

**Universidad Católica San Pablo (UCSP)**  
**Escuela Profesional de**  
**Ciencia de la Computación**  
**SILABO**



**CS251. Computación Gráfica (Obligatorio)**

**1. Información general**

1.1 Escuela	:	Ciencia de la Computación
1.2 Curso	:	CS251. Computación Gráfica
1.3 Semestre	:	7 <sup>mo</sup> Semestre.
1.4 Prerrequisitos	:	<ul style="list-style-type: none"><li>• CS312. Estructuras de Datos Avanzadas. (6<sup>to</sup> Sem)</li><li>• MA306. Análisis Numérico. (5<sup>to</sup> Sem)</li></ul>
1.5 Condición	:	Obligatorio
1.6 Modalidad de aprendizaje	:	Presencial
1.7 horas	:	2 HT; 4 HP;
1.8 Créditos	:	4
1.9 Plan	:	Plan Curricular 2016

**2. Profesores**

**Titular**

- Manuel Loaiza Fernandez <meloaza@ucsp.edu.pe>
  - Doctor en Informática, Pontificia Universidad Católica do Rio de Janeiro (PUC-RIO), Brasil, 2009.
  - Master en Informática, Pontificia Universidad Católica do Rio de Janeiro (PUC-RIO), Brasil, 2005.

**3. Fundamentación del curso**

Ofrece una introducción para el área de Computación Gráfica, la cual es una parte importante dentro de Ciencias de la Computación. El propósito de este curso es investigar los principios, técnicas y herramientas fundamentales para esta área.

**4. Resumen**

1. Conceptos Fundamentales 2. Rendering Básico 3. Programación de Sistemas Interactivos 4. Modelado Geométrico 5. Renderizado Avanzado 6. Animación por computadora

**5. Objetivos Generales**

- Acercar al alumno a conceptos y técnicas usados en aplicaciones gráficas 3-D complejas.
- Dar al alumno las herramientas necesarias para determinar que software gráfico y que plataforma son los más adecuados para desarrollar una aplicación específica.

**6. Contribución a los resultados (Outcomes)**

Esta disciplina contribuye al logro de los siguientes resultados de la carrera:

- 1) S.O. Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. **(Usar)**
- 6) S.O. Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. **(Usar)**

## 7. Contenido

### UNIDAD 1: Conceptos Fundamentales (6)

#### Competencias:

#### Contenido

- 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.

#### Objetivos Generales

- 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:** Hearn and Baker (1990)

UNIDAD 2: Rendering Básico (12)	
Competencias:	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> <li>• Renderizado en la naturaleza, por ejemplo, la emisión y dispersión de la luz y su relación con la integración numérica.</li> <li>• Renderizado Forward and Backward (i.e., <i>ray-casting</i> y rasterización)</li> <li>• Radiometría básica, triángulos similares y modelos de proyecciones</li> <li>• Afinamiento y Transformaciones de Sistemas de coordenadas</li> <li>• <i>Ray tracing</i></li> <li>• Visibilidad y oclusión, incluyendo soluciones a este problema, como el almacenamiento en búfer de profundidad, algoritmo del pintor, y el trazado de rayos.</li> <li>• Rasterización triangular simple.</li> <li>• Renderización con una API basada en shader.</li> <li>• Aplicación de la representación de estructuras de datos espaciales.</li> <li>• Muestreo y anti-aliasing.</li> <li>• Renderizado Forward and Backward (i.e., <i>ray-casting</i> y rasterización)</li> </ul>	<ul style="list-style-type: none"> <li>• 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]</li> <li>• Describir la tubería básica gráficos y cómo el factor de representación va hacia adelante y atrás en esta [Familiarizarse]</li> <li>• Crear un programa para visualizar modelos 3D de imágenes gráficas simples [Usar]</li> <li>• Obtener puntos en 2-dimensiones y 3-dimensiones por aplicación de transformaciones afín [Usar]</li> <li>• 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]</li> <li>• Contrastar la renderización hacia adelante <i>forward</i> y hacia atrás <i>backward</i> [Evaluar]</li> <li>• Explicar el concepto y las aplicaciones de mapeo de texturas, muestreo y el <i>anti-aliasing</i> [Familiarizarse]</li> <li>• Explicar la dualidad de rastreo de rayos/rasterización para el problema de visibilidad [Familiarizarse]</li> <li>• Implementar un sencillo renderizador en tiempo real utilizando una API de rasterización (por ejemplo, OpenGL) utilizando buffers de vértices y <i>shaders</i> [Usar]</li> <li>• Calcular las necesidades de espacio en base a la resolución y codificación de color [Evaluar]</li> <li>• Calcular los requisitos de tiempo sobre la base de las frecuencias de actualización, técnicas de rasterización [Evaluar]</li> </ul>
Lecturas: Marschner2016	

<b>UNIDAD 3: Programación de Sistemas Interactivos (2)</b>	
<b>Competencias:</b>	
<b>Contenido</b>	<b>Objetivos Generales</b>
<ul style="list-style-type: none"> <li>• Manejo de eventos e interacción de usuario.</li> <li>• Enfoques para el diseño, implementación y evaluación de la interacción sin mouse <ul style="list-style-type: none"> <li>– Interfaces táctiles y multitáctiles.</li> <li>– Interfaces compartidas, incorporadas y grandes</li> <li>– Nuevas modalidades de entrada (tales como datos de sensores y localización)</li> <li>– Nuevas ventanas, por ejemplo, iPhone, Android</li> <li>– Reconocimiento de voz y procesamiento del lenguaje natural</li> <li>– Interfaces utilizables y tangibles</li> <li>– Interacción persuasiva y emoción</li> <li>– Tecnologías de interacción ubicuas y contextuales (Ubicomp)</li> <li>– Inferencia bayesiana (por ejemplo, texto predictivo, orientación guiada)</li> <li>– Visualización e interacción de ambiente / periféricos</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Discute las ventajas (y desventajas) de las interfaces no basadas en ratón [Evaluar]</li> </ul>
<b>Lecturas: Marschner2016</b>	

<b>UNIDAD 4: Modelado Geométrico (15)</b>	
<b>Competencias:</b>	
<b>Contenido</b>	<b>Objetivos Generales</b>
<ul style="list-style-type: none"> <li>• Operaciones geométricas básicas como cálculo de intersección y pruebas de proximidad.</li> <li>• Volúmenes, voxels y representaciones basadas en puntos.</li> <li>• Curvas polinomiales y Superficies paramétricas.</li> <li>• Representación implícita de curvas y superficies.</li> <li>• 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.</li> <li>• 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.</li> <li>• Técnicas de subdivisión espacial.</li> <li>• Modelos procedimentales como fractales, modelamiento generativo y sistemas L.</li> <li>• Modelos deformables de forma libre y elásticamente deformables.</li> <li>• Subdivisión de superficies.</li> <li>• Modelado multiresolución.</li> <li>• Reconstrucción.</li> <li>• Representación de Geometría Sólida Constructiva (GSC)</li> </ul>	<ul style="list-style-type: none"> <li>• Representar curvas y superficies utilizando formas tanto implícitas y paramétricas [Usar]</li> <li>• Crear modelos poliédrico simples por teselación de superficies [Usar]</li> <li>• Generar una representación de malla de una superficie implícita [Usar]</li> <li>• Generar una malla de un conjunto de puntos adquiridos por un scanner laser [Usar]</li> <li>• Construct modelos de geometría sólida constructiva a partir de simples primitivas, tales como cubos y superficies cuádricas [Usar]</li> <li>• Contrastar métodos de modelización con respecto a espacio y tiempo de complejidad y calidad de imagen [Evaluar]</li> </ul>
<b>Lecturas: Marschner2016</b>	

UNIDAD 5: Renderizado Avanzado (6)	
Competencias:	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> <li>• Tiempo (desenfoque de movimiento), la posición del objetivo (enfoco), y la frecuencia continua (color) y su impacto en la representación.</li> <li>• Mapeo de Sombras.</li> <li>• Selectiva de oclusión.</li> <li>• Dispersión de la Superficie.</li> <li>• Renderizado no fotorealístico.</li> <li>• Arquitectura del GPU.</li> <li>• Sistemas visuales humanos incluida la adaptación a la luz, la sensibilidad al ruido, y la fusión de parpadeo.</li> </ul>	<ul style="list-style-type: none"> <li>• Demostrar como un algoritmo calcula una solución a la ecuación de renderización [Evaluar]</li> <li>• Demostrar las propiedades de un algoritmo de renderización, por ejemplo, completo, consistente, e imparcial [Evaluar]</li> <li>• 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]</li> <li>• Discutir como una técnica artística particular puede ser implementada en un renderizador [Familiarizarse]</li> <li>• Explicar como reconocer las técnicas gráficas usadas para crear una imagen en particular [Familiarizarse]</li> </ul>
<b>Lecturas: Marschner2016</b>	

UNIDAD 6: Animación por computadora (4)	
Competencias:	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> <li>• Cinemática directa e inversa.</li> <li>• Detección de colisiones y respuesta.</li> <li>• Animación procedimental empleando ruido, reglas (boids/crowds) y sistemas de partículas.</li> <li>• Algoritmos Skinning.</li> <li>• 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.</li> <li>• Animación de Cuadros Principales</li> <li>• Splines</li> <li>• Estructuras de datos para rotaciones, como cuaterniones.</li> <li>• Animación de Cámara.</li> <li>• Captura de Movimiento.</li> </ul>	<ul style="list-style-type: none"> <li>• Calcular la localización y orientación de partes de un modelo usando un enfoque de cinemática hacia delante [Usar]</li> <li>• Implementar el método de interpolación <i>spline</i> para producir las posiciones y orientaciones en medio [Usar]</li> <li>• Implementar algoritmos para el modelamiento físico de partículas dinámicas usando simplemente la mecánica de Newton, por ejemplo Witkin &amp; Kass, serpientes y gusanos, Euler simpléctica, Stormer/Verlet, o métodos de punto medio de Euler [Usar]</li> <li>• 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]</li> <li>• Usar el software de animación común para construir formas orgánicas simples usando <i>metaball</i> y el esqueleto [Usar]</li> </ul>
<b>Lecturas: Marschner2016</b>	

## 8. Metodología

1. El profesor del curso presentará clases teóricas de los temas señalados en el programa propiciando la intervención de los alumnos.
2. El profesor del curso presentará demostraciones para fundamentar clases teóricas.

3. El profesor y los alumnos realizarán prácticas

4. Los alumnos deberán asistir a clase habiendo leído lo que el profesor va a presentar. De esta manera se facilitará la comprensión y los estudiantes estarán en mejores condiciones de hacer consultas en clase.

**9. Evaluar 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.

**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.

**Sistema de Evaluación:**

La nota final se obtiene a través de:

EVALUACIONES PERMANENTES	EVALUACIONES
<b>Evaluación Permanente 1</b> : 20 %	<b>Evaluación Parcial</b> : 20 %
<b>Evaluación Permanente 2</b> : 40 %	<b>Evaluación Final</b> : 20 %
60%	40%

Donde:

Evaluación Permanente: Comprende trabajos grupales, participación activa en clase, test de ejercicios.

- Permanente 1 (Semanas 1 - 9)
- Permanente 2 (Semanas 10 - 17)

Para aprobar el curso, el alumno debe obtener 11.5 o más en la nota final.

## References

Hearn, Donald and Pauline Baker (1990). *Computer Graphics in C*. Prentice Hall.