

GG1003

Windows编程

李俊琴

李俊琴

实验3-2-文本输出实验

余庆祥

2220631136

BDP

2022/9/30

**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3-2-文本输出实验 |
| **2、实验日期** |  |
| **3、实验目的** | |
| 1．掌握常用的文本输出函数TextOut、DrawText的使用，熟悉其参数设置。  2．掌握字体颜色、背景颜色、背景透明等属性的修改。  3．掌握字体创建函数CreateFont的使用，掌握计算文本大小的两种方法，进而能确定文本输出的合适位置 | |
| **4、实验环境** | |
| 1、vs2019 | |
| **5、实验的步骤和方法** | |
| 1. 在窗口处理函数中增加绘图代码，增加文本输出内容，分别使用TextOut函数、DrawText函数输出文本。使用DrawText绘制的文本在矩形范围内居中显示，先将矩形绘制出来作为参照。注意函数中字符串输出长度、对齐参数的设置。效果参照课本 2. 修改上题代码，将字体颜色、背景颜色分别修改，设置背景颜色透明，将两行字体重叠显示。参考效果如下：      1. 创建新的字体，设置字体大小为30高，15宽，字体样式为隶书，计算合适位置输出多行文本，避免重叠。参考效果如下：    思考题 设计三种（或n种）文本的样式（例如，1红色20号黑体；2黑色26号宋体；3蓝色30号隶属），将一串文本显示在屏幕上，通过鼠标点击次数切换三种不同字体效果。 | |
| **6、数据记录和计算** | |
| **（代码编写思路及关键代码）**  1、第一题代码：  LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)  {  HDC hdc;  PAINTSTRUCT ps;  TCHAR str\_t[100] = L"使用TextOut输出文本";  TCHAR str\_d[100] = L"使用DrawText输出文本";  int len\_t = wcslen(str\_t);  int len\_d = wcslen(str\_t);  RECT rt = { 100, 100, 400, 200 };  switch (message)  {  case WM\_COMMAND:  {  int wmId = LOWORD(wParam);  // 分析菜单选择:  switch (wmId)  {  case IDM\_ABOUT:  DialogBox(hInst, MAKEINTRESOURCE(IDD\_ABOUTBOX), hWnd, About);  break;  case IDM\_EXIT:  DestroyWindow(hWnd);  break;  default:  return DefWindowProc(hWnd, message, wParam, lParam);  }  }  break;  case WM\_PAINT:  {  hdc = BeginPaint(hWnd, &ps);  TextOut(hdc, 10, 10, str\_t, len\_t);  Rectangle(hdc, rt.left, rt.top, rt.right, rt.bottom);  DrawText(hdc, str\_d, -1, &rt, DT\_SINGLELINE | DT\_CENTER | DT\_VCENTER);  // TextOut(hdc, 50, 100, lpStr, len);  EndPaint(hWnd, &ps);  }  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  default:  return DefWindowProc(hWnd, message, wParam, lParam);  }  return 0;  }  2、第二题代码：  LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)  {  HDC hdc;  PAINTSTRUCT ps;  TCHAR str1[100] = L"红色字体颜色，绿色背景的文本";  TCHAR str2[100] = L"被遮盖的文本，背景透明";  TCHAR str3[100] = L"显示在前面的文本，背景透明";  int len1 = wcslen(str1);  int len2 = wcslen(str2);  int len3 = wcslen(str3);  switch (message)  {  case WM\_COMMAND:  {  int wmId = LOWORD(wParam);  // 分析菜单选择:  switch (wmId)  {  case IDM\_ABOUT:  DialogBox(hInst, MAKEINTRESOURCE(IDD\_ABOUTBOX), hWnd, About);  break;  case IDM\_EXIT:  DestroyWindow(hWnd);  break;  default:  return DefWindowProc(hWnd, message, wParam, lParam);  }  }  break;  case WM\_PAINT:  {  hdc = BeginPaint(hWnd, &ps);  SetTextColor(hdc, RGB(255, 0, 0));  SetBkColor(hdc, RGB(0, 255, 0));  TextOut(hdc, 20, 20, str1, len1);  SetBkMode(hdc, TRANSPARENT);  TextOut(hdc, 20, 60, str2, len2);  TextOut(hdc, 20, 66, str3, len3);  EndPaint(hWnd, &ps);  }  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  default:  return DefWindowProc(hWnd, message, wParam, lParam);  }  return 0;  }  3、第三题代码：  LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)  {  HDC hdc;  PAINTSTRUCT ps;  TCHAR str[30] = L"Hello, 世界你好";  int len = wcslen(str);  static HFONT hFont;  static TEXTMETRIC tm;  SIZE size;  switch (message)  {  case WM\_COMMAND:  {  int wmId = LOWORD(wParam);  // 分析菜单选择:  switch (wmId)  {  case IDM\_ABOUT:  DialogBox(hInst, MAKEINTRESOURCE(IDD\_ABOUTBOX), hWnd, About);  break;  case IDM\_EXIT:  DestroyWindow(hWnd);  break;  default:  return DefWindowProc(hWnd, message, wParam, lParam);  }  }  break;  case WM\_CREATE:  hdc = GetDC(hWnd);  hFont = CreateFont(30, 15, 0, 0, FW\_BOLD, 0, 0, 0, GB2312\_CHARSET,  0, 0, 0, 0, L"隶书");  SelectObject(hdc, hFont);  GetTextMetrics(hdc, &tm);  ReleaseDC(hWnd, hdc);  return 0;  case WM\_PAINT:  {  hdc = BeginPaint(hWnd, &ps);  SetTextColor(hdc, RGB(255, 0, 0));  TextOut(hdc, 20, 30, str, len);  TextOut(hdc, 20, 30 + tm.tmHeight + tm.tmExternalLeading, str, len);  GetTextExtentPoint32(hdc, str, len, &size);  TextOut(hdc, 20 + size.cx, 30, str, len);  EndPaint(hWnd, &ps);  return 0;  }  break;  case WM\_DESTROY:  DeleteObject(hFont);  PostQuitMessage(0);  break;  default:  return DefWindowProc(hWnd, message, wParam, lParam);  }  return 0;  }  思考题代码：  LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)  {  HDC hdc;  PAINTSTRUCT ps;  HFONT hFont = { 0 };  TCHAR str[100] = L"测试文本";  int len = wcslen(str);  int red = 0;  int green = 0;  int blue = 0;  HBRUSH hBrush;  switch (message)  {  case WM\_COMMAND:  {  int wmId = LOWORD(wParam);  // 分析菜单选择:  switch (wmId)  {  case IDM\_ABOUT:  DialogBox(hInst, MAKEINTRESOURCE(IDD\_ABOUTBOX), hWnd, About);  break;  case IDM\_EXIT:  DestroyWindow(hWnd);  break;  default:  return DefWindowProc(hWnd, message, wParam, lParam);  }  }  break;  case WM\_PAINT:  {  }  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  case WM\_LBUTTONDOWN://  hdc = GetDC(hWnd);//  hBrush = (HBRUSH)GetStockObject(WHITE\_BRUSH);//  SelectObject(hdc, hBrush);//  Rectangle(hdc, WINDOW\_POSITON\_X - 1, WINDOW\_POSITON\_Y - 1, WINDOW\_WIDTH, WINDOW\_HEIGHT);//  switch (text\_style\_index)//  {  case 0://  red = 255, green = 0, blue = 0;//  hFont = CreateFont(20, 0, 0, 0, 0, 0, 0, 0,//  GB2312\_CHARSET, 0, 0, 0, 0, L"黑体");//  break;//  case 1://  red = 0, green = 0, blue = 0;//  hFont = CreateFont(26, 0, 0, 0, 0, 0, 0, 0,//  GB2312\_CHARSET, 0, 0, 0, 0, L"宋体");//  break;//  case 2://  red = 0, green = 0, blue = 255;//  hFont = CreateFont(30, 0, 0, 0, 0, 0, 0, 0,//  GB2312\_CHARSET, 0, 0, 0, 0, L"隶书");//  break;//  }  DeleteObject(hBrush);//  SelectObject(hdc, hFont);//  SetTextColor(hdc, RGB(red, green, blue));//  TextOut(hdc, 20, 20, str, len);//  text\_style\_index++;//  text\_style\_index %= 3;//  DeleteObject(hFont);//  ReleaseDC(hWnd, hdc);//  break;//  default:  return DefWindowProc(hWnd, message, wParam, lParam);  }  return 0;  } | |
| **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）**   1. 第一题效果图：      1. 第二题效果图：      1. 第三题效果图：     思考题效果图：  刚打开状态：    鼠标点击一次后的状态：    鼠标点击第二次后的状态：    鼠标点击第三次后的状态：    鼠标点击第四次后的状态：    从此处开始重复出现3种状态。 | |
| **8、备注或说明**  **）** | |
|  | |