Chronicler for CK3

Jacob Hilker

Table of Contents

1. Chronicler for Crusader Kings 3

Chronicler (for CK3) is a terminal application for taking notes on a dynasty during a particular campaign of Crusader Kings 3. It is written in Python. To get started right away, see the Quickstart section.

1.1 Quickstart

1.2 Contributing

If you are interested in contributing in any way to the application's codebase, please submit a pull request. If you are interested in contributing to the documentation of the project, please send a message on the Discord or the Slack.

2. Installation

lorem ipsum

3. Configuration

4. Usage

5. Contribute

6. API Reference

6.1 Chronicler

Main TUI implementation for Chronicler.

6.1.1 `ChroniclerApp`

Main TUI implementation for Chronicler.

Attributes:

Name	Type	Description
`__color_map__`	`dict{str, py_cui.color}`	Used to map a string from the config file to the interface.

`__init__(self, screen)` `special`

Used to initialize the application.

Parameters:

Name	Type	Description	Default
`screen`	`py_cui.PyCUI`	The screen the app will be rendered on.	*required*

Source code in `chronicler/__init__.py`

```
def __init__(self, screen):
    """
    Used to initialize the application.

    Args:
        screen (py_cui.PyCUI): The screen the app will be rendered on.
    """
    self.screen = screen
    self.character_list = self.screen.add_scroll_menu('Characters', 0, 0, row_span = 8, column_span = 3, pady = -1)
    self.character_war_info = self.screen.add_block_label('', 0, 3, row_span = 8, column_span = 5, pady = -1, center=False)
    self.widgets = {'character_list': self.character_list, 'character_info': self.character_war_info, 'title_bar': self.screen.title_bar, 'status_bar':
self.screen.status_bar}
    self.screen.set_status_bar_text(' q: Quit Chronicler |  o: Open AAR | H: Print Keybindings')
```

`run(self)`

Used to run Chronicler.

This function is an alias for running the Chronicler application.

Source code in `chronicler/__init__.py`

```
def run(self):
    """
    Used to run Chronicler.

    This function is an alias for running the Chronicler application.
    """
    self.screen.start()
```

6.1.2 `main()`

Run at the command line.

This function is executed when you run `chronicler` at the command line. It sets up the UI, parses the config, and then runs the application.

Source code in `chronicler/__init__.py`

```
def main():
    """
    Run at the command line.

    This function is executed when you run `chronicler` at the command line. It sets up the UI, parses the config, and then runs the application.
    """
    root = pycui.PyCUI(8, 8)
    root.set_title('Chronicler for Crusader Kings 3')
    app =  ChroniclerApp(root)
    #root.add_key_command(pycui.keys.KEY_H_UPPER, app.show_help_menu)
    parser = argparse.ArgumentParser(description='A tool for managing after-action reports for Crusader Kings 3.')
    parser.add_argument('-c', '--config', help='Path to another config file.', action='store', dest='cfg', nargs='?', const="~/.config/chronicler/chronicler.yml")
    args = vars(parser.parse_args())
    #print(args)
    if (args['cfg'] == None):
        parse_config(app)
    else:
        parse_config(app, args['cfg'])
```

6.1.3 `parse_config(app, config_file='/home/docs/.config/chronicler/config.yml')`

Used to parse the config file.

This function parses the configuration file and applies any relevant options to the application.

Parameters:

Name	Type	Description	Default
`app`	`ChroniclerApp`	Application to apply changes to.	*required*
`config_file`	`str`	The path to the configuration file. Defaults to "~/.config/chronicler/config.yml".	`'/home/docs/.config/chronicler/config.yml'`

Exceptions:

Type	Description
`FileNotFoundError`	If the file passed does not exist.

Source code in `chronicler/__init__.py`

```
def parse_config(app, config_file=os.path.expanduser("~/.config/chronicler/config.yml")):
    """
    Used to parse the config file.

    This function parses the configuration file and applies any relevant options to the application.

    Args:
        app (ChroniclerApp): Application to apply changes to.
        config_file (str, optional): The path to the configuration file. Defaults to "~/.config/chronicler/config.yml".

    Raises:
        FileNotFoundError: If the file passed does not exist.
    """
    with open(config_file, 'r') as config_file:
        config = yaml.safe_load(config_file)
        keys = config.keys()


        #print(keys)
        for key in keys:
            try:
                print(key)
            except:
                pass
```

6.2 Assets

Contains the assets for an AAR.

The `chronicler.assets` module contains all the assets and resources for an AAR.

6.2.1 `Character`

`__init__(self)` `special`

Creates a Character.

Source code in `chronicler/assets.py`

```
def __init__(self):
    """
    Creates a Character.
    """
    self.__name = ""
    __culture = ""
    __birth_date = None
    __status = ""
    __age = 0
    __gender = ""
    __rank = ""
    #__homeland = None
    __regions_ruled = []
    #__dynasty = None
    __titles = []
    __traits = []
```

6.3 AAR File Type