



Oculus Rift Getting Started Guide

Version 1.11

Copyrights and Trademarks

© 2017 Oculus VR, LLC. All Rights Reserved.

OCULUS VR, OCULUS, and RIFT are trademarks of Oculus VR, LLC. (C) Oculus VR, LLC. All rights reserved. BLUETOOTH is a registered trademark of Bluetooth SIG, Inc. All other trademarks are the property of their respective owners. Certain materials included in this publication are reprinted with the permission of the copyright holder.

Contents

Getting Started with the SDK	4
Recommended Specifications	4
Oculus Rift Setup	4
Oculus Rift SDK Setup	5
Installation	5
Compiler Settings	5
Build Solutions	5
Getting Started with the Demos	6
Software Developers and Integration Engineers	6
Artists and Game Designers	7
OculusWorldDemo Demo	8
OculusWorldDemo Controls	8
OculusWorldDemo Usage	9

Getting Started with the SDK

This guide describes how to install the SDK and try the demos.

Recommended Specifications

Presence is the first level of magic for great VR experiences: the unmistakable feeling that you've been teleported somewhere new. Comfortable, sustained presence requires a combination of the proper VR hardware, the right content, and an appropriate system.

For the full Rift experience, we recommend the following system:

- Graphics Card: GTX 1060 / AMD Radeon RX 480 or greater
- Alternative Graphics Card: NVIDIA GTX 970 / AMD Radeon R9 290 or greater
- CPU: Intel i5-4590 equivalent or greater
- Memory: 8GB+ RAM
- Video Output: Compatible HDMI 1.3 video output
- USB Ports: 3x USB 3.0 ports plus 1x USB 2.0 port
- OS: Windows 7 SP1 64 bit or newer, plus the DirectX platform update

Additionally, make sure you also have the latest GPU drivers:

- [NVIDIA Driver Version 355.83 or later](#)
- [AMD Catalyst Display Driver Version 15.200.1062.1005 or later](#)

The goal is for all Rift games and applications to deliver a great experience on this configuration. Ultimately, we believe this will be fundamental to VR's success, as you can optimize and tune your experiences for a known specification, consistently achieving presence and simplifying development.

Oculus Rift Setup

Before installing the SDK, Oculus recommends making sure the firmware, runtime, and hardware are correctly configured and tested.

To install the runtime and hardware, follow the steps in Oculus Setup. If you are not running the latest firmware while using Oculus Setup, it will automatically install the latest firmware and drivers.

For more information about setting up the Oculus Rift, see <https://www.oculus.com/setup/>.



Note: Because you are doing development work, make sure to enable unknown sources. To enable Oculus to run software from unknown sources, click the gear icon and select Settings. Then, click General and enable Unknown Sources.

Oculus Rift SDK Setup

This section describes how to set up the SDK.

Installation

The latest version of the Oculus SDK is always available from the Oculus Developer Center.

To download the latest package, go to <http://developer.oculus.com>.

SDK versions use a *major.minor.patch* format. For example, 5.0.1 means Major 5, Minor 0, Patch 1.



Note: The instructions in this section assume you have installed the Oculus Rift and software through Oculus Setup.

Compiler Settings

The `LibOVR` libraries do not require exception handling or RTTI support.

Your game or application can disable these features for efficiency.

Build Solutions

Developers can rebuild the samples and LibOVR using the projects and solutions in the `Samples` and `LibOVR/Projects` directories.

Windows

Solutions and project files for Visual Studio 2010, 2012, 2013, and 2015 are provided with the SDK. `Samples/Projects/Windows/VSxxxxx/Samples.sln`, or the 2012/2013/2015 equivalent, is the main solution that allows you to build and run the samples, and LibOVR itself.



Note: The SDK does not support Universal Windows Platform (UWP).

Getting Started with the Demos

Now that the Rift is plugged in, the drivers are installed, and the SDK is installed, you are ready to begin using the SDK.



Note: If you haven't already, take a moment to adjust the Rift headset so that it's comfortable for your head and eyes.

Software Developers and Integration Engineers

If you're integrating the Oculus SDK into your game engine, Oculus recommend starting with the sample projects.

Open the following projects, build them, and experiment with the provided sample code:

- Samples/Projects/Windows/VSxxxx/Samples.sln

OculusRoomTiny

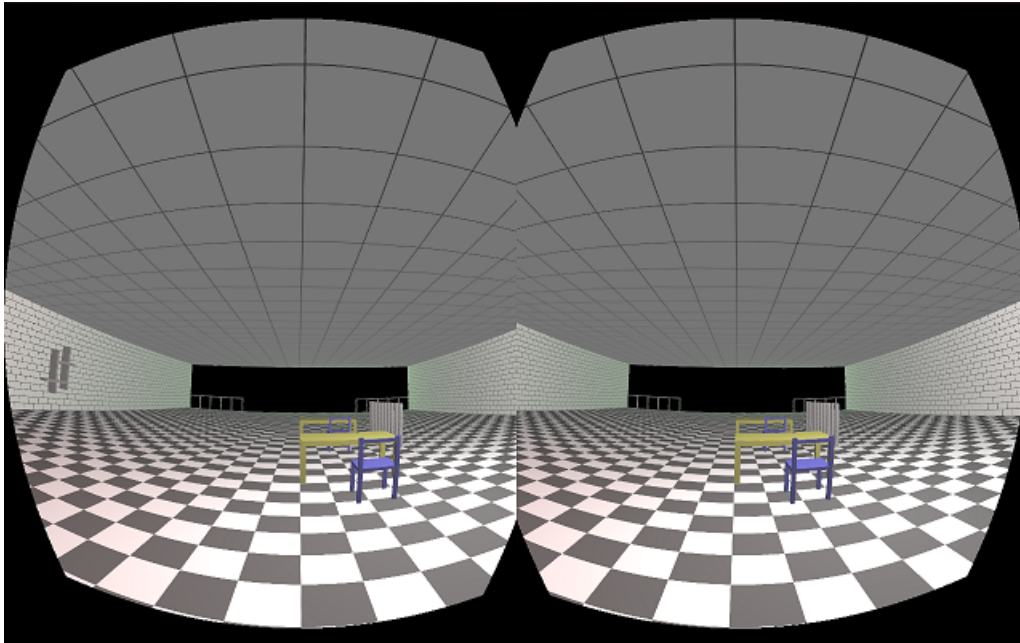
This is a good place to start, because its source code compactly combines all critical features of the Oculus SDK. It contains logic necessary to initialize LibOVR core, access Oculus devices, use the player's profile, implement head-tracking, sensor fusion, stereoscopic 3D rendering, and distortion processing. OculusRoomTiny comes with Direct3D 11, OpenGL and Direct3D 12 variants each with their own separate projects and source files.



Note: The Oculus Room Tiny (DX12) sample projects for each of VS2010 through VS2015 requires that an appropriate Windows 10 SDK be set for the build. If you have a different Windows 10 SDK than what is expected, then you will likely get compile errors about 'missing dx12.h' or similar. To fix this for VS2010 - VS2013, you need to edit the Samples\OculusRoomTiny\OculusRoomTiny (DX12)\Projects\Windows\Windows10SDKPaths.props file with a text editor and change the numbers to refer to your Windows 10 SDK, typically installed with headers at C:>Program Files (x86)\Windows Kits\10\Include.

To fix this for VS2015 you need to edit the Project **Properties** → **General** → **Target Platform Version** drop down box.

Figure 1: OculusRoomTiny



OculusWorldDemo

This is a more complex sample. It is intended to be portable and support many more features. These include windowed/full-screen mode switching, XML 3D model and texture loading, movement collision detection, adjustable view size and quality controls, 2D UI text overlays, and so on.

This is a good application to experiment with after you are familiar with the Oculus SDK basics. It also includes an overlay menu with options and toggles that customize many aspects of rendering including FOV, render target use, timewarp and display settings. Experimenting with these options may provide developers with insight into what the related numbers mean and how they affect things behind the scenes.

When running OculusWorldDemo in Windows, it uses Direct3D 11 by default. However, you can choose the OpenGL rendering path by appending the command-line argument "-r GL" to the executable.

Beyond experimenting with the provided sample code, Oculus recommends reading the rest of this guide. It covers LibOVR initialization, head-tracking, rendering for the Rift, and minimizing latency.

Artists and Game Designers

If you're integrating the Oculus SDK into your game engine, Oculus recommends starting with the sample projects.

If you're an artist or game designer unfamiliar with C++, we recommend downloading Unity along with the corresponding Oculus integration. You can use our out-of-the-box integrations to begin building Oculus-based content immediately.

We also recommend reading through the *Oculus Best Practices Guide*, which has tips, suggestions, and research oriented around developing great VR experiences. Topics include control schemes, user interfaces, cut-scenes, camera features, and gameplay. The *Best Practices Guide* should be a go-to reference when designing your Oculus-ready games.

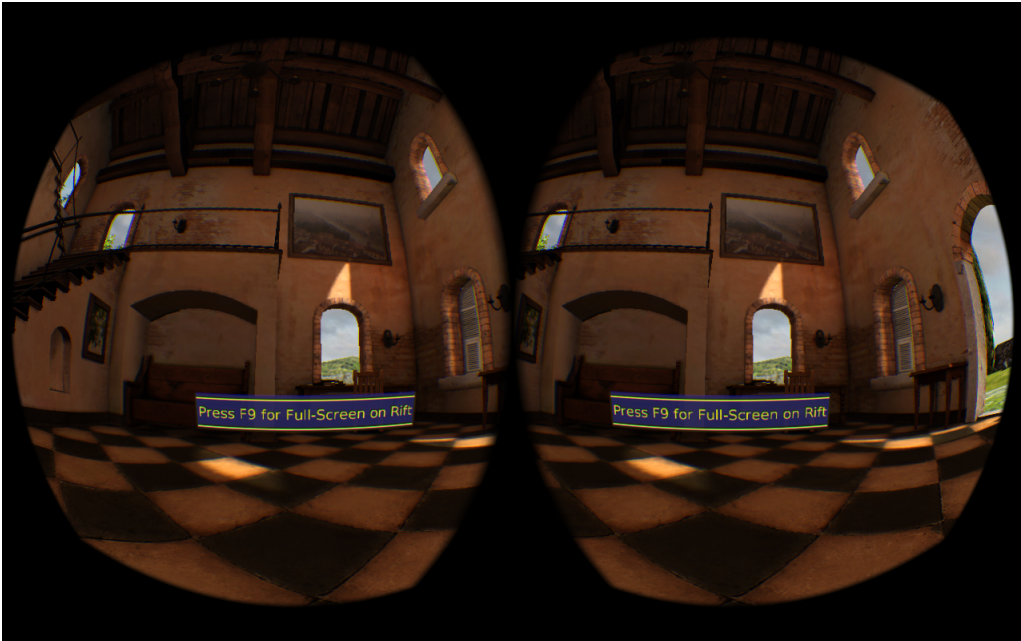
Aside from that, the next step is to start building your own Oculus-ready game or application. Thousands of other developers are out building the future of virtual reality gaming. To see what they are talking about, go to forums.oculus.com.

OculusWorldDemo Demo

Oculus recommends running the pre-built OculusWorldDemo to explore the SDK. You can find a link to the executable file in the root of the Oculus SDK installation.

The following is a screenshot of the OculusWorldDemo application:

Figure 2: OculusWorldDemo Application



OculusWorldDemo Controls

The OculusWorldDemo uses a mix of standard and specialized controls.

The following table describes keys and devices that you use for movement:

Table 1: Movement

Key or Input	Movement
W, S	Move forward, back
A, D	Strafe left, right
Mouse	Look left, right
Left gamepad stick	Move
Right gamepad stick	Turn

The following table describes keys that you use for functions:

Table 2: Functions

Key(s)	Function
F4	Multisampling toggle
F5	sRGB toggle
F7	Mono/stereo view mode toggle
F9	Hardware full-screen (low latency)
F11	Performance HUD toggle
E	Motion relative to head/body
R	Reset sensor orientation
Esc	Cancel full-screen
-, +	Adjust eye height
L	Adjust fourth view value
Tab	Options Menu
Spacebar	Toggle debug info overlay
T	Reset player position
Ctrl+Q	Quit
G	Cycle grid overlay mode
U, J	Adjust second view value
I, K	Adjust third view value
;	Cycle rendered scenes
+Shift	Adjust values quickly
O	Toggle Time-Warp
C	Toggle FreezeEyeUpdate
V	Toggle Vsync

OculusWorldDemo Usage

Once you've launched OculusWorldDemo, you should see a window on your PC monitor similar to the previous screenshot.

When the image is correctly displayed inside the Rift then take a moment to look around in VR and double-check that all of the hardware is working properly. You should be able to see that physical head translation is now also recreated in the virtual world as well as rotation.

Important: If you need to move the sensor for any reason after initial calibration, be sure to minimize the movement of the HMD for a few seconds while holding it within the tracking frustum. This will give the system chance to recalibrate the sensor pose.

If you would like to explore positional tracking in more detail, you can press the semicolon (;) key to bring the "sea of cubes" field that we use for debugging. In this mode, cubes are displayed that allow you to easily observe positional tracking behavior. Cubes are displayed in green when head position is being tracked and in red when sensor fusion falls back onto the head model.

There are a number of interesting things to take note of the first time you experience OculusWorldDemo. First, the level is designed to scale. Thus, everything appears to be roughly the same height as it would be in the

real world. The sizes for everything, including the chairs, tables, doors, and ceiling, are based on measurements from real world objects. All of the units are measured in meters.

Depending on your actual height, you may feel shorter or taller than normal. The default eye height of the player in OculusWorldDemo is 1.61 meters (approximately the average adult eye height), but this can be adjusted using the using the '+' and '-' keys.