

DIDACTICIEL

Écrire son troisième 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 troisième petit script qui vous permettra de découvrir de nouvelles possibilités du langage tout en pouvant vous rendre service. Son but : harmoniser l'aspect d'une partition en uniformisant la taille des portées et en égalisant les intervalles les séparant.

Ce script est toutefois un didacticiel, il reste très simple et n'envisage pas toutes les variantes possibles des partitions. Il se peut qu'il ne fonctionne pas correctement si la partition comporte des portées fusionnées ou multivoix.

 <p>The image shows a musical score with ten staves. The staves are of varying heights and are separated by irregular vertical intervals. A red letter 'A' is placed at the beginning of the first staff. The score includes various musical notations such as notes, rests, and dynamic markings like 'f' and 'ff'. A tempo marking 'J=120' is visible at the bottom of the score.</p>	 <p>The image shows the same musical score as the left panel, but the staves are now uniform in height and separated by equal vertical intervals. The red letter 'A' and all other musical notations remain the same.</p>
La partition avant le script	La partition après l'application du script

Comme pour les didacticiels précédents, voici tout d'abord le script.

Par commodité, les couleurs ont été réduites à deux : vert pour les commentaires, bleu pour le code.

```
----- COMMON SECTION -----  
--CREATOR: VERPEAUX J-Paul  
--DIFFUSION_MODE: 1  
--MENU_LOCATION: Staves  
--VERSION: 1.1.3
```

```
----- ENGLISH SECTION -----  
--DATE: Feb 8th, 2006  
--NAME_IN_MENU: Adjust staves  
--NAME: Adjust Staves  
--ABSTRACT: Set staves interval and size to same value  
--INFO: Make the score more balanced by setting size and intervals between staves.
```

```

----- FRENCH SECTION -----
--DATE-FR: 08 Février 2006
--NAME_IN_MENU-FR: Réajustement des portées
--NAME-FR: Réajustement des portées
--ABSTRACT-FR: Mettre aux mêmes valeurs les tailles et intervalles entre les portées.
--INFO-FR: Rendre la partition plus équilibrée en réajustant la taille et l'intervalle des portées.

myScore=FrontScore()
if(myScore ~= nil) then
    myScore.Preserve("ajustement des portées")

    -- Variables
    staff_count = 0 -- compte les portées
    mean_height = 0 -- hauteur moyenne
    height_sum = 0 -- total des hauteurs
    greatest_base = 0 -- plus grand intervalle
    mean_baseline = 0 -- moyenne des baseline
    baseline_sum = 0 -- total des baseline

    -- nombre de portées
    print (myScore.NumberOfStaves.." portées dans cette partition")

    -- Boucle principale de lecture
    myStaff=myScore.FirstStaff
    while myStaff~=nil do

        staff_count=staff_count + 1
        print (myStaff.Title)
        print ( "Hauteur en dpi = " .. myStaff.Height )
        height_sum=height_sum +myStaff.Height

        print ("BaseLine =" .. myStaff.BaseLine)
        baseline_sum=baseline_sum +myStaff.BaseLine

        -- plus grand intervalle
        if myStaff.BaseLine> greatest_base then greatest_base=myStaff.BaseLine end

        myStaff=myStaff.Next
    end -- of loop while

    -- Calcule des moyennes
    mean_height=height_sum/staff_count
    mean_baseline=baseline_sum/staff_count

    --print(height_sum)
    --print(baseline_sum)
    --print (mean_height)
    --print ("Moyenne deBaseLine = "..mean_baseline)
    --print ("Plus grande BaseLine ="..greatest_base)

    -- Boucle principale d'écriture
    myStaff=myScore.FirstStaff
    while myStaff~=nil do
        myStaff.Height=mean_height
        myStaff.BaseLine=greatest_base
        myStaff=myStaff.Next
    end -- of loop while
else
    Alert("Si vous n'avez pas de partition ouverte, le script ne sert à rien")
end -- of script

```

Voici maintenant, à l'exception de l'entête dont nous avons déjà expliqué le rôle des balises, le script commenté en détail.

Les lignes en rouge sont des lignes que vous pouvez supprimer car elles ne sont pas utiles au fonctionnement du script. Elles permettent d'afficher des informations ou calculs intermédiaires pour vérifier ou déboguer le script. Normalement on les efface quand le script est terminé et qu'il tourne correctement.

Le script comporte deux boucles, mais elles ne sont pas imbriquées. La première sert à calculer les nouvelles caractéristiques des portées, la seconde sert à les appliquer.

<code>myScore=FrontScore()</code>	Utilisation de la variable myScore pour représenter la partition dont on veut retoucher la disposition des portées,
<code>if(myScore ~= nil) then</code>	Vérifier que cette partition a bien été ouverte,
<code>myScore.Preserve("ajustement des portées")</code>	Sauver en mémoire la partition pour pouvoir annuler au besoin les effets du script. Remarquez une variante de l'utilisation de Preserve() : le texte entre parenthèses affichera dans le menu Edition : « annuler l'action du script ajustement des portées ».
<code>-- Variables</code>	Nous allons initialiser ci-dessous plusieurs variables dont nous aurons besoin.
<code>staff_count = 0 -- compte les portées</code>	Il faudra savoir combien de portées compte la partition.
<code>mean_height = 0 -- hauteur moyenne</code>	Variable qui servira à recueillir le résultat de notre calcul de hauteur moyenne.
<code>height_sum = 0 -- total des hauteurs</code>	Variable nécessaire pour faire le cumul des hauteurs.
<code>greatest_base = 0 -- plus grand intervalle</code>	Variable qui mémorisera le plus grand intervalle (baseline) séparant deux portées.
<code>mean_baseline = 0 -- moyenne des baseline</code>	Variable pour calculer et fournir l'écartement moyen.
<code>baseline_sum = 0 -- total des baseline</code>	Variable pour calculer la somme des écartements.
<code>-- nombre de portées</code>	
<code>print (myScore.NumberOfStaves.." portées dans cette partition")</code>	Cette ligne a été conservée pour vous montrer un exemple d'utilisation de la propriété NumberOfStaves .
<code>-- Boucle principale de lecture</code>	
<code>myStaff=myScore.FirstStaff</code>	myStaff est une variable. On l'initialise en l'associant à la première portée de la partition.
<code>while myStaff~=nil do</code>	Début d'une boucle while ... do dont on ne sortira que quand myStaff ne désignera plus rien (nil) car on aura parcouru toutes les portées de la partition.
<code>staff_count=staff_count + 1</code>	staff_count est la variable qui compte les portées.
<code>print (myStaff.Title)</code>	Ligne laissée pour vous montrer comment lire le nom d'une portée.
<code>print ("Hauteur en dpi = " .. myStaff.Height)</code>	Information bonne à connaître.
<code>height_sum=height_sum +myStaff.Height</code>	On additionne les hauteurs pour pouvoir calculer la hauteur moyenne des portées.
<code>print ("BaseLine =" .. myStaff.BaseLine)</code>	Encore un affichage facultatif que vous pourrez supprimer.
<code>baseline_sum=baseline_sum +myStaff.BaseLine</code>	Nous faisons ici la somme des espaces (intervalles) occupés par chaque portées,
<code>-- plus grand intervalle</code>	
<code>if myStaff.BaseLine> greatest_base then greatest_base=myStaff.BaseLine end</code>	Si on rencontre un intervalle encore plus grand que greatest_base alors greatest_base prend comme valeur cet intervalle.
<code>myStaff=myStaff.Next</code>	On passe à la portée suivante.
<code>end -- of loop while</code>	Fin de la boucle while do .
<code>-- Calcul des moyennes</code>	
<code>mean_height=height_sum/staff_count</code>	La hauteur moyenne d'une portée est la somme des hauteurs divisée par le nombre de portées. La barre oblique (/) est le symbole de division.

<code>mean_baseline=baseline_sum/staff_count</code>	De même, l'écartement moyen des portées est la somme des intervalles divisée par le nombre de portées.
<code>--print(height_sum) --print(baseline_sum) --print(mean_height) --print("Moyenne deBaseLine = "..mean_baseline) --print("Plus grande BaseLine ="..greatest_base)</code>	Quelques instructions facultatives pour contrôler le bon fonctionnement du script. Elles ont été écrites comme des remarques. Il suffit d'enlever les deux tirets du début de ligne pour les activer.
<code>-- Boucle principale d'écriture</code>	Nous avons fait tous les calculs théoriques. Maintenant nous allons appliquer les résultats trouvés à la partition, portée après portée, via une nouvelle boucle while .
<code>myStaff=myScore.FirstStaff</code>	Une variable qui ne sert plus peut être réutilisée. On se sert donc une seconde fois la variable myStaff , <u>mais après l'avoir de nouveau réinitialisée.</u>
<code>while myStaff~=nil do</code>	Début de la seconde boucle.
<code>myStaff.Height=mean_height</code>	Modification de la hauteur de chaque portée. Chaque portée aura comme hauteur la hauteur moyenne que nous avons calculée.
<code>myStaff.BaseLine=greatest_base</code>	Modification de l'espacement des portées.
<code>myStaff=myStaff.Next</code>	On enchaîne avec la portée suivante.
<code>end -- of loop while</code>	Et c'est fini, la partition est mise en forme.
<code>else</code>	Ça, c'est au cas où l'on aurait oublié d'ouvrir une partition.
<code>Alert("Si vous n'avez pas de partition ouverte, le script ne sert à rien")</code>	Alert permet d'afficher un message d'avertissement dans une fenêtre que l'utilisateur devra refermer.
<code>end -- of script</code>	Cette fois, c'est vraiment terminé.

Le script est téléchargeable sur le forum, mais vous pouvez aussi copier-coller dans l'éditeur de script le code présenté au tout début de cet article.

J'espère que la leçon n'a pas été trop compliquée. A bientôt pour passer à des scripts de force supérieure.