

# Introducción al Blending

## Informática Gráfica I

Material de: **Ana Gil Luezas**  
Adaptado por: **Elena Gómez y Rubén Rubio**  
`{mariaelena.gomez,rubenrub}@ucm.es`



# Contenido

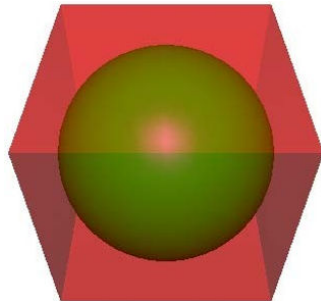
- 1 Blending
- 2 Depth Buffer y Depth Test
  - Depth Buffer
  - Depth Test
  - Blending
- 3 Frame Buffer
- 4 Stencil Buffer
- 5 Culling

# Colores RGBA

- El valor por defecto de la componente Alpha es 1 ( $\approx 255$ ):  
👉 **Un color RGB se completa con  $A = 1$  (opaco)**
- Se utiliza para modelar objetos traslúcidos:  
 $A = 1 \rightarrow$  opaco  
 $A = 0 \rightarrow$  transparente
- Los colores de objetos que se superponen en la vista se combinan para determinar el color final:  
👉 **Cuando un objeto traslúcido aparece delante de otro.**

# Colores RGBA

- **Blending**: suma ponderada de los colores correspondientes a distintos objetos.
- **Alpha blending**: los factores de ponderación se determinan en función de la componente alpha.



# Depth Buffer

- El **Depth buffer** o **Z-buffer** contiene la distancia con respecto a la cámara (al plano cercano) de cada píxel (componente Z del fragmento).
- Generalmente se almacena como un entero de 24 *bits*.
- Los valores están en el rango  $[0, 1]$ , siendo 0 el más cercano y 1 el más lejano.

# Depth Buffer

- Al realizarse la **proyección de los vértices**, se calcula la distancia relativa a la cámara y se guarda en la **componente Z** del vértice.
- Se realiza el **relleno de triángulos**: genera los fragmentos del interior mediante interpolación de los valores de los vértices. Datos de cada fragmento:
  - Coordenadas de ventana  $(x, y, z)$  (proyectadas y ajustadas al puerto de vista).
  - Color  $(r, g, b, a)$ .
  - Coordenadas de textura  $(s, t)$ .

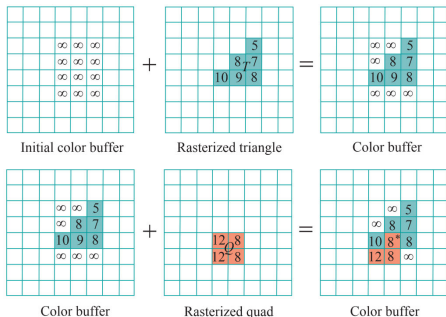
# Depth Test

- El **Depth Test** es el proceso que determina qué objetos están delante de otros en una escena:
  - Cuando se procesa un fragmento, se compara la distancia del fragmento con el valor del Z-buffer.
  - Si es menor, el fragmento en proceso reemplaza el valor de ambos buffers (el de colores y el de profundidad).
  - En otro caso, el fragmento queda descartado y no modifica ningún buffer.

Sólo los píxeles más cercanos a la cámara son visibles.

# Sin Blending

Con el test de profundidad por defecto activo y **sin blending**.

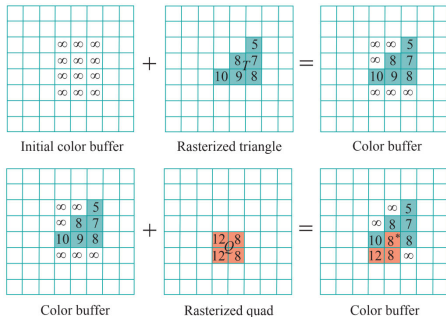


Un fragmento pasa el test si su profundidad es menor que la del buffer y sobrescribe ambos buffers con los valores del nuevo fragmento.



# Con Blending

Con el test de profundidad por defecto activo y **blending activo**.



Un fragmento pasa el test si su profundidad es menor que la del buffer y **mezcla** el color del buffer con el color del fragmento. El valor del Z-buffer se reemplaza.

# Ecuación de blending

- La mezcla de los dos colores se obtiene con los factores de blending que estén establecidos:

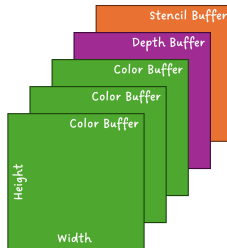
$$dstColor = (srcBFactor \times srcColor) + (dstBFactor \times dstColor)$$

siendo:

- **srcColor** = (**srcR**, **srcG**, **srcB**, **srcA**) el color RGBA del fragmento en proceso (*source color*),
- **dstColor** = (**dstR**, **dstG**, **dstB**, **dstA**) el color del Color Buffer correspondiente al mismo píxel (*destination color*), y
- **srcBFactor** y **dstBFactor** los correspondientes factores de **blending**.

# Frame Buffer

- El **Frame buffer** consta de varios buffers del mismo tamaño:
  - 1 **Colors buffers**: front and back
  - 2 **Depth buffer**: depth test
  - 3 **Stencil Buffer**: stencil test
- **Los tests permiten descartar fragmentos** para que no aporten color al color buffer.
- Se pueden utilizar frame buffers auxiliares (*frame buffer objects*).



# Stencil Buffer

- 1 Renderizar el cubo.
- 2 Renderizar el suelo.
- 3 Renderizar el cubo invertido.

## Problemas

- El Depth Test descarta los fragmentos tapados por el suelo.
- El reflejo aparece fuera del suelo.

## Ejemplos

reflejos, sombras, perfiles



# Stencil Buffer

## Ejemplos: reflejos, sombras, perfiles

- 1 Renderizar el cubo de forma habitual.
- 2 Activar la escritura en el Stencil Buffer (valor 1).
- 3 Renderizar el suelo escribiendo en el Stencil Buffer y no en el Z-buffer.
- 4 Configurar el Stencil test para que pasen los fragmentos con valor 1 en el Stencil Buffer.
- 5 Renderizar el cubo invertido.
- 6 Desactivar el Stencil Buffer y configurar el Z-buffer para lectura/escritura.



# Culling

- **Culling** significa: eliminar elementos, sacrificar selectivamente, entresacar o descartar.
- Tipos principales de culling:
  - **Backface culling**: eliminación de caras traseras.
  - **(view) Frustum culling**: eliminación de caras de una malla que están fuera del *frustum*.
  - **Occlusion culling**: eliminación de objetos que están ocultos por otros objetos.

