## 概述

### 基本信息

1）软件信息

|  |  |
| --- | --- |
| 中文全称 | 插件最大值编辑器 |
| 英文全称 | DrillLengthEditor |
| 标识号 | LE |
| 编写时间 | 2020年8月 - 2020年9月 |
| 软件版本 | v1.10（编制文档时） |
| 软件类型 | 单机版应用软件 |

2）源码信息

|  |  |
| --- | --- |
| 代码文件 | 39个文件（编制文档时统计） |
| 代码行数 | 1250行（不精确统计，映射57个文件，其中包含了自动生成的头文件行数） |
| 编程语言 | C++ |
| 编程框架 | Qt 5.5 |
| 编程工具 | vs2013 |
| 其他说明 | 使用上述组合，能够将程序打包成32位和64位程序，32位程序能够在xp系统上运行。  （qt5.5是最高支持32位编译的版本，之后qt版本都不再支持32位，也就是说xp系统没法运行了。） |

### 作者提醒

这是一篇口语化的说明文档，非正式软件文档，理解难度较高。

|  |
| --- |
| 这篇文档很多细节会很难理解，因为都是想到哪写到哪。  因为很多点太难详细说明了，都是靠编程习惯和经验得出的 伪代码和知识碎片。 |

插件最大值编辑器 是一个试验性质的小工具。

|  |
| --- |
| 后续作者我会写更多的编辑器等工具，随时可能与此工具的功能重叠。  虽然该编辑器的代码量并不大，但是要全理解、复用内容，似乎不太合适。 |

该文档的主要目的是归纳代码。

|  |
| --- |
| 后期的编辑器代码量会成倍增加，  所以这里只是对模块的一个基本分类与梳理。 |

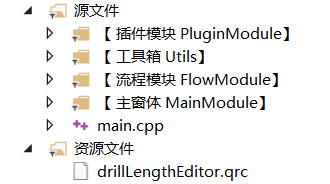
该编辑器有部分C++的工具，但比js插件难用，需要仔细看说明。

|  |
| --- |
| C++的代码封装程度比较高，  由于每个接口提供的功能都非常唯一，这就使得.h头文件会有大量函数提供。  建议先看.h头文件结构，再看cpp实现过程。 |

## 插件最大值编辑器

### 模块结构

编辑器整体分为4个模块，如下图。

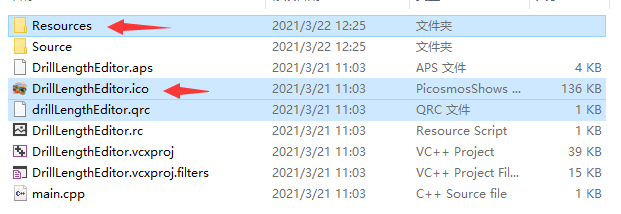


|  |  |  |
| --- | --- | --- |
| 模块 | 耦合度 | 备注 |
| 主窗体 | 极高 | 主窗体的ui界面。 |
| 流程模块 | 极高 | 主窗体操作的流程部分，被封装成固定的数据函数。 |
| 插件模块 | 低 | 当前工程只有 最大值编辑器工具类。 |
| 工具箱 | 低 | 常用的工具类。 |

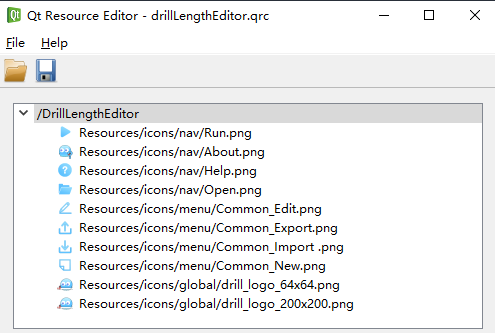
### 资源文件

所有qrc相关资源文件，都在Resources文件夹下。

（ico文件是exe的图标文件，不属于qrc资源文件）



其中放入的都是原创png图片，都符合drill图标的主题色（个人强迫症）



色卡如下：

白色：#ffffff 蓝深色：#449ed4

深灰：#525252 蓝中色：#5fc2ff

浅灰：#a6a6a6 蓝浅色：#8ad8ff

黑色：#000000

颜色在上述色卡之间徘徊（取色可以看出来），部分图标有渐变色。

另外：

图标完全打开一定有像素锯齿。

其实这个主题色，就是从我自己logo上衍生而来的。ヽ(\*。>Д<)o゜

### 流程

流程是程序内部无法改变的固定业务逻辑结构，如下图所示。

（C++的流程 复杂度比 菜单插件的流程 更复杂且庞大，这里就不截图介绍了。）



## 主窗体（MainModule）

**1）界面**

流程模块和主窗体 对应同一个功能中的 数据操作 与 ui操作；

这里特地将 数据与ui 二者分开，以免理解困难。（防止作者我自己理解困难）



**2）流程交互**

单文件操作 和 批量文件操作 被转移到了流程模块中。

其他简单按钮比如 文档按钮、关于作者按钮。直接现生成现用即可。

## 流程模块（FlowModule）

**1）封装数据函数**

封装数据函数的一个好处就是，封装后，就不需要考虑里面的执行方式了，只要看看.h头文件的注释说明即可理解使用。

简单来说就是将其变得“插件化”。

**2）单文件流程**

这部分看代码就行了，没什么必要说明的。

**3）批量文件流程**

这里也没有特别要注意的地方。

数据id通过setToolTip，存在了列表中。（你也可以用setData函数存到 项 中）

用户点击编辑时，根据藏在具体一项的id，来生成填写的表单。（表单的生成依靠的是工具箱的 快速表单类。）

## 插件模块（PluginModule）

插件模块具体指所有与rmmv/js路径下对js脚本进行相关操作的模块。

由于这里只有最大值编辑部分，其它功能用不上，所以留个索引。

### 最大值编辑器（读取器）

**1）使用说明**

最大值编辑器就是一个打包好的完整工具，看说明调用函数即可。

（里面的正则可能并不是一个很好的学习对象，毕竟比较混乱）

目前未发现会出现读取出错的特殊情况。

**2）读取结构**

读取器读取以下注解结构：

|  |
| --- |
| 只读一次：  @plugindesc  @author  @Drill\_LE\_editForbidden  多次分组读取：  @Drill\_LE\_param  @Drill\_LE\_parentKey  @Drill\_LE\_var |

**3）读取数据**

你需要注意下面两个参数：

varLen：代码中读取到的 ”xxxLength = 20；” 这个20就是正则读到的值。

realLen：注释中遍历读取到 ”xxx-21” ”xxx-22”等，遍历读到的最大值。

**4）注释规范**

详细的内容在”关于插件最大值编辑器.docx” 脚本开发 章节中有说明。

定义的注释规范如下：

|  |
| --- |
| 变量键（@Drill\_LE\_param）：  变量名的标识，必填项，  格式必须要有”%d”，”-”用来划分名称。  编辑器将根据-%d，生成”-1”、”-2”、”-1000”任意数量的参数注释。  变量组键（@Drill\_LE\_parentKey）：  变量分组的标识，  格式必须含有两个”%d”。  也可以为空字符串，空字符串表示不分组。  编辑器将根据%d，生成” 1和20”“21和40”的分组格式，固定20个为一组。  定义名（@Drill\_LE\_var）：  变量键数量的定义名。  格式必须为一个有效的变量名称。  也可以为”null”，表示没有相关定义变量。  编辑器将根据改变的最大值，将” xxxx = 10; ” 改成 ” xxxx = 最大值; ”。  三个注释必须依次按顺序写，可以写多个表示多个参数。  @Drill\_LE\_param ”xxxxxx-%d”  @Drill\_LE\_parentKey ”xxxx%dxxx%dxxxx”  @Drill\_LE\_var ”xxxxx”    禁用编辑（@Drill\_LE\_editForbidden）：  如果插件出现了这一条注释，将会禁止编辑器编辑最大值。  当你的插件出现”-1”，”-2”，”-3”，”-4”时，又不想让编辑器编辑插件中任何一个变量时添加。 |

### 最大值编辑器（修改器）

**1）使用说明**

注意，在执行doOverwritePlugin函数时，C\_LEConfigData的realLen必须要有值。（程序需要先通过realLen删注释，再新建加回去）

也就是说，单用最大值的修改器，是不行的，必须先读完插件，再进行覆写。

**2）生成方式**



## 工具箱（Utils）

工具箱中是一套完整的可拆解或组合的C++工具插件，这里对几个主要工具进行说明。

### 通用工具

TTool.h和TTool.cpp是作者我常用的静态函数集合。

TTool意为tinytool，这个名字因为有两个T和两个o，名字不正经，所以一眼就能找到，而且能感觉到是作者写的工具类，而不是什么专业的业务逻辑类。

该集合更新很频繁，因为只要在写代码的时候，有一两个函数反反复复出现，我就会把这个函数扔到TTool里面。简化操作。

### 文本解析

常用类，打开长篇文章，使用正则，对每行进行 d\_indexOf ，找出关键行。

|  |
| --- |
| 关于文本解析，你最好根据自己的需求，造一个自用的正则表达式匹配的封装类。  因为现成的读取器都比较难懂，都是用的seeker光标，是增量读取的（读一行解析一行），之前尝试过在增量中获取数据，结果某一行要前面几行的数据……就手足无措了。  相比而言，全部读取后，再对每行作定量划分，容易理解代码原理和用法。  有时候就按行划分字符串都能弄死你，因为可能：存在空行、\r表示换行、\r\n表示换行、\n\r表示换行，这些复杂情况必须统一工具接口，把所有情况考虑到，才能实现完整读取数据。 |

这里分为两个工具，一个是直接扫描行的读取器，另一个是扫描行并且记录你用indexof捕获过的行。

后者的存在主要是为了校验我写的正则有没有捕获到具体的行。

之前写了一个文本比较窗口，左边是原文本，右边是含置灰捕获标记的文本。

不过应用场景实在不大，等以后用到了，再详细介绍吧。

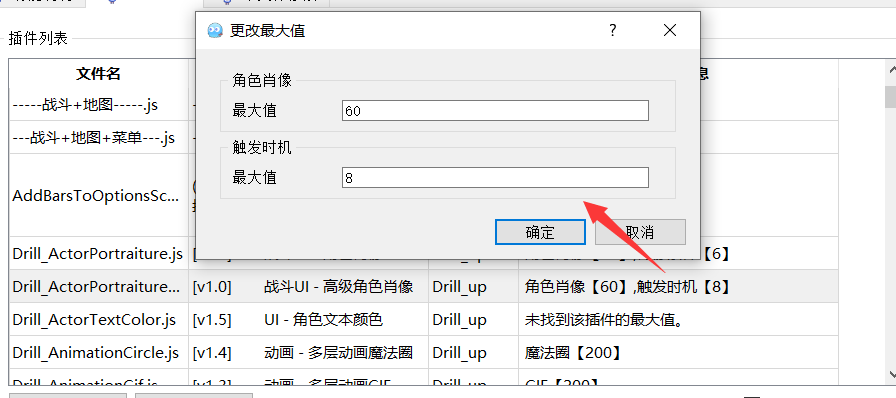
### 快速表单

快速表单的功能是：

输入一大堆预置文本，然后转成数据表类；

系统根据数据表再转成ui输入框，并能从输入框中打包成实际数据存储。

没错，这个功能就是对标rmmv通过注释@param就可以定义一个填写表单的功能模仿出来的。（毕竟这个数据字典的功能太实用了）



事实上我在工作期间，就反复琢磨、研究实现了这个类，2020年6月到9月左右，经过反复推敲、捶打、优化、测试，终于有了成形的结构了。

快速表单在流程模块中，函数“editOneWithWindow打开窗口编辑插件”有应用。

根据扫描到的插件变量名，自动生成最大值修改表单。

表单存储的数据是以中文存储的，比如图中存储的数据就会为：

{“Drill\_ActorPortraitureExtend角色肖像最大值”:60 , “Drill\_ActorPortraitureExtend触发时机最大值”:8}

如果你导出了drillle文件，右键用文本打开可以看到类似上述的json数据格式。

这样在下次导入文件时，会根据相关的 插件名+变量 进行键值匹配。

## 其他说明

### 注解缺陷

最大值编辑器，本质上是对插件进行文本修改替换。

替换的过程中，正则表达式非常容易遗漏部分特殊的写法或者公式。

而这类问题，只能通过反复测试才能找出，暂时没有其他方法。

毕竟注解的灵活性太大了，没有严格固定的标准。

使用特例匹配查找，就像是在操作一个不稳定的黑盒子数据一样，脚本程序员多打一个空格一个tab键，都可能存在匹配不到的问题。