|  |  |
| --- | --- |
|  | **윈도우프로그래밍**  **LAB 06** |
| **분반** | 2 |
| **학번** | 1815060 |
| **이름** | 문정현 |

**# 실습 문제 1**

#1 tictactoe.cpp 프로그램 소스 및 설명

#include <afxwin.h>

#define IDC\_BUTTON0 100

#define IDC\_BUTTON1 101

#define IDC\_BUTTON2 102

#define IDC\_BUTTON3 103

#define IDC\_BUTTON4 104

#define IDC\_BUTTON5 105

#define IDC\_BUTTON6 106

#define IDC\_BUTTON7 107

#define IDC\_BUTTON8 108

enum { OFF, ON };

// Declare the application class

class CToggleApp : public CWinApp {

public:

virtual BOOL InitInstance();

};

// Create an instance of the application class

CToggleApp ToggleApp;

// Declare the main window class

class CToggleWindow : public CFrameWnd {

CButton\* button[9];

int status[9];

int turn;

void init();

public:

CToggleWindow();

void HandleButton(int \_ik);

afx\_msg void HandleButton0();

afx\_msg void HandleButton1();

afx\_msg void HandleButton2();

afx\_msg void HandleButton3();

afx\_msg void HandleButton4();

afx\_msg void HandleButton5();

afx\_msg void HandleButton6();

afx\_msg void HandleButton7();

afx\_msg void HandleButton8();

DECLARE\_MESSAGE\_MAP();

// the InitInstance function is called once

// when the application first executes

};

BOOL CToggleApp::InitInstance()

{

m\_pMainWnd = new CToggleWindow();

m\_pMainWnd->ShowWindow(m\_nCmdShow);

m\_pMainWnd->UpdateWindow();

return TRUE;

}

// the message map

BEGIN\_MESSAGE\_MAP(CToggleWindow, CFrameWnd)

ON\_BN\_CLICKED(IDC\_BUTTON0, HandleButton0)

ON\_BN\_CLICKED(IDC\_BUTTON1, HandleButton1)

ON\_BN\_CLICKED(IDC\_BUTTON2, HandleButton2)

ON\_BN\_CLICKED(IDC\_BUTTON3, HandleButton3)

ON\_BN\_CLICKED(IDC\_BUTTON4, HandleButton4)

ON\_BN\_CLICKED(IDC\_BUTTON5, HandleButton5)

ON\_BN\_CLICKED(IDC\_BUTTON6, HandleButton6)

ON\_BN\_CLICKED(IDC\_BUTTON7, HandleButton7)

ON\_BN\_CLICKED(IDC\_BUTTON8, HandleButton8)

END\_MESSAGE\_MAP()

void CToggleWindow::init()

{

turn = 0;

for (int i = 0; i < 9; i++) {

status[i] = OFF;

}

}

// the contructor for the window class

CToggleWindow::CToggleWindow()

{

//Create the window itself

Create(NULL, \_T("Tic-Tac-Toe"),WS\_OVERLAPPEDWINDOW,CRect(0, 0, 460, 480));

init();

// Create button

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

button[3\*i+j] = new CButton();

button[3\*i+j]->Create(\_T(""), WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON, CRect(6+131\*j, 6+131\*i, 136+131\*j, 136+131\*i), this, IDC\_BUTTON0+3\*i+j);

}

}

}

void CToggleWindow::HandleButton0() { HandleButton(0); }

void CToggleWindow::HandleButton1() { HandleButton(1); }

void CToggleWindow::HandleButton2() { HandleButton(2); }

void CToggleWindow::HandleButton3() { HandleButton(3); }

void CToggleWindow::HandleButton4() { HandleButton(4); }

void CToggleWindow::HandleButton5() { HandleButton(5); }

void CToggleWindow::HandleButton6() { HandleButton(6); }

void CToggleWindow::HandleButton7() { HandleButton(7); }

void CToggleWindow::HandleButton8() { HandleButton(8); }

// message handler function for number buttons

void CToggleWindow::HandleButton(int \_ik)

{

if (status[\_ik] == OFF) {

if (turn % 2 == 0) {

button[\_ik]->SetWindowText(\_T("X"));

status[\_ik] = ON;

turn = (turn + 1) % 2;

}

else {

button[\_ik]->SetWindowText(\_T("O"));

status[\_ik] = ON;

turn = (turn + 1) % 2;

}

}

else{MessageBeep(-1);}

}

9개의 버튼에 대한 id를 100~108로 설정하였고 init()에서 id순서대로 버튼을 처음으로 눌렀는지 확인하기 위해 status라는 int 배열에서 OFF에 해당하는 수로 초기화시켰다. (id가 n인 버튼의 정보를 status배열의 n-100번째 인덱스에 저장되어 있다.)

CToggleWindow()에서는 9개의 CButton 포인터 배열을 선언하고 이중 for문을 이용해서 각 버튼 객체를 생성하고 위치를 i행 j열에 배치되도록 하였다. 메세지맵에서는 각 버튼에 대한 메세지 핸들러 함수를 등록시켜 놓은 뒤 메세지 핸들러 함수 내에서는 해당하는 버튼id-100에 대한 매개변수를 전달하고 HandleButton(int \_ik)를 실행시켜 각 버튼에 대한 버튼글자 제어를 한 함수에서 처리할 수 있도록 했다.

HandleButton(int \_ik)에서는 누른 버튼이 처음으로 눌렀는지 status[\_ik]에서 확인하여 OFF이면 해당하는 순서에 맞는 문자(O/X)를 버튼에 표시하게 하고 이미 ON인 경우는 경고음을 내게 했다. 순서를 저장한 변수는 turn으로 0이면 버튼에 X가 표시되게 하고 1이면 O가 표시되게 했다.

#2 실행 화면



