



GUÍA DOCENTE

PROGRAMACIÓN GRÁFICA

GRADO EN INGENIERÍA DEL SOFTWARE

MODALIDAD: PRESENCIAL

CURSO ACADÉMICO: 2023-2024

Denominación de la asignatura:	Programación Gráfica
Titulación:	Ingeniería del Software
Facultad o Centro:	Centro Universitario de Tecnología y Arte Digital
Materia:	Programación Gráfica, Sistemas Inmersivos y Videojuegos
Curso:	3º
Cuatrimestre:	2
Carácter:	OBM
Créditos ECTS:	6
Modalidad/es de enseñanza:	Presencial
Idioma:	Castellano
Profesor/a - email	Marcos Novalbos Mendiguchía / marcos.novalbos@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

En esa asignatura se enseñan los conceptos y técnicas de programación gráfica a bajo nivel, con el objeto de aprender los procesos de modelado, texturizado y renderizado desde la perspectiva del programador gráfico.

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

Programación pipeline gráfico

Transformaciones y modelado 2D/3D

Proyecciones

Técnicas de Renderizado

APIs de Computación gráfica

TEMARIO

Tema 1: Introducción a los sistemas de visualización 3D

Presentación de la asignatura. Definición de las técnicas de visualización 3D, herramientas disponibles y planificación del curso.

Tema 2: Transformaciones

Introducción a los sistemas de representación 3D.

Explicar como se definen los datos que se usarán para representar objetos virtuales. - Vértices, polígonos, mallas poligonales...

Explicar transformaciones y matrices de transformación:

-Posición-rotación-escalado

-Modelo-Vista-Proyección

Implementación de librería matemática para uso en prácticas

Tema 3: Opendgl y uso de cámara

Introducción a opengl

- Dibujado usando opengl 1.2
- Uso de bufferObjects
- Carga de mallas básicas, colores planos
- Uso de matrices de transformación:
 - o Modelo: Movimiento de objetos
 - o Vista: Creación de una cámara y movimientos básicos
 - o Proyección: Cálculo de una matriz de proyección
- Implementación de motor gráfico “sencillo” aplicando lo anterior

Tema 4: Shaders

Introducción a GLSL, uso de GPU para cómputo básico de dibujado

- Shaders de vértices
- Shaders de fragmentos
- Enlazado de bufferObjects con variables uniform/attribute
- Implementación de motor gráfico usando Opendgl 4.0

Tema 5: Iluminación Texturas

Sistemas de iluminación HW y SW.

Representación teórica de los algoritmos de iluminación de Phong y Goraud. Aplicación de texturas en mallas 3D

- Formatos de color
- Carga de imágenes 2D
- Mipmas, filtros, etc..
- Transparencias, aplicar mezcla de color por funciones alfa

- Uso de samplers2D en shader de fragmentos
- Uso de algoritmos de iluminación básicos en shader de fragmentos
- Implementación en motor gráfico de las técnicas anteriores

Tema 6: Detección de colisiones

- Teoría básica de detección de colisiones:
 - o División espacial
 - o Volúmenes envolventes
 - o Spatial hashing
 - o Árboles de volúmenes envolventes
- Implementación de detección de colisiones en un escenario con varios objetos.

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
Tema 1: Introducción a los sistemas de visualización 3D	2 Semanas
Tema 2: Transformaciones	3 Semanas
Tema 3: OpenGL y uso de cámara	3 Semanas
Tema 4: Shaders	2 semanas
Tema 5: Iluminación Texturas	2 semanas
Tema 6: Detección de colisiones	2 semanas

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>	70	70
<i>Prueba Objetiva</i>	20	20

Consideraciones generales acerca de la evaluación

Evaluación en convocatoria Ordinaria

- La evaluación de la participación en clase, en prácticas o en proyectos de la asignatura se realizará a partir de la asistencia y la participación activa en clase y en el resto de las actividades desarrolladas durante el curso. Eso incluirá la entrega de trabajos obligatorios a tiempo. Este aspecto representará el 10% de la calificación final de la asignatura en la convocatoria ordinaria.
- A lo largo del curso se plantearán actividades, ejercicios y problemas que deberán ser entregadas antes de la fecha indicada a través de la plataforma virtual. Este trabajo se evaluará a través de la propia plataforma virtual y supondrá un 70% de la calificación final de la asignatura en la convocatoria ordinaria. No se admitirán trabajos fuera de forma y fecha sin causa justificada. En caso de que la nota de un ejercicio no

alcance el aprobado, se admitirá una nueva entrega que si tiene la suficiente calidad tendrá una calificación de 5.0.

- Evaluación en convocatoria Extraordinaria
- En caso de no conseguir el aprobado en la convocatoria ordinaria, el alumno podrá presentarse a la convocatoria extraordinaria. Se realizará un examen final que representará el 20% de su calificación en dicha convocatoria, y en el que formará parte de la materia exigible al alumno todo el contenido de la asignatura visto en clase. El otro 70% se calculará a partir de los ejercicios prácticos entregables pedidos a lo largo de la asignatura. Por último se mantendrá el 10% de la participación obtenida durante la primera convocatoria. Para aprobar la asignatura en la convocatoria extraordinaria, es imprescindible que la nota final sea al menos 5.0 (sobre 10). Además de ese requisito, es necesario que la media de los trabajos planteados sea al menos 5.0 (sobre 10), y que la nota del examen final sea al menos de 5.0 (sobre 10).
- En convocatoria extraordinaria se guardará la nota de todo lo entregado hasta el momento.
- En caso de haber suspendido o no haber entregado prácticas en las fechas pedidas, se pedirá una nueva entrega de las mismas solucionando errores. Estas entregas tendrán como mínimo una penalización del 10% de la nota máxima, la cual no afectará al mínimo exigible para el aprobado (5). Eso quiere decir que tendrán una nota máxima de 9 sobre 10.

BIBLIOGRAFÍA / WEBGRAFÍA

Bibliografía Básica:

- The OpenGL® Programming Guide 9th Edition John Kessenich, Graham Sellers, and Dave Shreiner

MATERIALES, SOFTWARE Y HERRAMIENTAS NECESARIAS

Tipología del aula

Aula teórica

Equipo de proyección y pizarra

Materiales:

Ordenadores con tarjetas gráficas compatibles con OPENGL 4.2 o superior.

Software:

Windows 10 o superior

- Visual Studio 2022 o superior

- GLFW

- GLM

- GLAD

Linux ubuntu 20.04 o superior

- Gcc/G++

- GLFW

- GLM

- GLAD