

Programación básica para adolescentes

Guillermo Londoño Acosta

glondono@icesi.edu.co

Gustavo Adolfo Paz Loboguerrero

gapi20@yahoo.com

Fecha de recepción: 10-02-2007

Fecha de selección: 10-05-2007

Fecha de aceptación: 30-04-2007

ABSTRACT

Adolescents don't show interest in the computer programming, they consider it difficult and boring. This paper explores the idea to teach the new generations of adolescents to program in a way that attracts them, that is both easy and amusing.

This paper shows the experience at the Icesi University with high school students to teach basic programming by developing video games, with tool easy to use.

KEY WORDS

KPL, BASIC, programming, teaching, KPL Project, games, Morrison&Schwartz, Phrogram

RESUMEN

Los adolescentes están perdiendo interés por la programación de las computadoras, la consideran una temática complicada y poco motivadora. Es necesario iniciar en la programación a las nuevas generaciones de una manera fácil, divertida y muy gráfica; como por ejemplo, aprender los conceptos básicos de la programación construyendo juegos de video, con herramientas que oculten la complejidad y permitan conocer y aprender a utilizar esos elementos básicos.

Este artículo muestra el material desarrollado y los resultados de la experiencia que hemos tenido en el pro-

grama de Ingeniería de Sistemas de la universidad Icesi, con estudiantes de grados 10 y 11, en la enseñanza de los elementos básicos de la programación, a través de la construcción de juegos de video en 2D.

PALABRAS CLAVE

KPL, BASIC, programación, enseñanza, Proyecto KPL, juegos, Morrison&Schwartz, Phrogram, Alice

Clasificación Colciencias: B

INTRODUCCIÓN

En los años ochenta los jóvenes tenían la oportunidad de conocer el mundo de la programación de computadoras a través de un lenguaje de programación para aficionados, fácil de aprender y de utilizar, de propósito general y con características avanzadas pero simples para los principiantes; aunque los expertos lo criticaban por su lentitud y sencillez. Este lenguaje, llamado BASIC, se adecuaba muy bien a las computadoras personales de la época, y era el lenguaje estándar usado por los adolescentes como mecanismo para introducirse en el mundo de la programación. Un adolescente podía experimentar y conocer, de forma autodidacta, los elementos básicos de la programación.

En los años noventa la complejidad de las computadoras personales y la llegada de los ambientes gráficos exigían nuevas características a los lenguajes de programación, que BASIC no tenía. Lenguajes ya no para aficionados como C, C++ y VisualBasic permitían desarrollar aplicaciones complejas, que explotaban toda la potencialidad de las computadoras personales y de las interfaces gráficas usuario. El desarrollo de las computadoras personales permitió un gran crecimiento de muchas compañías de software. Mientras que las compañías de software para micros de los ochenta vendían cientos o miles de copias de sus programas, las compañías de software de esta década vendían decenas e incluso centenas de miles de copias. Las computadoras personales se convirtieron en herramientas usadas principalmente para ejecutar aplicaciones escritas por otros; la programación empezó a ser

menos importante para la creciente mayoría de usuarios y de poco interés para los adolescentes.

Hoy en día, lenguajes como Java, C# y VBasic.NET, con sus populares entornos de desarrollo Eclipse, JDeveloper, Visual Studio.NET, etc., no son adecuados para que un adolescente, estudiante de secundaria, se inicie en el mundo de la programación. Esta carencia en la actualidad de un lenguaje básico estándar ha conducido a la eliminación de la enseñanza de la programación en muchos colegios, y aquellos que lo hacen con lenguajes complejos como C, Java, etc., están logrando que muchos jóvenes pierdan todo interés por la programación.

Nuestros adolescentes se han convertido sólo en usuarios de paquetes como Office; algunos piensan que un ingeniero de sistemas se dedica a estudiar Office y otros paquetes en profundidad, perciben la informática como el uso de aplicaciones, pues desconocen el poder de la programación para comunicarse con las computadoras y lograr que hagan lo que uno quiere, como por ejemplo construir juegos de video, tema de su total interés.

La diferencia entre dar instrucciones a una persona y dar instrucciones a una computadora está en la complejidad de cada instrucción y en el lenguaje utilizado. El computador no puede suponer ni imaginar nada, las instrucciones que le damos son más elementales, precisas, en un estricto orden y sometidas a reglas de un lenguaje más restrictivo que los lenguajes utilizados por los humanos. No estamos acostumbrados a dar instrucciones en forma tan lógica, precisa, ordenada y con un

lenguaje tan restrictivo; este es un excelente ejercicio mental para un adolescente.

Con el objetivo de motivar de nuevo a los estudiantes de grados 10 y 11 por la programación, nos dimos a la tarea de desarrollar un material para los colegios que permita a un adolescente de grado 10 o grado 11 conocer y aplicar los principios básicos de la programación en la construcción de un juego de video en 2D.

KPL COMO ENTORNO DE PROGRAMACIÓN

Existe un entorno de programación moderno llamado *KPL*, siglas en inglés de Kids Programming Language, con un lenguaje de programación tan simple y legible como BASIC, que facilita la programación de juegos y programas educativos con gráficos y sonido.

KPL1 fue desarrollado con la dirección de Jonah Stagner y sus colegas, Jon Schwartz, antiguo director de

programación de Microsoft, y Walt Morrison, antes ingeniero de NCR. A finales del 2006 fue adquirido por Phrogram Company (www.phrogram.com).

KPL1 es un entorno de programación moderno compuesto por un editor, un compilador, un depurador, un entorno gráfico de ejecución y una biblioteca de estructuras, funciones y procedimientos que permiten manipular fácilmente imágenes, sonidos y dispositivos periféricos como el ratón y el teclado.

La versión 2 de KPL, liberada a finales del 2006 y llamada Phrogram, fue desarrollada sobre el framework .NET, incluye no sólo un potente motor 2D como el de la versión original, sino que introduce un motor 3D basado en DirectX 9.0c, el cual permite el uso de controles y joysticks para el manejo de los juegos, y en un futuro cercano los juegos desarrollados en KPL2 se podrán ejecutar en el Xbox



Figura 1. Entorno de programación de KPL.

360. También se han introducido nuevas características en el lenguaje, como la programación basada en clases y el manejo de archivos.

A finales del 2005 escogimos KPL1 como entorno de programación para la construcción del material, y como herramienta en los talleres dictados a los estudiantes de grados 10 y 11, y a los estudiantes de primer semestre de Ingeniería de Sistemas de la Universidad Icesi. Este entorno de programación es bastante amigable, sencillo de aprender, y ha permitido que la programación usando el lenguaje de KPL sea ampliamente aceptada por todos los participantes en los talleres.

LA RAZÓN PARA EMPLEAR JUEGOS

La programación se puede utilizar para resolver problemas matemáti-

cos, o problemas relacionados con el manejo de información en una organización, o para animar imágenes en la pantalla del computador al construir juegos de video, o en muchos otros campos. Todos estos dominios de aplicación requieren los mismos elementos básicos de programación. Empleamos los juegos de video para aprender a programar, puesto que este es un tema de mucho interés para los adolescentes; además, les permite ver resultados gráficos e inmediatos de sus programas y estimula su creatividad, mientras se les enseñan conceptos básicos como el de variable, operador, tipo de datos, instrucción de selección y de repetición, arreglo, estructura, método, función y clase.

Estos son algunos ejemplos de juegos clásicos que se pueden lograr con KPL1:

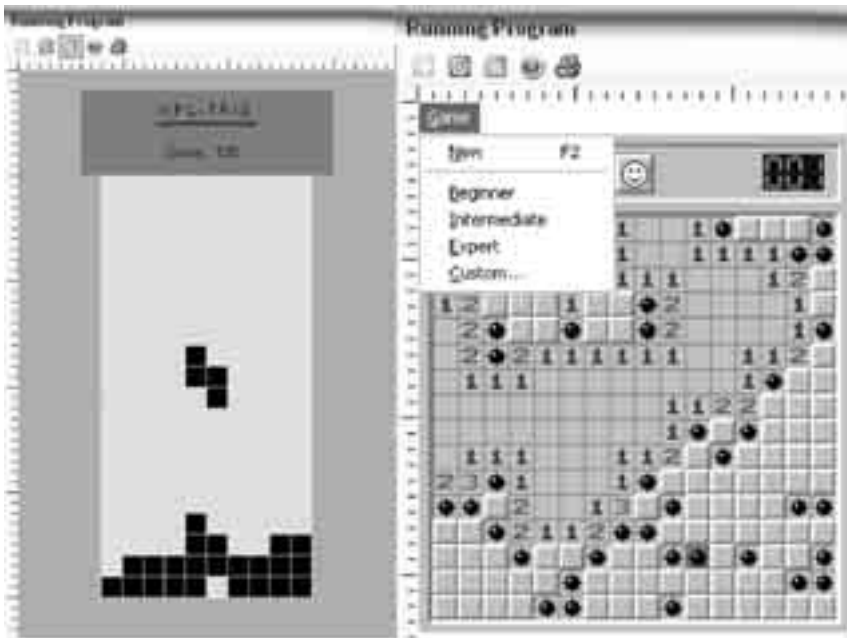


Figura 2. Ejemplos de juegos realizados en KPL1.

y con KPL2:

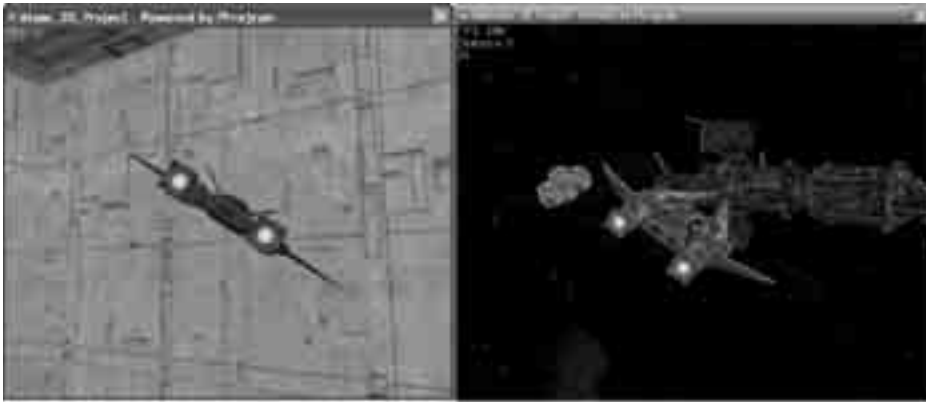


Figura 3. Ejemplos de juegos en 3D hechos en la versión 2 de KPL llamada Phrogram.

MATERIAL DESARROLLADO PARA LOS COLEGIOS

En diciembre de 2005 invitamos estudiantes de grados 10 y 11 de varios colegios de Cali a la charla “Aprender Algoritmos Desarrollando Juegos de Video”. Esta invitación tuvo tanta acogida que decidimos construir un material que permitiera a los estudiantes de los colegios conocer los elementos básicos de la programación construyendo juegos de video. El material está publicado en <http://www.icesi.edu.co/colegios/kpl/index.htm>

El material para el taller tiene una parte básica y una parte avanzada con el siguiente contenido:

Contenido básico

1. Introducción a la programación.
2. Manejo del lápiz.
3. Usar la geometría y la trigonometría en la construcción de figuras geométricas trazadas por las imágenes animadas.
4. Técnicas básicas de animación en 2D: Traslación, rotación y colisiones.

5. Elementos básicos de la programación: identificador, tipo de dato, palabra reservada de un lenguaje, operador, variable, literal, arreglo, instrucción de selección, instrucción de repetición.
6. Gif animados.
7. Movimiento de imágenes con el teclado.
8. Estructuras.
9. Funciones y métodos.

Contenido avanzado

1. Juegos con dos jugadores.
2. Uso del ratón en los juegos.
3. Desplegar estados del juego.
4. Cuadros fluidos.
5. Mover el fondo.
6. Ajuste dinámico de la velocidad de las animaciones.
7. Manejo de disparos y explosiones.
8. Uso de sonidos.
9. Uso de vectores.

A lo largo de todo el taller se construye un juego en el que se van aplicando gradualmente todos los conceptos discutidos.

TALLERES REALIZADOS

Hemos realizado cinco talleres con estudiantes de grados 10 y 11, cuatro abiertos a los colegios en general y uno dirigido sólo a los estudiantes de grado 11 del colegio Jefferson de

la ciudad de Cali. Este fue un taller de 40 horas, dictado en las salas de la Universidad Icesi durante un período de dos meses, en el que los estudiantes asistían tres horas por semana.



Figura 4. Fotos del taller con el Colegio Jefferson en 2006-1.

Para este año 2007:

1. Vamos a utilizar Phrogram.
2. Haremos los cambios necesarios para que todos los programas desarrollados en KPL1 incluyan las nuevas características de Phrogram.
3. Vamos a complementar la programación de juegos mediante la inclusión de talleres sobre producción y edición de audio e
4. Vamos a desarrollar nuevos ejemplos que exploten toda la potencialidad del motor 3D de Phrogram.

La Tabla 1 resume los resultados combinados de dos encuestas realizadas en los talleres.

Tabla 1. Resultados de encuestas realizadas sobre el taller.

Número de Encuestados:		22					Calificación	
							Bajo 1 – Alto 5	
1.	Motivación para estudiar el taller							
	Calificación	1	2	3	4	5	Total	
	Votos	0	0	2	8	12	22	
2.	Conocimiento alcanzado en programación							
	Calificación	1	2	3	4	5	Total	
	Votos	0	0	9	9	4	22	
3.	Motivación para hacer otro taller de programación							
	Calificación	1	2	3	4	5	Total	
	Votos	1	0	5	6	10	22	
4.	Calificación del material							
	Calificación	1	2	3	4	5	Total	
	Votos	0	0	1	6	15	22	
5.	Utilidad del Taller							
	Calificación	1	2	3	4	5	Total	
	Votos	0	0	2	9	11	22	
6.	Calificación del Taller							
	Calificación	1	2	3	4	5	Total	
	Votos	0	0	0	9	13	22	

CONCLUSIONES

Hace treinta años los adolescentes disponían de tecnologías basadas en texto y pobres en gráficos, sonido, animación, etc.; para ellos tenía mucho sentido la frase “Una imagen vale más que mil palabras”. Hoy en día los jóvenes utilizan diversos medios para obtener información, aprender y entretenerse. Los adolescentes de hoy tienen una mente muy gráfica y utilizan tecnologías con texto, imágenes, sonido, animación e interactividad. La frase más adecuada para los jóvenes de hoy es: una animación interactiva vale más que mil imágenes. Desafortunadamente seguimos enseñando los principios básicos de la programación utilizando el texto como único medio, lo cual es frustran-

te y poco motivador para una mente gráfica como la de los jóvenes.

En los últimos años hemos tenido pocas innovaciones en la enseñanza y el aprendizaje de los principios básicos de la programación. Los profesores en los colegios y en las universidades son reacios a cambiar su forma de enseñar algoritmos y programación. Afortunadamente ya están apareciendo entornos de programación modernos como KPL o como Alice, que permiten, mediante nuevos enfoques, desarrollar el pensamiento algorítmico en los estudiantes. En la universidad Carnegie Mellon se construyó una herramienta llamada Alice,^{1,2} un entorno de programación que permite gráficos en 3D, progra-

mación visual, animación e interactividad, y permite el desarrollo de aplicaciones de una manera fácil y rápida, debido a que oculta la complejidad al desarrollador. Alice permite concentrarse en el diseño de la solución y no en la forma de implementar los algoritmos, la interfaz gráfica, la animación, la interactividad, etc.

Alice se está utilizando como herramienta en los cursos introductorios a ciencias de la computación, para enseñar los conceptos fundamentales de programación, en las siguientes instituciones educativas norteamericanas:³

- Bucknell University
- California Lutheran University
- California State University at Humboldt
- Camden County College
- Carnegie Mellon University
- Clemson University
- Colorado School of Mines
- Community College of Philadelphia
- Cornell University
- Duke University
- Georgetown College
- Haverford College
- Ithaca College
- Manor College
- Mississippi Valley State University
- Plymouth State University
- Saint Edward's University
- Saint Joseph's University
- Saint Lawrence College
- San Diego State University
- Sierra Nevada College

- Southwestern University
- Tompkins Cortland Community College
- University of Colorado
- University of Illinois
- University of Mississippi
- Virginia Tech
- And several high schools

Por ejemplo, en la Universidad de Saint Joseph los estudiantes toman el curso CSC 1301 Building Virtual Worlds, de tres créditos, como introducción a la programación utilizando el software Alice. En este curso los estudiantes usan mundos virtuales, animados, interactivos y en 3D, para entender los elementos básicos de la programación. Luego toman el curso CSC 1401 de introducción a la programación de computadores, de cuatro créditos, utilizando un lenguaje como java, en el cual aprenden el concepto de variable, estructura selectiva y repetitiva, arreglo, método, objeto y clase.

Con en el apoyo de National Science Foundation (NSF) se llevó a cabo un proyecto de investigación en Saint Joseph's University (SJU) and Ithaca College (IC),⁷ sobre la efectividad del uso de Alice en la enseñanza de los conceptos básicos de programación, y en especial de la programación orientada a objetos. El estudio se enfocó en estudiantes con poca fundamentación en matemáticas y en programación que tomaron el curso CSC 1301 y demostró que los estudiantes que tomaron el curso CSC 1301, mejoraron sus calificaciones de C a B en el curso CSC 1401, comparándolas con las obtenidas históricamente por los estudiantes que también tenían las características ya mencionadas y no

tomaron el curso CSC 1301. Además, el porcentaje de estudiantes que avanzaban a cursos superiores de programación pasó del 47% al 88%. También señala el estudio que los estudiantes que toman el curso CSC 1301 tienen mayor facilidad para entender conceptos más avanzados de programación.

Herramientas como KPL y Alice nos brindan la posibilidad de explorar nuevos enfoques para el aprendizaje de los conceptos básicos de la programación. Tanto en los colegios como en las universidades un primer curso de programación con un enfoque pedagógico que enfatice en desarrollar la capacidad de abstracción para entender un problema, diseñar una solución algorítmica e implementarla en una computadora, contribuiría mucho a la formación de un adolescente, no importa si eventualmente decide llegar a ser un ingeniero de sistemas o no; estas habilidades y conocimiento lo beneficiarán por el resto de su vida, pues tendrá que desenvolverse en un mundo inmerso en la tecnología. Por otra parte, el conocimiento de los elementos básicos de la programación debe ser del dominio de toda persona que desee explotar la potencialidad de herramientas de productividad personal como Excel.

BIBLIOGRAFÍA

1. <http://www.alice.org>
2. <http://www.aliceprogramming.net>
3. <http://www.aliceprogramming.net/overview/overview.html>
4. <http://www.icesi.edu.co/colegios/kpl/index.htm>

5. <http://www.kidsprogramminlanguage.com/>
6. <http://www.phrogram.com/>
7. Moskal, Barb; Lurie, Deborah y Cooper, Stephen. Evaluating the Effectiveness of a New Instructional Approach. Pensilvania: Carnegie Mellon University, 2006. 5 p.
<http://www.alice.org/publications/pubs/EvaluatingTheEffectivenessOfANewApproach.pdf>

CURRÍCULOS

Guillermo Londoño Acosta. Físico de la Universidad del Valle. Magíster en Física del Instituto Venezolano de Investigaciones Científicas (IVIC), Caracas, Venezuela. Profesional asociado a Investigación del Instituto Venezolano de Investigaciones Científicas (IVIC), Caracas, Venezuela. Asistente de Investigación de la Universidad de Wisconsin, Milwaukee, U.S.A. Magíster en Ingeniería de Sistemas de la Universidad del Valle. Cursos de especialización en Ingeniería del Software en la Universidad Politécnica de Madrid. Actualmente Director del Programa de Ingeniería de Sistemas de la Universidad Icesi, y Profesor en el Área de Modelado de Sistemas de Información.

Gustavo Adolfo Paz Loboguerro. Estudiante de Ingeniería de Sistemas e Ingeniería Telemática en la Universidad Icesi. Actualmente Monitor del Taller KPL que se realiza en la misma Universidad y coautor del material desarrollado. ☼