

# 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. Ajustement des éléments graphiques,
6. récupération du code final,
7. impression en PDF.

## Synopsis détaillé de l'analyse

### Création du dossier de l'analyse

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:

### Découpage de la partition en « images-systèmes »

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`.

### Inscription des images-systèmes

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.



Figure 3:

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

### Définition de tous les éléments de l'analyse

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).

### Ajustement des éléments graphiques

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.

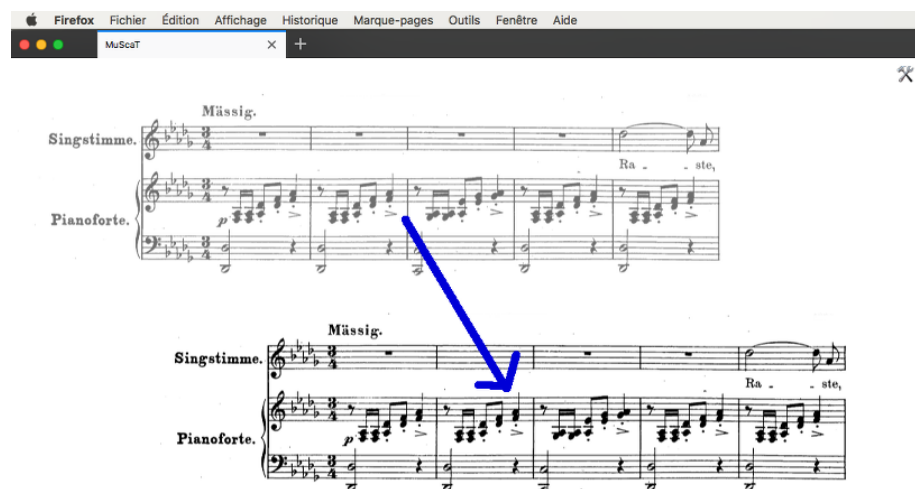


Figure 5: Exemple de déplacement d'élément

## Récupération du code final

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).

## Impression de l'analyse en PDF

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

Et voilà, c'est fait ! Et vous pourrez retoucher à votre analyse à n'importe quel moment grâce au fichier `tags.js` qui contient tout le code et les positions de l'analyse courante.

## 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      ``cadence <degré accord> type=<type cadence> x=... y=... w=...``  
Exemple : ``cadence I type=italienne w=200 x=12 y=100``

ligne        ``ligne <type ligne> x=... y=... w=...``  
Exemples : ``ligne U w=120 x=100 y=50``  
              ``line |---| w=50 x=100 y=50``

degré        ``degre <indice> x=... y=...``  
Exemple : ``degre 5 x=100 y=120``

texte        ``texte <contenu> x=... y=... type=...``  
Exemple : ``texte Exposition x=100 y=50 type=partie``

## 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

Voyons quelques méthodes de découpage de la partition en « images-systèmes ». Je les présente ici de la meilleure à la moins bonne. Cette qualité a été définie fonction des deux critères suivants :

- rapidité d'exécution,
- précision du découpage.

### Avec capture sélection dans Aperçu (Mac)

Méthode la plus rapide, mais également la moins précise. Ce manque de précision oblige parfois à reprendre des systèmes pour mieux les découper. Cependant, elle est tellement plus rapide que les autres que je la privilégie sans problème.

- Ouvrir la partition PDF dans l'application Aperçu,
- jouer **CMD Maj 4** pour activer la sélection par souris,
- sélectionner la zone de la partition à capturer (un système),
- recommencer l'opération pour tous les systèmes,
- récupérer les captures sur le bureau — sauf si l'astuce ci-dessous (1) a été utilisée — et les mettre dans le dossier **images** de votre analyse,
- modifier les noms des fichiers — sauf si vous avez utilisé l'astuce ci-dessous (1) — en les indiquant de 1 (ou 0) à N pour les insérer plus facilement dans l'analyse.

(1) Astuce : pour aller encore plus vite, vous pouvez :

- utiliser l'utilitaire Muscat **change\_folder\_captures** pour définir le dossier des captures écran ou consulter la procédure décrite ici. Vos captures iront directement dans ce dossier,
- effectuer les captures,
- utiliser l'utilitaire Muscat **rename\_images** pour renommer instantanément vos fichiers.

Note : vous pouvez voir ou revoir la procédure dans les tutoriels consacrés sur ma chaîne YouTube.

### Avec sélection rectangulaire dans Aperçu (Mac)

Une méthode qui ressemble à la précédente et permet d'être plus précis. Mais cette précision se fait au détriment du temps, notamment pour l'enregistrement des fichiers images.

- ouvrir la partition PDF dans Aperçu,
- choisir la sélection rectangle (p.e. Outils > Sélection rectangulaire),
- sélectionner le système grossièrement,
- ajuster parfaitement la sélection à l'aide des poignées,

- copier la sélection (CMD C),
- activer la combinaison CMD N pour créer une nouvelle image à partir du presse-papier,
- enregistrer l'image (CMD S) avec le nom voulu, dans le dossier voulu, en choisissant le format voulu.

### **Avec Aperçu, sélection souris et rectangle (Mac)**

On peut bien entendu imaginer une méthode intermédiaire qui reprendrait les deux méthodes précédentes. Lorsque la découpe est facile, on utilise la première, lorsque la découpe demande plus de précision, on privilégie la seconde.

### **Avec MuScaT et convert**

C'est une méthode qui souffre parfois d'un manque de qualité de rendu.

On tire déjà les images du PDF à l'aide de la commande à jouer dans le Terminal (adapter la qualité du traitement en fonction du résultat) :

```
# Se trouver dans le dossier contenant la partition (cd ...)
convert[ options] partition.pdf partition.jpg # ou .png
```

Autant d'images que de pages sont produites.

On insert la première dans le code du fichier `tags.js`, avec l'option `crop image` :

```
# Dans tags.js
option('crop image')
Tags=`
partition partition-0.jpg
`;
```

On ouvre le fichier `partition.html` dans Firefox.

Maintenant, il suffit de sélectionner, à la souris, la zone de l'image à prendre puis de coller le code du presse-papier dans la console du Terminal. Puis de jouer ce code.

Répéter l'opération avec chaque système, puis avec chaque page de la partition.

### **Avec Gimp/Photoshop (ou autre logiciel de traitement de l'image)**

Si Gimp présente une précision de découpage inégalable, l'application offre en revanche la méthode la plus chronophage, même avec l'habitude du logiciel.

- ouvrir le PDF dans Gimp,
- sélectionner chaque système en le découpant,
- le placer en haut,
- « cropper » l'image à la taille du plus haut système,
- exporter chaque image-système (avec le bon nom).

Ce mode d'emploi n'étant pas destiné à maîtriser Gimp, je vous renvoie au manuel d'utilisation de l'application.

## 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. On le voit ci-dessous dans la boîte noire.



Figure 6: Code à côté de la partition

### Option « crop image » (ou « découpe image »)

L'option « crop image » fait passer dans un mode d'utilisation qui va permettre de découper l'image de façon aisée (par simple [copié]-collé).

TODO:

## Utilitaires

L'application **MuScaT**, comme tout bon vin, est fournie avec quelques utilitaires pour se faciliter la vie, en tout cas sur Mac. En voici la liste avec leur mode d'utilisation.

### Renommage des fichiers images (Mac/Unix)

Ce script, qui se trouve dans le dossier `utils` de l'application, permet de renommer les images d'un dossier de façon cohérente et indexée.

Pour utiliser ce script :

- ouvrir l'application Terminal,
- rejoindre (commande `cd`) le dossier de l'application MuScaT (ATTENTION : ça n'est pas le dossier de l'analyse, ici, c'est bien le dossier de l'application),
- taper `./utils/rename_images.rb -h` et la touche Entrée pour tout savoir du script.

### Changement du dossier des captures écran (Mac)

Par défaut, les captures d'écran sont enregistrés sur le bureau. Ça n'est pas gênant en soit, il suffit de les glisser ensuite dans le dossier `image` de l'analyse. Mais si on veut encore gagner du temps, ce script permet de changer le dossier de destination.

Voici la procédure :

- ouvrir l'appli Terminal,
- rejoindre (commande `cd`) le dossier de l'application MuScaT (ATTENTION : ça n'est pas le dossier de l'analyse, ici, c'est bien le dossier de l'application),
- taper `./utils/change_folder_captures.rb -h` et la touche Entrée pour tout savoir du script.

Pour remettre la valeur par défaut (le bureau), jouer simplement `./utils/change_folder_captures.rb` sans aucun argument.