写在前面

此文档不是程序使用说明，如果未了解程序基本使用方法请查看打包内 使用说明 – 简中.pdf。

自定义读取模块设计的初衷是让用户能够自己定义弹幕文件的格式，比如影视工作者想要在自己的视频上加入弹幕的效果，而剪辑软件做出来的弹幕非常地生硬，就可用利用此功能来按照需求快速地设计自己的弹幕文件。

目前该模块处于试验阶段，很多东西都不完善，如果不按照语法要求，有小概率不会报错，但解析会出现问题。

如果您没有一定的计算机基础，这篇文章可能会有点难，还请谅解，语法已经尽我所能去精简了，这个模块在未来有可能会适配在 GUI 版本上，但暂时没有比较好的交互方式。

先来看看例子（希望不是劝退）

```
DanmakuFactory_Template_Version: 1.0 # 这是第一个版本哦

format:
{"time":$time$,"type":$type$,"fontsize":$fontsize$,"content":"$text$","color":$color$}

set $time$:
    Ratio_to_seconds = 1000

set $type$:
    right_to_left = 0
    left_to_right = 1
    top_fix = 2
    bottom_fix = 3
    default_type = right_to_left

set $color$:
    default_color = 16777215

set $fontsize$:
    default_fontsize = 38
```

一个模板文件由模板文件说明、格式说明、变量说明三部分组成。
其中字符'#'开头表示注释，每一行 '#' 以后的内容都会被忽略，缩进和部分换行不是必须的，但推荐这样做,推荐按上例这样写。
如果觉得还行的话，那开始吧!

模板文件说明

模板文件的第一行必须是模板版本说明，以方便向下兼容，正确的说明如下:

```
DanmakuFactory_Template_Version: 1.0
```

格式说明

语法：

```
format:
格式说明语句
```

例子：

如果你定义的文件如下：

```
{"time":1.00,"type":1,"fontsize":25,"content":"233","color":000000},
{"time":2.00,"type":1,"fontsize":25,"content":"666","color":000000},…
```

正确的格式的说明应该是这样的：

```
format:
{"time":$time$,"type":$type$,"fontsize":$fontsize$,"content":"$text$","color":$color$}
```

如果你接触过编程语言，你会发现这与格式输入输出语法很相似，我们来细说一下。

先写 format: 表明后面是格式说明，格式说明分为字符串常量和变量，上面的例子中，字符串常量已经用浅绿色标明，变量由橙黄色标明。字符串常量表示一些每一条弹幕都有，而且不会变的内容。变量的用途是替换和提取弹幕中会改变的内容，比如时间、评论内容等。

可以用的变量：

$time$
用于替换提取时间，必须是一个整数或小数。
$type$
用于替换提取弹幕类型，必须是一个整数。
$color$
用于替换提取颜色，必须是一个整数。
$fontsize$
用于替换提取文字大小，必须是一个整数。
$text$
用于替换提取评论文本内容，应当是一个字符串。
$anynumber$
用于替换会变化但又没有提取出来意义的任意数字。
$anystring$
用于替换会变化但又没有提取出来意义的任意文本。

格式说明的约定：

1. (1)格式说明开头不能是变量；(2)两个变量之间必须要有字符串常量分隔；（3）字符串变量（$text$、$anystring$）不能作为结尾。否则无法定位，如果这样写，就算是人工也无法准确分割。下面举几个例子。

```
$time$,$text$,$fontsize$    # 错误 变量不能作为开头
[$time$$text$$fontsize$]    # 错误 两个变量之间必须要有字符串常量分隔
[$time$,$fontsize$,$text$   # 错误 字符串变量（$text$、$anystring$）不能作为结尾
[$time$,$text$,$fontsize$] # 正确（就算只有一个逗号，也是一个字符串常量，但不建议这样分隔）
```

2. 字符串常量中不能出现字符'#'，不然会解析失败，可以出现字符 '$' 但不能出现变量名（如：$time$）。

3. 为了方便模板文件格式化，从 format:开始到第一个非空白字符之间的任意空白字符（换行符、空格、tab 等）将会被忽略，格式说明行最后一个非空白字符后的空白字符也会被忽略。但是中间的空白字符不会忽略。
如下↓表示换行符，→表示空格，会被忽略的符号用红色底色表示，不会被忽略的符号用绿色底色表示。

```
DanmakuFactory_Template: V1.0↓
format: ↓
↓
→→→{↓
 "time":$time$,"fontsize":$fontsize$,"count":$anynumber$,"content":"$text$"↓
→→→}↓
↓
set $time$:↓
Ratio_to_seconds = 1 ↓
```

4. 弹幕文件不需要保证开头便是弹幕内容，两条弹幕之前也可以有一些用于分隔的字符而无需在格式说明中声明，下面举个例子：

假设你定义的弹幕文件如下：

```
{"type":"json","title":"MyOwnDanmakuFile",[
{"time":1.00,"type":1,"fontsize":25,"content":"233","color":000000},
{"time":2.00,"type":1,"fontsize":25,"content":"666","color":000000},
{"time":3.00,"type":2,"fontsize":30,"content":"666","color":000000}
]}
```

模板文件的格式说明只需要说明单条弹幕文件的格式即可，前后内容以及两条弹幕之间的分隔符号无需且不能声明：

```
format:
{"time":$time$,"type":$type$,"fontsize":$fontsize$,"content":"$text$","color":$color$}
```

5. 在格式说明中，$time$ 和 $text$ 必须且仅出现 1 次；$type$、$color$、$fontsize$ 可以出现 0 或 1 次，$anystring$ 和 $anynumber$ 没有次数限制。

变量说明

语法：

```
set 变量名
    配置项 1 = 值 1
    配置项 2 = 值 2
    ...
```

例子：

```
set $type$:
    right_to_left = 0
```

可用的配置项：

变量 $time$

1. Ratio_to_seconds 定义$time$跟实际秒数的比例，可以是小数，应当满足：(Ratio_to_seconds =

$time$ ÷ 实际秒数)
如果 $time$ 提取到的值是分钟值，那 Ratio_to_seconds = 0.016667（1/60）;
如果不指定，则默认是 1，$time$ 提取到的值是秒数。

变量 $type$
1. right_to_left 定义右向左滚动弹幕的数字值，这个数字值是 $type$ 替换的值。
2. left_to_right 定义左向右滚动弹幕的数字值。
3. top_fix 定义顶部固定弹幕的数字值。
4. bottom_fix 定义底部固定弹幕的数字值。
5. default_type 定义解析错误的弹幕将会解析到的类型，可选的值有 right_to_left、left_to_right、top_fix、bottom_fix。

使用例子
```
set $type$:
    right_to_left = 0 # 右向左滚动弹幕，例 {"type":0,"font... 将会被解析成右左弹幕
    left_to_right = 1 # 左向右滚动弹幕，例 {"type":1,"font... 将会被解析成左右弹幕
    top_fix = 2       # 顶部固定弹幕
    bottom_fix = 3    # 底部固定弹幕
    default_type = right_to_left
```

如果你自定义的弹幕文件没有定义弹幕类型，可以用 default_type 来统一指定而无需再写 right_to_left、left_to_right、top_fix、bottom_fix。
如果不设定 $type$ 的任何配置项，则默认全部是右左滚动弹幕（right_to_left）。

变量 $color$
1. default_color 表示将颜色读入错误的弹幕全部设定为某种颜色，若不设定默认为白色。

变量 $fontsize$
1. default_fontsize 表示将字体大小读入错误的弹幕全部设定为某个大小，如不设定默认为 25。

变量说明的约定：

1. 全部变量及其配置项可以按需设定，没有指定值的全部为默认值。

https://github.com/hihkm/DanmakuFactory
Copyright © 2019-2020 TIKM.ORG