Win API

1. 다음과 같이 메뉴를 만드시오

[메뉴 리소스]

FILE

MENU1

MENU2

EXIT

|  |
| --- |
| MENU1 – 메시지박스로 현재 시간 출력  MENU2 – 화면에 현재 시간 출력  (현재시간은 메뉴 선택시 시간으로 내용이 실시간으로 변하지 않아도 됨)  EXIT – 프로그램을 종료 할것인지 묻는 YESNO 메시지박스 출력 후 YES 시 종료 |
| 답안  LRESULT CALLBACK WndProc(HWND hWnd, UINT iMsg, WPARAM wParam, LPARAM lParam) {  HDC hDC;  PAINTSTRUCT ps;    time\_t mytime;  static HANDLE hTimer;  static TCHAR str[26];  HBRUSH hMyBrush, hOldBrush;  HPEN hMyPen, hOldPen;  switch (iMsg) {  case WM\_COMMAND:  switch (LOWORD(wParam)) {  case ID\_FILE\_MENU1:  time(&mytime);  ctime\_s(str, sizeof(str), &mytime);  if (MessageBox(hWnd, str, TEXT("Time"), MB\_OK) == IDOK) return 0;  break;  case ID\_FILE\_MENU2:  time(&mytime);  ctime\_s(str, sizeof(str), &mytime);  InvalidateRect(hWnd, NULL, TRUE);  break;  case ID\_FILE\_EXIT:  if (MessageBox(hWnd, TEXT("Are You Want ?"), TEXT("EXIT"), MB\_YESNO) == IDYES) break;  else return 0;  break;  }  case WM\_PAINT:  hDC = BeginPaint(hWnd, &ps);  TextOut(hDC, 100, 100, str, \_tcslen(str) - 1);  EndPaint(hWnd, &ps);  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  }  return DefWindowProc(hWnd, iMsg, wParam, lParam);  } |
| 결과(화면 캡쳐) |
|  |

1. 다음과 같이 도형을 출력하시오

[도형 출력 파트, 펜과 브러시]

|  |
| --- |
|  |
| 답안  LRESULT CALLBACK WndProc(HWND hWnd, UINT iMsg, WPARAM wParam, LPARAM lParam)  {  HDC hDC;  PAINTSTRUCT ps;  HBRUSH hMyBrush, hOldBrush;  HPEN hMyPen, hOldPen;  switch (iMsg)  {  case WM\_PAINT:  hDC = BeginPaint(hWnd, &ps);  hMyPen = CreatePen(PS\_SOLID, 5, 0x0);  hOldPen = reinterpret\_cast<HPEN>(SelectObject(hDC, hMyPen));  hMyBrush = CreateSolidBrush(RGB(0, 170, 0));  hOldBrush = reinterpret\_cast<HBRUSH>(SelectObject(hDC, hMyBrush));  Rectangle(hDC, 100, 100, 1100, 500);  SelectObject(hDC, hOldBrush);  hMyPen = CreatePen(PS\_DASH, 1, 0x0);  hOldPen = reinterpret\_cast<HPEN>(SelectObject(hDC, hMyPen));  hMyBrush = CreateSolidBrush(RGB(0, 0, 170));  hOldBrush = reinterpret\_cast<HBRUSH>(SelectObject(hDC, hMyBrush));  Rectangle(hDC, 150, 150, 300, 300);  SelectObject(hDC, hOldBrush);  hMyPen = CreatePen(PS\_DOT, 1, 0x0);  hOldPen = reinterpret\_cast<HPEN>(SelectObject(hDC, hMyPen));  hMyBrush = CreateSolidBrush(RGB(0, 0, 170));  hOldBrush = reinterpret\_cast<HBRUSH>(SelectObject(hDC, hMyBrush));  Rectangle(hDC, 400, 150, 550, 300);  SelectObject(hDC, hOldBrush);  hMyPen = CreatePen(PS\_DASHDOT, 1, 0x0);  hOldPen = reinterpret\_cast<HPEN>(SelectObject(hDC, hMyPen));  hMyBrush = CreateSolidBrush(RGB(0, 0, 170));  hOldBrush = reinterpret\_cast<HBRUSH>(SelectObject(hDC, hMyBrush));  Rectangle(hDC, 650, 150, 800, 300);  SelectObject(hDC, hOldBrush);  hMyPen = CreatePen(PS\_DASHDOTDOT, 1, 0x0);  hOldPen = reinterpret\_cast<HPEN>(SelectObject(hDC, hMyPen));  hMyBrush = CreateSolidBrush(RGB(0, 0, 170));  hOldBrush = reinterpret\_cast<HBRUSH>(SelectObject(hDC, hMyBrush));  Rectangle(hDC, 900, 150, 1050, 300);  SelectObject(hDC, hOldBrush);  hMyPen = CreatePen(PS\_SOLID, 3, 0x0);  hOldPen = reinterpret\_cast<HPEN>(SelectObject(hDC, hMyPen));  hMyBrush = reinterpret\_cast<HBRUSH>(GetStockObject(GRAY\_BRUSH));  hOldBrush = reinterpret\_cast<HBRUSH>(SelectObject(hDC, hMyBrush));  Ellipse(hDC, 300, 400, 500, 600);  Ellipse(hDC, 700, 400, 900, 600);  SelectObject(hDC, hOldBrush);  hMyPen = CreatePen(PS\_SOLID, 5, 0x0);  hOldPen = reinterpret\_cast<HPEN>(SelectObject(hDC, hMyPen));  MoveToEx(hDC, 250, 450, nullptr);  LineTo(hDC, 300, 360);  LineTo(hDC, 500, 360);  LineTo(hDC, 550, 450);  MoveToEx(hDC, 650, 450, nullptr);  LineTo(hDC, 700, 360);  LineTo(hDC, 900, 360);  LineTo(hDC, 950, 450);  DeleteObject(hMyPen);  DeleteObject(hMyBrush);  EndPaint(hWnd, &ps);  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  }  return DefWindowProc(hWnd, iMsg, wParam, lParam);  } |
| 결과 |

1. 랜덤한 방향(360도)으로 움직이고 창 가장자리에 도달하면 반대 방향으로 튕겨져 움직이는 다음과 같은 도형을 출력하시오

|  |
| --- |
| 답안 |
| 결과  static INT x = 100, y = 100;  static INT rnd;  LRESULT CALLBACK WndProc(HWND hWnd, UINT iMsg, WPARAM wParam, LPARAM lParam)  {  HDC hDC;  PAINTSTRUCT ps;  HBRUSH hMyBrush, hOldBrush;  HPEN hMyPen, hOldPen;  static RECT rt;  static HANDLE hTimer;  INT speed = 3;  switch (iMsg)  {  case WM\_CREATE:  hTimer = (HANDLE)SetTimer(hWnd, 1, 0, NULL);  GetClientRect(hWnd, &rt);  rnd = rand() % 4;  SendMessage(hWnd, WM\_TIMER, 1, 0);  break;  case WM\_TIMER:  switch (rnd) {  case 0:  x += speed;  y += speed;  break;  case 1:  x -= speed;  y += speed;  break;  case 2:  x += speed;  y -= speed;  break;  case 3:  x -= speed;  y -= speed;  break;  }  InvalidateRect(hWnd, NULL, TRUE);  break;  case WM\_PAINT:  hDC = BeginPaint(hWnd, &ps);  hMyPen = CreatePen(PS\_SOLID, 3, RGB(255, 127, 0));  hOldPen = reinterpret\_cast<HPEN>(SelectObject(hDC, hMyPen));  hMyBrush = CreateHatchBrush(HS\_DIAGCROSS, RGB(0, 170, 0));  hOldBrush = reinterpret\_cast<HBRUSH>(SelectObject(hDC, hMyBrush));  Ellipse(hDC, x, y, x + 100, y + 100);  if (x + 100 >= rt.right) {  if (rnd == 2) rnd = 3;  else if (rnd == 0) rnd = 1;  }  else if (y + 100 >= rt.bottom) {  if (rnd == 0) rnd = 2;  else if (rnd == 1) rnd = 3;  }  else if (x <= rt.left) {  if (rnd == 3) rnd = 2;  else if (rnd == 1) rnd = 0;  }  else if (y <= rt.top) {  if (rnd == 2) rnd = 0;  else if (rnd == 3) rnd = 1;  }  Ellipse(hDC, x, y, x + 100, y + 100);  SelectObject(hDC, hOldBrush);  EndPaint(hWnd, &ps);  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  }  return DefWindowProc(hWnd, iMsg, wParam, lParam);  } |

전역변수로 x,y 그리고 랜덤으로 이동할 때 값을 썼습니다

남은 시간은 2일차 과제 심화문제 풀어봅시다