

L'API WiiRemoteJ

UE optionnelle EVC
Master Informatique 2 GL et Mitic

2009-2010

Thierry Duval
IFSIC / IRISA - Bunraku
thierry.duval@irisa.fr

API pour la Wiimote

- http://wiibrew.org/wiki/Wiimote_Driver
- API C/C++ :
 - ✓ CWiid (Linux)
 - ✓ Wiiuse
 - ✓ WiiuseCpp
- API Java :
 - ✓ WiimoteSimple
 - ✓ **WiiRemoteJ**
 - ✓ motej
 - ✓ WiiuseJ
- API Python :
 - ✓ PyWii

WiiRemoteJ

- API 100% Java destinée à piloter une wiimote Nintendo
 - ✓ basée sur Java Bluetooth API, JSR-082
 - x « platform-independent »
- Permet d'accéder ou de contrôler :
 - ✓ la caméra infra-rouge (IR)
 - ✓ l'accéléromètre
 - ✓ les extensions
 - x « Classic Controller »
 - x « Nunchuk »
 - x « Guitar »
 - ✓ les boutons
 - ✓ les LEDs
 - ✓ la vibration
 - ✓ le haut-parleur

Installation WiiRemoteJ

➤ Sous Eclipse :

- ✓ ajouter des jar externes dans le « Java Build Path »
 - x bluecove-2.1.0.jar
 - x bluecove-gpl-2.1.0.jar
 - x WiiRemoteJ.jar
- ✓ penser à les inclure lors de la création d'un exécutable

TestWiimoteJ.java (1/9)

```
import java.io.IOException;
import com.intel.bluetooth.BlueCoveConfigProperties;
import wiiremotej.AccelerationConstants ;
import wiiremotej.ButtonMap ;
import wiiremotej.NunchukExtension ;
import wiiremotej.WiiRemote ;
import wiiremotej.WiiRemoteExtension ;
import wiiremotej.WiiRemoteJ;
import wiiremotej.event.WRAccelerationEvent ;
import wiiremotej.event.WRButtonEvent ;
import wiiremotej.event.WRCombinedEvent ;
import wiiremotej.event.WRExtensionEvent ;
import wiiremotej.event.WRIREvent ;
import wiiremotej.event.WRNunchukExtensionEvent ;
import wiiremotej.event.WRStatusEvent ;
import wiiremotej.event.WiiRemoteListener ;
```

TestWiimoteJ.java (2/9)

```
public class TestWiimote {  
  
    WiiRemote wiimote = null ;  
  
    public TestWiimote () {  
        wiimote = null ;  
        try {  
            //wiimote = WiiRemoteJ.findRemote();  
            wiimote = WiiRemoteJ.connectToRemote ("0021BD776E90") ;  
            wiimote.addWiiRemoteListener (new WiimoteListener () ) ;  
            wiimote.setIRSensorEnabled (true, WRIREvent.BASIC) ;  
            wiimote.setAccelerometerEnabled (true) ;  
        } catch (Exception e) {  
            e.printStackTrace () ;  
        }  
    }  
}
```

TestWiimoteJ.java (3/9)

```
public static void main (String [] args) {  
    System.setProperty (  
        BlueCoveConfigProperties.PROPERTY_JSR_82_PSM_MINIMUM_OFF,  
        "true") ;  
    new TestWiimote () ;  
}
```

TestWiimoteJ.java (4/9)

```
class WiimoteListener implements WiiRemoteListener {
```

```
    public void IRInputReceived(WRIREvent ire) {  
        for (int i = 0 ; i < ire.getIRLights ().length ; i++) {  
            if (ire.getIRLights () [i] != null) {  
                System.out.println (ire.getIRLights () [i].getX () + " " +  
                    ire.getIRLights () [i].getY () );  
            }  
        }  
    }  
}
```

```
    public void accelerationInputReceived(WRAccelerationEvent ae) {  
        System.out.println ("wiimote x acceleration : " + ae.getXAcceleration () );  
        System.out.println ("wiimote y acceleration : " + ae.getYAcceleration () );  
        System.out.println ("wiimote z acceleration : " + ae.getZAcceleration () );  
    }  
}
```


TestWiimoteJ.java (5/9)

```
public void buttonInputReceived (WRButtonEvent be) {  
    if (be.wasPressed (WRButtonEvent.ONE)) System.out.println ("1 pressed") ;  
    if (be.wasPressed (WRButtonEvent.TWO)) ... ;  
    if (be.wasPressed (WRButtonEvent.B)) ... ;  
    if (be.wasPressed (WRButtonEvent.A)) ... ;  
    if (be.wasPressed (WRButtonEvent.MINUS)) ... ;  
    if (be.wasPressed (WRButtonEvent.HOME)) ... ;  
    if (be.wasPressed (WRButtonEvent.LEFT)) ... ;  
    if (be.wasPressed (WRButtonEvent.RIGHT)) ... ;  
    if (be.wasPressed (WRButtonEvent.DOWN)) ... ;  
    if (be.wasPressed (WRButtonEvent.UP)) ... ;  
    if (be.wasPressed (WRButtonEvent.PLUS)) ... ;  
}
```

TestWiimoteJ.java (6/9)

```
if (be.wasReleased (WRButtonEvent.ONE)) System.out.println ("1 released") ;
if (be.wasReleased (WRButtonEvent.TWO)) ... ;
if (be.wasReleased (WRButtonEvent.B)) ... ;
if (be.wasReleased (WRButtonEvent.A)) ... ;
if (be.wasReleased (WRButtonEvent.MINUS)) ... ;
if (be.wasReleased (WRButtonEvent.HOME)) ... ;
if (be.wasReleased (WRButtonEvent.LEFT)) ... ;
if (be.wasReleased (WRButtonEvent.RIGHT)) ... ;
if (be.wasReleased (WRButtonEvent.DOWN)) ... ;
if (be.wasReleased (WRButtonEvent.UP)) ... ;
if (be.wasReleased (WRButtonEvent.PLUS)) ... ;
}

public void combinedInputReceived(WRCombinedEvent arg0) {
    //System.out.println ("combinedInputReceived") ;
}
```

TestWiimoteJ.java (7/9)

```
public void disconnected() {
    System.out.println ("disconnected") ;
}

public void extensionConnected(WiiRemoteExtension arg0) {
    System.out.println ("extensionConnected") ;
    try {
        wiimote.setExtensionEnabled (true) ;
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void extensionDisconnected(WiiRemoteExtension arg0) {
    System.out.println ("extensionDisconnected") ;
}
```

TestWiimoteJ.java (8/9)

```
public void extensionInputReceived (WRExtensionEvent ee) {
    if (ee instanceof WRNunchukExtensionEvent) {
        WRNunchukExtensionEvent nee = (WRNunchukExtensionEvent)ee ;
        WRAccelerationEvent ae = nee.getAcceleration () ;
        System.out.println ("nunchuk x acceleration : " + ae.getXAcceleration ()) ;
        System.out.println ("nunchuk y acceleration : " + ae.getYAcceleration ()) ;
        System.out.println ("nunchuk z acceleration : " + ae.getZAcceleration ()) ;
        if (nee.wasReleased (WRNunchukExtensionEvent.C)) { ... } ;
        if (nee.wasReleased (WRNunchukExtensionEvent.Z)) { ... } ;
        if (nee.wasPressed (WRNunchukExtensionEvent.C)) { ... } ;
        if (nee.wasPressed (WRNunchukExtensionEvent.Z)) { ... } ;
        System.out.println ("x = " + nee.getAnalogStickData ().getX ()) ;
        System.out.println ("y = " + nee.getAnalogStickData ().getY ()) ;
        System.out.println ("angle = " + nee.getAnalogStickData ().getAngle ()) ;
    }
}
```

TestWiimoteJ.java (9/9)

```
public void extensionPartiallyInserted () {  
    System.out.println ("extensionPartiallyInserted") ;  
}  
  
public void extensionUnknown () {  
    System.out.println ("extensionUnknown") ;  
}  
  
public void statusReported (WRStatusEvent arg0) {  
    System.out.println ("statusReported") ;  
}  
  
}  
}
```