

INTRODUCCIÓN A LA COMPUTACIÓN Y A LA PROGRAMACIÓN: UN ENFOQUE VISUAL

Ing. Raúl V. Ramírez Velarde
Ciencias Computacionales - Campus Monterrey
Aulas VII - 410

Participantes

Carlos Salmerón. Alumno ARQ.

Ing. Enrique Arratia. (Durante 8° y 9° semestre ISC).

Introducción y Antecedentes

El propósito de este proyecto es el construir un curso introductorio de computación, incluyendo el funcionamiento real de la computadora y sus componentes; y de programación. Para lograr este objetivo, el problema se dividió en dos, lo cual en realidad creó dos proyectos en uno. Cada nuevo proyecto requiere de diferentes enfoques y tecnologías.

Para la parte de introducción a la computación, se diseñaron una serie de esquemas que explican el funcionamiento de cada uno de los componentes de la computadora (CPU, teclado, disco, monitor, mouse, etc.) y se implementaron en un paquete de dibujo de tres dimensiones 3D Studio. Esto tuvo como resultado una serie de gráficas altamente atractivas visualmente, muy al estilo del libro How Computers Work [WHITE]. La diferencia principal con nuestro enfoque, es que cada gráfica va acompañada de una explicación exhaustiva del funcionamiento del dispositivo, pero tomando en cuenta una audiencia con conocimientos casi nulos sobre computación. Las explicaciones contenidas en el libro mencionado, además de estar en inglés, son muy pobres. Para lograr un mejor entendimiento del funcionamiento de los dispositivos nos fundamentaremos en los conceptos estudiados en el curso de Equipo Periférico impartido a las carreras de ISE e ISC y en el material de apoyo diseñado para dicha materia. Las explicaciones sin embargo, utilizarán una terminología a nivel introductorio. Al final de este documento se incluye una gráfica de ejemplo.

Para la parte de programación, se diseñó un software interactivo que permite la creación de programas por medio de la edición de diagramas de flujo llamado Newt. Dichos diagramas de flujo son mapeados a lenguajes de alto nivel (C, Pascal, Basic, etc.) que el usuario puede ejecutar para verificar si el programa está correcto. La ventaja de este enfoque, es que la utilización de una interfase gráfica intuitiva de diseño de programa, permite aligerar la carga conosciativa sobre el estudiante. Esto es, que el estudiante se ve libre de entender el funcionamiento de un medio ambiente de programación, sistema operativo, editor, depurador y compilador, y se concentra en el diseño de algoritmos. El programa cuenta adicionalmente con ayuda en línea y ejemplos. La sintaxis es extremadamente sencilla y el programa constantemente corrige al estudiante la sintaxis (usando mensajes en los cuales la claridad es lo más importante) de tal forma que el programa siempre ejecuta, aunque no haga lo que el usuario quiere. Debido a que Newt no diseña el programa del usuario, sino que sólo se encarga de la sintaxis y de la edición del programa, Newt en realidad es un laboratorio virtual donde el estudiante puede

experimentar. Dicha experimentación puede ser guiada por la propia curiosidad del estudiante o, más académicamente, por una serie de prácticas que tocan sucesivamente diferentes conceptos, las cuales ya se encuentran diseñadas.

Newt ya fue presentado en la X Reunión de Intercambio de Experiencias de Estudios sobre Educación en Agosto de 1992 [RAMIREZ] y aquí se presentan solamente las mejoras y actualizaciones que se han realizado sobre el programa desde entonces. Estas son: versiones para MS Windows, Macintosh y Solaris, ayuda en línea, mejora en la captura de argumentos, mejoras en la interpretación del programa, mejoras en la ejecución, inclusión de llamadas a subrutinas, inclusión de Qbasic, y sobre todo, una mejora sustancial en la interfase de diseño del diagrama de flujo. Durante el verano de 1996 se tiene la firme intención de incluir una herramienta de visualización de algoritmos. Se presenta además una versión mejorada del libro de texto y se presenta un manual de trabajo orientado a estudiante de niveles inferiores al profesional.

Fundamento Teórico

Los fundamentos de la aplicación de programas educativos en la educación se pueden basar en tres teorías del aprendizaje que se enlistan en seguida. No se describen las teorías en este documento, pero se refiere al lector a las referencias bibliográficas enlistadas al final.

I. Teoría Conductista [COOPER]

II. Aprendizaje Significativo [SMITH]

III. Empowering en la Era de la Información [VAN DER EMBSE]

Basadas en las tres teorías descritas, se pueden encontrar una serie de características que deben caracterizar al aprendizaje de la era de la información.

1. El aprendizaje debe ser estimulante.
2. El aprendizaje debe ser práctico.
3. El aprendizaje debe ser orientado.
4. El aprendizaje debe ser constructivo.
5. El aprendizaje debe ser estructurado.
6. El aprendizaje debe ser funcional.
7. El aprendizaje debe ser funcional.
8. El aprendizaje debe ser evaluado.
9. El aprendizaje de ser en base a metas.
10. El aprendizaje debe ser personalizado.
11. El aprendizaje debe ser activo.
12. El aprendizaje debe ser total.

Las herramientas computacionales para la enseñanza propuestas en este proyecto cumplen con todas las características mencionadas arriba.

Modelo de diseño integral de la instrucción

Robert D. Tennyson [TENNYSON a], propone el Paradigma Cognoscitivo como base para la enseñanza tecnológica y propone un modelo de desarrollo de Instrucción Ayudada por Computadora [TENNYSON b] que se relaciona directamente con las estructuras mentales involucradas en el aprendizaje descritas en su paradigma cognoscitivo. La tabla 1 muestra ésta relación. Tamto el paradigma conoscitivo de Tennyson y como su modelo de diseño integral de la instrucción han sido incorporados al fundamento teórico del proyecto.

Metas Educativas

Componentes del modelo de desarrollo	Adquisición del conocimiento	Utilización del conocimiento			
Conocimiento	Declarativo	Procedural	Contextual	Complejidad Cognoscitiva	Sistema Total
Objetivos del Aprendizaje	Información Verbal	Habilidad Intelectual	Habilidad Contextual	Estrategia Cognoscitiva	Proceso Creativo
Tiempo de Instrucción	10%	20%	25%	30%	15%
Estrategia de Enseñanza	Exposición	Práctica	Resol. de Problemas	Complejidad y Dinámica	Autodirec- cionamiento

Tabla 1 Relación entre las estructuras mentales que componen el aprendizaje, el Modelo de Desarrollo de IAC y las estrategias de enseñanza.

El diseño del curso de introducción a la computación y a la programación cumple con todos los niveles del modelo de Tennyson. La parte de introducción a la computación, utilizando las gráficas tridimensionales cumple con la primera columna. El programa Newt cumple con las otras cuatro. Adicionalmente, utilizando tanto las gráficas computacionales en 3D para la introducción a la computación como el software newt, es logra que la educación cumpla con las 12 características de la educación mencionadas anteriormente.

Todas la personas que han utilizado Newt se han dado cuenta que Newt es un programa enteramente diferente a los que han utilizado hasta ese momento. Esta sensación se deriva de la interfase de usuario avanzada con que cuenta Newt. En Newt se utilizaron muchos de los conceptos avanzados de diseño de interfases de usuario interactivas que se pueden encontrar en [SNEIDERMAN] y [KEARSLEY]. Adicionalmente, conceptos específicamente enfocados a los programas educativos encontrados en [SCHAEFERMEYER] y [HUNKA] se aplicaron para el diseño de Newt.

La visualización de conceptos en la sección de introducción a la computación (con gráficas en 3D) se hace en forma directa. Es decir, que se realiza un representación Fotorealista del dispositivo a explicar. No obstante, en algunas secciones donde la naturaleza del dispositivo es altamente compleja se utilizan las técnicas que se mencionan en el siguiente párrafo, como en la sección de explicación del funcionamiento interno del microprocesador, la composición del ALU y en la interacción entre el CPU y sus periféricos.

Las principales técnicas utilizadas para la visualización de conceptos utilizadas en Newt (y en algunos esquemas de funcionamiento de dispositivos de la computadora) son: Abstracción, Discretización, Coloración, Codificación por Geometría y Animación. Adicionalmente, se hace uso extensivo de la manipulación directa. Se refiere al lector a la bibliografía se desea saber más acerca de estos conceptos [BROWN], [FOLEY] y [CUNNINGHAM].

Posibles Extensiones a Realidad Virtual

Las gráficas fueron realizadas en el paquete 3D Studio de AutoDesk directamente en tres dimensiones. Estas gráficas pueden importarse directamente a los paquetes de realidad virtual que existen actualmente y crear un museo virtual en 3 dimensiones, en tiempo real sobre el funcionamiento de la computadora. Al presente, se encuentra en el proceso de obtener fondos para realizar esta extensión al proyecto y ya se está trabajando en un prototipo para ser utilizado a través de Live3D de NetScape.¹

Objetivos

- Realizar una serie de gráficas en 3 dimensiones, que sean visualmente estimulantes que expliquen el funcionamiento de los componentes más importantes de la computadora.
- Elaborar un paquete integrado de diseño, simulación y visualización de algoritmos para enseñar a programadores novatos las estructuras mentales esenciales para el diseño de programas.
- Desarrollar un libro de texto que integre los conceptos de funcionamiento de computadoras y que guíe al estudiante en el aprendizaje de algoritmos de programación, depuración, eficientización, visualización y aplicación.
- Desarrollar un cuaderno de trabajo orientado a estudiantes de niveles inferiores a profesional para que puedan utilizar tanto el software como las gráficas.

Ventajas e Importancia

- El enfoque visual de la enseñanza del funcionamiento interno de la computadora ha probado ser universal (se refiere al lector al libro How Computers Work), en el sentido de que funciona para explicar conceptos complejos a una gran cantidad de personas, dedicados a muy diferentes actividades, con diferentes escolaridades y edades.
- El enfoque visual de la programación por medio de diagramas de flujo aligera la carga cognoscitiva del estudiante permitiendo concentrarse en el aprendizaje de los conceptos fundamentales de programación.
- El material instructivo podrá ser utilizado por el estudiante en su casa.

Metodología

¹ Para Febrero de 1997 se han hecho realidad las extensiones a realidad virtual y las animaciones en formato Qtime. Las animaciones quedaran terminadas para Mayo de 1997. Los archivos de realidad virtual (VRML) quedarán terminados para Setiembre de 1997.

- 1. Recopilación de versión 2.5 de Newt en Windows y Solaris. [Ago-Dic 95]
- 2. Capacitación de programador en lenguaje de desarrollo. [Ago-Sep 95]
- 3. Diseño de versión 3.0 de Newt [Oct-Nov 95]
- 4. Diseño de editor de diagramas de flujo [Nov-Dic 95][Ene-Mar 96]
- 5. Diseño de gráficas en 3D Studio. [Ene-Dic 96]
- 6. Intérprete del programa. [Abr-May 96]
- 7. Ayuda y ejemplos en línea. [May-Jun 96]
- 8. Revisión, aplicación y actualización de libro de texto. [Jun-Jul 96]

Resultados Obtenidos

- 1. Se tienen las siguientes gráficas: Externas: teclado, monitor, consola, impresora tinta, impresora de pines. Internos: CPU, tarjeta madre, CRT, Tarjeta de video.
- 2. El software Newt 25. Se ha compilado en Windows, DOS y Solaris.
- 3. El software Newt 3.0 lleva un avance del 85%. Para Junio de 1996 estará al 100%.
- 4. El Libro de texto lleva un avance del 85%.
- 5. El cuaderno de trabajo lleva un avance del 85%.
- 6. Pruebas preliminares del software Newt.

Conclusiones

La naturaleza visual, la manipulación directa y la interfase de usuario avanzada de Newt estimulan la curiosidad de los estudiantes y aligeran la carga cognoscitiva.

Para que el software pueda tener éxito, es necesario que sea estable. Es decir, que no aparezcan errores que sorprendan y arruinen el trabajo del estudiante, por lo que es muy importante enfocar esfuerzos importantes en la confiabilidad del software. También el ambiente de trabajo debe ser intuitivo. Es decir que se requiera de poca capacitación para comenzar a trabajar con el software. De aquí el largo tiempo de desarrollo de Newt y la importancia de esta última versión.

Otro aspecto importante de Newt es que el aprendizaje obtenido con este paquete es independiente del lenguaje. Una vez entendidos los conceptos más indispensables de diseño de algoritmos, a los estudiantes se les facilita entender cualquier lenguaje de Programación.

El enorme éxito del libro How Computers Work, Garantiza el éxito del enfoque visual que se ha seguido para la enseñanza del funcionamiento de la computadora. Para que esto funcione al 100%, las explicaciones deben ser claras y en lenguaje al nivel de la audiencia. Esto se garantiza gracias a la experiencia obtenida en la clase de equipo periférico. Las explicaciones sobre el funcionamiento de la computadora se basan en aquellas encontradas en [DUNCAN], [TANNENBAUM] y [JOURDAIN].

Bibliografía

BROWN, J. R. y Cunningham, S. "Visualization in Higher Education". En Academic Computing. Marzo 1990.

CUNNINGHAM, S. y Zimmerman, W. "Visualization in Teaching and Learning Mathematics". Sufragado por Mathematical Association of America. 1990.

DUNCAN, Ray. Advanced MSDOD Programming. Microsoft Press, 1986.

ELLSON, R. "Visualization at Work". En Academic Computing. Marzo 1990.

FOLEY, J. D. y Wallace V. L. "The Art of Natural Graphic Man-Machine Conversation". En Proceeding of the IEEE. Abril 1974.

HUNKA, S. "Designing Guidelines for CAI Authoring Systems". En Educational Technology. Noviembre 1989.

JOURDAIN, Robert. Programmer's Problem Solver. Brady, 1992.

KEARSLEY, G. y Halley, R. "Designing Interactive Software". Park Row Press. La Jolla, California. 1985.

RAMÍREZ, Raúl. NEWT, Una Herramienta de Programación Gráfica para la Enseñanza del Pensamiento Algorítmico. 1990.

SCHAEFERMEYER, S. "Standards for Instructional Computing Software Design and Development". En Educational Technology. Junio 1990.

SHNEIDERMAN, B. "Designing the User Interfase". Addison-Wesley Company Publishing Inc. 1987.

SMITH, P. E. "Some Learning and Instructional Theory Considerations for the Development of Computer Related Instructional Materials". En Educational Technology. Noviembre 1989.

TANNEBAUM, Andrew. Organización de Computadoras. Prentice Hall, 1985.

TAYLOR, James "Novex Analysis: A Cognitive Science Approach to Instructional Design". En: Educational Technology. Mayo-Junio 1994 pp. 5-12.

TENNYSON, Robert D. *A Proposed Cognitive Paradigm of Learning for Educational Technology*. Educational Technology. Junio 1990. pp. 16-19.

TENNYSON, Robert D. *Integrated Instructional Design Theory: Advancements from Cognitive Science and Instructional Technology*. Educational Technology. Julio 1990. pp. 9-15.

VON DER EMBSE. *Powerful ideas about Empowerment*. Manage, Nov/Dic 1989. pp. 25-28.

WHITE, Ron. How Computers Work. (PC COMPUTING) Ziff-DAvis Press, 1993.