

## Sistema Gráfico en Turbo Pascal

Juan L. Almará

Facultad de Ingeniería, UNER, Computación I  
Ingeniería Electrónica, UTN Reg. Pná, Informática II  
San Lorenzo 41, Dpto 2, 3100 Paraná, Entre Ríos, Argentina.

### RESUMEN

En el presente trabajo se detalla la implementación de un conjunto de rutinas, escritas en Turbo Pascal, para la realización de gráficos en alta resolución.

El objetivo es facilitar el acceso y uso de salidas gráficas en computadoras personales, equipadas con dispositivos HGA, CGA, MCGA, EGA, VGA, trazadores e impresoras.

Se hallan considerados: interfaz con los dispositivos gráficos, gráficos bidimensionales, con transformaciones, poligonales y funciones analíticas, y dibujos tridimensionales, con transformaciones, proyecciones, poliedros, líneas y superficies ocultas y funciones.

### ABSTRACT

In the present paper is detailed the implementation of a set of Turbo Pascal routines, for high resolution graphics.

The goal is to provide facilities for the access and use of graphic outputs in personal computers, equipped with devices HGA, CGA, MCGA, EGA, VGA, plotters and printers.

The topics included are: interface with graphic devices, bidimensional graphics, with transformations, polygons and analytic functions, and three dimensional drawings, with transformations, projections, polyhedra, hidden lines and surfaces and functions.

### 1. INTRODUCCION

El presente trabajo detalla las posibilidades de un conjunto de rutinas para la realización de gráficos en alta resolución, en computadoras personales compatibles.

El objetivo de la implementación es aportar un conjunto de subprogramas destinados a facilitar el acceso y uso de salidas gráficas, en dispositivos de amplia difusión, como HGA, CGA, MCGA, EGA y VGA.

Las rutinas están escritas en Turbo Pascal, y se han incorporado los elementos básicos para gráficos que aporta el proveedor [1].

Las constantes, tipos de variables, variables, procedimientos y funciones han sido agrupadas en diversas unidades de código (Unit), facilidad que provee la versión del lenguaje. Posteriormente, con sólo nombrar las unidades que contienen los elementos necesarios, el

compilador encadena los items utilizados y crea un programa ejecutable. Las rutinas se han compilado con las directivas de llamadas largas (far calls) y se ha habilitado la posibilidad de recubrirlas (overlays), además de la detección y uso de co-procesador matemático. Se ha codificado una Unit para el manejo de hardware, y otra para la realización de dibujos bidimensionales. El paquete se completa con rutinas para la presentación de curvas, y de escenas tridimensionales.

## 2. RECURSOS BÁSICOS

Los componentes relativos al manejo del equipo gráfico fueron agrupados en una Unit, para:

- Proveer cierta independencia del hardware.
- Facilitar la incorporación de nuevas tecnologías.
- Ampliar las posibilidades de uso, ya que los elementos básicos para el manejo de gráficos pueden, de esta forma, ser utilizados desde diversos programas.
- Proveer un sencillo sistema de escalas.
- Incluir las facilidades de impresión, salvado y recuperación de imágenes.

En esta Unit, la descripción de los recursos básicos podría indicarse como:

### a. Modos gráficos.

Turbo Pascal incluye rutinas que detectan el tipo de hardware existente, y, con su ayuda, se ha construido un módulo que inicializa el modo gráfico a la mayor resolución existente. Como un subproducto de la inicialización, los datos acerca del equipo quedan memorizados en diversas variables, que se hacen globales a todos los programas que utilicen esta Unit.

La inicialización a la mayor resolución no implica que el usuario no pueda diseñar su propia rutina de modo gráfico, ya que los procedimientos se han codificado de forma tal que no interfieran con ampliaciones particulares.

La orden para volver al modo texto se incluye también aquí, además de un procedimiento que, borrando el buffer de teclado, espera a que se digite una tecla, y recuerda cual ha sido.

Los manejadores de dispositivos (drivers) necesarios se encuentran formando parte del código de esta Unit, por lo que no son necesarios otros archivos en el disco.

### b. Texto en modo gráfico.

La definición de formatos de caracteres (fonts) a ser presentados en alta resolución también se ha encadenado con el presente código, con lo que se posibilita la escritura con tipografía normal, sans serif, pequeña y gótica, en diversos tamaños y orientaciones.

### c. Escalas.

Al entrar al modo gráfico, automáticamente se establecen las escalas que hacen corresponder un pixel a una unidad del mundo físico. Mediante un procedimiento se puede "redimensionar" el rectángulo que representa al mundo físico [2] [3]. Otra orden permite definir ventanas en la pantalla, que pueden ser coincidentes.

La conversión de coordenadas del mundo real al de la pantalla, y viceversa, se logra mediante cuatro funciones apropiadas.

### d. Impresión de la pantalla gráfica.

Se utiliza la tecla de Imprimir Pantalla que, estando en modo de alta resolución, llama a una rutina codificada en esta Unit, la que

accede a la impresora [4], similar a otras en el mercado. La copia en papel también se puede solicitar mediante software, en el transcurso de un programa.

La calidad de la salida impresa depende de la impresora y de la cinta; sin embargo, puede servir como una primera aproximación. En otra Unit (3.f) se incluyen las rutinas para lograr salida en lenguaje HPGL, compatible con varios plotters.

#### e. Almacenamiento y recuperación de la pantalla gráfica.

Especialmente en la generación de imágenes tridimensionales, el tiempo de proceso puede ser elevado, según las expectativas del usuario. En otras circunstancias, se desearía obtener un archivo de imágenes generadas, con fines de comparación, por ejemplo. Se proveen dos rutinas: una de ellas captura la imagen de la pantalla, la comprime, y la guarda en un archivo; la otra, lee los datos de un archivo, descomprime la imagen, y la presenta en pantalla. Cada vez que se solicita guardar una pantalla, la rutina crea un archivo con extensión numérica creciente, lo que permite obtener, manual o automáticamente, una serie ordenada de vistas. Mediante un sencillo programa, se pueden cargar y visualizar secuencialmente.

Estas rutinas de interfaz con el hardware fueron comprobadas en equipos PS/2 Mod 30 y 60 (con co-procesador), XT y AT 286 y 386, con monitores monocromáticos y color, y adaptadores MCGA, CGA, HGA, EGA y VGA.

Con el sólo uso de los elementos de esta Unit se pueden construir rápida y fácilmente gráficos sencillos, que incluyen la transformación de escala, y las órdenes provistas en la Unit Graph (líneas, rectángulos, texto en modo gráfico, y otras [1]).

### 3. GRÁFICOS BIDIMENSIONALES

En ciertas aplicaciones, no es suficiente contar con la generación elemental de imágenes que permite la Unit de Recursos Básicos. Obtener gráficas de poligonales, polígonos y sectores llevaría a un mayor esfuerzo de programación. Por otra parte, es frecuente la necesidad de contar con rutinas para:

- Transformaciones en el plano.
- Recorte de segmentos.
- Sombreado y relleno de polígonos.
- Salida compatible con graficadores.
- Presentación interactiva.

Los elementos adecuados para lograr éstas y otras facilidades se encuentran en una Unit de gráficos bidimensionales [3] [5], cuyo contenido es:

#### a. Poligonales y polígonos.

La estructura de los datos para almacenar las informaciones relativas a líneas poligonales y polígonos consiste en una matriz de dos columnas y una cierta cantidad de filas. Se memoriza, en cada renglón, la posición (x,y) de los puntos constitutivos del esquema, estando la topología determinada por el orden secuencial. Los algoritmos que presentan la figura geométrica reconocen la presencia de un polígono, y conectan el último punto con el primero.

#### b. Arcos de circunferencia.

Los puntos correspondientes a circunferencias, sus arcos y sectores, y a elipses, se calculan mediante sus ecuaciones paramétricas [2] [6], y se almacenan como poligonales.

### c. Transformaciones.

Se hallan implementadas las transformaciones planas de traslación, rotación, escalado y homotecia [5] [6], que actúan sobre puntos individuales. La transformación de un conjunto de puntos se obtiene por repetición, manteniendo la topología.

### d. Recorte en el plano.

Los segmentos y puntos son analizados para determinar si se encuentran dentro del plano físico en uso, o fuera del mismo, o, para segmentos, parcialmente dentro [3] [5] [6]. En todos los casos, se determina su condición, y, de ser necesario, se obtienen los extremos del segmento perteneciente al plano físico, mediante un subprograma de [6].

### e. Sombreado y relleno de polígonos.

Los polígonos pueden ser sombreados mediante líneas con diversas separaciones entre sí, y distintos ángulos con respecto a la horizontal [5]. Se utilizan las coordenadas del mundo físico. Para realizar el relleno o pintado, se transforman los vértices a coordenadas de la pantalla, y se analizan secuencialmente las líneas horizontales, determinando los segmentos interiores, y trazando líneas. Este algoritmo puede manejar polígonos no convexos.

### f. Salida para plotter.

En las rutinas de presentación de puntos y segmentos en la pantalla se han incorporado las acciones necesarias para generar caracteres en un archivo de texto, compatibles con el lenguaje de graficadores HPGL. Existe una etapa previa de inicialización de la salida (nombre de archivo, escalas), y un epílogo del proceso (cierre del archivo). El código generado comprende unas muy pocas órdenes del repertorio mencionado, por lo que, de ser necesario, se puede traducir fácilmente a otros entornos. Sin embargo, esta salida puede no dar resultados completamente satisfactorios en el caso de relleno de figuras (3.e.).

### g. Presentación interactiva.

Mediante el llamado a un procedimiento, uno de cuyos parámetros es otro procedimiento que contiene las órdenes para generar un dibujo, es posible controlar la visualización de diversas maneras: el rectángulo del mundo físico puede moverse a la izquierda, derecha, arriba o abajo, y cambiar de dimensiones; se puede avanzar un valor fijo, o páginas de pantalla, y la historia de los desplazamientos se va memorizando, con el objeto de conocer cómo se obtuvo cierta presentación.

## 4. FIGURAS Y CURVAS

Las rutinas para imágenes bidimensionales fueron ampliadas en varios sentidos, con el objeto de administrar procesos frecuentes, presentar mayor información, y preservar el trabajo realizado, según:

- Presentación de gráficas matemáticas y estadísticas.
- Visualización de texto en modo gráfico.
- Inclusión de ejes, etiquetas y títulos.
- Generación y suavizado de datos.
- Manejo de tablas de valores.

Los subprogramas pertinentes se encuentran agrupados en una Unit para el manejo de curvas, cuyos puntos salientes son:

### a. Gráficas matemáticas.

Mediante facilidades provistas por Turbo Pascal (funciones y procedimientos como parámetros), se han diseñado rutinas que reciben el

nombre de una función explícita, y presentan su diagrama como una poligonal. Diversos subprogramas aceptan funciones en forma cartesiana, paramétrica y polar, y es posible presentar gráficos en coordenadas lineales, semilogarítmicos y logarítmicos, como se vé en la Figura 1.

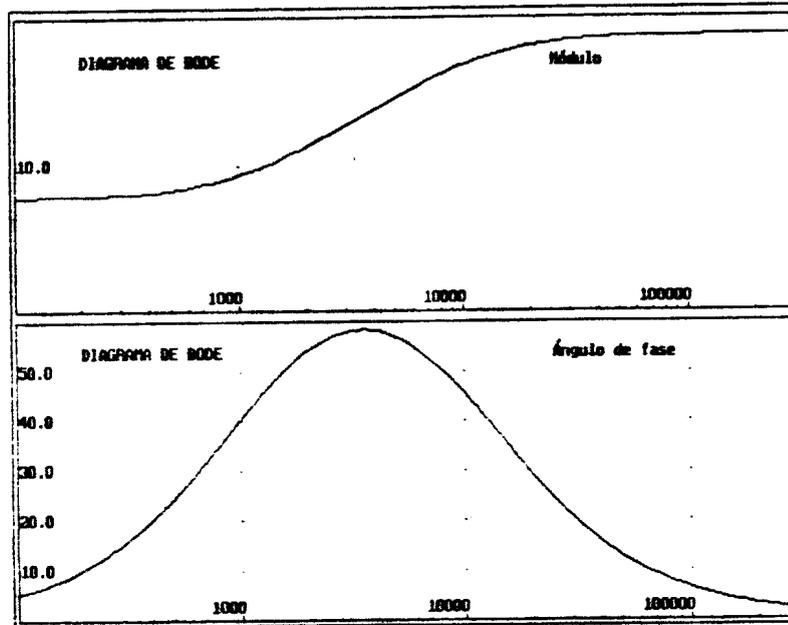


Figura 1: ejemplo de función en el plano.

**b. Gráficas estadísticas.**

La presentación de histogramas y barras horizontales se logra mediante el paso de los valores apropiados a un subprograma. En ambos casos, es posible controlar el espesor de las barras, y marcar puntos de interés mediante pequeños cuadrados, círculos, triángulos o cruces.

**c. Grillas.**

Con el objeto de hacer más explicativos los diagramas, se han previsto órdenes para el dibujo de los ejes coordenados con sus respectivas etiquetas y demarcaciones, grilla de elementos equidistantes y marcos alrededor del rectángulo de la representación. Estas facilidades están disponibles tanto para coordenadas cartesianas como para polares.

**d. Texto en modo gráfico.**

El uso de rutinas propias de la versión del lenguaje [1], y mediante la adecuada transformación de escalas, es posible presentar títulos y leyendas, con diversos tipos de letras, tamaños y orientaciones.

**e. Suvizado de curvas.**

Se ha previsto la graficación de datos experimentales; los mismos pueden, opcionalmente, ser preprocesados mediante un algoritmo de

aproximación de mínimos cuadrados, global o por conjuntos de un número seleccionable de puntos, siendo también una opción el grado.

#### f. Generación de curvas.

Los algoritmos de Bezier y Spline [5] [6] [7] fueron adaptados a la estructura de datos presente, e integrados en esta Unit.

#### g. Administración de tablas.

Las tablas de valores experimentales, o las de descripción de imágenes, pueden ser guardadas y recuperadas en y desde el disco, mediante un par de procedimientos que tienen en cuenta varias líneas de documentación en el archivo de texto de los datos.

### 5. GRÁFICOS TRIDIMENSIONALES

Muchas aplicaciones de gráficos en computadoras requieren o mejoran con la presentación de escenas y objetos tridimensionales. Con el objetivo de generar gráficos relativamente realistas, se han implementado las rutinas para:

- La reducción de la representación de objetos de tres dimensiones a equivalentes bidimensionales.
- El dibujo de estos objetos con cierto grado de realismo.

El logro de estas metas implica la utilización de las Units anteriores (básica, bidimensional y curvas), así como un conjunto de nuevos algoritmos.

#### a. Proyecciones.

La sensación de profundidad, al pasar de tres dimensiones a dos, puede lograrse en mayor o menor grado según el tipo de proyección que se realice. En la presente implementación se han incorporado diversos tipos de proyecciones paralelas y centrales: frontal, aérea, ortogonal, isométrica, dimétrica, trimétrica, caballera y perspectiva [3] [5] [6] [8] [9].

#### b. Transformaciones.

Una de las transformaciones más importantes es la de llevar coordenadas del espacio tridimensional a coordenadas del plano de proyección, para luego pasarlas al plano de la pantalla. Sin embargo, varios algoritmos se benefician con la consideración de un sistema de referencia en el volumen de visión, por lo que también se ha incluido esta posibilidad.

Otras transformaciones de interés, operando en tres dimensiones, comprenden traslaciones, rotaciones alrededor de una recta arbitraria, escalados y homotecias. Todas ellas se hallan implementadas en el presente desarrollo, mediante coeficientes matriciales [2] [5].

#### c. Recorte en el espacio.

No siempre es posible o deseable representar todo el espacio que corresponde a una escena, por lo que está presente una rutina de recorte en el espacio, similar a la usada para dibujo bidimensionales (3.d.) [5] [6].

#### d. Representación de escenas.

Las escenas en el espacio (que pueden contener varios objetos) pueden representarse según tres modalidades, disponibles en este paquete:

- Gráfico de líneas (wire frame): donde todos los segmentos especificados son dibujados. Aún para conjuntos relativamente sencillos, resulta en una imagen confusa.

- Líneas ocultas (hidden lines): donde no se dibujan los segmentos que están detrás de caras planas convexas. Se programó el algoritmo de Warnock [6] [8] [9].
- Superficies ocultas (hidden surfaces): se presentan sólo las caras (poligonales) que son visibles desde la posición del observador. Se ha implementado el algoritmo de líneas de barrido [3] [6], que permite caras que se penetran.

A lo largo de los últimos 30 años fueron desarrollados varios algoritmos y sus derivados, para la resolución de los problemas de líneas y superficies ocultas. Los mismos difieren en sus limitaciones (convexidad, penetración), complejidad, necesidades de memoria y velocidad [3]. Los incluidos en el presente conjunto de rutinas son sólo algunas de las posibilidades, no necesariamente los más apropiados para una aplicación en particular.

Un ejemplo de los resultados obtenibles se ve en la Figura 2.

Phi = 30  
theta = 70

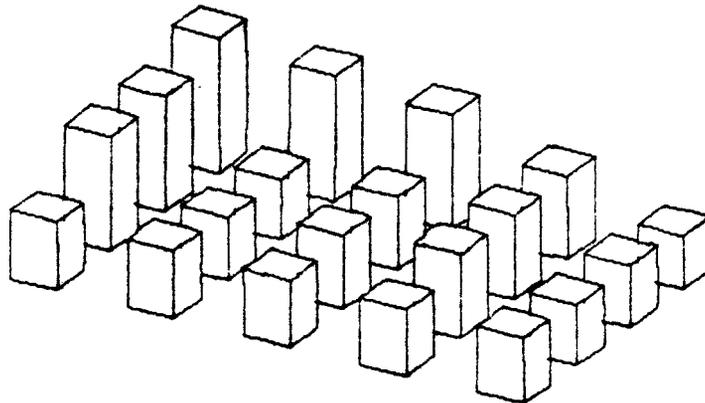


Figura 2: ejemplo de líneas ocultas.

#### e. Visualización interactiva.

La presentación estática de la proyección de una escena tridimensional, en ciertas oportunidades, no es suficiente. Por tal motivo, se ha incluido un procedimiento que, teniendo como parámetro otro procedimiento que construye la escena, permite obtener diversas vistas modificando los ángulos que definen la posición del observador, el tipo de proyección, y la modalidad de líneas ocultas. Sin embargo, no se logra un efecto de animación, debido a la falta de velocidad; este inconveniente se remedia, en parte, mediante las facilidades de captura de imágenes (l.e.).

#### f. Graficación de funciones.

Funciones explícitas de dos variables, de la forma  $z=f(x,y)$  pueden ser graficadas, indicando un rectángulo base en el plano  $x-y$ , además

de la cantidad de planos paralelos a z-y a utilizar. La gráfica puede obtenerse como una cuadrícula con todos los segmentos visibles, o mediante una representación con líneas ocultas, siendo la Figura 3 un ejemplo.

$$z = \sin(x)/x + \sin(y)/y$$

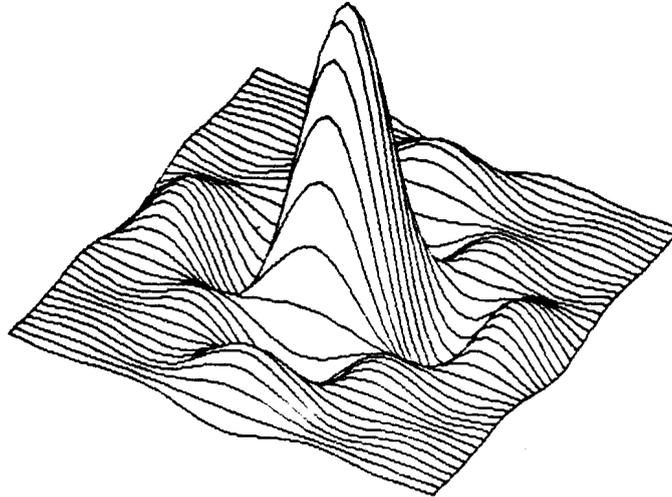


Figura 3: ejemplo de función en tres dimensiones.

## 6. CONCLUSIONES

Del detalle precedente puede comprobarse que este conjunto de rutinas permite, con múltiples alternativas, el uso de las posibilidades gráficas de las computadoras personales. Está implementado en un eficiente lenguaje estructurado de alto nivel, por lo que cumple con el objetivo original del desarrollo.

Los principales elementos que se han incorporado son:

- Compatibilidad y cierta independencia del hardware.
- Transformaciones de diversa índole.
- Presentación de varias geometrías elementales, con las que pueden construirse muchas figuras complejas.
- Gráfica de funciones matemáticas y estadísticas bi- y tridimensionales.
- Visualización interactiva de objetos tridimensionales con líneas y superficies ocultas.

El desarrollo se ha orientado hacia aplicaciones científicas, técnicas, comerciales y docentes.

Algunas de las limitaciones de este conjunto de rutinas gráficas en Turbo Pascal son:

- Archivo de imagen: la estructura de los datos que manejan estos subprogramas no incluyen un "display file" [5] [6]; si bien es un

inconveniente, fue diseñado de este modo para expandir la cantidad posible de usos.

- Escenas tridimensionales realistas [3] [6] [8]: no forman parte de este paquete las rutinas que implementan modelos de iluminación, transparencias, reflexiones, texturas, sombras ni visión estereoscópica.
- Construcción interactiva: sólo se ha implementado la visualización interactiva de gráficos, no así su generación.
- Velocidad de presentación: principalmente, en las rutinas para tres dimensiones y partes ocultas, puede comprobarse que las imágenes no se generan con una rapidez muy elevada.

Esta implementación de rutinas gráficas no pretende competir con programas comerciales ampliamente difundidos y de probada calidad. Se trata de ofrecer un paquete gráfico que pueda ser integrado con otros programas, con los que se logre una solución económica y fiable para aplicaciones en que se desea generar un software de calidad, independiente y autocontenido.

## 7. REFERENCIAS

El siguiente conjunto de referencias menciona algunas de las fuentes donde pueden encontrarse ampliaciones sobre lo comentado en este trabajo, junto a pseudocódigo y programas que fueron guía en el desarrollo.

1. Turbo Pascal Reference Guide. Version 5.0. Borland International. Inc.
2. Dony, R.: "Grafismo científico con ordenador". Ed. Masson S.A., 1986.
3. Hearn, D. y Baker, P. M.: "Gráficas por computadora" .Ed. Prentice-Hall Hispanoamericana S.A., 1988.
4. EPSON LX-810 User's Manual. Epson America Inc., California.
5. Harrington, S.: "Computer graphics - A programming approach". Ed. McGraw-Hill, 1983.
6. Newman, W. y Sproull, R.: "Principles of interactive computer graphics". 2 edic. Ed. McGraw-Hill, 1979.
7. Formica, D. L. y Delrieux, C.: "Descripción y evaluación de técnicas paramétricas de representación de curvas". Revista Telegráfica-Electrónica, Abril 1989, págs. 338-346.
8. Tori, R., Arakaki, R, Massola, A. M. y Filgueiras, L. V.: "Fundamentos de computação gráfica". Livros Técnicos e Científicos Editora. 1987.
9. Magalhaes, L. P.: "Computação gráfica". Editora da UNICAMP, Campinas, 1986.

