

# LilyPond

---

Le système de gravure musicale

## Nouveautés

### L'équipe de développement de LilyPond

Ce document recense les modifications et les nouvelles fonctionnalités de LilyPond pour la version 2.23.11 (depuis la 2.22).

Pour connaître la place qu'occupe ce manuel dans la documentation, consultez la page Section "Manuels" dans *Informations générales*.

Si vous ne disposez pas de certains manuels, la documentation complète se trouve sur <https://lilypond.org/>.

Ce document a été placé dans le domaine public ; en France, les auteurs renoncent à tous leurs droits patrimoniaux.

Pour LilyPond version 2.23.11

---

## Nouveautés en matière de notation musicale

### Améliorations de la représentation des hauteurs

- La prise en charge d'altérations alternatives s'améliore. Grâce à la propriété `alterationGlyphs` des contextes de niveau portée, peuvent se définir globalement les glyphes à utiliser pour tous les objets – voir Section “Glyphes d'altération alternatifs” dans *Manuel de notation*.

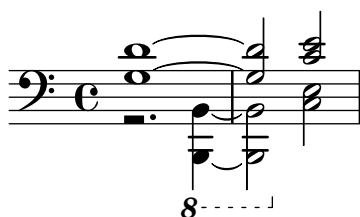


- Dans le cadre de la fonte Emmentaler, les têtes de notes dont l'aspect est identique et pour lesquelles la seule différence réside dans l'orientation de leur hampe ont été consolidées en un glyphe unique. Par exemple, les glyphes `noteheads.u2triangle` et `noteheads.d2triangle` ont été remplacés par le seul glyphe `noteheads.s2triangle`. Les paires de tête de note ayant un aspect différent selon l'orientation de la hampe conservent leur distinction.

Par ailleurs, la propriété `stem-attachment` des objets graphiques `NoteHead` renvoie le point d'attachement réel selon l'orientation de hampe en lieu et place d'un hypothétique point d'attachement d'une hampe ascendante.

- Les crochets d'octaviation peuvent s'appliquer à une seule voix plutôt qu'à l'intégralité de la portée. Ceci demandait auparavant quelques circonvolutions.

```
\layout {
  \context {
    \Staff
    \remove Ottava_spanner_engraver
  }
  \context {
    \Voice
    \consists Ottava_spanner_engraver
  }
}
```

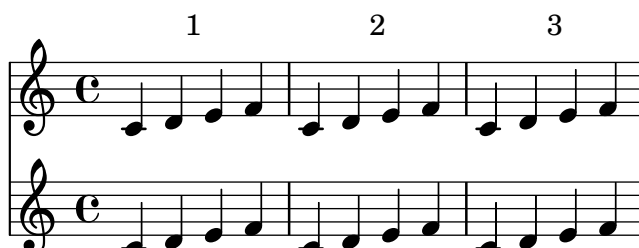


### Améliorations en matière de rythme

- Les numéros de mesure peuvent se présenter au centre de la mesure comme il est d'usage dans les musiques de film.

```
\layout {
  \context {
    \Score
    centerBarNumbers = ##t
    barNumberVisibility = #all-bar-numbers-visible
  }
}
```

```
<<
{ \repeat unfold 3 { c'4 d' e' f' } }
{ \repeat unfold 3 { c'4 d' e' f' } }
>>
```



- Les compteurs de mesures prennent en considération les mesures à compter et les alternatives.



- `\numericTimeSignature` et `\defaultTimeSignature` s'appliquent désormais en même temps à toutes les portées – plus précisément à toutes les portées d'un même contexte Timing – afin de correspondre au comportement de `\time`.
- Le nouveau `Mark_tracking_translator` prend le pas sur le `Mark_engraver` en matière de décision quant à la création d'une marque. Le `Mark_engraver` se charge toutefois de contrôler la mise en forme et le positionnement vertical du repère.  
Par défaut, des `Mark_engravers` dans de multiples contextes créent une séquence commune de repères. Lorsque des séquences indépendantes sont préférables, il faut utiliser plusieurs `Mark_tracking_translator`.
- La commande `\enablePolymeter` agit comme un raccourci lors de la saisie pour déplacer les graveurs selon le besoin et permettre d'avoir différentes métriques en parallèle. Le code :

```
\layout {
  \context {
    \Score
    \remove Timing_translator
    \remove Default_bar_line_engraver
  }
  \context {
    \Staff
    \consists Timing_translator
    \consists Default_bar_line_engraver
  }
}
```

peut ainsi s'abrégier en :

```
\layout {
  \enablePolymeter
}
```

Par ailleurs, le `Default_bar_line_engraver` a été supprimé.

- Au lieu de répéter le nom de glyphe en cours de ligne, `\defineBarLine` accepte désormais la valeur `#t`.
- Le type de barre de mesure "-" a été supprimé. `convert-ly` le convertit en "'". Il en résulte une légère différence dans l'espacement horizontal lors d'un saut de ligne.
- `automaticBars` a été supprimé. `convert-ly` convertit `automaticBars = ##f` en `measureBarType = #'()`.
- L'instruction `\bar "`, " crée une barre de mesure raccourcie.



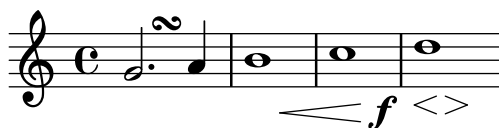
## Améliorations en matière d'expressivité

- L'apparence des extensions de trille a été modifiée pour mieux correspondre aux conventions classiques de la gravure. L'extension s'arrête désormais juste avant la note qui suit, sans la chevaucher. Ce sera juste avant l'altération si la note qui suit en est pourvue, et au niveau de la barre si la prochaine note est dans la mesure suivante.



- Les événements attachés à des notes, tels que nuances ou articulations, peuvent être différés d'une durée arbitraire à l'aide de `\after`. Ceci permet de simplifier de nombreuses situations qui jusqu'alors demandaient de recourir à une polyphonie et des silences invisibles.

```
{
  \after 2 \turn g'2. a'4
  \after 2 \< b'1
  \after 2. \f c''
  <>\< \after 4 \> \after 2\! d''
}
```



- Les terminaisons des soufflets peuvent désormais s'aligner sur la gauche, au centre ou sur la droite des `grobs` `NoteColumn` à l'aide d'une dérogação à la propriété `endpoint-alignments`.



- La nouvelle commande `\vshape` se comporte comme `\shape`, à ceci près qu'elle affiche les points de contrôle et le polygone qui les enveloppe, afin de faciliter les ajustements.

```
{ a1\vshape #'((0 . 0) (0 . 0.5) (0 . 0.9) (0 . 0.4))^( c'1) }
```



- La nouvelle propriété de contexte `breathMarkType` sélectionne le marqueur produit par la commande `\breathe` parmi plusieurs types prédéfinis.

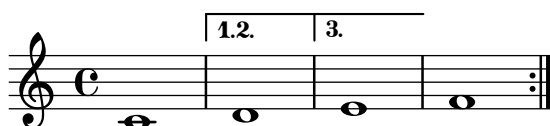
```
\fixed c' {
  \set breathMarkType = #'tickmark
  c2 \breathe d2
}
```



## Améliorations en matière de reprises

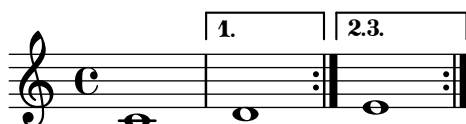
- Les alternatives peuvent se placer au sein même du bloc de répétition.

```
\repeat volta 3 { c'1 \alternative { d' e' } f' }
```



- Les numéros d'alternative peuvent se définir à l'aide de la commande `\volta`.

```
\repeat volta 3 c'1 \alternative { \volta 1 d' \volta 2,3 e' }
```

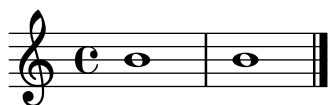


- La commande `\volta` supprime la musique lorsque la répétition est expansée.
- La commande `\unfolded` ajoute la musique lorsque la répétition est expansée.
- La nouvelle commande `\repeat segno` permet de gérer automatiquement un certain nombre de formes *da-capo* et *dal-segno*.

```
music = \fixed c' {
  \repeat segno 2 {
    b1
  }
  \fine
}

\score { \music }
\score { \unfoldRepeats \music }
```

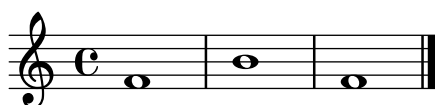




- La nouvelle commande `\fine` insère une barre de mesure finale qui interagit de façon optimale avec les barres de reprise. Placée à l'intérieur d'une reprise, elle ajoute une instruction *Fine* et termine la musique lorsque les reprises sont expansées.

```
music = \fixed c' {
  \repeat volta 2 {
    f1
    \volta 2 \fine
    b1
  }
}

\score { \music }
\score { \unfoldRepeats \music }
```



## Améliorations en matière de notation sur la portée

- La nouvelle fonction `Scheme universal-color` procure une palette de huit couleurs conçue pour lever certaines ambiguïtés dont souffrent les gens atteints de dyschromatopsie.

**black**

**orange**

**skyblue**

**bluegreen**

**yellow**

**blue**

**vermillion**

**redpurple**

- Il est désormais possible de contrôler la largeur et l'aspect de certains chiffres Emmentaler à l'aide de fonctionnalités OpenType.

**0123456789**

(time signatures)

**0123456789**

(fingering)

# 0123456789 (fixed-width digits)

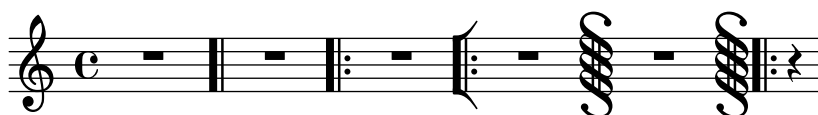
- L'ajout du `Melody_engraver` à un contexte `Voice` prend désormais en compte le sens de la mélodie pour orienter la hampe de la note sur la ligne médiane. Il fallait auparavant apporter une dérogation spécifique à la propriété `Stem.neutral-direction`.

```
\new Voice \with {
  \consists Melody_engraver
}
\relative c'' {
  \autoBeamOff
  g8 b a e g b a g |
  c b d c b e d c |
}
```



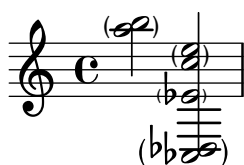
La propriété de contexte `suspendMelodyDecisions` permet de désactiver temporairement ce comportement, tout comme le faisait `\override Stem.neutral-direction = #DOWN`.

- L'instruction `\bar ""` n'est plus un préalable à l'affichage du numéro de la première mesure. Il suffit désormais de régler la propriété `barNumberVisibility` à `all-bar-numbers-visible` ou l'un des autres réglages pour lesquels le premier numéro de mesure est visible. Notez bien qu'il s'agit d'un changement de comportement pour les partitions dans lesquelles `barNumberVisibility` est réglé à `all-bar-numbers-visible` ou équivalent et `BarNumber.break-visibility` activé sans avoir de `\bar ""`. Un numéro de mesure est désormais affiché au début. Il s'agit bien du comportement attendu (**tous** les numéros devraient être visibles) mais, en raison d'une documentation probablement pas assez explicite, certains utilisateurs ont opté pour ces réglages afin d'afficher les numéros de mesure en cours de portée sauf pour la première mesure. En pareil cas, il convient de tout simplement supprimer la clause `\set Score.barNumberVisibility = #all-bar-numbers-visible` puisque `\override BarNumber.break-visibility = ##t` est le réglage approprié et suffisant.
- Les types de barre prédéfinies qui suivent n'apparaissent plus sous forme de simple barre lorsqu'ils surviennent en fin de ligne. Des types de barre annotée, tel que `\bar "S-|"`, ont été ajoutés en conséquence.



## Améliorations en matière d'annotations éditoriales

- Il est désormais possible de placer un accord entre parenthèses. Toutefois, la taille de la fonte utilisée pour les parenthèses devra être ajustée manuellement.



- Les objets étendus ou bandeaux peuvent être mis entre parenthèses.



- Une version « temporelle » de la commande `\parenthesize` est disponible. Elle prend en considération le chemin vers un objet graphique : `\parenthesize NomGrob` ou `\parenthesize ContextName.GrobName`. Cette commande se comporte comme un `\once \override`. Cette interface vient en complément de la forme déjà disponible `\parenthesize événement`, de manière identique à `\footnote`.

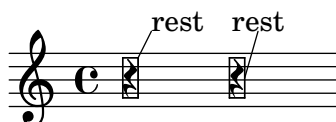
```
{
  \parenthesize NoteHead
  c'1
  \parenthesize Staff.KeySignature
  \key g \major
  c'1
}
```



- En matière de basse chiffrée, il est désormais possible de placer les altérations entre crochets.

```
7 [4]5 [45]
[ $\sharp$ ]3
```

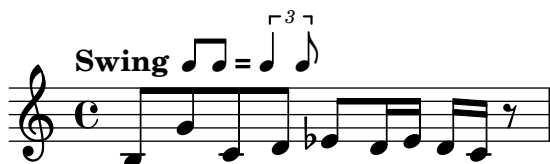
- La mise en forme de info-bulles est désormais modifiable.



## Améliorations en matière de mise en forme du texte

- La commande de `markup \rhythm` fait son apparition. Il s'agit d'un moyen simple de mélanger du texte et du rythme comme, par exemple, pour afficher une indication de « swing ».

```
\relative {
  \tempo \markup {
    Swing
    \hspace #0.4
    \rhythm { 8[ 8] } = \rhythm { \tuplet 3/2 { 4 8 } }
  }
  b8 g' c, d ees d16 ees d c r8
}
```





- Sont maintenant disponibles les commandes de *markup* `\with-true-dimension` et `\with-true-dimensions`. Elles procurent l'étendue réelle de la surface encrée, qui peut être quelque peu différente de l'étendue par défaut pour certains glyphes en raison des contraintes de régularité du texte.



- Sont maintenant disponibles les deux commandes de *markup* `\with-dimension` et `\with-dimension-from`. Elles sont similaires à `\with-dimensions` et `\with-dimensions-from`, à ceci près qu'elles permettent de ne jouer que sur une seule des deux dimensions.
- Sont désormais disponibles dans la fonte Emmentaler, des variantes textuelles pour les glyphes de dièse, bémol, bécarre, double dièse et double bémol. En mode *markup*, ils sont accessibles par leur valeur Unicode standard.

**1 # 2 ♭ 3 ♮ 4 ♯ 5 × 6**

- La nouvelle commande de *markup* `\align-on-other` déplace un *markup* comme s'il était aligné sur un autre.

```
1
12
  12345
123
```

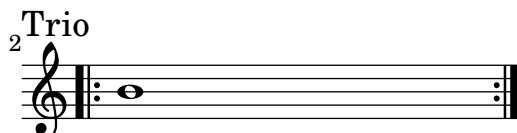
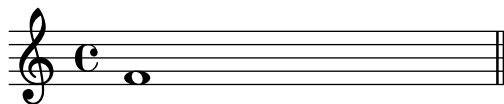
- La nouvelle commande de *markup* `string-lines` permet de couper une chaîne à un caractère donné. La coupure intervient par défaut au saut de ligne, et les éventuels espaces sont ignorés. La liste résultante de *markups* peut faire l'objet d'une mise en forme particulière. Cette fonctionnalité constitue un moyen tout à fait adapté pour ajouter des couplets à une chanson.

```
Twinkle, twinkle, little star,
How I wonder what you are!
Up above the world so high,
Like a diamond in the sky.
Twinkle, twinkle, little star,
How I wonder what you are!
```

- La nouvelle commande `\section` insère une double barre qui interagit de manière optimale avec les barres de reprise. Un passage peut être nommé à l'aide de la commande `\sectionLabel`.

```
\fixed c' {
  f1
  \break
  \section
  \sectionLabel "Trio"
  \repeat volta 2 {
    b1
  }
}
```

}



- L'instruction `\markup \path` est désormais également opérationnelle pour une sortie SVG même si le chemin n'est pas initialisé par une commande `moveto` ou `rmoveto`. Elle accepte par ailleurs les raccourcis équivalents SVG (`moveto` = M, etc.).
- La fonction `markup->string` convertit un *markup* dans une représentation approximative de chaîne de caractères. Ceci est utile pour générer des métadonnées PDF ainsi que les paroles et repères en MIDI. Il est donc possible de créer des commandes pour *markups* personnalisées afin de retraiter les chaînes converties par `markup->string`, comme par exemple

```
#(define-markup-command (upcase layout props arg) (string?)
  #:as-string (string-upcase arg)
  (interpret-markup layout props (string-upcase arg)))
```

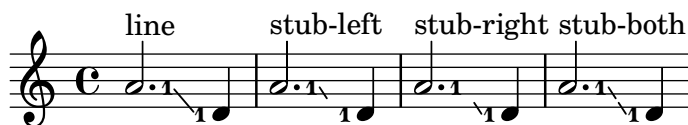
## Nouveautés en matière de notation spécialisée

### Améliorations pour la musique vocale

- Le nouveau contexte `VaticanaLyrics` est similaire au contexte `Lyrics`, à ceci près qu'il fournit un style d'hyphénation utilisé communément dans le style des éditions vaticanes, à savoir un trait d'union simple accolé à la syllabe de gauche.

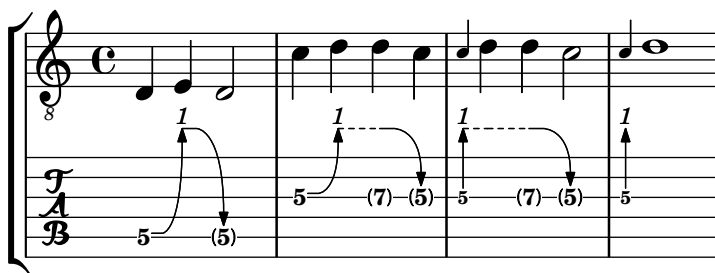
### Améliorations pour les cordes frettées ou non

- Ajout des accordages pour banjo `banjo-double-c` et `banjo-double-d`.
- Le nouvel objet graphique `FingerGlideSpanner` permet d'indiquer le glissé d'un doigt sur une corde passant d'une position à une autre. Il peut se présenter sous différentes formes selon le style adopté ; l'image ci-dessous affiche les styles `line`, `stub-left`, `stub-right` et `stub-both`.



D'autres styles sont disponibles : `dashed-line`, `dotted-line`, `zigzag`, `trill`, `bow` et `none`.

- Le nouvel objet graphique `BendSpanner` permet, dans le cadre d'un `TabStaff`, d'indiquer une désinance. En plus de l'apparence par défaut sont disponibles les styles `'hold`, `'pre-bend` et `'pre-bend-hold`.

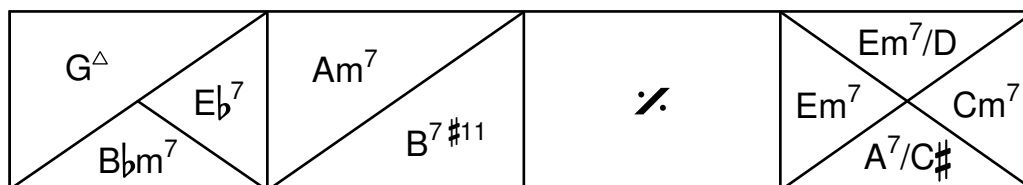


## Améliorations pour la notation des percussions

- Ajout du style de notation pour percussions `weinberg-drums-style` basé sur les travaux de standardisation de Norman Weinberg.

## Améliorations pour la notation des accords

- Les grilles harmoniques sont désormais prises en charge.



+

- Dans un `ChordNames`, les silences multimesure forcent désormais l'apparition du symbole « N.C. » comme pour des silences normaux.
- En matière de basse chiffrée, un `_` crée désormais un chiffre vide occupant de l'espace.

```
\figures {
  <8 _ 4> <_ 5+ 3>
}
```

```
8
4 #5
3
```

- La mise en forme de la basse chiffrée s'est améliorée. En particulier, sa taille par défaut est réduite à une valeur que l'on retrouve dans nombre d'éditions *Urtext* de musique baroque.
- La basse chiffrée utilise désormais par défaut des glyphes dessinés spécifiquement dans ce cadre, à savoir les `6\`, `7\` et `9\`. De même, des glyphes spécifiques seront utilisés par défaut en présence d'un signe plus après le chiffre pour les symboles `2\+`, `4\+` et `5\+`.

```
7\ 4+ 3 6 9
6\ b 5+ 4
4 3 3
2.
```

## Améliorations pour les musiques du monde

- LilyPond prend désormais en charge la musique persane. Deux glyphes d'altération ont été ajoutés à cet effet : *sori* et *koron*.

```
\include "persian.ly"

\relative c' {
  \key d \chahargah
  bk'8 a gs fo r g ak g |
```

```
fs ek d c d ef16 d c4 |
}
```



## Nouveautés en matière d'entrée et sortie

### Améliorations pour la structure

- Une même définition d'objet graphique peut désormais s'utiliser pour créer des *grobs* de classe différente (Item, Spanner, Paper\_column, System). Dans le cadre de cette évolution, les types de *grob* FootnoteItem et FootnoteSpanner ont été rassemblés en un unique type Footnote. De la même manière, BalloonTextSpanner et BalloonTextItem ont été unifiés en BalloonText.

Lorsque la définition du *grob* ne requiert pas de classe, les graveurs doivent décider de la classe à utiliser pour créer le *grob*. Pour ceux qui créent leurs propres graveurs en Scheme, cela signifie qu'il faudra utiliser soit `ly:engraver-make-item`, soit `ly:engraver-make-spanner`. La fonction utilitaire `ly:engraver-make-sticky` permet de prendre en charge le cas des *sticky grobs* tels que notes de bas de page et infobulles. Elle crée un *grob* de la même classe qu'un autre et gère la parenté et les attachements.

### Améliorations pour les titrages et entêtes

- La syntaxe permettant de gérer des conditions pour les *markups* a gagné en flexibilité et en simplicité. Elle utilise les nouvelles commandes de *markup* `\if` et `\unless`. Voici quelques uns des changements apportés :

#### Syntaxe version 2.22

```
\on-the-fly #first-page ...
\on-the-fly #not-part-first-page ...
\on-the-fly #(on-page n) ...
```

#### Syntaxe version 2.24

```
\if \on-first-page ...
\unless \on-first-page-of-part ...
\if \on-page #n ...
```

### Améliorations pour les fichiers résultants

- L'option `-dembed-source-code` permet maintenant d'embarquer également les images ajoutées par `\epsfile` ainsi que les fichiers inclus par `\verbatim-file`.
- `set-default-paper-size` et `set-paper-size` acceptent désormais des tailles de papier personnalisées.

```
 #(set-default-paper-size '(cons (* 100 mm) (* 50 mm)))
```

- `lilypond-book` prend en charge deux nouvelles options pour le traitement des extraits : `paper-width` et `paper-height` permettent de déterminer une taille particulière de papier.
- La valeur par défaut de l'option `aux-files` est désormais fixée à `#f`. Lorsque LilyPond est lancé avec l'argument `-dbackend=eps` et que sont nécessaires les fichiers `.tex` et `.texi`, il faut désormais spécifier explicitement l'option `-daux-files`. Les formats pour les images `lilypond-book` peuvent se définir séparément selon qu'il s'agit de la page (donc une sortie PNG pour du format HTML) ou des images indépendantes par système (pour des formats imprimables EPS ou PDF) à l'aide respectivement des sous-options `-dtall-page-formats` et `-dseparate-page-formats`.

## Améliorations pour le MIDI

- Le nouveau `Mark_performer` crée des événements MIDI à l'instar du `Mark_engraver` pour les sorties imprimables.

## Nouveautés en matière d'espacements

### Améliorations pour les sauts de ligne

- Désormais, la commande `\break` insère toujours un saut, outrepassant toutes les décisions par défaut quant aux points de rupture. Par exemple, il n'est désormais plus nécessaire d'ajouter `\bar ""` pour obtenir une rupture en cours de mesure.


La nouvelle commande `\allowBreak` insère un point de rupture potentiel, sans le forcer, mais outrepassa les décisions par défaut à l'instar de `\break`.

## Nouveautés en matière de modification des réglages par défaut

- Deux glyphes redondants ont été supprimés des fontes Emmentaler : `scripts.trillelement` – utilisez `scripts.trill_element` en remplacement – et `scripts.augmentum` – à remplacer par `dots.dotvaticana`.
- Les propriétés de `PaperColumn` et `NonMusicalPaperColumn` telle que `NonMusicalPaperColumn.line-break-system-details` acceptent désormais des dérogations au file de la musique à l'aide d'une simple instruction `\once \override`. Elles consituaient jusqu'alors une exception, requérant l'utilisation de la commande `\overrideProperty`.
- La nouvelle fonction de rappel pour objet graphique `break-alignment-list` permet de renvoyer une valeur différente selon le positionnement d'un *grob* par rapport à une rupture. Elle permet, par exemple, de fournir un alignement du *grob* différent selon qu'il est positionné en début, en cours de ligne ou à la fin.



## Nouveautés en matière d'interfaces et fonctions internes

- Le script `lilypond-book` autorise désormais l'utilisation d'accolades dans les arguments des commandes `\lilypond` (pour  $\text{\LaTeX}$ ) et `@lilypond` (pour `Texinfo`).
- `lilypond-book` ajoute désormais le répertoire courant comme dernière entrée pour la recherche des fichiers inclus, au lieu de le placer en premier dans la liste des chemins spécifiés. Ainsi, les fichiers situés dans les dossiers d'inclusion prennent le pas sur ceux du même nom dans le répertoire courant. L'effet ne sera visible que dans le cas où existent des fichiers du même nom de part et d'autre.
- `lilypond-book` prend en charge la nouvelle option `inline` pour le traitement d'extraits musicaux. Ceci permet de faire apparaître des bribes de partition, comme  au sein même d'un paragraphe de texte.
- Les traducteurs Scheme peuvent désormais contenir des *listeners* libellés ainsi :

```
(listeners
  ((event-class engraver event #:once)
   ...))
```

Ils ne sont jamais déclenchés plus d'une fois par pas dans le temps. Ils émettront un avertissement lorsqu'ils recevront deux événements dans un même pas, sauf si ces événements sont équivalents.

- Les traducteurs définis en Scheme et utilisables à la fois dans le cadre d'un '`\layout`' et d'un '`\midi`' peuvent désormais être créés à l'aide de `make-translator`. Les exécutants définis en Scheme, qui ne peuvent s'utiliser de dans un '`\midi`' se créent à l'aide de `make-performer`. Ces macros fonctionnent tout comme la macro préexistante `make-engraver` qui permet de créer un graveur, uniquement utilisable dans un '`\layout`'.
- Les traducteurs définis en Scheme peuvent désormais définir un nouveau connecteur dénommé `pre-process-music`. Celui-ci sera appelé par tous les traducteurs, après tous les *listeners* mais préalablement aux connecteurs `process-music`. Ceci peut s'utiliser pour un traitement qui dépend de l'intégralité des événements entendus mais nécessite de définir des propriétés de contexte avant que d'autres traducteurs les lisent.
- Les nouvelles propriétés `show-horizontal-skylines` et `show-vertical-skylines` permettent d'afficher les lignes d'horizon d'un objet. Ces propriétés sont beaucoup plus flexibles que d'utiliser l'option `debug-skylines` dans la mesure où elles sont opérationnelles avec tous les objets graphiques. Bien que prévu à l'origine pour le débogage de LilyPond, ceci s'avère fort utile lorsque l'on tente de comprendre les décisions en matière d'espacement ou la modification de stencils en Scheme.

