**NOLO VR Android SDK**

**Documentation**

NOLO Co., Ltd

March 2018

目录

[1. Overview 3](#_Toc3906)

[1.1 About NOLO 3](#_Toc4665)

[1.2 About NOLO CV1 3](#_Toc7538)

[2. Set Up Development Environment 4](#_Toc5472)

[2.1 Import library 4](#_Toc7928)

[2.2 Acquire Authorization 4](#_Toc13177)

[3. API Description 5](#_Toc4556)

[3.1 Basic Information 5](#_Toc21566)

[3.2 Positional Information 7](#_Toc8666)

[4. Notes 12](#_Toc12957)

[4.1 Set Origin 12](#_Toc10818)

[4.2 Set AppKey 12](#_Toc16758)

[4.3 Obtain A NOLOVR Instance 13](#_Toc19867)

# Overview

## About NOLO

NOLO is dedicated to combine desktop-grade VR gaming experience with the convenience of mobile VR devices, redefining a mobile VR gaming experience like never before.

NOLO kit is compatible with some 87,000,000 VR headsets of all kinds currently on the market, indicating huge market potential. In addition, we’ve partnered with VR headset companies, robotic companies, and drone companies around the globe.

## About NOLO CV1

NOLO CV1 is a short-range high-accuracy motion tracking equipment kit for VR/AR gaming systems, composed of a Base Station, a headset marker and 2 controllers.

NOLO CV1 is compatible with all mainstream mobile VR headsets, some PC VR/AR headsets, and All-In-One’s based on NibiruOS, providing position information and interaction functionality to these devices. With simple setup at first launch, users can move around in the virtual world like in real world, and interact with virtual objects using our hand controllers.

# Set Up Development Environment

## Import library

Import androidsdkclient-release.aar to your Android project.

## Acquire Authorization

Add the following scripts in AndroidManifest.xml:

<uses-permission android:name="android.permission.BROADCAST\_STICKY" />

<uses-permission android:name="android.permission.PACKAGE\_USAGE\_STATS" />

<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />

<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission android:name="com.android.launcher.permission.WRITE\_SETTINGS" />

<uses-permission android:name="android.permission.WRITE\_APN\_SETTINGS" />

# API Description

## Basic Information

|  |  |
| --- | --- |
| function name | public static NoloVR getInstance(Context context)； |
| function description | Generate a NoloVR instance |
| input parameters | context: must be Activity instances, otherwise prompt windows won’t display properly. |
| return value | NoloVR instance |
| prerequisites |  |

|  |  |
| --- | --- |
| function name | public boolean isStallServer()； |
| function description | To check if NOLO HOME (server) has been installed on the phone. Notice that NOLO HOME handles all incoming data from NOLO hardware |
| input parameters |  |
| return value | bool |
| prerequisites | NoloVR getInstance(Context context) |

|  |  |
| --- | --- |
| function name | public void setAppKey(String appKey)； |
| function description | Set appkey |
| input parameters | String type appKey, obtained from NOLO Developer Center. |
| return value |  |
| prerequisites | NoloVR getInstance(Context context) |

|  |  |
| --- | --- |
| function name | public void openServer()； |
| function description | Connect to NOLO HOME(server), if successful, SDK can read data from server. |
| input parameters |  |
| return value |  |
| prerequisites | NoloVR getInstance(Context context) |

|  |  |
| --- | --- |
| function name | public void closeServer()； |
| function description | Disconnect from server, upon exiting the game. |
| input parameters |  |
| return value | bool |
| prerequisites | NoloVR getInstance(Context context) |

## Positional Information

|  |  |
| --- | --- |
| function name | NTrackedDevicePose getPoseByDeviceType(int type); |
| function description | To get positional information of a designated device. |
| input parameters | 0: headset marker  1: left controller  2: right controller  3: base station |
| return value | NTrackedDevicePose |
| prerequisites | NoloVR getInstance(Context context) |

|  |  |
| --- | --- |
| function name | int getElectricityByDeviceType(int type) ; |
| function description | To get battery information of a designated device. |
| input parameters | 0: headset marker  1: left controller  2: right controller  3: base station |
| return value | int |
| prerequisites | NoloVR getInstance(Context context) |

|  |  |
| --- | --- |
| function name | NControllerState getControllerStatesByDeviceType(int type); |
| function description | To get button status of a designated device. |
| input parameters | 0: headset marker  1: left controller  2: right controller  3: base station |
| return value | NControllerState |
| prerequisites | NoloVR getInstance(Context context) |

|  |  |
| --- | --- |
| function name | void SetHmdTrackingCenter(float x,float y,float z); |
| function description | To set headset’s rotational center |
| input parameters | According to the fixed position of the helmet positioner, the human eye is used as the coordinate origin, and the coordinate information corresponding to the helmet positioner is used to calculate the positioning information of the head and calculate the position of the adult eye. |
| return value | Void |
| prerequisites | NoloVR getInstance(Context context) |

|  |  |
| --- | --- |
| function name | void triggerHapticPulse(int deviceType,int intensity); |
| function description | To trigger controller vibration. |
| input parameters | Deice type:  0: headset marker  1: left controller  2: right controller  3: base station  Vibration intensity:  0~100 (int) |
| return value |  |
| prerequisites | NoloVR getInstance(Context context) |

|  |  |
| --- | --- |
| function name | void setHmdType(int hmdType); |
| function description | To set headset type (legacy function) |
| input parameters | (legacy function, not to be used anymore) |
| return value | (legacy function, not to be used anymore) |
| prerequisites | (legacy function, not to be used anymore) |

|  |  |
| --- | --- |
| function name | void reportError(String errorLog); |
| function description | Log error messages. |
| input parameters | String errorLog: error log collected by game. |
| return value |  |
| prerequisites | NoloVR getInstance(Context context) |

|  |  |
| --- | --- |
| function name | int add(int i,int j) |
| function description | Check connection status with NOLO HOME(server) |
| input parameters | Arbitrary integers i and j chosen by you |
| return value | An integer, equals i + j if connection with NOLO HOME(server) is successful, -1 otherwise. |
| prerequisites | NoloVR getInstance(Context context) |

|  |  |
| --- | --- |
| function name | int getElectricityByDeviceType(int type); |
| function description | Get battery level information based on device type |
| input parameters | 0：headsetmarker  1：leftcontroller  2：rightcontroller  3：basestation |
| return value | int,range（0~5） |
| prerequisites | NoloVR getInstance(Context context) |

|  |  |
| --- | --- |
| function name | int setPredictionTime(int predictionTime); |
| function description | Set the underlying inertia prediction time, the default bit is 25 milliseconds. |
| input parameters | predictionTime :The unit is milliseconds |
| return value | 1 |
| prerequisites | NoloVR getInstance(Context context) |

|  |  |
| --- | --- |
| function name | int getElectricityValueByDeviceType(int type); |
| function description | Get accurate battery information based on device type, the battery value will fluctuate up and down |
| input parameters | 0：headsetmarker  1：leftcontroller  2：rightcontroller  3：basestation |
| return value | int,range（0~100） |
| prerequisites | NoloVR getInstance(Context context) |

|  |  |
| --- | --- |
| function name | int getNoloHardwareVersionByDeviceType(int type); |
| function description | Obtain the hardware version information of the NOLO device |
| input parameters | 0：headsetmarker  1：leftcontroller  2：rightcontroller  3：basestation |
| return value | int |
| prerequisites | NoloVR getInstance(Context context) |

|  |  |
| --- | --- |
| function name | float getNoloSoftwareVersionByDeviceType(int type); |
| function description | Obtain the software version information of the NOLO device |
| input parameters | 0：headsetmarker  1：leftcontroller  2：rightcontroller  3：basestation |
| return value | float |
| prerequisites | NoloVR getInstance(Context context) |

# Notes

## Set Origin

Turn on all NOLO devices, place the headset marker on the ground, press the button on the headset marker. The headset marker’s current position will be the origin in the game. The origin’s coordinates will be saved. This process only needs to be repeated if the Base Station has been moved.

## Set AppKey

A game must acquire an AppKey to run properly with NOLO CV1. An AppKey will be generated automatically when developers apply for their game on NOLO Developer Center.

When the game does not upload NOLO HOME, you can use this public Appkey for development testing.

Public Appkey：4e4f4c4f484f4d457eff82725bc694a5

## Obtain A NOLOVR Instance

Before you try to connect with NOLO HOME(server), please call isStallServer() to check if NOLO HOME(server) has been installed on the phone. When the game starts, call openServer() to connect to the server, and closeServer() to disconnect from the server upon exiting the game.