

1. BasePartGenerator とは？

- 共通基底クラスを用意し、compose(section, overrides, groove) などのシグネチャを統一します。
 - 休符・ゴースト化、グルーヴ適用、人間味付けなど全パート共通の処理を相続させ、各楽器ジェネレーター(GuitarGenerator など) は演奏ロジックだけに専念。
- これにより “同じ機能を各ファイルでコピペ→片方だけ修正漏れ” という無駄を排除できます。

2. 現状は無駄が多いか？

- 機能重複とヘルパー未完成が混在し、保守コストが上がっているのは事実です。
- ただしコードが “崩れた” というより パッチごとに仕様差が広がったイメージ。
→ 基底クラスで インターフェースを固定し、雛形メソッドを実装すれば整理できます。

楽器ジェネレーター監査 & 改修ロードマップ (2025-06-04)

0 | 対象スコープと入力ファイル

- **Chordmap** : *processed_chordmap_with_emotion.yaml* (全6セクション、感情／強度タグ付き) [?filecite?turn2file0?](#)
- **Overrides** : *arrangement_overrides.json* [?filecite?turn2file1?](#)
- **Groove** : *groove_profile.json* [?filecite?turn2file2?](#)
- 最新ジェネレーター : guitar / bass / drums / piano / melody

1 | 現在の機能力バレッジと要件差分

パート	現状の主機能	要件との差分	リスク
Bass	✓ 感情→パターン LUT✓ セクション別オーバーライド (weak_beat / approach)✓ Groove 同期スタブ	○ <code>_insert_approach()</code> が雛形のみ○ anticipate 弱拍スタイル 未実装○ Velocity / Octave 階層が一部ハードコード	中
Piano	✓ RH/LH の 2・4 拍ゴースト／休符✓ 4 拍目フィル試作	○ フィルパターンが固定 (スタイルマップ不足)○ ペダル制御がグローバルのみ	中～低

Guitar	✓ rhythm_key オーバーライド読 込✓ ヒューマナイ ズ (humanize_timi ng_sec)	✕ 感情マッピング 未対応✕ Palm-mute フラ グ無視✕ ストラム 方向サイクルがス タブ	高
Drums	✓ ゴースト HH / Kick 密度フラグ 読込	✕ 感情 → パターン 切替未実装✕ Groove 同期未適 用	中
Melody	✓ 歌詞タイミング ヒント利用	○ Overrides フ ック無し○ ヒュー マナイズがコード イベント依存	低

2 | 主な痛点

1. インターフェース崩れ：Bass だけ compose(section, overrides) など、シグネチャが不統一。
2. オーバーライド未伝播：Bass/Piano 以外は overrides キーを無視。
3. ヘルパー未完成：_insert_*, _apply_* が雛形のまま放置 → サイレントで機能しない。
4. テスト不足：Bass のオフセット単調性のみ。Guitar/Piano などゼロ。

3 | 推奨リファクタ・スプリント

フェーズ	目的	主タスク	見積
S-1 インターフェース統一	全ジェネレーターで (section, overrides, groove) を受理	・ BasePartGenerator を新設し標準 compose() を定義・ weak-beat や groove-apply ヘルパーを共通化	0.5日
S-2 Bass TODO 完了	アプローチ&アンティシペイト完成	・ _insert_approach() をダイアトニック/クロマチック別に実装・ 弱拍 anticipate (16 分前倒し) を実装	0.5日

S-3 <i>Guitar</i> エンジン拡充	仕様準拠のリズム & ストラム	<ul style="list-style-type: none"> • rhythm_key → pattern JSON・ Palm-mute → velocity/EQ 曲線 • down,down,up,up パーサー 	1.0日
S-4 <i>Drums</i> 高度化	感情連動グルーヴ	<ul style="list-style-type: none"> • 感情×強度 → ドラムパターン LUT・ゴースト HH 密度スカラー • Groove 同期実装 	0.5日
S-5 テスト & CI	退行を防ぐ	<ul style="list-style-type: none"> • pytest で各パート 8 件の核心テスト・GitHub Actions でミニ Chordmap 生成 → diff ゼロ確認 	0.5日

合計 約3日 (集中スプリント)

4 | 即効性のある改善

1. 全パートに **Groove** 同期を適用 — ノート生成後 `apply_groove_pretty()` を呼ぶ。
2. **Palm-mute** 実装 — `palm_mute:true` 時は `velocity-8`、`duration×0.6`。
3. **Override** バリデーション — 未知キーをログ警告。

5 | 未決事項

1. ギターのテンション省略ルール — `b9/#11` を自動除外すべき？
2. ピアノのダイナミックペダル — 和音単位で踏むか、小節単位か？
3. **Melody AI** 強化 — 感情連動メロディ・コンターを将来組み込む？

6 | 結論

アーキテクチャの大枠は初期ビジョンに沿っていますが、**Guitar & Drums** が未達要件、かつヘルパー雛形のままの部分が散見されます。BasePartGenerator から着手する3日スプリントで一気に足並みを揃え、オーバーライド&グルーヴの恩恵を全パートへ波及させるのが最短ルートです。

次のアクション：

1. Sprint S-1 の実装範囲を確定 → パッチ草案を提示
2. 優先パート (Guitar or Bass) を選定し並行着手