



Universidad Nacional de Ingeniería (UNI)

Escuela Profesional de
Ciencia de la Computación
Sílabo 2024-II

1. CURSO

CS251. Computer graphics (Mandatory)

2. INFORMACIÓN GENERAL

2.1 Curso	: CS251. Computer graphics
2.2 Semestre	: 7 th Semester.
2.3 Créditos	: 4
2.4 horas	: 2 HT; 4 HP;
2.5 Duración del periodo	: 16 semanas
2.6 Condición	: Mandatory
2.7 Modalidad de aprendizaje	: Face to face
2.8 Prerrequisitos	: MA106FCCS. Numerical Methods. (4 th Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

It offers an introduction to the area of Computer Graphics, which is an important part of Computer Science. The purpose of this course is to investigate the fundamental principles, techniques and tools for this area.

5. OBJETIVOS

- Bring students to concepts and techniques used in complex 3-D graphics applications.
- Give the student the necessary tools to determine which graphics software and which platform are best suited to develop a specific application.

6. RESULTADOS DEL ESTUDIANTE

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

7. TEMAS

Unidad 1: Conceptos Fundamentales (6 horas)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Aplicaciones multimedia, incluyendo interfaces de usuario, edición de audio y vídeo, motores de juego, cad, visualización, realidad virtual. • Soluciones de compensación entre el almacenamiento de datos y los datos re-computing es personalizado por vectores y raster en representaciones de imágenes. • Modelos de color sustractivo Aditivo y (CMYK y RGB) y por qué estos proporcionan una gama de colores. • Animación como una secuencia de imágenes fijas.	• Explicar en términos generales cómo las señales analógicas pueden ser representadas por muestras discretas, por ejemplo, cómo las imágenes pueden ser representadas por píxeles [Familiarizarse] • Describir modelos de color y su uso en los dispositivos de visualización de gráficos [Familiarizarse] • Describir las ventajas y desventajas entre el almacenamiento de información vs almacenar suficiente información para reproducir la información, como en la diferencia entre el vector y la representación de la trama [Familiarizarse] • Describir los procesos básico de la producción de movimiento continuo a partir de una secuencia de cuadros discretos(algunas veces llamado it flicker fusion) [Familiarizarse]
Lecturas : [HB90]	

Unidad 2: Rendering Básico (12 horas)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Renderizado en la naturaleza, por ejemplo, la emisión y dispersión de la luz y su relación con la integración numérica. • Renderizado Forward and Backward (i.e., <i>ray-casting</i> y rasterización) • Radiometría básica, triángulos similares y modelos de proyecciones • Afinamiento y Transformaciones de Sistemas de coordenadas • <i>Ray tracing</i> • Visibilidad y oclusión, incluyendo soluciones a este problema, como el almacenamiento en búfer de profundidad, algoritmo del pintor, y el trazado de rayos. • Rasterización triangular simple. • Renderización con una API basada en shader. • Aplicación de la representación de estructuras de datos espaciales. • Muestreo y anti-aliasing. • Renderizado Forward and Backward (i.e., <i>ray-casting</i> y rasterización)	• Discutir el problema de transporte de la luz y su relación con la integración numérica, es decir, se emite luz, dispersa alrededor de la escena, y es medida por el ojo [Familiarizarse] • Describir la tubería básica gráficos y cómo el factor de representación va hacia adelante y atrás en esta [Familiarizarse] • Crear un programa para visualizar modelos 3D de imágenes gráficas simples [Usar] • Obtener puntos en 2-dimensiones y 3-dimensiones por aplicación de transformaciones afín [Usar] • Aplicar sistema de coordenadas de 3-dimensiones y los cambios necesarios para extender las operaciones de transformación 2D para manejar las transformaciones en 3D [Usar] • Contrastar la renderización hacia adelante <i>forward</i> y hacia atrás <i>backward</i> [Evaluar] • Explicar el concepto y las aplicaciones de mapeo de texturas, muestreo y el <i>anti-aliasing</i> [Familiarizarse] • Explicar la dualidad de rastreo de rayos/rasterización para el problema de visibilidad [Familiarizarse] • Implementar un sencillo renderizador en tiempo real utilizando una API de rasterización (por ejemplo, OpenGL) utilizando buffers de vértices y <i>shaders</i> [Usar] • Calcular las necesidades de espacio en base a la resolución y codificación de color [Evaluar] • Calcular los requisitos de tiempo sobre la base de las frecuencias de actualización, técnicas de rasterización [Evaluar]
Lecturas : [HB90], [Hug+13], [Wol11], [Shr+13]	

Unidad 3: Programación de Sistemas Interactivos (2 horas)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Manejo de eventos e interacción de usuario. • Enfoques para el diseño, implementación y evaluación de la interacción sin mouse – Interfaces táctiles y multitáctiles. – Interfaces compartidas, incorporadas y grandes – Nuevas modalidades de entrada (tales como datos de sensores y localización) – Nuevas ventanas, por ejemplo, iPhone, Android – Reconocimiento de voz y procesamiento del lenguaje natural – Interfaces utilizables y tangibles – Interacción persuasiva y emoción – Tecnologías de interacción ubicuas y contextuales (Ubicomp) – Inferencia bayesiana (por ejemplo, texto predictivo, orientación guiada) – Visualización e interacción de ambiente / periféricos	• Discute las ventajas (y desventajas) de las interfaces no basadas en ratón [Evaluar]
Lecturas : [HB90]	

Unidad 4: Modelado Geométrico (15 horas)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Operaciones geométricas básicas como cálculo de intersección y pruebas de proximidad. • Volúmenes, voxels y representaciones basadas en puntos. • Curvas polinomiales y Superficies paramétricas. • Representación implícita de curvas y superficies. • Técnicas de aproximación, tales como curvas polinómicas, curvas Bezier, curvas spline y superficies, y base racional no uniforme (NURB) espinas, y el método de ajuste de nivel. • Técnicas de superficie de representación incluyendo teselación, la representación de malla, carenado malla, y las técnicas de generación de mallas, como la triangulación de Delaunay, marchando cubos. • Técnicas de subdivisión espacial. • Modelos procedimentales como fractales, modelamiento generativo y sistemas L. • Modelos deformables de forma libre y elásticamente deformables. • Subdivisión de superficies. • Modelado multiresolución. • Reconstrucción. • Representación de Geometría Sólida Constructiva (GSC)	• Representar curvas y superficies utilizando formas tanto implícitas y paramétricas [Usar] • Crear modelos poliédrico simples por teselación de superficies [Usar] • Generar un modelo fractal o terreno usando un método de procedimiento [Usar] • Generar una malla de un conjunto de puntos adquiridos por un scanner laser [Usar] • Construct modelos de geometría sólida constructiva a partir de simples primitivas, tales como cubos y superficies cuádricas [Usar] • Contrastar métodos de modelización con respecto a espacio y tiempo de complejidad y calidad de imagen [Evaluar]
Lecturas : [HB90], [Shr+13]	

Unidad 5: Renderizado Avanzado (6 horas)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Tiempo (desenfoco de movimiento), la posición del objetivo (enfoco), y la frecuencia continua (color) y su impacto en la representación. • Mapeo de Sombras. • Selectiva de oclusión. • Dispersión de la Superficie. • Renderizado no fotorealístico. • Arquitectura del GPU. • Sistemas visuales humanos incluida la adaptación a la luz, la sensibilidad al ruido, y la fusión de parpadeo.	• Demostrar como un algoritmo calcula una solución a la ecuación de renderización [Evaluar] • Demostrar las propiedades de un algoritmo de renderización, por ejemplo, completo, consistente, e imparcial [Evaluar] • Implementar un algoritmo no trivial de sombreado (por ejemplo, sombreado caricaturizado (<i>toon shading</i>), mapas de sombras en cascada (<i>cascaded shadow maps</i>)) bajo una API de rasterización [Usar] • Discutir como una técnica artística particular puede ser implementada en un renderizador [Familiarizarse] • Explicar como reconocer las técnicas gráficas usadas para crear una imagen en particular [Familiarizarse]
Lecturas : [HB90], [Hug+13], [Wol11], [Shr+13]	

Unidad 6: Animación por computadora (4 horas)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Cinemática directa e inversa. • Detección de colisiones y respuesta. • Animación procedimental empleando ruido, reglas (boids/crowds) y sistemas de partículas. • Algoritmos Skinning. • Movimientos basado en la física, incluyendo la dinámica del cuerpo rígido, sistemas de partículas físicas, redes de masa-muelle de tela y la carne y el pelo. • Animación de Cuadros Principales • Splines • Estructuras de datos para rotaciones, como cuaterniones. • Animación de Cámara. • Captura de Movimiento.	• Calcular la orientación de partes articuladas de un modelo de una localización y orientación usando un enfoque de cinemática inversa [Usar] • Implementar el método de interpolación <i>spline</i> para producir las posiciones y orientaciones en medio [Usar] • Implementar algoritmos para el modelamiento físico de partículas dinámicas usando simplemente la mecánica de Newton, por ejemplo Witkin & Kass, serpientes y gusanos, Euler simpléctica, Stormer/Verlet, o métodos de punto medio de Euler [Usar] • Discutir las ideas básicas detrás de algunos métodos para dinámica de fluidos para el modelamiento de trayectorias balísticas, por ejemplo salpicaduras, polvo, fuego, o humo [Familiarizarse] • Usar el software de animación común para construir formas orgánicas simples usando <i>metaball</i> y el esqueleto [Usar]
Lecturas : [HB90], [Shr+13]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [HB90] Donald Hearn and Pauline Baker. *Computer Graphics in C*. Prentice Hall, 1990.
- [Wol11] David Wolff. *OpenGL 4.0 Shading Language Cookbook*. Packt Publishing, 2011.
- [Hug+13] John F. Hughes et al. *Computer Graphics - Principles and Practice 3rd Edition*. Addison-Wesley, 2013.
- [Shr+13] Dave Shreiner et al. *OpenGL, Programming Guide, Eighth Edition*. Addison-Wesley, 2013.