CÓMO ARCHIVAR IMÁGENES CREADAS POR EL USUARIO EN LENGUAJE C++

Bruno López Takeyas
Instituto Tecnológico de Nuevo Laredo
Reforma Sur 2007, C.P. 88250, Nuevo Laredo, Tamps. México
http://www.itnuevolaredo.edu.mx/takeyas
E-mail: takeyas@itnuevolaredo.edu.mx

Resumen: *El siguiente artículo muestra el código fuente para graficar imágenes en lenguaje C++, grabarlas en un archivo y posteriormente cargarlas y reposicionarlas en la pantalla. El artículo está orientado a estudiantes con conceptos de graficación y archivos que desean almacenar figuras previamente dibujadas.*

Palabras claves: *Graficación, resolución, píxel, lenguaje C++, monitor, archivo.*

1 INICIALIZAR EL MONITOR EN MODO GRÁFICO

Para dibujar figuras en la pantalla es necesario habilitar el monitor en modo gráfico y se requiere incluir el encabezado `#include <graphics.h>` que contiene las declaraciones y funciones relacionadas con graficación e inicializar el monitor en modo gráfico y utilizar sus píxeles con la función ***initgraph()***. Dicha función requiere las declaraciones mostradas en la Fig. 1.

```
int monitor=DETECT;
// Variable para detectar el tipo
// de monitor
int modo;
// Modo de operación del monitor
```

Fig. 1. Inicialización del monitor en modo gráfico

también se puede declarar e inicializar con un tipo de monitor específico como el ejemplo mostrado en la Fig. 2.

```
int monitor=VGA;
// Variable para usar el monitor
// tipo VGA
int modo=VGAHI;
// Usar el monitor VGA a su
//maxima resolución
```

Fig. 2. Inicialización del monitor VGA en modo gráfico.

Para terminar de usar el monitor en modo gráfico y devolverlo a su modo de texto se usa la función ***closegraph().***

1.1 Uso de coordenadas

Una vez que se inicializa el monitor en modo gráfico, las coordenadas tienen al píxel como unidad de medida. La función ***getmaxx()*** calcula la cantidad de píxeles por renglón y la función ***getmaxy()*** calcula la cantidad de renglones de la pantalla.

Las funciones de gráficos tienen como estándar el orden de manejo de coordenadas como columna, renglón; es decir, primero se anota la columna y después el renglón para posicionarse en dicha coordenada. Esta notación también se utilizará para dibujar los escudos.

2 FUNCIONES DE MANEJO DE IMÁGENES

En esta sección se presentan las funciones necesarias para el manejo de imágenes con la finalidad de grabarlas en un archivo.

2.1 Declaración de variables necesarias

Cuando se dibuja una imagen en la pantalla, ésta puede ser almacenada en una variable de memoria para manipularse completamente. Esto se logra mediante un apuntador de tipo `char far`, el cual puede almacenar en memoria una imagen de hasta 1MB. Esto se logra con las declaraciones de variables mostradas en la Fig. 3.

Declaración	Uso
long tamano;	Variable de tipo entero largo para calcular el tamaño en bytes de la imagen que se desea archivar.
char far *imagen;	Apuntador para capturar en memoria la imagen (puede ser direccionada hasta 1 MB)

Fig. 3. Declaración de variables necesarias para cargar una imagen en memoria.

2.2 ¿Cómo cargar una imagen en memoria?

Para almacenar una imagen previamente dibujada en la pantalla, debe calcularse su tamaño en bytes, reservar espacio de memoria para posteriormente "cargarla" en una variable de memoria mediante las declaraciones de la Fig. 3. Esto se logra con las funciones que se presentan a continuación.

2.2.1 La función imagesize()

Se utiliza esta función para calcular el espacio (en bytes) ocupado por una imagen. Requiere de 4 argumentos que representan las coordenadas de las esquinas superior-izquierda e inferior-derecha respectivamente y devuelve un valor que representa su tamaño, el cual debe almacenarse en una variable de tipo entero largo (Fig. 4).

2.2.2 La función farmalloc()

Esta función reserva espacio en memoria para almacenar la imagen cuyo bloque posiblemente exceda 64k (según el tamaño calculado por la función `imagesize`) y requiere incluir el encabezado `#include <alloc.h>` (Fig. 4).

2.2.3 La función getimage()

Se usa esta función para almacenar en un apuntador de tipo `char far` (Fig. 3) la imagen comprendida entre los puntos indicados por las coordenadas de sus esquinas superior-izquierda e inferior-derecha (Fig. 4).

Función	Uso
tamano=(long int) imagesize(x1,y1,x2,y2)	Calcula el espacio necesario para capturar en memoria la imagen comprendida entre las esquinas (x1,y1) a (x2,y2)
imagen=(char far *) farmalloc(tamano)	Reserva el espacio de memoria indicado por ***tamano***
getimage(x1,y1,x2,y2,imagen)	Captura en el apuntador ***imagen*** el dibujo comprendido entre (x1,y1) y (x2,y2)

Fig. 4 .- Funciones necesarias para archivar una imagen

3 ¿CÓMO ARCHIVAR UNA IMAGEN CREADA POR EL USUARIO?

Igual que se puede almacenar cualquier variable con un tipo de datos en particular como enteros, reales, caracteres, cadenas, registros, o cualquier estructura de datos, también es posible almacenar en un archivo una imagen que se despliega en la pantalla. Para ello es necesario capturar en memoria la imagen que se desea grabar usando un apuntador, apoyándose en las declaraciones de la Fig. 3 y las funciones mostradas en la Fig. 4. Para manipular el archivo se declara el alias correspondiente como se muestra en la Fig. 5. Una vez capturada en memoria la imagen, se abre un archivo binario con un nombre específico, para proceder a escribir el apuntador con la imagen capturada (Fig. 6).

Declaración	Uso
FILE *alias	Declaración del alias para el archivo.

Fig. 5 .- Declaración del alias del archivo

Función de manejo de archivos	Uso
alias=fopen("IMAGEN.IMG","wb")	Crea el archivo binario "IMAGEN.IMG"
fwrite(imagen,sizeof(imagen),1,alias)	Graba la imagen capturada en el archivo
fclose(alias)	Cierra el archivo

Fig. 6 .- Declaraciones necesarias para archivar una imagen

3.1 Ejemplo de aplicación

El programa de la Fig. 7 muestra el código completo de un ejemplo que dibuja en la pantalla el escudo del Instituto Tecnológico de Nuevo Laredo (incluido en el encabezado TEC.H) para posteriormente grabarlo en un archivo.

```
/*
Programa para graficar el escudo del
Tec, grabarlo en un archivo para
cargarlo y desplegarlo posteriormente

 MiniTaller: Tecnicas avanzadas de
programacion en Lenguaje C++
 Instructor: M.C. Bruno Lopez Takeyas
*/

#include <graphics.h>  /* Encabezado
con declaraciones de graficos*/
#include <conio.h>
#include <stdio.h>
#include <iostream.h>
#include <alloc.h> /* Para usar la
funcion farmalloc*/
#include <string.h>
#include <tec.h> /* Encabezado
desarrollado por el programador con el
codigo fuente de los escudos del Tec y
de ISC*/

void main(void)
{
  int monitor=DETECT, modo; /*
Declaracion de tipo de monitor y modo
Automaticamente detecta el tipo de
monitor*/
  char *archivo; /* Para capturar el
nombre del archivo*/
  long tamano; /* Variable para
calcular el tamaño en bytes de la
imagen que se desea archivar*/
  char far *imagen; /* Variable para
capturar en memoria la imagen que se
desea archivar*/
  FILE *alias_archivo;
  int columna,renglon;



initgraph(&monitor,&modo,"\\tc\\bgi");
/* Inicializa el modo grafico indicando
el monitor y modo utilizado*/
/* El subdirectorio \\tc\\bgi indica la
ruta de localizacion de los
archivos *.BGI (monitores) y *.CHR
(tipos de letras)*/


  TEC(320,200); /*Escudo del Tec en
columna=320 y renglon=200*/

  setlinestyle(DOTTED_LINE,1,1);
  rectangle(250,130,390,270);

  gotoxy(1,19); cout << "Anote el
nombre del archivo (incluyendo la
extension): ";
  gets(archivo);
  /* Capturar el nombre del archivo*/

  tamano = (long
int)imagesize(250,130,390,270);
/* Calcula el tamaño de la imagen que
se desea archivar*/

  imagen = (char far *)
farmalloc(tamano);
/*Reserva la memoria suficiente*/

  getimage(250,130,390,270,imagen);
/* Cargar en memoria la imagen que
contiene el cuadro de las coordenadas
indicadas*/
  alias_archivo=fopen(archivo,"wb");
/* Crear el archivo con el nombre
capturado*/

fwrite(imagen,1,tamano,alias_archivo);
```

```
/* Graba en el archivo la imagen
cargada en memoria*/
  fcloseall(); /* Cierra el archivo*/

  cout << "\n\ntamaño=" << tamano << "
bytes";
  cout << "\nOprima cualquier tecla
para limpiar la pantalla y cargar la
imagen";
  getch();

  clearviewport();
/* Limpia la pantalla en modo grafico*/
  alias_archivo=fopen(archivo,"rb");
/* Abre el archivo en modo de solo
lectura*/
  fread(imagen,1,tamano,alias_archivo);
/* Lee desde el archivo la imagen*/
  fcloseall(); /* Cierra el archivo*/

  cout << "\nAnote las coordenadas
donde desea desplegar la imagen ...";
  cout << "\n\nColumna="; cin >>
columna;
  cout << "\n\nRenglon="; cin >>
renglon;

  putimage(columna,renglon,imagen,1);
/* Coloca en pantalla la imagen cargada
del archivo en las coordenadas
especificadas*/
  getch();
  closegraph(); /* Termina el modo
grafico (vuelve a su modo normal)*/
  return;
}
```

Fig. 7.- Programa para archivar una imagen.

4 CONCLUSIONES

En este artículo se muestran las operaciones fundamentales de manejo de imágenes en memoria y se presenta el código fuente para graficar el escudo del ITNL y grabarlo en un archivo en lenguaje C++ para posteriormente cargarlo y reposicionarlo en la pantalla. Dicho código puede obtenerse en el sitio http://www.itnuevolaredo.edu.mx/takeyas o bien solicitarse al autor escribiendo un correo electrónico a takeyas@itnuevolaredo.edu.mx.

5 BIBLIOGRAFÍA

• Barkakati Nabajyoti. **"The Waite Group´s. Turbo C Bible"**. Howard W. Sams & Company. Estados Unidos. 1990.

• Deitel y Deitel. **"C++ Cómo programar"**. Segunda edición. Pearson-Prentice Hall. Estados Unidos. 1999.

• Lafore, Robert. **"The Waite Group´s. Turbo C. Programming for the PC"**. Revised Edition. Howard W. Sams & Company. Estados Unidos. 1990.

• López Takeyas, Bruno. **"Minitaller: Técnicas avanzadas de programación en lenguaje C++".** Instituto Tecnológico de Nuevo Laredo, Tam. México. 2003.

• Schildt, Herbert. **"Turbo C. Programación avanzada"**. Segunda edición, McGraw Hill. Estados Unidos. 1990.