



UNIVERSIDAD TECNICA  
FEDERICO SANTA MARIA

DEPARTAMENTO  
DE INFORMÁTICA

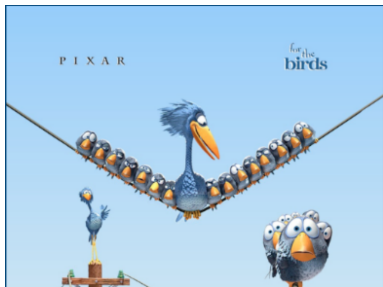
## Computación Gráfica – Viewing

Versión 220317

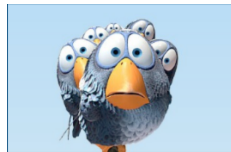


- Todo dispositivo de salida (pantalla) tienen su propio sistema de coordenadas (y dimensiones).
- Cada objeto tienen un sistema de coordenadas. Generalmente corresponde a las coordenadas del mundo (world).
- Por lo general, sólo una parte del objeto tiene interés en ser proyectada (window).
- ¿Cómo se despliega la información en cada tipo de pantalla?

# Viewing 2D

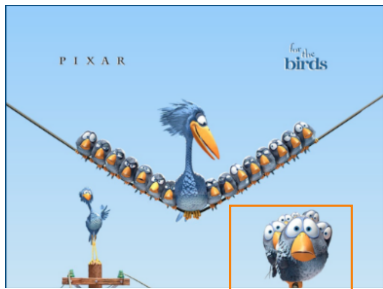


Objeto (mundo)



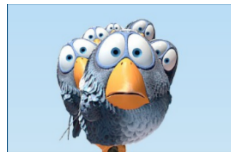
Dispositivo

¿Cómo realizar la proyección?



Objeto (mundo)

$$f_1(x, y)$$

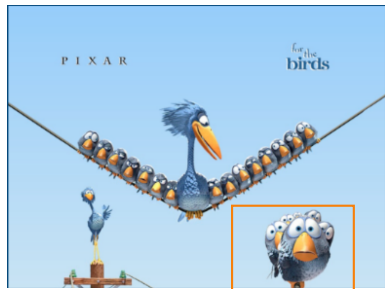


Dispositivo

## Solución

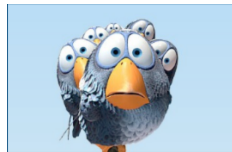
- Decidir que parte del objeto mostrar.
- Convertirla al sistemas de coordenadas del dispositivo.

# Viewing 2D



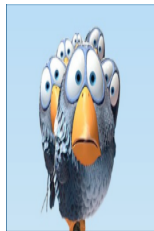
Objeto (mundo)

$$f_1(x, y)$$



Dispositivo 1

$$f_2(x, y)$$



Dispositivo 2

Problema de la solución: es dependiente del dispositivo de salida.

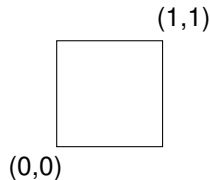
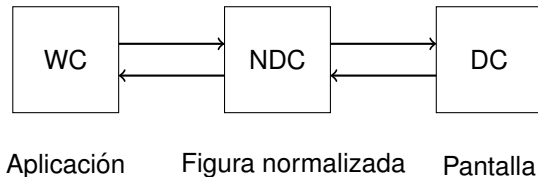


- Coordenadas del Dispositivo (DC): corresponden a las coordenadas del actual dispositivo y especifican en qué parte de la pantalla aparecerá el objeto
- Sistema de coordenadas universal (WC): Sistema cartesiano utilizado por las coordenadas del objeto. Las transformaciones antes vistas se basan en este sistema.

Para no depender del tipo de dispositivo, se utiliza un sistema de coordenadas normalizado (*NDC*).

*NDC*: Sistema de proyección unitario.

Una *window* en coordenadas *WC* es llevada a *NDC* y luego a *DC*. Permite independizar la aplicación del dispositivo.



# Viewing 2D



Objeto (mundo)

$$f_n(x, y)$$

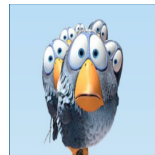
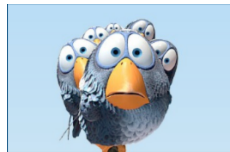


Imagen normalizada

$$f_d(x, y)$$

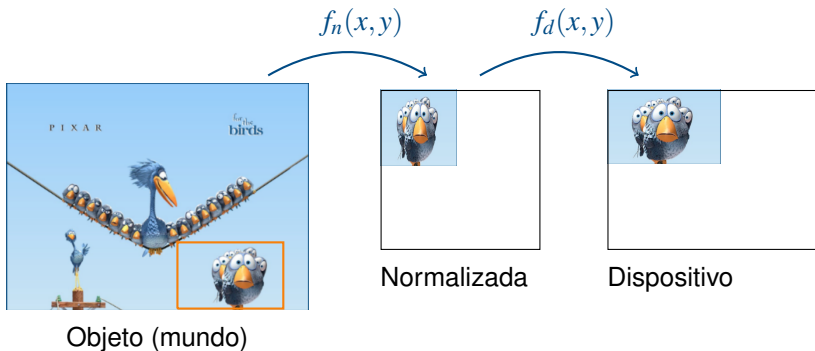


Dispositivo



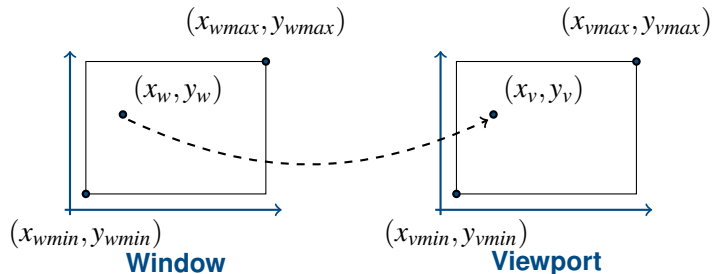
**Window:** zona de interés del *WC*

**Viewport:** zona del *NDC* que especifica dónde será proyectada la *window*.



# Viewing 2D

## Pasar un punto de la Window al Viewport



Usando proporsionalidad:

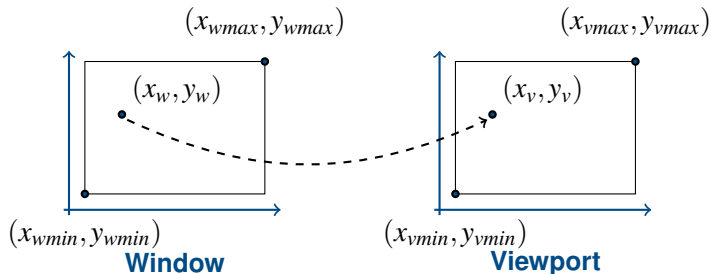
$$\frac{x_w - x_{wmin}}{x_{wmax} - x_{wmin}} = \frac{x_v - x_{vmin}}{x_{vmax} - x_{vmin}}$$

Despejando:

$$x_v = (x_w - x_{wmin}) \left( \frac{x_{vmax} - x_{vmin}}{x_{wmax} - x_{wmin}} \right) + x_{vmin}$$

# Viewing 2D

## Pasar un punto de la Window al Viewport



Usando proporsionalidad:

$$\frac{x_w - x_{wmin}}{x_{wmax} - x_{wmin}} = \frac{x_v - x_{vmin}}{x_{vmax} - x_{vmin}}$$

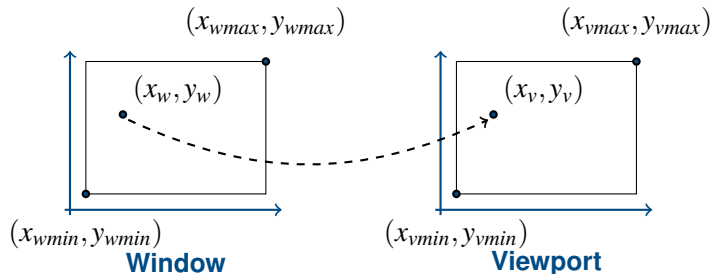
Despejando:

$$x_v = (x_w - x_{wmin}) \left( \frac{x_{vmax} - x_{vmin}}{x_{wmax} - x_{wmin}} \right) + x_{vmin}$$

Constante:  $S_x$

# Viewing 2D

## Pasar un punto de la Window al Viewport



Usando proporsionalidad:

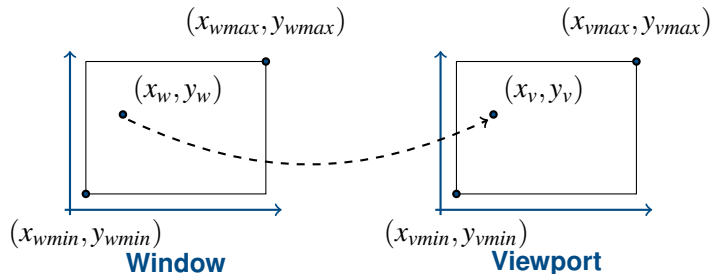
$$\frac{y_w - y_{wmin}}{y_{wmax} - y_{wmin}} = \frac{y_v - y_{vmin}}{y_{vmax} - y_{vmin}}$$

Despejando:

$$y_v = (y_w - y_{wmin}) \left( \frac{y_{vmax} - y_{vmin}}{y_{wmax} - y_{wmin}} \right) + y_{vmin}$$

# Viewing 2D

## Pasar un punto de la Window al Viewport



Usando proporsionalidad:

$$\frac{y_w - y_{wmin}}{y_{wmax} - y_{wmin}} = \frac{y_v - y_{vmin}}{y_{vmax} - y_{vmin}}$$

Despejando:

$$y_v = (y_w - y_{wmin}) \left( \frac{y_{vmax} - y_{vmin}}{y_{wmax} - y_{wmin}} \right) + y_{vmin}$$

Constante:  $S_y$



Se puede apreciar que:

- $S_x$  y  $S_y$  son valores constantes para todos los puntos mapeados.
- Si  $S_x \neq S_y$  se observarán distorsiones en la figura.
- Se define el Aspect Ratio para cada rectángulo como:

$$AR = \frac{x_{max} - x_{min}}{y_{max} - y_{min}}$$

Si el  $AR$  del *window* y *viewport* son iguales, entonces no hay distorsiones.

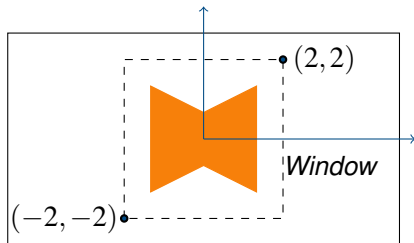
- Propuesto: Determine la matriz de transformación que permite llevar un punto  $p$  desde el *window* al *viewport*.

# Window a Viewport

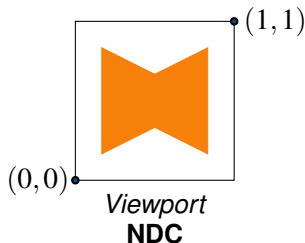
## Ejemplo 1



$$\text{Window}(WP_i, WP_f) \rightarrow \text{Viewport}(VP_i, VP_f)$$
$$\text{Window}((-2, -2), (2, 2)) \rightarrow \text{Viewport}((0, 0), (1, 1))$$



**Objeto (mundo)**



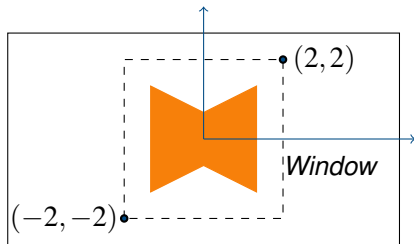
Se observa que no hay distorsión ya que *Window* y *Viewport* tienen el mismo *AR*. Además, el *Viewport* calza exacto sobre el **NDC**.

# Window a Viewport

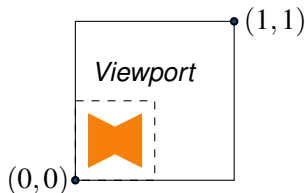
## Ejemplo 2



$$\text{Window}(WP_i, WP_f) \rightarrow \text{Viewport}(VP_i, VP_f)$$
$$\text{Window}((-2, -2), (2, 2)) \rightarrow \text{Viewport}((0, 0), (0, 5, 0, 5))$$



**Objeto (mundo)**



**NDC**

Se observa que no hay distorsión ya que *Window* y *Viewport* tienen el mismo *AR*. Sin embargo el *Viewport* ahora ocupa  $\frac{1}{4}$  del **NDC** (*zoom out*).

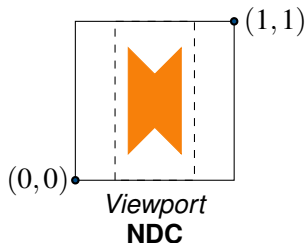
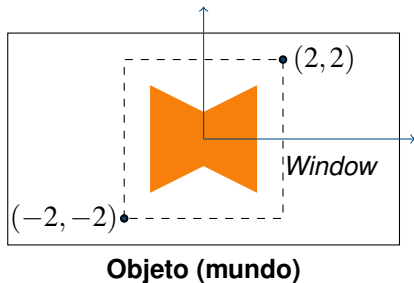


# Window a Viewport

## Ejemplo 3



$$\text{Window}(WP_i, WP_f) \rightarrow \text{Viewport}(VP_i, VP_f)$$
$$\text{Window}((-2, -2), (2, 2)) \rightarrow \text{Viewport}((0, 25, 0), (0, 75, 1))$$



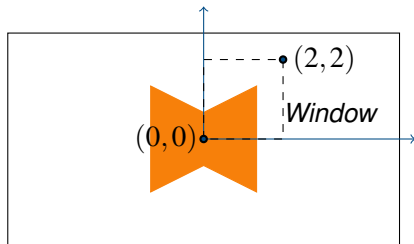
Se produce distorsión ya que *Viewport* ocupa solo el 50 % del ancho de *Window*. No tienen el mismo *AR*.

# Window a Viewport

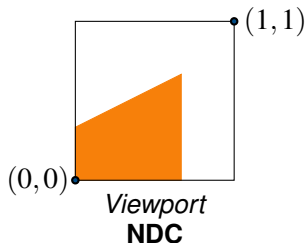
## Ejemplo 4



$$\text{Window}(WP_i, WP_f) \rightarrow \text{Viewport}(VP_i, VP_f)$$
$$\text{Window}((0,0), (2,2)) \rightarrow \text{Viewport}((0,0), (1,1))$$



**Objeto (mundo)**



No se produce distorsión ya que *Viewport* y *Window* tienen el mismo *AR*. Sin embargo se produce un efecto de *zoom in*. El *Viewport* y **NDC** son del mismo tamaño.