

Universidad Católica San Pablo
Facultad de Ingeniería y Computación
Escuela Profesional de
Ciencia de la Computación
SILABO



CS356. Programación de Video Juegos (Electivo)

2018-I

1. DATOS GENERALES

1.1 CARRERA PROFESIONAL	:	Ciencia de la Computación
1.2 ASIGNATURA	:	CS356. Programación de Video Juegos
1.3 SEMESTRE ACADÉMICO	:	10 ^{mo} Semestre.
1.4 PREREQUISITO(S)	:	CS261T. Inteligencia Artificial. (7 ^{mo} Sem) , CS355. Tópicos en Computación Gráfica. (9 ^{no} Sem) , CS250W. Interacción Humano Computador. (7 ^{mo} Sem)
1.5 CARÁCTER	:	Electivo
1.6 HORAS	:	2 HT; 2 HP; 2 HL;
1.7 CRÉDITOS	:	4

2. DOCENTE

Dr. Juan Carlos Gutiérrez Cáceres

- Dr. Ciencia de la Computación, Universidad Nacional de San Agustín, Perú, 2013.
- Mag. Ciencia de la Computación, ICMC-USP, Brasil, 2003.
- Prof. Ingeniero de Sistemas, Universidad Nacional de San Agustín, Perú, .

3. FUNDAMENTACIÓN DEL CURSO

La industria de los video juegos ha tenido un crecimiento exponencial en las últimas dos décadas y puede ser aplicada a diversas áreas del conocimiento humano.

El potencial que ofrece esta área para un egresado es muy amplio y como tal se considera como un área crítica para el desarrollo de la industria del software.

4. SUMILLA

1. GV/Técnicas Avanzadas.2. GV/Visualización.3. HC/Fundamentos de la Interacción Hombre-Computador (HCI)4. GV/Rendering Avanzado.5. GV/Programación de motores de juegos.

5. OBJETIVO GENERAL

- Que el alumno conozca las técnicas fundamentales que permiten la creación de video juegos.
- Que el alumno construya videos juegos de complejidad media incorporando conceptos de Inteligencia Artificial.

6. CONTRIBUCIÓN A LA FORMACIÓN PROFESIONAL Y FORMACIÓN GENERAL

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

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. [Nivel Bloom: 4]
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. [Nivel Bloom: 4]
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. [Nivel Bloom: 4]
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. [Nivel Bloom: 4]

7. CONTENIDOS

UNIDAD 1: GV/Técnicas Avanzadas.(8 horas)

Nivel Bloom: 3

OBJETIVO GENERAL

- Describir las técnicas identificadas en esta sección.
- Explicar como reconocer las técnicas gráficas usadas para crear una imagen particular.
- Implementar cualquiera de las técnicas gráficas especificadas utilizando un sistema gráfico primitivo a nivel de pixel.
- Utilizar un software de animación común para construir una forma orgánica simple usando metabolas y esqueletos.

CONTENIDO

- Cuantización de colores.
- Conversión de primitivas 2D de escaneo, diferenciación hacia adelante (*forward differencing*).
- Poligonización (*tessellation*) de superficies curvas.
- Métodos de remoción de superficies ocultas.
- *Z-buffer* y *frame buffer*, canales de color (un canal para la opacidad).
- Técnicas de modelamiento de geometría avanzada.

Lecturas: [Foley and van Dam, 1990], [Hearn and Baker, 1994]

UNIDAD 2: GV/Visualización.(4 horas)

Nivel Bloom: 3

OBJETIVO GENERAL

- Describir los algoritmos básicos detrás de la visualización de escalares y vectores.
- Comparar los algoritmos en términos de precisión y desempeño.
- Emplear la teoría disponible para explicar los efectos de las operaciones de visualización.
- Describir el impacto de la presentación y la interacción del usuario en exploración.

CONTENIDO

- Vista básica y funciones de interrogación para visualización.
- Visualización de campos de vectores, tensores y flujo de datos.
- Visualización de campos escalares o de campos de altura: iso-superficies usando el método *marching cubes*.
- *Rendering* volumétrico directo: *ray-casting*, funciones de transferencia, segmentación, hardware.
- Visualización de información: métodos de coordenadas paralelas y proyección.

Lecturas: [Foley and van Dam, 1990], [Hearn and Baker, 1994]

UNIDAD 3: HC/Fundamentos de la Interacción Hombre-Computador (HCI)(4 horas)**Nivel Bloom: 4****OBJETIVO GENERAL****CONTENIDO**

- Discutir las razones por las cuales es importante el desarrollo de software centrado en el usuario.
- Explicar porqué los modelos humanos individuales y los modelos sociales son importantes a la hora de diseñar la Interacción Humano-Computador.
- Definir y ejemplificar procesos centrados en el usuario que explícitamente evidencien que las expectativas del desarrollador y sus conocimientos previos son muy diferentes de las de los usuarios.
- Describir y ejemplificar casos en los que un diseño centrado en el usuario puede fallar.
- Explicar los distintos procesos aplicados a la definición de interfaces para diferentes contextos.
- Considerar el rol de la hipótesis y las diferencias entre resultados experimentales versus correlaciones, al utilizar métricas de evaluación de la Interacción Humano-Computador.
- Escoger entre métodos de evaluación cualitativos y cuantitativos para una evaluación dada.
- Usar un vocabulario especializado para referirse a la interacción humana con el software: potencialidad percible, modelo conceptual, modelo mental, metáforas, diseño de la interacción, retroalimentación, etc.
- Ejemplificar cómo determinados símbolos, íconos, palabras o colores pueden tener diferentes interpretaciones en dos culturas humanas distintas o incluso entre una cultura y alguna de sus subculturas.
- Estar preparado para describir al menos un estándar nacional o internacional de diseño estándar de interfases.

- Relevancia de la Interacción Hombre-Computador (HCI). ¿Por qué el estudio de la interacción entre las personas y la tecnología es vital para el desarrollo de sistemas más usables y aceptables?
- Terminología clave en la Interacción Humano-Computador: usabilidad, accesibilidad, diseño para todos, diseño inclusivo, acceso universal, diseño de sistemas centrados en el usuario (UCSD).
- Contextos de Interacción Humano-Computador: equipos (PC's, equipos industriales, dispositivos de consumo, dispositivos móviles) y aplicaciones (de negocios, en tiempo real, web, sistemas colaborativos, juegos, etc.).
- Proceso de desarrollo centrado en el usuario (UCSD): foco temprano en los usuarios, pruebas empíricas, diseño iterativo.
- Categorías de evaluación: utilidad, eficiencia, usabilidad, facilidad de aprendizaje, satisfacción del usuario.
- Consideraciones psicológicas para el modelamiento de usuarios y la evaluación de la Interacción Humano-Computador (atención, percepción y reconocimiento, memoria de corto y largo plazo, movimiento, abstracción, y procesamiento cognitivo).
- Aspectos sociales que influyen en el diseño y en el uso de Interfaces Humano-Computador: cultura, comunicación y organizaciones.
- Adaptación a la diversidad humana, incluyendo diseño y accesibilidad universal, diseño para múltiples contextos culturales y lingüísticos.
- Los errores más frecuentes en el diseño de interfaces.
- Estándares para el diseño de interfaces de sistemas interactivos (reglas y guías de diseño de organismos reguladores, fabricantes de software, y estilos corporativos).

Lecturas: [Baecker et al., 2000]

UNIDAD 4: GV/ <i>Rendering</i> Avanzado.(10 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Describir varias ecuaciones de transporte al detalle, resaltando sus efectos. ▪ Describir algoritmos eficientes para radiocidad y compararlos de acuerdo a sus desempeños algorítmicos y de exactitud. ▪ Describir el impacto de los esquemas de mallas. ▪ Explicar las técnicas de <i>rendering</i> basadas en imágenes, campos de luz y tópicos asociados. 	<ul style="list-style-type: none"> ▪ Ecuaciones de transporte. ▪ Algoritmos de trazo de rayos (<i>ray tracing</i>). ▪ <i>Photon tracing</i>. ▪ Radiocidad para el cálculo de la iluminación global, factores de forma. ▪ Métodos eficientes para iluminación global. ▪ Métodos Monte Carlo para iluminación global. ▪ <i>Rendering</i> basado en imágenes, visión panorámica, modelaje de la función plenóptica. ▪ <i>Rendering</i> de fenómenos complejos naturales. ▪ <i>Rendering</i> no fotorealístico.
Lecturas: [Foley and van Dam, 1990], [Hearn and Baker, 1994]	

UNIDAD 5: GV/Programación de motores de juegos.(26 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Estar informado del amplio rango de posibilidades para motores de juegos incluyendo su potencial y sus limitaciones. ▪ Usar un motor de juegos para construir un juego simple. 	<ul style="list-style-type: none"> ▪ La naturaleza de los motores de juegos (como un entorno de desarrollo integrado) y su propósito. ▪ Soporte de hardware incluyendo uso de paralelismo, desempeño, dispositivos de entrada. ▪ Componentes típicos incluyendo renderización 3D y soporte para gráficos en tiempo real e interacción así como simulación física, detección de colisiones, sonido, inteligencia artificial renderización de terreno.
Lecturas: [Penton, 2002], [Llopis, 2006], [Sherrod, 2007]	

8. METODOLOGÍA
<p>El profesor del curso presentará clases teóricas de los temas señalados en el programa propiciando la intervención de los alumnos.</p> <p>El profesor del curso presentará demostraciones para fundamentar clases teóricas.</p> <p>El profesor y los alumnos realizarán prácticas</p> <p>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.</p>

9. EVALUACIONES

Evaluación Permanente 1 : 20 %

Examen Parcial : 30 %

Evaluación Permanente 2 : 20 %

Examen Final : 30 %

Referencias

- [Baecker et al., 2000] Baecker, R., Buxton, W., and Grudin, J. (2000). *Readings in Human-Computer Interaction: Toward the Year 2000*. The Morgan Kaufmann Series in Interactive Technologies. Morgan Kaufmann, 2nd edition edition.
- [Foley and van Dam, 1990] Foley, J. and van Dam, A. (1990). *Computer Graphics: Principles and Practice*. Addison-Wesley.
- [Hearn and Baker, 1994] Hearn, D. and Baker, M. P. (1994). *Computer Graphics in C*. Prentice Hall.
- [Llopis, 2006] Llopis, N. (2006). *C++ For Game Programmers*. Charles River Media, 2 edition edition.
- [Penton, 2002] Penton, R. (2002). *Data Structures for Game Programmers*. Muska & Lipman/Premier-Trade, 1st edition. Premier Press Game Development.
- [Sherrod, 2007] Sherrod, A. (2007). *Data Structures and Algorithms for Game Developers*. Charles River Media, 1 edition edition.