Отчет по лабораторной работе № 1

«Работа с большими массивами пикселов»

7 вариант

Выполнил: студент группы P3317

Плюхин Д.А.

Преподаватель: Зинчик Александр Адольфович

# **Задание**

Написать программу, выполняющую следующее задание. Для произвольного графического файла формата BMP размером не менее чем 2000х2000 выполнить следующие действия:

1) Загрузить и отобразить файл.

2) Установить интенсивности синего канала изображения значение 128. Пользоваться функциями GetPixel и SetPixel запрещено.

3) Отобразить результат и сохранить новый файл.

Оценить отношение скорости работы функции BitBlt и пары функций GetPixel и SetPixel и построить график зависимости этого отношения от размеров изображения. Размеры изображения от 200х200 до 2000х2000 с шагом увеличения каждого размера на 200 (200х200, 400х400 и т.д.)

# **Исходный код**

## Загрузка и отображение файла

LPCTSTR fileName = getFileName();

hBitmap = (HBITMAP)LoadImage(0, fileName, IMAGE\_BITMAP, 0, 0, LR\_LOADFROMFILE | LR\_CREATEDIBSECTION);

InvalidateRect(hwnd, NULL, TRUE);

isLoaded = true;

## Установка интенсивности синего канала изображения равным 128 и отображение файла

BITMAP bm;

HDC hCompatibleDC = CreateCompatibleDC(NULL);

HBITMAP hOldBitmap = (HBITMAP)SelectObject(hCompatibleDC, hBitmap);

GetObject(hBitmap, sizeof(bm), &bm);

SelectObject(hCompatibleDC, GetStockObject(DC\_BRUSH));

SetDCBrushColor(hCompatibleDC, RGB(255, 255, 0));

BitBlt(hCompatibleDC, 0, 0, bm.bmWidth, bm.bmHeight, hCompatibleDC, 0, 0, MERGECOPY);

//dest\_color = src\_color \* 0x00FFFF

BitBlt(hCompatibleDC, 0, 0, bm.bmWidth, bm.bmHeight, hCompatibleDC, 0, 0, DSTINVERT);

// dest\_color = not(src\_color \* 0x00FFFF)

SetDCBrushColor(hCompatibleDC, RGB(255, 255, 127));

BitBlt(hCompatibleDC, 0, 0, bm.bmWidth, bm.bmHeight, hCompatibleDC, 0, 0, MERGECOPY);

//dest\_color = not(src\_color \* 0x00FFFF) \* 0x800000

BitBlt(hCompatibleDC, 0, 0, bm.bmWidth, bm.bmHeight, hCompatibleDC, 0, 0, DSTINVERT);

//dest\_color = (src\_color \* 0x00FFFF) + 0x800000

saveBitmap(hCompatibleDC, hBitmap, bm.bmWidth, bm.bmHeight);

InvalidateRect(hwnd, NULL, TRUE);

SelectObject(hCompatibleDC, hOldBitmap);

DeleteDC(hCompatibleDC);

## Сохранение нового файла

BITMAPFILEHEADER bmfHeader;

BITMAPINFOHEADER bi;

bi.biSize = sizeof(BITMAPINFOHEADER);

bi.biWidth = width;

bi.biHeight = height;

bi.biPlanes = 1;

bi.biBitCount = 24;

bi.biCompression = BI\_RGB;

bi.biSizeImage = 0;

bi.biXPelsPerMeter = 0;

bi.biYPelsPerMeter = 0;

bi.biClrUsed = 0;

bi.biClrImportant = 0;

DWORD dwBmpSize = ((width \* bi.biBitCount + 31) / 32) \* 4 \* height;

HANDLE DIB = GlobalAlloc(GHND, dwBmpSize);

char \*lpbitmap = (char \*)GlobalLock(hDIB);

GetDIBits(hdc, H, 0,

(UINT)height,

lpbitmap,

(BITMAPINFO \*)&bi, DIB\_RGB\_COLORS);

HANDLE hFile = CreateFile("images\\image\_new.bmp",

GENERIC\_WRITE,

0,

NULL,

CREATE\_ALWAYS,

FILE\_ATTRIBUTE\_NORMAL, NULL);

DWORD dwSizeofDIB = dwBmpSize + sizeof(BITMAPFILEHEADER) + sizeof(BITMAPINFOHEADER);

bmfHeader.bfOffBits = (DWORD)sizeof(BITMAPFILEHEADER) + (DWORD)sizeof(BITMAPINFOHEADER);

bmfHeader.bfSize = dwSizeofDIB;

bmfHeader.bfType = 0x4D42;

DWORD dwBytesWritten = 0;

WriteFile(hFile, (LPSTR)&bmfHeader, sizeof(BITMAPFILEHEADER), &dwBytesWritten, NULL);

WriteFile(hFile, (LPSTR)&bi, sizeof(BITMAPINFOHEADER), &dwBytesWritten, NULL);

WriteFile(hFile, (LPSTR)lpbitmap, dwBmpSize, &dwBytesWritten, NULL);

//Unlock and Free the DIB from the heap

GlobalUnlock(hDIB);

GlobalFree(hDIB);

//Close the handle for the file that was created

CloseHandle(hFile);

# **Оценка скорости работы функций GetPixel/SetPixel и BitBlt**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| bitblt duration (seconds): | 0,012 | 0,013 | 0,072 | 0,037 | 0,034 | 0,044 | 0,062 | 0,08 | 0,102 | 0,109 |
| pixel duration (seconds): | 0,329 | 1,309 | 2,85 | 5,045 | 7,962 | 11,424 | 15,63 | 20,376 | 25,876 | 31,206 |
| size (pixels): | 200 | 400 | 600 | 800 | 1000 | 1200 | 1400 | 1600 | 1800 | 2000 |
| ratio: | 27,417 | 100,69 | 39,583 | 136,35 | 234,18 | 259,636 | 252,097 | 254,7 | 253,69 | 286,294 |

# **Вывод**

Таким образом, скорость работы пары функций SetPixel/GetPixel значительно меньше скорости работы функции BitBlt, причем отношение их скоростей значительно увеличивается при увеличении размеров изображения и а.