

윈도우프로그래밍 Lab 06

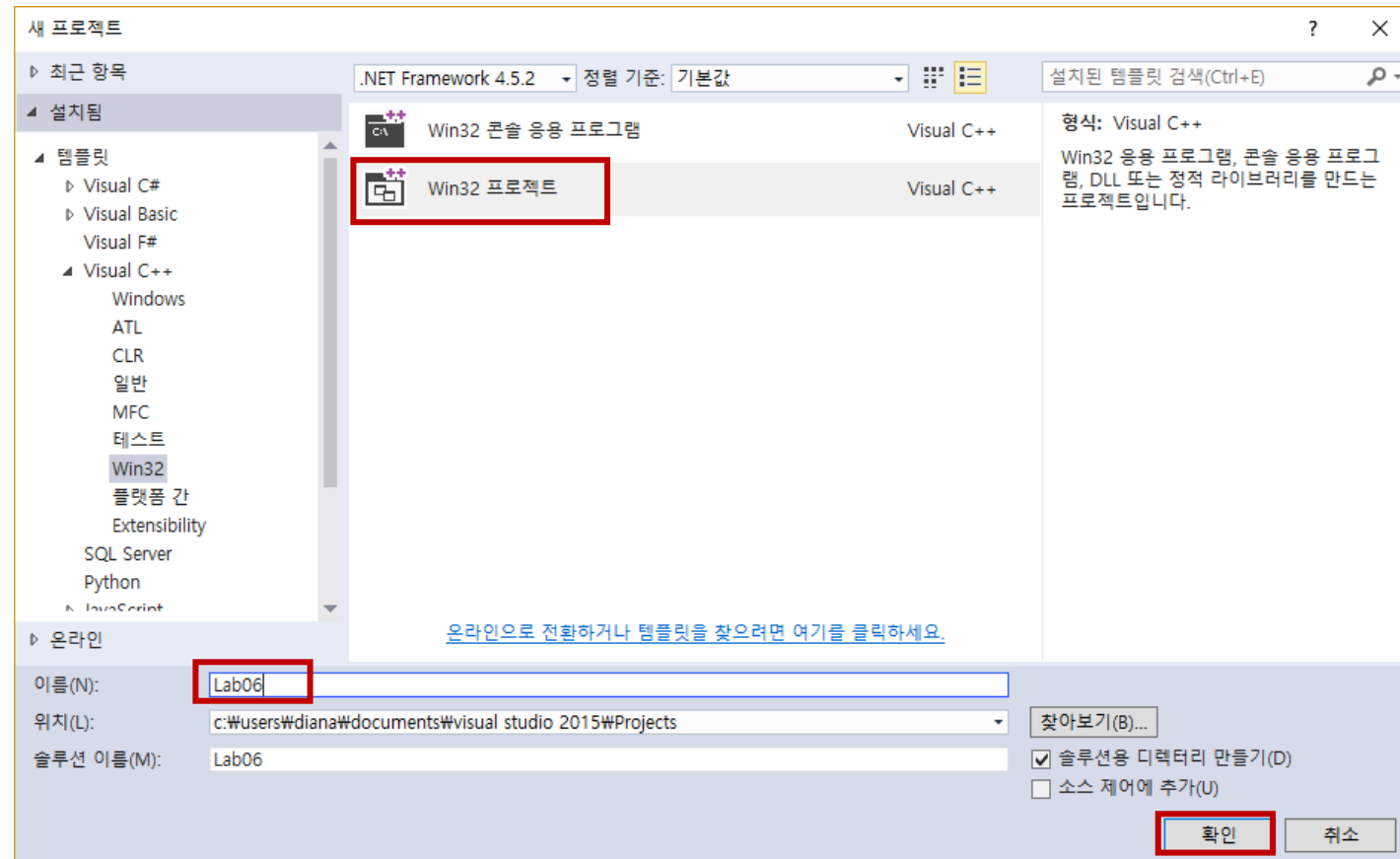
- MFC 프로그래밍: Tic-Tac-Toe 게임 프로그래밍하기 -

김한슬

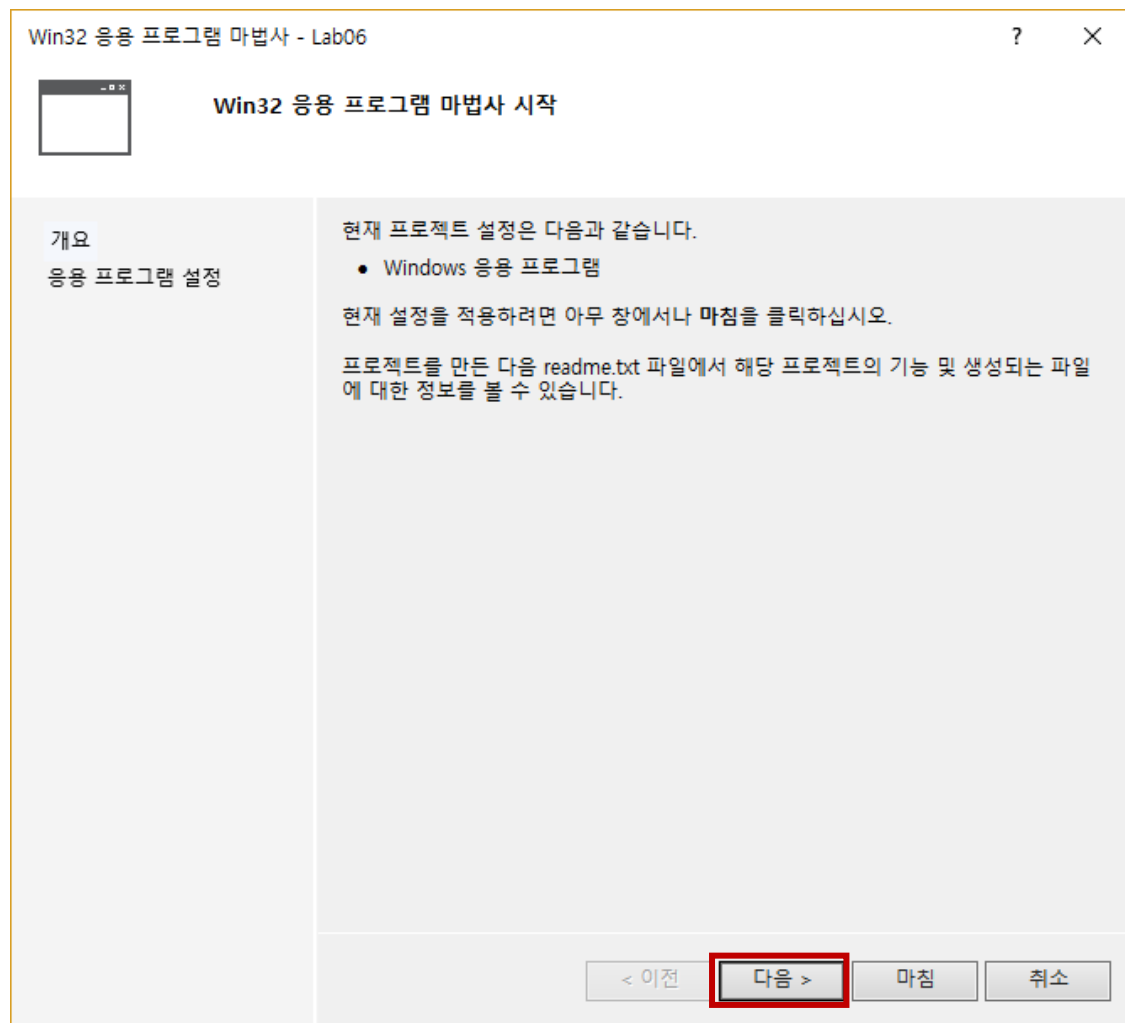
uo3359@sookmyung.ac.kr

win32 프로젝트 생성하기

- [파일] 메뉴 -> [새로 만들기] -> [프로젝트] -> <Win32 프로젝트> -> 프로젝트 이름을 입력 -> <확인>

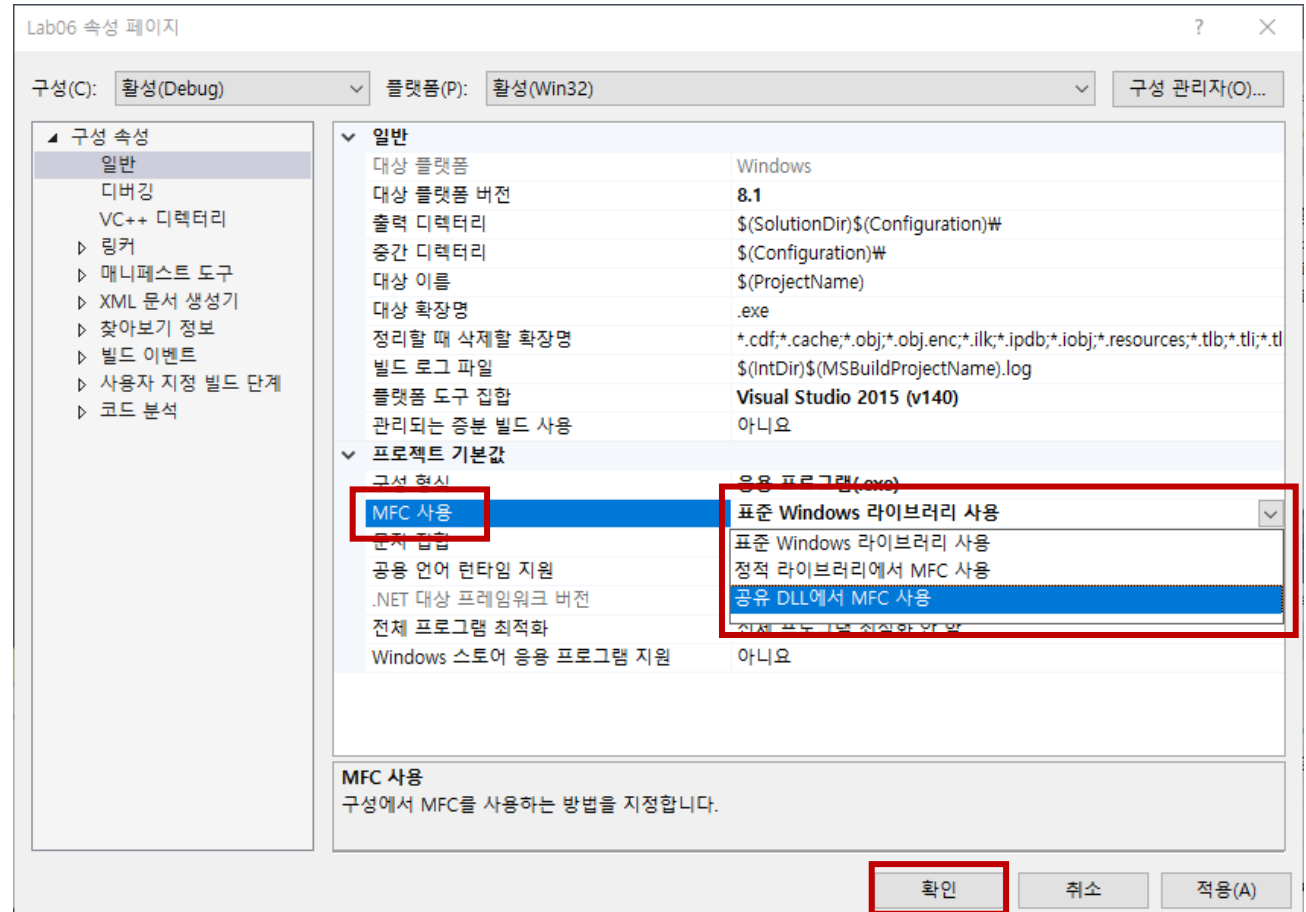
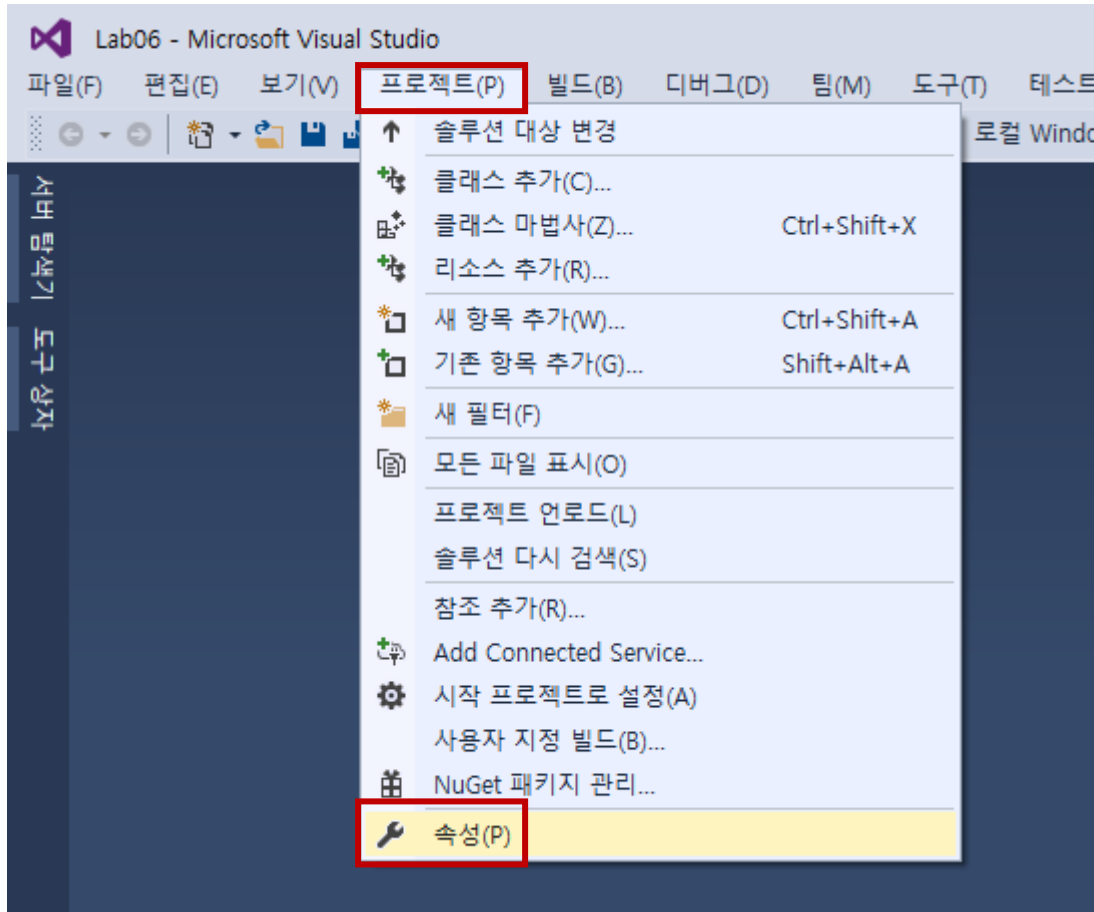


win32 프로젝트 생성하기



프로젝트 속성 설정

- [프로젝트] 메뉴 -> 속성 -> [구성 속성] -> 일반 -> MFC 사용 -> 공유 DLL에서 MFC 사용



실습 해보기 (1)

- CStatic

```
//hello.cpp
#include <afxwin.h>

// Declare the application class
class CHelloApp : public CWinApp {
public:
    virtual BOOL InitInstance();
};

// Create an instance of the application class
CHelloApp HelloApp;

// Declare the main window class
class CHelloWindow : public CFrameWnd {
    CStatic* cs;
public:
    CHelloWindow();
};
```

```
// The InitInstance function is called each
// time the application first executes.
BOOL CHelloApp::InitInstance() {
    m_pMainWnd = new CHelloWindow();
    m_pMainWnd->ShowWindow(m_nCmdShow);
    m_pMainWnd->UpdateWindow();
    return TRUE;
}

// The constructor for the window class
CHelloWindow::CHelloWindow() {
    // Create the window itself
    Create(NULL,
        _T("Hello World!"),
        WS_OVERLAPPEDWINDOW,
        CRect(0, 0, 300, 200));
    // Create a static label
    cs = new CStatic();
    cs->Create(_T("hello world"),
        WS_CHILD | WS_VISIBLE | SS_CENTER,
        CRect(50, 80, 250, 150),
        this);
}
```

실습 해보기 (2)

- CButton

```
// button1.cpp
#include <afxwin.h>
#include <stdlib.h>
#define IDC_BUTTON 100

// Declare the application class
class CButtonApp : public CWinApp {
public:
    virtual BOOL InitInstance();
};

// Create an instance of the application class
CButtonApp ButtonApp;

// Declare the main window class
class CButtonWindow : public CFrameWnd {
    CButton *button;
public:
    CButtonWindow();
};
```

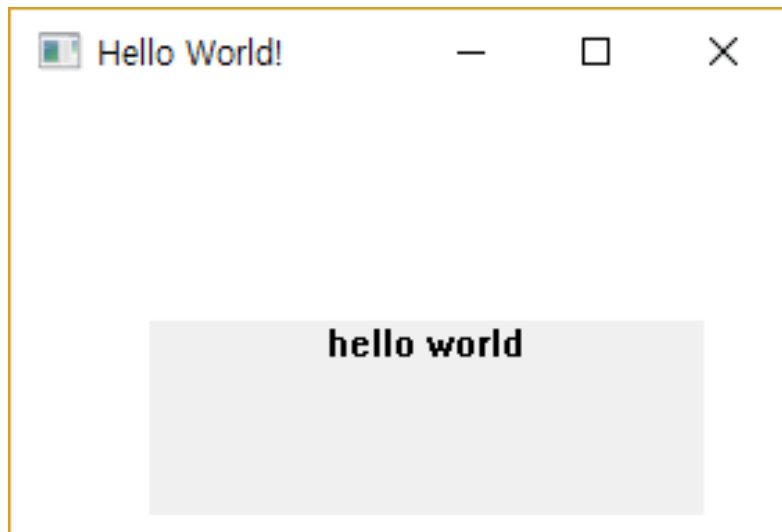
```
// The InitInstance function is called once
// when the application first executes
BOOL CButtonApp::InitInstance() {
    m_pMainWnd = new CButtonWindow();
    m_pMainWnd->ShowWindow(m_nCmdShow);
    m_pMainWnd->UpdateWindow();
    return TRUE;
}

// The constructor for the window class
CButtonWindow::CButtonWindow() {
    // Create the window itself
    Create(NULL,
        _T("CButton Tests"),
        WS_OVERLAPPEDWINDOW,
        CRect(0, 0, 300, 200));

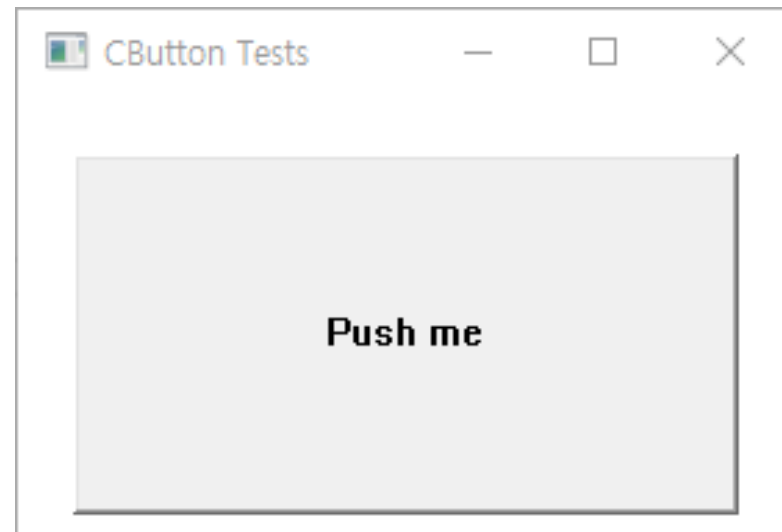
    // Create a button
    button = new CButton();
    button->Create(CString("Push me"),
        WS_CHILD | WS_VISIBLE | BS_PUSHBUTTON,
        CRect(20, 20, 260, 150),
        this,
        IDC_BUTTON);
}
```

실습 해보기

- 실행 결과



CStatic



CButton

실습 해보기 (3)

```
#include <afxwin.h>

#define IDC_BUTTON0 100

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;
    int status;
    void init();
public:
    CToggleWindow();
    afx_msg void HandleButton0();
    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)
END_MESSAGE_MAP()

void CToggleWindow::init()
{
    status = -1
}
```


실습 해보기 (3)

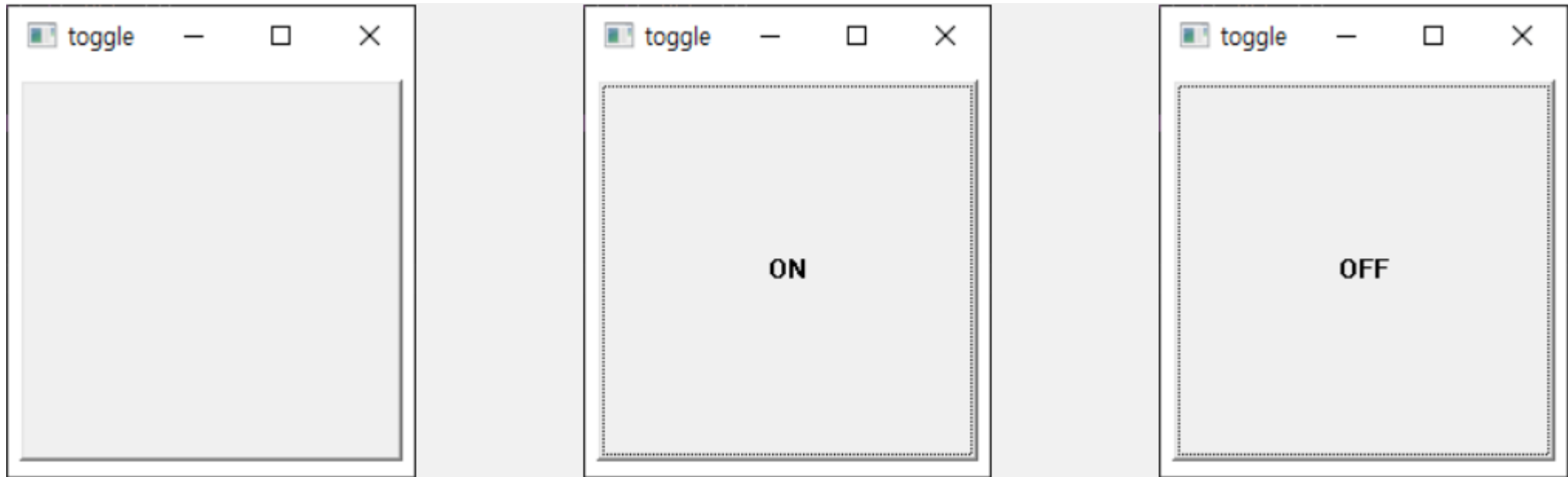
```
// the contructor for the window class
CToggleWindow::CToggleWindow()
{
    //Create the window itself
    Create(NULL, _T("toggle"),
    WS_OVERLAPPEDWINDOW,
    CRect(0,0,230,255));

    init();

    // Create button
    button = new CButton();
    button->Create(_T(""),
    WS_CHILD | WS_VISIBLE |
    BS_PUSHBUTTON,
    CRect(6,6,204,204),
    this,
    IDB_BUTTON0);
}
```

```
// message handler function for number buttons
void CToggleWindow::HandleButton0()
{
    if(status == ON){
        button->SetWindowText(_T("OFF"));
        status = OFF;
    }
    else
    {
        button->SetWindowText(_T("ON"));
        status = ON;
    }
}
```

실습 해보기 (3)



과제 : Tic-Tac-Toe 게임 프로그래밍 하기

- tictactoe.cpp
 - 9개의 버튼 (3x3)
 - **눌리지 않은 버튼은 아무것도 나타나지 않는 상태로 처리.**
 - 버튼 클릭 시 "X" - "O" - "X" - "O" 가 클릭 순서대로 번갈아 가며 나오도록
 - 이미 선택된 버튼을 누르면 "삐" 소리를 통해 경고
 - 버튼의 크기와 위치
 - (6, 6, 34, 34), (36, 6, 64, 34), (66, 6, 94, 34),
 - (6, 66, 34, 64), (36, 36, 64, 64), (66, 36, 94, 64),
 - (6, 66, 34, 94), (36, 66, 64, 94), (66, 66, 94, 94)
 - 전체 창의 크기는 적절히 정한다.

과제 : Tic-Tac-Toe 게임 프로그래밍 하기

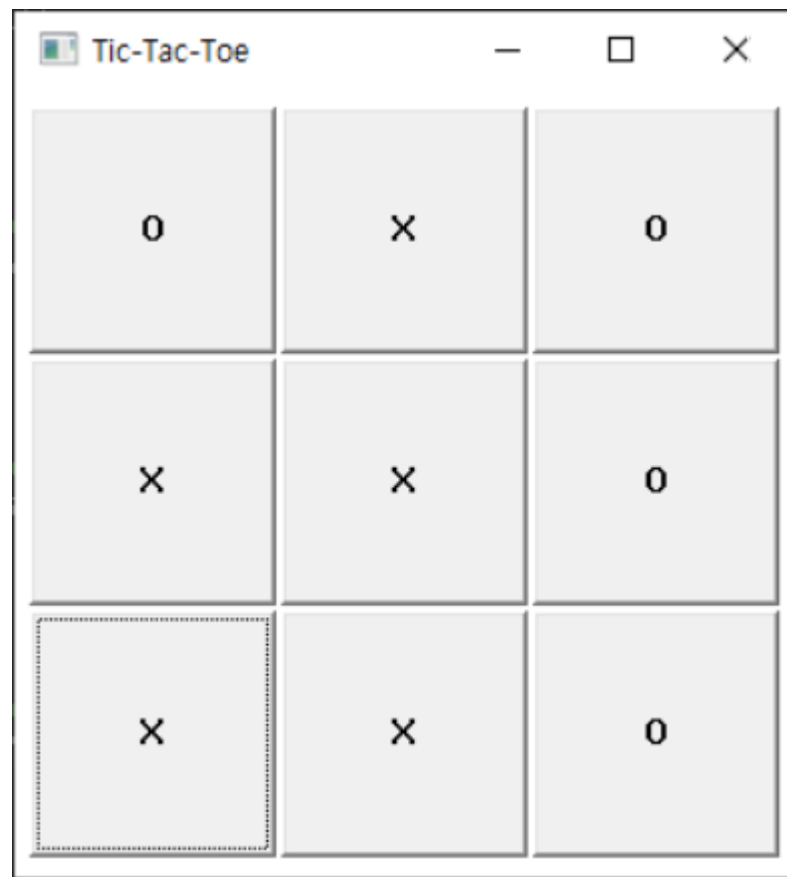
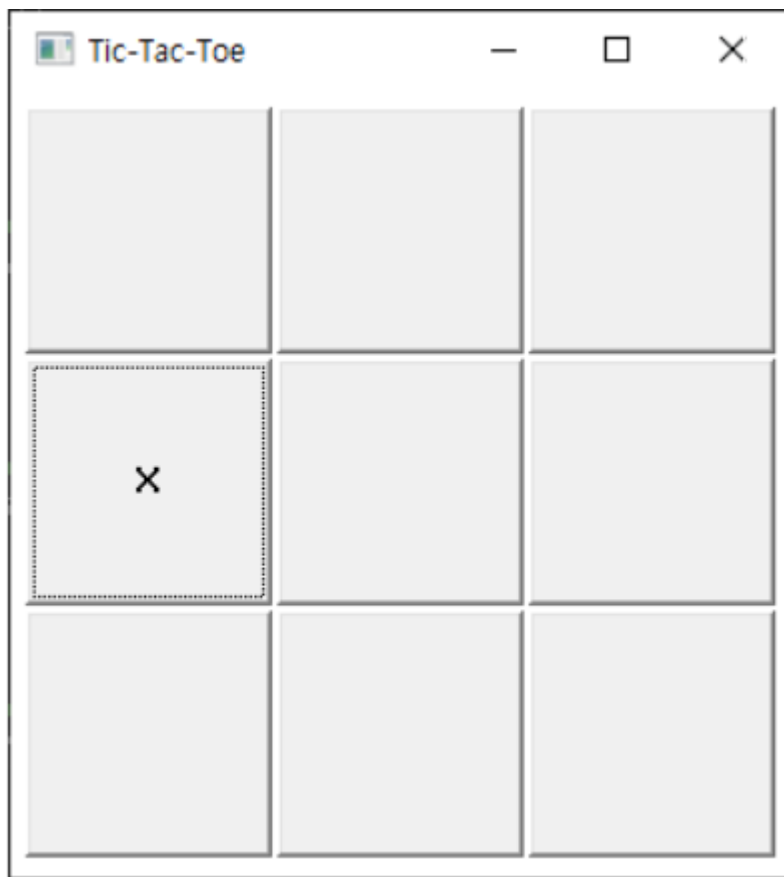
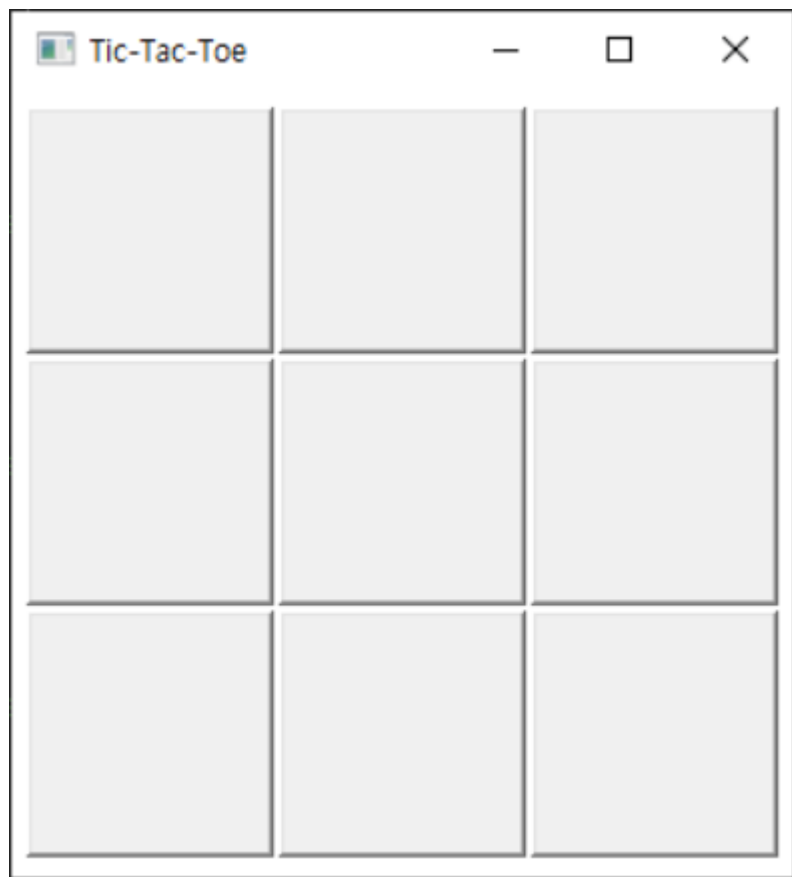
- tictactoe.cpp
 - 9개의 버튼 (3x3) 생성
 - Cbutton 객체 포인터들을 크기가 9개인 배열로 정의
 - Cbutton *button[9];
 - 버튼의 크기와 간격이 일정하므로 for문 / 이중 for문을 통해서 CRect()함수를 이용한 버튼의 위치 및 크기 지정
 - 버튼이 이미 클릭 되었는지 확인하기 위해 배열 사용
 - 메시지 핸들러 함수는 각각 생성하지만 함수 호출을 통해 하나의 함수에서 처리되도록
 - Afx_msg void HandleButton0(), afx_msg void HandleButton1() ...
 - HandleButton(int _iK)

과제 : Tic-Tac-Toe 게임 프로그래밍 하기

- tictactoe.cpp
 - 버튼 ID 상수로 define
 - #define IDB_BUTTON0 100
 - #define IDB_BUTTON1 101
 - $IDB_BUTTON0 + 1 = 100 \Rightarrow IDB_BUTTON1$
 - "X" - "O" 번갈아가면서 출력
 - 변수를 통해 이번 turn을 알 수 있음
 - 경고음 출력
 - 배열을 통해 클릭 된 적이 있는 버튼인지 확인

과제 : Tic-Tac-Toe 게임 프로그래밍 하기

- 실행 결과 (예시)



과제 제출: 보고서 + 솔루션 폴더

- 내용
 - 각 과제의 프로그램 소스와 설명
 - 각 과제의 실행 화면 캡처
- 제출 형식
 - 스노우보드에서 Lab06_이름_학번.doc 다운로드
 - 스노우보드에 제출: Lab06_이름_학번.zip (아래 두 파일 반드시 포함하도록)
 1. 솔루션 폴더 : Lab06_이름_학번 (압축하지 않은 솔루션 폴더)
 2. 보고서: Lab06_이름_학번.doc (설명 없을 경우 감점)
- 제출 기한
 - 5월 4일(화) PM 11:59 까지