实验一：BMP 图像文件格式研究

课程号：310025010　**课序号**：01　**课程名称**：生物医学图像处理实验　**任课教师**：林江莉

组长：徐广玄

小组成员：**无**

实验报告日期：　　2023 年　　3 月　　9 日

一、实验内容：

1. 编写程序读取各种位数（1 位、2 位，4 位、8 位，16 位，24 位，32 位）的 bmp 文件，显示各个像素点的坐标位置和颜色值。

答：见下文。

2. 与 MFC 的 GetPixel 函数得到 RGB 值对比，看是否能很好的匹配。

答：非常匹配。

3. 思考：那些具有调色板，那些没有调色板？

答：仅有 8、4、1 位图像有调色板，但是 16 位中 565 格式根据操作系统的不同存在 12/16 字节的调色盘掩码。

二、编程软件：

VS2022 MFC 窗口视图

三、实验步骤及关键代码：

- **打开图像并显示**

本次实验中添加了如下函数：

// 定义掩码和全局变量

```
#define RGB555_MASK_RED      0x7C00
#define RGB555_MASK_GREEN    0x03E0
#define RGB555_MASK_BLUE     0x001F

static BYTE image[4][1500][1500];
static BYTE image_sew[4][1500][1500];
static BYTE palette[256][4];
static BYTE color[1];
static short bit;
static short is555;
static int index;
```

// 显示图像函数

```
void DispColorImage(CDC* p, BYTE image[4][1500][1500], long height, long width, int dx, int dy)
{
    for (int i = 0; i < height; i++)
    {
        for (int j = 0; j < width; j++)
        {
            p->SetPixel(j + dx, i + dy, RGB(image[0][i][j], image[1][i][j], image[2][i][j]));
        }
    }
}
```

// 按键1打开图像

```
void CDIPexperiment01Dlg::OnBnClickedButton01()
{
    // TODO: 在此添加控件通知处理程序代码
    long height, width;
    CFileDialog dlg(true);
    if (dlg.DoModal() == IDOK)
    {
        CDC* p = GetDlgItem(ID_IMAG)->GetDC();
        ReadBmpImage(dlg.GetFileName(), image, height, width);
        DispColorImage(p, image, height, width, 0, 0);  // dx dy表示相比左上角的偏移量
    }
    else
        return;
}
```

// 按键2函数：输入xy坐标，输入rgb坐标

```
void CDIPexperiment01Dlg::OnBnClickedButton02()
{
    // TODO: 在此添加控件通知处理程序代码
    CString str1, str2, strR, strG, strB;
    long height, width;
    CFileDialog dlg(true);

    int x, y;
    EDIT02.GetWindowText(str1);
    x = _ttol(str1) - 1;
    EDIT01.GetWindowText(str2);
    y = _ttol(str2) - 1;
```

```
    strR.Format(_T("%d"), image[0][x][y]);
    strG.Format(_T("%d"), image[1][x][y]);
    strB.Format(_T("%d"), image[2][x][y]);

    EDIT03.SetWindowText(strR + " " + strG + " " + strB);
}
```

- **获取像素点相关信息（着重写这部分关键代码）**

头代码（该函数分为开头读取和if语句判断多少位数字信息）：

```
BOOL ReadBmpImage(CString fileName, BYTE image[4][1500][1500], long &height, long &width)
{
    char bmp[2];
    CFile file;
    int c;

    file.Open(fileName, CFile::modeRead);
    file.Read(bmp, 2);

    if (bmp[0] != 'B' || bmp[1] != 'M')
    {
        return 0;
    }

    file.Seek(16, CFile::current);  // 指针指向宽度信息
    file.Read(&width, 4);
    file.Read(&height, 4);
    file.Seek(2, CFile::current);
    file.Read(&bit, 2);  // 指针指向像素的颜色信息
    file.Read(&is555, 4);  // 查找16位储存类型
    file.Seek(20, CFile::current);
}
    // 仅有8、4、1位图像有调色板
```

黑白：

```
if (bit == 1)
    {
        c = (width / 8 + 1) % 4;  // 多余字节
        // 给调色盘赋值
        for (int x = 0; x < 2; x++)
        {
            for (int y = 0; y < 4; y++)
            {
                file.Read(&palette[x][y], 1);
```

```
            }
        }
        for (int i = height - 1; i >= 0; i--)
        {
            for (int j = 0; j <= width - 1; j++)
            {
                file.Read(&index, 1);
                for (int k = 2; k >= 0; k--)
                {
                    image[k][i][j] = palette[index][2 - k];
                }
            }
            if (c != 0)
            {
                file.Seek(4 - c, CFile::current);
            }
        }
        file.Close();
        return(1);
    }
```

2位：

```
if (bit == 2)
    {
        c = (width / 4 + 1) % 4;  // 多余字节
        // 给调色盘赋值
        for (int x = 0; x < 4; x++)
        {
            for (int y = 0; y < 4; y++)
            {
                file.Read(&palette[x][y], 1);
            }
        }
        for (int i = height - 1; i >= 0; i--)
        {
            for (int j = 0; j <= width - 1; j++)
            {
                file.Read(&index, 1);
                for (int k = 2; k >= 0; k--)
                {
                    image[k][i][j] = palette[index][2 - k];
                }
            }
            if (c != 0)
```

```
            {
                file.Seek(4 - c, CFile::current);
            }
        }
        file.Close();
        return(1);
    }
```

4位：

```
if (bit == 4)
    {
        c = (width / 2 + 1) % 4;  // 多余字节
        // 给调色盘赋值
        for (int x = 0; x < 16; x++)
        {
            for (int y = 0; y < 4; y++)
            {
                file.Read(&palette[x][y], 1);
            }
        }
        for (int i = height - 1; i >= 0; i--)
        {
            for (int j = 0; j <= width - 1; j++)
            {
                file.Read(&index, 1);
                for (int k = 2; k >= 0; k--)
                {
                    image[k][i][j] = palette[index][2 - k];
                }
            }
        }
        file.Close();
        return(1);
    }
```

8位：

```
if (bit == 8)
    {
        c = width % 4;  // 多余字节
        // 给调色盘赋值
        for (int x = 0; x < 256; x++)
        {
            for (int y = 0; y < 4; y++)
```

```
            {
                file.Read(&palette[x][y], 1);
            }
        }
        for (int i = height - 1; i >= 0; i--)
        {
            for (int j = 0; j <= width - 1; j++)
            {
                file.Read(&index, 1);
                for (int k = 2; k >= 0; k--)
                {
                    image[k][i][j] = palette[index][2 - k];
                }
            }
            if (c != 0)
            {
                file.Seek(4 - c, CFile::current);
            }
        }
        file.Close();
        return(1);
    }
```

16位（此处根据变量is555分别两种格式，其中565格式根据操作系统的不同存在12/16字节的调色盘掩码）：

```
if (bit == 16 && is555 == BI_BITFIELDS)  // 565格式
    {
        file.Seek(12, CFile::current);  // 此处为12字节
        c = width * 2 % 4;  // 多余字节
        for (int i = height - 1; i >= 0; i--)
        {
            for (int j = 0; j <= width - 1; j++)
            {
                for (int k = 0; k <= 2; k++)
                {
                    if (k == 0)
                    {
                        file.Seek(1, CFile::current);
                        file.Read(&image[k][i][j], 1);
                        image[k][i][j] = (image[k][i][j] >> 3) << 3;
                    }
                    else if (k == 1)
                    {
```

```
                    file.Seek(-1, CFile::current);
                    file.Read(&image[k][i][j], 1);
                    image[k][i][j] = image[k][i][j] << 5;
                    file.Seek(-2, CFile::current);
                    file.Read(&image_sew[k][i][j], 1);
                    image_sew[k][i][j] = (image_sew[k][i][j] >> 5) << 2;
                    image[k][i][j] = image[k][i][j] | image_sew[k][i][j];
                }
                else
                {
                    file.Seek(-1, CFile::current);
                    file.Read(&image[k][i][j], 1);
                    image[k][i][j] = image[k][i][j] << 3;
                }
            }
            file.Seek(1, CFile::current);
        }
        if (c != 0)
        {
            file.Seek(4 - c, CFile::current);
        }
    }
    file.Close();
    return(1);
}

if (bit == 16 && is555 == BI_RGB)  // 555格式
{
    c = width * 2 % 4;  // 多余字节
    for (int i = height - 1; i >= 0; i--)
    {
        for (int j = 0; j <= width - 1; j++)
        {
            for (int k = 0; k <= 2 ; k++)
            {
                if (k == 0)
                {
                    file.Seek(1, CFile::current);
                    file.Read(&image[k][i][j], 1);
                    image[k][i][j] = (image[k][i][j] >> 2) << 3;
                }
                else if (k == 1)
                {
                    file.Seek(-1, CFile::current);
```

```
                        file.Read(&image[k][i][j], 1);
                        image[k][i][j] = image[k][i][j] << 6;
                        file.Seek(-2, CFile::current);
                        file.Read(&image_sew[k][i][j], 1);
                        image_sew[k][i][j] = (image_sew[k][i][j] >> 5) << 3;
                        image[k][i][j] = image[k][i][j] | image_sew[k][i][j];
                    }
                    else
                    {
                        file.Seek(-1, CFile::current);
                        file.Read(&image[k][i][j], 1);
                        image[k][i][j] = image[k][i][j] << 3;
                    }
                }
                file.Seek(1, CFile::current);
            }
            if (c != 0)
            {
                file.Seek(4 - c, CFile::current);
            }
        }
        file.Close();
        return(1);
    }
```

24位：

```
if (bit == 24)
    {
        c = width * 3 % 4;  // 多余字节
        for (int i = height - 1; i >= 0; i--)
        {
            for (int j = 0; j <= width - 1; j++)
            {
                for (int k = 2; k >= 0; k--)
                {
                    file.Read(&image[k][i][j], 1);
                }
            }
            if (c != 0)
            {
                file.Seek(4 - c, CFile::current);
            }
        }
        file.Close();
```

```
        return(1);
    }
```

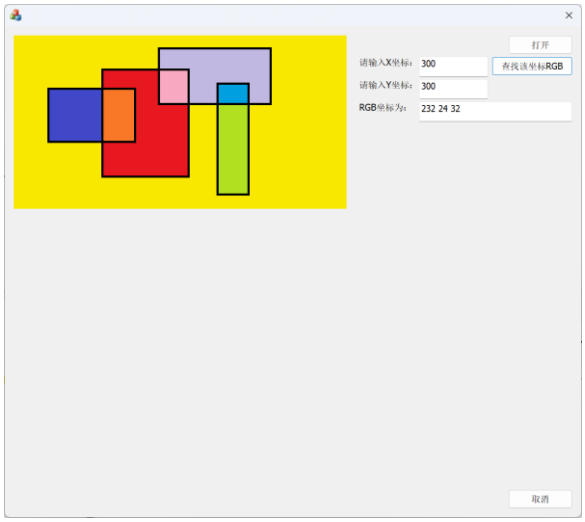
32位：

```
if (bit == 32)
    {
        for (int i = height - 1; i >= 0; i--)
        {
            for (int j = 0; j <= width - 1; j++)
            {
                for (int k = 3; k >= 0; k--)
                {
                    file.Read(&image[k][i][j], 1);
                }
            }
        }
        file.Close();
        return(1);
    }
```

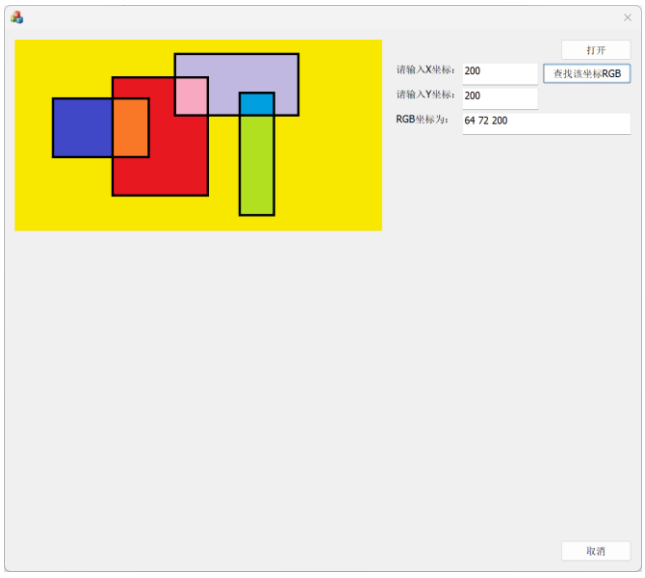
四、实验结果：

图一 群实验测试文件 565 格式

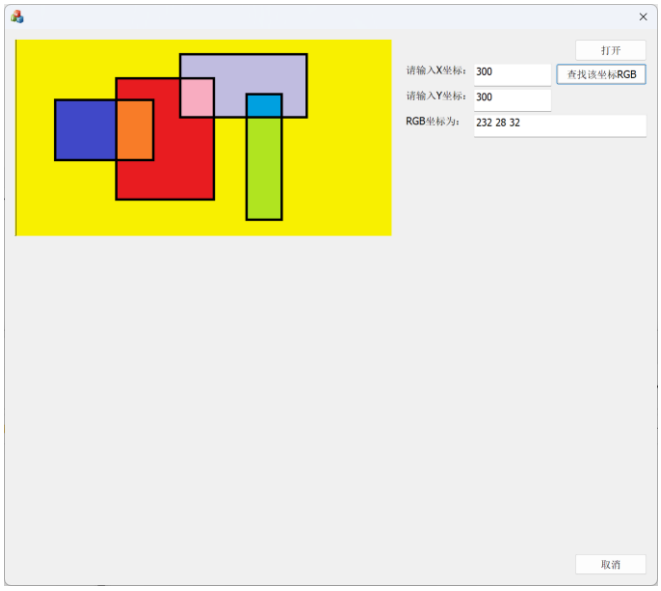
图二 群实验测试文件 555 格式



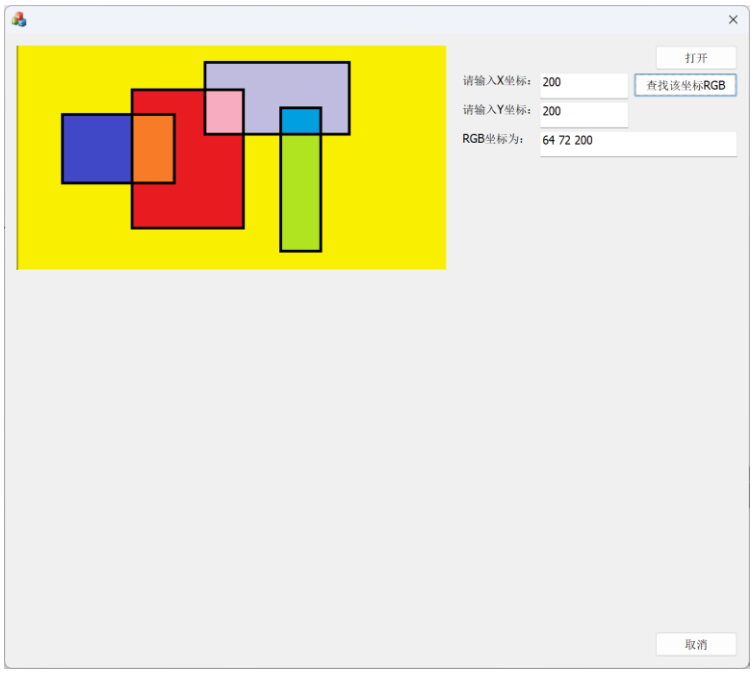
图三 群实验测试文件 555 格式 坐标一



图四 群实验测试文件 555 格式 坐标二



图五 群实验测试文件 565 格式 坐标一



图六 群实验测试文件 565 格式 坐标二

总结：根据图一图二可以发现，此文档可以打开测试文件，根据三到六的图片可以发现，虽然 555 和 565 肉眼看不出区别，但是在 RGB 查询功能中可以发现，其坐标还是有少量区别。

五、小组各成员的分工

全都由徐广玄一人完成。

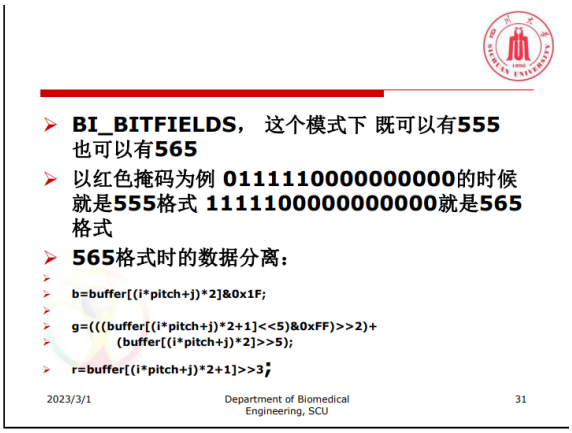
六、结果分析及其实验小结：

编程中问题及其解决：

1、 因为版本过高，使用单文档时出现大量无法解决 bug。解决办法：改用基于对话框。

2、 16 位文件打开后出现大量问题，如色彩不对、偏移量大。解决办法：注意到 565 格式仍有调色盘，其次 ppt 上的伪代码有大量错误，如：

图七 第四部分掩码在 win11 不存在



图八 555 和 565 不能高位补零，而是应该低位补零，更不应该做“color”的乘除法

3、 发现色彩有错误。解决办法：win 系统采用小段对齐，高八位信息储存在后一位字节。

编程中的注意事项：如 取整，变量类型 等。

1、 由于 MFC 仅存在 CString 类别，故无法像常规 C++一样轻松转换 String 和 Int 类型。我们采用：`_ttol()`和`str.Format(_T("%d"), image[0][x][y])`来进行String和Int类型的相互转换。

2、 需要注意变量类型：

```
static BYTE image[4][1500][1500];
static BYTE image_sew[4][1500][1500];
static BYTE palette[256][4];
static BYTE color[1];
```

为static BYTE类型，方便进行位操作。