Rapport Base de Données

Librairie OSCsend pour C#

Professeur : R. Giot

Zhang Mingming
2ème MA Informatique

Année académique 2012-2013

Table des matières

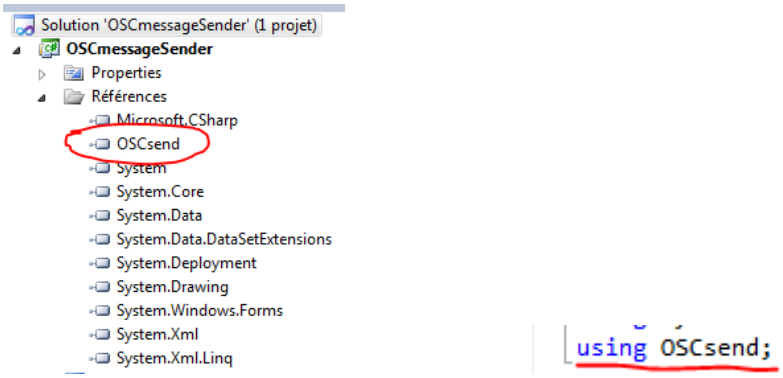
1. Introduction

Dans le cadre de ce cours de Base de Données, il nous a été demandé de coder une librairie (.dll) pour CSharp permettant de facilement envoyer des messages OSC correctement formatés pour des logiciels comme KymaX ou MaxMSP.

2. OSCsend : utilisation & explication

La librairie « OSCsend.dll » est composée de quatre classes. La plus importante des quatre est la classe « OSC ». En effet, c'est dans cette classe que se trouvent les méthodes les plus essentielles.

Pour utiliser OSCsend.dll, il faut tout d'abord l'importer dans les Ressources du projet et ajouter « using OSCsend ; » au début des classes qui ont besoin de l'utiliser pour l'appeler.



Ensuite, il faut créer un nouvel objet OSC en lui donnant comme paramètres l'IP distant (à qui l'on veut envoyer les messages) et le port d'écoute distant.

```
string ip = "192.168.12.16";
int port = 8000;

OSC oscm = new OSC(ip, port);
```

Pour envoyer un message OSC, il faut d'abord créer un objet OSCmsg en donnant comme paramètre l'entête OSC du message.

```
OSCmsg omess = new OSCmsg("/test_send_int_float_string");
omess.addValue(357);
omess.addValue(25.75f);
omess.addValue("test");
oscm.sendOSCmsg(omess);
```

Ici, nous avons créé « omess », un objet OSCmsg et avons donné à ce message OSC un entête « /test_send_int_float_string ». Nous avons ajouté à ce message, une valeur réelle int 357, une valeur décimale flottante float 25.75f et une chaine de caractère string « hello ». Nous avons ensuite envoyé le message OSC « omess » par la méthode sendOSCmsg de l'objet OSC « oscm ».

Nous avons suivi la spécification 1.0 d'OSC, donc il est possible d'ajouter des « int », des « float », des « string » et des « blob » dans un message OSC. Un « blob » est en quelque sorte un tableau de « int », de « float » ou de « string ».

Pour envoyer un blob, il suffit de créer un objet OSCblob, d'ajouter des valeurs dedans, d'ajouter le blob à une OSCmsg et enfin, d'envoyer l'OSCmsg.

```
OSCmsg omess = new OSCmsg("/test send blob");
OSCblob oscb = new OSCblob();
oscb.addBlob(258);
oscb.addBlob(2.3f);
oscb.addBlob("yahoo!");
omess.addBlob(oscb);
oscm.sendOSCmsg(omess);
```

Et enfin, nous avons aussi implémenté la possibilité d'envoyer une OSCbundle, qui consiste en un entête #bundle de 8 bytes suivit d'un OSC Time Tag (par défaut 0) de 8 bytes aussi. Ensuite vient la taille du bundle, un int32 en 4 bytes, et finalement le bundle en lui-même, qui peut contenir divers messages OSC.

Pour créer une OSC bundle, il faut tout d'abord créer un objet OSCbundle, créer des OSCmsg, d'ajouter les OSCmsg dans l'OSCbundle et d'envoyer l'OSCbundle.

```
OSCbundle obu = new OSCbundle();
OSCmsg omess = new OSCmsg("/test bundel 1");
omess.addValue(357);
omess.addValue(25.75f);
obu.addOSCmsg(omess);

OSCmsg omess2 = new OSCmsg("/test bundel 2");
omess2.addValue(85);
omess2.addValue(0.3f);
obu.addOSCmsg(omess2);
oscm.sendOSCBundle(obu);
```

3. Conclusion

Nous sommes arrivés en fin de compte à créer une librairie pour CSharp qui permet d'envoyer relativement facilement des messages OSC. Par manque de moyens, les blob et les bundles n'ont pas pu être validés correctement, mais la fonction de base, qui est d'envoyer des int, float, et string marche à merveille, ce qui est en soi, déjà pas mal.