DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico Final

Renderizado 3D por software

Organización del computador 2

Integrante	LU	Correo electrónico
Germán Pinzón	475/13	pinzon.german.94@gmail.com
Angel More	931/12	angel_21_fer@hotmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2160 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (54 11) 4576-3359
http://www.fcen.uba.ar

Resumen

El propósito de este trabajo es desarrollar un renderizador 3D sin la ayuda de las librerías gráficas (Direct3D y OpenGL) que, junto con la placa de video, facilitan la tarea de dibujar primitivas tridimensionales. De esta manera, podemos ver que tan performante puede ser realizar este tipo de cómputos pura y exclusivamente con la CPU. El desarrollo será principalmente en lenguaje C y para las partes críticas de cálculo utilizaremos Assembler junto con SIMD para poder equilibrar un poco el hecho de no apoyarnos en la placa de video. Además, debido a que necesitamos de alguna manera dibujar píxeles en la pantalla, haremos uso de una librería gráfica de bajo nivel llamada SDL que nos permite realizar este tipo de tareas usando exclusivamente la CPU.

Índice

1. Introducción

1.1. ¿Qué es un renderizador 3D?

Para dibujar los gráficos 3D que vemos en muchos de los juegos o software de simulación que podemos encontrar hoy en día, se hacen uso de librerías que facilitan este trabajo. En general siempre existe una abstracción de alto nivel que nos permite dibujar en pantalla un modelo o primitiva 3D de manera directa, indicando su posición en el espacio, rotación o demás características. Sin embargo, si empezamos a bajar el nivel de abstracción, siempre vamos a converger en el uso de dos librerías de bajo nivel: Direct3D y OpenGL.

Direct3D y OpenGL son librerías gráficas que trabajan con la placa de video independientemente del fabricante, permitiendo de esta manera que el desarrollador pueda concentrarse en el código de más alto nivel y no tenga que preocuparse por los detalles de hardware. Además, brindan el soporte matemático necesario para que el programador no tenga que realizar ciertos cálculos o construir matrices desde 0, si no que con algunos datos (por ejemplo para construir una matriz de rotación basta con pasarle el ángulo de rotación) ya se encarga de realizar la tarea por completo. De esta manera, si se desea dibujar una primitiva 3D, basta con pasar sus vértices a alguna función específica que provean estas librerías.

Podemos pensar entonces a un renderizador 3D como un software que se encarga de hacer ésto último, es decir, tomar la información necesaria sobre un modelo 3D (en un formato tridimensional) y convertirla a píxeles para posteriormente dibujarlos en la pantalla. Cuando decimos “información necesaria” estamos resumiendo muchísimas cosas ya que para dibujar un modelo 3D necesitamos bastantes más datos que solo sus vértices. El objetivo de este trabajo va a ser programar una suerte de DirectX u OpenGL propio pero que interaccione solamente con el CPU.

1.2. SDL

SDL es una librería gráfica open source de bajo nivel, multiplataforma, que trabaja con C y que brinda soporte para el desarrollo de aplicaciones gráficas 2D (aunque puede ser usada también con OpenGL). Esta librería nos permite realizar tareas como crear la ventana, imprimir texto en pantalla, detectar eventos de mouse y teclado y pintar píxeles en la ventana de manera sencilla.

Elegimos SDL básicamente por dos motivos, el primero porque es una librería “madura” en el sentido de que lleva varios años en desarrollo y tiene una comunidad importante detras. El segundo motivo fue que también nos permite trabajar con la CPU en el sentido de que podemos evitar la aceleración por hardware de la placa de video, pudiendo de esta manera medir de manera más exacta la performance a la hora de realizar experimentos y poder sacar conclusiones más objetivamente.

1.3. Otras features

Si bien nuestro principal objetivo es el de realizar los cálculos algebraicos necesarios para convertir los vértices de un modelo 3D en píxeles y así poder dibujarlos en la pantalla, existen otras cosas como por ejemplo computar transformaciones básicas sobre el objeto como rotarlo y estirarlo en distintas direcciones, aplicarle texturas y también un tipo específico de iluminación. Además, dado que en general los modelos 3D son generados por programas de diseño como 3D Studio Max o Blender y existen numerosos formatos de archivos de modelos 3D, resulta conveniente elegir uno e implementar un lector para poder cargar la información fácilmente. Nosotros elegimos el formato *.obj e implementamos dicho lector.