

DIDACTICIEL

Écrire son premier script

par Jean-Paul Verpeaux

Ce didacticiel utilise **MyrScript**, le langage permettant de piloter les fonctionnalités d'Harmony Assistant et inclus par défaut dans celui-ci.

A la demande de l'un d'entre vous, voici un petit didacticiel pour vous aider à écrire votre premier script.

Je dois d'abord vous préciser que je ne suis pas un programmeur professionnel, mais que mon métier d'électronicien m'a amené à écrire de nombreux logiciels dans différents langages. Pour Harmony-assistant j'ai déjà écrit une vingtaine de scripts, mais je ne les diffuse pas tous car certains ne correspondent qu'à des besoins strictement personnels.

Tout d'abord permettez-moi de vous faire quelques suggestions ou vous donner quelques conseils. Il s'agit en fait de méthodes pratiques que j'applique et qui s'avèrent à l'usage fort utiles.

1er conseil. Le manuel du langage est très volumineux et en anglais. Alors, achetez un carnet répertoire (avec des onglets pour chaque lettre de l'alphabet) et recopiez y les instructions que vous êtes conduit à utiliser souvent, avec une brève description de son usage et de sa syntaxe.

2nd conseil. Quand vous écrirez une boucle d'instructions (par exemple **While ...**) ou une condition (**If... Then**) qui se terminent toujours par le mot « **end** », prenez l'habitude d'indenter (décaler vers la droite) les lignes incluses dans la boucle ou le test conditionnel et d'écrire non pas « end » mais « **end --while** » ou « **end -- if** » selon le cas. Cela évitera de se perdre dans un méandre de boucles imbriquées.

Voici maintenant comme exemple, comment écrire pas à pas un script qui coloriera en rouge les notes dont la vitesse est supérieure à 64, seulement dans les portées sélectionnées.

Ouvrez l'éditeur de script en faisant « Nouveau script ».

Écrivez l'entête de votre script (copiez-collez celui ci-dessous), l'absence d'entête n'empêche pas le script de fonctionner, mais personne ne saura à quoi il sert ni où le trouver.

Sauvegardez ce scripts sous un nom approprié (**color_64_velocity** par exemple) dans le répertoire « **Myriad documents/settings/scripts** » ou un sous-répertoire de votre choix dans ce répertoire.

```
----- COMMON SECTION -----
--CREATOR: Tartempion
--MENU_LOCATION: Edit
--DIFFUSION_MODE: 1
--VERSION: 1.0.0
----- ENGLISH SECTION -----
--DATE: October, 2007
--NAME_IN_MENU: Coloured Velocity
--NAME: Coloured Velocity
--ABSTRACT: Coloured >64 Velocity
--INFO: Change notes color if their velocity > 64.

----- FRENCH SECTION -----
--DATE-FR: Octobre 2007
--NAME_IN_MENU-FR: Colorier Vélacité
--NAME-FR: Colorier Vélacité
--ABSTRACT-FR: Colorier Vélacité > 64
--INFO: Change la couleur des notes dont la vélocité est > 64.
```

Tout de suite après, collez :

```
Include "MSDefine.mys"  
Include "MSLibrary.mys"
```

Cela consiste à inclure dans le script, sans le surcharger visuellement, un ensemble de mots, variables ou constantes prédéfinis. Vous n'en aurez peut-être pas besoin pour ce script, mais c'est une habitude à prendre, Au besoin, vous supprimerez ces lignes ultérieurement.

Voici maintenant le corps du script, dans sa version minimaliste. Expliquons-le ligne par ligne.

```
score=FrontScore()  
score.Preserve()  
-- For all selected staves  
staff=score.FirstSelectedStaff  
while staff and staff.IsSelected==true do  
    symbol=staff.FirstSelectedSymbol  
    while symbol do  
        if symbol.IsNote==true then  
            velocity=symbol.Velocity  
            -- modifier la couleur des notes  
            if velocity >64 then  
                symbol.RGBColor="800000"  
            end -- if velocity >64  
        end -- if symbol.IsNote  
        symbol=symbol.Next  
        -- Tester user break  
        Application.UserBreak()  
    end -- while symbol do  
    staff=staff.Next  
end -- while
```

Explications :

score=FrontScore()	Cela veut dire : travaillons avec la partition ouverte (celle qui est en premier plan), score est une variable dont j'ai choisi le nom.
score.Preserve()	Nous sauvegardons la partition avant de la modifier. Une ligne dans le menu Édition permettra de la restaurer en cas de manipulation malencontreuse.
-- For all selected staves	Commentaire indiquant que nous allons traiter toutes les portées sélectionnées. Les commentaires commencent toujours par deux tirets.
staff=score.FirstSelectedStaff	staff est une variable dont j'ai moi-même choisi le nom. On lui affecte la valeur score.FirstSelectedStaff c'est à dire que staff représente la première portée sélectionnée de la partition score .
while staff and staff.IsSelected==true do	Ceci indique le début d'une boucle signifiant « tant que staff existe et que staff est sélectionné exécuter les instructions qui vont suivre, jusqu'à ce qu'on rencontre l'instruction end ». Cette boucle explorera les portées sélectionnées une par une.
symbol=staff.FirstSelectedSymbol	symbol est ma troisième variable. On l'initialise avec staff.FirstSelectedSymbol ce qui veut dire que l'on commencera à analyser la partition score en lisant le premier symbol de staff (la première portée sélectionnée).
while symbol do	Dans la boucle while staff... on insère une nouvelle boucle signifiant « tant que symbol existe... faire... ». Les deux boucles sont imbriquées. Grâce à cette boucle, les symboles de la portée étudiée seront analysés un par un.
if symbol.IsNote==true then	On teste si chaque symbol analysé par la boucle est bien une note. symbol pourrait être autre chose qu'une note, un silence par exemple. Dans ce cas symbol ne pourrait pas avoir de vitesse.
velocity=symbol.Velocity	Si symbol est bien une note, on affecte à la variable velocity la valeur de la vitesse de la note analysée.
-- modifier la couleur des notes	Commentaire
if velocity >64 then	On teste la variable velocity pour savoir si la note analysée a une vitesse supérieure à 64.
symbol.RGBColor="800000"	RGBColor est une des caractéristiques de l'objet symbol . En affectant la valeur 800000 à symbol.RGBColor on colorie en rouge le symbole analysé. 800000 signifie 80 (valeur hexadécimale) pour la composante rouge de la couleur, 00 pour le ton vert et 00 pour le bleu.
end -- if velocity >64	end indique que la dernière boucle ouverte ou le dernier test if then entamés prennent fin ici. Cette fois, il s'agit de la fin du test de la vitesse. -- if velocity >64 est un commentaire qui permet repérer à quelle ligne associer le mot end .
end -- if symbol.IsNote	Fin du test if... then
symbol=symbol.Next	On passe au symbole suivant. Cette ligne équivaut à symbol=symbol +1 que l'on pourrait rencontrer dans un autre langage.
-- Tester user break	Commentaire annonçant que l'on va regarder si l'utilisateur veut interrompre le script avant sa fin.
Application.UserBreak()	Commande pour interrompre le script. En appuyant sur la touche Esc ou en faisant Commande+Maj+point pour arrêter l'exécution d'un script.
end -- while symbol do	Fin de la boucle while explorant les symboles de portées sélectionnées.
staff=staff.Next	Passer à la portée suivante.
end -- while	Fin de la boucle principale (boucle extérieure). A ce stade le script est terminé.