

Advanced Programming*

课程设计报告

ℒ_Tℒ_X

* Teacher: Shujian Huang. TA: Yiyu Zhang

1st 张逸凯 171840708

Department of Computer Science and Technology

Nanjing University

zykhelloha@gmail.com

摘要

本文是171840708 张逸凯高级程序设计(*Advanced Programming*)第一次课程设计报告.

本次课程设计实现了基于面向对象思想的一个简易Markdown编辑器, 可解析并生成HTML格式的输出.

本次课设思路清晰, 构建了基于位置优先的多层次分治算法模型, 对Markdown语法转换成HTML格式较鲁棒, 特别对嵌套的子列表采用了递归树解析的方法, 此外本课设不仅完成了PDF上的基本需求, 还进行了全面的拓展, 比如参考Markdown标准语法对语法进行扩充(包含5项), 在GUI版本可载入文件、自由编辑(修改多行内容), 而且可输出带有CSS样式的HTML文件.

*谢谢老师和助教哥的耐心批改.

目录

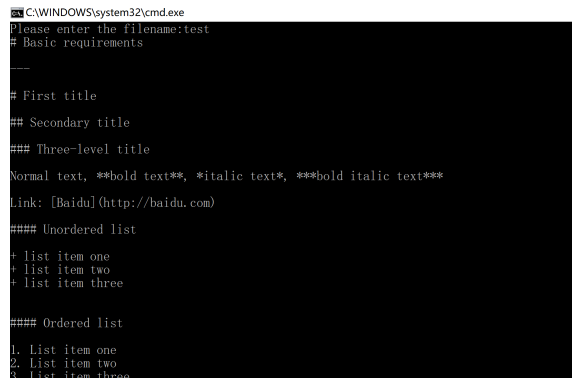
I	主要内容、目标	3
I-A	基本内容	3
I-A1	可以从控制台录入Markdown格式的文本(只录入文本即可, 无自由编辑功能), 并保存生成 .md 文件	3
I-A2	载入生成的 .md 文件, 根据以上定义的Markdown语法子集进行解析, 生成HTML文件 (保证可以通过浏览器打开, 正常展示)	3
I-A3	对于生成的HTML文件, 无附带CSS样式, 只生成基本的标签即可	3
I-A4	语法规则, Markdown文本, HTML标签等应合理表示	4
I-B	拓展部分实现内容	4
I-B1	CSS样式	4
I-B2	附加语法(5项)	4
I-B3	GUI界面(自由编辑)	6
II	设计思路	7
II-A	如何解析复杂的嵌套句子	7
II-B	如何处理匹配标记的先后顺序	8
II-C	如何解析嵌套列表	9
III	主要类的设计、类的数据和操作、类之间的关系	11
IV	附加内容提要: 拓展语法与GUI界面	12
V	如何运行代码	12
V-A	控制台版本	12
V-B	图形界面版本	13
VI	总结	14

I. 主要内容、目标

A. 基本内容

基本内容由控制台版本实现, 代码对应附件: **console_version** 文件夹

1) 可以从控制台录入Markdown格式的文本(只录入文本即可, 无自由编辑功能), 并保存生成 *.md* 文件: .



```
C:\WINDOWS\system32\cmd.exe
Please enter the filename:test
# Basic requirements

---

# First title
## Secondary title
### Three-level title

Normal text, bold text, italic text, bold italic text

Link: [Baidu] (http://baidu.com)

#### Unordered list
+ list item one
+ list item two
+ list item three

#### Ordered list
1. List item one
2. List item two
3. List item three
```

图 1. 控制台录入Markdown格式运行截图

2) 载入生成的 *.md* 文件, 根据以上定义的Markdown语法子集进行解析, 生成HTML文件 (保证可以通过浏览器打开, 正常展示): .

3) 对于生成的HTML文件, 无附带CSS样式, 只生成基本的标签即可: .

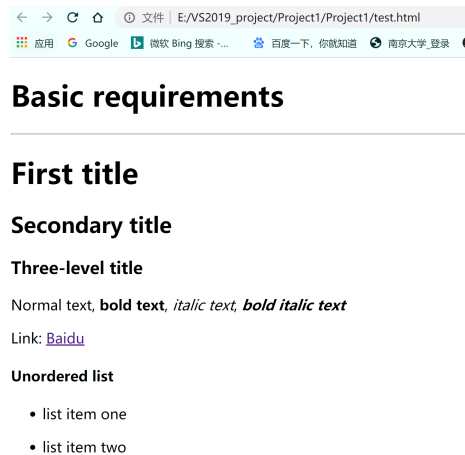


图 2. 程序运行后生成的未带CSS样式生成的HTML文件(Chrome打开)

4) 语法规则，*Markdown*文本，*HTML*标签等应合理表示: .

```
5 robust nested lists

look!

• list item 1
  1. sublist item 1
    1. subsublist item 1
    2. subsublist item 2
      1. subsubsublist item 1
      2. subsubsublist item 2
      3. subsubsublist item 3
    3. sublist item 2
  2. sublist item 2
• list item 2
  ◦ sublist item 1
    ▪ subsublist item 1
    ▪ subsublist item 2
      ▪ subsubsublist item 1
      ▪ subsubsublist item 2
    ▪ subsublist item 3
  ◦ sublist item 2
• list item 3
```

图 3. 深层嵌套list(接下来会详述)

B. 拓展部分实现内容

1) CSS样式: .

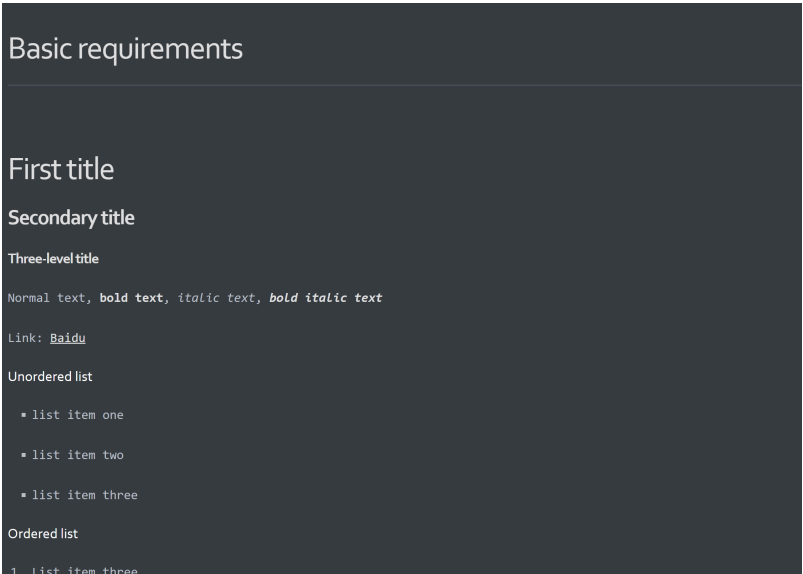


图 4. CSS样式(以Night.css为例)

2) 附加语法(5项): .

1. 代码块:

1 Enhanced code block

```
// c++ code:

int main() {
    for (int i = 0; i < 10; ++i) {
        std::cout << i << std::endl;
    }

    return 0;
}
```

图 5. 附加语法: 代码块

2. 行内代码块:

2 code environment in one line

```
CODE in the text: printf("Hello World"); NOTE: The previous one was the code format.
```

图 6. 附加语法: 行内代码块

3. 分割线:

3 cut-off rule

图 7. 附加语法: 分割线

4. 引用语句块:

4 quote environment

Hello!

Here is the quote.

Bye!

图 8. 附加语法: 引用语句块

5. 深层嵌套列表:

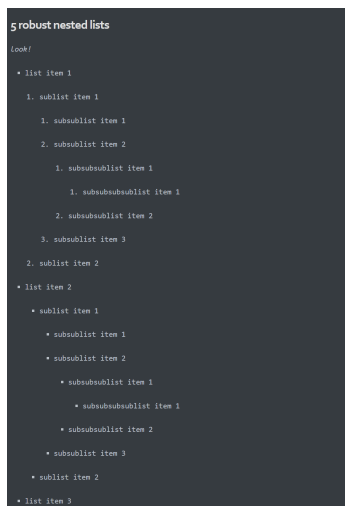


图 9. 深层嵌套列表

3) GUI界面(自由编辑): .

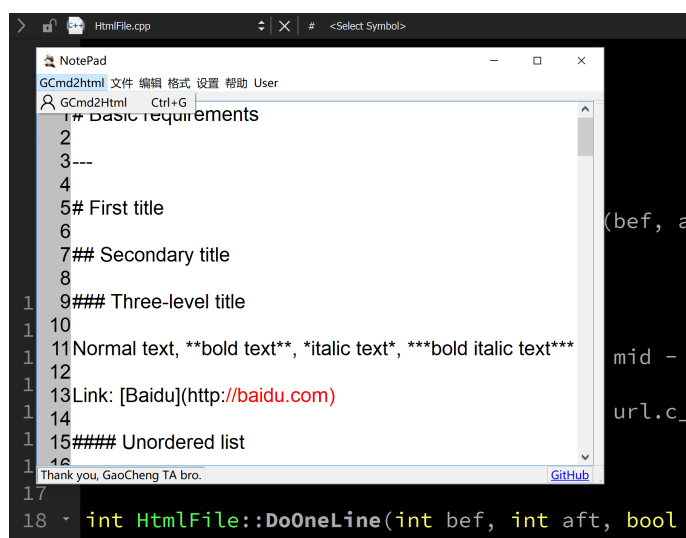


图 10. GUI

实现了基于Qt5的GUI界面, 可以自由编辑, 带来良好的用户交互体验,

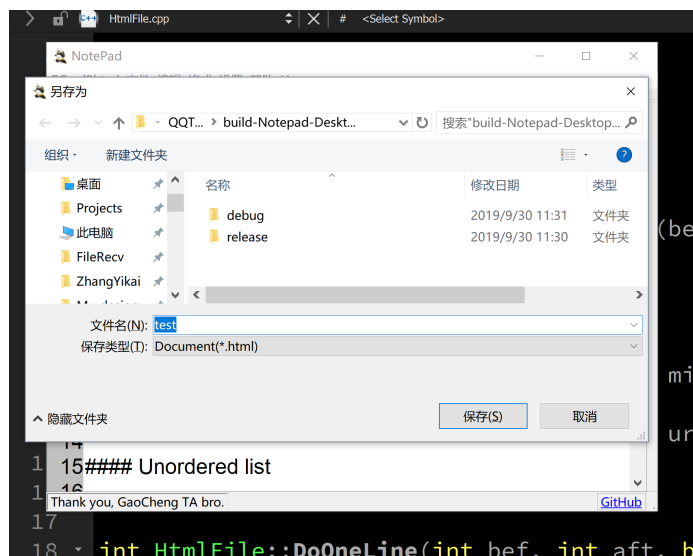


图 11. 点击GCmd2html后截图

图10中下拉的菜单选项GCmd2html(或者按快捷键ctrl + G)就是GC(gaocheng), markdown文件转HTML的选项, 点击后可以命名想要保存的文件名字, 当前编辑的内容即可转成HTML格式, 并保存在指定的文件夹下, 如图11所示.

II. 设计思路

本节介绍了基于面向对象思想, markdown文本解析并转换成HTML的设计思路. 详述了本次课设使用的核心方法: 基于位置优先的多层次分治算法模型.

首先学习HTML语言, 我们发现本次课设需要的可分为两种形式, 一是双端标记, 比如斜体: **italic**, 加粗 ****bold****; 二是单端标记, 比如标题 # 一级标题等.

A. 如何解析复杂的嵌套句子

因为看到测试用例涉及复杂的嵌套Markdown解析, 并不可以暴力地把所有可能都列出来, 而且我们仔细分析发现暴力地列出所有可能情况, 强行地采用规则匹配, 对于嵌套的复杂格式处理能力很差, 这样看起来问题比较复杂, 如何设计一套对各种情况都能合法处理的自动化解析方式呢? 由嵌套, 首先想到的设计方法就是分治, “分而治之”, 就是把一个复杂的问题分成两个或更多的相同或相似的子问题, 再把子问题分成更小的子问题, 直到最后子问题可以简单的直接求解.

从最简单的一个分治例子开始: 我们看如下简单嵌套Markdown语句递归树构成, 展示了一个Markdown应该以怎样的递归思路进行展开.

```
"##### Title Link List Item [Computer System Homepage](http://cs.nju.edu.cn)"
```

```
<h5> + "Title Link List Item [Computer System Homepage](http://cs.nju.edu.cn)" + </h5>
```

```
<h5>Title Link List Item + "[Computer System Homepage](http://cs.nju.edu.cn)" + </h5>
```

```
<h5>Title Link List Item <a href='http://cs.nju.edu.cn'>Computer System Homepage</a></h5>
```

B. 如何处理匹配标记的先后顺序

很容易我们就发现了一个问题,不妨称每种标记格式(比如加粗,斜体等)为一种格式环境,当两个格式环境相互嵌套的时候,或者出现多个环境同时并列的时候,该如何进行匹配确定先后匹配顺序呢?本次课设提出了基于位置优先的方法,在每个Markdown语句中的匹配格式环境标记中,优先匹配位置靠前的单端标记(比如标题等),单端标记在分析后是可以被证明具有更高的优先级的,然后再根据双端标记的`pair.first`优先级顺序进行解析与转换,总是优先解析并匹配双端标记的第一个位置更前面的那个标记。

解析之后如果该解析的格式环境里面有其他嵌套的格式环境,则像上图一样递归处理,如果该解析的格式环境里有与其他格式环境产生交集的标记,则所有存在交集的格式环境消亡。这样就可以保证算法是鲁棒的。

对于某些错误的Markdown输入,比如*这是含有交集的**两个*格式环境**,像这样加粗环境和斜体环境产生了交集“两个”,但是基于位置优先的分治算法模型不会出错,不会存在HTML格式内出现环境交叉的情况。

综上所述,我们依靠基于位置优先的分治算法已经可以对任何行进行处理,但是在学习HTML后发现,列表的格式环境是跨行的,简单的行处理行不通。注意,无序标记(或者有序标签)将与标签一起使用,但是又有差别,下面就是一个典型的例子。

```
<ul>
<li>list item 1</li>
<li>list item 2</li>
</ul>
```

每一个列表的item项需要(每一个嵌套列表的周围也要包含),其实在列表的每一个层次递进中,和都会被打上标记,但是结束的时候,是子列表要包含,同级列表不用,而是子列表和同级列表都需要包含。这一点细微的差异怎么控制呢,这就是算法模型的多层次的体现了,算法通过一系列层次性的标志物,保证了递归函数在回溯的时候都可以稳定地判断是否需要打或者的标记,保证了HTML语法的正确性。

C. 如何解析嵌套列表

5 robust nested lists

look!

- list item 1
 - 1. sublist item 1
 - 1. subsublist item 1
 - 2. subsublist item 2
 - 1. subsubsublist item 1
 - 1. subsubsubsublist item 1
 - 2. subsubsubsublist item 2
 - 3. subsubsubsublist item 3
- 2. sublist item 2

图 12. 复杂的嵌套列表

但是遇到嵌套列表该怎么做呢? 学习HTML之后发现嵌套列表其实是一个递归树的形式, 下面这个例子将展示这一思想(这是加强的测试用例, 很形象地展示了多层次算法模型的本质):

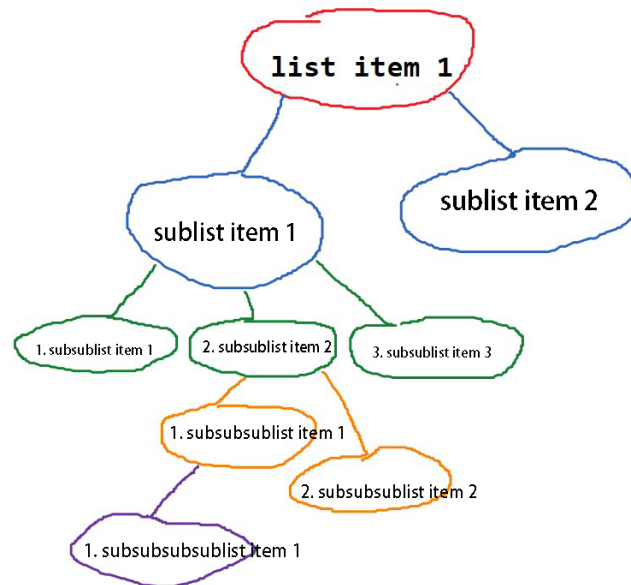


图 13. 复杂的嵌套列表转化成递归树的形式

其实算法模型对列表的递归树做了一次先序遍历, 这样我们就把HTML语法转换成了我们容易理解的算法形式, 发现了解析一种语言的本质, 当然因为我还没学编译原理, 可能还有更厉害的方法, 但是这样多层次递归的解析, 已经是结构清晰明了, 效果稳定的方法了呢!

算法伪代码如下(L^AT_EX打得好辛苦!), 其实具体的实现, 回溯细节都是非常复杂和严谨, 这个算法模型结构简单, 但是具体实现可不一般.

Algorithm 1: 基于位置优先的多层次分治算法模型

Data: Markdown text
Result: HTML text

```
1 initialization;  
2 function DoFunc(line): ;  
3 foreach line of Input do  
4     // 优先定位 ;  
5     foreach A and B in line && A and B is TAG do  
6         something to be processed ;  
7         priority queue.push(make pair(A, B)) ;  
8     end  
9     While priority queue is not empty { ;  
10    q.push(TAG A and B);    // 识别标记, 找到解析段 ;  
11    if line is Title n then  
12        Out ← < hn > ;  
13        DoFunc(rest of line(without TITLE TAG)) ;    // 递归 ;  
14        Out ← < /hn > ;  
15    end  
16    if Other environment case then  
17        Out ← Tag one ;  
18        DoFunc(rest of line);    // 递归 ;  
19        Out ← Tag two ;  
20    end  
21 }  
22 end
```

III. 主要类的设计、类的数据和操作、类之间的关系

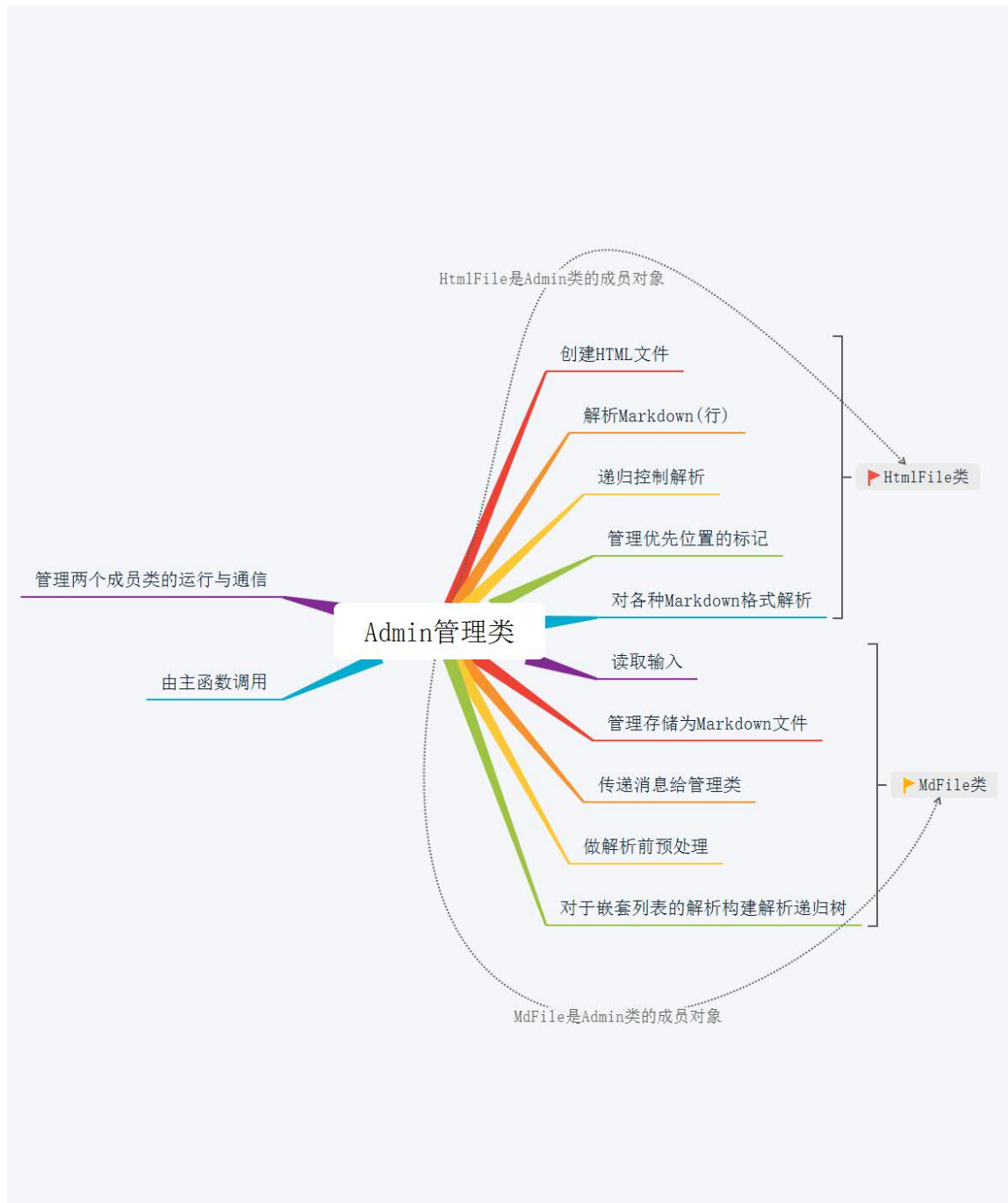


图 14. 类的设计与类间关系

设置了一个类似操作系统的调度类, 这样程序的可扩展行就会变好, 也有利于不同类之间耦合性降低. 对于Markdown和HTML设置了两个类, 一个类的功能单一, 内聚高, 一个类在扩展性方面是开放的而在更改性方面是封闭的.

类之间的数据传递是通过封装好的接口实现的, 保证了接收方不能修改数据, 数据重复性冗余度也尽量降低, 数据安全性也得到加强. 充分基于面向对象的特性.

IV. 附加内容提要: 拓展语法与GUI界面

拓展的5个语法(代码块, 文本行内代码, 分割线, CSS样式, 引用块)也是同样应用基于位置优先的多层次分治算法模型实现的, 具有良好的鲁棒性.

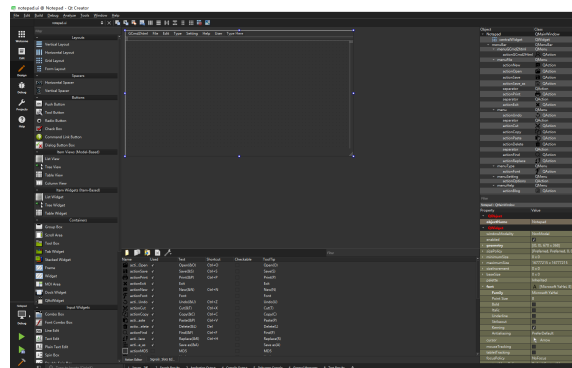


图 15. Qt5 实现自由编辑.ui文件

GUI界面由Qt5实现, 具有良好的用户界面, 自由编辑, 内容可一键导出HTML文件.

V. 如何运行代码

可执行文件的运行这里不再赘述, 主要讲如何运行源代码.

A. 控制台版本

常规操作, Visual Studio编译运行即可.

界面出现, 我们首先输入要写入的文件名:

接着输入md格式即可.

```
int main() {  
    C:\WINDOWS\system32\cmd.exe  
    Please enter the filename:test  
    # Basic requirements  
    ---  
    # First title
```

ctrl + z结束输入
输入是否需要CSS样式:

```
---  
**Thank you for your patience in correcting. Please refer to the experiment report for details about the assignment.**  
*Thank you, TA bro.*  
^Z  
Do you need CSS styles? (1: Yes!, 0: No, thanks) :1  
请按任意键继续...
```

B. 图形界面版本

请使用Qt5.10, Qt creator 4.5.0, Qt 5.10 MinGW 及以上版本进行编译.

```
1 # Basic requirements  
2  
3 ---  
4 # First title  
5  
6  
7 ## Secondary title  
8  
9 ### Three-level title  
10  
11 Normal text, ""bold text", "italic text", ""bold italic text""  
12  
13 Link: (Baidu)(http://baidu.com)  
14  
15 ##### Unordered list  
16  
17 + list item one  
18 + list item two  
19 + list item three  
20  
21  
22 ##### Ordered list  
23  
24 1. List item one  
25 2. List item two  
26 3. List item three  
27  
28  
29 # Enhanced hierarchical list (robust performance)  
30  
31 ##### Nested list
```

图 16. GUI界面可自由编辑

可以导入文件或者直接自由编辑, 按Ctrl + G或者第一个选项GCmd2html:



按步骤保存文件即可.

如果不能复现结果请联系我, 或者直接编译运行源代码, 电话: 18051988316, QQ: 645064582

	已经提交
	未评分
	2019年10月13日 星期日 23:55
	2019年10月1日 星期二 09:55
	 171840708_张逸航_高程课设一.zip

VI. 总结

总计大约50小时完成了本次课设, 花了很多心思, 希望能得到高分, 谢谢助教哥的批改.

10月13日做了一个code review, 还发现了一点小问题. 现在基本OK啦!

祝好~