**Music- Score player**

****

**휴먼ICT 소프트웨어 공학 Team 6**

**──────────────────────**

|  |  |
| --- | --- |
| 분반 | 월 7,8,9 |
| 교수 | 송인식 교수님 |
| 팀 | Team 6 |
| 팀원 | 20141261 송제웅  20140786 조장연  20140661 신주환  20141001 이지호  20141359 정유석  20146235 김정훈 |

# Documentation for developers.

### How to obtain the source code.

1. <https://github.com/jjiho/Music-Score-2019-SW-Team-6->에 접속한다.

2.Music-Score Player 파일을 다운받는다.

3.Open CV 라이브러리를 설치한다.

4.Visual Studio 2017을 이용하여 빌드한다.

The layout of your directory structure.

Music-Score Player 2.4

└ Music Program

└ Perfect Pitch

└ Midi

└ Image Process

└ Score Processor

### How to build the software.

Empty Parking Space Project는 아래와 같이 총 5개의 Module로 구성된다.

* MIDI
* Score processor
* UI

각 Module의 Build 방법은 아래와 같다.

**[MIDI]**

-따로 설치할 필요 없이 Visual Studio 에서 제공되는 MIDI를 사용하였으므로 MIDI.h를 Include해준다.

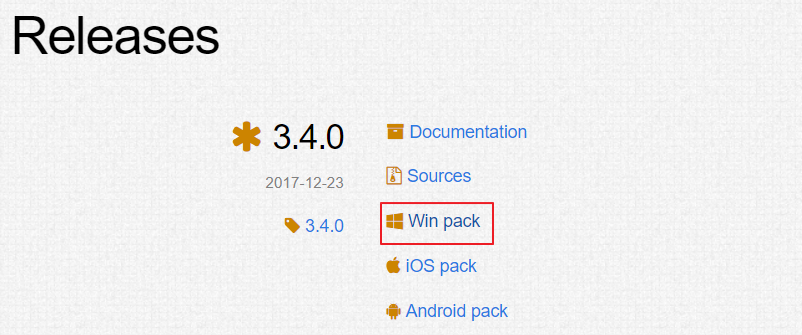
**[Score processor]**

Score proccessor를 위해서는 Visual Studio IDE와 Open cv API를 필요로한다.

**[OpenCV - 설치]**

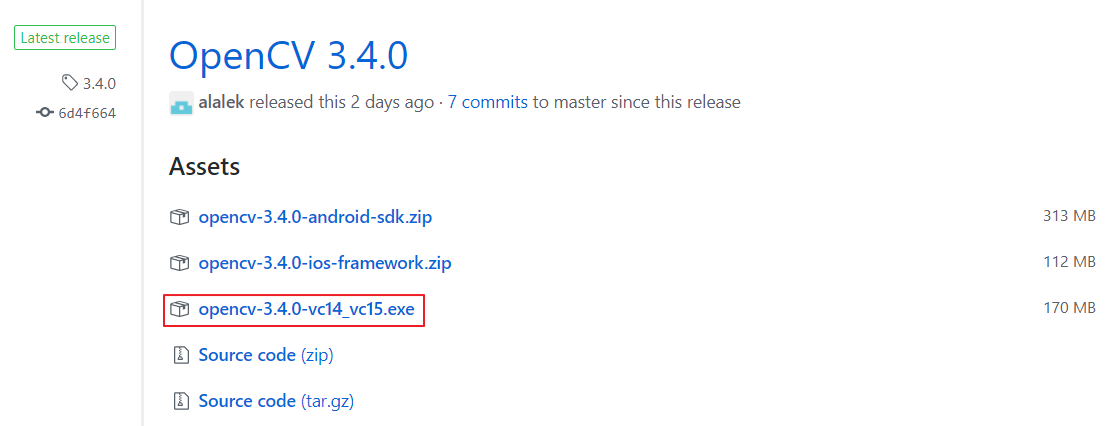
1. OpenCV 공식 사이트에서 Win pack을 클릭하여 다운로드 받습니다.

<http://opencv.org/releases.html>

[](http://opencv.org/releases.html)

1-1. github에서도 다운로드 가능합니다.

<https://github.com/opencv/opencv/releases>

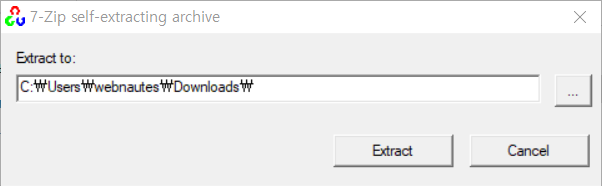
[](https://github.com/opencv/opencv/releases)

2. opencv-3.4.0-vc14\_vc15.exe 이름의 파일이 다운로드 됩니다

이름에서도 알수 있듯이 이제는 vc14(Visual Studio 2015) 외에 vc15(Visual Studio 2017)도 미리 빌드된 라이브러리가 제공됩니다.

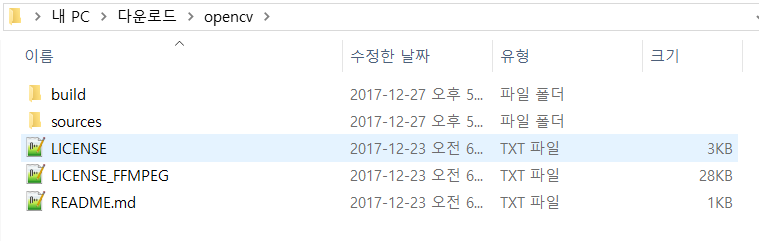
3. 실행하면 압축을 해제할 경로를 물어보는데 우선은 다운로드 받은 폴더로 지정했습니다.

**Extract**를 클릭하면 압축 해제가 시작됩니다.



4. opencv 폴더가 생성됩니다.

opencv 폴더는 미리 빌드된 라이브러리가 포함되어 있는 **build** 폴더와 OpenCV 3.4 소스 코드가 포함되어 있는 **sources** 폴더로 구성되어 있습니다.



5. **opencv** 폴더 이름을 **opencv-3.4**로 변경하고 C:\로 복사했습니다.

포스팅에서는 **OpenCV 라이브러리** 위치가 **C:\opencv-3.4**인 것으로 하고 진행하겠습니다.

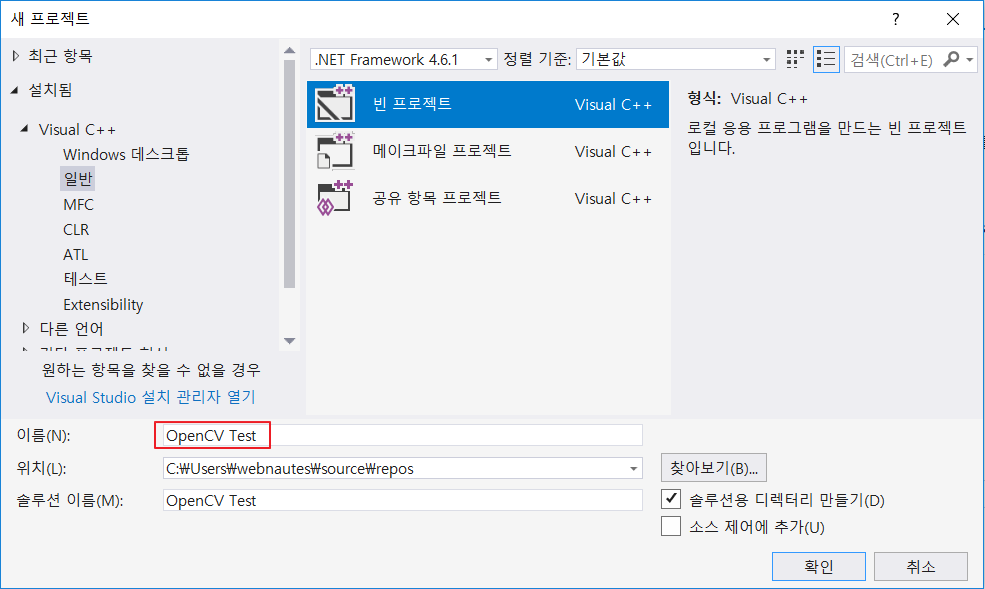
**[OpenCV - 환경 설정]**

1. 비주얼스튜디오의 메뉴에서 **파일 > 새로 만들기 > 프로젝트**를 선택합니다.

2. **새 프로젝트** **창**에서 **빈 프로젝트**를 선택합니다.

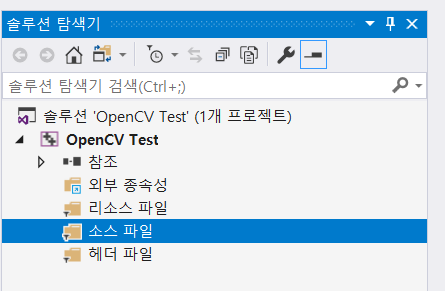
(예전에 있던 Win32 콘솔 프로젝트가 Visual Studio 버전업에 따라 사라져서 대신 빈 프로젝트를 사용합니다.)

**프로젝트 이름**을 적어주고 **확인**을 클릭합니다.

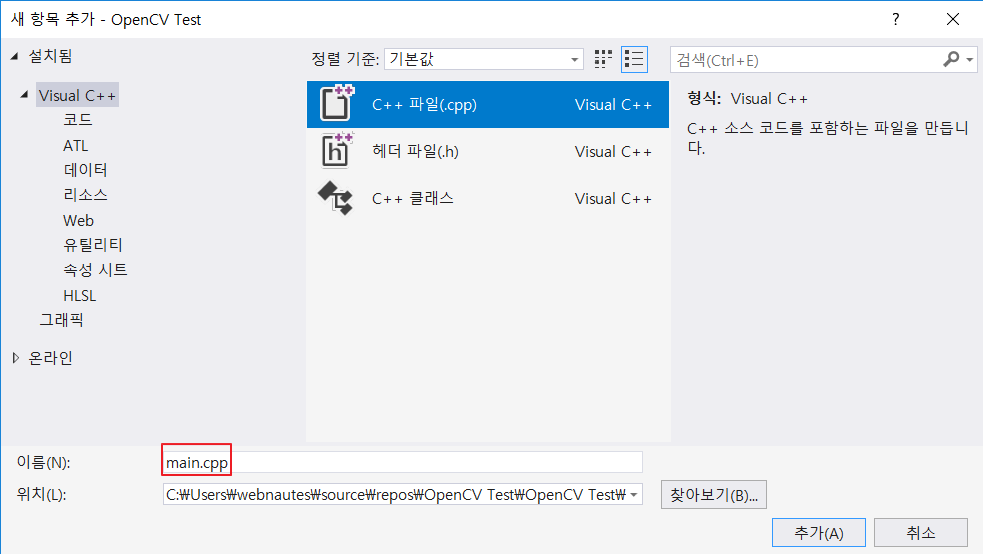


3. **솔루션 탐색기**에서 **소스 파일**을 선택합니다.

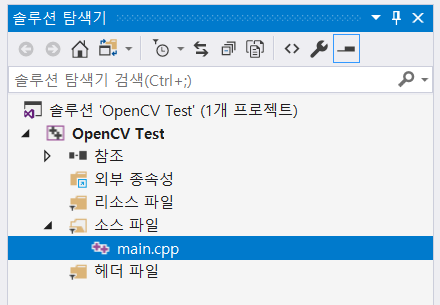
마우스 오른쪽 버튼을 눌러 보이는 메뉴에서 **추가 > 새 항목**을 클릭합니다.



**C++ 파일(.cpp)**를 선택하고 소스 파일의 **이름**을 적고 **추가**를 클릭합니다.



소스 파일 항목에 cpp 파일이 추가됩니다.

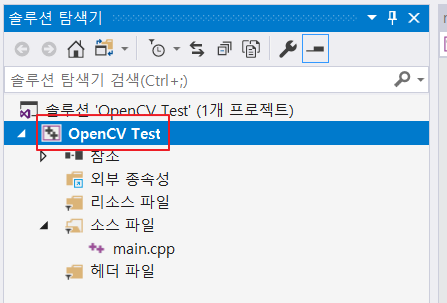


4.솔루션 플랫폼을 x64로 변경합니다.

x86용이 필요시 OpenCV 소스 코드를 x86에 맞추어 컴파일해줘야 합니다.

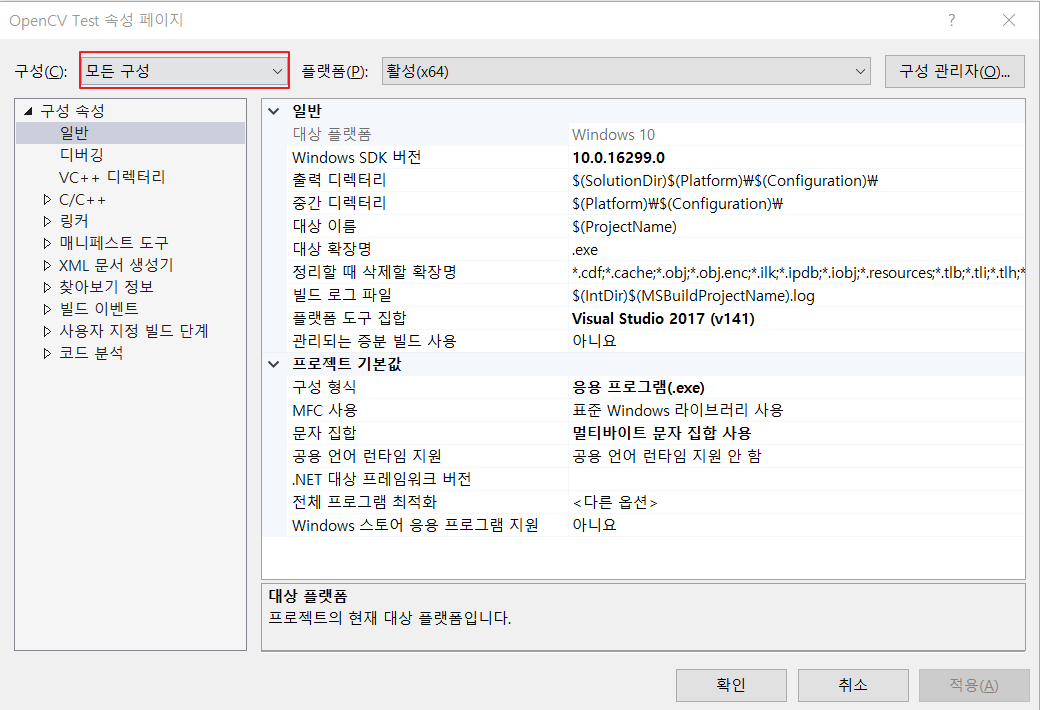


5. 솔루션 탐색기에서 프로젝트 이름을 선택하고 메뉴에서 **프로젝트 > 속성**을 선택합니다.



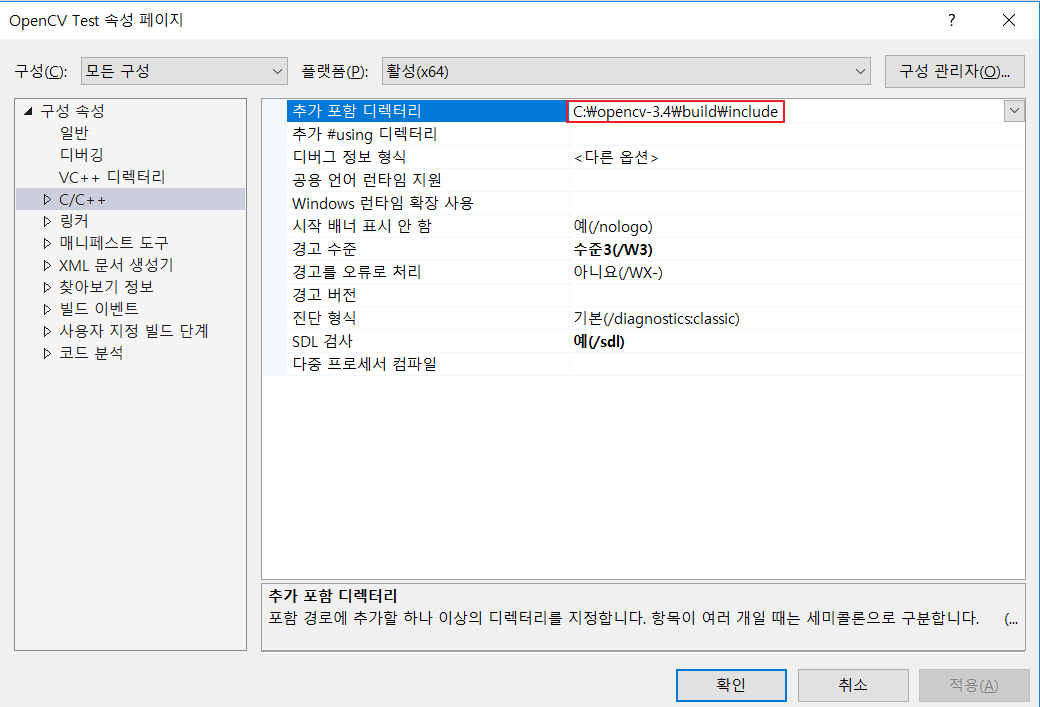
6. **속성 페이지 창**에서 **구성**을 **모든 구성**으로 변경합니다.

Debug 모드와 Release 모드의 공통 부분이 같이 설정됩니다.



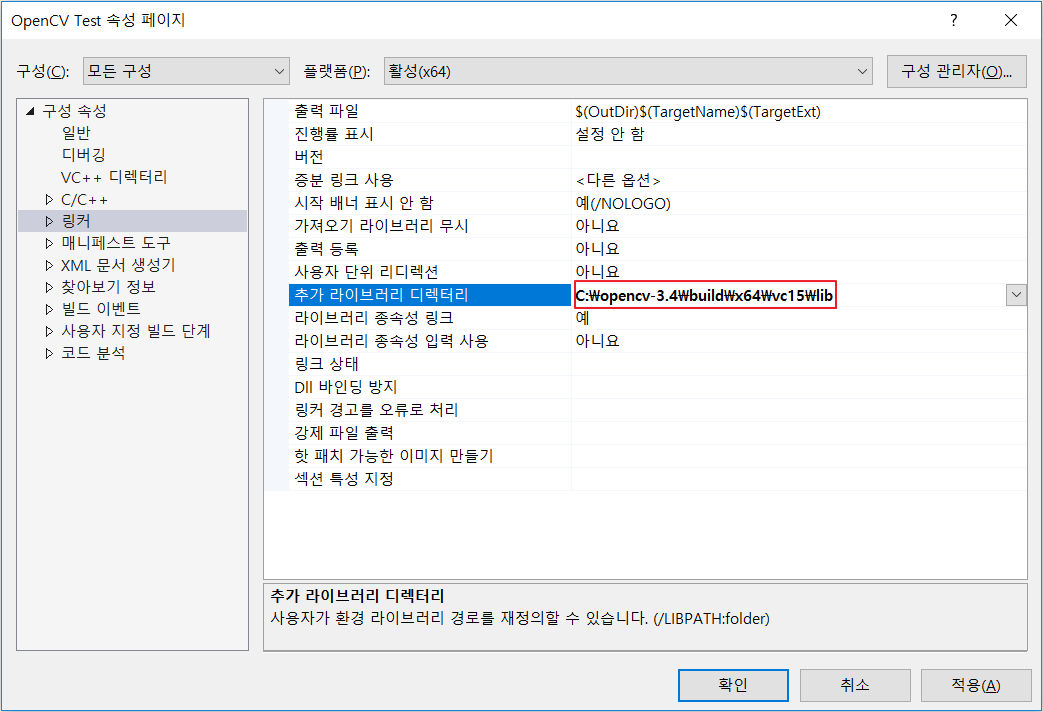
7. **속성 페이지** 창에서 **C/C++**를 선택하고 **추가 포함 디렉터리** 항목에 **OpenCV 헤더 파일 위치**를 입력합니다.(설치한 openCV의 경로에 맞춰 설정합니다.)

**ex> C:\opencv-3.4\build\include**

****

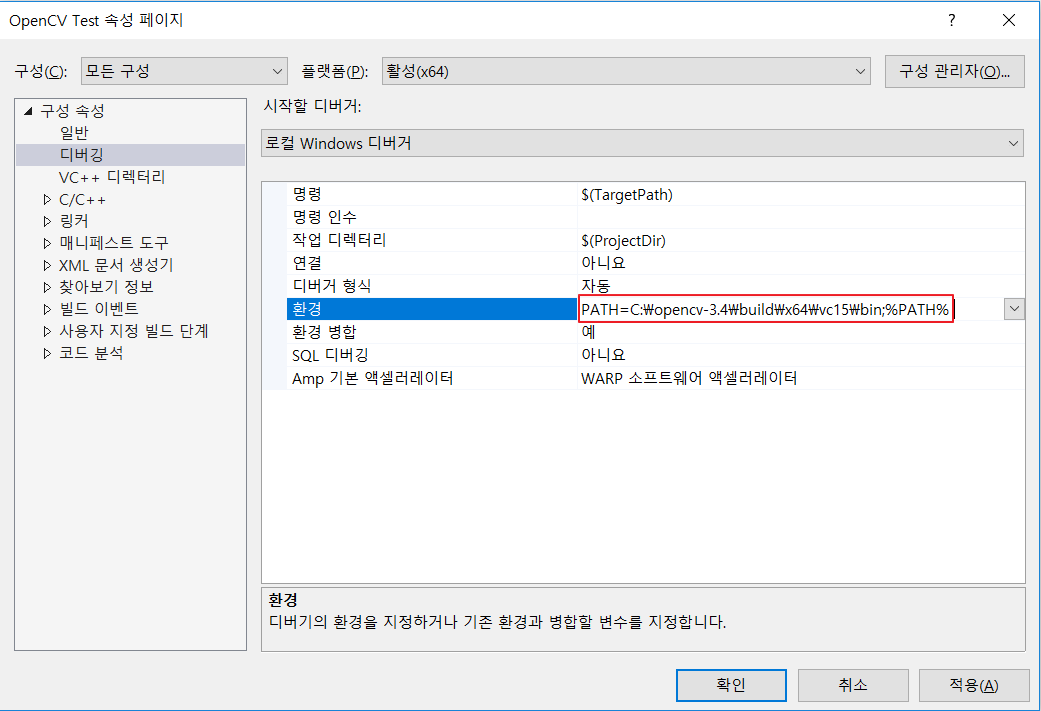
8. **속성 페이지** 창에서 **링커**를 선택하고 **추가 라이브러리 디렉터리** 항목에 **OpenCV 라이브러리 파일** **위치**를 입력합니다.(설치한 openCV의 경로에 맞춰 설정합니다.)

**ex> C:\opencv-3.4\build\x64\vc15\lib**

****

9. **속성 페이지** 창에서 **구성 속성 > 디버깅**을 선택하고 **환경** 항목에 **OpenCV 라이브러리 DLL 파일 위치**를 입력합니다.(설치한 openCV의 경로에 맞춰 설정합니다.)

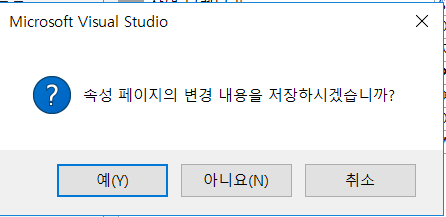
**ex> PATH=C:\opencv-3.4\build\x64\vc15\bin;%PATH%**

****

10. 구성을 활성(Debug)로 변경하고

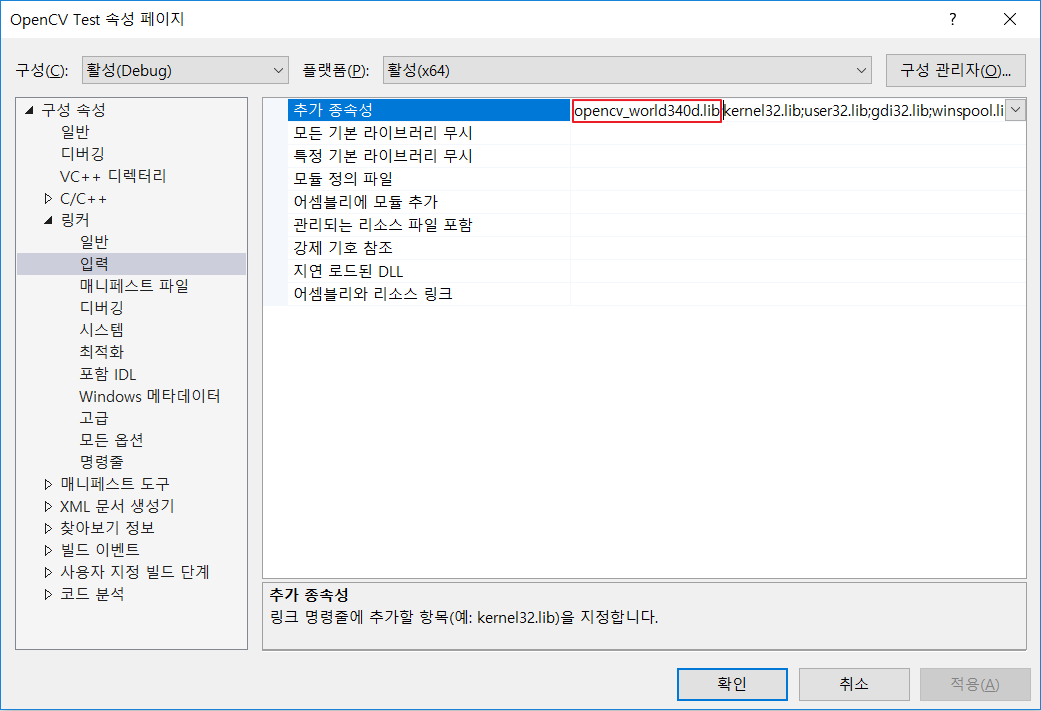


예를 눌러서 설정을 저장합니다.



11. **링커 > 입력**을 선택하고 **추가 종속성**의 값 맨앞에 다음 내용을 입력합니다.

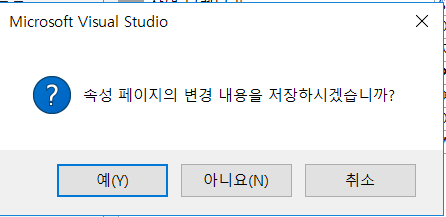
opencv\_world340d.lib;



12. 구성을 Release로 변경하고

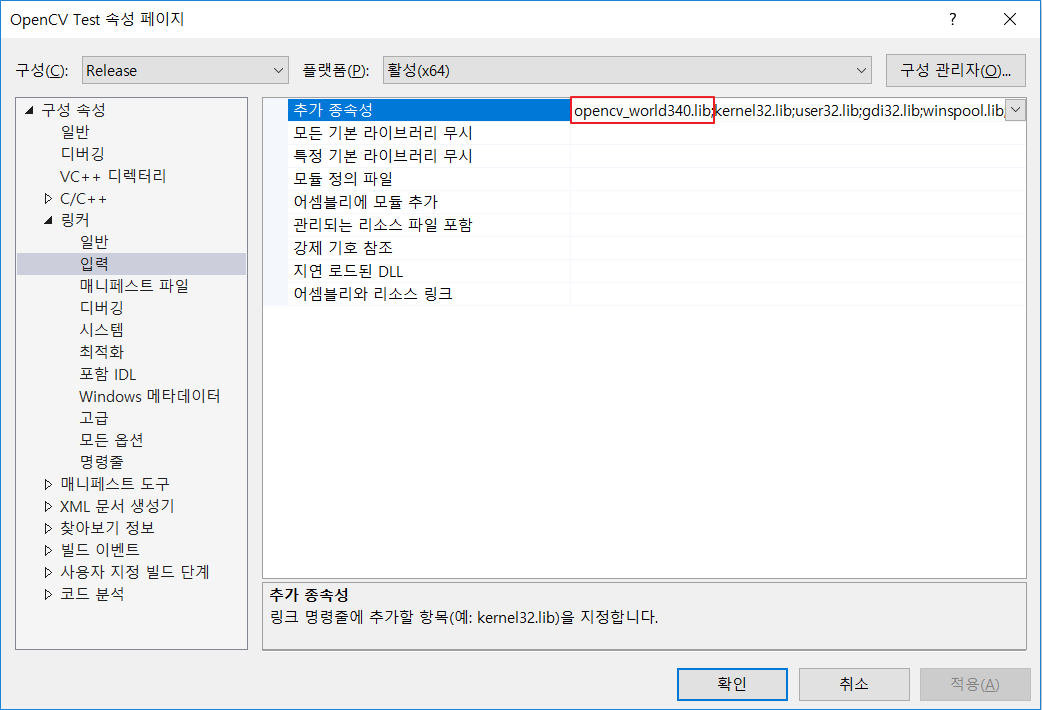


예를 눌러서 설정을 저장합니다.



13. **링커 > 입력**을 선택하고 추가 종속성의 값 맨앞에 다음 내용을 입력합니다.

opencv\_world340.lib;



14. 설정은 마무리 되었으며, git에서 다운로드 한 파일들을 추가한 후 프로그램을 실행한다.

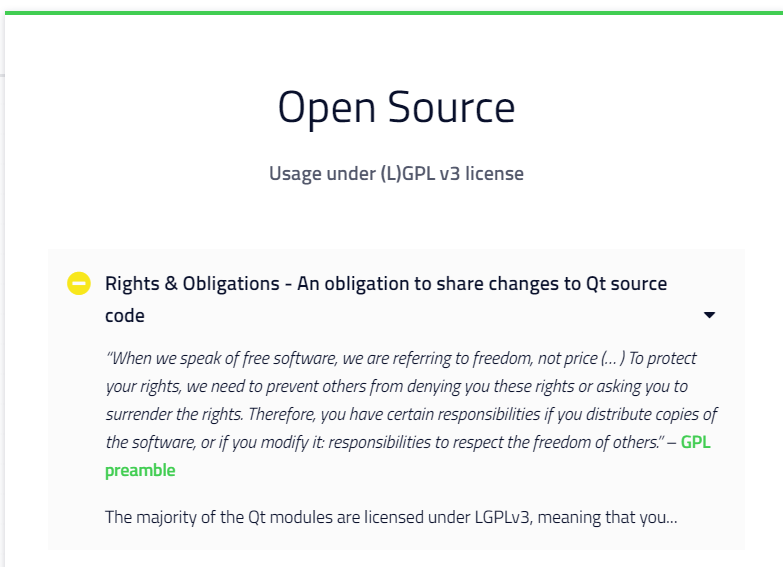
**[UI]**

-Qt는 C++ GUI를 제공하는 API로 운영체제에 상관없이 QtCreator IDE를 통하여 공통된 개발 환경을 제공한다. 아래의 문서는 Qt Creator를 통한 Build 방법을 소개한다.

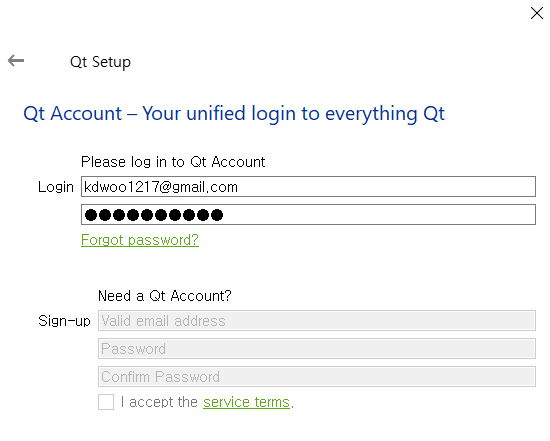
1. [Qt 홈페이지](https://qt.io)에 접속하여 아래 Download 버튼을 클릭한다.



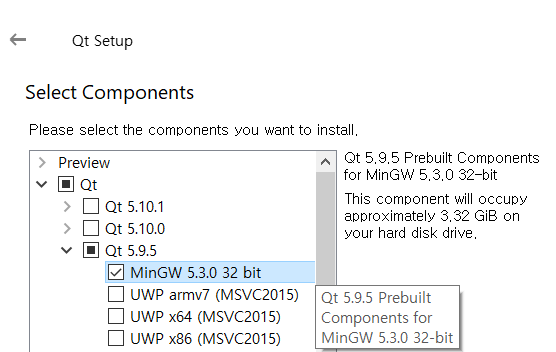
1. Open Source 버전의 Qt를 다운받는다.



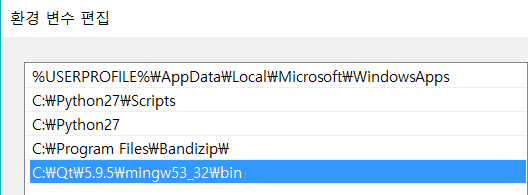
1. Qt를 설치하기 위해서는 로그인을 요구하므로 가입 후 ID를 입력한다.



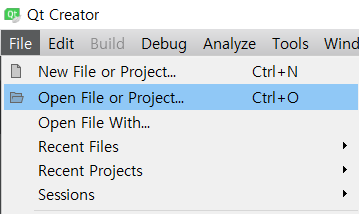
1. Qt 5.9.5 를 클릭한 후 MinGW 를 설치한다. Qt Creator IDE는 MinGW를 기반으로 동작하므로 다른 파일을 설치할 경우 정상적으로 동작하지 않을 수 있다.



1. 고급 시스템 설정 - 환경변수에 MinGW 폴더 하위에 bin 폴더를 추가한다.



1. Qt Creator IDE를 실행한 후 Qt Project File을 열어 Build 한다.



### How to release a new version of your software.

1. <https://github.com/jjiho/Music-Score-2019-SW-Team-6->에 접속한다.

2. 해당 페이지의 Pull requests 탭을 선택한다.

3. 브라우저 중앙 우측에 위치한 New pull requests 버튼을 클릭한다.

4. 이전 버전과의 변경된 사항을 상세히 기술하고 Contributor를 통하여 새로운 버전을 등록한다.

### How to access the list of outstanding bugs and the list of resolved bugs, and how to resolve a bug.

**[Access the list of outstanding bugs]**

1. <https://github.com/jjiho/Music-Score-2019-SW-Team-6->에 접속한다.

2. 해당 페이지의 Issues 탭을 선택한다.

3. 브라우저 중앙 좌측에 위치한 Open 탭에 있는 Bug를 체크한다.

**[How to resolve bug]**

1. 사용자의 Report를 통하여 상황을 파악한다.

2. 의심이 되는 부분이 어느 곳인지 추측한다.

! 사용자의 Report가 충분히 추측할 수 없다면 Comment를 통하여 사용자에게 더 상세한 사항을 물어본다.

3. Unit Test 등을 통하여 해당 상황을 재현한다.

4. 발견된 Bug를 수정하고 이에 대한 다른 오류가 발생하지 않는지 전체 Unit Test를 다시 진행한다.

5. 수정된 파일을 Pull Requests를 통하여 업데이트 한다.

6. Bug를 작성한 User에게 어느 버전에서 문제가 해결되었는지 안내하고 해당 Issue를 Closed 한다.

**[Access the list of resolved bugs]**

1. <https://github.com/jjiho/Music-Score-2019-SW-Team-6->에 접속한다.

2. 해당 페이지의 Issues 탭을 선택한다.

3. 브라우저 중앙 좌측에 위치한 Closed 탭에 있는 Bug를 체크하고 어떤 버전에서 Bug가 해결되었는지 확인한다.

### The source code should contain.

**[Add new methods & Class]**

“\*\*/ \* …\* /\*\*” 형태의 주석을 사용하여 아래와 같은 내용이 반드시 포함되도록 한다.

* 작성자
* 마지막 작성일자
* 함수 또는 클래스에 대한 자세한 설명

또한 메소드나 클래스를 정의할 때는 다른 개발자가 빠르게 이해할 수 있도록 기능을 담은 이름으로 명명한다.

**[Edit current methods & Class]**

기존 메소드나 클래스의 변경이 이루어 지는 경우 반드시 아래와 같은 내용을 주석에 추가한다.

* 수정자
* 수정일자
* 수정이유
* 변경된 내용
* 변경사항에 대한 자세한 설명