BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN

MÔN HỌC: KỸ THUẬT LẬP TRÌNH

ĐỀ TÀI: BITMAP

Người thực hiện:

Nguyễn Gia Phúc

MSSV 21120529

TP. HỒ CHÍ MINH THÁNG 5/2022

Mục lục

Mục lục hình ảnh

1. Hàm printusage

Hàm $void\ printusage()$ có chức năng in ra cách sử dụng chương trình khi thực thi mà không có tham số dòng lệnh hoặc sai cú pháp.

```
Usage:

-conv <input file path> <output file path>              Convert 24bpp or 32bpp image to 8bpp image.
-zoom <input file path> <output file path> <scale>      Downsampling the input image.
```

Ảnh 1-1: Hàm printusage

2. Hàm choose

Cú pháp: $int\ choose(int\ n, char*\ str)$

Có chức năng trả về một số từ tham số dòng lệnh mà người dùng nhập vào.

Khi người dùng nhập tham số $-conv$ thì sẽ trả về 1 và $-zoom$ sẽ trả về 2. Từ đó thực hiện các bước kế tiếp trong hàm $main$.

3. Hàm cleanup

Cú pháp: $void\ cleanup(BMP\&\ bmp)$

Nhận vào một struct BMP có tham chiếu.

Có chức năng giải phóng bộ nhớ được cấp phát động trong struct BMP như $rDIB$, $colourTable$, và $imgData$.

4. Hàm BMPread

Cú pháp: $int\ BMPread(BMP\&\ bmp, char*\ input)$

Nhận vào một struct BMP và đường dẫn file cần đọc. Hàm trả về 1 nếu thực thi thành công, ngược lại trả về 0.

Hàm sẽ đọc lần lượt từ file (dạng nhị phân) có đường dẫn mà người dùng nhập vào: 14 bytes đầu cho $bmp.Header$, 40 bytes tiếp theo cho các thông tin cần thiết vào $bmp.DIB$ (các thông tin khác của DIB nếu có sẽ được lưu vào $bmp.rDIB$), Nếu vẫn chưa đọc đến $dataOffset$ thì phần dữ liệu từ sau DIB đến

trước $dataOffset$ sẽ được lưu vào $bmp.colourTable$. Dữ liệu ảnh bắt đầu từ $dataOffset$ đến hết file sẽ lưu vào $bmp.imgData$.

5. Hàm BMPsave

Cú pháp: $int\ BMPsave(BMP\ bmp, char*\ output)$

Nhận vào một struct BMP và đường dẫn nơi cần lưu file. Hàm trả về 1 nếu thực thi thành công, ngược lại trả về 0.

Hàm sẽ ghi lần lượt các dữ liệu $bmp.Header$, $bmp.DIB$, $bmp.rDIB$ (nếu có), $bmp.colourTable$ (nếu có) và $bmp.imgData$ vào file (sẽ được tạo trong lúc thực thi) tại đường dẫn chỉ định.

6. Hàm conv

Cú pháp: $int\ conv(BMP\ srcBmp, BMP\&\ dstBmp)$

Nhận vào hai struct BMP nguồn (không tham chiếu) và đích (có tham chiếu).

Đầu tiên, hàm sẽ kiểm tra ảnh nguồn có hợp lệ hay không (24bpp hoặc 32bpp), nếu không sẽ kết thúc hàm.

Tiếp đến, copy và trích xuất các thông tin cần thiết từ ảnh nguồn sang ảnh đích và chỉnh sửa cho phù hợp: từ 24bpp hoặc 32bpp thành 8bpp, vị trí $dataOffset$ và kích thước ảnh.

Ảnh 8bpp cần colour table để có thể hiển thị được, đoạn code sau sẽ dựng lại colour table cho ảnh:

```
//create colour table
dstBmp.colourTable = new char[256 * 4];
for (int i = 0; i < 256; i++)
{
    dstBmp.colourTable[i * 4 + 0] = (char)i;
    dstBmp.colourTable[i * 4 + 1] = (char)i;
    dstBmp.colourTable[i * 4 + 2] = (char)i;
    dstBmp.colourTable[i * 4 + 3] = (char)i;
}
```

Ảnh 6-1: Phục dựng colour table

Tiếp theo, dùng 2 vòng lặp for, những con trỏ dòng và pixel để dịch chuyển trong vùng dữ liệu điểm ảnh của ảnh nguồn (bỏ qua được padding byte bằng cách này).

Ở ảnh 24bpp, thứ tự sẽ là BGR. Còn ở 32bpp là ABGR hoặc BGRA (do không tìm thấy ảnh ABGR nên trong phần code cũng như ảnh demo sẽ dùng ảnh BGRA để thay thế).

Cuối cùng, lấy trung bình cộng cả B, G, R gán vào ảnh mới và dịch chuyển sang pixel tiếp theo. Lặp lại quá trình trên đến khi duyệt qua tất cả các pixel.

7. Hàm zoom

Cú pháp: $int\ zoom(BMP\ srcBmp, BMP\&\ dstBmp, int\ s)$

Nhận vào hai struct BMP nguồn (không tham chiếu), đích (có tham chiếu) và tỉ lệ thu nhỏ s.

Đầu tiên, hàm sẽ kiểm tra ảnh nguồn có hợp lệ hay không (8bpp hoặc 24bpp hoặc 32bpp), nếu không sẽ kết thúc hàm.

Tiếp đến, copy và trích xuất các thông tin cần thiết từ ảnh nguồn sang ảnh đích và chỉnh sửa cho phù hợp: chiều cao, chiều rộng và kích thước ảnh

Tiếp theo, dùng 2 vòng lặp for, những con trỏ dòng và pixel để dịch chuyển trong vùng dữ liệu điểm ảnh của ảnh nguồn (bỏ qua được padding byte bằng cách này).

Trong 2 vòng lặp for trên còn có thêm 2 vòng lặp for để duyệt qua từng ô vuông có kích thước $s * s$ sau đó tính trung bình cộng từng kênh màu (BGR với 24bpp hoặc 32bpp và Grey với 8bpp) và gán vào ảnh mới.

Dịch chuyển sang ô vuông tiếp theo và lặp lại công việc cho đến khi duyệt qua tất cả các pixel trong ảnh mới.