

Programa Computación Gráfica 2016

Ing John Coppens (john@jcoppens.com)

22 de agosto de 2016

2016-05-19

Parte I Teórico

<i>Semana</i>	<i>Sección</i>	<i>Práctico</i>
1	11	Introducción
2	2.12.1	Visualización de gráficos
3	2.22.2	Almacenamiento
4	33	Librerías gráficas
5	4.14.1	Introducción a OpenGL
6	4.24.2	Procesamiento en OpenGL
7	55.1	Matemática aplicada
8	5.15.2	Aplicación de la matemática
9	66	Modelación y edición de gráficos
10	7.17.1	Efectos: Luz
11	7.27.2	Efectos: Superficies
12	88.1	Rasterización por rayos: Técnica
13	8.28.2	Implementación
14	8.29	Influencia de la internet
15		Recuperación

1. Introducción

Presentación personal - Documentación - Organización de clases - Sobre prácticos y exámenes (formato de presentación, e-mails) - Trabajos - Contenido de la materia - Técnicas de generación sintética de imágenes - Necesidades gráficas en la computación (técnico, arquitectura, arte, ...) - Métodos de sintetización de imágenes: OpenGL o rasterización por rayos - Máquina virtual con herramientas útiles para diferentes tipos de gráficos, ejemplos de imágenes - Referencias a fuentes de información en la red: Programas, teoría, imágenes, ...

2. Representación de gráficos

2.1. Parte 1 Visualización

Representación en pantalla: Vectorizado o pixelado? - Historia del hardware gráfico, tipos de terminales, representación de imágenes en memoria - Interfaz software-hardware - Estado actual - Resolución - El problema del tamaño de los archivos: Compresión - Primitivas gráficas.

2.2. Parte 2 Almacenamiento

Ejemplos de programas para generación de diferentes tipos de imágenes - Estructuras de definición y almacenamiento de graficos - Ejemplos de formatos de imágenes pixelados en disco: PNM, GIF,

PNG, BMP, JPG - Formatos de animaciones y video: GIF, MPEG - Codecs. Fallas (?) del ojo - y como se aprovechan en computación gráfica.

3. Uso de librerías de graficación 2D(/3D)

Identificación de la librerías en la máquina virtual - Trazar una línea - Otros elementos fundamentales disponibles - Visualización y generación del archivo (pixelado: PNG y vectorizado: PoSema-nastScript y SVG) - Integración de imágenes en una página web - Vistas bidimensionales. Conceptos tridimensionales - Transformaciones geométricas y de modelado tridimensional - Transformaciones geométricas.

4. Introducción al proceso de OpenGL

4.1. Modelación y edición de gráficosIntro

Qué es OpenGL - Variantes de OpenGL: OpenGL, OpenGL ES, WebGL: Puntos en común y diferencias - Como desarrollar en OpenGL: Conexión con diferentes idiomas: C++, Python, Javascript, etc. La necesidad de generar imágenes en la Internet: Flash, SVG, HTML 5, WebGL - El 'tubo' de procesamiento de OpenGL - Problemas de integración de OpenGL (posibilidad de ejecución de OpenGL por software) - Limitaciones de OpenGL.

4.2. Procesamiento de OpenGL

El 'tubo' de procesamiento de OpenGL - Descripción de los paso que sigue OpenGL para procesar la descripción de la imagen y convertirla en visual - El rasterizador - Shaders - Acceso y control de cada paso del proceso - Ejemplos básico de elementos fundamentales.

5. Matemática gráfica

5.1. Matemática utilizada

Matemática requerida por los gráficos por computadora: Representación de 'cosas' en programación: Ejes - coordenadas - texturas - colores - tiempo. - Repaso de la matemática de matrices y vectores - Matrices: Peligro: Columnas o Filas primero? Matemática requerida por los gráficos por computadora - Operaciones con vectores: Suma - productos - significado en 2D/3D

5.2. Aplicación

Semana

Operaciones: Traducir, rotar, escalar - Importancia del orden de las transformaciones - Matrices correspondientes. Fórmula de Rodrigues. Librerías que implementan las transformación - Metodos de detección de superficies visibles.

Visualización de 3D: Convertir 3D a 2D: Matrices de proyección: Ortográfico - Perspectiva (pro-blemas). Otras formas de visualizar 3D: Seudo-estereo (la moda 3D) - El futuro: Holografía?

6. Modelación/edición de imágenes

Programas para todos los gustos: Rasterizado: Photoshop, GIMP. Vectorizado: Inkscape, Corel-Draw. Rasterizado por rayos: Editor de texto (!), o programas auxiliares como KPovModeller, y otros. Animación por computadora: Necesitamos la 4ta dimensión! Ejemplo: Blender.

7. Mejorar la calidad: Efectos

7.1. Luz

Para mejorar la presentación de las imágenes (y el valor comercial!) - Iluminación detallada: Luz puntual, dirigida, y paralelo - Modelos de iluminación - Técnicas de realismo en reflexión: Phong, especular, ... (área de investigación)

7.2. Las superficies

y métodos de presentación de superficies - - Cosas transparentes: simulación de refracción - Texturas sintéticas y fotográficas - Modelos con color y aplicaciones del color.

8. Rasterización por rayos

8.1. Filosofía

Filosofía de esta técnica - Técnicas extras necesarias y diferencias fundamentales con OpenGL - El estado del arte - Las tendencias y el interés de fabricantes de GPU - Técnica más importante: Determinar si el rayo toca objetos - Reflección y refracción - Necesidad de recursión

8.2. Implementación

Esqueleto de un programa para RayTracing - Pasos a seguir para 'renderizar' una escena - Algunos problemas a solucionar (precisión) - Texturas (sintéticas y mapeadas) - Paralelismo (OpenMP)

9. La Internet

La presión de la internet: El navegador se convirtió al elemento central en computadoras (y otros medios) - Como enviar imágenes a clientes con ancho de banda limitado - Generación de imágenes en la aplicación del cliente en lugar del servidor - Trasladar el proceso de renderizado al cliente - 'Estado del arte': Siguen algunos problemas, pero se están solucionando rápidamente - Problema del navegador gigante e incompatibilidades entre navegadores.

Convertir un proyecto OpenGL para uso via la internet (utilizando WebGL).

Parte II

Prácticos

<i>Semana</i>	<i>Sección</i>	<i>Práctico</i>
1	1	Mínima introducción a Linux - Instalación máquina virtual - Comandos mínimos de navegación - Ubicación de software - Pruebas - Solución de problemas
2	2.1	Muestra de imágenes - Programas visualizadores (Geeqie, ...) - Identificación de programas (comando 'file')
3	2.2	Investigar archivos en disco de diferentes tipos de archivos - 'Número mágico' - Comparar tamaños de archivos de diferentes tipos (Raster: BMP, GIF, PNG, JPEG, TIFF, JPEG2000 - Vector: SVG, Postscript) - Investigar programas de conversión
4	3	Realizar mínimos programas utilizando diferentes librerías de graficación (Python + GooCanvas, Python + Matplotlib)
5	4.1	Sigue del punto anterior (con el agregado de programas para uso de OpenGL en 2D)
6	4.2	Experimentos con Shaders, y OpenGL en 3D
7	5.1	Ejercicios de matemática básica de matrices y vectores.
8	5.2	Aplicación de las operaciones matemáticas en operaciones gráficas - rotación, traslado, escala, proyección
9	6	Practicar con diferentes programas de modelación (raster, vector, blender)
10	7.1	Edición de un programa con OpenGL, con el agregado de diferentes tipos de iluminación
11	7.2	Continuación del ejercicio anterior, con texturas
12	8.1	Investigación de software de rasterización por rayos - crear tabla comparativa
13	8.2	Implementación de ciertas rutinas de un rasterizador
14	8.2	Sigue del punto anterior - Meta: realizar imágenes con luz ambiental
15	9	Sigue del punto anterior - Meta: realizar imágenes diferentes tipos de luz, y reflexiones.

Categorías:

<i>Letra</i>	<i>Rubro</i>	Horas
R	Problemas rutinarios	15
X	Experimental	15
I	Problemas de ingeniera	-