RLily

Permet de construire rapidement une partition à l'aide de Lilypond.

1. Écrire le code en ruby (dans un fichier .rb)
2. Taper `CMD + ALT + CTRL + S`

Et la partition est produite.

Pour le moment, cette partition est très simple, elle est produite pour le piano (clé de sol et clé de Fa) et ne supporte que les ajouts de notes (et autres marques du code normal de Lilypond comme les doigtés, les liaisons, les accords, etc.)

Quick référence pour la création

1. Créer le fichier RLily (.rlily) qui va contenir le code pour la partition (c'est le "fichier source")
2. Définir les données du score (Données générales du score)
3. Définir le contenu de chaque main avec `MD << "<notes>"` et `MG << "<notes>"`

La suite dépend de l'utilisation ou non du Bundle `Phil:RLily`. S'il est présent :

1. Avec le fichier source activé, utiliser le menu `Produire la partition` (ou jouer le raccourci `CMD + MAJ + P`)

Si le bundle n'est pas présent :

1. Ouvrir le fichier `_rlily.rb`
2. Définir la valeur de `SCORE_PATH` en mettant le path du fichier source
3. Runner `_rlily.rb` pour produire la partition

En cas d'échec

Dans certains dossiers, les permissions peuvent manquer. Dans ce cas, le programme ouvre le fichier `.rlily` dans TextMate, et il suffit alors de taper `CMD + MAJ + P` pour produire la partition.

Données générales du score

Définir dans le code du fichier source (toutes les valeurs peuvent être omises) :

```
SCORE::titre = "LE TITRE DU MORCEAU"
SCORE::compositeur = "LE COMPOSITEUR"
SCORE::armmure = "L'ARMURE ('A', 'B', etc.)"
SCORE::metrique = "4/4"
```

Constantes utiles

Le fichier `lib/music/constantes_textuelles.rb` contient des

constantes textuelles utiles, pour marquer des barres, etc.

Utilisation de la commande `rlily`

On peut construire une partition à l'aide de :

```
rlily "path/to/file_partition.rb"
```

Pour que ça fonctionne, il faut que la commande `rlily` ait été définie :

```
$ cd /usr/bin
$ sudo touch rlily
[Entrer mot de passe]
$ sudo chmod 0777 rlily
```

ci-dessous, vous devez remplacer `PATH/TO/RUBY2LILY/` par le path à votre dossier téléchargé de ruby2lily :

```
$ sudo echo 'exec PATH/TO/RUBY2LILY/ruby2lily.rb "$@"' > rl
ily
# $ sudo echo 'exec /Users/philippeperret/Programmation/Pro
grammes/RLily/_rlily.rb "$@"' > rlily
$ sudo chmod u+x rlily
```

Indication de l'opus

```
SCORE::opus = <opus>
```

Par exemple :

```
SCORE::opus = 13
```

Note : Ne pas mettre "Opus" ou "Op."

Indication de l'armure

```
SCORE::armure = <armure>
```

`<armure>` se désigne comme les accords : "A#" pour La dièse majeur, "Bbm" pour Si bémol mineur, etc.

Indication du tempo

Tempo pour la valeur de la noire

```
SCORE::tempo = <tempo|tempo name>
```

Par exemple :

```
SCORE::tempo = 126

SCORE::tempo = "Andante"
```

Tempo pour une valeur autre que noire

```
SCORE::tempo = {:duree => <durée>, :tempo => <tempo>, :name
 => <nom tempo>}
```

Par exemple, pour mettre la croche (8) à 60 :

```
SCORE::tempo = {:duree => 8, :tempo => 60, :name => "Allegr
o"}
```

Définir l'instrument

```
SCORE::instrument = <instrument>
```

Définir l'espace entre titre et premier système

```
SCORE::espace_titre_systemes = <valeur>
```

Définir l'espacement entre les systèmes

```
SCORE::espace_entre_systemes = <valeur>
```

Définir l'espace entre les portées

```
SCORE::espace_entre_portees = <valeur> # défaut : 3 (ne pas
 mettre une valeur trop grande)
```

Définir la taille de la partition

```
SCORE::taille = <valeur>
```

Où `<valeur>` peut être :

```
PARTITION_CLASSIQUE/20    Taille normale pour les partition
s classiques
LIVRET_POCHE/11           Les livrets de poche
13, 14, 16
LIVRE_CHANT / 18          Les livres de chant
23 et 26
```

Écrire le code de la partition

Concaténation

Pour éviter les erreurs en additionnant des strings, la class `String` a été modifiée :

```
"a" + "b"   => "a b"
"a" << "b"  => "a b"
```

Donc il est inutile de terminer ou de commencer les strings de notes par des espaces.

Options pour les fonctions musicales

Ces options sont le second argument de toutes les fonctions musicales. Elles peuvent définir :

```
:duree        La durée de la note/des notes
:octave       L'octave (note : en notation anglosaxone, don
c le DO sous la portée
              de Sol est un Do 4)
:doigte       Le doigté à utiliser
:jeu          Le mode de jeu, défini explicitement avec du
code
              lilypond ("."， "-", etc.) ou les constantes t
extuelles
```

Définir l'octave

Si des variables sont employées pour simplifier le code et le rendre plus explicite, il est bon de définir l'octave précise des motifs pour qu'ils puissent être enchainés. On utilise pour ça la fonction `rel` (pour "relative") :

```
rel(<notes>, <octave>)
# La première des <notes> commencera toujours à l'octave <o
ctave>
```

Par exemple :

```
rel("c e g", 5)
# => \relative c''{ c e g }
```

Note : Pour définir une écriture avec "8ve----", cf. Écrire à l'octave.

Définir la clé

```
MD << <clé>
```

Par exemple :

```
# Par constante (cf. ci-dessous)
MD << CLE_FA

# Explicitement
MD << "\\clef \"bass\""
```

Constantes :

```
CLE_F/CLE_FA      Clé de fa
CLE_G/CLE_SOL     Clé de sol
```

Écrire un accord

```
chord(<"notes"|[notes]>[, <options>])
```

Par exemple :

```
MD << chord("c e g")
# => "<c e g>"

MD << chord([c, E, g], :duree => 8)
# => "<c e g>8"
# Noter que les notes peuvent être définies par des minuscu
les ou des majuscules

MD << chord("c e g", {:duree => 4, :jeu => PIQUE})
# => "<c e g>4-."
```

Pour les `<options>`, cf. Options pour les fonctions musicales

Écrire un triolet

```
triolet(<notes>[, <options>])
```

Par exemple :

```
MD << triolet([c, d, e])
# => "\tuplet 3/2 { c d e }"

MD << triolet("c d e", :duree => 16)
# => "\tuplet 3/2 { c16 d e }"
```

Pour les `<options>`, cf. Options pour les fonctions musicales

Écrire à l'octave

```
octave(<notes>[, <nombre d'octave (1 par défaut)>])
```

Par exemple :

```
MD << octave("c'' e f g")
# => Ecrit les notes une octave en dessous, avec la marque
"8ve-----"

MD << octave("c''' e f g", 2)
# => Ecrit les notes une octave en dessous, avec la marque
"15ma-----"
```

Réglage des options

Demander l'affichage des dimensions

```
SCORE::option :display_spacing => true

ou

SCORE::option :display_spacing, true
```

Ajouter un slash entre les systèmes

```
SCORE::option :slash_between_systemes => true
```