Edit: March 21 2023

Caligrafo
Planning Document

GitHub.com/Zvorky/Caligrafo

The creative process to rise up the project.

Enzo Zavorski Delevatti
@Zvorky

March 10 2023

The Idea

Work with alignments for texts, essentially when printing at terminal.

The same functions to align and stylize the printed text can be used to write into text files.

It can support lists, tables, and some others text structures in the future…

I'd like to use many blocks of text in the same page, maybe with overlaps, it would be cool…

The goal is to turn simple text formatting easier, allowing us to make more fancy CLIs and text files quickly.

The Code

Object Oriented, the Caligrafo can be a class itself, who makes the actions, or…

The formatted text as a class, that can be converted from/into *str*.

Or maybe both…

With the Caligrafo being a class itself I mean the text would be the simple *str* yet, but with the formatting applied.

But with the formatted text as a class, we have more versatility to use the functions one just before other, and still can get the simple text.

Enzo Zavorski Delevatti
@Zvorky

March 10 2023

Basic Utilities

`The initial goal`

- Paragraph spacing
- Horizontal Alignment (Left, Center, Right, Justified)
- Vertical Alignment (Top, Center, Bottom)
- Border Spacing
- Max Width
- Max Height

General Requirements

`A basic requirement list`

- Any string must be able to be converted into formatted text
- Any formatted text must be able to export its basic string
- The formatted text must be able to convert into string, applying the formatting
- The '*page*' size must be able to be passed by parameter
- The '*page*' size must be automatically set with the terminal size when possible and not passed by parameter
- The '*page*' size must have a default value for non-terminal use

Enzo Zavorski Delevatti
@Zvorky

Edit: March 14 2023

Data Structure

It needs to have a *str* with the simple text, and be able to map the formatting…

'*Page*' size is the same for a group of texts, but it can't be global so we can work in a file and terminal at same time, for example.

To be able to make combinations of texts, or set different formatting in different points of the same text, it will need to use lists.

It is a nice idea to make it self-describing, so let set a *class* for spacing instead of a *list* :^)

…and use *str* instead of *char* for alignments too

- Spacing:

Type	Name
int	paragraph
int	left
int	right
int	top
int	bottom

- Formatted Text:

Type	Name	Description
str	text	- simple text
int	width	- max width (0 = limitless)
int	height	- max height (0 = limitless)
Spacing	spacing	- *int* spacing amounts
str	horizontal_alignment	- Left, Center, Right, Justified
str	vertical_alignment	- Top, Center, Bottom

Enzo Zavorski Delevatti
@Zvorky

March 21 2023

Text Box to String

The conversion routine

Non-limited boxes:

Order	Operation
0	Start with Paragraph
1	New Line into Paragraph

Width limited boxes:

Order	Operation
0	Start with Paragraph
1	New Line into Paragraph
2	Width Limit into New Line

Height limited boxes:

Order	Operation
0	Start with Paragraph
1	If Height Limit, New Line into “[…]"
2	New Line into Paragraph

Width and Height limited boxes:

Order	Operation
0	Start with Paragraph
1	If Height Limit, New Line into “[…]"
2	New Line into Paragraph
3	Width Limit into New Line

Need to track the letter, line and column indexes.

Enzo Zavorski Delevatti
@Zvorky