

# MuScaT

## Manuel d'utilisation

Suite à la diffusion de mon Initiation à l'analyse musicale — dont je ne pensais pas qu'elle rencontrerait un tel succès ;-) — nombreux ont été les professeurs et autres pédagogues musicologues à me demander le logiciel que j'avais utilisé pour créer l'animation de cette initiation.

C'est malheureusement une application personnelle un peu trop... personnelle (comprendre : indomptable pour qui ne l'a pas créé), une usine à gaz ne fonctionnant qu'à la ligne de code.

Mais pour répondre à ces intérêts et à mes propres moyens, j'avais besoin d'un outil simple et pratique qui me permettrait de réaliser rapidement des analyses de partitions.

C'est ainsi qu'est né **MuScaT** (qui est composé de « Mu » pour « Musique », « Sc » pour « Score » — « partition » en anglais — et « Ta » à l'envers pour « Tag », le sens en français, comme les tags qu'on dépose sur les murs).

**MuScaT** permet donc de **réaliser rapidement, de façon très propre et très pratique, des analyses de partitions musicales**. Elle est semi-graphique, et permet d'ajuster très finement les éléments — au pixel près — de façon visuelle et agréable.

- Synopsis de fabrication
- Composition d'un tag
- Désignation des images (partitions)
- Tous les types (natures) d'éléments
- Les types de textes

## Synopsis général de l'analyse

Commençons par un aperçu général de la fabrication d'une analyse musicale à l'aide de **MuScaT**.

1. Création du dossier de l'analyse,
2. découpage de la partition en « images-systèmes»,
3. inscription des images-systèmes dans l'analyse,
4. ajouts des accords, des chiffrages, des cadences, de tous les éléments d'analyse,
5. impression en PDF,
6. récupération du code

On commence par **créer un dossier pour son analyse** en copiant-collant le dossier **template** du dossier principal de **MuScaT**. Nous vous conseillons

vivement de ne pas toucher à ce modèle.



Figure 1:

Appelons ce dossier `monAnalyse` pour illustrer.



Figure 2:

Si la partition que l'on s'apprête à analyser est suffisamment aérée (espace entre les systèmes), on peut la garder telle qu'elle. Dans le cas contraire (et le plus fréquent), il faut découper cette partition en systèmes, c'est-à-dire faire une image de chaque système.

Dans tous les cas, on place la ou les images dans le dossier `monAnalyse/images`.



Figure 3:

On ouvre ensuite son fichier `monAnalyse/tags.js`. C'est **le fichier principal de l'analyse**, celui qui va définir tous les éléments, les images, les marques de modulations, les accords, les cadences, les parties, tout ce qui constitue l'analyse.

On définit d'abord les images de la partition, en ajoutant des commentaires pour pouvoir se retrouver, plus tard, lorsque le fichier deviendra conséquent.

```

38
39
40 Tags = `
41 # Définir ci-dessous vos tags (aide: [TODO])
42 # -----
43
44 # Premier système
45 # -----
46
47 score ligne-01.png x=2 y=100
48
49 # Deuxième système
50 # -----
51
52 score ligne-02.png x=2 y=203
53
54 # Troisième système
55 # -----
56
57 score ligne-03.png x=2 y=269
58
59 `;
60
61
62
63
64
65
66
67
68

```

Figure 4: Exemple d'images dans tags.js

On définit ensuite tous les autres éléments graphiques : marque de parties, accords, chiffrages, numéros de portée, cadences, etc. On s'arrange pour les placer, dans `tags.js`, à peu près en fonction des positions des images de la partition. C'est-à-dire que si une cadence doit se produire sur le troisième système, il vaut mieux la définir après la ligne insérant l'image de ce troisième système (remarquez cependant qu'il n'y a aucune obligation là-dessus).

On ouvre le fichier `monAnalyse/partition.html` dans un navigateur internet (Firefox est le meilleur choix, pour **MuScaT**).

On place les éléments aux bons endroits simplement en les déplaçant à la souris, ou avec les flèches de son clavier.

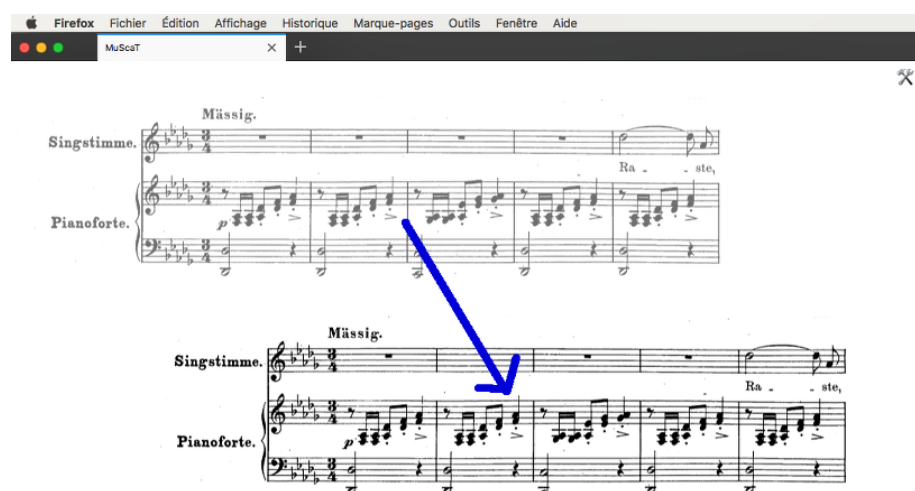


Figure 5: Exemple de déplacement d'élément

On demande enfin le code final de l'analyse, que l'on colle dans notre fichier `monAnalyse/tags.js` pour le conserver (si vous voulez en garder une trace ou pouvoir le modifier plus tard).

On imprime la page HTML du navigateur en choisissant le format PDF (ou on enregistre la page au format HTML et on utilise un outil de transformation des pages HTML en PDF).

Et voilà, c'est fait !

## Composition d'un tag

Un tag — image de la partition comprise — se compose d'une ligne dans le fichier de données.

Cette ligne a le format général suivant :

```
<nature>[ <contenu>][ <coordonnées>][ <options, type>]
```

Par exemple, pour une cadence (nature = 'cadence') de « V I » (contenu = 'V\_I') qu'on veut placer à 200 pixels depuis le haut (coordonnée y = 200) et 100 pixels de la gauche (coordonnées x = 100), de type « cadence parfaite » (type = 'parfaite'), on insérera dans son fichier `tags.js`, sous la définition de l'image (« score ») :

```
Tags = `  
  
    score ma_partition.jpg y=100 x=10  
  
    cadence V_I type=parfaite y=200 x=100  
  
`;
```

L'intégralité des natures d'éléments est détaillé ici.

## Désignation des images

Il existe trois mots clés pour indiquer la nature d'une image : **image**, **score** ou **partition**. C'est le premier mot à trouver sur la ligne d'une image. Juste après, on doit trouver le nom de cette image, ou son chemin relatif depuis le dossier **image** du dossier de votre analyse.

```
partition haydn/premier_mouvement.png [...]
```

Ci-dessus, l'image `premier_mouvement.png` doit donc se trouver dans le dossier `./images/haydn/` de votre dossier d'analyse.

Le plus souvent, il n'est pas pratique d'utiliser une seule image pour toute une partition. Il y a trop peu d'espace entre les systèmes. On conseille donc fortement de découper les partitions en systèmes (vous pouvez trouver des indications sur la procédure de découpage de la partition ci-dessous).

Mais il serait fastidieux d'entrer la ligne de chaque image de système dans notre fichier `tags.js`. Au lieu de ça, si les images des systèmes ont été correctement nommées (avec des suites de nombres), il suffit d'une seule ligne pour entrer toute la partition :

```
score haydn/mouvement_1-[1-35].png
```

Le texte ci-dessus indique qu'il y a 35 images de système dans ce mouvement.  
Le code qui en résultera sera :

```
score haydn/mouvement_1-1.png
score haydn/mouvement_1-2.png
score haydn/mouvement_1-3.png
score haydn/mouvement_1-4.png
...
...
score haydn/mouvement_1-35.png
```

Nous vous invitons vivement à commencer par cette opération avant insertion de toute autre marque sur la partition.

Une fois ce code établi, vous pouvez déplacer les images dans la page pour les ajuster à vos besoins. Cela créera automatiquement les *x* et les *y* des coordonnées spatiales de chaque système au bout des lignes de score.

Astuce : si votre écran est assez grand et que vous adoptez l'option `code beside` (ou `code à côté`), vous pourrez voir en direct votre code s'actualiser.

## Nature des éléments

Détaillons ces éléments.

Dans la ligne, le premier mot qui définit la *<nature>* du tag peut être (note : les deux mots, français et anglais, sont utilisables) :

Français	Anglais	Description	Exemple
image	score	Pour ajouter l'image d'une partition à tagger.	mon.png
accord	chord	Le nom d'un accord, placé au-dessus de la portée.	Dm7
harmonie	harmony	L'accord dans l'harmonie avec son renversement.	I**
cadence		Marque la cadence.	I
degre	degree	Marque le degré dans la gamme d'une note.	4
ligne	line		
mesure	measure	Pour ajouter un numéro de mesure	12
texte	texte	Pour écrire un texte quelconque.	

Le seconde « mot » définit le plus souvent le contenu textuel ou, pour les images, le nom du fichier dans le dossier **images**.

On peut par exemple écrire un texte quelconque à une position quelconque avec la ligne :

```
Tags = `
    texte Et_si_j'étais_un_texte_quelconque x=300 y=400
`;
```

Remarquez comme les espaces ont été remplacées par des tirets plats (qu'on obtient sur Mac avec la combinaison de touches Maj- — touche majuscule et tiret).

Ce deuxième sert aussi par exemple à définir le type des lignes à obtenir (cf. ).

Les deux autres informations capitales sont les positions verticale et horizontale du tag à poser (ou de la partition).

NOTE IMPORTANTE : dans votre fichier **tags.js**, ces valeurs peuvent dans un premier temps être approximatives, et seront affinées directement à l'écran.

On définit position verticale avec **y=** et la position horizontale avec **x=**, comme nous l'avons vu dans les exemples précédents. Le nombre est exprimé en pixels.

Pour les lignes et les cadences par exemple, on peut définir aussi la largeur avec la lettre « w » qui signifie « width » (largeur) en anglais : **w=200**. Le nombre correspond là aussi au nombre de pixels.

Ensuite, on peut définir certaines choses comme le « type » du tag. On l'a vu pour la cadence, par exemple. Les autres tags pouvant définir leur type sont le **texte** ou la **ligne** (bien que la **ligne** se définit plutôt par son contenu).

## Écrire des textes

Ce que l'on appelle les « textes », ici, ce sont tous les textes hors des accords, modulations, chiffrage, etc. Ce sont vraiment des textes qu'on peut placer n'importe où. À commencer pour définir les parties de l'ouvrage (« Introduction », « Coda », etc.).

Dans un texte, il est impérative de remplacer toutes les espaces par des traits plats (on les obtient, sur mac, à l'aide de Maj+tiret).

Par exemple, pour écrire sur la partition :

Premier couplet

Il faut définir la ligne :

ligne Premier\_couplet type=partie y= 50 x=200

Note : ici, c'est le type **partie** qui fera que le texte s'écrit de travers, dans une boîte.

## Tous les types (natures) d'éléments

Si vous avez déjà consulté ce manuel, vous pouvez trouver une aide rapide ici.

partition	<code>`image &lt;source&gt; x=... y=... z=...`</code> Exemple : <code>`image monScore.png z=50 x=100 y=100`</code> - "z" désigne le zoom en pourcentage - l'image doit se trouver dans le dossier 'images' Alias : score, image Note : un astérisque ("*") indique au départ une suite d'image (image1, image2, image3 etc.)
accord	<code>`accord &lt;nom&gt; x=... y=...`</code> Exemple : <code>`accord Cm7 x=230 y=520`</code> Alias : chord
harmonie	<code>`harmonie &lt;degré accord et renversement&gt; x=... y=...`</code> Exemple : <code>`harmonie II** x=200 y=230`</code> Alias : harmony, chiffrage
cadence	<code>`cadence &lt;degré accord&gt; type=&lt;type cadence&gt; x=... y=... w=...`</code> Exemple : <code>`cadence I type=italienne w=200 x=12 y=100`</code>
ligne	<code>`ligne &lt;type ligne&gt; x=... y=... w=...`</code> Exemples : <code>`ligne U w=120 x=100 y=50`</code> <code>`line  ---  w=50 x=100 y=50`</code>
degré	<code>`degre &lt;indice&gt; x=... y=...`</code> Exemple : <code>`degre 5 x=100 y=120`</code>
texte	<code>`texte &lt;contenu&gt; x=... y=... type=...`</code> Exemple : <code>`texte Exposition x=100 y=50 type=partie`</code>

## Les types de textes

type	anglais	Description
partie	part	Titre de partie, comme Exposition ou Coda
mesure	measure	Numéro de mesure, dans un carré.



## Dessiner des lignes

Les lignes se définissent par `line` ou `ligne`.

Le premier élément définit le **type** de la ligne. On trouve les types suivants.

U ou  __	Ligne inférieure et trait vertical avant/après
N ou  ---	Ligne supérieure et trait vertical avant/après
L ou  __	Ligne inférieure et trait vertical avant
K ou  ---	Ligne supérieure et trait vertical avant
V ou __	(Virgule) Trait inférieur et trait vertical après
^ ou ---	(Virgule inversée) Trait supérieur et trait vertical après

On peut ensuite définir sa taille et sa position avec les lettres habituelles `x` (position horizontale), `y` (position verticale) et `w` (largeur en pixels).

## Écrire un texte

TODO

## Procédure de découpage de la partition

TODO

## Options

### Option « code à côté »

L'option « code à côté » permet d'avoir le fichier contenant le code juste à côté de la partition, ce qui est très pratique pour le modifier sans avoir à changer d'application.

```

partition haydn/1.png x=213 y=24
partition haydn/2.png x=137 y=43
partition haydn/3.png x=171 y=64
partition haydn/4.png
partition haydn/5.png
partition haydn/6.png
// score sonata_haydn-0.jpg-1.png x=37 y=16
// harmony I x=251 y=129
// harmony I x=319 y=128
// harmony I x=249 y=131
// score sonata_haydn-0.jpg-3.png x=27 y=181
// score sonata_haydn-0.jpg-4.png x=30 y=340
//
// harmony I x=107 y=132
// line type=0 x=105 y=133 w=150

```

Actualiser (ALT →)

Figure 6: Code à côté de la partition