Introducción: Blending

- Blending

Ana Gil Luezas
Departamento de Sistemas Informáticos y Computación
Facultad de Informática
Universidad Complutense de Madrid

Colores RGBA

El valor por defecto de la componente Alpha es 1:

Un color RGB se completa con A=1

Su uso mas habitual es para modelar objetos traslúcidos:

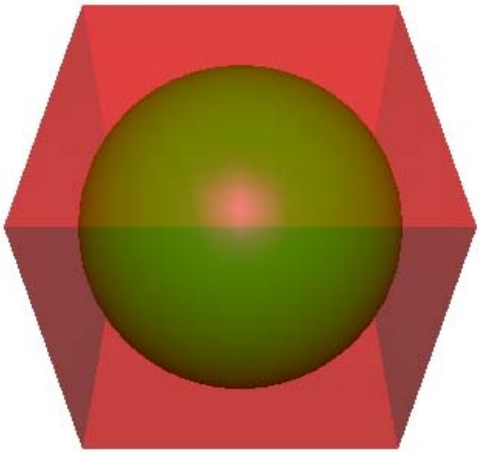
A = 1 -> opaco

A = 0 -> transparente

Se utiliza para combinar los colores de objetos que se superponen en la vista:

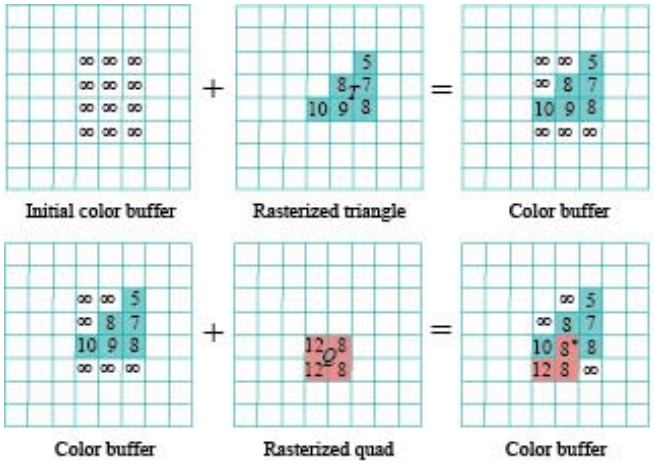
Cuando un objeto traslúcido

aparece delante de otro



Color and Depth Buffers

- glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BIT);
  T.draw(); // triángulo azul opaco
  Q.draw(); // rectángulo rojo translúcido



Con el test de profundidad por defecto activado

y sin blending

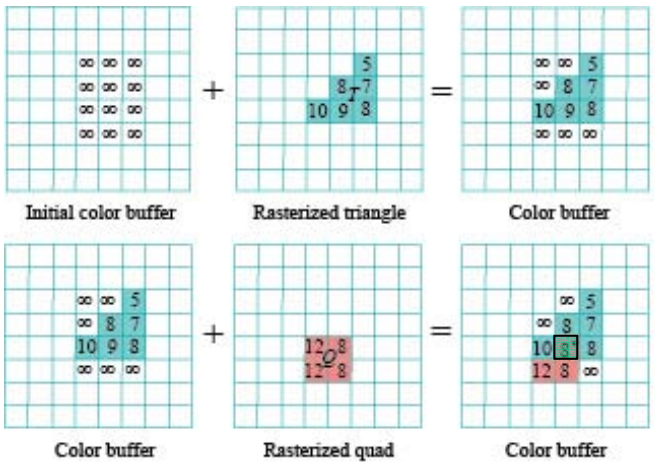
↓

Un fragmento pasa el test si su profundidad es menor que la del buffer

y sobrescribe el color del buffer con el del fragmento

Color and Depth Buffers

- glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BIT);
  T.draw(); // triángulo azul opaco
  Q.draw(); // rectángulo rojo translúcido



Con el test de profundidad por defecto

y blending activos

↓

Un fragmento pasa el test si su profundidad es menor que la del buffer

y mezcla el color del buffer con el del fragmento

Blending equation

Activar y desactivar el *Blending*

glEnable(GL_BLEND) / glDisable(GL_BLEND)

Ecuación: la mezcla de los dos colores se obtiene con los factores de blending que estén establecidos

(dstR, dstG, dstB, dstA) = sBF * (srcR, srcG, srcB, srcA)
+ dBF * (dstR, dstG, dstB, dstA)

siendo

(srcR, srcG, srcB, srcA) el color RGBA del fragmento en proceso,

(dstR, dstG, dstB, dstA) el color correspondiente del Color Buffer

y dBF y sBF los factores de blending

Los factores de la ecuación se establecen con:

glBlendFunc(sBFactor, dBFactor)

Blending equation

Configuración de los factores de la ecuación:

(dstR, dstG, dstB, dstA) = sBF * (srcR, srcG, srcB, srcA)
+ dBF * (dstR, dstG, dstB, dstA)

glBlendFunc(sBFactor, dBFactor) ambos factores en [0 , 1]:

glBlendFunc(1, 0) -> valores por defecto

glBlendFunc(0.5, 0.5) -> mismo peso para los dos colores

Para transparencias se utiliza la componente Alpha de los colores

glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);

Si el fragmento en curso es opaco (srcA =1) -> sBF =1 y dBF = 0

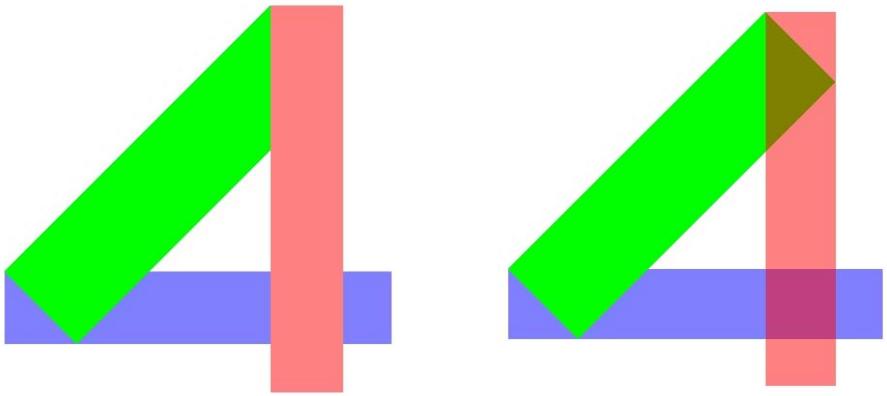
Si el fragmento en curso es transparente (srcA=0) -> sBF =0 y dBF = 1

Color and Depth Buffers

- glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BIT);

  Rectángulos: rojo translúcido (cercano), azul translúcido (lejano) y verde opaco (en medio)

  Con el test de profundidad por defecto y blending activos



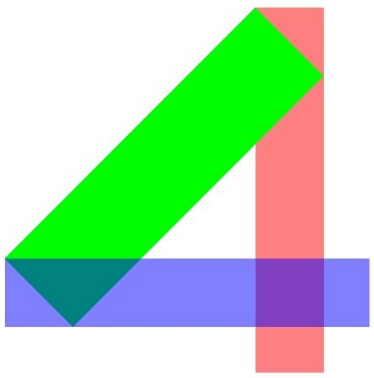
Orden: rojo, verde, azul -> azul, verde, rojo

Color and Depth Buffers

- glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BIT);

  Rectángulos: rojo translúcido (cercano), azul translúcido (lejano) y verde opaco (en medio)

  Test de profundidad desactivado y blending activo



Orden: rojo, verde, azul

- Orden para 2D
  1- Desactivar el test de profundidad
  2- Dibujar los objetos por capas: empezando por el más "lejano"

- Orden para 3D con objetos opacos y traslúcidos

  1- Test de profundidad activado por defecto
  Dibujar los objetos opacos

  2- Usar el buffer de profundidad solo para lectura:
  glDepthMask(GL_FALSE)
  Dibujar los objetos traslúcidos
  Los que están delante de los opacos se combinaran

  3- glDepthMask(GL_TRUE)