Using the Logitech Gaming SDK with Unity

Contents

Overview

The Logitech Gaming SDK is divided in four categories:

1. Logitech Gaming LCD SDK that enables control to the LCDs on supported Logitech gaming keyboards
2. Logitech Gaming Led SDK that enables led backlighting control on supported Logitech gaming mice and keyboards
3. Logitech Gaming G-Key SDK that enables control to the G-Key featured Logitech mice, headset and keyboard.
4. Logitech Gaming Steering Wheel SDK that enables control to Logitech gaming controllers, like steering wheels, flight system or gamepads.

It can be easily integrated in a Unity game using C# scripting.

Please refer to the related documentation in the package for each SDK:

- LED SDK -> LogitechGamingLEDSDK.pdf
- GKEY SDK -> LogitechGamingLCDSDK.pdf
- LCD SDK -> LogitechGamingGkeySDK.pdf
- Steering Wheel SDK -> LogitechGamingSteeringWheelSDK.pdf

Making the SDK work in your Unity game

The following steps show how to make the Logitech SDK work with a new Unity game. Please adapt the steps to your game for things to work.

Steps

1. Create a new project in Unity editor. Under Import the following selcect Scripts.unityPackage.
2. Check the build settings in the Unity project under File->Build Settings->Architecture
3. In the package downloaded from the asset store, you will find two folders containing version x86 and x64. Choose the one that matches your Unity build settings and copy the library file (dll) in the Unity project folder.
4. Import the LogitechGSDK.cs script that comes as part of the package into your scripts folder
5. Create a c# script and attach it to any object in the stage.
6. Call the functions as needed in the script just created, referencing statically to the LogitechGSDK class. Example: *LogitechGSDK.LogiLedInit()*

For more help on how to call those functions or more detailed examples, please have a look at the sample scripts contained in this package: LogitechGkey.cs, LogitechLCD.cs, LogitechLed.cs, LogitechSteeringWheel.cs
Specifications on the Logitech SDKs can be found in the SDK Documentation folder in the package.

Notes and Remarks

Please note that there are few exceptions in the function names. These differences are due to the type incompatibility in the interop conventions between c# script and c++ dlls.

- G-Key functions
    - **LogiGkeyGetMouseButtonString** has been renamed as **LogiGkeyGetMouseButtonStr** in the Unity version
    - **LogiGkeyGetKeyboardGkeyString** has been renamed as **LogiGkeyGetKeyboardGkeyStr** in the Unity version
    - **LogiGkeyIsMouseButtonPressed** the return type of this function is int, you need to check if it's equal to 1 in order to check its Boolean value.
- Steering wheel function
    - **LogiGetFriendlyProductName** has been renamed as **LogiSteeringGetFriendlyProductName** in the Unity version
    - **LogiGetState** has been renamed as **LogiGetStateUnity** in the Unity version.

To avoid any errors follow the sample scripts in the package.

For questions/comments, email cjuncker@logitech.com, tpigliucci@logitech.com or lbyrne@logitech.com .