

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики
Кафедра информатики и прикладной математики

Проектирование человеко-машинных интерфейсов

Лабораторная работа №1

Выполнил Бакшенов В.О.

Группа Р3318

Преподаватель : Зинчик А.А.

2017 г.

Текст Задания

1. Написать программу, выполняющую следующее задание. Для произвольного графического файла формата BMP размером не менее чем 2000x2000 выполнить следующие действия:

1) Загрузить и отобразить файл.
2) Применить действие, согласно варианту задания. Пользоваться функциями GetPixel и SetPixel запрещено. +`

3) Отобразить результат и сохранить новый файл.
2. Оценить отношение скорости работы функции BitBlt и пары функций GetPixel и SetPixel при решении вашего варианта задания и построить график зависимости этого отношения от размеров изображения. Размеры изображения от 200x200 до 2000x2000 с шагом увеличения каждого размера на 200 (200x200, 400x400 и т.д.)

Действие - Убрать синий канал изображения

Текст Программы

```
PAINTSTRUCT ps;  
  
HDC hdc = BeginPaint(hWnd, &ps);  
  
hBitmap = (HBITMAP)LoadImageW(NULL, L"600.bmp", IMAGE_BITMAP, 0, 0,  
LR_LOADFROMFILE | LR_CREATEDIBSECTION);  
  
//Получаем ее свойства в bm  
GetObject(hBitmap, sizeof(bm), &bm);  
HDC hdcBuffer = GetDC(hWnd);  
hdc = CreateCompatibleDC(hdcBuffer);  
  
//привязываем наш битмап к контексту  
SelectObject(hdc, hBitmap);  
  
RECT rcClient;  
GetClientRect(hWnd, &rcClient);  
  
BitBlt(hdcBuffer,  
    0, 0,  
    rcClient.right / 2, rcClient.bottom,  
    hdc,  
    0, 0,  
    SRCCOPY);  
  
HBRUSH hRed = CreateSolidBrush(RED);  
HBRUSH hOld = (HBRUSH) SelectObject(hdcBuffer, hRed);  
  
double start = clock();  
  
BitBlt(hdcBuffer,  
    rcClient.right / 2, 0,  
    rcClient.right / 2, rcClient.bottom,  
    hdc,  
    0, 0, MERGECOPY);  
  
double finish = clock();  
double res = (finish - start) / CLOCKS_PER_SEC;  
  
ReleaseDC(hWnd, hdcBuffer);  
EndPaint(hWnd, &ps);
```

Используя функции Get/SetPixels

```
//=====КЛЮЧЕВАЯ ЧАСТЬ=====
```

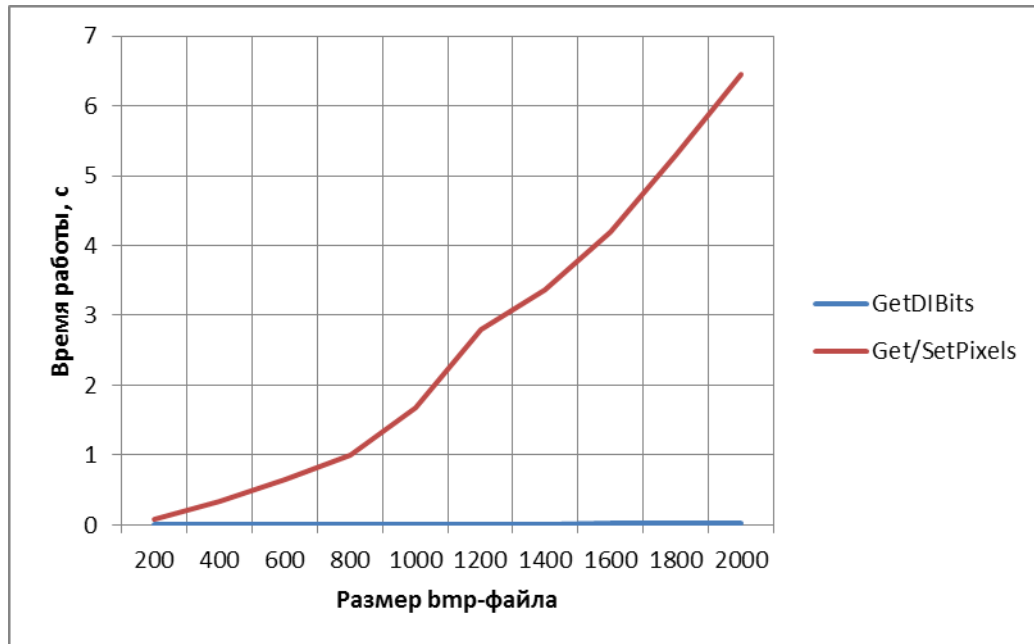
```
double start = clock();
```

```

COLORREF obgr;
for (int i = 0; i < bm.bmHeight; i++)
{
    for (int j = 0; j < bm.bmWidth; j++)
    {
        obgr = GetPixel(hdc, j, i); // получаем пиксель в формате 0bgr
        obgr = obgr << 16;           // сдвигаем до состояния gr00
        obgr = obgr >> 16;           // сдвигаем назад и получаем 00gr
        SetPixel(hdc, j, i, obgr);
    }
}
double finish = clock();
*res = (finish - start) / CLOCKS_PER_SEC;
//=====КЛЮЧЕВАЯ ЧАСТЬ=====

```

Графики



Вывод - Функции Get/SetPixels обрабатывают изображение на порядок медленнее, чем при непосредственной работе с массивом байт.