

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					
Nombre del curso en inglés	<i>Game Engine Architecture</i>					
Código	CC5512		Créditos	6		
Horas semanales	Docencia	3	Auxiliares	0	Trabajo personal	7
Carácter del curso	Obligatorio			Electivo de Licenciatura	X	
Requisitos	CC3501 Modelación y Computación Gráfica para Ingenieros CC4302 Sistemas Operativos CC4401 Ingeniería de Software					

B. Propósito del curso:

Distintas aplicaciones encapsuladas en el concepto de videojuegos poseen requerimientos similares, tales como presentar un mundo virtual tridimensional e interactivo, simulando elementos del mundo real tales como efectos auditivos y comportamientos según leyes de la física, o predeterminados según ciertas reglas. Las tecnologías desarrolladas con los videojuegos han evolucionado desde estar inseparablemente en el videojuego mismo, a sistemas completamente independientes que nos permiten generar una serie de aplicaciones similares, pero con diferencias importantes que en la práctica se traducen en un producto distinto y nuevo. Es así como la industria ha desarrollado estas capas intermedias de software, llamadas motores de videojuegos, las que unifican una serie de subsistemas típicamente utilizados.

Este curso explora la arquitectura de un motor de juegos en 2D/3D, profundizando en la descripción y decisiones de diseño de subsistemas típicos como: motor de renderizado, sistema de audio, detección de colisiones, simulaciones físicas, sistema de animaciones y sistema de inteligencia artificial. Todo esto, manteniendo la separación clara entre el motor y la aplicación de videojuego. El propósito es implementar y comprender de forma teórica y práctica dificultades inherentes al tipo de software calificado como *Game Engine*, o motor de juegos. Disponer de este conocimiento se vuelve clave al momento de comprender, trabajar y expandir motores de juegos utilizados en la industria.

A término del curso, la/él estudiante habrá implementado su propio motor de juegos básico pero específico, desarrollando tanto las habilidades para comprender y modificar este tipo de sistemas, así como la integración efectiva de bibliotecas de terceros para resolver problemas específicos del mundo de los videojuegos. El curso incluye una breve introducción a herramientas de desarrollo e ingeniería de software para orientar y facilitar el desarrollo del motor 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	RA1: Programa un motor de videojuegos para un caso de uso específico, incluyendo la programación e interconexión de los distintos subsistemas estudiados: gráfico, audio, física, colisiones, animaciones, inteligencia artificial, mensajes/eventos y scripting.
CE1, CE2, CE5	RA2: Desarrolla una arquitectura de software apropiada para la implementación de un motor de videojuegos.
CE1, CE6	RA3: Desarrolla software complejo integrando distintas bibliotecas de software de terceros con el fin de satisfacer requerimientos específicos a su propio motor de videojuegos.
CE1, CE8	RA4: Utiliza efectivamente herramientas de desarrollo de software como IDE, debugger, profiler e inspector de memoria, en el desarrollo de un motor de videojuegos.
CE1, CE2, CE5	RA5: Utiliza los fundamentos teóricos y técnicos de los subsistemas involucrados en un motor de videojuegos tradicional (gráfico, audio, física, colisiones y animaciones), a fin de desarrollar programas propios, logrando identificar limitantes de los distintos enfoques de cada subsistema.
CE1, CE2	RA6: Analiza y discute motores de videojuegos de terceros.
Competencias genéricas	Resultados de aprendizaje
CG1	RA7: Produce reportes y presentaciones donde se ilustran los avances periódicos en el desarrollo de su propio motor de videojuegos, con especial énfasis en las decisiones técnicas según los casos de uso considerados.
CG2	RA8: Investiga distintas fuentes de información teórica y técnica relevante a las distintas tecnologías de desarrollo de videojuegos, aplicando activamente estos conocimientos en el progreso de su proyecto individual.

C. Unidades temáticas:

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
1	RA5, RA7	Introducción	1 semana
Contenidos		Indicador de logro	
1. Presentación del curso 2. Concepto de Game Engine		La/el estudiante: <ul style="list-style-type: none"> - Analiza el producto de software “Game Engine”, sus aplicaciones, su evolución histórica y su relevancia en la actualidad. 	
Bibliografía de la unidad		Diapositivas del curso.	

Número	RA al que tributa	Nombre de la unidad	Duración en Semanas
2	RA4	Desarrollo de Software	2 semanas
Contenidos		Indicador de logro	
1. Herramientas de Desarrollo <ul style="list-style-type: none"> 1.1. C++ y sistema de construcción de software 1.2. Control de versiones 1.3. Software: IDE, debugger, profiler, memory leak/corruption detection 2. Ingeniería de Software para Videojuegos <ul style="list-style-type: none"> 2.1. Datos, código y memoria 2.2. Atrapando y manejando errores 2.3. Manejo de memoria 2.4. Estructuras de datos contenedoras 		La/el estudiante: <ul style="list-style-type: none"> - Utiliza distintas herramientas de software para analizar problemas típicos del desarrollo de software, como bugs, velocidad o problemas de memoria. - Programa una pequeña aplicación que compare el desempeño de al menos dos estructuras de datos contenedoras utilizando un sistema de construcción de software, control de versiones y demostrando un manejo apropiado de errores. 	
Bibliografía de la unidad		(1) Capítulo 2: Tools of the Trade. (1) Capítulo 3.1-3.3: Fundamentals of Software Engineering for Games	

Número	RA al que tributa	Nombre de la unidad	Duración en Semanas
3	RA3, RA4, RA5, RA7, RA8	Sistemas Elementales	3 semanas
Contenidos		Indicador de logro	
<ol style="list-style-type: none"> 1. Manejo de tiempo y ciclo de juego 2. Dispositivos de Interfaz Humanas 3. Pipeline de Rendering Gráfico <ol style="list-style-type: none"> 3.1. GPU & Shaders 3.2. Transformaciones 3.3. Texturizado 3.4. Iluminación 4. Arquitectura de un motor de renderizado <ol style="list-style-type: none"> 4.1. Optimización 4.2. Ordenamiento, Alpha blending y Z pre-pass 4.3. Determinación de visibilidad y Grafos de Escena 4.4. Efectos: Partículas, Overlays, Decals, Post Processing 5. Sistema de audio <ol style="list-style-type: none"> 5.1. Audio Digital 5.2. Renderizado de audio 2D y 3D 		<p>La/el estudiante:</p> <ul style="list-style-type: none"> - Programa un motor de renderizado 2D y/o 3D que incluya etapas de optimización y aplicación de distintos efectos visuales de interés. - Programa una aplicación simple que utilice el motor de renderizado, tecnologías de audio y algún dispositivo de interfaz humano computador para generar un primer mundo virtual interactivo. - Utiliza herramientas de desarrollo de software para la implementación, optimización e iteración de las aplicaciones solicitadas. 	
Bibliografía de la unidad		<p>(1) Capítulo 6.1-6.3: Engine Support Systems (1) Capítulo 8: The Game Loop and Real-Time Simulation (1) Capítulo 6: Human Interface Devices (1) Capítulo 11: The Rendering Engine (5) Capítulo 8: Visibility Determination (1) Capítulo 14.4-14.5: Audio</p>	

Número	RA al que tributa	Nombre de la unidad	Duración en Semanas
4	RA1, RA2, RA6, RA7, RA8	Arquitectura de un Motor de Videojuegos	3 semanas
Contenidos		Indicador de logro	
<ol style="list-style-type: none"> 1. Modelo de Game Objects <ol style="list-style-type: none"> 1.1. Runtime versus offline 1.2. Generadores de Game Objects 1.3. Actualización de Game Objects 1.4. Integración de Game Objects en un Motor 2. Administrador de Recursos <ol style="list-style-type: none"> 2.1. Sistema de archivos 2.2. Pipeline de Assets 2.3. Administrando recursos 3. Gameplay Foundation Systems <ol style="list-style-type: none"> 3.1. Componentes de la capa gameplay 3.2. Modelos de arquitecturas para sistemas en tiempo real 3.3. Manejo de memoria 3.4. Streaming de mundos 3.5. Manejo de memoria para objetos dinámicos 		<p>La/el estudiante:</p> <ul style="list-style-type: none"> - Programa un Game Engine básico exponiendo una cantidad limitada de funcionalidades al usuario. El Game Engine debe incluir los sistemas de renderizado gráfico y de audio, y considerar además una administración correcta de recursos. - Utiliza su propio Game Engine en el desarrollo de un videojuego minimalista que ejercite los sistemas de renderizado gráfico y de audio. 	
Bibliografía de la unidad		(1) Capítulo 15: Introduction to Gameplay Systems (1) Capítulo 7: Resources and the File System (1) Capítulo 16.1-16.5: Runtime Gameplay Foundation Systems	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
5	RA3, RA5, RA7, RA8	Física e Interacciones	2 semanas
Contenidos		Indicador de logro	
1. Detección de Colisiones 1.1. Colisiones Básicas 1.2. Optimizaciones 1.3. Estrategias y Limitaciones 1.4. Ray Casting 1.5. Sphere Casting 2. Motor de Física 2.1. Integración numérica 2.2. Dinámica Lineal 2.3. Dinámica Rotacional 2.4. Restricciones y Ragdolls 2.5. Integrando física en un videojuego		La/el estudiante: <ul style="list-style-type: none"> - Utiliza estrategias eficientes de detección de colisiones y de simulaciones físicas para simular interacciones de mayor complejidad. - Analiza limitantes tanto de sistemas de colisiones como de simulaciones de física en el contexto de aplicaciones interactivas. - Desarrolla su aplicación, logrando determinar restricciones apropiadas para trabajar dentro de las limitantes posibles. 	
Bibliografía de la unidad		(1) Capítulo 13: Collision and Rigid Body Dynamics (2) Capítulo 9: Collision Detection	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
6	RA1, RA2, RA5, RA6	Integración de Sistemas	2 semanas
Contenidos		Indicador de logro	
1. Integración de Subsistemas 1.1. Simulación multi objetos 1.2. Integración de renderizado gráfico, audio y física 2. Eventos y Scripting 2.1. Envío de mensajes 2.2. Lenguajes de scripting 2.3. Flujo del juego		La/el estudiante: <ul style="list-style-type: none"> - Incorpora a su Game Engine los sistemas de detección de colisiones y de simulación de física, manteniendo siempre una interfaz apropiada para desarrollar aplicaciones sobre él. - Describe y analiza la relevancia de los sistemas de eventos y scripting en Game Engines, identificando distintos enfoques posibles y como estos agilizan el proceso de desarrollo. 	
Bibliografía de la unidad		(1) Capítulo 16.6-16.10: Runtime Gameplay Foundation Systems	

Número	RA al que tributa	Nombre de la unidad	Duración en Semanas
7	RA1, RA5, RA6, RA7, RA8	Control de Movimiento	2
Contenidos		Indicador de logro	
<ol style="list-style-type: none"> 1. Sistema de Animación <ol style="list-style-type: none"> 1.1. Esqueletos y poses 1.2. Clips y manejo de tiempo local 1.3. Skinning 1.4. Cinemática Inversa 1.5. Pipeline de Animación 1.6. Técnicas de compresión 1.7. Máquinas de estado de animación 2. Inteligencia Artificial en Videojuegos <ol style="list-style-type: none"> 2.1. Proceso tomador de decisiones 2.2. Planificación de movimiento 		<p>La/el estudiante:</p> <ul style="list-style-type: none"> - Programa un sistema de animación simple y lo incorpora a su Game Engine, concentrando esfuerzos en implementar al menos una técnica avanzada de animación - Programa un pequeño sistema tomador de decisiones y lo incorpora a su Game Engine. - Utiliza tanto su Game Engine personal como algún otro para desarrollar una aplicación/videojuego simple que incorpore los sistemas desarrollados y estudiados en el curso. 	
Bibliografía de la unidad		(1) Capítulo 12: Animation Systems (2) Capítulo 4: Skeletal Animation (2) Capítulo 6: Kinematics (4) Capítulo 3: Solving Problems by Searching (4) Capítulo 25.4: Planning to Move	

D. Estrategias de enseñanzas:

El curso se estructura en base a distintas metodologías que fomenta la participación del estudiante e incluyen principalmente:

- Cátedras
- Presentaciones de proyectos
- Casos de estudio

E. Estrategias de evaluación:

La propuesta de evaluación es de proceso y considera distintas instancias, entre las que se pueden mencionar:

Proyectos computacionales individuales y grupales
Proyectos de investigación
Presentaciones

F. Recursos bibliográficos:

Bibliografía obligatoria:

- (1) Gregory, J. (2019). *Game Engine Architecture*. CRC Press, Taylor & Francis Group: third edition.
- (2) Mukundan R. (2012). *Advance Methods in Computer Graphics with examples in OpenGL*. Springer.

Bibliografía Complementaria:

- (3) Meyers, S. (2015). *Effective Modern C++, 42 Specific Ways to Improve Your use of C++11 and C++14*. O'Reilly 2015.
- (4) Russell, S., Norvig, P. (2010). *Artificial Intelligence. A Modern Approach*. Pearson: third edition.
- (5) Lengyel, E. (2012). *Mathematics for 3D Game Programming and Computer Graphics*. Course Technology, Cengage Learning: third edition.
- (6) Sellers G., Wright, Jr. R. S., Haemel N. (2016). *OpenGL SuperBible. Comprehensive Tutorial and Reference*. Addison Wesley: seventh edition.

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

Vigencia desde:	Primavera, 2021
Elaborado por:	Daniel Calderón
Validado por:	
Revisado por:	AGC