**NOLO VR UE4 SDK**

**インターフェースについて**

NOLO Co., Ltd

2018年3月

目次

[1. 紹介 3](#_Toc510201059)

[1.1 NOLOについて 3](#_Toc510201060)

[1.2 NOLO CV1について 3](#_Toc510201061)

[1.3 NOLO HOMEについて 3](#_Toc510201062)

[1.4 NOLO VR Unity SDKについて 4](#_Toc510201063)

[1.5 モバイル端末のSDK構造 4](#_Toc510201064)

[2. 接続準備 6](#_Toc510201065)

[3. インターフェースについて 6](#_Toc510201066)

[3.1 ブループリント・インターフェースについて 6](#_Toc510201070)

[3.2 インターフェース詳細説明 8](#_Toc510201071)

[3.3 コントローラー･ボタン説明 11](#_Toc510201072)

[4. 注意事項 15](#_Toc510201073)

[4.1 原点位置を設定する。 15](#_Toc510201075)

[4.2 設置 AppKey 15](#_Toc510201076)

[4.3 方向設定 16](#_Toc510201077)

[4.4 UE4カメラ設定 16](#_Toc510201078)

[4.5 第三者 VR SDK 17](#_Toc510201079)

# 紹介

## NOLOについて

NOLOはVRボードゲームの素晴らしい体験及びVR移動の利便性を結びつけ、次世代のVR移動システムのインタラクティブな操作方法を改めて定義する。

現在、NOLOは世界で計8,700万個のVRヘルメットに対応でき、非常に市場の将来性がある。それから、NOLOは国内外で数多くの一体型VRヘルメット、PC VR、AR、サービスロボット及びドローンの関連会社と連携している。

## NOLO CV1について

NOLO CV1はNOLO Inc.自社で開発された世界初の完全没入型VR/AR 製品で、BS1つ、ヘルメットロケーター1つ及びコントローラー2つを含む。

NOLO CV1は現在市販されるVRヘルメット、一部のPC VR、ARのヘルメット及びNibiruOSの一体型に対応でき、空間定位及びインタラクティブの機能を支える。ユーザーは初めて使用する場合、簡単に設置すれば、仮想の現実世界を体験し、コントローラーで仮想世界との多様なインタラクションを体験することができる。

## NOLO HOMEについて

NOLO HOME は NOLO傘下のモバイルVRプラットフォームであり、世界初の携帯電話用6-DoFモバイルVRプラットフォームである。ユーザーはNOLO HOME 、NOLO CV1 及び携帯電話を利用して、時間や場所を問わず、無線で完全な没入型VR体験を享受することができる。一般消費者のために斬新なゲームプレー方法を提供する。現在、Androidのユーザーはダウンロードできる。iOS 版もまもなく使用できる。

## NOLO VR Unity SDKについて

NOLO VR Unity SDKはUnityの開発者がNOLO設備ﾃﾞｰﾀを取得し、Cardboard及びGearVR のSDKを搭載し、モバイル6DoF VRゲームを開発するために、NOLO Inc.により開発されたものである。

## モバイル端末のSDK構造

NOLO SDKはC/S構造を採用する。NOLO HOMEはServeであり、NOLO CV1設備ﾃﾞｰﾀを読込み、処理する。ゲームはClientである。SDKはAIDLによってゲームとNOLOHOMEの間にﾃﾞｰﾀを転送させ、NOLO CV1設備ﾃﾞｰﾀを取得する。ゲームはUSB設備ﾃﾞｰﾀを読み込む必要がなく、NOLO HOMEとの接続のみによってﾃﾞｰﾀの送受信をする。(図 1)

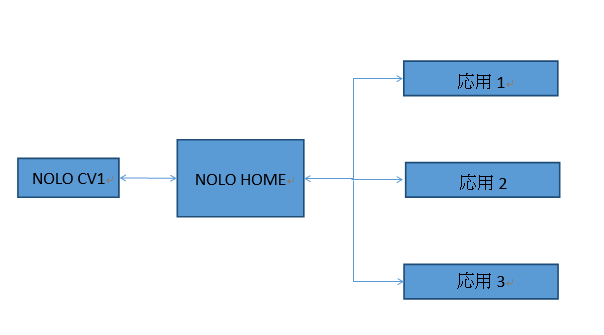


図 1

NOLO VR Android SDKはAIDL及びNOLO HOMEでプロセス間通信を実現することによって、設備ﾃﾞｰﾀを取得する。getInstance()でNOLOVRの実例を取得する。ゲーム開始前にisStallServer()でユーザーがNOLO HOMEをインストールしたかどうかを判断する。インストールを確認された場合、openServer()でServerをバインディングする。ゲーム終了後、closeServer()でバインディングを外す。NOLO VR Unity SDK及びNOLO VR UE4 SDKはNOLO VR Android SDKに対して、部分のインタフェースに対してカプセル化する。開発者はインターフェースを通じて、位置及びボタン値を取得すれば、ゲームを開発することができる。 (図2).

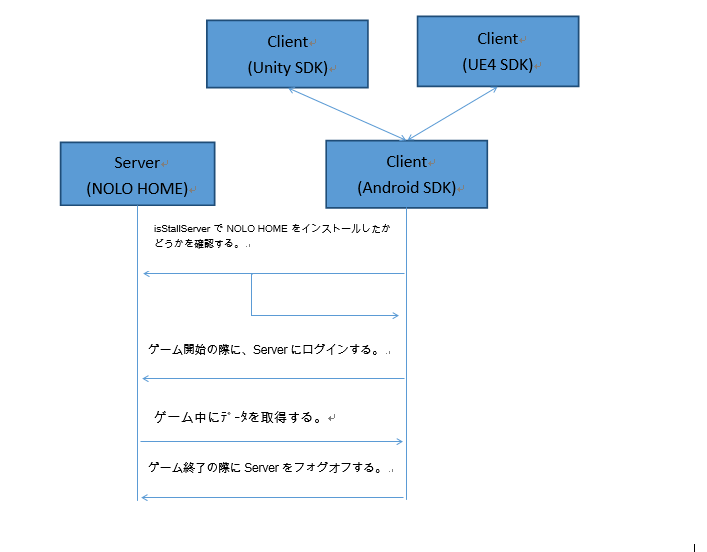


図 2

# 接続準備

開発及びテストの準備：UnrealEngine4.18.2/4.18.3

携帯電話で利用する場合、NOLO HOMEまずインストールする必要がある。

# インターフェースについて



## ブループリント・インターフェースについて

(図 3)

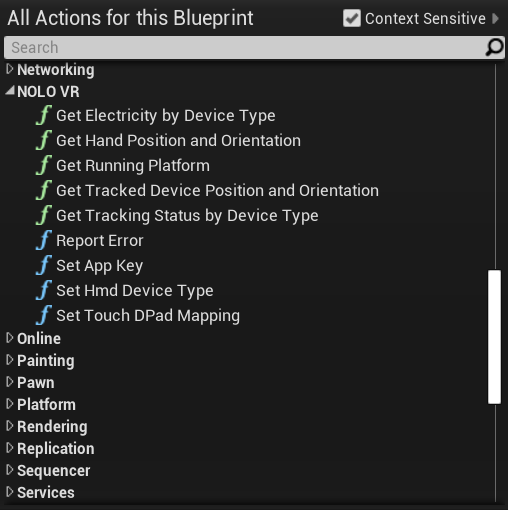


図3

1. GetRunningPlatform：ゲームプラットフォームを取得する。
2. GetTrackedDevicePositionAndOrientation：ヘルメットのロケーターの位置及び状態に関する情報を取得する。
3. GetHandPositionAndOrientation：コントローラーの位置及び状態に関する情報を取得する。
4. GetTrackingStatusByDeviceType：設備の追跡状況を取得する。
5. GetElectricityByDeviceType：設備の消費電力情報を取得する。
6. SetHmdDeviceType：ゲーム開発に使用されるヘルメット種類を設定する。
7. SetAppKey：設置AppKey
8. ReportError：異常情報を取得する。
9. SetTouchDPadMapping: タッチパネルの上下左右の4つのボタンに関連するイベントを定義する。

## インターフェース詳細説明

|  |  |
| --- | --- |
| **関数名** | **FString GetRunningPlatform（）** |
| **機能説明** | ゲームプラットフォームを取得する。 |
| **パラメーター** | **NULL** |
| **戻り値** | **FString : Android or Default** |
| **先決条件** |  |

|  |  |
| --- | --- |
| **関数名** | **bool GetTrackedDevicePositionAndOrientation（FVector & OutPosition, FRotator & OutOrientation）** |
| **機能説明** | 頭部ロケーターの位置及び状態に関する情報を取得する。 |
| **パラメーター** | **OutPosition : (out) Current position of the device**  **OutOrientation (out) Current orientation of the device** |
| **戻り値** | **Bool : True if the specified device id had a valid tracking pose this frame, false otherwise** |
| **先決条件** |  |

|  |  |
| --- | --- |
| **関数名** | **Bool GetHandPositionAndOrientation（EControllerHand Hand, FVector& OutPosition, FRotator& OutOrientation）** |
| **機能説明** | コントローラーの位置及び状態に関する情報を取得する。 |
| **パラメーター** | **Hand : Which hand's controller to get the position and orientation for**  **OutPosition : (out) Current position of the device**  **OutOrientation (out) Current orientation of the device** |
| **戻り値** | **Bool : True if the specified controller is Available** |
| **先決条件** |  |

|  |  |
| --- | --- |
| **関数名** | **bool GetTrackingStatusByDeviceType(const ENoloDeviceType DeviceType)** |
| **機能説明** | 設備の追跡状況を取得する。 |
| **パラメーター** | **ENoloDeviceType : Hmd,Left,Right,Basestation** |
| **戻り値** | **True : Tracked ,false : not Tracked** |
| **先決条件** |  |

|  |  |
| --- | --- |
| **関数名** | **Bool bool GetElectricityByDeviceType(const ENoloDeviceType DeviceType,int & battery)** |
| **機能説明** | 設備仕様によって設備の消費電力情報を取得する。 |
| **パラメーター** | **ENoloDeviceType : Hmd,Left,Right,Basestation**  **Battery : (out)0:shut down 、1:low 、2:middle 、3:high** |
| **戻り値** | **Bool : True if the battery info is Available** |
| **先決条件** |  |

|  |  |
| --- | --- |
| **関数名** | **void SetHmdDeviceType(const EMobileVR\_HMDDeviceType DeviceType)** |
| **機能説明** | ゲーム開発に使用されるヘルメットの種類を設定する。 |
| **パラメーター** | **EMobileVR\_HMDDeviceType : DT\_GearVR、 DT\_GoogleVR\_Cardboard、DT\_GoogleVR\_Daydream、DT\_Other** |
| **戻り値** | **Void** |
| **先決条件** |  |

|  |  |
| --- | --- |
| **関数名** | **void SetAppKey (FString appKey)** |
| **機能説明** | 位置指定数値を取得するappkey権限を設定する。 |
| **パラメーター** | **FString** |
| **戻り値** | **Void** |
| **先決条件** |  |

|  |  |
| --- | --- |
| **関数名** | **void ReportError (FString appKey)** |
| **機能説明** | LOGｹﾞｰﾑの中のエラー情報を収集する。 |
| **パラメーター** | **FString** |
| **戻り値** | **Void** |
| **先決条件** |  |

|  |  |
| --- | --- |
| **関数名** | **void SetTouchDPadMapping (ENoloVRTouchDPadMapping NewMapping)** |
| **機能説明** | **タッチパネルの上下左右の4つのボタンに関連するイベントを定義する。** |
| **パラメーター** | **ENoloVRTouchDPadMapping** |
| **戻り値** | **Void** |
| **先決条件** |  |

## コントローラー･ボタン説明

(図 4) (図 5) (図 6).

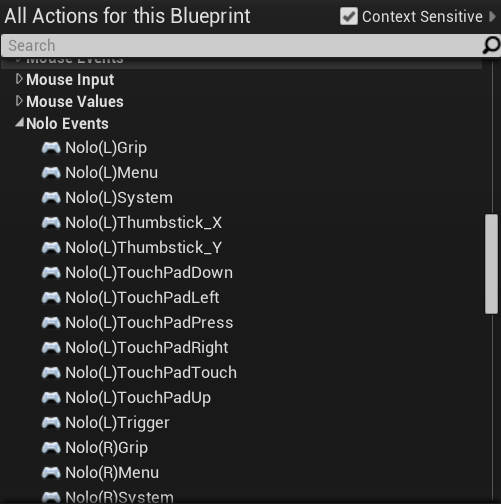


図 4

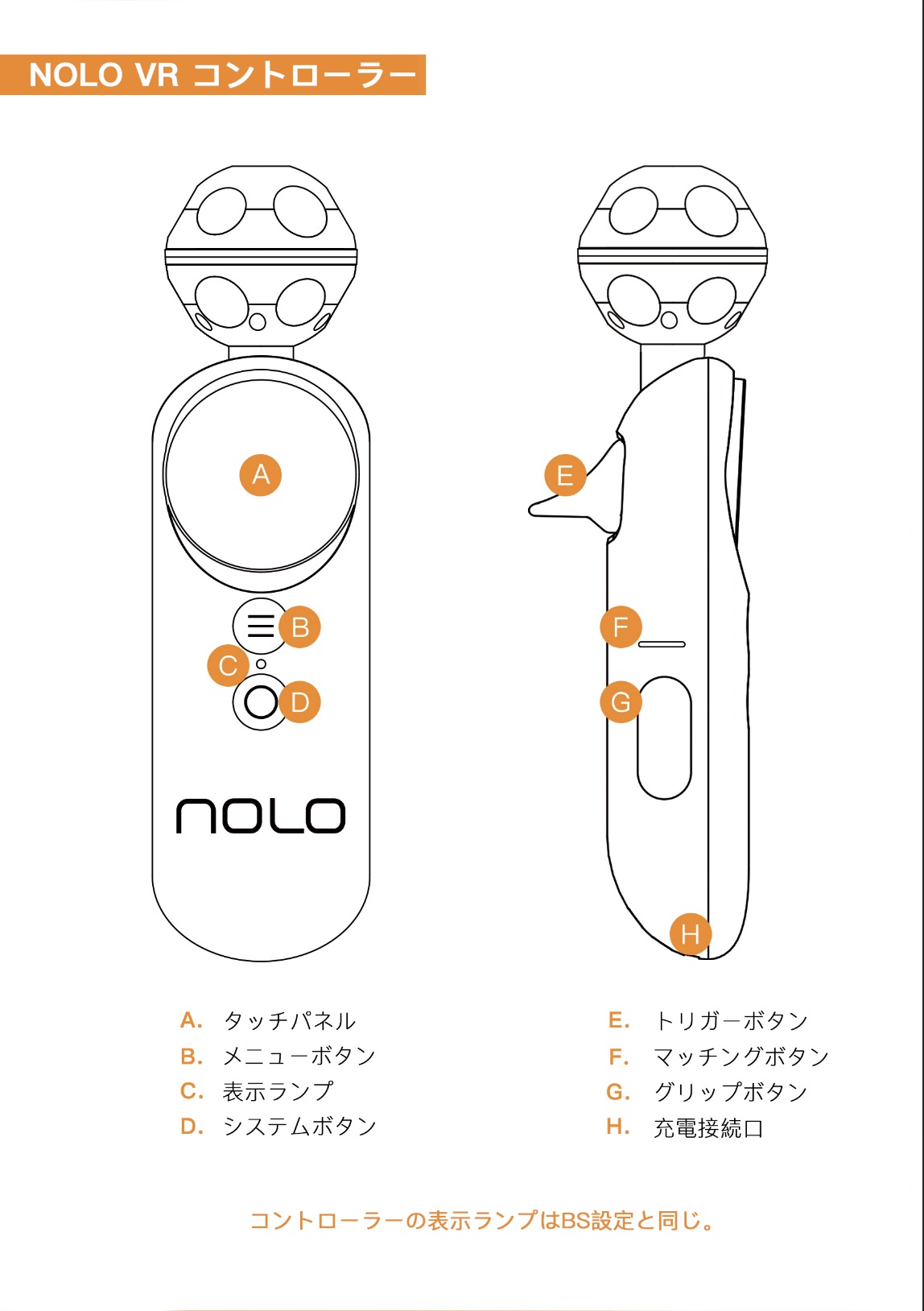


図 5

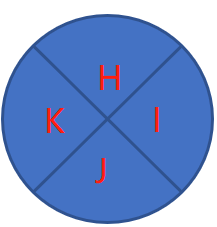
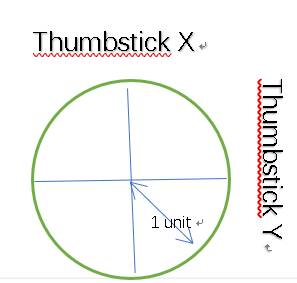


図 6

ボタン一覧表は以下の通り。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ボタン凡例** | **描述**  **説明** | **NOLO Events** | **Gamepad Events** | **SteamEvents** |
| A | TouchPadPress | Nolo\_Left\_TouchPadPress | MotionController\_Left\_Thumbstick |  |
| A | TouchPadTouch | Nolo\_Left\_TouchPadTouch |  | Steam\_Touch\_0 |
| B | Menu | Nolo\_Left\_Menu | MotionController\_Left\_Shoulder |  |
| D | System | Nolo\_Left\_System | SpecialLeft |  |
| E: | Trigger | Nolo\_Left\_Trigger | MotionController\_Left\_Trigger |  |
| G | Grip | Nolo\_Left\_Grip | MotionController\_Left\_Grip1 |  |

タッチパネルのボタン値一覧表は以下の通り。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ボタン凡例** | **描述**  **説明** | **NOLO Events** | **Face Buttos**  **(Gamepad Events)** | **ThumbstickDirections**  **(Gamepad Events)** |
| H | Up | Nolo\_Left\_TouchPadUp | MotionController\_Left\_FaceButton1 | MotionController\_Left\_Thumbstick\_Up |
| I | Right | Nolo\_Left\_TouchPadRight | MotionController\_Left\_FaceButton2 | MotionController\_Left\_Thumbstick\_Right |
| J | Down | Nolo\_Left\_TouchPadDown | MotionController\_Left\_FaceButton3 | MotionController\_Left\_Thumbstick\_Down |
| K | Left | Nolo\_Left\_TouchPadLeft | MotionController\_Left\_FaceButton4 | MotionController\_Left\_Thumbstick\_Left |

# 注意事項



## 原点位置を設定する。

NOLO正常に稼動する場合、ヘルメットを床に置き、ロケーターのボタンを押せば、当位置がｹﾞｰﾑの初期位置となる。即ち、ゲームエンジン“”の位置。原点位置を設定すれば、設備に登録される。BSの位置が大きく変更した場合のみ、改めて原点位置を設定する必要がある。

## 設置 AppKey

セッティングappkeyの後、ｹﾞｰﾑプレイができる。appkeyそれは開発者がプラットフォームを利用する際に自動生成される。

ゲームがNOLO HOMEをアップロードしないときは、この公開Appkeyを開発テストに使用することができます。

公開Appkey：4e4f4c4f484f4d457eff82725bc694a5。

## 方向設定

ゲーム開始後、ゲームの方向がBSの方向と異なる場合、BSに向かってコントローラーの電源をダブルクリックし、方向を設定する。コントローラーの方向が異なる場合、BSに向かって当コントローラーの電源ｽｲｯﾁをダブルクリックし、方向を設定する。

## UE4カメラ設定

NOLO VRを提供した6DOFの位置決めデータが、NOLO VRプラグインを接続しないUE4 HMDインタフェース（IHeadMountedDisplay）。NOLO VR頭部ロケーターの位置情報のみ使用する場合、以下の図面で設計してもよい。即ち、カメラの親ノードの位置情報を変更する。NOLO VR頭部ロケーターの位置及び状態のﾃﾞｰﾀを使用する場合、にチェックを入れないでくださいLock to Hmd、あるいは他のVRプラグインを選択しない。携帯電話のカメラの状態（Rotation）を調整してください。 (図 7)

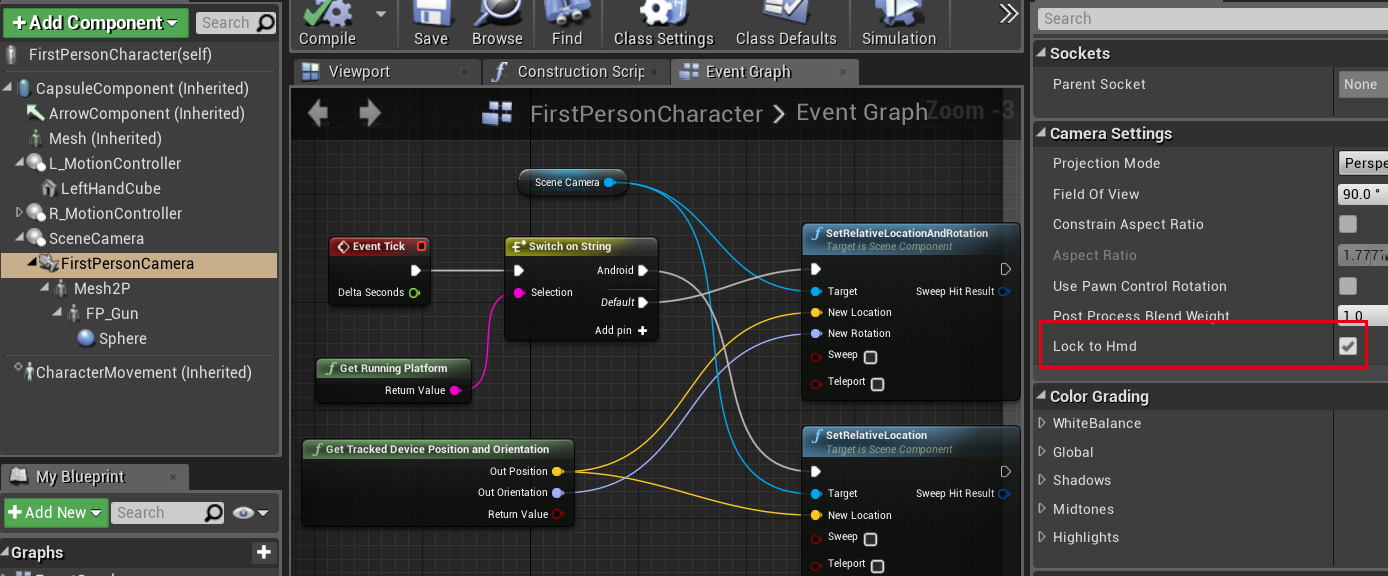


図 7

## 第三者 VR SDK

NOLO VRプラグインを完全に実現することUE4エンジンのハンドル制御のインターフェイス、頭の位置追跡このを提供した想いはインタフェース、位置決めデータを得るので、全く配合Google VR、Gear VRをゲーム開発、移動端は3DOFデータ問題解決。