Win API

1. 프로그램을 실행하면 다음과 같은 스태틱과 버튼이 존재하고 ‘게임시작’ 버튼을 누르면 스태틱과 버튼이 모두 사라지고 2일차 과제 4번 문제가 나오게 하고 ‘게임종료’버튼을 누르면 프로그램이 종료되게 하시오

버튼을 사라지게 할 때에는 ShowWindow함수를 사용하면 된다

[푸시버튼, 스태틱]

|  |
| --- |
| 재밌는 걔임  게임종료  게임시작 |
| 답안  #define STARTCODE 100  #define EXITCODE 101  LRESULT CALLBACK WndProc(HWND hWnd, UINT iMessage, WPARAM wParam, LPARAM lParam)  {  HDC hDC;  PAINTSTRUCT ps;  static INT x = 0;  static INT y = 0;  static BOOL Draw = FALSE;  static HWND title, startbt, exitbt;    switch (iMessage) {  case WM\_CREATE:  title = CreateWindow("static", "재밌는 걔임", WS\_CHILD | WS\_VISIBLE | SS\_SIMPLE, 200, 200, 100, 25, hWnd, (HMENU)-1, GetModuleHandle(NULL), NULL);  startbt = CreateWindow(TEXT("button"), TEXT("게임 시작"), WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,  200, 300, 100, 25, hWnd, (HMENU)STARTCODE, GetModuleHandle(NULL), NULL);  exitbt = CreateWindow(TEXT("button"), TEXT("게임 종료"), WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,  200, 350, 100, 25, hWnd, (HMENU)EXITCODE, GetModuleHandle(NULL), NULL);  break;  case WM\_COMMAND:  switch (LOWORD(wParam)) {  case STARTCODE:  ShowWindow(title, SW\_HIDE);  ShowWindow(startbt, SW\_HIDE);  ShowWindow(exitbt, SW\_HIDE);  SendMessage(hWnd, WM\_PAINT, STARTCODE, 0);  Draw = TRUE;  InvalidateRect(hWnd, NULL, TRUE);  break;  case EXITCODE:  PostQuitMessage(0);  break;  }  case WM\_PAINT:  hDC = BeginPaint(hWnd, &ps);  if (Draw) {  SetTextAlign(hDC, TA\_LEFT | TA\_TOP);  TextOut(hDC, 200, 200, TEXT("B"), 1);  TextOut(hDC, x, y, TEXT("A"), 1);  if (x == 200 && y == 200) SendMessage(hWnd, WM\_DESTROY, 0, 0);  }  EndPaint(hWnd, &ps);  break;  case WM\_KEYDOWN:  switch (wParam) {  case VK\_LEFT:  x -= 10;  if (x < 0) x = 0;  break;  case VK\_RIGHT:  x += 10;  break;  case VK\_UP:  y -= 10;  if (y < 0) y = 0;  break;  case VK\_DOWN:  y += 10;  break;  }  InvalidateRect(hWnd, NULL, TRUE);  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  }  return(DefWindowProc(hWnd, iMessage, wParam, lParam));  } |
| 결과(화면 캡쳐) |
|  |

1. 다음과 같은 리스트 박스를 누르면 화면에 출력되는 비트맵이 달라지도록 하시오

비트맵은 4일차 과제에 사용한 Mario.bmp를 활용한다

리스트 박스의 항목은 자유롭게 구성한다

[비트맵 출력, 리스트 박스]

|  |
| --- |
| Brick1  Brick2  Brick3  Goomba  Mario |
| 답안  #define ID\_LIST 100  char Choice[][15] = { "Brick1", "Brick2", "Brick3", "Goomba", "Mario" };  char str[15];  HWND hList;  LRESULT CALLBACK WndProc(HWND hWnd, UINT iMsg, WPARAM wParam, LPARAM lParam)  {  HDC hDC, hMem;  PAINTSTRUCT ps;  HBITMAP hMyBitmap, hOldBitmap;  static INT i = 0;  switch (iMsg)  {  case WM\_CREATE:  hList = CreateWindow("listbox", NULL, WS\_CHILD | WS\_VISIBLE | WS\_BORDER |  LBS\_NOTIFY, 10, 10, 100, 100, hWnd, (HMENU)ID\_LIST, GetModuleHandle(NULL), NULL);  for (i = 0; i < 5; i++)  SendMessage(hList, LB\_ADDSTRING, 0, (LPARAM)Choice[i]);  break;  case WM\_COMMAND:  switch (LOWORD(wParam)) {  case ID\_LIST:  switch (HIWORD(wParam)) {  case LBN\_SELCHANGE:  i = SendMessage(hList, LB\_GETCURSEL, 0, 0);  SendMessage(hList, LB\_GETTEXT, i, (LPARAM)str);  InvalidateRect(hWnd, NULL, TRUE);  SetWindowText(hWnd, str);  break;  }  }  break;  case WM\_PAINT:  hDC = BeginPaint(hWnd, &ps);  hMem = CreateCompatibleDC(hDC);  hMyBitmap = LoadBitmap(GetModuleHandle(NULL), MAKEINTRESOURCE(IDB\_BITMAP2));  hOldBitmap = reinterpret\_cast<HBITMAP>(SelectObject(hMem, hMyBitmap));  //바닥 0, 0, 16, 16  //부술 수 있는 벽돌 16, 0, 16, 16  //캐릭터, 적  //StretchBlt(hDC, 130, 144, 12, 16, hMem, 274, 64, 12, 16, SRCCOPY);  //StretchBlt(hDC, 306, 144, 16, 16, hMem, 176, 80, 16, 16, SRCCOPY);    switch (i) {  case 0:  StretchBlt(hDC, 200, 10, 100, 100, hMem, 0, 0, 16, 16, SRCCOPY);  break;  case 1:  StretchBlt(hDC, 200, 10, 100, 100, hMem, 16, 0, 16, 16, SRCCOPY);  break;  case 2:  StretchBlt(hDC, 200, 10, 100, 100, hMem, 48, 0, 16, 16, SRCCOPY);  break;  case 3:  StretchBlt(hDC, 200, 10, 100, 100, hMem, 176, 80, 16, 16, SRCCOPY);  break;  case 4:  StretchBlt(hDC, 200, 10, 100, 100, hMem, 274, 64, 12, 16, SRCCOPY);  break;  }  SelectObject(hMem, hOldBitmap);  DeleteObject(hOldBitmap);  DeleteDC(hMem);  EndPaint(hWnd, &ps);;  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  }  return DefWindowProc(hWnd, iMsg, wParam, lParam);  } |
| 결과(화면 캡쳐) |
|  |

1. 다음과 같은 라디오 박스를 누르면 화면에 출력되는 마리오의 상태가 달라지도록 하시오 2 , 224 528

비트맵은 4일차 과제에 사용한 Mario.bmp를 활용한다

각각 걷기,점프,앉기 모양이다

타일맵 상에 모양이 여러가지 존재할 경우 그중 한가지 선택하면 된다

[비트맵 출력, 라디오 박스]

|  |
| --- |
| STATE  □ WALK  ▣ JUMP  □ SITDOWN |
| 답안  #define ID\_R 100  #define ID\_R1 101  #define ID\_R2 102  #define ID\_R3 103  HWND sit1, sit2, sit3;  INT shape;  LRESULT CALLBACK WndProc(HWND hWnd, UINT iMsg, WPARAM wParam, LPARAM lParam)  {  HDC hDC, hMem;  PAINTSTRUCT ps;  HBITMAP hMyBitmap, hOldBitmap;  static INT i = 0;  switch (iMsg)  {  case WM\_CREATE:  CreateWindow("button", "STATE", WS\_CHILD | WS\_VISIBLE | BS\_GROUPBOX,  5, 5, 120, 110, hWnd, (HMENU)ID\_R, GetModuleHandle(NULL), NULL);  sit1 = CreateWindow("button", "WALK", WS\_CHILD | WS\_VISIBLE |  BS\_AUTORADIOBUTTON, 10, 25, 100, 20, hWnd, (HMENU)ID\_R1, GetModuleHandle(NULL), NULL);  sit2 = CreateWindow("button", "JUMP", WS\_CHILD | WS\_VISIBLE |  BS\_AUTORADIOBUTTON, 10, 55, 100, 20, hWnd, (HMENU)ID\_R2, GetModuleHandle(NULL), NULL);  sit3 = CreateWindow("button", "SITDOWN", WS\_CHILD | WS\_VISIBLE |  BS\_AUTORADIOBUTTON, 10, 85, 100, 20, hWnd, (HMENU)ID\_R3, GetModuleHandle(NULL), NULL);  CheckRadioButton(hWnd, ID\_R1, ID\_R3, ID\_R1);  break;  case WM\_COMMAND:  if (HIWORD(wParam) == BN\_CLICKED) {  switch (LOWORD(wParam)) {  case ID\_R1:  shape = 0;  break;  case ID\_R2:  shape = 1;  break;  case ID\_R3:  shape = 2;  break;  }  InvalidateRect(hWnd, NULL, TRUE);  }  break;  case WM\_PAINT:  hDC = BeginPaint(hWnd, &ps);  hMem = CreateCompatibleDC(hDC);  hMyBitmap = LoadBitmap(GetModuleHandle(NULL), MAKEINTRESOURCE(IDB\_BITMAP2));  hOldBitmap = reinterpret\_cast<HBITMAP>(SelectObject(hMem, hMyBitmap));  switch (shape) {  case 0:  StretchBlt(hDC, 200, 10, 100, 100, hMem, 192, 64, 15, 16, SRCCOPY);  break;  case 1:  StretchBlt(hDC, 200, 10, 100, 100, hMem, 240, 64, 16, 16, SRCCOPY);  break;  case 2:  StretchBlt(hDC, 200, 10, 100, 100, hMem, 306, 64, 12, 16, SRCCOPY);  break;  }  SelectObject(hMem, hOldBitmap);  DeleteObject(hOldBitmap);  DeleteDC(hMem);  EndPaint(hWnd, &ps);;  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  }  return DefWindowProc(hWnd, iMsg, wParam, lParam);  } |
| 결과(화면 캡쳐) |
|  |

1. 에디트에 문자열을 입력하면 해당 문자열이 화면상에서 나타나고 방향키를 누르면 문자열이 움직이게 하시오

에디트가 움직이는 것이 아니다

[문자열 출력, 키보드 입력, 에디트]

|  |
| --- |
| 닉네임  닉네임 |
| 답안  #define ID\_MOVE 100  HWND hEdit;  static INT x = 100, y = 100;  char str[128];  INT length;  LRESULT CALLBACK WndProc(HWND hWnd, UINT iMsg, WPARAM wParam, LPARAM lParam)  {  HDC hDC;  PAINTSTRUCT ps;  switch (iMsg)  {  case WM\_CREATE:  hEdit = CreateWindow("edit", NULL, WS\_CHILD | WS\_VISIBLE | WS\_BORDER | ES\_AUTOHSCROLL,  10, 10, 200, 25, hWnd, (HMENU)ID\_MOVE, GetModuleHandle(NULL), NULL);  break;  case WM\_COMMAND:  switch (LOWORD(wParam)) {  case ID\_MOVE:  switch (HIWORD(wParam)) {  case EN\_CHANGE:  GetWindowText(hEdit, str, 128);  length = strlen(str);  SetWindowText(hWnd, str);  InvalidateRect(hWnd, NULL, TRUE);  }  }  break;  case WM\_LBUTTONDOWN:  SetFocus(hWnd);  break;  case WM\_PAINT:  hDC = BeginPaint(hWnd, &ps);  TextOut(hDC, x, y, str, length);  EndPaint(hWnd, &ps);  break;  case WM\_KEYDOWN:  switch (wParam) {  case VK\_LEFT:  x -= 10;  if (x < 0) x = 0;  break;  case VK\_RIGHT:  x += 10;  break;  case VK\_UP:  y -= 10;  if (y < 0) y = 0;  break;  case VK\_DOWN:  y += 10;  break;  }  InvalidateRect(hWnd, NULL, TRUE);  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  }  return DefWindowProc(hWnd, iMsg, wParam, lParam);  } |
| 결과(화면 캡쳐) |

1. 수평 스크롤을 스크롤하면 마리오가 헤엄치도록 하시오

스크롤이 스크롤되면 마리오가 헤엄치는 각각의 타일을 복사원의 좌표를 바꾸어 줘 차례로 화면에 출력한다

[비트맵, 스크롤]

|  |
| --- |
|  |
| 답안  #define ID\_MOVE 100  LRESULT CALLBACK WndProc(HWND hWnd, UINT iMsg, WPARAM wParam, LPARAM lParam)  {  HDC hDC, hMem;  PAINTSTRUCT ps;  HBITMAP hMyBitmap, hOldBitmap;  static HWND hPos;  static INT x = 100, y = 100;  static INT scrpos = 0, cnd = 0;  switch (iMsg)  {  case WM\_CREATE:  hPos = CreateWindow("scrollbar", NULL, WS\_CHILD | WS\_VISIBLE | SBS\_HORZ,  150, 200, 200, 20, hWnd, (HMENU)ID\_MOVE, GetModuleHandle(NULL), NULL);  SetScrollRange(hPos, SB\_CTL, 0, 50, TRUE);  SetScrollPos(hPos, SB\_CTL, 0, TRUE);  break;  case WM\_HSCROLL:  switch (LOWORD(wParam)) {  case SB\_LINELEFT:  scrpos = max(0, scrpos - 1);  break;  case SB\_LINERIGHT:  scrpos = min(100, scrpos + 1);  break;  case SB\_PAGELEFT:  scrpos = max(0, scrpos - 3);  break;  case SB\_PAGERIGHT:  scrpos = min(100, scrpos + 3);  break;  case SB\_THUMBTRACK:  scrpos = HIWORD(wParam);  break;  }  SetScrollPos((HWND)lParam, SB\_CTL, scrpos, TRUE);  InvalidateRect(hWnd, NULL, TRUE);  break;  case WM\_PAINT:  hDC = BeginPaint(hWnd, &ps);  hMem = CreateCompatibleDC(hDC);  hMyBitmap = LoadBitmap(GetModuleHandle(NULL), MAKEINTRESOURCE(IDB\_BITMAP2));  hOldBitmap = reinterpret\_cast<HBITMAP>(SelectObject(hMem, hMyBitmap));  cnd = scrpos % 5;  switch (cnd) {  case 0:  StretchBlt(hDC, 200, 100, 100, 100, hMem, 321, 64, 13, 16, SRCCOPY);  break;  case 1:  StretchBlt(hDC, 200, 100, 100, 100, hMem, 337, 64, 13, 16, SRCCOPY);  break;  case 2:  StretchBlt(hDC, 200, 100, 100, 100, hMem, 353, 64, 13, 16, SRCCOPY);  break;  case 3:  StretchBlt(hDC, 200, 100, 100, 100, hMem, 369, 64, 13, 16, SRCCOPY);  break;  case 4:  StretchBlt(hDC, 200, 100, 100, 100, hMem, 385, 64, 14, 16, SRCCOPY);  break;  }  SelectObject(hMem, hOldBitmap);  DeleteObject(hOldBitmap);  DeleteDC(hMem);  EndPaint(hWnd, &ps);  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  }  return DefWindowProc(hWnd, iMsg, wParam, lParam);  } |
| 결과(화면 캡쳐) |
|  |