

ヴルームSDK リファレンス

VROOM SDK FOR UNITY DOCUMENTATION

対応SDKバージョン : 2.33

公開日 : 2017年 1月26日

目次

1. 使用方法	3
2. VvrController	3
2.1. プロパティ	3
2.2. Static変数	3
2.3. 関数	4
2.3.1. デバイスのスキャン	4
2.3.2. デバイスの接続／切断	5
2.3.3. 接続デバイスの情報	8
2.3.4. モーションセンサー	9
2.3.5. ジョイスティック	11
2.3.6. Appボタン	13
2.3.7. Homeボタン	14
2.3.8. トリガー	15
2.3.9. ボリュームボタン	16
2.3.10. バッテリー情報 (Battery Service)	18
2.3.11. デバイス情報 (Device Information Service)	18
3. VvrPermissionRequester	20
3.1. Static変数	20
3.2. 関数	20

1. 使用方法

- VvrControllerのunitypackageをインポートします。
- VvrControllerプレファブをシーンにアタッチします。
- VvrControllerプレファブはAssets > Vroom > Prefabにあります。

2. VvrController

- コントローラAPIクラス。
- このAPIを使用するには、VvrControllerプレハブをシーンにアタッチします。
- このオブジェクトはシーンに1つのみ追加可能で、DontDestroyOnLoadを設定したオブジェクトとなります。

2.1. プロパティ

- 接続スタイル

VvrConnectionStyle ConnectionStyle = VvrConnectionStyle.RightHandOnly;

RightHandOnly : 片手モード(1本接続)

RightHandAndLeftHand : 両手モード(2本接続)

- 接続モード

VvrConnectionMode ConnectionMode

Auto : 自動接続モード

Manual : 手動接続モード

2.2. Static変数

- 最大接続可能コントローラ数

int maxConnectableDevices

- 接続スタイル

public static int connectionStyle

スタイル:

1 : 片手(デフォルト)

2 : 両手

備考:

プロパティのConnection Styleの設定値になります。

- 接続モード

```
public static VvrConnectionMode connectionMode
```

モード:

VvrConnectionMode.Auto : 自動接続モード(デフォルト)

VvrConnectionMode.Manual : 手動接続モード

備考:

プロパティのConnection Modeの設定値になります。

- スキャン状態

```
public static bool scanningState
```

状態:

true : スキャン動作中

false : スキャン停止中

2.3. 関数

2.3.1. デバイスのスキャン

```
public static void StartScan();
```

パラメーター:

なし

戻り値:

なし

説明:

- デバイスのスキャンを開始します。
- スキャンはStopScanをコールするか、開始から20秒後に自動停止します。
- また、スキャンが開始されると、scanningStateにtrueが設定されます。
- スキャン結果は、GetScanResultにより取得します。
- 自動接続モードの場合、最初に見つかったデバイスに自動的に接続します。

スキャン条件:

- アドバタイズパケットの値に以下の値が含まれていること。
- Shortened Device Name(Flag:0xXX)が“Vroom”であること。
- GATTにVroom Controller Measurement Service (UUID:)が含まれていること。

```
public static void StopScan ();
```

パラメーター:

なし

戻り値:

なし

説明:

- デバイスのスキャンを停止します。
- スキャンが停止すると、scanningStateにfalseが設定されます。

```
public static void GetScanResult (out int size, out string [][]  
peripherals);
```

パラメーター:

size : 検出デバイス数
peripherals : 検出デバイス一覧

戻り値:

なし

説明:

- スキャンによって検出したデバイスの一覧を取得します。
- 検出デバイス一覧のデータ構造は以下の通りです。

```
peripherals[m][n]  
  m      : デバイス番号(0 ~ (size -1))  
  n      : デバイス情報  
           0      : デバイスID  
           1      : デバイス名  
           2      : RSSI値
```

2.3.2. デバイスの接続／切断

```
public static bool ConnectRequest(int hand, string peripheral);
```

パラメーター:

hand : コントローラ番号(0 or 1)、省略時 0
peripheral : デバイスID、省略時 string.Empty

戻り値:

デバイス接続リクエストが正常に行われた場合にtrueを返します。

説明:

- 指定されたコントローラに接続します。

- デバイスが接続されたかどうかは、ConnectionState()で確認してください。
- なお、指定されたコントローラの接続状態と、デバイスIDの指定、および接続モードの値によって挙動が変わります。

接続モード: 自動(connectionMode == VvrConnectionMode.Auto)

接続状態	動作
Connected	接続中のデバイスと同じ場合、何もせず、trueを返します。 接続中のデバイスと異なる場合、何もせず、falseを返します。
Connecting	接続処理中のデバイスと同じ場合、何もせず、trueを返します。 接続処理中のデバイスと異なる場合、何もせず、falseを返します。
Scanning	デバイスID省略時、スキャンを開始します。なお、最初に検出したデバイスに自動接続します。 デバイスID指定時、接続リクエストを出して、trueを返します。
Disconnecting	何もせず、falseを返します。
Disconnected	デバイスID省略時、スキャンを開始します。なお、最初に検出したデバイスに自動接続します。 デバイスID指定時、接続リクエストを出して、trueを返します。

接続モード: 手動(connectionMode == VvrConnectionMode.Manual)

接続状態	動作
Connected	接続中のデバイスと同じ場合、何もせず、trueを返します。 接続中のデバイスと異なる場合、何もせず、falseを返します。
Connecting	接続処理中のデバイスと同じ場合、何もせず、trueを返します。 接続処理中のデバイスと異なる場合、何もせず、falseを返します。
Scanning	デバイスID省略時、何もせず、falseを返します。 デバイスID指定時、接続リクエストを出して、trueを返します。
Disconnecting	何もせず、falseを返します。
Disconnected	デバイスID省略時、何もせず、falseを返します。 デバイスID指定時、接続リクエストを出して、trueを返します。

備考:

- 30秒以内に接続が完了しない場合、内部で切断リクエストを行い、デバイスを切断しようとしてします。

```
public static bool DisconnectRequest(int hand = RightHand);
```

パラメーター:

hand :コントローラ番号(0 or 1)、省略時 0

戻り値:

デバイス切断リクエストが正常に行われた場合にtrueを返します。
デバイスが切断されたかどうかは、ConnectionState()で確認してください。

説明:

- 指定されたコントローラを切断します。

```
public static VvrConnectionState ConnectionState(int hand);
```

パラメーター:

hand :コントローラ番号(0 or 1)、省略時 0

戻り値:

コントローラの接続状態

説明:

- 指定されたコントローラの接続状態を取得します。

状態	説明
Error	エラーにより接続／切断ができません。
Disconnected	コントローラ未接続
Disconnecting	コントローラ切断中
Scanning	指定されたコントローラ番号はスキャン中
Connecting	コントローラ接続中
Connected	コントローラ接続完了

```
public enum VvrConnectionState  
{  
    Error = -1,  
    Disconnected = 0,  
    Disconnecting,  
    Scanning,  
    Connecting,  
    Connected  
}
```

```
        Scanning,  
        Connecting,  
        Connected  
};
```

2.3.3. 接続デバイスの情報

```
public static string PeripheralID(int hand);
```

パラメーター:

hand :コントローラ番号(0 or 1)、省略時 0

戻り値:

デバイスID

説明:

- 指定されたコントローラのデバイスIDを取得します。
- AndroidとiOSではデバイスIDのフォーマットが異なります。

Android	ハードウェアアドレス(48bit) 全ての端末で一意的な値となります。
iOS	UUID(128bit) OSで生成される値であるため、端末ごとに異なる値となります。

```
public static string ShortenedDeviceName(int hand);
```

パラメーター:

hand :コントローラ番号(0 or 1)、省略時 0

戻り値:

デバイス名(ショート)

説明:

- 指定されたコントローラのShortened Device Nameを取得します。
- Shortened Device Nameの値は、アダプタイズパケットに含まれる値で常に“Vroom”となります。

```
public static string DeviceName(int hand);
```

パラメーター:

hand :コントローラ番号(0 or 1)、省略時 0

戻り値:

デバイス名

説明:

- 指定されたコントローラのデバイス名を取得します。
 - ここで取得する値は、GAP(Generic Access Profile)のDeviceNameに設定されている値で常に“Vroom Controller”となります。
 - コントローラからデバイス名を読み出せていない場合、Shortened Device Nameが返されます。
 - Shortened Device Nameについては、ShortenedDeviceName()関数を参照してください。
-

public static string RSSI(int hand);

パラメーター:

hand :コントローラ番号(0 or 1)、省略時 0

戻り値:

RSSI値

説明:

- 指定されたコントローラのRSSI値を取得します。
 - スキャンによって検出されたタイミングでのRSSI値となり、リアルタイムで変化はしません。
 - (iOS)他アプリに接続中のデバイスも検出されますが、該当デバイスのRSSI値は、string.Emptyとなります。
-

2.3.4. モーションセンサー

public static Quaternion Orientation (int hand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

クォータニオン

説明:

- 指定されたコントローラのクォータニオンを取得します。
 - コントローラの左右方向がX軸、上下方向がY軸、前後方向がZ軸となります。
-

public static Vector3 Gyro (int hand = RightHand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

ジャイロ

説明:

- 指定されたコントローラの角速度(ジャイロ)を取得します。
- コントローラの左右方向がX軸、前後方向がY軸、上下方向がZ軸となります。
- 角速度センサーの精度は 2000degree/sです。

public static Vector3 Accel (int hand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

加速度

説明:

- 指定されたコントローラの加速度(G)を取得します。
- コントローラの左右方向がX軸、前後方向がY軸、上下方向がZ軸となります。
- 加速度センサーの精度は ±16Gです。

public static bool Recentering (int hand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

モーションセンサーのリセット状態

説明:

- Homeボタンを1秒以上長押しすることでモーションセンサーがリセットされます。
- リセット状態になると、リセットが終了するまでtrueを返します。

public static bool Recentered (int hand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

モーションセンサーのリセット終了

説明:

- Homeボタンを1秒以上長押しすることでモーションセンサーがリセットされます。
- コントローラのリセット終了時にtrueを返します。

2.3.5. ジョイスティック

public static Vector2 Joystick (int hand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

ジョイスティックの状態

説明:

- ジョイスティックの状態を返します。
X軸 :コントローラの奥から手前(縦方向)
Y軸 :コントローラの左から右(横方向)
- コントローラの中点を(0.0F,0.0F)とし、左奥が(-1.0F,-1.0F)、右手前が(1.0F,1.0F)となります。

public static VvrJoystickAction JoystickAction (int hand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

アクション

説明:

- ジョイスティックの状態からアクションを返します。
- アクション毎のジョイスティックの数値(Joystick()で取得可能)の設定は下表の通りです。

アクション	ジョイスティック(X)	ジョイスティック(Y)
UpLeft	-1.0F ~ -0.7F	-1.0F ~ -0.7F
Up	-1.0F ~ -0.7F	-0.69F ~ 0.69F
UpRight	-1.0F ~ -0.7F	0.7F ~ 1.0F
Left	-0.69F ~ 0.69F	-1.0F ~ -0.7F
Neutral	-0.69F ~ 0.69F	-0.69F ~ 0.69F
Right	-0.69F ~ 0.69F	0.7F ~ 1.0F
DownLeft	0.7F ~ 1.0F	-1.0F ~ -0.7F
Down	0.7F ~ 1.0F	-0.69F ~ 0.69F

DownRight	0.7F ~ 1.0F	0.7F ~ 1.0F
-----------	-------------	-------------

備考:

```
public enum VvrJoystickAction {
    Neutral = 0,
    UpLeft,
    Up,
    UpRight,
    Left,
    Right,
    DownLeft,
    Down,
    DownRight,
}
```

public static bool ClickButton (int hand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

Clickボタンの状態

説明:

- 指定されたコントローラのジョイスティックのClickボタンの状態を取得します。
- Clickボタンを押している時にtrueを返します。

public static bool ClickButtonDown (int hand = RightHand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

Clickボタンのプレスタイミング

説明:

- 指定されたコントローラのジョイスティックのClickボタン押したタイミングでtrueを返します。

public static bool ClickButtonUp (int hand = RightHand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

Clickボタンのリリースタイミング

説明:

指定されたコントローラのClickボタンを離れたタイミングでtrueを返します。

2.3.6. Appボタン

public static bool AppButton (int hand = RightHand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

Appボタンの状態

説明:

- 指定されたコントローラのAppボタンの状態を取得します。
 - Appボタンを押している時にtrueを返します。
-

public static bool AppButtonDown (int hand = RightHand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

Appボタンのプレスタイミング

説明:

- 指定されたコントローラのAppボタンを押したタイミングでtrueを返します。
-

public static bool AppButtonUp (int hand = RightHand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

Appボタンのリリースタイミング

説明:

- 指定されたコントローラのAppボタンを離れたタイミングでtrueを返します。
-

2.3.7. Homeボタン

```
public static bool HomeButton (int hand = RightHand);
```

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

Homeボタンの状態

説明:

- 指定されたコントローラのHomeボタンの状態を返します。
- Homeボタンを押している時はtrueになります。

備考:

- HomeButtonState()関数と機能は同じです。
-

```
public static bool HomeButtonState (int hand = RightHand);
```

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

Homeボタンの状態

説明:

- 指定されたコントローラのHomeボタンの状態を返します。
- Homeボタンを押している時はtrueになります。

備考:

- HomeButton()関数と機能は同じです。
-

```
public static bool HomeButtonDown (int hand = RightHand);
```

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

Homeボタンのプレスタイミング

説明:

- 指定されたコントローラのHomeボタンを押したタイミングでtrueを返します。
-

```
public static bool HomeButtonUp (int hand = RightHand);
```

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

Homeボタンのリリースタイミング

説明:

- 指定されたコントローラのHomeボタンを離れたタイミングでtrueを返します。
 - モーションセンサーのリセットと挙動を分けたい場合、Recentring()関数でリセット状態とっていないか事前に確認してください。
-

2.3.8. トリガー

public static bool Trigger (int hand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

トリガーボタンの状態

説明:

- 指定されたコントローラのトリガーボタンの状態を返します。
 - トリガーボタンが押されている場合にtrueを返します。
-

public static bool TriggerDown (int hand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

トリガーボタンのプレスタイミング

説明:

- 指定されたコントローラのトリガーボタンを押したタイミングでtrueを返します。
-

public static bool TriggerUp (int hand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

トリガーボタンのリリースタイミング

説明:

- 指定されたコントローラのトリガーボタンを離れたタイミングでtrueを返します。
-

2.3.9. ボリュームボタン

public static VvrVolumeState Volume (int hand);

パラメーター:

hand :コントローラ番号(0 ~ n)

戻り値:

ボリュームアップ・ダウン

説明:

- ボリュームアップ・ダウンの状態を返します。
- ボリューム+、ボリューム-の2ボタンの押下状態から決定されます。
- 同時押しの場合、後に押したボタンが優先されます。
 1. 2ボタンとも押していない場合、Noneとなります。
 2. ボリューム+ボタンを押した状態(TurnUp状態)でボリューム-ボタンを押すとTurnDownとなります。
 3. 2の状態から、ボリューム-ボタンを離した場合、状態はNoneとなります。
 4. 3の状態から、ボリューム-ボタンを押すと、VurnDownとなります。
 5. 2の状態から、ボリューム+ボタンを離した場合、状態は変わりません。
 6. 5の状態から、ボリューム+ボタンを押すと、TurnDownとなります。

状態	説明
TurnDown	ボリュームアップ
None	未操作
TurnUp	ボリュームダウン

備考:

- 現状、SDK側で音量のコントロールはしておりません。本情報を元に、UnityのAudioMixer等を制御することで、コントロールをしてください。

```
public enum VvrVolumeState
{
    TurnDown,
    None,
    TurnUp
}
```

public static VvrVolumeState VolumeState (int hand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

ボリュームアップ・ダウン

説明:

- Volume()関数と同じです。Volume()関数を参照してください。
-

public static bool VolumeUp (int hand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

ボリューム+ボタンの状態

説明:

- ボリューム+ボタンの状態を取得します。
 - ボリューム+ボタンを押されている場合にtrueを返します。
-

public static bool VolumeDown (int hand);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

説明:

ボリューム-ボタンの状態

説明:

- ボリューム-ボタンの状態を取得します。
 - ボリューム-ボタンを押されている場合にtrueを返します。
-

2.3.10. バッテリー情報 (Battery Service)

public static int BatteryLevel (int hand = 0);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

バッテリー残量 (0 ~ 100, -1)

説明:

- 指定されたコントローラのバッテリー残量の値を取得します。(0 ~ 100%)
- 未取得または一度もコントローラが接続されていない場合は、-1が返却されます。

2.3.11. デバイス情報 (Device Information Service)

public static string ManufacturerName(int hand = 0);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

Manufacturer Name Stringの値

説明:

- 指定されたコントローラのManufacturer Name Stringの値を取得します。
- 未取得または一度もコントローラが接続されていない場合は、string.Emptyが返却されます。

public static string ModelNumber (int hand = 0);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

Model Number Stringの値

説明:

- 指定されたコントローラのModel Number Stringの値を取得します。
- 未取得または一度もコントローラが接続されていない場合は、string.Emptyが返却されます。

public static string SerialNumber (int hand = 0);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

Serial Number Stringの値

説明:

- 指定されたコントローラのSerial Number Stringの値を取得します。
- 未取得または一度もコントローラが接続されていない場合は、string.Emptyが返却されます。

public static string HardwareRevision (int hand = 0);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

Hardware Revision Stringの値

説明:

- 指定されたコントローラのHardware Revision Stringの値を取得します。
- 未取得または一度もコントローラが接続されていない場合は、string.Emptyが返却されま
す。

public static string FirmwareRevision (int hand = 0);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

Firmware Revision Stringの値

説明:

- 指定されたコントローラのFirmware Revision Stringの値を取得します。
- 未取得または一度もコントローラが接続されていない場合は、string.Emptyが返却されま
す。

public static string SoftwareRevision (int hand = 0);

パラメーター:

hand :コントローラ番号(0 ~ n)、省略時 0

戻り値:

Software Revision Stringの値

説明:

- 指定されたコントローラのSoftware Revision Stringの値を取得します。
- 未取得または一度もコントローラが接続されていない場合は、string.Emptyが返却されま
す。

3. VvrPermissionRequester

- Android Runtime Permission API。
- Android以外は不要。

3.1. Static変数

- VvrPermissionRequesterインスタンス
public static VvrPermissionRequester Instance

3.2. 関数

```
public bool IsPermissionGranted();
```

パラメーター:

なし

戻り値:

パーミッション権限

true :パーミッション権限あり

false :パーミッション権限なし

説明:

- パーミッション(Access.CoarseLocation)の権限有無を取得します。
- Android 6.0未満、またはAndroid以外で実行された場合、常にtrueを返します。

```
public bool ShouldShowRequestPermissionRationale ();
```

パラメーター:

なし

戻り値:

パーミッションの付与許可

true :パーミッションの付与拒否

false :パーミッションの付与許可、または、「今後表示しない」をチェックして付与拒否している場合

説明:

- パーミッション権限なしの場合に、ユーザへ告知するUIを表示する必要があるかを返します。
- このメソッドで、パーミッションの権限有無をチェックすべきではありません。
- パーミッションの権限有無をチェックする場合は、IsPermissionGranted()を使用してください。
- RequestPermission()で表示されるダイアログで、「今後表示しない」をチェックして拒否したかどうかを判定することにも使用可能です。

```
public void RequestPermission (Action<PermissionStatus> callback);
```

パラメーター:

callback コールバック

戻り値:

なし

説明:

- パーミッション(Access_COARSE_LOCATION)の権限をリクエストします。
- リクエストするとダイアログが表示されます。ユーザによるダイアログの応答は、引数で指定されたコールバックに返されます。

コールバック関数:

void callback (PermissionStatus status);

パラメーター : status : パーミッション応答

戻り値 : なし

説明: RequestPermission()に引数として指定する関数。

PermissionStatus

メンバー

string Name : パーミッション名

bool Granted : 結果(true: 許可, false: 拒否)

void OpenApplicationSettings ();

パラメーター:

なし

戻り値:

なし

説明:

- Androidの設定 > アプリ配下の該当アプリのページを開きます。

Copyright© 2017–2018 WonderLeague Corporation