Guía docente
804226 - PROG1VJ - Programación I

Última modificación: 15/09/2024

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 obligatoria).

Curso: 2024 **Créditos ECTS:** 6.0 **Idiomas:** Catalán, Inglés

PROFESORADO

Profesorado responsable: Díaz Iriberri, Jose

Otros: Martín Mínguez, Mónica

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Específicas:
3. 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.

Genéricas:
1. Interpretar los fundamentos del uso y programación de los computadores, los sistemas operativos, las bases de datos y, en general, los programas informáticos con aplicación en ingeniería.
2. Interpretar y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para el tratamiento automático de la información por medio de sistemas computacionales y su aplicación para la resolución de problemas propios de la ingeniería.

Transversales:
4. COMUNICACIÓN EFICAZ ORAL Y ESCRITA: Comunicarse de forma oral y escrita con otras personas sobre los resultados del aprendizaje, de la elaboración del pensamiento y de la toma de decisiones; participar en debates sobre temas de la propia especialidad.
5. 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.

METODOLOGÍAS DOCENTES

Las clases de aprendizaje dirigido estructuran en sesiones de dos horas. Durante parte de las sesiones, el profesor/a 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 propuestos por el profesorado y, cuando proceda, también se dedicará tiempo para la resolución de dudas y problemas con los que se hayan encontrado durante la realización de los ejercicios.
Se hará un uso intensivo del campus virtual (Atenea), tanto para publicar el material de la asignatura (apuntes, enunciados de problemas, soluciones propuestas, recopilación de links, etc.) como mecanismo de comunicación para publicar avisos, pedir las revisiones de las diferentes pruebas, etc.
Planificación de actividades
Distinguimos tres tipos de actividades:
- Dos tests de ejercicios prácticos de evaluación sobre temas concretos de la asignatura (P).
- Un examen parcial realizado durante la semana 7, prevista en el calendario académico (EP).
- El examen final de la asignatura. (EF).

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

- Mostrar comprensión y capacidad de explicar el concepto de algoritmo y de traducir algoritmos elementales expresados en lenguaje natural a funciones
correctas.
- Ser capaz de diseñar e implementar un programa de tamaño pequeño acompañado de baterías sencillas de prueba, dadas unas especificaciones.
- Mostrar comprensión y capacidad de explicar los conceptos de tipo, valor y variable.
- Ser capaz de diferenciar entre interpretación y compilación.
- Ser caz de efectuar cálculos que requieran almacenamiento de resultados intermedios en variables atendiendo a un uso correcto de los tipos.
- Ser capaz de implementar funciones sin y con parámetros y sin y con valor de retorno.
- Ser capaz de usar librerías existentes en el desarrollo de programas.
- Ser capaz de implementar programas capaces de realizar acciones de entrada/salida línea a línea sobre ficheros de texto; programas que hagan uso
de estructuras de control de repetición (posiblemente anidadas y en combinación con estructuras de selección): bucles con condición de parada y bucles
con variable de control de iteración y, programas que manejen cadenas, vectores, registros y matrices.

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTADO

Tipo	Horas	Porcentaje
Horas actividades dirigidas	20,0	13.33
Horas grupo mediano	16,0	10.67
Horas aprendizaje autónomo	90,0	60.00
Horas grupo grande	24,0	16.00

Dedicación total: 150 h

CONTENIDOS

1.Conceptos básicos de los algoritmos

Descripción:
1.1. Definiciones. Algoritmo y Programa.
1.2. Lenguajes de programación. Clasificación. lenguaje C.
1.3. Fase de compilación o interpretación, y linkado (link, montado o enlace).
1.4. Fase de ejecución de un programa.
1.5. El compilador de C. Características generales.
1.6. Ejemplos de juegos programados en C.
1.7 Introducción a Visual Studio.

Actividades vinculadas:
Clases de explicación teórica con prácticas de aplicación.

Dedicación: 11h
Grupo grande/Teoría: 4h 30m
Aprendizaje autónomo: 6h 30m

2.Tipos, operadores y expresiones

Descripción:
2.1 Nombres de variables.
2.2 Tipos y tamaños de datos.
2.3 Constantes.
2.4 Declaraciones.
2.5 Operadores aritméticos.
2.6 Operadores de relación y lógicos.
2.7 Conversiones de tipos.
2.8 Operadores de incremento y decremento (postfijos y prefijos).
2.9 Operadores de asignación y expresiones.
10.2 Expresiones condicionales.
2.11 Precedencia y orden de evaluación.

Actividades vinculadas:
Clases de explicación teórica con prácticas de aplicación.

Dedicación: 10h
Grupo grande/Teoría: 4h
Aprendizaje autónomo: 6h

3.Control de Flujo del programa

Descripción:
3.1 Proposiciones y bloques.
3.2 If-else.
3.3 Else-if.
3.4 Switch.
3.5 Ciclos-while y for.
3.6 Ciclos-do-while.
3.7 Break y continue.

Actividades vinculadas:
Clases de explicación teórica con prácticas de aplicación.

Dedicación: 25h
Grupo grande/Teoría: 10h
Aprendizaje autónomo: 15h

4.Funciones y la estructura del programa

Descripción:
4.1 Conceptos básicos de funciones.
4.2 Funciones que retornan valores no-enteros.
4.3 Variables externas.
4.4 Reglas de alcance.
4.5 Archivos header.
4.6 Variables estáticas.
4.7 Variables registro.
4.8 Estructura de bloque.
4.9 Inicialización.
4:10 El preprocesador de C.
4.11 Librerías estándar.

Actividades vinculadas:
Clases de explicación teórica con prácticas de aplicación.

Dedicación: 30h
Grupo grande/Teoría: 12h
Aprendizaje autónomo: 18h

Examen Parcial

Descripción:
Examen Parcial

Dedicación: 2h
Grupo grande/Teoría: 2h

5.Arrays

Descripción:
5.1 Arrays unidimensionales.
5.2 Arrays multidimensionales.
5.3 Recorridos y búsquedas.

Actividades vinculadas:
Clases de explicación teórica con prácticas de aplicación.

Dedicación: 30h
Grupo grande/Teoría: 12h
Aprendizaje autónomo: 18h

6.Estructuras

Descripción:
6.1 Conceptos básicos sobre estructuras.
6.2 Estructuras y funciones.
6.3 Arrays de estructuras.
6.4 Typedef.
6.5 Uniones.

Actividades vinculadas:
Clases de explicación teórica con prácticas de aplicación.

Dedicación: 25h
Grupo grande/Teoría: 10h
Aprendizaje autónomo: 15h

7.Ficheros

Descripción:
7.1. Introducción.
7.2. Leer ficheros.
7.3. Escribir ficheros.

Actividades vinculadas:
Clases de explicación teórica con prácticas de aplicación.

Dedicación: 15h
Grupo grande/Teoría: 6h
Aprendizaje autónomo: 9h

Examen Final

Descripción:
Examen teórico y práctico

Dedicación: 2h
Grupo grande/Teoría: 2h

SISTEMA DE CALIFICACIÓN

La nota de la asignatura (NF) se calculará utilizando la siguiente fórmula:
NF = 0.2 * EP + 0.4 * P + 0.3 * EF + 0.1*PAA

Siendo EP la nota del examen parcial, P las notas de las 2 prácticas a realizar durante el curso (cada uno vale un 20% de la nota final), EF la nota del examen final y PAA es la participación y actitud de aprendizaje.

Los alumnos que no superen la asignatura mediante la evaluación continua, tendrán la posibilidad de presentarse a la prueba de reevaluación. En esta prueba se reevalúan las calificaciones de los exámenes (EP y EF). La nota final de asignatura que resulte no podrá superar el 5.

Las acciones irregulares que puedan llevar a una variación significativa de la calificación de uno o más estudiantes constituyen una realización fraudulenta de un acto de evaluación. Esta acción comporta la calificación descriptiva de suspenso y numérica de 0 del acto de evaluación ordinario global de la asignatura, sin derecho a reevaluación.

Si los docentes tienen indicios de la utilización de herramientas de IA no permitidas en las pruebas de evaluación, podrán convocar a los estudiantes implicados a una prueba oral o a una reunión para verificar la autoría.

NORMAS PARA LA REALIZACIÓN DE LAS PRUEBAS.

Las pruebas y exámenes puntuables se realizarán presencialmente en el aula el día y hora estipulados de manera individual. Todas las pruebas tendrán una duración máxima de 2 horas. Para realizar correctamente los ejercicios se deberán seguir estrictamente las indicaciones especificadas en el documento de trabajo.

No se admitirán entregas fuera del horario establecido en el día y hora de la prueba.

Los archivos entregados a evaluación deben poderse abrir y ejecutar en cualquier ordenador de la universidad. Aquellas pruebas que no puedan ejecutarse correctamente supondrán un suspenso automáticamente.

Los documentos deben ser completados, siguiendo las instrucciones, especialmente con respecto a los nombres de los archivos. La correcta gestión de la documentación aportada es un aspecto de las competencias a adquirir y parte de la evaluación.

BIBLIOGRAFÍA

Básica:
- Kernighan, B.W.; Ritchie, D.M. El lenguaje de programación C. 2ª ed. México: Prentice Hall, 1991. ISBN 9688802050.

Complementaria:
- Joyanes Aguilar, L. Fundamentos de programación: algoritmos y estructura de datos [en línea]. 5ª ed. Madrid [etc.]: McGraw-Hill, 2020 [Consulta: 06/05/2022]. Disponible a: https://www-ingebook-com.recursos.biblioteca.upc.edu/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=10211. ISBN 9781456279448.
- Joyanes , L.; Zahonero, I. Algoritmos y estructuras de datos: una perspectiva en c. Madrid: McGraw-Hill, 2004. ISBN 9788448140779.

RECURSOS

Enlace web:
- Documentació llenguatge C. https://www.cplusplus.com/reference/