



# Logitech Gaming G-key SDK V1.02

## Overview and Reference

---

© 2012 Logitech. Confidential

**The Logitech Gaming G-key SDK, including all accompanying documentation, is protected by intellectual property laws. All use of the Logitech Gaming G-key SDK is subject to the License Agreement found in the "Logitech Gaming G-key SDK License Agreement" file and at the end of this document. If you do not agree to the terms and conditions of the License Agreement, you must immediately return any documentation, the accompanying software and all other material provided to you by Logitech. All rights not expressly granted by Logitech are reserved.**

## Contents

Overview .....	4
User experience.....	4
SDK basics .....	4
SDK Package .....	5
Requirements .....	5
Interfacing with the SDK .....	5
Using LogitechGkey.h and LogitechGkey.lib to access LogitechGkey.dll .....	5
Using LogitechGkey.dll directly .....	5
Features of Logitech devices with G-keys .....	5
G710.....	5
G510.....	6
G110.....	6
G19 .....	7
G103.....	7
G105.....	7
G105 Call Of Duty .....	8
G11 .....	8
G13 .....	9
G15 v1.....	9
G15 v2.....	10
G35 .....	10
G930.....	11
Features of Logitech gaming mice with extra buttons .....	12
G600.....	12
G300.....	12
G400.....	12
Do's and Don'ts .....	13
Sample usage of the SDK .....	13
Using header and lib .....	13
Using DLL directly .....	14
Reference.....	16
logiGkeyCBContext Structure .....	16
Parameters for callback function.....	16
LogiGkeyInit.....	17
Parameters .....	17
Return value .....	17

LogiGkeyIsMouseButtonPressed.....	17
Parameters .....	17
Return value .....	17
LogiGkeyGetMouseButtonString .....	17
Parameters .....	18
Return value .....	18
LogiGkeyIsKeyboardGkeyPressed.....	18
Parameters .....	18
Return value .....	18
LogiGkeyGetKeyboardGkeyString .....	18
Parameters .....	18
Return value .....	18
LogiGkeyShutdownFunction.....	18
End-User License Agreement for Logitech Gaming G-key SDK.....	19

## Overview

The Logitech Gaming G-key SDK enables to get the current state of G-keys and extra mouse buttons for supported Logitech gaming mice and keyboards. It can be used via callback or polling. It only works when the Logitech Gaming Software is running (8.30 or later).

## User experience

When the Logitech Gaming Software (LGS) is installed and running (8.30 or later), the user needs to specifically allow their device buttons to report G-keys/buttons directly to the game rather than using macros. If an associated game profile exists in LGS for your application, users can do so by dragging a "G-key" command on top of their G-keys/buttons.

The "G-key" command only appears in LGS for a specific profile after the first time the corresponding application with G-key SDK integration is run. Before that it is not visible anywhere.

If no profile exists when the application is run for the first time, a corresponding profile is created with "G-key" in the commands list.

The SDK will not forward G-key/button events if the user is running an older Logitech Gaming Software (8.20 and earlier).

For mice with on-board modes (G600 and G300), the SDK only works if the mouse is not in on-board mode.

## SDK basics

The SDK is a Windows based API for C/C++ programmers. Games based on the Microsoft Win32 API do not access hardware directly. Instead, the Logitech Gaming G-key SDK interacts with supported Logitech devices on behalf of the games.

When you want to enable your application to receive the G-key and mouse button events, you call `LogiGkeyInit()`, passing in the callback context. After the call to `LogiGkeyInit()`, whenever the user presses/releases a G-key or extra mouse button in your application on the supported Logitech device, your application's callback function will get called with the proper G-key event message.

Applications will only get G-key/button events from keyboard/mouse if the application is currently in the foreground.

The callback function is executed in the context of a thread within the library. Therefore, take the necessary precautions for thread safety should the callback share resources with other threads in your application.

If you would rather poll every frame than to use the callback you can use the `LogiGkeyIsMouseButtonPressed()` and `LogiGkeyIsKeyboardGkeyPressed()` functions. Internally the devices will not get polled every time these functions are called. Instead they simply return the current state based on events received internally. The downside is that a button press could potentially get missed in case a user manages to press and release it within a single game loop.

For convenience when needing to define a string for the G-key or button that was pressed, you can use the `LogiGkeyGetMouseButtonString()` and `LogiGkeyGetKeyboardGkeyString()` functions.

When you are done listening to G-key events, call `LogiGkeyShutdown()` to free up the SDK resources.

For gaming mice, the G-key SDK only reports button events from mouse button 6 and above. Mouse button 1-5 are standard mouse buttons. They are not considered as G-keys.

NOTE: When developing and launching different versions of the same application (debug vs. release) and those different versions have the same executable name, make sure your corresponding profile in LGS points to the same executable. For example if running for the first time with the debug version, a profile will be created in LGS. But then when running a release version, if it has the same executable name, a new profile will not be created, but because the existing profile points to the debug version, things will not work.

## SDK Package

The following files are included:

- LogitechGkey.h: C/C++ header file containing function prototypes
- LogitechGkey.lib: companion lib file to access DLL exported functions (32 and 64 bit)
- LogitechGkey.dll: library of SDK functions (32 and 64 bit)

## Requirements

The Logitech Gaming G-key SDK can be used on the following platforms:

- Windows XP SP2 (32-bit and 64-bit)
- Windows Vista (32-bit and 64-bit)
- Windows 7 (32-bit and 64-bit)
- Windows 8 (32-bit and 64-bit)

The Logitech Gaming G-key SDK is a C based interface and is designed for use by C/C++ programmers. Familiarity with Windows programming is required.

## Interfacing with the SDK

### Using LogitechGkey.h and LogitechGkey.lib to access LogitechGkey.dll

The application can include LogitechGkey.h and link to LogitechGkey.lib (see "Sample usage of the SDK" further below or sample program in Samples folder). Installation folder for the DLL needs to be the same as the main executable, or needs to be part of the Path in the system environment.

### Using LogitechGkey.dll directly

Alternatively the game can use the DLL directly by loading it via [LoadLibrary](#), and accessing its functions using [GetProcAddress](#) (see "Sample usage of the SDK" further below or sample program in Samples folder).

## Features of Logitech devices with G-keys

### G710

G-keys: 1 to 6

Number of modes: 3



### G510

G-keys: 1 to 18

Number of modes: 3



### G110

G-keys: 1 to 12

Number of modes: 3



## G19

G-keys: 1 to 12

Number of modes: 3



## G103

G-keys: 1 to 6

Number of modes: 3



## G105

G-keys: 1 to 6

Number of modes: 3



### G105 Call Of Duty

G-keys: 1 to 6

Number of modes: 3



### G11

G-keys: 1 to 18

Number of modes: 3





### G13

The SDK treats this device as a keyboard.

G-keys: 1 to 29

Number of modes: 3



### G15 v1

G-keys: 1 to 18

Number of modes: 3



### G15 v2

G-keys: 1 to 6

Number of modes: 3



### G35

G-keys: 1 to 3

Number of modes: 1



### G930

G-keys: 1 to 3

Number of modes: 1



## Features of Logitech gaming mice with extra buttons

### G600

Extra buttons: 6 to 20



### G300

Extra buttons: 6 to 9



### G400

Extra buttons: 6 to 8



## Do's and Don'ts

These are a few guidelines that may help you implement 'better' support in your game:

- If using the callback works for you, use it rather than polling. When polling if a user clicks and releases a button within a single game loop the button press will not be seen.

## Sample usage of the SDK

### Using header and lib

```
#include "LogitechGkey.h"

...

void __cdecl GkeySDKCallback(GkeyCode gkeyCode, wchar_t* gkeyOrButtonString, void*
/*context*/)
{
    // Look at gkeyCode to figure out which G-key/mode or button has been pressed or
    released
    ...
}

...

LogiGkeyCBContext gkeyContext;
ZeroMemory(&gkeyContext, sizeof(gkeyContext));
gkeyContext.gkeyCallBack = (LogiGkeyCB)GkeySDKCallback;
gkeyContext.gkeyContext = NULL;

// If polling instead of callback use NULL as argument
LogiGkeyInit(&gkeyContext);

...

// If not using callback, check all G-keys to see if they are being pressed
for (int index = 6; index <= LOGITECH_MAX_MOUSE_BUTTONS; index++)
{
```

```
    if (LogiGkeyIsMouseButtonPressed(index))
    {
        ...
    }
}

for (int index = 1; index <= LOGITECH_MAX_GKEYS; index++)
{
    for (int mKeyIndex = 1; mKeyIndex <= LOGITECH_MAX_M_STATES; mKeyIndex++)
    {
        if (LogiGkeyIsKeyboardGkeyPressed(index, mKeyIndex))
        {
            ...
        }
    }
}

...

LogiGkeyShutdown();
```

## Using DLL directly

```
#define LOGITECH_MAX_MOUSE_BUTTONS 20
#define LOGITECH_MAX_GKEYS 29
#define LOGITECH_MAX_M_STATES 3

typedef struct
{
    unsigned int keyIdx          : 8;          // index of the G key or mouse button, for
example, 6 for G6 or Button 6
    unsigned int keyDown         : 1;          // key up or down, 1 is down, 0 is up
    unsigned int mState          : 2;          // mState (1, 2 or 3 for M1, M2 and M3)
    unsigned int mouse           : 1;          // indicate if the Event comes from a mouse,
1 is yes, 0 is no.
    unsigned int reserved1       : 4;          // reserved1
    unsigned int reserved2       : 16;         // reserved2
} GkeyCode;

// Callback used to allow client to react to the Gkey events. It is called in the context
of another thread.
typedef void (WINAPI *logiGkeyCB)(GkeyCode gkeyCode, const wchar_t* gkeyOrButtonString,
void* context);

typedef struct
{
    logiGkeyCB    gkeyCallBack;
    void*         gkeyContext;
}logiGkeyCBContext;

typedef BOOL (* LPFNDLLGKEYINIT)(logiGkeyCBContext*);
typedef BOOL (* LPFNDLLGKEYISMOUSEBUTTONPRESSED)(int);
typedef wchar_t* (* LPFNDLLGKEYGETMOUSEBUTTONSTRING)(int);
typedef BOOL (* LPFNDLLGKEYISKEYBOARDGKEYBUTTONPRESSED)(int, int);
typedef wchar_t* (* LPFNDLLGKEYGETKEYBOARDGKEYSTRING)(int, int);
typedef void (* LPFNDLLGKEYSHUTDOWN)();
```

```
LPFNDLLGKEYINIT g_lpfndllgkeyinit = NULL;
LPFNDLLGKEYISMOUSEBUTTONPRESSED g_lpfndllgkeyismousebuttonpressed = NULL;
LPFNDLLGKEYGETMOUSEBUTTONSTRING g_lpfndllgkeygetmousebuttonstring = NULL;
LPFNDLLGKEYISKEYBOARDGKEYBUTTONPRESSED g_lpfndllgkeyiskeyboardgkeypressed = NULL;
LPFNDLLGKEYGETKEYBOARDGKEYSTRING g_lpfndllgkeygetkeyboardgkeystring = NULL;
LPFNDLLGKEYSHUTDOWN g_lpfndllgkeyshutdown = NULL;

...

void __cdecl GkeySDKCallback(GkeyCode gkeyCode, wchar_t* gkeyOrButtonString, void*
/*context*/)
{
    // Look at gkeyCode to figure out which G-key/mode or button has been pressed or
    released
    ...
}

...

HINSTANCE logiDllHandle = LoadLibrary(L"LogitechGkey.dll");
if (logiDllHandle != NULL)
{
    g_lpfndllgkeyinit = (LPFNDLLGKEYINIT)GetProcAddress(logiDllHandle, "LogiGkeyInit");
    g_lpfndllgkeyismousebuttonpressed =
(LPFPNDLLGKEYISMOUSEBUTTONPRESSED)GetProcAddress(logiDllHandle,
"LogiGkeyIsMouseButtonPressed");
    g_lpfndllgkeygetmousebuttonstring =
(LPFPNDLLGKEYGETMOUSEBUTTONSTRING)GetProcAddress(logiDllHandle,
"LogiGkeyGetMouseString");
    g_lpfndllgkeyiskeyboardgkeypressed =
(LPFPNDLLGKEYISKEYBOARDGKEYBUTTONPRESSED)GetProcAddress(logiDllHandle,
"LogiGkeyIsKeyboardGkeyPressed");
    g_lpfndllgkeygetkeyboardgkeystring =
(LPFPNDLLGKEYGETKEYBOARDGKEYSTRING)GetProcAddress(logiDllHandle,
"LogiGkeyGetKeyboardGkeyString");
    g_lpfndllgkeyshutdown = (LPFNDLLGKEYSHUTDOWN)GetProcAddress(logiDllHandle,
"LogiGkeyShutdown");

    if (g_lpfndllgkeyinit)
    {
        g_lpfndllgkeyinit(&gkeyContext);
    }
}

...

for (int index = 6; index <= LOGITECH_MAX_MOUSE_BUTTONS; index++)
{
    if (g_lpfndllgkeyismousebuttonpressed(index))
    {
        ...
    }
}

for (int index = 1; index <= LOGITECH_MAX_GKEYS; index++)
{
```

```
for (int mKeyIndex = 1; mKeyIndex <= LOGITECH_MAX_M_STATES; mKeyIndex++)
{
    if (g_lpfnDllGkeyIsKeyboardGkeyPressed(index, mKeyIndex))
    {
        ...
    }
}
...
g_lpfnDllGkeyShutdown();
```

## Reference

### logiGkeyCBContext Structure

The logiGkeyCBContext is used to give the SDK enough information to allow the G-key events to be sent back to your application. The registered callback is called when the user presses/releases the G-key/mouse buttons, and the SDK client is currently in the foreground.

```
typedef struct
{
    logiGkeyCB    gkeyCallBack;
    void*         gkeyContext;
} logiGkeyCBContext;
```

```
typedef struct
{
    unsigned int keyIdx      : 8;          // index of the G key or mouse button, for
example, 6 for G6 or Button 6
    unsigned int keyDown    : 1;          // key up or down, 1 is down, 0 is up
    unsigned int mState     : 2;          // mState (1, 2 or 3 for M1, M2 and M3)
    unsigned int mouse      : 1;          // indicate if the Event comes from a mouse,
1 is yes, 0 is no.
    unsigned int reserved1  : 4;          // reserved1
    unsigned int reserved2  : 16;         // reserved2
} GkeyCode;
```

The callback function logiGkeyCB is defined as follows:

```
typedef void (__cdecl *logiGkeyCB)(GkeyCode gkeyCode, const wchar_t* gkeyOrButtonString,
void* context);
```

#### Parameters for callback function

- gkeyCode
  - This parameter tells you all the information about a G-key event, including the index of the G-key or mouse button which generates the event, the direction of the event (up or down), the M states of the G-key (1, 2 or 3), and if the event comes from a mouse.
- gkeyOrButtonString
  - Friendly string for the G-key or button event received. Can be used anywhere the name of the G-key/button needs to be displayed to the user. An example for keyboards is "G3/M1", and for mice "Mouse Btn 7".



- context
  - This specifies an arbitrary context value of the application that is passed back to the client in the event that the registered gkeyCallback function is invoked.

#### Remarks

Note that the callback function is executed in the context of a thread within the library. Therefore, take the necessary precautions for thread safety should the callback share resources with other threads in your application.

### LogiGkeyInit

The **LogiGkeyInit()** function initializes the G-key SDK. It must be called before your application can see G-key/button events.

```
BOOL LogiGkeyInit(logiGkeyCBContext* gkeyCBContext);
```

#### Parameters

- gkeyCBContext: context for callback. See sample code above or sample program in Samples folder. This value can be set to NULL if the game wants to use the polling functions instead of a callback.

#### Return value

If the function succeeds, it returns TRUE. Otherwise FALSE.

#### Remarks

Use this initialization if working with any application that is not built using Unreal Engine or Unity game engine. For these two game engines use appropriate function as follows :

- Unreal Engine -> LogiGkeyInitWithoutCallback()
- Unity -> LogiGkeyInitWithoutContext

See the examples in the relative documentation to see how to get those functions to work.

### LogiGkeyIsMouseButtonPressed

The **LogiGkeyIsMouseButtonPressed()** function indicates whether a mouse button is currently being pressed.

```
BOOL LogiGkeyIsMouseButtonPressed(int buttonNumber);
```

#### Parameters

- buttonNumber: number of the button to check (for example between 6 and 20 for G600).

#### Return value

TRUE if the specified button is currently being pressed, FALSE otherwise.

### LogiGkeyGetMouseButtonString

The **LogiGkeyGetMouseButtonString()** function returns a button-specific friendly string.

```
wchar_t* LogiGkeyGetMouseButtonString(int buttonNumber);
```

### Parameters

- buttonNumber: number of the button to check (for example between 6 and 20 for G600).

### Return value

Friendly string for specified button number. For example "Mouse Btn 8".

## LogiGkeyIsKeyboardGkeyPressed

The **LogiGkeyIsKeyboardGkeyPressed()** function indicates whether a keyboard G-key is currently being pressed.

```
BOOL LogiGkeyIsKeyboardGkeyPressed(int gkeyNumber, int modeNumber);
```

### Parameters

- gkeyNumber: number of the G-key to check (for example between 1 and 6 for G710).
- modeNumber: number of the mode currently selected (1, 2 or 3)

### Return value

TRUE if the specified G-key for the specified Mode is currently being pressed, FALSE otherwise.

## LogiGkeyGetKeyboardGkeyString

The **LogiGkeyGetKeyboardGkeyString()** function returns a G-key-specific friendly string.

```
wchar_t* LogiGkeyGetKeyboardGkeyString(int gkeyNumber, int modeNumber);
```

### Parameters

- gkeyNumber: number of the G-key to check (for example between 1 and 6 for G710).
- modeNumber: number of the mode currently selected (1, 2 or 3)

### Return value

Friendly string for specified G-key and Mode number. For example "G5/M1".

## LogiGkeyShutdownFunction

The **LogiGkeyShutdown** function unloads the corresponding DLL and frees up any allocated resources.

```
void LogiGkeyShutdown();
```

## End-User License Agreement for Logitech Gaming G-key SDK

This End-User License Agreement for Logitech Gaming G-key SDK ( "Agreement") is a legal agreement between you, either an individual or legal entity ("You" or "you") and Logitech Inc. ("Logitech") for use of the Logitech Gaming G-key software development kit, which includes computer software and related media and documentation (hereinafter "Logitech Gaming G-key SDK"). By using this Logitech Gaming G-key SDK, you are agreeing to be bound by the terms and conditions of this Agreement. If you do not agree to the terms and conditions of this Agreement, promptly return the Logitech Gaming G-key SDK and other items that are part of this product in their original package, or if you have downloaded this software from a Logitech or a Distributor web site, then you must stop using the software and destroy any copies of the software in your possession or control.

**1 Grant of License and Restrictions.** This Agreement grants You the following rights provided that You comply with all terms and conditions of this Agreement.

- (a) Logitech grants You a limited, non-exclusive, nontransferable license to install and use an unlimited number of copies of the Logitech Gaming G-key SDK on computers. All other rights are reserved to Logitech.
- (b) You shall not reverse engineer, decompile or disassemble any portion of the Logitech Gaming G-key SDK, except and only to the extent that this limitation is expressly prohibited by applicable law.
- (c) At your option, you may provide reasonable feedback to Logitech, including but not limited to usability, bug reports and test results, with respect to the Logitech Gaming G-key SDK. All bug reports, test results and other feedback provided to Logitech by You shall be the property of Logitech and may be used by Logitech for any purpose.
- (d) In the event Logitech, in its sole discretion, elects to provide copies of the Logitech Gaming G-key SDK to more than one individual employed by You (if You are not a single individual), each such individual shall be entitled to exercise the rights granted in this Agreement and shall be bound by the terms and conditions herein.

**2 Updates.** Logitech is not obligated to provide technical support or updates to You for the Logitech Gaming G-key SDK provided to You pursuant to this Agreement. However, Logitech may, in its sole discretion,

provide further pre-release versions, technical support, updates and/or supplements (“Updates”) to You, in which case such Updates shall be deemed to be included in the “Logitech Gaming G-key SDK” and shall be governed by this Agreement, unless other terms of use are provided in writing by Logitech with such Updates.

- 3 Intellectual Property Rights.** The Logitech Gaming G-key SDK is licensed, not sold, to You for use only under the terms and conditions of this Agreement. Logitech and its suppliers retain title to the Logitech Gaming G-key SDK and all intellectual property rights therein. The Logitech Gaming G-key SDK is protected by intellectual property laws and international treaties, including U.S. copyright law and international copyright treaties. All rights not expressly granted by Logitech are reserved.
- 4 Disclaimer of Warranty.** TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, LOGITECH, ITS SUPPLIERS AND DISTRIBUTORS PROVIDE THE LOGITECH GAMING G-key SDK AND OTHER LOGITECH PRODUCTS AND SERVICES (IF ANY) AS IS AND WITHOUT WARRANTY OF ANY KIND. LOGITECH AND ITS SUPPLIERS AND DISTRIBUTORS EXPRESSLY DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD-PARTY RIGHTS WITH RESPECT TO THE LOGITECH GAMING G-KEY SDK AND ANY WARRANTIES OF NON-INTERFERENCE OR ACCURACY OF INFORMATIONAL CONTENT. NO LOGITECH DISTRIBUTOR, AGENT, OR EMPLOYEE IS AUTHORIZED TO MAKE ANY MODIFICATION, EXTENSION, OR ADDITION TO THIS WARRANTY. Some jurisdictions do not allow limitations on how long an implied warranty lasts, so the above limitation may not apply to you.
- 5 Limitation of Liability.** IN NO EVENT WILL LOGITECH, ITS SUPPLIERS, OR DISTRIBUTORS BE LIABLE FOR ANY COSTS OF PROCUREMENT OF SUBSTITUTE PRODUCTS OR SERVICES, LOST PROFITS, LOSS OF INFORMATION OR DATA, OR ANY OTHER SPECIAL, INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES ARISING IN ANY WAY OUT OF THE SALE OF, USE OF, OR INABILITY TO USE THE LOGITECH GAMING G-KEY SDK OR ANY LOGITECH PRODUCT OR SERVICE, EVEN IF LOGITECH HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO CASE SHALL LOGITECH'S, ITS SUPPLIERS' AND DISTRIBUTORS' TOTAL LIABILITY EXCEED THE ACTUAL MONEY PAID FOR THE LOGITECH PRODUCT OR SERVICE GIVING RISE TO THE LIABILITY.

Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you. The above limitations will not apply in case of personal injury where and to the extent that applicable law requires such liability.

- 6 U.S. Government Rights.** Use, duplication, or disclosure of the software contained in the Logitech Gaming G-key SDK by the U.S. Government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (OCT 1988) FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable. Logitech Inc. 7600 Gateway Blvd, Newark, CA 94560.
- 7 Export Law Assurances.** You agree and certify that neither the Logitech Gaming G-key SDK nor any other technical data received from Logitech will be exported outside the United States except as authorized and as permitted by the laws and regulations of the United States. If you have rightfully obtained the Logitech Gaming G-key SDK outside of the United States, you agree that you will not re-export the Logitech Gaming G-key SDK nor any other technical data received from Logitech, except as permitted by the laws and regulations of the United States and the laws and regulations of the jurisdiction in which you obtained the Logitech Gaming G-key SDK.
- 8 Termination:** This Agreement is effective until terminated. Upon any violation of any of the provisions of this Agreement, or any provisions of any agreement between you and a Distributor, rights to use the Logitech Gaming G-key SDK shall automatically terminate and the Logitech Gaming G-key SDK must be returned to Logitech or all copies of the Logitech Gaming G-key SDK destroyed. You may also terminate this Agreement at any time by destroying all copies of the Logitech Gaming G-key SDK in your possession or control. If Logitech makes a request via public announcement or press release to stop using the copies of the Logitech Gaming G-key SDK, you will comply immediately with this request. The provisions of paragraphs 3, 7, 8 and 12 will survive any termination of this Agreement.
- 9 General Terms and Conditions.** If You are an individual signing this Agreement on behalf of a company, then You represent that You have authority to execute this Agreement on behalf of such company. This Agreement will be governed by and construed in accordance with the laws of the United States and the State of California, without regard to or application of its choice of law rules or principles. If for any reason a

court of competent jurisdiction finds any provision of this Agreement, or portion thereof, to be unenforceable, that provision of the Agreement shall be enforced to the maximum extent permissible so as to affect the intent of the parties, and the remainder of this Agreement shall continue in full force and effect. This Agreement constitutes the entire agreement between You and Logitech respect to the use of the Logitech Gaming G-key SDK and supersedes all prior or contemporaneous understandings, communications or agreements, written or oral, regarding such subject matter.