

LyricsRoller

- 功能：为已经打好轴的歌词字幕自动生成类似 Apple Music / 网易云音乐 的滚动歌词

- ✓ 支持卡拉OK效果（需要将参数**字体颜色**设置为**主要颜色**）
- ✓ 支持非英文字符的**自动换行**（但是会把单个英文单词分割到两行中）
- ✓ 支持自定义歌词的**对齐方式**（详见[更多用法](#)）

- 效果对比

原歌词	现歌词
 <p>二度寝(回笼觉) Creepy Nuts 作曲: DJ松永 作词: R-指定</p> <p>想要尝试逃避 エスケープしてみたい</p> <p>【主歌】 【副歌】</p>	 <p>二度寝(回笼觉) Creepy Nuts 作曲: DJ松永 作词: R-指定</p> <p>想要尝试逃避 エスケープしてみたい</p> <p>于是乘坐巴士前往未来 このバスに乗って未来へ 此曲是电视剧『不合适也要有个限度！』的主题曲，剧中主角乘坐一辆时间巴士从昭和时代前往令和时代，并在两个时代往返。剧情讨论昭和时代过于开放令和时代过于死板的社会问题。</p> <p>不回到过去吧 いや、はるか昔</p> <p>唉 还是哪儿都别去了 まあどっちもとんでもないぜ</p> <p>过去的人们早就不在了 【主歌】 【副歌】</p>

- 视频效果演示

(可见示例文件 `二度寝_lyrics_roller.mp4`)

<https://github.com/user-attachments/assets/d689d208-b66d-488e-a678-5682f5c63efe>

- 示例位于 `/example` 文件夹下，其包含以下文件：

- 二度寝.ass (已生成滚动歌词的字幕文件，可自行复原)
- 二度寝.mp4 (字幕用到的视频文件)
- 二度寝_lyrics_roller.mp4 (使用上述两个文件压制成的最终视频文件)
- 二度寝-karaok.ass (卡拉OK示例的字幕文件)
- 二度寝-karaok_lyrics_roller.mp4 (卡拉OK字幕文件生成的最终视频文件)

! 所有图片和视频演示均为v1.0版本，新增功能请在[更新历史](#)查看

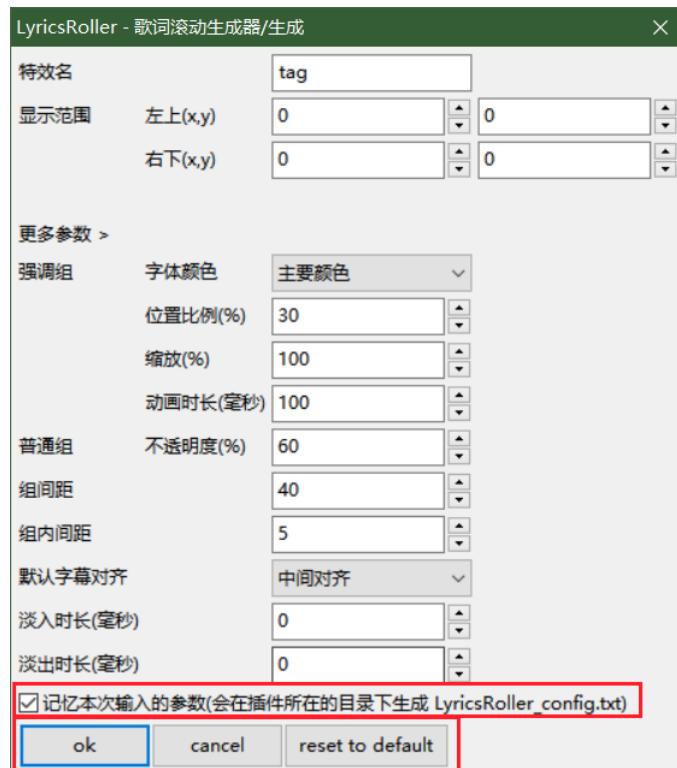
目录

- [LyricsRoller](#)
 - [更新历史](#)
 - [目录](#)
 - [使用方法](#)
 - [载入插件](#)
 - [基本使用方式](#)
 - [你需要注意的](#)
 - [名词解释](#)
 - [参数解释](#)
 - [更多用法](#)
 - [卡拉OK效果](#)
 - [单独指定对齐方式](#)
 - [修改插件对话框中的默认值](#)

更新历史

- v1.2.2
 - 增加透明度不渐变的选项
 - 勾选此选项，对应着之前版本中默认的情况：显示范围内第一组和最后一组字幕呈现逐渐清晰/逐渐透明效果
 - 不勾选此选项，则直接在显示范围边界截断字幕
 - 修改透明度渐变的计算方式为余弦函数而非线性
 - 修复某些情况下生成滚动字幕会使得Aegisub爆内存的bug
- v1.2.1
 - 新增卡拉OK模式型字体颜色选项
 - 勾选此选项后，强调组和普通组的字体颜色配置会覆盖对话框中单独指定的字体颜色，且配置如下：
 - 强调组为次要颜色转主要颜色
 - 强调组上方普通组使用主要颜色
 - 强调组下方普通组使用次要颜色
 - 新增普通组-字体颜色选项，作用类似强调组-字体颜色
 - 修复卡拉OK模式下出现的这些问题：
 - 单行字幕换行后的第一个字符没有颜色切换动画
 - 没有强调组的时间段中所有行会同时进行颜色切换动画
- v1.1.1
 - 修复缩小强调行时没有动画的问题
 - 修复复原功能无效的问题
 - 增加生成 & 复原后的消息提示
- v1.1
 - 新增保存输入参数的功能

- 在对话框中勾选 **记忆本次输入的参数** 即可
- 点击 **reset to default** 可恢复最初的默认参数
- 现在可以通过**Enter键**和**Esc键**快捷选择对话框按钮的**ok**和**cancel**了



- v1.0

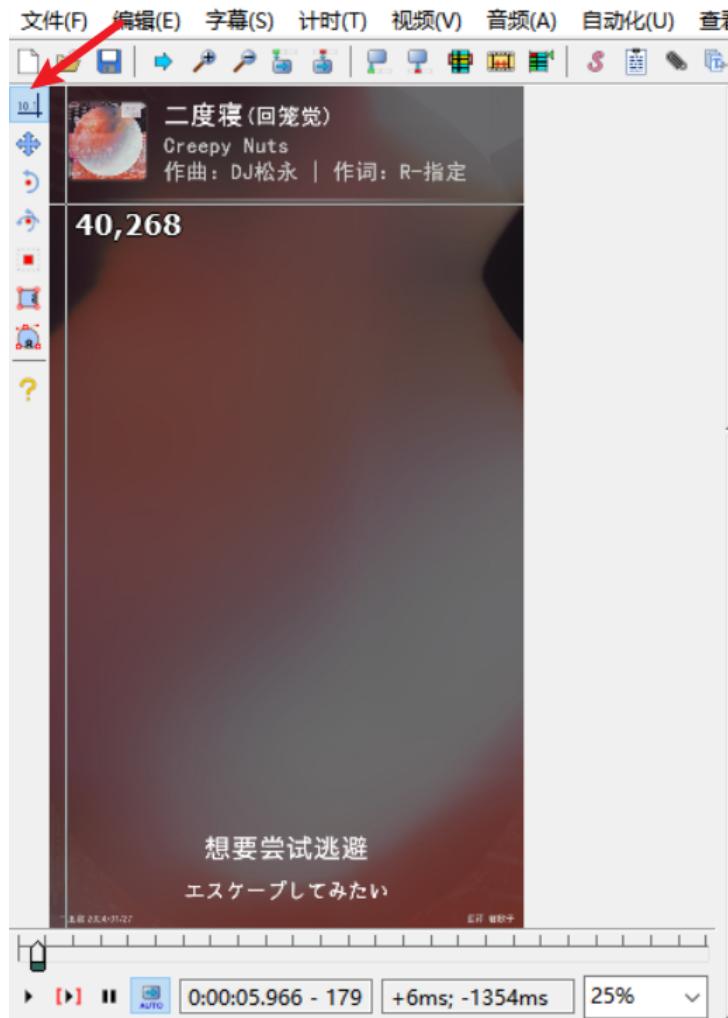
使用方法

载入插件

1. 在**Release**列表中下载最新的文件 `LyricsRoller.lua`
2. 在**Aegisub**中载入插件
 - 方法一（推荐）：打开Aegisub的本地文件夹目录，进入子目录 `Aegisub/automation/autoload`，将下载的文件 `LyricsRoller.lua` 放入此文件夹中
 - ▶ 图片示例
 - 方法二：（菜单）自动化 → 自动化 → 载入 → 选择 `LyricsRoller.lua`
 - ▶ 图片演示

基本使用方式

1. 在**标准模式**找出想要显示滚动歌词的矩形范围的**左上角**和**右下角**的点的坐标

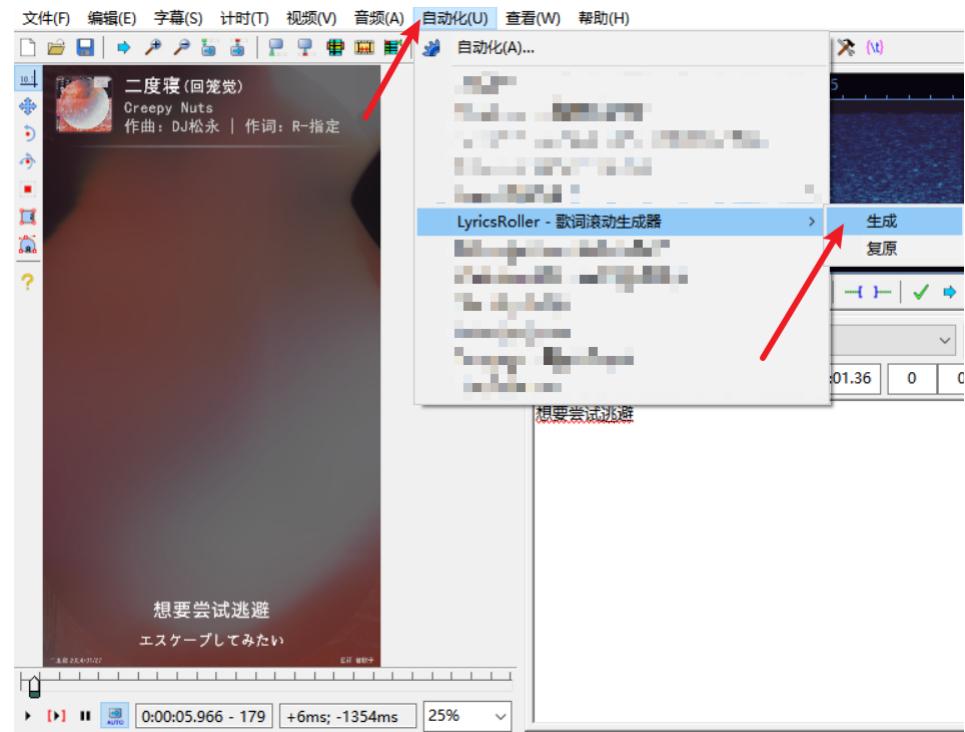


2. 将所有需要用于滚动歌词的行设置一个相同的特效名

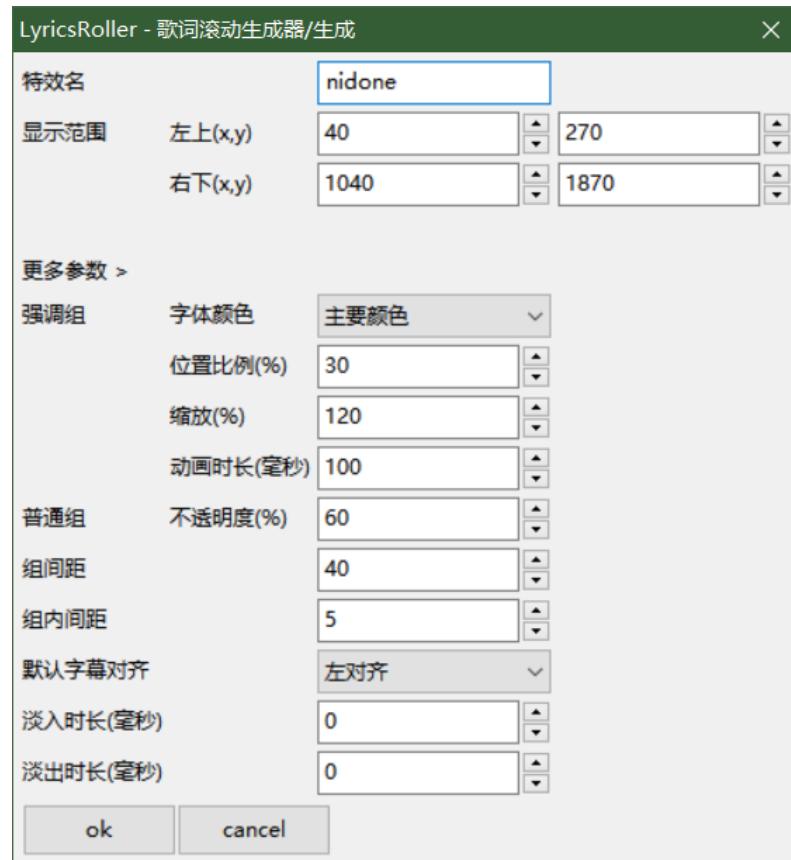
#	开始时间	结束时间	字/秒	样式	特效	文本
1	0:00:00.00	0:04:14.00	0	滚动歌曲介绍 - 创作者		◎作曲: DJ松永 作词: R-指定
2	0:00:00.00	0:04:14.00	0	滚动歌曲介绍 - 创作者		◎Creepy Nuts
3	0:00:00.00	0:04:14.00	0	滚动歌曲介绍 - 标题		◎二度寝(回笼觉)
4	0:00:00.00	0:00:05.96	0	滚动歌词 - 原文	nidone	
5	0:00:05.96	0:00:07.32	4	滚动歌词 - 译文	nidone	想要尝试逃避
6	0:00:07.32	0:00:08.71	7	滚动歌词 - 译文	nidone	于是乘坐巴士前往未来
7	0:00:08.71	0:00:10.05	5	滚动歌词 - 译文	nidone	不回到过去吧
8	0:00:10.05	0:00:11.44	7	滚动歌词 - 译文	nidone	唉还是哪儿都别去了
9	0:00:11.44	0:00:12.77	7	滚动歌词 - 译文	nidone	过去的人们早就不在了
10	0:00:12.77	0:00:14.12	6	滚动歌词 - 译文	nidone	我们也没有用处了吗?
11	0:00:14.12	0:00:16.81	10	滚动歌词 - 译文	nidone	无论生活在何时都会想着“像这样的时代”yeah yeah
12	0:00:16.81	0:00:18.24	9	滚动歌词 - 译文	nidone	oh shit 看那浦岛太郎
13	0:00:18.24	0:00:19.60	5	滚动歌词 - 译文	nidone	轻轻合上了玉匣
14	0:00:19.60	0:00:20.92	5	滚动歌词 - 译文	nidone	从中升起的白烟
15	0:00:20.92	0:00:22.32	7	滚动歌词 - 译文	nidone	让世间一切都发生了变化
16	0:00:22.32	0:00:23.69	10	滚动歌词 - 译文	nidone	oh shit 我就像是浦岛
17	0:00:23.69	0:00:25.04	5	滚动歌词 - 译文	nidone	而传说故事是烟
18	0:00:25.04	0:00:26.39	6	滚动歌词 - 译文	nidone	当它从我身边飘过时
19	0:00:26.39	0:00:27.94	7	滚动歌词 - 译文	nidone	我咳嗽几声皱起了眉头
20	0:00:27.94	0:00:30.66	8	滚动歌词 - 译文	nidone	warning warning 不合适的语录
21	0:00:30.66	0:00:33.32	4	滚动歌词 - 译文	nidone	咔擦咔擦 没注意就被火烧伤

! 建议在第一句歌词前增加一个从音乐开始时间到第一句歌词开始时间的空行（如上图的第四行）、并在最后一句歌词后增加一句空行持续到音乐结尾，从而让歌词显示效果更加完整
（目的：延长插件所生成的歌词的显示时间至整首歌的时长）

3. (菜单) 自动化 → LyricsRoller - 歌词滚动生成器 → 生成

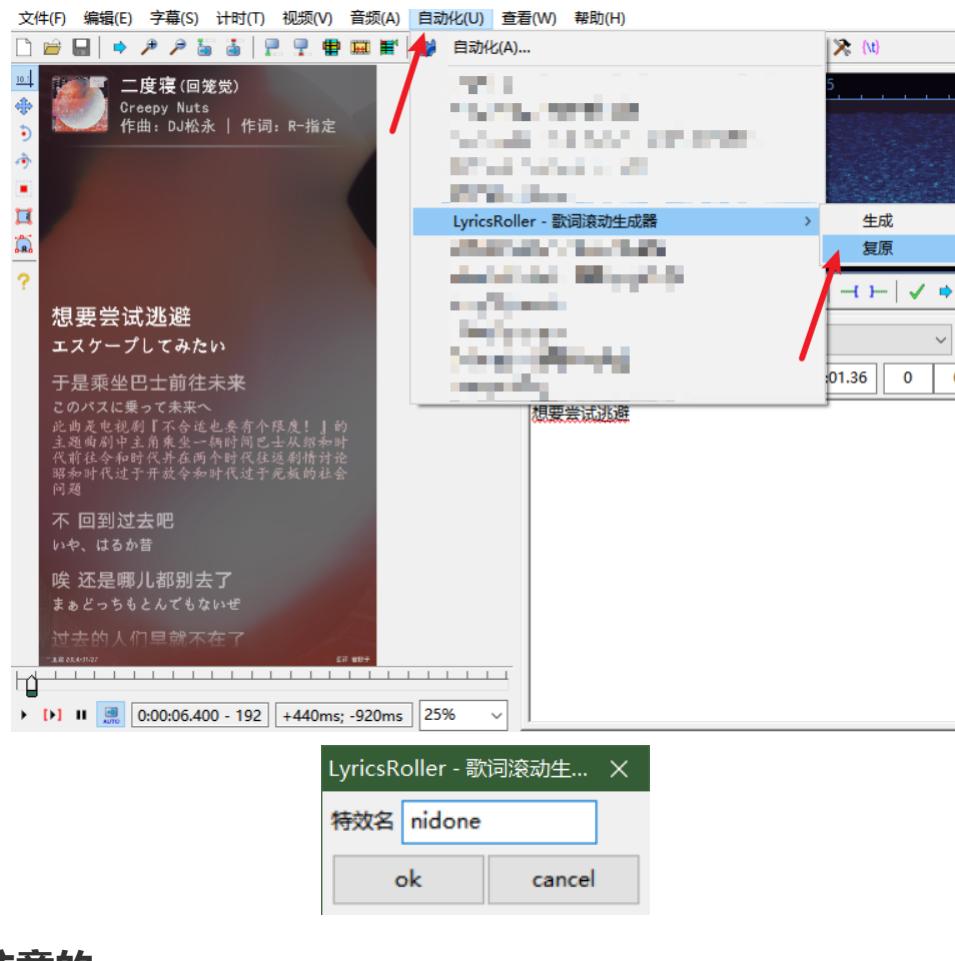


2. 在**特效名**中填写你在**2**中设置的特效名、在**显示范围**中填写你在**1**记录的矩阵范围的坐标点，并根据自己的需要设置**更多参数**



! **更多参数**的解释及效果见[参数解释](#)部分

3. 点击**ok**, 会生成滚动歌词，并自动覆盖之前使用相同特效名所生成的滚动歌词，生成结果可见[一开始的对比图](#)
4. 若要手动删除所有使用此特效名生成的滚动歌词，点击（菜单）自动化 → LyricsRoller - 歌词滚动生成器 → 复原，输入使用的特效名，并点击**ok**



你需要注意的

- LyricsRoller会自动将具有相同开始时间的字幕行视作**同一组**，因此请为需要在同一时间内显示的歌词（如原文+译文+注释）设置相同的开始时间
- 对于**同一组**的字幕行，LyricsRoller会按照原来的字幕顺序将其放在一起
 - 例如，如果你想在最终的滚动歌词中，让译文显示在原文上方，那么你应该在使用插件前将译文对应的行放在原文的行的上方
 - 一个简单的方法是按照**样式名**将所有行进行**排序**，并把所有译文样式的行剪切放在原文样式的行的上方

	译文在原文上方						译文在原文下方							
	#	开始时间	结束时间	字/秒	样式	特效	文本	#	开始时间	结束时间	字/秒	样式	特效	文本
◦	1	0:00:05.96	0:00:07.32	4	滚动歌词 - 译文	nidone	想要尝试逃避	1	0:00:05.96	0:00:07.32	7	滚动歌词 - 原文	nidone	エスケープしてみたい
	2	0:00:07.32	0:00:08.71	7	滚动歌词 - 译文	nidone	于是乘坐巴士前往未来	2	0:00:07.32	0:00:08.71	7	滚动歌词 - 原文	nidone	このバスに乗って未来へ
	3	0:00:05.96	0:00:07.32	7	滚动歌词 - 原文	nidone	エスケープしてみたい	3	0:00:05.96	0:00:07.32	4	滚动歌词 - 译文	nidone	想要尝试逃避
	4	0:00:07.32	0:00:08.71	7	滚动歌词 - 原文	nidone	このバスに乗って未来へ	4	0:00:07.32	0:00:08.71	7	滚动歌词 - 译文	nidone	于是乘坐巴士前往未来

生成结果

想要尝试逃避
エスケープしてみたい
于是乘坐巴士前往未来
このバスに乗って未来へ

エスケープしてみたい
想要尝试逃避
このバスに乗って未来へ
于是乘坐巴士前往未来

- 请在使用LyricsRoller前，尽量确保字幕已经正常打好了轴，否则你应当清楚LyricsRoller对不同开始时间、结束时间的统一方式：
 - LyricsRoller会自动统一**同一组**字幕的**结束时间**，此时间来自于此组的字幕里**最晚的结束时间**
 - 若某组的结束时间晚于**下一组的开始时间**，则修改此结束时间为下一组的开始时间
 - 若两个**组**之间的间隔时间小于**强调组的动画时长**（见[参数解释](#)），则会将上一组的结束时间修改为下一组的开始时间

- 请尽量不要在行中手动修改**字体大小**或进行其他有可能影响到字体大小的操作（比如修改字体），因为**自动换行**功能会忽略行内的所有特效
- 请尽量不要在行中使用换行符 \n 或 \N 或手动空格等字符，因为**自动换行**功能不会解析这些代码，且会视为一个个**独立的字符**，从而导致自动换行可能出现错误
- LyricsRoller理论上**不支持**Aegisub的**注音** (furi) 功能，因为其生成的注音的位置是固定的，但LyricsRoller在生成滚动字幕时不会去计算同一组内不同行字幕之间的相对位置
 - 本人目前没有注音滚动字幕的需求，因此暂不考虑增加此功能，不过也欢迎各位进行改进

名词解释

- 组：所有开始时间相同的字幕行的集合
- 强调组：当前的音频时间处于某一组的开始时间与结束时间之内，则此组称为强调组
 - 对应音乐歌词中当前唱到的那一句
- 普通组：强调组以外的其他组

参数解释

参数名	参数名	数据类型	数据范围	作用
特效名		字符串	-	指定用于生成的字幕行
显示范围	(x,y)	整数	-	指定生成歌词的显示范围
强调组	字体颜色	-	主要颜色 / 次要颜色	指定 强调组 中字幕行的字体颜色为其对应样式中的 主要颜色 还是 次要颜色 ！改变样式中的字体颜色后需要 重新生成 才能应用颜色的修改
	位置比例	整数	0 - 100	指定 强调行 在显示范围内的位置 此数值表示强调组顶部的y轴位置在整个显示范围内的比例（最上面是0，最下面是1）
	缩放	整数	0 ~	指定 强调行 相对普通行的字体大小的比例 100即为相同大小 ！若要使此特效有效，需要用到插件 VSFilterMod ，使用方式和用于压制的方式请自行搜索
	动画时长	整数	0 ~	指定普通行变为强调行、强调行变为普通行时的动画的时长(毫秒) 动画包括：颜色变化、位置移动、缩放 动画所占用的时间均为强调行的时间

参数名	参数名	数据类型	数据范围	作用
普通组	字体颜色	-	主要颜色 / 次要颜色	指定 普通组 中字幕行的字体颜色为其对应样式中的 主要颜色还是次要颜色 ！改变样式中的字体颜色后需要 重新生成 才能应用颜色的修改
	不透明度	整数	0 - 100	指定 普通行 字体的透明度 100即为完全不透明 与之对应的，强调行的不透明度为100%
组内间距		整数	0 ~	组内不同字幕行之间的距离
组间距		整数	0 ~	两组之间的距离
默认字幕对齐		-	左对齐 / 中间对齐 / 右对齐	字幕在整个显示范围中的x轴对齐方式，单独指定方式见 更多用法
淡入时长		整数	0 ~	最早的开始时间时，淡入显示整个滚动歌词的时长(毫秒)
淡出时长		整数	0 ~	最晚的结束时间时，淡出显示整个滚动歌词的时长(毫秒)
首末组字幕透明度渐变		布尔型	勾选 / 未勾选	显示范围内的第一组和最后一组字幕会有透明度渐变效果 取消后只做硬裁剪
卡拉OK模式型字体颜色		布尔型	勾选 / 未勾选	勾选此选项后，强调组和普通组的字体颜色配置会覆盖对话框中单独指定的字体颜色，且配置如下： 强调组 为次要颜色转主要颜色 强调组上方普通组 使用主要颜色 强调组下方普通组 使用次要颜色
记忆本次输入的参数		布尔型	勾选 / 未勾选	勾选会在插件目录下生成 LyricsRoller_config.txt 以保存参数信息 点击 reset to default 可恢复最初的默认参数

更多用法

卡拉OK效果

- 使用方式：正常在字幕行中加入 \k 等标签
 - 若仅想要**正在唱的行**有特效变化，则指定强调行的颜色为**主要颜色、并不勾选 卡拉OK模式型字体颜色**这一选项

- 若希望整体都有颜色变化效果（已唱的为主要颜色，未唱的为次要颜色），则勾选 **卡拉OK模式型字体颜色** 这一选项
- v1视频效果演示（仅日语歌词有卡拉OK效果；新增的**卡拉OK模式型字体颜色**请自行体验）
(可见示例文件 [二度寝-karaok_lyrics_roller.mp4](#))
<https://github.com/user-attachments/assets/f505a33b-e40a-4a3c-b40a-1469cb47560e>

单独指定对齐方式

- 为某一行的特效名增加后缀 **-l / -L / -c / -C / -r / -R**，可指定此行的对齐方式，分别为左对齐、中间对齐、右对齐
- 当不指定时，会使用默认的字幕对齐方式
- 示例：

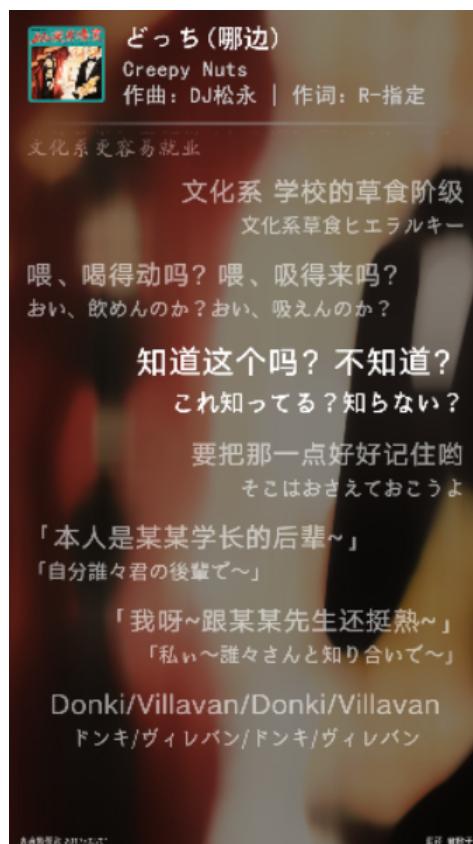
1. 指定特效名

#	开始时间	结束时间	字/秒	样式	特效	文本
1	0:00:00.00	0:04:56.00	0	滚动歌曲介绍 - 创作者		○作曲: DJ松永 作词: R-指定
2	0:00:00.00	0:04:56.00	0	滚动歌曲介绍 - 创作者		○Creepy Nuts
3	0:00:00.00	0:04:56.00	0	滚动歌曲介绍 - 标题		○どっち(哪边)
4	0:02:45.21	0:02:47.53	5	滚动歌词 - 译文	docchi-l	体育会系 学校的内食阶级
5	0:02:47.53	0:02:49.56	5	滚动歌词 - 译文	docchi-r	文化系 学校的草食阶级
6	0:02:49.56	0:02:51.51	5	滚动歌词 - 译文	docchi-l	喂、喝得动吗？喂、吸得来吗？
7	0:02:51.51	0:02:52.47	8	滚动歌词 - 译文	docchi-r	知道这个吗？不知道？
8	0:02:52.47	0:02:53.46	10	滚动歌词 - 译文	docchi-r	要把那一点好好记住哟
9	0:02:53.46	0:02:55.42	5	滚动歌词 - 译文	docchi-l	「本人是某某学长的后辈～」
10	0:02:55.42	0:02:57.35	6	滚动歌词 - 译文	docchi-r	「我呀~跟某某先生还挺熟～」
11	0:02:57.35	0:02:59.31	13	滚动歌词 - 译文	docchi	Donki/Villavan/Donki/Villavan
12	0:02:45.21	0:02:47.53	5	滚动歌词 - 原文	docchi-l	体育会系内食ヒエラルキ一
13	0:02:47.53	0:02:49.56	5	滚动歌词 - 原文	docchi-r	文化系草食ヒエラルキ一
14	0:02:49.56	0:02:51.51	7	滚动歌词 - 原文	docchi-l	おい、飲めんのか？おい、吸えんのか？
15	0:02:51.51	0:02:52.47	10	滚动歌词 - 原文	docchi-r	これ知ってる？知らない？
16	0:02:52.47	0:02:53.46	11	滚动歌词 - 原文	docchi-r	そこはおさえておこうよ
17	0:02:53.46	0:02:55.42	5	滚动歌词 - 原文	docchi-l	「自分誰々君の後輩で～」
18	0:02:55.42	0:02:57.35	7	滚动歌词 - 原文	docchi-r	「私い～誰々さんと知り合いで～」
19	0:02:57.35	0:02:59.31	8	滚动歌词 - 原文	docchi	ドンキ/ヴィレパン/ドンキ/ヴィレパン
20	0:02:45.21	0:02:47.53	10	滚动歌词 - 注释	docchi-l	体育会系：重视精神、本性、上下关系 相对文化系更容易就业

2. 默认中间对齐



3. 生成效果



修改插件对话框中的默认值

- 打开源文件 `LyricsRoller.lua`，修改数组 `generate_config` 里对应属性的 `value` 即可
 - ! 显示范围的 `右下(x,y)` 无法通过此方式修改，因为其会自动获取视频的尺寸并设置为右下角的位置
- 示例：参数名以及值的对应

generate_config=

```
{
    {class="label",x=0,y=0,label="特效名"}, {class="edit",name="tag",x=2,y=0,value="tag"}, {class="label",x=0,y=1,label="显示范围"}, {class="label",x=1,y=1,label="左上(x,y)"}, {class="intedit",name="bound_1x",x=2,y=1,value=0}, {class="intedit",name="bound_1y",x=3,y=1,value=0}, {class="label",x=1,y=2,label="右下(x,y)"}, {class="intedit",name="bound_2x",x=2,y=2}, {class="intedit",name="bound_2y",x=3,y=2}, {class="label",x=0,y=4,label="更多参数 >"}, {class="label",x=0,y=5,label="强调组"}, {class="label",x=1,y=5,label="字体颜色"}, {class="dropdown",name="color",x=2,y=5,items={"主要颜色","次要颜色"},value="主要颜色",hint="强调组使用字体"}}
```