

GG1003

Windows编程

胡学萱

李俊琴

实验5-2 Windows鼠标实验

余庆祥

2220631136

BDP

2022/11/4

**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5-2 Windows鼠标实验 |
| **2、实验日期** | **2022/11/4** |
| **3、实验目的** | |
| 1．理解鼠标消息中参数的含义，掌握鼠标消息的使用。  2．理解鼠标消息在绘图操作中与WM\_PAINT消息的配合使用。  3．对实验内容中给出的题目，运用所学知识，按照分析思考、代码编写、调试运行的解题步骤，在VS环境下完成最终效果。 | |
| **4、实验环境** | |
| 1、vs2019 | |
| **5、实验的步骤和方法** | |
| 1. 在窗口中显示多个矩形，响应鼠标消息，当点击某个矩形时，做出相应的响应，如“ClickRect.exe”所示。 2. 响应鼠标消息，在窗口中点击并拖动鼠标画矩形；若同时按下Shift，则画椭圆；如“Mouse-Draw.exe”所示。   **思考题：**  现在，我们可以将鼠标按下、移动、释放结合起来，再加上双缓冲，实现很多有趣的效果了。思考鼠标拖拽效果，如“PickRect.exe”该如何实现？  **请及时复习面向对象编程内容，准备后续的框架学习** | |
| **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)  {  static HDC hdc;  static PAINTSTRUCT ps;  static RECT rt;  static HBRUSH hBrush\_red, hBrush\_while, hOldBrush;  static bool one = false, two = false;  int x\_pos, y\_pos;  switch (message)  {  case WM\_CREATE:  GetClientRect(hwnd, &rt);  hBrush\_red = CreateSolidBrush(RGB(255, 0, 0));  hBrush\_while = CreateSolidBrush(RGB(255, 255, 255));  break;  case WM\_LBUTTONDOWN:  x\_pos = LOWORD(lParam);  y\_pos = HIWORD(lParam);  if (x\_pos >= 100 && x\_pos <= 200 && y\_pos >= 100 && y\_pos <= 200)  {  one = !one;  }  else if (x\_pos >= 400 && x\_pos <= 500 && y\_pos >= 100 && y\_pos <= 400) {  two = !two;  }  else {  one = false;  two = false;  }    InvalidateRect(hwnd, &rt, true);  break;  case WM\_PAINT:  hdc = BeginPaint(hwnd, &ps);  if (one)  {  hOldBrush = (HBRUSH)SelectObject(hdc, hBrush\_red);  }  else {  hOldBrush = (HBRUSH)SelectObject(hdc, hBrush\_while);  }  Rectangle(hdc, 100, 100, 200, 200);  if (two)  {  SelectObject(hdc, hBrush\_red);  }  else {  SelectObject(hdc, hBrush\_while);  }  Rectangle(hdc, 400, 100, 500, 400);  EndPaint(hwnd, &ps);  break;  case WM\_DESTROY:  SelectObject(hdc, hOldBrush);  DeleteObject(hBrush\_red);  DeleteObject(hBrush\_while);  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)  {  static HDC hdc;  static PAINTSTRUCT ps;  static RECT rt, rtDraw;  static bool shift\_down = false;  switch (message)  {  case WM\_CREATE:  GetClientRect(hwnd, &rt);  break;  case WM\_MOUSEMOVE:  if (!(wParam & MK\_LBUTTON))  return 0;  rtDraw.right = LOWORD(lParam);  rtDraw.bottom = HIWORD(lParam);  InvalidateRect(hwnd, &rt, true);  break;  case WM\_LBUTTONDOWN:  rtDraw.left = LOWORD(lParam);  rtDraw.top = HIWORD(lParam);  shift\_down = false;  if (GetKeyState(VK\_SHIFT) & 0x8000)  {  shift\_down = true;  }  break;  case WM\_PAINT:  hdc = BeginPaint(hwnd, &ps);  if (shift\_down)  {  Ellipse(hdc, rtDraw.left, rtDraw.top, rtDraw.right, rtDraw.bottom);  }  else {  Rectangle(hdc, rtDraw.left, rtDraw.top, rtDraw.right, rtDraw.bottom);  }  EndPaint(hwnd, &ps);  break;  case WM\_DESTROY:  PostQuitMessage(0);  return 0;  }  return DefWindowProc(hwnd, message, wParam, lParam);  }  思考题代码：  LRESULT CALLBACK WndProc(HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)  {  static HDC hdc;  static PAINTSTRUCT ps;  static RECT rt, sm\_rt;  static float rect\_x = 50, rect\_y = 50; // 可移动的矩形的位置  float pos\_x, pos\_y; // 鼠标位置  static float diff\_x, diff\_y; // 鼠标的偏移量  static bool rect\_can\_move = false, rect\_in\_RECT = false; //  switch (message)  {  case WM\_CREATE:  GetClientRect(hwnd, &rt);  break;  case WM\_MOUSEMOVE:  if (!(wParam & MK\_LBUTTON))  return 0;  pos\_x = LOWORD(lParam);  pos\_y = HIWORD(lParam);  // 如果小矩形允许移动且不在大矩形内  if (rect\_can\_move && !rect\_in\_RECT)  {  rect\_x = pos\_x - diff\_x;  rect\_y = pos\_y - diff\_y;  }  InvalidateRect(hwnd, &rt, true);  break;  case WM\_LBUTTONDOWN:  pos\_x = LOWORD(lParam);  pos\_y = HIWORD(lParam);  // 如果鼠标在可移动矩形内  if (pos\_x >= 50 && pos\_x <= 100 && pos\_y >= 50 && pos\_y <= 100)  {  // 获取偏移量  diff\_x = pos\_x - 50;  diff\_y = pos\_y - 50;  rect\_can\_move = true;  }  break;  case WM\_LBUTTONUP:  rect\_can\_move = false;  // 当可移动矩形在大矩形之外时  if (rect\_x < 200 || rect\_x > 400 - 50 || rect\_y < 200 || rect\_y > 400 - 50)  {  rect\_x = 50;  rect\_y = 50;  }  else {  rect\_in\_RECT = true;  }  InvalidateRect(hwnd, &rt, true);  break;  case WM\_PAINT:  hdc = BeginPaint(hwnd, &ps);  // 大矩形  Rectangle(hdc, 200, 200, 400, 400);  // 小矩形  Rectangle(hdc, rect\_x, rect\_y, rect\_x + 50, rect\_y + 50);  if (rect\_in\_RECT)  {  sm\_rt.left = rect\_x;  sm\_rt.top = rect\_y;  sm\_rt.right = rect\_x + 50;  sm\_rt.bottom = rect\_y + 50;  FillRect(hdc, &sm\_rt, (HBRUSH)CreateSolidBrush(RGB(255, 0, 0)));  }  EndPaint(hwnd, &ps);  break;  case WM\_DESTROY:  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）**  1、第一题效果图：  在“第一题录屏.mp4”   1. 第二题效果图：   在“第二题录屏.mp4”  思考题效果图：  在“思考题录屏.mp4” | |
| **8、备注或说明**  **）** | |
|  | |