Using the Logitech|G SDKs with Unity

Contents

Overview

The Logitech Gaming SDK is divided in five categories:

1. **Logitech|G Arx Control App SDK**,
   Easily create a second screen experience for the PC Game through an applet running on Logitech|G Arx Control app, available for free to everybody on iTunes App Store and Google Play Store.
2. **Logitech|g LED Backlighting**,
   Access backlighting features of Logitech|G devices, now including also per-key color backlighting on featured devices.
3. **Logitech|G G-key SDK,**
   Receive notifications on G-keys events coming from Logitech|G featured mice, headsets and keyboards.
4. **Logitech|G Steering Wheel SDK,**
   Control Logitech Gaming controllers, such as steering wheels, flight controllers or gamepads.

Making the SDK work in your Unity game

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

Steps

- Create a new project in Unity editor and import from asset store the package Logitech Gaming SDK.
- To call any function from Logitech's SDK In your code, import the LogitechGSDK.cs script and call the functions statically. E.g. LogitechGSDK.LogiLedInit();
- According to your build architecture (x86 or x64), copy the desired SDK's dll to your game exe's path from the Lib folder of this package.
- To see examples of function calls and effects of the SDKs, try attaching to an object in your scene one of the samples script that come with this package: LogitechArxControl.cs, LogitechLCD.cs, LogitechLed.cs, LogitechGKey.cs, LogitechSteeringWheel.cs

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

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

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 functions
    - LogiGetFriendlyProductName has been renamed as ogiSteeringGetFriendlyProductName 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 devtechsupport@logitech.com