Traduction française par Lynx et Maraudeur

Introduction

Gauge Composer est un outil qui permet de fabriquer des panneaux d'instruments graphiques, incluant des instruments analogiques, qui peuvent fonctionner dans une fenêtre ou sur un ordinateur indépendant.

Le programme acquiert des informations (data) d'un serveur IOCP, (IOCPServer pour FSimulator, XPLUIPC pour X-Plane, LO-IOCP pour Lock-ON, SIOC,...).
Ainsi ces instruments peuvent être utilisés sur n'importe quel ordinateur connecté à distance du serveur IOCP.

Avec cet outil on peut fabriquer des instruments facilement, car c'est un outil graphique, et tous les ajustements peuvent être réalisés à l'aide de la souris.
Le programme est basé dans plusieurs scripts par instrument. Ces scripts sont définis par des commandes basiques.

Le système peut acquérir des informations, **data**, d'un second serveur IOCP, il peut donc être connecté à 2 serveurs en même temps.

Scripts

Les Eléments (images et textes) d'un instrument peuvent changer à cause de modifications dans les variables IOCP. Le moyen de dire à **GaugeComposer** ce qu'il doit faire est défini via un script, un petit programme qui est lancé chaque fois que la valeur associée à la variable change.

Il n'y a pas besoin d'avoir des talents de programmeur pour travailler sur des scripts et sur **GaugeComposer**. La seule chose réellement importante est de comprendre les concepts basiques : comment le script est lancé, et l'ordre d'exécution des lignes de script.

Chaque script doit commencer par la ligne : **link offset oo server {1|2}**, ainsi **GaugeComposer** saura à quel serveur est relié l'instrument et sa valeur.

Avant de décrire le langage qui compose le script, nous verrons tout d'abord une petite chose.

GaugeComposer lancera le script pour tous les éléments reliés au même serveur à chaque fois qu'une des variables change. Si nous avons quatre scripts, deux reliés à des variables du serveur #1 et deux reliés à des variables du serveur #2, les deux scripts reliés à des variables du serveur #1 seront lancés quand **n'importe laquelle** des variables reliées dans le serveur #1 change.

Reserved words

GCScript comprend les "mots réservés" suivants : link, offset, server, value, data, if, else, endif, hide, show, rotate, print, print_frq, round, moveh, movev.

Juste seize mots pour gérer complètement n'importe quel instrument.

Avant d'expliquer chacun d'entre eux, une notification importante s'impose :

> **{A|B}** Indique que l'on doit trouver un A ou B (A ou B sont des options possibles).
> **{1..9}**Indique que l'on doit trouver une des valeurs possibles entre 1 et 9.
> **[]** Indique l'optionalité, Existe ou n'existe pas.

Liste des reserved words :

.-data
.-endif
.-else
.-if
.-link offset server
.-offset → Valeur Distante ou VD
.-server
.-hide
.-int
.-moveh
.-movev
.-print
.-print_frq
.-rotate
.-round
.-show
.-value

data

data est utilisé pour transférer des informations à un groupe de mots. Ces groupes de mots sont utilisés pour faire tourner ou bouger des éléments (visuels ou graphiques). Il peut aussi être utilisé et modifié pour des questions ou des assignations pour value.

data peut être modifié comme suit :

data = expresion	la valeur d'expression est assignée à data.
data + = expresion	la valeur d'expression est ajoutée à data.
data - = expresion	la valeur d'expression est soustraite à data.
data * = expresion	la valeur de data est multipliée par la valeur expression.
data / = expresion	la valeur de data est divisée par la valeur d'expression.

Nous pouvons donc obtenir un script qui commence ainsi :
link offset 63 server 1 (relier la VD n°63 du serveur 1)
data = value

Avec 'data=value', la valeur stockée dans value (par la phrase de lien) est stockée dans data. Maintenant nous pouvons changer cette valeur, par exemple data *= 0.36, et appeler l'instruction rotate pour faire pivoter les éléments liés à ce script du nombre de degrés indiqués par data.

link offset oo server {1|2}

link doit être la première phrase de tout script, et définit le serveur et la valeur distante qui vont définir le comportement de l'instrument.

oo se définit comme le n° de valeur distante IOCP à laquelle nous voulons nous connecter. Pour le serveur IOCP ce sera dans le champ de 0 à 1196 (voir la documentation du serveur IOCP).

link cette phrase peut aussi se retrouver dans d'autres endroits du script, si nous voulons connaître la valeur d'une autre variable distante dans le serveur ou dans un autre serveur.

Dans ce cas on peut l'écrire de cette façon :
link [offset oo][server {1|2}]

Cela signifie que la valeur distante ou le serveur peuvent n’être pas spécifiés.
Le résultat de cette phrase du script est une mise à jour de la valeur.

moveh

L’instruction **moveh** décale l'élément (graphique ou texte) horizontalement vers la droite si data>0, ou la gauche si data<0, de la valeur en pixel indiquée par data.

Si nous voulons dessiner un indicateur de volets avec une aiguille qui bouge horizontalement, nous savons que les volets de l'aile gauche sont représentés par la valeur distante 327. Cette VD peut avoir des valeurs comprises entre 0 et 16K (1K=1024) avec 16K signifiant « volets max ».

Tout cela signifie que si je veux bouger l'aiguille de 190 pixels quand les flaps sont au max, j'aurai besoin de calculer un facteur « f » f= 190/16*1024 (=0,0115966796875). Quand nous multiplions ce facteur « f » par la VD, nous obtenons une valeur entre 0 et 190.
Si les volets sont remontés position 0, 0*f=0, il n'y a pas de mouvement, si les flaps sont sortis au max (16K), 16K*f=190.

Exemple :
link offset 327 server 1
data = value
data *= 0.0115966796875

movev

L’instruction **movev** décale l'élément (graphique ou texte) verticalement vers le bas si data>0, ou le haut si data<0 de la valeur en pixel indiquée par data.

Si nous voulons dessiner un indicateur de volets avec une aiguille qui bouge verticalement, nous savons que les volets de l'aile gauche sont représentés par la valeur distante (VD) 327. Cette VD peut avoir des valeurs comprises entre 0 et 16K (1K=1024) avec 16K signifiant « volets max ».

Tout cela signifie que, si je veux bouger l'aiguille de 190 pixels quand les flaps sont au max, j'aurai besoin de calculer un facteur « f » f= 190/16*1024 (=0,0115966796875). Quand nous multiplions ce facteur « f » par la VD, nous obtenons une valeur entre 0 et 190.
Si les volets sont remontés position 0, 0*f=0, il n'y a pas de mouvement mais si les flaps sont sortis au max (16K), 16K*f=190.

Exemple :
link offset 327 server 1
data = value
data *= 0.0115966796875
movev

rotate

L’instruction **rotate** tourne l'élément (graphique ou texte) dans le sens des aiguilles d'une montre, du nombre de degrés indiqués par la variable data.

rotate peut être utilisé pour faire tourner des aiguilles dans des instruments analogiques. Si data peut prendre une valeur entre 0 et X, nous devons diviser la valeur maximale (en degrés) par X pour en trouver le facteur.

Si nous voulons dessiner la position des flaps avec une aiguille qui tourne, nous savons que la valeur des flaps de l'aile gauche est représentée par la VD 327. Cette VD peut avoir des valeurs comprises entre 0 et 16K (1K=1024) avec 16K signifiant « volets max ».
Tout cela signifie que si je veux bouger l'aiguille de 190 degrés quand les flaps sont au max, j'aurai besoin de calculer un facteur « f » f= 190/16*1024 (=0,0115966796875).
Quand nous multiplions ce facteur « f » par la VD, nous obtenons une valeur entre 0 et 190.
Si les volets sont remontés position 0, 0*f=0, il n'y a pas de mouvement mais si les flaps sont sortis au max (16K), 16K*f=190.

Exemple :
link offset 327 server 1
data = value
data *= 0.0115966796875
rotate

value

value placé après **link**, stocke la valeur de la VD du serveur recherché, nous pouvons ainsi la connaître et l'utiliser.
Cette valeur peut être changée et utilisée pour des questions ou pour des assignements de valeur data.

value peut être modifiée par les commandes suivantes:

value = = expression	la valeur 'expression' est stockée dans 'value'.
value + = expression	la valeur 'expression' est ajoutée à 'value'.
value - = expression	la valeur 'expression' est soustraite à 'value'.
value * = expression	'value' est multipliée par la valeur d' 'expression'.
value / = expression	'value' est divisée par la valeur d'expression'.

Nous pouvons donc avoir un script débutant par:
link offset 63 server 1
value /= 100

Comme nous sommes reliés à une VD du serveur, nous avons une variable à nom fixe appelée 'value'. Avec 'value' nous connaissons la VD. 'value' représente la valeur de la VD juste au moment ou le script est lancé, mais nous pouvons changer cette valeur depuis le script ou demander une autre valeur de VD.