Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра автоматизованих систем управління

Курсова  робота

з дисципліни  «Проблемно-орієнтовані мови програмування»

на тему

ВІДТВОРЕННЯ ГРАФІЧНИХ ЗОБРАЖЕНЬ З ФАЙЛІВ

ВМР-ФОРМАТУ

Завдання на курсову роботу

з дисципліни «Проблемно-орієнтовані мови програмування»

http://stud24.ru/programming-computer/vdtvorennya-grafchnih-zobrazhen-z-fajlv/347511-1055074-page1.html

Спеціальна частина завдання:

1.   Ознайомитись з варіантами структур ВМР-формату.

2.   Ознайомитись з поняттям палітри і керування палітрами.

3.   Ознайомитись з алгоритмами стиснення, що застосовуються для ВМР-файлів.

4.   Реалізувати відтворення зображення (невеликого розміру) в режимі 12h.

5.   Термін завершення роботи – 6 червня 2010 року.

Зміст

Вступ…………………………………………………………………………….4

1.      Огляд літератури

1.1.           Загальна характеристика ВМР-формату………………………….....5

1.2.           Варіанти структур ВМР-формату……………………………………6

1.2.1.     BITMAPFILEHEADER……………………………………………7

1.2.2.     BITMAPINFOHEADER, BITMAPV4HEADER, BITMAPV5HEADER………………………………………………..8

1.2.3.     Збереження даних у форматі BMP. Палітра……………………14

1.3.           Типи стиснення даних у зображеннях BMP формату…………….16

1.3.1.     Метод стиснення RLE8…………………………………………..17

1.3.2.     Метод стиснення RLE4…………………………………………..18

1.4.           Графіка в мові програмування С……………………………………19

2.      Формулювання задачі………………………………………………………20

3.      Опис алгоритму……………………………………………………………..21

3.1.           Загальний алгоритм розв’язку………………………………………21

3.2.           Блок-схема загального алгоритму програми ………………………23

4.      Програмна реалізація алгоритму ………………………………………….25

5.      Інструкція користувачеві програми ………………………………………27

6.      Контрольні приклади та аналіз результатів їх реалізації…..…………….28

Висновок………………………………………………………………………..30

Список літератури   .…………………………………………………………...31

Додатки…………………………………………………………………………32

Додаток 1. Код програми …………………………………………………32

Додаток 2. Зображення rainbow.bmp …………………………………….40

Вступ

Мета курсової роботи – практичне закріплення, поглиблення та розширення знань та навичок, набутих у процесі вивчення мови програмування С. З цією метою було поставлено конкретну задачу: розглянути варіанти структур ВМР-формату, ознайомитись з поняттям палітри і її керуванням, алгоритмами стиснення, що застосовуються для ВМР-файлів і користуючись набутою теоретичною базою відтворити невелике зображення ВМР-формату в режимі 12h за допомогою засобів мови програмування С.

Формат ВМР використовується для збереження даних без втрат якості. Це власний формат Windows, який використовується для цілей системи. Наприклад, саме у цьому форматі зображення включаються у склад виконуючих файлів і виводяться на екран. Будь-яка програма, яка використовує графічний інтерфейс Windows, містить засоби для відкриття, редагування та збереження ВМР-файлів. Цей формат зберігає зображення з достатньою якістю і глибиною кольору, однак має ряд недоліків, наприклад, не дозволяє використовувати корекцію кольору, зберігати шари і інші елементи зображення. Саме тому він використовується досить рідко, в основному як кінцевий формат при підготовці графіки для Windows.

Для здійснення поставленої задачі необхідно докладно розглянути структуру ВМР-формату

1. Огляд літератури

1.1 Загальна характеристика ВМР-формату

ВМР (від англ. Bitmap Picture – бітовий масив) – формат для збереження растрових зображень [1]. Початково формат міг зберігати лише апаратно-залежні растри (англ.. Device Dependent Bitmap, DDB), але з розвитком технологій відображення графічних даних формат ВМР став переважно зберігати апаратно-незалежні растри (англ.. Device Independent Bitmap, DIB).

Формат ВМР початково розроблявся двома корпораціями Intel і Microsoft, і в той час був однаковим для обох операційних систем Intel OS/2 Warp і Microsoft Windows 2.x. Далі фірма Microsoft розширила формат, розширивши структури (при цьому зберігши як зворотню, так і пряму сумісність для нестиснених різновидів) і додавши підтримку компресії.

З форматом ВМР працює велика кількість програм, так як його підтримка інтегрована в операційні системи Windows і OS/2. Файли формату ВМР можуть мати розширення .bmp, .dib і .rle. Крім того, дані цього формату включаються в двійкові файли ресурсів RES і в PE-файли.

Глибина кольору у даному форматі може бути від 1 до 48 біт на піксель, максимальний розмір зображення 65535\*65535 пікселів.

В форматі ВМР є підтримка стиснення по алгоритму RLE, однак тепер існують формати з сильнішим стисненням, і через великий об’єм ВМР рідко застосовується в Інтернеті, де для стиснення без втрат використовується PNG і більш старий GIF.

Апаратно-залежні або DDB бітмапи використовуються в Windows для збереження зображень в пам’яті різноманітних графічних пристроїв (відеопам’яті). Фактично такий бітмап являє собою урізану версію апаратно-незалежного. Його дані формуються таким чином, щоб відповідати конкретному графічному режиму, крім того, такий бітмап містить спрощений заголовок. Наприклад, для старого 16 кольорового EGA/VGA відеоадаптера такий бітмап буде являти собою 3-х колірні матриці (для кожного з кольорів), апаратно-незалежний бітмап буде містити всього одну матрицю розрядністю 4 біти на піксель.

Оскільки структура DDB бітмапу змінюється в залежності від пристрою до пристрою, то він, як правило, створюється прямо в пам’яті і не зберігається у файл. Для збереження у файл DDB конвертується у DIB.

1.2 Варіанти структур ВМР-формату

Встановлений формат бітмапу складається зі структури BITMAPFILEHEADER, за якою слідують структури BITMAPINFOHEADER, BITMAPV4HEADER або BITMAPV5HEADER [2]. Далі за головною інформаційною структурою бітмапу йде масив структур RGBQUAD (що називається палітрою). Палітра завершується другим масивом індексів кольорів палітри (фактичні дані бітмапу).

Структуру файлу можна умовно зобразити так (Табл. 1):

   Таблиця 1

|  |
| --- |
| BITMAPFILEHEADER |
| BITMAPINFOHEADER |
| RGBQUAD array |
| Color-index array |

Складові структури BITMAPFILEHEADER ідентифікують файл, визначають розмір файлу в байтах і визначають положення першого байта даних по бітмап відносно першого байта заголовку.

Складові структур BITMAPINFOHEADER, BITMAPV4HEADER або BITMAPV5HEADER визначають ширину і довжину зображення в пікселях, чи було до зображення застосоване стиснення і якого типу, кількість байтів даних зображення, початкова роздільна здатність зображення, кількість кольорів, використана в зображенні. Структура RGBQUAD визначає інтенсивність значень RGB для кожного кольору з палітри.

У наступній таблиці (Табл. 2) показані байти даних, які зв’язані зі структурами в ВМР-файлі.

Таблиця 2

|  |  |
| --- | --- |
| Структури | Відповідні байти |
| BITMAPFILEHEADER | 0x00 0x0D |
| BITMAPINFOHEADER | 0x0E 0x35 |
| RGBQUAD array | 0x36 0x75 |
| Color-index array | 0x76 0x275 |

Розглянемо докладніше окремі компоненти структури.

1.2.1 BITMAPFILEHEADER

На початку розташований заголовок файлу (BITMAPFILEHEADER)[3]. Структура BITMAPFILEHEADER містить інформацію про типе, розмір і представлення даних у файлі [2]. Цей заголовок файлу описаний наступним чином:

typedef struct tagBITMAPFILEHEADER {

  WORD  bfType;

  DWORD bfSize;

  WORD  bfReserved1;

  WORD  bfReserved2;

  DWORD bfOffBits;

} BITMAPFILEHEADER, \*PBITMAPFILEHEADER;

Тип WORD повинен мати розмір 16 біт, типи DWORD i LONG – 32 біти, тип LONG – знаковий [1].

bfType визначає тип файлу [3]. Тут він повинен бути ВМ. Якщо відкриєте будь-який файл ВМР у текстовому редакторі, то видно, що перші два символи – це ВМ (від слова BitMap). Символи «ВМ» в шістнадцятковій системі числення мають представлення: 0х42 0х4d [1].

bfSize – це розмір самого файлу в байтах. Більшість програм для того, щоб впевнитись, що перед ними справді ВМР, а не «підробка» повинні, по-перше, перевірити, що bfType містить «ВМ» (без лапок), і, по-друге, що bfSize дорівнює розміру файлу.

bfReserved1 и bfReserved2 – зарезервовані і повинні бути нулями.

bfOffBits – одне з найважливіших полів в цій структурі. Воно містить інформацію про те, де починається сам бітовий масив відносно початку файлу (від початку структури BITMAPFILEHEADER [2]), який описує зображення. Тобто, щоб гарантовано потрапити на початок масиву потрібно писати:

fsetpos (hFile, bfh.bfOffBits);

1.2.2 BITMAPINFOHEADER, BITMAPV4HEADER BITMAPV5HEADER

Після заголовку файлу стоїть одна з трьох варіантів структур: BITMAPINFOHEADER, BITMAPV4HEADER або BITMAPV5HEADER [1].

BITMAPINFOHEADER – найпростіший варіант заголовку. Додатки для Windows NT3.51 і більш ранніх можуть використовувати лише цю структуру. Вона оголошена наступним чином:

typedef struct tagBITMAPINFOHEADER

{

   DWORD  biSize;

   LONG   biWidth;

   LONG   biHeight;

   WORD   biPlanes;

   WORD   biBitCount;

   DWORD  biCompression;

   DWORD  biSizeImage;

   LONG   biXPelsPerMeter;

   LONG   biYPelsPerMeter;

   DWORD  biClrUsed;

   DWORD  biClrImportant;

} BITMAPINFOHEADER, \*PBITMAPINFOHEADER;

biSize – це розмір самої структури BITMAPINFOHEADER, тобто інформація про кожен піксель [3]. Її потрібно ініціалізувати наступним чином : bih.biSize = sizeof (BITMAPINFOHEADER); тут і далі ми вважаємо, що bih оголошена наступним чином: BITMAPINFOHEADER bih;

biWidth – ширина зображення в пікселях [1]. Для Win98/Me і Win200/XP: якщо поле biCompression містить BI\_JPEG або BI\_PNG – тут вказана ширина розпакованого зображення.

biHeight – задає висоту зображення у пікселях. Якщо містить додатне значення – зображення записане у порядку знизу догори, нульовий піксель в нижньому правому кутку. Якщо значення від’ємне – зображення записане згори донизу, нульовий піксель – у верхньому лівому кутку зображення. Поле biCompression повинно містити значення BI\_RGB або BI\_BITFIELDS. Таке зображення не може бути стиснуте.

biPlanes – задає кількість площин. Поки що це поле завжди містить 1.