





WINAPI

-CHAPTER 10-

SOULSEEK



목차

1. File In-Out
 2. 파일열기 대화상자
 3. Maptool
- 
- 

FILE IN-OUT

1. FILE IN-OUT

파일 핸들러 생성

HANDLE CreateFile(LPCSTR lpFileName, DWORD dwDesiredAccess, DWORD dwShareMode, LPSECURITY_ATTRIBUTES lpSecurityAttributes, DWORD dwCreationDisposition, DWORD dwFlagsAndAttributes, HANDLE hTemplateFile);

- 첫 번째 인자 : 파일이름(경로지정 포함)
- 두 번째 인자 : **GENERIC_READ**(파일읽기), **GENERIC_WRITE**(파일쓰기), **GENERIC_READ | GENERIC_WRITE**(파일 읽고 쓰기)
- 세 번째 인자 : 열려진 파일의 공유 모드를 지정한다. 잘 사용되지 않고 **0**을 지정해 준다.
- 네 번째 인자 : 잘 사용되지 않으므로 **NULL**로 지정한다.
- 다섯 번째 인자 : 파일 생성 및 추가 옵션, **CREATE_ALWAYS**(파일이 존재할 경우 새로 생성), **OPEN_EXISTING**(이미 있는 파일을 열되 만약 없으면 에러 코드를 되돌린다.)를 주로 사용한다.
- 여섯 번째 인자 : **FILE_ATTRIBUTE_NORMAL**을 사용한다.
- 일곱 번째 인자 : **NULL**로 지정

1. FILE IN-OUT

파일 쓰기, 파일 읽기

- **BOOL WriteFile**(HANDLE hFile, LPCVOID lpBuffer, DWORD nNumberOfBytesToWrite, LPDWORD lpNumberOfBytesWritten, LPOVERLAPPED lpOverlapped);
- **BOOL ReadFile**(HANDLE hFile, LPVOID lpBuffer, DWORD nNumberOfBytesToRead, LPDWORD lpNumberOfBytesRead, LPOVERLAPPED lpOverlapped);
 - 파일로부터 데이터를 읽어오거나 쓰는 역할을 한다. 파일 읽기 기능을 활용하면 비트맵도 리소스 추가 없이 사용할 수 있다.
 - 함수의 반환 값은 파일의 저장(불러오기)가 성공하면 **0** 이외의 값을 반환하고, 실패한다면 **0(FALSE)**를 반환한다.
 - 첫 번째 인수(**hFile**) : **CreateFile**로 불러온 파일의 핸들을 전달한다.
 - 두 번째 인수(**lpBuffer**) : 데이터를 저장할(혹은 불러올) 공간의 포인터를 전달한다, 이 인수는 매우 중요한데, 구조체나 클래스의 포인터 또한 전달할 수 있다.
 - 세 번째 인수(**nNumberOfBytesToWrite / nNumberOfBytesToRead**) : 저장할(불러올) 바이트 수를 지정
 - 네 번째 인수(**lpNumberOfBytesWritten / lpNumberOfBytesRead**) : 실제로 저장한(불러온) 바이트 수를 저장할 **DWORD**형 포인터를 전달한다, 파일의 끝일 경우 저장하고 불러오게 하는 크기와 실제로 저장하고 불러올 크기가 맞아 떨어지지 않을 경우가 많다.
 - 다섯 번째 인수(**lpOverlapped**) : 첫 번째 인수로 전달한 **hFile**의 파라미터에서 **FILE_FLAG_OVERLAPPED** 플래그를 가지고 있을 때, **OVERLAPPED**형 구조체를 전달한다. 사용하지 않는다면 **NULL(0)**을 전달한다.

핸들 삭제

- **BOOL CloseHandle**(HANDLE hObject);
 - 열려 있는 핸들을 닫는 역할을 한다.
 - 핸들을 닫는데 성공했다면 **0** 이외의 값을 반환하고, 실패했다면 **0(FALSE)**를 반환한다.

1. FILE IN-OUT

```
TCHAR buf[1024] = TEXT("abcde");  
HANDLE hFile;  
DWORD dwRead;
```

```
case WM_LBUTTONDOWN:
```

```
    hFile = CreateFile(TEXT("Test.txt"), GENERIC_READ, 0, NULL, OPEN_EXISTING,  
                       FILE_ATTRIBUTE_NORMAL, NULL);
```

```
    if (hFile != INVALID_HANDLE_VALUE)
```

```
    {
```

```
        ReadFile(hFile, buf, 1024, &dwRead, NULL);
```

```
        CloseHandle(hFile);
```

```
        MessageBox(hWnd, buf, TEXT("알림"), MB_OK);
```

```
    }
```

```
return 0;
```

```
case WM_RBUTTONDOWN:
```

```
    hFile = CreateFile(TEXT("Test.txt"), GENERIC_WRITE, 0, NULL, CREATE_ALWAYS,  
                       FILE_ATTRIBUTE_NORMAL, NULL);
```

```
    WriteFile(hFile, buf, strlen(buf), &dwRead, NULL);
```

```
    CloseHandle(hFile);
```

```
    MessageBox(hWnd, TEXT("Test.txt를 생성 했습니다."), TEXT("알림"), MB_OK);
```

```
return 0;
```

파일열기 대화상자

2. 파일열기 대화상자

BOOL GetOpenFileName(LOPENFILENAME lpofn);

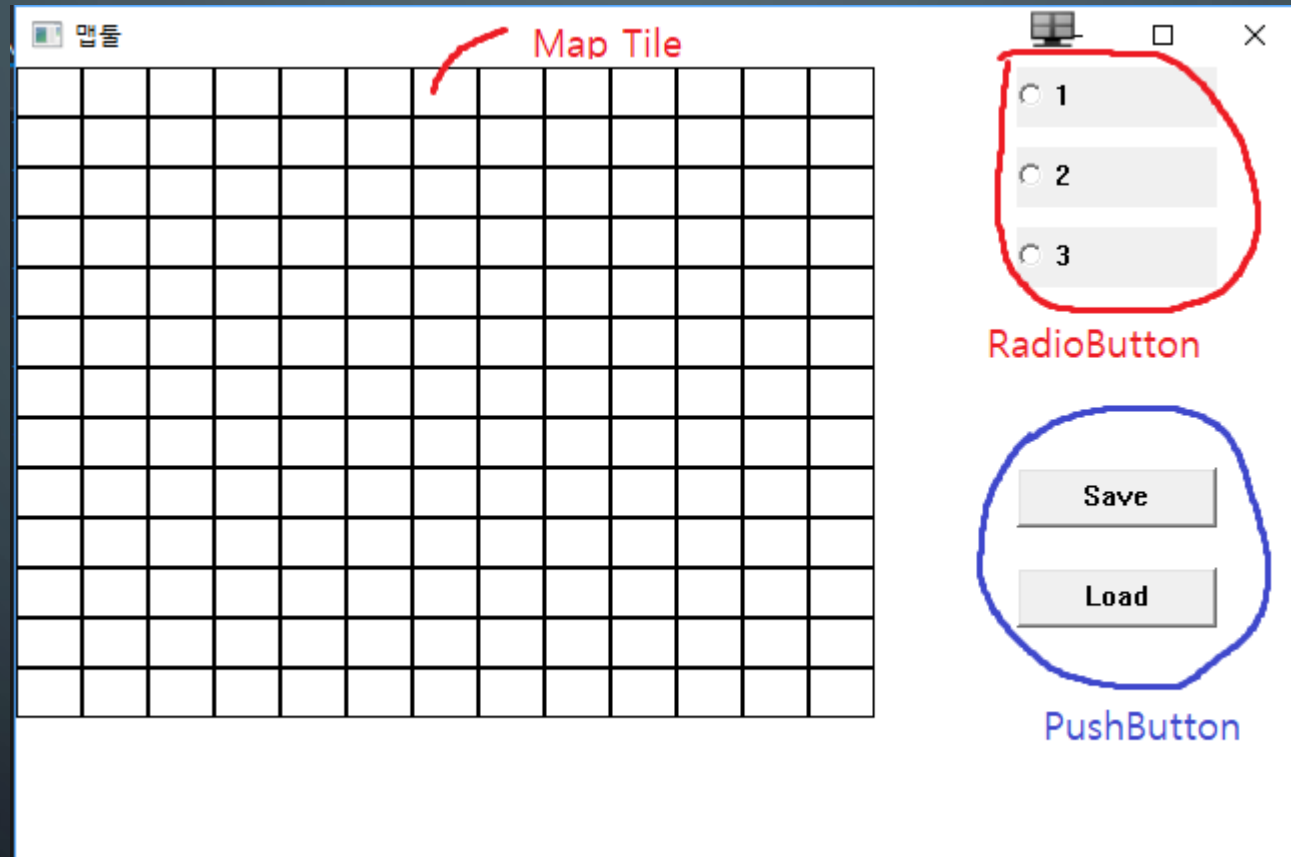
- **OPENFILENAME** 구조체의 포인터를 인자로 가진다.
- 파일을 제대로 입력 받았으면 **0**이 아닌 값을 리턴 하고 사용자가 입력을 취소 했을 때는 **0**을 리턴 한다.
- **OPENFILENAME** 구조체에 의해서 파일열기 대화창의 모양이나 기능이 달라진다.
- **IStructSize** : **OPENFILENAME** 구조체의 크기를 지정하며 이 값은 버전 확인을 위해 사용된다. **sizeof(OPENFILENAME)**을 대입해 주면 된다
- **hwndOwner** : 대화상자의 소유주를 지정한다. 소유주가 없을 경우는 **NULL**로 지정할 수도 있다.
- **hInstance** : 별도의 대화상자 템플릿을 사용할 경우 리소스를 가진 인스턴스 핸들을 지정한다. 필요없으면 **NULL**로 지정한다
- **lpstrFilter** : 파일 형식 콤보 박스에 나타낼 필터들이며 널 문자를 기준으로 "파일 형식\0필터"와 같이 기술한다. 여러 개의 필터를 \0로 구분해서 동시에 지정할 수 있다. **Ex) "Every File(*.*)\0*. *\0Text File\0*.txt;*.doc\0";**
- **lpstrFile** : 파일 이름 에디트에 처음 나타낼 파일명을 지정하되 필요없을 경우는 **NULL**로 지정한다. 또한 이 멤버는 사용자가 최종적으로 선택한 파일의 이름을 리턴하는 용도로 사용된다. 즉 파일 대화상자가 종료된 후 이 멤버를 통해 사용자가 어떤 파일을 선택했는지를 알게 된다.
- **nMaxFile** : **lpstrFile** 멤버의 길이이다. 최소 **256**문자 분의 길이를 가져야 한다.
- **lpstrInitiaDir** : 파일 찾기를 시작할 디렉토리를 지정한다. 대화상자가 열릴 때 이 멤버가 지정하는 디렉토리를 현재 디렉토리로 사용한다.
- **Flags** : 대화상자의 모양과 동작을 지정하는 옵션을 설정하는 플레그이다. 이 멤버에 지정하는 플레그에 따라 여러 파일 선택, 커스텀 대화상자, 혹은 함수 사용, 도움말 지원 여부, 공유 상태 확인 등의 기능이 추가된다.

The image features a dark blue gradient background with faint, concentric circular patterns. In the four corners, there are decorative white line art elements resembling circuit boards or neural networks, with lines and small circles connecting them.

MAPTOOL

3.MAPTOOL

Tile Map을 사용하는 게임의 맵툴을 제작해보자.(이차원 배열로 미로 찾기)



학습과제

1. 파일열기 **DLG** 예제들을 확인하고 공부해보자.
2. 파일열기 **DLG**를 이용해서 맵 툴을 만들어 보자.
3. 제공된 리소스를 활용해서 맵툴과 맵툴의 스테이지 들로 플레이 되는 **BattleCity**게임을 만들어보자.