

Utilización de prácticas con gráficos 3D animados en la enseñanza de la programación orientada a objetos

Javier Macías, José María Gutiérrez, José Ramón Hilera, José Antonio Gutiérrez

Dpto. Ciencias de la Computación

Universidad de Alcalá

28871 Alcalá de Henares

e-mail: {javier.macias, josem.gutierrez, jose.hilera, jantonio.gutierrez}@uah.es

Resumen

Se presenta una experiencia real basada en la utilización de prácticas con gráficos animados en tres dimensiones en la enseñanza de la programación orientada a objetos, indicando los resultados obtenidos tanto en relación al aprendizaje como en la motivación de los alumnos.

1. Introducción

Es evidente la importancia que la programación orientada a objetos ha adquirido tanto en la educación universitaria como en la propia industria [1]. Sin embargo, los conceptos clave de la programación y el diseño de este paradigma, como clase, objeto, atributo, método, encapsulación, herencia, polimorfismo, etc. [4] no son fáciles de entender recurriendo únicamente a la explicación teórica.

Por otro lado, los alumnos de programación sufren problemas de falta de motivación [7]. Una de las razones de esta falta de motivación es achacable a las propias prácticas de programación utilizadas en su enseñanza. Éste es el caso de las prácticas basadas en ejemplos poco realistas o muy forzados, o prácticas demasiado complejas, o aquellas que consisten en una entrada simple seguida de un proceso tras el cual se obtiene un triste número en la pantalla, o, por citar un último caso, prácticas cerradas de difícil ampliación por parte del alumno.

Es curioso observar que cuando se utilizan lenguajes como Pascal o Modula en las prácticas, los alumnos culpan al lenguaje de defectos atribuibles a las propias prácticas, tachándolos de

obsoletos e incluso de inútiles en lo relativo a su aprendizaje para la vida real, lo cual además crea en ellos un círculo vicioso desmotivador.

2. Objetivos

Dada la importancia de las prácticas en la enseñanza de la programación en general y de la programación orientada a objetos en particular, y considerando a su vez la necesidad de motivar a los alumnos en su aprendizaje, se buscaron unas prácticas que:

- Cubrieran una parte significativa de los conceptos básicos del diseño y la programación orientada a objetos.
- Fueran suficientemente cortas de forma que pudieran ser realizadas sin requerir grandes esfuerzos por parte del alumno, pero que a su vez resultaran vistosas y estimulantes.
- Fuera posible su ampliación de una forma fácil, si el alumno así lo deseaba.
- Se pudieran realizar utilizando el lenguaje conocido por los alumnos (en nuestro caso concreto, Turbo Pascal[®]) sin recurrir a artificios, aunque deberían ser fácilmente adaptables a cualquier otro lenguaje que soporte la orientación a objetos, como C++ o Java.

Una vez analizadas varias posibilidades de forma que se cubrieran los objetivos perseguidos, nos decantamos por diseñar prácticas fundamentadas en la utilización de gráficos en tres dimensiones y, en concreto, de figuras poliédricas, tal y como se desarrolla a continuación.

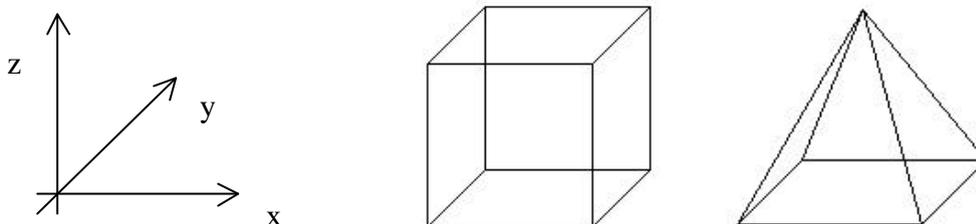


Figura 1. Ejes en perspectiva caballera y apariencia del cubo y pirámide solicitados en la primera práctica

3. Las prácticas

Las prácticas construidas al efecto han sido dos. A continuación se resumen ambas brevemente, así como se indican algunas cuestiones generales que se tuvieron en cuenta en las mismas.

La primera práctica se denomina “Figuras geométricas tridimensionales” [6]. Los alumnos deben crear un cubo y una pirámide cuadrangular (Figura 1) que giren en la pantalla sobre el eje z.

Antes de explicar a los alumnos el diseño recomendado, se les pidió que pensaran sobre cómo desarrollarían ellos la práctica utilizando programación estructurada. En general, ven mucha dificultad, lo cual hace que luego les resulte más impresionante la facilidad con la que se consigue utilizando un buen diseño y programación orientados a objetos.

El tiempo asignado, incluyendo tanto el destinado para su explicación como el utilizado por los alumnos para su desarrollo, fue de dos sesiones de dos horas de duración cada una.

La segunda práctica se denomina “Sistema solar de figuras geométricas” [8]. Los alumnos

deben colocar una figura geométrica en el centro de coordenadas y distribuir otras figuras a su alrededor. Seguidamente, deben animar el sistema creado de forma parecida a la manera en que giran los planetas alrededor del Sol (Figura 2).

El tiempo total destinado a esta práctica fue de una sesión de dos horas.

Para grupos formados por alumnos con poca base en programación orientada a objetos o que nunca habían llegado a escribir y probar un programa orientado a objetos en el ordenador, se añadió una práctica previa introductoria, con una duración de una sesión de dos horas. La práctica consistía en representar varios puntos (objetos) en la pantalla que el usuario podía mover utilizando para ello el teclado [5].

En esta práctica se explica detalladamente y paso a paso todo el proceso de diseño, obteniendo una descomposición modular consistente en un programa principal y una unidad donde se definen dos clases. La primera de las clases representa una coordenada de la pantalla, y a partir de ésta se define un descendiente (utilizando la herencia) que representa un punto de la pantalla.

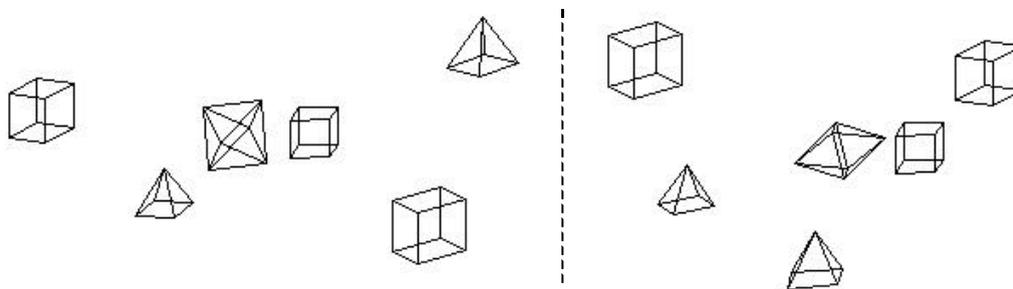


Figura 2. Dos instantáneas de un ejemplo de la segunda práctica

La práctica introductoria permite que los alumnos se familiaricen, entre otros, con los conceptos de clase, objeto y encapsulamiento, siendo también un primer acercamiento a la herencia. Además, al tener que utilizar gráficos, les ayuda a afrontar con más confianza las prácticas 3D posteriores.

Es de destacar que las prácticas aquí presentadas se destinaron fundamentalmente para la enseñanza de programación, por lo que se proporcionaron las ecuaciones necesarias para realizar las proyecciones sobre la pantalla, así como una unidad donde se definía la clase punto, debido a que esta clase debe contener métodos de traslación y rotación, para lo cual es necesario

tener ciertos conocimientos de geometría computacional [2]. Si los alumnos poseen estos conocimientos, pueden desarrollar partiendo de cero toda la práctica, pero entonces sería conveniente destinar algo más de tiempo.

Además, para que el alumno concentrara su atención en la programación orientada a objetos y no se perdiera en detalles auxiliares, se le proporcionaron las rutinas para inicializar y finalizar el modo gráfico, los nombres y sintaxis de las funciones necesarias para dibujar líneas (y puntos, en el caso de la práctica introductoria) en la pantalla, así como vértices predefinidos que le facilitarían la labor de definir tanto el cubo como la pirámide.

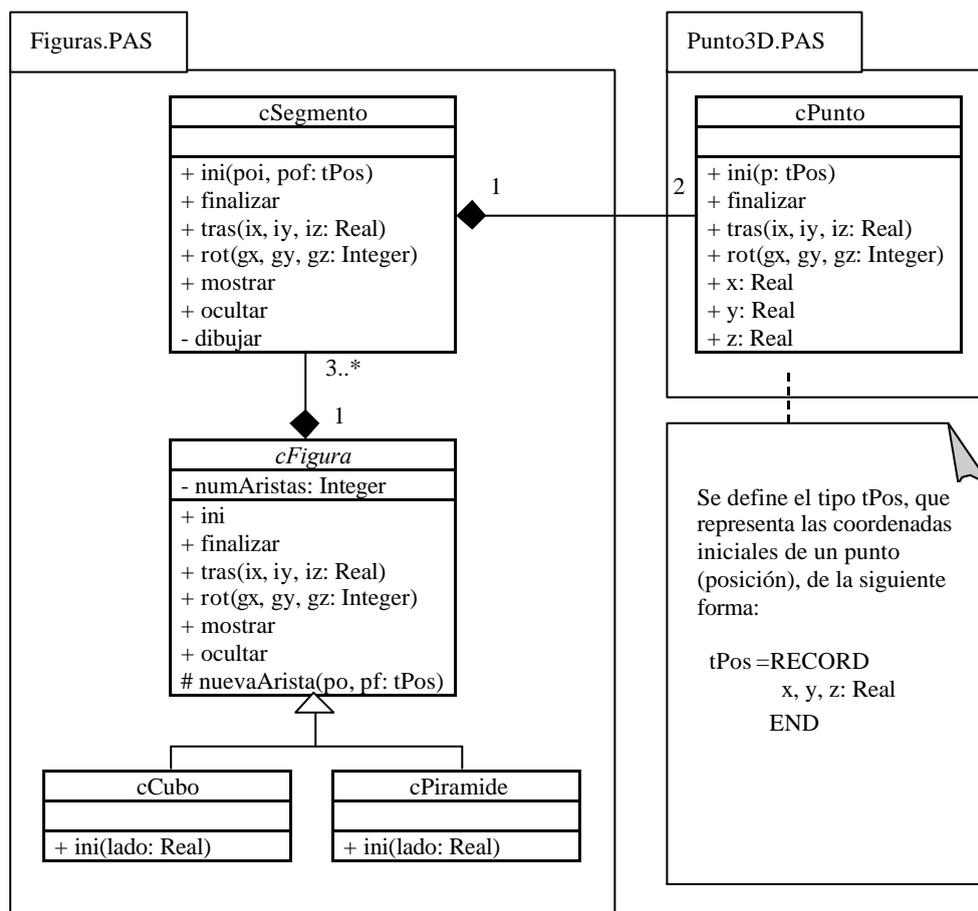


Figura 3. Diagrama simplificado de clases de la primera práctica



Figura 4. Instantáneas de la secuencia de giro de una pirámide

Por último, indicar que estas prácticas han sido utilizadas en el laboratorio de la asignatura "Estructura de Datos" de las titulaciones I.T. en Informática de Sistemas e I.T. en Informática de Gestión de la Universidad de Alcalá desde el curso 1999/00. También han sido utilizadas en la asignatura "Técnicas de Programación Estructurada", impartida a los Sargentos Alumnos del Cuerpo de Especialistas Escala de Suboficiales, en su segundo curso de Enseñanza Militar de Formación, en la Escuela de Técnicas Aeronáuticas (ESTAER) en la Base Aérea de Torrejón (Ejército del Aire).

3.1. Primera práctica

El diseño de la primera práctica se puede acometer fácilmente empleando una estrategia de arriba a abajo (*top-down*), esto es, partiendo de un cubo o una pirámide se va descomponiendo el problema hasta llegar al punto, o empleando una estrategia de abajo a arriba (*bottom-up*), esto es, partiendo del punto, vamos obteniendo elementos más complejos hasta llegar a formar un cubo o una pirámide.

Cualquiera que sea la estrategia empleada (podría usarse una y luego la otra para que los alumnos aprecien ambos enfoques, o incluso un único enfoque mixto), es necesario que el alumno entienda que una figura (poliédrica) no es sino un

conjunto de segmentos, y que un segmento se puede definir por sus puntos extremos. Nótese que se puede asociar inmediatamente el concepto de objeto (y, por tanto, de clase) a las figuras, segmentos y puntos.

Por otro lado, rotar una figura consiste en rotar todos sus segmentos. Y para rotar un segmento, basta con rotar sus puntos extremos y luego trazar una línea entre ambos. La traslación de una figura se hace de forma análoga. Como se puede apreciar, esto no es más que una aplicación del mecanismo de abstracción, que es fundamental en el diseño de sistemas complejos, hecho que se debe recalcar a los alumnos.

También es importante que el alumno entienda la utilidad de definir una clase abstracta figura, de la cual desciendan todos los poliedros. Esto además le permite ver lo sencillo que resulta añadir nuevas figuras. Para que los alumnos puedan entender fácilmente el diseño (Figura 3), se utiliza UML [9]. Nótese que en el diseño aparecen, entre otros, los conceptos de encapsulación y las relaciones de herencia y composición.

En el programa principal, el alumno debe instanciar un objeto cubo y otro pirámide. También definirá una función que reciba como parámetro una figura y muestre en la pantalla una animación consistente en rotar grado a grado dicha figura hasta completar 360°. Esta función se utilizará para introducir el concepto de polimor-

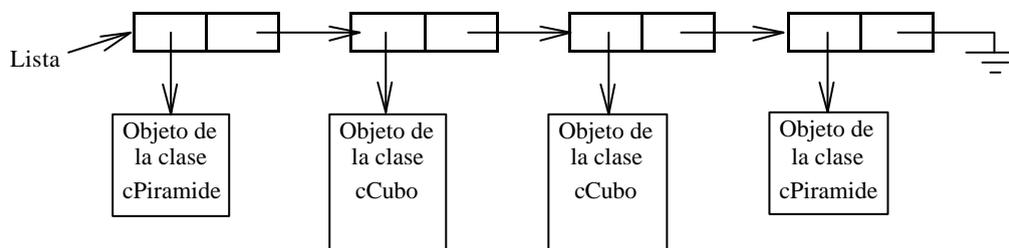


Figura 5. Ejemplo de lista de figuras

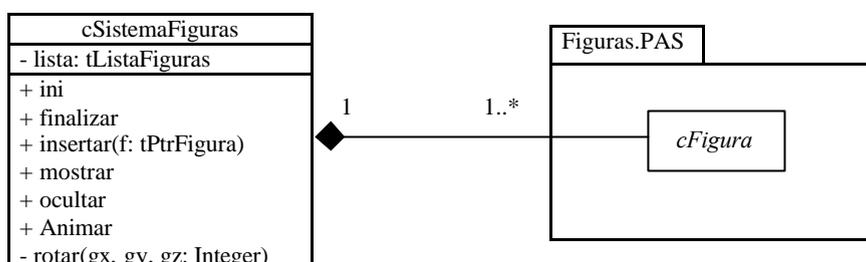


Figura 6. Diagrama simplificado de clases de la segunda práctica

fismo, ya que el alumno podrá comprobar que puede utilizar la misma función tanto para animar un cubo como una pirámide (Figura 4).

3.2. Segunda práctica

La práctica segunda está pensada para que el alumno se familiarice con el uso de punteros a objetos, para lo cual se utiliza una lista dinámica en la que se guardarán los objetos que formen el “sistema solar de figuras” (Figura 5). El diseño de esta práctica es muy sencillo (Figura 6), y en él se utiliza la unidad figuras creada en la práctica anterior.

Nótese que en esta práctica aparece de nuevo el polimorfismo, ya que la lista permite almacenar todo tipo de figuras.

4. Opinión de los alumnos

Es destacable la buena acogida que las prácticas aquí presentadas han venido teniendo estos años por parte de los alumnos. Una anécdota: el primer año de puesta en marcha de las mismas, alumnos del curso anterior, con la asignatura aprobada, mostraron su interés por las nuevas prácticas. Otra curiosidad: recientemente, se preguntó sobre las prácticas a alumnos que las habían realizado hace dos cursos, resultando grato el comprobar que todavía las recordaban y las guardaban.

Algunas de las opiniones expresadas por los alumnos sobre las prácticas han sido las siguientes:

- Muy interesantes y entretenidas.
- Llamativas: las mostraban a amigos y familiares.

- Pedagógicas y útiles. Les ayudaron a entender el diseño orientado a objetos, las ventajas de este tipo de programación, la diferencia entre clase y objeto, la herencia y otros conceptos que les han resultado muy útiles en asignaturas posteriores.
- Divertidas, ya que se podían ampliar fácilmente para, por ejemplo, construir nuevas figuras.
- Fomentadoras de la imaginación.

Por otro lado, señalar que resultó ligeramente más fácil entender las prácticas a los alumnos que habían visto dibujo técnico en bachillerato, aunque el resto de alumnos indicó que no tuvo especiales problemas para su realización una vez explicado, fundamentalmente, el concepto de proyección.

5. Ampliaciones de los alumnos

Es significativo destacar, como muestra de la motivación que estas prácticas produjeron entre los alumnos, algunas de las ampliaciones que éstos realizaron de forma voluntaria:

- Inclusión de otras figuras geométricas, algunas de ellas compuestas de un gran número de aristas, como las mostradas en la Figura 7.
- Adición de colores a las figuras.
- Representación en pantalla utilizando perspectiva isométrica, en vez de usar las ecuaciones dadas para perspectiva caballera.
- Interactividad mediante el teclado. Permitían al usuario definir el eje de giro, desplazar figuras por la pantalla usando los cursores,

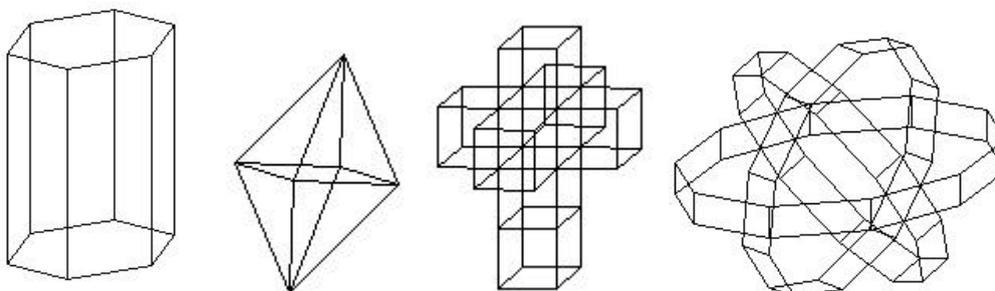


Figura 7. Ejemplos de figuras realizadas por los alumnos

- seleccionar las posiciones iniciales y figuras a mostrar en la práctica del sistema solar, etc.
- Adición de un procedimiento (*wait retrace*) para evitar el parpadeo en la visualización de gráficos debido al refresco de la pantalla.
- Giro simultáneo de la figura sobre varios ejes.

6. Conclusión

La utilización de prácticas con gráficos 3D animados han resultado satisfactorias en la enseñanza de la programación orientada a objetos, ya que han permitido introducir de una forma natural los conceptos más importantes de este paradigma y a su vez han resultado estimulantes para los alumnos.

Es importante resaltar que es un factor crítico de éxito el que las prácticas sean animadas, ya que un gráfico estático presentado en la pantalla resulta mucho menos impresionante. También se debe destacar la importancia de utilizar UML o algún otro lenguaje gráfico para modelar el diseño, de forma que éste sea fácilmente entendible por los alumnos.

7. Futuras líneas de trabajo

Tras el éxito obtenido por estas prácticas entre los alumnos, se ha empezado a trabajar en una nueva práctica continuadora del espíritu aquí presentado basada en la visión estereoscópica, empleando para ello estereogramas de puntos al azar [3] en la pantalla. Para experimentar la sensación de tridimensionalidad sólo es necesario el uso de

gafas con filtros rojo y verde, de fácil construcción, por otra parte, por los propios alumnos.

Las pruebas iniciales realizadas hasta ahora están resultando alentadoras. Conviene señalar que en su diseño se está utilizando una de las clases definidas en las prácticas de figuras, lo que permitirá, si en un futuro se usaran todas ellas conjuntamente, proporcionar a los alumnos un claro ejemplo de reutilización.

Referencias

- [1] ACM/IEEE-CS. *Computer Curricula 2001*. Final report, 2001.
- [2] Gisela Bielig-Schulz y Christoph Schulz. *3D graphics in Pascal*. Wiley, 1990.
- [3] John P. Frisby. *Del ojo a la visión*. Alianza, 1987.
- [4] Bertrand Meyer. *Construcción de software orientado a objetos*. Prentice Hall, 1999.
- [5] Práctica introductoria: <http://www.cc.uah.es/jmacias/jenui2002/Introduccion/>
- [6] Primera práctica: <http://www.cc.uah.es/jmacias/jenui2002/Primera/>
- [7] Maria Salomó y otros. *Iniciativas para motivar a los alumnos de programación*. En las actas de las VII Jornadas de Enseñanza Universitaria de la Informática. José Miró (editor). Universitat de les Illes Balears, 2001.
- [8] Segunda práctica: <http://www.cc.uah.es/jmacias/jenui2002/Segunda/>
- [9] Unified Modeling Language: <http://www.omg.org/uml/>