



GUÍA DOCENTE

PROGRAMACIÓN DE VIDEOJUEGOS AVANZADA

GRADO EN INGENIERÍA DEL SOFTWARE

MODALIDAD: PRESENCIAL

CURSO ACADÉMICO: 2023-2024

Denominación de la asignatura:	Programación de videojuegos avanzada
Titulación:	Ingeniería del Software
Facultad o Centro:	Centro Universitario de Tecnología y Arte Digital
Materia:	Programación Gráfica, Sistemas Inmersivos y Videojuegos
Curso:	4º
Cuatrimestre:	2
Carácter:	OBM
Créditos ECTS:	6
Modalidad/es de enseñanza:	Presencial
Idioma:	Castellano
Profesor/a - email	David Rocha Cerezo / david.rocha@live.u-tad.com
Página Web:	http://www.u-tad.com/

DESCRIPCIÓN DE LA ASIGNATURA

Descripción de la materia

La materia dota a los alumnos de las competencias y conocimientos necesarios para el desarrollo de experiencias interactivas como los videojuegos, inmersivas como las propias de la realidad virtual o realidad aumentada y la generación de gráficos a través

Descripción de la asignatura

La programación de videojuegos se realiza actualmente de forma generalizada con dos motores gráficos principales: Unity y Unreal Engine. El motor de Epic resulta de manejo más complejo que el de Unity, no obstante, el resultado obtiene mejores calidades y prestaciones.

El objetivo de este curso es dotar al alumno de conocimiento avanzado para la programación de videojuegos, tanto a nivel de dinámicas de juego como con el uso de un motor más complejo como Unreal Engine 5.

COMPETENCIAS Y RESULTADOS DE APRENDIZAJE

Competencias (genéricas, específicas y transversales)

COMPETENCIAS BÁSICAS Y GENERALES

CB1: Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.

CB2: Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

CB3: Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.

CB4: Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

CB5: Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía

CG1: Entender, planificar y resolver problemas a través del desarrollo de soluciones informáticas.

CG3: Aplicar los fundamentos científicos para la resolución de problemas informáticos

CG4: Entender la complejidad, simplificar y optimizar los sistemas informáticos

CG9: Aprender, modificar y producir nuevas tecnologías informáticas

CG10: Aplicar las técnicas creativas para la realización de proyectos informáticos

COMPETENCIAS ESPECÍFICAS

CE10: Generar documentación de una aplicación de forma automática así como entender y manejar adecuadamente un gestor de versiones de código

COMPETENCIAS TRANSVERSALES

CT1: Conocer la definición y el alcance, así como poner en práctica los fundamentos de las metodologías de gestión de proyectos de desarrollo tecnológico.

CT2: Conocer los principales agentes del sector y el ciclo de vida completo de un proyecto en desarrollo y comercialización de contenidos digitales

CT4: Actualizar el conocimiento adquirido en el manejo de herramientas y tecnologías digitales en función del estado actual del sector y de las tecnologías empleadas.

Resultados de aprendizaje

Al acabar la titulación, el graduado o graduada será capaz de:

- Conocer las posibilidades de la tecnología y las restricciones que impone en la construcción de videojuegos.

- Conocer la sintaxis y uso básico de los lenguajes de programación indicados para el diseño de videojuegos
- Implementar programas sencillos acompañados de baterías sencillas de pruebas
- Desarrollar juegos simples en lenguajes de scripting
- Manejar los conceptos de diseño 2D en la elaboración de un juego
- Aplicar el conocimiento de diseño de juegos a la construcción de un juego 3D básico
- Ser capaz de usar técnicas específicas de programación gráfica que permiten escribir código más eficiente.
- Entender el pipeline de producción gráfica, sus etapas, reglas y monitorización.
- Conocer y aplicar las soluciones más comunes para el renderizado de escenas digitales en la industria.
- Aprender a desarrollar soluciones en distintos mundos inmersivos dentro del continuo de la virtualidad.
- Aprender desarrollar aplicaciones y experiencias de realidad aumentada y mixta, potenciando sus características utilizando los distintos entornos software disponibles.
- Ser capaz de integrar los desarrollos software con los dispositivos de AR/VR.
- Asimilar los algoritmos de simulación numérica para generar imágenes fotorrealistas.
- Utilizar la API de OpenGL para la representación de vértices, aristas y superficies en el espacio tridimensional.
- Ser capaz de diseñar aplicaciones con algoritmos concurrentes, ejecutables en paralelo en hardware gráfico especial
- Diseñar, desarrollar y desplegar un proyecto completo de computación gráfica, AR/VR o videojuegos utilizando las técnicas adquiridas en esta mención.

CONTENIDO

Generación Procedural de contenidos

Técnicas de renderizado en tiempo real

Sistemas de animación

Métodos de Interacción e Interfaces de Usuario

Programación en red

Shaders y programación a bajo nivel

TEMARIO

Introducción:

Presentación de la asignatura, profesor y condiciones de evaluación.

Tema 1. Introducción a Unreal Engine 5 y su entorno de desarrollo:

- Instalación y puesta en marcha
- De Unity a Unreal Engine 5
- Editor

Tema 2. Fundamentos de videojuegos y Unreal Engine 5:

- Edición de niveles y escenas
- Iluminación
- Texturas y materiales

Tema 3. Blueprints.

- Introducción a los Blueprint
- Anatomía y tipos de Blueprint
- Editor de Blueprint
- Blueprint Class
- Blueprint Interface
- Blueprint Components
- Nodos y funciones especiales
- Implementando sonidos con Cues

Tema 4. Unreal Engine Avanzado.

- Inteligencia Artificial
- Animaciones
- Interfaz de usuario
- VFX con Niagara

Tema 5. Programación en C++

- Creando clases C++
- Añadiendo componentes
- Blueprint heredado de C++
- Variables
- Overriding
- Eventos

- Componentes Custom

ACTIVIDADES FORMATIVAS Y METODOLOGÍAS DOCENTES

Actividades formativas

Actividad Formativa	Horas totales	Horas presenciales
<i>Clases teóricas / Expositivas</i>	29,38	29,38
<i>Clases Prácticas</i>	23,25	23,25
<i>Tutorías</i>	4,00	0,00
<i>Estudio independiente y trabajo autónomo del alumno</i>	50,00	0,00
<i>Elaboración de trabajos (en grupo o individuales)</i>	31,88	0,00
<i>Actividades de Evaluación</i>	5,25	5,25
<i>Seguimiento de Proyectos</i>	6,25	6,25
TOTAL	150	64,13

Metodologías docentes

Método expositivo o lección magistral

Aprendizaje de casos

Aprendizaje basado en la resolución de problemas

Aprendizaje basado en proyectos

Aprendizaje cooperativo o colaborativo

Aprendizaje por indagación

Metodología Flipped classroom o aula invertida

Gamificación

Just in time Teaching (JITT) o aula a tiempo

Método expositivo o lección magistral

Método del caso

Aprendizaje basado en la resolución de problemas

Aprendizaje basado en proyectos
Aprendizaje cooperativo o colaborativo
Aprendizaje por indagación
Metodología flipped classroom o aula invertida
Gamificación

DESARROLLO TEMPORAL

UNIDADES DIDÁCTICAS / TEMAS	PERÍODO TEMPORAL
Introducción a Unreal Engine 5 y su entorno de desarrollo.	
Creación de grupos de trabajo	
Creación de un mini proyecto	Semana 1
Fundamentos de videojuegos y Unreal Engine	
Actividades practicas	
Ideación del proyecto final (videojuego)	
Entrega de documento del proyecto a realizar	
	Semanas 2, 3, 4
Blueprints	
Entrega de prototipo básico	
Presentación de demo POC	
	Semanas 5, 6, 7, 8
Unreal Engine Avanzado	
Actividades practicas	
Sesión de dudas proyecto final	
	Semanas 9, 10, 11
Programación C++ en Unreal	
Desarrollo final del proyecto (componentes avanzados)	Semanas 12, 13, 14
Presentación del proyecto final y evaluación	Semana 15

SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN MÍNIMA RESPECTO A LA CALIFICACIÓN FINAL (%)	VALORACIÓN MÁXIMA RESPECTO A LA CALIFICACIÓN FINAL (%)
<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	10	30
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	40	80
<i>Prueba Objetiva</i>	10	60

CRITERIOS ESPECÍFICOS DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	CONVOCATORIA ORDINARIA	CONVOCATORIA EXTRAORDINARIA
<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	10	10
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	50	50
<i>Prueba Objetiva</i>	40	40

Consideraciones generales acerca de la evaluación

Convocatoria Ordinaria

- El alumno deberá de entregar y aprobar con nota de cinco sobre diez los trabajos evaluables. Contará un 50% de la nota final.
- El alumno deberá aprobar el examen final obligatorio con nota de cinco sobre diez. Contará un 40% de la nota final. El examen será la defensa de su conocimiento y proyecto de forma oral e individual.
- Para aprobar la asignatura todas las partes han de tener una calificación superior a 5 sobre 10.

Convocatoria extraordinaria

- El alumno deberá de entregar y aprobar con nota de cinco sobre diez los trabajos evaluables. Contará un 50% de la nota final.
- El alumno deberá aprobar el examen final obligatorio con nota de cinco sobre diez. Contará un 40% de la nota final. El examen será la defensa de su conocimiento y proyecto de forma oral e individual.

- El 10% restante de participación se recuperará de la convocatoria ordinaria.
- Para aprobar la asignatura todas las partes han de tener una calificación superior a 5 sobre 10.
- En la convocatoria extraordinaria se deberán entregar todos los trabajos realizados durante el curso y realizar el examen (incluyendo actividades practicas realizadas durante las clases).
- Cualquier escrito que el alumno presente (informes de los casos prácticos, problemas, exámenes, comentarios de los programas, etc.) deberá estar bien presentado, correctamente redactado (con la adecuada puntuación) y sin faltas ortográficas. Los informes con faltas de ortografía no se corregirán, ya que a un universitario se le exige calidad máxima en su expresión escrita
- A pesar de tratarse de una asignatura fuertemente teórica, se espera la participación activa del alumno y se le estimulará para que haga preguntas concretas y muestre su interés sobre temas particulares del temario. Se animará al alumno a que realice investigaciones concretas de tecnologías en particular y que se expongan y discutan en clase, y los alumnos presentarán un proyecto de diseño que haga uso de una tecnología en concreto.

BIBLIOGRAFÍA / WEBGRAFÍA

Bibliografía básica

Epic Games, Unreal Engine 5.3 Documentation, desde <https://docs.unrealengine.com/5.3/en-US/>

"Game Programming Patterns" de Robert Nystrom

"Unreal Engine 4 Game Development Essentials" de Satheesh PV

Bibliografía recomendada

"Blueprints Visual Scripting for Unreal Engine 5: Unleash the true power of Blueprints to create impressive games and applications in UE5" de Marcos Romero, Brenden Sewell Luis Cataldi

"Unreal Engine 5 Character Creation, Animation, and Cinematics: Create custom 3D assets and bring them to life in Unreal Engine 5 using MetaHuman, Lumen, and Nanite" de Henk Venter, Wilhelm Ogterop

MATERIALES, SOFTWARE Y HERRAMIENTAS NECESARIAS

Tipología del aula

Aula teórica

Equipo de proyección y pizarra

Materiales:

Se recomienda que el alumno traiga su ordenador y cascos.

Conexión a Internet.

Software:

Powerpoint, doc, Excel, web.

Unreal Engine 5