Planificación de la Docencia Universitaria
Grado en Diseño y Desarrollo de Videojuegos

Guía Docente
Curso Académico 2020/2021

Desarrollo para Dispositivos Móviles

Datos de Identificación de la asignatura

Título
Grado en Diseño y Desarrollo de Videojuegos

Módulo
Ciencias Aplicadas y Tecnología

Denominación de la Asignatura
Desarrollo para Dispositivos Móviles

Código
40030

Curso
Tercero

Semestre
Primero

Tipo de asignatura (básica, obligatoria u optativa)
Obligatoria

Créditos ECTS
5

Modalidad/es de enseñanza
Presencial

Profesor
Ángel Rodríguez Ballesteros

Lengua vehicular
Español

Profesorado de la Asignatura

Profesor
Ángel Rodríguez Ballesteros

Datos de Contacto
angel.rodriguez@esne.es

Tutorías Académicas
Consultar en el Campus Virtual el documento "horarios de tutorías - Grado en Diseño y Desarrollo de Videojuegos"

Requisitos Previos

Esenciales

Los propios del título.

Aconsejables

Conocimientos básicos de programación.

Sentido y Aportaciones de la asignatura al Plan de Estudios

Campo de conocimiento al que pertenece la asignatura

Esta asignatura pertenece a la rama de Ingeniería y Arquitectura y es una materia de Expresión Gráfica.

Relación de interdisciplinariedad con otras asignaturas del currículum.

- Fundamentos de la Programación
- Programación II
- Introducción a la Industria del Videojuego
- Programación Orientada a Objetos.
- Motores de Videojuegos.
- Ingeniería del Conocimiento: IA
- Programación Gráfica.
- Motores Gráficos y Plugins.
- Middleware: Herramientas de Desarrollo.

Aportaciones al plan de estudios e interés profesional de la asignatura

En esta asignatura el estudiante aprenderá las características de los diversos entornos para dispositivos móviles que ofrece el mercado, así como las diversas plataformas de desarrollo. Así mismo, la asignatura contempla el manejo de lenguajes de programación específicos para la implementación de software en varios de los entornos estudiados.

Resultados de aprendizaje en relación con las competencias que desarrolla la materia

Competencias generales

CG10. Conocerá las especificaciones tecnológicas de las distintas plataformas de ejecución de videojuegos y sabrá tener en cuenta estas características para contextualizar el diseño y el desarrollo de un videojuego.

Competencias específicas

CE18. Será capaz de identificar las capacidades y las limitaciones del hardware de las diversas plataformas de videojuegos, y sabrá tenerlas en cuenta a la hora de planificar su diseño y desarrollo.

CE19. Conocerá diversas plataformas de desarrollo y habrá adquirido conocimientos de programación para implementar software que se ejecute en dispositivos móviles.

Resultados de aprendizaje relacionados con la asignatura

- Conocerá las especificaciones tecnológicas de las distintas plataformas de ejecución de videojuegos y sabrá tener en cuenta estas características para contextualizar el diseño y el desarrollo de un videojuego.
- El alumno diseñará un proyecto de software para dispositivo móvil, sabiendo identificar las limitaciones hardware de la arquitectura del dispositivo escogido.
- El alumno podrá desarrollar un proyecto software para un dispositivo móvil.

Contenidos / Temario / Unidades Didácticas

Breve descripción de los contenidos

- Introducción.
- Programación con C++.
- Desarrollo multiplataforma usando SDKs nativos.
- Gestión de recursos.
- Gestión de diferencias entre dispositivos.
- Desarrollo multiplataforma usando un engine nativo.
- Plataforma Android.
- Plataforma IOS.
- Publicación.

Temario desarrollado

1. Introducción.

1.1. Conceptos generales.
1.2. Historia de los dispositivos móviles y contexto actual.
1.3. Plataformas existentes.
1.4. Técnicas de desarrollo de videojuegos para dispositivos móviles.
1.4.1. Técnicas de desarrollo web aplicadas a dispositivos móviles.
1.4.2. Middleware para desarrollo multiplataforma de videojuegos.
1.4.3. Herramientas de creación de videojuegos integradas.
1.4.4. SDKs nativos.
1.4.5. Desarrollo multiplataforma con SDKs nativos.

2. Programación con C++.

2.1. Conceptos básicos.
2.1.1. Introducción.
2.1.2. Revisiones del lenguaje.
2.1.3. Proceso de compilación
2.1.4. Compiladores y plataformas.
2.1.5. Entornos de desarrollo.
2.2. Programación estructurada.
2.3. Programación orientada a objetos.
2.4. Sobrecarga de operadores.
2.5. Programación genérica con plantillas.
2.6. Librería estándar.
2.7. Gestión de recursos.
2.8. C++11 y C++14.

3. Desarrollo multiplataforma usando SDKs nativos.

3.1. Introducción.
3.1.1. C++ como lenguaje común a todas las plataformas.
3.2. Programación modular y encapsulación.
3.3. Separación del código en capas.
3.3.1. Capa de abstracción de la plataforma.
3.3.1.1. Integración de C++ con otros lenguajes.

3.3.1.2. Diferencias entre plataformas.
3.3.1.3. Interfaz pública multiplataforma.
3.3.2. Capa de código general para el desarrollo de videojuegos (motor).
3.3.3. Capa de código específico a cada videojuego.
3.3.3.1. Conceptos de scripting.
3.3.4. Extensiones específicas de cada plataforma.
3.4. Organización de proyectos multiplataforma.
3.5. Soluciones existentes destacables.
3.5.1. SDL.
3.5.2. Cocos2D-X.
3.5.3. Basics.

4. Gestión de recursos.

4.1. Recursos gráficos.
4.1.1. Características.
4.1.2. Mapas de bits.
4.1.3. Dibujos vectoriales.
4.1.4. Formatos de compresión en archivo.
4.1.5. Formatos de compresión en GPU.
4.2. Recursos de sonido.
4.2.1. Características.
4.2.2. Formatos sin compresión.
4.2.3. Formatos con compresión.
4.3. Recursos de música.
4.3.1. Características.
4.3.2. Formatos.
4.4. Otros recursos.
4.5. Preparación y organización de los recursos para diferentes dispositivos.
4.6. Optimización de los recursos.
4.7. Distribución de los recursos.

5. Gestión de diferencias entre dispositivos.

5.1. Diferencias entre CPUs.
5.2. Diferencias entre GPUs.
5.3. Sensores.

- 5.3.1. Tipos y características.
- 5.3.2. Disponibilidad.

5.4. Dispositivos de entrada.

- 5.4.1. Tipos y características.
- 5.4.2. Disponibilidad.

5.5. Dispositivos de salida.

- 5.5.1. Salida de audio.
- 5.5.2. Pantalla.
  - 5.5.2.1. Características.
    - 5.5.2.1.1. Tamaño, aspect ratio, resolución y densidad.
  - 5.5.2.2. Maquetación de escenas 2D adaptable.

5.6. Optimización del consumo de la batería.

6. Desarrollo multiplataforma usando un engine nativo.

6.1. Introducción.

- 6.1.1. Características.
- 6.1.2. Ejecución dirigida por eventos.

6.2. Instalación y configuración.

6.3. Creación de un nuevo proyecto multiplataforma.

- 6.3.1. Adición de código y recursos.

6.4. Edición, ejecución y depuración con un entorno de desarrollo integrado.

6.5. Creación y manipulación de escenas.

- 6.5.1. Elementos del grafo de escena.
- 6.5.2. Uso de capas.
- 6.5.3. Cambio de escenas.

6.6. Dinámica de un videojuego.

- 6.6.1. Interacción con el usuario.
- 6.6.2. Actualización del estado.
- 6.6.3. Detección y resolución de colisiones.
- 6.6.4. Render.

7. Plataforma Android.

7.1. Introducción a la plataforma Android.

- 7.1.1. Historia.
- 7.1.2. Arquitectura.

7.2. Herramientas de desarrollo oficiales.
7.3. Instalación y configuración del entorno de desarrollo.
    7.3.1. Instalación y desinstalación de componentes con el SDK Manager.
    7.3.2. Creación de emuladores con el AVD Manager.
7.4. Android Studio.
    7.4.1. Introducción al entorno de desarrollo.
    7.4.2. Importación y creación de proyectos.
    7.4.3. Características y estructura de un proyecto.
        7.4.3.1. Archivos de código Java y C/C++.
        7.4.3.2. Archivos de recursos.
        7.4.3.3. Archivos de configuración.
            7.4.3.3.1. Introducción a Gradle.
            7.4.3.3.2. El archivo de manifiesto.
    7.4.4. Edición de código, ejecución y depuración.
        7.4.4.1. El LogCat.
        7.4.4.2. Uso de los emuladores.
        7.4.4.3. Depuración con un terminal físico.
    7.4.5. Generación de binarios instalables.
        7.4.5.1. Características y estructura de un archivo APK.
        7.4.5.2. Ofuscación con Proguard.
7.5. Componentes de una aplicación o videojuego.
    7.5.1. Activities, Services, Content Providers y Broadcast Receivers.
    7.5.2. Ciclo de vida de una Activity.
7.6. Breve introducción al desarrollo de aplicaciones usando Java.
7.7. Uso de JNI para comunicar código Java con código C/C++.
7.8. Uso de la NativeActivity para el desarrollo de videojuegos multiplataforma.
    7.8.1. Ejemplo de arquitectura.

8. Plataforma IOS.

8.1. Introducción a la plataforma iOS.
    8.1.1. Historia.
    8.1.2. Arquitectura.
8.2. Instalación y configuración del entorno de desarrollo.
8.3. XCode.

8.3.1. Introducción al entorno de desarrollo.
8.3.2. Importación y creación de proyectos.
8.3.3. Características y estructura de un proyecto.
8.3.3.1. Archivos de código Objective-C y C/C++.
8.3.3.2. Archivos de recursos.
8.3.3.3. Archivos de configuración.
8.3.4. Edición de código, ejecución y depuración.
8.3.4.1. Uso del emulador.
8.3.4.2. Depuración con un terminal físico.
8.3.5. Generación de binarios instalables.
8.3.5.1. Características y estructura de un archivo IPA.
8.4. Introducción a Objective-C.
8.5. Objective-C++.
8.5.1. Organización del código Objective-C y C++.
8.5.2. Comunicación entre el código Objective-C y C++.
8.6. Ciclo de vida de una aplicación o videojuego.
8.7. Breve introducción al desarrollo de aplicaciones usando Objective-C.
8.8. Desarrollo de videojuegos multiplataforma en iOS.
8.8.1. Ejemplo de arquitectura.

9. Publicación.

9.1. Registro en Google Play.
9.2. Publicación en Google Play.
9.3. Otras tiendas de aplicaciones Android.
9.4. Registro en Apple Store.
9.5. Publicación en Apple Store.

Cronograma

Unidades Didácticas / Temas	Período Temporal
1. Introducción.	Septiembre
2. Programación con C++.	Septiembre-Octubre
3. Desarrollo multiplataforma usando SDKs nativos.	Noviembre
4. Gestión de recursos.	Noviembre
5. Gestión de diferencias entre dispositivos.	Noviembre
6. Desarrollo multiplataforma usando un engine nativo.	Noviembre
7. Plataforma Android.	Diciembre
8. Plataforma IOS.	Enero
9. Publicación.	Enero

Modalidades Organizativas y Métodos de Enseñanza

Modalidad organizativa	Método de enseñanza	Competencias relacionadas	Horas		
			Presencial	Trabajo autónomo	Total
Clases teóricas. Actividad formativa en el aula que, utilizando la metodología expositiva, prioriza la acción docente del profesor.	**Exposición de los temas.** Explicar planificación de la asignatura: programa, apuntes y bibliografía. Repasos al inicio de la clase. Resolución de dudas: temas y lecturas. Pruebas de evaluación.	CG10, CE18, CE19	30	10	40
Clases prácticas. Actividad formativa en el aula-taller que, bajo la guía del profesor, se ordena a la resolución individual o cooperativa de ejercicios y problemas o a la ejecución de trabajos técnicos o artísticos.	**Resolución de ejercicios.** Debates sobre los temas y especialmente sobre ejercicios y lecturas. Presentaciones. Pruebas de evaluación.	CG10, CE18, CE19	20	15	35

Tutorías. Actividad formativa fuera del aula que fomenta el aprendizaje autónomo, con el apoyo de la acción de guía y seguimiento por medio de un tutor.	**Preparación de clase mediante lectura de los temas.** Planificación de debates y comentarios mediante la preparación de las lecturas. Resolución de ejercicios. Comentarios y resolución de dudas presencialmente o por correo electrónico.	CG10, CE18, CE19	10	-	10
Trabajo personal del alumno. Actividad formativa fuera del aula que, sin una guía directa del profesor o tutor, fomenta el aprendizaje autónomo del alumno.	Lecturas: preparación y búsqueda de información complementaria. Estudio personal. Preparación de comentarios y debates. Tutorías libres y voluntarias.	CG10, CE18, CE19	-	40	40

Sistema de Evaluación

General

Actividades de Evaluación	Criterios de Evaluación	Valoración respecto a la Calificación Final
Exámenes/Pruebas objetivas	Examen presencial.	40%
Trabajos y Proyectos individuales y/o cooperativos	Práctica final individual.	40%
Asistencia Participativa	Participación y trabajo en clase. Actitud frente a los contenidos de la asignatura.	20%

Consideraciones generales acerca de la evaluación

Asistencia a Clase

- La asistencia a clase es obligatoria. Se aplica la norma del 80% de asistencia recogida en la Normativa Académica disponible en la pestaña de Documentos de Interés General del Campus Virtual.
- La Dirección/Coordinación de la Titulación podrá considerar situaciones excepcionales, previo informe documental, debiendo ser aprobadas por la Dirección Académica de ESNE.
- Se exigirá puntualidad al alumno en el comienzo de las clases. Una vez transcurridos cinco minutos de cortesía, el profesor podrá denegar la entrada en el aula.

Entregas de Trabajos

- Debe entregarse la práctica en la fecha que solicite el profesor, no admitiéndose entregas posteriores.
- Los trabajos se entregarán siempre y de manera individual a través del Campus Virtual.
- Cualquier detección de copia o uso de material no creado por el propio alumno de una entrega, da por suspensa dicha entrega.

Evaluación en Convocatoria Ordinaria

- Para superar la asignatura es necesario obtener una suma ponderada entre todas las actividades de evaluación de 5 puntos o superior. No hay nota mínima ni de corte para ninguna de ellas.
- Hay que realizar un examen práctico de programación presencial y entregar una práctica que consiste en el desarrollo de un videojuego según los requisitos que se plantean en su enunciado.
- Tanto la práctica como el examen tendrán un valor del 40% cada uno en la nota final (sumando entre los dos un 80%) y el 20% restante será el trabajo realizado en clase a lo largo del semestre (incluyendo los ejercicios pedidos en clase, que se pueden entregar hasta el final de la evaluación).

Evaluación en Convocatoria Extraordinaria

- En la convocatoria extraordinaria habrá que volver a entregar la misma práctica que en la convocatoria ordinaria y hacer un único examen. El alumno puede optar por volver a entregar la misma práctica que presentó en convocatoria ordinaria, manteniendo esa nota, o bien realizar en ella las mejoras que sean procedentes para mejorarla.
- No se guardará la nota del examen de la convocatoria ordinaria para la convocatoria extraordinaria, teniendo que repetirlo obligatoriamente.

Bibliografía / Webgrafía

Bibliografía básica

Programming: Principles and Practice Using C++ (2014, 2ª Edición)
Bjarne Stroustrup
Addison-Wesley
ISBN-13: 978-0321992789

C++ Primer Plus (2011, 6ª Edición)
Stephen Prata
Addison-Wesley Professional
ISBN-13: 978-0321776402

Android NDK: Beginner's Guide (2015, 2ª Edición)
Sylvain Ratabouil
Packt Publishing
ISBN-13: 978-1783989645

iOS Game Development: Developing Games for iPad, iPhone, and iPod Touch (2013)
Thomas Lucka
CRC Press
ISBN-13: 978-1466569935

Bibliografía complementaria – Webgrafía

C++ Pocket Reference (2003)
Kyle Loudon
O'Reilly Media
ISBN-13: 978-0596004965

STL Pocket Reference (2003)
Ray Lischner
O'Reilly Media
ISBN-13: 978-0596005566

Cocos2d-x by Example: Beginner's Guide (2015, 2nd Edition)
Roger Engelbert
Packt Publishing
ISBN-13: 978-1785288852