🎭 Thalie

v0.13a 2022-12-11

A LaTeX package to typeset drama plays

Louis Paternault

https://framagit.org/spalax/thalie

spalax(at)gresille(dot)org

This package is meant to typeset drama plays using LaTeX. It provides commands to introduce characters' lines, to render stage direction, to divide a play into acts and scenes, to automatically build the dramatis personæ, etc.

Table of Contents

This document introduces the thalie package, used to typeset drama plays.

Part I.
Introduction

1. About thalie

1.1. Why Thalie?

"Thalie" is the French translation of "Thalia" ("Θάλεια" in Greek), the Muse of comedy and idyllic poetry.[1] It seems an appropriate name for a tool dedicated to help writing drama.

1.2. Other classes and packages

This package is far from being the only one that can be used to render drama plays. If you do not like my work, you can use one of the following ones (and I guess there exists others): drama [7], dramatist [8], play [9], screenplay [10], sides [11].

I began to write this package in 2010, and now, at the end of 2012, I must admit that I wonder why I started this… There are already several such packages in CTAN,[2] and the dramatist package seems really nice (I borrowed some ideas and copied some code from it). There are actually a few improvements in my package compared to dramatist: in my package, plays, acts and scenes appear in the table of contents; it is possible to include several plays in a single document; there are more options when building the dramatis personæ; headers and footers are taken into account, etc. But these improvements are small; it might have been smarter to contribute to dramatist instead of starting my own package. The good part is that I have a package that fits my needs, and I learned how to write a LaTeX package.

Oh, yes! I know why I started this: I am a geek…

1. More information : `https://fr.wikipedia.org/wiki/Thalie_(Muse)`.
2. `http://www.ctan.org/topic/drama-script`

1.3. License

This work may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version.

Further information can be found in the `.dtx` file used to build this document.

In short (but this paragraph has no legal value), you can use this package freely to render your drama plays, and modify it almost freely. Nevertheless, if you like my work, you can invite me to the performance of the play you typeset using my package.[3]

1.4. Acknowledgements

I borrowed ideas and lines of code from the LaTeX packages drama[7] and dramatist[8].

I used the following guides to package my package: *How to Package Your LaTeX Package* [5], and *LaTeX 2ε for class and package writers* [6].

2. About this documentation

2.1. Examples

Most of the examples are taken from Edmond Rostand's *Cyrano de Bergerac* [2], Jean Giraudoux's *Électre* [4], or William Shakespeare's *A Midsummer Night's Dream* [3].

2.2. Overview

Installation instruction are given in section 3. Documentation about how to use this package is given in section II. In particular, section 5.1 explains how to use acts and scenes, sections 5.2 and 5.3 explains how to define characters, and use these definitions to introduce characters' lines, and section 5.4 describes commands used to render stage directions. At last, specific instruction about how to use this package with a language other than English (or a non-latin alphabet) are given in section 6. Examples are given in section A.

3. Download and Install

3.1. Gnu/Linux Distribution

If applicable, the easiest way to get thalie working is by installing it by your distribution package. With Debian (and Ubuntu, and surely other distributions that inherit from Debian), simply run:

3. You will not take a big risk, since there is little chance I travel half the world to see a play. But if by chance I can attend it, it would make me really happy.

```
sudo apt-get install texlive-humanities
```

3.2. LaTeX distribution

This package is included both in TeXLive and MiKTeX. It can be installed by their respective package managers.

3.3. Manual installation

- Download the lastest archive :

  Stable version `http://mirrors.ctan.org/install/macros/latex/contrib/thalie.tds.zip`

  Development version `https://framagit.org/spalax/thalie/repository/archive.zip?ref=main`

- Unzip the archive.
- Move file `tex/latex/thalie/thalie.sty` (or simply `thalie.sty` if you got the development version) in a LaTeX path.

Part II.
Usage

4. Package options

Note that any package option can be redefined anywhere in the document: the options given to `\usepackage[`⟨*options*⟩`]{`⟨*thalie*⟩`}` as passed as-is to command `\setthalieoptions` (see section 4.4 on page 6). This makes it possible to have several plays collected in a single document, while applying different styles or options to each play.

4.1. Space following character commands and stage directions

`xspace = true|false`

Introduced in version v0.7

As commands introducing characters' lines and displaying characters' names may be frequently used, it might be tempting to omit the following `{}`. For instance, one might prefer to write:

```
\lebret \cyranoname n'est pas là.
```

instead of:

```
\lebret \cyranoname{} n'est pas là.
```

If package option xspace is set (e.g. xspace = {true}), space is automatically added after those commands if necessary[4]; otherwise, it is not. The same rule applies after a \did command.

For historical reasons, the option xspace default is xspace = {true}, but this might change in some later non-backward compatible version.

4.2. Style

The way characters' lines are displayed, as well as play, act and scene titles, can be set when loading the options.

characterstyle = bold | margin | center | simple | arden | imprimerie-verse | imprimerie-prose
Set character style. Available styles, and indication to use a custom one, are described in section 5.3.3 on page 14.

playstyle = center|bigcenter|box|custom
Set play style. Available styles, and indication to use a custom one, are described in section 5.1.2 on page 8.

actstyle = center|bigcenter|box|custom
Set act style. Available styles, and indication to use a custom one, are described in section 5.1.2.

scenestyle = center|bigcenter|box|custom
Set scene style. Available styles, and indication to use a custom one, are described in section 5.1.2.

4.3. Sectioning levels

If you use a table of contents, or if you also use "usual" sectioning commands (\chapter, \section and so on), the relative importance of plays, acts and scenes is important.

4. This is done with the \xspace command (from the xspace package), hence the name.

playlevel = part | chapter | section | subsection | subsubsection | paragraph | subparagraph
Set play level in the table of contents.

actlevel = part | chapter | section | subsection | subsubsection | paragraph | subparagraph
Set act level in the table of contents.

scenelevel = part | chapter | section | subsection | subsubsection | paragraph | subparagraph
Set scene level in the table of contents.

interludelevel = play|act|scene
Setting the interlude level is slightly different. When setting it using interludelevel = {⟨*level*⟩}, instead of choosing one of LaTeX vanilla sectioning levels as the level, you may choose one of play|act|scene. It defines if an interlude is at the same level as a play, an act or a scene.

4.4. Redefining options

Introduced in version v0.9a

\setthalieoptions{⟨*package options*⟩}
Packages options can be reset in the document body using this command. Its only argument has the same syntax as the package options (see section 4 on page 4).

For instance, if you are writing a collections of plays, some in prose, some in verses, you might want to write something like the following.

```
\documentclass{book}
\usepackage{thalie}
\begin{document}

\play{First play}
\setthalieoptions{characterstyle=imprimerie-verse}
...

\play{Second play}
\setthalieoptions{characterstyle=impriemie-prose}
...

\end{document}
```

5. Commands and Environments

5.1. Sectioning

Here begin the parts explicitly relating to drama.

5.1.1. Levels

To introduce a new play, act or scene, use commands \play, \act and \scene. Their behaviour is as close as the "usual" sectioning commands (\chapter, \section and so on) as possible, i.e.:

- Their signature is \play[⟨*short title*⟩]{⟨*longtitle*⟩} (the optional short title is the one used in the table of content, and in headers and footers).
- A starred version (\play*, \act* and \scene*) is provided, which inserts a play (or act, or scene) which is not numbered, and does not insert any line in the table of content.
- Headers and footers are changed (more information in section 5.1.4 on the following page).

Both commands \act and \scene (and their starred versions) are designed to deal with empty titles. Indeed, it is common not to give any name to acts and scenes.

By default, a play is as deep (regarding to the table of contents) as a chapter, an act as a section, and a scene as a subsection. But this can be set using package options playlevel, actlevel and scenelevel (see packages options, page 4). That way, you can use in your document plays, acts and scenes as well as chapters, sections and so on. It can be useful if you want a foreword, and appendix, etc.

It is not compulsory to use all three commands \play, \act and \scene. The rule of thumb is: if only one element exists, skip the corresponding command: if your document has a single play, you should ignore \play; if your document has several single act plays, set playlevel = {section}, scenelevel = {subsection}, and ignore \act; etc.

\play[⟨*short title*⟩]{⟨*long title*⟩}
Start a new play (see above).

\act[⟨*short title*⟩]{⟨*long title*⟩}
Start a new act (see above).

\scene[⟨*short title*⟩]{⟨*long title*⟩}
Start a new scene (see above).

\interlude*[⟨*short title*⟩]{⟨*long title*⟩}
Insert an interlude, which is an act or a scene which is not numbered, but which should appear in the table of content (unless the starred version is used).

You may choose the sectioning level an interlude is equivalent to in the package options. If your interludes are acts, use `interludelevel = {act}`; if they are scenes, use `interludelevel = {scene}`.

`\curtain`
At last, to mark the end of an act or of the play, you can use command `\curtain`, which prints the word `\GetTranslation{Curtain}` in the middle of its own line.

5.1.2. Title styles

Several styles are available to render play, act and scene titles. Choose them using package options `playstyle`, `actstyle`, and `scenestyle`. Default is `playstyle = {box}`, `actstyle = {bigcenter}`, `scenestyle = {center}`.

Custom titles Unfortunately, as play, act and scene titles are not considered (by LaTeX) as usual sections, package titlesec[5] cannot be used to use alternative section titles. Here is the way to set your own one.

When loading the package, use `actstyle = {custom}` as the style of the title you want to customize. Then, commands `\customact{`⟨*counter*⟩`}{`⟨*title*⟩`}` and `\customact*{`⟨*title*⟩`}` will be called by this package to render titles. You *must* define them. Example A.2 on page 24 gives the example of the definition of the `center` style.

The first argument of `\customact` is the label of the act being printed (that is, `\theact` for an act, `\theplay` for a play, etc.), its second argument is its title. Command `\customact*` only has one argument, which is the act title.

5.1.3. Labels and counters

Using the same tools as `\chapter`, `\section` and so on, it is possible to define the way counters of plays, acts and scenes are displayed. You can do this by redefining `\theplay`, `\theact` and `\thescene`. For example, to have acts numbered using letters, use: `\renewcommand{\theact}{\Alph{act}}`.

5.1.4. Headers and footers

Once again, similar tools as those used by `\section` are provided to deal with headers and footers. When introducing, a new play, act or scene, respectively, commands `\playmark{`⟨*label*⟩`}`, `\actmark{`⟨*label*⟩`}` and `\scenemark{`⟨*label*⟩`}` are called, so that titles can be used in headers and footers. If the default behaviour does not suit you (which should be the case if you did not choose the default option for `playlevel`, `actlevel` or `scenelevel`), you can redefine them.

Examples are given in section A.6 on page 27.

5. `http://www.ctan.org/pkg/titlesec`

5.2. Dramatis personæ

This part explains how to build and display the dramatis personæ.

Definition of characters is done in document body. As it is possible to have several plays in a single documents (for a collection of plays or sketches), it is possible to define several dramatis personæ. A new one disables the character commands defined by the previous one.

`\begin{dramatis}[hidden = yes|no, defaultcast = {}]`

Definition of characters is done inside the `dramatis` environment.

The following options can be set.

`hidden = true|false`

If the `hidden` option is set to `true` (or given without any argument), the dramatis personæ is not printed (its only purpose is then to define the character commands). Default value is `false` (that is: dramatis personæ is not hidden).

⟨*defaultcast*⟩default (initially empty)

Introduced in version v0.13a

Default value of the `cast` option of command `\character` (see section 5.2.2 on page 11).

Then, several commands are available to define characters, and organize character definitions.

5.2.1. Character definition

Basic definition To define a character, use command `\character`.

`\character[cmd = {⟨command⟩}, drama = {⟨dramatis⟩}, desc = {⟨description⟩}, cast = {⟨cast⟩}]{⟨name⟩}`

`name`

The mandatory argument is the name of the character, as it will appear to introduce each of their line. It is later possible to redefine it using command `\setcharactername` (see part 5.3.2 on page 14). Optional arguments are:

`desc = {⟨description⟩}`

Description of the character, displayed in the dramatis personæ.

`cmd = {⟨command⟩}`

Name of the control sequence that will be used to introduce this character's lines in the remaining part of your play.

`drama = {⟨dramatis⟩}`

Name of the character, as it will appear in the dramatis personæ. The name of the character (mandatory argument) is used as a default value.

`cast = {⟨cast⟩}`

Name of the actor or actress playing this character, displayed in the dramatis personæ. More information in section 5.2.2 on page 11.

If cmd is defined, this command creates two new commands: \⟨*cmd*⟩ and \⟨*cmd*⟩name. The first one is used to introduce a character's line. The second one prints the character's name. An error is raised if a command with any of these two names already exists.

An example of the use of this command is given in section A.1.1 on page 20.

Special character definition Although optional arguments are not mandatory, not defining them, or leaving the mandatory argument blank, have special meaning. The combination are summed up in the following table, where a cross × means that the corresponding combination is forbidden. The superscript number refers to the list of special character definitions (right after the table).

		empty name		name	
		no cmd	cmd	no cmd	cmd
no drama	no desc	×	×	×	default[i]
	desc	description only[ii]	×	×	default[i]
drama = {⟨*drama*⟩}	no desc	silent[iv]	×	×	default[i]
	desc	silent[iv]	×	×	default[i]
drama = {}	no desc	×	×	×	hidden[iii]
	desc	×	×	×	×

Table: "To define or not to define" options of \character

(i) **Default definition (name and cmd are given; desc and drama may be omitted):** The character is defined as described in section 5.2.1 on the previous page. If desc is omitted, no description appear in the dramatis personæ; if no drama is given, character in the dramatis personæ has the same name as it will have in the document. Example:

```
\character[
  drama={A ghost},
  desc={the king's ghost},
  cmd={ghost},
  ]{The ghost}
```

(ii) **Description only (everything omitted but desc):** A description is inserted in the dramatis personæ. Useful to add characters such as *The kings' armies*. Example:

```
\character[desc={The kings' armies}]{}
```

(iii) **Hidden character (drama is empty (defined, but empty), cmd and name are defined, desc is omitted)** Definition of a character that does not appear in the dramatis personæ. In this case, even if defined, the cast argument is ignored. Example:

```
\character[drama={}, cmd={postman}]{The postman}
```

(iv) **Silent character (name is empty; drama is defined; desc may be omitted; cmd is omitted):** The character only appear in the dramatis personæ. It will not be used elsewhere in the document. An optional description may also appear in the dramatis personæ. Example:

```
\character[drama={A priest}]{}
```

Changed in version v0.6

Group of characters

`\begin{charactergroup}[`⟨*width*⟩`]{`⟨*description*⟩`}`

It is possible to group several characters' definition if they have the same description. The optional argument sets the group width (on the left of the brace). See an example on section A.1.2 on page 21.

The effect of this code will be, in the dramatis personæ, to have a nice brace mapping the three characters to their common description.

The behaviour of nested `charactergroup` environments is undefined.[6]

5.2.2. Cast names

Introduced in version 0.13a

The name of the actor or actress playing a character can be displayed next to their name in the dramatis personæ, using the cast option of the `\character` command. Special values are described here; see section A.1.4 on page 22 for an example.

`cast = {`⟨*name*⟩`}` The name given in argument is displayed next to the character definition.

6. It might work; it might not. It might work now, but stop working in the future; it might not work now, but work by accident in the future. Etc.

`cast = {}` No cast is displayed.

`cast = {\underline{\phantom{XXXXXXXXXX}}}` Display a line instead of the cast name: the name of the actor or actress may be added by hand later, when the play has been printed.

Default value is `cast = {}`, unless something else has been set using option `defaultcast` of environment `dramatis` (see 5.2 on page 9).

Introduced in version v0.13a

Cast group Some characters (or group of characters), like the chorus in Greek tragedy, can be played by several actors and actresses.

See an example in section A.1.3.

`\begin{castgroup}[cmd = {⟨command⟩}, drama = {⟨dramatis⟩}, desc = {⟨description⟩}]{⟨name⟩}`
Declare a character, that is played by several cast members. The arguments are the same as the arguments of the `\character` command (see section 5.2.1), excepted that the `cast` option is missing.

`\cast[⟨cast⟩]`
This command is the only one allowed inside a `castgroup` environment. It declares a new cast member (its only argument) for this group.

5.2.3. Vertical space

`\characterspace`
Command `\characterspace` put some vertical space into the dramatis personæ.

5.2.4. Customize style of dramatis personæ

Introduced in version v0.9a

A default style is used to display the dramatis personæ, but it is made to be customized. The following commands can be redefined to make dramatis personæ look different. They do nothing apart from displaying text (no internal definition, etc.). You can redefine them to change the look of the dramatis personæ.

`\begin{dramatisenv}`
This environment wraps the dramatis personæ.

`\begin{dramatischaractergroup}{⟨width⟩}{⟨description⟩}`
Renders a group of characters.

`\begin{dramatischaractercastgroup}{⟨name⟩}{⟨description⟩}{⟨cast⟩}`
Renders a group of cast members.

`\dramatischaracter{⟨name⟩}{⟨description⟩}{⟨cast⟩}`
Renders the character name, description, and cast. It should take care of cases where name or description (but not both) is missing, or cast is empty.

`\dramatischaractername{`⟨*name*⟩`}`
Renders the character name.

`\dramatischaracterdescription{`⟨*description*⟩`}`
Renders the character description.

`\dramatischaractercast{`⟨*cast*⟩`}`
Renders the cast name.

`\dramatiscast{`⟨*cast*⟩`}`
Renders the cast name, inside a `dramatischaractercastgroup`.

`\characterspace`
Adds a vertical character space.

Section A.1.5 on page 23 illustrates how those commands are used.
For example, to render character names in small caps, one would define :

```
\renewcommand{\dramatischaractername}[1]{\textsc{#1}}
```

Or to make dramatis personæ include a title, one can define :

```
\renewenvironment{dramatisenv}{%
  \act*{Dramatis person\ae}%
  \list{}{\rightmargin1cm\leftmargin2cm}\item[]
}{%
  \endlist%
}
```

5.3. Characters

This part explains how to define characters, and introduce character's lines.

It has been explained that definition of a character in the dramatis personæ (see 5.2 on page 9) also creates commands used to introduce lines of characters. For instance, defining a character using `\character[cmd = {cyrano}]{Cyrano}` means than Cyrano's lines can now be introduced using the following code.

```
\cyrano
```

```
Ah ! non ! c'est un peu court, jeune homme !
```

This would display the name *Cyrano* using the current character style, then his line.

5.3.1. "Disposable" characters

One can need to define characters that are used only once (or a few times).

`\disposablecharacter[`⟨*directions*⟩`]{`⟨*name*⟩`}`
This command, used in the body, introduces a line for character `{`⟨*name*⟩`}`, with optional stage directions. It is rendered the same way regular characters are rendered, but nothing is added to the dramatis personæ. It is used in section A.3.2 on page 25.

This command is also useful to introduce lines of characters in non standard ways, for instance if several characters speak at once. See examples in section A.3.1 on page 25.

5.3.2. Changing character's name

It is possible to change the name that appears to introduce character's lines within the text. This can be used, for example, when a character is first refered to as *A voice*, until we learn his real identity, which is *Cyrano*. An example is given in section A.3.2 on page 25.

`\setcharactername{`⟨*command*⟩`}{`⟨*name*⟩`}`

`command`
Control sequence used to introduce this character's line.

`name`
Name to display for this character.

5.3.3. Lines

To introduce characters' lines, use the commands defined in the character definition (see the `\character` command, or the example in section A.1.1 on page 20).

Choose style Several styles are available, to typeset character's name and lines in different ways. Change the style by loading package using option `characterstyle = {`⟨*style*⟩`}` (available styles are `bold`, `center`, `margin`, `simple`, `imprimerie-verse`, `imprimerie-prose`). A few notes:

- `simple` Default style.
- `imprimerie-verse` and `imprimerie-prose` French *de facto* standards for typesetting drama plays, respectively in verse and in prose, as defined by the *Imprimerie nationale* [1].

Introduced in version v0.6

Introduced in version v0.6

- `arden` Mimick typesetting of the Arden Shakespeare series.
- `bold` Needs a sans serif font to be defined (it should be done by default with latin alphabets, but must be done by hand with languages using non latin characters (greek, hindi, etc.)).

Customize style If available styles does not fit your need, you can also define your own one. To do so, you can redefine the following commands.

`\speakswithoutdirection{`⟨*name*⟩`}`
Invoked to display a character's name to introduce their line. It takes one argument, which is the character's name.

`\speakswithdirection{`⟨*name*⟩`}{`⟨*direction*⟩`}`
Also invoked to display a character's name to introduce their line, but it takes a second argument, which is stage directions to be printed together with character's name.

See also section 5.4 to see other ways to print stage directions.

5.4. Stage directions

Changed in version 0.12a

Let us begin with a warning: since we could not figure out a nice command name made from "stage directions", we used the French word *(didascalie)* to build up command names.

We define two ways to render stage directions: a short one, that is printed inline, and a long one, printed in its own paragraph.[7]

`\did{`⟨*directions*⟩`}`
Display a stage direction inline.

`\begin{dida}`
Display a stage direction in its own paragraph.

Note that option `xspace` may be used to automatically add a space, if necessary, after a `\did` command (see section 4.1 on page 4, page 4).

`\onstage{`⟨*directions*⟩`}`
Some stage directions are displayed centered under the scene or act definition (I often see it in classical drama, where the list of characters appearing in each scene is displayed that way). This command can be used to display such information.

`\pause`
As we often need to mark pauses in theater, we created the command `\pause`, which is a shortcut for `\did{\GetTranslation{Pause}}`.

7. For people writing math: They act a bit like `\(1+1=2\)` and `\[1+1=2\]` for formulas.

5.5. Splitting verses

Changed in version v0.10c

When writing verses, it is sometimes useful to split a verse between two characters: the first character starts it in their line, and the second one finishes it in a second line. Visually, the start of the second line is vertically aligned to the end of the first line.

Warning: this is still experimental. I present several recipes to achieve this, but they all have their pros and cons.

The recipes presented below all implement macros \pauseverse and \resumeverse. See section A.4.1 on page 26 for an example.

\pauseverse
Used at the end of the verse to be continued.

\resumeverse
Used at the beginning of the continuing verse.

When writing a play in verse, one can use a lot of those commands, which can be tedious because of their long names. So, defining "shortcuts" of those commands can be handy, as in the following example.

```
\newcommand{\pv}{\pauseverse}
\newcommand{\rv}{\resumeverse}
```

This is not done by default, because short command names are a scarce ressource in LaTeX, so, defining these commands by default might conflict with other packages, while most of thalie users will not use them.

5.5.1. Which recipe should I use?

As I have already stated, there are several ways to achieve this, but none of them is perfect.

- Do try the first recipe (section 5.5.2). If it works, stick to it. Otherwise…
- If you do not mind loading the TikZ package, use it (section 5.5.4). As far as I know, from a user point of view, it is the easiest recipe to use, with the less bugs. Otherwise…
- You can try building you own commands (section 5.5.3). Otherwise…
- The last recipe (section 5.5.5) always works. It is far less user-friendly, but it does the job.

5.5.2. Recipe 1 : Pure LaTeX

Warning: I do not understand the code behind this feature. I copied-pasted code by David Carlisle[8] without understanding it. There are ~~probably~~ bugs,[9] but I won't be able to solve them.

The implemented macros \pauseverse and \resumeverse are written in pure LaTeX(without any external package). For backward compatibility reasons, this will remain the default implementation for some time…

If the character names are displayed on the same line as the verse, the lines are not aligned the way one would expect them to. To correct this, command \adjustverse{⟨*length*⟩} can be used to add an extra (possibly negative) space when calling \resumeverse, to get a correct alignment. See section A.4.2 on page 26, for an example of those three commands.

The young inexperienced fool[10] who originally wrote those commands had not noticed that \adjustverse is needed because \resumeverse does not take into account the length of the character names, in styles where it appears on the same line as the verse. This (probably) won't be fixed, because some people might be using this, and fixing this might break their document.

If you encounter the same issue, you might prefer one of the other recipes.

5.5.3. Recipe 2 : With package `hanging`

Warning: This is a do-it-yourself recipe.

The bug announced at the end of the previous recipe, has been spotted and partially corrected by egreg on StackExchange.[11] He proposes the following code to fix this.

```
\usepackage{hanging}

\newlength{\brokenverse}
\renewcommand*{\speakswithoutdirection}[1]{%
  \settowidth{\brokenverse}{\textsc{#1}.}%
  \hangpara{2\parindent}{1}\noindent\textsc{#1}.%
}
\makeatletter
\renewcommand{\resumeverse}{%
  \hspace{\@verseadjust}\hspace{\dimexpr\dimen\@ne-\brokenverse-2em}
}
\makeatother
```

8. Originally written as an answer to a StackExchange question: see `https://tex.stackexchange.com/a/107727`.
9. One on them lead to the next recipe.
10. Me…
11. `https://tex.stackexchange.com/a/545326`

This code ignores the choosen style (it actually redesigns it), and it only works with lines without character directions. If needed, you should also define a `\speakswithdirection` command to implement the same feature.

5.5.4. Recipe 3 : With package `TikZ`

As far as I know, this recipe is the best one, from a user point of view. But it loads the `TikZ` package, which is a huge package to implement such a tiny feature.[12]

```
\usepackage{tikz}
\usetikzlibrary{tikzmark}
\usetikzlibrary{math}
\newcounter{thalieverse}
\tikzset{tikzmark suffix=-\thethalieverse}%
\renewcommand{\pauseverse}{%
  \stepcounter{thalieverse}
  \tikzmark{thaliepauseverse}
}
\renewcommand{\resumeverse}{%
  \tikzmark{thalieresumeverse}%
  \iftikzmark{thaliepauseverse}{%
    \begin{tikzpicture}[remember picture]%
      % Uncomment to see "pause" and "resume" tikz marks.
      %\draw[overlay,green, ultra thick] (pic cs:thaliepauseverse) -- (
  pic cs:thalieresumeverse);%
      \tikzmath{
        coordinate \c;
        \c1 = (pic cs:thaliepauseverse);
        \c2 = (pic cs:thalieresumeverse);
        \width = max(0, \cx1 - \cx2);
      }
      % Set opacity to 1 to see the space
      \draw[opacity=0, black, line width=1ex] (0, 0) -- (\width pt, 0);
    \end{tikzpicture}%
  }{}%
}
```

This recipe is implemented in the `examples/henry.tex` file in the repository of this project.

12. In French, we would say : “Un marteau-pilon pour écraser une mouche” (a far too heavy solution for a tiny problem).

5.5.5. Recipe 4 : With `\phantom{}`

At last, if nothing else workd, this feature can also be implemented using the \phantom command (see section A.4.3 on page 26 for an example). It is far less user-friendly, but it is also less error prone. Choose wisely…

6. Localization and Internationalization

6.1. Localization

Introduced in version v0.9a

By default, commands \act, \scene, etc. use English words, but they are translated to the current language used by babel or polyglossia.[13]

Language cannot be defined directly in this package. Instead, the language currently used by babel or polyglossia is used to translate the following words (with or without capital letter, singular or plural): *play*, *act*, *scene*, *interlude*, *pause*, *curtain*.

Changed in version v0.9a

If the option for your language does not exist,[14] or if you want to change the default words used here, you can set your own translation using command \DeclareTranslation (see the translations package for more information). For instance use `\DeclareTranslation {Spanish}{Curtain}{Telòn}` for Spanish.[15]

See section 6.3 on the next page for the list of supported languages.

6.1.1. Common translations

Commands defined here are shorter versions of strings commonly translated.

\playname
Shortcut for \GetTranslation{Play}.

\actname
Shortcut for \GetTranslation{Act}.

\scenename
Shortcut for \GetTranslation{Scene}.

\interludename
Shortcut for \GetTranslation{Interlude}.

\pausename
Shortcut for \GetTranslation{Pause}.

\curtainname
Shortcut for \GetTranslation{Curtain}.

13. Given that someone provided the translation. See section 6.3 on the following page for the list of available languages.
14. You can also send me the translation for your language, to improve this package.
15. I may be wrong: I cannot speak Spanish.

6.2. Non-latin alphabets

This package should work with languages using non-latin alphabets (greek, hindi, etc.). However, here are a few remarks.

- The `bold` character style (see section 5.3.3 on page 14) can only be used if a sans serif font has been defined. For instance, for hindi, you might write in the preamble: `\newfontfamily\hindifontsf{Noto Sans Devanagari}`.

6.3. Language specific comments

The following languages are available:

English

- The `arden` character style tries to mimick the typestting of the Arden Shakespeare series.

French

- The `imprimerie-verse` and `imprimerie-prose` character styles implements the French *de facto* standards for typesetting drama plays, respectively in verse and in prose, as defined by the *Imprimerie nationale* [1].
- The same book [1] states that act numbers should be typeset in capital roman numerals (which is done by default with thalie), and that scene numbers should be typeset in small capital roman numerals, which is *not* the default, but can be achieved by adding the following line in the preamble: `\renewcommand{\thescene}{\textsc{\roman{scene}}}`.

Introduced in version v0.10a

German

Introduced in version v0.11a

Italian

I am not a linguist or dramatist in any of those languages. If you have interesting advice for those languages that you would like to share (preferably with a reference to a reliable source), I would be happy to add them to this list!

Part III.
Appendix

A. Examples

Introduced in version v0.9a

A.1. Dramatis personæ

A.1.1. Character definition

```
\begin{dramatis}
  \character[cmd={cyrano}, drama={Cyrano de Bergerac}]{Cyrano}
  \character[cmd={lebret}]{Le Bret}
  \character[cmd={bellerose}]{Bellerose}
\end{dramatis}

\bigskip

\lebret[à \cyranoname, lui prenant le bras]
Çà, causons !\ldots

\cyrano
Laisse un peu sortir cette cohue\ldots{}
\did{À \bellerosename.}
Je peux rester ?
```

Cyrano de Bergerac
Le Bret
Bellerose

Le Bret, *à Cyrano, lui prenant le bras* : Çà, causons !…
Cyrano : Laisse un peu sortir cette cohue… *(À Bellerose.)* Je peux rester ?

A.1.2. Define a group of characters

```
\begin{dramatis}
\begin{charactergroup}[2.5cm]{Officers}
  \character[desc={a soldier}, cmd=marcellus]{Marcellus}
  \character[desc={a soldier}, cmd=bernardo]{Bernardo}
\end{charactergroup}
\end{dramatis}
```

Marcellus, a soldier
Bernardo, a soldier } Officers

A.1.3. Define a cast group

```
\begin{dramatis}
  \character[cmd=narses, cast={Raymone}]{La femme Narsès}
  \begin{castgroup}[drama={Les Euménides}]{}
    \cast{Marthe Herlin}
    \cast{Monique Mélinand}
    \cast{Denise Pezzani}
  \end{castgroup}
  \begin{castgroup}[drama={Les petites Euménides}]{}
    \cast{Véra Pharès}
    \cast{Nicole Munié-Berny}
    \cast{Clairette Fournier}
  \end{castgroup}
  \character[cmd=mendiant, cast={Louis Jouvet}]{Le Mendiant}
\end{dramatis}
```

La femme Narsès ..... Raymone

Les Euménides { Marthe Herlin, Monique Mélinand, Denise Pezzani

Les petites Euménides { Véra Pharès, Nicole Munié-Berny, Clairette Fournier

Le Mendiant ..... Louis Jouvet

A.1.4. Example of `defaultcast`.

We already know that John Doe will play Astérix. Actors for Obélix and Cléopatre have not been found yet. Idéfix will be played by a puppet: no need to put a cast name.

```
\begin{dramatis}[defaultcast={\underline{\phantom{XXXXXXX}}}]
  % Explicit cast
```

```
  \character[desc={guerrier gaulois}, cast={John Doe}, cmd=asterix]{
Astérix}
  % Default cast
  \character[desc={sculpteur de menhir, ami du précédent}, cmd=obelix]{
Obélix}
  \character[desc={reine d'Égypte}, cmd=cleopatre]{Cléopatre}
  % No cast
  \character[desc={chien d'Obélix}, cast={}, drama={Idéfix}]{}
\end{dramatis}
```

Astérix, guerrier gaulois ..... John Doe
Obélix, sculpteur de menhir, ami du précédent ..... _____
Cléopatre, reine d'Égypte ..... _____
Idéfix, chien d'Obélix

A.1.5. Commands used to display the dramatis personæ.

```
\begin{dramatis}
  \character[cmd=queen]{The queen}
  \character[cmd=king, desc={the queen's husband.}]{The king}
  \begin{charactergroup}{Lords and ladies}
    \character[drama={Lord Foo}]{}
    \character[desc={Lord Foo's sister.}, drama={Lady Bar}]{}
    \character[desc={Lord and Lady Baz.}]{}
  \end{charactergroup}
  \characterspace
  \character[desc={Several servants}]{}
  \begin{castgroup}[drama={Chorus}]{}
    \cast{Jane Doe}
    \cast{John Doe}
  \end{castgroup}
\end{dramatis}
```

To display the dramatis personæ above, the commands below are called.

```
\begin{dramatisenv}
  \dramatischaracter{The queen}{}{}
  \dramatischaracter{The king}{the queen's husband.}{}
  \begin{dramatischaractergroup}[5cm]{Lords and ladies}
    \dramatischaracter{Lord Foo}{}{}
    \dramatischaracter{Lady Bar}{Lord Foo's sister.}{}
    \dramatischaracter{}{Lord and Lady Baz.}{}
  \end{dramatischaractergroup}
  \characterspace
  \dramatischaracter{}{Several servants}{}
  \begin{dramatischaractercastgroup}{}{Chorus}{}
    \dramatiscast{Jane Doe}
    \dramatiscast{John Doe}
  \end{dramatischaractercastgroup}
\end{dramatisenv}
```

Command \dramatischaracter{⟨*name*⟩}{⟨*description*⟩}{⟨*cast*⟩} is defined as something like the following code (excepted that parts taking care of cases where there is no character name or description is omitted here).

```
\newcommand{\dramatischaracter}[3]{ %
  \dramatischaractername{#1}, \dramatischaracterdescription{#2}\dotfill\
  dramatischaractercast{#3}\\
}
```

A.2. Custom definition of acts

To redefine the definition of \act, one can use the following code.

```
\newcommand\customact[2]{
  \begin{center}
    \textsc{\GetTranslation{Act} #1}

    #2
  \end{center}
}
\WithSuffix\newcommand\customact*[1]{
  \begin{center}
    #1
  \end{center}
```

```
}
```

A.3. Charcters' lines

A.3.1. Introduce lines of several characters at the same time.

Since \disposablecharacter can display arbitrary text using the style of character names, it can be used to render a custom text when two characters are speaking together.

```
\eve
Let's open it.

\disposablecharacter{\alicename and \bobname}
No!
```

Eve: Let's open it.
Alice and Bob: No!

A.3.2. Change the name of a character

On the first scenes of *Cyrano de Bergerac*, the main character (Cyrano) is somewhere in a crowd, and only appears as *The voice* (*La voix* in French). He appears as Cyrano in the following verses.

```
\cyrano Sortez !

\disposablecharacter{Le parterre} Oh !

\montfleury[s'étranglant] \enquote{Heureux qui loin des cours\ldots}

\setcharactername{cyrano}{Cyrano}
\cyrano[surgissant du parterre, debout sur une chaise, les bras croisés,
 le feutre en bataille, la moustache hérissée, le nez terrible] Ah ! je
 vais me fâcher !\ldots
```

La voix : Sortez !
Le parterre : Oh !
Montfleury, *s'étranglant* : « Heureux qui loin des cours… »
Cyrano, *surgissant du parterre, debout sur une chaise, les bras croisés, le feutre*

en bataille, la moustache hérissée, le nez terrible : Ah ! je vais me fâcher !…

A.4. Splitting verses

A.4.1. Commands used to split verse

```
\setthalieoptions{characterstyle=imprimerie-verse}

\hermia
So is Lysander.\pauseverse

\theseus
\resumeverse In himself he is.
```

HERMIA
So is Lysander.
THESEUS
In himself he is.

A.4.2. Commands used to adjust splitted verse

```
\setthalieoptions{characterstyle=simple}
\adjustverse{-7em}

\hermia
So is Lysander.\pauseverse

\theseus
\resumeverse In himself he is.
```

HERMIA: So is Lysander.
THESEUS: In himself he is.

A.4.3. Commands used to split verse (using the `\phantom` command)

```
\hermia
So is Lysander.

\theseus
\phantom{So is Lysander.} In himself he is.
```

HERMIA: So is Lysander.
THESEUS: In himself he is.

A.5. Footnotes

Add a footnote on a character name Adding a footnote on the character name introducing its line cannot be done on the character command defined by the dramatis personæ. Instead, the \disposablecharacter command can be used instead.

```
\disposablecharacter{\alicename\footnote{An interesting note.}}
Hello, world!
```

Add a footnote in a stage direction Footnote must be protected to be included in stage directions.

```
\cyrano[frappant dans ses mains \protect\footnote{An interesting footnote
  .}]
Une !
```

A.6. Headers and footers

By default, with the article class, the header of the left page contains the play name, and the header of the right page contains the act number.

Header: Act number on left, scene number on right

```
\renewcommand{\actmark}[1]{%
  {\markboth{\MakeUppercase{\GetTranslation{Act}}\ \theact\ #1}{}}%
}
```

```
\renewcommand{\scenemark}[1]{%
  {\markright{\MakeUppercase{\GetTranslation{Scene}}\ \thescene\ #1}}%
}
```

Play name of left, act and scene name on right

```
\usepackage{fancyhdr}
\pagestyle{fancy}
\fancyhf{}
\fancyhead[RO]{%
  \textit{\MakeUppercase{%
    \GetTranslation{Act} \theact{}, %
    \GetTranslation{Scene} \thescene%
  }}%
}
\fancyhead[LE]{\leftmark}
```

A.7. Line numbers

Some styles write character names in the left margin, which can overlap with line numbers displayed by package lineno. To avoid this, line numbers can be displayed on the right of the page using option `[right]` of package lineno.

B. Change History

This is a raw copy of the *CHANGELOG.md* file that can be found in the git repository of thalie.

- thalie 0.13a (2022-12-11)
  - Documentation overhaul.
  - [\character] Fix tests of forbidden combination of arguments (some combinations were wrongfully forbidden, some were wrongfully allowed).
  - Add `defaultcast` option to dramatis personae, and `cast` option to `\character`.
  - Add a `castgroup` environment.

  – Louis Paternault spalax@gresille.org[16]

- thalie 0.12a (2022-10-09)
  - The `xspace` option also affects \did.

16. mailto:spalax@gresille.org

– Louis Paternault spalax@gresille.org[17]

- thalie 0.11b (2022-06-16)
  - Fix sloppy previous release.

  – Louis Paternault spalax@gresille.org[18]

- thalie 0.11a (2022-06-15)
  - New translation: Italian (thanks Giuseppe Palma).

  – Louis Paternault spalax@gresille.org[19]

- thalie 0.10c (2021-09-26)
  - Document different implementations of (sometimes buggy) \pauseverse and \resumeverse.

  – Louis Paternault spalax@gresille.org[20]

- thalie 0.10b (2019-07-31)
  - Minor fixes and updates of examples.

  – Louis Paternault spalax@gresille.org[21]

- thalie 0.10a (2019-02-24)
  - New translation: German (thanks Ekkart Kleinod).
  - Minor documentation updates.

  – Louis Paternault spalax@gresille.org[22]

- thalie 0.9b (2017-04-24)
  - Remove blank page at the beginning of documentation.
  - Add missing dependency.

  – Louis Paternault spalax@gresille.org[23]

- thalie 0.9a (2017-04-22)
  - Thalie.sty
    * Commands `\playmark`, `\actmark` and `\scenemark` no longer include label (e.g. "Act 1"). It is up to the user to add it or not.
    * Display default translations, even in language environments using non-latin characters where no latin font is available (closes #24).

17. mailto:spalax@gresille.org
18. mailto:spalax@gresille.org
19. mailto:spalax@gresille.org
20. mailto:spalax@gresille.org
21. mailto:spalax@gresille.org
22. mailto:spalax@gresille.org
23. mailto:spalax@gresille.org

    * Do not add a wrong indentation after character name in verse environment (closes #3).
    * Package options can be redefined anywhere in the document (closes #5).
    * Reduce vertical space around character names in style `imprimerie-verse` (closes #6).
    * Replace package ifthen[24] with package etoolbox[25] (closes #29).
    * Style of dramatis personæ can be customized (closes #9 #15 #18 #19).
    * Use translations[26] to translate words ("act", "scene", etc.).
  - Documentation
    * Add sections *Examples* and *Localization*.
    * Add a note about non-latin characters and character style `bold` (closes #26).
    * Fix `\customact` example.
    * Various minor improvements.
  - README
    * Convert README from text to markdown.
    * Add examples.

 – Louis Paternault spalax@gresille.org[27]

- thalie 0.8 (2015-12-30)
  - Fix release errors in previous release.

 – Louis Paternault spalax@gresille.org[28]

- thalie 0.7 (2015-12-28)
  - Update project URL to http://framagit.org/spalax/thalie.
  - Add commands `\pauseverse`, `\resumeverse` and `\adjustverse`.
  - Add package option `xspace`.
  - Minor documentation improvements.

 – Louis Paternault spalax@gresille.org[29]

- thalie 0.6 (2014-06-26)
  - Add character style `imprimerie-verse`, `imprimerie-prose`, and `arden`.
  - Correct a lot of typos in documentation (thanks Per).
  - Better alignment of groups of characters.
  - Improve spacing (thanks Caroline).
  - Several documentation and core improvements.

24. `http://ctan.org/pkg/ifthen`
25. `http://ctan.org/pkg/etoolbox`
26. `http://ctan.org/pkg/translations`
27. `mailto:spalax@gresille.org`
28. `mailto:spalax@gresille.org`
29. `mailto:spalax@gresille.org`

– Louis Paternault spalax@gresille.org[30]

- thalie 0.5 (2013-06-08)
  - Initial release.

  – Louis Paternault spalax@gresille.org[31]

References

[1] Imprimerie nationale, *Lexique des règles typographiques en usage à l'Imprimerie nationale*, 2002, ISBN : 978-2-7433-0482-9

[2] Edmond Rostand, *Cyrano de Bergerac*, 1897

[3] William Shakespeare, *A Midsummer Night's Dream*, 1600

[4] Jean Giraudoux, *Électre*, 1937

[5] Scott Pakin, *How to Package Your LaTeX Package — Tutorial on writing .dtx and .ins files*, 2004, `http://www.ctan.org/pkg/dtxtut/`

[6] The LaTeX Team, *LaTeX 2ε for class and package writers*, 2006, `http://www.ctan.org/pkg/clsguide`

[7] Matt Swift, *drama — Production-style stage script in LaTeX*, 2001, `http://www.ctan.org/pkg/drama`

[8] Massimiliano Dominici, *dramatist — Typeset dramas, both in verse and in prose*, 2005, `http://www.ctan.org/pkg/dramatist`

[9] James Kilfiger, *play — Typeset drama using LaTeX*, 2001, `http://www.ctan.org/pkg/play`

[10] John Pate, *screenplay — A class file to typeset screenplays*, 2012, `http://www.ctan.org/pkg/screenplay`

[11] Wing L Mui, *sides — A LaTeX class for typesetting stage plays*, 2005, `http://www.ctan.org/pkg/sides`

30. `mailto:spalax@gresille.org`
31. `mailto:spalax@gresille.org`

Index

U

X