# KATVR Unity engine plugin usage instructions

KATVR Unity engine plug-in instructions

-2022.03.23

1. **Check before use**
2. For Go and Walk Deluxe users, please confirm that you have received and connected the bottom sensor and foot receiver of the industry application version (the wired version only has the bottom sensor), the runtime of the industry application version, the dongle corresponding to the runtime number, and the runtime number SDK plug-in, and confirm that the sensor's USB cable has been properly connected to the computer, it is recommended to use USB3 0 and above interfaces;

For Walk and Walk Premium users, please confirm that you have received and connected the bottom sensor and foot receiver of the industry application version (the wired version only has the bottom sensor), the runtime of the industry application version, the dongle corresponding to the Runtime number, and the SDK plug-in , And confirm that the USB cable of the sensor has been correctly con nected to the computer. It is recommended to use USB3.0 and above ports.

1. Open the Runtime program and confirm that the data is normal and will change.

Note: If you do not need real-machine testing, you can use Testruntime to test it.

Open the Runtime program and confirm that the data is normal and subject to change.

Note: If you don't need real machine testing, you can use Test-runtime to test.

1. **The plugin uses Plug-in**
2. Open the SDK archive, get and import the Unity project file;

Open the SDK zip package, obtain and import into the Unity project file

1. Attach KATDevice.cs scripts to any scene object, or use them directly;

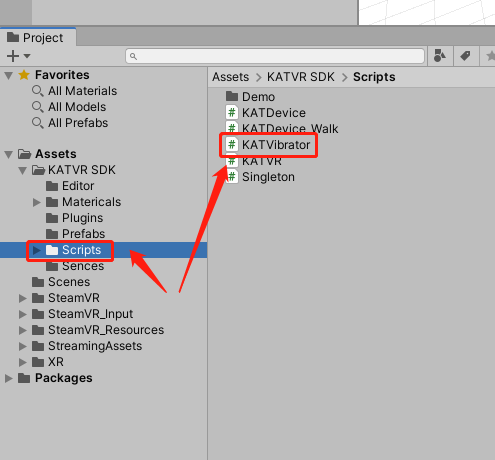
Attach the KATDevice.cs script to any scene object, or use it directly;

1. Prefabs can also be modified directly.

You can also modify the prefab directly.

1. A reference to the vibration module

References to the vibration module



[DllImport("WalkerBase", CallingConvention = CallingConvention.Cdecl)]

public extern static bool Haptic\_Module\_Control(int haptic\_level, int haptic\_time)

// Control vibration module: prioritise new commands

//[haptic\_level] Vibration level: 0-5

//[haptic\_time] duration of vibration: 0-3000 [unit: 100ms]

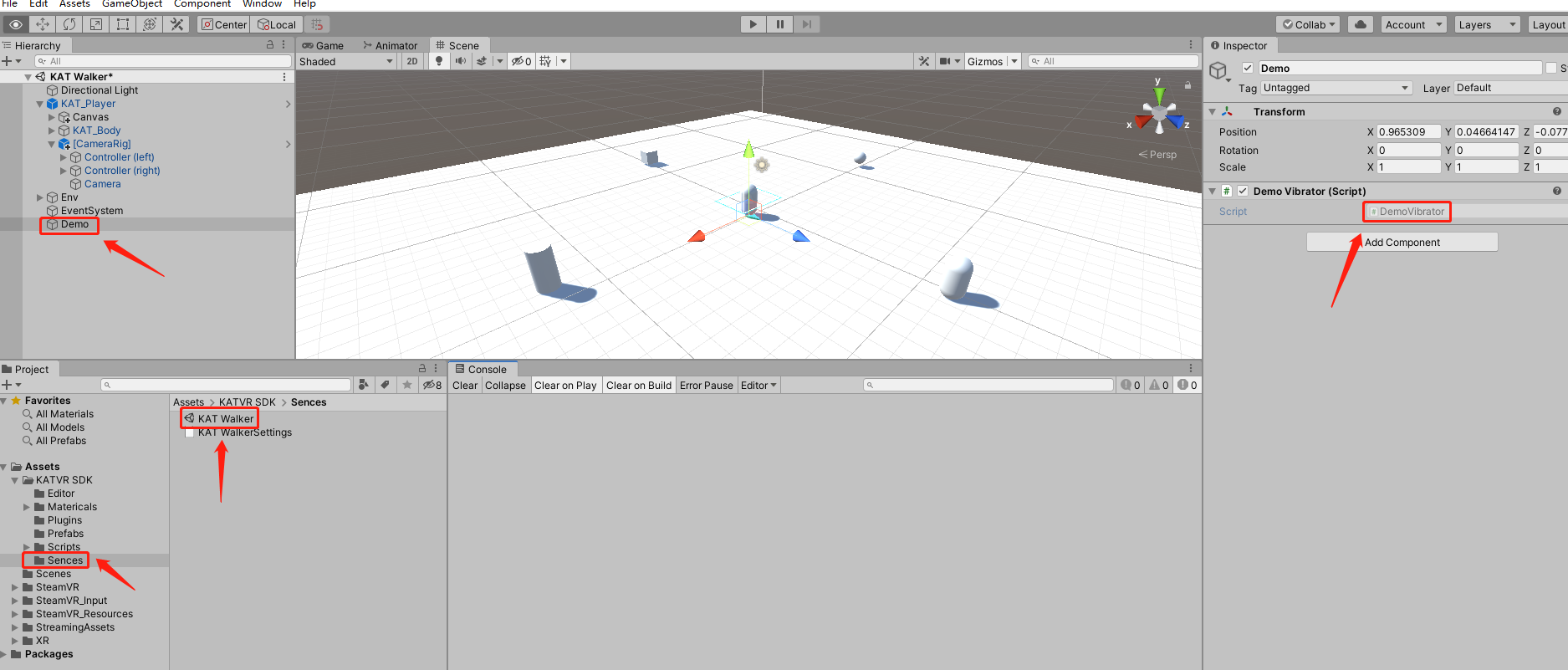
//return value.

//True: command sent successfully

//False: not started successfully, command sending failed

[DllImport("WalkerBase", CallingConvention = CallingConvention.Cdecl)]

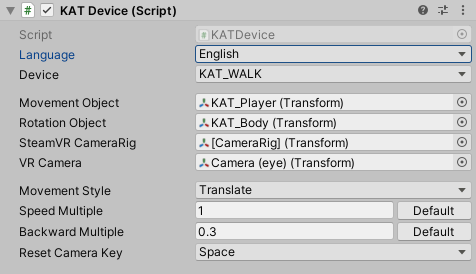
public extern static bool Haptic\_Module\_Control(int haptic\_level, int haptic\_time)



The KAT Walker in the SDK contains a demo to test whether the vibration module is called, and the keyboard key is K (or the handle trigger) with a vibration level of 5 Level, the time is 100s, the time can be adjusted by itself.

A demo is included in the KAT Walker SDK to test whether the vibration module is called, with a keyboard button of K (or the joystick trigger button), a vibration level of 5 and a time of 100s, which can be adjusted by yourself.

1. **The parameter description parameter description**



**Language settings: Select the** display language of the plug-in on the Inspector panel;

Language setting: select the display language of the plug-in on the Inspector panel;

**Device used: Select the type** of KAVR hardware used by the object;

Device: select the KATVR hardware type used by the object;

**Target Moving Object: Select the target object that needs to** be controlled to move, usually the object that the player character needs to move;

Movement object: Select the target object that needs to be controlled to move, usually the object that the player character needs to move;

**Target rotation object: Select the target object that needs to control the rotation, usually the object** that the player character needs to rotate, the target rotation and the target moving animal body can be set to one, and can also refer to the default Prefab object relationship in the plug-in;

Rotation object: Select the target object that needs to be rotated, usually the object that the player character needs to rotate. The target rotation and the target moving object can be set as one, or you can refer to the default Prefab object relationship in the plug-in

**SteamVR Prefabrication:** SteamVR's CameraRig object with the main character selected;

SteamVR CameraRig: select the main character's SteamVR CameraRig object;

**VR main camera:** Select the Camera(eye) object below the CameraRig object above;

VR Camera: select the Camera (eye) object below the CameraRig object above;

**Move:** You can choose the ordinary Translate offset and move by changing the velocity of RigidBody, and developers can also customize some mobile methods;

Movement Style: You can choose ordinary Translate displacement and move by changing the velocity of RigidBody, and developers can also customize some movement methods;

**Speed multiplier:** use this parameter to adjust the player's walking speed when moving forward and backward, and click the button to restore the default value; (It is recommended to use the default values here for magnification of 1 for both forward and backward, or to zoom in and out equally).

Speed Multiple: Use this parameter to adjust the player's walking speed when moving forward and backward, and click the button to restore the default value; ( it is recommended to use the default value for forward and backward magnifications to be 1, or to zoom in and out in equal proportions)

**Camera reset key:** This key is a corrective button to prevent the player from using an abnormal walking direction during the game, and it is recommended to place this button in a position convenient for the player to use when the game is released, such as the application button on the gamepad.

Reset Camera Key : This button is to prevent the player from using the corrective button when the walking direction is abnormal during the game. It is recommended to place this button in a convenient location for the player when the game is released, such as the application button on the handle .

1. **Package the release release**
2. When the project is published, please select the 64-bit version;
3. When the project is released, please choose to release it in a 64-bit version;
4. After the project is published, confirm that WalkBase \_Data files are stored in the Plugins directory in the game's .dll folder.

. After the project is released, confirm that there is a WalkBase.dll file in the Plugins directory in the \_Data folder of the game.

1. **Scene test scene test**
2. The SDK contains a demo scenario for testing to test whether the sensor data is normal;

. The SDK includes a demo scene for testing to test whether the sensor data is normal;

1. Packaged scenarios can be simply tested with Testruntime, and then tested with Developruntime.
2. The packaged scene can be simply tested with Test-runtime, and after no problem, use Developruntime to test.

Note: The SDK package contains instructions for using Testruntime and Developruntime.

Note: The SDK package contains instructions for using Testruntime and Developruntime.