# MIDI和弦生成工具快照总结 (2025年8月5日)

## 当前开发状态

### ✅ 已完成核心功能

​**和弦生成引擎**​：

* 1. 完整和弦数据库（日式ACG/华语流行/民谣等）
  2. 罗马数字-音符映射系统
  3. 支持转位和弦计算
  4. 柱式和弦/分解和弦生成逻辑

​**MIDI导出模块**​：

* 1. 支持标准MIDI文件格式(.mid)
  2. 完整的音符事件序列（note\_on/note\_off）
  3. 速度和节拍设置
  4. 直接DAW兼容导出

​**可视化组件**​：

* 1. 钢琴键盘可视化（带和弦高亮）
  2. 和弦信息显示面板（和弦名+音符名）
  3. 五线谱空间预留（未来可扩展）

### 🛠 进行中的模块

​**UI控制系统**​：

* 1. 预制和弦选择器
  2. 和弦网格编辑器
  3. 演奏风格切换面板

​**资源管理系统**​：

* 1. 皮肤管理框架设计
  2. 资源路径映射机制
  3. 配置解析接口

### ⏳ 待开发功能

​**进阶MIDI功能**​：

* 1. 自定义力度曲线
  2. 音符长度微调
  3. 多轨道支持

​**UI增强**​：

* 1. 皮肤实现细节
  2. 和弦网格交互
  3. 音阶选择器
  4. 预制和弦库选择器

​**扩展功能**​：

* 1. MIDI播放预览
  2. 和弦名称自动生成
  3. 和弦复杂度和弦分析

## 核心设计要点总结

### 1. 系统架构

graph TD

A[用户界面] --> B[和弦生成引擎]

A --> C[MIDI导出模块]

A --> D[可视化系统]

B --> E[和弦数据库]

C --> F[标准MIDI文件]

D --> G[钢琴键盘]

D --> H[五线谱视图]

D --> I[和弦信息]

style A fill:#f9f,stroke:#333

style B fill:#ccf,stroke:#333

style C fill:#cfc,stroke:#333

style D fill:#ffc,stroke:#333

### 2. 关键数据结构

# 和弦数据结构

ChordData = {

"roman": "IV", # 罗马数字级别

"type": "min7", # 和弦类型

"inversion": 0, # 转位

"root": "G", # 根音

"octave": 4, # 八度

"duration": 1.0, # 小节长度

}

# 和弦进行序列

Progression = [

ChordData, ChordData, ...

]

# UI配置项

UIConfig = {

"current\_style": "日式ACG",

"progression\_name": "经典进行1",

"key": "C",

"bpm": 120,

"style": "block",

"skin": "default"

}

### 3. 视觉设计方案

graph TB

A[主界面] --> B[左侧控制面板]

A --> C[中部可视化区]

A --> D[底部进度编辑]

B --> E[风格选择下拉菜单]

B --> F[预制进行选择器]

B --> G[调性设置]

B --> H[BPM控制]

B --> I[柱式/分解切换]

C --> J[钢琴键盘视图]

C --> K[和弦信息面板]

D --> L[和弦网格编辑器]

D --> M[和弦属性编辑]

### 4. 资源管理系统

资源结构:

skins/

default/

config.json # 颜色和尺寸配置

images/

button\_normal.png

button\_hover.png

panel\_bg.png

switch\_on.png

switch\_off.png

japanese/

modern/

皮肤管理器功能:

- 按需加载皮肤资源包

- 资源路径自动解析

- 热切换支持

- 提供默认回退机制

## 下一步开发重点

### 短期目标

1. 完成和弦网格编辑器的交互逻辑
2. 实现预制和弦选择器
3. 完成皮肤管理器基础功能
4. 添加基本控制UI元素（下拉菜单/开关/按钮）

### 中期目标

1. 实现完整UI皮肤系统
2. 添加和弦属性编辑弹窗
3. 实现MIDI播放预览功能
4. 添加五线谱可视化视图

### 长期目标

1. 创建和弦复杂度分析器
2. 实现高级演奏风格选项（摇摆节奏等）
3. 开发和弦学习模块
4. 构建扩展和弦库分发系统

## 当前技术栈

* Python 3.10+
* PyGame 2.1+
* Mido 1.2+
* Pygame-GUI (可选)
* Numpy (用于音符计算)

这个快照完整记录了当前设计方向、已完成功能和后续计划。所有模块将保持统一的设计原则，特别是UI系统的可换肤架构将确保未来视觉升级的可能性。