FINWE

Orion360 SDK For iOS

User Guide

v.1.0.0

Table of Contents

Disclaimer

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1 Foreword

Take a moment to think about the current status of viewing media content with a mobile device: photographs, videos, games, multimedia presentations, exhibitions – all these forms of art are seen through a small frame called a phone or a tablet. Everything is locked inside this frame, and there is no way to see what surrounds the camera; no method to experience what it feels like *to be there*. Or as Steven Spielberg put it[1]:

"We're never going to be totally immersive as long as we're looking at a square, whether it's a movie screen or a computer screen. We've got to get rid of that and put the player inside the experience; where no matter where you look you're surrounded by a three-dimensional experience. That's the future."

Imagine a presentation tool that could remove this boundary, and let people see the story told in full 360° view. Such tool would allow the user himself to decide where to look at and when. It would let him play the movie multiple times, each being a different, unique experience. And most remarkably: to begin interacting with the elements, and become an active participant in unwrapping the layers of the story.

We have created exactly that tool, and it has a name: *Orion360*. It is the most advanced and best performing solution for 360° video, and with support for 360° panoramic images, it takes you a long way forward towards Spielberg's vision of future. Welcome to the future.

Now it is up to you, the developer, to bring all this to life. Build your own story, and let people experience it with this revolutionizing style. Make it beautiful, and be bold! Reach for the limits of this new way of storytelling.

We thank you for choosing Orion360 and Finwe Ltd. as your partners for this journey. We believe you have made the right choice, and promise to do our best in helping you build apps that takes their breath away, makes them wonder how is this possible at all, and keeps them asking for more. In case you need a piece of advice of which route to take or a helping hand when in front of an obstacle, we are there to give you a hand.

Let's change the world.

Tapani Rantakokko

Orion360 Project Lead

Finwe Ltd., Oulu, Finland

[1] http://www.nytimes.com/2013/06/23/opinion/sunday/movies-of-the-future.html

2 Introduction

2.1 What is Orion360

In short, *Orion360 SDK For iOS* provides simple but powerful API for easily embedding 360° media to 3rd party iOS applications.

The SDK is based on our *Orion360 Engine*, an optimized library for 360° content. The engine provides fully hardware-accelerated path from video decoding to projection to rendering on screen. It is built upon the following principles:

- *Lightweight*
  Small binary size, low memory consumption
- *Ease of use*
  Simple API, few dependencies to external libraries
- *Best performance*
  High UI frame rate, responsive platform-agnostic sensor fusion
- *Best image quality*
  Fast, precise polygon based rendering
- *Natural interaction*
  Simultaneous gyro and touch based panning, zooming and rotating with Auto Horizon Leveler (AHL) feature
- *Fully configurable*
  Customizable content, graphics, interaction
- *Available for multiple platforms as a service or as an SDK*
  Android & iOS available now, PC/Mac/Linux on request

Orion360 SDK brings the power of the Orion360 Engine to the hands of 3rd party developers via easy-to-use API, code examples, user manual, and support.

2.2 Requirements

Orion360 SDK For iOS supports devices running iOS 7.0 or later.

In addition, the following is recommended:

- *Large high-resolution display*
  Video is best viewed from a large screen with enough resolution for details and space for performing touch gestures. For VR usage with a HMD frame, Full HD or higher resolution is recommended
- *Support for 1080p video playback*
  Orion360 Engine can use Full HD resolution video content, which is also currently the recommended video format
- *Support for 4k video playback (h.264 profile 5.1)*
  Orion360 Engine can use 4k resolution video content, which provides the best viewing

experience on supported devices

2.3 Video Formats

Orion360 SDK For iOS uses AV Foundation framework for video playback. Videos that work with the native iOS media player generally work with Orion360 as well.

Be aware that video playback support depends on the hardware used. As of June 2015, low-end iOS devices typically support 720p video (1280x720, h.264 profile 3.1), middle and high-end devices support 1080p (1920x1080, h.264 profile 4.1), and latest top devices can play 4k videos (3840x2160, h.264 profile 5.1).

Currently the recommended media format for Orion360 based projects is **h.264 encoded Full HD video (1920x1080, h.264 main profile 4.1)**. We advise configuring video encoder for **streaming and progressive downloading**. If necessary, use a video transcoding tool such as *Handbrake*[2] (free) for encoding the videos.

For more information about iOS's core media formats and video encoding recommendations, see iOS developer documentation[3].

Tip: Automatic recognition of device's video capabilities is currently not feasible. For public distributions we advice to select a tested safe video format, or let end-users choose the stream quality.

Tip: Beware of anamorphic videos that stretch during playback time. For example, setting aspect ratio to 2:1 with 1920x1080 resolution can result to video being stretched to 2160x1080 resolution during playback. This can be problematic as iOS devices have a hardware limit for maximum texture size typically at 2048x2048 or 4096x4096 pixels.

2.4 Image Formats

Orion360 SDK For iOS supports also panoramic images. The images must be stored in PNG or JPG format. See "Image Viewer" example application for details.

Tip: In case image resolution is larger than device's maximum texture size (typically 2048x2048 or 4096x4096), it will be automatically scaled down to fit inside a single texture.

2.5 Media Sources

Orion360 SDK For iOS supports the following media sources:

- Embedding media to app package (.ipa)
- Streaming media from network
- Copying or downloading media to local memory

App packages submitted through iTunes Connect has a size limit of 4GB. However, note that installation of big packages will take time (because of the cellular network delivery size limitations). Embedding media to app package removes the need for own content delivery network, but has drawbacks:

[2] http://handbrake.fr/

[3] https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/MediaLayer/MediaLayer.html

- User cannot remove any media delivered inside an app package without uninstalling the whole application.

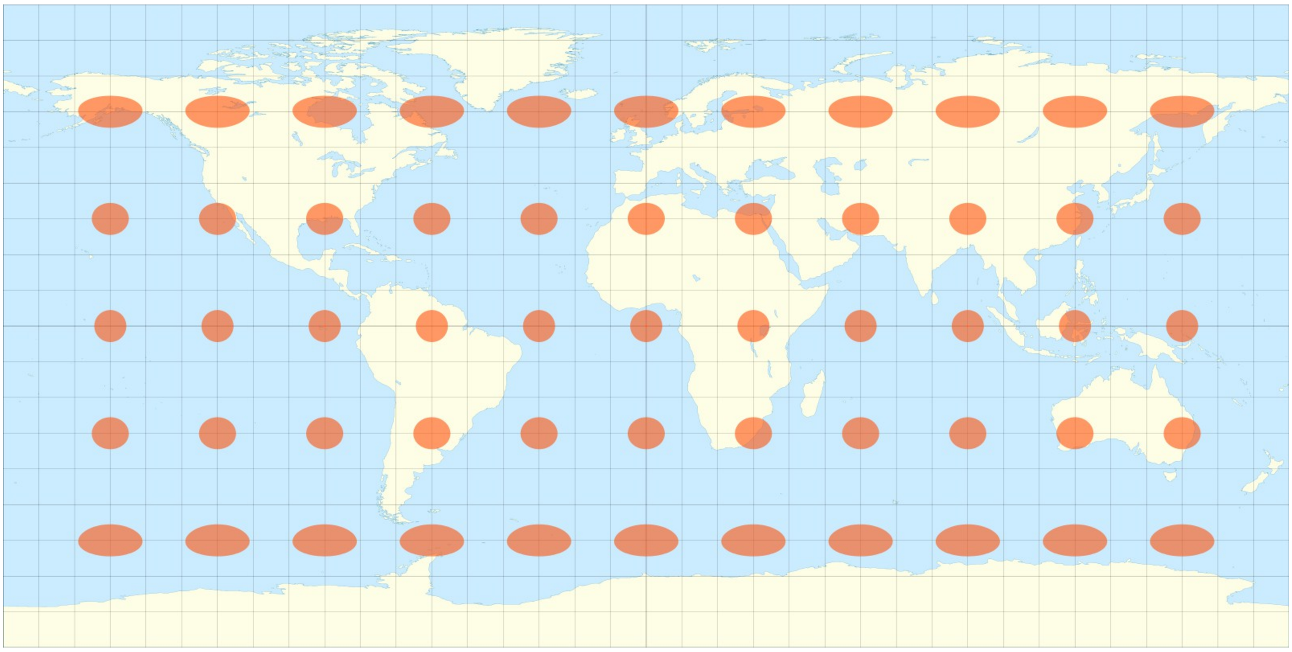
If you plan to stream videos from a network, supported streaming protocols can be found from iOS developer documentation[4].

Tip: Progressive streaming of video files that are hosted on a server running Apache web server works well over local Wi-Fi network; this is handy when developing with multiple devices.

2.6 Content Projection

Orion360 SDK For iOS v1.0.0 supports only spherical content. This means that videos and images should cover full angular field of 360 x 180 degrees.

All content is currently expected to utilize the most common panoramic projection, the **equirectangular projection**[5]. Equirectangular projection is neither equal area nor conformal. As seen in the illustration below, strong distortion is introduced by this projection. It is common because of the particularly simple relationship between the position of a source pixel and its corresponding location on the rendering surface.



© Eric Gaba, Wikipedia

It you wish to use the **cubic projection** / cube mapping[6] instead, contact us at support@finwe.fi.

Tip: Orion360 is flexible with content dimensions; all the pixels will be used and the 2:1 ratio typical to spherical panoramic photographs is NOT required. You can use for example the ordinary Full HD resolution 1920x1080 (1.77:1 ratio) or a more "panorama-friendly" resolution 1920x960 (2:1 ratio). The difference in image quality is usually negligible.

[4] https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/Introduction/Introduction.html

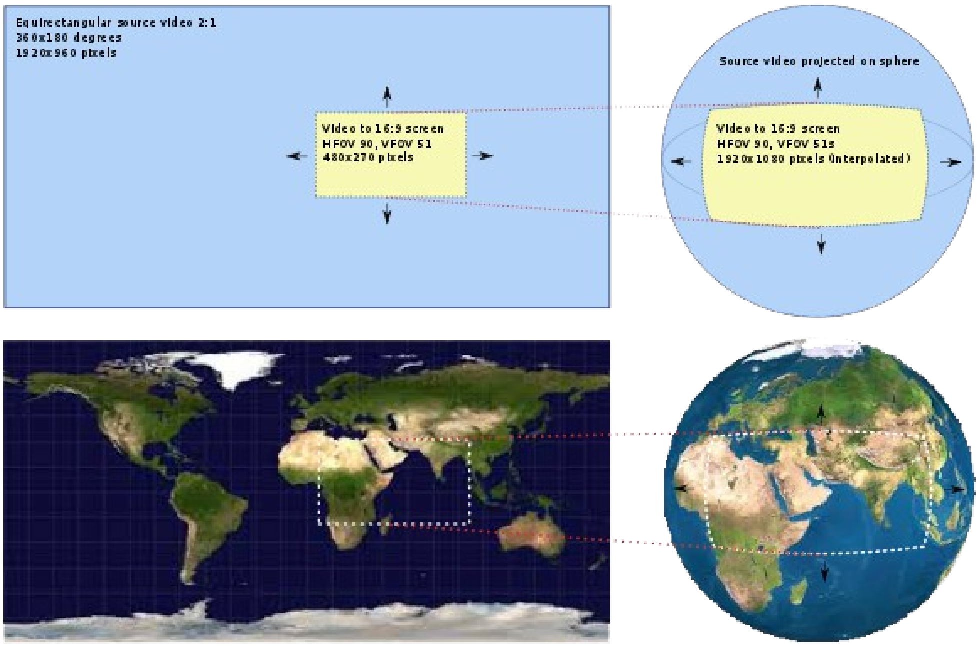
[5] http://en.wikipedia.org/wiki/Equirectangular_projection

[6] http://en.wikipedia.org/wiki/Cube_mapping

2.7 Content Production

As a general rule for creating media content, the producer should **aim for the highest possible resolution and quality**. Depending on the selected projection type, end users will typically see only a fraction of each video frame / image at a time. This is similar to looking at a zoomed-in view where all the details of the footage get exposed. It is most notable in VR mode where image is viewed via magnifying lenses of a HMD frame.

For illustration of the above, consider a video that has a resolution of 1920x960 pixels (2:1 ratio) and that contains full spherical image, i.e. 360x180 degrees. Also consider a smartphone that has Full HD resolution display (1920x1080 pixels, 16:9 ratio). If the view is zoomed so that it covers, say, 90 degree horizontal FOV (Field-of-view), then vertical FOV will be about 51 degrees. This corresponds to **only 480x270 pixels** of the source image, taken somewhere from the video depending on current panning position, and eventually interpolated to 1920x1080 target resolution to cover the whole screen. Notice that this is a highly simplified illustration; in practice one cannot simply copy a segment of the source video to display buffer due to distortion in the source image and required projection.



This is the reason why panoramic imaging professionals are always among the first to adopt higher resolutions when they become available. As of June 2015, only the latest high-end devices support 4k video playback. If a 4k video were used in the example above, viewport source resolution would increase from 480x270 to 960x540 pixels – a clear improvement but still far from native display resolution. In fact, an 8k video is needed to match with a Full HD display, and 90 degree horizontal FOV used in above calculations is still quite a wide angle.

Orion360 supports adjusting 3D camera field-of-view (FOV) i.e. zooming in/out. Be aware that zooming in requires high-resolution material to look decent, and zooming out far introduces significant image distortion at the edges of the viewport.

The video producer should design camera locations so that the camera is in the center of the action, thus providing interesting viewpoints to multiple directions and encouraging the viewer to look around. Getting "inside" is essential to make good use of this medium.

Orion360 supports panning around freely in true spherical form (360° x 360°), meaning that there are no artificial barriers at North and South poles that prevent moving the view any further. While this method gives the user full freedom to choose where to look at and pan around, it also frequently leads to tilting the horizon unintentionally. Consider the case where you first look straight ahead, then turn the 3D camera straight down, and then raise the camera right or left up to the level of the horizon: the World is now tilted 90 degrees! Orion360 includes a special aid for this particular problem: *Auto Horizon Leveler* (AHL) is a feature to keep the horizon in level.

When creating 360° video using a camera rig and a stitching process, be sure to keep enough empty space around the cameras as the parallax effect will produce nasty image stitching errors for objects that are too close to the camera, and these are difficult to fix.

As with any video material, multiple camera positions should be used and the final movie should be cut into a sequence of short scenes to keep up the action and to make it more interesting for viewers.

2.8 Hardware Performance

Video viewing performance strongly depends on the used hardware. There are two important bottlenecks:

Video decoding speed
This mostly depends on content resolution, frame rate, bit rate and other encoding parameters. If content is beyond device's capabilities, depending on device's implementation, video playback may skip frames, play only audio channel or play nothing at all. Be sure to use content that is within the device's limits.

User Interface (UI) update speed
This mostly depends on device's display resolution. Orion360 SDK For iOS uses polygon based spherical projection which provides best performance and in general works well on all devices. In order to provide some guidance on how devices currently on the market perform, we have measured 360° video playback performance with a number of devices.

The results show that when using polygon based spherical projection, most devices achieve 50-60 FPS UI update speed with ease. The downsides of polygon based projection are 1) straight lines cannot be fully preserved, and 2) image distortion appears at nadir and zenith (South and North poles of the sphere). Orion360 uses large, high-quality sphere model that provides a good compromise between image quality and speed. It is recommended for general purpose 360° video playback especially if a large number of device models need to be supported, and is the only alternative if the developer plans to support low-end devices.

2.9 Development Environment

Orion360 SDK requires Mac computer running OS X 10.9.4 or later and Apple's integrated development environment Xcode. The iOS SDK is included with the Xcode. Orion360 SDK will be integrated to you projects using Cocoapods. Thus, you also need to have Cocoapods installed to your Mac.

2.10 Application Distribution

Orion360 library binaries can be distributed as a part of the developer's own application to enable its intended usage. Note that **all SDK versions require using a license file. You must include the unaltered license file(s) to your application.** Failing to do this will lead into blank video/photo view.

2.11 License Restrictions

Orion360 SDK For iOS license agreement may limit available features and distribution rights of the binary modules. Refer to the license agreement document for details (included in SDK package).

As an example, depending on the SDK version and agreement made with the customer, Orion360 library may contain visual watermarks and technical limitations that reduce available features or limit support to certain application package namespaces.

We endorse openness and free software, and may some day release also Orion360 as free software. Currently we need to protect our IPR in order to be able to develop this magnificent technology further, and to provide our users support that they deserve. We wish that you respect our efforts and ask for permission or buy a license for needs that are not covered by your current license agreement.

3 Getting Started

3.1 SDK Contents

The Orion360 SDK For iOS contains the following entities:

- *Orion360_SDK_UserGuide.pdf*
  Orion360 SDK user manual (this document)

- *Orion360_SDK_API.zip*
  Orion360 API documentation in HTML format

- *Orion360_SDK_SampleApps.zip*
  Orion360 example Android projects

- *Orion360_SDK_LicenseAgreement.pdf*
  Legally binding license agreement

- *Orion360_SDK_ReadMe.txt*
  Release notes

- **.lic (Optional)*
  The SDK may contain one or more license files, to be copied to the application's resources

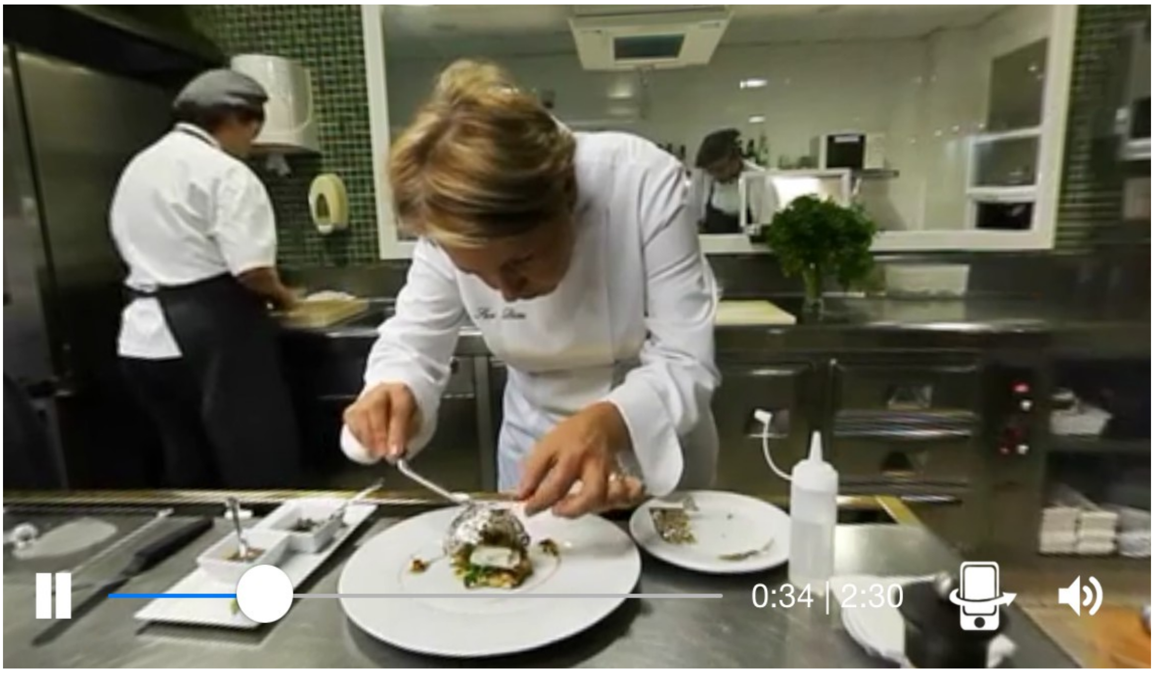
The documents and example codes listed above are delivered as a zip package named as *Orion360_SDKforIOS_vX.Y.Z.zip* (XYZ revealing SDK version). The *Orion360 library* is available on **GitHub** (https://github.com/FinweLtd/OrionSDK_iOS_Prd) and is integrated into the example applications using *CocoaPods*.

3.2 First Steps

We recommend reading first this User Guide document, and following the instructions on creating a 'Hello Orion360' example project (Chapter 3).

Next we advise skimming through included sample projects, building them, and testing them on a real device.

Finally, the developer should read the included API docs.



Orion SDK for iOS example app "Advanced Player".

4 Hello Orion360

4.1 Introduction

This chapter introduces the developer to the basics of using Orion SDK For iOS. We will create a simple demo app "*Hello Orion360*" step by step with clear instructions and screenshots.

After completing this exercise, the developer should be ready for integrating *Orion library* to his own iOS projects. Yet we advise to learn more by examining the sample applications and reading API docs.

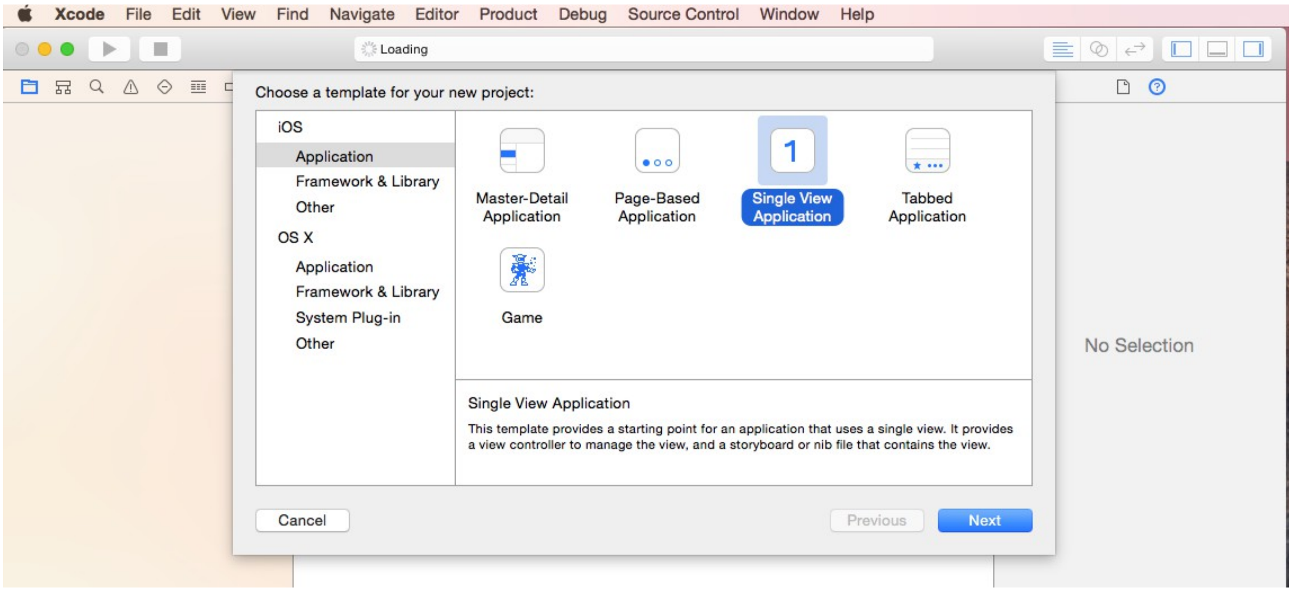
4.2 Prerequisites

The developer should have good knowledge of the iOS programming language (objective C) and be familiar with the iOS development and tool chain. In addition, the developer should have:

- An iOS phone or tablet with OS 7.0 or later
- A Mac computer running OS X 10.9.4 or later
- Xcode (latest version)
- iOS SDK (included with the Xcode)

4.3 Step1: Create a New iOS project

Open Xcode and select *File > New > Project...* (or *Create a new Xcode project* from the Welcome dialog).

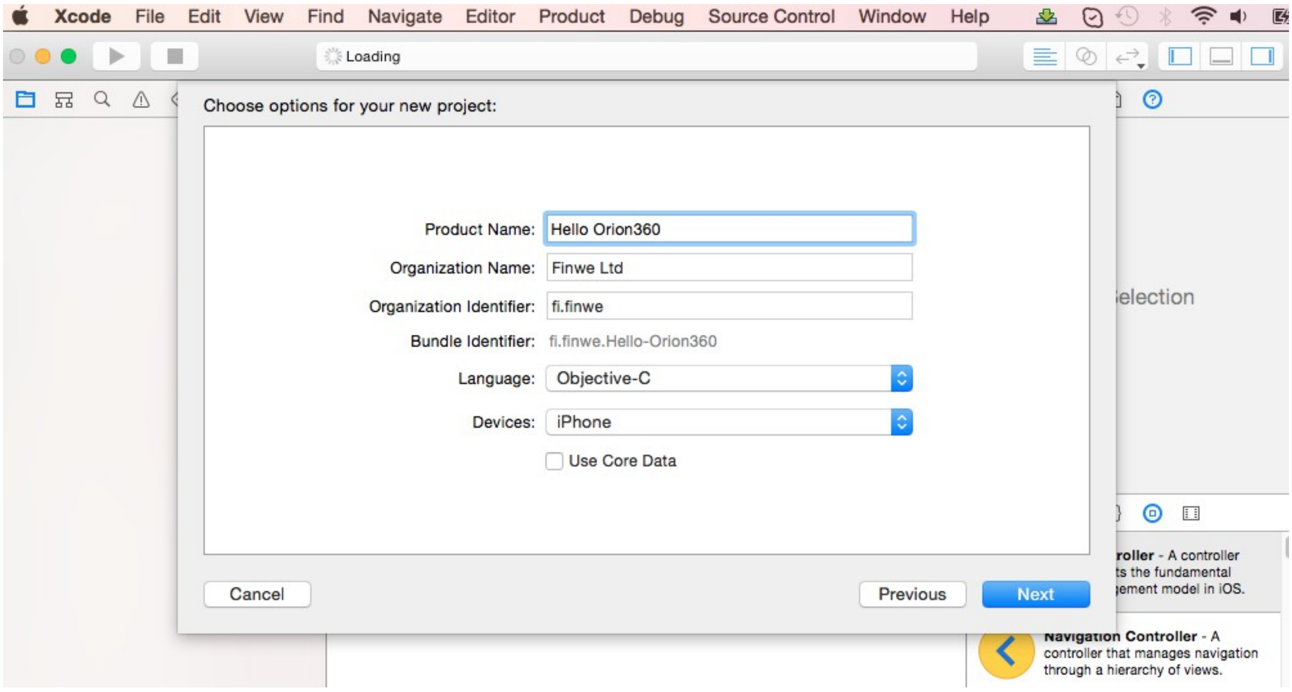


Single View Application

This template provides a starting point for an application that uses a single view. It provides a view controller to manage the view, and a storyboard or nib file that contains the view.

Select *Application* from the left section under the *iOS* and then, in the main area of the dialog, click *Single View Application*.

Click **Next**.



Choose options for your new project:

Product Name: Hello Orion360

Organization Name: Finwe Ltd

Organization Identifier: fi.finwe

Bundle Identifier: fi.finwe.Hello-Orion360

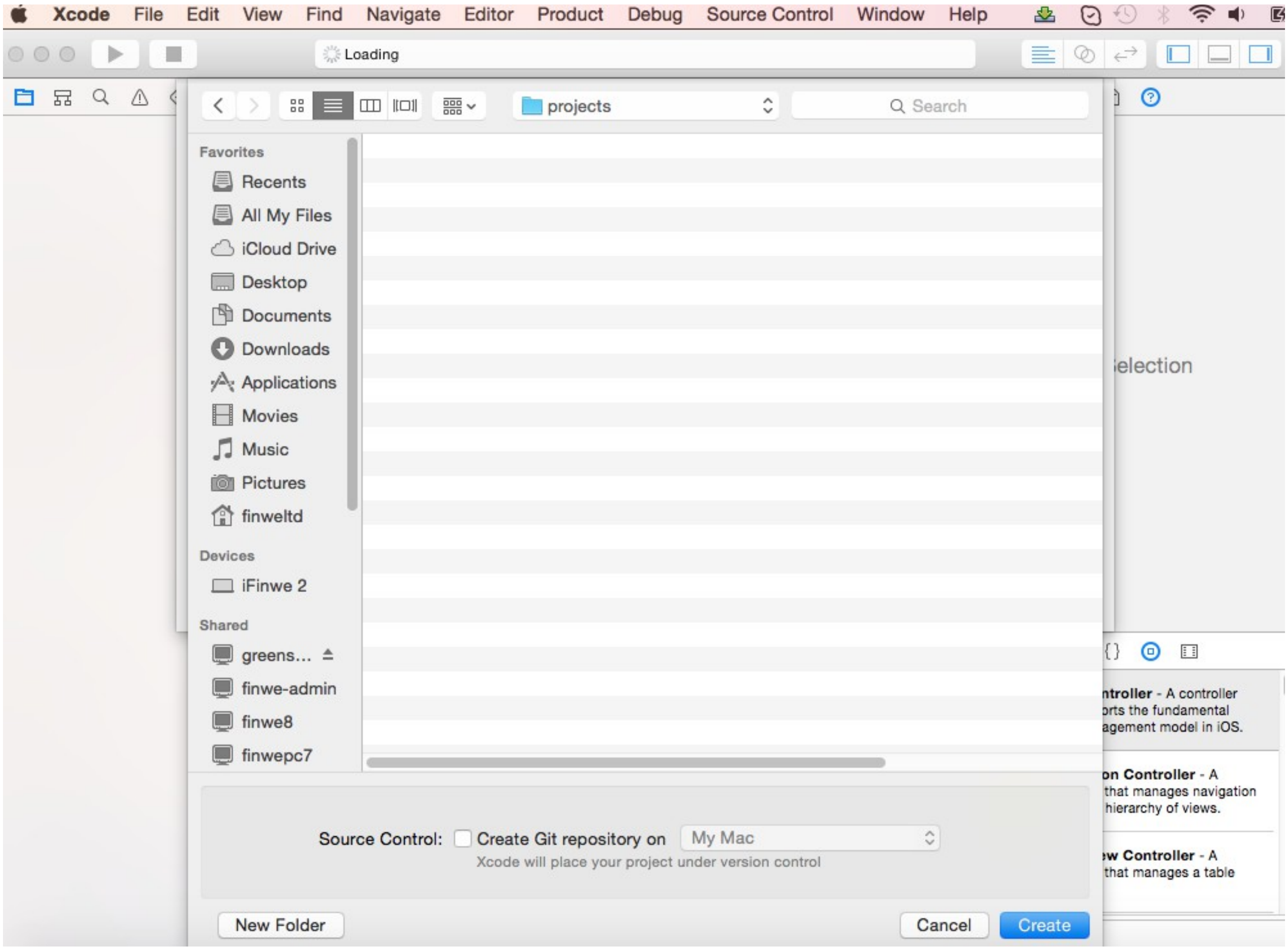
Language: Objective-C

Devices: iPhone

□ Use Core Data

Enter Product name “*Hello Orion360*”, Organization name “*Finwe Ltd*” (you can leave this blank) and Organization identifier “*fi.finwe*”. Bundle identifier is automatically generated based on the product name and organization identifier. Leave the rest of the settings as-is.

Click **Next**.

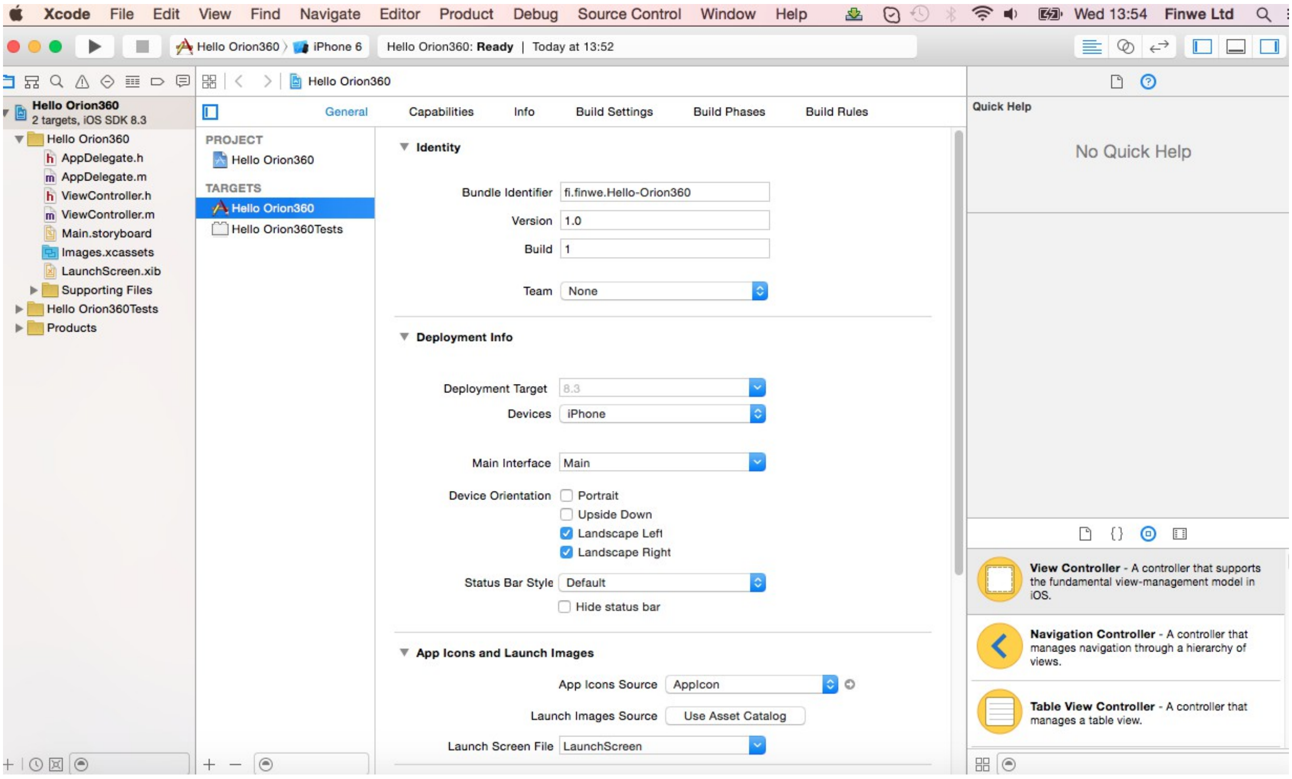


Source Control: □ Create Git repository on My Mac

Xcode will place your project under version control

Select the folder where you wish to save your project.

Click **Create**.


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Wed 13:54 Finwe Ltd
Hello Orion360 〉 iPhone 6
Hello Orion360: Ready | Today at 13:52
Hello Orion360
Hello Orion360
2 targets, iOS SDK 8.3
Hello Orion360
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
Main.storyboard
Images.xcassets
LaunchScreen.xib
Supporting Files
Hello Orion360Tests
Products
General
Capabilities
Info
Build Settings
Build Phases
Build Rules
PROJECT
Hello Orion360
TARGETS
Hello Orion360
Hello Orion360Tests
Quick Help
No Quick Help
View Controller - A controller that supports the fundamental view-management model in iOS.
Navigation Controller - A controller that manages navigation through a hierarchy of views.
Table View Controller - A controller that manages a table view.

Identity

Bundle Identifier fi.finwe.Hello-Orion360
Version 1.0
Build 1
Team None

Deployment Info

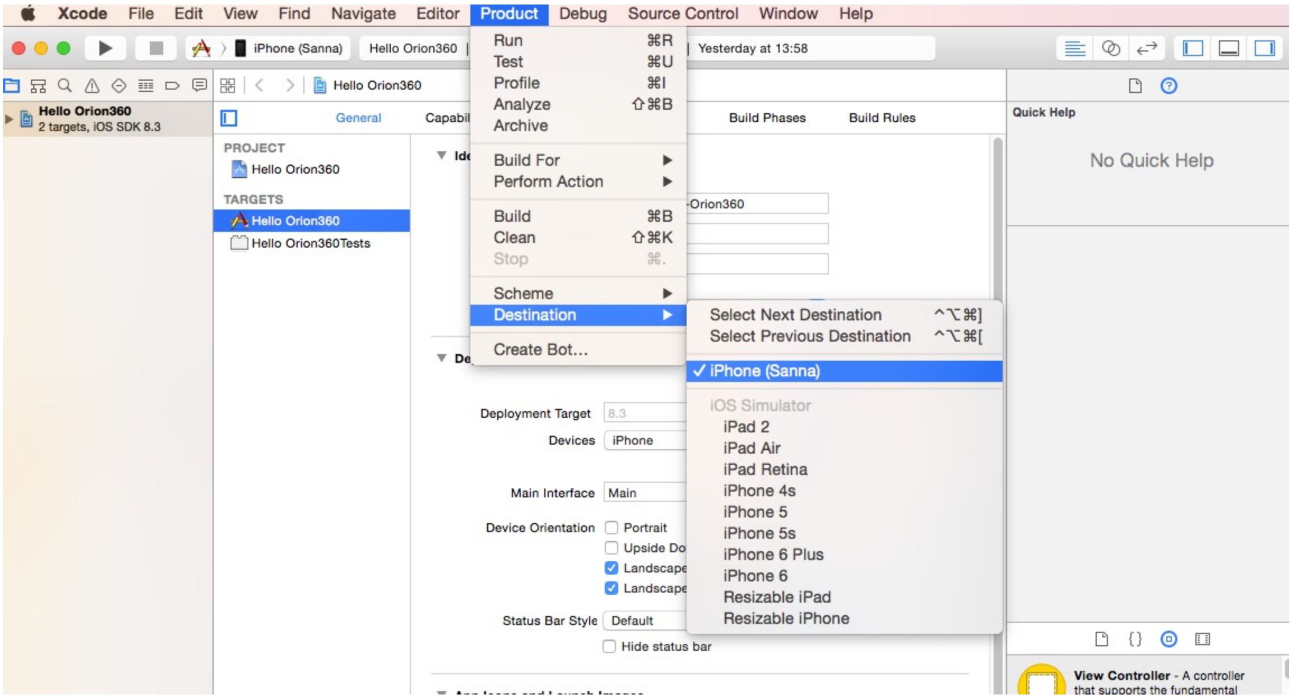
Deployment Target 8.3
Devices iPhone
Main Interface Main
Device Orientation ☐ Portrait
☐ Upside Down
☑ Landscape Left
☑ Landscape Right
Status Bar Style Default
☐ Hide status bar

App Icons and Launch Images

App Icons Source AppIcon
Launch Images Source Use Asset Catalog
Launch Screen File LaunchScreen

Xcode opens your new project in the workspace window.

We will make this application to appear in landscape orientation. This can be done by unselecting "**Portrait**" from device *Orientation* -section and leaving "**Landscape Left**" and "**Landscape right**" selected (see image above).



-Orion360
Deployment Target 8.3
Devices iPhone
Main Interface Main
Device Orientation ☐ Portrait
☐ Upside Do
☑ Landscape
☑ Landscape
Status Bar Style Default
☐ Hide status bar

Next, connect your iOS device to the computer via USB cable. From Xcode, select *Product > Destination > "Your device"* and then *Product > Run* from the pop-up menu to run the application on device.

Hello Orion360

Copyright (c) 2015 Finwe Ltd. All rights reserved.

The application will be now compiled and and installed to your device. After it is started, you will see a default launch image (somewhat similar to the one shown above) and then a default user interface.

You are now ready to start integrating Orion360 video view to your application.

4.4 Step 2: Add Orion360 Library to the Project

Orion360 library will be integrated to your project using a tool called Cocoapods (https://cocoapods.org). It is a dependency management tool that fetches a library source code, resolves dependencies between different libraries, and even creates and maintains the right environment to build your project.

To continue, you first need to install Cocoapods. Open Terminal application and type command

```
sudo gem install cocoapods
```

and press **Enter**.

Now you should have the *pod* command available in the terminal. The next step is to create a *Podfile* to your project (the *Podfile* is where the dependencies of you project are defined).

Close Xcode. Open Terminal application if not yet open and navigate to the folder where your Hello Orion360 project file (*Hello Orion360.xcodeproj*) exists and type command

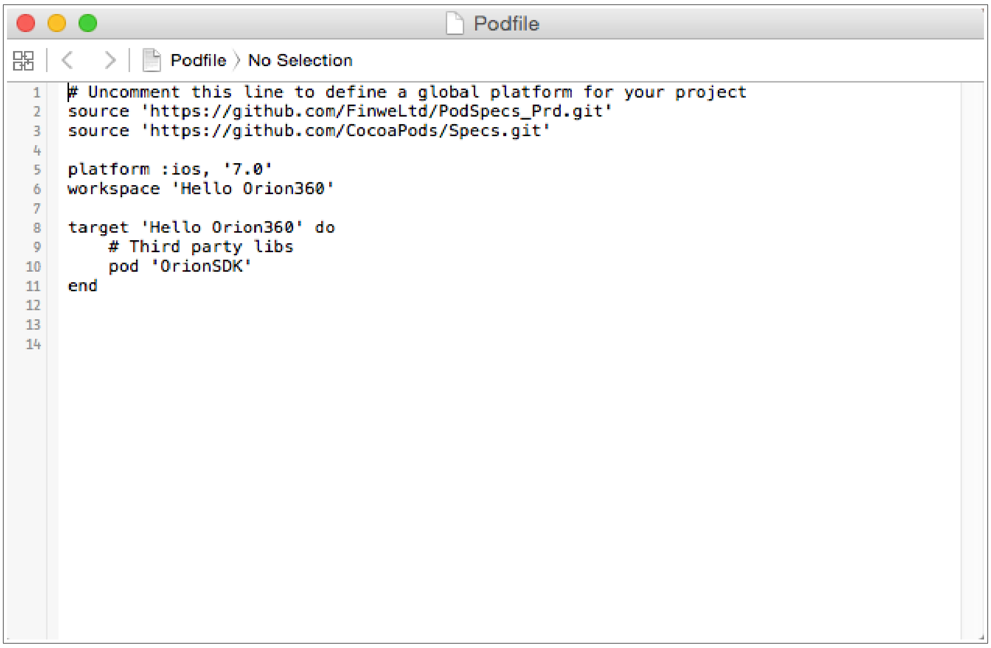
```
pod init
```

and press **Enter**. This command will create a default Podfile for your project.

Then type command

open -a Xcode Podfile

and press **Enter** to open the Podfile with Xcode for editing.

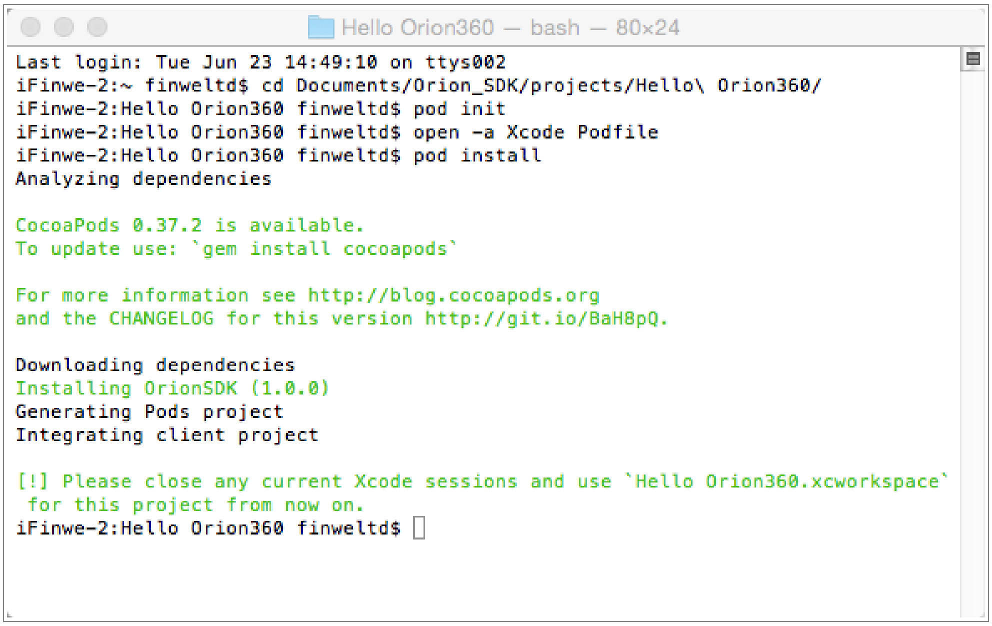


```
# Uncomment this line to define a global platform for your project
source 'https://github.com/FinweLtd/PodSpecs_Prd.git'
source 'https://github.com/CocoaPods/Specs.git'

platform :ios, '7.0'
workspace 'Hello Orion360'

target 'Hello Orion360' do
    # Third party libs
    pod 'OrionSDK'
end
```

Modify the opened Podfile to look like the Podfile above. Then save and close it. Go back to the terminal and type

pod install

and press **Enter**.



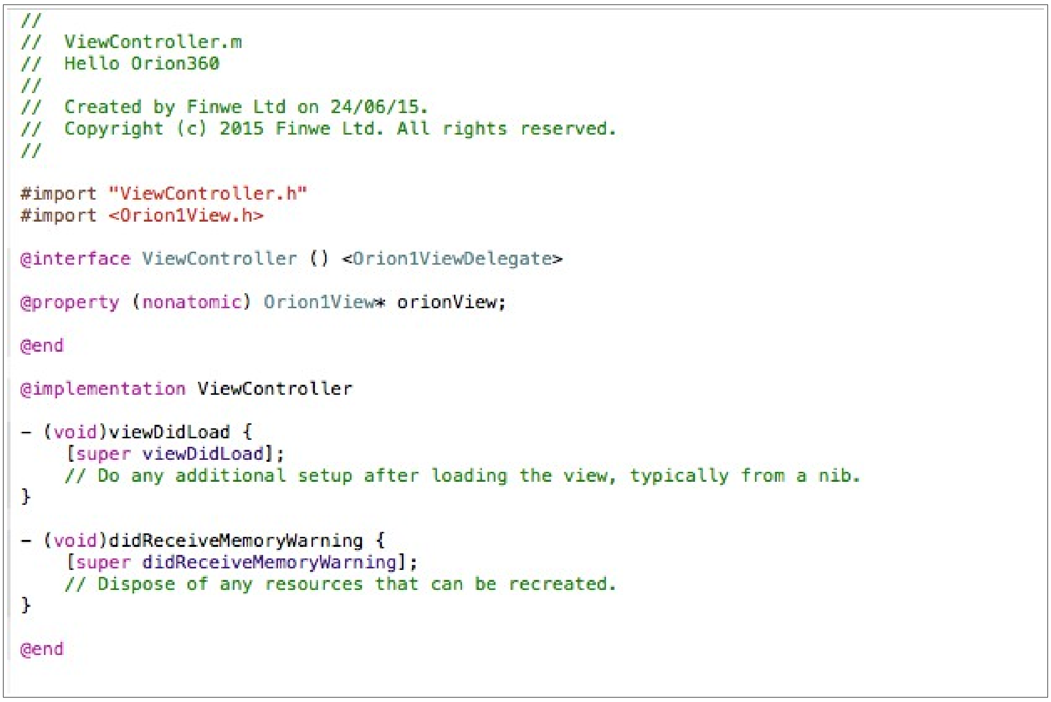
```
Last login: Tue Jun 23 14:49:10 on ttys002
iFinwe-2:~ finweltd$ cd Documents/Orion_SDK/projects/Hello\ Orion360/
iFinwe-2:Hello Orion360 finweltd$ pod init
iFinwe-2:Hello Orion360 finweltd$ open -a Xcode Podfile
iFinwe-2:Hello Orion360 finweltd$ pod install
Analyzing dependencies

CocoaPods 0.37.2 is available.
To update use: `gem install cocoapods`

For more information see http://blog.cocoapods.org
and the CHANGELOG for this version http://git.io/BaH8pQ.

Downloading dependencies
Installing OrionSDK (1.0.0)
Generating Pods project
Integrating client project

[!] Please close any current Xcode sessions and use `Hello Orion360.xcworkspace`
 for this project from now on.
iFinwe-2:Hello Orion360 finweltd$ 
```

Now you should have pods installed to your project. From now on it is very important that you open the workspace (created as pods were installed) and **NOT** the project.

Now, if you have Xcode project still open, close it. Then go to the *Hello Orion360* folder and double click *Hello Orion360.xcworkspace* file to open the workspace.

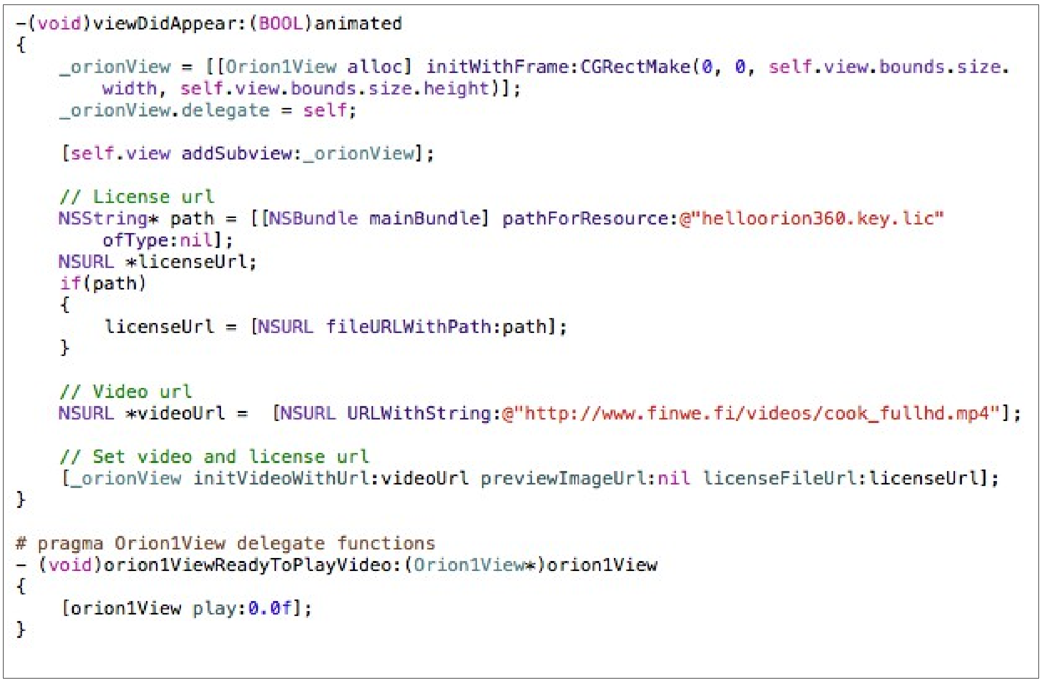
4.5 Step 3: Add Orion1View to ViewController

Open *Hello Orion360.xcworkspace* if not yet open. From the left navigator area, select *Hello Orion360* and under it *Hello Orion360* folder. Open **ViewController.m** by selecting it from the navigator. Note that in the left project navigator, there now exists also item **Pods**. Under it, there is folder **Pods** including folder **OrionSDK. OrionSDK** includes header **Orion1View.h**. We will now import it into your **ViewController.m**.



```
//
//  ViewController.m
//  Hello Orion360
//
//  Created by Finwe Ltd on 24/06/15.
//  Copyright (c) 2015 Finwe Ltd. All rights reserved.
//

#import "ViewController.h"
#import <Orion1View.h>

@interface ViewController () <Orion1ViewDelegate>

@property (nonatomic) Orion1View* orionView;

@end

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

@end
```

Modify the **ViewController.m** file by importing **Orion1View.h** and adding **Orion1ViewDelegate** as well as property **Orion1View* orionView**.



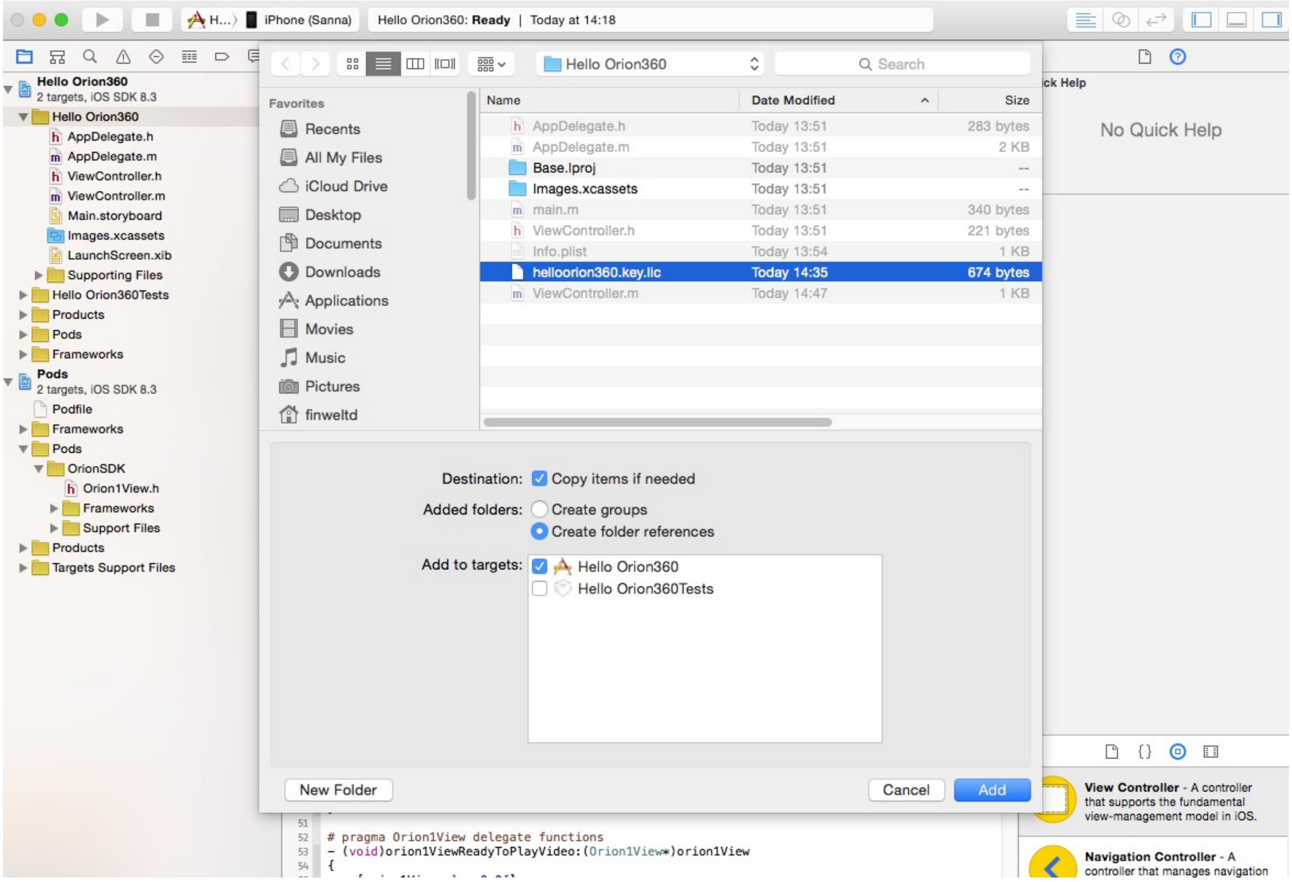
```
-(void)viewDidAppear:(BOOL)animated
{
    _orionView = [[Orion1View alloc] initWithFrame:CGRectMake(0, 0, self.view.bounds.size.
        width, self.view.bounds.size.height)];
    _orionView.delegate = self;

    [self.view addSubview:_orionView];

    // License url
    NSString* path = [[NSBundle mainBundle] pathForResource:@"helloorion360.key.lic"
        ofType:nil];
    NSURL *licenseUrl;
    if(path)
    {
        licenseUrl = [NSURL fileURLWithPath:path];
    }

    // Video url
    NSURL *videoUrl =  [NSURL URLWithString:@"http://www.finwe.fi/videos/cook_fullhd.mp4"];

    // Set video and license url
    [_orionView initVideoWithUrl:videoUrl previewImageUrl:nil licenseFileUrl:licenseUrl];
}

# pragma Orion1View delegate functions
- (void)orion1ViewReadyToPlayVideo:(Orion1View*)orion1View
{
    [orion1View play:0.0f];
}
```

Then add **ViewDidAppear** method and Orion1View delegate method **orion1ViewReadyToPlayVideo** as above.

4.6 Step 4: Add License Files

Locate the license file for the *Hello Orion360* application from the root of the SDK (*helloorion360.key.lic*). Copy it (without making any changes to the file) to the **~/Hello Orion360/Hello Orion360** folder (same folder where **ViewController.m** exists).

Open *Hello Orion360.xcworkspace* again.



Name	Date Modified	Size
AppDelegate.h	Today 13:51	283 bytes
AppDelegate.m	Today 13:51	2 KB
Base.lproj	Today 13:51	--
Images.xcassets	Today 13:51	--
main.m	Today 13:51	340 bytes
ViewController.h	Today 13:51	221 bytes
Info.plist	Today 13:54	1 KB
helloorion360.key.lic	Today 14:35	674 bytes
ViewController.m	Today 14:47	1 KB

Destination: ☑ Copy items if needed

Added folders: ○ Create groups
● Create folder references

Add to targets: ☑ Hello Orion360
□ Hello Orion360Tests

```
# pragma Orion1View delegate functions
- (void)orion1ViewReadyToPlayVideo:(Orion1View*)orion1View
{
```

Select *Hello Orion360* folder from the project navigator and click right mouse button. From the pop-up menu, select "*Add Files to "Hello Orion360*"...". From file selection dialog, navigate to **~/Hello Orion360/Hello Orion360/** folder and select file *helloorion360.key.lic*.

Click **Add**.

4.7 Build & Test

Build the application again (make sure that your device is connected). Select *Product* > *Destination* > *"Your Device"* from the pop-up menu and then *Product* > *Run* (or click Build and Run button). You should see that 360 video playback begins automatically when the app starts.

If your device has all three sensors, you can pan around simply by moving the device around you. Notice that the video view will adapt to your devices UI orientation changes (portrait, landscape). With touch, you can pan around (drag) and zoom (pinch-to-zoom). Automatic Horizon Leveler (AHL) keeps the horizon straight when panning with touch.

4.8 Next Steps

The developer should now be familiar with the minimum steps required for using 360 videos in an app. We recommend that you test the provided sample apps. Run pod install and open the workspace with Xcode, build and run them, and read the code to learn some useful practices.

To try out one of the sample applications, do the following:

- Open the Terminal application and navigate to the folder where the application's *Podfile* exists. Type *pod install.*
- Open finder, navigate to the folder where the application's *Podfile* exists and double click the created *workspace* file to open workspace with the Xcode.
- Build and run the application on your iOS device. Then study it's source code, and try to modify it as you wish. The API docs should be a helpful resource when customizing the player.

The following examples are included to the SDK:

- *ExampleMinimalPlayer*
  Very simple player project, feature-wise close to Hello Orion360 exercise.
- *ExampleAdvancedPlayer*
  Rich player that demonstrates many features of the SDK. Options can be selected via buttons.
- *ExampleImageViewer*
  Simple panoramic image player that demonstrates how to use a still image instead of a video.
- *ExamplePreviewImage*
  Simple video player that demonstrates usage of preview image in 360 video view.

5 FAQ

5.1 “I have included a video to the app resources. How do I play it from there”

Create a NSURL path for your video resource (in the same way as was created the path to license file in the *Hello Orion360* example).

```
NSString* path = [[NSBundle mainBundle] pathForResource:@"myVideo.mp4" ofType:nil];
NSURL *videoUrl;
if (path)
{
        videoUrl = [NSURL fileURLWithPath:path];
}
```

And then call initVideoWithUrl method (in the same way as in the *Hello Orion360* example)

```
[_orionView initVideoWithUrl:videoUrl previewImageUrl:nil licenseFileUrl:licenseUrl];
```

6 Troubleshooting

6.1 Video Is Not Showing

If your video is not showing (screen shows blank), make sure that you have copied all the .lic files to your application's resources. You must not alter these files in any way. If you still get an exception, check that you are not trying to use feature that is not covered by your current license. If this is not the case, the license file may be corrupted, contact us at support@finwe.fi

6.2 Certain Video Files Won't Play

Check that video resolution is within device limits. Some devices can play only 720p HD videos, some 1080p FullHD videos, some 4k videos. Check also that video has been encoded with suitable parameters.

7 Support

7.1 Getting Help

Finwe Ltd. Provides support as agreed in the license agreement. At the minimum support level, developers can always contact us by email: support@finwe.fi.

We suggest that you include the following information to your requests:

- Your name, organization and contact information
- iOS device model and software version
- Version number of Xcode that you are using for compiling and building the app
- Version of Orion360 SDK that you are using for 360 video playback
- Problem description and steps to reproduce

7.2 SDK Updates, Feature Requests, and Bug Reports

Orion360 SDK is currently under active development, and new features will be released frequently. Contact us at support@finwe.fi if you want to be notified when new versions become available.

You can use that same email address for posting feature requests and reporting bugs. We'd very much like to hear your opinion!

7.3 Licensing & Business proposals

If you want to use the SDK for something else that was originally agreed or have other business proposal in mind, contact us via sales@finwe.fi.

Glossary

360°

A rotation of 360 degrees makes a full circle. In the context of imaging describes a photograph or a video that covers everything around the camera, although more specific way is to express the horizontal and the vertical viewing angle separately: 360x180 degrees covers all.

AHL

Auto Horizon Leveler, a method developed by Finwe Ltd. for keeping the horizon in level when panning around in a virtual world that is modeled as a sphere around the user.

API

Application Programming Interface, specifies how some software components should interact with each other.

Orion360

Finwe Ltd’s software solution for viewing and controlling 360° content.

Orion360 engine

Optimized library that renders 360° content onto screen and controls the various rendering parameters.

Orion360 SDK

Software Development Kit that enables 3rd parties to use Orion360 Engine in building their own applications.