



## Guía docente

### 804260 - PGA - Programación Gráfica Avanzada

Última modificación: 12/02/2021

**Unidad responsable:** Centro de la Imagen y la Tecnología Multimedia  
**Unidad que imparte:** 804 - CITM - Centro de la Imagen y la Tecnología Multimedia.

**Titulación:** GRADO EN DISEÑO Y DESARROLLO DE VIDEOJUEGOS (Plan 2014). (Asignatura optativa).  
GRADO EN DISEÑO Y DESARROLLO DE VIDEOJUEGOS (Plan 2014). (Asignatura optativa).

**Curso:** 2020      **Créditos ECTS:** 6.0      **Idiomas:** Inglés

#### PROFESORADO

---

**Profesorado responsable:** Díaz García, Jesús

**Otros:**

#### COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

---

**Específicas:**

CEVJ 5. Utilizar lenguajes de programación, patrones algorítmicos, estructuras de datos, herramientas visuales de programación, motores de juego y librerías para el desarrollo y prototipado de videojuegos, de cualquier género y para cualquier plataforma y dispositivo móvil.

CEVJ 6. Analizar, decidir y aplicar técnicas de programación gráfica, física, inteligencia artificial, interacción, realidad aumentada y redes a un proyecto de videojuego.

**Transversales:**

04 COE N1. COMUNICACIÓN EFICAZ ORAL Y ESCRITA - Nivel 1: Planificar la comunicación oral, responder de manera adecuada a las cuestiones formuladas y redactar textos de nivel básico con corrección ortográfica y gramatical.

CT4. USO SOLVENTE DE LOS RECURSOS DE INFORMACIÓN: Gestionar la adquisición, la estructuración, el análisis y la visualización de datos e información en el ámbito de especialidad, y valorar de forma crítica los resultados de dicha gestión.

07 AAT N1. APRENDIZAJE AUTÓNOMO - Nivel 1: Llevar a cabo tareas encomendadas en el tiempo previsto, trabajando con las fuentes de información indicadas, de acuerdo con las pautas marcadas por el profesorado.

#### METODOLOGÍAS DOCENTES

---

Las clases de aprendizaje dirigido se estructuran en sesiones de dos horas. Durante parte de las sesiones, el profesor expone los conceptos teóricos y lo ejemplifica mediante ejemplos que se resuelven, en lo posible, de forma participativa por parte de los estudiantes. Otra parte de la sesión se dedica a que los estudiantes practiquen los conceptos introducidos resolviendo una serie de ejercicios o los proyectos propuestos por el profesorado y, cuando proceda, también se dedicará tiempo para la resolución de las dudas y problemas encontrados durante su realización.

#### OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

---

- Conocer el estado actual y las diferentes posibilidades que ofrece la informática gráfica en el desarrollo de videojuegos.
- Completar los conocimientos en gráficos y programación de GPUs adquiridos en asignaturas anteriores (principalmente Motores de Videojuegos y Realidad Aumentada) para desarrollar técnicas gráficas avanzadas.
- Ser capaz de sacar el máximo rendimiento a la potencia de cálculo de las GPUs actuales para desarrollar algoritmos gráficos complejos en tiempo real.



## HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTADO

Tipo	Horas	Porcentaje
Horas grupo grande	18,0	12.00
Horas actividades dirigidas	12,0	8.00
Horas grupo mediano	30,0	20.00
Horas aprendizaje autónomo	90,0	60.00

**Dedicación total:** 150 h

## CONTENIDOS

### Programación de shaders con GLSL

**Descripción:**

1. Introducción a GLSL
2. Herramientas básicas de GLSL
3. Fundamentos matemáticos de técnicas de trazado de rayos

**Actividades vinculadas:**

Práctica 1: Implementación de un shader de raycasting procedural

**Dedicación:** 50h

Grupo grande/Teoría: 10h

Grupo mediano/Prácticas: 10h

Aprendizaje autónomo: 30h

### Programación de GPUs con OpenGL

**Descripción:**

1. Vista global de la pipeline de renderizado
2. Espacio de coordenadas normalizadas de dispositivo (NDC)
3. Renderizado de mallas de triángulos
4. Paso de variables "globales" a los shaders
5. Manejo del estado (tests de visibilidad, blending, etc)
6. Herramientas de depuración y optimización
7. Transformaciones geométricas
8. Renderizado a textura

**Dedicación:** 50h

Grupo mediano/Prácticas: 10h

Actividades dirigidas: 10h

Aprendizaje autónomo: 30h



## Técnicas gráficas avanzadas

### Descripción:

1. Bump, normal, and relief mapping
2. Full-screen effects
3. Screen Space Ambient Occlusion (SSAO)
4. Environment mapping
5. Physically based rendering
6. Water
7. Bloom

### Actividades vinculadas:

Proyecto final: Implementación de un motor gráfico

### Dedicación: 50h

Grupo grande/Teoría: 10h

Grupo mediano/Prácticas: 10h

Aprendizaje autónomo: 30h

## SISTEMA DE CALIFICACIÓN

La evaluación de la asignatura se realizará mediante la entrega de dos ejercicios prácticos (EP) sobre aspectos concretos de la asignatura, además de un proyecto final (PR), dónde se implementará un motor gráfico que incluya buena parte de las técnicas gráficas explicadas en clase. También se evaluará la participación (PART) del alumno a partir de sus intervenciones y del interés de aprendizaje demostrado en clase. Al tratarse de una asignatura de proyecto no habrá posibilidad de presentarse a la reevaluación.

Nota final = 0.30 \* EP1 + 0.30 \* EP2 + 0.30 \* PR + 0.1 \* PART

## NORMAS PARA LA REALIZACIÓN DE LAS PRUEBAS.

Parte de los ejercicios se harán en clase con la asistencia del profesor, pero también se tendrá que dedicar tiempo fuera de clase para acabar las actividades y proyectos propuestos durante el curso.

Los proyectos pueden ser hechos por parejas y se han de entregar vía el Campus Virtual siguiendo las instrucciones proporcionadas en sus respectivos enunciados (nombres de archivos, etc.). Los proyectos que se entreguen después de la media noche de la fecha de entrega se considerarán como NP. Cualquier circunstancia que haga que no se pueda entregar un proyecto a tiempo debe ser justificada y comunicada con suficiente antelación al profesor. La evaluación de los proyectos no se hará sólo a partir del material entregado, sino que se puede pedir al estudiante que presente su trabajo en clase.

Los proyectos y actividades propuestas durante el curso se han de poder ejecutar en las aulas del CITM, por lo tanto, es responsabilidad del alumno asegurarse de que trabaja con la misma versión del software que la instalada en el centro y de que su proyecto se puede ejecutar sin errores en el CITM.

## BIBLIOGRAFÍA

### Básica:

- Möller, Tomas; Haines, Eric; Hoffman, Naty. Real-time rendering. 3rd ed. Wellesley: A K Peters, cop. 2008. ISBN 9781568814247.
- Kessenich, J.M.; Sellers, G.; Shreiner, D. OpenGL programming guide: the official guide to learning OpenGL, version 4.5 with SPIR-V. 9th ed. Upper Saddle River, NJ: Addison-Wesley, cop. 2017. ISBN 9780134495491.
- Rost, Randi J; Licea-Kane, Bill; Ginsburg, Dan. OpenGL shading language. 3rd ed. Upper Saddle River, NJ: Addison Wesley, 2010. ISBN 9780321637635.

### Complementaria:

- Nguyen, Hubert. GPU gems 3. Upper Saddle River, NJ: Addison-Wesley, 2007. ISBN 9780321515261.



## RECURSOS

---

### Otros recursos:

ShaderToy: <https://www.shadertoy.com>

GLSL Sandbox: <http://glslsandbox.com>

OpenGL: <https://www.opengl.org>

OpenGL loader: <https://glad.dav1d.de>

GLM library: <https://github.com/g-truc/glm>

GLFW library: <https://www.glfw.org>

ImGui library: <https://github.com/ocornut/imgui>

STB library: <https://github.com/nothings/stb>

Assimp library: <https://www.assimp.org>