

GG1003

Windows编程

李俊琴

李俊琴

实验4-1 位图和BitBlt

余庆祥

2220631136

BDP

2022/10/13

**内页写作格式**

**一、实验名称：**要用最简练的语言反映实验的内容。

**二、实验日期：**写明做实验的具体年、月、日及组別。

**三、实验目的：**使用简洁的文字或关键字来敘述，是以怎样的目的作此实验的。

**四、实验环境：**实验的操作系统和软件等。

**五、实验的步骤和方法：**这是实验报告极其重要的内容。这部分要写明经过哪几个步骤。

**六、数据记录和计算：**指从实验中测到的数据以及计算结果。

**七、实验结果或结论：**即根据实验过程中所见到的现象和测得的数据，得出结论。

**八、备注或说明**：可写上实验成功或失败的原因，实验后的心得体会、建议等。

|  |  |
| --- | --- |
| **1、实验名称** | 实验4-1 位图和BitBlt |
| **2、实验日期** |  |
| **3、实验目的** | |
| 1．理解位图需要在内存中存储，需要先加载到内存；理解位图也是GDI对象，要创建并且选入设备环境才能使用。  2．理解位图输出跟一般图形输出不同，需要将内存设备环境中的内容使用“位块传输”的方式输出到显示设备环境中。  3．掌握位块传输函数BitBlt、StretchBlt、TransparentBlt函数的使用。 | |
| **4、实验环境** | |
| 1、vs2019 | |
| **5、实验的步骤和方法** | |
| 1. 按照位图输出步骤，准备设备环境（目的、源），加载位图之后选入源设备环境，使用BitBlt函数将位图输出到目的设备环境中。其中，位图文件Monster.bmp要与main.cpp或项目exe文件在同一目录下。注意做好释放、清理工作。效果参照下图：      1. 在上题基础上扩充代码，使用StretchBlt函数将位图放大、缩小输出，注意位图缩小时有失真问题，需要通过SetStretchBltMode函数对缩放模式进行设置。效果参考下图：      1. 修改上题代码，使用TransparentBlt函数将位图透明输出，在函数中通过最后一个参数获取图像左上角像素颜色值作为关键色进行透明处理。注意使用TransparentBlt函数要先配置环境，将msimg32.lib加入项目链接器中的依赖项，或者在主函数的开始加入语句：#pragma comment(lib, "msimg32.lib")。效果参考下图：    思考题  1. 先在Monster的身体上输出文字“I am a monster！”，之后再将它输出到窗口，若它的位置、缩放发生改变，文本也随之发生改变。 2. 与键盘鼠标结合起来，Monster输出后，若按下键盘（任意键）则缩小；若按下鼠标左键则放大。 | |
| **6、数据记录和计算** | |
| **（代码编写思路及关键代码）**  第一题代码：  LRESULT CALLBACK WndProc(HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)  {  HDC hdc, hdcMem;  PAINTSTRUCT ps;  HBITMAP hBitmap;  BITMAP bm;  switch (message)  {  case WM\_PAINT:  hdc = BeginPaint(hwnd, &ps);  hdcMem = CreateCompatibleDC(hdc);  hBitmap = (HBITMAP)LoadImage(NULL, L"Monster.bmp", IMAGE\_BITMAP,  0, 0, LR\_LOADFROMFILE);  SelectObject(hdcMem, hBitmap);  GetObject(hBitmap, sizeof(BITMAP), &bm);  BitBlt(hdc, 100, 20, bm.bmWidth, bm.bmHeight, hdcMem, 0, 0, SRCCOPY);  DeleteDC(hdcMem);  DeleteObject(hBitmap);  EndPaint(hwnd, &ps);  return 0;  break;  case WM\_DESTROY:  PostQuitMessage(0);  return 0;  }  return DefWindowProc(hwnd, message, wParam, lParam);  }  第二题代码：  LRESULT CALLBACK WndProc(HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)  {  HDC hdc, hdcMem;  PAINTSTRUCT ps;  HBITMAP hBitmap;  BITMAP bm;  switch (message)  {  case WM\_PAINT:  hdc = BeginPaint(hwnd, &ps);  hdcMem = CreateCompatibleDC(hdc);  hBitmap = (HBITMAP)LoadImage(NULL, L"Monster.bmp", IMAGE\_BITMAP,  0, 0, LR\_LOADFROMFILE);  SelectObject(hdcMem, hBitmap);  GetObject(hBitmap, sizeof(BITMAP), &bm);  BitBlt(hdc, 50, 20, bm.bmWidth, bm.bmHeight, hdcMem, 0, 0, SRCCOPY);  StretchBlt(hdc, 250, 20, bm.bmWidth \* 2, bm.bmHeight \* 2, hdcMem,  0, 0, bm.bmWidth, bm.bmHeight, SRCCOPY); // 放大2倍  StretchBlt(hdc, 20, 200, bm.bmWidth \* 2 / 3, bm.bmHeight \* 2 / 3, hdcMem,  0, 0, bm.bmWidth, bm.bmHeight, SRCCOPY); // 放大2/3倍  SetStretchBltMode(hdc, COLORONCOLOR);  StretchBlt(hdc, 130, 200, bm.bmWidth \* 2 / 3, bm.bmHeight \* 2 / 3, hdcMem,  0, 0, bm.bmWidth, bm.bmHeight, SRCCOPY); // 放大2/3倍并保真  DeleteDC(hdcMem);  DeleteObject(hBitmap);  EndPaint(hwnd, &ps);  return 0;  break;  case WM\_DESTROY:  PostQuitMessage(0);  return 0;  }  return DefWindowProc(hwnd, message, wParam, lParam);  }  第三题代码：  LRESULT CALLBACK WndProc(HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)  {  HDC hdc, hdcMem;  PAINTSTRUCT ps;  HBITMAP hBitmap;  BITMAP bm;  switch (message)  {  case WM\_PAINT:  hdc = BeginPaint(hwnd, &ps);  hdcMem = CreateCompatibleDC(hdc);  hBitmap = (HBITMAP)LoadImage(NULL, L"Monster.bmp", IMAGE\_BITMAP,  0, 0, LR\_LOADFROMFILE);  SelectObject(hdcMem, hBitmap);  GetObject(hBitmap, sizeof(BITMAP), &bm);  BitBlt(hdc, 20, 20, bm.bmWidth, bm.bmHeight, hdcMem,  0, 0, SRCCOPY);  TransparentBlt(hdc, 180, 20, bm.bmWidth, bm.bmHeight, hdcMem,  0, 0, bm.bmWidth, bm.bmHeight, GetPixel(hdcMem, 0, 0));  TransparentBlt(hdc, 340, 50, bm.bmWidth / 2, bm.bmHeight / 2, hdcMem,  0, 0, bm.bmWidth, bm.bmHeight, GetPixel(hdcMem, 0, 0));  DeleteDC(hdcMem);  DeleteObject(hBitmap);  EndPaint(hwnd, &ps);  return 0;  break;  case WM\_DESTROY:  PostQuitMessage(0);  return 0;  }  return DefWindowProc(hwnd, message, wParam, lParam);  }  思考题第一题代码：  LRESULT CALLBACK WndProc(HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)  {  HDC hdc, hdcMem;  PAINTSTRUCT ps;  HBITMAP hBitmap;  BITMAP bm;  TCHAR str[20] = L"I am a monster！";  int len = lstrlen(str);  RECT rect;  switch (message)  {  case WM\_PAINT:  hdc = BeginPaint(hwnd, &ps);  hdcMem = CreateCompatibleDC(hdc);  hBitmap = (HBITMAP)LoadImage(NULL, L"Monster.bmp", IMAGE\_BITMAP,  0, 0, LR\_LOADFROMFILE);  SelectObject(hdcMem, hBitmap);  GetObject(hBitmap, sizeof(BITMAP), &bm);  SetBkMode(hdcMem, TRANSPARENT);  rect = { 0, 0, bm.bmWidth, bm.bmHeight };  DrawText(hdcMem, str, -1, &rect, DT\_VCENTER | DT\_CENTER | DT\_SINGLELINE);  BitBlt(hdc, 30, 50, bm.bmWidth, bm.bmHeight, hdcMem,  0, 0, SRCCOPY); // 初始样式  BitBlt(hdc, 200, 50, bm.bmWidth, bm.bmHeight, hdcMem,  0, 0, SRCCOPY); // 位置改变  SetStretchBltMode(hdc, COLORONCOLOR);  StretchBlt(hdc, 400, 50, bm.bmWidth \* 2, bm.bmHeight \* 2, hdcMem,  0, 0, bm.bmWidth, bm.bmHeight, SRCCOPY); // 长宽各放大两倍  StretchBlt(hdc, 30, 250, bm.bmWidth / 2, bm.bmHeight / 2, hdcMem,  0, 0, bm.bmWidth, bm.bmHeight, SRCCOPY); // 长宽各缩小两倍  DeleteDC(hdcMem);  DeleteObject(hBitmap);  EndPaint(hwnd, &ps);  return 0;  break;  case WM\_DESTROY:  PostQuitMessage(0);  return 0;  }  return DefWindowProc(hwnd, message, wParam, lParam);  }  思考题第二题代码：  // 全局变量  int n = 3;  HDC hdc, hdcMem;  PAINTSTRUCT ps;  HBITMAP hBitmap;  BITMAP bm;  HPEN hPen, hOldPen;  // 在循环GetMessage()之前  hdc = GetDC(hwnd);  hdcMem = CreateCompatibleDC(hdc);  hBitmap = (HBITMAP)LoadImage(NULL, L"Monster.bmp", IMAGE\_BITMAP,  0, 0, LR\_LOADFROMFILE);  SelectObject(hdcMem, hBitmap);  hPen = CreatePen(PS\_NULL, 1, RGB(255, 255, 255));  hOldPen = (HPEN)SelectObject(hdc, hPen);  GetObject(hBitmap, sizeof(BITMAP), &bm);  SetStretchBltMode(hdc, COLORONCOLOR);  // 在GetMessage()内  StretchBlt(hdc, 50, 50, bm.bmWidth \* n, bm.bmHeight \* n, hdcMem,  0, 0, bm.bmWidth, bm.bmHeight, SRCCOPY); // 长宽各放大两倍  // CALLBACK函数  LRESULT CALLBACK WndProc(HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)  {  switch (message)  {  case WM\_LBUTTONDOWN:  if (n == 5) return 0;  //Rectangle(hdc, 50, 50, bm.bmWidth \* n + 50 + 1, bm.bmHeight \* n + 50 + 1);  n++;  break;  case WM\_KEYDOWN:  if (n == 1) return 0;  Rectangle(hdc, 50, 50, bm.bmWidth \* n + 50 + 1, bm.bmHeight \* n + 50 + 1);  n--;  break;  case WM\_PAINT:  break;  case WM\_DESTROY:  DeleteDC(hdcMem);  DeleteObject(hBitmap);  DeleteObject(hPen);  SelectObject(hdc, hOldPen);  ReleaseDC(hwnd, hdc);  PostQuitMessage(0);  return 0;  }  return DefWindowProc(hwnd, message, wParam, lParam);  } | |
| **7、实验结果或结论（**实验结果怎么样？你从这个实验你学会了什么？得出了什么结论？） | |
| **（实验结果截图+文字说明）**  第一题效果图：    第二题效果图：    第三题效果图：    思考题第一题效果图：    思考题第二题效果图：  原来大小：    按下键盘任意键后：    原来大小的状态下点击鼠标左键后： | |
| **8、备注或说明**  **）** | |
|  | |