

GNU LilyPond

Le système de gravure musicale

Manuel de l'utilisateur

L'équipe de développement de LilyPond

Copyright © 1999–2007 by the authors

The translation of the following copyright notice is provided for courtesy to non-English speakers, but only the notice in English legally counts.

La traduction de la notice de droits d'auteur ci-dessous vise à faciliter sa compréhension par le lecteur non anglophone, mais seule la notice en anglais a valeur légale.

Vous avez le droit de copier, distribuer et/ou modifier ce document selon les termes de la Licence GNU de documentation libre, version 1.1 ou tout autre version ultérieure publiée par la Free Software Foundation, “sans aucune section invariante”. Une copie de la licence est fournie à la section “Licence GNU de documentation libre”.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections. A copy of the license is included in the section entitled “GNU Free Documentation License”.

(Pour LilyPond version 2.11.31)

Table des matières

Préface	1
1 Introduction	2
1.1 Gravure	2
1.2 Gravure automatisée	3
1.3 Quels signes graver ?	5
1.4 Représentation de la musique	6
1.5 Exemples d'application	8
1.6 À propos de ce manuel	9
2 Tutoriel	12
2.1 Premiers pas	12
2.1.1 Compiler un fichier	12
2.1.2 Notation simple	13
2.1.3 Travailler sur des fichiers texte	17
2.1.4 Comment lire le tutoriel	18
2.2 Notation sur une seule portée	18
2.2.1 Hauteurs de note relatives	18
2.2.2 Altérations et armure	18
2.2.3 Liaisons	20
2.2.4 Articulations et nuances	21
2.2.5 Barres de ligature automatiques et manuelles	22
2.2.6 Commandes rythmiques avancées	23
2.3 Notes simultanées	24
2.3.1 Les expressions musicales en clair	24
2.3.2 Plusieurs portées	25
2.3.3 Double portée	27
2.3.4 Polyphonie sur une portée	27
2.3.5 Combinaison de notes en accords	28
2.4 Chansons	29
2.4.1 Imprimer des paroles	29
2.4.2 Partition d'une chanson	30
2.5 Dernières précisions	31
2.5.1 Numéro de version	31
2.5.2 Ajout de titres	31
2.5.3 Noms de note absolus	32
2.5.4 Organiser ses pièces avec des identificateurs	33
2.5.5 Après le tutoriel	34
2.5.6 Comment lire le manuel	34
3 Tout mettre en ordre	35
3.1 Étendre les modèles	35
3.2 Comment les fichiers LilyPond sont organisés	38
3.3 La partition est une unique expression musicale	39
3.4 Partitions d'orchestre	41

4	Travailler sur des projets LilyPond	44
4.1	Suggestions de saisie des fichiers LilyPond	44
4.1.1	Suggestions générales	44
4.1.2	Graver de la musique existante	45
4.1.3	Pour des projets d'envergure	45
4.2	Gagner du temps grâce aux identificateurs et fonctions	45
4.3	Feuilles de style	47
4.4	Mettre à jour d'anciens fichiers	51
4.5	Résolution de problèmes — tout remettre à plat	51
4.6	Exemples minimaux	52
5	Retoucher les résultats	53
5.1	Déplacer des objets	53
5.2	Corriger les collisions d'objets	55
5.3	Retouches courantes	56
5.4	Fichiers fournis avec LilyPond	58
5.5	Réduire le nombre de pages de la partition	58
5.6	Retouches avancées avec Scheme	60
5.7	Éviter les retouches ralentissant le traitement	61
6	Notation de base	62
6.1	Hauteurs	62
6.1.1	Hauteurs naturelles	62
6.1.2	Altérations	63
6.1.3	Altérations de précaution	64
6.1.4	Altérations microtonales	64
6.1.5	Noms de note dans d'autres langues	64
6.1.6	Octaves relatives	65
6.1.7	Vérification d'octave	66
6.1.8	Transposition	67
6.1.9	Silences	68
6.1.10	Silences invisibles	69
6.2	Rythme	69
6.2.1	Durées	69
6.2.2	Notes pointées	70
6.2.3	Nolets	71
6.2.4	Changement d'échelle des durées	73
6.2.5	Vérification des limites de mesure	73
6.2.6	Vérification du numéro de mesure	74
6.2.7	Découpage automatique des notes	74
6.3	Polyphonie	75
6.3.1	Accords	75
6.3.2	Hampes	75
6.3.3	Polyphonie basique	75
6.3.4	Instantiation explicite des voix	77
6.3.5	Résolution des collisions	79
6.4	Notation sur la portée	81
6.4.1	Clefs	81
6.4.2	Armure	82
6.4.3	Métrique	83
6.4.4	Mesures incomplètes	85
6.4.5	Barres de mesures	86
6.4.6	Musique sans métrique	87

6.4.7	Groupements de systèmes	88
6.4.8	Symbole de la portée	90
6.4.9	Saisir la musique en parallèle	91
6.5	Relier des groupes de notes	92
6.5.1	Liaisons de prolongation	92
6.5.2	Liaisons d'articulation	94
6.5.3	Liaisons de phrasé	95
6.5.4	Liaisons "laissez vibrer"	96
6.5.5	Barres de ligature automatiques	96
6.5.6	Barres de ligature manuelles	96
6.5.7	Notes d'ornement	98
6.6	Signes d'interprétation	101
6.6.1	Articulations	101
6.6.2	Doigtés	103
6.6.3	Nuances	104
6.6.4	Signes de respiration	107
6.6.5	Trilles	107
6.6.6	Glissando	108
6.6.7	Arpège	108
6.6.8	Chutes et sauts	110
6.7	Répétitions	110
6.7.1	Types de répétition	110
6.7.2	Syntaxe des répétitions	110
6.7.3	Répétitions et MIDI	112
6.7.4	Commandes de reprise manuelles	113
6.7.5	Répétitions en trémolo	114
6.7.6	Subdivision des trémolos	115
6.7.7	Répétitions de mesure	115
7	Notation spécifique	117
7.1	Musique pour piano	117
7.1.1	Changements de portée automatiques	117
7.1.2	Changements de portée manuels	117
7.1.3	Pédales	117
7.1.4	Lignes de changement de portée	117
7.1.5	Accords sur plusieurs portées	117
7.2	Les noms des accords	117
7.2.1	Introduction aux noms d'accords	117
7.2.2	Mode accords	117
7.2.3	Imprimer des noms d'accord	117
7.3	Musique vocale	117
7.3.1	Écrire des chants simples	117
7.3.2	Saisie des paroles	117
7.3.3	Aligner des paroles sur une mélodie	117
7.3.3.1	Durée automatique des syllabes	117
7.3.3.2	Une autre manière de saisir des paroles	118
7.3.3.3	Affecter plus d'une syllabe à une note	118
7.3.3.4	Plus d'une note pour une même syllabe	118
7.3.3.5	Prolongateurs et traits d'union	118
7.3.4	Utiliser des identificateurs pour gérer les paroles	118
7.3.5	Flexibilité dans le placement	118
7.3.5.1	Paroles sur plusieurs notes d'un mélisme	118
7.3.5.2	Paroles alternatives	118
7.3.5.3	Changer la voix associée à une ligne de paroles	118

7.3.5.4	Paroles indépendantes des notes	118
7.3.6	Espacement des paroles.....	118
7.3.7	Traitement avancé des couplets	118
7.3.7.1	Adding stanza numbers	118
7.3.7.2	Ajouter des nuances	118
7.3.7.3	Ajouter le nom des chanteurs	118
7.3.7.4	Printing stanzas at the end	118
7.3.7.5	Printing stanzas at the end in multiple columns	118
7.3.8	Ambitus	118
7.3.9	Autres éléments relatifs à la voix	118
7.4	Musique rythmique.....	119
7.4.1	Graver des lignes rythmiques	119
7.4.2	Notation des percussions	119
7.4.3	Portée de percussions.....	119
7.4.4	Notes fantômes.....	119
7.5	Guitare	119
7.5.1	Indications de numéro de corde.....	119
7.5.2	Les tablatures : notions élémentaires.....	119
7.5.3	Tablatures autres que pour la guitare.....	119
7.5.4	Tablatures pour banjo	119
7.5.5	Tablatures	119
7.5.6	Doigtés pour la main droite.....	119
7.5.7	Autres éléments relatifs à la guitare.....	119
7.6	Cornemuse.....	119
7.6.1	Définitions pour la cornemuse	119
7.6.2	Exemple pour la cornemuse.....	119
7.7	Notations anciennes.....	120
7.7.1	Têtes de note anciennes	120
7.7.2	Altérations anciennes.....	120
7.7.3	Silences anciens	120
7.7.4	Clefs anciennes	120
7.7.5	Crochets anciens	120
7.7.6	Métriques anciennes.....	120
7.7.7	Articulations anciennes	120
7.7.8	Guidons.....	120
7.7.9	Divisions	120
7.7.10	Ligatures.....	120
7.7.10.1	Ligatures mensurales	120
7.7.10.2	Grégorien et neumes ligaturés	120
7.7.11	Contextes du chant grégorien.....	120
7.7.12	Le contexte mensural.....	120
7.7.13	Altérations et musica ficta.....	120
7.7.14	Basse chiffrée.....	120
7.8	Autres notations spécifiques à des instruments.....	121
7.8.1	Harmoniques artificiels (cordes).....	121

8	Notation avancée	122
8.1	Texte	122
8.1.1	Commentaires textuels	122
8.1.2	Indications textuelles et lignes d'extension	122
8.1.3	Extensions de texte	122
8.1.4	Indications textuelles	122
8.1.5	Étiquette textuelle	122
8.1.6	Partitions enchâssées	122
8.1.7	Texte avec sauts de page	122
8.1.8	Vue d'ensemble des commandes d'indication textuelle	122
8.1.9	Vue d'ensemble des commandes de listes d'indications textuelles	130
8.1.10	Sélection de la fonte	130
8.1.11	Personnalisation des indications de nuance	130
8.2	Préparer un matériel d'orchestre	131
8.2.1	Silences multi-mesures	131
8.2.2	Indications métronomiques	131
8.2.3	Indications de repère	131
8.2.4	Numéros de mesure	131
8.2.5	Noms d'instrument	131
8.2.6	Instruments transpositeurs	131
8.2.7	Marques d'octavation	131
8.2.8	Différentes éditions à partir d'une même source	131
8.3	Musique orchestrale	131
8.3.1	Regroupement automatique de parties	131
8.3.2	Masquer des portées	131
8.3.3	Citer d'autres voix	131
8.3.4	Mettre en forme une citation	131
8.3.5	Alignement et cadences	131
8.4	Notation contemporaine	131
8.4.1	Notation polymétrique	132
8.4.2	Gestion du temps	132
8.4.3	Introduction à la notation proportionnelle	132
8.4.4	Clusters	132
8.4.5	Têtes de note spécifiques	132
8.4.6	Barres de ligature en soufflet	132
8.4.7	Improvisation	132
8.4.8	Spécifier la taille de la fonte musicale	132
8.5	Utilisation en milieu éducatif	132
8.5.1	Info-bulle	132
8.5.2	Papier à musique	132
8.5.3	Dictée à trous	132
8.5.4	Têtes de note à forme variable	132
8.5.5	Têtes de note avec nom de note	132
8.5.6	Crochets d'analyse	132
8.5.7	Colorer des objets	132
8.5.8	Parenthèses	132
8.5.9	Quadrillage temporel	133

9	Changer les réglages prédéfinis	134
9.1	Comportements automatiques	134
9.1.1	Altérations accidentelles automatiques	134
9.1.2	Définir les règles de ligatures automatiques	134
9.2	Contextes d'interprétation	134
9.2.1	Tout savoir sur les contextes	134
9.2.2	Créer un contexte	134
9.2.3	Changer à la volée les propriétés de contexte	134
9.2.4	Modifier les greffons de contexte	134
9.2.5	Affiner le rendu au sein des contextes	134
9.2.6	Modifier les réglages par défaut d'un contexte	134
9.2.7	Définir de nouveaux contextes	134
9.2.8	Aligner des contextes	134
9.2.9	Groupement vertical d'objets graphiques	134
9.3	La commande <code>\override</code>	134
9.3.1	Élaborer une retouche	134
9.3.2	Manipuler le guide de référence du programme	135
9.3.3	Interfaces de rendu	135
9.3.4	Déterminer la propriété de l'objet graphique (Grob)	135
9.3.5	Objets liés à la source	135
9.3.6	Utiliser du code Scheme au lieu de <code>\tweak</code>	135
9.3.7	<code>\set</code> ou <code>\override</code>	135
9.3.8	Retouches complexes	135
10	Objets non musicaux	136
10.1	Fichiers d'entrée	136
10.1.1	Introduction à la structure de fichier	136
10.1.2	Structure de fichier	136
10.1.3	Une expression musicale unique	136
10.1.4	Plusieurs partitions dans un même ouvrage	136
10.1.5	Extraire des fragments	136
10.1.6	Insérer des fichiers LilyPond	136
10.1.7	Codage du texte	136
10.2	Titres et entêtes	136
10.2.1	Créer des titres	136
10.2.2	Titres personnalisés	136
10.2.3	Référence de numéro de page	136
10.2.4	Table des matières	136
10.3	Sortie MIDI	136
10.3.1	Créer des fichiers MIDI	136
10.3.2	Le bloc MIDI	137
10.3.3	Noms d'instruments MIDI	137
10.4	Afficher de la notation au format LilyPond	137
10.5	Ignorer des passages de la partition	137

11	Gestion de l'espace	138
11.1	Du papier et des pages	138
11.1.1	Format de papier	138
11.1.2	Formater une page	138
11.2	Mise en forme de la musique	138
11.2.1	Définir la taille de la portée	138
11.2.2	Mise en forme de la partition	138
11.3	Montrer les espacements	138
11.4	Sauts	138
11.4.1	Saut de ligne	138
11.4.2	Saut de page	138
11.4.3	Optimisation des sauts de page	138
11.4.4	Optimisation des tournes	138
11.4.5	Minimisation des sauts de page	138
11.4.6	Sauts explicites	138
11.4.7	Recourir à une voix supplémentaire pour gérer les sauts	138
11.5	Espacement vertical	139
11.5.1	Espacement vertical au sein d'un système	139
11.5.2	Espacement vertical entre les systèmes	139
11.5.3	Positionnement explicite des portées et systèmes	139
11.5.4	Optimiser le remplissage avec un deuxième passage	139
11.5.5	Résolution des collisions verticales	139
11.6	Espacement horizontal	139
11.6.1	Généralités sur l'espacement horizontal	139
11.6.2	Changement d'espacement au cours de la partition	139
11.6.3	Modifier l'espacement horizontal	139
11.6.4	Longueur de ligne	139
11.6.5	Notation proportionnelle	139
12	Interfaces pour les programmeurs	140
12.1	Fonctions musicales	140
12.1.1	Aperçu des fonctions musicales	140
12.1.2	Fonctions de substitution simple	140
12.1.3	Fonctions de substitution par paire	140
12.1.4	De l'usage des mathématiques dans les fonctions	140
12.1.5	Fonctions fantômes	140
12.1.6	Fonctions sans argument	140
12.1.7	Liste des fonctions musicales prédéfinies	140
12.2	Interfaces de programmation	143
12.2.1	Variables d'entrée et Scheme	143
12.2.2	Représentation interne de la musique	143
12.3	Construire des fonctions complexes	143
12.3.1	Afficher des expressions musicales	143
12.3.2	Propriétés de la musique	143
12.3.3	Exemple : redoubler une note avec liaison	143
12.3.4	Exemple : ajouter une articulation à plusieurs notes	143
12.4	Interface de programmation des marqueurs de texte	143
12.4.1	Construire un marqueur en Scheme	143
12.4.2	Fonctionnement interne des marqueurs	143
12.4.3	Définition d'une nouvelle commande de marqueur	144
12.4.4	Définition d'une nouvelle commande de liste de marqueurs	144
12.5	Contextes pour programmeurs	144
12.5.1	Évaluation d'un contexte	144

12.5.2	Appliquer une fonction à tous les objets de mise en forme	144
12.6	Utiliser des procédures Scheme en tant que propriétés	144
Annexe A	Bibliographie	145
Annexe B	Tutoriel Scheme	146
Annexe C	Tables du manuel de notation	147
C.1	Table des noms d'accord	147
C.2	Instruments MIDI	147
C.3	Liste des couleurs	147
C.4	La fonte Feta	147
C.5	Styles de tête de note	147
Annexe D	Modèles	148
D.1	Portée unique	148
D.1.1	Notes seules	148
D.1.2	Notes et paroles	148
D.1.3	Notes et accords	148
D.1.4	Notes, paroles et accords	148
D.2	Modèles pour claviers	148
D.2.1	Piano seul	148
D.2.2	Chant et accompagnement	148
D.2.3	Piano et paroles entre les portées	148
D.2.4	Piano et nuances entre les portées	148
D.3	Quatuor à cordes	148
D.3.1	Quatuor à cordes	148
D.3.2	Parties pour quatuor à cordes	148
D.4	Ensemble vocal	148
D.4.1	Partition pour chœur à quatre voix mixtes	148
D.4.2	Partition pour chœur SATB avec réduction pour piano	148
D.4.3	Partition pour chœur SATB avec alignement des contextes	148
D.5	Exemples de notation ancienne	148
D.5.1	Transcription de musique mensurale	148
D.5.2	Transcription du grégorien	148
D.6	Symboles de jazz	148
D.7	Squelettes pour lilypond-book	149
D.7.1	LaTeX	149
D.7.2	Texinfo	149
Annexe E	Aide-mémoire	150
Annexe F	Licence GNU de documentation libre	154
F.0.1	SUPPLÉMENT : comment utiliser cette licence pour vos documents	159
Annexe G	Index des commandes LilyPond	160
Annexe H	Index de LilyPond	162

Préface

Ce doit être pendant une répétition de l'Orchestre des Jeunes d'Eindhoven, un jour de 1995, que Jan, du pupitre des altistes tordus, aborda Han-Wen, un corniste déjanté, pour lui parler d'un mirifique projet dans lequel il venait de se lancer. Il s'agissait d'un système automatisé de gravure musicale — le préprocesseur MPP pour MusiXTeX pour être précis. Han-Wen, qui voulait justement imprimer certaines parties d'une oeuvre, jeta un premier coup d'oeil à ce programme, et devint très vite accro. Le MPP s'avéra bientôt une voie sans issue. Après avoir ratiociné et échangé un grand nombre de courriels enflammés, Han-Wen lança le projet LilyPond en 1996, dans lequel, cette fois, ce fut au tour de Jan de se sentir entraîné.

De bien des points de vue, coder un programme informatique, c'est comme apprendre à jouer d'un instrument. Au début c'est sympa, on découvre comment ça fonctionne, et on aborde tout ce qu'on n'arrive pas encore à faire comme autant de défis. L'exaltation de la nouveauté s'estompant, on doit s'entraîner encore et encore. Les gammes, les études deviennent vite ennuyeuses, et peuvent, si l'on n'est pas encouragé par d'autres — professeurs, chefs ou public — en décourager plus d'un. Pourtant, pour peu que l'on persévère, l'instrument devient progressivement une partie de notre vie. Si certains jours en jouer semble naturel, c'est un vrai bonheur. Et si d'autres jours on ne peut tout simplement rien en tirer, on continue quand même à travailler, jour après jour.

De même, développer LilyPond peut être une tâche harassante. Certains jours, c'est un monceau de bugs duquel il faut se dépêtrer. Pourtant, il fait maintenant partie de notre vie, et nous nous accrochons. Notre principale motivation est sans doute que notre logiciel est véritablement utile aux gens. En flânant sur Internet, nous trouvons beaucoup de gens qui se servent de LilyPond, et réalisent des partitions très impressionnantes : c'est incroyable, mais en même temps très flatteur.

Les utilisateurs ne se contentent pas de nous encourager en utilisant notre logiciel ; nombre d'entre eux nous aident aussi en faisant des suggestions et en signalant des bogues. Aussi, nous voudrions remercier ici tous les utilisateurs qui nous ont signalé des bugs, ont fait des suggestions ou ont contribué d'une façon ou d'une autre au développement de LilyPond.

Jouer de la musique ou en graver, il y a là plus qu'une comparaison séduisante. Même si l'on s'amuse beaucoup en programmant tous ensemble, et qu'on éprouve une satisfaction profonde à aider les gens, au bout du compte, notre travail sur LilyPond est avant tout une manière d'exprimer notre amour sincère de la musique. Puisse-t-il vous aider à créer de nombreuses et belles oeuvres !

Han-Wen et Jan

Utrecht/Eindhoven, Pays-Bas, juillet 2002.

1 Introduction

1.1 Gravure

L'art de la typographie musicale se nomme la *gravure*. Ce terme est issu du processus traditionnel d'impression musicale. Il y a seulement quelques dizaines d'années, on faisait les partitions en coupant et en embossant une plaque de zinc ou d'étain en image miroir. Cette plaque était ensuite encrée, les dépressions créées par les creux et les bosses retenant l'encre. Une image était formée en pressant du papier sur la plaque. La découpe et l'embossage étaient entièrement faits à la main. Il était pénible d'appliquer une correction, quand celle-ci n'était pas impossible, la gravure devait donc être parfaite du premier coup. La gravure demandait une qualification hautement spécialisée : un artisan devait accomplir environ cinq ans de formation avant de mériter le titre de maître graveur, et il lui fallait cinq années d'expérience supplémentaires pour devenir vraiment habile.

De nos jours, toutes les partitions récentes sont produites avec des ordinateurs. Ceci a des avantages évidents : le coût des impressions a diminué, et le travail d'éditeur peut être envoyé par courriel. Malheureusement, l'utilisation dominante des ordinateurs a également diminué la qualité graphique des partitions. L'impression informatisée leur donne un aspect fade et mécanique qui les rend désagréables à jouer.

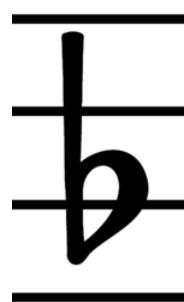
Les images ci-dessous illustrent la différence entre la gravure traditionnelle et l'impression typique par ordinateur, et la troisième image montre comment LilyPond mime l'aspect traditionnel. L'image de gauche est une numérisation d'un symbole bémol d'une édition publiée en 2000. Celle du centre montre un bémol d'une gravure à la main de l'édition Bärenreiter de la même musique. L'image de gauche illustre des défauts typiques de l'impression informatique : les lignes de portée sont minces, l'épaisseur de trait du bémol est la même que les lignes fines, et il y a un aspect rigide avec des angles pointus. Par contraste, le bémol Bärenreiter possède un aspect gras et arrondi, presque voluptueux. Notre symbole bémol est créé, entre autres, à partir de celui-là. Il est arrondi, et son épaisseur de trait s'harmonise avec nos lignes de portée, lesquelles sont également plus épaisses que celles de l'édition informatique.



Henle (2000)



Bärenreiter (1950)



Fonte Feta de LilyPond (2003)

En matière d'espacement, la répartition de l'espace devrait refléter les durées entre les notes. Cependant, beaucoup de partitions modernes se contentent des durées avec une précision mathématique, ce qui mène à de mauvais résultats. Dans l'exemple suivant, un motif est imprimé deux fois : une fois en utilisant un espacement mathématique exact, et une autre fois avec des corrections. Pouvez-vous les repérer ?





L'extrait n'utilise que des notes de même durée ; l'espacement devrait le refléter. Malheureusement, notre œil nous trompe quelque peu ; il ne se contente pas de remarquer la distance entre les têtes de notes, il prend en compte également la distance entre les hampes consécutives. Ainsi, par compensation, les notes avec une combinaison « hampe vers le haut »/« hampe vers le bas » doivent être éloignées l'une de l'autre, et les notes avec une combinaison « hampe vers le bas »/« hampe vers le haut » rapprochées, le tout dépendant de la position verticale des notes. Les deux premières mesures sont imprimées avec cette correction, les deux suivantes sans. Les notes dans les deux dernières mesures forment des blocs de notes « hampe vers le bas »/« hampe vers le haut ».

Les musiciens sont généralement plus absorbés par l'exécution que par l'étude de l'aspect graphique d'une partition, donc discuter sur les détails typographiques peut paraître peu important. Il n'en est rien. Dans de longues pièces avec des rythmes monotones, les corrections d'espacement engendrent de subtiles variations dans la mise en forme de chaque ligne, donnant à chacune une signature visuelle distincte. Sans cette signature, toutes les lignes auraient le même aspect, et ressembleraient à un labyrinthe. Si un musicien regarde ailleurs un instant ou se déconcentre momentanément, il peut avoir du mal à se retrouver sur la page.

De même, l'aspect robuste des symboles sur d'épaisses lignes de portée ressort mieux quand la partition est éloignée du lecteur, comme sur un pupitre par exemple. Une organisation minutieuse des espaces vides permet de minimiser l'espace qu'occupe la musique, tout en évitant que les symboles s'amassent les uns contre les autres. Le résultat permet de réduire le nombre de pages à tourner, ce qui est un grand avantage.

Ceci est une caractéristique commune à toute typographie. La disposition doit être belle, non seulement pour des raisons esthétiques, mais également pour l'aide apportée au lecteur dans la tâche qu'il doit accomplir. Pour du matériel d'exécution comme les partitions de musique, cela prend une double importance : les musiciens ont une quantité limitée d'attention. Moins ils en ont besoin pour lire, plus ils peuvent se concentrer sur la musique elle-même. Autrement dit, une meilleure typographie permet une meilleure interprétation.

Ces exemples démontrent que la typographie musicale est un art subtil et complexe, et que la produire demande une expertise considérable, que les musiciens n'ont généralement pas. LilyPond représente notre effort pour apporter l'excellence graphique de la gravure à la main à l'ère de l'ordinateur, et la rendre accessible à tous les musiciens. Nous avons conçu nos algorithmes, fontes et paramètres de programme pour retrouver la qualité d'édition des anciennes partitions que nous aimons tant lire et jouer.

1.2 Gravure automatisée

Comment pouvons-nous implémenter la typographie ? Si les artisans ont besoin de plus de dix ans pour devenir de vrais maîtres, comment nous, simples programmeurs, pourrions-nous jamais écrire un programme pour faire leur travail ?

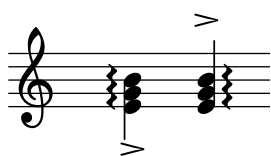
La réponse est : nous ne le pouvons pas. La typographie se base sur le jugement visuel humain, donc les humains ne peuvent pas être complètement remplacés. Si LilyPond arrive à résoudre la plupart des situations correctement, ce sera déjà une grande avancée sur les logiciels existants. Les autres situations peuvent être résolues à la main. Au fil des ans, le logiciel peut être affiné pour faire de plus en plus de choses automatiquement, pour que les ajustements manuels soient de moins en moins nécessaires.

Quand nous avons commencé, nous avons écrit le programme Lilypond entièrement dans le langage de programmation C++ ; les fonctions du programme étaient figées par les développeurs. Ceci s'est avéré insatisfaisant pour plusieurs raisons :

- Quand Lilypond fait des erreurs, les utilisateurs ont besoin de contredire les décisions de formatage. Les utilisateurs doivent donc avoir accès au moteur de formatage. Par conséquent, les règles et les propriétés ne peuvent pas être fixées par nous au moment de la compilation, mais doivent être accessibles aux utilisateurs au moment de l'exécution.
- La gravure est une question de jugement visuel, et donc de goût. Aussi bien informés que nous le sommes, les utilisateurs peuvent être en désaccord avec nos décisions personnelles. Par conséquent, les définitions du modèle typographique doivent également être accessibles à l'utilisateur.
- Enfin, nous affinons continuellement les algorithmes de formatage, donc nous avons besoin d'une approche souple des règles. Le langage C++ oblige à une certaine méthode de groupage des règles qui ne convient pas bien au fonctionnement de la notation musicale.

Ces problèmes ont été résolus en intégrant un interpréteur pour le langage de programmation Scheme, et en réécrivant des parties de LilyPond en Scheme. L'architecture actuelle de formatage est construite autour de la notion d'objets graphiques, décrits par des fonctions et des variables Scheme. Cette architecture comprend les règles de formatage, le style typographique, et des décisions individuelles de formatage. L'utilisateur a un accès direct à la plupart de ces contrôles.

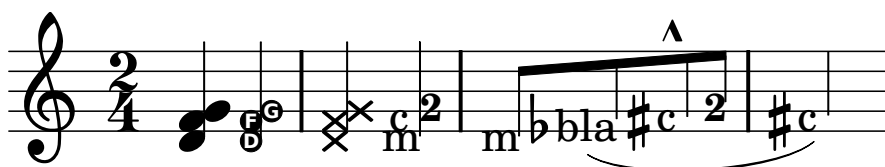
Les variables Scheme contrôlent les décisions de mise en page. Par exemple, beaucoup d'objets graphiques ont une variable de direction qui encode le choix entre haut et bas (ou gauche et droite). Vous pouvez voir ici deux accords, avec des accents, et des arpèges. Dans le premier accord, les objets graphiques sont tous dirigés vers le bas (ou la gauche). Dans le second accord ils sont tous dirigés vers le haut (droite).



Le processus de formatage d'une partition consiste à lire et écrire les variables d'objets graphiques. Certaines variables ont une valeur prédéfinie. Par exemple, l'épaisseur d'un grand nombre de lignes – une caractéristique du style typographique – est une variable avec une valeur prédéfinie. Vous êtes libres d'altérer cette valeur, ce qui vous donne une partition avec une impression typographique différente.



Les règles de formatage ont aussi des variables prédéfinies : chaque objet possède des variables contenant des procédures. Ces procédures exécutent le formatage, et en les substituant par d'autres, nous pouvons changer l'apparence des objets. Dans l'exemple suivant, la règle du choix de têtes de notes est changée au cours de l'extrait de musique.



1.3 Quels signes graver ?

Le processus de formatage décide où placer les symboles. Cependant, cela ne peut être fait qu'à partir du moment où il a été décidé *quels* symboles doivent être imprimés, c'est-à-dire quelle notation utiliser.

La notation musicale usuelle est un système d'écriture qui a évolué à travers les dix derniers siècles. La forme qui est aujourd'hui communément utilisée date du début de la Renaissance. Bien que la forme basique — les têtes de notes sur une portée de cinq lignes — n'a pas changé, les détails continuent d'évoluer pour exprimer les innovations de la notation contemporaine. Par conséquent, elle comprend quelque 500 ans de musique, avec des applications allant des mélodies monodiques à de monstrueux contrepoints pour grand orchestre.

Comment pouvons nous appréhender un tel monstre à plusieurs têtes, et le confiner dans l'espace réduit d'un programme informatique ? Notre solution consiste à diviser le problème de la notation — par opposition à la gravure, ou typographie — en morceaux digestes et programmables : chaque type de symbole est géré par un module séparé, couramment appelé greffon¹. Chaque greffon est entièrement modulaire et indépendant, et donc peut être développé et amélioré séparément. De tels greffons sont nommés **graveurs**², par analogie avec les artisans qui traduisent les idées musicales en symboles graphiques.

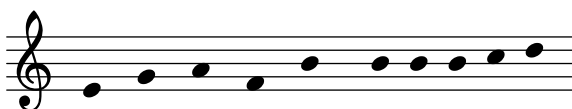
Dans l'exemple suivant, voyons comment nous commençons avec un greffon pour les têtes de notes, le graveur de têtes de note (`Note_heads_engraver`) :



Ensuite, le graveur du symbole de portée (`Staff_symbol_engraver`) ajoute la portée



le graveur de clef (`Clef_engraver`) définit un point de référence pour la portée



et le graveur de hampes (`Stem_engraver`) ajoute les hampes :



Le graveur de hampe est notifié de chaque tête de note qui survient. Chaque fois qu'une tête de note — plusieurs pour un accord — est rencontrée, un objet hampe est créé et connecté à la tête de note. En ajoutant des graveurs pour les barres de ligature, les liaisons, les accents, les altérations accidentelles, les barres de mesure, la métrique, et les armures, nous obtenons un jeu de notation complet.



¹ traduction de l'anglais *plug-in*.

² **engravers** en anglais.

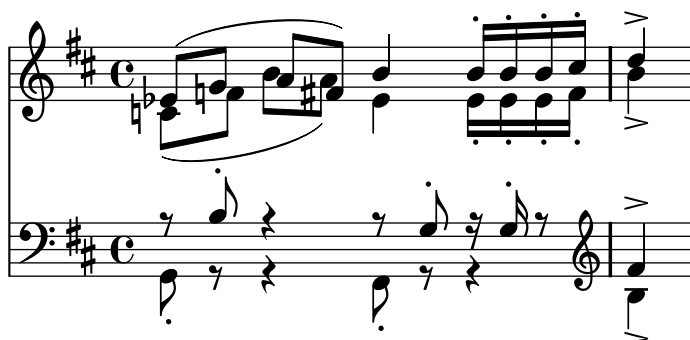
Ce système fonctionne bien pour de la musique monodique, mais qu'en est-il de la polyphonie ? En notation polyphonique, plusieurs voix peuvent partager une portée.



Dans cette situation, la portée et les altérations accidentelles sont partagées, mais les hampes, liaisons etc., sont spécifiques à chaque voix. Par conséquent, les graveurs doivent être groupés. Les graveurs des têtes de notes, hampes, liaisons etc., vont dans un groupe appelé « contexte de Voix »³, alors que les graveurs des clés, altérations accidentelles, barres de mesure etc., vont dans un groupe appelé « contexte de Portée ». Dans le cas de la polyphonie, un seul contexte de Portée contient plusieurs contextes de Voix. De même, plusieurs contextes de Portée peuvent être inclus dans un seul contexte de Partition. Le contexte de Partition est le contexte de notation de plus haut niveau.

Voir aussi

Program reference: `Contexts`.



1.4 Représentation de la musique

Idéalement, le format d'entrée pour n'importe quel système de formatage est une description abstraite du contenu. Dans ce cas-ci, ce serait la musique elle-même. Cela pose un formidable problème : comment pouvons-nous définir ce que la musique est réellement ? Plutôt que d'essayer de trouver une réponse, nous avons renversé la question. Nous écrivons un logiciel capable de produire de la musique écrite, et adaptons le format pour atteindre la plus grande concision possible. Quand le format ne peut plus être simplifié, il nous reste par définition le contenu lui-même. Notre logiciel sert de définition formelle d'un document de musique.

La syntaxe est également l'interface utilisateur pour LilyPond, par conséquent il est facile de saisir

```
c'4 d'8
```

c'est-à-dire un do central noire et, juste au-dessus un ré croche



³ 'Voice context' en anglais, 'Voice' commence par une majuscule comme tous les noms de contexte dans le programme LilyPond.

Sur une échelle microscopique, une telle syntaxe est facile à utiliser. A plus grande échelle, la syntaxe a besoin aussi de structure. Comment serait-il possible autrement de rentrer des pièces complexes comme des symphonies ou des opéras ? La structure est formée par le concept d'expression musicale : en combinant de petits fragments de musique pour en former de plus grands, on peut exprimer de la musique plus complexe. Par exemple

`c4`



Des accord peuvent être construits avec `<<` et `>>` autour des notes.

`<<c4 d4 e4>>`



Cette expression est mise dans une séquence grâce à l'encadrement par des accolades `{ ... }`

`{ f4 <<c4 d4 e4>> }`



Ceci est également une expression, et peut donc encore une fois être combinée avec d'autres expressions simultanées (une blanche) en utilisant `<<`, `\`, et `>>`

`<< g2 \ { f4 <<c4 d4 e4>> } >>`



De telles structures récursives peuvent être spécifiées formellement et de manière ordonnée dans une grammaire indépendante de tout contexte. Le code d'analyse est aussi générée à partir de cette grammaire. Autrement dit, la syntaxe de LilyPond est définie clairement et sans ambiguïté.

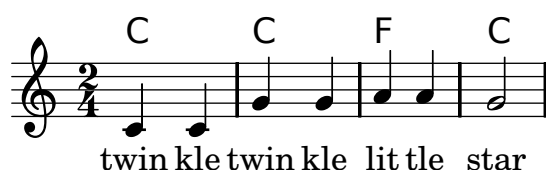
L'interface utilisateur et la syntaxe sont ce que les gens voient et manipulent le plus. Elles sont en partie une affaire de goût, et aussi sujettes à beaucoup de discussions. Même si ces discussions sur les goûts ont leur mérite, elles ne sont pas très productives. D'un point de vue plus large sur LilyPond, l'importance de la syntaxe est minime : il est facile d'inventer une syntaxe concise, alors qu'écrire un code de formatage décent est beaucoup plus difficile. Ceci est également illustré par le nombre de lignes de codes pour les composants respectifs : l'analyse et la représentation constituent moins de 10% du code source.

1.5 Exemples d'application

Nous avons conçu LilyPond comme une expérimentation visant à concentrer l'art de la gravure musicale dans un logiciel. Grâce à tout ce dur labeur, le programme peut maintenant être utilisé pour accomplir des travaux utiles. L'application la plus simple est d'imprimer des notes :



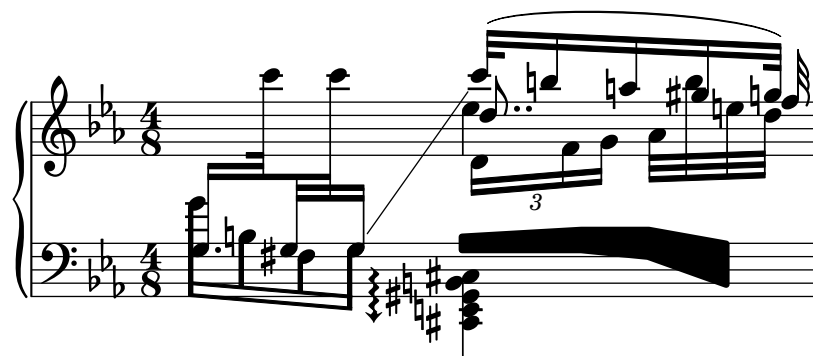
En ajoutant des noms d'accords et des paroles, nous obtenons une partition de chanson :



La notation polyphonique et la musique pour piano peuvent également être générées. L'exemple suivant associe quelques constructions plus exotiques :

Screech and boink Random complex notation

Han-Wen Nienhuys



Les extraits exposés ici ont tous été écrits à la main, mais ce n'est pas une obligation. Puisque le moteur de formatage est en grande partie automatique, il peut servir de sortie pour d'autres programmes qui manipulent la musique. Par exemple, il peut être utilisé pour convertir des bases de données d'extraits musicaux en images pour des sites Internet et des présentations multimédias.

Ce manuel montre également une application : le format d'entrée est du texte, et peut donc facilement être intégré dans d'autres formats basés sur le texte comme \LaTeX , HTML, ou dans le cas de ce manuel, Texinfo. À l'aide d'un programme spécial, les extraits de code peuvent être remplacés par des images de musiques dans les fichiers de sortie PDF ou HTML. Cela donne la possibilité de mélanger de la musique et du texte dans les documents.

1.6 À propos de ce manuel

Le manuel est divisé en quatre livres.

Manuel d'apprentissage

Ce livre explique comment débiter avec LilyPond, et expose de manière simple quelques concepts clés.

- *Chapitre 2 [Tutoriel], page 12* propose une introduction en douceur à la typographie musicale. Les utilisateurs débutants sont invités à commencer ici.
- *Chapitre 3 [Tout mettre en ordre], page 35* explique des concepts généraux du format de fichier ly. Si vous n'êtes pas certain de l'endroit où placer une commande, lisez ce chapitre !
- *Chapitre 4 [Travailler sur des projets LilyPond], page 44* montre des utilisations pratiques de LilyPond et donne des conseils afin d'éviter les problèmes les plus courants.
- *Chapitre 5 [Retoucher les résultats], page 53* est une introduction aux retouches de gravure avec LilyPond.

Manuel de référence

Ce livre détaille toutes les commandes LilyPond produisant une notation musicale. La lecture de cet ouvrage requiert une bonne compréhension des concepts exposés dans le manuel d'apprentissage.

- *Chapitre 6 [Notation de base], page 62* traite de sujets groupés par type de notation. Cette section détaille la notation de base, qui sera utile dans la plupart des projets de partition.
- *Chapitre 7 [Notation spécifique], page 117* traite de sujets groupés par type de notation. Cette section détaille des notations spéciales qui ne seront utiles que pour des types particuliers d'instruments ou la voix.
- *Chapitre 8 [Notation avancée], page 122* traite de sujets groupés par type de notation. Cette section donne des précisions à propos de notations compliquées et inhabituelles.
- *Chapitre 9 [Changer les réglages prédéfinis], page 134* explique comment ajuster finement la mise en page.
- *Chapitre 10 [Objets non musicaux], page 136* traite de sorties non musicales comme les titres, les mouvements multiples, et la sélection des instruments MIDI.
- *Chapitre 11 [Gestion de l'espace], page 138* traite de sujets affectant la sortie globale, comme sélectionner la taille de papier ou spécifier les sauts de page.
- *Chapitre 12 [Interfaces pour les programmeurs], page 140* explique comment créer des fonctions de musique.

Utilisation du programme

Ce livre explique comment exécuter le programme et comment intégrer les partitions LilyPond dans d'autres programmes.

- *program usage manual, Running LilyPond* montre comment lancer LilyPond et ses programmes auxiliaires. De plus, cette section explique comment mettre à jour des fichiers source écrits pour d'anciennes versions de LilyPond.
- *program usage manual, LilyPond-book* explique comment créer des documents intégrant des exemples musicaux, comme ce manuel.
- *program usage manual, Converting from other formats* explique comment exécuter les programmes de conversion. Ces programmes sont livrés avec le paquetage LilyPond, et convertissent une variété de formats musicaux vers le format .ly.

Annexes

Ce livre contient des tables de référence pratiques.

- *Annexe A [Bibliographie], page 145* contient un choix de livres de référence utiles pour ceux qui veulent en savoir plus sur la notation et la gravure.
- Le *Annexe B [Tutoriel Scheme], page 146* propose une courte introduction à Scheme, le langage de programmation utilisé dans les fonctions de musique.
- *Annexe C [Tables du manuel de notation], page 147* sont un ensemble de tableaux montrant les noms d'accord, les instruments MIDI, les noms de couleur, et la police Feta.
- Les *Annexe D [Modèles], page 148* de pièces LilyPond. Copiez et collez un modèle dans un fichier, ajoutez les notes, et c'est prêt !
- La *Annexe E [Aide-mémoire], page 150* est une référence pratique des commandes LilyPond les plus courantes.
- L' *Annexe G [Index des commandes LilyPond], page 160* est un index de toutes les `\commandes` LilyPond.
- L' *Annexe H [Index de LilyPond], page 162* est un index complet.

Autre documentation

D'autres documents constituent de précieuses sources d'information.

Le glossaire de musique explique les termes musicaux, et inclut leur traduction dans diverses langues. C'est un document séparé, disponible aux formats HTML et PDF. Si vous n'êtes pas familier avec la notation musicale ou la terminologie, il est conseillé de consulter le glossaire, notamment pour les parties non encore traduites de la documentation.

- Les Exemples de code sont une vaste sélection de petits exemples montrant des trucs, astuces et fonctionnalités particulières de LilyPond. La plupart de ces exemples sont également disponibles sur le [LilyPond Snippet Repository](#). Ce site Web possède également le manuel de LilyPond avec une fonctionnalité de recherche.
- La référence du programme est un ensemble de pages HTML étroitement liées entre elles, qui documente les moindres petits détails de chaque classe, objet et fonction de LilyPond. Cette documentation est produite directement à partir des définitions de formatage utilisées. Presque toutes les fonctions de formatage utilisées en interne sont directement disponibles pour l'utilisateur. Par exemple, toutes les variables qui contrôlent les épaisseurs, les distances etc., peuvent être modifiées dans les fichiers d'entrée. Il y a un grand nombre d'options de formatage, et elles sont toutes décrites dans ce document. Chaque section du manuel de notation a une section **Voir aussi** qui renvoie à la documentation générée automatiquement. Dans la documentation au format HTML, ces sous-sections ont des liens cliquables.

Lorsque vous serez un utilisateur expérimenté, vous pourrez consulter le manuel comme une référence : il y a un index complet⁴, mais le manuel est aussi disponible en une seule grande page, ce qui facilite la recherche avec la fonction de recherche de votre navigateur.

Dans tous les documents HTML qui incluent des fragments musicaux, le code Lilypond utilisé pour produire l'image peut être vu en cliquant l'image.

L'emplacement des fichiers de documentation mentionnés ici peut varier d'un système à l'autre. De temps en temps, ce manuel fait référence aux fichiers d'exemple et d'initialisation. Tout au long de ce manuel, nous donnons les emplacements des fichiers d'entrée relativement au répertoire racine de l'archive source. Par exemple, `'input/test/bla.ly'` peut référer au fichier `'lilypond2.x.y/input/test/bla.ly'`. Dans les paquets binaires pour les plateformes Unix, la

⁴ Si vous cherchez quelque chose sans le trouver dans la documentation, c'est un bogue. Dans ce cas, merci d'envoyer un rapport de bogue.

documentation et les exemples se trouvent généralement sous `‘/usr/share/doc/lilypond/’`. Les fichiers d’initialisation, par exemple `‘scm/lily.scm’`, ou `‘ly/engraver-init.ly’`, se trouvent généralement dans le répertoire `‘/usr/share/lilypond/’`.

Pour finir, ce manuel et les autres sont disponibles en ligne, à la fois aux formats PDF et HTML, à partir du site Web, accessible à l’adresse <http://www.lilypond.org/>.

2 Tutoriel

Ce tutoriel commence par une introduction au langage musical utilisé par LilyPond, qui vous permettra de faire fonctionner le logiciel pour produire une partition. Après ce premier contact, nous verrons comment créer des partitions utilisant une notation musicale courante.

2.1 Premiers pas

Cette section présente sommairement la façon de travailler avec LilyPond.

2.1.1 Compiler un fichier

Le premier exemple montre comment débiter avec LilyPond. Pour créer une partition, on écrit un fichier de texte qui décrit la notation musicale. Par exemple, si l'on écrit

```
{
  c' e' g' e'
}
```

le résultat ressemblera à



Il est aussi possible d'utiliser les noms de notes français 'do re mi fa sol la si', en insérant au début du fichier la ligne `\include "italiano.ly"`

Attention : tout extrait de code LilyPond doit être entouré d'une {paire d'accolades}. De plus, pour éviter toute ambiguïté, il est préférable d'entourer les accolades par des espaces ou retours à la ligne. Bien que certains exemples de ce manuel ne comportent pas d'accolades, ne les oubliez pas dans vos partitions !

De plus, LilyPond est sensible à la casse. `{ c d e }` est un code valide ; `{ C D E }` produira un message d'erreur.

Saisir la musique et visualiser la partition produite

Dans cette section nous expliquerons quelles commandes exécuter et comment voir ou imprimer le résultat de LilyPond.

MacOS X

Si vous double-cliquez sur `LilyPond.app`, un fichier d'exemple s'ouvrira. Sauvegardez-le, par exemple, sous `test.ly` sur votre bureau, et traitez-le ensuite avec la commande du menu `'Compile > Typeset File'`. Le fichier PDF résultant sera alors affiché sur votre écran.

Notez que le premier démarrage peut prendre une minute ou deux, car toutes les polices système doivent être d'abord analysées.

À l'avenir, vous aurez certainement recours aux commandes « Nouveau » ou « Ouvrir ». Vous devez enregistrer votre fichier avant de lancer la création de la partition. Si une erreur advient pendant le traitement, vous la trouverez dans la fenêtre « log ».

Windows

Sous Windows, lorsque vous double-cliquez sur l'icône LilyPond qui se trouve sur le Bureau, un fichier d'exemple s'ouvre dans un simple éditeur de texte. Enregistrez-le, par exemple en tant que `test.ly` sur votre Bureau, puis double-cliquez sur son icône (qui montre une note de musique)

pour le traiter. Après quelques secondes, vous obtiendrez un fichier ‘**test.pdf**’ sur votre Bureau, fichier que vous pourrez ouvrir pour voir la partition imprimée. Une autre méthode pour lancer le traitement du fichier ‘**test.ly**’ est de le glisser avec votre souris sur l’icône de LilyPond.

Pour modifier un fichier ‘**.ly**’ existant, faites un clic droit dessus et sélectionnez « Éditer la source ». Pour partir d’un fichier vide, lancez l’éditeur en ouvrant un fichier existant et utilisez la commande « New » du menu « File ».

En double-cliquant sur le fichier, vous obtiendrez, en plus du fichier PDF, un fichier ‘**.log**’ qui récapitule les opérations que LilyPond a effectuées sur votre fichier. Si une erreur advient, c’est ce fichier qu’il vous faudra étudier.

Notez qu’il existe d’autres éditeurs de texte, certains disposant d’un meilleur support pour LilyPond ; reportez-vous à program usage manual, **Editor support**.

Unix

Commencez par ouvrir une fenêtre de terminal et un éditeur de texte. Par exemple, vous pouvez ouvrir un xterm et exécuter **joe**.¹ Dans votre éditeur, entrez le texte suivant et sauvegardez le fichier sous ‘**test.ly**’

```
{
  c' e' g' e'
}
```

Pour traiter ‘**test.ly**’, procédez comme ceci :

```
lilypond test.ly
```

Vous verrez quelque chose ressemblant à :

```
lilypond test.ly
GNU LilyPond 2.10.0
Processing `test.ly'
Parsing...
Interpreting music... [1]
Preprocessing graphical objects...
Calculating line breaks... [2]
Layout output to `test.ps'...
Converting to `test.pdf'...
```

Suivant votre installation, ces messages peuvent être traduits.

De tout cela résulte un fichier ‘**test.pdf**’, que vous pouvez imprimer ou visualiser avec les outils standards de votre système d’exploitation.²

2.1.2 Notation simple

Il y a certains éléments graphiques que LilyPond ajoute automatiquement. Dans l’exemple suivant, nous n’avons fourni que quatre hauteurs, mais LilyPond a ajouté une clé, un chiffre de mesure et du rythme.

```
{
  c' e' g' e'
}
```

¹ Il existe des fichiers de macros pour les fans de VIM et un **LilyPond-mode** pour les fans d’Emacs. S’ils ne sont pas encore installés, consultez le fichier ‘**INSTALL.txt**’. L’outil d’édition le plus facile d’utilisation est de loin ‘**LilyPondTool**’. Vous trouverez plus d’informations dans program usage manual, **Editor support**.

² Si votre système ne dispose pas des outils nécessaires, vous pouvez essayer **Ghostsript**, un programme pour afficher et imprimer librement les fichiers PDF et PostScript.



Ce comportement peut être modifié, mais dans bien des cas ces attributions automatiques s'avèrent utiles.

Hauteurs

Le moyen le plus simple d'entrer des notes est d'utiliser le mode `\relative`. Avec ce mode, l'intervalle entre la note et celle qui la précède est supposé inférieur ou égal à une quarte. Commençons par entrer la partition la plus élémentaire qui soit, une *gamme*.

```
\relative c' {
  c d e f
  g a b c
}
```



La note de départ est **do central**. Chacune des notes qui suivent est à moins d'une quarte de la note précédente — en d'autres termes, le premier 'c' est le do central, entre la clé de sol et la clé de sa, puis est suivi par le ré le plus proche, et ainsi de suite. On peut bien sûr créer des mélodies d'intervalles plus étendus :

```
\relative c' {
  d f a g
  c b f d
}
```



Comme vous l'aurez remarqué, cet exemple ne commence plus sur le do du milieu. La première note — le 'd' — est le ré qui en est le plus proche.

Pour ajouter des intervalles supérieurs à une quarte, il suffit d'indiquer si la note est à l'octave supérieure ou inférieure, en ajoutant respectivement une apostrophe ' ou une virgule , au nom de la note.

```
\relative c'' {
  a a, c' f,
  g g'' a,, f'
}
```



Pour déplacer une note deux octaves (ou davantage !) plus haut ou plus bas, il suffit de mettre plusieurs '' ou plusieurs ,, — attention cependant à bien mettre deux apostrophes '', et non un guillemet " ! C'est de cette même manière que l'on peut modifier la valeur de départ de `\relative c'`.

Durées (rythme)

La durée d'une note est indiquée par un nombre qui suit son nom : '1' pour une **ronde**, '2' pour une **blanche**, '4' pour une **noire** et ainsi de suite. Les hampes sont ajoutées automatiquement.

```
\relative c'' {
  a1
  a2 a4 a8 a
  a16 a a a a32 a a a a64 a a a a a a a2
}
```



Si aucune durée n'est indiquée, la dernière durée entrée sera utilisée pour les notes suivantes. En l'absence d'indication, la première note est une noire.

Les notes pointées s'obtiennent en ajoutant un point '.' à la valeur rythmique.

```
\relative c'' {
  a a a4. a8
  a8. a16 a a8. a8 a4.
}
```



Silences

On saisit un **silence** tout comme une note, mais avec le caractère 'r'.

```
\relative c'' {
  a r r2
  r8 a r4 r4. r8
}
```



Métrique

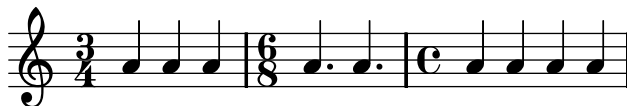
Le **métrique** peut être défini à l'aide de la commande `\time` :

```
\relative c'' {
  \time 3/4
  a4 a a
  \time 6/8
  a4. a
}
```

```

\time 4/4
a4 a a a
}

```



Clefs

La clef peut être définie à l'aide de la commande `\clef` :

```

\relative c' {
  \clef treble
  c1
  \clef alto
  c1
  \clef tenor
  c1
  \clef bass
  c1
}

```



Tout associer

Voici un bref exemple qui montre tous ces éléments ensemble :

```

\relative c, {
  \time 3/4
  \clef bass
  c2 e8 c' g'2.
  f4 e d c4 c, r4
}

```



Plus d'informations

Entrer des hauteurs et des durées

voir [Section 6.1 \[Hauteurs\]](#), page 62 and [Section 6.2.1 \[Durées\]](#), page 69.

Les silences

voir [Section 6.1.9 \[Silences\]](#), page 68.

Les chiffres de mesure et autres commandes de métrique

voir [Section 6.4.3 \[Métrique\]](#), page 83.

Les clés

voir [Section 6.4.1 \[Clefs\]](#), page 81.

2.1.3 Travailler sur des fichiers texte

Le traitement des fichiers source de LilyPond est semblable à celui du code de nombreux langages de programmation répandus : la casse est prise en compte, et les caractères considérés comme espaces ont généralement peu d'importance. Les expressions sont délimitées par des accolades { }, et les commentaires par % ou %{ ... %}.

Si cette phrase vous paraît incompréhensible, ne vous en faites pas ! Tous ces termes vont être expliqués :

- **La casse** : LilyPond est sensible à la casse, c'est à dire qu'une lettre capitale n'a pas la même valeur qu'une lettre minuscule. Les notes, par exemple, doivent être entrées en minuscules : { c d e } est un code valide, alors que { C D E } produira un message d'erreur.
- **Les espaces multiples** : LilyPond ne tient pas compte du nombre d'espaces, ou de retours à la ligne. { c d e } a le même sens que { c d e } ou que

```
{
  c                d
  e }
```

Bien sûr, ce dernier exemple est difficile à lire. Une bonne habitude à prendre est d'indenter les blocs de code avec soit des tabulations soit des doubles espaces :

```
{
  c d e
}
```

- **Expressions musicales** : Tout morceau saisi dans LilyPond doit être placé entre { **accolades** }. Ces caractères indiquent à LilyPond que ce bloc de texte est une et une seule expression musicale, tout comme les parenthèses '(') en mathématiques. Il est préférable, pour éviter toute ambiguïté, d'entourer tous ces crochets d'espaces, à moins qu'ils se trouvent au début ou à la fin d'une ligne.

Une fonction — \relative { } par exemple — compte également comme une seule expression musicale.

- **Les commentaires** : Un commentaire est une indication pour tout lecteur humain d'un fichier de musique ; il est ignoré par l'ordinateur, et n'a donc aucun effet sur la partition imprimée. On distingue deux types de commentaires :
 - la ligne de commentaire, introduite par le symbole '%' : tout ce qui suit ce symbole sur cette ligne sera ignoré.
 - le bloc de commentaire, qui peut être de plusieurs lignes voire de toute une section : tout ce qui se trouve entre %{ et %} est ignoré. Les blocs de commentaires ne peuvent s'imbriquer.

Le fragment suivant met en évidence quelques usages possibles des commentaires :

```
% voici les notes de "ah vous dirai-je maman"
c4 c g' g a a g2

%{
  Ces lignes et les notes qui suivent
  seront ignorées, car elles se trouvent
  dans un bloc de commentaire.

  g g f f e e d d c2
%}
```

Vous trouverez plus d'astuces pour organiser vos fichiers LilyPond dans [Section 4.1 \[Suggestions de saisie des fichiers LilyPond\]](#), page 44.

2.1.4 Comment lire le tutoriel

Comme nous l'avons vu dans [Section 2.1.3 \[Travailler sur des fichiers texte\]](#), page 17, un code LilyPond doit être encadré par des `{ }` ou bien par `\relative c'' { ... }` afin d'être compris. Cependant, dans la suite de ce manuel, la plupart des exemples ne feront pas apparaître ces signes.

Si vous consultez la documentation au format HTML, et que vous souhaitez voir la source exacte d'un exemple, il vous suffit de cliquer sur l'image. Si vous ne disposez pas de la version HTML, il vous est possible de simplement copier et coller le code affiché, mais **à condition** d'ajouter `\relative c'' { }` de la façon suivante :

```
\relative c'' {
  ... collez ici votre exemple...
}
```

Pourquoi avoir omis les accolades ? La plupart des exemples de ce manuel peuvent être insérés au milieu d'un morceau de musique plus long. Il n'y a donc aucune raison d'ajouter `\relative c'' { }` à ces exemples — en effet, il n'est pas possible d'insérer un `\relative` à l'intérieur d'un autre `\relative`. Il vous serait donc devenu impossible de copier un bref exemple de la documentation et de le coller dans une pièce de votre cru.

2.2 Notation sur une seule portée

Cette section présente la notation courante dont on a besoin pour écrire une seule voix sur une seule portée.

2.2.1 Hauteurs de note relatives

Comme nous l'avons vu dans [Section 2.1.2 \[Notation simple\]](#), page 13, LilyPond calcule la hauteur de chaque note en fonction de la précédente³. Si aucune indication supplémentaire d'octaviation n'est ajoutée, il en conclura que chaque hauteur est située à une quarte au plus de la note précédente.

Lilypond tient compte des intervalles induits par les noms des notes — en d'autres termes, une quarte augmentée n'est *pas* équivalente à une quinte diminuée. Ainsi, si l'on part d'un Do, un Fa dièse sera placé au-dessus, tandis qu'un Sol bémol sera placé au-dessous.

```
c2 fis
c2 ges
```



Plus d'informations

Les hauteurs de note relatives
voir [Section 6.1.6 \[Octaves relatives\]](#), page 65.

Les vérifications d'octaves
voir [Section 6.1.7 \[Vérification d'octave\]](#), page 66.

2.2.2 Altérations et armure

³ Il existe un autre mode de saisie des hauteurs, le mode [Section 2.5.3 \[Noms de note absolus\]](#), page 32, mais en pratique il est bien plus aisé et sûr d'avoir recours au mode de hauteurs relatives.

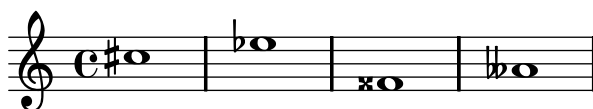
Altérations

Dans la notation par défaut, un **dièse** s’obtient en ajoutant ‘**is**’ au nom de la note, et un **bémol** en ajoutant ‘**es**’. Comme vous pouvez vous y attendre, un double dièse ou double bémol est alors obtenu en ajoutant ‘**isis**’ ou ‘**eses**’⁴.

Cependant, si vous utilisez la commande ‘`\include "italiano.ly"`’ pour entrer les noms de notes français au lieu des noms hollandais, il faudra ajouter un ‘**d**’ pour un dièse, et un ‘**b**’ pour un bémol. Le double dièse et le double bémol s’obtiennent en ajoutant respectivement ‘**dd**’ et ‘**bb**’.

Pour en savoir plus sur les autres langues disponibles, voir [Section 6.1.5 \[Noms de note dans d’autres langues\]](#), page 64.

```
cis1 ees fisis, aeses
```



Armures

L’armure est déterminée par la commande `\key`, suivie d’une hauteur puis de `\major` (majeur) ou `\minor` (mineur) :

```
\key d \major
a1
\key c \minor
a
```



Attention aux armures et aux hauteurs

La combinaison de l’armure et des hauteurs de note — y compris les altérations — permet à LilyPond de déterminer dans quel cas afficher les altérations accidentelles. L’armure n’affecte que les altérations *imprimées*, et non les hauteurs réelles ! Cette fonctionnalité est souvent source de confusion pour les nouveaux utilisateurs, aussi expliquons-la en détail.

LilyPond fait une nette distinction entre le contenu musical et la mise en forme. L’altération d’une note — bémol, bécarré ou dièse — fait partie de sa hauteur, et relève donc du contenu musical. La présence ou non d’une altération accidentelle — un *signe* bémol, bécarré ou dièse — devant la note correspondante est une question qui relève de la mise en page. Mettre en page une partition se fait selon des règles ; les altérations accidentelles seront donc automatiquement imprimées suivant ces règles. Les hauteurs de note, en revanche, relèvent de ce que vous voulez entendre ; et, dans la mesure où la musique que vous entrez est censée être celle que vous voulez entendre, LilyPond (qui n’est chargé que de la gravure) ne les choisira pas à votre place.

Dans cet exemple,

```
\key d \major
d cis fis
```

⁴ Cette syntaxe est dérivée de la convention de dénomination des notes dans les langues nordiques et germaniques, comme l’allemand ou le hollandais.



Aucune note n'a ici d'altération accidentelle, et pourtant vous devrez entrer le 'is' pour les notes cis et fis.

Le code 'd' ne veut pas dire « Imprimez-moi un point noir juste en-dessous de la portée. » Cela signifie plutôt : « Ici se trouve une note dont la hauteur est un ré naturel. » Avec l'armure de la bémol majeur, ce ré sera flanqué d'un bémol accidentel :

```
\key aes \major
e
```



Ajouter explicitement toutes les altérations requiert un peu plus d'effort dans la phase de saisie, mais la transposition en sera grandement facilitée. De plus les altérations accidentelles peuvent être imprimées suivant plusieurs conventions. Regardez [Section 9.1.1 \[Altérations accidentelles automatiques\]](#), page 134 pour connaître les différentes manières dont les altérations accidentelles peuvent être imprimées, suivant les règles que vous choisirez.

Plus d'informations

Les altérations

voir [Section 6.1.2 \[Altérations\]](#), page 63 and [Section 9.1.1 \[Altérations accidentelles automatiques\]](#), page 134.

Les armures

voir [Section 6.4.2 \[Armure\]](#), page 82.

2.2.3 Liaisons

Liaisons de prolongation

Une liaison de prolongation⁵ se crée en ajoutant un tilde '~' à la première note liée.

```
g4~ g c2~
c4 ~ c8 a8 ~ a2
```



Liaisons d'articulation

Une liaison d'articulation d'articulation (ou « legato ») peut englober plusieurs notes. La note de départ et d'arrivée reçoivent respectivement un signe '(' et ')'.

```
d4( c16) cis( d e c cis d) e( d4)
```



⁵ parfois aussi appelée liaison de tenue

Liaisons de phrasé

De plus longues liaisons, dites de phrasé, sont délimitées par \ (et \). Il est possible d'avoir en même temps des legatos et des phrasés, mais pas plusieurs liaisons de phrasé ou de legato à la fois.

```
a8(\( ais b c) cis2 b'2 a4 cis,\)
```



Attention aux types de liaison

Une liaison d'articulation ou de phrasé ressemble à une liaison de tenue, mais n'a pas la même signification. Alors qu'une liaison de tenue ne peut relier que deux notes de même hauteur, le legato indique une articulation de plusieurs notes, éventuellement nombreuses. Les liaisons de tenue peuvent être enchâssées dans un legato ou un phrasé.

```
c2~( c8 fis fis4 ~ fis2 g2)
```



Plus d'informations

Section 6.5.1 [Liaisons de prolongation], page 92,

Section 6.5.2 [Liaisons d'articulation], page 94,

Section 6.5.3 [Liaisons de phrasé], page 95.

2.2.4 Articulations et nuances

Articulations

Des articulations peuvent être ajoutées à une note, au moyen d'un tiret '-' suivi d'un caractère :

```
c- . c-- c-> c-^ c-+ c-_
```



Doigtés

De même, des indications de doigté peuvent être ajoutées à une note en utilisant un tiret ('-') et le chiffre à écrire :

```
c-3 e-5 b-2 a-1
```



Articulations et doigtés sont habituellement placés automatiquement, mais vous pouvez spécifier une direction en utilisant ‘^’ (en haut) ou ‘_’ (en bas). Vous pouvez aussi utiliser plusieurs articulations sur la même note. Dans la plupart des cas, cependant, il est mieux de laisser LilyPond déterminer l’emplacement de l’articulation.

```
c_-^1 d^ . f^4_2-> e^_+ 
```



Nuances

Les signes de nuances sont obtenus en ajoutant à la note les noms des nuances, précédées d’un anti-slash ‘\’ :

```
c\ff c\mf c\p c\pp
```



Crescendos et decrescendos débutent avec les commandes \< et \>. Ils se terminent soit par une nuance d’arrivée, par exemple \f, soit par la commande \! :

```
c2\< c2\ff\> c2 c2\!
```



Plus d’informations

Section 6.6.1 [Articulations], page 101.

Section 6.6.2 [Doigtés], page 103.

Section 6.6.3 [Nuances], page 104.

2.2.5 Barres de ligature automatiques et manuelles

Toutes les barre de ligatures sont dessinées automatiquement :

```
a8 ais d ees r d c16 b a8
```



Lorsqu’on n’aime pas la manière dont les notes sont automatiquement groupées, il est possible de les entrer à la main, en marquant la première note à attacher d’un ‘[’ et la dernière d’un ‘]’.

```
a8[ ais] d[ ees r d] a b
```



Plus d'informations

Groupements de notes et ligatures automatiques

voir [Section 6.5.5 \[Barres de ligature automatiques\]](#), page 96.

Groupements et ligatures manuels

voir [Section 6.5.6 \[Barres de ligature manuelles\]](#), page 96.

2.2.6 Commandes rythmiques avancées

Mesure incomplète

Une levée (ou *anacrouse*) est entrée avec la commande `\partial`, suivie d'une durée : `\partial 4` est une levée d'une noire et `\partial 8` d'une croche.

```
\partial 8
f8 c2 d
```



Nolets

Les nolets sont créés avec la commande `\times`, qui prend deux arguments : une fraction et une expression musicale. La durée des notes de l'expression musicale est multipliée par la fraction. Par exemple les notes d'un triolet durent les deux tiers du temps de leur notation réelle, cette fraction est donc de $2/3$ pour les triolets :

```
\times 2/3 { f8 g a }
\times 2/3 { c r c }
\times 2/3 { f,8 g16[ a g a] }
\times 2/3 { d4 a8 }
```



Notes d'ornement

Des notes d'ornement sont produites par la commande `\grace`, mais aussi en préfixant une expression musicale avec le mot-clé `\appoggiatura` ou `\acciaccatura` :

```
c2 \grace { a32[ b] } c2
c2 \appoggiatura b16 c2
c2 \acciaccatura b16 c2
```



Plus d'informations

Notes d'ornement

voir [Section 6.5.7 \[Notes d'ornement\]](#), page 98,

nolets

voir [Section 6.2.3 \[Nolets\]](#), page 71,

levées

voir [Section 6.4.4 \[Mesures incomplètes\]](#), page 85.

2.3 Notes simultanées

Cette section traite des situations où l'on a plus d'une note à la fois : plusieurs instruments, plusieurs portées pour un même instrument (le piano, par exemple), et les accords.

La polyphonie, en théorie musicale, est la notion d'une musique constituée de plusieurs voix ; dans LilyPond, ce terme désigne les situations où il y a plus d'une voix sur une même portée.

2.3.1 Les expressions musicales en clair

Dans les fichiers source LilyPond, la musique est représentée par ce qu'on appelle des *expressions musicales*. En soi, une seule note peut constituer une expression musicale, si tant est qu'elle soit correctement encadrée :

```
a4
```



Mettre un groupe de notes entre accolades crée une nouvelle expression musicale :

```
{ a4 g4 }
```



Placer une séquence d'expressions musicales — des notes par exemple — entre accolades signifie qu'elles doivent être jouées successivement, les unes après les autres. Le résultat est une expression, qui peut elle-même être regroupée séquentiellement avec d'autres expressions. Ici, l'expression de l'exemple précédent est combinée à deux notes :

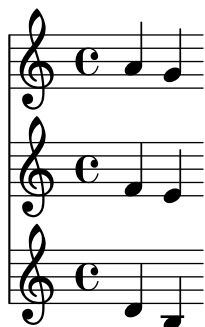
```
{ { a4 g } f g }
```



Expressions musicales simultanées – plusieurs portées

Cette technique est utile pour de la musique polyphonique. Pour entrer une musique avec plusieurs voix ou plusieurs portées, nous pouvons aussi combiner en parallèle les expressions. Deux voix qui doivent être jouées en même temps, sont entrées comme une combinaison simultanée de deux expressions. Une expression musicale 'simultanée' est formée en entourant les expressions entre << et >>. Dans l'exemple suivant, trois expressions (contenant chacune deux notes distinctes) sont combinées simultanément.

```
\relative c'' {
  <<
    { a4 g }
    { f e }
    { d b }
  >>
}
```



Notez que nous avons ici indenté chaque niveau du fichier d'entrée avec un nombre d'espaces différent. LilyPond se moque de l'espace qu'il peut y avoir (ou pas) au début d'une ligne, mais en indentant votre code vous le rendrez bien plus facile à lire pour les humains.

Attention : chaque note saisie est relative à la précédente, mais pas au `c''` de la commande `\relative` de départ.

Expressions musicales simultanées – une seule portée

Pour déterminer le nombre de portées, LilyPond regarde le premier élément autre qu'une accolade. Si c'est une seule note, il y a une portée ; si c'est une expression simultanée, il y aura plus d'une portée.

```
\relative c'' {
  c2 <<c e>>
  << { e f } { c <<b d>> } >>
}
```



Analogie avec les expressions mathématiques

Ce mécanisme est similaire aux formules mathématiques : une grosse formule est créée en assemblant plusieurs petites formules. Ces types de formules, appelées expressions, ont une définition récursive, de telle sorte que vous pouvez fabriquer des expressions arbitrairement longues et complexes. Par exemple,

1

1 + 2

(1 + 2) * 3

((1 + 2) * 3) / (4 * 5)

Ceci est une suite d'expressions, où chacune est contenue dans la suivante. Les expressions les plus simples sont les nombres, et de plus grandes expressions sont produites en combinant des expressions avec des opérateurs — comme '+', '*' et '/' — et des parenthèses. Tout comme les expressions mathématiques, les expressions musicales peuvent être imbriquées avec une profondeur arbitraire, ce qui est nécessaire pour de la musique complexe comme des partitions polyphoniques.

2.3.2 Plusieurs portées

Comme nous l'avons vu dans [Section 2.3.1 \[Les expressions musicales en clair\]](#), page 24, un fichier d'entrée LilyPond est fait d'expressions musicales. Si la partition commence par plusieurs

expressions simultanées, LilyPond créera plusieurs portées. Cependant, il est plus facile de voir ce qu'il advient si l'on crée explicitement chacune des portées.

Pour créer plus d'une portée, chaque partie de la musique constituant une portée est entrée en la faisant précéder de `\new Staff`. Ces éléments `Staff` sont ensuite combinés en parallèle avec `<<` et `>>`, comme ceci :

```
\relative c'' {
  <<
    \new Staff { \clef treble c }
    \new Staff { \clef bass c,, }
  >>
}
```



La commande `\new` introduit un « contexte de notation ». Un contexte de notation est un environnement dans lequel les événements musicaux — comme les notes ou les commandes `\clef` — sont interprétés. Pour des pièces simples, ces contextes sont créés automatiquement. Pour des pièces plus complexes, il est préférable de spécifier explicitement les contextes. Cela assure que chaque fragment aura sa propre portée.

Il existe différents types de contextes. Les contextes `Staff` (portée), `Voice` (voix) et `Score` (partition) gèrent la notation de la mélodie, alors que `Lyrics` gère les paroles et `ChordNames` imprime le nom des accords.

En termes de syntaxe, faire précéder une expression musicale de `\new` crée une plus grosse expression musicale. En reprenant la comparaison, cela ressemble au signe « moins » en mathématiques. La formule $(4 + 5)$ est une expression, donc $-(4 + 5)$ est une plus grosse expression.

Les chiffres de mesure figurant sur une portée affectent toutes les autres portées ; en revanche l'armure d'une portée n'affectera *pas* les autres portées.⁶

```
\relative c'' {
  <<
    \new Staff { \clef treble \time 3/4 c }
    \new Staff { \clef bass \key d \major c,, }
  >>
}
```



⁶ Ce comportement peut être modifié si nécessaire : voir pour cela [Chapitre 9 \[Changer les réglages prédéfinis\]](#), page 134.

2.3.3 Double portée

La musique pour piano s'écrit sur deux portées reliées par une accolade. Imprimer ce type de portée revient au même que dans l'exemple de musique polyphonique de [Section 2.3.2 \[Plusieurs portées\]](#), [page 25](#), mais maintenant cette expression entière doit être interprétée dans un contexte `PianoStaff` :

```
\new PianoStaff <<
  \new Staff ...
  \new Staff ...
>>
```

Voici un bref exemple :

```
\relative c'' {
  \new PianoStaff <<
    \new Staff { \time 2/4 c4 e g g, }
    \new Staff { \clef bass c,, c' e c }
  >>
}
```



Plus d'informations

Voir [Section 7.1 \[Musique pour piano\]](#), [page 117](#).

2.3.4 Polyphonie sur une portée

Quand différentes lignes mélodiques sont combinées sur une seule et même portée, elles sont imprimées comme des voix polyphoniques ; chaque voix a ses propres hampes⁷, liaisons et ligatures, la voix supérieure ayant les hampes vers le haut, la voix inférieure vers le bas.

Ce type de partition est réalisé en entrant chaque voix comme une séquence (avec `{...}`), en combinant simultanément les voix et en les séparant par `\\` :

```
<<
  { a4 g2 f4~ f4 } \\
  { r4 g4 f2 f4 }
>>
```



Pour l'écriture de musique polyphonique, les silences invisibles s'avèrent bien pratiques : ce sont des silences qui ne s'impriment pas. Ils sont utiles pour remplir des voix qui, temporairement, ne jouent rien. On peut voir ici le même exemple avec un silence invisible (`s`) à la place d'un silence normal (`r`) :

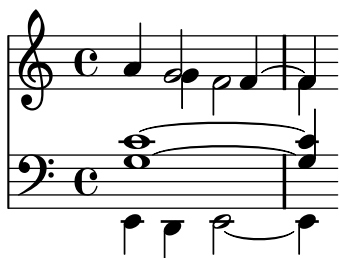
⁷ familièrement appelées queues de note.

```
<<
  { a4 g2 f4~ f4 } \\
  { s4 g4 f2 f4 }
>>
```



Là encore, ces expressions peuvent s’imbriquer arbitrairement :

```
<<
  \new Staff <<
    { a4 g2 f4~ f4 } \\
    { s4 g4 f2 f4 }
  >>
  \new Staff <<
    \clef bass
    { <c g>1 ~ <c g>4 } \\
    { e,,4 d e2 ~ e4 }
  >>
>>
```



Plus d’informations

Voir [Section 6.3.3 \[Polyphonie basique\]](#), page 75.

2.3.5 Combinaison de notes en accords

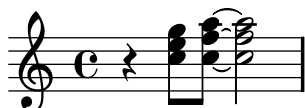
Des accords peuvent être produits en entourant les hauteurs de notes par des angles gauche et droit — ‘<’ et ‘>’ —

```
r4 <c e g>4 <c f a>2
```

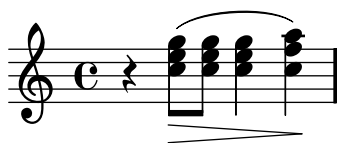


Vous pouvez combiner les indications comme les liaisons et les ligatures de croches avec les accords. Ils doivent cependant être placés en dehors des angles :

```
r4 <c e g>8[ <c f a>]~ <c f a>2
```



```
r4 <c e g>8\>( <c e g> <c e g>4 <c f a>\!)
```



2.4 Chansons

Cette section présente l'écriture vocale et les partitions de variété.

2.4.1 Imprimer des paroles

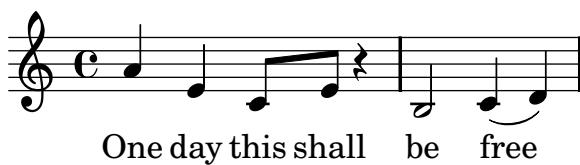
Prenons une mélodie toute simple :

```
\relative c'' {
  a4 e c8 e r4
  b2 c4( d)
}
```



Des paroles peuvent être associées à ces notes, en les combinant avec la commande `\addlyrics`. Les paroles sont entrées en séparant chaque syllable par un espace :

```
<<
  \relative c'' {
    a4 e c8 e r4
    b2 c4( d)
  }
  \addlyrics { One day this shall be free }
>>
```



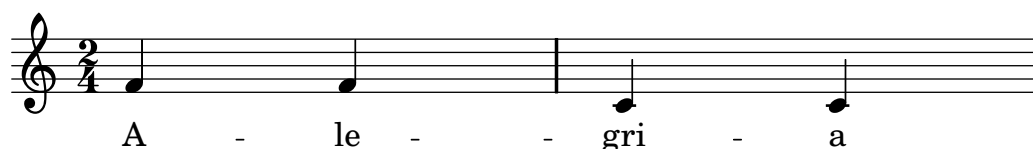
Cette mélodie se termine sur un *mélisme*, c'est-à-dire qu'une seule syllable (« free ») correspond à plus d'une note. Ceci est indiqué avec une *ligne d'extension*. Elle est entrée avec deux caractères souligné (_), c'est-à-dire :

```
<<
  \relative c'' {
    a4 e c8 e r4
    b2 c4( d)
  }
  \addlyrics { One day this shall be free _ _ }
>>
```



De la même manière, les séparations syllabiques d'un mot peuvent être entrées avec deux tirets (-), ce qui produit un tiret centré entre les deux syllabes :

```
<<
  \relative c' {
    \time 2/4
    f4 f c c
  }
  \addlyrics { A -- le -- gri -- a }
>>
```



Plus d'informations

Plus de possibilités, comme celle d'ajouter plusieurs lignes de paroles en dessous d'une même mélodie sont exposées dans [Section 7.3 \[Musique vocale\], page 117](#).

2.4.2 Partition d'une chanson

En musique de variété, il est courant d'indiquer l'accompagnement par le nom des accords. De tels accords peuvent être entrés comme les notes :

```
\chordmode { c2 f4. g8 }
```



Maintenant, chaque hauteur est lue comme la base de l'accord à la place de la note. Ce mode est activé avec `\chordmode`. D'autres accords peuvent être créés en ajoutant des modificateurs après deux points. L'exemple suivant montre quelques modificateurs usuels :

```
\chordmode { c2 f4:m g4:maj7 gis1:dim7 }
```



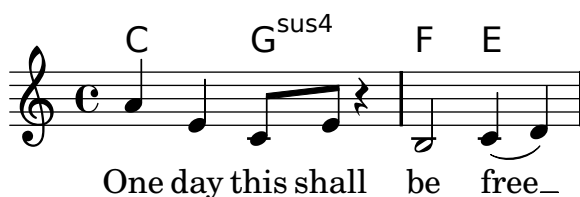
Pour la musique improvisée, les accords ne sont pas imprimés sur des portées mais comme des lignes à part entière. Ceci s'obtient en utilisant `\chords` à la place de `\chordmode`. La même syntaxe sera utilisée que dans le cas de `\chordmode`, mais le rendu des notes interviendra dans un contexte de `ChordNames`, avec le résultat suivant :

```
\chords { c2 f4.:m g4.:maj7 gis8:dim7 }
```

C FmG[△] G^{#07}

Une fois assemblés, les accords, paroles et mélodie forment une partition de chanson :

```
<<
  \chords { c2 g:sus4 f e }
  \relative c'' {
    a4 e c8 e r4
    b2 c4( d)
  }
  \addlyrics { One day this shall be free __ }
>>
```



Plus d'informations

Une liste complète de modificateurs et d'autres options de mise en forme se trouve à la section [Section 6.3.1 \[Accords\]](#), page 75.

2.5 Dernières précisions

L'ultime section de ce tutoriel montre comment ajouter la touche finale à des morceaux simples, et constitue une introduction au reste du manuel.

2.5.1 Numéro de version

La déclaration `\version` stipule le numéro de la version de LilyPond pour laquelle le fichier a été écrit. Pour donner à un fichier la version 2.11.20, ajoutez

```
\version "2.11.20"
```

tout en haut de votre fichier LilyPond.

Cette annotation permet de faciliter les prochaines mises à jour de LilyPond. Les changements dans la syntaxe sont gérés avec un programme spécial, `'convert-ly'` — voir `program usage manual`, `Updating files with convert-ly` — et il utilise `\version` pour déterminer les règles de conversion à appliquer au fichier.

2.5.2 Ajout de titres

Les informations bibliographiques (nom du morceau, du compositeur, etc) sont entrées dans un bloc séparé, le bloc d'en-tête (`\header`), qui existe indépendamment des expressions musicales principales. Le bloc `\header` est habituellement placé en début de fichier.

```
\version "2.11.20"
\header {
  title = "Symphonie"
  composer = "Moi"
  opus = "Op. 9"
}

{
  ... music ...
}
```

Quand le fichier est traité, le titre et le compositeur sont imprimés en haut de la partition. Vous trouverez plus d'informations sur les titres à [Section 10.2.1 \[Créer des titres\]](#), page 136.

2.5.3 Noms de note absolus

Jusqu'ici nous n'avons utilisé que le mode `\relative` pour définir les hauteurs de notes. Si c'est effectivement le moyen le plus simple d'entrer la majeure partie de votre musique, il existe une autre façon de procéder : le mode de hauteurs absolues.

Si vous omettez la commande `\relative`, LilyPond considérera toutes les hauteurs comme des hauteurs absolues. Un `c'` sera toujours un do du milieu, un `b` sera toujours une note au-dessous du précédent, et un `g`, sera toujours la note la plus grave dans la portée de clé de fa.

```
{
  \clef bass
  c' b g, g,
  g, f, f c'
}
```



Voici une gamme sur 4 octaves :

```
{
  \clef bass
  c, d, e, f,
  g, a, b, c
  d e f g
  a b c' d'
  \clef treble
  e' f' g' a'
  b' c'' d'' e''
  f'' g'' a'' b''
  c'''1
}
```



Comme vous pouvez le voir, il faut beaucoup d'apostrophes pour écrire de la musique dans un registre aigu. Regardez cet extrait de Mozart :

```
{
  \key a \major
  \time 6/8
  cis''8. d''16 cis''8 e''4 e''8
  b'8. cis''16 b'8 d''4 d''8
}
```



Toutes ces apostrophes rendent le fichier moins lisible, et c'est donc une source d'erreurs. En mode `\relative`, le même exemple devient bien plus facile à lire :

```
\relative c'' {
  \key a \major
  \time 6/8
  cis8. d16 cis8 e4 e8
  b8. cis16 b8 d4 d8
}
```



Si d'aventure vous faites une erreur d'octaviation, le mode `\relative` la rendra frappante — toutes les notes suivantes seront placées à la mauvaise octave. En mode de hauteurs absolues, une erreur isolée ne serait pas aussi visible, et donc aussi facile à dénicher.

Cependant, le mode de hauteurs absolues reste utile pour les musiques où les intervalles sont étendus, surtout pour les fichiers LilyPond créés par ordinateur.

2.5.4 Organiser ses pièces avec des identificateurs

Lorsque l'on combine tous les éléments étudiés plus haut pour produire des fichiers plus volumineux, les blocs `\score` deviennent beaucoup plus gros parce que les expressions musicales sont plus longues et, dans le cas des pièces polyphoniques, profondément imbriquées. De telles expressions imposantes finissent par devenir peu maniables. Cet inconvénient peut être résolu par l'utilisation d'*identificateurs*.

En utilisant ces identificateurs, que l'on pourrait aussi appeler variables ou macros, il est possible de découper des expressions musicales complexes. Un identificateur se définit comme suit :

```
MusiqueToto = { ... }
```

Le contenu de l'expression musicale `MusiqueToto` pourra être utilisé plus loin en faisant précéder son nom d'un anti-slash, c'est-à-dire `\MusiqueToto`, juste comme n'importe quelle commande LilyPond. Tous les identificateurs doivent être définis *avant* l'expression musicale principale.

```
violin = \new Staff { \relative c'' {
  a4 b c b
}}
cello = \new Staff { \relative c {
  \clef bass
  e2 d
}}
{
  <<
    \violin
    \cello
  >>
}
```



Le nom d'un identificateur ne doit comporter que des caractères alphanumériques non accentués, aucun nombre ni tiret.

Il est possible d'utiliser des variables de types variés. Par exemple,

```
width = 4.5\cm
name = "Wendy"
aFivePaper = \paper { paperheight = 21.0 \cm }
```

En fonction de son contenu, un identificateur peut être utilisé à différents endroits. L'exemple suivant utilise la variable ci-dessus :

```
\paper {
  \aFivePaper
  line-width = \width
}
{ c4^\name }
```

2.5.5 Après le tutoriel

Après avoir parcouru ce tutoriel, vous devriez vous essayer à écrire un morceau ou deux. Commencez par copier l'un des [Annexe D \[Modèles\], page 148](#) types et ajoutez-y des notes. Si vous voulez employer une notation que vous n'avez pas trouvée dans le tutoriel, consultez la référence de notation, en commençant par la [Chapitre 6 \[Notation de base\], page 62](#). Si vous désirez écrire pour un ensemble instrumental non couvert par les [Annexe D \[Modèles\], page 148](#), lisez la section [Section 3.1 \[Étendre les modèles\], page 35](#).

Après avoir écrit quelques pièces courtes, lisez les chapitres 3 à 5 du manuel d'apprentissage. Rien ne s'oppose à ce que vous consultiez dès à présent les autres chapitres, bien sûr ! Néanmoins, le reste du manuel de l'utilisateur part du principe que vous avez déjà bien assimilé la syntaxe de LilyPond. Vous pouvez toujours survoler le reste du manuel, et y revenir plus tard après avoir acquis de l'expérience.

2.5.6 Comment lire le manuel

Comme nous l'avons déjà vu dans [Section 2.1.4 \[Comment lire le tutoriel\], page 18](#), de nombreux exemples du tutoriel n'ont pas fait apparaître `\relative c' { ... }` dans l'extrait de code affiché.

Dans le reste du manuel, les exemples utilisés sont encore beaucoup plus souples : parfois il leur manque le `\relative c' { ... }`, mais d'autres fois ils ont recours à une autre hauteur de référence, telle que `c'` ou `c,`, et dans certains cas c'est même l'exemple entier qui est en mode de hauteurs absolues ! Cependant, de telles ambiguïtés ne se trouvent que dans des contextes où les hauteurs n'ont que peu d'importance. Dans tous les exemples où elles en ont, le mode `\relative` ou absolu `{ }` est explicitement spécifié.

Si vous ne vous y retrouvez toujours pas pour savoir quel code LilyPond produit précisément tel ou tel exemple, consultez la version HTML de ce manuel si ce n'est pas déjà le cas, et cliquez sur l'image de la partition. La source exacte utilisée pour générer ce manuel s'affichera alors.

3 Tout mettre en ordre

Ce chapitre traite des principes généraux de LilyPond, et de la manière de créer des blocs `\score`¹.

3.1 Étendre les modèles

Bon, vous avez lu le tutoriel, vous savez écrire de la musique. Mais comment obtenir les portées que vous voulez ? Les modèles, c'est bien beau, mais que faire quand ils ne traitent pas ce que l'on veut précisément ?

Commencez par le modèle qui vous semblera le plus proche de ce à quoi vous voulez aboutir. Disons par exemple que vous voulez écrire une pièce pour soprano et violoncelle : dans ce cas l'on pourrait commencer par les « notes et paroles », pour la partie de soprano.

```
\version "2.11.20"
melodie = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

texte = \lyricmode {
  Laaa Siii Dooo Rééé
}

\score{
  <<
    \new Voice = "voixUn" {
      \autoBeamOff
      \melodie
    }
    \new Lyrics \lyricsto "voixUn" \texte
  >>
  \layout { }
  \midi { }
}
```

Maintenant, on veut ajouter une partie de violoncelle. Jetons un coup d'oeil sur l'exemple avec les notes seules :

```
\version "2.11.20"
melodie = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

\score {
```

¹ Le terme 'score' signifie partition en anglais.

```

\new Staff \melodie
\layout { }
\midi { }
}

```

On n'a pas besoin de deux commandes `\version`. Ce dont on a besoin, c'est la section `melodie`. De même, on n'a pas besoin de deux sections `\score` — si nous les gardions toutes les deux, on obtiendrait deux parties séparées ; mais nous voulons un vrai duo, avec les deux parties ensemble. Dans la section `\score`, on n'a pas besoin non plus de deux `\layout` ou de deux `\midi`.

Si on se contente de couper et coller les sections `melodie`, on se retrouvera avec deux sections de ce nom ; il nous faut donc les renommer. Appelons la section pour la soprano `musiqueSoprano` et celle pour le violoncelle `musiqueVioloncelle`. Tant qu'on y est, renommons `texte` en `parolesSoprano`. Attention à bien renommer les deux occurrences de chacune de ces dénominations : c'est-à-dire la définition de départ, où l'on trouve `mélodie = relative c' {` , et l'endroit où cette dénomination est utilisée, dans la section `\score`.

Et puis, toujours tant qu'on y est, mettons le violoncelle en clé de Fa, comme le veut l'usage, et donnons-lui d'autres notes.

```

\version "2.11.20"
musiqueSoprano = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

parolesSoprano = \lyricmode {
  Laaa Siii Dooo Rééé
}

musiqueVioloncelle = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  d4 g fis8 e d4
}

\score{
  <<
    \new Voice = "voixUn" {
      \autoBeamOff
      \musiqueSoprano
    }
    \new Lyrics \lyricsto "voixUn" \parolesSoprano
  >>
  \layout { }
  \midi { }
}

```

Voilà qui est mieux, mais la partie de violoncelle n'apparaît pas sur la partition — en effet, nous ne l'avons pas utilisée dans la section `\score`. Si l'on veut que la partie de violoncelle s'imprime sous la partie de soprano, on va devoir ajouter :

```
\new Staff \musiqueVioloncelle
```

en dessous de tout ce qui concerne la soprano. Il nous faut également encadrer la musique par des `<<` et `>>`, qui feront comprendre à LilyPond que plusieurs événements — ici, des objets `Staff` — se déroulent en même temps. Le bloc `\score` ressemble maintenant à

```
\score{
  <<
    <<
      \new Voice = "voixUn" {
        \autoBeamOff
        \musiqueSoprano
      }
      \new Lyrics \lyricsto "voixUn" \parolesSoprano
    >>
    \new Staff \musiqueVioloncelle
  >>
  \layout { }
  \midi { }
}
```

C'est un peu le bazar dans tout ça ; mais il vous sera facile de mettre un peu d'ordre dans l'indentation. Voici le modèle pour soprano et violoncelle au complet :

```
\version "2.11.15"
sopranoMusic = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

sopranoLyrics = \lyricmode {
  Aaa Bee Cee Dee
}

celloMusic = \relative c {
  \clef bass
  \key c \major
  \time 4/4

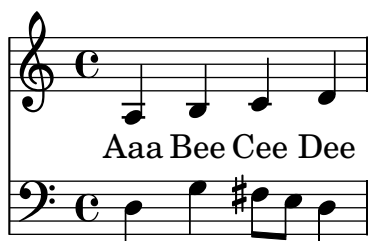
  d4 g fis8 e d4
}

\score{
  <<
    <<
      \new Voice = "one" {
        \autoBeamOff
        \sopranoMusic
      }
    >>
    \new Staff \celloMusic
  >>
  \layout { }
  \midi { }
```

```

\new Lyrics \lyricsto "one" \sopranoLyrics
>>
\new Staff \celloMusic
>>
\layout { }
\midi { }
}

```



3.2 Comment les fichiers LilyPond sont organisés

La mise en forme des fichiers d'entrée de LilyPond est vraiment peu astreignante, afin d'offrir assez de souplesse aux utilisateurs expérimentés pour qu'ils puissent organiser leurs fichiers comme ils l'entendent. Cependant, les nouveaux utilisateurs peuvent parfois se perdre en raison de cette souplesse. Cette section présente sommairement l'organisation du code LilyPond, en privilégiant la simplicité au détriment de certains détails. Vous trouverez une description plus complète dans [Section 10.1.2 \[Structure de fichier\]](#), page 136.

La plupart des exemples de ce manuel sont de courts fragments, par exemple

```
c4 a b c
```

Comme vous le savez maintenant (du moins nous l'espérons), ceci ne peut pas être traité en tant que tel. Il s'agit de formes abrégées des exemples complets ; pour pouvoir être traitées, elles doivent au moins être encadrées par des accolades :

```
{
  c4 a b c
}
```

La plupart des exemples font aussi intervenir la commande `\relative`, suivie de `c'` ou `c''`. Elle n'est pas à proprement parler nécessaire pour le traitement des exemples, mais dans la plupart des cas le résultat sera vraiment déplorable si vous l'oubliez.

```
\relative c'' {
  c4 a b c
}
```



C'est ici que nous entamons les choses sérieuses : le code LilyPond, sous cette forme, est en réalité un *autre* raccourci. Même s'il est traité sans problème, et aboutit au bon résultat, c'est une forme abrégée de :

```
\score {
  \relative c'' {
    c4 a b c
  }
}
```

```
}
```

Un bloc `\score` doit commencer par une et une seule expression musicale. Rappelez-vous que cette expression peut être ce que vous voulez, d’une note toute seule à un gigantesque

```
{
  \new GrandStaff <<
    collez ici la partition complète de votre opéra de Wagner préféré
  >>
}
```

Dès lors que tout cela est entre accolades : `{ ... }`, c’est une et une seule expression musicale.

Le bloc `\score` peut contenir d’autres éléments, tels que

```
\score {
  { c'4 a b c' }
  \layout { }
  \midi { }
  \header { }
}
```

Certains préfèrent mettre ces commandes en dehors du bloc `\score` — par exemple, on met souvent le `\header` au-dessus. C’est juste là une autre forme abrégée que LilyPond accepte.

Un autre raccourci pratique est la possibilité de définir des variables — également appelées « identificateurs ». Dans tous les modèles, vous trouverez :

```
melodie = \relative c' {
  c4 a b c
}

\score {
  { \melodie }
}
```

Lorsque LilyPond examinera ce fichier, il va prendre la valeur de la variable `melodie`, c’est-à-dire tout ce qui suit le signe `=`, et l’insérer partout où il tombera sur `\melodie`. Vous êtes libre de choisir comment dénommer vos variables² ; ce peut être `melodie`, `global`, `maindroitepiano`, ou `laTeteAToto`. Pour plus de détails, voir [Section 4.2 \[Gagner du temps grâce aux identificateurs et fonctions\]](#), page 45.

Pour une description complète du format des fichiers d’entrée, voir [Section 10.1.2 \[Structure de fichier\]](#), page 136.

3.3 La partition est une unique expression musicale

Dans la section précédente, [Section 3.2 \[Comment les fichiers LilyPond sont organisés\]](#), page 38, nous avons vu l’organisation générale des fichiers d’entrée de LilyPond. Mais c’est comme si nous avions éludé la question essentielle : comment diable peut-on savoir quoi mettre après `\score` ?

En fait, nous ne l’avons pas éludée du tout : le grand mystère est tout simplement qu’il n’y a *pas* de mystère. Allez, expliquons-le en une ligne :

Un bloc `\score` doit commencer par une et une seule expression musicale.

Peut-être serait-il judicieux de relire la section [Section 2.3.1 \[Les expressions musicales en clair\]](#), page 24, dans laquelle vous avez appris à construire de grandes expressions musicales petit bout par petit bout — nous avons vu les notes, puis les accords, etc. Maintenant, nous allons partir d’une grande expression musicale, et redescendre la pente.

² Les noms de variables sont sensibles à la casse, et ne peuvent contenir ni chiffre, ni tiret, ni caractère accentué.

```

\score {
  { % cette accolade marque le début de l'expression musicale
    \new GrandStaff <<
      insérez ici votre opéra de Wagner préféré
    >>
  } % cette accolade marque la fin de l'expression musicale
  \layout { }
}

```

Un opéra de Wagner multiplierait facilement la longueur de ce manuel par deux ou trois, alors faisons-le en version chant/piano. On n'a plus besoin d'une partition d'orchestre — **GrandStaff** — donc laissons cela de côté. Par contre, un chanteur et un piano *pourraient* nous être utiles.

```

\score {
  {
    <<
      \new Staff = "chanteur" <<
        >>
      \new PianoStaff = piano <<
        >>
    >>
  }
  \layout { }
}

```

Vous vous souvenez que nous avons recours à << et >> pour mettre en place des musiques simultanées. Et, pour le coup, on aimerait *vraiment* que la partie vocale et l'accompagnement soient imprimés ensemble...

```

\score {
  {
    <<
      \new Staff = "chanteur" <<
        \new Voice = "chant" { }
      >>
      \new Lyrics \lyricsto chant \new Lyrics { }
      \new PianoStaff = "piano" <<
        \new Staff = "mainDroite" { }
        \new Staff = "mainGauche" { }
      >>
    >>
  }
  \layout { }
}

```

On y voit nettement plus clair maintenant. Nous voici donc avec la partie du chanteur, qui contient un ensemble **Voice**, ce qui dans LilyPond correspond à une voix, au sens de voix d'une polyphonie plutôt que de voix chantée — ce pourrait être une partie de violon par exemple.

Nous avons également une partie de piano, qui contient deux portées : une pour la main droite, une autre pour la main gauche.

À ce point, on pourrait commencer à ajouter les notes. Dans les accolades qui suivent **\new Voice = chant**, on pourrait commencer à écrire

```

\relative c'' {
  a4 b c d
}

```

Mais si l'on procédait ainsi, la section `\score` deviendrait vite assez touffue, et très rapidement on ne s'y retrouverait plus. C'est pourquoi on utilisera plutôt des variables, ou identificateurs :

```
melodie = { }
texte = { }
mainDroite = { }
mainGauche = { }
\score {
  {
    <<
    \new Staff = "chanteur" <<
    \new Voice = "chant" { \melodie }
    >>
    \new Lyrics \lyricsto chant \new Lyrics { \texte }
    \new PianoStaff = "piano" <<
    \new Staff = "mainDroite" { \mainDroite }
    \new Staff = "mainGauche" { \mainGauche }
    >>
  }
  \layout { }
}
```

Souvenez-vous que vous pouvez donner aux variables le nom que vous voulez, à condition de respecter les caractères autorisés. Ces limitations sont décrites dans [Section 10.1.2 \[Structure de fichier\], page 136](#).

Quand on écrit, ou que l'on lit, une section `\score`, mieux vaut y aller lentement et soigneusement. Commencez par le niveau le plus large, puis travaillez sur chaque niveau plus détaillé. À ce propos, une indentation stricte et propre est vraiment d'une aide précieuse : assurez-vous que chaque élément d'un même niveau a le même décalage horizontal dans votre éditeur de texte !

3.4 Partitions d'orchestre

Dans la musique d'orchestre, toutes les notes sont imprimées deux fois. D'abord dans les parties séparées destinées aux musiciens, et ensuite dans le conducteur destiné au chef. Les variables sont là pour vous éviter un double travail. La musique n'est entrée qu'une seule fois, et stockée dans une variable, dont le contenu servira à imprimer à la fois la partie séparée et la partition d'orchestre.

Il est judicieux de définir les notes dans un fichier séparé. Par exemple, supposons que le fichier `'musique-Cor.ly'` contienne la partie suivante pour un duo cor/basson.

```
notesCor = \relative c {
  \time 2/4
  r4 f8 a cis4 f e d
}
```

On établira alors une partie séparée en constituant un nouveau fichier :

```
\include "musique-Cor.ly"
\header {
  instrument = "Cor en Fa"
}

{
  \transpose f c' \notesCor
}
```

À la ligne

```
\include "musique-Cor.ly"
```

sera substitué le contenu du fichier ‘musique-Cor.ly’, et de ce fait la variable `notesCor` se trouvera définie. La commande `\transpose f c'` indique que son argument `\notesCor` sera transposé à la quinte supérieure : le son réel ‘f’ s’écrit ‘c’, ce qui est la caractéristique d’un Cor en Fa. La transposition est visible comme suit :



Dans les pièces d’ensemble, il arrive souvent qu’une voix ne joue pas pendant plusieurs mesures. Un silence spécial, appelé silence multi-mesures, l’indique alors. On l’obtient par un ‘R’ majuscule, suivi d’une durée : 1 pour une pause, 2 pour une demi-pause, etc. Cette durée peut être multipliée pour établir de plus longs silences. Par exemple, le silence suivant dure 3 mesures à 2/4.

```
R2*3
```

Dans une partie séparée, les silences multi-mesures sont compressés. Il faut pour cela définir la propriété `skipBars` à ‘vrai’ :

```
\set Score.skipBars = ##t
```

Cette commande donne la valeur ‘vrai’ — ‘true’ en anglais, ‘#t’ dans le langage Scheme — à cette propriété dans le contexte `Score`. Si l’on ajoute dans la musique ci-dessus le silence multi-mesures et cette option, on obtient le résultat suivant :

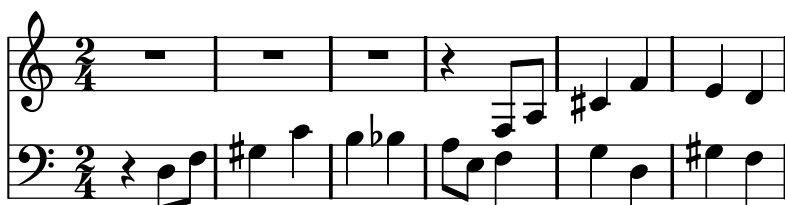


Le conducteur rassemble toute la musique. Si l’on suppose que l’autre voix de notre duo se trouve dans le fichier ‘musique-Basson.ly’ en tant que variable `notesBasson`, on établira un conducteur avec

```
\include "musique-Basson.ly"
\include "musique-Cor.ly"

<<
  \new Staff \notesCor
  \new Staff \notesBasson
>>
```

ce qui équivaut à



Des informations plus détaillées sur la mise en place de conducteurs et de parties séparées se trouvent dans le manuel : voir [Section 8.3 \[Musique orchestrale\]](#), page 131.

Les variables (‘propriétés’) réglables sont abordées en détail dans [Section 9.2.3 \[Changer à la volée les propriétés de contexte\]](#), page 134.

4 Travailler sur des projets LilyPond

Cette section explique comment résoudre ou éviter certains problèmes courants. Si vous avez de l'expérience en programmation, beaucoup de ces astuces peuvent vous paraître évidentes, mais vous ne perdrez tout de même pas votre temps à lire ce chapitre.

4.1 Suggestions de saisie des fichiers LilyPond

Maintenant vous êtes prêt à travailler sur de plus gros fichiers LilyPond — des pièces entières, et plus seulement les petits exemples du tutoriel. Mais comment devriez-vous vous y prendre ?

Tant que LilyPond parvient à comprendre vos fichiers et produit le résultat que vous souhaitez, peu importe la manière dont le code est organisé. Néanmoins, quelques critères doivent être pris en compte lorsque l'on écrit un fichier LilyPond.

- Si vous faites une erreur, la structure même du fichier LilyPond peut permettre de la localiser plus ou moins facilement.
- Et si vous souhaitez partager vos fichiers avec quelqu'un d'autre, ou si vous souhaitez modifier vos propres fichiers dans quelques années ? Si certains fichiers LilyPond sont compréhensibles au premier coup d'oeil, d'autres vous feront vous arracher les cheveux pendant une heure.
- Et si vous souhaitez mettre à jour votre fichier pour l'utiliser avec une version plus récente de LilyPond ? La syntaxe du langage d'entrée change parfois lorsque LilyPond s'améliore. La plupart des changements peuvent être appliqués automatiquement avec `convert-ly`, mais quelques-uns peuvent requérir une intervention manuelle. Vos fichiers LilyPond peuvent être structurés de manière à faciliter leur mise à jour.

4.1.1 Suggestions générales

Voici quelques conseils qui peuvent vous éviter certains problèmes ou en résoudre d'autres.

- **Ajoutez le numéro de version dans chaque fichier.** Notez que chaque fichier modèle contient une ligne `\version "2.11.15"`. Nous vous conseillons fortement d'inclure cette ligne, même pour de petits fichiers. Par expérience, il est très difficile de se rappeler quelle version de LilyPond on utilisait quelques années auparavant. L'utilitaire `convert-ly` demande que vous spécifiez quelle version de LilyPond vous utilisez.
- **Ajoutez des contrôles:** [Section 6.2.5 \[Vérification des limites de mesure\]](#), [page 73](#), [Section 6.1.7 \[Vérification d'octave\]](#), [page 66](#) et [Section 6.2.6 \[Vérification du numéro de mesure\]](#), [page 74](#). Si vous avez ajouté des contrôles de loin en loin, et que vous faites une erreur, vous pourrez la retrouver plus rapidement. « De loin en loin », qu'est-ce à dire ? Cela dépend de la complexité de la musique. Pour de la musique très simple, peut-être une ou deux fois. Pour de la musique très complexe, peut-être à chaque mesure.
- **Une mesure par ligne de texte.** Si la musique en elle-même ou le résultat que vous désirez contient quelque chose de compliqué, il est souvent bon de n'écrire qu'une seule mesure par ligne. Économiser de la place en tassant huit mesures par ligne, ça ne vaut pas vraiment le coup si l'on doit corriger vos fichiers.
- **Ajoutez des commentaires.** Utilisez soit des numéros de mesure (assez souvent), soit des références au contenu musical — « second thème des violons », « quatrième variation », etc. Vous pouvez ne pas avoir besoin des commentaires lorsque vous écrivez une pièce pour la première fois, mais si vous souhaitez y revenir deux ou trois ans plus tard pour changer quelque chose, ou si vous donnez le fichier source à un ami, ce sera beaucoup plus difficile de déterminer vos intentions ou la manière dont votre fichier est structuré si vous n'y avez pas adjoint de commentaires.
- **Indentez les accolades.** Beaucoup de problèmes viennent d'un défaut de parité entre `{` et `}`.

- **Séparez les affinages de mise en forme** de la musique elle-même. Voyez [Section 4.2 \[Gagner du temps grâce aux identificateurs et fonctions\]](#), page 45 et [Section 4.3 \[Feuilles de style\]](#), page 47.

4.1.2 Graver de la musique existante

Si vous saisissez de la musique à partir d'une partition existante, c'est-à-dire de la musique déjà écrite,

- n'entrez qu'un seul système de la partition originale à la fois — mais toujours une seule mesure par ligne de texte —, et vérifiez chaque système lorsqu'il est terminé. Vous pouvez utiliser la commande `showLastLength` pour accélérer la compilation — voir [Section 10.5 \[Ignorer des passages de la partition\]](#), page 137 ;
- définissez `mBreak = {\break }` et insérez `\mBreak` dans le fichier d'entrée des sauts de ligne identiques à la partition originale. Cela facilite la comparaison entre la partition originale et la partition de LilyPond. Lorsque vous avez fini de relire votre musique, vous pouvez définir `mBreak = { }` pour enlever tous ces sauts de ligne, et laisser LilyPond placer les sauts de ligne selon son propre algorithme.

4.1.3 Pour des projets d'envergure

Lorsque l'on travaille sur un gros projet, il devient vital de structurer clairement ses fichiers LilyPond.

- **Utilisez un identificateur pour chaque voix**, avec un minimum de structure dans la définition. La structure de la section `\score` est la plus susceptible de changer, notamment dans une nouvelle version de LilyPond, alors que la définition du violon l'est beaucoup moins.

```
violin = \relative c' {
  g4 c'8. e16
}
...
\score {
  \new GrandStaff {
    \new Staff {
      \violin
    }
  }
}
```

- **Séparez les retouches** des définitions de musique. Ce conseil a été vu dans [Section 4.1.1 \[Suggestions générales\]](#), page 44, mais pour les gros projets c'est absolument vital. Nous pouvons avoir besoin de changer la définition de `fthenp`, mais dans ce cas nous n'aurons besoin de le faire qu'une seule fois, et nous pourrons encore éviter de modifier quoi que ce soit à l'intérieur de la définition du violon.

```
fthenp = _\markup{
  \dynamic f \italic \small { 2nd } \hspace #0.1 \dynamic p }
violin = \relative c' {
  g4\fthenp c'8. e16
}
```

4.2 Gagner du temps grâce aux identificateurs et fonctions

Jusqu'à maintenant, vous avez vu ce type de code :

```
hornNotes = \relative c' { c4 b dis c }
\score {
```

```
{
  \hornNotes
}
```



Vous comprendrez combien cela peut être utile pour écrire de la musique minimaliste :

```
fragA = \relative c'' { a4 a8. b16 }
fragB = \relative c'' { a8. gis16 ees4 }
violin = \new Staff { \fragA \fragA \fragB \fragA }
\score {
  {
    \violin
  }
}
```



Cependant, vous pouvez aussi utiliser ces identificateurs — aussi connus sous le nom de variables, macros, ou commandes (définies par l'utilisateur) — pour des retouches :

```
dolce = \markup{ \italic \bold dolce }
padText = { \once \override TextScript #'padding = #5.0 }
fthenp = \markup{ \dynamic f \italic \small { 2nd } \hspace #0.1 \dynamic p }
violin = \relative c'' {
  \repeat volta 2 {
    c4._\dolce b8 a8 g a b |
    \padText
    c4.^"hi there!" d8 e' f g d |
    c,4.\fthenp b8 c4 c-. |
  }
}
\score {
  {
    \violin
  }
}
\layout{ragged-right=##t}
}
```



Ces identificateurs sont évidemment utiles pour économiser de la frappe. Mais ils peuvent l'être même si vous ne les utilisez qu'une seule fois : ils réduisent la complexité. Regardons l'exemple précédent sans aucun identificateur. C'est beaucoup plus laborieux à lire, et particulièrement la dernière ligne.

```
violin = \relative c'' {
  \repeat volta 2 {
    c4._\markup{ \italic \bold dolce } b8 a8 g a b |
    \once \override TextScript #'padding = #5.0
    c4.^"hi there!" d8 e' f g d |
    c,4.\markup{ \dynamic f \italic \small { 2nd }
      \hspace #0.1 \dynamic p } b8 c4 c-. |
  }
}
```

Jusqu'ici nous avons vu des substitutions statiques : quand LilyPond rencontre `\padText`, il le remplace par le contenu que nous lui avons défini — c'est-à-dire le contenu à droite de `padText=`).

LilyPond gère également des substitutions non-statiques — vous pouvez les voir comme des fonctions.

```
padText =
#(define-music-function (parser location padding) (number?)
  #{
    \once \override TextScript #'padding = #$padding
  #})

\relative c''' {
  c4^"piu mosso" b a b
  \padText #1.8
  c4^"piu mosso" d e f
  \padText #2.6
  c4^"piu mosso" fis a g
}
```



Utiliser les identificateurs est aussi un bon moyen pour vous épargner du travail si la syntaxe de LilyPond change un jour — voir [Section 4.4 \[Mettre à jour d'anciens fichiers\]](#), page 51. Si vous avez une seule définition, par exemple `\dolce`, pour tous vos fichiers (voir [Section 4.3 \[Feuilles de style\]](#), page 47), et que la syntaxe change, alors vous n'aurez qu'à mettre à jour votre seule définition `\dolce`, au lieu de devoir modifier chaque fichier `.ly`.

4.3 Feuilles de style

La sortie que produit LilyPond peut être largement modifiée — voir [Chapitre 5 \[Retoucher les résultats\]](#), page 53 pour plus de détails. Mais que faire si vous avez beaucoup de fichiers auxquels vous souhaitez appliquer vos retouches ? Ou si vous souhaitez simplement séparer les retouches de la musique elle-même ? Rien de plus facile.

Prenons un exemple. Ne vous inquiétez pas si vous ne comprenez pas les parties avec tous les `#()`. Celles-ci sont expliquées dans [Section 5.6 \[Retouches avancées avec Scheme\]](#), page 60.

```
mpdolce = #(make-dynamic-script (markup #:hspace 1 #:translate (cons 5 0)
  #:line(:dynamic "mp" #:text #:italic "dolce" )))
tempoMark = #(define-music-function (parser location markp) (string?)
  #{
    \once \override Score . RehearsalMark #'self-alignment-X = #left
    \once \override Score . RehearsalMark #'extra-spacing-width = #'(+inf.0 . -inf.0)
    \mark \markup { \bold $markp }
  })

\relative c'' {
  \tempo 4=50
  a4.\mpdolce d8 cis4--\glissando a | b4 bes a2
  \tempoMark "Poco piu mosso"
  cis4.\< d8 e4 fis | g8(\! fis)-. e( d)-. cis2
}
```



Il y a quelques problèmes de chevauchement ; nous allons arranger cela en utilisant les techniques de [Section 5.1 \[Déplacer des objets\]](#), page 53. On peut aussi faire quelque chose pour les définitions de `mpdolce` et `tempoMark`. Elles produisent le résultat que nous désirons, mais nous pourrions aussi vouloir les utiliser dans une autre pièce. Il suffirait de les copier et les coller au début de chaque fichier, mais c'est fastidieux. De plus, cela laisse les définitions dans nos fichiers de musique, et je trouve personnellement tous ces `#()` assez laids. Stockons-les dans un autre fichier :

```
%%% enregistrez ceci dans un fichier nommé "definitions.ly"
mpdolce = #(make-dynamic-script (markup #:hspace 1 #:translate (cons 5 0)
  #:line(:dynamic "mp" #:text #:italic "dolce" )))
tempoMark = #(define-music-function (parser location markp) (string?)
  #{
    \once \override Score . RehearsalMark #'self-alignment-X = #left
    \once \override Score . RehearsalMark #'extra-spacing-width = #'(+inf.0 . -inf.0)
    \mark \markup { \bold $markp }
  })
```

Maintenant, modifions notre musique (enregistrez ce fichier sous "musique.ly").

```
\include "definitions.ly"

\relative c'' {
  \tempo 4=50
  a4.\mpdolce d8 cis4--\glissando a | b4 bes a2
  \once \override Score.RehearsalMark #'padding = #2.0
  \tempoMark "Poco piu mosso"
  cis4.\< d8 e4 fis | g8(\! fis)-. e( d)-. cis2
}
```



C'est mieux, mais effectuons encore quelques retouches. Le glissando est peu visible, c'est pourquoi nous allons l'épaissir et le rapprocher des têtes de notes. Déplaçons l'indication métronomique au-dessus de la clef, au lieu de la laisser au-dessus de la première note. Et pour finir, mon professeur de composition déteste les chiffrages de mesure en « C », nous allons donc le transformer en « 4/4 ».

Cependant, ne changez pas le fichier 'musique.ly'. Remplacez le fichier 'definitions.ly' par ceci :

```
%% definitions.ly
mpdolce = #(make-dynamic-script (markup #:hspace 1 #:translate (cons 5 0)
  #:line( #:dynamic "mp" #:text #:italic "dolce" )))
tempoMark = #(define-music-function (parser location markp) (string?)
#{
  \once \override Score . RehearsalMark #'self-alignment-X = #left
  \once \override Score . RehearsalMark #'extra-spacing-width = #'(+inf.0 . -inf.0)
  \mark \markup { \bold $markp }
#})

\layout{
  \context { \Score
    \override MetronomeMark #'extra-offset = #'(-9 . 0)
    \override MetronomeMark #'padding = #'3
  }
  \context { \Staff
    \override TimeSignature #'style = #'numbered
  }
  \context { \Voice
    \override Glissando #'thickness = #3
    \override Glissando #'gap = #0.1
  }
}
```



C'est encore mieux ! Mais supposons maintenant que je veuille publier cette pièce. Mon professeur de composition n'aime pas les chiffrages de mesure en « C », mais moi je les aime bien. Copions l'actuel 'definitions.ly' dans le fichier 'publication-web.ly', et modifions ce dernier. Puisque la musique est destinée à produire un fichier PDF affiché sur écran, nous allons aussi augmenter la taille globale de police.

```
%% definitions.ly
mpdolce = #(make-dynamic-script (markup #:hspace 1 #:translate (cons 5 0)
  #:line( #:dynamic "mp" #:text #:italic "dolce" )))
tempoMark = #(define-music-function (parser location markp) (string?)
```

```

#{
  \once \override Score . RehearsalMark #'self-alignment-X = #left
  \once \override Score . RehearsalMark #'extra-spacing-width = #'(+inf.0 . -inf.0)
  \mark \markup { \bold $markp }
#})

#(set-global-staff-size 23)
\layout{
  \context { \Score
    \override MetronomeMark #'extra-offset = #'(-9 . 0)
    \override MetronomeMark #'padding = #'3
  }
  \context { \Staff
  }
  \context { \Voice
    \override Glissando #'thickness = #3
    \override Glissando #'gap = #0.1
  }
}

```



Il ne nous reste plus qu'à remplacer `\include "definitions.ly"` par `\include "publication-web.ly"` dans notre fichier de musique.

Il est possible, bien sûr, de rendre cela encore plus pratique. Nous pourrions créer un fichier `'definitions.ly'` qui ne contiendrait que les définitions de `mpdolce` et de `tempoMark`, un fichier `'publication-web.ly'` qui ne contiendrait que la section `layout` décrite ci-dessus et un fichier `'universite.ly'` qui ne contiendrait que les retouches pour produire le résultat que mon professeur préfère. Le début du fichier `'musique.ly'` ressemblerait alors à

```

\include "definitions.ly"

%%% Décommentez seulement une de ces deux lignes !
\include "publication-web.ly"
%\include "universite.ly"

```

Cette approche peut être utile même si vous ne produisez qu'un seul jeu de partitions. J'utilise une demi-douzaine de fichiers de « feuille de style » pour mes projets. Je commence chaque fichier de musique par `\include "../global.ly"` qui contient :

```

%%% global.ly
\version "2.11.15"

```

```
#(ly:set-option 'point-and-click #f)
\include "../init/init-defs.ly"
\include "../init/init-mise-en-page.ly"
\include "../init/init-en-tetes.ly"
\include "../init/init-papier.ly"
```

4.4 Mettre à jour d'anciens fichiers

La syntaxe de LilyPond change de temps en temps. Ces changements de syntaxe du langage d'entrée accompagnent les améliorations du logiciel. Ces changements sont parfois destinés à rendre les fichiers plus faciles à lire et à écrire, ou permettent d'intégrer de nouvelles fonctionnalités.

LilyPond est fourni avec un utilitaire qui facilite cette mise à jour : `convert-ly`. Pour savoir comment utiliser ce programme, voir `program usage manual`, `Updating files with convert-ly`.

Malheureusement, `convert-ly` ne peut pas réaliser toutes les modifications. Il s'occupe des changements qui ne requièrent qu'une simple substitution de texte — comme `raggedright` devenant `ragged-right` —, les autres étant trop compliqués à effectuer. Les changements de syntaxe qui ne sont pas gérés par `convert-ly` sont énumérés dans `program usage manual`, `Updating files with convert-ly`.

Par exemple, dans les versions 2.4 et antérieures de LilyPond, les accents et les lettres non anglaises étaient entrées en utilisant LaTeX — par exemple, `'No\''el'`. À partir de la version 2.6, le caractère 'ë' doit être entré directement dans le fichier LilyPond comme caractère UTF-8. `convert-ly` ne peut pas changer tous les caractères LaTeX en caractères UTF-8 ; vous devez mettre à jour vos vieux fichiers LilyPond manuellement.

4.5 Résolution de problèmes — tout remettre à plat

Tôt ou tard, vous écrirez un fichier que LilyPond ne peut pas compiler. Les messages que LilyPond affiche peuvent vous aider à trouver l'erreur, mais dans beaucoup de cas vous aurez besoin de faire quelques recherches pour déterminer la source du problème.

Pour ce faire, les outils les plus puissants sont le commentaire de fin de ligne, indiqué par `%`, et le commentaire multilignes (ou bloc de commentaire), indiqué par `%{ ... %}`. Si vous ne pouvez localiser le problème, commencez par mettre en commentaire de grandes parties de votre fichier d'entrée. Après avoir mis en commentaire une section, essayez de compiler à nouveau. Si cela fonctionne, c'est que le problème se situe dans cette partie du fichier. Si cela ne fonctionne pas, continuez à mettre en commentaire d'autres sections, jusqu'à ce que vous ayez quelque chose qui compile.

Dans un cas extrême, vous pourriez en arriver à

```
\score {
  <<
    % \melodie
    % \harmonie
    % \basse
  >>
  \layout{}
}
```

c'est-à-dire un fichier sans aucune musique.

Si cela arrive, ne vous découragez pas. Décommentez un peu, la partie de basse par exemple, et voyez si ça fonctionne. Si ce n'est pas le cas, placez en commentaire toute la partie de basse, mais laissez `\basse` décommenté dans le bloc `\score`.

```

basse = \relative c' {
  %{
    c4 c c c
    d d d d
  %}
}

```

Maintenant commencez à décommenter petit à petit le partie de `basse` jusqu'à ce que vous localisiez la ligne qui pose problème.

Une autre technique de débogage très utile est la construction de [Section 4.6 \[Exemples minimaux\]](#), page 52.

4.6 Exemples minimaux

Un exemple minimal est un exemple de code aussi court que possible. De tels exemples sont bien plus compréhensibles que des exemples longs. Les exemples minimaux sont utilisés pour

- les rapports de bogue,
- les demandes d'aide sur les listes de diffusion,
- un ajout à [LilyPond Snippet Repository](#).

Pour construire un exemple minimal, la règle est très simple : enlevez tout ce qui n'est pas nécessaire. Il est préférable de commenter les lignes non nécessaires plutôt que de les effacer : ainsi, si vous vous apercevez que certaines étaient *réellement* nécessaires, vous pouvez les décommenter au lieu de les resaisir.

Il y a deux exceptions à cette règle du strict nécessaire :

- incluez le numéro de `\version` en début de fichier
- si possible, utilisez `\paper{ ragged-right=##t }` au début de votre exemple.

Tout l'intérêt d'un exemple minimal réside dans sa facilité de lecture :

- évitez d'utiliser des notes, armures ou métriques compliquées, à moins que vous vouliez montrer quelque chose en rapport avec celles-ci,
- n'utilisez pas de commandes `\override` sauf si elles font l'intérêt de l'exemple.

5 Retoucher les résultats

Ce chapitre indique comment modifier le résultat que vous obtiendrez. LilyPond offre de nombreuses possibilités de réglages, permettant de modifier quasiment chaque élément de votre partition.

5.1 Déplacer des objets

Aussi surprenant que cela puisse paraître, LilyPond n'est pas parfait. Certains éléments sur la partition peuvent se chevaucher, ce qui est regrettable mais, le plus souvent, facile à corriger.

À FAIRE : les modifications de la gestion des espacements de la version 2.12 feront perdre leur pertinence aux exemples suivants. Ils démontrent cependant la puissance de LilyPond, et justifient à ce titre leur présence dans ces lignes, tant que d'autres exemples n'auront pas été proposés.

```
% temporary code to break this example:
\override TextScript #'outside-staff-priority = ##f
e4~\markup{ \italic ritenuto } g b e
```



Le plus simple est ici d'augmenter la distance entre l'objet (du texte comme ici, ou bien des nuances ou des doigtés) et la note. Dans LilyPond, il s'agit de la propriété `padding`, qui se mesure en espaces relatifs à la taille de la portée. Pour la plupart des objets (chacun ayant sa propre valeur), elle est définie à 1.0, ou un peu moins. Nous voulons ici l'augmenter : essayons 1.5.

```
% temporary code to break this example:
\override TextScript #'outside-staff-priority = ##f
\once \override TextScript #'padding = #1.5
e4~\markup{ \italic ritenuto } g b e
```



C'est déjà mieux ! Mais on peut certainement encore améliorer le résultat. Il nous semble, après plusieurs essais, que la meilleure valeur dans ce cas soit 2.3. Toutefois, ce constat est le fruit d'expérimentations et de goût personnel en matière de notation. Essayez le même exemple avec 2.3... mais également avec des valeurs plus grandes (ou plus petites). À votre avis, quelle est la meilleure version ?

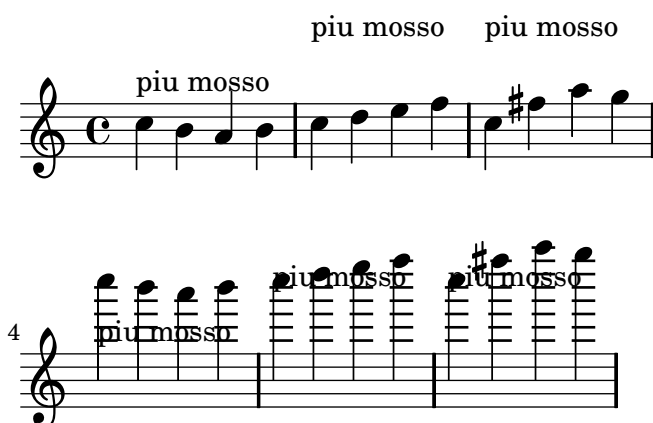
La propriété `staff-padding` est de nature similaire. `padding` détermine l'espace minimum entre un objet et l'objet le plus proche (le plus souvent une note ou les lignes de la portée) ; `staff-padding` détermine pour sa part l'espace minimum entre un objet et la portée. La différence est subtile, mais vous apparaîtra clairement ici :

```
% temporary code to break this example:
\override TextScript #'outside-staff-priority = ##f
c4~"piu mosso" b a b
\once \override TextScript #'padding = #4.6
```

```

c4^"piu mosso" d e f
\once \override TextScript #'staff-padding = #4.6
c4^"piu mosso" fis a g
\break
c'4^"piu mosso" b a b
\once \override TextScript #'padding = #4.6
c4^"piu mosso" d e f
\once \override TextScript #'staff-padding = #4.6
c4^"piu mosso" fis a g

```



Une autre démarche permet de contrôler totalement la position d'un objet — on peut le déplacer horizontalement ou verticalement. Il suffit d'avoir recours à la propriété `extra-offset`. En fait c'est une méthode plus complexe, qui peut en outre poser des problèmes. Quand on déplace un objet à l'aide de `extra-offset`, le déplacement est effectué après que LilyPond a placé tous les autres objets. Par conséquent, l'objet ainsi déplacé peut venir recouvrir d'autres objets déjà placés.

```

% temporary code to break this example:
\override TextScript #'outside-staff-priority = ##f
\once \override TextScript #'extra-offset = #'( 1.0 . -1.0 )
e4^\markup{ \italic ritenuto } g b e

```



Lorsqu'on utilise `extra-offset`, le premier nombre décrit le déplacement horizontal (négatif pour un déplacement vers la gauche) tandis que le deuxième décrit un déplacement vertical (négatif pour le haut). Après quelques essais, on peut choisir les valeurs suivantes qui semblent donner un résultat satisfaisant.

```

% temporary code to break this example:
\override TextScript #'outside-staff-priority = ##f
\once \override TextScript #'extra-offset = #'( -1.6 . 1.0 )
e4^\markup{ \italic ritenuto } g b e

```



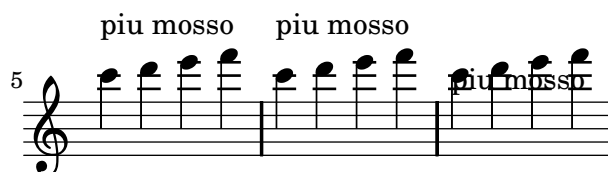
Une fois encore, c'est après quelques tâtonnements que l'on a abouti à ces nombres, au regard du résultat final. Si vous souhaitez que le texte soit plus haut, plus à gauche, etc. essayez vous-même et choisissez après avoir regardé le résultat.

Une dernière mise en garde : dans cette section, nous avons eu recours à

```
\once \override TextScript ...
```

ce qui permet de régler le placement du texte pour la note suivante. Mais si cette note n'a pas de texte, le réglage ne s'appliquera pas et n'attendra **pas** le prochain texte. Pour que ce comportement persiste après la commande, ne mettez pas `\once`. Votre réglage s'appliquera alors partout, jusqu'à ce que vous l'annuliez au moyen de la commande `\revert`. Ceci est expliqué en détail dans [Section 9.3 \[La commande \override\], page 134](#).

```
% temporary code to break this example:
\override TextScript #'outside-staff-priority = ##f
c4^"piu mosso" b
\once \override TextScript #'padding = #4.6
a4 b
c4^"piu mosso" d e f
\once \override TextScript #'padding = #4.6
c4^"piu mosso" d e f
c4^"piu mosso" d e f
\break
\override TextScript #'padding = #4.6
c4^"piu mosso" d e f
c4^"piu mosso" d e f
\revert TextScript #'padding
c4^"piu mosso" d e f
```



Voir aussi

Dans ce même manuel : [Section 9.3 \[La commande \override\], page 134](#), [Section 5.3 \[Retouches courantes\], page 56](#).

5.2 Corriger les collisions d'objets

Dans la section [Section 5.1 \[Déplacer des objets\], page 53](#), nous avons vu comment déplacer un objet `TextScript`. Ce même procédé peut être appliqué à d'autres types d'objets : il vous suffira de remplacer `TextScript` par le nom de l'objet en question.

Pour trouver cette dénomination, regardez les liens ‘**Voir aussi**’ en bas des pages de la documentation. Par exemple, en bas de la page [Section 6.6.3 \[Nuances\]](#), page 104, nous trouvons

Voir aussi

Référence du programme : `DynamicText`, `Hairpin`. Le placement vertical de ces symboles est contrôlé par `DynamicLineSpanner`.

Ce qui implique que, pour modifier la hauteur d’une nuance, nous utiliserons

```
\override DynamicLineSpanner #'padding = #2.0
```

Nous ne listerons pas ici tous les types d’objets, mais seulement les plus communs :

Type d’objet	Nom de l’objet
Nuances (verticalement)	<code>DynamicLineSpanner</code>
Nuances (horizontalement)	<code>DynamicText</code>
Liaisons de tenue	<code>Tie</code>
Liaisons	<code>Slur</code>
Indications d’articulation	<code>Script</code>
Doigtés	<code>Fingering</code>
Textes (“ <code>texte</code> ”)	<code>TextScript</code>
Repères	<code>RehearsalMark</code>

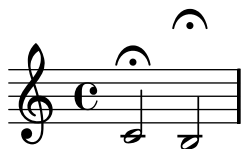
5.3 Retouches courantes

Certains réglages sont si courants que des raccourcis sont fournis sous forme de commandes telles que `\slurUp` ou `\stemDown`. Toutes ces commandes sont décrites dans les différentes sections de la Référence de notation.

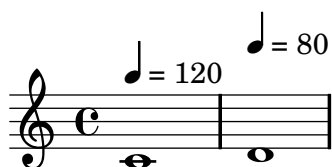
La liste complète des modifications possibles pour chaque type d’objet (tels que les liaisons ou les ligatures) se trouve dans la Référence du programme. Cependant, certaines propriétés sont communes à de nombreux objets, et on peut de ce fait définir quelques réglages génériques.

- La propriété `padding` peut être définie de manière à accroître (ou décroître) la distance entre les symboles qui se placent au-dessus ou au-dessous des notes. Ce qui s’applique à tous les objets régis par `side-position-interface`.

```
c2\fermata
\override Script #'padding = #3
b2\fermata
```



```
% This will not work, see below:
\override MetronomeMark #'padding = #3
\tempo 4=120
c1
% This works:
\override Score.MetronomeMark #'padding = #3
\tempo 4=80
d1
```



Notez, dans le second exemple, l'importance de savoir à quel contexte correspond l'objet. Dans la mesure où l'objet `MetronomeMark` appartient au contexte `Score`, les modifications affectant le contexte `Voice` ne l'affecteront pas. Pour plus de détails, voir [Section 9.3.1 \[Élaborer une retouche\]](#), page 134.

- La propriété `extra-offset` permet de déplacer latéralement et verticalement ; c'est pourquoi elle requiert deux nombres. Le premier affecte le placement horizontal (un nombre positif déplace l'objet vers la droite) ; le second le placement vertical (un nombre positif déplace l'objet vers le haut). Cette propriété est de bas niveau : le moteur de formatage ne tient aucun compte des placements qu'elle induit.

Dans l'exemple suivant, le second doigté est déplacé un peu vers la gauche, et plus bas de 1.8 espaces :

```
\stemUp
f-5
\once \override Fingering
      #'extra-offset = #'(-0.3 . -1.8)
f-5
```



- La propriété `transparent` imprime les objets avec de l'‘encre invisible’ : l'objet n'est pas visible, mais tous les comportement le concernant s'appliquent quand même. Il occupe une certaine place, intervient dans la gestion des collisions, et on peut lui attacher des liaisons ou des ligatures.

L'exemple suivant montre comment tenir des notes entre différentes voix, au moyen de liaisons. Ces liaisons de tenue, en principe, ne peuvent relier que deux notes d'une même voix. On introduit donc la liaison dans une autre voix :



et on efface la première croche (hampe vers le haut) de ladite voix ; maintenant la liaison semble passer d'une voix à l'autre :

```
<< {
  \once \override Stem #'transparent = ##t
  b8~ b8\noBeam
} \ {
  b[ g8]
} >>
```



Pour s'assurer que le crochet de la hampe que nous avons effacée ne raccourcira pas la liaison, nous allons également rallonger cette hampe, en attribuant à la propriété `length` la valeur 8 :

```
<< {
  \once \override Stem #'transparent = ##t
  \once \override Stem #'length = #8
  b8~ b8\noBeam
} \ {
  b[ g8]
} >>
```



Les distances dans LilyPond sont mesurées dans l'unité staff-space (espace de portée) tandis que la plupart des propriétés relatives aux épaisseurs sont mesurées à l'aide de l'unité line-thickness (épaisseur de ligne). Toutefois, certaines d'entre-elles échappent à cette règle : par exemple l'épaisseur des liens de croches est mesurée à l'aide de l'unité staff-space. Pour de plus amples informations, consultez les sections correspondantes de la Référence du programme.

5.4 Fichiers fournis avec LilyPond

La Référence du programme contient beaucoup d'informations sur LilyPond. Cependant vous pouvez en découvrir encore plus en consultant les fichiers internes de LilyPond.

Des réglages par défaut (tels que les définitions des blocs `\header{}`) sont contenus dans des fichiers `.ly`. D'autres (comme les définitions des commandes « markup ») sont contenus dans des fichiers `.scm` (Scheme). Malheureusement, des explications plus complètes dépassent le cadre de ce manuel. Les utilisateurs qui souhaiteraient comprendre le fonctionnement de ces fichiers de configuration doivent être avertis que des connaissances techniques substantielles et beaucoup de temps sont nécessaires.

- Linux : `'dossierduprogramme/lilypond/usr/share/lilypond/current/'`
- Mac OS X : `'dossierduprogramme/LilyPond.app/Contents/Resources/share/lilypond/current/'`.
Pour accéder à ce dossier, deux possibilités : soit, dans un Terminal, taper `cd` suivi du chemin complet ci-dessus ; soit Control-clicquer (ou clic droit) sur l'application LilyPond et sélectionner 'Afficher le contenu du paquet'.
- Windows : `'dossierduprogramme/LilyPond/usr/share/lilypond/current/'`

Les répertoires `'ly/'` et `'scm/'` sont tout particulièrement intéressants. En effet les fichiers du type `'ly/property-init.ly'` ou encore `'ly/declarations-init.ly'` déterminent toutes les définitions avancées communes.

5.5 Réduire le nombre de pages de la partition

Parfois, une partition peut se terminer avec seulement un ou deux systèmes sur la dernière page. Ceci peut être ennuyeux surtout si vous constatez, en regardant les pages précédentes, qu'il reste encore beaucoup de place sur celles-ci.

Si vous vous intéressez aux problèmes de mise en page, `annotate-spacing`, peut alors être un outil d'une valeur inestimable. Cette commande imprime les valeurs de nombreuses commandes d'espacement concernant la mise en page. Consultez [Section 11.3 \[Montrer les espacements\]](#), [page 138](#) pour de plus amples informations. À l'aide des informations données par `annotate-spacing` on peut voir quelles marges il est souhaitable de modifier afin de résoudre le problème.

En plus d'agir sur les marges, il existe d'autres possibilités qui permettent de gagner de la place.

- Vous pouvez demander à LilyPond de placer les systèmes aussi près que possible les uns des autres (pour en disposer autant que possible sur une page), tout en répartissant les systèmes afin de ne pas laisser de blanc en bas de la dernière page.

```
\paper {
  between-system-padding = #0.1
  between-system-space = #0.1
  ragged-last-bottom = ##f
  ragged-bottom = ##f
}
```

- Vous pouvez obliger LilyPond à mettre un certain nombre de systèmes par page. Par exemple, si LilyPond veut placer onze systèmes dans une page, vous pouvez l'obliger à n'en mettre que dix.

```
\paper {
  system-count = #10
}
```

- Supprimer (ou réduire) les objets qui augmentent la hauteur du système. C'est le cas en particulier de certaines reprises (avec des alternatives) qui placent des crochets au dessus des portées. Si ces crochets de reprise se poursuivent sur deux systèmes, ils prendront plus de place que s'ils sont regroupés sur un même système.

Un autre exemple : déplacer les nuances qui « débordent » d'un système.

```
\relative c' {
  e4 c g\ff c
  \override DynamicLineSpanner #'padding = #-1.8
  \override DynamicText #'extra-offset = #'(-2.1 . 0)
  e4 c g\ff c
}
```



- Modifier l'espacement vertical avec `SpacingSpanner`. Reportez-vous à [Section 11.6.3 \[Modifier l'espacement horizontal\]](#), page 139 pour plus de détails.

```
\score {
  \relative c'' {
    g4 e e2 | f4 d d2 | c4 d e f | g4 g g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
    d4 d d d | d4 e f2 | e4 e e e | e4 f g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
  }
  \layout {
    \context {
      \Score
      \override SpacingSpanner
        #'base-shortest-duration = #(ly:make-moment 1 4)
    }
  }
```



5.6 Retouches avancées avec Scheme

Nous avons déjà vu comment le résultat obtenu avec LilyPond peut être largement personnalisé à l'aide de commandes comme `\override TextScript #'extra-offset = (1 . -1)`. Cependant, l'utilisation de Scheme ouvre des possibilités encore plus grandes. Pour des explications complètes là-dessus, consultez [Annexe B \[Tutoriel Scheme\]](#), page 146 et [Chapitre 12 \[Interfaces pour les programmeurs\]](#), page 140.

On peut utiliser Scheme simplement à l'aide des commandes `\override`.

```

padText = #(define-music-function (parser location padding) (number?)
#{
  \once \override TextScript #'padding = #$padding
#})

\relative c'' {
  c4^"piu mosso" b a b
  \padText #1.8
  c4^"piu mosso" d e f
  \padText #2.6
  c4^"piu mosso" fis a g
}

```



On peut s'en servir pour créer de nouvelles commandes :

```

tempoMark = #(define-music-function (parser location padding marktext)
                                     (number? string?)
#{
  \once \override Score . RehearsalMark #'padding = $padding
  \once \override Score . RehearsalMark #'extra-spacing-width = #'(+inf.0 . -inf.0)
  \mark \markup { \bold $marktext }
#})

\relative c'' {
  c2 e
  \tempoMark #3.0 #"Allegro"
}

```

```

      g c
    }

```

Allegro



On peut même y inclure des expressions musicales :

```

pattern = #(define-music-function (parser location x y) (ly:music? ly:music?)
#{
  $x e8 a b $y b a e
#})

\relative c''{
  \pattern c8 c8\f
  \pattern {d16 dis} { ais16-> b\p }
}

```



5.7 Éviter les retouches ralentissant le traitement

LilyPond peut effectuer des vérifications supplémentaires lors du traitement des fichiers, cependant le rendu nécessitera alors plus de temps. En contrepartie, il y aura moins d'ajustements manuels à réaliser.

```

%% Ceci sert à s'assurer que les indications textuelles resteront à l'intérieur des m
\override Score.PaperColumn #'keep-inside-line = ##t

```

6 Notation de base

De l'utilisation des fonctions de notation de base.

6.1 Hauteurs

De la manière de spécifier les hauteurs de notes.

6.1.1 Hauteurs naturelles

La hauteur s'écrit — à moins de préciser une autre langue — avec la notation anglaise, en utilisant les lettres a à g. Une gamme ascendante de do majeur s'écrit

```
\clef bass
c d e f g a b c'
```



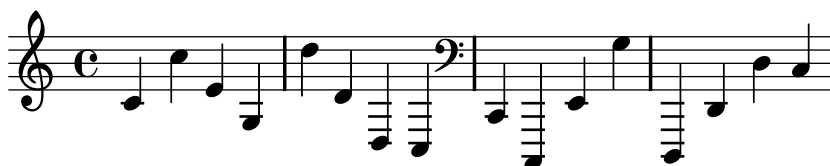
La note c est écrite une octave sous le do central.

```
\clef treble
c1
\clef bass
c1
```



L'octave peut être précisée sous forme d'une série d'apostrophes ''' ou d'une série de virgules ', '. Chaque ''' hausse la note d'une octave, chaque ', ' baisse la note d'une octave.

```
\clef treble
c' c'' e' g d'' d' d c
\clef bass
c, c,, e, g d,, d, d c
```



Il existe une autre méthode pour préciser à quelle octave se situe la note à graver ; cette méthode demande moins d'indications d'octave (' and ,) — voir [Section 6.1.6 \[Octaves relatives\]](#), [page 65](#).

6.1.2 Altérations

Dans la notation par défaut, un dièse est formé en ajoutant **-is** après le nom de note, un bémol en ajoutant **-es**. Les double-dièses et double-bémols sont obtenus en ajoutant **-isis** ou **-eses** au nom de note.

```
a2 ais a aes
a2 aisis a aeses
```



Ce sont les noms de note hollandais. En hollandais, on élide **aes** pour écrire **as**, mais les deux formes sont acceptées. De manière similaire, on accepte aussi bien **es** que **ees**.

```
a2 as e es
```



Un bécarré annule l'effet d'une altération, qu'elle soit accidentelle ou à l'armure. Cependant, dans la syntaxe des noms de note, les bécarrés ne s'écrivent pas avec un suffixe ; un simple nom de note donnera une note bécarré.

```
a4 aes a2
```



LilyPond interprète l'entrée **d e f** comme « imprimer un ré naturel, un mi naturel et un fa naturel », sans tenir compte de l'armure. Pour plus d'information à propos de la distinction entre le contenu musical et sa représentation, voir [Section 2.2.2 \[Altérations et armure\]](#), page 18.

```
\key d \major
d e f g
d e fis g
```



Propriétés couramment modifiées

En accord avec les règles standards de l'écriture musicale, on imprime un bécarré avant un dièse ou un bémol si on a besoin d'annuler une altération précédente. Pour modifier ce comportement, spécifiez `\set Staff.extraNatural = ##f`

```
ceses4 ces cis c
\set Staff.extraNatural = ##f
ceses4 ces cis c
```



Voir aussi

Référence du programme : `LedgerLineSpanner`, `NoteHead`.

6.1.3 Altérations de précaution

Normalement, les altérations sont imprimées automatiquement, mais il se peut que vous vouliez les imprimer manuellement. On peut forcer l'impression d'une altération, dite « de précaution », en ajoutant un point d'exclamation '!' après la hauteur de note. Une altération entre parenthèses peut être obtenue en ajoutant un point d'interrogation '?' après la hauteur de note. Ces signes peuvent aussi être utilisés pour imprimer des bécarrés.

```
cis cis cis! cis? c c? c! c
```



Voir aussi

L'impression automatique des altérations peut être affinée de plusieurs manières. Pour plus d'information, voir [Section 9.1.1 \[Altérations accidentelles automatiques\]](#), page 134.

6.1.4 Altérations microtonales

Les demi-bémols et demi-dièses s'écrivent en ajoutant respectivement `-eh` et `-ih`. Voici une série de dos altérés en hauteurs croissantes :

```
\set Staff.extraNatural = ##f
ceseh ceh cih csih
```



Les micro-intervalles sont aussi exportés dans le fichier MIDI.

Bogues

Il n'y a pas de standard universellement accepté pour noter le bémol et demi (qui abaisse la hauteur trois quarts de ton), le symbole de LilyPond n'est donc conforme à aucun standard.

6.1.5 Noms de note dans d'autres langues

Vous disposez de jeux prédéfinis de noms de notes pour plusieurs autres langues. Pour les utiliser, incluez le fichier d'initialisation spécifique à la langue voulue. Par exemple, pour les langues romanes, ajoutez `\include "italiano.ly"` au début du fichier source. Les fichiers de langues disponibles ainsi que les noms de notes utilisés sont les suivants :

	Noms de note							dièse	bémol	double dièse
nederlands.ly	c	d	e	f	g	a	bes b	-is	-es	-isis
english.ly	c	d	e	f	g	a	bf b	-s/-sharp	-f/-flat	-ss/-x/

												-sharpshsarp
deutsch.ly	c	d	e	f	g	a	b	h	-is	-es		-isis
norsk.ly	c	d	e	f	g	a	b	h	-iss/-is	-ess/-es		-ississ/-isis
svenska.ly	c	d	e	f	g	a	b	h	-iss	-ess		-ississ
italiano.ly	do	re	mi	fa	sol	la	sib	si	-d	-b		-dd
catalan.ly	do	re	mi	fa	sol	la	sib	si	-d/-s	-b		-dd/-ss
espanol.ly	do	re	mi	fa	sol	la	sib	si	-s	-b		-ss

Notez qu'en hollandais, en allemand, en norvégien et en suédois, un 'la' altéré de bemol se note **aes** et **aeses**. Ces formes sont contractées en **as** et **ases** (ou plus communément **asas**). Dans certains fichiers linguistiques, seules ces formes abrégées ont été définies ; ceci s'applique aussi aux suffixes pour les quarts de ton.

Certaines musiques utilisent des microtonalités, pour lesquelles les altérations sont des fractions de dièse ou bémol « normaux ». Le tableau suivant répertorie les noms de note en quart de ton, tels que définis dans plusieurs fichiers linguistiques. Les préfixes 'semi-' et 'sesqui-' correspondent au 'demi-' et 'trois demis'. À noter qu'aucune définition n'existe à ce jour pour le norvégien, le suédois, le catalan et l'espagnol.

	Noms de note								semi- dièse	semi- bémol	sesqui- dièse	sesqui- bémol
nederlands.ly	c	d	e	f	g	a	bes	b	-ih	-eh	-isih	-eseh
english.ly	c	d	e	f	g	a	bf	b	-qs	-qf	-tqs	-tqf
deutsch.ly	c	d	e	f	g	a	b	h	-ih	-eh	-isih	-eseh
norsk.ly	c	d	e	f	g	a	b	h				
svenska.ly	c	d	e	f	g	a	b	h				
italiano.ly	do	re	mi	fa	sol	la	sib	si	-sd	-sb	-dsd	-bsb
catalan.ly	do	re	mi	fa	sol	la	sib	si				
espanol.ly	do	re	mi	fa	sol	la	sib	si				

6.1.6 Octaves relatives

On spécifie les octaves en ajoutant ' et , aux noms de hauteurs. En recopiant de la musique, on a vite fait de mettre une note à la mauvaise octave, et ce genre d'erreur est difficile à retrouver. Le mode d'écriture `\relative` prévient ces erreurs dans la mesure où elles deviennent beaucoup plus évidentes : une seule erreur décale le reste de la pièce à une mauvaise octave.

`\relative startpitch musicexpr`

ou

`\relative musicexpr`

c' est utilisé par défaut si aucune hauteur de départ n'est définie.

L'octave des notes mentionnées dans *musicexpr* va être calculée de la manière suivante : si aucun signe de changement d'octave n'est utilisé, l'intervalle de base entre la note actuelle et la précédente sera toujours au plus d'une quarte. Cet intervalle est déterminé sans tenir compte des altérations ; ainsi un **fisis** après un **ceses** sera placé au-dessus du **ceses**. En d'autres termes, une quarte doublement augmentée demeure considérée comme un intervalle plus petit qu'une quinte diminuée, bien que la quarte doublement augmentée soit de sept demi-tons et la quinte diminuée de seulement six demi-tons.

Les signes de changement d'octave ' et , peuvent être ajoutés pour hausser ou baisser la note d'une octave supplémentaire. Lorsque l'on entre en mode `\relative`, une hauteur absolue de départ peut être spécifiée, et agira dès lors comme si elle précédait la première note de *musicexpr*. Si aucune hauteur de départ n'est spécifiée, le do central sert de point de départ.

Voici le mode `\relative` en action.

```
\relative c'' {
  b c d c b c bes a
}
```



On utilise les signes de changement d'octave pour les intervalles dépassant la quarte.

```
\relative c'' {
  c g c f, c' a, e''
}
```



Si l'expression précédente est un accord, c'est la première note de l'accord qui détermine l'emplacement de la première note du prochain accord.

```
\relative c' {
  c <c e g>
  <c' e g>
  <c, e' g>
}
```



La hauteur après `\relative` contient un nom de note.

La conversion en mode `\relative` n'affectera pas les sections `\transpose`, `\chordmode` ou `\relative` situées dans son argument. Pour utiliser `\relative` dans de la musique transposée, un code `\relative` additionnel doit être placé dans `\transpose`.

6.1.7 Vérification d'octave

Les tests d'octave rendent la correction d'erreurs d'octave plus facile dans le mode d'octave `relative` : une note peut être suivie de *=apostrophes/virgules* pour indiquer à quelle octave absolue elle devrait être. Dans l'exemple suivant,

```
\relative c'' { c='' b='' d,='' }
```

le `d` générera un avertissement, puisqu'on attend un `d''` mais qu'on obtient un `d'` — il n'y a qu'une tierce entre `b'` et `d'`. Sur la partition, l'octave sera corrigée pour donner un `d''` et la prochaine note sera calculée en fonction de `d''` et non de `d'`.

Il existe aussi une vérification d'octave qui ne produit pas de musique imprimée, ayant pour syntaxe

```
\octave hauteur
```

Cette commande vérifie que la *hauteur* considérée sans apostrophe ni virgule et comme hauteur relative, est la même que la *hauteur* absolue considérée avec ses éventuelles apostrophes

ou virgules. Sinon, un avertissement est émis et l'octave est corrigée. La *hauteur* n'est pas considérée comme une note, et donc n'est pas imprimée.

Dans l'exemple ci-dessous, le premier test réussit, puisque le **e** — dans le mode **relative** — est au plus à une quarte de **a'**. Cependant, le deuxième test produit un avertissement, puisque le **e** est à quinte de **b'**. Le message d'avertissement est émis, et l'octave est corrigée afin que les notes suivantes soient à nouveau à la bonne octave.

```
\relative c' {
  e
  \octave a'
  \octave b'
}
```

L'octave d'une note qui suit un test d'octave est déterminée selon la note précédente. Dans l'exemple suivant, la dernière note est un **a'**, au-dessus du do central. Cela veut dire que le test **\octave** réussit, et peut donc être enlevé sans changer le résultat sur la partition.

```
\relative c' {
  e
  \octave b
  a
}
```



6.1.8 Transposition

Une expression musicale peut être transposée avec **\transpose**. Voici la syntaxe :

```
\transpose note_de départ note_d_arrivée musicexpr
```

Cela signifie que *musicexpr* est transposé d'un intervalle entre les notes *note_de_départ* et *note_d_arrivée* : toute note dont la hauteur était *note_de_départ* est changée en *note_d_arrivée*.

Prenons comme exemple une pièce écrite en ré majeur. Si cette pièce est un peu trop basse pour l'interprète, elle peut être transposée en mi majeur avec

```
\transpose d e ...
```

Regardons maintenant une partie écrite pour violon — un instrument en ut). Si cette partie doit être jouée par une clarinette en la (écrite à la tierce mineure supérieure, un do écrit donnant un la réel), la transposition suivante créera la partie appropriée.

```
\transpose a c ...
```

\transpose fait la distinction entre les notes enharmoniques : **\transpose c cis** et **\transpose c des** transposeront la pièce un demi-ton plus haut. Mais la première version écrira des dièses et la deuxième des bémols.

```
mus = { \key d \major cis d fis g }
\new Staff {
  \clef "F" \mus
  \clef "G"
  \transpose c g' \mus
  \transpose c f' \mus
}
```



On peut aussi utiliser `\transpose` pour entrer des notes écrites pour un instrument transpositeur. Normalement, les hauteurs dans LilyPond sont écrites en ut, i.e. en sons réels, mais elles peuvent être écrites dans un autre ton. Quand, par exemple, on écrit pour une trompette en si bémol, commençant sur ré note réelle, on pourrait écrire

```
\transpose c bes { e4 ... }
```

Pour imprimer cette musique en si bémol à nouveau — et de ce fait produire une partie de trompette, au lieu d'un conducteur en notes réelles, on utilisera un deuxième `transpose`

```
\transpose bes c { \transpose c bes { e4 ... } }
```

Voir aussi

Référence du programme : `TransposedMusic`.

Exemple : `'scheme/transpose-pitches-with-minimum-accidentals.ly'`.

Bogues

Si vous voulez utiliser en même temps `\transpose` et `\relative`, vous devez mettre `\transpose` en dehors de `\relative`, puisque `\relative` n'aura aucun effet sur la musique apparaissant dans un `\transpose`.

6.1.9 Silences

Les silences sont écrits comme des notes avec le nom de note `r`.

```
r1 r2 r4 r8
```



Les pauses d'une mesure complète, qui sont placées au centre de la mesure, doivent être entrées comme des mesures de silence. Elles peuvent être utilisées pour une seule mesure comme pour plusieurs, et leur utilisation est expliquée dans la section [Section 8.2.1 \[Silences multiples\]](#), page 131.

Pour spécifier explicitement la position verticale d'un silence, écrivez une note suivie de `\rest`. Un silence sera placé à la position où serait imprimée la note.

```
a'4\rest d'4\rest
```



Cela rend plus facile la mise en place de la musique polyphonique, puisque le formateur automatique de collision des silences laissera ces silences tranquilles.

Voir aussi

Référence du programme : `Rest`.

6.1.10 Silences invisibles

Un silence invisible — que l’on pourrait appeler un « saut » — peut être entré comme une note avec le nom de note “s” ou avec `\skip durée`

```
a4 a4 s4 a4 \skip 1 a4
```



La syntaxe `s` est seulement disponible pour les modes d’entrée de notes et d’accords. Dans d’autres situations, pour l’entrée de paroles par exemple, vous devrez utiliser la commande `\skip`.

```
<<
  \relative { a'2 a2 }
  \new Lyrics \lyricmode { \skip 2 bla2 }
>>
```



La commande de saut génère simplement une case musicale vide. Elle ne produit rien sur la partition, pas même un symbole transparent. Le code de saut `s` crée tout de même une **Staff** et une **Voice** lorsque nécessaire, tout comme les commandes de note et de silence. Ainsi, le code suivant aboutit à une portée vide.

```
{ s4 }
```



Le fragment `{ \skip 4 }` produirait une page vide.

Voir aussi

Référence du programme : `SkipMusic`.

6.2 Rythme

Des rythmes, durées et mesures.

6.2.1 Durées

Dans les modes de notes, d’accords et de paroles, les durées sont écrites avec des chiffres et des points : les durées sont indiquées par leur valeur fractionnaire par rapport à la durée d’une ronde. Une noire, par exemple, qui équivaut à un 1/4 de ronde — « quarter note » en anglais — s’écrit `4`, alors qu’une blanche — « half-note », 1/2 ronde — s’écrit `2`. Pour des notes plus longues qu’une ronde, vous devrez utiliser les commandes `\longa` pour une longue, et `\breve` pour une brève, aussi appelée carrée.

```

c'\breve
c'1 c'2 c'4 c'8 c'16 c'32 c'64 c'64
r\longa r\breve
r1 r2 r4 r8 r16 r32 r64 r64

```



Si la durée n'est pas précisée, elle est alors assimilée à la durée précédente. La valeur par défaut pour la première note est une noire (4).

```
{ a a a2 a a4 a a1 a }
```



6.2.2 Notes pointées

Pour obtenir des notes pointées, ajoutez simplement un point ('.') au chiffre. Les notes doublement pointées sont créées de la même façon.

```
a'4 b' c'4. b'8 a'4. b'4.. c'8.
```



Commandes prédéfinies

Les points sont normalement haussés pour éviter les lignes de portées, sauf dans certaines polyphonies. Les commandes suivantes peuvent être utilisées pour demander manuellement une orientation particulière des points.

```
\dotsUp, \dotsDown, \dotsNeutral.
```

Voir aussi

Référence du programme : `Dots`, `DotColumn`.

6.2.3 Nolets

Les nolets — triolets, quintolets, etc. — sont obtenus en multipliant toutes les durées d’une expression musicale par une fraction.

```
\times fraction expr_musique
```

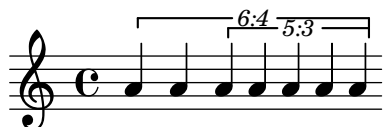
La durée de *expr_musique* sera multipliée par la fraction. Le dénominateur de la fraction sera imprimé au-dessus des notes, parfois avec un crochet. Le nolet le plus courant est le triolet, dans lequel 3 notes ont la durée de 2, et où les notes durent donc $\frac{2}{3}$ de leur valeur écrite.

```
g'4 \times 2/3 {c'4 c' c'} d'4 d'4
```



Les nolets peuvent être imbriqués ; par exemple,

```
\override TupletNumber #'text = #tuplet-number::calc-fraction-text
\times 4/6 {
  a4 a
  \times 3/5 { a a a a a }
}
```



Commandes prédéfinies

`\tupletUp`, `\tupletDown`, `\tupletNeutral`.

Propriétés couramment modifiées

La propriété `tupletSpannerDuration` spécifie la longueur voulue de chaque crochet. Avec elle, vous pouvez faire plusieurs nolets en ne tapant `\times` qu’une fois, ce qui évite une longue saisie. Dans le prochain exemple, deux triolets sont imprimés avec une seule fonction `\times`.

```
\set tupletSpannerDuration = #(ly:make-moment 1 4)
\times 2/3 { c8 c c c c c }
```



Pour plus d’information sur `make-moment`, voir [Section 8.4.2 \[Gestion du temps\]](#), page 132.

L’apparence du chiffre est déterminée par la propriété `text` dans `TupletNumber`. La valeur par défaut imprime seulement le dénominateur, mais si elle est définie par la fonction `tuplet-number::calc-fraction-text`, la fraction entière *num:den* sera imprimée à la place.

Pour éviter d’imprimer les chiffres des nolets, utilisez

```
\times 2/3 { c8 c c } \times 2/3 { c8 c c }
\override TupletNumber #'transparent = ##t
\times 2/3 { c8 c c } \times 2/3 { c8 c c }
```



Utilisez la fonction `\tweak` pour définir une priorité dans le cas de nolets imbriqués débutant au même moment. Dans cet exemple, `\tweak` spécifie un texte sous forme de fraction pour le `TupletNumber` externe et de dénominateur pour les trois triolets internes.

```
\new Staff {
  \tweak #'text #tuplet-number::calc-fraction-text
  \times 4/3 {
    \tweak #'text #tuplet-number::calc-denominator-text
    \times 2/3 { c'8[ c'8 c'8] }
    \times 2/3 { c'8[ c'8 c'8] }
    \times 2/3 { c'8[ c'8 c'8] }
  }
}
```

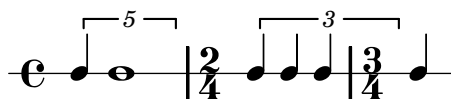


Ici, `\tweak` et `\override` agissent de concert pour spécifier le positionnement du `TupletBracket`. Le premier `\tweak` positionne le `TupletBracket` du nolet externe au dessus de la portée. Le second `\tweak` positionne le `TupletBracket` du premier triolet interne au dessous de la portée. Notez que cette paire de fonctions `\tweak` n'affecte que le triolet extérieur et le premier des trois triolets imbriqués parce qu'ils commencent au même instant musical. Nous utilisons `\override` de manière tout à fait normale, pour forcer le positionnement du `TupletBracket` des deuxième et troisième triolets en dessous de la portée.

```
\new Staff {
  \tweak #'text #tuplet-number::calc-fraction-text
  \tweak #'direction #up
  \times 4/3 {
    \tweak #'direction #down
    \times 2/3 { c'8[ c'8 c'8] }
    \override TupletBracket #'direction = #down
    \times 2/3 { c'8[ c'8 c'8] }
    \times 2/3 { c'8[ c'8 c'8] }
  }
}
```



Les crochets de nolets peuvent aller jusqu'aux prémisses de la mesure suivante, ou à la prochaine note.



Voir aussi

Référence du programme : `TupletBracket`, `TupletNumber`, `TimeScaledMusic`.

6.2.4 Changement d'échelle des durées

Vous pouvez altérer la durée des notes en leur joignant une fraction N/M , donnant $'*N/M'$ — ou $'*N'$ si $M=1$. Ceci n'affectera pas l'apparence des notes ou silences produits.

Dans l'exemple suivant, les trois premières notes prennent exactement deux temps, mais aucun triolet n'est imprimé.

```
\time 2/4
a4*2/3 gis4*2/3 a4*2/3
a4 a4 a4*2
b16*4 c4
```



Voir aussi

Dans ce manuel : [Section 6.2.3 \[Nolets\]](#), page 71.

6.2.5 Vérification des limites de mesure

Les tests de limites de mesure (ou tests de mesure) aident à détecter les erreurs dans les durées. Un test de mesure s'écrit avec une barre verticale, `|`. Lors du traitement, elle doit correspondre à une barre de mesure. Sinon, un avertissement est émis. Dans l'exemple suivant, le deuxième test de mesure signale une erreur.

```
\time 3/4 c2 e4 | g2 |
```

Le test de mesure peut être aussi utilisé dans les paroles, par exemple :

```
\lyricmode {
  \time 2/4
  Twin -- kle | Twin -- kle
}
```

Des durées incorrectes font échouer les tests de mesure, et peuvent souvent mettre la partition sens dessus dessous, particulièrement s'il s'agit de musique polyphonique. Vérifier les tests de mesure qui ont échoué et les durées incorrectes est un bon moyen de commencer à corriger sa partition.

Il est aussi possible d'attribuer une autre valeur au symbole `|`, en assignant une expression musicale à `pipeSymbol`,

```
pipeSymbol = \bar "||"
```

$$\{ c'2 \ c' \mid c'2 \ c' \}$$


Voir aussi

Référence du programme : Completion_heads_engraver.

6.3 Polyphonie

La notion musicale de polyphonie fait référence au fait d'avoir plus d'une voix simultanément dans une pièce. Dans LilyPond, la notion de polyphonie fait référence au fait d'avoir plus d'une voix sur la même portée.

6.3.1 Accords

Un accord est formé en mettant une série de hauteurs entre < et >. Un accord peut être suivi d'une durée et d'indications d'articulation, comme une simple note.

$\langle c_{eg} \rangle_4 \quad \langle c \rangle_8$



Pour plus d'information à propos des accords, voir [Section 7.2 \[Les noms des accords\]](#), [page 117](#).

6.3.2 Hampes

Dès qu'une note est rencontrée, un objet **Stem** se crée automatiquement. Pour les rondes et les silences, ils sont aussi créés, mais en mode invisible.

Commandes prédéfinies

`\stemUp, \stemDown, \stemNeutral.`

Propriétés couramment modifiées

Pour changer la direction des hampes au milieu de la portée, utilisez

```
a4 b c b
\override Stem #'neutral-direction = #up
a4 b c b
\override Stem #'neutral-direction = #down
a4 b c b
```



6.3.3 Polyphonie basique

La manière la plus facile d'entrer des fragments avec plus d'une voix sur une portée est d'entrer chaque voix comme une suite de notes — entre accolades : $\{\dots\}$ — puis de les combiner simultanément, en séparant les voix avec `\l`

```
\new Staff \relative c' {
  c16 d e f
  <<
```

```

{ g4 f e | d2 e2 } \\  

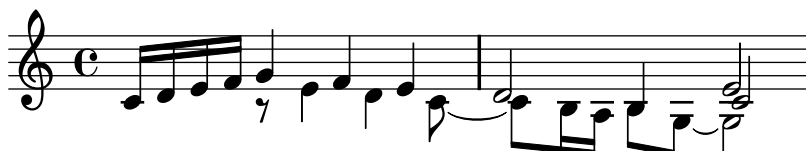
{ r8 e4 d c8 ~ | c b16 a b8 g ~ g2 } \\  

{ s2. | s4 b4 c2 }  

>>  

}

```



Le séparateur permet aux contextes `Voice`¹ d'être identifiés. Les contextes de voix portent les noms "1", "2", etc. Dans chacun de ces contextes, la direction verticale des liaisons, hampes, etc. est réglée de manière appropriée.

Ces voix sont toutes distinctes de la voix qui contient les notes en dehors de la construction `<< \ \ >>`. On doit le prendre en compte lorsqu'on fait des changements au niveau des voix. Cela veut aussi dire que les liaisons de prolongation et d'articulation ne peuvent ni entrer ni sortir de la construction `<< \ \ >>`. À l'inverse, des voix parallèles venant de constructions `<< \ \ >>` séparées sur la même portée sont dans les mêmes voix. Voici le même exemple, avec des têtes de note différentes pour chaque voix. Notez que le changement de style de tête de note n'affecte pas l'intérieur des constructions `<< \ \ >>`. Aussi, le changement à la deuxième voix dans la première construction `<< \ \ >>` l'affecte aussi dans la deuxième construction `<< \ \ >>`, et la voix est liée entre les deux constructions.

```

\new Staff \relative c' {  

  \override NoteHead #'style = #'cross  

  c16 d e f  

  <<  

    { g4 f e } \\  

    { \override NoteHead #'style = #'triangle  

      r8 e4 d c8 ~ }  

  >> |  

  <<  

    { d2 e2 } \\  

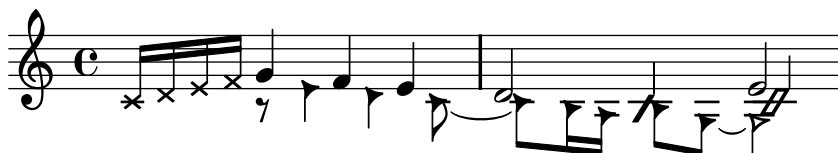
    { c8 b16 a b8 g ~ g2 } \\  

    { \override NoteHead #'style = #'slash s4 b4 c2 }  

  >>  

}

```



La polyphonie ne change pas la relation des notes dans un bloc `\relative { }`. Chaque note est calculée en fonction de la note qui la précède immédiatement, sans tenir compte des différentes voix.

```
\relative { noteA << noteB \ \ noteC >> noteD }
```

`noteC` est calculé relativement à `noteB`, non pas à `noteA` ; `noteD` est calculé relativement à `noteC`, non pas à `noteB` ou `noteA`.

¹ Ces voix polyphoniques sont parfois appelées « couches » ou « calques » dans d'autres logiciels de notation.

6.3.4 Instantiation explicite des voix

Les contextes `Voice` peuvent être déclarés manuellement dans un bloc `<< >>` pour créer de la musique polyphonique, en utilisant `\voiceOne`, ... jusqu'à `\voiceFour` pour assigner des directions de hampes et un déplacement horizontal pour chaque partie.

En particulier,

```
<< \upper \\ \lower >>
```

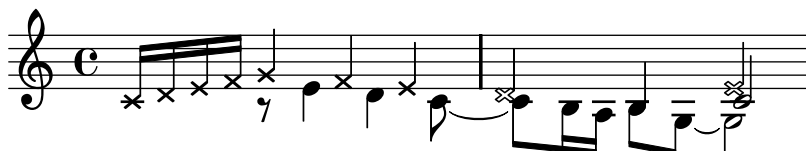
équivalent à

```
<<
  \new Voice = "1" { \voiceOne \upper }
  \new Voice = "2" { \voiceTwo \lower }
>>
```

Les commandes `\voiceXXX` fixent la direction des hampes, des liaisons de prolongations et de phrasé, des articulations, des annotations, des points d'augmentation des notes pointées et des doigtés. `\voiceOne` et `\voiceThree` font pointer ces objets vers le haut, alors que `\voiceTwo` et `\voiceFour` les font pointer vers le bas. La commande `\oneVoice` les ramène aux critères normaux.

Une expression séquentielle qui apparaît en premier dans un `<< >>` appartient à la voix principale. Ceci est utile lorsque des voix supplémentaires apparaissent pendant que la voix principale est jouée. Voici une meilleure réalisation de l'exemple de la section précédente. Les notes en croix mettent en évidence le fait que la mélodie principale est maintenant dans un seul contexte de voix.

```
\new Staff \relative c' {
  \override NoteHead #'style = #'cross
  c16 d e f
  \voiceOne
  <<
    { g4 f e | d2 e2 }
    \new Voice="1" { \voiceTwo
      r8 e4 d c8 ~ | c8 b16 a b8 g ~ g2
      \oneVoice
    }
    \new Voice { \voiceThree
      s2. | s4 b4 c2
      \oneVoice
    }
  >>
  \oneVoice
}
```



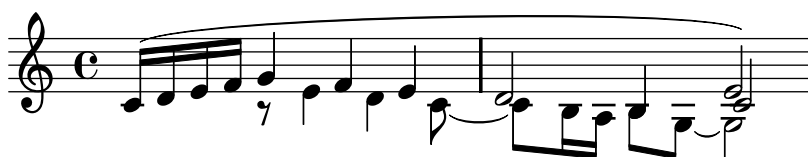
La définition correcte des voix permet à la mélodie d'être liée.

```
\new Staff \relative c' {
  c16^( d e f
  \voiceOne
  <<
```

```

{ g4 f e | d2 e2) }
\context Voice="1" { \voiceTwo
  r8 e4 d c8 ~ | c8 b16 a b8 g ~ g2
  \oneVoice
}
\new Voice { \voiceThree
  s2. s4 b4 c2
  \oneVoice
}
>>
\oneVoice
}

```

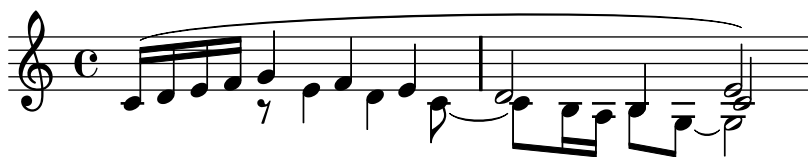


Le fait d'éviter le séparateur `\\` permet aussi d'imbriquer des constructions polyphoniques, ce qui peut être une manière plus naturelle de saisir la musique.

```

\new Staff \relative c' {
  c16^( d e f
  \voiceOne
  <<
    { g4 f e | d2 e2) }
    \context Voice="1" { \voiceTwo
      r8 e4 d c8 ~ |
      <<
        {c8 b16 a b8 g ~ g2}
        \new Voice { \voiceThree
          s4 b4 c2
          \oneVoice
        }
      >>
      \oneVoice
    }
  >>
  \oneVoice
}

```



Dans certaines circonstances de polyphonie complexe, vous pourrez être amené à recourir à des voix supplémentaires afin d'éviter des collisions de notes. Ces voix additionnelles s'ajoutent en définissant un identificateur, comme le montre l'exemple suivant :

```
voiceFive = #(context-spec-music (make-voice-props-set 4) 'Voice)
```

```
\relative c''' <<
  { \voiceOne g4 ~ \stemDown g32[ f( es d c b a b64 )g] } \\
  { \voiceThree b4} \\
  { \voiceFive d,} \\
  { \voiceTwo g,}
>>
```



6.3.5 Résolution des collisions

D'ordinaire, les têtes de notes pointées et non-pointées ne sont pas fusionnées, mais lorsque la propriété `merge-differently-dotted` de l'objet `NoteCollision` est définie, elles se trouvent fusionnées :

```
\new Voice << {
  g8 g8
  \override Staff.NoteCollision
    #'merge-differently-dotted = ##t
  g8 g8
} \\ { g8.[ f16] g8.[ f16] } >>
```



De même, vous pouvez fusionner une tête de blanche avec une tête de croche, en définissant `merge-differently-headed`

```
\new Voice << {
  c8 c4.
  \override Staff.NoteCollision
    #'merge-differently-headed = ##t
  c8 c4. } \\ { c2 c2 } >>
```



LilyPond décale aussi verticalement les silences à l'opposé des hampes, par exemple

```
\new Voice << c''4 \\ r4 >>
```



Lorsque trois notes ou plus s'agglutinent dans un même empilement, `merge-differently-headed` ne peut mener à bien la fusion des deux notes qui devraient l'être. Pour obtenir une fusion optimale, appliquez un décalage (`\shift`) à la note qui ne devrait pas s'empiler. Dans la première mesure de l'exemple suivant, `merge-differently-headed` ne fonctionne pas — la tête de la blanche est noire. Dans la seconde mesure, `\shiftOn` s'applique pour sortir le sol (g) de l'alignement, et `merge-differently-headed` fonctionne correctement.

```
\override Staff.NoteCollision #'merge-differently-headed = ##t
<<
  { d='2 g2 } \
  { \oneVoice d='8 c8 r4 e,8 c'8 r4 } \
  { \voiceFour e,,2 e'2}
>>
<<
  { d='2 \shiftOn g2 } \
  { \oneVoice d='8 c8 r4 e,8 c'8 r4 } \
  { \voiceFour e,,2 e'2}
>>
```



Commandes prédéfinies

`\oneVoice`, `\voiceOne`, `\voiceTwo`, `\voiceThree`, `\voiceFour`.

`\shiftOn`, `\shiftOnn`, `\shiftOnnn`, `\shiftOff` : toutes ces commandes précisent le degré de décalage des notes de la voix courante. Les voix externes — habituellement, les voix une et deux — ont `\shiftOff`, alors que les voix internes — trois et quatre — ont `\shiftOn`. `\shiftOnn` et `\shiftOnnn` sont des niveaux supplémentaires de décalage.

Quand LilyPond est dépassé, la propriété `force-hshift` de l'objet `NoteColumn`, et des silences à hauteur déterminée, peuvent s'avérer utiles pour dicter au programme les choix de placement.

```
\relative <<
{
  <d g>
  <d g>
} \ {
  <b f'>
  \once \override NoteColumn #'force-hshift = #1.7
  <b f'>
} >>
```



Voir aussi

Référence du programme : les objets appropriés pour résoudre les collisions sont `NoteCollision` et `RestCollision`.

Bogues

Quand `merge-differently-headed` est utilisé avec une croche ou une valeur plus courte à hampe vers le haut, et une blanche hampe vers le bas, la croche se retrouve à la mauvaise place.

Il n'y a aucune prise en charge des agrégats dans lesquels une même note apparaît avec différentes altérations. Il est conseillé d'avoir recours aux enharmoniques, ou d'utiliser la notation spécifique de cluster — voir [Section 8.4.4 \[Clusters\], page 132](#).

6.4 Notation sur la portée

De la notation musicale concernant la portée, comme les armures, les clefs ou les chiffres de mesure.

6.4.1 Clefs

La clé indique quelles lignes de la portée correspondent à telles hauteurs. Elle est réglée par la commande `\clef`.

```
{ c'2 \clef alto g'2 }
```



Les clés prises en charge sont

Clef	Position
treble, violin, G, G2	Clé de sol 2e ligne
alto, C	Clé d'ut 3e ligne
tenor	Clé d'ut 4e ligne
bass, F	Clé de fa 4e ligne
french	Clé de sol 1e ligne
soprano	Clé d'ut 1e ligne
mezzosoprano	Clé d'ut 2e ligne
baritone	Clé d'ut 5e ligne
varbaritone	Clé de fa 3e ligne
subbass	Clé de fa 5e ligne
percussion	Clé de percussion
tab	Clé de tablature

En ajoutant `_8` ou `^8` au nom de la clé, celle-ci est transposée à l'octave respectivement inférieure ou supérieure, et `_15` ou `^15` la transpose de deux octaves. L'argument *clefname* doit être mis entre guillemets lorsqu'il contient un caractère « souligné » ou des chiffres. Par exemple,

```
\clef "G_8" c4
```



Propriétés couramment modifiées

La commande `\clef "treble_8"` équivaut à définir `clefGlyph`, `clefPosition` — qui contrôle la position verticale de la clé — `middleCPosition` et `clefOctavation`. Une clé est imprimée lorsque l'une de ces propriétés est changée. Les exemples suivant font apparaître des possibilités de réglage manuel de ces propriétés.

```
{
  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  c'4
  \set Staff.clefGlyph = #"clefs.G"
  c'4
  \set Staff.clefGlyph = #"clefs.C"
  c'4
  \set Staff.clefOctavation = #7
  c'4
  \set Staff.clefOctavation = #0
  \set Staff.clefPosition = #0
  c'4
  \clef "bass"
  c'4
  \set Staff.middleCPosition = #4
  c'4
}
```



Voir aussi

Dans ce manuel : [Section 6.5.7 \[Notes d'ornement\]](#), page 98.

Référence du programme : `Clef`.

6.4.2 Armure

L'armure indique la tonalité dans laquelle la pièce doit être jouée. Elle comprend un ensemble d'altérations (dièses ou bémols) à la clé, c'est-à-dire au début de la portée.

On définit ou modifie l'armure avec la commande `\key`

```
\key hauteur type
```

Ici, *type* doit être `\major` ou `\minor` afin d'avoir respectivement *hauteur-major* ou *hauteur-minor*. Vous pouvez aussi avoir recours aux modes anciens que sont `\ionian`, `\locrian`, `\aeolian`, `\mixolydian`, `\lydian`, `\phrygian`, et `\dorian`.

Cette commande fixe la propriété de contexte `Staff.keySignature`. Des armures inhabituelles peuvent être spécifiées en modifiant directement cette propriété.

Les nouveaux utilisateurs s'embrouillent souvent dans les altérations et les armures, car des notes naturelles prennent ou non un bémol selon l'armure. Pour plus d'informations, voir [Section 6.1.2 \[Altérations\]](#), page 63 ou [Section 2.2.2 \[Altérations et armure\]](#), page 18.

```
\key g \major
f1
fis
```



Propriétés couramment modifiées

Un bécarre est imprimé pour annuler toute altération précédente. Ceci peut être supprimé en réglant la propriété `Staff.printKeyCancellation`.

```
\key d \major
a b cis d
\key g \minor
a bes c d
\set Staff.printKeyCancellation = ##f
\key d \major
a b cis d
\key g \minor
a bes c d
```



Voir aussi

Référence du programme : `KeyCancellation`, `KeySignature`.

6.4.3 Métrique

Le chiffre de mesure indique le mètre d'une pièce : une alternance régulière de temps forts et de temps faibles. Il est indiqué par une fraction au début de la portée.

Le chiffre de mesure est réglé par la commande `\time`.

```
\time 2/4 c'2 \time 3/4 c'2.
```



Propriétés couramment modifiées

Le symbole imprimé peut être modifié avec la propriété `style`. En la réglant sur `#'()`, une fraction sera utilisée pour les chiffres de mesure 4/4 et 2/2.

```
\time 4/4 c'1
\time 2/2 c'1
\override Staff.TimeSignature #'style = #'()
\time 4/4 c'1
\time 2/2 c'1
```

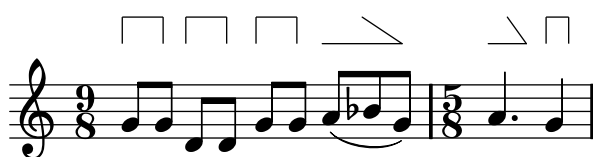


Il y a bien d'autres options pour sa mise en place. Voir [Section 7.7.6 \[Métriques anciennes\]](#), [page 120](#) pour plus d'exemples.

`\time` définit les propriétés `timeSignatureFraction`, `beatLength` et `measureLength` dans le contexte `Timing`, qui en principe est assimilé à `Score`. La propriété `measureLength` détermine où des barres de mesure doivent être insérées, et comment les groupements de notes doivent être gérés. La modification de la valeur de `timeSignatureFraction` donne également lieu à l'impression d'un symbole.

Plus d'options sont accessibles au moyen de la fonction Scheme `set-time-signature`. De concert avec le `Measure_grouping_engraver`, elle crée les signes de `MeasureGrouping`, qui facilitent la lecture de musiques modernes, complexes rythmiquement. Dans l'exemple suivant, les mesures à 9/8 sont subdivisées en (2 2 2 3), ce qui est donné comme argument à la commande `set-time-signature`, en troisième position.

```
\score {
  \relative c'' {
    #(set-time-signature 9 8 '(2 2 2 3))
    g8[ g] d[ d] g[ g] a8[( bes g]) |
    #(set-time-signature 5 8 '(3 2))
    a4. g4
  }
  \layout {
    \context {
      \Staff
      \consists "Measure_grouping_engraver"
    }
  }
}
```



Voir aussi

Référence du programme : `TimeSignature` et `Timing_translator`.

Exemples : ‘`contemporary/compound-time-signature.ly`’.

Bogues

Le groupement automatique des ligatures n'utilise pas les groupements spécifiés par `set-time-signature`.

6.4.4 Mesures incomplètes

Les mesures incomplètes, telles que les anacrouses ou levées, doivent être entrées avec la commande

```
\partial 16*5 c16 cis d dis e | a2. c,4 | b2
```



La syntaxe de cette commande est

```
\partial durée
```

durée étant la valeur rythmique devant être ajoutée avant la mesure suivante.

Le programme traduit cette commande en

```
\set Timing.measurePosition = -durée
```

La propriété `measurePosition` contient un nombre rationnel qui indique, à ce point précis, où l'on en est de la mesure. Notez qu'il peut s'agir d'un nombre négatif ; `\partial 4` signifie, pour le programme : « Dans cette mesure, il reste juste une noire ».

Bogues

Cette commande ne prend pas en compte les notes d'ornement ou appoggiatures au début de la musique. Lorsqu'un morceau commence par une levée et avec des petites notes, la commande `\partial` devrait être placée après celles-ci.

```
\grace f16
\partial 4
g4
a2 g2
```



`\partial` n'est destiné à être utilisé qu'en début de pièce. Si on l'utilise ailleurs qu'au début, des messages d'erreurs peuvent s'afficher.

6.4.5 Barres de mesures

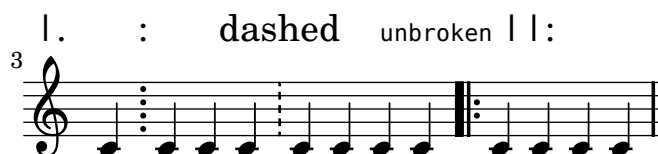
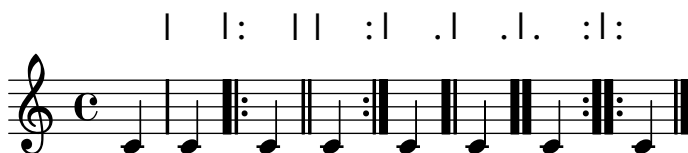
Les barres de mesures délimitent les mesures, mais peuvent aussi indiquer une reprise. En principe, elles sont insérées automatiquement, et les sauts de ligne ne peuvent avoir lieu qu'au niveau de ces barres.

Il est possible de forcer l'impression d'une barre de mesure spéciale, avec la commande `\bar` :

```
c4 \bar "|:" c4
```



Les styles de barres de mesure disponibles sont



En plus de cela, on peut demander `"||:"`, qui équivaut à `"|:"`, mais qui donnera, en cas de saut de ligne, une double barre en fin de ligne, et une barre de reprise au début de la ligne suivante.

Il est possible d'autoriser un saut de ligne même s'il n'y a pas de barre de mesure visible, en utilisant :

```
\bar ""
```

Ceci insérera une barre de mesure invisible, et permettra de sauter de ligne à cet endroit, sans incrémenter le numéro de mesure.

Dans une partition comprenant plusieurs portées, la commande `\bar` placée sur une portée s'applique automatiquement à toutes les portées. Les barres de mesure que l'on obtient alors sont d'un seul tenant sur les portées d'un `StaffGroup`, d'un `PianoStaff` ou d'un `GrandStaff`.

```
<<
\new StaffGroup <<
  \new Staff {
    e'4 d'
    \bar "||"
    f' e'
```

```

    }
    \new Staff { \clef bass c4 g e g }
  >>
  \new Staff { \clef bass c2 c2 }
  >>

```



Propriétés couramment modifiées

La commande `\bar bartype` sert de raccourci pour `\set Timing.whichBar = bartype`. Dès que l'on définit `whichBar`, une barre de mesure est créée selon le style défini.

Dès que la propriété `whichBar` est définie, une barre de mesure est créée. À chaque début de mesure, elle prend la valeur de `Timing.defaultBarType`. La valeur de `repeatCommands` sert à remplacer les barres de mesure par défaut.

Nous vous invitons à utiliser `\repeat` pour indiquer les reprises. Voyez à ce sujet [Section 6.7 \[Répétitions\]](#), page 110.

Voir aussi

Dans ce manuel : [Section 6.7 \[Répétitions\]](#), page 110, [Section 6.4.7 \[Groupements de systèmes\]](#), page 88.

Référence du programme : `BarLine` (faisant partie du contexte `Staff`), `SpanBar` (sur plusieurs portées).

6.4.6 Musique sans métrique

Les barres de mesure et les numéros de mesure sont calculés automatiquement, ce qui n'est pas souhaitable dans le cas d'une musique non mesurée — les cadences, par exemple. Les commandes `\cadenzaOn` et `\cadenzaOff` permettent de désactiver et de rétablir la métrique automatique.

```

c4 d e d
\cadenzaOn
c4 c d8 d d f4 g4.
\cadenzaOff
\bar "|"
d4 e d c

```



Bogues

LilyPond ne change de ligne ou de page qu'au niveau des barres de mesure. Si votre musique non mesurée dure plus d'une ligne, il vous faudra insérer des barres de mesure invisibles, pour indiquer à quels endroits un saut de ligne peut intervenir.

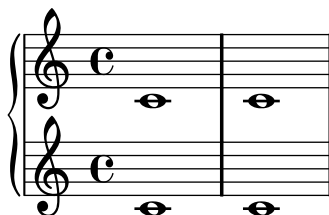
```
\bar ""
```

6.4.7 Groupements de systèmes

De nombreuses partitions sont écrites sur plusieurs portées. Ces portées peuvent être regroupées de quatre manières différentes.

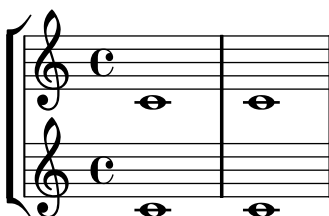
- Le groupe de portées est attaché par une accolade sur la gauche, et les barres de mesure sont d'un seul tenant. Il s'agit du contexte **GrandStaff**.

```
\new GrandStaff
\relative <<
  \new Staff { c1 c }
  \new Staff { c c }
>>
```



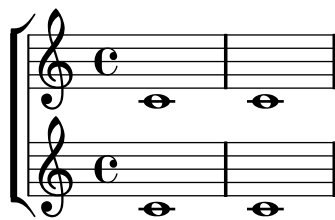
- Le groupe de portées est attaché par un crochet, et les barres de mesure sont d'un seul tenant. Il s'agit du contexte **StaffGroup**.

```
\new StaffGroup
\relative <<
  \new Staff { c1 c }
  \new Staff { c c }
>>
```



- Le groupe de portées est attaché par un crochet, mais les barres de mesure sont séparées d'une portée à l'autre. Il s'agit du contexte **ChoirStaff**.

```
\new ChoirStaff
\relative <<
  \new Staff { c1 c }
  \new Staff { c c }
>>
```



- Les portées du groupe ne sont pas attachées (hormis par une simple ligne verticale). Les barres de mesure sont détachées. Il s'agit de l'assemblage par défaut.

```
\relative <<
  \new Staff { c1 c }
  \new Staff { c c }
>>
```



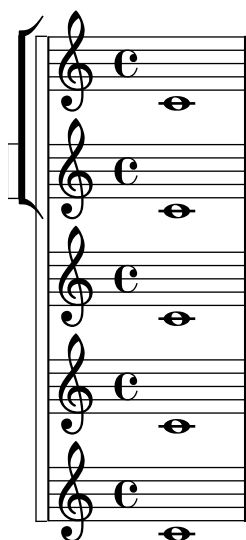
Voir aussi

Les barres de mesure au début de chaque système prennent l'un des styles `SystemStartBar`, `SystemStartBrace`, `SystemStartBracket`. Dans chaque contexte, seul l'un de ces styles est utilisé, et c'est la propriété `systemStartDelimiter` qui détermine lequel.

Propriétés couramment modifiées

Les accolades et crochets délimitant les systèmes peuvent être imbriqués en profondeur,

```
\new StaffGroup
\relative <<
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare (SystemStartBracket a (SystemStartSquare b)) d)
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
>>
```



6.4.8 Symbole de la portée

Les notes, nuances, etc. sont regroupés dans un assemblage de lignes horizontales, que l'on nomme la portée (en anglais « staff », et « staves » au pluriel). Dans LilyPond, ces lignes sont dessinées au moyen d'un objet de mise en forme à part entière, nommé **staff symbol** — symbole de portée.

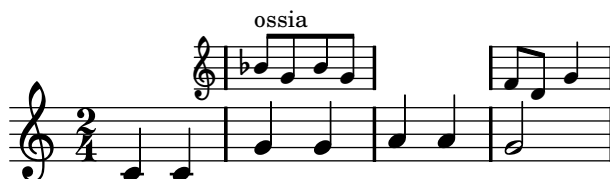
L'aspect du symbole de portée peut être modifié selon différentes propriétés, telles que le nombre de lignes, leur épaisseur, ou leur éloignement. Vous en trouverez des exemples dans '[staff/changing-the-number-of-lines-in-a-staff.ly](#)' et '[staff/changing-the-staff-size.ly](#)'.

De plus, la portée peut commencer et s'arrêter où l'on veut, grâce aux commandes `\startStaff` et `\stopStaff`.

```
b4 b
\override Staff.StaffSymbol #'line-count = 2
\stopStaff \startStaff
b b
\revert Staff.StaffSymbol #'line-count
\stopStaff \startStaff
b b
```



Cette manière de procéder peut être utilisée pour introduire des « ossias », ou dans des partitions à nombre de portées variable, comme sur l'exemple suivant :



Voir aussi

Référence du programme : `StaffSymbol`.

D'autres exemples : '[staff](#)'.

6.4.9 Saisir la musique en parallèle

On peut écrire plusieurs voix de façon entremêlée :

```
\parallelMusic #'(voiceA voiceB) {
  r8      g'16[ c'' ] e''[ g' c'' e'' ] r8      g'16[ c'' ] e''[ g' c'' e'' ] |
  c'2                                          c'2 |
  r8      a'16[ d'' ] f''[ a' d'' f'' ] r8      a'16[ d'' ] f''[ a' d'' f'' ] |
  c'2                                          c'2 |
}
\new StaffGroup <<
  \new Staff \new Voice \voiceA
  \new Staff \new Voice \voiceB
>>
```



Ceci fonctionne bien avec la musique pour piano :

```
music = {
  \key c \major
  \time 4/4
  \parallelMusic #'(voiceA voiceB voiceC voiceD) {
    % Bar 1
    r8 g'16[ c'' ] e''[ g' c'' e'' ] r8 g'16[ c'' ] e''[ g' c''
e'' ] |
    c'2                                          c'2 |
    r8 a16[ d' ] f'[ a d' f']          r8 a16[ d' ] f'[ a d' f'] |
    c2                                          c2 |

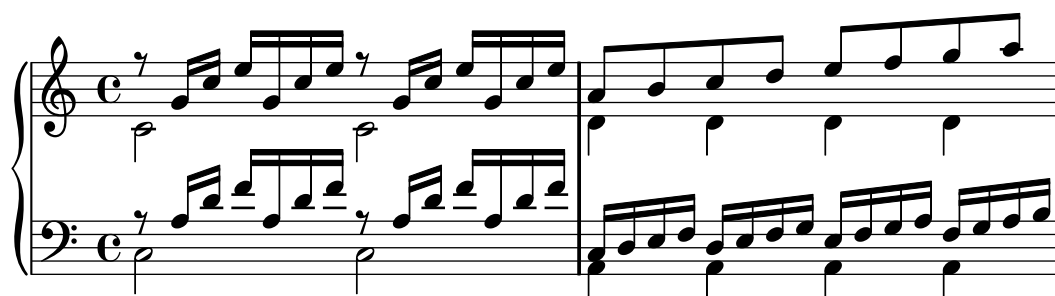
    % Bar 2
    a'8 b'      c'' d''      e'' f''      g'' a'' |
    d'4         d'         d'         d' |
    c16 d e f    d e f g    e f g a    f g a b |
    a,4         a,4         a,4         a,4 |

    % Bar 3 ...
  }
}
```

```

\score {
  \new PianoStaff <<
    \music
    \new Staff <<
      \voiceA \\\
      \voiceB
    >>
    \new Staff {
      \clef bass
      <<
        \voiceC \\\
        \voiceD
      >>
    }
  >>
}

```



6.5 Relier des groupes de notes

De la notation qui affecte des groupes de notes.

6.5.1 Liaisons de prolongation

Une liaison de tenue (ou de prolongation) relie deux notes adjacentes de même hauteur. Dans les faits, elle prolonge la durée d'une note, et ne doit donc pas être confondue avec les liaisons d'articulation ou de phrasé. Une liaison de tenue est indiquée au moyen d'un tilde '~'.

$e' \sim e' < c' e' g' > \sim < c' e' g' >$



Quand une liaison de tenue se trouve entre deux accords, toutes les notes de même hauteur entre ces deux accords sont reliées. S'il n'y en a aucune, aucune liaison n'est créée. Il est également possible de lier partiellement deux accords, en mettant les liaisons à l'intérieur des accords.

$< c \sim e \ g \sim b > < c \ e \ g \ b >$



Une liaison de tenue est un moyen parmi d'autres pour prolonger la durée d'une note, tout comme les points. L'exemple suivant montre deux manières de matérialiser exactement la même idée :



Les liaisons de tenues sont utilisées soit lorsque la note dépasse de la mesure, soit quand les points ne suffisent pas à donner la bonne durée. Lorsque l'on utilise ces liaisons, les valeurs rythmiques les plus longues doivent s'aligner sur les subdivisions de la mesure, comme ici :



Lorsque l'on doit lier de nombreuses notes sur plusieurs mesures, il devient plus facile d'avoir recours à la division automatique des notes — voir [Section 6.2.7 \[Découpage automatique des notes\]](#), page 74. Ce procédé divise automatiquement les notes trop longues, et les lie par-delà les barres de mesure.

Lorsqu'une mesure de seconde fois après une reprise commence sur une note liée, la liaison doit être répétée. C'est à cela que sert la commande `\repeatTie` :



Propriétés couramment modifiées

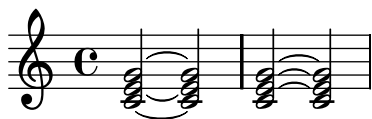
Les liaisons de tenue servent parfois à rendre un accord arpégé. Dans ce cas, les notes liées ne sont pas toutes consécutives. Il faut alors définir à *vrai* ('t' pour 'true') la propriété `tieWaitForNote`. Cette même méthode peut servir, par exemple, à lier un trémolo à un accord.

```
\set tieWaitForNote = ##t
\grace { c16[~ e~ g]~ } <c, e g>2
\repeat "tremolo" 8 { c32~ c'~ } <c c,>1
e8~ c~ a~ f~ <e' c a f>2
```



Il est possible de graver manuellement les liaisons de tenue, en changeant la propriété `tie-configuration`. Pour chaque paire, le premier nombre indique la distance à la portée, en espaces de portée, et le deuxième nombre indique la direction — 1 pour haut, -1 pour bas.

```
<c e g>2~ <c e g> |
\override TieColumn #'tie-configuration =
  #'((0.0 . 1) (-2.0 . 1) (-4.0 . 1))
<c e g>~ <c e g> |
```



Commandes prédéfinies

`\tieUp`, `\tieDown`, `\tieNeutral`, `\tieDotted`, `\tieDashed`, `\tieSolid`.

Voir aussi

Dans ce même manuel : [Section 6.2.7 \[Découpage automatique des notes\]](#), page 74.

Référence du programme : `Tie`.

Bogues

Un changement de portée, lorsqu'une liaison de tenue est active, ne peut produire une liaison oblique.

Le changement de clé ou d'octave pendant une liaison de tenue produit un résultat indéfini. Dans ces cas-là, il est préférable d'utiliser un legato.

6.5.2 Liaisons d'articulation

Une liaison d'articulation indique que les notes doivent être jouées liées, ou *legato*. Ces liaisons s'indiquent au moyen de parenthèses.

```
f( g a) a8 b( a4 g2 f4)
<c e>2( <b d>2)
```



On peut indiquer l'orientation des liaisons suivantes avec `\slurDIR`, `DIR` pouvant être `Up` pour une liaison vers le haut, `Down` pour une liaison vers le bas, ou `Neutral` pour laisser LilyPond décider.

Il existe également un raccourci pratique pour forcer l'orientation d'une seule liaison. Il suffit pour cela d'ajouter `_` ou `^` avant d'ouvrir une parenthèse.

```
c4_( c) c^( c)
```

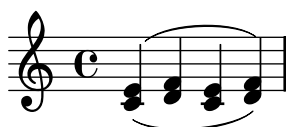


Une seule liaison d’articulation peut être imprimée à la fois. S’il est nécessaire d’imprimer une liaison plus longue, englobant des liaisons plus courtes, utilisez des [Section 6.5.3 \[Liaisons de phrasé\]](#), page 95.

Propriétés couramment modifiées

Certains auteurs utilisent deux liaisons lorsqu’ils veulent lier des accords. Dans LilyPond, il faut pour cela définir à *vrai* (‘true’) la propriété `doubleSlurs` :

```
\set doubleSlurs = ##t
<c e>4 ( <d f> <c e> <d f> )
```



Commandes prédéfinies

`\slurUp`, `\slurDown`, `\slurNeutral`, `\slurDashed`, `\slurDotted`, `\slurSolid`.

Voir aussi

Référence du programme : `internals document`, `Slur`.

6.5.3 Liaisons de phrasé

Une liaison de phrasé relie plusieurs notes en délimitant une phrase musicale. On indique les points de départ et d’arrivée avec `\(` et `\)` respectivement.

```
\time 6/4 c'\( d( e) f( e) d\)
```



D’un point de vue typographique, rien ne distingue une liaison de phrasé d’une liaison d’articulation. Cependant, LilyPond les considère comme des objets différents. Une commande `\slurUp` n’affectera donc pas une liaison de phrasé : il faut plutôt utiliser `\phrasingSlurUp`, `\phrasingSlurDown` ou `\phrasingSlurNeutral`.

Il n’est pas possible d’avoir plusieurs liaisons de phrasé en même temps.

Commandes prédéfinies

`\phrasingSlurUp`, `\phrasingSlurDown`, `\phrasingSlurNeutral`.

Voir aussi

Référence du programme : `PhrasingSlur`.

6.5.4 Liaisons “laissez vibrer”

Les liaisons « Laissez vibrer » (L.V.) sont utilisées pour le piano, la harpe, et certains instruments de percussion. Elles indiquent à l’instrumentiste de laisser sonner la note ou l’accord au lieu de l’étouffer. Cet effet s’obtient avec la commande `\laissezVibrer`.

```
<c f g>\laissezVibrer
```



Voir aussi

Référence du programme : `LaissezVibrerTie LaissezVibrerTieColumn`

D’autres exemples : ‘`connecting/laissez-vibrer-ties.ly`’

6.5.5 Barres de ligature automatiques

LilyPond décide automatiquement de la manière de grouper les notes et d’imprimer les ligatures.

```
\time 2/4 c8 c c c \time 6/8 c c c c8. c16 c8
```



Lorsque ces décisions automatiques ne sont pas satisfaisantes, on peut définir des groupements manuellement — voir section suivante — ou personnaliser les groupements automatiques — voir [Section 9.1.2 \[Définir les règles de ligatures automatiques\], page 134](#).

La commande `\noBeam` peut servir à empêcher des notes individuelles d’être rattachées aux autres.

```
\time 2/4 c8 c\noBeam c c
```



Voir aussi

Référence du programme : `Beam`.

6.5.6 Barres de ligature manuelles

Dans certaines situations, il peut s’avérer nécessaire de supplanter l’algorithme de groupement automatique des notes, par exemple pour prolonger une ligature par-dessus un silence ou une barre de mesure. Le début et la fin de la ligature sont alors indiqués par [et].

```
{
  r4 r8[ g' a r8] r8 g[ | a] r8
}
```



Propriétés couramment modifiées

En principe, les sous-groupes à l'intérieur d'un groupement de notes sont déterminés automatiquement. Cependant, les propriétés `stemLeftBeamCount` et `stemRightBeamCount` permettent d'ajuster ce comportement. Lorsque l'une ou l'autre de ces propriétés est définie, elle ne s'applique qu'une seule fois, après quoi sa définition est effacée.

```
{
  f8[ r16
    f g a]
  f8[ r16
    \set stemLeftBeamCount = #1
    f g a]
}
```



La propriété `subdivideBeams` sert à grouper les double-croches ou les valeurs plus brèves pulsation par pulsation, la pulsation étant définie par la propriété `beatLength`.

```
c16[ c c c c c c c]
\set subdivideBeams = ##t
c16[ c c c c c c c]
\set Score.beatLength = #(ly:make-moment 1 8)
c16[ c c c c c c c]
```



Pour plus d'information sur `make-moment`, voir [Section 8.4.2 \[Gestion du temps\]](#), page 132.

Lorsqu'une ligature entre des notes franchit une barre de mesure, le saut de ligne est en principe interdit à cet endroit. Ce comportement peut être outrepassé en définissant `breakable`.

LilyPond insère automatiquement des ligatures coudées — certaines notes vers le haut, d'autres vers le bas — lorsqu'il détecte un espace important entre des têtes de notes. Ce comportement peut être changé par l'intermédiaire de l'objet `auto-knee-gap`

Bogues

Les ligatures coudées à cheval sur deux portées ne peuvent être employées en même temps que des portées invisibles. Voir [Section 8.3.2 \[Masquer des portées\]](#), page 131.

Les ligatures peuvent entrer en collision avec des symboles entourant les notes, contrairement aux textes ou aux altérations.

6.5.7 Notes d'ornement

Les petites notes sont des ornements entièrement écrits. Les plus courantes sont les acciaccatures, qui doivent se jouer très vite, et qui s'écrivent sous forme d'une petite note barrée (sur la hampe) et liée. L'appoggiature est une petite note non barrée, qui vole une fraction à la durée de la note réelle qui la suit.

Ces petites notes sont entrées avec les commandes `\acciaccatura` et `\appoggiatura`, comme le montre l'exemple suivant :

```
b4 \acciaccatura d8 c4 \appoggiatura e8 d4
\acciaccatura { g16[ f] } e4
```



Ce sont là deux formes spéciales de la commande `\grace`, qui prend en charge toutes les petites notes. Si on la fait suivre d'une expression musicale, un groupe de petites notes sera créé, sans impact sur la métrique.

```
c4 \grace c16 c4
\grace { c16[ d16] } c2 c4
```



Contrairement à `\acciaccatura` ou `\appoggiatura`, la commande `\grace` ne provoque pas de liaison.

La durée des petites notes est interprétée par le programme en fonction d'un deuxième compteur de temps, le chronomètre `grace`. Chaque instant est défini par deux nombres rationnels : le premier compte les durées réelles, le second compte la durée des petites notes. Reprenons l'exemple ci-dessus en y ajoutant ces couples de nombres :



Les petites notes se placent de façon synchrone entre les différentes portées. Dans l'exemple suivant, il y a deux petites double-croches pour chaque petite croche.

```
<< \new Staff { e4 \grace { c16[ d e f] } e4 }
\new Staff { c4 \grace { g8[ b] } c4 } >>
```



La commande `\afterGrace` sert à placer une petite note après une note réelle — et non *avant* comme d'ordinaire. Cette commande requiert deux arguments : la note réelle, et la ou les petites notes qui suivent.

```
c1 \afterGrace d1 { c16[ d] } c4
```



Les petites notes se placent alors aux 3/4 de la durée de la note réelle. Cette fraction peut être changée en définissant `afterGraceFraction` ; ainsi,

```
#(define afterGraceFraction (cons 7 8))
```

placera la petite note à 7/8 de la note réelle.

On peut obtenir le même effet manuellement, de la façon suivante :

```
\new Voice {
  << { d1^\trill_( }
    { s2 \grace { c16[ d] } } >>
  c4)
}
```



Le silence invisible peut être plus ou moins long — ici c'est une demi-pause — afin d'ajuster l'espace entre la note réelle et les petites notes.

Les expressions `\grace` obéissent à des règles typographiques particulières, notamment pour régler l'orientation et la taille des objets. De ce fait, toute subtilité de mise en forme devra être indiquée à l'intérieur de l'expression introduite par `\grace` :

```
\new Voice {
  \acciaccatura {
    \stemDown
    f16->
    \stemNeutral
  }
  g4
}
```



Tous les réglages ajoutés doivent également être désactivés dans cette même expression.

Il est possible de changer globalement la mise en forme des petites notes dans un morceau, au moyen de la fonction `add-grace-property`. Ici, par exemple, on ôte la définition de l'orientation des objets `Stem` pour toutes les petites notes, afin que les hampes ne soient pas toujours orientées vers le haut.

```
\new Staff {
  #(add-grace-property 'Voice 'Stem 'direction '())
  ...
}
```

Il est par ailleurs possible de modifier les variables `startGraceMusic`, `stopGraceMusic`, `startAcciaccaturaMusic`, `stopAcciaccaturaMusic`, `startAppoggiaturaMusic`, et `stopAppoggiaturaMusic`. Pour plus de détails, voir le fichier `ly/grace-init.ly`.

Le trait que l'on trouve sur les hampes des acciaccatures peut être appliqué dans d'autres situations en définissant `\override Stem #'stroke-style = #"grace"`.

Propriétés couramment modifiées

Il est possible de forcer l'élasticité de l'espacement des notes d'agrément.



Voir aussi

Référence du programme : `GraceMusic`.

Bogues

Une partition commençant par une expression `\grace` doit faire intervenir la commande `\new Voice`, sans quoi la note réelle et la petite note se retrouveront sur des portées différentes.

La synchronisation des petites notes se fait de façon parfois surprenante, car les autres objets de la portée — barres de mesure, armures, etc. — sont eux aussi synchrones. Pensez-y lorsque vous mêlez des portées comprenant des petites notes et d'autres sans :

```
<< \new Staff { e4 \bar "|:" \grace c16 d4 }
    \new Staff { c4 \bar "|:" d4 } >>
```



Il est possible de remédier à cela en insérant sur les autres portées des silences invisibles dans une expression précédée de `\grace`, correspondant à la durée des petites notes.

```
\new Staff { c4 \bar "|:" \grace s16 d4 }
```

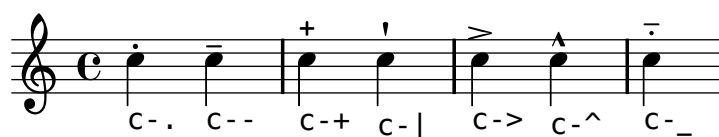
Seules des expressions musicales séquentielles peuvent être utilisées pour des petites notes ; il n'est pas possible d'imbriquer ni de juxtaposer des sections de petites notes, faute de quoi le traitement du code peut échouer ou produire des erreurs.

6.6 Signes d'interprétation

Les signes d'interprétation aident les musiciens à faire en sorte que la musique soit plus qu'un assemblage de notes et de rythmes.

6.6.1 Articulations

Différents symboles peuvent être ajoutés au-dessus ou au-dessous des notes pour indiquer des ponctuations ou des modes de jeu différents. On les ajoute à chaque note au moyen d'un tiret suivi du caractère correspondant à l'articulation désirée. En voici une démonstration :



Il est possible de changer la signification de ces raccourcis : voir des exemples dans `'ly/script-init.ly'`.

Même si LilyPond place automatiquement ces symboles, il est possible de l'obliger à les placer au-dessus ou en-dessous de la note, tout comme d'autres objets, en utilisant respectivement `^` et `_`.

```
c''4^^ c''4_^
```

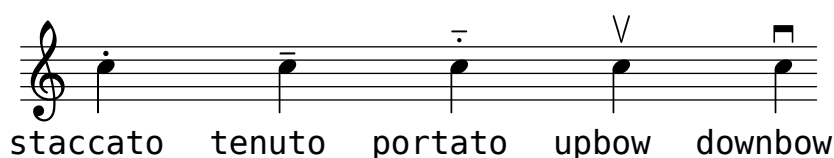
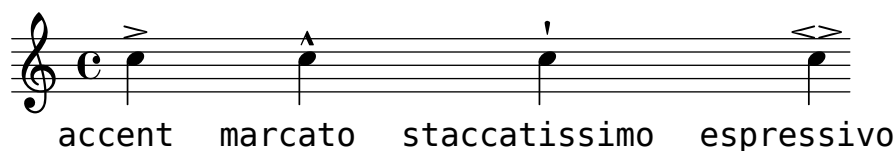


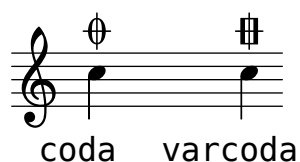
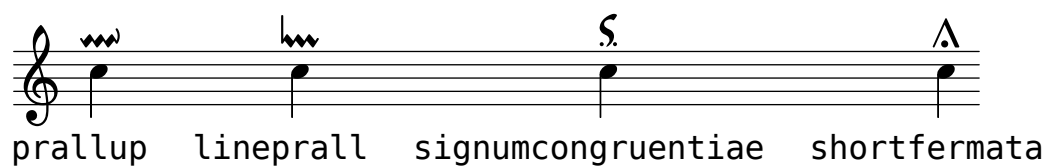
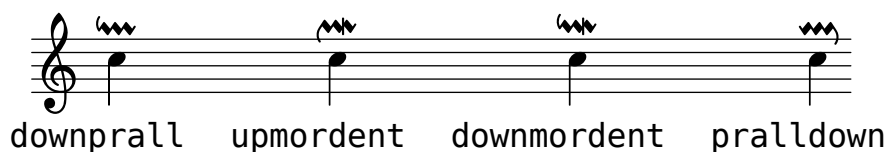
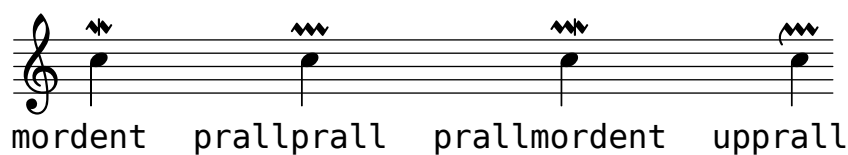
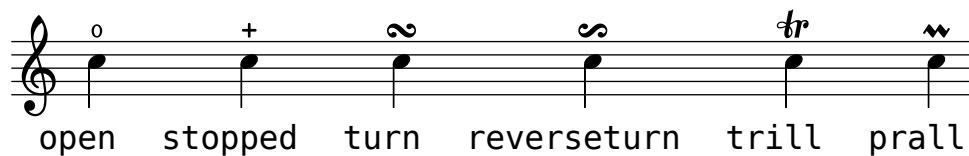
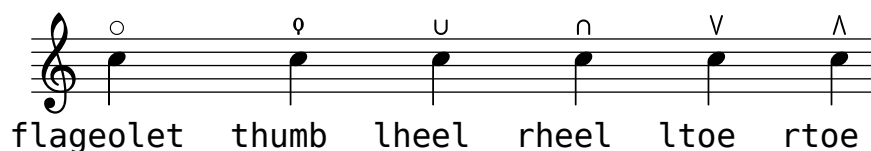
On peut ajouter d'autres symboles, avec la syntaxe `note\symbole`. Ici encore, on peut forcer leur orientation avec `^` and `_`.

```
c\fermata c^\fermata c_\fermata
```



Voici la liste exhaustive des symboles :





Propriétés couramment modifiées

Les symboles s'ordonnent verticalement suivant la propriété `script-priority`. Plus sa valeur numérique est faible, plus le symbole sera proche de la note. Dans l'exemple suivant, l'objet `TextScript` — le dièse — a d'abord la propriété la plus basse, et il est donc placé plus près de la note ; ensuite c'est l'objet `Script` qui a la propriété la plus basse, et il se place alors sous le dièse. Lorsque deux objets ont la même priorité, c'est l'ordre dans lequel ils sont indiqués qui détermine lequel sera placé en premier.

```
\once \override TextScript #'script-priority = #-100
a4^\prall^\markup { \sharp }
```

```
\once \override Script #'script-priority = #-100
a4^\prall^\markup { \sharp }
```



Voir aussi

Référence du programme : `Script`.

Bogues

Ces symboles sont présents sur la partition imprimée, mais n'ont pas d'effet sur le rendu de la musique en MIDI.

6.6.2 Doigtés

Les doigtés peuvent être indiqués comme suit :

note-chiffre_du_doigt

Pour les substitutions de doigtés, on a recours à une indication textuelle (commande `\markup`) de doigté (commande `\finger`).

```
c4-1 c-2 c-3 c-4
c^\markup { \finger "2 - 3" }
```



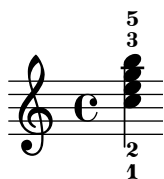
La commande `\thumb` peut être utilisée pour indiquer, par exemple dans une partition de violoncelle, si une note doit être jouée avec le pouce ('thumb' en anglais).

```
<a_\thumb a'-3>8 <b_\thumb b'-3>
```



Les doigtés des accords peuvent être saisis note par note, en les indiquant après chaque hauteur de note.

```
< c-1 e-2 g-3 b-5 >4
```



Propriétés couramment modifiées

On peut contrôler précisément les doigtés des accords en réglant la propriété `fingeringOrientations`.

```
\set fingeringOrientations = #'(left down)
<c-1 es-2 g-4 bes-5 > 4
\set fingeringOrientations = #'(up right down)
<c-1 es-2 g-4 bes-5 > 4
```



Cette propriété permet également, dans de la musique monophonique, d'indiquer des doigtés très proches des têtes de notes.

```
\set fingeringOrientations = #'(right)
<es'-2>4
```



Voir aussi

Référence du programme : `Fingering`.

D'autres exemples : `'expressive/fingering-chords.ly'`.

6.6.3 Nuances

À chaque nuance absolue correspond une commande, qui peut être indiquée après une note : `c4\ff` par exemple. Les commandes de nuances disponibles sont `\ppppp`, `\pppp`, `\ppp`, `\pp`, `\p`, `\mp`, `\mf`, `\f`, `\ff`, `\fff`, `\ffff`, `\fp`, `\sf`, `\sff`, `\sp`, `\spp`, `\sfz`, et `\rfz`.

```
c\ppp c\pp c \p c\mp c\mf c\f c\ff c\fff
c2\fp c\sff c\sff c\sp c\spp c\sff c\rfz
```



Un crescendo est délimité par `\<` et `\!`, ou peut se terminer par une commande de nuance. Au lieu de `\<` ou `\>`, on peut aussi utiliser `\cr` et `\decr`. Ces indications sont attachées aux notes ; aussi si l'on veut faire se succéder plusieurs nuances pendant une note tenue, il faudra avoir recours à des silences invisibles :

```
c\< c\! d\> e\!  
<< f1 { s4 s4\< s4\! \> s4\! } >>
```



En principe, un soufflet — (de)crescendo imprimé sous forme graphique — commence au bord gauche de la note de départ, et se termine au bord droit de la note d'arrivée. Cependant, si la note d'arrivée est sur un premier temps, le soufflet s'arrêtera au niveau de la barre de mesure qui la précède. Ce comportement peut être annulé en définissant à *faux* (lettre 'f') la propriété `hairpinToBarline` :

```
\set hairpinToBarline = ##f  
c4\< c2. c4\!
```



On peut avoir recours à l'indication `\espressivo` pour indiquer un crescendo puis un decrescendo sur une seule note.

```
c2 b4 a g1\espressivo
```



Cependant, ces deux indications graphiques peuvent se trouver très comprimées horizontalement. Pour les rallonger, on peut modifier la propriété `minimum-length` de `Voice.Hairpin` — `Voice` étant le contexte, et `Hairpin` l'objet affecté. Par exemple :

```
\override Voice.Hairpin #'minimum-length = #5
```

Des crescendos ou decrescendos *al niente* peuvent être indiqués de manière graphique, en définissant à *vrai* (lettre 't') la propriété `circled-tip`, ce qui affiche un cercle à leur extrémité.

```
\override Hairpin #'circled-tip = ##t  
c2\< c\!  
c4\> c\< c2\!
```



Au lieu d'une notation graphique des crescendos, on peut utiliser une notation textuelle.

```
\setTextCresc
c\< d e f\!
\setHairpinCresc
e\> d c b\!
\setTextDecresc
c\> d e f\!
\setTextDim
e\> d c b\!
```



On peut même définir ses propres indications textuelles :

```
\set crescendoText = \markup { \italic "cresc. poco" }
\set crescendoSpanner = #'dashed-line
a'2\< a a a\!\mf
```



Pour créer des indications de nuances qui restent alignées avec les nuances habituelles, voir [Section 8.1.11 \[Personnalisation des indications de nuance\]](#), page 130.

Le positionnement vertical des nuances est géré par le `DynamicLineSpanner`.

Propriétés couramment modifiées

Des nuances différentes situées — ou commençant — sur une même note seront alignées verticalement. Pour aligner des nuances qui ne se situeraient pas sur une même note, il est possible d'augmenter la propriété `staff-padding`.

```
\override DynamicLineSpanner #'staff-padding = #4
```

Cette propriété peut aussi servir à régler des problèmes de collision entre des nuances et d'autres objets.

Les crescendos ou decrescendos qui aboutissent sur la première note d'une nouvelle ligne ne sont imprimés que jusqu'à la fin de la ligne précédente. Ce comportement peut être outrepassé en définissant :

```
\override Score.Hairpin #'after-line-breaking = ##t
```

Les crescendos et decrescendos indiqués textuellement — tels que *cresc.* ou *dim.* — sont suivis de pointillés qui montrent leur étendue. On peut empêcher l'impression de ces pointillés avec :

```
\override DynamicTextSpanner #'dash-period = #-1.0
```

Commandes prédéfinies

`\dynamicUp`, `\dynamicDown`, `\dynamicNeutral`.

Voir aussi

Référence du programme : `DynamicText`, `Hairpin`.

Le placement vertical de ces éléments graphiques est géré par le `DynamicLineSpanner`.

6.6.4 Signes de respiration

Les indications de respiration sont indiquées par la commande `\breathe`.

```
c'4 \breathe d4
```



Propriétés couramment modifiées

On peut choisir le glyphe imprimé par cette commande, en modifiant la propriété `text` de l'objet `BreathingSign`, pour lui affecter n'importe quelle indication textuelle. Par exemple :

```
c'4
\override BreathingSign #'text
= #(make-musicglyph-markup "scripts.rvarcomma")
\breathe
d4
```



Voir aussi

Référence du programme : `BreathingSign`.

6.6.5 Trilles

Les trilles brefs s'indiquent comme n'importe quelle ponctuation : voir [Section 6.6.1 \[Articulations\]](#), page 101.

Les trilles plus longs sont délimités par `\startTrillSpan` et `\stopTrillSpan` :

```
\new Voice {
  << { c1 \startTrillSpan }
    { s2. \grace { d16[\stopTrillSpan e] } } >>
  c4 }
```



Les trilles qui font intervenir une hauteur précise peuvent être indiqués par la commande `pitchedTrill`.

```
\pitchedTrill c4\startTrillSpan fis
f\stopTrillSpan
```



Le premier argument est la note réelle ; le second est une hauteur qui sera imprimée comme une tête de note noire entre parenthèses.

Commandes prédéfinies

`\startTrillSpan`, `\stopTrillSpan`.

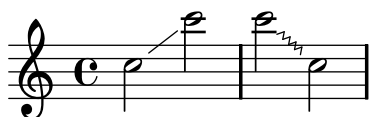
Voir aussi

Référence du programme : `TrillSpanner`.

6.6.6 Glissando

Un glissando relie une hauteur à une autre en passant par chaque hauteur intermédiaire. Il est indiqué graphiquement, par une ligne ou des vaguelettes entre ces deux notes. On l'obtient en accolant la commande `\glissando` à la première note.

```
c2\glissando c'
\override Glissando #'style = #'zigzag
c2\glissando c,
```



Voir aussi

Référence du programme : `Glissando`.

D'autres exemples : `'expressive/glissando.ly'`, `'expressive/line-styles.ly'`.

Bogues

Il n'est pas possible d'imprimer un texte (tel que *gliss.*) le long de la ligne de glissando.

6.6.7 Arpège

On peut indiquer qu'un accord doit être arpégé en lui accolant la commande `\arpeggio` :

```
<c e g c>\arpeggio
```



Pour spécifier qu'un autre accord doit être plaqué et non arpégé, on peut remplacer ce signe par un crochet :

```
\arpeggioBracket
<c' e g c>\arpeggio
```



Le sens de l'arpège est parfois indiqué par une pointe de flèche au bout de la vaguelette :

```
\new Voice {
  \arpeggioUp
  <c e g c>\arpeggio
  \arpeggioDown
  <c e g c>\arpeggio
}
```



Propriétés couramment modifiées

Quand un arpège couvre plusieurs portées, il est possible d'indiquer l'arpège sur chacune des portées, puis de relier la ligne de vaguelettes en définissant à *vrai* la propriété `connectArpeggios`, par exemple dans le contexte `PianoStaff` :

```
\new PianoStaff <<
  \set PianoStaff.connectArpeggios = ##t
  \new Staff { <c' e g c>\arpeggio }
  \new Staff { \clef bass <c,, e g>\arpeggio }
  >>
```



Commandes prédéfinies

`\arpeggio`, `\arpeggioUp`, `\arpeggioDown`, `\arpeggioNeutral`, `\arpeggioBracket`.

Voir aussi

Dans ce même manuel : [Section 6.5.1 \[Liaisons de prolongation\]](#), page 92, pour noter explicitement des arpèges.

Référence du programme : `Arpeggio`.

Bogues

Il est impossible de mêler au même instant, dans un contexte `PianoStaff`, des lignes d'arpèges connectées et d'autres non connectées.

6.6.8 Chutes et sauts

Des indications de désinence peuvent être obtenues au moyen de la commande `\bendAfter` :



6.7 Répétitions

Une des notions essentielles, dans la musique, est le fait que des éléments soient joués plusieurs fois. Il existe de nombreuses façons de noter ces reprises.

6.7.1 Types de répétition

On peut indiquer des répétitions des façons suivantes :

- unfold** La musique qui doit être répétée sera entièrement imprimée (et jouée). Ceci est particulièrement utile dans de la musique répétitive. Ce type de reprise est le seul pris en compte dans le rendu MIDI.
- volta** Le passage répété ne sera pas écrit explicitement, mais il sera encadré sur la partition par des barres de reprises, et peut se terminer par plusieurs fins alternatives, imprimées de gauche à droite sous des crochets. Il s'agit là de la notation courante des reprises avec des fins alternatives. Ces dernières, par défaut, ne sont pas jouées dans le rendu MIDI.
- tremolo** Pour réaliser des trémolos. Ceux-ci, par défaut, ne sont pas joués dans le rendu MIDI.
- percent** Pour répéter des temps ou des mesures, sous forme de signes de pourcentage. Ceux-ci, par défaut, ne sont pas joués dans le rendu MIDI. Les répétitions indiquées par ces symboles doivent être déclarées dans un contexte `Voice`.

6.7.2 Syntaxe des répétitions

Tous les différents types de reprise se spécifient dans LilyPond avec une même construction syntaxique, qui est :

```
\repeat type_de_la_reprise nombre_de_répétitions expression_à_répéter
```

On peut ajouter, pour indiquer une fin alternative :

```
\alternative {
  alternative1
  alternative2
  alternative3
  ...
}
```

}

chaque *alternative* étant une expression musicale. Si l'on donne trop peu d'alternatives en regard du nombre de fois où le passage doit être rejoué, la première alternative sera jouée plusieurs fois.

Les reprises courantes s'indiquent comme ceci :

```
c1
\repeat volta 2 { c4 d e f }
\repeat volta 2 { f e d c }
```



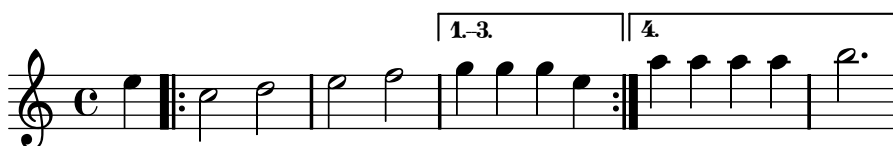
Et avec des fins alternatives :

```
c1
\repeat volta 2 {c4 d e f}
\alternative { {d2 d} {f f,} }
```



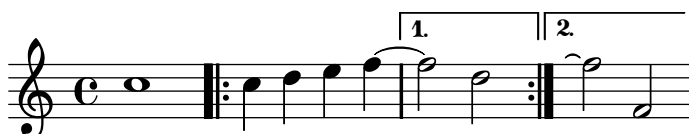
Il est possible de créer des répétitions avec une levée.

```
\new Staff {
  \partial 4 e |
  \repeat volta 4 { c2 d2 | e2 f2 | }
  \alternative { { g4 g g e } { a a a a | b2. } }
}
```



Des liaisons de tenue peuvent être ajoutées à toute alternative :

```
c1
\repeat volta 2 {c4 d e f ~ }
\alternative { {f2 d} {f\repeatTie f,} }
```



On peut raccourcir les crochets indiquant les fins alternatives, en définissant la propriété `voltaSpannerDuration`. Dans l'exemple suivant, le crochet ne se prolonge que sur une mesure à 3/4.

```

\relative c''{
  \time 3/4
  c c c
  \set Staff.voltaSpannerDuration = #(ly:make-moment 3 4)
  \repeat "volta" 5 { d d d }
  \alternative { { e e e f f f }
    { g g g } }
}

```



Si l'on souhaite à la fois terminer une ligne par une double barre et débiter la ligne suivante avec une section reprise, on peut utiliser

```

... \bar "||:" \break
\repeat volta 2 { ...

```

Consultez [Section 6.4.5 \[Barres de mesures\]](#), [page 86](#) pour plus d'informations.

Voir aussi

Les crochets précédant une reprise s'impriment d'ordinaire seulement au-dessus de la portée du haut. On peut ajuster cela en déplaçant le graveur `Volta_engraver` vers les contextes de portée (`Staff`) qui doivent comporter ces crochets ; voir [Section 9.2.4 \[Modifier les greffons de contexte\]](#), [page 134](#) et `'repeats/volta-multi-staff.ly'`.

Bogues

Des reprises imbriquées telles que

```

\repeat ...
\repeat ...
\alternative

```

présentent une ambiguïté, dans la mesure où l'on ne sait à quelle section `\repeat` attribuer la section `\alternative`. Pour résoudre cette ambiguïté, il convient de toujours insérer la commande `\alternative` à l'intérieur de la section `\repeat`. Il est préférable, dans une telle situation, d'utiliser des accolades pour plus de clarté.

Lorsqu'une alternative commence, les informations de métrique sont perdues, il faut donc les rappeler après une reprise, par exemple en définissant `Score.measurePosition` ou en invoquant la commande `\partial`. De même, aucune liaison (de tenue ou autre) n'est répétée.

Les crochets qui indiquent les alternatives ne sont pas alignés verticalement.

6.7.3 Répétitions et MIDI

Au prix de quelques réglages, les reprises de toutes sortes peuvent être rendues dans le fichier MIDI. Il suffit pour cela de recourir à la fonction `\unfoldRepeats`, qui développe toutes les reprises, i.e. qui transforme toutes les reprises en reprises de type `unfold`.

```

\unfoldRepeats {
  \repeat tremolo 8 {c'32 e' }
  \repeat percent 2 { c''8 d'' }
}

```

```

\repeat volta 2 {c'4 d' e' f'}
\alternative {
  { g' a' a' g' }
  { f' e' d' c' }
}
\bar "|."

```



Lorsque l'on veut utiliser `\unfoldRepeats` seulement pour le rendu MIDI, il faut établir deux blocs `\score` : un pour le MIDI, avec des reprises explicites, et l'autre pour la partition, avec des reprises notées sous forme de barres de reprise, de trémolo ou de symboles de pourcentage. Par exemple

```

\score {
  ..musique..
  \layout { .. }
}
\score {
  \unfoldRepeats ..musique..
  \midi { .. }
}

```

6.7.4 Commandes de reprise manuelles

La propriété `repeatCommands` sert à contrôler la mise en forme des reprises. On la définit par une suite de commandes de reprise Scheme.

start-repeat

Pour imprimer une barre de reprise | :

end-repeat

Pour imprimer une barre de reprise :|

(volta *texte*)

Pour imprimer un crochet indiquant une alternative. Le texte *texte* mentionné dans le crochet peut être n'importe quelle chaîne de caractères ou indication textuelle — voir [Section 8.1.5 \[Étiquette textuelle\], page 122](#). Attention cependant à changer la police, car la police employée par défaut pour les chiffres ne contient aucun caractère alphabétique.

(volta #f)

Pour terminer un crochet indiquant une alternative.

```

c4
\set Score.repeatCommands = #'((volta "93") end-repeat)
c4 c4
\set Score.repeatCommands = #'((volta #f))
c4 c4

```



Voir aussi

Référence du programme : `VoltaBracket`, `RepeatedMusic`, `VoltaRepeatedMusic`, et `UnfoldedRepeatedMusic`.

6.7.5 Répétitions en trémolo

On peut placer une notation de trémolo entre deux notes, avec la commande `\repeat` suivie du style trémolo :

```

\new Voice \relative c' {
  \repeat "tremolo" 8 { c16 d16 }
  \repeat "tremolo" 4 { c16 d16 }
  \repeat "tremolo" 2 { c16 d16 }
}

```



On peut aussi indiquer un trémolo sur une seule note, qu'il faudra alors laisser sans accolades.

```
\repeat "tremolo" 4 c'16
```



La subdivision des trémolos aboutit à un résultat semblable : voir [Section 6.7.6 \[Subdivision des trémolos\]](#), page 115.

Voir aussi

Dans ce manuel : [Section 6.7.6 \[Subdivision des trémolos\]](#), page 115, [Section 6.7 \[Répétitions\]](#), page 110.

Référence du programme : `Beam`, `StemTremolo`.

6.7.6 Subdivision des trémolos

Un trémolo peut être indiqué sur une seule note, en la faisant suivre de deux points et d'un nombre :

```
note:[nombre].
```

Le nombre en question correspond à la valeur de la subdivision ; il doit être au moins de 8, auquel cas la hampe sera barrée par une seule ligature. Si ce nombre est omis, la dernière valeur — telle que mémorisée dans `tremoloFlags`) — sera utilisée.

```
c'2:8 c':32 | c': c': |
```



Bogues

Les trémolos entrés de cette manière ne sont pas rendus dans le fichier MIDI.

Voir aussi

Dans ce manuel : [Section 6.7.5 \[Répétitions en trémolo\]](#), page 114.

Référence du programme : `StemTremolo`.

6.7.7 Répétitions de mesure

Le style de « reprise en pourcent » sert à répéter une séquence de notes. Elle sera imprimée une fois, puis remplacée par un symbole spécial. Les séquences d'une ou deux mesures sont remplacées par un symbole qui ressemble au symbole de pourcentage, tandis que les séquences inférieures à une mesure sont remplacées par une barre oblique. Toutes ces répétitions doivent être déclarées dans un contexte `Voice`.

```
\new Voice \relative c' {
  \repeat "percent" 4 { c4 }
  \repeat "percent" 2 { c2 es2 f4 fis4 g4 c4 }
}
```



Les répétitions de plus de 2 mesures sont surmontées d'un compteur, si l'on définit à *vraie* (lettre 't') la propriété `countPercentRepeats`.

```
\new Voice {
  \set countPercentRepeats = ##t
  \repeat "percent" 4 { c1 }
}
```



Des symboles de pourcentage isolés peuvent aussi être obtenus, au moyen d'un silence multi-mesures dont on modifie l'aspect :

```
\override MultiMeasureRest #'stencil
= #ly:multi-measure-rest::percent
R1
```



Voir aussi

Référence du programme : RepeatSlash, PercentRepeat, DoublePercentRepeat, DoublePercentRepeatCounter, PercentRepeatCounter, PercentRepeatedMusic.

7 Notation spécifique

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.1 Musique pour piano

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.1.1 Changements de portée automatiques

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.1.2 Changements de portée manuels

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.1.3 Pédales

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.1.4 Lignes de changement de portée

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.1.5 Accords sur plusieurs portées

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.2 Les noms des accords

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.2.1 Introduction aux noms d'accords

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.2.2 Mode accords

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.2.3 Imprimer des noms d'accord

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3 Musique vocale

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.1 Écrire des chants simples

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.2 Saisie des paroles

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.3 Aligner des paroles sur une mélodie

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.3.1 Durée automatique des syllabes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.3.2 Une autre manière de saisir des paroles

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.3.3 Affecter plus d'une syllabe à une note

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.3.4 Plus d'une note pour une même syllabe

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.3.5 Prolongateurs et traits d'union

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.4 Utiliser des identificateurs pour gérer les paroles

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.5 Flexibilité dans le placement

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.5.1 Paroles sur plusieurs notes d'un mélisme

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.5.2 Paroles alternatives

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.5.3 Changer la voix associée à une ligne de paroles

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.5.4 Paroles indépendantes des notes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.6 Espacement des paroles

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.7 Traitement avancé des couplets

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.7.1 Adding stanza numbers

7.3.7.2 Ajouter des nuances

7.3.7.3 Ajouter le nom des chanteurs

7.3.7.4 Printing stanzas at the end

7.3.7.5 Printing stanzas at the end in multiple columns

7.3.8 Ambitus

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.3.9 Autres éléments relatifs à la voix

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.4 Musique rythmique

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.4.1 Graver des lignes rythmiques

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.4.2 Notation des percussions

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.4.3 Portée de percussions

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.4.4 Notes fantômes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.5 Guitare

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.5.1 Indications de numéro de corde

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.5.2 Les tablatures : notions élémentaires

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.5.3 Tablatures autres que pour la guitare

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.5.4 Tablatures pour banjo

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.5.5 Tablatures

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.5.6 Doigtés pour la main droite

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.5.7 Autres éléments relatifs à la guitare

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.6 Cornemuse

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.6.1 Définitions pour la cornemuse

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.6.2 Exemple pour la cornemuse

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.7 Notations anciennes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.7.1 Têtes de note anciennes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.7.2 Altérations anciennes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.7.3 Silences anciens

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.7.4 Clefs anciennes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.7.5 Crochets anciens

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.7.6 Métriques anciennes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.7.7 Articulations anciennes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.7.8 Guidons

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.7.9 Divisions

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.7.10 Ligatures

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.7.10.1 Ligatures mensurales

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.7.10.2 Grégorien et neumes ligaturés

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.7.11 Contextes du chant grégorien

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.7.12 Le contexte mensural

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.7.13 Altérations et musica ficta

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.7.14 Basse chiffrée

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.8 Autres notations spécifiques à des instruments

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

7.8.1 Harmoniques artificiels (cordes)

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8 Notation avancée

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.1 Texte

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.1.1 Commentaires textuels

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.1.2 Indications textuelles et lignes d'extension

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.1.3 Extensions de texte

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.1.4 Indications textuelles

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.1.5 Étiquette textuelle

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.1.6 Partitions enchâssées

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.1.7 Texte avec sauts de page

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.1.8 Vue d'ensemble des commandes d'indication textuelle

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

`\arrow-head` *axis* (integer) *direction* (direction) *filled* (boolean)

Produce an arrow head in specified direction and axis. Use the filled head if *filled* is specified.

`\beam` *width* (number) *slope* (number) *thickness* (number)

Create a beam with the specified parameters.

`\bigger` *arg* (markup)

Increase the font size relative to current setting.

`\bold` *arg* (markup)

Switch to bold font-series.

`\box` *arg* (markup)

Draw a box round *arg*. Looks at **thickness**, **box-padding** and **font-size** properties to determine line thickness and padding around the markup.

`\bracket` *arg* (markup)

Draw vertical brackets around *arg*.

`\caps` *arg* (markup)

Emit *arg* as small caps.

- `\center-align` *args* (list of markups)
Put *args* in a centered column.
- `\char` *num* (integer)
Produce a single character. For example, `\char #65` produces the letter ‘A’.
- `\circle` *arg* (markup)
Draw a circle around *arg*. Use `thickness`, `circle-padding` and `font-size` properties to determine line thickness and padding around the markup.
- `\column` *args* (list of markups)
Stack the markups in *args* vertically. The property `baseline-skip` determines the space between each markup in *args*.
- `\combine` *m1* (markup) *m2* (markup)
Print two markups on top of each other.
- `\concat` *args* (list of markups)
Concatenate *args* in a horizontal line, without spaces inbetween. Strings and simple markups are concatenated on the input level, allowing ligatures. For example, `\concat { "f" \simple #"i" }` is equivalent to `"fi"`.
- `\dir-column` *args* (list of markups)
Make a column of *args*, going up or down, depending on the setting of the `#'direction` layout property.
- `\doubleflat`
Draw a double flat symbol.
- `\doublessharp`
Draw a double sharp symbol.
- `\draw-circle` *radius* (number) *thickness* (number) *fill* (boolean)
A circle of radius *radius*, thickness *thickness* and optionally filled.
- `\draw-line` *dest* (pair of numbers)
A simple line. Uses the `thickness` property.
- `\dynamic` *arg* (markup)
Use the dynamic font. This font only contains **s**, **f**, **m**, **z**, **p**, and **r**. When producing phrases, like ‘più **f**’, the normal words (like ‘più’) should be done in a different font. The recommended font for this is bold and italic.
- `\epsfile` *axis* (number) *size* (number) *file-name* (string)
Inline an EPS image. The image is scaled along *axis* to *size*.
- `\fill-line` *markups* (list of markups)
Put *markups* in a horizontal line of width *line-width*. The markups are spaced or flushed to fill the entire line. If there are no arguments, return an empty stencil.
- `\filled-box` *xext* (pair of numbers) *yext* (pair of numbers) *blot* (number)
Draw a box with rounded corners of dimensions *xext* and *yext*. For example,
`\filled-box #'(-.3 . 1.8) #'(-.3 . 1.8) #0`
creates a box extending horizontally from -0.3 to 1.8 and vertically from -0.3 up to 1.8, with corners formed from a circle of diameter 0 (i.e. sharp corners).
- `\finger` *arg* (markup)
Set the argument as small numbers.
- `\flat`
Draw a flat symbol.

`\fontCaps` *arg* (markup)

Set `font-shape` to `caps`.

`\fontsize` *increment* (number) *arg* (markup)

Add *increment* to the font-size. Adjust baseline skip accordingly.

`\fraction` *arg1* (markup) *arg2* (markup)

Make a fraction of two markups.

`\fret-diagram` *definition-string* (string)

Make a (guitar) fret diagram. For example, say

```
\markup \fret-diagram #"s:0.75;6-x;5-x;4-o;3-2;2-3;1-2;"
```

for fret spacing 3/4 of staff space, D chord diagram

Syntax rules for *definition-string*:

- Diagram items are separated by semicolons.
- Possible items:
 - `s:number` – Set the fret spacing of the diagram (in staff spaces). Default: 1.
 - `t:number` – Set the line thickness (in staff spaces). Default: 0.05.
 - `h:number` – Set the height of the diagram in frets. Default: 4.
 - `w:number` – Set the width of the diagram in strings. Default: 6.
 - `f:number` – Set fingering label type (0 = none, 1 = in circle on string, 2 = below string). Default: 0.
 - `d:number` – Set radius of dot, in terms of fret spacing. Default: 0.25.
 - `p:number` – Set the position of the dot in the fret space. 0.5 is centered; 1 is on lower fret bar, 0 is on upper fret bar. Default: 0.6.
 - `c:string1-string2-fret` – Include a barre mark from *string1* to *string2* on *fret*.
 - `string-fret` – Place a dot on *string* at *fret*. If *fret* is ‘o’, *string* is identified as open. If *fret* is ‘x’, *string* is identified as muted.
 - `string-fret-fingering` – Place a dot on *string* at *fret*, and label with *fingering* as defined by the `f:` code.
- Note: There is no limit to the number of fret indications per string.

`\fret-diagram-terse` *definition-string* (string)

Make a fret diagram markup using terse string-based syntax.

Here an example

```
\markup \fret-diagram-terse #"x;x;o;2;3;2;"
```

for a D chord diagram.

Syntax rules for *definition-string*:

- Strings are terminated by semicolons; the number of semicolons is the number of strings in the diagram.
- Mute strings are indicated by ‘x’.
- Open strings are indicated by ‘o’.
- A number indicates a fret indication at that fret.
- If there are multiple fret indicators desired on a string, they should be separated by spaces.
- Fingerings are given by following the fret number with a -, followed by the finger indicator, e.g. ‘3-2’ for playing the third fret with the second finger.

- Where a barre indicator is desired, follow the fret (or fingering) symbol with `-(` to start a barre and `-)` to end the barre.

`\fret-diagram-verbose` *marking-list* (list)

Make a fret diagram containing the symbols indicated in *marking-list*.

For example,

```
\markup \fret-diagram-verbose
#'( (mute 6) (mute 5) (open 4)
    (place-fret 3 2) (place-fret 2 3) (place-fret 1 2) )
```

produces a standard D chord diagram without fingering indications.

Possible elements in *marking-list*:

`(mute string-number)`

Place a small ‘x’ at the top of string *string-number*.

`(open string-number)`

Place a small ‘o’ at the top of string *string-number*.

`(barre start-string end-string fret-number)`

Place a barre indicator (much like a tie) from string *start-string* to string *end-string* at fret *fret-number*.

`(place-fret string-number fret-number finger-value)`

Place a fret playing indication on string *string-number* at fret *fret-number* with an optional fingering label *finger-value*. By default, the fret playing indicator is a solid dot. This can be changed by setting the value of the variable *dot-color*. If the *finger* part of the `place-fret` element is present, *finger-value* will be displayed according to the setting of the variable *finger-code*. There is no limit to the number of fret indications per string.

`\fromproperty` *symbol* (symbol)

Read the *symbol* from property settings, and produce a stencil from the markup contained within. If *symbol* is not defined, it returns an empty markup.

`\general-align` *axis* (integer) *dir* (number) *arg* (markup)

Align *arg* in *axis* direction to the *dir* side.

`\halign` *dir* (number) *arg* (markup)

Set horizontal alignment. If *dir* is `-1`, then it is left-aligned, while `+1` is right. Values inbetween interpolate alignment accordingly.

`\hbracket` *arg* (markup)

Draw horizontal brackets around *arg*.

`\hcenter-in` *length* (number) *arg* (markup)

Center *arg* horizontally within a box of extending *length*/2 to the left and right.

`\hcenter` *arg* (markup)

Align *arg* to its X center.

`\hspace` *amount* (number)

This produces an invisible object taking horizontal space. For example,

```
\markup { A \hspace #2.0 B }
```

puts extra space between A and B, on top of the space that is normally inserted before elements on a line.

`\huge` *arg* (markup)

Set font size to `+2`.

- `\italic arg` (markup)
Use italic **font-shape** for *arg*.
- `\justify-field symbol` (symbol)
Justify the data which has been assigned to *symbol*.
- `\justify args` (list of markups)
Like wordwrap, but with lines stretched to justify the margins. Use `\override #'(line-width . X)` to set the line width; *X* is the number of staff spaces.
- `\justify-string arg` (string)
Justify a string. Paragraphs may be separated with double newlines
- `\large arg` (markup)
Set font size to +1.
- `\larger arg` (markup)
Copy of the bigger-markup command.
- `\left-align arg` (markup)
Align *arg* on its left edge.
- `\line args` (list of markups)
Put *args* in a horizontal line. The property **word-space** determines the space between each markup in *args*.
- `\lookup glyph-name` (string)
Lookup a glyph by name.
- `\lower amount` (number) *arg* (markup)
Lower *arg* by the distance *amount*. A negative *amount* indicates raising; see also `\raise`.
- `\magnify sz` (number) *arg* (markup)
Set the font magnification for its argument. In the following example, the middle A is 10% larger:

`A \magnify #1.1 { A } A`
 Note: Magnification only works if a font name is explicitly selected. Use `\fontsize` otherwise.
- `\markalphabet num` (integer)
Make a markup letter for *num*. The letters start with A to Z and continue with double letters.
- `\markletter num` (integer)
Make a markup letter for *num*. The letters start with A to Z (skipping letter I), and continue with double letters.
- `\medium arg` (markup)
Switch to medium font series (in contrast to bold).
- `\musicglyph glyph-name` (string)
glyph-name is converted to a musical symbol; for example, `\musicglyph #"accidentals.natural"` selects the natural sign from the music font. See user manual, **The Feta font** for a complete listing of the possible glyphs.
- `\natural`
Draw a natural symbol.
- `\normal-size-sub arg` (markup)
Set *arg* in subscript, in a normal font size.

`\normal-size-super` *arg* (markup)

Set *arg* in superscript with a normal font size.

`\normal-text` *arg* (markup)

Set all font related properties (except the size) to get the default normal text font, no matter what font was used earlier.

`\normalsize` *arg* (markup)

Set font size to default.

`\note-by-number` *log* (number) *dot-count* (number) *dir* (number)

Construct a note symbol, with stem. By using fractional values for *dir*, you can obtain longer or shorter stems.

`\note` *duration* (string) *dir* (number)

This produces a note with a stem pointing in *dir* direction, with the *duration* for the note head type and augmentation dots. For example, `\note #"4." #-0.75` creates a dotted quarter note, with a shortened down stem.

`\null`

An empty markup with extents of a single point.

`\number` *arg* (markup)

Set font family to **number**, which yields the font used for time signatures and fingerings. This font only contains numbers and some punctuation. It doesn't have any letters.

`\on-the-fly` *procedure* (symbol) *arg* (markup)

Apply the *procedure* markup command to *arg*. *procedure* should take a single argument.

`\override` *new-prop* (pair) *arg* (markup)

Add the first argument in to the property list. Properties may be any sort of property supported by **font-interface** and **text-interface**, for example

```
\override #'(font-family . married) "bla"
```

`\pad-around` *amount* (number) *arg* (markup)

Add padding *amount* all around *arg*.

`\pad-markup` *padding* (number) *arg* (markup)

Add space around a markup object.

`\pad-to-box` *x-ext* (pair of numbers) *y-ext* (pair of numbers) *arg* (markup)

Make *arg* take at least *x-ext*, *y-ext* space.

`\pad-x` *amount* (number) *arg* (markup)

Add padding *amount* around *arg* in the X direction.

`\page-ref` *label* (symbol) *gauge* (markup) *default* (markup)

Reference to a page number. *label* is the label set on the referenced page (using the `\label` command), *gauge* a markup used to estimate the maximum width of the page number, and *default* the value to display when *label* is not found.

`\postscript` *str* (string)

This inserts *str* directly into the output as a PostScript command string. Due to technicalities of the output backends, different scales should be used for the T_EX and PostScript backend, selected with `-f`.

For the T_EX backend, the following string prints a rotated text

```
0 0 moveto /ecrm10 findfont
1.75 scalefont setfont 90 rotate (hello) show
```

The magical constant 1.75 scales from LilyPond units (staff spaces) to T_EX dimensions.

For the postscript backend, use the following

```
gsave /ecrm10 findfont
10.0 output-scale div
scalefont setfont 90 rotate (hello) show grestore
```

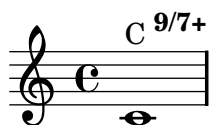
`\put-adjacent arg1 (markup) axis (integer) dir (direction) arg2 (markup)`

Put *arg2* next to *arg1*, without moving *arg1*.

`\raise amount (number) arg (markup)`

Raise *arg* by the distance *amount*. A negative *amount* indicates lowering, see also `\lower`.

```
c1^\markup { C \small \raise #1.0 \bold { "9/7+" } }
```



The argument to `\raise` is the vertical displacement amount, measured in (global) staff spaces. `\raise` and `\super` raise objects in relation to their surrounding markups.

If the text object itself is positioned above or below the staff, then `\raise` cannot be used to move it, since the mechanism that positions it next to the staff cancels any shift made with `\raise`. For vertical positioning, use the `padding` and/or `extra-offset` properties.

`\right-align arg (markup)`

Align *arg* on its right edge.

`\roman arg (markup)`

Set font family to `roman`.

`\rotate ang (number) arg (markup)`

Rotate object with *ang* degrees around its center.

`\sans arg (markup)`

Switch to the sans serif family.

`\score score (unknown)`

Inline an image of music.

`\semiflat`

Draw a semiflat.

`\semisharp`

Draw a semi sharp symbol.

`\sesquiflat`

Draw a 3/2 flat symbol.

`\sesquisharp`

Draw a 3/2 sharp symbol.

`\sharp`

Draw a sharp symbol.

`\simple str` (string)

A simple text string; `\markup { foo }` is equivalent with `\markup { \simple #"foo" }`.

`\slashed-digit num` (integer)

A feta number, with slash. This is for use in the context of figured bass notation.

`\small arg` (markup)

Set font size to -1.

`\smallCaps text` (markup)

Turn `text`, which should be a string, to small caps.

`\markup \smallCaps "Text between double quotes"`

`\smaller arg` (markup)

Decrease the font size relative to current setting.

`\stencil stil` (unknown)

Use a stencil as markup.

`\strut`

Create a box of the same height as the space in the current font.

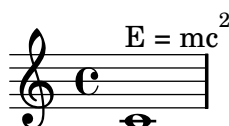
`\sub arg` (markup)

Set `arg` in subscript.

`\super arg` (markup)

Raising and lowering texts can be done with `\super` and `\sub`:

`c1~\markup { E "=" \concat { "mc" \super "2" } }`



`\teeny arg` (markup)

Set font size to -3.

`\text arg` (markup)

Use a text font instead of music symbol or music alphabet font.

`\tied-lyric str` (string)

Like simple-markup, but use tie characters for ‘~’ tilde symbols.

`\tiny arg` (markup)

Set font size to -2.

`\translate offset` (pair of numbers) `arg` (markup)

This translates an object. Its first argument is a cons of numbers.

`A \translate #(cons 2 -3) { B C } D`

This moves ‘B C’ 2 spaces to the right, and 3 down, relative to its surroundings. This command cannot be used to move isolated scripts vertically, for the same reason that `\raise` cannot be used for that.

`\translate-scaled offset` (pair of numbers) `arg` (markup)

Translate `arg` by `offset`, scaling the offset by the `font-size`.

- `\transparent` *arg* (markup)
Make the argument transparent.
- `\triangle` *filled* (boolean)
A triangle, either filled or empty.
- `\typewriter` *arg* (markup)
Use `font-family typewriter` for *arg*.
- `\upright` *arg* (markup)
Set font shape to `upright`. This is the opposite of `italic`.
- `\vcenter` *arg* (markup)
Align *arg* to its Y center.
- `\verbatim-file` *name* (string)
Read the contents of a file, and include it verbatim.
- `\whiteout` *arg* (markup)
Provide a white underground for *arg*.
- `\with-color` *color* (list) *arg* (markup)
Draw *arg* in color specified by *color*.
- `\with-dimensions` *x* (pair of numbers) *y* (pair of numbers) *arg* (markup)
Set the dimensions of *arg* to *x* and *y*.
- `\with-url` *url* (string) *arg* (markup)
Add a link to URL *url* around *arg*. This only works in the PDF backend.
- `\wordwrap-field` *symbol* (symbol)
Wordwrap the data which has been assigned to *symbol*.
- `\wordwrap` *args* (list of markups)
Simple wordwrap. Use `\override #'(line-width . X)` to set the line width, where *X* is the number of staff spaces.
- `\wordwrap-string` *arg* (string)
Wordwrap a string. Paragraphs may be separated with double newlines.

8.1.9 Vue d'ensemble des commandes de listes d'indications textuelles

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

- `\column-lines` *args* (list of markups)
Like `\column`, but return a list of lines instead of a single markup. `baseline-skip` determines the space between each markup in *args*.
- `\justified-lines` *args* (list of markups)
Like `\justify`, but return a list of lines instead of a single markup. Use `\override #'(line-width . X)` to set the line width; *X* is the number of staff spaces.
- `\override-lines` *new-prop* (pair) *args* (list of markups)
Like `\override`, for markup lists.
- `\wordwrap-lines` *args* (list of markups)
Like `\wordwrap`, but return a list of lines instead of a single markup. Use `\override #'(line-width . X)` to set the line width, where *X* is the number of staff spaces.

8.1.10 Sélection de la fonte

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.1.11 Personnalisation des indications de nuance

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.2 Préparer un matériel d'orchestre

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.2.1 Silences multi-mesures

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.2.2 Indications métronomiques

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.2.3 Indications de repère

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.2.4 Numéros de mesure

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.2.5 Noms d'instrument

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.2.6 Instruments transpositeurs

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.2.7 Marques d'octavation

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.2.8 Différentes éditions à partir d'une même source

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.3 Musique orchestrale

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.3.1 Regroupement automatique de parties

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.3.2 Masquer des portées

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.3.3 Citer d'autres voix

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.3.4 Mettre en forme une citation

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.3.5 Alignement et cadences

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.4 Notation contemporaine

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.4.1 Notation polymétrique

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.4.2 Gestion du temps

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.4.3 Introduction à la notation proportionnelle

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.4.4 Clusters

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.4.5 Têtes de note spécifiques

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.4.6 Barres de ligature en soufflet

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.4.7 Improvisation

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.4.8 Spécifier la taille de la fonte musicale

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.5 Utilisation en milieu éducatif

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.5.1 Info-bulle

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.5.2 Papier à musique

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.5.3 Dictée à trous

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.5.4 Têtes de note à forme variable

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.5.5 Têtes de note avec nom de note

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.5.6 Crochets d'analyse

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.5.7 Colorer des objets

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.5.8 Parenthèses

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

8.5.9 Quadrillage temporel

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9 Changer les réglages prédéfinis

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.1 Comportements automatiques

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.1.1 Altérations accidentelles automatiques

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.1.2 Définir les règles de ligatures automatiques

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.2 Contextes d'interprétation

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.2.1 Tout savoir sur les contextes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.2.2 Créer un contexte

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.2.3 Changer à la volée les propriétés de contexte

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.2.4 Modifier les greffons de contexte

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.2.5 Affiner le rendu au sein des contextes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.2.6 Modifier les réglages par défaut d'un contexte

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.2.7 Définir de nouveaux contextes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.2.8 Aligner des contextes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.2.9 Groupement vertical d'objets graphiques

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.3 La commande `\override`

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.3.1 Élaborer une retouche

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.3.2 Manipuler le guide de référence du programme

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.3.3 Interfaces de rendu

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.3.4 Déterminer la propriété de l'objet graphique (Grob)

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.3.5 Objets liés à la source

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.3.6 Utiliser du code Scheme au lieu de `\tweak`

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.3.7 `\set` ou `\override`

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

9.3.8 Retouches complexes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10 Objets non musicaux

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.1 Fichiers d'entrée

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.1.1 Introduction à la structure de fichier

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.1.2 Structure de fichier

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.1.3 Une expression musicale unique

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.1.4 Plusieurs partitions dans un même ouvrage

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.1.5 Extraire des fragments

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.1.6 Insérer des fichiers LilyPond

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.1.7 Codage du texte

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.2 Titres et entêtes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.2.1 Créer des titres

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.2.2 Titres personnalisés

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.2.3 Référence de numéro de page

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.2.4 Table des matières

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.3 Sortie MIDI

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.3.1 Créer des fichiers MIDI

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.3.2 Le bloc MIDI

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.3.3 Noms d'instruments MIDI

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.4 Afficher de la notation au format LilyPond

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

10.5 Ignorer des passages de la partition

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11 Gestion de l'espace

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.1 Du papier et des pages

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.1.1 Format de papier

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.1.2 Formater une page

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.2 Mise en forme de la musique

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.2.1 Définir la taille de la portée

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.2.2 Mise en forme de la partition

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.3 Montrer les espacements

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.4 Sauts

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.4.1 Saut de ligne

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.4.2 Saut de page

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.4.3 Optimisation des sauts de page

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.4.4 Optimisation des tournes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.4.5 Minimisation des sauts de page

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.4.6 Sauts explicites

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.4.7 Recourir à une voix supplémentaire pour gérer les sauts

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.5 Espacement vertical

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.5.1 Espacement vertical au sein d'un système

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.5.2 Espacement vertical entre les systèmes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.5.3 Positionnement explicite des portées et systèmes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.5.4 Optimiser le remplissage avec un deuxième passage

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.5.5 Résolution des collisions verticales

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.6 Espacement horizontal

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.6.1 Généralités sur l'espacement horizontal

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.6.2 Changement d'espacement au cours de la partition

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.6.3 Modifier l'espacement horizontal

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.6.4 Longueur de ligne

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

11.6.5 Notation proportionnelle

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

12 Interfaces pour les programmeurs

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

12.1 Fonctions musicales

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

12.1.1 Aperçu des fonctions musicales

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

12.1.2 Fonctions de substitution simple

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

12.1.3 Fonctions de substitution par paire

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

12.1.4 De l'usage des mathématiques dans les fonctions

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

12.1.5 Fonctions fantômes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

12.1.6 Fonctions sans argument

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

12.1.7 Liste des fonctions musicales prédéfinies

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

`transposedCueDuring` - *what* (string) *dir* (direction) *pitch-note* (music) *main-music* (music)
(undocumented; fixme)

`displayLilyMusic` - *music* (music)
(undocumented; fixme)

`tocItem` - *text* (markup)
Add a line to the table of content, using the `tocItemMarkup` paper variable markup

`appoggiatura` - *music* (music)
(undocumented; fixme)

`cueDuring` - *what* (string) *dir* (direction) *main-music* (music)
(undocumented; fixme)

`removeWithTag` - *tag* (symbol) *music* (music)
(undocumented; fixme)

`breathe` -
(undocumented; fixme)

`clef` - *type* (string)
(undocumented; fixme)

`overrideProperty` - *name* (string) *property* (symbol) *value* (any type)
(undocumented; fixme)

withMusicProperty - *sym* (symbol) *val* (any type) *music* (music)
 (undocumented; fixme)

assertBeamSlope - *comp* (procedure)
 (undocumented; fixme)

acciaccatura - *music* (music)
 (undocumented; fixme)

pitchedTrill - *main-note* (music) *secondary-note* (music)
 (undocumented; fixme)

applyContext - *proc* (procedure)
 (undocumented; fixme)

allowPageTurn -
 (undocumented; fixme)

assertBeamQuant - *l* (pair) *r* (pair)
 (undocumented; fixme)

includePageLayoutFile -
 (undocumented; fixme)

transposition - *pitch-note* (music)
 (undocumented; fixme)

applyOutput - *ctx* (symbol) *proc* (procedure)
 (undocumented; fixme)

afterGrace - *main* (music) *grace* (music)
 (undocumented; fixme)

label - *label* (symbol)
 (undocumented; fixme)

oldaddyrics - *music* (music) *lyrics* (music)
 (undocumented; fixme)

pageTurn -
 (undocumented; fixme)

compressMusic - *fraction* (pair of numbers) *music* (music)
 (undocumented; fixme)

featherDurations - *factor* (moment) *argument* (music)
 (undocumented; fixme)

displayMusic - *music* (music)
 (undocumented; fixme)

parallelMusic - *voice-ids* (list) *music* (music)
 (undocumented; fixme)

resetRelativeOctave - *reference-note* (music)
 (undocumented; fixme)

addQuote - *name* (string) *music* (music)
 (undocumented; fixme)

octave - *pitch-note* (music)
 (undocumented; fixme)

`parenthesize` - *arg* (music)
 (undocumented; fixme)

`balloonGrobText` - *grob-name* (symbol) *offset* (pair of numbers) *text* (markup)
 (undocumented; fixme)

`instrumentSwitch` - *name* (string)
 (undocumented; fixme)

`makeClusters` - *arg* (music)
 (undocumented; fixme)

`spacingTweaks` - *parameters* (list)
 (undocumented; fixme)

`tag` - *tag* (symbol) *arg* (music)
 (undocumented; fixme)

`noPageTurn` -
 (undocumented; fixme)

`bendAfter` - *delta* (integer)
 (undocumented; fixme)

`partcombine` - *part1* (music) *part2* (music)
 (undocumented; fixme)

`grace` - *music* (music)
 (undocumented; fixme)

`noPageBreak` -
 (undocumented; fixme)

`pageBreak` -
 (undocumented; fixme)

`bar` - *type* (string)
 (undocumented; fixme)

`shiftDurations` - *dur* (integer) *dots* (integer) *arg* (music)
 (undocumented; fixme)

`unfoldRepeats` - *music* (music)
 (undocumented; fixme)

`balloonText` - *offset* (pair of numbers) *text* (markup)
 (undocumented; fixme)

`quoteDuring` - *what* (string) *main-music* (music)
 (undocumented; fixme)

`barNumberCheck` - *n* (integer)
 (undocumented; fixme)

`addInstrumentDefinition` - *name* (string) *lst* (list)
 (undocumented; fixme)

`scoreTweak` - *name* (string)
 (undocumented; fixme)

`autochange` - *music* (music)
 (undocumented; fixme)

rightHandFinger - *finger* (number or string)
(undocumented; fixme)

endSpanners - *music* (music)
(undocumented; fixme)

musicMap - *proc* (procedure) *mus* (music)
(undocumented; fixme)

applyMusic - *func* (procedure) *music* (music)
(undocumented; fixme)

killCues - *music* (music)
(undocumented; fixme)

keepWithTag - *tag* (symbol) *music* (music)
(undocumented; fixme)

tweak - *sym* (symbol) *val* (any type) *arg* (music)
(undocumented; fixme)

12.4.3 Définition d'une nouvelle commande de marqueur

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

12.4.4 Définition d'une nouvelle commande de liste de marqueurs

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

12.5 Contextes pour programmeurs

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

12.5.1 Évaluation d'un contexte

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

12.5.2 Appliquer une fonction à tous les objets de mise en forme

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

12.6 Utiliser des procédures Scheme en tant que propriétés

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

Annexe A Bibliographie

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

Annexe B Tutoriel Scheme

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

Annexe C Tables du manuel de notation

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

C.1 Table des noms d'accord

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

C.2 Instruments MIDI

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

C.3 Liste des couleurs

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

Couleurs de base

Noms de couleurs X

Noms de couleur sans suffixe numérique

Noms de couleur avec suffixe numérique

Échelle de gris

C.4 La fonte Feta

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

C.5 Styles de tête de note

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

Annexe D Modèles

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

D.1 Portée unique

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

D.1.1 Notes seules

D.1.2 Notes et paroles

D.1.3 Notes et accords

D.1.4 Notes, paroles et accords

D.2 Modèles pour claviers

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

D.2.1 Piano seul

D.2.2 Chant et accompagnement

D.2.3 Piano et paroles entre les portées

D.2.4 Piano et nuances entre les portées

D.3 Quatuor à cordes

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

D.3.1 Quatuor à cordes

D.3.2 Parties pour quatuor à cordes

D.4 Ensemble vocal

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

D.4.1 Partition pour chœur à quatre voix mixtes

D.4.2 Partition pour chœur SATB avec réduction pour piano

D.4.3 Partition pour chœur SATB avec alignement des contextes

D.5 Exemples de notation ancienne

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

D.5.1 Transcription de musique mensurale

D.5.2 Transcription du grégorien

D.6 Symboles de jazz

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

D.7 Squelettes pour lilypond-book

Cette section n'est pas encore traduite ; reportez-vous à l'édition de ce manuel en anglais.

D.7.1 LaTeX

D.7.2 Texinfo

Annexe E Aide-mémoire

Syntaxe

1 2 8 16

Description

valeurs rythmiques

Exemple



c4. c4..

notes pointées



c d e f g a b

gamme



fis bes

altérations



\clef treble \clef bass

clés



\time 3/4 \time 4/4

chiffre de mesure



r4 r8

silences



d ~ d

liaison de tenue



`\key es \major`

armure

`note'`

monter d'une octave

`note,`

baisser d'une octave

`c(d e)`

liaisons

`c\ (c(d) e\)`

liaisons de phrasé

`a8[b]`

ligatures

`<< \new Staff ... >>`

ajouter des portées

`c-> c-.`

indications d'articulation



`c\mf c\s fz`

nuances

`a\< a a\!`

crescendo

`a\> a a\!`

decrescendo

`< >`

accords

`\partial 8`

levées

`\times 2/3 {f g a}`

triolet

`\grace`

appoggiatures

`\lyricmode {twinkle }`

ajouter des paroles

twinkle

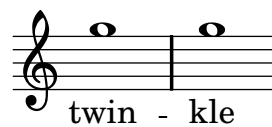
`\new Lyrics`

imprimer les paroles

twinkle

`twin -- kle`

diviser un mot en
plusieurs syllabes



`\chordmode { c:dim f:maj7 }`

accords chiffrés



`\context ChordNames`

imprimer les chiffrages
d'accords

$C^{\circ} F^{\Delta}$

`<<\{e f\} \\\{c d\}>>`

polyphonie



`s4 s8 s16`

silences invisibles

Annexe F Licence GNU de documentation libre

Version 1.1, March 2000

Copyright © 2000 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of ‘copyleft’, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The ‘Document’, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as ‘you’.

A ‘Modified Version’ of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ‘Secondary Section’ is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ‘Invariant Sections’ are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The ‘Cover Texts’ are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A ‘Transparent’ copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file

format whose markup has been designed to thwart or discourage subsequent modification by readers is not ‘Transparent’. A copy that is not ‘Transparent’ is called ‘Opaque’.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The ‘Title Page’ means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, ‘Title Page’ means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section entitled 'History', and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled 'History' in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the 'History' section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. In any section entitled 'Acknowledgments' or 'Dedications', preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgments and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section entitled 'Endorsements'. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section as 'Endorsements' or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to

the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled 'Endorsements', provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled 'History' in the various original documents, forming one section entitled 'History'; likewise combine any sections entitled 'Acknowledgments', and any sections entitled 'Dedications'. You must delete all sections entitled 'Endorsements.'

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an 'aggregate', and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License ‘or any later version’ applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

F.0.1 SUPPLÉMENT : comment utiliser cette licence pour vos documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.1
or any later version published by the Free Software Foundation;
with the Invariant Sections being list their titles, with the
Front-Cover Texts being list, and with the Back-Cover Texts being list.
A copy of the license is included in the section entitled `GNU
Free Documentation License'
```

If you have no Invariant Sections, write ‘with no Invariant Sections’ instead of saying which ones are invariant. If you have no Front-Cover Texts, write ‘no Front-Cover Texts’ instead of ‘Front-Cover Texts being *list*’; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Annexe G Index des commandes LilyPond

!		\ionian.....	82
!	64	\key.....	82
,		\locrian.....	82
'	62	\longa.....	69
,		\lydian.....	82
,	62	\major.....	82
.		\maxima.....	69
.	70	\mf.....	104
?		\minor.....	82
?	64	\mixolydian.....	82
[\mp.....	104
[.....	96	\oneVoice.....	80
]		\p.....	104
]	96	\partial.....	85
\		\phrasingSlurDown.....	95
\!	105	\phrasingSlurNeutral.....	95
\<	105	\phrasingSlurUp.....	95
\>	105	\phrygian.....	82
\\	75	\pp.....	104
\aeolian.....	82	\ppp.....	104
\afterGrace.....	99	\pppp.....	104
\alternative.....	110	\relative.....	65
\arpeggio.....	108	\repeat.....	110
\arpeggioBracket.....	109	\repeatTie.....	93, 111
\arpeggioDown.....	109	\rest.....	68
\arpeggioNeutral.....	109	\rfz.....	104
\arpeggioUp.....	109	\sf.....	104
\bar.....	86	\sff.....	104
\breve.....	69	\sfz.....	104
\cadenzaOff.....	87	\shiftOff.....	80
\cadenzaOn.....	87	\shiftOn.....	80
\clef.....	81	\shiftOnn.....	80
\dorian.....	82	\shiftOnnn.....	80
\dotsDown.....	70	\skip.....	69
\dotsNeutral.....	70	\slurDashed.....	95
\dotsUp.....	70	\slurDotted.....	95
\dynamicDown.....	107	\slurDown.....	95
\dynamicNeutral.....	107	\slurNeutral.....	95
\dynamicUp.....	107	\slurSolid.....	95
\f.....	104	\slurUp.....	95
\ff.....	104	\sp.....	104
\fff.....	104	\spp.....	104
\ffff.....	104	\startTrillSpan.....	108
\fp.....	104	\stemDown.....	75
\glissando.....	108	\stemNeutral.....	75
\grace.....	98	\stemUp.....	75
		\stopTrillSpan.....	108
		\tieDashed.....	94
		\tieDotted.....	94
		\tieDown.....	94
		\tieNeutral.....	94
		\tieSolid.....	94
		\tieUp.....	94
		\time.....	83
		\times.....	71
		\transpose.....	67
		\tupletDown.....	71
		\tupletNeutral.....	71
		\tupletUp.....	71
		\unfoldRepeats.....	112
		\voiceFour.....	80

<code>\voiceOne</code>	80
<code>\voiceThree</code>	80
<code>\voiceTwo</code>	80

<code> </code>	73
----------------------	----

<code>~</code>	92
----------------------	----

B

<code>barCheckSynchronize</code>	73
<code>breakable</code>	97

D

<code>defaultBarType</code>	87
-----------------------------------	----

P

<code>pipeSymbol</code>	73
-------------------------------	----

R

<code>r</code>	68
<code>repeatCommands</code>	87, 113

S

<code>s</code>	69
<code>stemLeftBeamCount</code>	97
<code>stemRightBeamCount</code>	97
<code>subdivideBeams</code>	97

T

<code>tremoloFlags</code>	115
<code>tupletNumberFormatFunction</code>	71

W

<code>whichBar</code>	87
-----------------------------	----

Annexe H Index de LilyPond

!		\ionian.....	82
!	64	\key.....	82
,		\locrian.....	82
'	62	\longa.....	69
,		\lydian.....	82
,	62	\major.....	82
.		\maxima.....	69
.	70	\mf.....	104
?		\minor.....	82
?	64	\mixolydian.....	82
[\mp.....	104
[.....	96	\oneVoice.....	80
]		\p.....	104
]	96	\partial.....	85
\		\phrasingSlurDown.....	95
\!	105	\phrasingSlurNeutral.....	95
\<.....	105	\phrasingSlurUp.....	95
\>.....	105	\phrygian.....	82
\\.....	75	\pp.....	104
\aeolian.....	82	\ppp.....	104
\afterGrace.....	99	\pppp.....	104
\alternative.....	110	\relative.....	65
\arpeggio.....	108	\repeat.....	110
\arpeggioBracket.....	109	\repeatTie.....	93, 111
\arpeggioDown.....	109	\rest.....	68
\arpeggioNeutral.....	109	\rfz.....	104
\arpeggioUp.....	109	\sf.....	104
\bar.....	86	\sff.....	104
\breve.....	69	\sfz.....	104
\cadenzaOff.....	87	\shiftOff.....	80
\cadenzaOn.....	87	\shiftOn.....	80
\clef.....	81	\shiftOnn.....	80
\dorian.....	82	\shiftOnnn.....	80
\dotsDown.....	70	\skip.....	69
\dotsNeutral.....	70	\slurDashed.....	95
\dotsUp.....	70	\slurDotted.....	95
\dynamicDown.....	107	\slurDown.....	95
\dynamicNeutral.....	107	\slurNeutral.....	95
\dynamicUp.....	107	\slurSolid.....	95
\f.....	104	\slurUp.....	95
\ff.....	104	\sp.....	104
\fff.....	104	\spp.....	104
\ffff.....	104	\startTrillSpan.....	108
\fp.....	104	\stemDown.....	75
\glissando.....	108	\stemNeutral.....	75
\grace.....	98	\stemUp.....	75
		\stopTrillSpan.....	108
		\super.....	129
		\tieDashed.....	94
		\tieDotted.....	94
		\tieDown.....	94
		\tieNeutral.....	94
		\tieSolid.....	94
		\tieUp.....	94
		\time.....	83
		\times.....	71
		\transpose.....	67
		\tupletDown.....	71
		\tupletNeutral.....	71
		\tupletUp.....	71
		\unfoldRepeats.....	112

<code>\voiceFour</code>	80
<code>\voiceOne</code>	80
<code>\voiceThree</code>	80
<code>\voiceTwo</code>	80

|

.....	73
-------	----

~

~	92
---------	----

A

accacciatura	98
accent	101
accents	21
acciaccatura	141
accolade verticale	88
accord arpégé	108
accords	28, 75
accords, noms	30
<code>addInstrumentDefinition</code>	142
<code>addQuote</code>	141
<code>afterGrace</code>	141
al niente	105
<code>allowPageTurn</code>	141
altération de précaution	64
altération entre parenthèses	64
altération, de précaution	64
altération, entre parenthèses	64
anacrouse	23, 85
<code>applyContext</code>	141
<code>applyMusic</code>	143
<code>applyOutput</code>	141
appoggiatura	140
appoggiature	23, 98
armure	82
armure, définition de	19
arpège	108
<code>arrow-head</code>	122
<code>arrow-head-markup</code>	122
articulation	21
articulations	101
<code>assertBeamQuant</code>	141
<code>assertBeamSlope</code>	141
auto-knee-gap	97
<code>autochange</code>	142

B

<code>balloonGrobText</code>	142
<code>balloonText</code>	142
<code>bar</code>	142
<code>barCheckSynchronize</code>	73
<code>barNumberCheck</code>	142
baroque, ornementation	101
barres de mesure	86
barres de reprise	86
<code>beam</code>	122
<code>beam-markup</code>	122
<code>bendAfter</code>	142
<code>bigger</code>	122

<code>bigger-markup</code>	122
<code>bold</code>	122
<code>bold-markup</code>	122
bouché	101
<code>box</code>	122
<code>box-markup</code>	122
<code>bracket</code>	122
<code>bracket-markup</code>	122
<code>breakable</code>	97
<code>breathe</code>	140

C

cadence	87
calques	76
<code>caps</code>	122
<code>caps-markup</code>	122
casse, prise en compte de	12, 17
<code>center-align</code>	123
<code>center-align-markup</code>	123
changement de portée manuel	27
chanson, partition complète	30, 31
chansons	29
<code>char</code>	123
<code>char-markup</code>	123
chiffage de mesure	83
chiffre indicateur de mesure	83
choral tenor clef	81
<code>circle</code>	123
<code>circle-markup</code>	123
clé d'ut	81
clé de fa	81
clé de sol	81
<code>clef</code>	140
coda	101
<code>column</code>	123
<code>column-lines</code>	130
<code>column-lines-markup-list</code>	130
<code>column-markup</code>	123
<code>combine</code>	123
<code>combine-markup</code>	123
commentaire de fin de ligne	17
commentaire-bloc	17
commentaires	17
<code>compressMusic</code>	141
<code>concat</code>	123
<code>concat-markup</code>	123
couches	76
crescendo	22, 106
crochet vertical	88
<code>cueDuring</code>	140

D

decrescendo	22, 106
<code>defaultBarType</code>	87
demi-bémols, demi-dièses	64
diminuendo	106
<code>dir-column</code>	123
<code>dir-column-markup</code>	123
<code>displayLilyMusic</code>	140
<code>displayMusic</code>	141
distances	58
documentation du fonctionnement interne	11

doigtés	21
doubleflat	123
doubleflat-markup	123
doublesharp	123
doublesharp-markup	123
draw-circle	123
draw-circle-markup	123
draw-line	123
draw-line-markup	123
durées	69
dynamic	123
dynamic-markup	123

E

écrire la musique en parallèle	91
endSpanners	143
épaisseur des caractères	2
épaisseur des lignes de portées	90
epsfile	123
epsfile-markup	123
équilibre	2
espacement optique	2
espacement régulier	3
espressivo	101
étendre lilypond	11
exemples de code	10
expression	25
expression musicale	25
extra-offset	54, 57

F

FDL, GNU Free Documentation License	154
featherDurations	141
fichier PDF	13
fill-line	123
fill-line-markup	123
filled-box	123
filled-box-markup	123
finger	123
finger change	103
finger-markup	123
fingering	103
flageolet	101
flat	123
flat-markup	123
fontCaps	124
fontCaps-markup	124
fonte	2
fontsize	124
fontsize-markup	124
fraction	124
fraction-markup	124
fret-diagram	124
fret-diagram-markup	124
fret-diagram-terse	124
fret-diagram-terse-markup	124
fret-diagram-verbose	125
fret-diagram-verbose-markup	125
fromproperty	125
fromproperty-markup	125

G

general-align	125
general-align-markup	125
glissando	108
grace	142
gravure	5
groupements de note manuels	96
gruppetto	101

H

halign	125
halign-markup	125
hauteurs	62
hbracket	125
hbracket-markup	125
hcenter	125
hcenter-in	125
hcenter-in-markup	125
hcenter-markup	125
hspace	125
hspace-markup	125
huge	125
huge-markup	125

I

identificateurs	39, 45
includePageLayoutFile	141
index	11
indication d'octave relative	65
instrumentSwitch	142
invisibles, objets	57
italic	126
italic-markup	126

J

jargon	10
justified-lines	130
justified-lines-markup-list	130
justify	126
justify-field	126
justify-field-markup	126
justify-markup	126
justify-string	126
justify-string-markup	126

K

keepWithTag	143
killCues	143

L

label	141
laissez vibrer	96
langage	10
langue	10
langues étrangères	10
large	126
large-markup	126
larger	126

<code>larger-markup</code>	126
<code>left-align</code>	126
<code>left-align-markup</code>	126
<code>legato</code>	94
<code>levée</code>	23, 85
<code>liaison d'articulation</code>	94
<code>liaison de prolongation</code>	92
<code>liaison de prolongation, répétition</code>	93
<code>liaison, laissez vibrer</code>	96
<code>liaisons d'articulation</code>	20
<code>liaisons de phrasé</code>	21, 95
<code>liaisons de phrasé et de prolongation, différences</code> ..	21
<code>liaisons de prolongation</code>	20
<code>ligatures coudées</code>	97
<code>ligatures de trémolo</code>	114
<code>ligatures et sauts de ligne</code>	97
<code>ligatures manuelles</code>	22, 96
<code>ligne d'extension</code>	29
<code>lignes de portée, épaisseur des</code>	90
<code>lignes de portée, nombre de</code>	90
<code>lilypond-internals</code>	11
<code>line</code>	126
<code>line-markup</code>	126
<code>lookup</code>	126
<code>lookup-markup</code>	126
<code>lower</code>	126
<code>lower-markup</code>	126
<code>lowering text</code>	129
<code>LSR</code>	10

M

<code>magnify</code>	126
<code>magnify-markup</code>	126
<code>makeClusters</code>	142
<code>marcato</code>	101
<code>markalphabet</code>	126
<code>markalphabet-markup</code>	126
<code>markletter</code>	126
<code>markletter-markup</code>	126
<code>masquage d'objets</code>	57
<code>measure repeats</code>	115
<code>medium</code>	126
<code>medium-markup</code>	126
<code>mélisme</code>	29
<code>mesure incomplète</code>	23, 85
<code>mesures, vérification des limites</code>	73
<code>métrique</code>	83
<code>modes anciens</code>	82
<code>mordant</code>	101
<code>moving text</code>	129
<code>musicglyph</code>	126
<code>musicglyph-markup</code>	126
<code>musicMap</code>	143
<code>musique en parallèle</code>	91
<code>musique entremêlée</code>	91

N

<code>natural</code>	126
<code>natural-markup</code>	126
<code>niente, al</code>	105
<code>nolets</code>	23, 71
<code>nolets, formatage</code>	71

<code>nombre de lignes de portée</code>	90
<code>noms d'accords</code>	30
<code>noms de note</code>	62
<code>noms de note, autres langues</code>	64
<code>noms de note, hollandais</code>	63
<code>noms de note, par défaut</code>	63
<code>noPageBreak</code>	142
<code>noPageTurn</code>	142
<code>normal-size-sub</code>	126
<code>normal-size-sub-markup</code>	126
<code>normal-size-super</code>	127
<code>normal-size-super-markup</code>	127
<code>normal-text</code>	127
<code>normal-text-markup</code>	127
<code>normalsize</code>	127
<code>normalsize-markup</code>	127
<code>note</code>	127
<code>note-by-number</code>	127
<code>note-by-number-markup</code>	127
<code>note-markup</code>	127
<code>notes d'ornement</code>	23, 98
<code>nuances</code>	22, 104
<code>null</code>	127
<code>null-markup</code>	127
<code>number</code>	127
<code>number-markup</code>	127

O

<code>octave</code>	141
<code>oldaddyrics</code>	141
<code>on-the-fly</code>	127
<code>on-the-fly-markup</code>	127
<code>orgue, marque de pédale d'</code>	101
<code>ornementation</code>	23
<code>ornementation baroque</code>	101
<code>ornementations, symboles</code>	101
<code>ornements</code>	98
<code>ossia</code>	90
<code>ouvert</code>	101
<code>override</code>	127
<code>override-lines</code>	130
<code>override-lines-markup-list</code>	130
<code>override-markup</code>	127
<code>overrideProperty</code>	140

P

<code>pad-around</code>	127
<code>pad-around-markup</code>	127
<code>pad-markup</code>	127
<code>pad-markup-markup</code>	127
<code>pad-to-box</code>	127
<code>pad-to-box-markup</code>	127
<code>pad-x</code>	127
<code>pad-x-markup</code>	127
<code>padding</code>	53, 56
<code>page-ref</code>	127
<code>page-ref-markup</code>	127
<code>pageBreak</code>	142
<code>pageTurn</code>	141
<code>parallelMusic</code>	141
<code>parenthesize</code>	142
<code>paroles</code>	29

partcombine	142
partition de chanson	30, 31
percent repeats	115
petite note	98
phrasé, liaisons de	95
pipeSymbol	73
pitchedTrill	141
point d'arrêt	101
point d'orgue	101
police	2
polyphonie	27, 75
portato	101
portée multiple	88
portée, lignes de	90
portées, groupe de	88
postscript	127
postscript-markup	127
pouce	101
pousser l'archet	101
première fois	110
propriétés	11
put-adjacent	128
put-adjacent-markup	128

Q

quarts de ton	64
quoteDuring	142

R

r	68
raise	128
raise-markup	128
raising text	129
régulier, espacement	3
régulier, rythme	3
relatif	65
removeWithTag	140
repeatCommands	87, 113
reprises ambiguës	112
reprises avec alternatives et liaisons de prolongation	93
reprises développées	112
reprises et répétitions	110
resetRelativeOctave	141
retoucher	11
retouches, distances	58
right-align	128
right-align-markup	128
rightHandFinger	143
roman	128
roman-markup	128
rotate	128
rotate-markup	128
rotated text	127

S

s	69
sans	128
sans-markup	128
sauts de durée	69
Scheme	11

score	128
score-markup	128
scoreTweak	142
seconde fois	110
segno	101
semiflat	128
semiflat-markup	128
semisharp	128
semisharp-markup	128
sesquiflat	128
sesquiflat-markup	128
sesquisharp	128
sesquisharp-markup	128
sharp	129
sharp-markup	129
shiftDurations	142
silences	68
silences d'espacement	69
silences invisibles	69
simple	129
simple-markup	129
slashed-digit	129
slashed-digit-markup	129
small	129
small-markup	129
smallCaps	129
smallCaps-markup	129
smaller	129
smaller-markup	129
sourdine	101
spacingTweaks	142
staccatissimo	101
staccato	21, 101
Staff notation	81
stemLeftBeamCount	97
stemRightBeamCount	97
stencil	129
stencil-markup	129
strut	129
strut-markup	129
sub	129
sub-markup	129
subdivideBeams	97
super	129
super-markup	129
suppression d'objets	57
symbole de portée	90
symboles d'ornementation	101
symboles musicaux	2
système, début de	88

T

tag	142
teeny	129
teeny-markup	129
tenuto	101
terminologie	10
têtes de note, styles	76
text	129
text-markup	129
tied-lyric	129
tied-lyric-markup	129
tiny	129

<code>tiny-markup</code>	129
tirer l'archet	101
<code>tocItem</code>	140
<code>translate</code>	129
<code>translate-markup</code>	129
<code>translate-scaled</code>	129
<code>translate-scaled-markup</code>	129
translating text	129
<code>transparent</code>	130
<code>transparent-markup</code>	130
transparent, objets	57
<code>transposedCueDuring</code>	140
transposition	67
<code>transposition</code>	141
transposition des hauteurs	67
trémolo, indication de	115
trémolo, ligatures de	114
<code>tremoloFlags</code>	115
<code>triangle</code>	130
<code>triangle-markup</code>	130
trille	101
trilles avec hauteur explicite	108
triolet	23, 71
<code>tupletNumberFormatFunction</code>	71
<code>tweak</code>	143
<code>typewriter</code>	130
<code>typewriter-markup</code>	130
typographie	3, 5

U

<code>unfoldRepeats</code>	142
<code>upright</code>	130
<code>upright-markup</code>	130

V

<code>varcoda</code>	101
variables	11, 39, 45
<code>vcenter</code>	130
<code>vcenter-markup</code>	130
<code>verbatim-file</code>	130
<code>verbatim-file-markup</code>	130
vérification d'octave	66
vérification des limites de mesure	73
versions	31
visionnage de la musique	13
voix changeant manuellement de portée	27
voix multiples sur une portée	27
<code>volta</code>	110

W

<code>whichBar</code>	87
<code>whiteout</code>	130
<code>whiteout-markup</code>	130
<code>with-color</code>	130
<code>with-color-markup</code>	130
<code>with-dimensions</code>	130
<code>with-dimensions-markup</code>	130
<code>with-url</code>	130
<code>with-url-markup</code>	130
<code>withMusicProperty</code>	141
<code>wordwrap</code>	130
<code>wordwrap-field</code>	130
<code>wordwrap-field-markup</code>	130
<code>wordwrap-lines</code>	130
<code>wordwrap-lines-markup-list</code>	130
<code>wordwrap-markup</code>	130
<code>wordwrap-string</code>	130
<code>wordwrap-string-markup</code>	130