

【优化记录】2022/10/26内部讨论

方案：

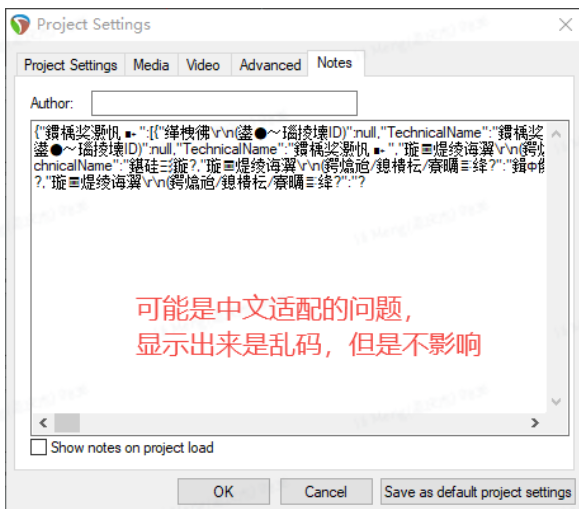


工程的item数量太多会直接导致自适应reascript 0.2版本卡顿，为解决这个问题，并实现item打metadata的功能，遵循以下优化方案。

- Excel的key words属性以JSON的形式写入 Reaper_Project 的 NOTES里。因为模板工程中的region、maker、track都是由这个JSON生成的，所以打标签时可以直接通过调用region、marker、track的id 将单块的信息对应到相应的item。

NOTES只支持字符串形式，所以需要先转成String类型，打Metadata调用时再转成JSON。预演形式如下图。

reaper工程的NOTES：



将reaper工程的NOTES重新解析为JSON：

```

▼ {暗影小兔: Array(6), 匹诺曹: Array(6), 爱丽丝: Array(6), 白雪公主: Array(6), 快乐小丑: Array(6), ...} ⓘ
▼ 丘比特: Array(6)
  ▼ 0:
    Audio File Name(音频文件命名): "Vo_Story_Entry_Cupid"
    Character: "Cupid"
    Dialogue(CN)(台词): "正中红心"
    TechnicalName: "丘比特"
    Trigger: "Entry"
    Type: "Story"
    备注: null
    情绪: "得意"
    播放场景/触发条件: "入场"
    "编号 (可填角色ID)": null
    "语音类型 (功能/战斗/引导等)": "剧情语音"
    音频文件命名: "Vo_Story_Entry_Cupid"
    ▶ [[Prototype]]: Object
    ▶ 1: {"编号 (可填角色ID)": null, TechnicalName: '丘比特', "语音类型 (功能/战斗/引导等)": '剧情语音', 情绪: '得意'}
    ▶ 2: {"编号 (可填角色ID)": null, TechnicalName: '丘比特', "语音类型 (功能/战斗/引导等)": '呼喝声', 情绪: null}
    ▶ 3: {"编号 (可填角色ID)": null, TechnicalName: '丘比特', "语音类型 (功能/战斗/引导等)": '呼喝声', 情绪: null}
    ▶ 4: {"编号 (可填角色ID)": null, TechnicalName: '丘比特', "语音类型 (功能/战斗/引导等)": '呼喝声', 情绪: null}
    ▶ 5: {"编号 (可填角色ID)": null, TechnicalName: '丘比特', "语音类型 (功能/战斗/引导等)": '呼喝声', 情绪: null}
    length: 6
    ▶ [[Prototype]]: Array(0)
  ▶ 仙灵小店: (3) [{"..."}, {"..."}, {"..."}]
  ▶ 力量工坊: (3) [{"..."}, {"..."}, {"..."}]
  ▶ 匹诺曹: (6) [{"..."}, {"..."}, {"..."}, {"..."}, {"..."}, {"..."}]
  ▶ 占星室: (3) [{"..."}, {"..."}, {"..."}]
  ▶ 和平小屋: (3) [{"..."}, {"..."}, {"..."}]
  ▶ 奸商: (5) [{"..."}, {"..."}, {"..."}, {"..."}, {"..."}]
  ▶ 小猫商店: (3) [{"..."}, {"..."}, {"..."}]

```

- 为解决CPU的效率问题，region及marker不再进行实时监测。而是通过GUI可以单开一个录音轨，等录音导演剪辑完成后计算所需region长度后，再一并进行延长、平移操作。

注意事项：

1、region、marker、item 的操作顺序并不是按照 id 进行的，而是按照实际在工程中的前后顺序。为了避免region、marker之间因程序指令执行的时间差而出现相互重叠（或超越）的情况，应分情况考虑是否倒序批量操作。

```

1  if moveLength > 0
2  then
3    for j = num_regions - 1, inRegionNum + 1, -1 do
4      --for循环从最后一个region开始
5    end
6  else
7    for j = inRegionNum + 1, num_regions - 1, 1 do
8      --for循环从第一个region开始
9    end
10 end

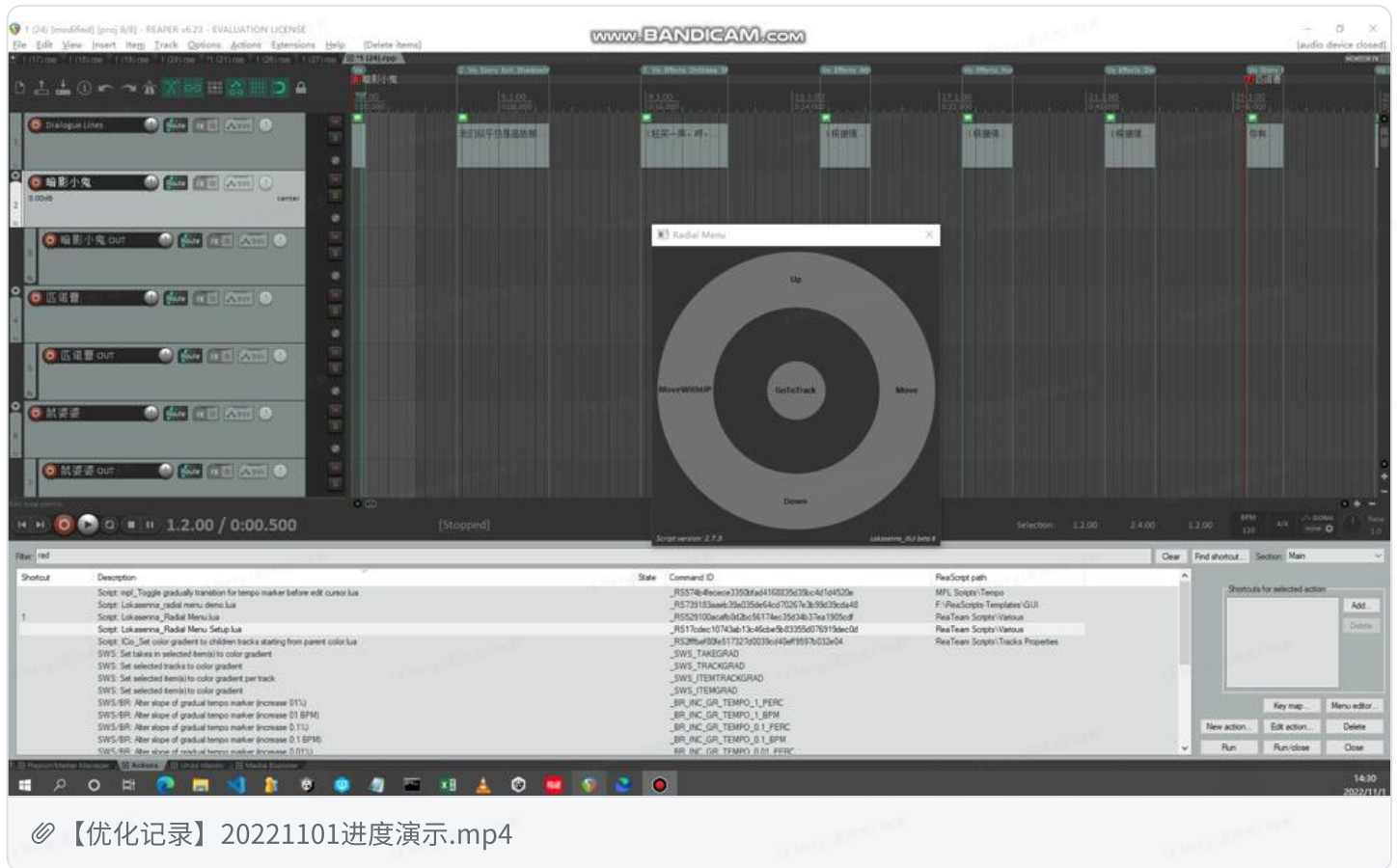
```

2、用GUID作为item的唯一标识

```
1 ItemGUID[i] = reaper.BR_GetMediaItemGUID(item);
```

成果：

自适应



GoToTrack:

通过输入speaker的名字到达录音轨

UP:

向上移动选择的Items一个track

Down:

向下移动选择的Item一个track

MoveWithUp:

将选中的takes从Out轨移动到主轨，并对应Region，其他Region自动补齐。

Move:

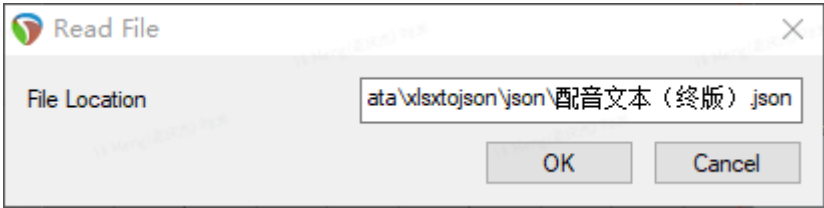
将选中的takes对应Region，其他Region自动补齐。

名称	类型	Command ID
GoToTrack	Custom: Giant_SelectTrackByName	_9d1211125bbba34486801b93c1f289c
	Script: Giant_SelectTrackByName.lua	_RS2d6773ea306b81e5bd8c7bffa57dk
	Item navigation: Select and move to item in next track	40419
Up	Item edit: Move items/envelope points up one track/a bit	40117
Down	Item edit: Move items/envelope points down one track/a bit	40118
MoveWithUp	Custom: Giant_MoveWithCursorWithUp_0.3	_8233f43284ed9a40a679704262ad82f
	Script: Giant_MoveWithCursor_0.3.lua	_RS24955eb42afd72862db4eb33db5d
	Item edit: Move items/envelope points up one track/a bit	40117
Move	Script: Giant_SelectTrackByName.lua	_RS2d6773ea306b81e5bd8c7bffa57dk

打MetaData

reaper的project note长度有限制，当语音很多时，会漏掉最大限度之后的信息。所以打算从reaper内直接读取源数据Json，然后通过解析，将键值对发送到相应的item 的note里。然后通过item note设置每一句语音的metadata。

Step 1、输入元数据的绝对地址，点击OK，会将所有的item的note都写入对应的键值对



Step 2、render时的设置

