

Sistemas de referencia

Introducción a la Computación Gráfica

Andrea Rueda

Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas

- **The implementation principle:**

If you understand a mathematical process well enough, you can write a program that executes it.

J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes. Computer graphics: principles and practice, 3rd edition. Addison-Wesley, 2013.

Quiz diagnóstico

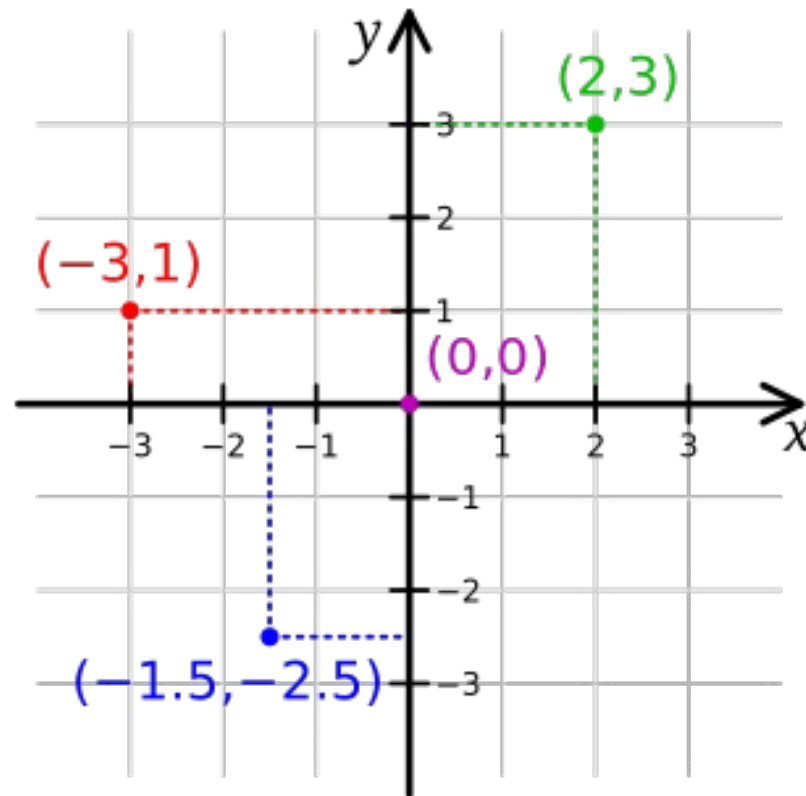
- ¿Qué es una coordenada y cómo se expresa?
- ¿Qué es un sistema de referencia y qué relación tiene con las coordenadas?
- Siendo $\mathbf{A} = [1 \ 2 \ 3]^T$ y $\mathbf{B} = [0 \ 1 \ 1]$, calcular \mathbf{AB} y \mathbf{BA}
- ¿La definición $\|\mathbf{v}\| \|\mathbf{w}\| |\sin \theta|$, a qué concepto corresponde y qué representa?
- ¿Cuál es la ecuación de una línea?

Recordando...

conceptos de álgebra lineal
y geometría

Coordenadas

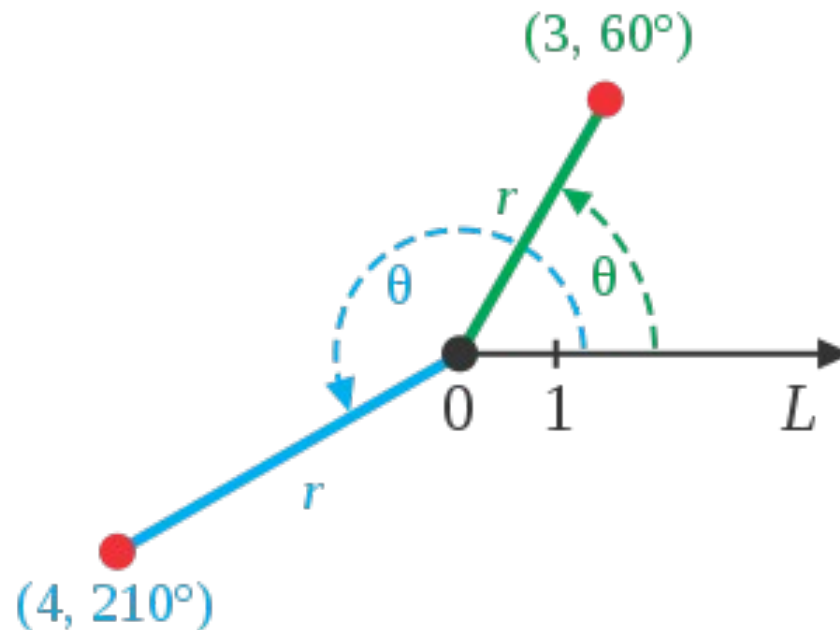
- Plano cartesiano



- Cada punto se define con una dupla de números reales \rightarrow coordenada.

Coordenadas

- Coordenadas cartesianas:
desplazamiento en los ejes de coordenadas (x, y).
- Coordenadas polares:
distancia radial al centro de coordenadas y
desplazamiento angular sobre el eje horizontal.



Coordenadas

- Polares a cartesianas:

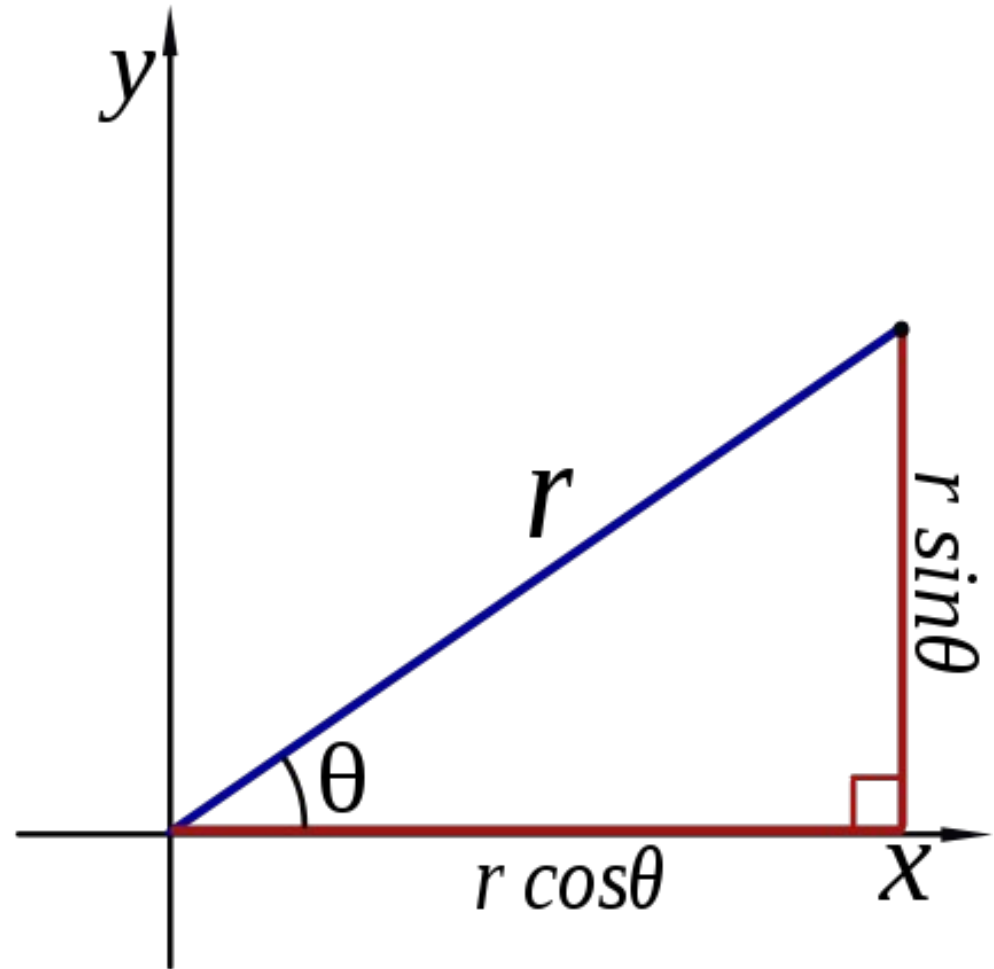
$$x = r \cos \theta$$

$$y = r \sin \theta$$

- Cartesianas a polares:

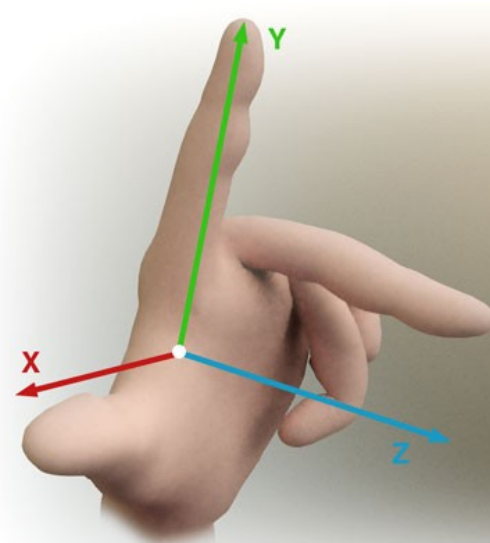
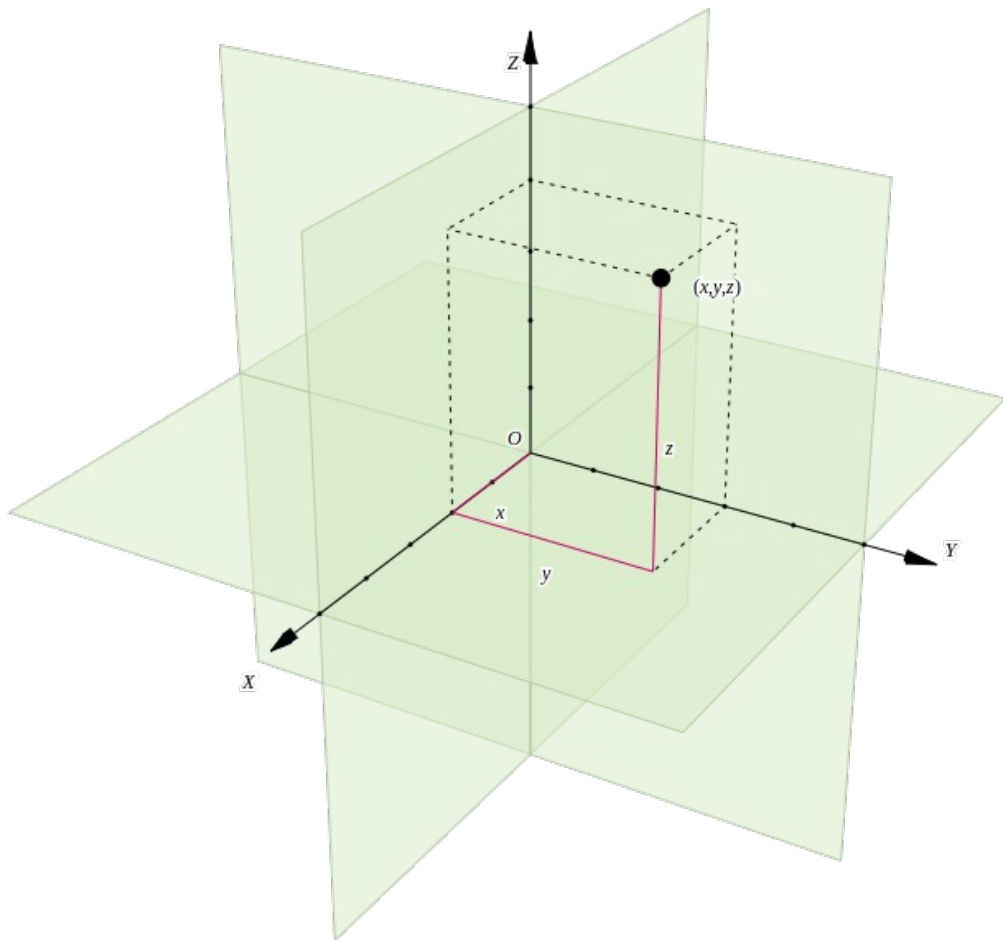
$$r = \sqrt{x^2 + y^2}$$

$$\theta = \tan^{-1} \left(\frac{y}{x} \right)$$

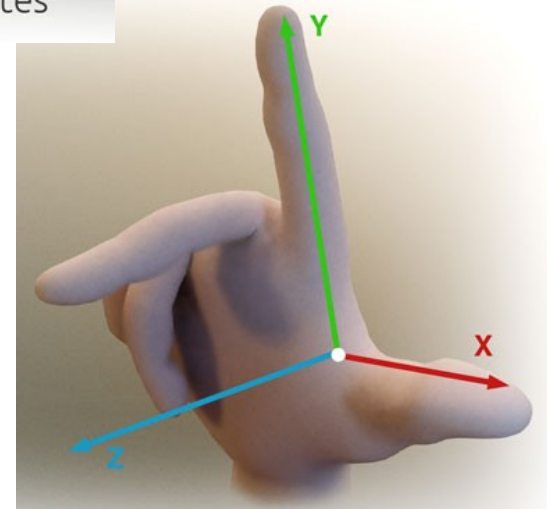


Coordenadas

- Coordenadas 3D:



Left Handed Coordinates



Right Handed Coordinates

Coordenadas

- Coordenadas cartesianas:
desplazamiento en los ejes de coordenadas (x, y, z) .
- Coordenadas cilíndricas:
distancia radial al centro de coordenadas,
desplazamiento angular en el plano x - y ,
desplazamiento en el eje z .
- Coordenadas esféricas (polares en 3D):
distancia radial al centro de coordenadas,
desplazamiento angular en el plano x - y ,
desplazamiento angular desde el eje z .

Coordenadas

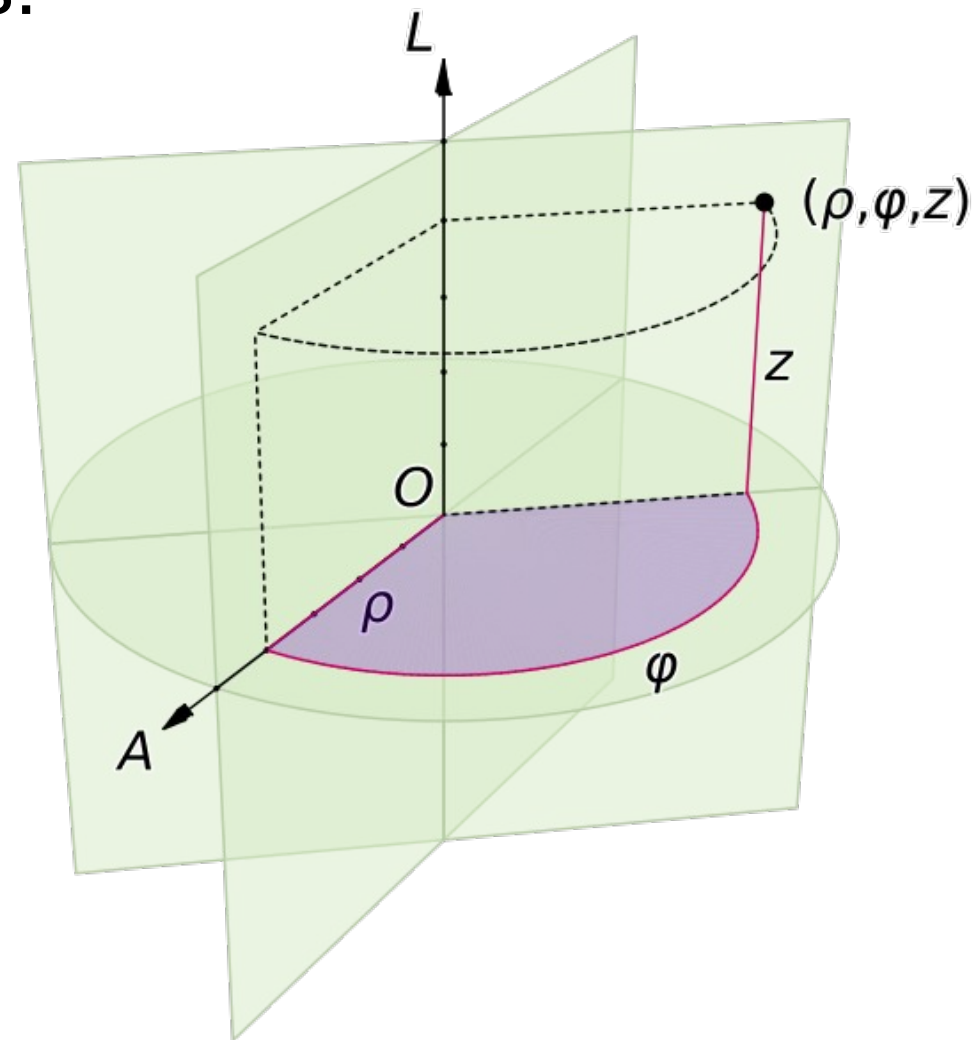
- Cilíndricas a cartesianas:

$$x = \rho \cos \phi$$

$$y = \rho \sin \phi$$

$$z = z$$

- Esféricas a cartesianas



Coordenadas

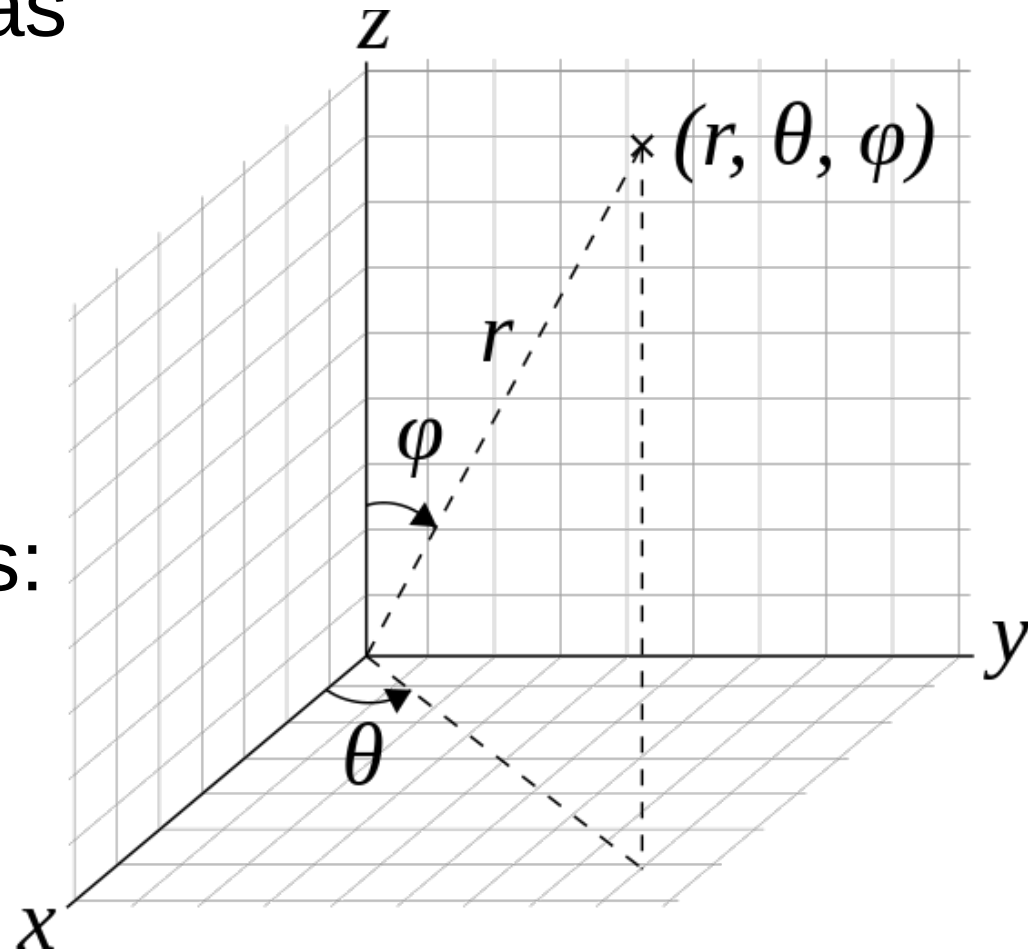
- Cilíndricas a cartesianas

- Esféricas a cartesianas:

$$x = r \cos \theta \sin \phi$$

$$y = r \sin \theta \sin \phi$$

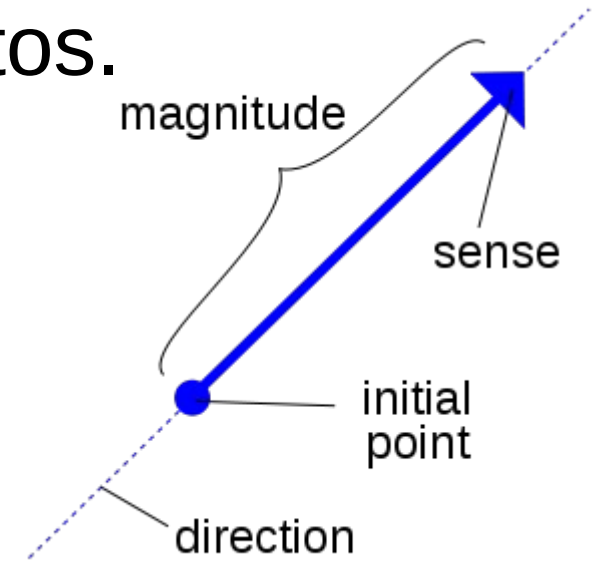
$$z = r \cos \phi$$



Vectores

- Vector: diferencia entre dos puntos.

$$\begin{aligned}\mathbf{V} &= \mathbf{P}_2 - \mathbf{P}_1 \\ &= (x_2 - x_1, y_2 - y_1) \\ &= (V_x, V_y)\end{aligned}$$



en.wikipedia.org/wiki/File:Vector_by_Zureks.svg

- Segmento de línea dirigido, con magnitud (módulo) y dirección.

$$|\mathbf{V}| = \sqrt{V_x^2 + V_y^2}$$

$$\alpha = \tan^{-1}\left(\frac{V_y}{V_x}\right)$$

Vectores

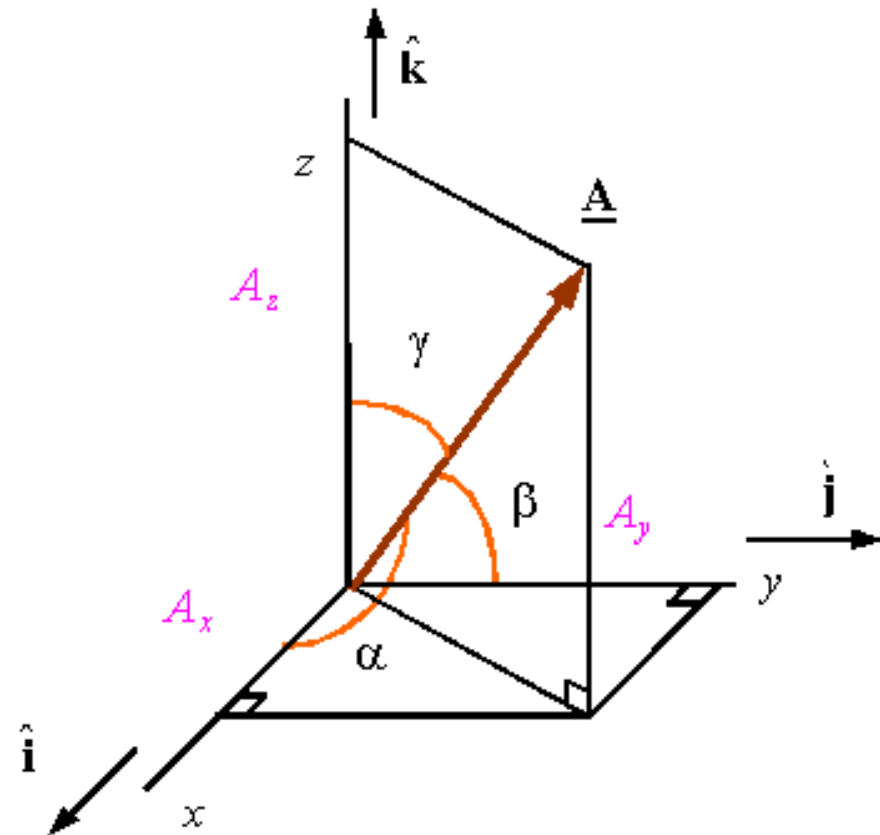
- Vector cartesiano tridimensional:

$$\mathbf{V} = (V_x, V_y, V_z)$$

$$|\mathbf{V}| = \sqrt{V_x^2 + V_y^2 + V_z^2}$$

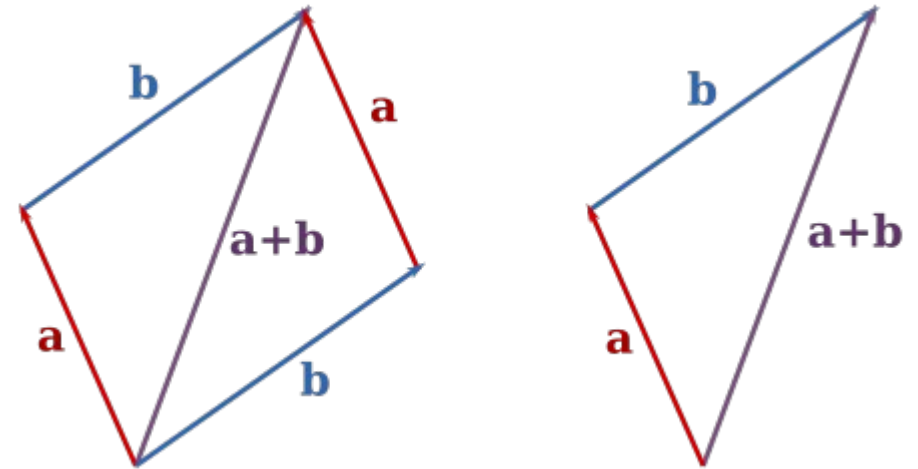
$$\cos \alpha = \frac{V_x}{|\mathbf{V}|} \quad \cos \beta = \frac{V_y}{|\mathbf{V}|}$$

$$\cos \gamma = \frac{V_z}{|\mathbf{V}|}$$



Vectores

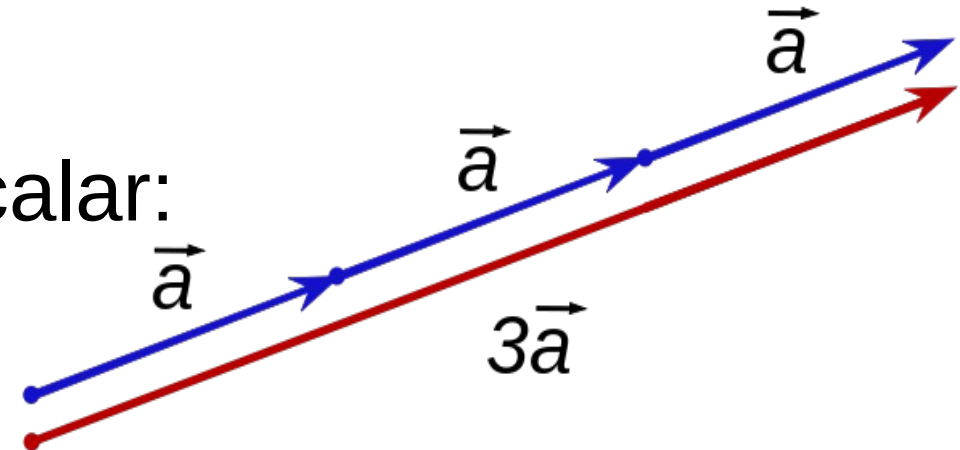
- Suma de vectores:



$$\mathbf{V}_1 + \mathbf{V}_2 = (V_{1x} + V_{2x}, V_{1y} + V_{2y}, V_{1z} + V_{2z})$$

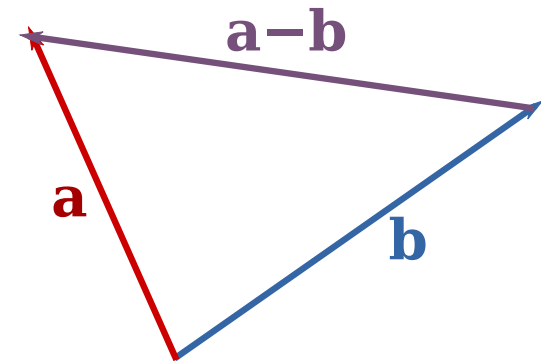
- Multiplicación por un escalar:

$$s \mathbf{V} = (s V_x, s V_y, s V_z)$$



Vectores

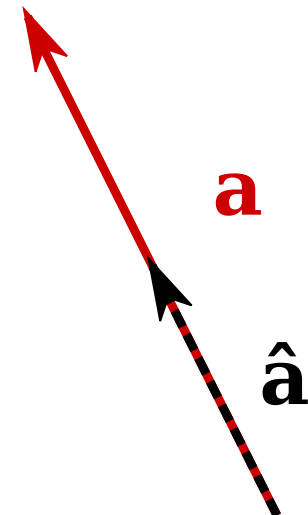
- Resta de vectores:



$$\mathbf{V}_1 - \mathbf{V}_2 = (V_{1x} - V_{2x}, V_{1y} - V_{2y}, V_{1z} - V_{2z})$$

- Normalización → vector unitario:

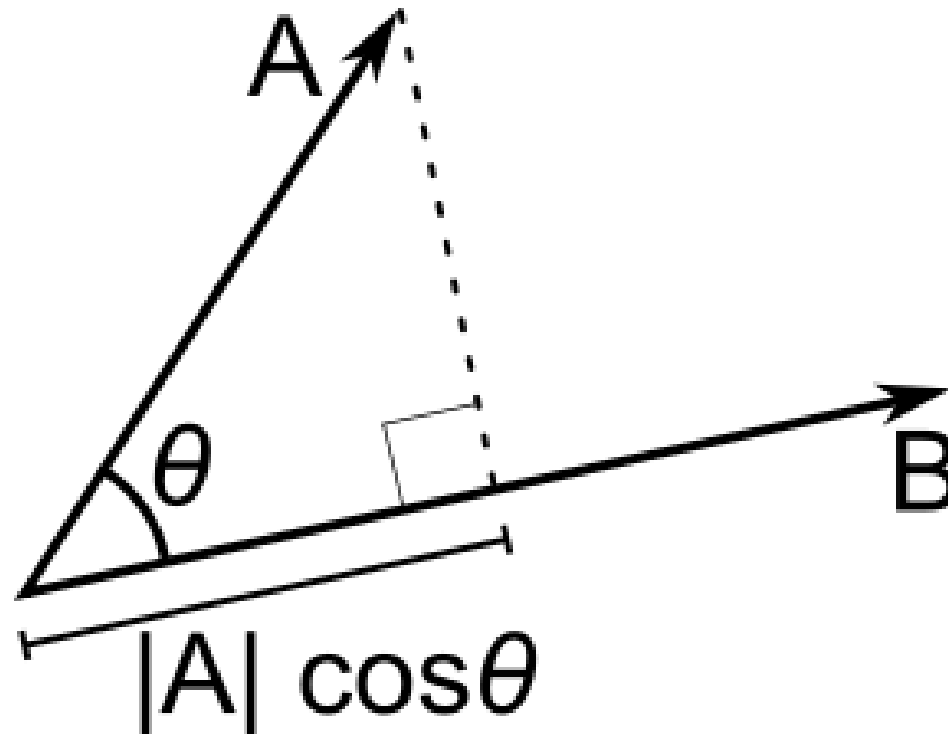
$$\hat{\mathbf{V}} = \frac{\mathbf{V}}{|\mathbf{V}|} = \frac{V_1}{|\mathbf{V}|} + \frac{V_2}{|\mathbf{V}|} + \frac{V_3}{|\mathbf{V}|}$$



Vectores

- Producto escalar (punto, interno):

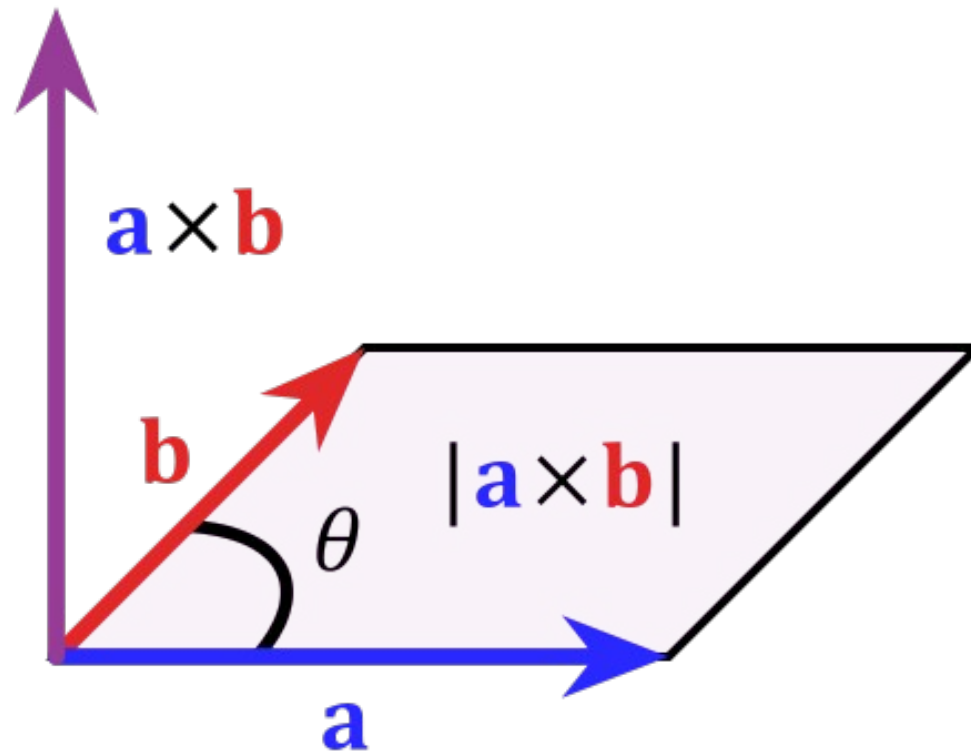
$$\mathbf{V}_1 \cdot \mathbf{V}_2 = |\mathbf{V}_1| |\mathbf{V}_2| \cos \theta, \quad 0 \leq \theta \leq \pi$$



Vectores

- Producto vectorial (cruz):

$$\mathbf{V}_1 \times \mathbf{V}_2 = \mathbf{u} \|\mathbf{V}_1\| \|\mathbf{V}_2\| \sin \theta, \quad 0 \leq \theta \leq \pi$$



Vectores base

- Vectores de eje: $\vec{u}_k, \quad k = 1, 2, \dots, n$
- Vector de posición: $\vec{r} = x \vec{u}_x + y \vec{u}_y + z \vec{u}_z$
- Vectores base unitarios: $u_k = \frac{\vec{u}_k}{|\vec{u}_k|}$
- Vectores base mutuamente perpendiculares \rightarrow base ortogonal.
- Vectores base unitarios y perpendiculares \rightarrow base ortonormal.

Matrices

- Disposición rectangular de magnitudes, organizadas en filas y columnas.
- Colección de vectores fila, o colección de vectores columna.
- Como convención, un vector suele representarse como una matriz columna.

$$\mathbf{V} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

Matrices

- Suma de matrices: **A** y **B**, tamaño $n \times m$

$$(\mathbf{A+B})_{i,j} = \mathbf{A}_{i,j} + \mathbf{B}_{i,j}, \quad 1 \leq i \leq n ; 1 \leq j \leq m$$

- Multiplicación por un escalar: **A**, tamaño $n \times m$

$$(c\mathbf{A})_{i,j} = c \cdot \mathbf{A}_{i,j}, \quad 1 \leq i \leq n ; 1 \leq j \leq m$$

Matrices

- Transpuesta de una matriz: **A**, tamaño $n \times m$

$$(\mathbf{A}^T)_{i,j} = \mathbf{A}_{j,i}, \quad 1 \leq i \leq n ; 1 \leq j \leq m$$

- Multiplicación de matrices: **A**, tamaño $n \times m$, **B**, tamaño $m \times p$

$$(\mathbf{A} \mathbf{B})_{i,j} = \sum_{k=1}^m \mathbf{A}_{i,k} \mathbf{B}_{k,j}, \quad 1 \leq i \leq n; 1 \leq j \leq p; 1 \leq k \leq m$$

Sistemas de referencia

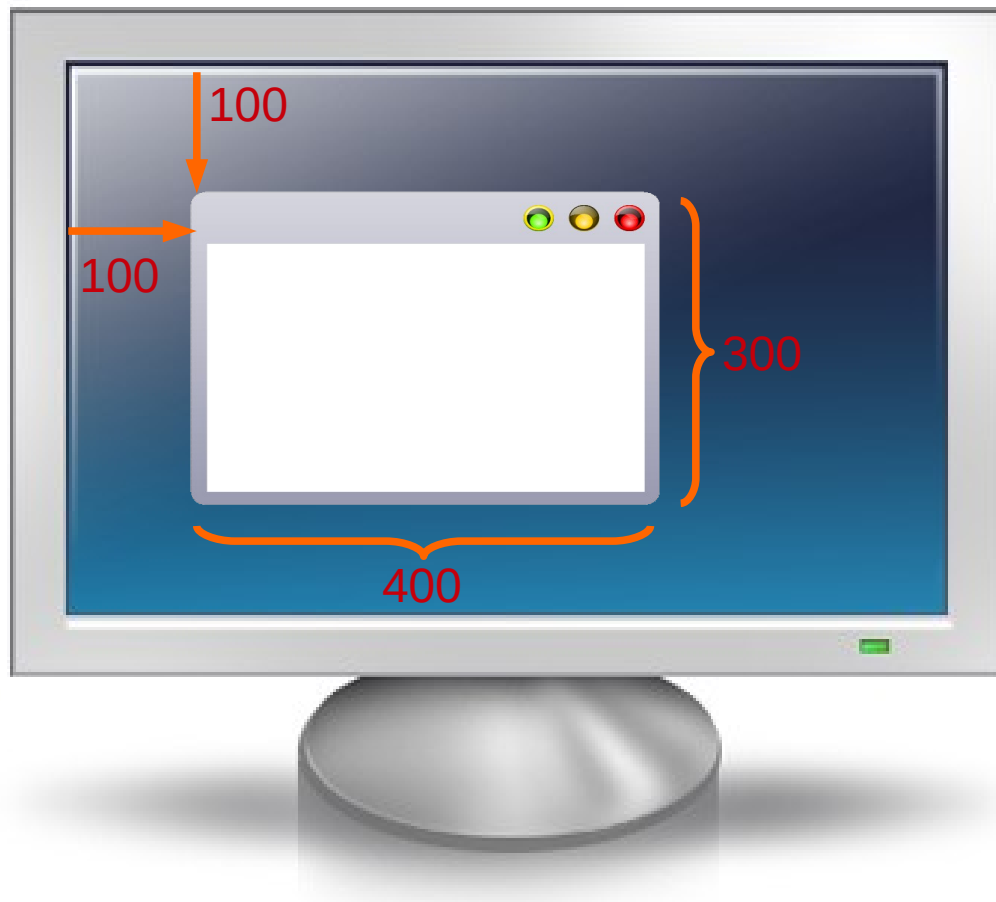
Sistemas de referencia

- Sistema de referencia de la pantalla:
 - Origen (0,0): punto superior-izquierdo de la pantalla. Coordenada x se incrementa hacia la derecha. Coordenada y se incrementa hacia abajo.



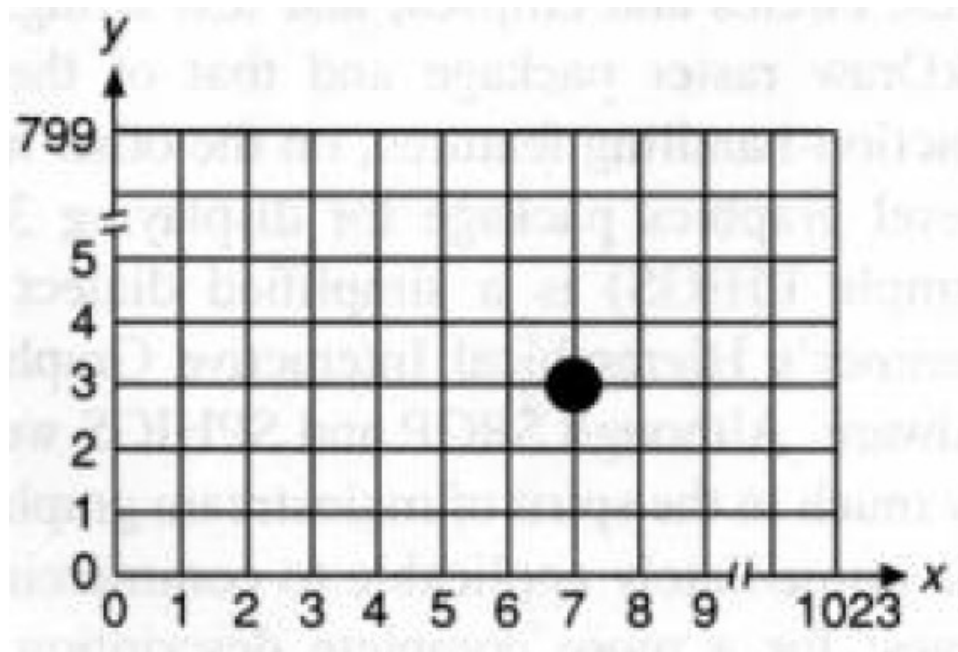
Sistemas de referencia

- Sistema de referencia de la pantalla:
 - Gestión de la ventana de visualización:



Sistemas de referencia

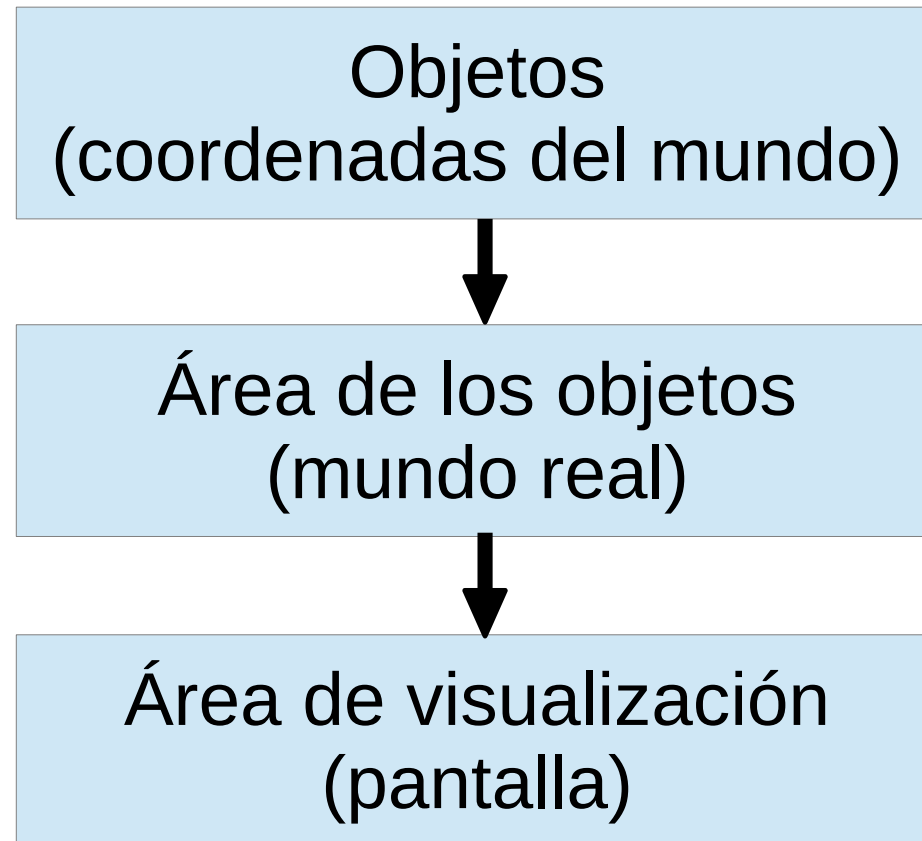
- Sistema de coordenadas cartesianas:
 - Origen (0,0): punto inferior-izquierdo de la ventana.
Coordenada x se incrementa hacia la derecha.
Coordenada y se incrementa hacia arriba.



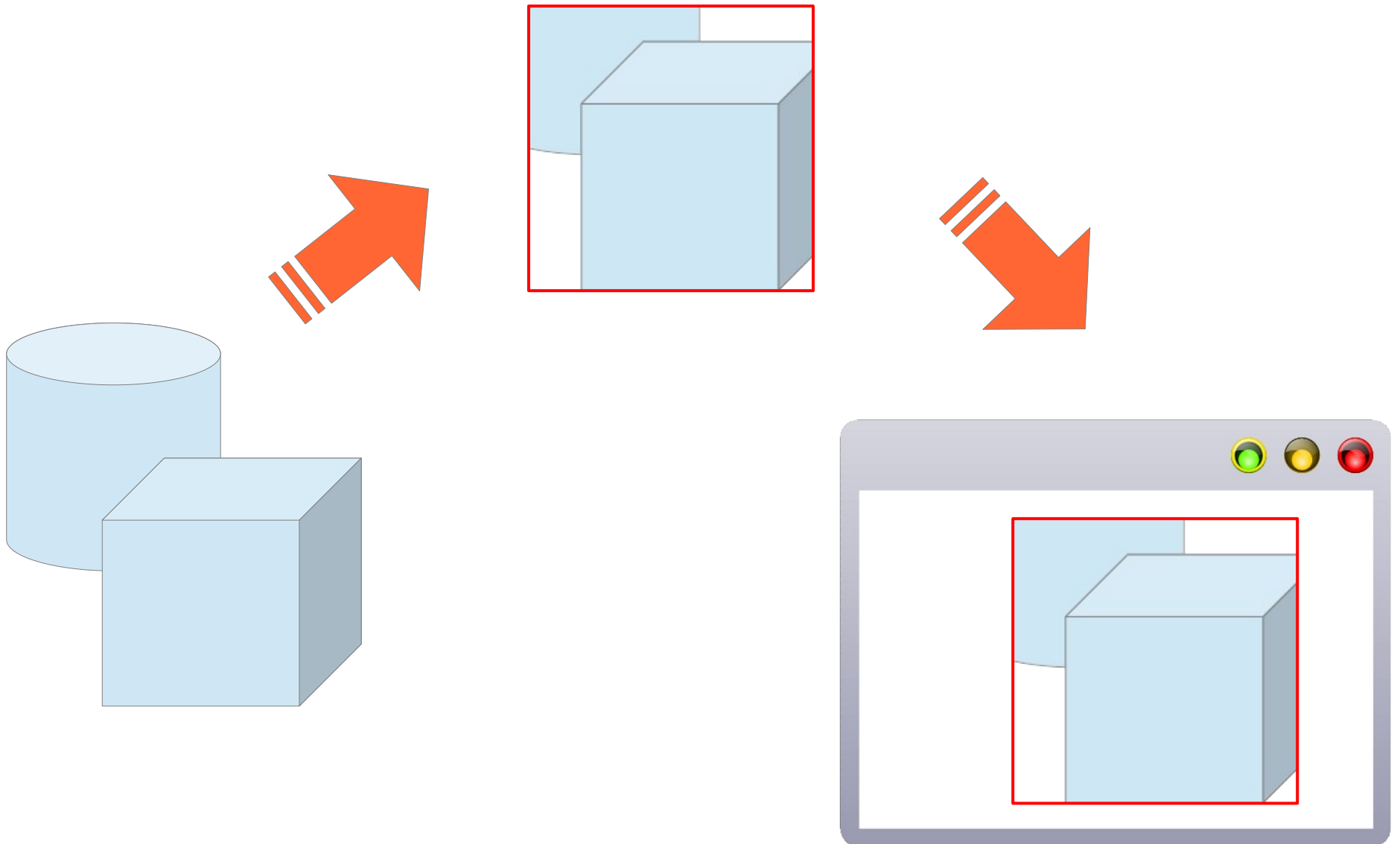
Sistemas de referencia

- Sistema de coordenadas de la pantalla:
Definido de forma relativa a la ventana.
- Desventajas:
 - Tamaño de la ventana puede variar.
 - Especificación de los objetos en sus tamaños originales (relación de aspecto).

Sistemas de referencia

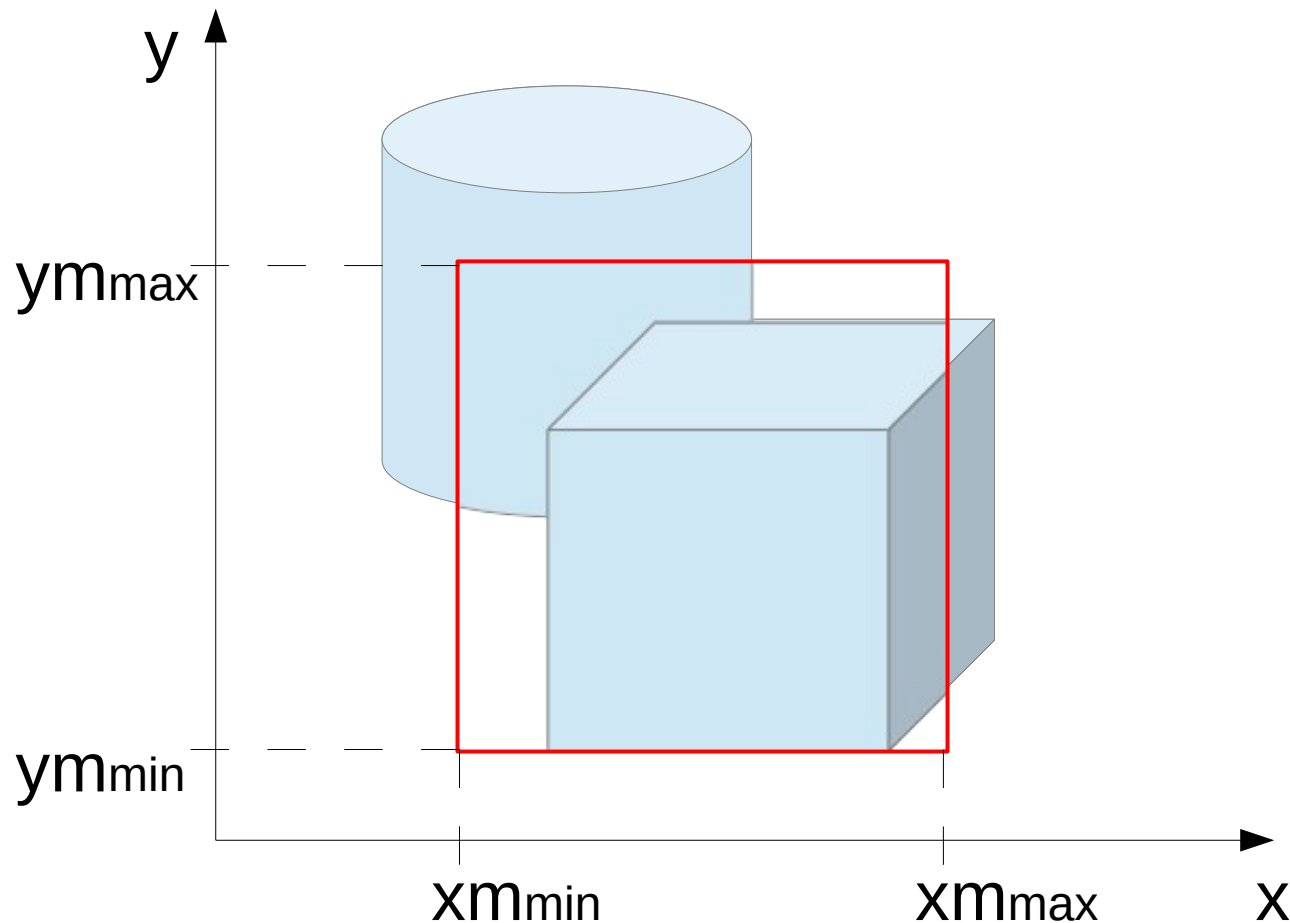


Sistemas de referencia



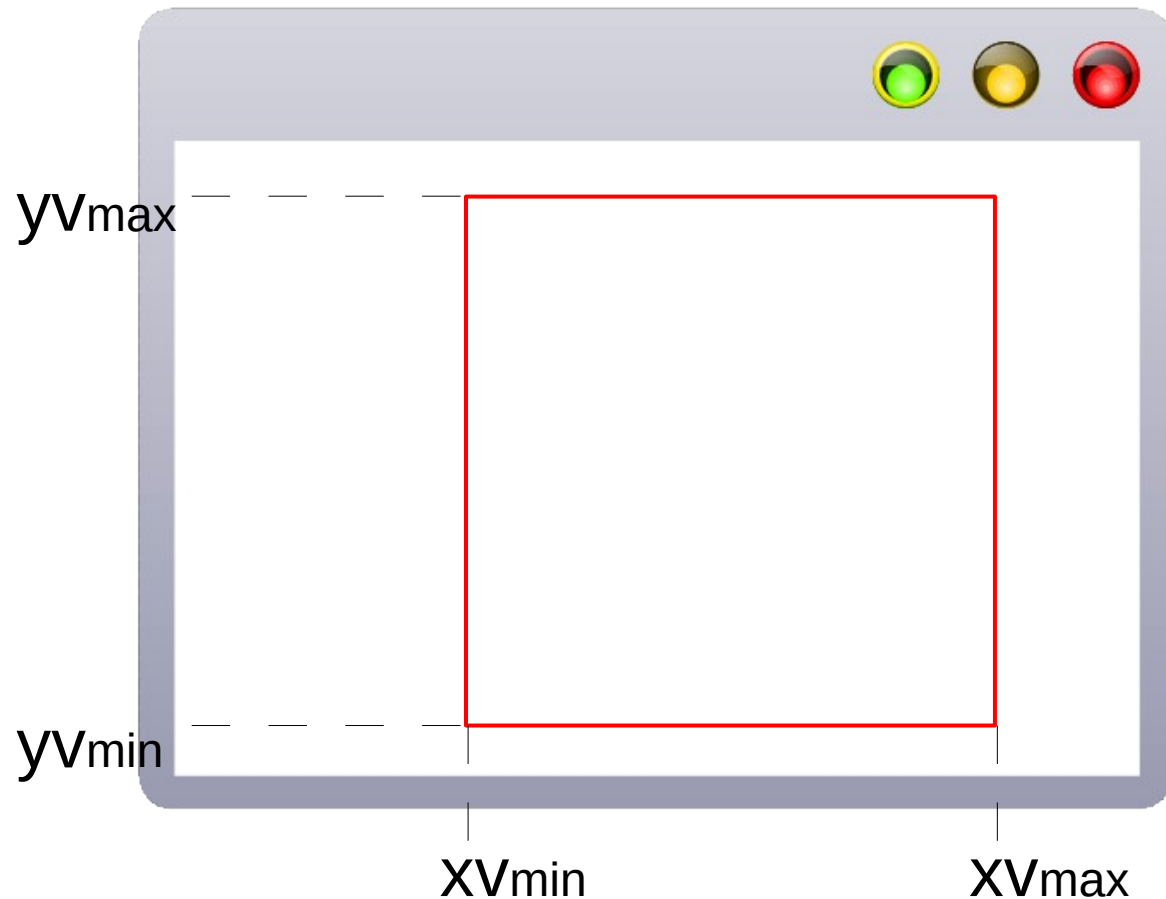
Sistemas de referencia

- Área de los objetos (ventana del mundo real):



Sistemas de referencia

- Área de visualización (en pantalla):

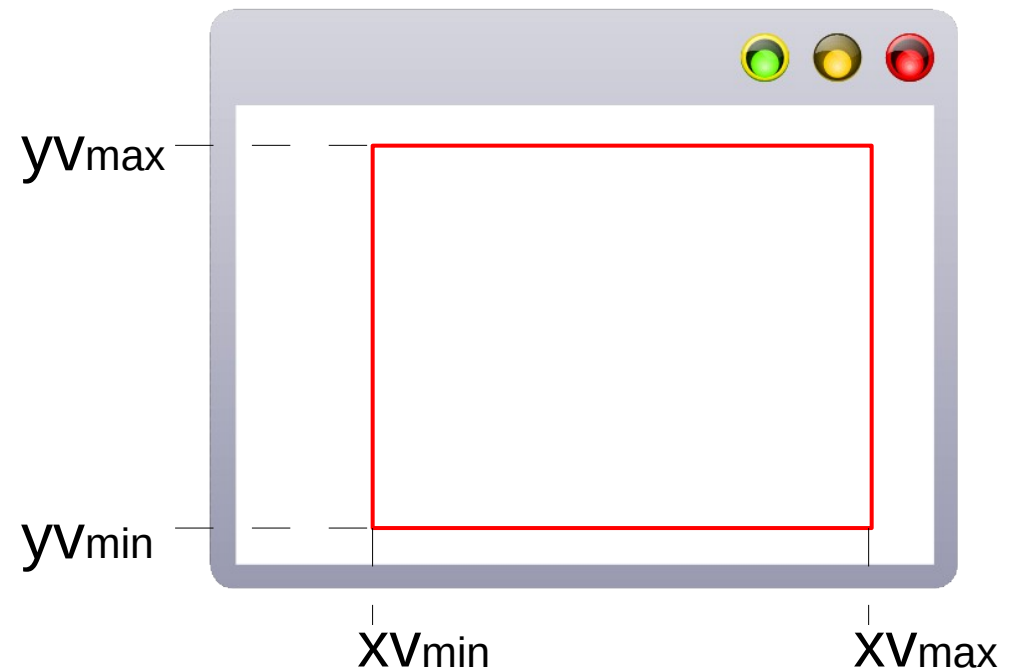
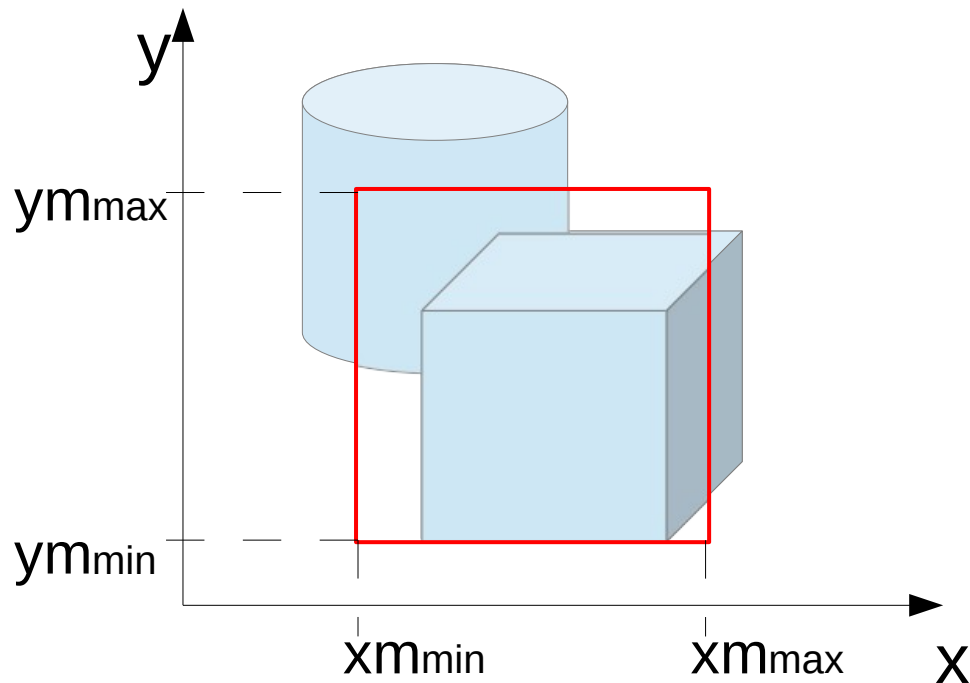


Sistemas de referencia

- En el procedimiento de inicialización, el área de visualización se fija al mismo tamaño de la ventana.
- Al cambiar el tamaño de la ventana (procedimiento de actualización), ¿qué pasa con el tamaño del área de visualización?

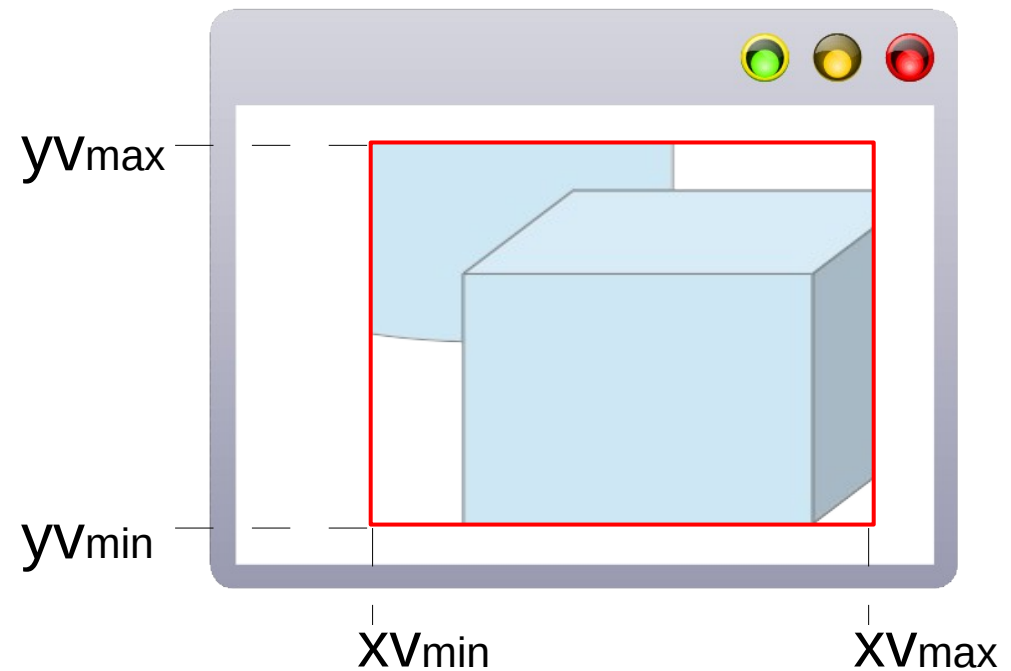
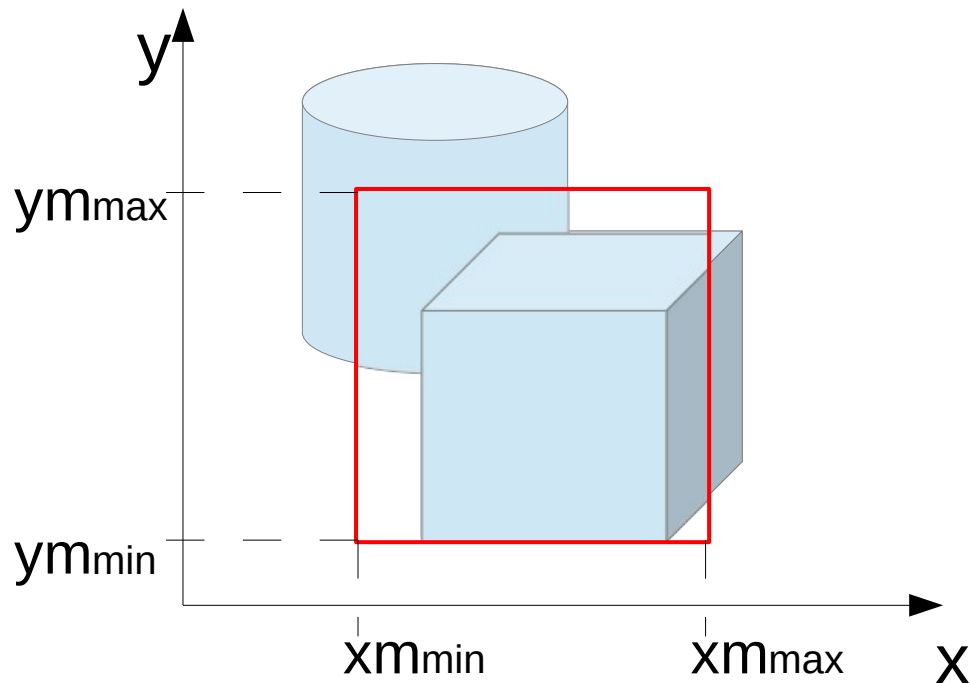
Sistemas de referencia

- Diferencias en las proporciones, debido a los tamaños independientes:



Sistemas de referencia

- Diferencias en las proporciones, debido a los tamaños independientes:



Sistemas de referencia

- En el procedimiento de inicialización, el área de visualización se fija al mismo tamaño de la ventana.
- Al cambiar el tamaño de la ventana (procedimiento de actualización), ACTUALIZAR el tamaño del área de visualización para mantener la relación de aspecto.

Sistemas de referencia

- Coordenadas absolutas:

Posiciones reales dentro del sistema de coordenadas.

- Coordenadas relativas:

Desplazamiento a partir de la última posición a la que se ha hecho referencia (posición actual).

posición actual: (3,8)

nueva coordenada: (2, -1) \rightarrow (5,7)

Introducción a OpenGL

Software gráfico

- Sistemas de propósito específico:

- Photoshop (Adobe).
- AutoCAD (Autodesk).
- SolidEdge (Siemens).



- Paquetes de programación general:

- GL (Graphics Library), OpenGL (Silicon Graphics).
- VRML (Virtual-Reality Modelling Language).
- Java 2D y 3D.

GL

Graphics Library (Silicon Graphics):

- Uno de los primeros paquetes gráficos, muy popular debido a su uso en las estaciones gráficas de Silicon Graphics.
- Su creciente popularidad y extensión a otros sistemas de hardware, generó el desarrollo y liberación de OpenGL.

OpenGL



- Interfaz de programación de aplicaciones (API) que provee un amplio rango de funciones de renderizado, mapeo de texturas, efectos especiales y visualización de elementos 2D y 3D.
- Desarrollado y actualizado por la organización OpenGL Architecture Review Board.

OpenGL

- Independiente del sistema operativo:
Funcional en Windows, Linux, Mac, ...
- Independiente del lenguaje de programación:
API implementada mayoritariamente en hardware.
Correspondencia de lenguajes en C, C++, Java, ...

OpenGL

- Librerías asociadas:
 - GLU (OpenGL Utility Library):
subrutinas de visualización y proyección adicionales, incluida en todas las implementaciones de OpenGL.
 - GLUT (OpenGL Utility Toolkit):
añade la funcionalidad para el manejo de ventanas de visualización (independientes del dispositivo).

OpenGL

- Dos aproximaciones:
 - *Pipeline* fija (OpenGL v 1.x):
transformaciones y caracterizaciones de los pixeles se encuentran definidos en el hardware (GPU) y no se pueden modificar.
 - *Pipeline* programable (OpenGL v 2.0 en adelante):
las etapas para visualización se pueden modificar por el usuario a través de “shaders”, permitiendo mayor flexibilidad.

OpenGL

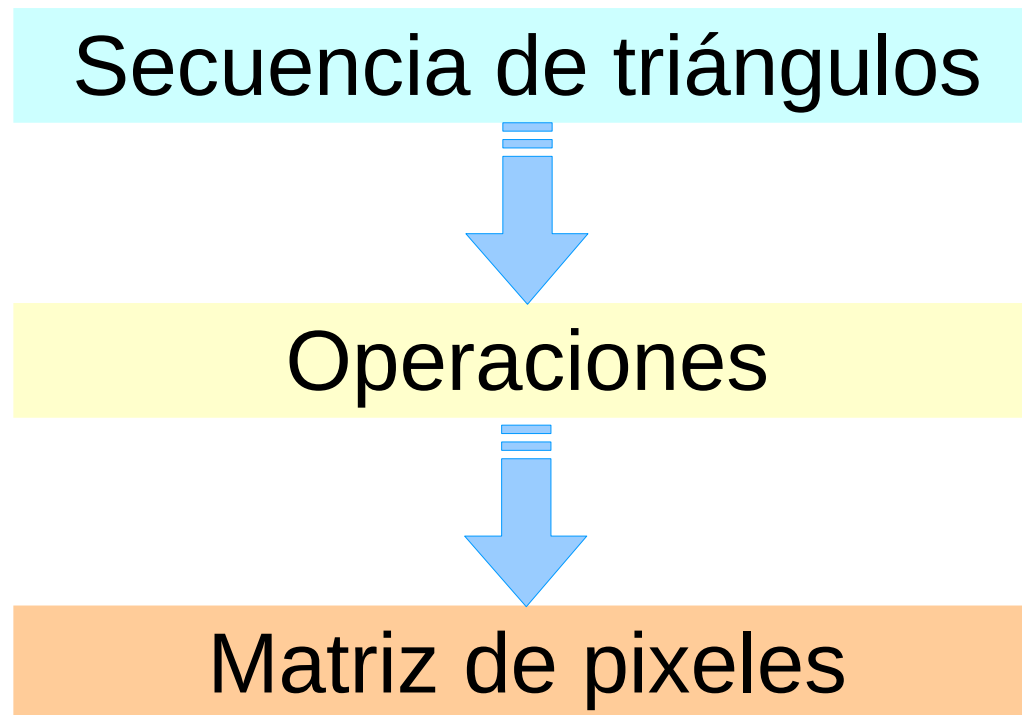
- Las nuevas versiones siempre son para mejorar, sin embargo...
 - La *pipeline* programable hace que las tareas difíciles se puedan programar fácilmente, pero algunas tareas simples ahora resultan más complejas de programar.
 - La *pipeline* programable es más robusta, pero es menos intuitiva.
- Como no estamos en un curso de sólo OpenGL, utilizaremos la *pipeline* fija.

Pipeline de visualización

- Elemento básico de visualización: pixel (*picture element*).
- Imagen: arreglo bidimensional (matriz) de pixeles.
- Visualización 3D: aplicar efectos a los pixeles para crear la sensación de tridimensionalidad.
- Renderización (*rendering*): convertir una escena 3D en una imagen 2D.
- Rasterización (*rasterization*): proceso de renderización a partir de triángulos.

Pipeline de visualización

- Proceso de rasterización:



- Triángulos: definidos por tres vértices, coordenadas (X,Y,Z).

Pipeline de visualización

- Proceso de rasterización (simplificación):
 1. Transformación al espacio de visualización (*clip space*).
 2. Normalización de coordenadas.
 3. Transformación a la ventana de visualización.
 4. Conversión a fragmentos.
 5. Procesamiento de fragmentos.
 6. Escritura de fragmentos.

Pipeline de visualización

1. Transformación al espacio de visualización:

- *Clip space*: volumen de visualización.
- *Clip coordinates*: coordenadas de visualización.

Coordenadas 3D espaciales: 3 componentes.

Coordenadas 3D de visualización: 4 componentes.

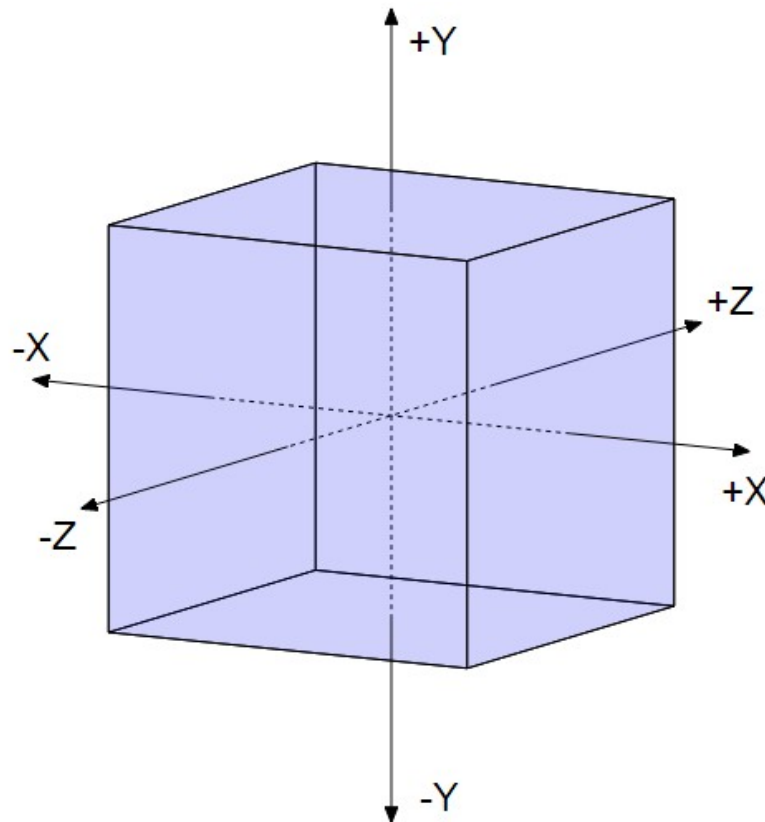
$(X, Y, Z, W) \rightarrow W$ indica pertenencia al volumen de visualización

- Triángulos con vértices fuera del volumen son divididos (*clipping*) en triángulos más pequeños.
- Triángulos fuera del volumen son descartados.

Pipeline de visualización

2. Normalización de coordenadas:

Se dividen las coordenadas X, Y, Z por W , para que el rango efectivo esté en $[-1, 1]$.



Pipeline de visualización

3.Transformación a la ventana de visualización:

X de izquierda a derecha.

Y de abajo a arriba.

Z de adelante hacia atrás.

ahora con respecto a la ventana en la que se está ejecutando OpenGL.

Pipeline de visualización

4. Conversión a fragmentos:

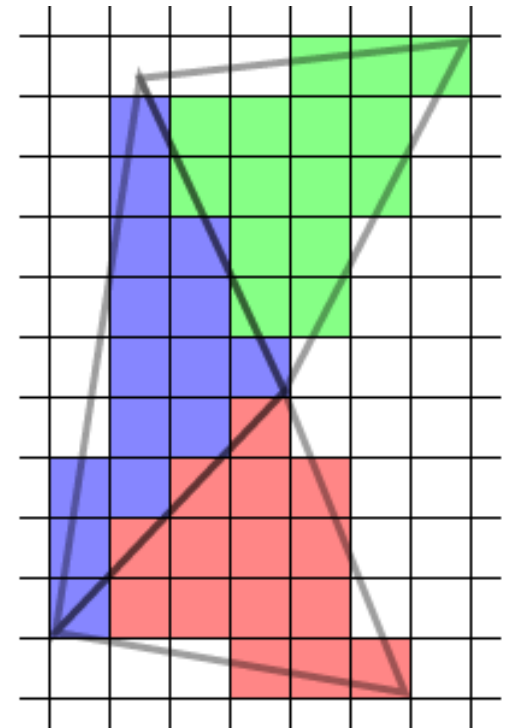
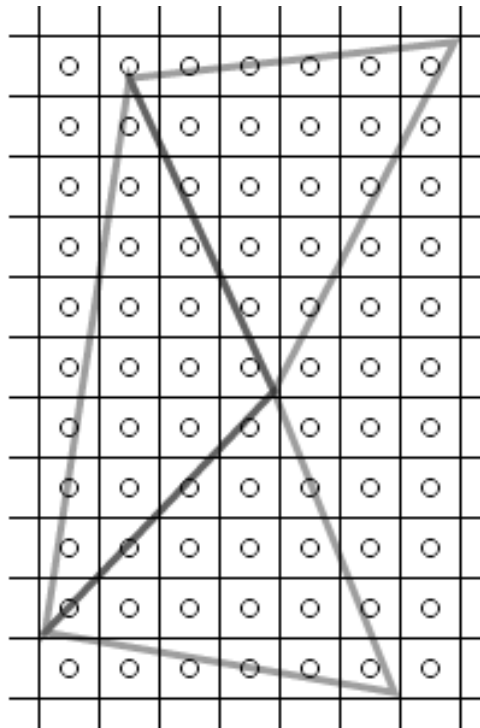
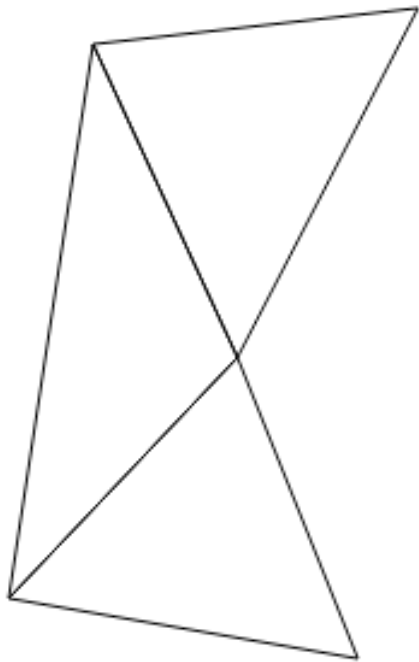
Cada triángulo se analiza en términos de los pixeles que cubre.

De esta forma, cada triángulo es una secuencia de fragmentos que lo cubren.

Se utiliza solamente la información de las coordenadas X,Y.

Pipeline de visualización

4. Conversión a fragmentos:



Pipeline de visualización

5. Procesamiento de fragmentos:

Por cada triángulo, los fragmentos se procesan para transformarlos en valores de color y profundidad.

Para evitar sobrelapamientos en la información entre triángulos, todos los fragmentos de un triángulo se procesan primero antes de pasar al siguiente triángulo.

Pipeline de visualización

6.Escritura de fragmentos:

Escribir cada fragmento a la imagen final (búfer), teniendo en cuenta la información de color y profundidad.

Referencias

- D. Hearn, M.P. Baker. Gráficos por computadora con OpenGL, 3a edición. Pearson Prentice Hall, 2006.
- J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes. Computer graphics: principles and practice, 2nd edition in C. Addison-Wesley, 1996.
- en.wikipedia.org/wiki/OpenGL
- www.opengl.org/about/
- www.arcsynthesis.org/gltut/
- www.cse.ohio-state.edu/~whmin/courses/cse581-2012-winter/2D.pdf
- www.sci.brooklyn.cuny.edu/~weiss/cisc3620/Lectures/02-Chapter02.html