

## PROGRAMA DE CURSO

### ARQUITECTURA DE MOTORES DE JUEGOS

#### A. Antecedentes generales del curso:

Departamento	Ciencias de la Computación					
Nombre del curso	Arquitectura de Motores de Juegos	Código	CC5512	Créditos	6	
Nombre del curso en inglés	<i>Game Engine Architecture</i>					
Horas semanales	Docencia	3	Auxiliares	1	Trabajo personal	6
Carácter del curso	Obligatorio			Electivo	X	
Requisitos	CC3501: Modelación y computación gráfica para ingenieros, CC3301: Programación de software de sistemas, CC3002: Metodologías de diseño y programación					

#### B. Propósito del curso:

El curso busca que la/el estudiante pueda, por un lado, comprender cuáles son las componentes técnicas de un videojuego y, por otro lado, analizar y extender motores de videojuegos.

Un motor de videojuegos es una componente de software que permite reutilizar componentes técnicas, de alta complejidad en su desarrollo, para la elaboración de distintos videojuegos o aplicaciones de requerimientos similares, como simulaciones gráficas interactivas en tiempo real.

Este curso explora la arquitectura de un motor de videojuegos en 2D y 3D, profundizando en la descripción y decisiones de diseño tanto del motor mismo como de sus subsistemas típicos: motor de renderizado, sistema de audio, detección de colisiones, simulaciones físicas, sistema de animaciones, sistema de networking y sistemas de inteligencia artificial. Todo esto, manteniendo una separación clara entre el motor de videojuegos y la implementación del videojuego mismo.

El propósito del curso es que la/el estudiante comprenda la arquitectura de este tipo de software, y aprenda a modificarlo y a extenderlo, según sea requerido. Este conocimiento se vuelve clave al momento de trabajar en la industria especializada.

El curso utiliza un lenguaje de programación cercano al hardware en conjunto con varias bibliotecas de terceros. Esto con el fin de comprender y analizar las problemáticas de este tipo de software, junto con un acercamiento práctico a las plataformas utilizadas en industria.

Al término del curso, la/el estudiante podrá analizar técnicamente un motor de videojuegos; identificando potencialidades y limitantes; habrá implementado videojuegos simples, con y sin motor; podrá extender un motor de videojuegos y utilizar subsistemas especializados del mismo.

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, CE2, CE5, CE6, CE8	RA1: Programa un videojuego simple, en lenguaje cercano al hardware, utilizando bibliotecas de terceros que permitan manipular componentes especializadas de hardware (gráficos, audio, entradas de usuario) y software (física, matemática, inteligencia artificial, etc.)
CE1, CE2, CE5, CE6, CE8	RA2: Programa un videojuego simple utilizando un motor de videojuegos de código disponible, con especial énfasis en su arquitectura y uso de sus distintos sistemas.
CE2, CE6	RA3: Analiza y discute la arquitectura y el funcionamiento de motores de videojuegos de terceros.
CE1, CE5	RA4: Aplica los fundamentos teóricos y técnicos de los subsistemas involucrados en un motor de videojuegos tradicional (gráfico, audio, física, etc.), a fin de desarrollar programas propios, logrando identificar capacidades y limitaciones de cada subsistema.
CE1, CE2, CE5, CE6, CE8	RA5: Programa una extensión para un motor de videojuegos de código abierto, permitiendo ejecutar funcionalidades de mayor complejidad no cubiertas nativamente por el motor de videojuegos utilizado.
CE8	RA6: Utiliza herramientas de desarrollo de software como IDE, control de versiones, depurador, perfilador de tiempo y memoria, en el desarrollo de aplicaciones de videojuegos, con el fin de organizar su proceso de desarrollo y de respaldar las decisiones técnicas adoptadas en las distintas implementaciones.
Competencias genéricas	Resultados de aprendizaje
CG1	RA7: Elabora reportes y presentaciones breves donde se transmite con claridad el estado de desarrollo de sus tareas computacionales, describiendo con especial énfasis las decisiones técnicas que se tomaron.
CG2	RA8: Investiga en distintas fuentes de información teórica y técnica relevante a las distintas tecnologías de desarrollo de videojuegos, aplicando activamente estos conocimientos en el progreso de sus tareas computacionales.



#### D. Unidades temáticas:

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
1	RA3, RA8	Introducción a Motores de Videojuegos y sus aplicaciones	0,5 semanas
Contenidos		Indicador de logro	
1.1. Concepto de Motor de Videojuegos.		La/el estudiante:  1. Analiza el producto de software “Motor de Videojuegos”, considerando sus aplicaciones, evolución histórica y su relevancia en la actualidad.	
Bibliografía de la unidad		(1) Capítulo 1: Introduction	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
2	RA1, RA6, RA7, RA8	Programación de Videojuegos en Bajo Nivel	4 semanas
Contenidos		Indicador de logro	
2.1. C++, Clases y Objetos, Templates. 2.2. References, Pointers & Smart Pointers. 2.3. STL Containers & Allocators. 2.4. Organización de un proyecto, Sistema de construcción de software, CMake. 2.5. Gráficos 2D, Audio 2D y Dispositivos de Interacción Humana. 2.6. Herramientas de desarrollo: IDE & Debugger.		La/el estudiante:  1. Implementa un videojuego simple directamente en lenguaje cercano al hardware, sin utilizar un motor de videojuegos.  2. Documenta los resultados de su trabajo en un breve reporte técnico, ajustando su escrito a criterios de claridad y precisión frente a lo expuesto.  <i>*La complejidad del videojuego se estima similar a algunos videojuegos clásicos de Atari. Ejemplos: Breakout, Pong, Space Invaders, etc.</i>	
Bibliografía de la unidad		(2) Capítulo 5: Designing with Objects (2) Capítulo 7: Memory Management (2) Capítulo 18: Standard Library Containers (1) Capítulo 2.2: Compilers, Linkers and IDEs (1) Capítulo 9: Human Interface Devices (1) Capítulos 14.1, 14.2, 14.3: Audio	



Número	RA al que tributa	Nombre de la unidad	Duración en semanas
3	RA3, RA8	Modelos de Objetos	2 semanas
Contenidos		Indicador de logro	
3.1. Game Objects en tiempo de ejecución. 3.2. Estrategias de implementación: Jerarquía Monolítica, Componentes y Sistemas Entidad-Componente. 3.3. Game Objects en el Editor. 3.4. Mapas y Serialización.		La/el estudiante:  1. Analiza y discute distintas estrategias y problemáticas de los temas tratados en cátedra, centrándose en la arquitectura de uno o más motores de videojuegos disponibles en la industria.	
Bibliografía de la unidad		(1) Capítulo 15: Introduction to Gameplay Systems (1) Capítulo 16: Runtime Gameplay Foundation Systems	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
4	RA2, RA3, RA6, RA7, RA8	Módulos de Soporte	3,5 semanas
Contenidos		Indicador de logro	
4.1. Módulos de un Motor de Videojuegos. 4.2. Administrador de Recursos. 4.3. Scripting. 4.4. Manejo del tiempo e implementación básica de un perfilador. 4.5. Hardware de videojuegos y ejecución concurrente de un motor.		La/el estudiante:  1. Implementa un videojuego simple utilizando un Motor de Videojuegos de código disponible (Ejemplos: O3DE, Godot, Unreal Engine, etc.), para esto, utiliza tanto el lenguaje nativo como el lenguaje de scripting provistos por el motor.  2. Redacta un breve reporte técnico donde analiza y documenta su trabajo, considerando los aspectos vistos en cátedra respecto del motor utilizado.  <i>*La complejidad del videojuego se estima similar a algunos videojuegos clásicos de Atari. Ejemplos: Breakout, Pong, Space Invaders, etc.</i>	
Bibliografía de la unidad		(1) Capítulo 6.1: Subsystem Start-Up and Shutdown. (1) Capítulo 7.2: The Resource Manager. (1) Capítulo 16.9: Scripting. (1) Capítulo 8: The Game Loop and Real-Time Simulation.	



Número	RA al que tributa	Nombre de la unidad	Duración en semanas
5	RA2, RA3, RA4, RA5, RA6, RA7, RA8	Subsistemas Especializados	5 semanas
Contenidos		Indicador de logro	
<p>5.1. Detección de Colisiones: Primitivas de Colisión, Estructuras Aceleradoras, Ray Casting, Detección Continua de Colisiones (CCD).</p> <p>5.2. Renderizado de Audio: Espacialización, Distorsiones y Sistemas Dedicados.</p> <p>5.3. Sistemas de Inteligencia Artificial: Agentes Inteligentes, Planificación vía búsqueda, Path Planning, Máquinas de Estado, Árboles de Decisión, Árboles de Conducta, Goal-Oriented Action Planning.</p> <p>5.4. Renderizado Gráfico: Pipeline 2D/3D, Determinación de Visibilidad, Grafos de Escena y Sistemas Dedicados como Texto y Efectos Ambientales.</p> <p>5.5. Sistema de Animación: Clips de Animación, Animación vía Sprites y Esqueletos.</p> <p>5.6. Simulaciones Físicas: Integración Numérica, Dinámica Lineal y Rotacional.</p> <p>5.7. Networking en Videojuegos: Estrategias de Videojuegos en la Red y Replicación de Game Objects.</p>		<p>La/El estudiante:</p> <ol style="list-style-type: none"> <li>1. Programa una extensión para un motor de videojuegos de código disponible que añada funcionalidades no disponibles nativamente en el motor.</li> <li>2. Modifica/extiende un videojuego, haciendo uso efectivo de la extensión implementada.</li> <li>3. Implementa desarrollos técnicos que ejerciten al menos dos de los subsistemas dígame, detección de colisiones, renderizado de audio, sistemas de inteligencia artificial, renderizado gráfico, animaciones, simulaciones físicas y/o networking.</li> <li>4. Aplica perfiladores de tiempo y memoria para evaluar el rendimiento de un videojuego.</li> <li>5. Redacta un breve reporte donde documenta su trabajo, y justifica las decisiones técnicas de este.</li> <li>6. Elabora argumentos técnicos basados tanto en respaldo teórico, como en evidencia empírica proporcionada por las herramientas de desarrollo, como perfiladores de tiempo y memoria.</li> <li>7. Presenta oralmente su trabajo demostrando un dominio técnico apropiado y un buen uso de los conceptos relevantes del curso.</li> </ol>	
Bibliografía de la unidad		<p>(3) Capítulo 12: Collision Detection; 13: Generating Contacts</p> <p>(1) Capítulo 14.4, 14.5, 14.6: Audio</p> <p>(4) Capítulos: 1.2: Model of Game AI; 4: Pathfinding; 5: Decision Making; 2.4: The AI Engine</p> <p>(1) Capítulo 11: The Rendering Engine</p> <p>(1) Capítulo 12: Animation Systems</p> <p>(1) Capítulo 13: Collision and Rigid Body Dynamics</p> <p>(5) Capítulos: 1: Overview of Networked Games; 4: Object Serialization; 5: Object Replication</p>	



### E. Estrategias de enseñanza-aprendizaje:

El curso considera las siguientes estrategias (se listan el tipo de estrategias)

- Cátedras
- Desarrollo y discusión de tareas computacionales
- Aprendizaje basado en proyectos
- Estudio de casos

### F. Estrategias de evaluación:

Al inicio de cada semestre, el académico o académica informará a los y las estudiantes sobre los tipos de evaluaciones, así como las ponderaciones correspondientes.

Tipo de evaluación	RA asociado a la evaluación
▪ Tareas computacionales individuales	Evalúa RA1, RA2, RA4, RA5, RA6, RA8
▪ Evaluación teórica	Evalúa RA3
▪ Reportes técnicos	Evalúa RA3, RA7, RA8

### G. Recursos bibliográficos:

#### Bibliografía obligatoria:

- [1] Gregory, J. (2019). *Game Engine Architecture*. CRC Press, Taylor & Francis Group: third edition.

#### Bibliografía Complementaria:

- [2] Gregoire, M. (2021). *Professional C++*. WROX: third edition.  
 [3] Millington, I. (2010). *Game Physics Engine Development*. CRC Press, Taylor & Francis Group: second edition.  
 [4] Millington, I. (2019). *AI for Games*. CRC Press, Taylor & Francis Group: third edition.  
 [5] Glazer, J.; Madhav S. (2016). *Multiplayer Game Programming*. Addison Wesley.

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

Vigencia desde:	Primavera 2024
Elaborado por:	Daniel Calderón S.
Validado por:	Académico Nelson Baloian T., CTD 05/06/24
Revisado por:	Área Gestión Curricular