

# Oculus Go

---



**Software environment:** Unity 2018.3.8/Android

**Hardware environment:** Oculus Go

## Practice objectives:

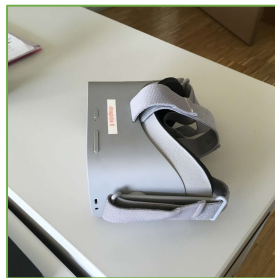
1. Become familiar with a low-cost, standalone VR Headset
2. Setup an Android VR application
3. Make UI changes to alter the User Experience
4. Package and preview in Oculus Go

## Description:

The Oculus Go is a low-cost (250-300 CHF) headset that runs on Android, like the Samsung Gear VR. This means that no external computer is required, which opens up many opportunities, but brings with it many limitations, as well, particularly in the controller (of which there is only one). You will setup your computer and Unity Environment in order to allow development with Oculus Go, modify an existing sample scene, and then deploy that to an Oculus Go headset.

## Warning:

**The Oculus Go (like many headsets) is easily damaged by sunlight landing on the Fresnel lenses. NEVER leave it sitting flat, with the lenses upright. A stray ray of sunshine will burn out pixels quickly!**



**YES! (AWAY FROM WINDOW)**



**NO!**

## Getting Started:

Setup your environment for Android development with Oculus Go. Follow the instructions in this tutorial: <https://medium.com/virtual-reality-virtual-people/setting-up-unity-for-oculus-go-development-43d3e8c12f59>

### You will need:

1. Android build support in Unity,
2. the Android SDK, included in Android Studio,
3. the ADB, Android debugging bridge, also included in Android Studio.

## Play a project from Unity directly in Oculus Go:

- Create a new project
  - If you have already installed Android build Support, you can start from the Unity Hub, selecting Android as your build target
    - Otherwise, go to File>Build Settings and choose Android
    - Click Switch Platform
  - From the Build Settings dialog, select Player Settings
    - Go to the Android tab (small robot)
    - XR Settings, make sure Oculus SDK is selected, and VR supported is clicked
  - Close the Build Settings
- From the asset store, download and import Oculus Integration (publisher Oculus)
- From the asset store, download and import VR Samples (publisher Unity)
- Go to VRSampleScenes, under your project assets, look in the Scenes folder
  - Find the “Flyer” scene and open it
  - Try playing it in Unity and see what happens (i.e. not much)
  - Stop playing the scene
- Go to File>Build Settings
  - Uncheck all the scenes listed in the box at the top
  - Click add open scenes (if it doesn't add the Flyer scene, or gives an error message, just ignore it)
- Turn on your Oculus Go and connect it with a micro USB cable
  - You may need to wait for Windows to configure your device
  - Look in the headset and wait for a message asking if you want to connect to this computer. Check the box to remember this connection and click OK.
- In Unity, make sure the target is Android, select your Oculus Go from the *Run Device* drop-down menu
- Click build and run
- Put on the Oculus Go and wait.... Eventually the game will load, but you may need to “Enter VR” by pressing and holding the Oculus button on the controller (if it has timed out)
- Play the game on the headset, paying attention to the control options.
- In Unity, press play to stop the game streaming in the Oculus Go.

- Modify the game
  - Change the generation of the asteroids.
  - Change the look or behaviour of the game.
- Go back and build and run again, observe the differences.
- Try one or two other sample scenes to familiarize yourself with the Oculus Go controls.

## Package the APK and deploy to Oculus Go:

- From the same project, go to Build Settings
- Click Build, create a new directory, called Builds, and give the game a name, like *Flyer*
- When the build has finished, navigate to the directory and copy the path
- Open the CMD prompt (or terminal, on Mac)
- Navigate to the Android SDK to find adb.exe (usually under something like C:\Users\USERNAME\AppData\Local\Android\Sdk\platform-tools)
- Type adb.exe devices
  - You should see the Oculus Go listed, with a serial number
- Type “adb.exe install path/Flyer.apk”, where your path is the path you just copied and the name is the name of the APK you just built (like *Flyer*)
- Once installed on the Go, you should see “Success” in the CMD prompt or terminal
- Put on the Oculus Go, and navigate to your library on the home screen. Click on unknown sources, and find your application. Note the package name identifier in addition to the app name. You may need this to uninstall or replace the app.
- Launch the app and see if there are performance differences relative to playing in emulator mode as you did previously.
- Also note that each subsequent build of the APK takes much less time than the original build.

## Screen Mirror your Oculus Go (useful for sharing and documentation):

- Connect your Oculus Go with the USB Cable (this also works over wifi, but our Oculus are not connected to EPFL wifi)
- Make sure you have VLC installed (<https://www.videolan.org/vlc/index.html>)
- Find the default location for Mac or PC
  - "C:\Program Files\VideoLAN\VLC\vlc.exe",
  - or for Mac, the default location is: "/Applications/VLC.app/Contents/MacOS/VLC"
- Go back to the ADB directory from above
- Type
  - PC: adb exec-out "while true; do screenrecord --bit-rate=16m --output-format=h264 --time-limit 180 -; done" | "C:\Program Files\VideoLAN\VLC\vlc.exe" --demux h264 --h264-fps=60 --clock-jitter=0 -
  - Mac: ./adb exec-out "while true; do screenrecord --bit-rate=8m --output-format=h264 --time-limit 180 -; done" |

```
"/Applications/VLC.app/Contents/MacOS/VLC" --demux h264 --  
h264-fps=60 --clock-jitter=0 --network-caching=100 --sout-mux-  
caching=100 -
```

## Experiment with a novel VR navigation technique:

- Download the GullivR APK here: <https://drive.google.com/open?id=19S5ZYdhO-aXBrPDf4QHIsbYBFzDVHz4p>
  - Credit to Andrey Krekhov, from Duisburg-Essen, and Hugo Decroix from EPFL for their work in developing this package
- Load the APK onto the Oculus Go with the ADB command link tool, as you have done above
- Start the Application from Unknown Sources in your library
- Experiment with the “Giant Mode” navigation technique, there are several different permutations available within the menu, experiment with different settings

**If you have any questions during the session, do not hesitate to ask the TAs!**

**Good Luck!**