

CGRAPHIC: una herramienta gráfica para la enseñanza de los fundamentos de la programación (usando C)

Antonio J. Fernández, Jesús Millán Sánchez
Dept. de Lenguajes y Ciencias de la Computación, ETSII,
Campus de Teatinos, Universidad de Málaga
29071 Málaga
e-mail: afdez@lcc.uma.es

Resumen

Se describe una herramienta multimedia de apoyo docente para la enseñanza de los conceptos básicos de la programación. Esta herramienta consiste en un tutorial-simulador gráfico accesible en la Web que permite al alumno ahondar, gráfica y gradualmente, en los conceptos fundamentales de la programación imperativa.

1. Motivación y trabajo relacionado

En las ingenierías universitarias, la programación es una de las asignaturas básicas que más tiempo necesita para su comprensión y además, es un hecho reconocido que el lenguaje C es elegido en muchas universidades españolas como primer lenguaje de programación. Particularmente, [3] discute las adecuación de utilizar C como primer lenguaje de programación en las ingenierías. Entre sus ventajas citamos su practicidad, la generalidad de uso, motivación para el alumno, existencia de numerosas aplicaciones industriales escritas en C así como la disponibilidad gratuita de múltiples entornos para C. La principal desventaja consiste en que “es un lenguaje difícil tanto de enseñar como de aprender en un primer curso” universitario.

Un aspecto que contribuye a una mejor comprensión de la programación es la visualización dinámica de sus conceptos coordinada con la ejecución gradual de las sentencias fuente. En realidad existen muchos trabajos con fines educativos para facilitar el aprendizaje, aunque la mayoría se enfocan en el nivel físico sobre el funcionamiento interno de la máquina [2,5] y otros incluyen algún tipo de animación visual [4,6,9]. Sin embargo, se aprecia una carencia de simuladores orientados a

lenguajes de alto nivel (e.g., [1] simula ejecuciones de código en PASCAL).

Para reducir en alguna medida la dificultad de enseñar y aprender los conceptos básicos de la programación usando el lenguaje C, hemos elaborado una herramienta que permite ahondar, de forma gradual y gráfica, en los fundamentos de la programación imperativa. Esta herramienta, que llamamos *CGRAPHIC*, es original (i.e., no está basada en otras) y complementa, de forma gráfica e interactiva, los conocimientos adquiridos en clase, los cuales se explican de forma intuitiva.

2. CGRAPHIC como software educativo

CGRAPHIC puede ser fácilmente catalogado como software educativo [8] puesto que cumple muchas de sus características tales como

- facilidad de uso,
- capacidad de motivar al alumno,
- tratamiento de temas relevantes,
- versatilidad y accesibilidad,
- interactividad con el alumno y
- originalidad.

Además, CGRAPHIC cumple las principales funciones del software educativo [8], tales como:

- *una función instructiva* pues orienta el aprendizaje de los estudiantes;
- *una función motivadora* ya que incluye elementos gráficos que captan la atención de los alumnos, mantienen su interés y los guía hacia los aspectos más importantes de los conceptos;
- *una función evaluadora* porque da una respuesta inmediata a las acciones de los alumnos según éstos las demanden.

3. Teoría + práctica

CGRAPHIC integra un tutorial on-line, de los conceptos básicos del lenguaje C, y un simulador visual de la ejecución de programas escritos en C, combinando pues una parte teórica con otra eminentemente práctica.

En la parte teórica, cada concepto básico de programación viene acompañado primeramente de una explicación teórica similar a la que pueda aparecer en cualquier libro básico de programación. Primero se define un concepto y luego se muestran, a nivel teórico, ejercicios explicativos del mismo. Además, se proponen una lista de conceptos afines que el alumno puede consultar de forma inmediata on-line.

En la parte práctica, asociado a cada concepto existe un conjunto de ejercicios interactivos cuya ejecución es visualizada gráficamente. En este contexto práctico, se combinan características de un intérprete de C con las de un depurador del mismo. Básicamente CGRAPHIC visualiza la ejecución de un programa diseñado mediante el lenguaje de programación C. La versión actual incluye un conjunto amplio de programas -de dificultad diversa- que se asocian a conceptos fundamentales de la programación imperativa, tales como: tipos básicos, tipos compuestos (registros-estructuras, arrays, etc), constantes, variables locales y globales, ámbitos de visibilidad, estructuras de control, bucles, direcciones de memoria, punteros, ficheros, funciones, procedimientos, paso de parámetros y tipos abstractos de datos (TADs).

4. El entorno de visualización

El entorno gráfico de ejecución de la parte práctica consta de lo siguientes elementos, los cuales son fácilmente identificables en la Figura 1: (1) *Una pantalla gráfica* para mostrar una representación gráfica de los elementos del programa C a medida que éstos son creados, modificados y/o eliminados; (2) *Una pantalla de código* que muestra el código del ejercicio actual que se está representando; además en cada uno de los estados intermedios, se resalta cuál es la siguiente línea de código que va a ser ejecutada; (3) *una pantalla de salida*, que indica tanto la salida estándar (es decir el monitor) del programa como instrucciones al usuario para un correcto

funcionamiento; (4) *un control del modo de ejecución* (ver explicación posterior); (5) *un cuadro de entrada*, que proporciona interactividad pues permite introducir datos como si del teclado se tratase; (6) un botón para iniciar la visualización (*ejecución*) y (7) *un botón de acción*, para pasar de un estado a otro en la ejecución.

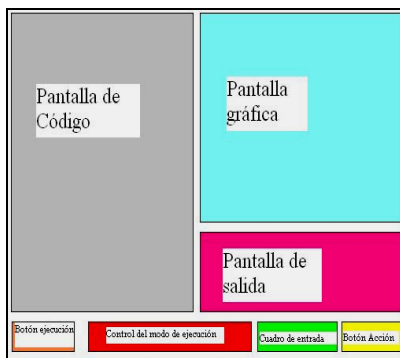


Figura 1. Entorno de ejecución.

La implementación: Debido a las limitaciones de espacio, comentaremos brevemente la implementación de CGRAPHIC. La parte teórica ha sido diseñada básicamente mediante páginas HTML; el entorno gráfico ha sido desarrollado en Java y sólo requiere un programa navegador para ser ejecutado. La Figura 2 muestra el esquema de todas las clases empleadas: se han implementado una serie de objetos genéricos que son usados para visualizar diferentes conceptos de programación. Entre esos objetos tenemos: variables, arrays, funciones, punteros, mapas de memoria, ficheros, estructuras y buffer de teclado. Cada objeto se implementa mediante una clase Java y cada ejercicio (e.g., *Ex*), implementado en otra clase Java (e.g., *Ex.java*), contiene un programa C que se carga durante la ejecución en la ventana de código del entorno (ver Figura 1). La ventana de código es definida en el fichero *Code.java*. El resto del entorno gráfico se implementa en el fichero *GraphicBox.java*, en cual es compartido por todos los ejercicios (los cuales también comparten la misma plantilla, implementada en el fichero *Template.java*). Más información puede ser encontrada en la siguiente dirección:

http://campusvirtual.uma.es/fundinfo/online/English_CGRAPHIC/default.htm.

Modos de ejecución: Se permiten dos modos posibles de ejecución a elegir por el usuario: un modo *directo*, en el cual el programa se ejecuta tal y como lo haría en un entorno clásico de programación en C, y un modo *pausado o paso a paso*, en el cual los programas son ejecutados instrucción a instrucción y la resolución (trazas) de cada instrucción es mostrada gráficamente por pantalla. En este modo, el usuario indica cuándo se quiere pasar al siguiente estado mientras se observan cómo varían los elementos del programa (variables, punteros, memoria,...) en los estados intermedios; consiste pues en una interpretación gráfica del ejercicio.

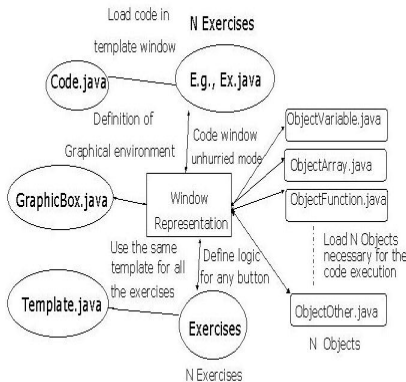


Figura 2. Esquema de Clases.

Ejemplos: La Figura 3 muestra un estado intermedio derivado de la ejecución paso a paso de un ejercicio (concretamente, el de intercambio de enteros usando una función *swap*). En esta figura se observan algunos de los *objetos* tratados gráficamente en la herramienta tales como las variables globales (en verde), las variables locales (en azul), el mapa de memoria, la línea de código a ejecutarse (resaltada en rojo), etc.

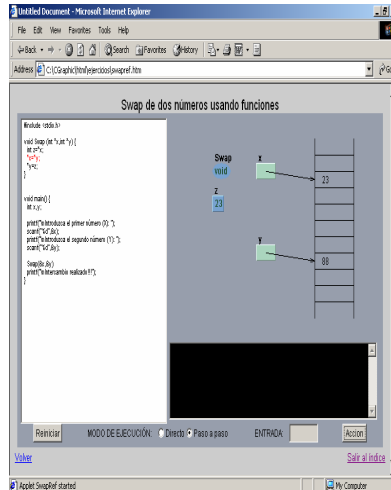


Figura 3. Estado intermedio de ejecución *paso a paso*.

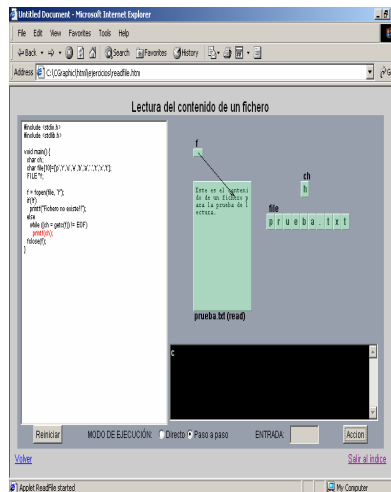


Figura 4. Otros estados intermedios. Ficheros-Arrays.

Las Figuras 4 y 5 muestran el estado intermedio de la ejecución pausada de otros

ejercicios relacionados con otros conceptos (en la Figura 4 la lectura de ficheros y las cadenas de caracteres; en la Figura 5, las direcciones de memorias, punteros, funciones y el TAD pila).

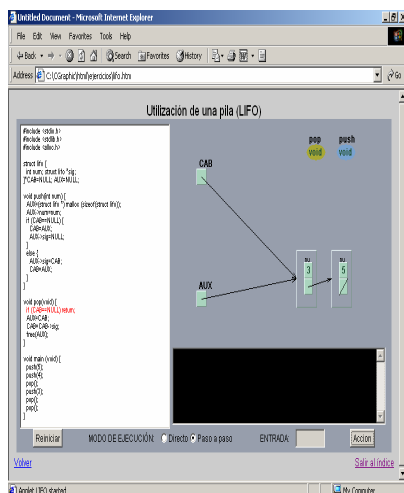


Figura 5. Otros estados intermedios. TADs-punteros

5. Conclusión

Hemos presentado CGRAPHIC una herramienta original diseñada exclusivamente para la enseñanza, usando C, de los conceptos fundamentales de la programación imperativa. Esta herramienta reduce el enorme hueco existente entre la asimilación de la teoría y su utilización práctica cuando se utiliza el C como primer lenguaje de programación. CGRAPHIC está totalmente operativa y accesible en la siguiente dirección web:

<http://campusvirtual.uma.es/fundinfo/> (enlace On-line).

Evaluación Con respecto a la evaluación de la herramienta, todavía es pronto para analizar su uso por parte de los alumnos pues ha comenzado a ser operativa en este segundo cuatrimestre. Sin embargo diremos que, desde el punto de vista del profesor, nos está resultando muy útil a la hora de

explicar conceptos difíciles tales como la "entrada de datos bufferizada". En este sentido estamos usando el típico cañón para proyectar en las clases prácticas la visualización paso a paso de ejercicios, la cual es complementada con la explicación teórica del profesor. Esto facilita la comprensión del concepto por parte del alumno.

Al final del curso, proyectamos realizar una evaluación plena, siguiendo criterios ya establecidos [7].

Referencias

- [1] Brette J-F. Transparent running and contextual help to learn and to teach an imperative language. *SIGCSE Bulletin*, 27(2):7-12, 1995.
- [2] Campbell R. Introducing computer concepts by simulating a simple computer. *SIGCSE Bulletin*, 28(3):9-11, 1996.
- [3] Fernández A.J., Trella M., Aranda M.C. y Galindo P. *Nuevas tecnologías y metodologías para la enseñanza de una introducción a la programación de ordenadores en ingenierías*. III Congreso Chileno de Educación Superior en Computación, Chile, 2001.
- [4] Garralón J., Contreras D. y Núñez P. *PERICO: Programa Educativo de programación Imperativa con visualización de elementos físicos*. II Jornadas Andalúses de Informática Gráfica, pp:63-72, 2000.
- [5] Hassapis G. An interactive electronic book approach for teaching computer implementation of industrial control systems. *IEEE Transactions on Education*, 46(1):177-184, 2003.
- [6] Henderson W. Animated models for teaching aspects of computer systems organizations. *IEEE Transactions on Education*, 37 (3):247-256, 1994.
- [7] Iglesias O., Paniagua C. y Pessacq R. Evaluation of University Educational Software. *Computer Applications in Engineering Education*, 5(3):181-188, 1997.
- [8] Marques P. *Software educativo. Guía de uso y metodología de diseño*. Estes, 1995.
- [9] Robbins S. y Robbins K. A microprogramming animation. *IEEE Transactions on Education*, 41(4):293-300, 1998.