

Computación Gráfica

Blending & Antialiasing

Jonathan Antognini C.

Universidad Técnica Federico Santa María

29 de noviembre de 2012

1 Introducción

2 Blending

- Blending
- Transparencia
- Alpha Blending
- Implementación

3 Antialiasing

- Antialiasing
- Funciones antialiasing
- Línea y puntos
- Polígono

4 Conclusiones

Introducción

Los capítulos 6 y 7 definen dos procesos bastante utilizados en la computación gráfica:

- Blending: OpenGL proporciona un control a nivel de pixel del almacenamiento de valores RGBA en el buffer de color. Permite funcionalidades como composición, transparencias, etc.
- Antialiasing: la apariencia que tienen los puntos, líneas, polígonos en algunas ocasiones pueden presentar aliasing, y OpenGL proporciona opciones para evitar estos casos.

La idea del trabajo fue explorar dichas áreas y observar experimentalmente en qué influyen las distintas funciones.

Blending

Blend o mezcla, está presente en la etapa de renderizado de OpenGL, y permite aprovechar las salidas de color de fragmento, y las combina con los colores de destino que se está agregando. La utilización está ligada a la composición alpha, esto es, a nivel color de un pixel, se utilizan los 3 colores RGB, pero además (24bits) se añaden 8bits al canal Alpha (32 bits total), el cual hace referencia a la opacidad del color (o transparencia).

Los principales usos son para:

- Transparencia
- Composición
- Pintura, etc.

Transparencia

Una de las aplicaciones de blending, y la que fue estudiada en el trabajo fue transparencia.

Blending se puede utilizar para que los objetos aparezcan con cierto grado de transparencia.

Cuando se dibuja algo con la opción de blending activada, el renderizado vuelve a leer los pixeles del frame buffer, se mezcla con el nuevo color y pone los pixeles de vuelta de donde vinieron.

Alpha Blending

Se aplica alpha blending para crear el efecto de transparencia. Esto es útil en escenas que destacan objetos de cristal o líquidos. Esto lo hace combinando un primer plano translúcido con un color de fondo para crear una mezcla intermedia.

Para animaciones, el alpha blending también se puede usar para decolorar gradualmente una imagen y convertirla en otra.

El factor alpha (del RGBA) puede tomar cualquier valor de 0 a 1. Cuando se pone a 0 el primer plano es completamente transparente. Cuando se pone a 1, se hace opaco y totalmente obscurece el fondo. Cualquier valor intermedio crea una mezcla de las dos imágenes.

Implementación

En este ejemplo se dibujan dos cubos, uno dentro del otro, el exterior es rojo y el interior es verde.

Con las teclas -z "se rotan los dos cubos, al ejecutar el ejemplo son transparentes, al pulsar la tecla "1."el factor alpha del cubo rojo exterior aumenta, mientras que si se pulsa la tecla "2" se aumenta el valor del factor alpha del cubo interior.

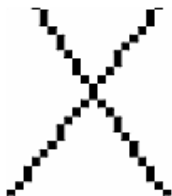
Cuando el factor alpha está al máximo, si lo intentamos aumentar, se vuelve a hacer transparente.

- `glDisable(GL_BLEND)`
- `void glBlendFunc (GLenum sfactor, GLenum dfactor);`

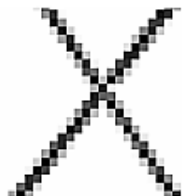
Antialiasing

Algunas líneas en OpenGL especialmente las casi horizontales o verticales con cierto grado de inclinación, parecen irregulares. Estos bordes dentados aparecen porque la línea ideal se aproxima por una serie de píxeles que deberá situarse en la cuadrícula de píxeles.

Por ejemplo:



Aliased



Antialiased

Funciones de antialiasing

Existen varios casos en donde ocurre el fenómeno de aliasing, y para ello también, existen varias técnicas para realizar antialiasing, como:

- Líneas y Puntos Antialiasing
- Polígonos Antialiasing
- Multisampling
- Antialiasing con texturas
- Antialiasing with accumulation buffer

Para este trabajo se tomaron las primeras dos técnicas para experimentar.

Antialiasing Línea y puntos

Debe considerarse por separado de antialiasing polígono ya que las técnicas suelen ser bastante diferente. Matemáticamente, una línea es infinitamente delgada. El intento de calcular el porcentaje de un pixel cubierto por un objeto infinitamente delgada sería imposible, por lo general, uno de los dos métodos siguientes se utiliza:

- La línea se modela como un largo, delgado, de un solo píxel de ancho cuadrilátero. El porcentaje de cobertura de píxel se calcula para cada píxel de la línea y el porcentaje de cobertura se utiliza como el valor alfa de la mezcla.
- La línea se modela como un objeto brillante infinitamente delgada y transparente.

Para realizar este procedimiento en OpenGL, se habilita `glEnable` pasando `GL_POINT_SMOOTH` o `GL_LINE_SMOOTH`. Además se puede sugerir una calidad llamando `glHint`, y se le puede pasar `GL_FASTEST` para indicar por ejemplo la opción más eficiente, o `GL_NICEST` para indicar la mejor calidad, o `GL_DONT_CARE` para no elegir nada.

Ejemplo Tomado del redbook.

Polígono antialiasing

En el caso de los polígonos, el fenómeno de aliasing ocurre en los bordes o aristas del mismo (dependiendo la forma).

En los bordes de polígonos rellenos es similar a los puntos de suavizado de líneas y líneas. Sin embargo, los polígonos anti-aliasing en modo color índice, no es práctico ya que las intersecciones son los objetos más comunes y que realmente necesita usar Blending de OpenGL para obtener resultados decentes.

El funcionamiento es similar, se llama glEnable con GL_POLYGON_SMOOTH, esto hace que los píxeles en los bordes del polígono sean asignados a valores fraccionarios de alfa sobre la base de su cobertura. También se puede proporcionar valores para GL_POLYGON_SMOOTH_HINT.

Conclusiones

Las conclusiones del trabajo específicas fueron:

- Blending: habilitar el blend dentro de OpenGL permite bastantes funcionalidades que de manera visual son útiles como en el caso de transparencia. Para este caso es importante entender la distribución en el buffer de color de un pixel y que este tiene 4 capas (color + opacidad).
- Antialiasing: este fenómeno aparece en distintas áreas de la ingeniería, pero del punto de vista de computación gráfica, es bueno saber que con un par de líneas OpenGL permite realizar cálculos internos para evitar el problema. Además las diferencias entre como aplicar antialiasing entre puntos, líneas, polígonos es bastante similar.

EOF