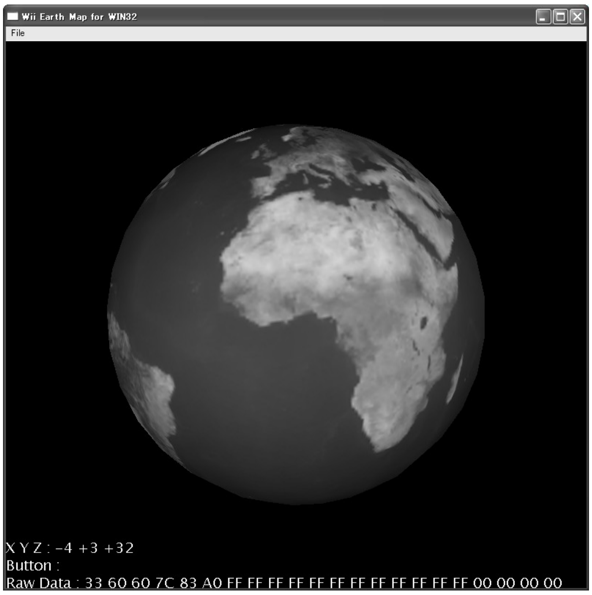


```
X Y Z : -4 +3 +32
Button :
Raw Data : 33 60 60 7C 83 A0 FF FF FF FF FF FF FF FF FF FF FF FF 00 00 00 00
```

CyberGarage

Wiimote for C++

User’s Guide

Document Revision History

Modified	Description
May 29, 2007	The first release.
Jun 6, 2007	Added the installation about Windows DDK

Table of Contents

Introduction

CyberGarage Wiimote for C++ is a development library to use Wiimote. I would like to release the library for any platforms such as Linux and MacOSX finally, but I have released it only for WIN32 platforms at the first release.

To know about Wiimote in more detail, see http://www.wiili.org/index.php/Wiimote.

Setup

Package Contents

The package has the header files, the souce files and the project files to build the library with the samples. The files are included the following directries.

File Type		Directory
Source files		src
Header Files		include
Sample files		sample
Project files	WindowsXP (Visual Studio 2005)	*/win32/vs2005

Building library and samples

On Windows platforms, build using the projects for Visual Studio 2005. I checked the package to build and execute normally on the following platforms.

Platform	Version	Environment
WindowsXP	Service Pack2	Visual Studio 2005

On Windows platforms, you have to install the following library. Then you have to add the include directory which has setupapi.h and hidsdi.h, add the library directory which has hid.lib and setupapi.lib into the directory setting of Visual Studio 2005.

Package	URL
DDK - Windows Driver Development Kit	http://www.microsoft.com/whdc/DevTools/ddk/default.mspx

Using Wiimote

Class

The following is the main class to use Wiimote. The class is defined using a name space, CyberGarage.

```
namespace CyberGarage {

class Wiimote : public UsbHID

{

public:

        Wiimote();

        virtual ~Wiimote();

        bool open();

        bool setLEDs(bool led1, bool led2, bool led3, bool led4);

        bool IsAPressed()

        bool IsBPressed()

        bool IsOnePressed()

        bool IsTwoPressed()

        bool IsLeftPressed()

        bool IsRightPressed()

        bool IsUpPressed()

        bool IsDownPressed()

        bool IsMinusPressed()

        bool IsPlusPressed()

        bool IsHomePressed()

        unsigned char getXMotion()

        unsigned char getYMotion()

        unsigned char getZMotion()

};


}
```

Initialize

Use Wiimote::open() to initialize a Wiimote. The method returns true if Wiimote is initialized normally, otherwise false.

```
using namespace CyberGrage;

Wiimote *wiimote = new Wiimote();

if (wiimote->open() == false) {

        delete wiimote;
        printf("Could not find Wiimote !!");
}
```

Update

To update data of Wiimote, use Wiimote::read(). The method returns a number of the read data. The method doesn’t lock when the Wiimote doesn’t send any data.

```
Wiimote *wiimote = …
  ..........
wiimote->read();
```

Button/Motion

To get the button status, use the following methods of the Wiimote class.

Method	Type	Data
IsAPressed()	Button	A
IsBPressed()		B
IsOnePressed()		One
IsTwoPressed		Two
IsLeftPressed()		Left
IsRightPressed()		Rigth
IsUpPressed()		Up
IsDownPressed()		Down
IsMinusPressed()		Minus
IsPlusPressed()		Plus
IsHomePressed()		Home
getXMotion()	Motion	X
getYMotion()		Y
getZMotion()		Z

Sample

The package has the following samples using the Wiimote for C++.

Name	Platform	Directory
Wiimote Console	WindowsXP (Visual Studio 2005)	sample/console/win32/vs2005
Wiimote Earth	WindowsXP (Visual Studio 2005)	sample/wiiearth/win32/vs2005

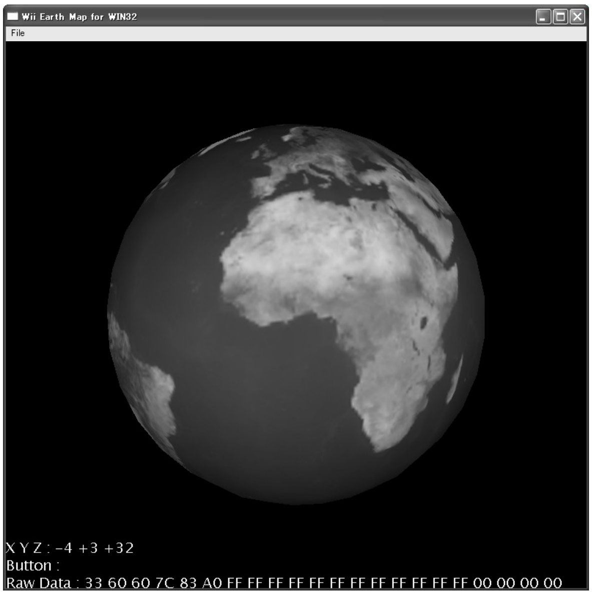
Wiimote Console

This is a most simple sample which outputs data of a Wiimore to the standard console. The main source code is the following.

```
#include <stdio.h>
#include <cybergarage/wii/Wiimote.h>
using namespace CyberGarage;
int main( int argc, char *argv[] )
{
       Wiimote *wiimote;
       wiimote = new Wiimote();
       if (wiimote->open() == false) {
                delete wiimote;
                printf("Could not find Wiimote !!");
                return 0;
       }
       do {
                wiimote->read();
                printf("\r%+d %+d %+d %s %s %s %s %s %s %s %s %s %s %s",
                          (wiimote->getXMotion()-0x80),
                          (wiimote->getYMotion()-0x80),
                          (wiimote->getZMotion()-0x80),
                          (wiimote->IsAPressed() ? "A" : " "),
                          (wiimote->IsBPressed() ? "B" : " "),
                          (wiimote->IsOnePressed() ? "1" : " "),
                          (wiimote->IsTwoPressed() ? "2" : " "),
                          (wiimote->IsUpPressed() ? "U" : " "),
                          (wiimote->IsDownPressed() ? "D" : " "),
                          (wiimote->IsLeftPressed() ? "L" : " "),
                          (wiimote->IsRightPressed() ? "R" : " "),
                          (wiimote->IsMinusPressed() ? "-" : " "),
                        (wiimote->IsPlusPressed() ? "+" : " "),
                        (wiimote->IsHomePressed() ? "H" : " ")
                        );
                Sleep(100);
        } while (wiimote->IsHomePressed() == false);
        wiimote->close();
        delete wiimote;
        return 0;
}
```

Wiimote Earth

This is a 3D sample using Wiimote to rotate and translate a earth model. The image is the following.



```
X Y Z : -4 +3 +32
Button :
Raw Data : 33 60 60 7C 83 A0 FF FF FF FF FF FF FF FF FF FF FF FF 00 00 00 00
```

To use a Wiimote, you can rotate the earth easily. To build the sample, you have to install the following package. The sample is based on the other 3D library.

Package	URL
CyberX3D for C++	

License

Subject to the below, redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the names of the Copyright Owners nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT OWNERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

While any of the respective Copyright Owners licenses and/or distributes this software and any files contained therein under its or third party copyrights, for clarification it is hereby stated that no patent license explicit or implied is granted by any of the Copyright Owners in connection to this license or distribution of this software.

In addition to the disclaimer above and not limiting its generality, no assurances are provided by the Copyright Owners that the software does not infringe patents of either third parties or any of the Copyright Owners in any particular jurisdiction. As a condition to exercising the rights and licenses granted hereunder in any particular jurisdiction, each recipient hereby assumes sole responsibility to procure licenses under any relevant patents in that jurisdiction. For example, if a third party patent license is required to allow recipient to distribute the program in a particular country, it is the recipient's responsibility to acquire that license before distributing the program.