

Aplicación del uso de interfaces tangibles de usuario a la enseñanza de lenguajes de programación.

José María Rodríguez Corral*, Arturo Morgado Estévez⁺

*Departamento de Ingeniería Informática, Escuela Superior de Ingeniería, ⁺Departamento de Ingeniería en Automática, Electrónica, Arquitectura y Redes de Computadores, Escuela Superior de Ingeniería.

josemaria.rodriguez@uca.es

RESUMEN: A menudo los estudiantes tienen dificultades a la hora de comprender los conceptos de la Programación Orientada a Objetos (POO). Este artículo presenta una aportación a la enseñanza de los lenguajes de POO a través de un enfoque orientado a los juegos basado en la interacción con interfaces tangibles de usuario (TUIs). El uso de un tipo específico de TUI comercial (cubos Sifteo), donde varios dispositivos de tamaño reducido poseen capacidades sensitivas, de comunicación inalámbrica y de salida al usuario (visuales), se aplica a la enseñanza del lenguaje de programación C#, puesto que el funcionamiento de estos dispositivos se controla mediante programas escritos en dicho lenguaje. Para realizar los ensayos experimentales, se seleccionó una muestra de estudiantes con un nivel suficiente de conocimientos en programación imperativa, la cual se dividió en dos grupos: El primero asistió a un curso introductorio estándar de C#, mientras que el segundo asistió a un curso experimental que, además de los contenidos del anterior, incluía un programa de demostración que ilustraba determinados conceptos básicos de la POO haciendo uso de las características de las TUIs. Finalmente, ambos grupos participaron en dos pruebas de evaluación: Un test para evaluar el grado de adquisición de los conceptos básicos de la POO y un ejercicio de programación en C#. El análisis de los resultados obtenidos a partir de las pruebas anteriores indica que el grupo de estudiantes que participaron en el curso que incluía las demostraciones con TUIs mostraron un mayor nivel de interés durante la exposición del curso que el grupo que participó en el curso introductorio estándar. Además, los estudiantes del grupo experimental obtuvieron una mejor calificación global.

PALABRAS CLAVE: estrategias de enseñanza/aprendizaje; innovación educativa; interacción persona-computador; interfaces tangibles de usuario; lenguajes de programación; redes inalámbricas de sensores

INTRODUCCIÓN

En la actualidad, el ordenador forma parte de la vida de los estudiantes y la mayoría de ellos posee cierta experiencia en el uso de juegos de ordenador, motores de búsqueda, redes sociales, procesadores de textos y aplicaciones de mensajería instantánea. Sin embargo, los estudiantes a menudo tienen dificultades a la hora de comprender los conceptos abstractos de la programación orientada a objetos (clases, objetos, métodos, atributos, herencia, etc.), puesto que se les hace difícil encontrar la equivalencia de dichos conceptos en la vida real (1). Asimismo, la programación tradicionalmente se enseña como la creación de programas basados en la escritura de texto, que no es familiar ni atractiva para los estudiantes. La mayoría de ellos tienen más experiencia en el manejo del ratón y los entornos gráficos (interfaz gestual) que en el uso de la línea de órdenes (interfaz textual).

El diseño orientado a objetos es muy natural, dado que en la vida real pensamos en términos de objetos, que poseen ciertas propiedades y comportamientos. Sin embargo, a la hora de escribir programas, los estudiantes iniciados en la programación imperativa suelen adoptar la visión tradicional, consistente en instrucciones que se ejecutan y estructuras que definen el flujo de control de los programas (2). En este sentido, los programadores esperan que los programas tengan un inicio y un final, de modo que puedan entenderse a partir de una lectura de principio a fin. Sin embargo, comprender el funcionamiento de un programa orientado a objetos implica entender qué son los objetos y cómo se transfieren los mensajes entre ellos para realizar determinadas tareas (3).

En este trabajo, pretendemos traer para los estudiantes los conceptos abstractos de POO al "mundo visible". El uso de los cubos Sifteo (4) permite materializar conceptos básicos de

la POO tales como *objeto*, *atributo* y *método*, de este modo se consigue estimular en los estudiantes el aprendizaje significativo (5). En primer lugar, se desarrollará un programa de demostración con objeto de ilustrar algunos conceptos de la POO haciendo uso de las características de los cubos Sifteo. Posteriormente, se elaborará material para dos cursos introductorios de POO en C#: Uno de ellos incluirá el programa de demostración ya referido, mientras que el otro consistirá en un curso estándar. Finalmente, se diseñarán dos pruebas de evaluación: Un test para evaluar el grado de adquisición de los conceptos básicos de POO, y un ejercicio de programación.

Una vez que los dos grupos de estudiantes - experimental y control - hayan completado ambas pruebas de evaluación, se evaluarán los resultados y se extraerán las conclusiones, las cuales se presentarán en la última sección del artículo.

FUNDAMENTOS TECNOLÓGICOS

Una *red inalámbrica de sensores* (6) consta de un conjunto de nodos computerizados que cooperan para realizar una amplia variedad de tareas. En (7), se propone la aplicación de los principios tecnológicos de las *redes de sensores* a las *Interfaces Tangibles de Usuario* (TUIs) con el fin de abrir nuevas posibilidades en cuanto a modos de interacción.

La concepción de una interfaz de usuario como una red de sensores (SNU), utilizada para desarrollar las "Siftables" (7) y más tarde los cubos Sifteo (4, 8), toma la forma de una TUI distribuida donde varios dispositivos de tamaño reducido poseen capacidad sensorial, de comunicación inalámbrica y de salida dirigida al usuario (visuales). Estos dispositivos dan lugar a una plataforma genérica de interacciones que combina las capacidades flexibles de presentación de una interfaz gráfica

de usuario (GUI) con el carácter físico de una TUI, a las cuales se unen las capacidades propias de una red de sensores.



Figura 1. Un juego para los cubos Sifteo.

Los cubos Sifteo (4, 8) son la versión evolucionada y comercial de las "Siftables" (figura 1). Cada cubo reemplaza el antiguo microcontrolador de las "Siftables" por un microprocesador ARM de 32 bits, incluye una memoria Flash de 8 MB y está alimentado por una batería recargable de polímero de litio (figura 2). Posee una pantalla TFT en color de 128x128 pixels que también funciona como botón, y utiliza una señal de radio de 2.4 GHz para comunicarse con el ordenador anfitrión. Los cubos pueden detectarse entre ellos mediante una tecnología de localización de objetos en proximidad propia de Sifteo.

En una tarea típica de manipulación, se proporciona a cada cubo mediante radiotransmisión una imagen que representa una instancia de los datos a ordenar. Las manipulaciones físicas realizadas por el usuario sobre el conjunto de cubos son detectadas y usadas como entradas que se suministran al programa concreto que controla la tarea de manipulación. La realimentación visual que se genera durante el transcurso de la tarea se muestra al usuario a través de las pantallas de los cubos, mientras que la realimentación auditiva se genera por el ordenador que ejecuta el programa de control de la tarea de manipulación.

Es posible desarrollar aplicaciones para los cubos Sifteo usando un *kit de desarrollo de software* (SDK) proporcionado por Sifteo (9), que contiene *Siftdev* - la aplicación de escritorio que provee el *soporte en tiempo de ejecución* de las aplicaciones para los cubos Sifteo (*Siftrunner*) junto con herramientas adicionales para desarrolladores -, un *simulador* (*Siftulator*) que permite comprobar el funcionamiento de las aplicaciones sin utilizar los cubos (figura 3) y, finalmente, una serie de *aplicaciones de ejemplo* y una *documentación de referencia* sobre las clases y los paquetes de C# contenidos en la API de Sifteo, que permite a los desarrolladores utilizar todas las características funcionales de los cubos Sifteo (tecnologías de detección de acciones de manipulación y pantalla TFT).

C# (10) es un lenguaje de programación orientado a objetos simple y moderno que incorpora características de seguridad en cuanto a la utilización de los tipos de datos (11). Tiene su origen en la familia de lenguajes de C y resulta familiar para los programadores de C, C++ y Java. ISO/IEC ha

desarrollado una especificación del lenguaje (estándar ISO/IEC 23270).



Figura 2. Cubos Sifteo en la base de carga.

PROGRAMA DE DEMOSTRACIÓN PARA LOS CUBOS SIFTEO APLICADO AL ESTUDIO DEL LENGUAJE DE PROGRAMACIÓN C#

En esta sección se describe brevemente una sencilla aplicación de demostración escrita en C# que emplea la tecnología de los cubos Sifteo para ilustrar algunos conceptos importantes de la POO (12-13), como *clase*, *objeto*, *atributo*, *método* e incluso *herencia*. Asimismo, ciertos conceptos de la programación basada en eventos como los *eventos* (14) y los *tipos delegados* de C# (11), pueden ilustrarse de igual modo.

El programa¹ desarrollado muestra secuencias ordenadas de colores en las pantallas de los cubos ("Screen Colors"). Dichas secuencias progresan de modo independiente en cada cubo por medio de pulsaciones en su pantalla (*ButtonEvent*). Después del último color se muestra de nuevo el primero de la secuencia. La acción de agitar un cubo (*ShakeStartedEvent* y *ShakeStoppedEvent*) reinicia la secuencia de colores para dicho cubo, cuya pantalla vuelve a mostrar el primer color.

Esta aplicación permite ilustrar algunos conceptos básicos de la *programación orientada a objetos* y la *programación basada en eventos*, para aquellos estudiantes que se inician en el conocimiento de estos paradigmas de programación mediante el aprendizaje del lenguaje C#.

RESULTADOS Y DISCUSIÓN.

En la Escuela Superior de Ingeniería (Universidad de Cádiz), la asignatura "Fundamentos de Informática" se imparte en el primer curso del Grado en Ingeniería en Tecnologías Industriales (15). Los contenidos de dicha asignatura giran en torno a dos temas centrales: Introducción a la Informática (Arquitectura de ordenadores (16) y Sistemas Operativos (17)) y Programación de Ordenadores en Lenguaje C (18).

Se ha seleccionado una muestra de treinta alumnos con edades comprendidas entre 18 y 19 años: Quince de ellos (el grupo control) han participado en un curso² estándar de POO en C#, que incluye demostraciones prácticas de programas de

¹ Ver código fuente de la aplicación *Screen Colors* en anexo *PI_14_002_Anexo 1.pdf*.

² Ver curso de C# en anexo *PI_14_002_Anexo 2.pdf*.

ejemplo en ordenador que no hacen uso de la contribución tecnológica y didáctica de los cubos Sifteo. Los otros quince (el grupo experimental) han participado en un curso en el que sí se utilizan las aplicaciones de demostración para los cubos Sifteo descritas en la sección anterior.

La tabla 1 muestra un conjunto de indicadores, que corresponden a los valores medios de los resultados obtenidos por cada grupo (experimental y control), junto con las correspondientes desviaciones típicas:

- Percepción subjetiva del nivel de claridad (de 1 a 4) en la exposición del curso.
- Percepción subjetiva del nivel de interés (de 1 a 4) en la exposición del curso.
- Tiempo empleado (expresado en minutos) en el estudio de los contenidos del curso.
- Calificación obtenida (de 0 a 10) para los tests 1 y 2 respectivamente.
- Tiempo empleado (expresado en minutos) en la realización de los tests 1 y 2 respectivamente.
- Percepción subjetiva del nivel de dificultad (de 1 a 4) en la realización de los tests 1 y 2 respectivamente.

A partir de los resultados obtenidos, podemos observar que, en lo que respecta a los indicadores correspondientes al nivel de interés y a la calificación obtenida (test2), el grupo experimental obtiene claramente valores más altos que el grupo control. En primer lugar, un mayor grado de interés en la exposición de un tema se relaciona con una mayor motivación en el proceso de aprendizaje del estudiante.

Indicador	Grupo experimental		Grupo control	
	Valor	Desv. típica	Valor	Desv. típica
Nivel de claridad (presentación)	3.57	0.62	3.32	0.69
Nivel de interés (presentación)	4.00	0.65	2.93	0.57
Tiempo empleado (estudio)	134.36	30.81	139.93	15.87
Calificación obtenida (test 1)	8.24	0.54	8.22	0.31
Calificación obtenida (test 2)	6.64	1.23	4.20	1.38
Tiempo empleado (test 1)	12.07	0.70	13.40	0.48
Tiempo empleado (test 2)	12.64	0.81	12.80	1.05
Nivel de dificultad (test 1)	2.43	0.90	2.53	0.83
Nivel de dificultad (test 2)	2.57	0.49	2.81	0.56

Tabla 1. Valores de los indicadores y desviaciones típicas para los dos grupos de estudiantes.

Por otra parte, el curso de C# para el grupo experimental incluye un programa de demostración que utiliza los cubos Sifteo. Por tanto, la obtención de un valor más alto del indicador calificación obtenida (test 2) en este grupo puede explicarse en base al uso de ejemplos más significativos e iluminadores en lugar de un conjunto típico de programas estándares de C# para ejecutar y probar en un ordenador.

Finalmente, a pesar de que las diferencias más importantes entre el grupo experimental y el grupo control en cuanto a los resultados obtenidos se encuentran en los valores de los indicadores correspondientes al grado de interés y a la calificación obtenida (test 2), debe tenerse en cuenta que todas las diferencias existentes están a favor del grupo experimental. Aunque se trata de un estudio preliminar, este hecho es significativo y debe tomarse en consideración.

CONCLUSIONES

En este trabajo se ha aplicado el uso de una *interfaz de usuario tangible distribuida* (DTUI) a la enseñanza del lenguaje de programación orientado a objetos C#, puesto que el funcionamiento de los cubos Sifteo se controla mediante programas escritos en este lenguaje. Dicha interfaz funciona como una red inalámbrica de sensores conectada a un ordenador a través de un compacto transmisor USB de radio.

A partir del análisis de resultado presentado en la sección anterior, es posible concluir que la contribución tecnológica de los cubos Sifteo - utilizados como una TUI distribuida mediante la cual los conceptos básicos de la POO son representados de un modo tangible y visible - a la enseñanza del lenguaje C# ejerce una influencia positiva en el aprendizaje de este lenguaje, así como de dichos conceptos básicos.

REFERENCIAS

1. Yan, L. Teaching Object-Oriented Programming with Games. *ITNG 2009 6th International Conference on Information Technology: New Generations*. Las Vegas, Nevada (USA), **2009**, 969-974.
2. Overmars, M. Learning Object-Oriented Design by Creating Games. *IEEE Potentials*, **2004**, 23(5), 11-13.
3. Rosson, M.B., Carroll, J.M. Climbing the Smalltalk Mountain. *ACM SIGCHI Bulletin*, **1990**, 21(3), 76-79.
4. Sifteo Inc. *Sifteo Cubes*. <https://www.sifteo.com/product>. Último acceso el 20 de enero de 2014.
5. Fink, L.D. *Creating Significant Learning Experiences: An Integrated Approach to Designing College Courses*. John Wiley and Sons, Inc. **2003**.
6. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E. (2002). Wireless Sensor Networks: A Survey. *Computer Networks*, **2002**, 38(4), 393-422.
7. Merrill, D., Kalanithi, J., Maes, P. Siftables: Towards Sensor Network User Interfaces. *TEI'07 1st International Conference on Tangible and Embedded Interaction*. Louisiana (USA), **2007**, 75-78.
8. Merrill, D., Sun, E., Kalanithi, J. Sifteo Cubes. *CHI 2012 Conference on Human Factors in Computing Systems*. Austin (Texas), **2012**, 1015-1018.
9. Sifteo Inc. *Sifteo Software Development Kit for Windows (version 1.1.3)*. https://s3.amazonaws.com/updates.sifteo.com/Sifteo_SDK_win_1_1_3.exe. Último acceso el 20 de enero de 2014.
10. Michaelis, M. *Essential C# 4.0* (3rd ed.). Pearson Education, Inc. **2010**.
11. Microsoft Corporation. *C# Language Specification. Version 5.0*. <http://www.microsoft.com/en-us/download/details.aspx?id=7029>. Último acceso el 27 de enero de 2014.
12. Détienne, F. Assessing the Cognitive Consequences of the Object-Oriented Approach: A Survey of Empirical Research on Object-Oriented Design by Individuals and Teams. *Interacting with Computers*, **1997**, 9(1), 47-72.
13. Rumbaugh, J. *Object-Oriented Modeling and Design*. Prentice Hall. **1991**.

14. Faison, T. (2006). *Event-Based Programming: Taking Events to the Limit*. Apress. **2006**.
15. European Commission. *The Bologna Process - Towards the European Higher Education Area*.
http://ec.europa.eu/education/higher-education/bologna_en.htm. Último acceso el 8 de enero de 2014.
16. Hennesy, J.L., Patterson, D.A. *Computer Architecture. A Quantitative Approach* (4th ed.). Morgan-Kaufmann. **2006**.
17. Tanenbaum, A.S., Woodhull, A.S. *Operating Systems. Design and Implementation* (3rd ed.). Prentice Hall. **2006**.
18. Kernighan, B.W., Ritchie, D.M. *The C Programming Language* (2nd ed.). Prentice Hall. **1998**.

ANEXOS

PI_14_002_Anexo 1.pdf (código fuente de la aplicación *Screen Colors*)

PI_14_002_Anexo 2.pdf (curso de introducción a C#)