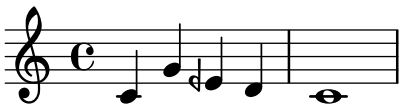Nouvelles fonctionnalités de la version 2.22 (depuis 2.20)

Nouveautés en matière de notation musicale

Améliorations de la représentation des hauteurs

- le noms des notes en quart de ton est désormais disponible dans toutes les langues de saisie.
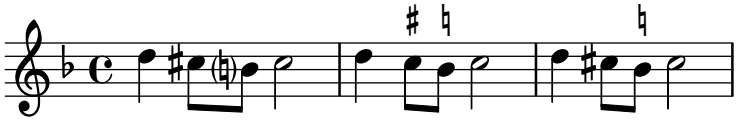
```
\language "català"
\relative { do'4 sol' miqb re do1 }
```



- Toutes les langues de saisie (instruction `\language`) peuvent se mentionner avec leur propre orthographe UTF-8 – autrement dit y compris avec des caractères spéciaux. Les dénominations `català` et `português` s'ajoutent désormais aux appellations originelles `catalan` et `portugues`.
- Définir `suggestAccidentals` à `'cautionary` a pour effet d'appliquer le style `AccidentalSuggestion` uniquement aux altérations accidentelles de précaution. Ceci permet de distinguer les accidentelles *facsimile* de celles à vocation éditoriale.
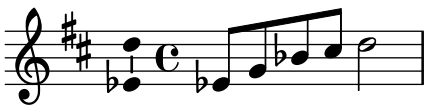
```
\relative {
  \key d \minor
  d''4 cis8 b? cis2
  \set suggestAccidentals = ##t
  d4 cis8 b? cis2
  \set suggestAccidentals = #'cautionary
  d4 cis8 b? cis2

}
```



- Ajout de la commande `\ambitusAfter`, qui permet de positionner l'ambitus à un endroit particulier.

```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```
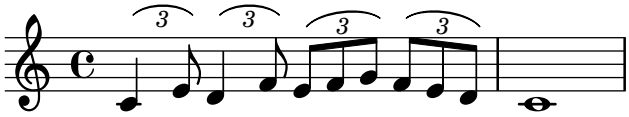


- La commande `\partcombine`, ainsi que les commandes, sous-routines et noms de propriété préfixées de `partCombine`, s'écrivent désormais avec un C majuscule, comme `\partCombine`, `\partCombineApart`, etc. La commande `\autochange` suit le même principe et se libelle désormais `\autoChange`.
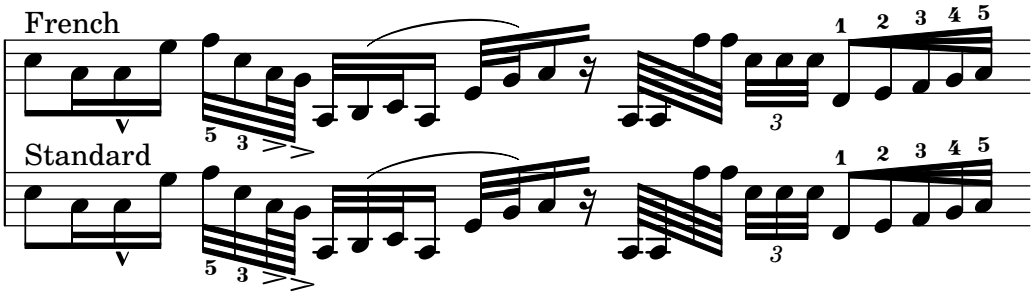
Améliorations en matière de rythme

- Les n-olets peuvent désormais s'indiquer à l'aide d'une liaison en lieu et place d'un crochet :

```
\relative {
  \tuplet 3/2 4 {
    \override TupletBracket.tuplet-slur = ##t
    c'4 e8 d4 f8
    \override TupletBracket.bracket-visibility = ##t
    e f g f e d
  } c1
}
```

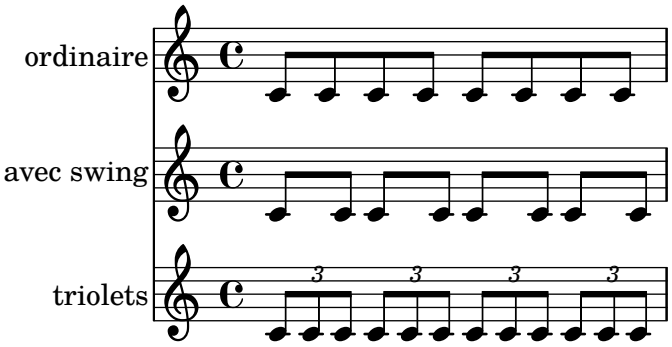


- Les ligatures à la française (`\override Stem.french-beaming = ##t`) se comportent à l'identique des ligatures standard (par défaut) dans toutes leurs composantes (position de la ligature et placement des articulations, doigtés, etc.) à ceci près que les hampes intermédiaires ne traversent pas le trait de ligature.

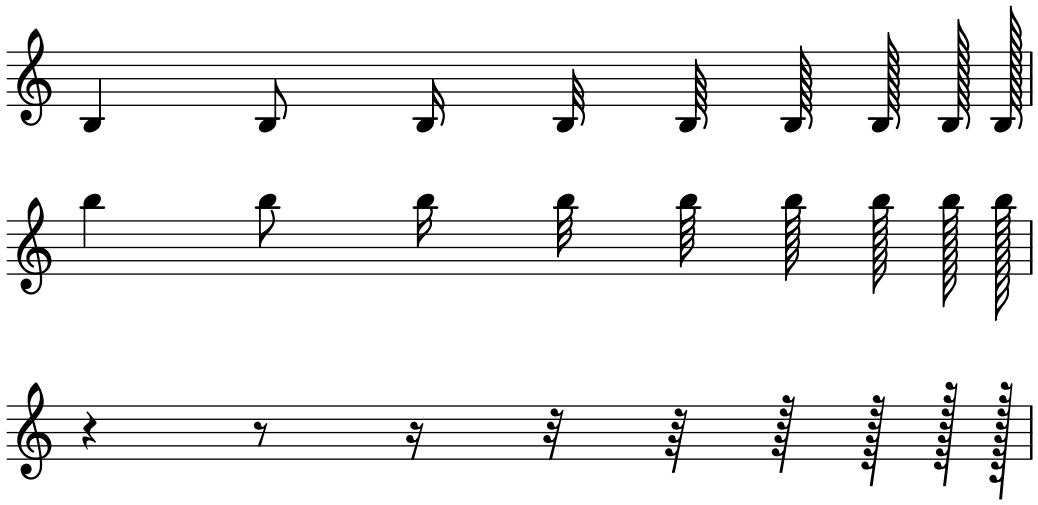


- Le swing et les motifs rythmiques irréguliers peuvent désormais s'appliquer à des expressions musicales composées de durées régulières qui peuvent s'utiliser pour rendre une interprétation inégale en MIDI.

```
\include "swing.ly"
<<
  \new Staff \with {
    instrumentName = "ordinaire"
  } \repeat unfold 8 c'8
  \new Staff \with {
    instrumentName = "avec swing"
  } \repeat unfold 4 \tripletFeel 8 { c'8 c' }
  \new Staff \with {
    instrumentName = "triolets"
  } \tuplet 3/2 4 \repeat unfold 12 c'8
>>
```
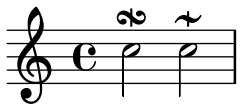


- Plusieurs glyphes, correspondant aux 256^{e}, 512^{e} et 1024^{e} de soupir, ainsi que les crochets respectifs, ont été ajoutés.



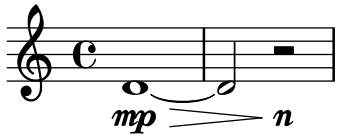
Améliorations en matière d'expressivité

- Ajout de deux nouveaux ornements.

```
{
  c''2\slashturn c''\haydnturn
}
```



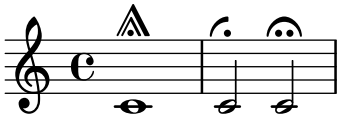
- Ajout de la commande de nuance `\n` pour *niente*.



Améliorations en matière de notation sur la portée

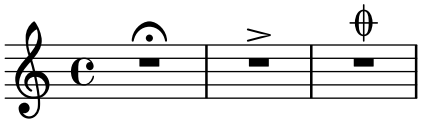
- Ajout d'un point d'arrêt ultracourt et de variantes à la Henze pour les points d'orgue plus ou moins longs.

```
{
  c'1\veryshortfermata
  c'2\henzeshortfermata c'\henzelongfermata
}
```



- Point d'orgue et autres articulations peuvent désormais s'ajouter directement à un silence multimesure. Par conséquent, l'utilisation de la commande `\fermataMarkup` est désapprouvée.

```
{
  R1\fermata
  R->
  R\coda
}
```

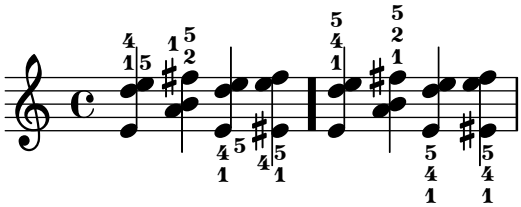


- `\compressFullBarRests` a été renommé en `\compressEmptyMeasures` afin d'éviter toute confusion avec `\compressMMRests`. Dans le même esprit, `\expandFullBarRests` devient `\expandEmptyMeasures`.

Améliorations en matière d'annotations éditoriales

- Le `New_fingering_engraver` prend désormais en charge la propriété `X-align-on-main-noteheads` de la `self-alignment-interface`. Dès lors qu'elle est activée (`##t`), tous les doigtés positionnés `up` ou `down` seront rangés sur une unique colonne alignée sur les têtes de note, du côté adéquat de la hampe :

```
{
  \set fingeringOrientations = #'(up)
  <e'-1 d''-4 e''-5>4 <a'-1 b'-2 fis''-5>
  \set fingeringOrientations = #'(down)
  <e'-1 d''-4 e''-5> <eis'-1 e''-4 fis''-5>
  \bar "."
  \override Fingering.X-align-on-main-noteheads = ##t
  \set fingeringOrientations = #'(up)
  <e'-1 d''-4 e''-5> <a'-1 b'-2 fis''-5>
  \set fingeringOrientations = #'(down)
  <e'-1 d''-4 e''-5> <eis'-1 e''-4 fis''-5>
}
```
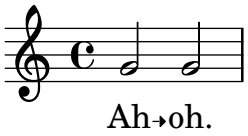


Nouveautés en matière de notation spécialisée

Améliorations pour la musique vocale

- Un changement graduel de voyelle (ou une consonne maintenue) peut s'indiquer en ajoutant un signe de transition entre les syllabes à l'aide de la commande `\vowelTransition`.

```
{ g'2 g' }
\addlyrics { Ah \vowelTransition oh. }
```



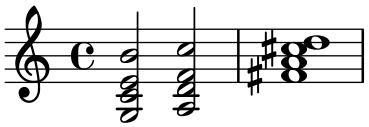
Améliorations pour les cordes frettées ou non

- Quelques diagrammes de fret pour ukulele ont été corrigés, et d'autres ont été ajoutés.

Améliorations pour la notation des accords

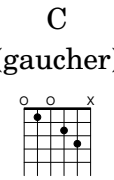
- Les accords peuvent être désormais automatiquement renversés, ou avoir l'une de leurs composantes sautée.

```
\chordmode {
  \dropNote 2 {
    c2:maj7 d:m7
  }
  \invertChords 1 d1:maj7
}
```



- Les fonctions historiques de nommage d'accord `banter-chord-names` et `jazz-chord-names` ont été supprimées de la base de code, tout comme les propriétés `chordNamesExceptionsFull` et `chordNamesExceptionsPartial`. Dans le cadre d'une réécriture, le *power chord* est désormais inclus dans les exceptions par défaut et sera imprimé correctement (annotation en exposant comme tous les accords) sans avoir besoin de bricolage – la commande prédéfinie `\powerChords` devient donc inutile. Les adeptes des fonctions historiques peuvent en trouver une copie (plus ou moins fonctionnelle) dans le morceau choisi `chord-names-alternative.ly`.
- Les diagrammes de fret peuvent désormais s'imprimer en version « gaucher » dès lors que `handedness` (sous-propriété de `fret-diagram-details`) aura été définie à `LEFT`.

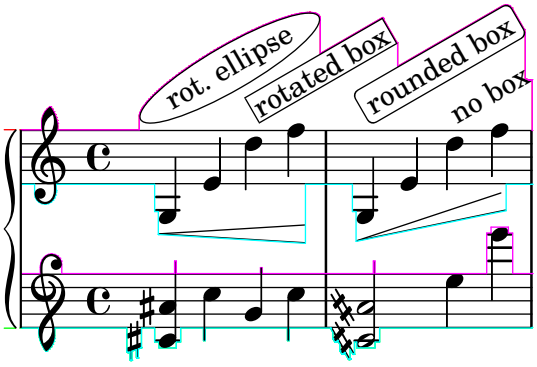
```
\markup
  \center-column {
    "C"
    "(gaucher)"
     \override #`(fret-diagram-details . ((handedness . ,LEFT)))
     \fret-diagram "6-x;5-3-3;4-2-2;3-o;2-1;1-o;"
  }
```



Nouveautés en matière d'entrée et sortie

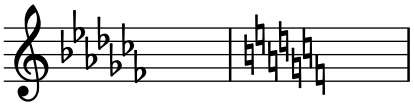
Améliorations pour les fichiers résultants

- Les lignes d'horizon (*skylines*) prennent désormais en compte la propriété `rotation` des objets de rendu. Faire pivoter un soufflet de crescendo en lui appliquant par exemple `\override Hairpin.rotation = #'(15 0 0)` aura une influence sur les lignes d'horizon et donc aidera à améliorer l'espacement.



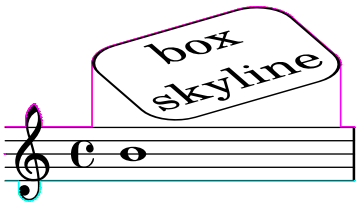
- Un léger décalage a été ajouté au glyphe de bécarre dont les extrémités se touchaient à l'occasion d'un changement d'armure.

```
{
  \omit Staff.TimeSignature
  \key ces \major s1 \key c \major s
}
```



- Les lignes d'horizon des boîtes reflètent désormais l'emprise réelle de la boîte, y compris les bords arrondis et la rotation.

```
#(ly:set-option 'debug-skylines #t)
{
  b'1^\markup \scale #'(2 . 1)
              \rotate #30
              \override #'(corner-radius . 3)
              \rounded-box \center-column { box skyline }
}
```



Améliorations en matière d'extraction de la musique

- La procédure `output-classic-framework` et l'option `-dclip-systems` sont désormais disponibles avec le moteur `SVG`.
- L'argument `-dcrop` a été ajouté, permettant de formater les sorties `SVG` et `PDF` sans marge ni saut de page.

Nouveautés en matière d'interfaces et fonctions internes

- Comme il avait été annoncé lors de la diffusion de la version 2.17.3, l'option `'relative-include` est désormais activée par défaut. Lorsqu'un fichier faisant l'objet d'une inclusion comporte lui-même une instruction `\include`, le chemin alors recherché sera relatif à ce fichier, non par rapport au répertoire du fichier principal. Il est possible de contrevenir à ce comportement en affectant `#f` à la directive `'relative-includes`, que ce soit en ligne de commande ou par l'utilisation d'un `ly:set-option` dans le fichier source.
- LilyPond pour Windows (MinGW) gère les noms de fichier Unicode à compter de Windows 10-1903.