

## PROGRAMA DE CURSO

### MODELACIÓN Y COMPUTACIÓN GRÁFICA PARA INGENIEROS

#### A. Antecedentes generales del curso:

Departamento	Ciencias de la computación					
Nombre del curso	Modelación y Computación Gráfica para Ingenieros	Código	CC3501	Créditos	6	
Nombre del curso en inglés	<i>Modeling and Computer Graphics for Engineers</i>					
Horas semanales	Docencia	3	Auxiliares	1,5	Trabajo personal	5,5
Carácter del curso	Obligatorio	X		Electivo		
Requisitos	CC1002: Introducción a la programación, MA2001: Cálculo en varias variables, MA2601: Ecuaciones diferenciales ordinarias, FI2001: Mecánica					

#### B. Propósito del curso:

El curso se ubica en el V semestre, en el ciclo de Licenciatura. Introduce a las y los estudiantes en la disciplina de la computación gráfica, sus principales problemáticas y aplicaciones en el contexto de las ciencias y la ingeniería. Al término del curso se espera que los y las estudiantes modelen, solucionen y visualicen computacionalmente problemas aplicados, que involucren datos, geometrías y escenas complejas en 2 y 3 dimensiones (2D y 3D). En las actividades prácticas se programa la GPU, utilizando la librería de bajo nivel *OpenGL*.

A modo de ejemplo, se pueden desarrollar aplicaciones gráficas interactivas funcionales simples como videojuegos, y adquirir herramientas con las cuales se podrían trabajar simulaciones físicas, sistemas CAD, visualización arquitectónica, modelación de terrenos, entre otras aplicaciones posibles.

El curso tributa a las siguientes competencias específicas (CE) y genéricas (CG):

CE1: Analizar problemas computacionales, construir modelos, expresándolos en representaciones y lenguajes formales adecuados.

CE2: Analizar, diseñar y/o adoptar, algoritmos y estructuras de datos que cumplan con las garantías requeridas de correctitud y eficiencia.

CE5: Concebir, diseñar y construir soluciones de software, siguiendo un proceso sistemático y cuantificable, acorde a los fundamentos, eligiendo el paradigma y las técnicas más adecuadas.

CE6: Desarrollar software en una amplia variedad de plataformas y lenguajes de programación.

CE8: Diagnosticar y resolver problemas en el funcionamiento de software cercano a la plataforma para mejorar su desempeño.

CG1: Comunicación académica y profesional

Comunicar en español de forma estratégica, clara y eficaz, tanto en modalidad oral como escrita, puntos de vista, propuestas de proyectos y resultados de investigación fundamentados, en situaciones de comunicación compleja, en ambientes sociales, académicos y profesionales.

CG2: Comunicación en inglés

Leer y escuchar de manera comprensiva en inglés una variedad de textos e informaciones sobre temas concretos o abstractos, comunicando experiencias y opiniones, adecuándose a diferentes contextos y a las características de la audiencia.

### C. Resultados de aprendizaje:

Competencias específicas	Resultados de aprendizaje
CE1, CE5	RA1: Modela computacionalmente aplicaciones gráficas e interactivas en 2D y 3D, considerando el patrón de arquitectura de software Modelo-Vista-Controlador (MVC), objetos geométricos, escenas complejas, datos y dispositivos periféricos.
CE2	RA2: Utiliza mallas poligonales, curvas, modelación jerárquica y de sólidos para modelar objetos geométricos complejos, considerando las especificaciones del usuario, la <i>discretización</i> y el tipo de <i>visualización</i> deseable.
	RA3: Aplica el proceso de rasterización de modelos tridimensionales y <i>renderiza</i> o genera imágenes, usando algoritmos de pintado como mallas de alambre, colores, degradaciones, texturas y técnicas de iluminación y sombreado.
CE1, CE8	RA4: Resuelve problemas de modelación geométrica y visualización, programando a nivel de procesador gráfico (GPU) vía OpenGL/GLSL, con el fin de construir aplicaciones interactivas en tiempo real.
CE1, CE2	RA5: Implementa simulaciones gráficas simples de movimiento basadas en fenómenos físicos y sus correspondientes ecuaciones diferenciales ordinarias (EDOs), considerando aspectos como la detección de colisiones, la inestabilidad numérica y las limitaciones del método de simulación.

CE6	RA6: Analiza la aplicabilidad de sistemas computacionales gráficos, como motores de juego, sistemas de diseño asistido por computador (CAD), sistemas de información geográficos (GIS), y software de animación, considerando el aporte de estos sistemas a otras disciplinas.
Competencias genéricas	Resultados de aprendizaje
CG1	RA7: Produce reportes donde justifica decisiones sobre el diseño, implementación y funcionalidades de aplicaciones gráficas desarrolladas y expone oralmente, con criterios de concisión y claridad, sobre una de estas aplicaciones, evidenciando manejo conceptual del proceso.
CG1, CG2	RA8: Analiza, a partir de una lectura analítica y crítica, textos sobre modelación gráfica para utilizar, con propiedad y pertinencia, conceptos teóricos, técnicas y herramientas al diseño de diversas aplicaciones gráficas y al análisis de ejemplos de esta disciplina.
	RAXX: Genera ideas o inputs novedosos como posibilidades para optar por aquella con la cual construir aplicaciones gráficas interactivas en tiempo real, considerando las características del usuario potencial y el uso de sistemas computacionales gráficos, motores de juego, software de animación, entre otros.

#### D. Unidades temáticas:

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
1	RA1, RA4	Introducción a la Computación Gráfica	3,5 semanas
Contenidos		Indicador de logro	
1.1. Aplicaciones de Computación Gráfica. 1.2. Tecnologías de Display. 1.3. <i>Graphics Rendering Pipeline</i> . 1.4. <i>OpenGL</i> y <i>Shaders</i> . 1.5. Discretizaciones. 1.6. Transformaciones Geométricas. 1.7. Interpolación lineal.		La/el estudiante:  1. Analiza la evolución del hardware gráfico, considerando su desarrollo y avances en el contexto de la computación gráfica. 2. Describe y analiza el <i>pipeline</i> de renderizado, considerando sus etapas, con énfasis en la <i>API OpenGL core profile</i> como ejemplo. 3. Describe el proceso de rasterización y cómo este permite generar imágenes a partir de líneas y polígonos. 4. Discretiza manualmente figuras simples en 2D, generando una malla geométrica. 5. Elabora animaciones simples vía interpolación lineal.	

	6. Programa aplicaciones interactivas simples en 2D utilizando OpenGL, transformaciones y mallas geométricas.
Bibliografía de la unidad	(5) Apuntes del curso. (1) Capítulo 1: <i>The Rendering Pipeline</i> ; Capítulo 4: <i>Transforms</i> . (2) Capítulo 1: <i>Introduction</i> ; Capítulo 2: <i>Our First OpenGL Program</i> .

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
2	RA1, RA2, RA4, RA7, RA8	Técnicas de Modelación	3 semanas
Contenidos		Indicador de logro	
2.1. Introducción a Texturas. 2.2. Modelación Jerárquica. 2.3. Modelo-Vista-Controlador. 2.4. Curvas y sus aplicaciones. 2.5. Animaciones.		La/el estudiante: <ol style="list-style-type: none"> <li>1. Modela escenas complejas que involucren texturas y múltiples objetos con posibles jerarquías entre sí.</li> <li>2. Desarrolla aplicaciones interactivas, utilizando el patrón de arquitectura Modelo-Vista-Controlador.</li> <li>3. Utiliza curvas de Hermite, Bezier y Catmull-Rom en la modelación geométrica y animación, considerando restricciones en sus parámetros.</li> <li>4. Desarrolla una aplicación gráfica interactiva en <i>OpenGL</i>, utilizando conceptos de modelación geométrica y computación gráfica en 2D.</li> <li>5. Reporta, de manera concisa, las decisiones tomadas para el desarrollo de la aplicación gráfica interactiva en 2D.</li> <li>6. Identifica y analiza distintas técnicas de animación: transformaciones simples, esqueletos, cambios de color, máquinas de estado, considerando aplicabilidad y ventajas de cada una.</li> <li>7. Utiliza conceptos, teorías para identificar y analizar técnicas de modelación útiles al momento de desarrollar aplicaciones gráficas, a partir de una lectura analítica de textos tanto en inglés como español.</li> </ol>	
Bibliografía de la unidad		(1) Capítulo 7.5: <i>Texture Mapping</i> ; Capítulo 11: <i>Curves and Surfaces</i> . (3) Capítulo 3: <i>Scene Graphs</i> . (11) Capítulo 9.6: The Model-View-Controller architectural pattern. (8) Capítulo 11: <i>Computer Animation</i> .	

Número	RA al que tributa	Nombre de la unidad	Duración en Semanas
3	RA1, RA2, RA3, RA4, RA6, RA7	Modelación y Visualización Tridimensional	5 semanas
Contenidos		Indicador de logro	
<p>3.1. Proyecciones.</p> <p>3.2. Mallas de Polígonos.</p> <p>3.3. Iluminación y Sombreado.</p> <p>3.4. Introducción a Iluminación Global.</p> <p>3.5. Recorte.</p> <p>3.6. Eliminación de Caras Ocultas.</p> <p>3.7. Modelación de Sólidos.</p> <p>3.8. Aplicaciones: impresión 3D, sistemas CAD y sistemas de información geográfica (GIS).</p>		<p>La/el estudiante:</p> <ol style="list-style-type: none"> <li>1. Ajusta transformaciones de vista y de proyección a conveniencia, considerando el proceso de visualización 3D.</li> <li>2. Genera mallas de polígonos con propósitos de visualización, utilizando estructuras de datos apropiadas.</li> <li>3. Identifica y analiza las especificaciones de formatos de archivos en mallas poligonales estándares: <i>STL, OBJ, OFF, entre otros</i>.</li> <li>4. Utiliza mallas poligonales para la modelación de terrenos, a partir de ejemplos.</li> <li>5. Analiza y utiliza el modelo de iluminación de Phong considerando las estrategias de <i>shading Flat, Gouraud y Phong</i>.</li> <li>6. Identifica y explica las diferencias entre iluminación local y global, en relación con modelos geométricos.</li> <li>7. Determina las ventajas de utilizar la iluminación global, dado que esta permite un mejor realismo de la escena.</li> <li>8. Analiza y concluye cómo el recorte y la eliminación de caras ocultas mejoran el desempeño de una aplicación gráfica interactiva.</li> <li>8. Desarrolla una aplicación gráfica interactiva en <i>OpenGL</i>, utilizando conceptos de modelación geométrica y computación gráfica en 3D.</li> <li>9. Reporta, de manera concisa, las decisiones tomadas para el desarrollo de la aplicación gráfica interactiva en 3D.</li> <li>10. Expone, de manera concisa, las decisiones tomadas para el desarrollo de una aplicación gráfica interactiva en 3D.</li> <li>11. Compara y contrasta las técnicas de representación de sólidos como geometría sólido constructiva, volúmenes de barrido, representación en celdas, considerando la capacidad de representación de cada una.</li> <li>12. Analiza la aplicabilidad y aportes de tipos de sistemas computacionales gráficos a otras</li> </ol>	

	disciplinas tales como Mecánica, Arquitectura, Diseño, Geología, Medicina, entre otros.
Bibliografía de la unidad	<p>(1) Capítulo 5.5: <i>Projections</i>; Capítulo 7: <i>Lighting and Shading</i>.</p> <p>(2) Capítulo 4: <i>Math for 3D Graphics, Understanding Transformations</i>; Capítulo 3: <i>Following the Pipeline, Primitive Assembly, Clipping and Rasterization</i>.</p> <p>(3) Capítulo 8.1: <i>Mesh Representation</i>; Capítulo 8.2: <i>Polygonal Manifolds</i>, Capítulo 8.3 <i>Mesh Data Structures</i>.</p> <p>(7) Capítulo 12.4: <i>The Rendering Equation</i>; Capítulo 12.5: <i>Radiosity</i> Capítulo 12.6 <i>Global Illumination and Path Tracing</i>.</p> <p>(10) Capítulo 12: <i>Solid Modeling</i>.</p>

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
4	RA1, RA2, RA5, RA6, RA7	Colisiones y Simulaciones Físicas	2 semanas
Contenidos		Indicador de logro	
<p>4.1. Simulación de problemas de Física Lineal.</p> <p>4.2. Diferencias Finitas para EDOs.</p> <p>4.3. Introducción a la Detección de Colisiones.</p> <p>4.4. Aplicaciones: Simulaciones computacionales gráficas y motores de juego.</p>		<p>La/el estudiante:</p> <ol style="list-style-type: none"> <li>Identifica y resuelve, en el contexto de una simulación, el problema de detección de colisiones.</li> <li>Aplica métodos de Diferencias Finitas a la resolución de EDOs, como el método de Euler y el método de Series de Taylor, comprendiendo las limitantes de precisión de cada uno de estos métodos.</li> <li>Desarrolla una aplicación gráfica que simule movimientos basado en física, utilizando detección de colisiones y los métodos de diferencias finitas para EDOs.</li> <li>Argumenta por escrito sobre las decisiones tomadas para el desarrollo de una aplicación gráfica que simule movimiento basado en fenómenos físicos.</li> </ol>	
Bibliografía de la unidad		(1) Capítulo 13: <i>Linear Physics</i> ; Capítulo 16.4: <i>Numerical Methods, Ordinary Differential Equations</i> ; Capítulo 12: <i>Collision Detection</i> .	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
5	RA1, RA2, RA3, RA6	Visualización Científica	1,5 semanas
Contenidos		Indicador de logro	
5.1. Visualización de Campos Escalares y Vectoriales. 5.2. Visualización de data volumétrica: Algoritmo del Cubo Marchante.		La/el estudiante: <ol style="list-style-type: none"> <li>1. Analiza técnicas de visualización, considerando su función y características.</li> <li>2. Genera gráficas de visualización de campos escalares y vectoriales en 2D y 3D, utilizando curvas de nivel, superficies, imágenes, diagramas de flujo y de flechas, etc.</li> <li>3. Identifica y aplica técnicas de visualización, según el tipo de dato y las funcionalidades de cada técnica.</li> <li>4. Analiza la aplicabilidad del algoritmo del cubo marchante en la generación de isosuperficies, considerando sus ventajas para la aplicación de técnicas de sombreado.</li> </ol>	
Bibliografía de la unidad		(5) Apuntes del curso. (9) Capítulo 26: <i>Visualization</i> . (4) <i>Marching Cubes: A high Resolution 3D Surface Construction Algorithm</i> .	

### E. Estrategias de enseñanza - aprendizaje:

El curso considera las siguientes estrategias:

- **Clases expositivas interactivas** que incluyen presentación de los temas de estudio, con los principales aprendizajes de cada sesión y el análisis de ejemplos en código.
- **Resolución de problemas** donde los y las estudiantes desarrollan aplicaciones gráficas interactivas que modelan, simulan y visualizan computacionalmente problemas aplicados que involucren datos, geometrías y escenas complejas en 2D y 3D. La implementación práctica se realiza programando la GPU con la librería de bajo nivel *OpenGL*.

### F. Estrategias de evaluación:

La propuesta de evaluación se informará al inicio de cada semestre, indicando cantidad de evaluaciones y tipo, así como la ponderación para cada actividad evaluativa.

Para este curso se consideran las siguientes estrategias evaluativas:

- **Controles** (2).
- **Tareas computacionales** individuales y/o grupales, con su respectivo reporte (mínimo 3 tareas).
- **Presentación oral** de una tarea computacional (solo 1), evidenciando manejo de conceptos, procesos y técnicas de modelación geométrica y computación gráfica).

- **Examen (1):** se evalúan de manera integral los aprendizajes del curso.

*En algunos de los controles o tareas se incluirá la evaluación de lecturas complementarias específicas, lo que se informará con antelación a los estudiantes.*

## G. Recursos bibliográficos:

### Bibliografía obligatoria:

- (1) Lengyel, E. (2012). *Mathematics for 3D Game Programming and Computer Graphics*. Course Technology, Cengage Learning: third edition.
- (2) Sellers G., Wright, Jr. R. S., Haemel N. (2016). *OpenGL SuperBible. Comprehensive Tutorial and Reference*. Addison Wesley: seventh edition.
- (3) Mukundan R. (2012). *Advance Methods in Computer Graphics with examples in OpenGL*. Springer.
- (4) Lorensen W. E., Cline H. E. (1987). *Marching Cubes: A high Resolution 3D Surface Construction Algorithm*. Computer Graphics, ACM, 21(4): 164-169.
- (5) Apuntes del curso, disponibles en Material Docente de U-Cursos.

### Bibliografía complementaria:

- (6) Kessenich J., Sellers G., Shreiner D. (2017). *OpenGL Programming Guide. The Official Guide to Learning OpenGL. Version 4.5 with SPIR-V*. Addison Wesley: ninth edition.
- (7) Angel E., Shreiner D. (2014). *Interactive Computer Graphics, a top-down approach with WebGL*. Pearson: seventh edition.
- (8) Hearn D., Baker M. P., Carithers W. R. (2014). *Computer Graphics with OpenGL*. Pearson Prentice Hall: fourth edition.
- (9) Marschner S., Shirley P. (2016). *Fundamentals of Computer Graphics*. CRC Press Taylor & Francis Group: fourth edition.
- (10) Foley J., van Dam A., Feiner S., Hughes J. (1995). *Computer Graphics: Principles and Practice in C*. Addison-Wesley Professional: second edition.
- (11) Lethbridge T., Laganière R. (2005). *Object-Oriented Software Engineering. Practical software development using UML and Java*. Mc Graw Hill.

## H. Datos generales sobre elaboración y vigencia del programa de curso:

Vigencia desde:	Otoño, 2021
Elaborado por:	Nancy Hitschfeld, Daniel Calderón
Validado por:	Revisión y validación académico par: Iván Sipirán. Validación de CTD Computación noviembre de 2020
Revisado por:	Área de Gestión Curricular