Staff Animation

(Animation de portée)

Cette application permet de faire des animations musicales (écrites), à des fins pédagogiques, pour les insérer dans des screencasts.

Création d'une animation

Table des matières

Introduction

Une animation est composée de `pas` (step) qui vont être exécutés l'un après l'autre, produisant des choses aussi diverses que :

- L'apparitioin d'une portée ;
- L'écriture d'une note ou d'un accord sur la portée ;
- Le déplacement de notes ;
- L'écriture de textes explicatifs ;

- etc.

Chaque pas (chaque “step”) exécute une action. Ces pas se définissent dans la console située à gauche de l'écran (le bloc noir). Chaque pas se trouve sur une ligne distincte. Donc chaque ligne est un nouveau pas, une nouvelle étape de l'animation.

Bien noter que chaque ligne représente un pas, il est impossible de définir plusieurs actions sur la même ligne (en réalité c'est faux, mais c'est dangereux ;-)).

Ces pas peuvent :

- Construire une portée ;
- Construire une note (et l'afficher) ;
- Construire un accord ;
- Construire une gamme ;
- Écrire un texte associé à un objet (note, portée, barre, etc.) ;
- Déplacer un objet quelconque (des notes, des textes, etc.) ;
- Supprimer une note ;
- Mettre une note en exergue (entourée d'un cercle de couleur) ;
- etc.

Composer le code de l'animation

Écrire un commentaire

Les commentaires s'écrivent avec `#` en début de ligne.

Noter que les commentaires ne peuvent pas se trouver sur un pas proprement, même après le code. Il faut absolument qu'ils soient sur une ligne seule, qui sera passée.

Faire une pause dans l'animation.

On utilise la méthode `WAIT` pour faire une pause, avec en argument le nombre de secondes (qui peut être un flottant, pour des temps très courts) :

```
WAIT(<nombre de secondes>)
```

Par exemple, pour attendre 4 secondes :

```
maNote=NOTE(a3)
WAIT(4)                 // 4 secondes avant de construire l
'accord
monAccord('c3 eb3 g3')
```

Note : C'est un “pas” comme les autres, donc il doit être mis sur une ligne seule comme toute étape.

Se déplacer sur la portée

La commande pour écrire à la suite des dernières notes sur la portée, on utilise la commande :

```
LEFT([<nombre pixels>])
```

Par défaut (sans argument), le déplacement sera de 40px (`Anim.defaut.hoffset`). Sinon, le déplacement sera de la valeur précisée.

Par exemple :

```
LEFT()
// => les notes suivantes s'écriront 40px plus à gauche

LEFT(100)
// => les notes suivantes s'écriront 100px plus à gauche
```

Écrire un texte général

Pour écrire un texte en rapport avec l'animation (en haut à gauche), on utilise la méthode `WRITE` :

WRITE()

Par exemple :

WRITE("Ce qu'il faut remarquer à ce moment-là.")

Noter que les objets tels que les notes ont également leur propre méthode textuelle, qui permet d'aligner le textes directement à ces notes, accords, etc.

Nettoyer l'animation

“Nettoyer l'animation” signifie supprimer tous les éléments affiché. On peut ou non demander que les portées restent en place, though.

Le pas à utiliser est :

```
CLEAR(<avec les portées>)
```

… `<avec les portées>` est FALSE par défaut, donc il faut ajouter `true` pour effacer aussi les portées :

```
CLEAR(true) // efface aussi les portées
```

Les Notes

Table des matières

Désignation des notes

Les notes doivent être désignées par :

```
<nom note><alteration><octave>
```

- On désigne le `nom des notes` par une seule lettre, anglaise, de "a" (la) à "g" (sol).
- L'altération est soit rien (note naturelle), soit un signe (cf. ci-dessous "b", "d", "x", ou "t").
- Vient ensuite l'octave, un nombre, négatif si nécessaire (*mais pour le moment, on va seulement jusqu'à l'octave 0, les autres ne sont pas gérés*).

Marque des altérations

La valeur `<alteration>` ci-dessus peut être :

- `b` pour bémol
- `t` (comme "ton") pour double bémol
- `d` pour dièse
- `x` pour double-dièse

Constantes notes

De l'octave 0 (qui n'existe pas en français) à l'octave 7 il existe des constantes pour chaque note avec l'altération bémol ("b") et dièze ("d").

Les deux formules suivantes sont donc possible :

no=NOTE(ad5) no=NOTE('ad5')

Note: Attention à ne pas donner à une variable note le nom d'une de ces constantes. Par exemple, si on fait :

```
a5=NOTE(a5)
```

Cela génèrera une erreur de constante déjà définie.

On peut utiliser plutôt :

```
na5=NOTE(a5)
```

Créer une note

```
<variable name>=NOTE(<note>)
```

- `<variable>` peut avoir le nom qu'on veut, HORMIS un nom de constante, comme `a5` .
- La `<note>` peut être soit un string soit une constante (cf. Désignation des notes)
- Une telle séquence (un pas) ne doit pas comporter d'espaces.

Déplacer une note

Pour déplacer une note, on utilise le pas :

```
<nom variable note>.moveTo(<nouvelle note>[,<params>])
```

Exemple :

```
maNote=NOTE(a4) // crée la note LA 4
maNote.moveTo(g4) // descend la note vers SOL4
```

Ne mettre aucune espace dans ce code.

Noter que la note de destination devra vraiment la nouvelle valeur de la note. Si la note "a4" se déplace vers "a3", cette note deviendra "a3" dans ses données.

Placer une note sur une portée précise

Si on veut placer une note sur une portée hors de la portée active, on indique l'indice de cette portée avant la note, puis ":" :

```
<indice portée>:<note>
```

Par exemple :

```
note_autre_staff=NOTE('2:a4')
OU
note_autre_staff=NOTE('2:'+a4)
```

… placera un LA4 sur la deuxième portée, même si elle n'est pas active.

Noter que cela ne rend pas la portée active.

Détruire d'une note

Pour détruire la note (la supprimer de l'affichage, utiliser :

```
<nom variable note>.remove()
```

Les Accords

Table des matières

Création d'un accord

On crée un accord avec :

```
monAccord=CHORD('<note1> <note2>... <noteN>')
```

… où chaque note doit correspondre à la définition normale.

Par exemple :

```
accDom=CHORD('c3 eb3 g3')
```

Création d'un accord sur plusieurs portées

On peut poser l'accord sur différentes portées en ajoutant `<indice portée>:` devant la note (*rappel : l'indice portée est "1-start", donc "1" pour la première portée en comptant depuis le haut*).

Noter qu'il est inutile d'indiquer l'indice portée de la portée active.

Par exemple (en imaginant que la portée 1 est la portée active) :

```
acc=CHORD('2:c3 2:g3 e4 g4')
```

… placera "c3" et "g3" sur la 2e portée (certainement la clé de fa) et les notes "e4" et "g4" sur la 1ère portée qui est la portée active.

Référence aux notes de l'accord

On fait appel aux notes de l'accord par :

```
<nom accord>[<indice note>]
```

… où `<indice note>` est l'indice 1-start.

Par exemple :

```
accDom=CHORD('c3 eb3 g3')
accDom[1].moveTo('c4')
// Prends la première note (c3) et la déplace en c4.
```

Destruction d'un accord

Pour détruire l'accord, utiliser :

```
<nom variable accord>.remove()
```

Les portées

Table des matières

Créer une portée

Pour afficher une portée, utiliser le pas :

```
NEW_STAFF(<clé>)
```

… où `<cle>` peut être `SOL`, `FA`, `UT3`, `UT4`.

Par exemple :

```
NEW_STAFF(SOL)
```

… qui affichera une portée en clé de sol en dessous de la dernière portée.

Noter que cette portée deviendra la portée active, c'est-à-dire celle où seront placées les objets définis par la suite.

Activer une portée

Activer une portée signifie que tous les pas suivants la viseront. Par exemple, les notes se déposent toujours sur la portée active.

```
ACTIVE_STAFF(<indice de la portee>)
```

… où `<indice de la portee>` est son rang dans l'affichage, en partant de 1 et du haut. Donc la portée la plus en haut s'active par :

```
ACTIVE_STAFF(1)
```

Supprimer des lignes supplémentaires

Pour le moment, la suppression de lignes supplémentaires n'est pas automatique, afin de laisser toute liberté à la programmation de l'animation.

On détruit ces lignes à l'aide de la commande :

```
REMOVE_SUPLINE(<parameters>)
```

Une ligne supplémentaire est caractéristée par :

- La portée qui la porte ;
- Son indice à partir de la portée ;
- Sa position supérieure ou inférieure ;
- Son décalage à gauche (“frame” de l'animation).

Cela détermine les paramètres de `<parameters>`.

```
{
  staff: indice 1-start de la portée (depuis le haut),
  bottom: liste d'indices ou indice de la ligne à supprimer
 en bas,
  top: liste d'indices ou indice de la ligne à supprimer en
 haut,
  xoffset: décalage gauche (frame)
}
```

Toutes les valeurs à part `bottom` xou `top` sont optionnelles :

Si `staff` n'est pas précisé, on prendra la portée active.

Si `xoffset` n'est pas précisé, on prendra le décalage courant (ce qui représente le cas le plus fréquent, entendu qu'on va rarement supprimer une ligne supplémentaire "en arrière").

Note : lors d'un déplacement, une suppression ou tout autre effet qui doit rendre obsolète la ligne supplémentaire, il est préférable de déclencher la suppression des lignes supplémentaires AVANT la commande sur la note. Par exemple, pour un déplacement :

```
note=NOTE(c4) // ajoute une ligne supplément en bas
WAIT(2)
REMOVE_SUPLINE({bottom:1})
note.moveTo(c5)
```

Précision des indices

Les indices peuvent être une simple valeur numérique :

```
bottom:1 / top:1
```

… ou une liste d'indices

```
top:[1,2] / bottom:[1,2]
```

Noter que ces indices sont "1-start" et se comptent toujours À PARTIR DE LA portée, donc en descendant pour `bottom` et en montant pour `top` .

Noter aussi que `bottom` et `top` sont complètement indépendants, pour `bottom` on ne tient compte QUE des lignes supplémentaires inférieures et pour `top` on ne tient compte QUE des lignes supplémentaires supérieures.

Les Textes

Table des matières

Introduction

Les textes peuvent exister pour l'animation en général (ils sont alors écrits en haut à gauche et chaque nouveau texte remplace l'ancien), mais ils peuvent être associés aussi à tout objet de l'animation, note, portée, mesure, barre, etc.

Créer un texte

Créer un texte pour l'animation

Utiliser la commande :

```
WRITE("<le texte>")
```

Créer un texte pour un objet

Pour associer un texte à un objet, il faut bien sûr créer l'objet puis ensuite appeler sa méthode `write` (écrire) :

```
maNote=NOTE(a4)
maNote.write("C'est un LA 4")
```

Supprimer un texte

Supprimer un texte d'objet

Pour supprimer le texte de l'objet, c'est-à-dire de le faire disparaitre de l'affichage, utiliser la méthode `hide` (cacher) du texte de l'objet :

```
<note>.texte.hide()
```

Par exemple :

```
maNote=NOTE(a4)
# Écrire le texte
maNote.write("C'est un LA 4")
# Attendre 2 secondes
WAIT(2)
# Supprimer le texte
maNote.texte.hide()
```

Noter que cette méthode supprime l'affichage du texte, mais l'objet `texte` existe toujours pour l'objet et on peut le ré-utiliser plus tard, par exemple avec :

```
maNote.texte.show()
```

… qui ré-affichera ce texte.