DIDACTICIEL

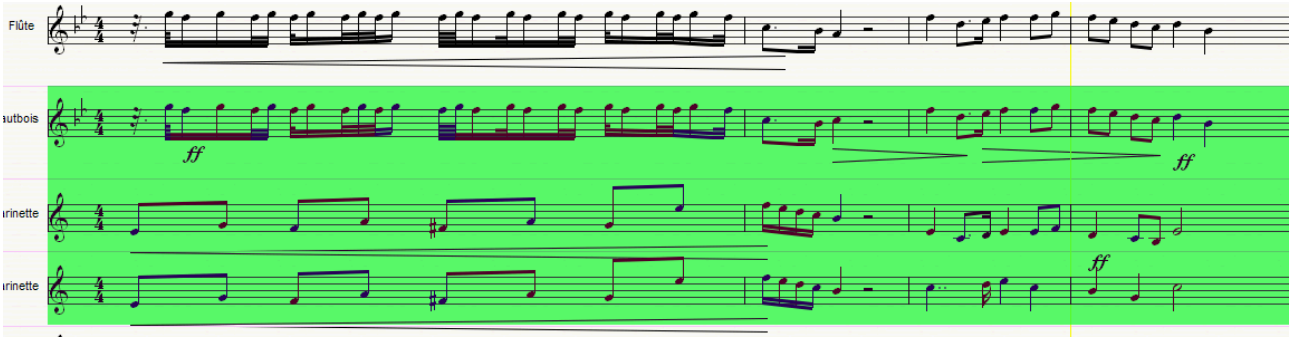
Écrire son second script

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.

Voici un nouveau script. Pas très long, mais déjà un peu plus compliqué que le premier. Il va nous permettre de découvrir de nouvelles possibilités du langage script.
Pour faciliter l'écriture de la leçon, le code du script sera écrit en bleu, les commentaires en vert. Dans son éditeur de texte, on observerait plusieurs couleurs différentes pour distinguer les commentaires des mots clefs ou des instructions mal orthographiées.
Le script ci-dessous peut être copié-collé dans l'éditeur de script (après avoir fait « nouveau script ») ou installé dans votre ordinateur en double-cliquant sur le fichier « human_feeling.msa ».

Ce script modifie aléatoirement le début des notes, en avance ou en retard puis change légèrement, en sens inverse la durée d'appui des notes. Enfin, les notes modifiées seront coloriées en rouge ou en bleu selon qu'elles auront été anticipées ou retardées. Le script n'agit que sur les portées préalablement sélectionnées.
Les notes jouées trop tard seront donc moins longues que prévu (temps d'appui réduit). Celles jouées en avance seront raccourcies. Voici un exemple visuel du résultat.




```
---------- COMMON SECTION ------
--CREATOR: Tartempion
--MENU_LOCATION: Edit>Actions
--DIFFUSION_MODE: 1
--VERSION: 1.0.0
------ ENGLISH SECTION -------
--DATE: October, 2007
--NAME_IN_MENU: Human feeling
--NAME: Human feeling
--ABSTRACT: Human feeling
--INFO: Change notes delays to make human feeling and set notes color  according to delay
------ FRENCH SECTION -------
--DATE: Octobre, 2007
--NAME_IN_MENU: Human feeling
--NAME-FR: Human feeling
--ABSTRACT-FR: Human feeling
--INFO-FR: Création d'un effet Human feeling en jouant aléatoirement sur le positionnement exact des notes.
--INFO-FR: Ce script tient compte de la durée de chaque note et génère un retard proportionnel.
--INFO-FR: Le retard est compensé par une modification contradictoire du temps d'appui.
--INFO-FR: Les notes sont coloriées en rouge ou en bleu selon le sens de leur retard-NAME-FR: Human feeling
Include "MSDefine.mys"
Include "MSLibrary.mys"
effect=0.05 -- retard en % de la durée de la note, exemple 0.05 pour 5%
sens = 0 --sens du retard
ratio=0 -- pourcentage du retard par rapport à la durée
score=FrontScore()
if score ~= nil  then
        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
				duree=symbol.Duration
				duree= duree /157.5 -- conversion en 1/256 noire
				retard=random(0,duree)
				retard=retard * effect
				sens=random(0,2)
				if sens <1 then sens=-1 else sens=1 end
				retard=retard * sens
				ratio= retard/duree
				-- modifier les notes
				symbol.Delay=retard -- nouveau retard
				-- modifier l'appui dans le meme pourcentage
				-- l'appui est modifié dans le sens inverse du retard
				print("retard : "..retard..."	ratio : "..ratio)
				symbol.PressureTime=symbol.PressureTime*( 1-ratio)
				-- changer la couleur
				if retard >0 then symbol.RGBColor="C00000" end
				if retard <0 then symbol.RGBColor="C0" end
			end -- if symbol.IsNote
			symbol=symbol.Next
			-- Tester user break
			Application.UserBreak()
		end -- while symbol do
		staff=staff.Next
	end  -- while staff ...
end  -- if score...
```

Expliquons le code ligne par ligne, si nécessaire.

---------- COMMON SECTION ------ --CREATOR: Tartempion .... --INFO-FR: Création d'un effet Human feeling en jouant aléatoirement sur le positionnement exact des notes. --INFO-FR: Ce script tient compte de la durée de chaque note et génère un retard proportionnel. --INFO-FR: Le retard est compensé par une modification contradictoire du temps d'appui. --INFO-FR: Les notes sont coloriées en rouge ou en bleu selon le sens de leur retard.	Ici je ne décrirai pas à nouveau le rôle de l'entête mais apporterai juste quelques précisions. Par exemple : FRENCH SECTION indique le début d'une série d'informations que pourront lire seulement les utilisateurs de langue française. --MENU_LOCATION: Edit>Actions fait apparaître le script dans le menu d'Harmony-Assistant **Edition\Actions.** Ces informations se présentent comme des commentaires (couleur verte dans l'éditeur), mais chaque premier mot en majuscule est une balise qui active une fonction précise. C'est la raison pour laquelle il faut bien respecter l'orthographe de ces mots et les absences d'espace pour les séparer des deux tirets de gauche et du deux-points de droite. Pour décrire le script, on peut placer autant de lignes INFO qu'on le souhaite.
Include "MSDefine.mys" Include "MSLibrary.mys"	Ces deux lignes peuvent parfois être supprimées mais c'est une bonne habitude de les mettre si on compte utiliser dans le script des constantes prédéfinies comme par exemple les mots **true** et f**alse**.
effect=0.05 -- retard en % de la durée de la note, exemple 0.05 pour 5%	**effect** est une variable. Je rappelle ici que c'est la personne qui écrit le script qui choisit le nom donné à ses variables. Attention, les noms sont sensibles à la casse (majuscule/minuscule) et doivent commencer par une lettre. **effect** va servir à définir l'intensité de l'effet **human feeling** recherché. Quand vous aurez compris le fonctionnement du script, vous pourrez modifier cette valeur pour vous amuser.
sens = 0 --sens du retard	La variable **sens** servira à choisir entre un retard positif ou un retard négatif (c'est à dire une note jouée en avance). s**ens=0** initialise la variable à 0, mais en fait cette instruction n'est pas utile, on peut supprimer cette ligne.
ratio=0 -- pourcentage du retard par rapport à la durée	Autre variable. **ratio** sert à calculer le rapport entre le retard et la durée de la note (valeur exprimée en %). Idem, cette variable sera calculée plus bas, son initialisation est superflue.

score=FrontScore()	**score** = la partition sur laquelle on va travailler.
if score ~= nil then	On s'assure avant tout que l'on a bien une partition d'ouverte (score ~= nil.).
score.Preserve()	Nous préservons la partition en mémoire pour pouvoir au besoin annuler les actions effectuées par le script.
-- For all selected staves	Commentaire expliquant que nous allons traiter toutes les portées sélectionnées.
staff=score.FirstSelectedStaff	On commence par la première portée sélectionnée. **staff** est une variable qui représentera à tour de rôle toutes les portées sélectionnées en commençant par celle placée le plus en haut sur la page.
while staff and staff.IsSelected==true do	On teste s'il y a bien une portée et qu'elle est sélectionnée et on commence une boucle (suite d'instructions) de type **while ... do.** Cette boucle (nommons-la mentalement « grande boucle ») sert à analyser les portées sélectionnées les unes après les autres. Le mot clef **and** exige que les deux conditions soient réunies.
symbol=staff.FirstSelectedSymbol	Nous affectons à la variable **symbol** l'identité du premier symbole rencontré sur la portée en cours d'analyse. Si vous préférez, nous initialisons ici la variable **symbol** avec le symbole le plus à gauche de la portée.
while symbol do	On réalise une seconde boucle d'instruction (« petite boucle »), imbriquée dans la première, Elle sert à analyser un par un les symboles contenus dans la portée représentée par la variable **staff**.
if symbol.IsNote==true then	Si **symbol** est une note, le script effectuera les instructions suivantes, sinon il les sautera jusqu'au prochain code **end**.
duree=symbol.Duration	La variable **duree** (sans accent) prend ici la valeur de la note représentée par la variable **symbol**. Par exemple 40320 pour une noire, 20160 pour une croche ....
duree= duree /157.5 -- conversion en 1/256 noire	Les valeurs très élevées (40320 ...) sont dues à la très haute résolution interne du logiciel. En divisant **duree** par 157,5 on change d'unité. La nouvelle unité de durée devient le 1/256 de noire. On se rapproche des valeurs employées par certains séquenceurs midi.
retard=random(0,duree)	Pour avoir un retard aléatoire, on utilise l'instruction **random**. Grâce aux 2 paramètres entre parenthèse, cette instruction nous retournera un nombre compris entre 0 et la valeur actuelle de **duree**. Donc un nombre aléatoire qui pourra être plus grand pour une noire que pour une croche.
retard=retard * effect	Comme on ne veut pas trop déplacer la position des notes, on applique à chaque valeur de retard un même coefficient. Ici 5%. Pour ce, on multiplie **retard** par 0,05 (la valeur que nous avons donné à **effect**).
sens=random(0,2)	Pour le besoin nous créons une nouvelle variable : **sens**. Elle va servir à décider si l'effet « human feeling » jouera la note en avance ou en retard. Avec l'instruction **random**, **sens** pourra prendre au hasard les valeurs 0, ou 1 ou 2.
if sens <1 then sens=-1 else sens=1 end	Selon la valeur de **sens**, on décidera si on place la note en avance (pour cela il faudra donner à **retard** une valeur négative) ou en retard. Remarquez qu'on peut placer plusieurs instructions sur la même ligne. Ici trois instruction se suivent : **if, else** et **end**. Quand ces instructions auront été exécutées, sens vaudra +1 ou -1 exclusivement;
retard=retard * sens	Comme sens vaut maintenant +1 ou -1, retard sera positif ou négatif.
ratio= retard/duree	**ratio** est une nouvelle variable dont nous allons avoir besoin pour modifier le temps d'appui d'une note. Elle exprime le rapport entre le retard et la durée de la note.
-- modifier les notes	Pour l'instant nous n'avons fait que des calculs. Passons au concret, c'est à dire à la modification des notes.
symbol.Delay=retard -- nouveau retard	**Delay** est une des propriété de l'objet **symbol**. Par défaut **Delay** vaut 0 (quand on pose une note sur une portée avec la souris, son retard est toujours 0). Maintenant cette valeur va changer et sera égale à retard.
-- modifier l'appui dans le même pourcentage -- l'appui est modifié dans le sens inverse du retard	

print("**retard :** "..retard.." **ratio** : "..ratio)	**print** est une commande qui affiche dans une petite fenêtre le texte qu'on lui demande entre parenthèses. Cela sert à debugger le programme. J'ai laissé cette ligne pour vous le montrer. Ici on affiche 4 choses : le texte « **retard :** » (placé entre guillemets), la valeur de la variable **retard** (sans guillemets car c'est une variable), le texte « **ratio :** » et la valeur de la variable **ratio**. Notez que ces quatre choses sont réunies en une seule chaîne grâce à l'outil de concaténation représenté par deux points (que j'ai coloriés en rouge et exagérément grossis pour ne pas qu'ils échappent à votre vigilance).
symbol.PressureTime=symbol.PressureTime*(1-ratio)	Ici nous modifions le temps d'appui d'une note. Si la note est jouée en avance son temps sera rallongé, si elle est retardée, il sera raccourci.
-- changer la couleur	
if retard >0 then symbol.RGBColor="C00000" end	Si la note est retardée (retard positif) elle sera coloriée en rouge. Notez encore la présence des deux instructions complémentaires **if** et **then** sur la même ligne, C'est pratique quand il n'y a qu'une seule instruction à exécuter si le test est positif.
if retard <0 then symbol.RGBColor="C0" end	Si le retard est négatif (note jouée en avance), la note sera colorée en bleu. « C0 » est équivalent à « 0000C0 ».
end -- if symbol.IsNote	Fin du test vérifiant que **symbol** est une note.
symbol=symbol.Next	Nous passons au symbole suivant.
-- Tester user break	
Application.UserBreak()	Petite sécurité pour pouvoir interrompre le script en cas de besoin.
end -- while symbol do	Fin de la petite boucle.
staff=staff.Next	On enchaîne sur la portée sélectionnée suivante.
end -- while staff	Fin de la grande boucle. On sort ici quand on a exploré toutes les portées sélectionnées.
End -- if score ...	Fin du script. On aboutit directement à cette ligne si on a oublié d'ouvrir une partition et sélectionner une ou plusieurs portées avant de lancer le script.