

Prácticas de programación en C.

ET1032 Informática Industrial - 2017

Este boletín se refiere a la primera parte de las prácticas de la asignatura, dedicadas a la programación en lenguaje C sobre un ordenador personal. Además de este documento, que describe el trabajo a realizar y cómo va a ser evaluado, en la web de la asignatura hay información de apoyo sobre el uso del compilador **MinGW** y su entorno integrado, información sobre la biblioteca gráfica **mgsLib**, así como algunos ejemplos.

Cada grupo de prácticas deberá completar, a su elección, sea dos ejercicios de programación (**opción 1**) sea un proyecto de simulación (**opción 2**), tal y como se explica más adelante. Cada uno de los ejercicios que se programen deberá estar en un programa individual que permita probar su funcionamiento. Por ello deberá incluir una interfaz de usuario y las operaciones de entrada y salida de datos que se consideren oportunas para realizar la prueba, además de las que se especifiquen en la descripción del ejercicio, si las hubiere.

La nota de esta parte constituye al menos el 60% de la calificación final de las prácticas. El peso de los ejercicios aparece especificado más adelante. Los criterios que se tendrán en cuenta para evaluarlos son: la corrección de los programas y de los algoritmos utilizados; la calidad y robustez de la interfaz de usuario; la estructuración de los programas y la presencia de comentarios pertinentes para explicar el código.

La entrega de las prácticas debe consistir del código debidamente comentado, los archivos de prueba donde sea necesario incluirlos y, en algún caso excepcional, la inclusión de un archivo de texto explicando el porqué de algunos resultados. Los archivos de la entrega constituyen un documento oficial para la evaluación de la asignatura, deberán ser originales y desarrollados por cada grupo de trabajo sin colaboraciones de terceros. Cualquier transgresión de este requisito será sujeto de las medidas disciplinarias recogidas en la normativa de exámenes de la Universitat.

En la web de la asignatura, como se ha dicho, aparecen numerosos ejemplos de programación cuyo código puede usarse e incluirse libremente en los ejercicios si así se desea.

Opción 1.- Tipos de datos y programación gráfica.

En esta opción se deberá realizar un ejercicio de cada uno de los dos grupos que se describen a continuación. Los ejercicios en particular se seleccionarán de entre los propuestos usando el programa `pr1op1.c` que se puede descargar de la web de la asignatura, introduciendo la dirección de correo electrónico con el número mayor de entre todas las de los miembros del grupo. Existe también la posibilidad de proponer otros ejercicios similares en cuyo caso se deberá solicitar conformidad al profesor de la asignatura.

Tipos de datos: entrada/salida y operaciones específicas.

Los ejercicios de este apartado requieren en primer lugar de la especificación de un tipo de datos adecuado a los que se deben tratar según la propuesta. Para este tipo de datos habrá que crear funciones de entrada y salida desde consola y las funciones de tratamiento u operación propuestas. Se debe crear además un programa principal que permita solicitar datos al usuario, seleccionar las operaciones a realizar y mostrar los resultados por pantalla. Se valorará que exista la posibilidad de entrada y salida de datos utilizando ficheros en disco y que se programen funciones adicionales además de las propuestas.

- Polinomios en x . Se deberán incluir las funciones de suma y resta, multiplicación, derivada e integral y cálculo del valor para una x dada.
- Matrices. Se deberán incluir las funciones de suma y resta, multiplicación de matrices, multiplicación por un escalar, transpuesta y determinante.
- Complejos. Se deberán incluir las funciones de suma y resta, multiplicación, división y conversión entre representaciones. Todas las funciones se harán para representación cartesiana y polar.
- Geometría en el espacio. Se deberán definir tipos de datos para puntos, rectas y planos e implementar las funciones de distancia e intersección (o pertenencia, en el caso de un punto).
- Aritmética multiprecisión con enteros. Se deberán incluir las funciones de suma y resta, multiplicación y división, esta última al menos con un entero en precisión simple como divisor.
- Operaciones de bits multiprecisión. Se deberán incluir las operaciones lógicas **and**, **or**, **xor**, **nand**, **nor**, **xnor**, la negación y el complemento a 2, así como desplazamiento de bits a izquierda y derecha y búsqueda del primer **0** y del primer **1** tanto desde la izquierda como desde la derecha. La entrada y salida de datos debe poder hacerse en binario y en hexadecimal.

Programación gráfica y animaciones.

Los ejercicios de este apartado consisten en aplicaciones gráficas, animaciones en su mayor parte, que se programarán utilizando la biblioteca **mgsLib**. Todos ellos requieren de una fase de configuración y otra de ejecución. La primera se puede realizar desde consola, aunque se valorará que se utilice la entrada de datos de forma gráfica también.

- Representación animada de funciones $y = f(x, t)$, donde t representa el tiempo. Se deben poder configurar los límites de representación en todas las dimensiones, así como el incremento temporal. La función se deberá poder cambiar en el código fuente, aunque se valorará que se pueda seleccionar en ejecución. Se recomienda comenzar con la función de propagación de una onda, $y = A \text{sen}(wt - kx)$
- Representación en perspectiva de una función en el espacio $y = f(x, z)$. Se deben poder configurar los límites de representación en todas las dimensiones. En el caso de utilizar una malla para la representación, se podrá configurar también el incremento entre curvas en x y z . La función se deberá poder cambiar en el código fuente, aunque se valorará que se pueda seleccionar en ejecución.

- c. Simulación del movimiento de un cuerpo circular que se desplaza dentro de un polígono, rebotando al colisionar con sus aristas. Se podrá configurar el número de lados y el radio del polígono, así como el del círculo, y su velocidad inicial. En principio se trata de movimiento con velocidad lineal constante y un cuerpo perfectamente elástico. Se valorará poder configurar también el coeficiente de elasticidad.
- d. Simulación del movimiento de un polígono cuyo centro se desplaza por la pantalla con velocidad lineal uniforme y además rota a su alrededor con velocidad angular constante, y rebota al chocar contra los extremos de aquélla. Se podrá configurar el número de lados, radio y las velocidades lineal y angular.
- e. Simulación de un conjunto de cuerpos circulares que se mueven por la pantalla con velocidad lineal constante y que rebotan entre ellos y con los extremos de la pantalla. Se podrá configurar el radio y las velocidades de los cuerpos. El programa deberá gestionar al menos tres cuerpos, aunque se valorará que su número sea también configurable.
- f. Desde un punto de la pantalla se generan objetos que se mueven con velocidad lineal constante y desaparecen al salir de la pantalla. Cada vez que se desaparece un objeto se genera uno nuevo, de forma que el número de objetos en pantalla permanece constante. Debe poder configurarse el número de objetos y la posición del origen. Se deja a la creatividad del grupo de trabajo las consideraciones acerca de los objetos, su velocidad, etcétera.
- g. Realizad un programa que muestre mediante colores el conjunto de Mandelbrot. Se debe poder realizar ampliaciones de regiones seleccionadas de manera dinámica.
- h. Simulación de arte con hilos o hilorama. Dado uno o varios conjuntos de puntos, se trazarán líneas de distinto color entre puntos según cierto patrón. Por ejemplo, con dos conjuntos de puntos alineados, desde todos los de un conjunto a todos los de otro; para un conjunto dispuesto como los vértices de un polígono regular, desde cada punto a todos los demás no adyacentes, etcétera. Se deben poder configurar varios tipos, realizar superposiciones, etcétera.
- i. Simulación de espejos. Se dispone de un conjunto de espejos, cuya orientación puede cambiarse dinámicamente. Desde un punto de la pantalla y con una dirección también cambiabile, se traza un rayo de luz y todas sus posibles reflexiones en los espejos.
- j. Se tiene que simular una serie de móviles que se desplazan horizontalmente a diferentes alturas en pantalla, con distintas velocidades y sentidos, y que o bien rebotan al llegar a un extremo o bien aparecen de nuevo por el extremo opuesto. En este escenario se debe simular un objeto que cae desde la parte superior de la pantalla y sigue un movimiento uniformemente acelerado, y sus posibles interacciones con los móviles. El número de móviles y la altura en la que se desplazan deben ser configurables. El objeto debe poder dejarse caer dinámicamente.

Opción 2.- Ejercicio de simulación.

El segundo trabajo a realizar en las prácticas consiste en un proyecto de simulación de algún sistema físico utilizando la biblioteca **msLib** para la representación gráfica y animación de tal simulación. Un ejemplo de este tipo de sistemas que es interesante simular sería una esfera con cierta elasticidad que se desliza sobre un plano inclinado y acaba cayendo a una superficie plana y rebotando sobre ella. El proyecto debería permitir configurar parámetros tales como la inclinación del plano, el coeficiente de rozamiento entre él y la esfera y la elasticidad de ésta, para posteriormente mostrar el sistema en funcionamiento, con la pelota rodando primero y luego cayendo y rebotando sobre el plano.

Para llevar a cabo este proyecto se deben seguir y documentar los pasos que se indican a continuación:

1. Elegir adecuadamente el sistema físico considerando, sin entrar en gran detalle, su comportamiento para identificar qué parámetros lo rigen y cómo se comporta y evoluciona. De este primer estudio se pueden esbozar los procesos de configuración y simulación para el programa.
2. Estudiar en detalle el sistema, identificando las leyes físicas que lo rigen y su formulación. De esta forma se identificará cómo los parámetros de configuración determinan la simulación del sistema y la animación asociada a la misma. Una vez realizado este estudio se definirán los parámetros a configurar en el programa, así como el rango de valores que pueden tomar, bien intrínsecamente o por decisión de diseño.
3. Una vez conocido el sistema se debe estructurar adecuadamente el programa para realizar tanto la configuración como la simulación.
 - a. En primer lugar se diseñará la entrada de datos por parte del usuario, es decir la configuración. Se valorará que se realice dentro de la propia interfaz gráfica ofrecida por **mgsLib**.
 - b. Posteriormente se establecerá la parte dinámica del sistema y su simulación. Se identificará cómo evoluciona el sistema y cómo esto se refleja de forma gráfica. Se establecerán los distintos estados de ejecución y los algoritmos que rigen el sistema en cada uno de ellos.
4. Por último se realizará el programa basándose en lo anterior y se realizarán las verificaciones adecuadas hasta que todo funcione según el diseño del sistema.

A continuación aparecen algunos ejemplos de sistemas que pueden simularse de manera sencilla. Se valorará sin embargo la creatividad y la iniciativa de los grupos de trabajo en la elección del sistema a simular.

- Tiro parabólico, con perturbaciones adicionales como viento, o efectos como caída sobre una superficie deformable, etcétera.
- Móvil magnetizable que atraviesa un espacio perturbado por diversos campos magnéticos.
- Móvil que atraviesa una superficie con desniveles que afectan su movimiento.
- Sistema de cuerpos esféricos que colisionan, como una bola de billar impactando sobre otras.
- Móvil que desciende en el seno de un fluido con una cierta densidad, y se ve afectado por corrientes que afectan a su movimiento.
- ...

Existe la posibilidad de realizar la representación gráfica de la simulación en tres dimensiones, con un motor 3d desarrollado enteramente sobre **mgsLib**. Si algún grupo de trabajo está interesado deberá comunicárselo al profesor de la asignatura para que les facilite la información y ayuda necesaria.