
컴퓨터 공학 설계 및 실험 | openFrameworks

6주차 실험

Introduction To openFrameworks (oF)

□ openFrameworks (oF) 에 대한 소개

- oF 는 C++ 를 기반으로 한 오픈 소스 라이브러리로, “창의적인 코딩”을 목적으로 설계되었다.
- 아래의 공용 라이브러리들을 포함하여 범용적인 “접착제” 역할을하도록 디자인되었다.
 - 그래픽: OpenGL, GLEW, GLUT etc.
 - 오디오 입력, 출력 및 분석: rtAudio, PortAudio, OpenAL etc.
 - 이미지 불러오기 및 저장: FreeImage
 - 컴퓨터 비전: OpenCV
 - 3D 모델 불러오기: Assimp
- 또한, 크로스플랫폼을 지원한다. 현재는 Windows, Mac OSX, Linux 등에 대한 IDE (Visual Studio, Xcode, Eclipse, Code::Blocks) 를 지원한다.
- <https://openframeworks.cc/>

First Things First

- openFrameworks (oF) 기반의 프로젝트를 만들기 위한 Integrated Development Environment (IDE) 가 필요하다.
 - IDE 는 코드를 쓰고, 컴파일, 실행 그리고 디버깅까지 할 수 있는 소프트웨어의 일종이다.
 - Visual Studio (for Windows)
 - Xcode (for Mac OSX)
 - 대부분의 IDE 들은 비슷한 인터페이스를 갖는다.

Xcode

- Xcode 는 Apple 사의 IDE 로 iOS 에서 쓰인다.

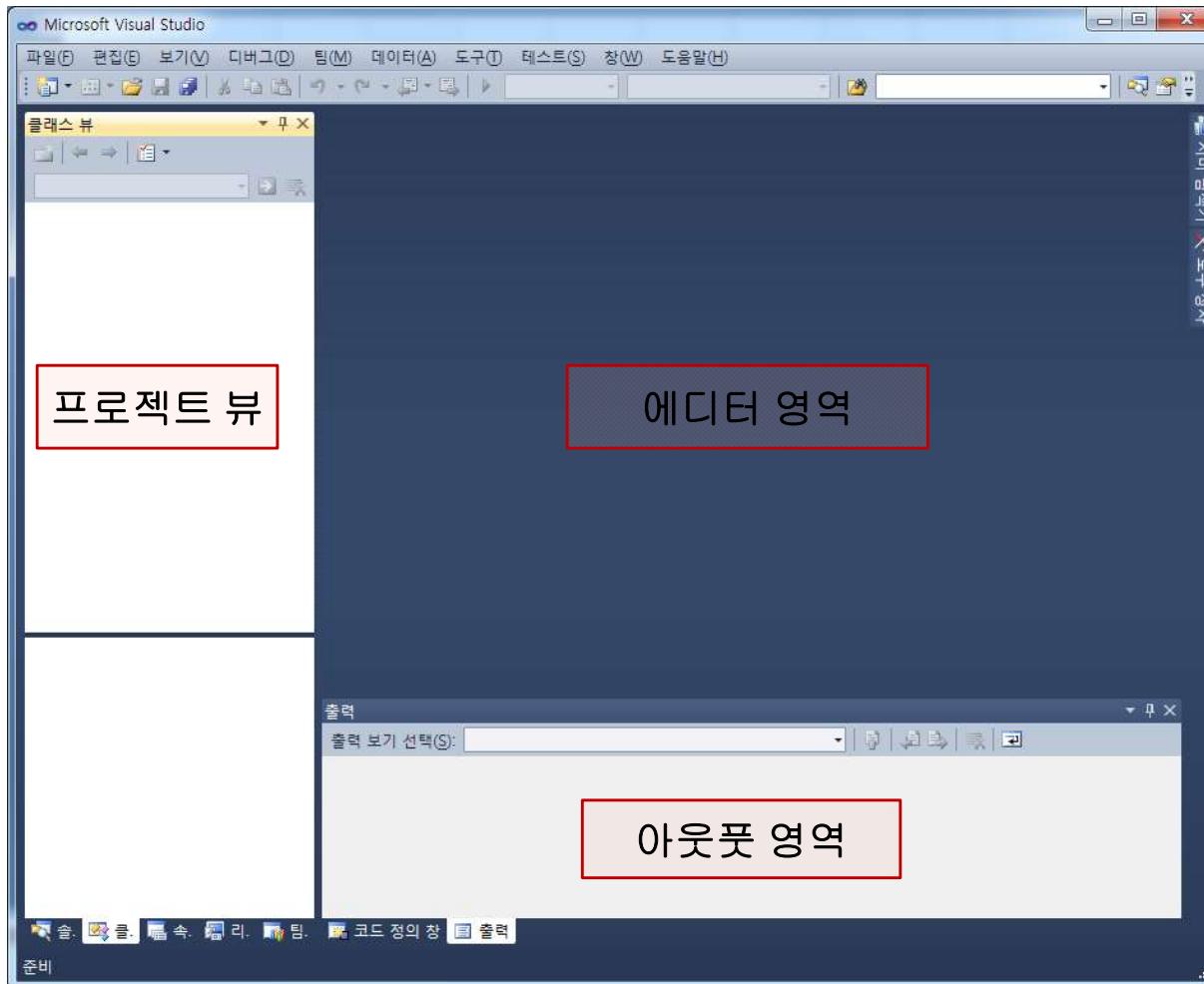
The screenshot shows the Xcode interface with the following details:

- Project Structure:** The left sidebar shows a project named "Kitura-Starter" with files like Package.swift, Sources, and main.swift.
- Editor:** The main area displays the contents of main.swift:

```
1 /**
2 * Copyright IBM Corporation 2016,2017
3 *
4 * Licensed under the Apache License, Version 2.0 (the "License");
5 * you may not use this file except in compliance with the License.
6 * You may obtain a copy of the License at
7 *
8 * http://www.apache.org/licenses/LICENSE-2.0
9 *
10 * Unless required by applicable law or agreed to in writing, software
11 * distributed under the License is distributed on an "AS IS" BASIS,
12 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 * See the License for the specific language governing permissions and
14 * limitations under the License.
15 */
16
17 import Foundation
18 import Kitura
19 import LoggerAPI
20 import SwiftMetrics
21 import SwiftMetricsDash
22
23 // Start SwiftMetrics
24 let sm = try SwiftMetrics()
25 let monitor = sm.monitor()
26
27 // Set up monitoring for latency data
28 func processLatency(latency: LatencyData) {
29     print("Received a latency event")
30 }
31 monitor.on(processLatency)
32
33 // Start the dashboard
34 let router = Router()
35 let smd = try SwiftMetricsDash(swiftMetricsInstance: sm, endpoint: router)
36 Kitura.addHTTPServer(onPort: 8080, with: router)
37 Kitura.run()
```
- Output Window:** The bottom pane shows the console output with the message "Received a latency event" repeated five times.

Visual Studio (VS)

- Visual Studio 는 Microsoft 사의 IDE 로 Windows 개발에 쓰임.



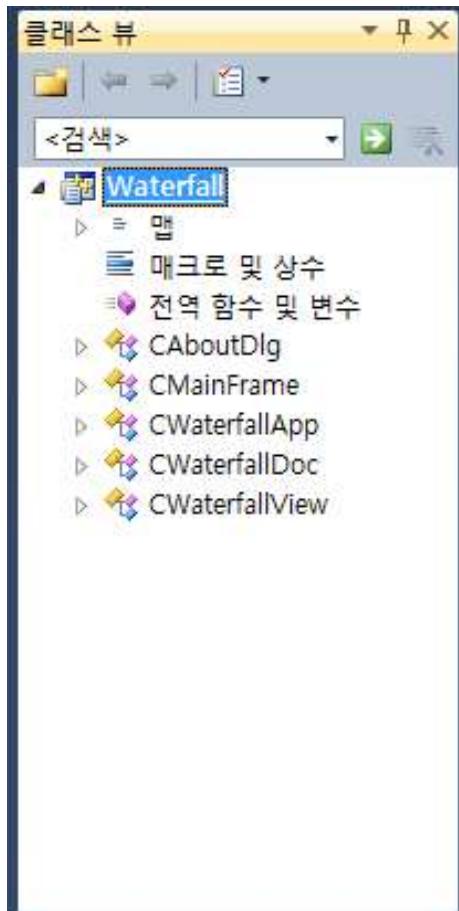
Construction for Visual Studio

□ 프로젝트 뷰 영역

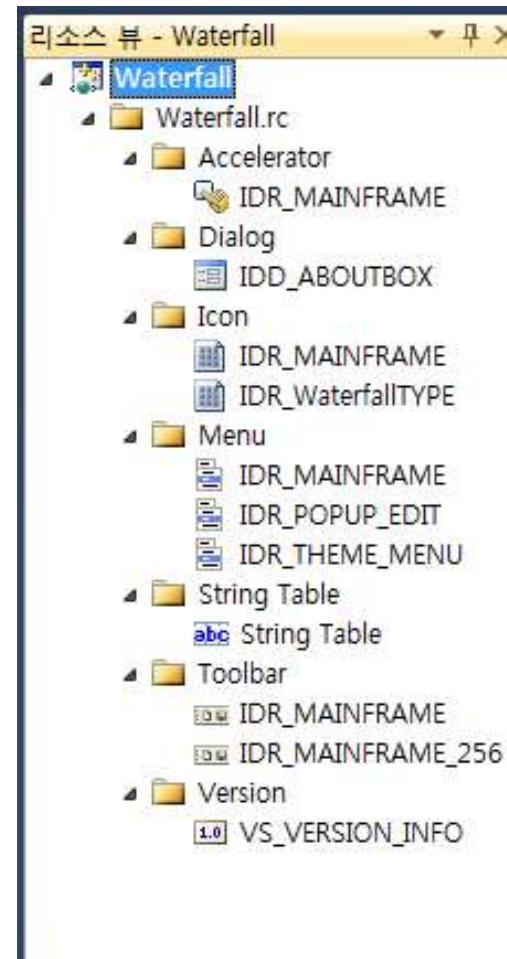
- 프로젝트를 구성하고 있는 여러 요소들을 일관성 있게 살펴볼 수 있도록 해준다.
- 아래의 세 가지 형태의 뷰를 제공한다.
 - 클래스 뷰 (Class view): 소스 코드를 C++ 클래스 수준에서 살펴보고 조작할 수 있게 해준다.
 - 리소스 뷰 (Resource view): 애플리케이션에서 사용하는 여러 가지 리소스인 다이얼로그 박스, 아이콘, 메뉴 등을 찾아서 편집할 수 있도록 해준다.
 - 파일 뷰 (File view): 애플리케이션을 만드는 데 필요한 모든 파일을 찾아볼 수 있게 해준다.

Construction for Visual Studio (Cont')

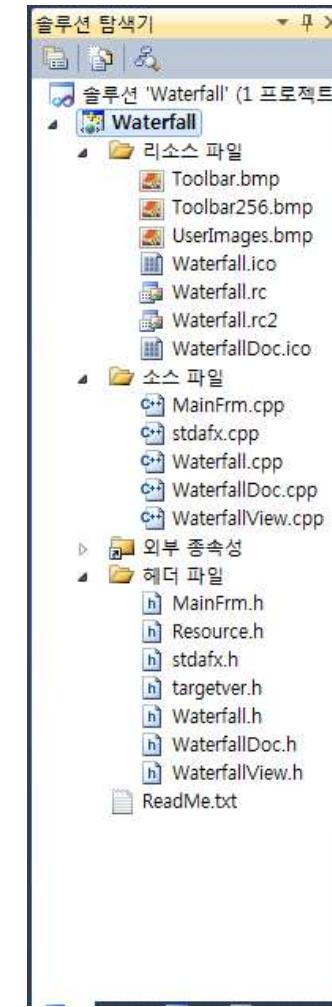
□ 프로젝트 뷰 영역의 세 가지 내용



Class View



Resource View



Solution Explorer

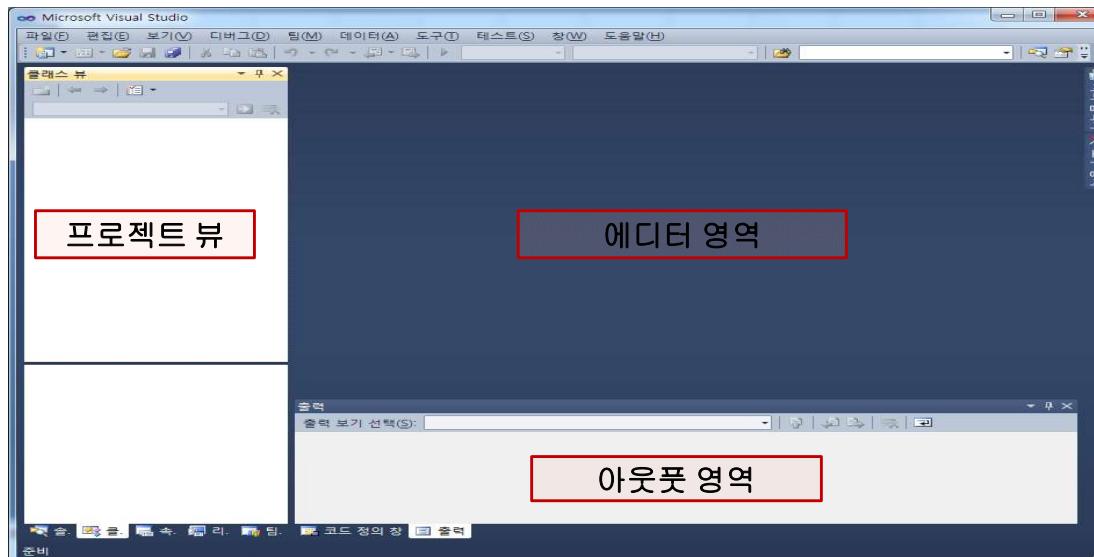
Construction for Visual Studio (Cont')

□ 에디터 뷰 영역

- 필요한 모든 편집을 할 수 있는 곳

□ 아웃풋 영역

- 컴파일 진행 과정 시 발생하는 경고 메시지, 에러 메시지 등을 볼 수 있다.
- 디버깅을 해주는 Debugger 를 실행시켰을 경우에는 실행중인 모든 변수들의 값도 볼 수 있다.



Construction for Visual Studio (Cont')

□ Visual Studio (VS) 생성파일

- VS로 개발되는 프로그램은 프로젝트 단위로 관리한다.
- 프로젝트
 - 하나의 프로그램을 만들 때 필요한 모든 구성 요소들에 대한 정보를 갖는다.
 - 어떤 파일들이 컴파일되고 링크되어 실행 파일을 만들어 낼 것인가에 대한 정보를 갖는다.
 - 프로젝트 파일: vsproj
- 솔루션
 - 프로젝트 및 워크스페이스가 한 단계 발전된 형태
 - 프로젝트의 모든 정보를 포함함과 동시에 작업 윈도우의 여러 가지 설정 상태 및 원도우의 위치 등 개발 환경에 대한 정보까지 갖는다.
 - 솔루션 파일: sln

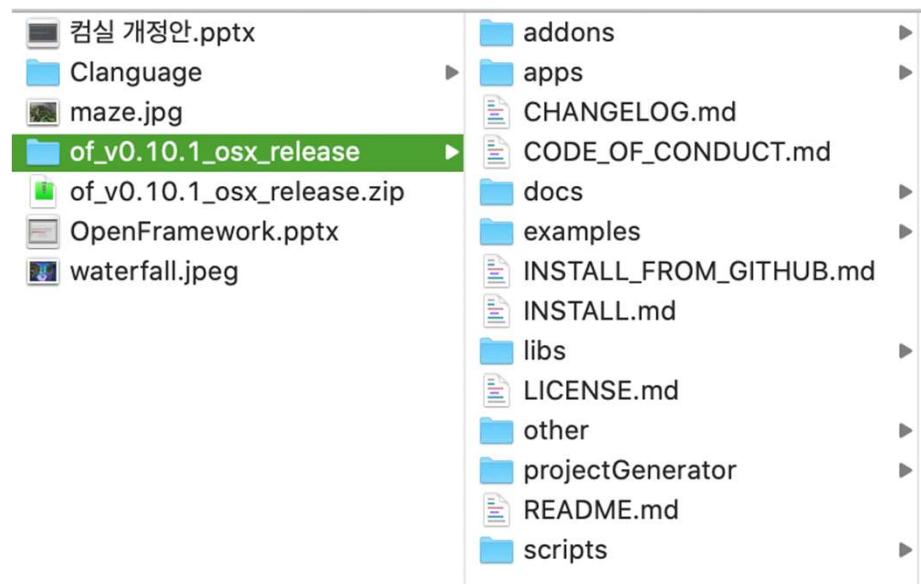
Construction for Visual Studio (Cont')

- Visual Studio (VS) 생성파일
 - 프로젝트를 구성하고 있는 관련 파일들

파일 확장자	설명
.h	헤더 파일 (클래스 정의 파일)
.cpp / .c	소스 파일
.clw	Class Wizard에서 사용되는 정보를 보관하는 파일
.sln	솔루션 파일
.dsp	프로젝트 파일
.rc	Resource가 정의된 파일
.pch	프리 컴파일 헤더 파일
.pdb	프로그램 정보 파일
.res	.rc 파일이 컴파일된 바이너리 파일
Res 폴더	리소스에 사용되는 이미지를 비롯한 기타 등등의 파일이 저장된 폴더
Debug 폴더	컴파일 후에 생성되는 파일(.obj, .pch, .exe 등)이 들어 있는 폴더

oF Folder Structure

- openFramework (oF) 에서 제공하는 프레임워크 다운로드 및 앱 축 (<https://openframeworks.cc/download/>)
 - 프레임워크 파일 폴더는 oF root 폴더라고 명명한다.
- oF root folder 는 다음과 같은 여러 폴더로 이루어져져 있다.
 - Apps
 - Examples
 - libs
 - other
 - projectGenerator
 - Addons



oF root Folder

Running Examples

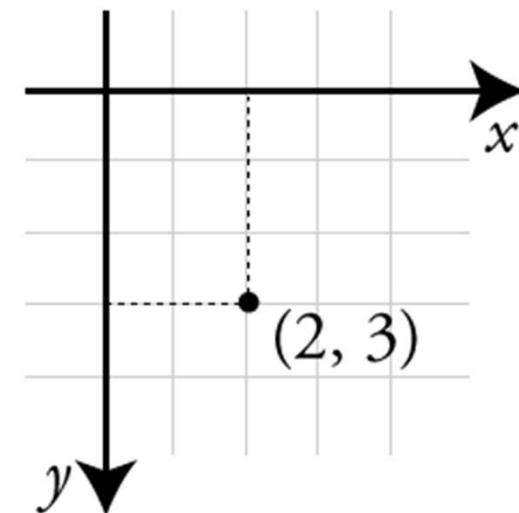
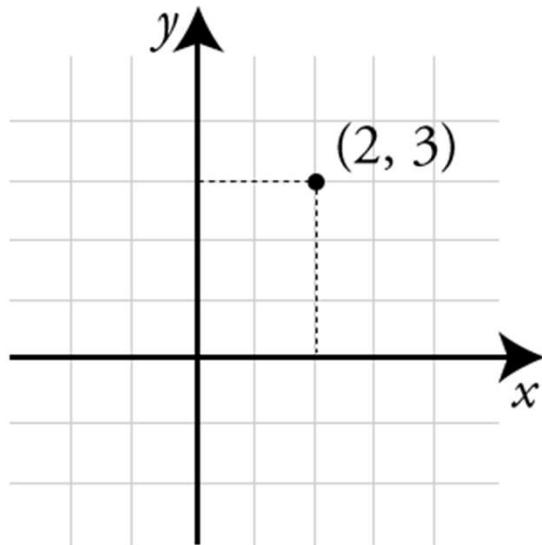
- oF root 폴더 아래의 examples 폴더에 다양한 예제들이 존재한다.
- 예제를 먼저 실행해보기.
 - Xcode 사용자는 .xcodeproj 확장자로 된 파일을 클릭하여 실행 가능
 - VS 사용자는 .sln. 확장자로 된 파일을 클릭하여 실행 가능.

Brush

- 다양한 형태의 Brush 를 만들기 위해서, 먼저 그래픽의 Basic Building Block (BBB) 의 정의가 필요하다.
 - 2D 그래픽 함수를 두 종류의 카테고리로 나눌 수 있다.
 - 기본 모양 (Basic Shapes): 선, 사각형, 삼각형 등
 - 자유 형태 모양 (Freeform Shapes): polygon, path 등

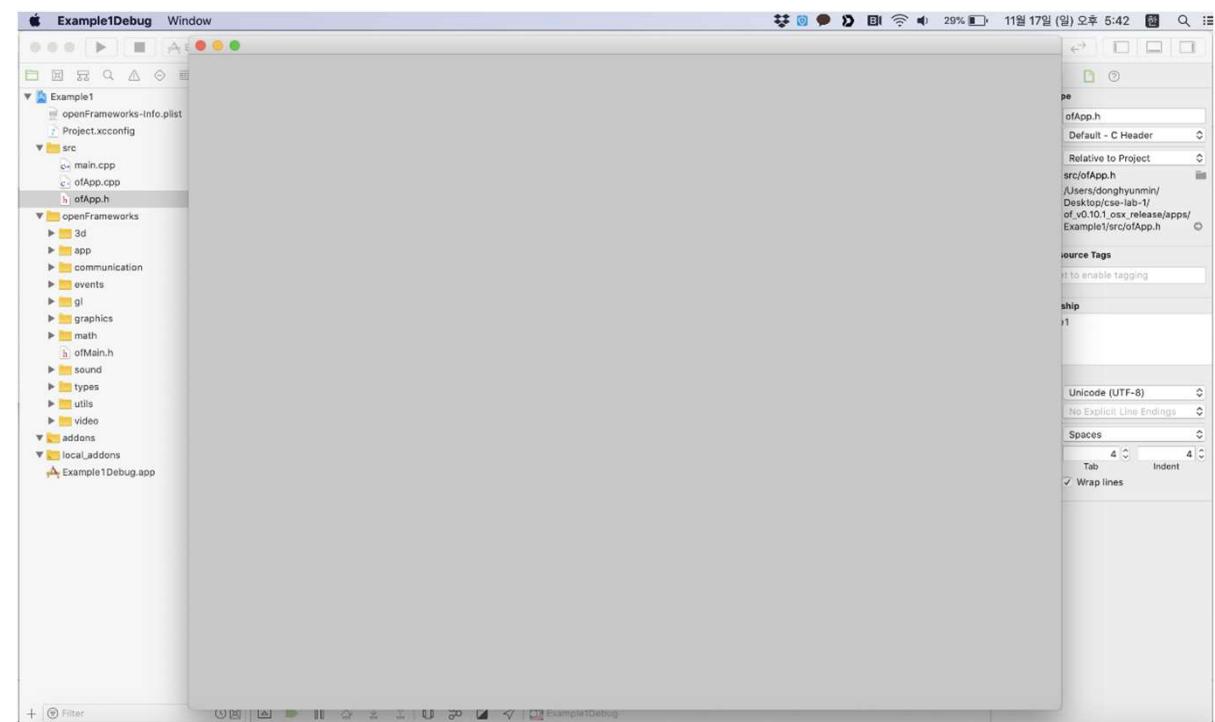
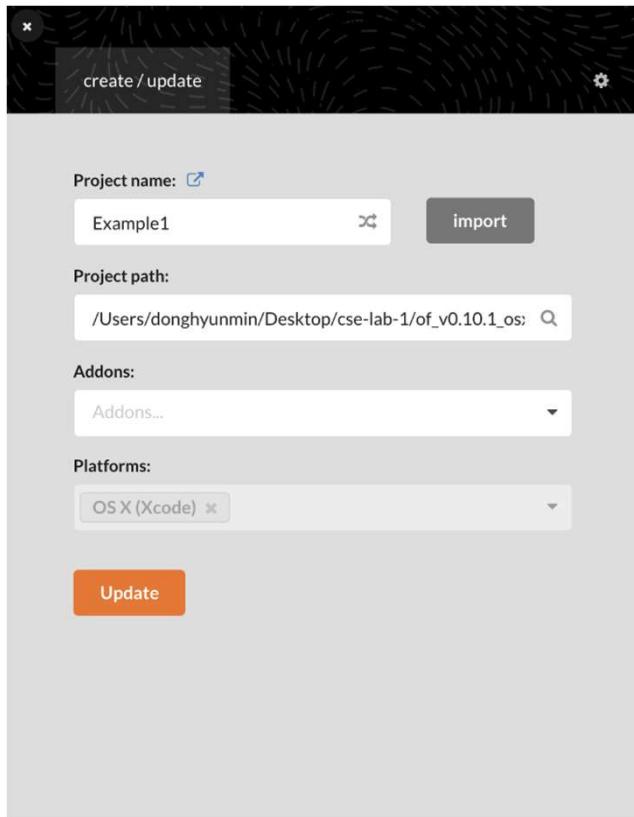
Before Basic Shapes..

- 어떤 모양을 그리기 전에, oF에서 어떻게 물체의 location 을 결정하는지, 어떻게 해석되는지를 이해해야 한다.
- 컴퓨터 그래픽스는 아래 왼쪽 그림과 같은 Cartesian coordinate system 을 기본적으로 사용하되 다음과 같은 두 가지 다른 점을 반영한 coordinate system 을 사용한다.
 - 1) 원점 (Origin)이 upper leftmost.
 - 2) y 축의 방향이 플립 되어있음.



Before Basic Shapes.. (Cont')

- Empty 프로젝트를 만들어보기.



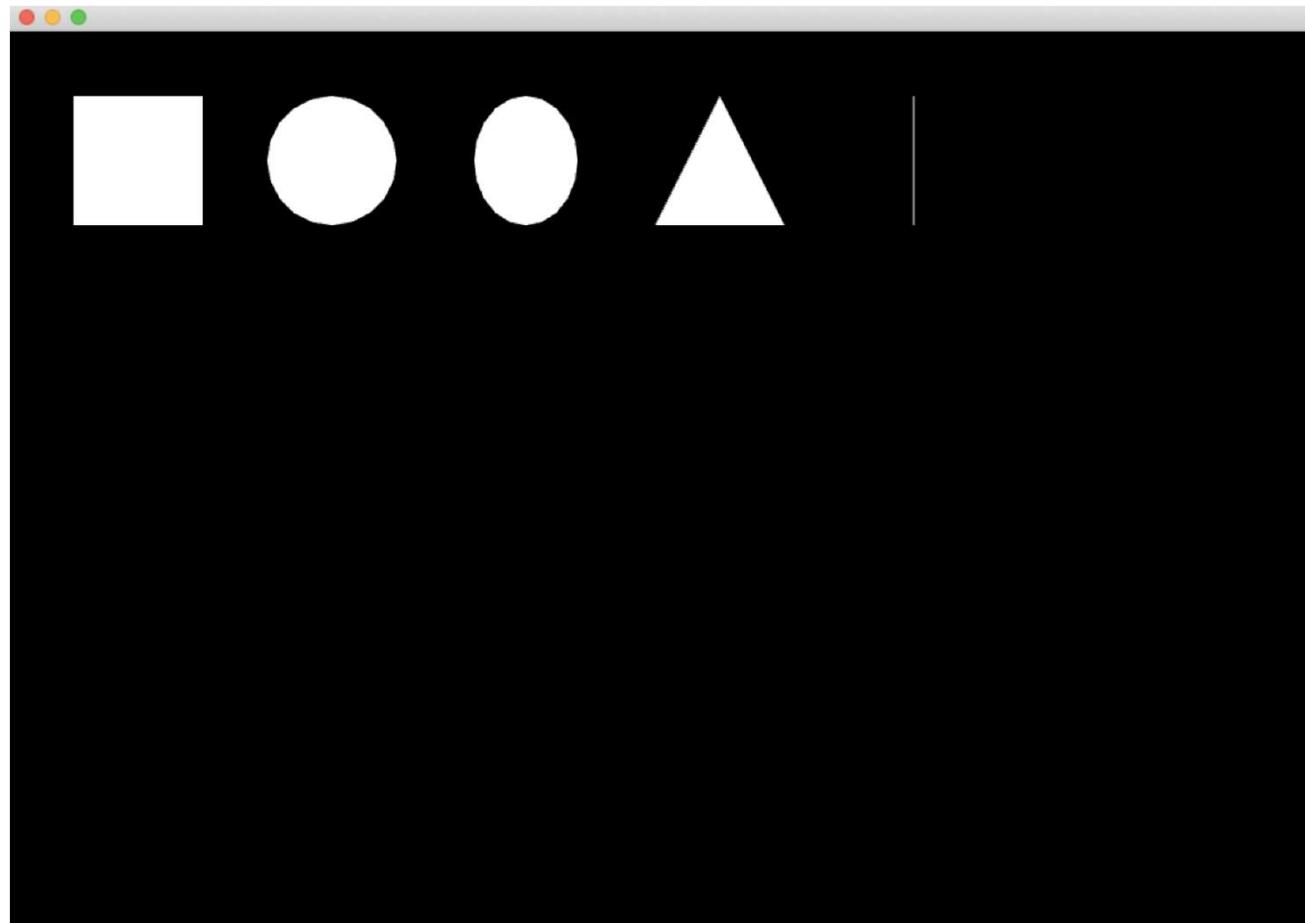
Basic Shapes (1)

- 빈 프로젝트를 만들고, source file 을 열어서 **ofApp.cpp** 의 **draw()** 함수를 찾아 아래 코드를 넣어 보기.

```
ofBackground(0);
ofSetColor(255);
ofDrawRectangle(50, 50, 100, 100);
ofDrawCircle(250, 100, 50);
ofDrawEllipse(400, 100, 80, 100);
ofDrawTriangle(500, 150, 550, 50, 600, 150);
ofDrawLine(700, 50, 700, 150);
```

Basic Shapes (1) (Cont')

- 결과 화면



Basic Shapes (1) (Cont')

- Explanation for our first `draw()` function

```
ofBackground(0); // Clear the screen with a black color, 0 means completely white  
color in grayscale color.  
ofSetColor(255); // Set the drawing color to white.  
// Draw some shapes below.  
ofDrawRectangle(50, 50, 100, 100); // Top left corner at (50, 50), 100 wide x 100 high.  
ofDrawCircle(250, 100, 50); // Centered at (250, 100), radius of 50.  
ofDrawEllipse(400, 100, 80, 100); // Centered at (400 100), 80 wide x 100 high.  
ofDrawTriangle(500, 150, 550, 50, 600, 150); // Three corners: (500, 150), (550, 50),  
(600, 150).  
ofDrawLine(700, 50, 700, 150); // Line from (700, 50) to (700, 150).
```

Basic Shapes (2)

- 이번에는 `draw()` 안에서 `ofFill()`, `ofNoFill()` 함수를 사용해보기.

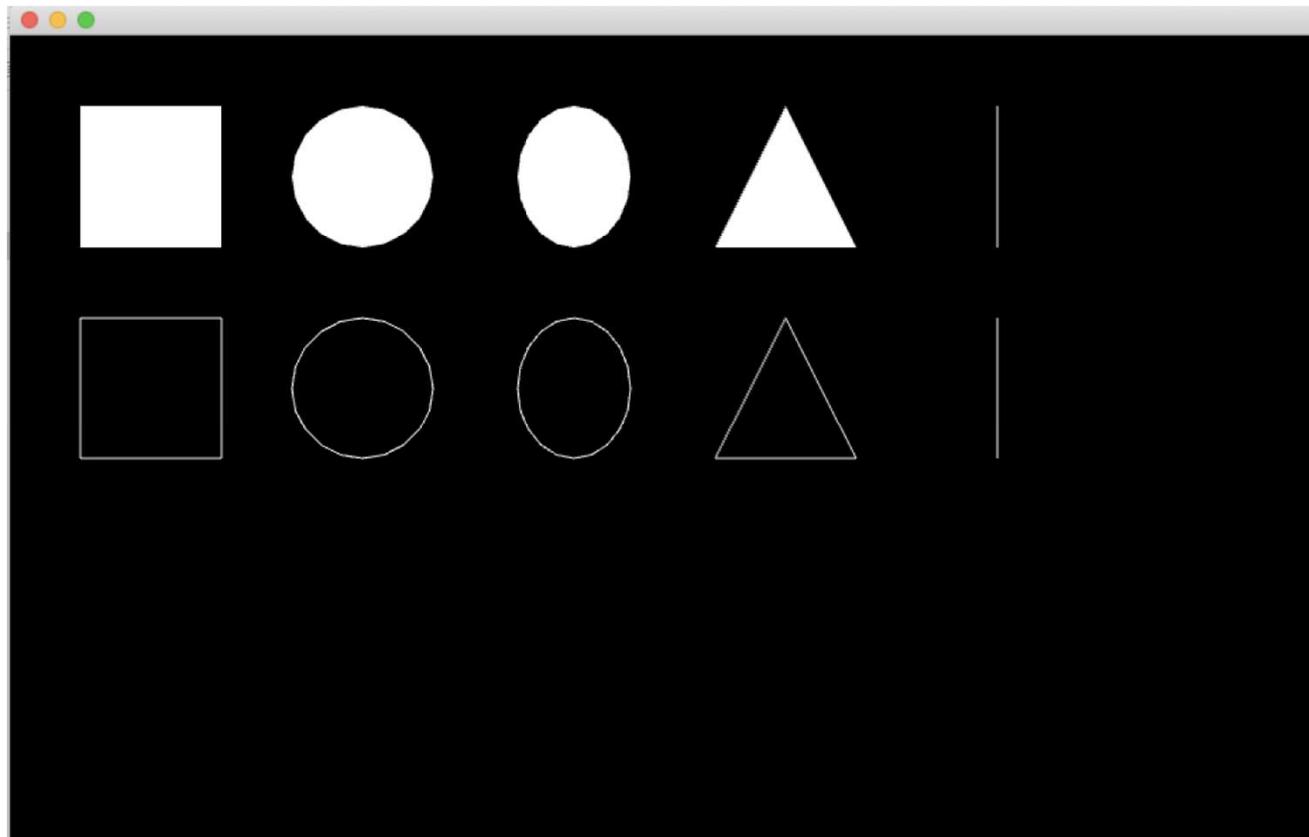
```
ofBackground(0);
ofSetColor(255);

ofFill();
ofDrawRectangle(50, 50, 100, 100);
ofDrawCircle(250, 100, 50);
ofDrawEllipse(400, 100, 80, 100);
ofDrawTriangle(500, 150, 550, 50, 600, 150);
ofDrawLine(700, 50, 700, 150);

ofNoFill();
ofDrawRectangle(50, 200, 100, 100);
ofDrawCircle(250, 250, 50);
ofDrawEllipse(400, 250, 80, 100);
ofDrawTriangle(500, 300, 550, 200, 600, 300);
ofDrawLine(700, 200, 700, 300);
```

Basic Shapes (2) (Cont')

□ 결과 화면



Basic Shapes (2) (Cont')

- Explanation for ofFill(), ofNoFill().

```
...
```

ofFill(); // If we omit this and leave ofNoFill(), all the shapes will be outlines! // Draw some shapes (code omitted).

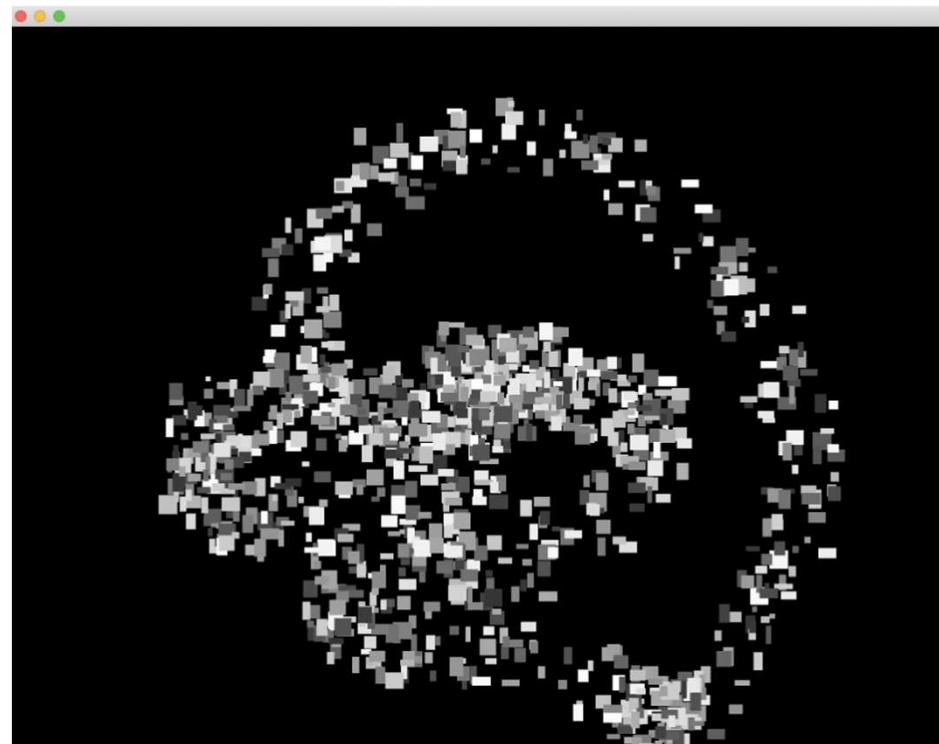
```
...
```

ofNoFill(); // If we omit this and leave ofFill(), all the shapes will be filled! // Draw some shapes (code omitted).

```
...
```

Brushes from Basic Shapes

- 지금까지 배운 기본 모양을 갖고 아래 목표를 갖는 작은 프로젝트를 만들 것이다.
 - 목표: 왼쪽 마우스 버튼을 눌러서 이동할 때마다 폭발하는 모양을 무작위로 떨어트리는 brush 만들기.



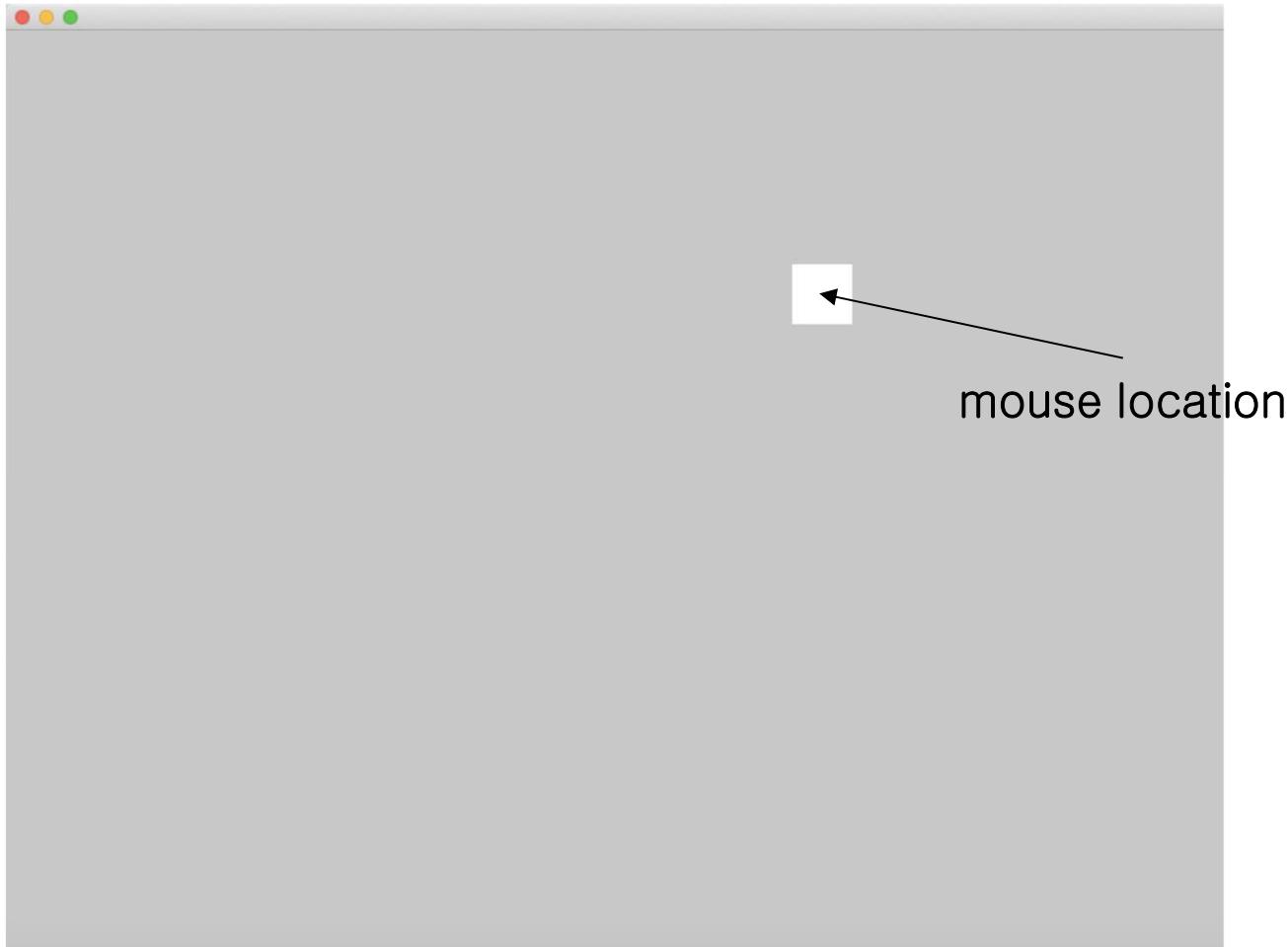
Single Rectangle Brush: Using the Mouse (1)

- 마우스를 잡으면 직사각형 하나만 그리는 간단한 brush 를 만들어 붓의 기초를 다지기.
 - 이를 위해, 우리는 1) 마우스 위치, 2) 마우스 왼쪽 버튼이 눌려지는지를 알아야 한다.
 - 1)을 위해, ofGetMouseX() 와 ofGetMouseY() 함수를 draw() 안에서 사용하여 마우스 위치를 알 수 있다.
 - 2)를 위해, ofGetMousePressed(...) 함수를 사용하여 왼쪽 마우스가 눌려졌는지 아닌지를 확인할 수 있다.
- 아래 코드를 **draw()** 함수에 넣어 보기.

```
if (ofGetMousePressed(OF_MOUSE_BUTTON_LEFT)) {  
    ofSetColor(255);  
    ofSetRectMode(OF_RECTMODE_CENTER); ofDrawRectangle(ofGetMouseX(),  
ofGetMouseY(), 50, 50);  
}
```

Single Rectangle Brush: Using the Mouse (1) (Cont')

□ 결과 화면



Single Rectangle Brush: Using the Mouse (1) (Cont')

- Explanation for our second `draw()` function.

```
if (ofGetMousePressed(OF_MOUSE_BUTTON_LEFT)) { // If the left mouse button is pressed...
    ofSetColor(255);
    ofSetRectMode(OF_RECTMODE_CENTER); ofDrawRectangle(ofGetMouseX(),
ofGetMouseY(), 50, 50); // Draw a 50 x 50 rect centered over the mouse
}
```

Single Rectangle Brush: Using the Mouse (2)

- 왼쪽 마우스 버튼을 떼면 흰색 상자가 바로 사라진다.
 - 왜냐하면 기본적으로 screen 은 매 `draw()` 가 불릴 때, clear 되기 때문이다.
- 금새 사라지는 것을 막기 위해, 아래의 함수를 `ofApp.cpp` 의 `setup()` 함수에 추가 해보기.

```
ofSetBackgroundAuto(false);
```

```
ofBackground(0);
```

Single Rectangle Brush: Using the Mouse (2) (Cont')

- 결과 화면



Single Rectangle Brush: Using the Mouse (2) (Cont')

- Explanation for `ofSetBackgroundAuto()`.

```
ofSetBackgroundAuto(false); // We still want to draw on a black background, so we need to  
// draw // the background before we do anything with the brush  
ofBackground(0);
```

Single Rectangle Brush: Using the Mouse (3)

- draw() 함수를 아래와 같이 수정하여 나타나는 상자의 색깔을 랜덤하게 지정해보기.

```
if (ofGetMousePressed(OF_MOUSE_BUTTON_LEFT)) {  
  
    float randomColor = ofRandom(50, 255);  
    ofSetColor(randomColor);  
  
    ofSetRectMode(OF_RECTMODE_CENTER);  ofDrawRectangle(ofGetMouseX(),  
ofGetMouseY(), 50, 50);  
}
```

Single Rectangle Brush: Using the Mouse (3) (Cont')

- 결과 화면



Single Rectangle Brush: Using the Mouse (3) (Cont')

- Explanation for ofRandom() and ofSetColor().

```
if (ofGetMousePressed(OF_MOUSE_BUTTON_LEFT)) { // If the left mouse button is pressed...
    float randomColor = ofRandom(50, 255);
    ofSetColor(randomColor); // Exclude dark grayscale values (0 - 50) that won't
show on black background

    ofSetRectMode(OF_RECTMODE_CENTER); ofDrawRectangle(ofGetMouseX(),
ofGetMouseY(), 50, 50); // Draw a 50 x 50 rect centered over the mouse
}
```

Bursting Rectangle Brush: Creating Randomized Bursts (1)

- 단순한 하나의 사각형 대신, 여러 개의 폭발하는모양 만들기.
 - for-loop 돌면서 color, width, height, 마우스 위치로 부터 x, y 축 방향으로 얼만큼 떨어져있는지 (offset) 등을 무작위적으로 정하자.
- 아래 수정된 코드를 **draw()** 함수에 넣기.

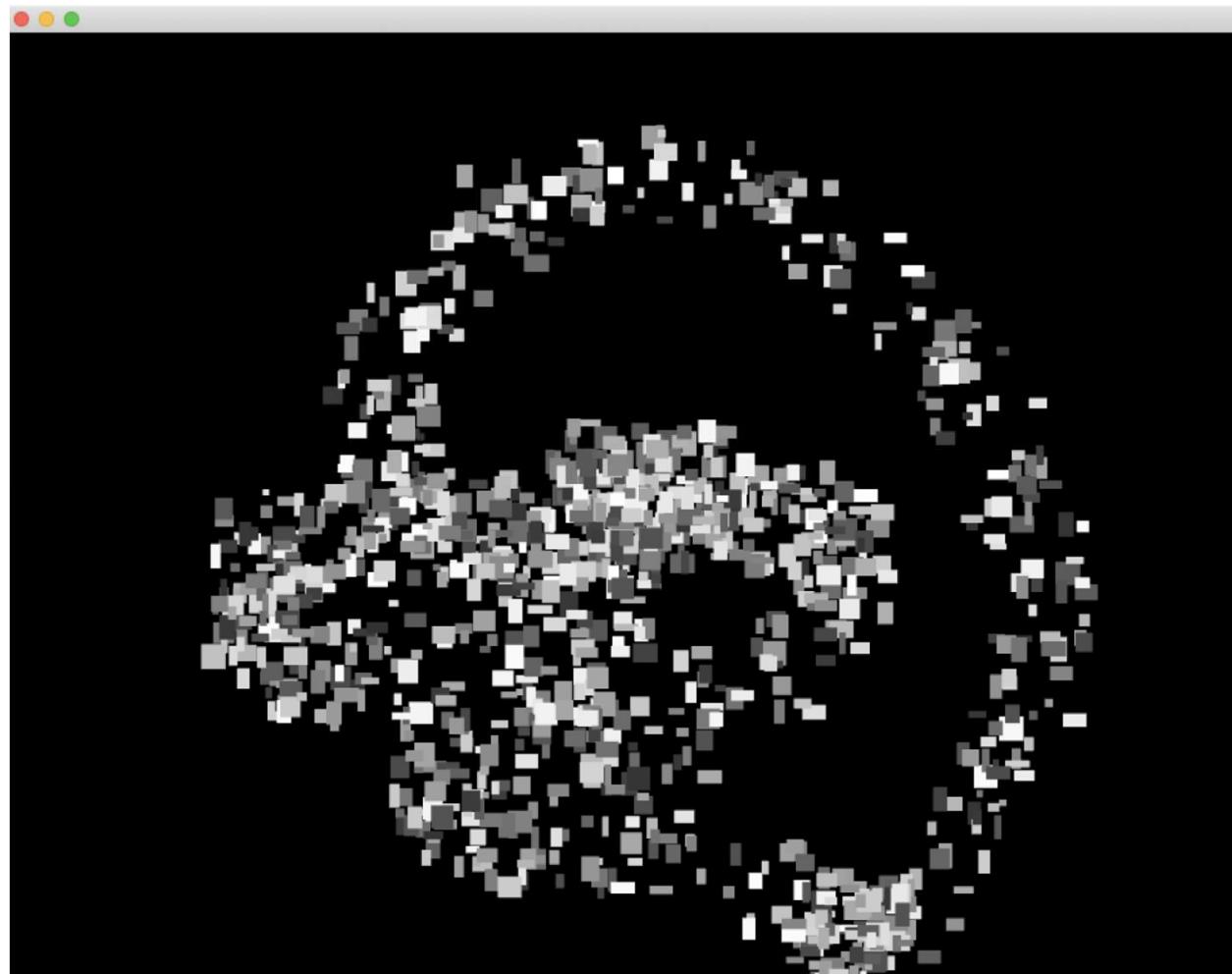
```
if (ofGetMousePressed(OF_MOUSE_BUTTON_LEFT)) { // If the left mouse button is pressed...  
    ofSetRectMode(OF_RECTMODE_CENTER);  
    int numRects = 10;  
    for (int r=0; r<numRects; r++) {  
        ofSetColor(ofRandom(50, 255));  
        float width = ofRandom(5, 20);  
        float height = ofRandom(5, 20);  
        float xOffset = ofRandom(-40, 40);  
        float yOffset = ofRandom(-40, 40);  
        ofDrawRectangle(ofGetMouseX() + xOffset, ofGetMouseY() + yOffset, width, height);  
    }  
}
```

- 아래 코드를 **setup()** 함수 안에서 처음에 넣기.

```
ofSetFrameRate(60);
```

Bursting Rectangle Brush: Creating Randomized Bursts (1) (Cont')

- 결과 화면



Glowing Circle Brush: Using Color

- Color 를 어떻게 바꾸는 지 이해하기.
- https://github.com/openframeworks/ofBook/tree/master/chapters/intro_to_graphics/code/1_ii_b_Bursting_Rect_Brush/src

```
ofColor myOrange(255, 132, 0); // Defining an opaque orange color - specified using RGB
ofColor myBlue(0, 0, 255, 50); // Defining a transparent blue color - specified using RGBA
// We can access the red, green, blue and alpha channels like this:

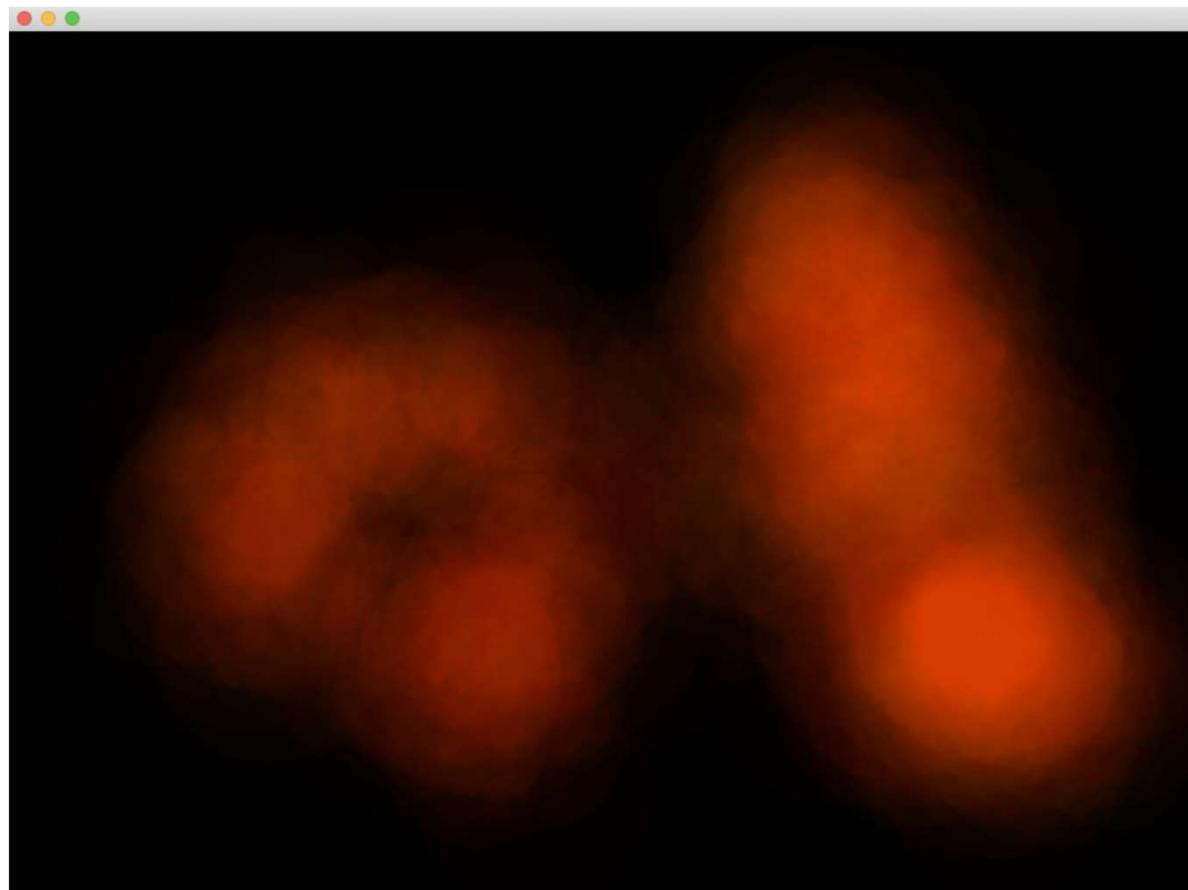
ofColor myGreen(0, 0, 255, 255);

cout << "Red channel:" << myGreen.r << endl;
cout << "Green channel:" << myGreen.g << endl;
cout << "Blue channel:" << myGreen.b << endl;
cout << "Alpha channel:" << myGreen.a << endl;
// We can also set the red, green, blue and alpha channels like this:
ofColor myYellow;
myYellow.r = 255; myYellow.b = 0; myYellow.g = 255; myYellow.a = 255;
// We can also make use of some predefined colors provided by openFrameworks:
ofColor myAqua = ofColor::aqua;
ofColor myPurple = ofColor::plum;
```

Glowing Circle Brush: Using Color (Cont')

□ 결과 화면

- https://github.com/openframeworks/ofBook/tree/master/chapters/intro_to_graphics/code/1_ji_b_Bursting_Rect_Brush/src



Fleeing Triangle Brush: Vectors

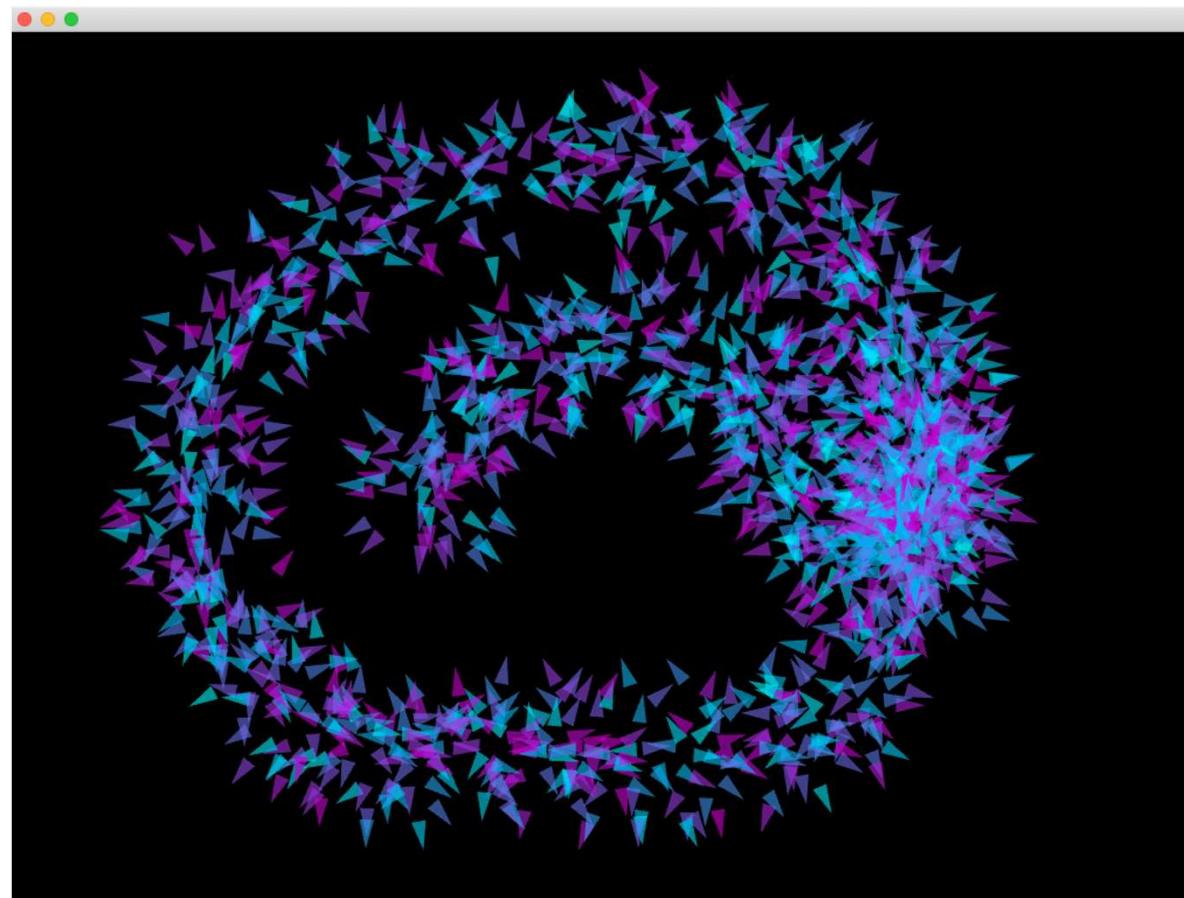
- Vector 를 어떻게 다루는 지 이해하기.
- https://github.com/openframeworks/ofBook/tree/master/chapters/intro_to_graphics/code/1_ii_e_Triangle_Brush

```
ofVec2f mousePos(ofGetMouseX(), ofGetMouseY());  
  
// Define a triangle at the origin (0,0) that points to the right  
ofVec2f p1(0, 25.0);  
ofVec2f p2(100, 0);  
ofVec2f p3(0, -25.0);  
  
// Shift the triangle to the mouse position  
p1 += mousePos;  
p2 += mousePos;  
p3 += mousePos;
```

Fleeing Triangle Brush: Vectors and Rotations (Cont')

□ 결과 화면

- https://github.com/openframeworks/ofBook/tree/master/chapters/intro_to_graphics/code/1_ji_e_Triangle_Brush



update() 함수

- update() 함수는 non-drawing 작업들을 위한 것으로 한 frame 당 한 번 불려지도록 설계된 함수이다.
- 아래 코드를 ofApp.h 안의 클래스의 멤버 변수로 넣기.

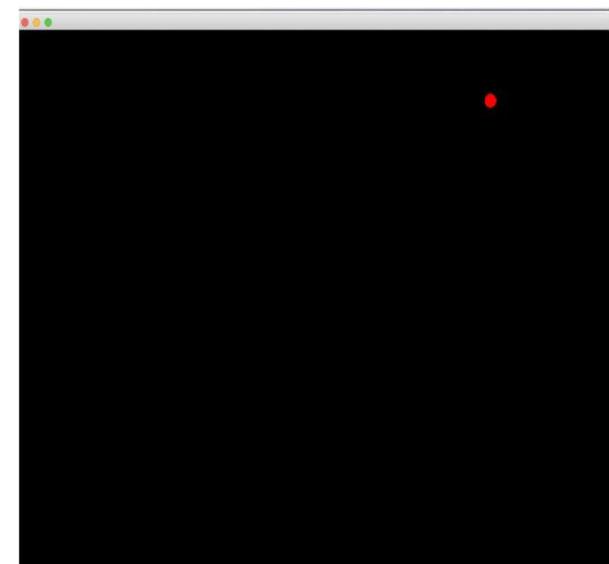
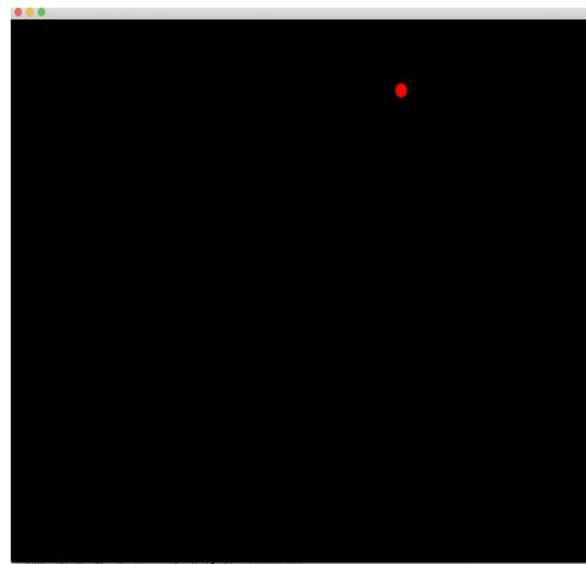
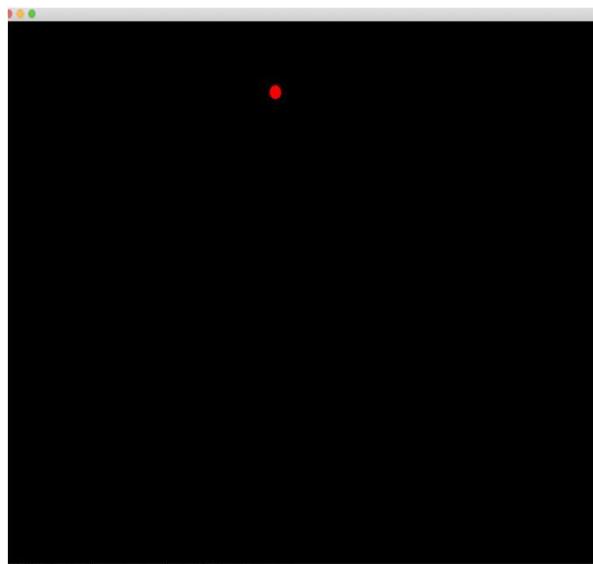
```
int xPos;
```

- 아래 코드를 참고하여 ofApp.cpp 코드 작성하기

```
void ofApp::setup(){
    xPos = 5; // horizontal start position
    ofBackground(ofColor::black); // black background
}
void ofApp::update(){
    xPos += 2;
    if(ofGetWidth()<xPos){ // if horizontal position is off the screen (width)
        xPos = 5; // reset horizontal position
    }
}
void ofApp::draw(){
    ofSetColor(ofColor::red); // draw everything in red
    ofDrawCircle(xPos, 100, 10); // draw a circle at the (variable) horizontal position, 100 pixels from the top with a 10 pixel diameter
}
```

update() 함수 (Cont')

- 결과 화면



Running Example (Con't)

- List of Toy Examples
 - math/particleExample
 - events/simpleEventsExample
 - strings/sortingExample