## CONTENTS:

1 Coordinate Canvas  
1.1 Installation ............................................................... 1  
1.1.1 Install the development version .................................. 1  
1.2 Configuration ............................................................. 1  
1.3 Execution ................................................................. 2  
1.3.1 Built-in prompts ....................................................... 2  
1.3.2 Command Line Interface ............................................. 2  
1.4 Usage ....................................................................... 2  
1.4.1 Data retrieval .......................................................... 4

2 core  
2.1 coordinate_canvas package .............................................. 5  
2.1.1 Module contents ....................................................... 5  
2.2 config module ............................................................. 5  
2.3 input_handler module .................................................... 5  
2.4 line_builder module ...................................................... 6

3 Indices and tables  
Python Module Index ......................................................... 11  
Index .............................................................................. 13
CHAPTER ONE

COORDINATE CANVAS

Interactive canvas that allows you to draw 2D coordinates in a plane and output their corresponding coordinates to a JSON file.

1.1 Installation

Instructions below assume that your Python interpreter is linked to the `python` alias:

```
python -m pip install coordinate-canvas
```

1.1.1 Install the development version

In case you’d like to be able to update the package code occasionally with the latest bug fixes and improvements, see the source code, or even make your own changes, you can always clone the code directly from the repository:

```
git clone https://github.com/erlete/coordinate-canvas.git
cd coordinate-canvas
pip install -e .
```

1.2 Configuration

The package allows customization of three parameters:

- **Width (integer or float)**: width of the canvas.
- **Height (integer or float)**: height of the canvas.
- **Number of lines (integer)**: amount of lines to draw.

Said parameters are configured upon execution.
1.3 Execution

Currently, the package supports two separate execution modes. The first one uses Python built-in prompts, while the other allows parameters to be passed via command line interface:

1.3.1 Built-in prompts

```python
python -m coordinate-canvas
>>> Width: <width of the canvas>
>>> Height: <height of the canvas>
>>> Number of lines: <amount of lines to draw>
```

After all prompts have been filled accordingly, the canvas will pop up in a separate window.

1.3.2 Command Line Interface

```python
python -m coordinate-canvas <width of the canvas> <height of the canvas> <amount of lines to draw>
```

Again, if all fields have been filled properly, the canvas will open in another window.

1.4 Usage

Once the canvas has been opened, you will be able to click on any part of it and add a new coordinate. Lines can be switched using the numeric pad on the keyboard, as explained on the header of the window.
Once you have added all the desired coordinates, just press the “Escape” key or close the window. A coordinates.json file will be generated in the same directory where the program was executed. The structure of the JSON file will be as follows:

```json
{
    "line_1": { 
        "x": [ ...
```
1.4.1 Data retrieval

JSON data can easily be retrieved with a few lines of code.

```python
import json

# Assuming the script is located in the same directory where the program is
# being executed (if not, modify the path below):
with open("coordinates.json", mode="r", encoding="utf-8") as fp:
    data = json.load(fp)  # Loads all data in a dictionary.

line_1 = data["line_1"]
line_1_x = data["line_1"]["x"]
line_1_y = data["line_1"]["y"]
```
2.1 coordinate_canvas package

2.1.1 Module contents

2.2 config module

Configuration module.
This module contains several configuration constants used by the program.

Author:
Paulo Sanchez (@erlete)

2.3 input_handler module

Input handling module.
This module contains all methods needed to handle the input of the package. It is mean to be used internally by the package and not by the user, since method implementations are highly specific to the package’s needs.

Author:
Paulo Sanchez (@erlete)

\textbf{output_format} (values: \text{tuple[str]}) \rightarrow \text{tuple[float, float, int]} | \text{None}

Format the output of the input.
This method receives a tuple of strings and validates them. If the amount of values is less than 3, then a ValueError is raised. If any of the values is not a string, then a TypeError is raised. Otherwise, the formatted input values are returned.

The method is meant to be used in combination with the \textit{cli_input} and the \textit{python_input} methods.

\textbf{Parameters}
\begin{itemize}
  \item \textbf{values} (tuple) – input values.
\end{itemize}

\textbf{Raises}
\begin{itemize}
  \item \textbf{ValueError} – If the input does not have at least 3 values.
  \item \textbf{TypeError} – If any of the input values is not a string.
\end{itemize}

\textbf{Returns}
The formatted input values.
Return type
tuple[float, float, int]

cli_input(\texttt{arguments: list[str]} \rightarrow \text{tuple[str, str] | None})

Handle CLI input.

This class receives a list of arguments and validates them. If the amount of arguments is less than 4, then None is returned. If any of the arguments is not numeric, then None is returned. Otherwise, the raw input values are returned.

\textbf{Parameters}
- arguments (\texttt{list[str]}) – list of arguments from \texttt{sys.argv}.

\textbf{Returns}
The raw input values or None.

\textbf{Return type}
tuple[str, str, str] | None

\textbf{python_input}(\texttt{message: str}) \rightarrow \text{str}

Handle Python input.

This class receives a message and validates the input. If the input is not numeric, then a ValueError is raised. Otherwise, the input is returned.

\textbf{Parameters}
- message (\texttt{str}) – message to be displayed to the user.

\textbf{Raises}
- ValueError – if the input is not numeric.

\textbf{Returns}
validated input or None.

\textbf{Return type}
value (str)

2.4 line_builder module

LineBuilder container module.

This module contains the LineBuilder class, which is used to build a line based on click events. The line is built by clicking on the matplotlib plot.

\textbf{Author:}
- Paulo Sanchez (@erlete)

\textbf{class LineBuilder}(\texttt{line: Line2D, ax: Axes, width: float, height: float, color: str})

\textbf{Bases: object}

Line builder on click events.

This class is used to build a line based on click events. The line is built by clicking on the matplotlib plot. The coordinates are stored in a list of tuples.

\textbf{line}
line to be built.

\textbf{Type}
\texttt{matplotlib.lines.Line2D}
ax
    axes where the line is drawn.
    
    Type
    matplotlib.axes.Axes

width
    width of the plot.

    Type
    float

height
    height of the plot.

    Type
    float

color
    color of the line.

    Type
    str

is_connected() → bool
    Check if the line builder is connected to the matplotlib plot.

    Parameters
    cid (bool) – True if the line builder is connected, False otherwise.

connect() → None
    Connect the line builder to the matplotlib plot.

disconnect() → None
    Disconnect the line builder from the matplotlib plot.
CHAPTER
THREE

INDICES AND TABLES

• genindex
• modindex
• search
C

coordinate_canvas, 5
coordinate_canvas.core.config, 5
coordinate_canvas.core.input_handler, 5
coordinate_canvas.core.line_builder, 6
null
INDEX

A
ax (LineBuilder attribute), 6

C
cli_input() (in module coordinate_canvas.core.input_handler), 6
color (LineBuilder attribute), 7
connect() (LineBuilder method), 7
coordinate_canvas module, 5
coordinate_canvas.core.config module, 5
coordinate_canvas.core.input_handler module, 5
coordinate_canvas.core.line_builder module, 6

D
disconnect() (LineBuilder method), 7

H
height (LineBuilder attribute), 7

I
is_connected() (LineBuilder method), 7

L
line (LineBuilder attribute), 6
LineBuilder (class in coordinate_canvas.core.line_builder), 6

M
module
  coordinate_canvas, 5
  coordinate_canvas.core.config, 5
  coordinate_canvas.core.input_handler, 5
  coordinate_canvas.core.line_builder, 6

O
output_format() (in module coordinate_canvas.core.input_handler), 5

P
python_input() (in module coordinate_canvas.core.input_handler), 6

W
width (LineBuilder attribute), 7