

WiiFlash-Einführung

1. Einleitung

Dieses kleine Tutorial soll in den grundlegenden Umgang mit dem von Thibault Imbert und Joa Ebert erstellten WiiFlashServer einführen, der es ermöglicht, die verschiedenen Eingabegeräte für die Nintendo Wii mit Flash anzusprechen.

Konkret geht es an dieser Stelle nur um die Wiimote, obwohl es damit auch möglich ist, das BalanceBoard oder die Nunchuk usw. anzusteuern.

Als Hinweis vorweg sei gesagt, dass es Unterschiede zwischen der Syntax der PC- und Mac-Version des Servers gibt. Bis jetzt zeigten sich diese nicht besonders gravierend und zu umgehen. Der weitere Verlauf des Experiments bleibt allerdings spannend.

2. Erste Schritte

2.1. Einrichten der Serververbindung

Zuerst wird das Serverprogramm installiert, die Macversion ist unter lab.tojio.com und die PC-Version unter wiiflash.org zu finden.

Wenn der heruntergeladene Server entpackt ist, wird die Wiimote mit dem Bluetooth synchronisiert. Das funktioniert auf den verschiedenen Systemen leicht unterschiedlich. Man startet zuerst den Bluetooth-Dialog und stellt dann die Verbindung mit der Wiimote her, die dann zum Beispiel als „NINTENDO-RVL-CNT-01“ angezeigt wird. Die Wiimote selbst beginnt die Synchronisation über das gleichzeitigen Druck auf den „1“- und „2“-Button oder auf den Synchronisationsbutton im Batteriefach. Im Bluetooth-Dialog selbst muss man meist erst die Geräte durchsuchen und dann die gefundene Wiimote verbinden.

Der nächste Schritt ist es, den Server zu starten. Dieser erkennt die Wiimote automatisch, wenn sie korrekt verbunden ist und das Symbol für die angeschlossene Wiimote erscheint in der Serveranzeige (beim PC vibriert die Wiimote dann auch).

2.2. Anlegen der Flash-Datei

Wenn dann die Arbeitsdatei in Flash erstellt wird muss auf Folgendes geachtet werden: Die Datei muss in ActionScript 3 angelegt werden und es empfiehlt sich, sie sofort in einem Extraordner zu speichern. Das ist sehr wichtig, weil Flash sonst nicht die notwendigen WiiFlash-Klassen finden kann (eine echte Fehlerquelle).

Als Nächstes wird der .org-Ordner, der sich in dem heruntergeladenen Paket befindet, in das Arbeitsverzeichnis kopiert.

Nun kann Flash auf die Klassen zugreifen, die für den WiiFlash-Server notwendig sind und es kann die eigentliche Programmierung beginnen.

2.3. Import der WiiFlash-Klassen in ActionScript

Zunächst werden die notwendigen Klassen in das erste Bild, das den ActionScript-Code enthält, importiert.

```
import org.wiiflash.Wiimote;
import org.wiiflash.events.ButtonEvent;
import org.wiiflash.events.WiimoteEvent;
import flash.events.*;
```

Hier sieht man, dass Flash auf den Ordner „org“ zugreift. `import` ist standardmäßig bei AS die Methode, mit der Klassen importiert werden. Danach wird auf das Verzeichnisse „org.wiimote“ und „org.wiimote.events“ verwiesen, in dem sich die Klassen für die Wiimote befinden. Mit `import flash.events.*;` werden alle Eventklassen von AS3 importiert.

Als nächstes wird das Wiimote-Objekt instanziiert, mit dem weitergearbeitet werden soll.

```
var myWiimote:Wiimote = new Wiimote();
```

`myWiimote` ist ein neues Objekt der Klasse `Wiimote` und erbt damit ihre ganzen Events, Eigenschaften und Methoden. Der Name des instanziierten Objekts steht frei, die berühmten Beispiele „hund“ oder „katze“ können natürlich auch verwendet werden.

Gibt es mehrere Wiimotes, die verbunden sind, muss man mehrere Wiimote-Objekte instanziiieren. Diesen wird eine ID zugewiesen (mehr dazu später).

Danach verbindet man das Wiimote-Objekt über die Methode `connect`.

```
myWiimote.connect ();
```

Damit ist die Wiimote verbunden und die Grundlagen für die weitere Programmierung gelegt.

3. Programmieren

3.1. EventListener

Mit AS3 wurde das Konzept des Eventlisteners erweitert und es ersetzt einiges, was in AS2 grundlegend war. Eventlistener können an Objekte gehängt werden und haben die Funktion, das Eintreten bestimmter Ereignisse (wie Eingaben durch den Nutzer oder das Betreten eines Schlüsselbilds) zu überwachen. Treten die erwarteten Ereignisse ein, werden andere, festgelegte Ereignisse ausgelöst, was ungefähr dem Konzept des „on“ in AS2 entspricht.

Das Wiimote-Objekt besitzt auch einige dieser Eventlistener. Diese funktionieren so:

```
myWiimote.addEventListener( ButtonEvent.A_PRESS,  
onAPressed );
```

myWiimote erhält den Eventlistener `ButtonEvent.A_PRESS` über die Methode `addEventListener`. Dieser fragt ab, ob der „A“-Button gedrückt wird. Tritt dieses Ereignis ein wird die Funktion `onAPressed` ausgelöst die dann folgendermaßen definiert werden könnte:

```
function onAPressed ( pEvt:ButtonEvent ):void {  
    trace("A wird gedrückt");  
}
```

Es ist wichtig, dass bei der Funktionsdefinition darauf geachtet wird, welche Art von Ereignis eintritt, hier trat zum Beispiel mit `pEvt:ButtonEvent`, ein `ButtonEvent` ein. **Dringend zu beachten ist, dass `pEvt` nur bei den Button- und den Wiimote-events eingesetzt wird. Bei den anderen wird `pEvent` geschrieben.**

Die anderen Eventlistener können dann einfach an der Stelle, an der jetzt `ButtonEvent.A_PRESS` stand, eingefügt werden.

Folgende `ButtonEvents` sind noch möglich:

`ButtonEvent.A_RELEASE` „A“ wird losgelassen

`ButtonEvent.B_PRESS` „B“ wird gedrückt

`ButtonEvent.B_RELEASE` „B“ wird losgelassen

`ButtonEvent.LEFT_PRESS`,
`ButtonEvent.RIGHT_PRESS`,
`ButtonEvent.UP_PRESS`,
`ButtonEvent.DOWN_PRESS` Das Steuerkreuz wird nach links, rechts, oben oder unten gedrückt

`ButtonEvent.LEFT_RELEASE`,
`ButtonEvent.RIGHT_RELEASE`,
`ButtonEvent.UP_RELEASE`,
`ButtonEvent.DOWN_RELEASE` Das Steuerkreuz wird links, rechts, oben oder unten losgelassen

```
ButtonEvent.MINUS_RELEASE,  
ButtonEvent.PLUS_PRESS,  
ButtonEvent.PLUS_RELEASE,  
ButtonEvent.HOME_PRESS  
ButtonEvent.HOME_RELEASE,  
ButtonEvent.ONE_PRESS,  
ButtonEvent.ONE_RELEASE,  
ButtonEvent.TWO_PRESS,  
ButtonEvent.TWO_RELEASE,
```

Genauso funktioniert das bei der „Minus“-, der „Plus“, der „Home“-, der „1“- und der „2“-Taste

```
Event.CONNECT,  
IOErrorEvent.IO_ERROR,  
Event.CLOSE
```

Hier treten die Ereignisse ein, wenn die Wiimote verbunden wurde, es dabei einen Fehler gab oder die Verbindung getrennt wurde.

```
WiimoteEvent.UPDATE
```

Bei diesem Eventlistener wird ein Ereignis ausgelöst, wenn von der Wiimote Daten empfangen werden. Das hat eine recht hohe Frequenz und kann somit als besserer Ersatz für `Event.ENTER_FRAME` verwendet werden.

3.2. Eigenschaften

Die Wiimote besitzt verschiedene Eigenschaften, also Variablen, die abgefragt oder zurückgesandt werden können. Diese können ganz einfach verwendet werden. Beispielsweise so: `kreis_mc.rotation = myWiimote.roll`; Will man eine Variable an das Wiimote-Objekt senden funktioniert das beispielsweise so : `myWiimote.mouseControl = true`;

Hier sollen die wichtigsten aufgelistet werden.

```
myWiimote.roll
```

Gibt an, ob die Wiimote nach vorne oder hinten gedreht ist. Der Wert ist bei +1, wenn die Wiimote 90 Grad nach vorn und bei -1, wenn sie 90 Grad nach hinten gekippt ist

```
myWiimote.pitch
```

Gibt an, ob die Wiimote nach rechts oder links gedreht ist. Der Wert ist bei +1, wenn die Wiimote 90 Grad nach rechts und bei -1, wenn sie 90 Grad nach links gekippt ist

```
myWiimote.yaw
```

Diese Variable entspricht der Drehung der Wiimote entlang der Längsachse. +1 entspricht der „normalen“ Lage und -1 der Drehung nach unten. 0 entspricht einer 90°-Drehung zur Seite.

```
myWiimote.sensorX, myWiimote.sensorY,  
myWiimote.sensorZ
```

Diese drei Variablen entstammen dem Beschleunigungssensor der Wiimote. Je nachdem wie stark die Wiimote in eine bestimmte Richtung (entlang der x-, y- und z-Achse) bewegt wird entstehen Werte zwischen 0 und ca. 1.5. Dabei ist zu beachten, dass die Achsen anders angeordnet sind als im Standardkoordinatensystem (siehe Abb. unten).

```
myWiimote.batteryLevel
```

Gibt mit einem Wert zwischen 0 und 1 an, wie voll die Batterie der Wiimote ist

```
myWiimote.connected
```

Eine boolesche Variable, die angibt, ob die Wiimote verbunden ist

```
myWiimote.rumbleTimeout
```

Löst die Vibration der Wiimote aus, der Wert wird in Millisekunden angegeben

```
myWiimote.mouseControl
```

Boolesche Variable, mit der die Mauskontrolle aktiviert werden kann. Ist die Mauskontrolle aktiviert entspricht die Lage des Cursors dem x- und y-Wert des Sensorpunkts 1 (siehe unten unter `ir`). Bei der PC-Version heißt „Mauskontrolle“ zusätzlich, dass das Drücken der B-Taste an der Wiimote dem Drücken der linken Maustaste entspricht

```
myWiimote.id
```

Wenn man mit mehreren Wiimotes arbeiten will, muss man dem Wiimote-Objekt eine ID zuweisen, was über den Wert `myWiimote.id` funktioniert. Der Wert kann eine Ganzzahl von 1 bis 4 sein. So können vier verschiedene Wiimote-Objekte erstellt werden, die jeweils eine festgelegte ID besitzen.

```
myWiimote.leds
```

`leds` ist eine Eigenschaft, mit der sich die LED-Leiste der Wiimote ansteuern lässt. Hier gibt es auch Unterschiede in der PC- und Mac-Syntax. Beim PC kann man 15 verschiedene Ganzzahlen angeben, die 15 verschiedene Stellungen der LEDs bedeuten.

Eine Besonderheit stellt die Variable `ir` dar. Sie ist eigentlich vielmehr ein Array-Objekt, da sie mehrere Werte enthält. Die Wiimote kann über ihren Infrarotsensor bis zu vier Lichtpunkte erkennen. Daraus ergeben sich die folgenden Variablen:

```
myWiimote.ir.p1, myWiimote.ir.p2,  
myWiimote.ir.p3, myWiimote.ir.p4
```

Boolesche Variablen, die angeben, ob die Wiimote einen Punkt erkennt. Ergibt `myWiimote.ir.p3` den Wert `true` heißt das, dass die Wiimote drei Punkte bzw. den dritten Punkt erkennt

```
myWiimote.ir.x1, myWiimote.ir.x2,  
myWiimote.ir.x3, myWiimote.ir.x4
```

Die Variablen geben den x-Wert des jeweiligen Punkts an

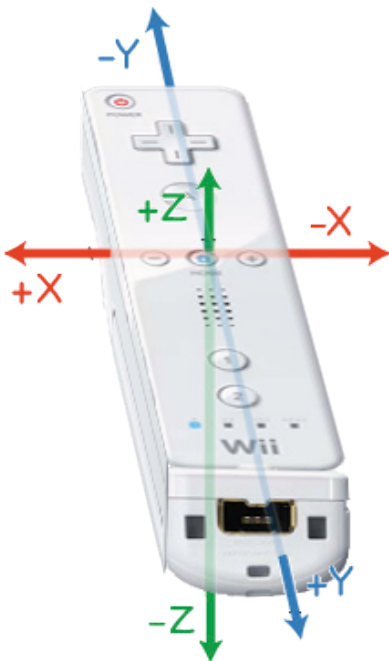
```
myWiimote.ir.y1, myWiimote.ir.y2,  
myWiimote.ir.y3, myWiimote.ir.y4
```

Die Variablen geben den y-Wert des jeweiligen Punkts an

```
myWiimote.ir.size1, myWiimote.ir.size2,  
myWiimote.ir.size3, myWiimote.ir.size4
```

Die Variablen geben die Größe des jeweiligen Punkts an

Dazu, wie die einzelnen Werte sich aus der Bewegung der Wiimote ergeben, habe ich eine Beispieldatei erstellt. Die Abbildung unten zeigt, wie das Koordinatensystem der Wiimote funktioniert.



3.3. Methoden

Als wichtige Methoden sind, neben den bereits bekannten `myWiimote.connect` und `MyWiimote.addEventListener` nur noch `myWiimote.close`, die die Verbindung zur Wiimote trennt, und `removeEventListener`, die `EventListener` wieder entfernt, zu nennen.

4. Troubleshooting

Bei dem Umgang mit dem Wiiflash-Server sind bei mir ein paar typische Fehler aufgetreten, die ich natürlich nicht geheim halten will. Anhand der Checkliste unten kann man überprüfen, ob einer der Fehler vielleicht der Grund ist, warum nichts funktioniert.

Steht die Verbindung zwischen der Wiimote und Flash?

Erstens kann es immer wieder passieren, dass man vergessen hat, überhaupt die Verbindung herzustellen und zweitens hat der Wiiflash-Server beim PC die Tendenz, einfach auszugehen. Flash bringt dann meistens folgende Fehlermeldung:

```
Error #2044: ioError unverarbeitet. text=Error #2031: Socketfehler. URL: 127.0.0.1.
```

Befindet sich der .org-Ordner im gleichen Verzeichnis wie meine Arbeitsdatei bzw. habe ich meine Datei überhaupt gespeichert?

Wie bereits gesagt, muss der org-Ordner sich als Unterverzeichnis in dem gleichen Ordner wie meine Datei befinden, dass Flash auf die notwendigen Klassen zugreifen kann. Dazu muss, logischerweise, die Datei bereits irgendwo abgespeichert sein. Flash beschwert sich darüber, dass man auf nicht definierte Methoden oder Eigenschaften zugreifen will.

Stimmt die Reihenfolge bei der Programmierung?

Mir ist es beispielsweise passiert, dass ich die Wiimote über `connect()` ; verbinden wollte bevor ich überhaupt das Wiimote-Objekt instanziiert hatte.

Habe ich den richtigen Eventlistener eingesetzt?

Wie oben bereits beschrieben muss man darauf achten, welches Ereignis abgefragt wird.

5. Links

<http://lab.tojio.com> - die Seite für den Mac-Server

<http://www.wiiflash.org> - die Seite für den PC-Server

<http://www.wiiflash.org/wiiflash-asdoc> - die Dokumentation

<http://code.google.com/p/wiiflash> - Wiiflash bei Google Code

<http://www.wiili.org> - eine Wiki über die Wii für Linux, hier gibt es auch eine genauere Beschreibung der Wiimote