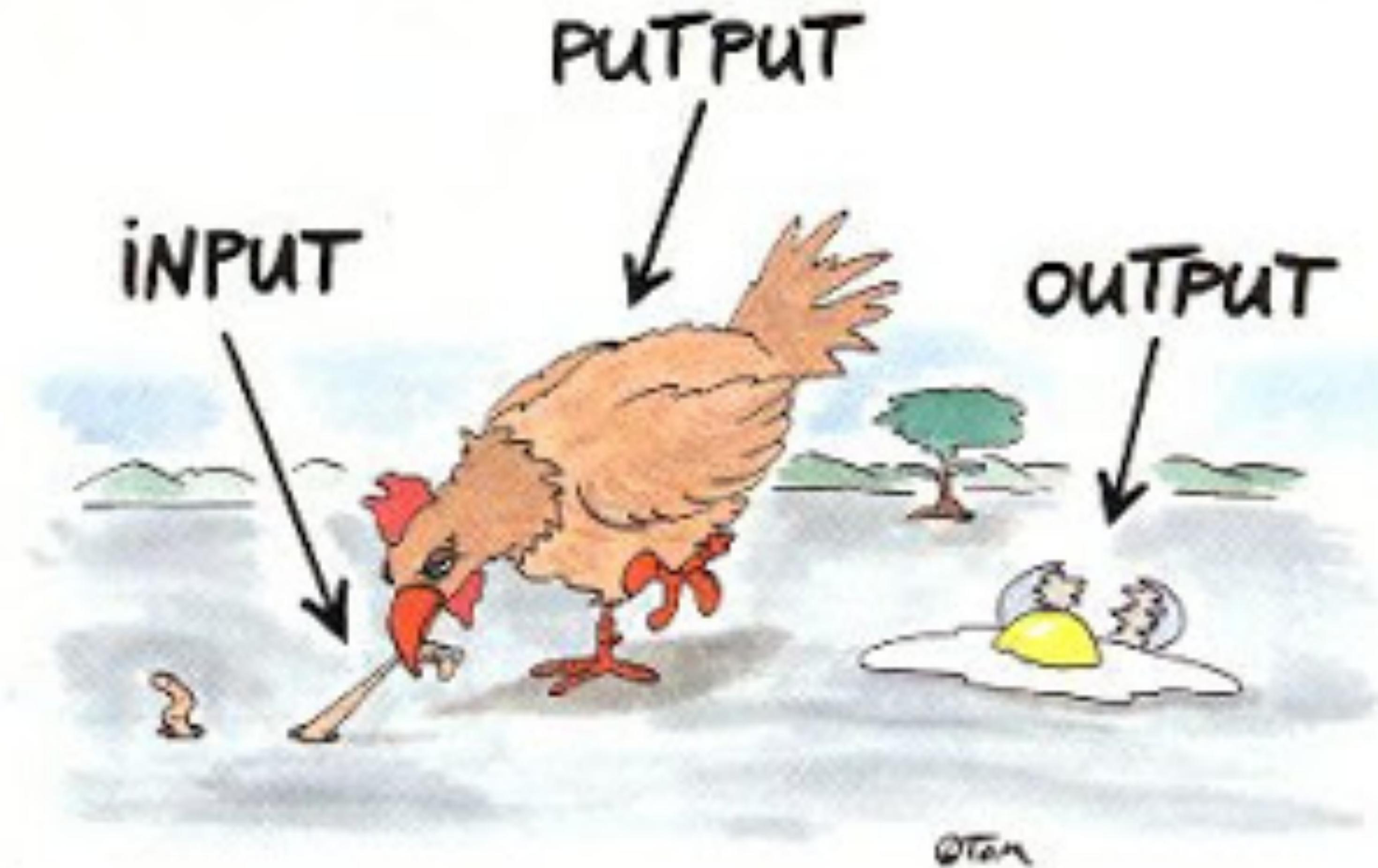


Inputs, Outputs, and Addons

CC Lab 2016 openFrameworks Week 4



openFrameworks is powerful because it handles various inputs & outputs.

ofxaddons.com

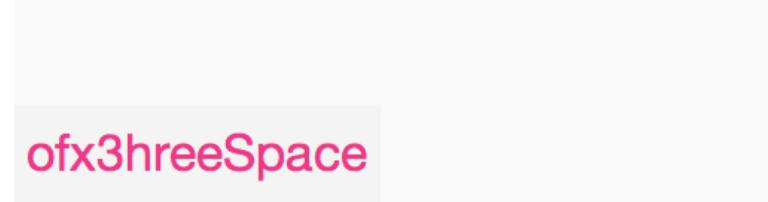
ofxAddons categories popular freshest unsorted contributors faq/help login

Categories

Algorithms	35
Animation	56
Bridges	111
Computer Vision	185
Game Engine	10
Geometry	38
Graphics	397
GUI	147
Hardware Interface	199
iOS	52
Machine Learning	34
Physics	28
Sound	100
Typography	34
Utilities	390
Video/Camera	166
Web/Networking	143

ofxAddons is directory of extensions and libraries for the [openFrameworks](#) creative coding toolkit. Compiled fresh from [Github](#) daily.

Computer Vision

 **ofx3hreeSpace**

3hreeSpace offers a method for 3D tracking using two cameras

Maintained by  [islandrabe](#)
Last updated [2 years ago](#)
Release? ~0.8.3
★ 1
Examples 1
Categories: Computer Vision

 **ofxActiveScan**

Active 3D scan for uncalibrated projector-camera/Kinect

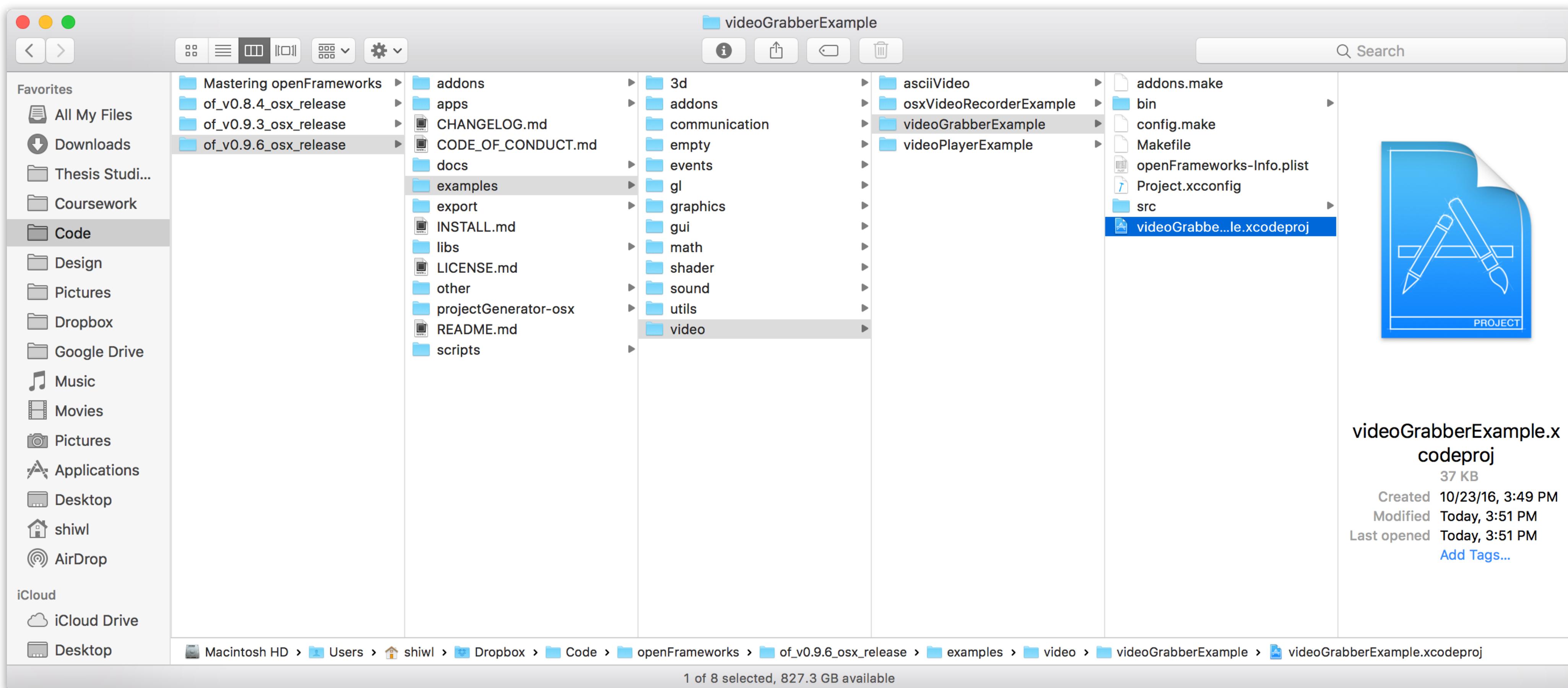
Maintained by  [micuat](#)
Last updated [2 years ago](#)
Release? ~0.8.3
★ 41
Makefile
Examples 6
Categories: Computer Vision

 **OfxAddonCombo**

Combining open frameworks libraries can be challenging. This app combines the ofx postprocessing library with the ofx optical flow library.

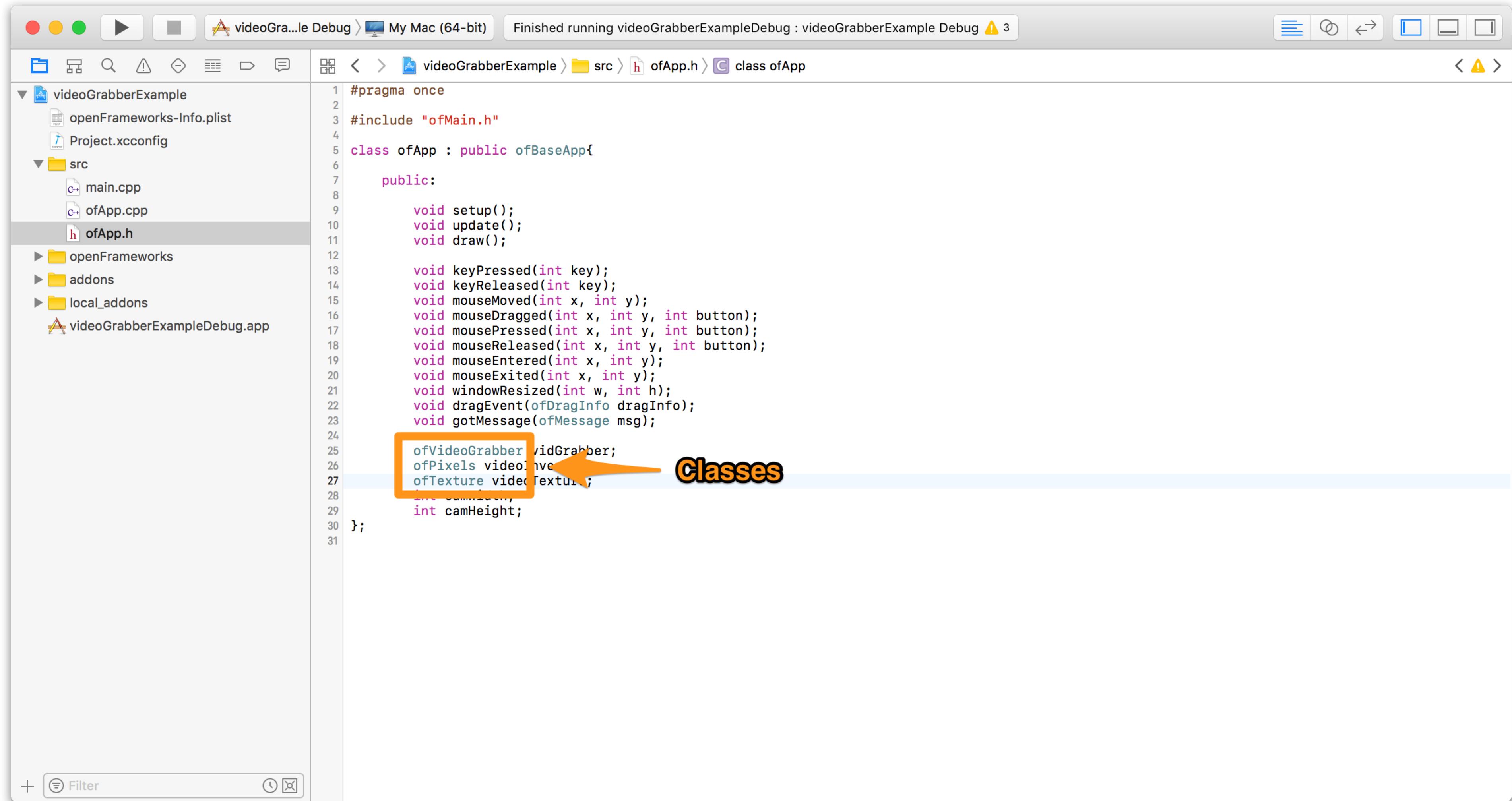
Maintained by  [ccrecord](#)
Last updated [4 years ago](#)
Release? ~0.7.3
★ 1
Categories: Computer Vision

openFrameworks is even enhanced by addons!



Input: Video Camera

examples/video/videoGrabberExample

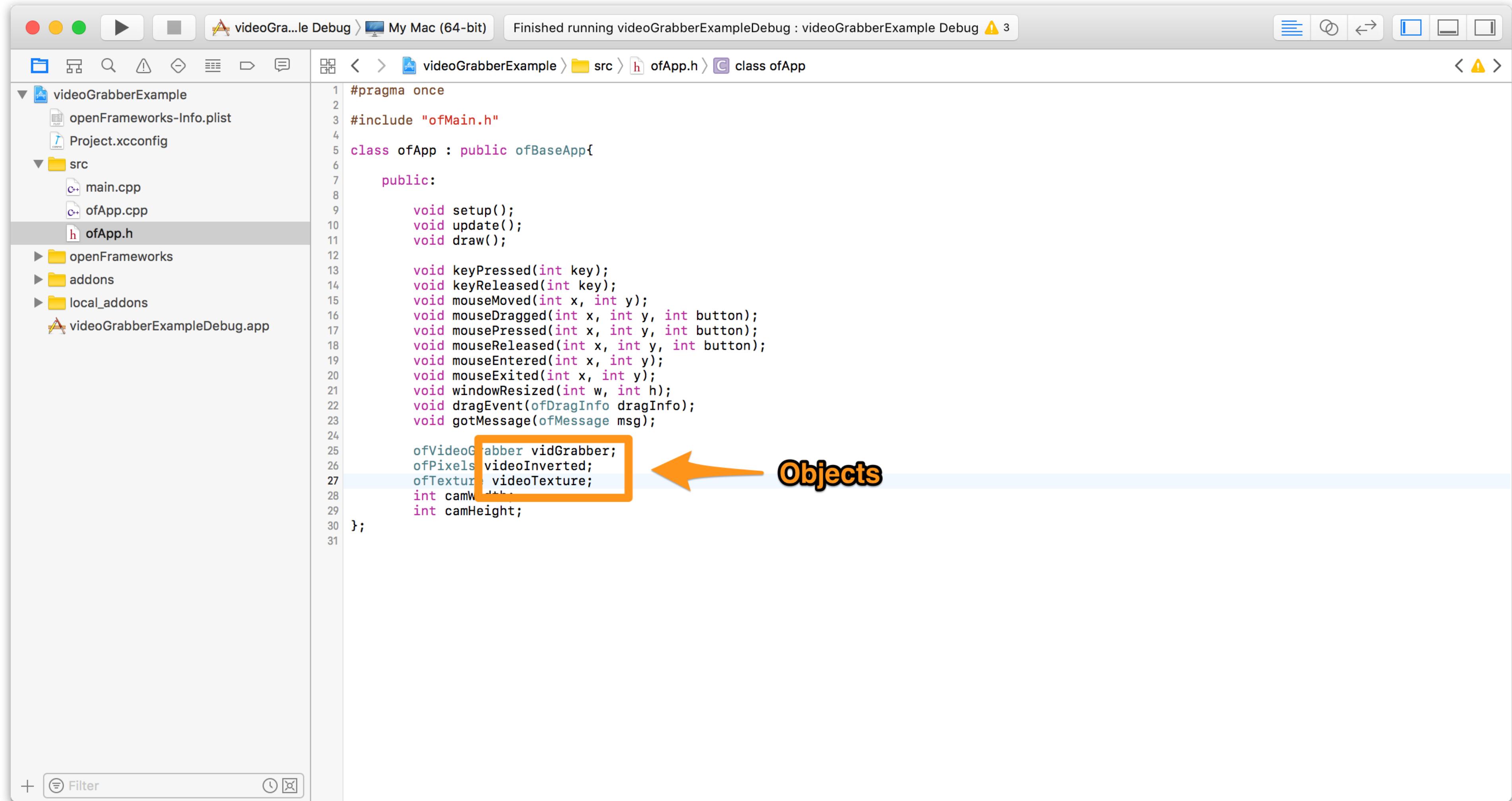


The screenshot shows the Xcode interface with the project "videoGrabberExample" open. The current file is "ofApp.h" located in the "src" folder. The code defines a class `ofApp` that inherits from `ofBaseApp`. The class contains several public methods and event handlers. At the bottom, it declares three member variables: `ofVideoGrabber vidGrabber;`, `ofPixels videoIn;`, and `ofTexture videoTexture;`. An orange box highlights these variable declarations, with an arrow pointing to the word "Classes".

```
1 #pragma once
2
3 #include "ofMain.h"
4
5 class ofApp : public ofBaseApp{
6
7     public:
8
9         void setup();
10        void update();
11        void draw();
12
13        void keyPressed(int key);
14        void keyReleased(int key);
15        void mouseMoved(int x, int y);
16        void mouseDragged(int x, int y, int button);
17        void mousePressed(int x, int y, int button);
18        void mouseReleased(int x, int y, int button);
19        void mouseEntered(int x, int y);
20        void mouseExited(int x, int y);
21        void windowResized(int w, int h);
22        void dragEvent(ofDragInfo dragInfo);
23        void gotMessage(ofMessage msg);
24
25        ofVideoGrabber vidGrabber;
26        ofPixels videoIn;
27        ofTexture videoTexture;
28
29        int camWidth;
30        int camHeight;
31    };

```

ofApp.h



The screenshot shows the Xcode interface with the project "videoGrabberExample" selected. The current file is "ofApp.h" located in the "src" folder. The code defines the `ofApp` class, which inherits from `ofBaseApp`. The class contains several methods for handling user input and window events, as well as member variables for managing a video grabber and texture.

```
#pragma once
#include "ofMain.h"

class ofApp : public ofBaseApp{
public:
    void setup();
    void update();
    void draw();

    void keyPressed(int key);
    void keyReleased(int key);
    void mouseMoved(int x, int y);
    void mouseDragged(int x, int y, int button);
    void mousePressed(int x, int y, int button);
    void mouseReleased(int x, int y, int button);
    void mouseEntered(int x, int y);
    void mouseExited(int x, int y);
    void windowResized(int w, int h);
    void dragEvent(ofDragInfo dragInfo);
    void gotMessage(ofMessage msg);

    ofVideoGrabber vidGrabber;
    ofPixels videoInverted;
    ofTexture videoTexture;
    int camWidth;
    int camHeight;
};

};
```

ofApp.h

The screenshot shows the Xcode interface with the project "videoGrabberExample" open. The "src" folder contains "main.cpp" and "ofApp.cpp". The "ofApp.cpp" file is the active editor, showing the following code:

```
1 #include "ofApp.h"
2
3 //-----
4 void ofApp::setup(){
5     camWidth = 320; // try to grab at this size.
6     camHeight = 240;
7
8     //we can now get back a list of devices.
9     vector<ofVideoDevice> devices = vidGrabber.listDevices();
10
11    for(int i = 0; i < devices.size(); i++){
12        if(devices[i].bAvailable){
13            ofLogNotice() << devices[i].id << ":" << devices[i].deviceName;
14        }else{
15            ofLogNotice() << devices[i].id << ":" << devices[i].deviceName << " - unavailable ";
16        }
17    }
18
19    vidGrabber.setDeviceID(0);
20    vidGrabber.setDesiredFrameRate(60);
21    vidGrabber.initGrabber(camWidth, camHeight);
22
23    videoInverted.allocate(camWidth, camHeight, OF_PIXELS_RGB);
24    videoTexture.allocate(videoInverted);
25    ofSetVerticalSync(true);
26}
27
28
29 //-----
30 void ofApp::update(){
31     ofBackground(100, 100, 100);
32     vidGrabber.update();
33
34     if(vidGrabber.isFrameNew()){
35         ofPixels & pixels = vidGrabber.getPixels();
36         for(int i = 0; i < pixels.size(); i++){
37             videoInverted[i] = 255 - pixels[i];
38         }
39         videoTexture.loadData(videoInverted);
40     }
41 }
42
43 //-----
44 void ofApp::draw(){
45     ofSetHexColor(0xffffffff);
46     vidGrabber.draw(20, 20);
47     videoTexture.draw(20 + camWidth, 20, camWidth, camHeight);
48 }
```

ofApp.cpp

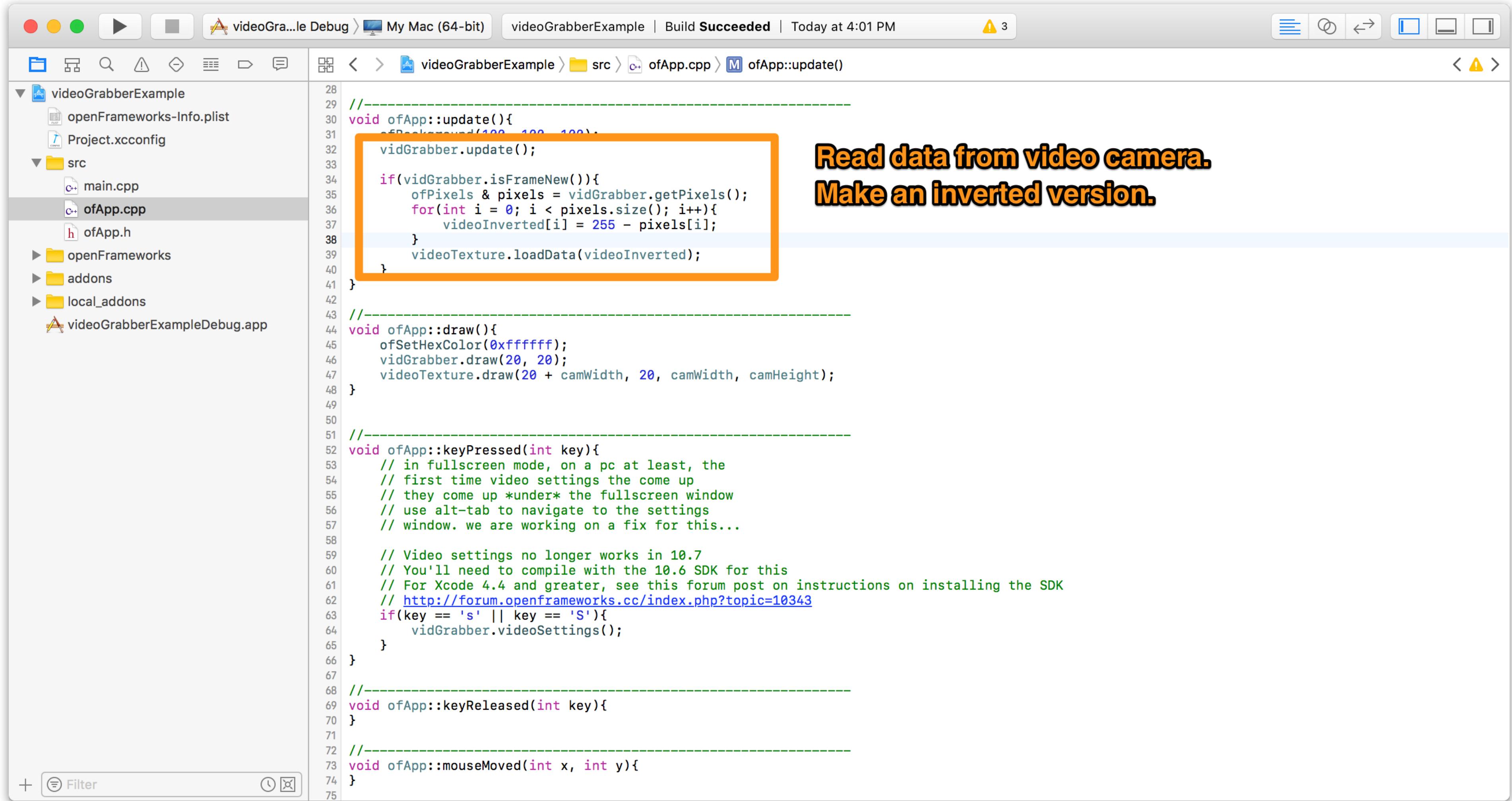
Setup vidGrabber

The screenshot shows the Xcode IDE interface with the following details:

- Title Bar:** Shows the project name "videoGrabberExample Debug" and the build status "Build Succeeded" at "Today at 4:01 PM".
- File Navigator:** Displays the project structure:
 - videoGrabberExample
 - openFrameworks-Info.plist
 - Project.xcconfig
 - src
 - main.cpp
 - ofApp.cpp
 - ofApp.h
 - openFrameworks
 - addons
 - local_addons
 - videoGrabberExampleDebug.app
- Editor Area:** Shows the content of the ofApp.cpp file. The code is annotated with a callout pointing to the line where a video texture is allocated.

```
1 #include "ofApp.h"
2
3 //-----
4 void ofApp::setup(){
5     camWidth = 320; // try to grab at this size.
6     camHeight = 240;
7
8     //we can now get back a list of devices.
9     vector<ofVideoDevice> devices = vidGrabber.listDevices();
10
11    for(int i = 0; i < devices.size(); i++){
12        if(devices[i].bAvailable){
13            ofLogNotice() << devices[i].id << ":" << devices[i].deviceName;
14        }else{
15            ofLogNotice() << devices[i].id << ":" << devices[i].deviceName << " - unavailable ";
16        }
17    }
18
19    vidGrabber.setDeviceID(0);
20    vidGrabber.setDesiredFrameRate(60);
21    vidGrabber.initGrabber(camWidth, camHeight);
22
23    videoInverted.allocate(camWidth, camHeight, OF_PIXELS_RGB);
24    videoTexture.allocate(videoInverted);
25    ofSetVerticalSync(true);
26}
27
28
29 //-----
30 void ofApp::update(){
31     ofBackground(100, 100, 100);
32     vidGrabber.update();
33
34     if(vidGrabber.isFrameNew()){
35         ofPixels & pixels = vidGrabber.getPixels();
36         for(int i = 0; i < pixels.size(); i++){
37             videoInverted[i] = 255 - pixels[i];
38         }
39         videoTexture.loadData(videoInverted);
40     }
41 }
42
43 //-----
44 void ofApp::draw(){
45     ofSetHexColor(0xffffffff);
46     vidGrabber.draw(20, 20);
47     videoTexture.draw(20 + camWidth, 20, camWidth, camHeight);
48 }
```
- Annotations:** A yellow callout box highlights the line `videoInverted.allocate(camWidth, camHeight, OF_PIXELS_RGB);` with the text "Setup videoInverted".

ofApp.cpp



The screenshot shows the Xcode interface with the project "videoGrabberExample" open. The "src" folder contains "main.cpp", "ofApp.cpp", and "ofApp.h". The "ofApp.cpp" file is the current editor tab, showing the following code:

```
28 //-
29 void ofApp::update(){
30     ofBackground(100, 100, 100);
31     vidGrabber.update();
32
33     if(vidGrabber.isFrameNew()){
34         ofPixels & pixels = vidGrabber.getPixels();
35         for(int i = 0; i < pixels.size(); i++){
36             videoInverted[i] = 255 - pixels[i];
37         }
38         videoTexture.loadData(videoInverted);
39     }
40
41 //-
42 void ofApp::draw(){
43     ofSetHexColor(0xffffffff);
44     vidGrabber.draw(20, 20);
45     videoTexture.draw(20 + camWidth, 20, camWidth, camHeight);
46 }
47
48 //-
49 void ofApp::keyPressed(int key){
50     // in fullscreen mode, on a pc at least, the
51     // first time video settings the come up
52     // they come up *under* the fullscreen window
53     // use alt-tab to navigate to the settings
54     // window. we are working on a fix for this...
55
56     // Video settings no longer works in 10.7
57     // You'll need to compile with the 10.6 SDK for this
58     // For Xcode 4.4 and greater, see this forum post on instructions on installing the SDK
59     // http://forum.openframeworks.cc/index.php?topic=10343
60     if(key == 's' || key == 'S'){
61         vidGrabber.videoSettings();
62     }
63 }
64
65 //-
66 void ofApp::keyReleased(int key){
67 }
68
69 //-
70 void ofApp::mouseMoved(int x, int y){
71 }
72
73 //-
74
```

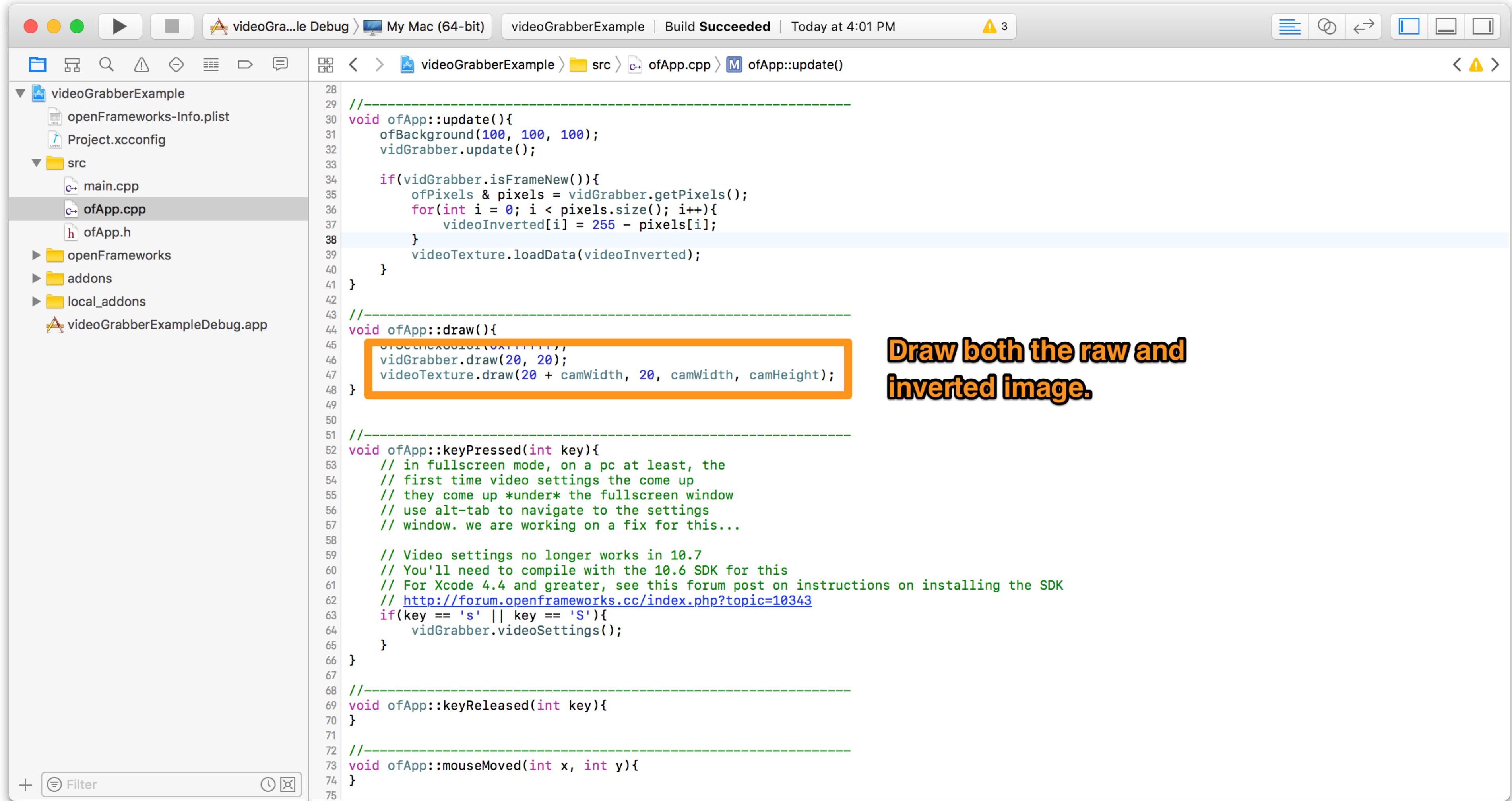
A callout box highlights the following code block:

```
vidGrabber.update();

if(vidGrabber.isFrameNew()){
    ofPixels & pixels = vidGrabber.getPixels();
    for(int i = 0; i < pixels.size(); i++){
        videoInverted[i] = 255 - pixels[i];
    }
    videoTexture.loadData(videoInverted);
}
```

To the right of this highlighted code, the text "Read data from video camera. Make an inverted version." is displayed in bold orange font.

ofApp.cpp

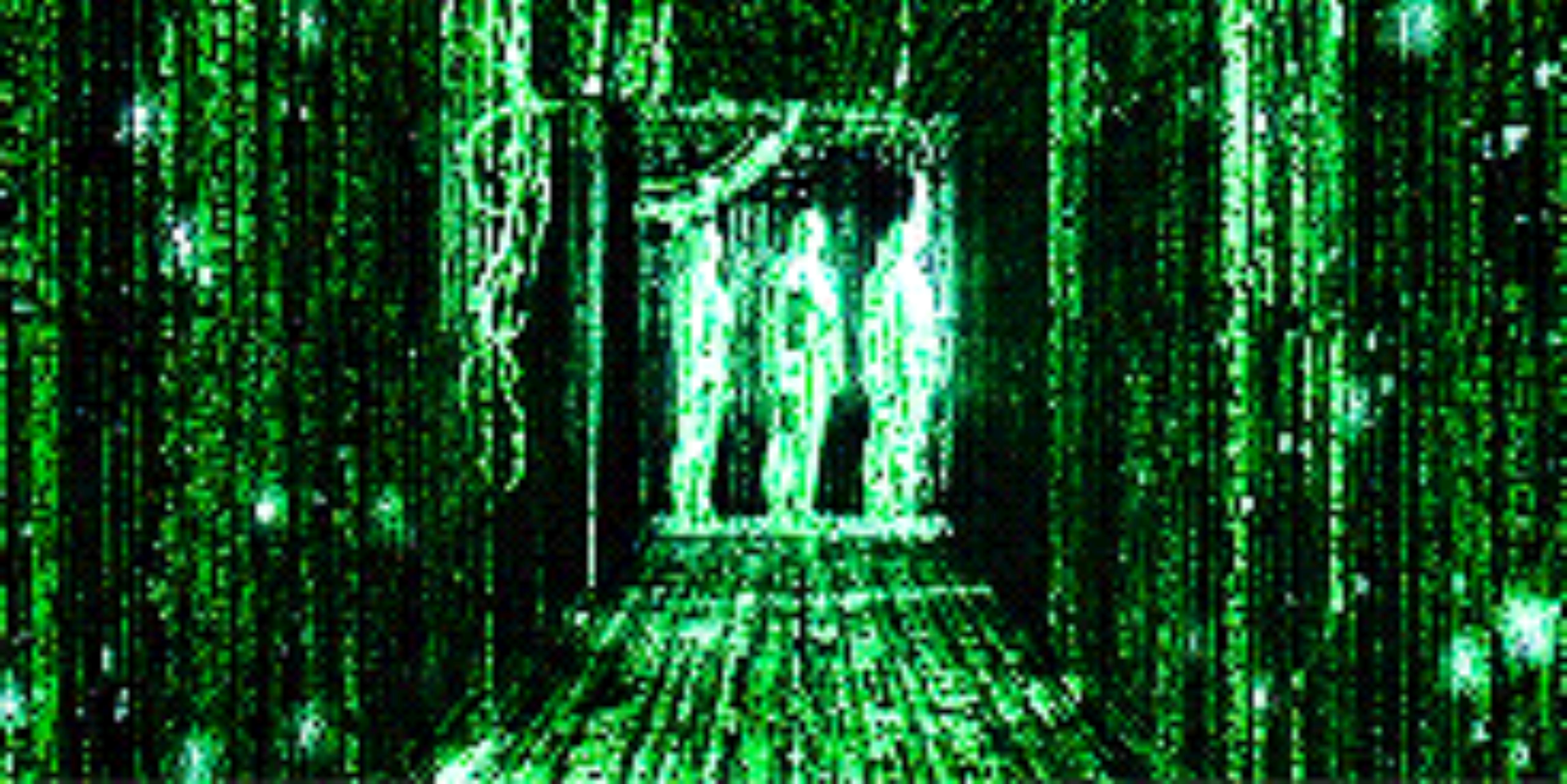


The screenshot shows the Xcode interface with the project "videoGrabberExample" open. The "src" folder contains "main.cpp", "ofApp.cpp", and "ofApp.h". The "ofApp.cpp" file is the current editor tab, showing the following code:

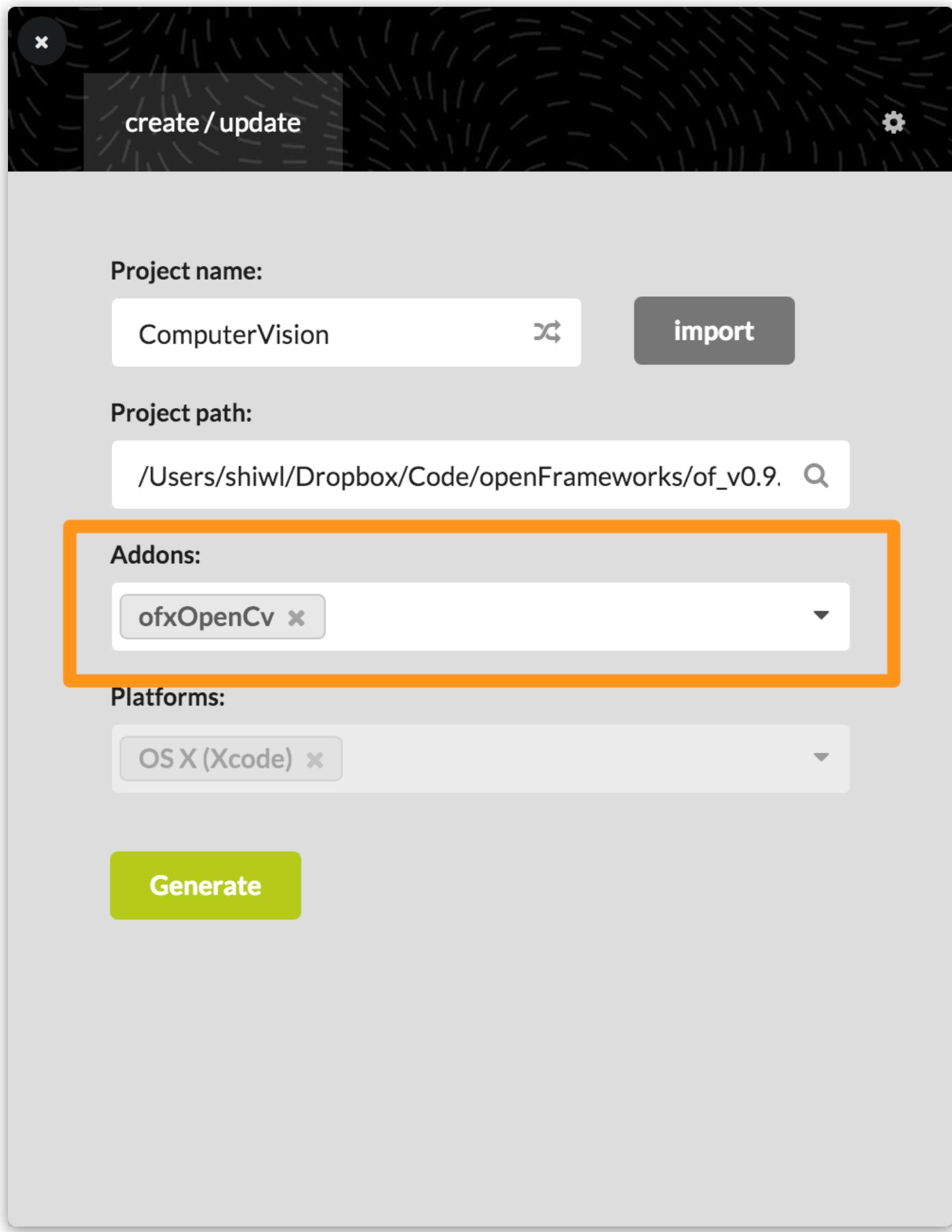
```
28 //-
29 void ofApp::update(){
30     ofBackground(100, 100, 100);
31     vidGrabber.update();
32
33     if(vidGrabber.isFrameNew()){
34         ofPixels & pixels = vidGrabber.getPixels();
35         for(int i = 0; i < pixels.size(); i++){
36             videoInverted[i] = 255 - pixels[i];
37         }
38         videoTexture.loadData(videoInverted);
39     }
40 }
41
42 //-
43 void ofApp::draw(){
44     ofRect(0, 0, 1000, 1000);
45     vidGrabber.draw(20, 20);
46     videoTexture.draw(20 + camWidth, 20, camWidth, camHeight);
47 }
48
49
50 //-
51 void ofApp::keyPressed(int key){
52     // in fullscreen mode, on a pc at least, the
53     // first time video settings the come up
54     // they come up *under* the fullscreen window
55     // use alt-tab to navigate to the settings
56     // window. we are working on a fix for this...
57
58     // Video settings no longer works in 10.7
59     // You'll need to compile with the 10.6 SDK for this
60     // For Xcode 4.4 and greater, see this forum post on instructions on installing the SDK
61     // http://forum.openframeworks.cc/index.php?topic=10343
62     if(key == 's' || key == 'S'){
63         vidGrabber.videoSettings();
64     }
65 }
66
67 //-
68 void ofApp::keyReleased(int key){
69 }
70
71 //-
72 void ofApp::mouseMoved(int x, int y){
73 }
```

A yellow box highlights the draw call in line 46: `vidGrabber.draw(20, 20);`. To its right, the text "Draw both the raw and inverted image." is displayed in bold orange font.

ofApp.cpp



Example: Computer Vision!



Add an addon to your oF project

ofxaddons.com

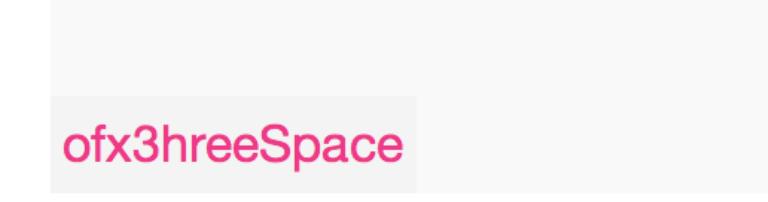
ofxAddons categories popular freshest unsorted contributors faq/help login

Categories

Algorithms	35
Animation	56
Bridges	111
Computer Vision	185
Game Engine	10
Geometry	38
Graphics	397
GUI	147
Hardware Interface	199
iOS	52
Machine Learning	34
Physics	28
Sound	100
Typography	34
Utilities	390
Video/Camera	166
Web/Networking	143

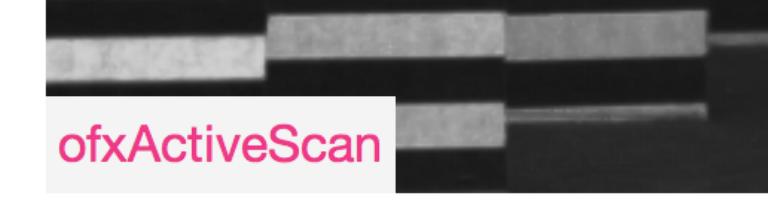
ofxAddons is directory of extensions and libraries for the [openFrameworks](#) creative coding toolkit. Compiled fresh from [Github](#) daily.

Computer Vision

 ofx3hreeSpace

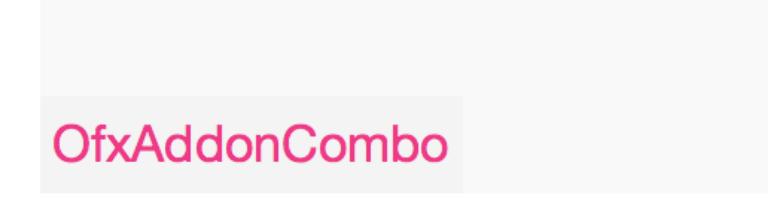
3hreeSpace offers a method for 3D tracking using two cameras

Maintained by  [islandrabe](#)
Last updated [2 years ago](#)
Release? ~0.8.3
★ 1
Examples 1
Categories: Computer Vision

 ofxActiveScan

Active 3D scan for uncalibrated projector-camera/Kinect

Maintained by  [micuat](#)
Last updated [2 years ago](#)
Release? ~0.8.3
★ 41
Makefile
Examples 6
Categories: Computer Vision

 OfxAddonCombo

Combining open frameworks libraries can be challenging. This app combines the ofx postprocessing library with the ofx optical flow library.

Maintained by  [ccrecord](#)
Last updated [4 years ago](#)
Release? ~0.7.3
★ 1
Categories: Computer Vision

Look for cool addons from ofxaddons.com



Bonus Example: ofxKinect!
examples/addons/kinectExample



1st Generation
(Kinect for Xbox 360)
Only buy model 1414!



2nd Generation
(Kinect for Xbox One)
Higher performance,
Harder to use!

The screenshot shows a web browser window with the URL openframeworks.cc in the address bar. A red banner at the top reads: "Work in progress! This is a preliminary version of ofBook, a collaboratively written book about openFrameworks. Please post any issues, suggestions, comments on our [repo](#)." The main content area is titled "Foreword" in large, bold, black font. Below the title, it says "by [Zach Lieberman](#)". The text discusses the history of openFrameworks, mentioning its start around 2004/2005, its initial use of tools like Director/Lingo and Flash, and its connection to the ACU library developed at MIT. It also notes the transition to Processing. On the left side, there is a sidebar with a navigation menu:

- Foreword (selected)
- [about this book](#)
- [credits](#)
- Basics
- Approaches
- I/O
- Graphics
- C++
- Advanced topics
- Platforms
- Tools
- Case studies

What's next?

<http://openframeworks.cc/ofBook/chapters/foreword.html>

Homework

Choose (at least) one of the following:

- Instead than display the detected blobs, use computer vision to do some interesting things.
- Try to make your oF app talk with other programs (such as Max/MSP, Ableton Live, MadMapper, etc. Use ofxOSC or ofxSyphon to do these.)