosamente o canto inferior esquerdo da janela. Normalmente, o que dita o ponto que se escolhe, é o ponto para o qual, no visor, se conhecem bem as suas coordenadas finais.
- A sequência de transformações resulta numa matriz **M** (para usar na operação **M.p**, para transformar um ponto **p**), e é sempre do tipo **M**=**T.S.T**. No caso geral, de janelas e visores com orientação arbitrária, seria do tipo **M** = **T.R.S.R.T**

Exercício

O conteúdo duma janela, definida em WC pelos seus limites $40 \leq x \leq 60$ e $70 \leq y \leq 100$, deverá ser mapeado num ecrã dum telemóvel, ao alto, de dimensões 480x960 em DC, ocupando a maior área possível, encostado ao canto superior direito, sem deformação e sem recorte, e tendo o cuidado de não ocupar uma faixa com 160 pixels de altura na base desse mesmo ecrã. Como habitual, o referencial do ecrã tem a origem no canto superior esquerdo.

a) Indique, justificando, as dimensões do visor pretendido, bem como os limites do mesmo.

b) Especifique matematicamente o enquadramento Janela-Visor em causa, através duma matriz M (a usar na forma $P' = M.P$), deduzida e apresentada em termos duma composição natural de transformações geométricas elementares (S,R ou T) em 2D, com instanciação apropriada de todos os parâmetros (Nota: sempre que for o caso, indique, em parâmetro, os cálculos aritméticos necessários, mas sem os efetuar).

c) Suponha agora que se pretende ocupar todo o ecrã com o visor. Indique que modificações faria por forma a que os gráficos visualizados nas condições da alínea b) se mantivessem no mesmo local e com a mesma dimensão.