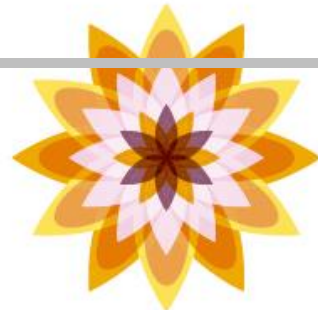
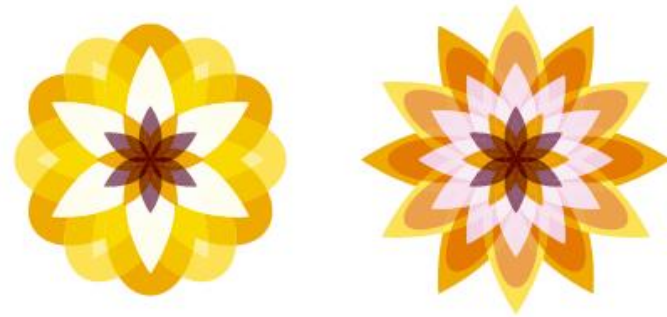


*Chapter 01*

# Visual Studio와 openFrameworks 기초



# 1. Visual Studio

- C 언어를 이용해서 '벽돌 깨기' 게임을 만들어볼 것인데 이번 장에서는 이를 위한 2개의 새로운 프로그램 개발 환경에 대해서 소개한다.
- 먼저 Visual Studio라는 통합 개발 환경 (Integrated Development Environment: IDE)이다.
- Visual Studio는 Microsoft사의 프로그램으로써 C 언어 이외에 C#, Visual Basic, F#, JavaScript, C++, Python 등의 다양한 언어로 프로그래밍하는 것을 지원하는 개발 환경이다.
- Visual Studio는 윈도우 운영체제에서 PC용 소프트웨어 개발뿐만 아니라 모바일 등 다양한 장치를 대상으로 앱을 개발할 수도 있다
- 두번째는 openFrameworks라는 C++ 언어를 이용한 보다 재미있는 프로그래밍을 위한 라이브러리 (library)이다.

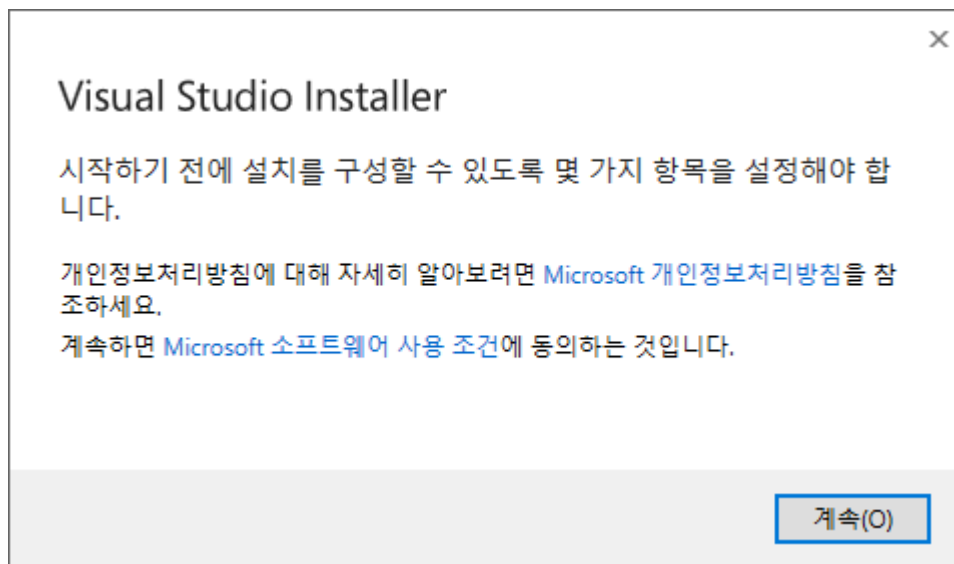
# 1. Visual Studio

- openFrameworks 라이브러리는 “창의적인 코딩”을 위한 C++ 오픈 소스 라이브러리로서, C++ 언어를 이용한 그래픽 기반 인터랙션(interaction, 상호 작용)을 쉽게 구현할 수 있도록 해주는 라이브러리이다.
- openFrameworks는 Windows, Mac OSX, Linux, Android, iOS 등에서도 실행할 수 있는 프로그램을 개발할 수 있다.
- 또한 자바 기반인 Processing 언어, Flash AS3.0, Cinder 등에 비해서, openFrameworks는 C++ 기반이기 때문에 빠른 속도와 다른 라이브러리와 결합이 쉽다는 장점이 있다.

## 2. Visual Studio와 openFrameworks 설치

### ■ Visual Studio 2019 Community 버전 설치

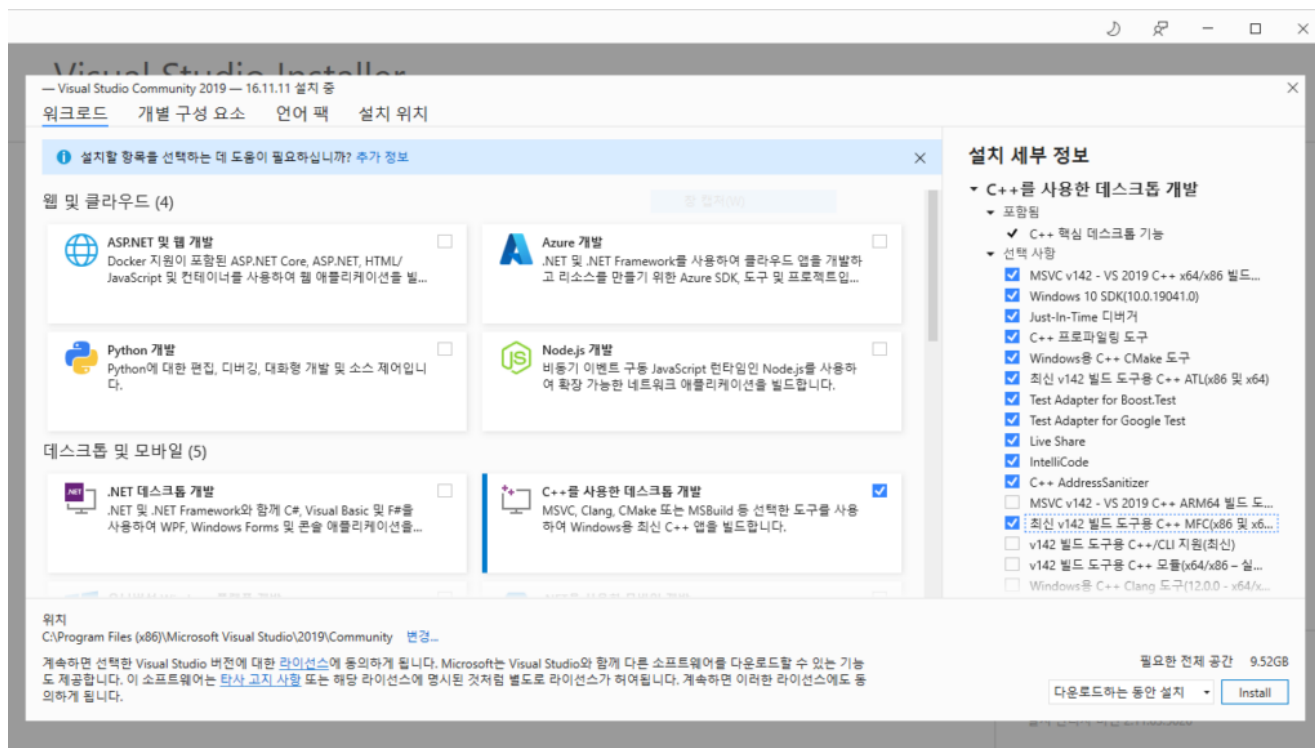
- 다운로드 페이지: <https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=Community&rel=16>
- 다운로드 페이지를 클릭하면 약 1.4 MB의 설치 파일이 다운로드 된다.
- 파일 이름은 “vs\_Community.exe”이다.
- 다운로드 받은 파일을 더블클릭해 설치를 시작한다.
- 만일 “이 앱이 디바이스를 변경할 수 있도록 허용하시겠습니까?”라는 창이 뜬다면 “예”를 클릭한다.
- 아래와 같이 사용 조건에 동의하는지를 묻는 창이 뜬다.
- “계속”을 클릭한다.



## 2. Visual Studio와 openFrameworks 설치

### ■ Visual Studio 2019 Community 버전 설치

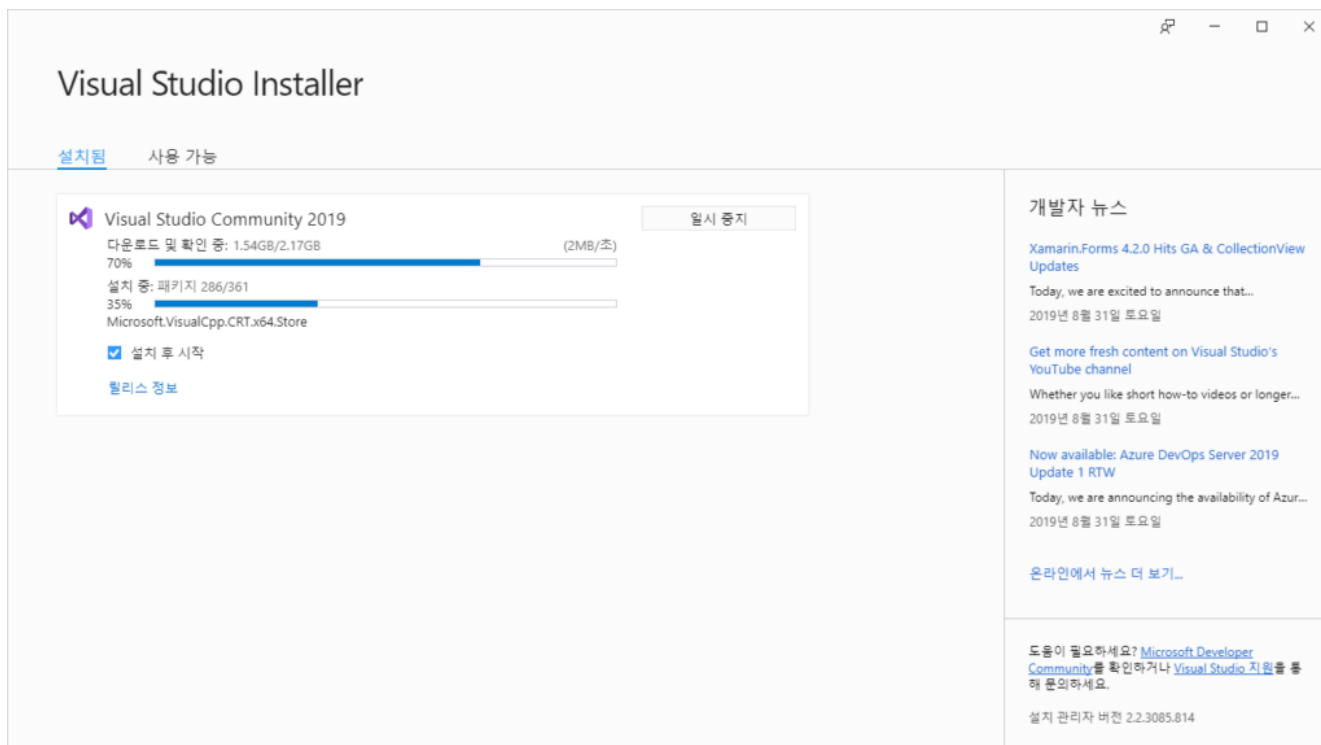
- 아래와 같이 설치할 항목을 고르는 창이 뜬다.
- 저는 C++을 사용할 것이므로, “워크로드” 탭에서 “C++을 사용한 데스크톱 개발”에 체크했다.
- 또한, MFC를 이용할 것이므로, 화면의 오른쪽 부분에서 추가적으로 “최신 v143 빌드 도구용 C++ MFC(x86 및 x64)”에 체크했다.
- “설치”를 클릭한다.



## 2. Visual Studio와 openFrameworks 설치

### ■ Visual Studio 2019 Community 버전 설치

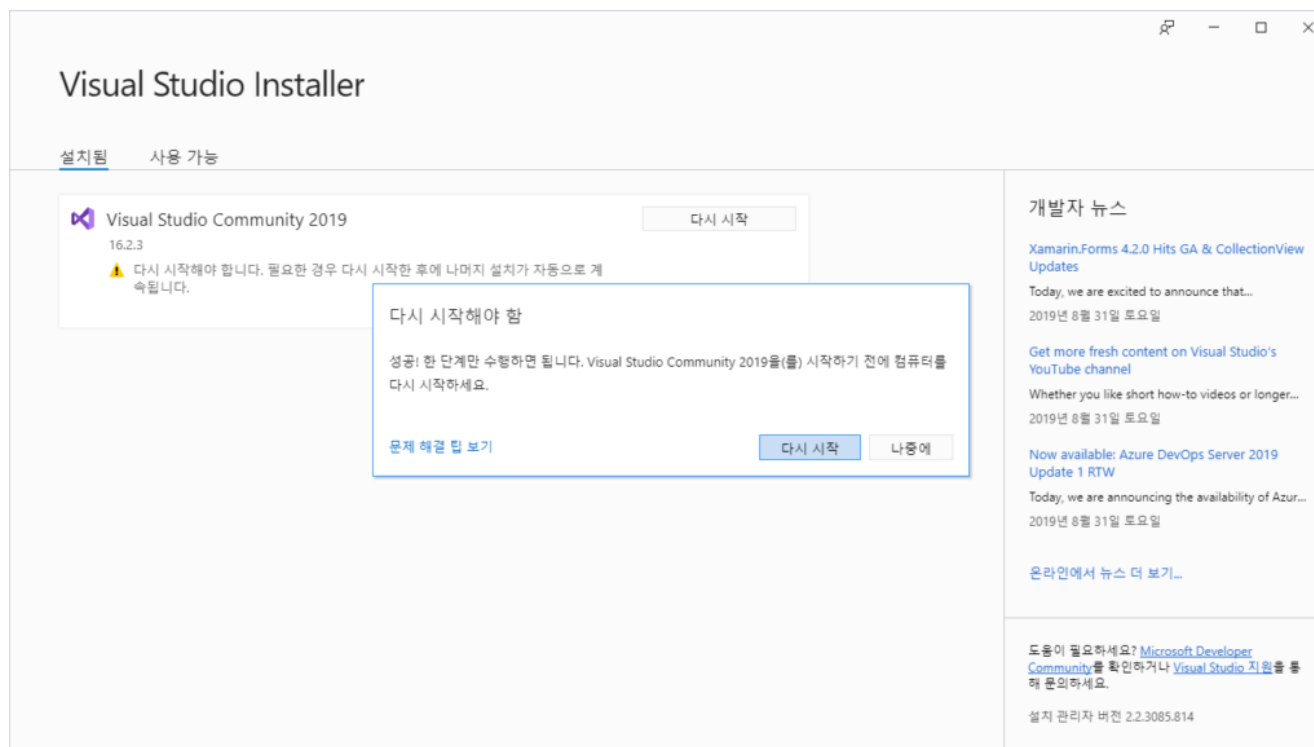
- 설치 과정에서 체크를 더 했거나 덜 한채로 진행했더라도 나중에 “Visual Studio Installer”를 실행시켜 필요 항목을 추가하거나 삭제할 수 있다.
- 설치가 진행된다.



## 2. Visual Studio와 openFrameworks 설치

### ■ Visual Studio 2019 Community 버전 설치

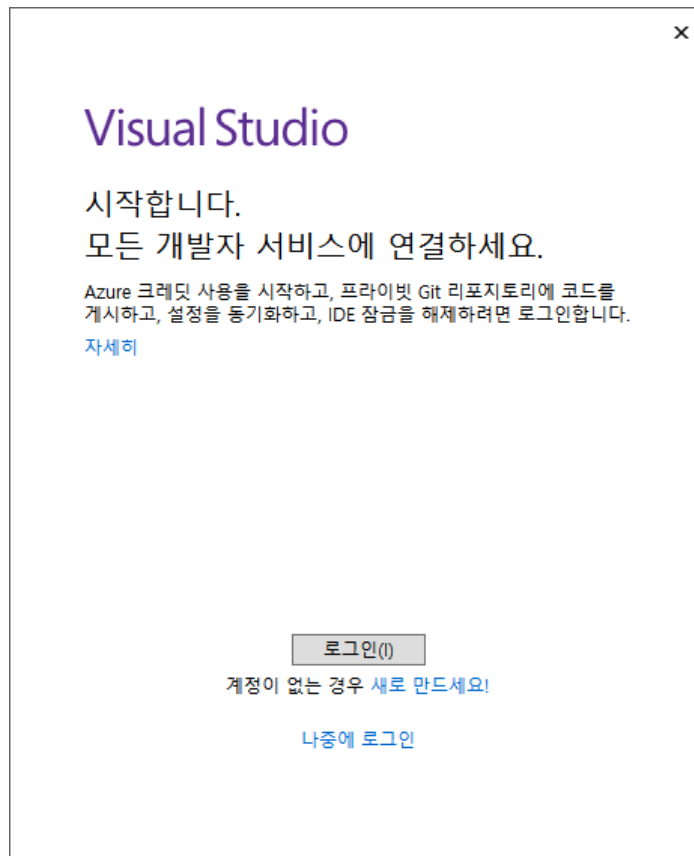
- 설치가 완료되었다.
- 다시 시작해야 한다는 창이 뜬다.
- “다시 시작”을 클릭한다.



## 2. Visual Studio와 openFrameworks 설치

### ■ Visual Studio 2019 Community 버전 설치

- 설치 후, 처음 Visual Studio 2019를 실행하게 되면 아래와 같은 창이 뜬다.
- “나중에 로그인”을 클릭한다.

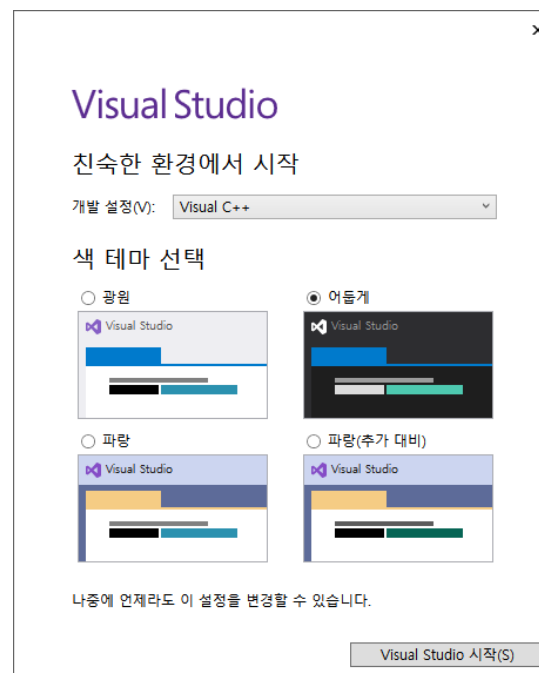




## 2. Visual Studio와 openFrameworks 설치

### ■ Visual Studio 2019 Community 버전 설치

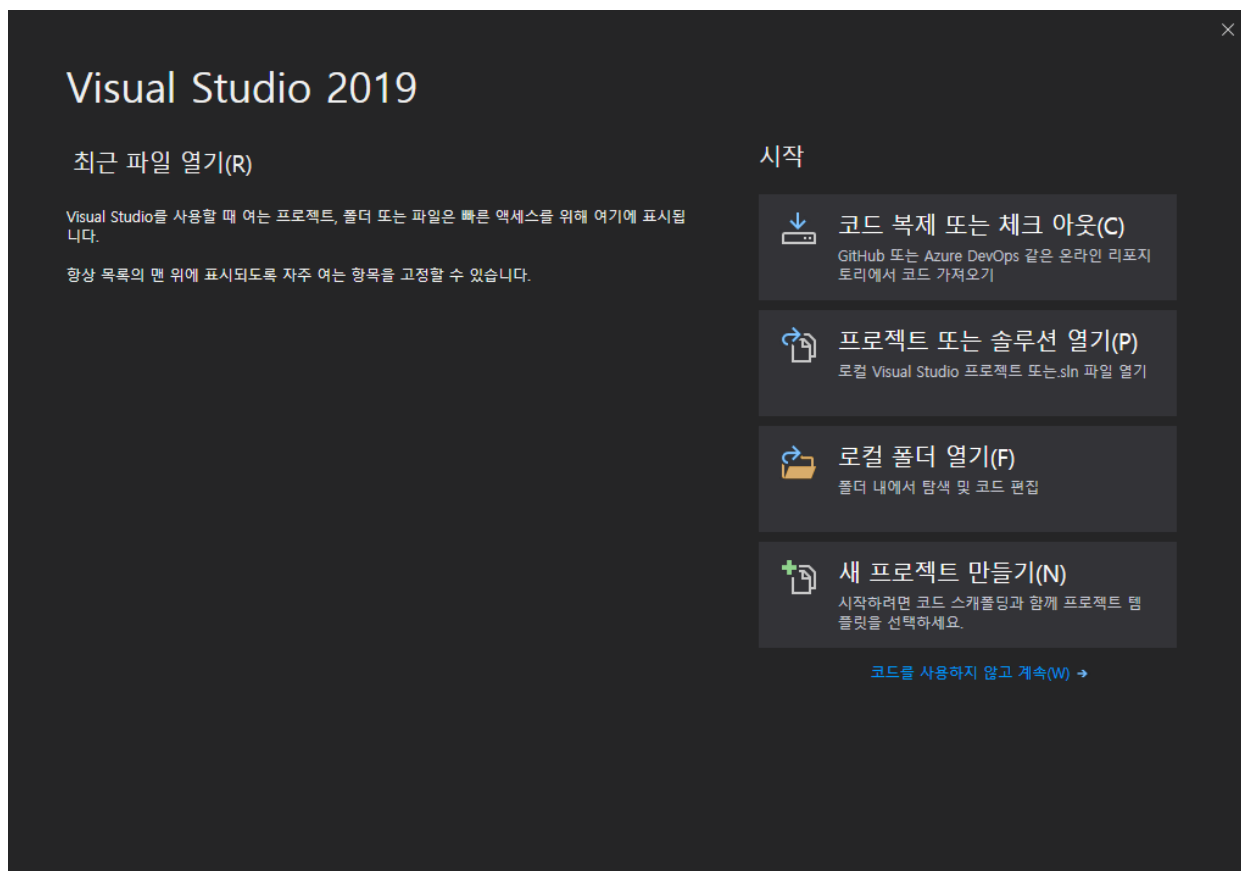
- 위의 창에서 “나중에 로그인”을 누르게 되면 아래와 같은 창이 뜬다.
- 개발 설정 및 색 테마를 선택하는 창이다.
- 개발 설정을 “Visual C++”로 선택한다.
- 여기서 개발 설정을 다르게 선택할 경우, 일부 화면 구성이나 단축키 설정 등이 바뀌게 되니 주의가 필요하다.
- 잘못 선택했더라도 나중에 다시 바꿀 수 있다.
- 색 테마는 기본이 “파랑”으로 되어 있지만, 각자 편한대로 선택하면 된다.
- 마지막으로 “Visual Studio 시작(S)” 버튼을 클릭한다.



## 2. Visual Studio와 openFrameworks 설치

### ■ Visual Studio 2019 Community 버전 설치

- 아래 그림은 비주얼 스튜디오 커뮤니티 2019의 시작 화면이다.



## 2. Visual Studio와 openFrameworks 설치

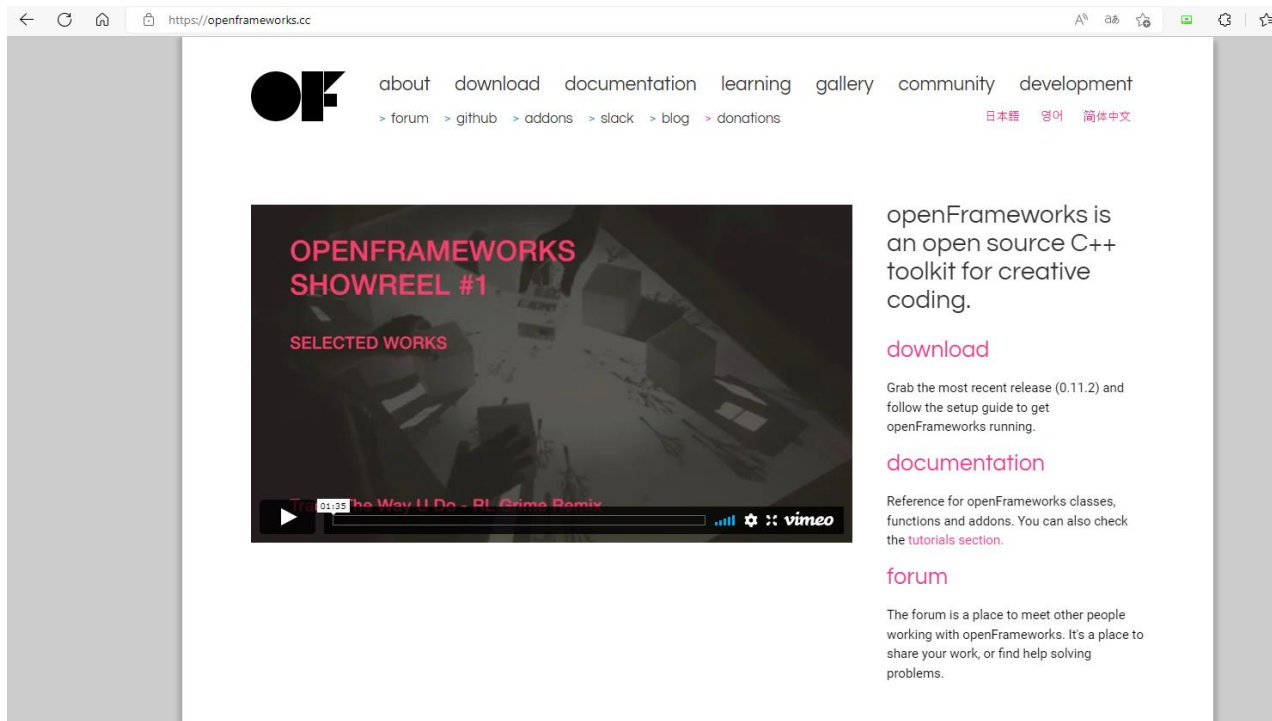
### ■ Visual Studio 2019 Community 버전 설치

- 비주얼 스튜디오 커뮤니티 버전을 무료로 사용하기 위해서는 로그인을 최소한 한번은 해야 한다.
- 로그인 한 후 바로 로그아웃하더라도 계속해서 무료로 사용이 가능하다.
- 따라서 로그인을 한번 하는 것이 좋다.
- 대부분의 대학생들은 대학생에게 무료로 제공되는 MS 오피스용 계정을 가지고 있을 것이다.
- 그 계정으로 로그인하시면 된다.
- 한번도 로그인하지 않고 사용하게 되면, 설치 한 달 후 로그인 창이 강제로 뜨게 되고 로그인 하지 않으면 사용할 수 없게 된다.

## 2. Visual Studio와 openFrameworks 설치

### ■ openFrameworks 설치

- 먼저 openFrameworks를 설치해보자.
- 아래의 사이트에서 openFrameworks의 최신 버전을 다운받을 수 있다.
- <http://openframeworks.cc/>



## 2. Visual Studio와 openFrameworks 설치

### ■ openFrameworks 설치

- download 탭을 클릭한다.
- openFrameworks는 osx, linux, windows, ios, android 등 다양한 운영 체제에서 사용할 수 있다.
- 따라서, 한번의 프로그래밍으로 다양한 운영 체제에서 수행되는 프로그램을 제작할 수 있다.
- 따라서 자신의 운영 체제에 맞는 openFrameworks 파일을 다운받자.
- Windows 운영체제를 사용하는 경우라면 화면의 "windows" 부분의 "visual studio" 링크를 눌러 다운받으면 된다.



0.11.2 is the most recent release. It's a minor version, with bugfixes only, so it is fully compatible with projects developed for 0.11.0 or any newer version. Please see the [changelog](#) to get an overview of the differences between versions.

To use openFrameworks you will need an IDE, and the setup guide for your platform can walk you through this. Please post any bugs on the [issues](#) page, and post to the [forum](#) if you have any other questions. openFrameworks is distributed under the [MIT License](#).  
NOTE: If the Github links below do not work for you, please try the [mirror links here](#).

<p>OSX</p> <p>download openFrameworks for osx</p> <p>setup guides <a href="#">xcode</a> <a href="#">qt creator (experimental)</a> <a href="#">emscripten</a></p>	<p>linux</p> <p>download openFrameworks for linux gcc6 or later</p> <p>setup guides <a href="#">linux install</a> <a href="#">qt creator</a> <a href="#">emscripten</a></p>	<p>windows</p> <p>download openFrameworks for <a href="#">visual studio (2017-2022)</a> <a href="#">qt creator / msys2 32bit</a> <a href="#">qt creator / msys2 64bit</a></p> <p>setup guides <a href="#">visual studio</a> <a href="#">qt creator</a> <a href="#">msys2</a></p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 2. Visual Studio와 openFrameworks 설치

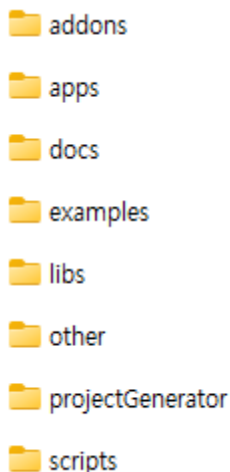
### ■ openFrameworks 설치

- 다운받은 파일의 압축을 풀고, 압축을 푼 폴더의 위치를 잘 기억하자.
- openFrameworks 라이브러리의 위치는 향후에 Visual Studio에서 간단한 추가 설정 작업이 필요하기 때문이다.

## 2. Visual Studio와 openFrameworks 설치

### ■ openFrameworks 폴더 구조

- 이제 openFrameworks의 폴더 구조를 알아보자.
- 압축을 풀면, libs, addons, examples 같은 일련의 하위 폴더들을 발견할 수 있다.
- 각 폴더의 내용은 다음과 같다.

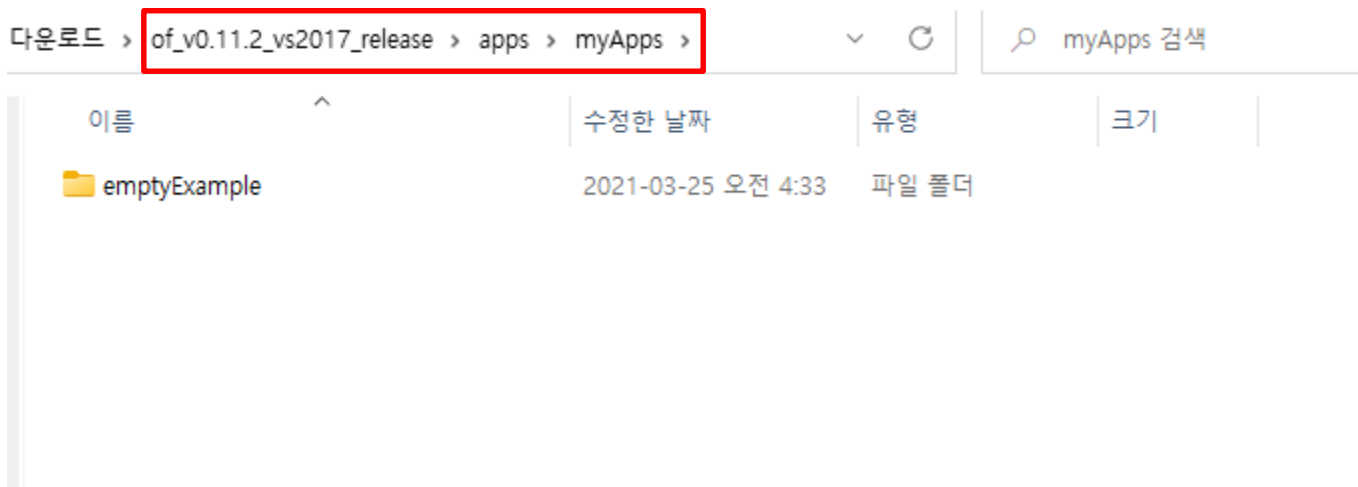


- addons: 오픈프레임웍스(openFrameworks)에서 제공되는 다양한 추가 기능이 담겨 있는 폴더
  - apps: 프로그래머가 직접 작성한 응용 프로그램이 저장되는 폴더. 우리가 작성할 프로그램도 이 폴더에 저장해야 한다.
  - libs: 오픈프레임웍스가 사용하는 코어(core) 라이브러리와 함께 다양한 라이브러리가 있는 폴더
- 오픈프레임웍스를 사용할 때 위의 폴더들의 계층 구조를 유지하는 것이 매우 중요하다.

## 2. Visual Studio와 openFrameworks 설치

### ■ openFrameworks 폴더 구조

- 새로운 프로그래밍을 위해서는 apps 폴더 안의 myApps 폴더에 소스 파일을 저장해야 한다.





## 2. Visual Studio와 openFrameworks 설치

### ■ openFrameworks 폴더 구조

- 이제부터 여러분이 작성하는 프로그램은 아래의 폴더 구조를 가진다.

» apps

• myApps

■ 프로젝트 폴더

• src: 프로그램 소스 프로젝트 파일

• bin: 컴파일된 응용 프로그램

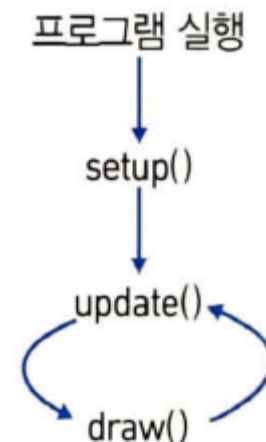
◦ data: 프로그램에서 사용할 사진/폰트 등 저장

- 오픈프레임웍스는 컴파일된 라이브러리 형태로 제공되는 것이 아니라 소스 코드 형태로 제공되기 때문에, 개발하는 소프트웨어가 내부 폴더들과 지역적으로 링크(link)될 수 있도록 구성되어 있다.
- 만약 폴더들을 옮기거나, 변경하게 되면 컴파일이 되지 않을 수 있다.
- 주의하자.

## 2. Visual Studio와 openFrameworks 설치

### ■ openFrameworks 소스 코드 구조

- 오픈프레임웍스를 사용하기 위해서는 몇 가지 규칙을 지켜야 한다.
- 이를 위해서 오픈프레임웍스의 소스 코드 구조를 알아야 한다.
- 오픈프레임웍스에서는 기본적으로 3개의 파일을 사용한다.
  - main.cpp: main.cpp 파일은 C++ 컴파일러에게 어떠한 리소스(resource)를 프로그램 내에서 사용할 것인지를 알려주고, 프로그램을 실행시키는 역할을 한다. 우리는 본 강좌에서 소개되는 내용을 배우기 위해서는 이 파일의 내용을 변경할 필요는 없다.
  - ofApp.cpp와 ofApp.h: 실제 코딩은 ofApp.h와 ofApp.cpp 파일 내에서 이루어지게 된다. 사용자는 기본적으로 ofApp.cpp 파일의 `setup()`, `update()`, `draw()` 함수를 사용할 수 있다. 이 3개의 함수는 오픈프레임웍스에서 미리 정의되어 있고, 우리는 이 함수의 바디를 채우는 방식으로 프로그래밍한다.
    - `setup()`: 프로그램이 실행되는 첫 순간에 한번만 실행되며, 변수 초기화나 업데이트가 필요하지 않은 초기 설정을 하는 부분이다.
    - `update()`: 매 프레임 별로 반복적으로 수행된다. 이 함수는 주로 변수 값을 변경하는 역할을 하며 내부에서 드로잉 함수(그림을 그리는 함수)를 호출할 수 없다.
    - `draw()`: 매 프레임 별로 반복적으로 수행되며 매 프레임에서 그림을 그리는 명령어들을 포함한다.
- 이 3개의 함수는 옆의 그림 같은 순서로 동작한다.
- 즉, 프로그램이 실행되면, `setup()` 함수가 1번 수행되고, 그 다음에 `update()` 함수와 `draw()` 함수가 반복적으로 수행된다.



## 2. Visual Studio와 openFrameworks 설치

### ■ openFrameworks 소스 코드 구조

- 오픈프레임웍스를 사용하기 위해서는 몇 가지 규칙을 지켜야 한다.
- 이를 위해서 오픈프레임웍스의 소스 코드 구조를 알아야 한다.
- 오픈프레임웍스에서는 기본적으로 3개의 파일을 사용한다.
  - main.cpp: main.cpp 파일은 C++ 컴파일러에게 어떠한 리소스(resource)를 프로그램 내에서 사용할 것인지를 알려주고, 프로그램을 실행시키는 역할을 한다. 우리는 본 강좌에서 소개되는 내용을 배우기 위해서는 이 파일의 내용을 변경할 필요는 없다.
  - ofApp.cpp와 ofApp.h: 실제 코딩은 ofApp.h와 ofApp.cpp 파일 내에서 이루어지게 된다. 사용자는 기본적으로 ofApp.cpp 파일의 setup(), update(), draw() 함수를 사용할 수 있다. 이 3개의 함수는 오픈프레임웍스에서 미리 정의되어 있고, 우리는 이 함수의 바디를 채우는 방식으로 프로그래밍한다.
    - setup(): 프로그램이 실행되는 첫 순간에 한번만 실행되며, 변수 초기화나 업데이트가 필요하지 않은 초기 설정을 하는 부분이다.
    - update(): 매 프레임 별로 반복적으로 수행된다. 이 함수는 주로 변수 값을 변경하는 역할을 하며 내부에서 드로잉 함수(그림을 그리는 함수)를 호출할 수 없다.
    - draw(): 매 프레임 별로 반복적으로 수행되며 매 프레임에서 그림을 그리는 명령어들을 포함한다.
  - 이 3개의 함수는 옆의 그림 같은 순서로 동작한다.
  - 즉, 프로그램이 실행되면, setup() 함수가 1번 수행되고, 그 다음에 update() 함수와 draw() 함수가 반복적으로 수행된다.

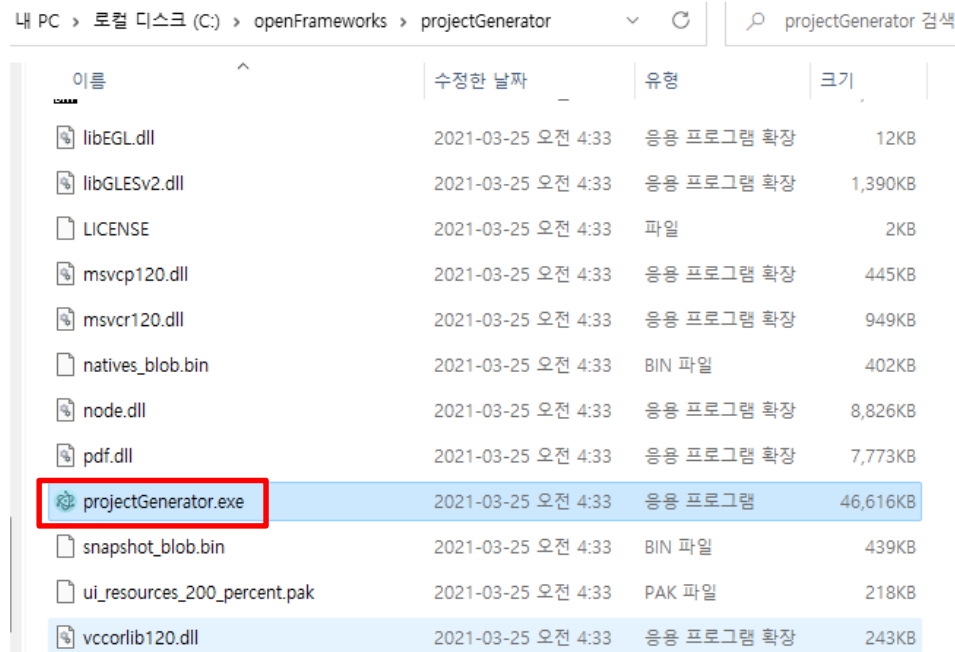
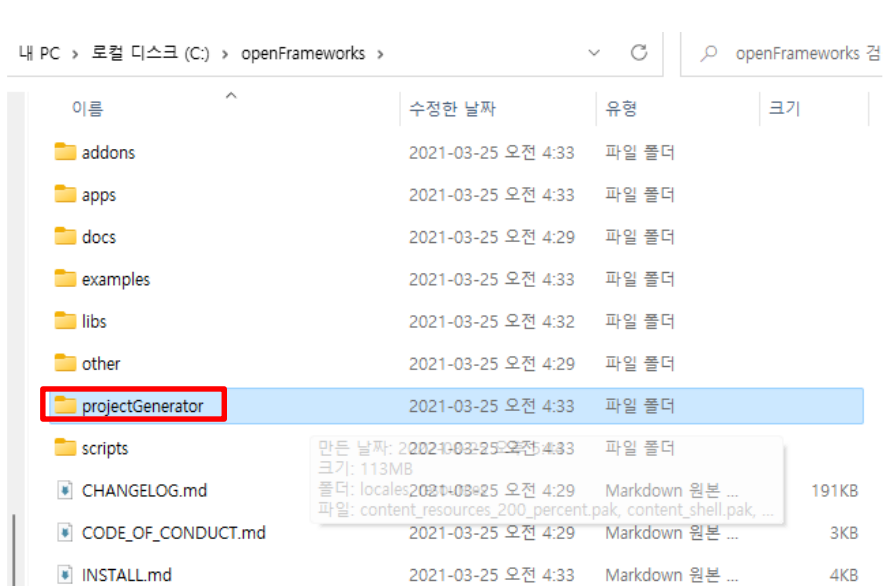
### 3. openFrameworks 시작

- 오픈프레임웍스 라이브러리를 이용하여 Visual Studio에서 프로그래밍을 해보자
- 본격적인 그래픽 프로그래밍이다.
- 아래의 3가지 단계로 프로그래밍을 시작하자.
  - 오픈프레임웍스를 이용해서 empty 프로그램을 작성한다. 아무 일도 하지 않는 프로그램이다.
  - 오픈프레임웍스를 이용하여 텍스트를 출력하는 프로그램을 작성한다.
  - 오픈프레임웍스를 이용하여 그래픽 상태에서 원을 하나 그린다.

### 3. openFrameworks 시작

#### ■ openFrameworks 시작 #1: empty 프로그램 제작

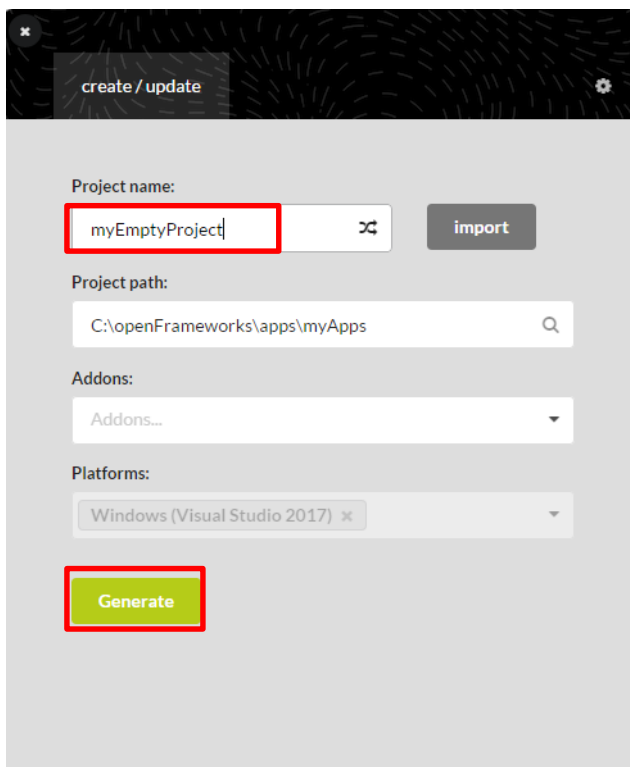
- 이제는 openFrameworks를 이용해서 C 프로그래밍을 시작해보자.
- openFrameworks 압축을 푼 폴더에서 projectGenerator 폴더를 들어간다.
- projectGenerator 폴더에서 projectGenerator.exe 파일을 실행한다.



# 3. openFrameworks 시작

## ■ openFrameworks 시작 #1: empty 프로그램 제작

- Project name에 프로젝트 이름을 입력한다.
- Generate를 클릭한다.
- Open in IDE를 클릭한다.



create / update

Project name:  
myEmptyProject

Project path:  
C:\openFrameworks\apps\myApps

Addons:  
Addons...

Platforms:  
Windows (Visual Studio 2017)

Generate

Success!

You can now find your project in

[C:\openFrameworks\apps\myApps\myEmptyProject](#)

```
[notice ] -----  
[notice ] setting OF path to: C:\openFrameworks  
[notice ] from -o option  
[notice ] target platform is: vs  
[notice ] project path is: C:\openFrameworks\apps\myApps\myEmptyProject  
[notice ] setting up new project C:\openFrameworks\apps\myApps\myEmptyProject  
[notice ] saving addons.make
```

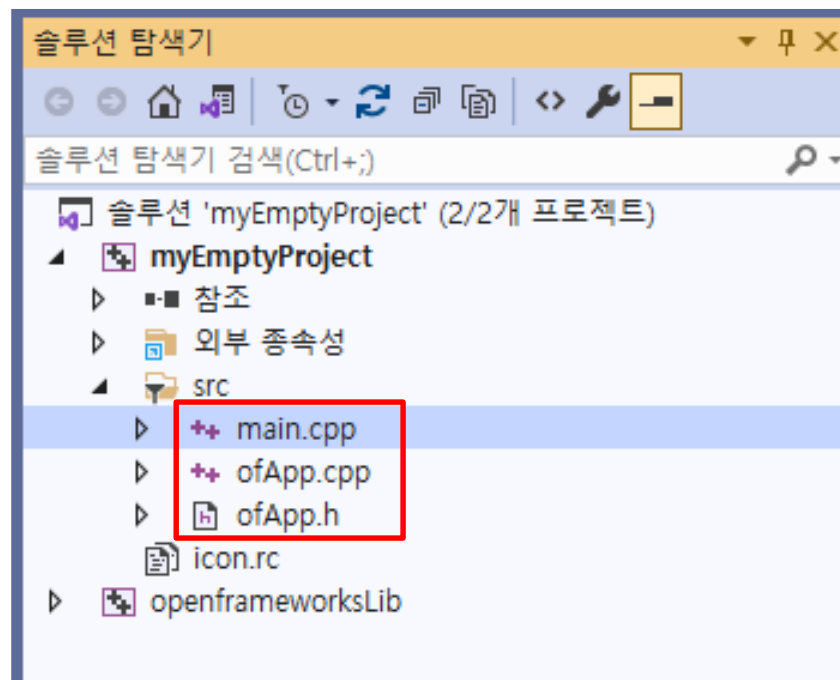
Open in IDE

Close

### 3. openFrameworks 시작

#### ■ openFrameworks 시작 #1: empty 프로그램 제작

- 이제 코드 마법사 기능에 의해서 기본 프로그램이 구성되었다.
- 화면 오른쪽의 [솔루션 탐색기] 장에서, [src]라는 '폴더를 클릭하면 3개의 파일(main.cpp, ofApp.cpp, ofApp.h)을 볼 수 있다.
- 각 파일을 클릭하면 소스 코드를 확인할 수 있다.



### 3. openFrameworks 시작

#### ■ openFrameworks 시작 #1: empty 프로그램 제작

- 실행해보자.
- 메뉴에서 [빌드] → [솔루션 빌드]를 클릭한다.
- 컴파일에서 오류가 없으면 [디버그] → [디버그하지 않고 시작] 메뉴로 프로그램을 실행하자.
- 그림처럼 빈 화면이 출력되면 성공이다.
- 지금까지는 빈 프로그램(empty program)을 만들었다.
- 이제 이 비어있는 화면에 그림을 그리면 된다.

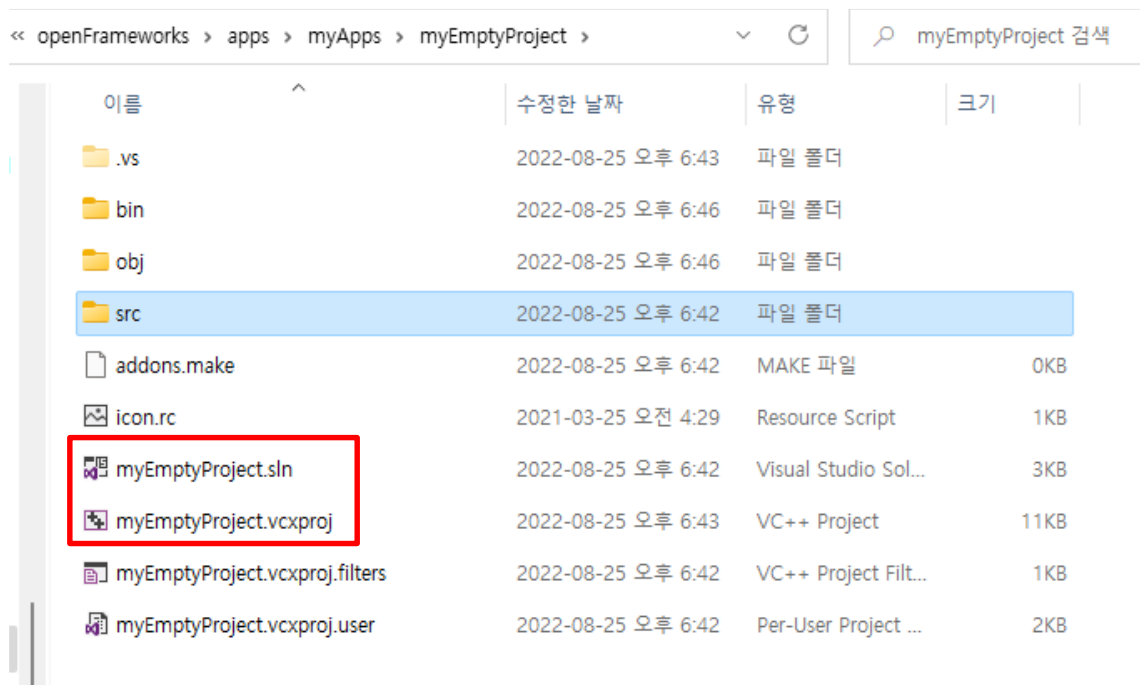




### 3. openFrameworks 시작

#### ■ openFrameworks 시작 #1: empty 프로그램 제작

- 소스 파일이 저장된 폴더의 구조를 다시 한 번 더 설명하자.
  - apps > myApps > 솔루션 폴더 >
- 솔루션 폴더에는 확장자가 \*.sln인 솔루션 파일(파일 유형이 “Microsoft Visual Studio Solution”)이 있다.
- 향후에 이 파일을 더블 클릭하면 Visual Studio를 이용해서 프로그램을 수정할 수 있다.
- 해당 폴더 내부에는 솔루션 이름과 동일한 이름의 프로젝트 파일(\*.vcxproj)이 있다(파일 유형이 “VC++ Project”).



이름	수정된 날짜	유형	크기
.vs	2022-08-25 오후 6:43	파일 폴더	
bin	2022-08-25 오후 6:46	파일 폴더	
obj	2022-08-25 오후 6:46	파일 폴더	
src	2022-08-25 오후 6:42	파일 폴더	
addons.make	2022-08-25 오후 6:42	MAKE 파일	0KB
icon.rc	2021-03-25 오전 4:29	Resource Script	1KB
myEmptyProject.sln	2022-08-25 오후 6:42	Visual Studio Sol...	3KB
myEmptyProject.vcxproj	2022-08-25 오후 6:43	VC++ Project	11KB
myEmptyProject.vcxproj.filters	2022-08-25 오후 6:42	VC++ Project Filt...	1KB
myEmptyProject.vcxproj.user	2022-08-25 오후 6:42	Per-User Project ...	2KB

### 3. openFrameworks 시작

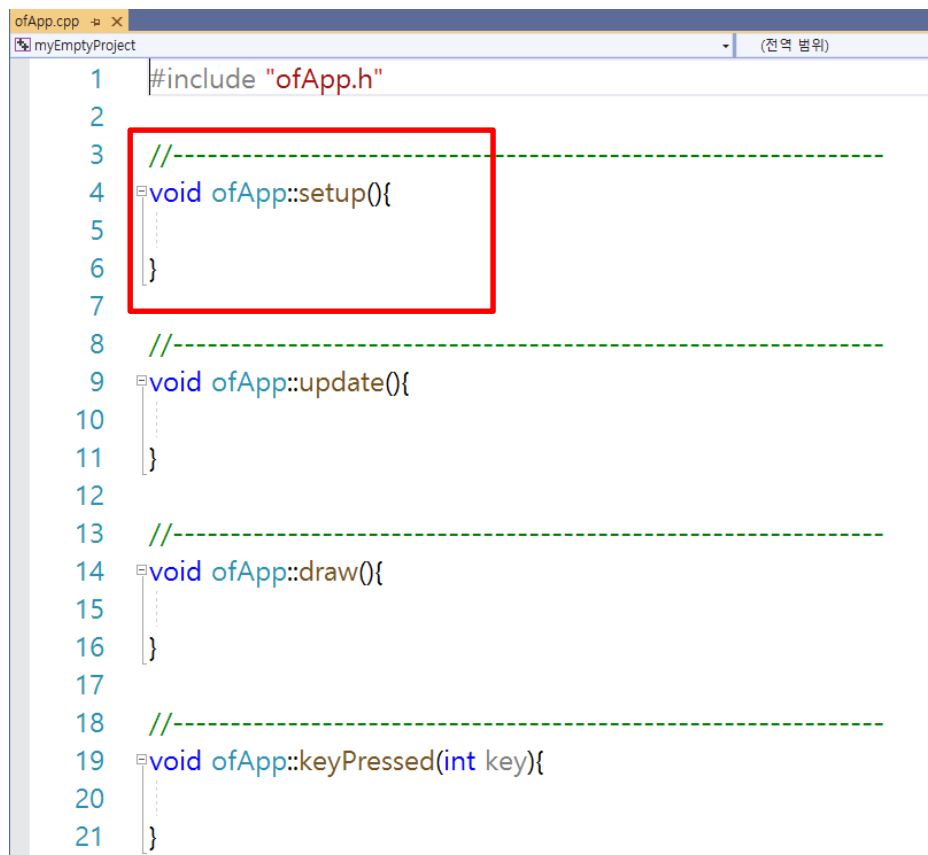
#### ■ openFrameworks 시작 #2: 텍스트 모드 프로그램 제작

- 스무 고개' 프로그램을 openFrameworks를 이용해서 구현해보자.
- 콘솔 화면에서 수행되는 프로그램을 작성해 볼 것이다.
- 물론 앞에서 이야기했던 Visual Studio에서의 콘솔 프로그래밍 방식을 사용해도 되지만, 여기에서는 openFrameworks를 사용하는 환경에서의 콘솔 프로그래밍을 소개한다.
- 이제부터는 주의하자!
- openFrameworks는 기본적으로 객체 지향 언어인 C++ 언어를 위한 라이브러리이다.
- 그래서 중간 중간에 C++ 언어와 관련된 설명이 나온다.
- C 언어를 배우는 프로그래머의 대부분은 차후에 C++ 언어를 배우게 되는데, 아직 C++을 배우지 않은 사람을 위해서 필요할 경우 C++에 대한 기초적 지식은 조금씩 별도로 설명한다.

### 3. openFrameworks 시작

#### ■ openFrameworks 시작 #2: 텍스트 모드 프로그램 제작

- 지금까지 작성했던 empty 프로그램을 이용하자.
- Visual Studio 화면 우측의 [솔루션 탐색기]에서 ofApp.cpp 파일을 클릭해서 열어보자.



```
1 #include "ofApp.h"
2
3 //-----
4 void ofApp::setup(){
5
6 }
7
8 //-----
9 void ofApp::update(){
10
11 }
12
13 //-----
14 void ofApp::draw(){
15
16 }
17
18 //-----
19 void ofApp::keyPressed(int key){
20
21 }
```

### 3. openFrameworks 시작

#### ■ openFrameworks 시작 #2: 텍스트 모드 프로그램 제작

- ofApp.cpp 파일 내부의 setup( ) 함수를 아래와 같이 수정해서 실행해보자.
- 이 프로그램은 앞에서 작성했던 명령어 모드에서의 프로그램(웹 컴파일러를 사용한 경우)과 동일하다.
- 이와 같이 마우스나 키보드 등을 이용한 인터랙션이 빈번하지 않은 프로그램은 setup( ) 함수 내에 코딩하면 된다.

```
3  //-----
4  void ofApp::setup(){
5      int opponent = 72;
6      int guess;
7
8      do {
9          std::printf("Guess what between 1 to 100 ? \n");
10         std::scanf("%d", &guess);
11         if (guess == opponent) {
12             std::printf("Your guess %d is Correct!\n", guess);
13         }
14         else {
15             std::printf("Your guess %d is Incorrect!\n", guess);
16         }
17     } while (guess != opponent);
18 }
```

### 3. openFrameworks 시작

#### ■ openFrameworks 시작 #2: 텍스트 모드 프로그램 제작

- 코딩을 완료하면, [빌드] → [솔루션 빌드]하고 [디버그] → [디버그하지 않고 시작]을 클릭해서 실행하자.
- 아래는 콘솔 실행 화면이다.
- 스무 고개 게임을 위해서 숫자를 입력하고 엔터키를 눌러보자.

```
C:\openFrameworks\apps\myApps\myEmptyProject\bin\myEmptyProject_debug.exe
Guess what between 1 to 100 ?
44
Your guess 44 is Incorrect!
Guess what between 1 to 100 ?
72
Your guess 72 is Correct!
```

### 3. openFrameworks 시작

#### ■ openFrameworks 시작 #2: 텍스트 모드 프로그램 제작

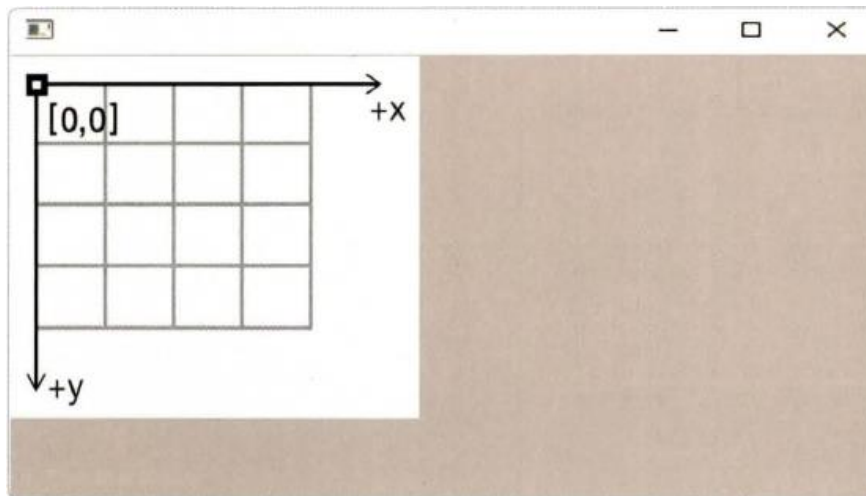
- 콘솔 프로그램이 끝나면 하단의 그래픽 화면도 같이 출력되지만 아직은 이 그래픽 화면에는 출력 내용이 없다.



# 3. openFrameworks 시작

## ■ openFrameworks 시작 #3: 그래픽 모드 프로그램 제작

- 이제 비로써 그래픽 출력을 시작한다.
- 화면에 도형 몇 개를 그려볼 것이다.
- 이를 위해서 먼저 그래픽 화면의 좌표에 대한 설명을 하자.
- 좌표계
  - openFrameworks에서 사용하는 좌표계는 아래 그림과 같다.
  - 화면의 왼쪽 상단이 기준점(원점)이다.
  - 좌표값은 0에서 시작한다.
  - 즉, 출력 화면의 왼쪽 상단이 원점으로 써 좌표가  $[0, 0]$ 이 된다.
  - 가로가  $x$ , 세로 방향이  $y$ 축이다.
  - 즉  $[120, 240]$  크기의 윈도우인 경우는  $[0, 0]$ 에서 시작해서 가로 세로가 각각  $[319, 239]$ 까지이다.



# 3. openFrameworks 시작

## ■ openFrameworks 시작 #3: 그래픽 모드 프로그램 제작

### ■ 좌표계

- openFrameworks를 이용한 기본적인 프로젝트 설정 작업을 마치면, 솔루션 탐색기에서 아래의 3개의 파일을 확인할 수 있다.
  - main.cpp
  - ofApp.cpp
  - ofApp.h

### ■ main.cpp

- main.cpp는 아래와 같다.
- 본 강좌에서는 변경할 필요가 없기 때문에 main.cpp 파일이 있구나 ... 정도로만 알고 넘어가자.

```
1 #include "ofMain.h"
2 #include "ofApp.h"
3
4 //=====
5 int main() {
6     ofSetupOpenGL(1024,768,OF_WINDOW);    // <----- setup the GL context
7
8     // this kicks off the running of my app
9     // can be OF_WINDOW or OF_FULLSCREEN
10    // pass in width and height too:
11    ofRunApp(new ofApp());
12
13 }
14
```



# 3. openFrameworks 시작

## ■ openFrameworks 시작 #3: 그래픽 모드 프로그램 제작

### ■ ofApp.cpp

- ofApp.cpp 파일의 setup( )과 draw( ) 함수를 아래와 같이 수정해보자.
- update( ) 함수는 수정하지 않는다.

```
3  //-----
4  void ofApp::setup(){
5      ofSetWindowTitle("Sample Progame #1"); // 실행 화면의 상단 타이틀 바의 문자열 설정
6      ofSetWindowShape(1024, 768);           // 출력창의 크기 설정
7      ofSetFrameRate(60);                     // 화면의 업데이트 속도
8      ofBackground(ofColor::white);           // 화면의 배경색 설정
9      ofSetColor(ofColor::black);             // 화면의 도형색 설정
10     ofSetLineWidth(1.0);                    // 선 굵기 설정
11 }
12
13 //-----
14 void ofApp::update(){
15
16
17 }
```

# 3. openFrameworks 시작

## ■ openFrameworks 시작 #3: 그래픽 모드 프로그램 제작

### ■ ofApp.cpp

- ofApp.cpp 파일의 setup( )과 draw( ) 함수를 아래와 같이 수정해보자.
- update( ) 함수는 수정하지 않는다.

```
19  //-----
20  void ofApp::draw(){
21      // (0, 0)과 (700, 700) 사이의 직선 그리기
22      ofLine(0, 0, 700, 700);
23
24      // 중심이 (100, 100)이고 반지름이 100인 원 그리기
25      ofCircle(100, 100, 100);
26
27      // 좌측 상단 꼭지점이 (500, 100)이고, 가로 폭이 200, 세로 높이가 200인 사각형 그리기
28      ofRect(500, 100, 200, 200);
29
30      // 꼭지점이 (300, 400), (200, 600), (50, 600)인 삼각형 그리기
31      ofTriangle(300, 400, 200, 600, 50, 600);
32
33  }
```

# 3. openFrameworks 시작

## ■ openFrameworks 시작 #3: 그래픽 모드 프로그램 제작

### ■ ofApp.h

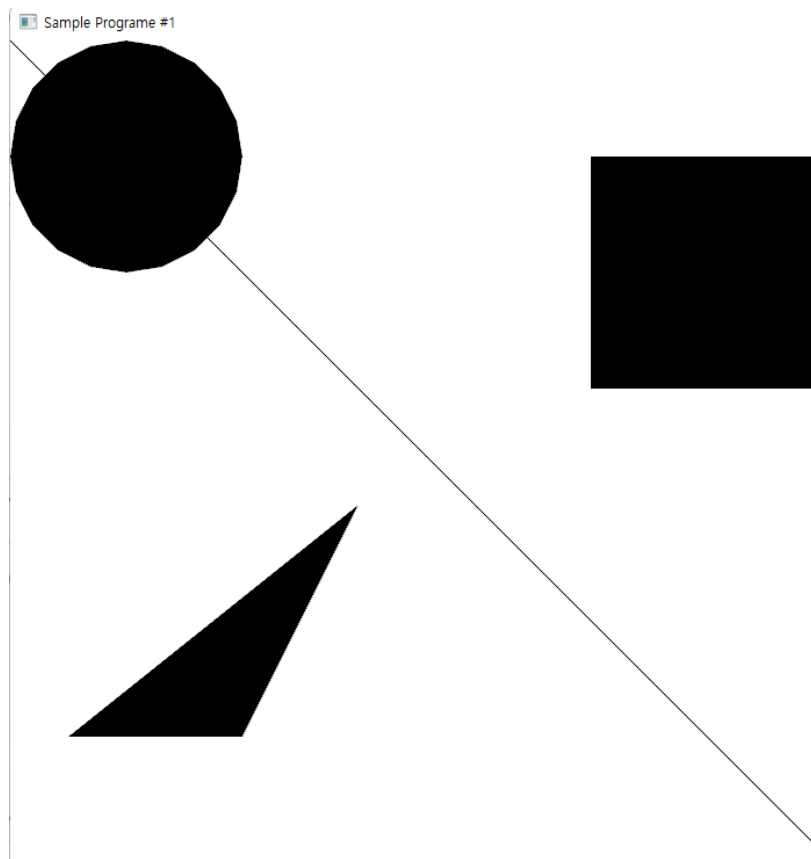
- 본 강좌에서는 ofApp.h 파일도 변경할 필요가 없다.
- 아래 그림처럼 ofApp.cpp 파일에서 사용하는 함수들이 선언되어 있다.

```
1  #pragma once
2
3  #include "ofMain.h"
4
5  class ofApp : public ofBaseApp{
6
7      public:
8          void setup();
9          void update();
10         void draw();
11
12         void keyPressed(int key);
13         void keyReleased(int key);
14         void mouseMoved(int x, int y );
15         void mouseDragged(int x, int y, int button);
16         void mousePressed(int x, int y, int button);
17         void mouseReleased(int x, int y, int button);
18         void mouseEntered(int x, int y);
19         void mouseExited(int x, int y);
20         void windowResized(int w, int h);
21         void dragEvent(ofDragInfo dragInfo);
22         void gotMessage(ofMessage msg);
23
24     };
25
```

### 3. openFrameworks 시작

#### ■ openFrameworks 시작 #3: 그래픽 모드 프로그램 제작

- 코딩을 완료하면, [빌드] → [솔루션 빌드]하고 [디버그] → [디버그하지 않고 시작]을 클릭해서 실행하자.
- 아래는 콘솔 실행 화면이다.





Thank You

---