A Calligraphy Grid in ConTeXt/LMTX

This project is a generator of grids for medieval calligraphy in ConTeXt/LMTX and MetaPost.

Files included

```
pautado-context
├── doc
│   └── context
│       └── third
│           └── pauta
│               ├── env-pauta.tex
│               ├── pauta-doc.pdf
│               ├── pauta-doc.tex
│               ├── pauta-example.pdf
│               └── pauta-example.tex
├── tex
│   └── context
│       ├── interface
│       │   └── third
│       │       └── t-pauta.xml
│       └── third
│           └── pauta
│               └── t-pauta.mkxl
├── .gitignore
├── install-module.sh
├── LICENSE
├── make-release.sh
├── README.md
└── VERSION
```

Installation

Option 1: Automated Installation Script

The easiest way to install the module is using the provided installation script:

```
./install-module.sh [context-installation-path]
```

The script will:

- Automatically detect your ConTeXt installation (if in common locations)
- Check for existing installations and compare versions
- Prompt you before overwriting existing versions
- Install all files to the correct TDS locations
- Provide instructions for rebuilding the database

Example:

```
./install-module.sh /home/user/Apps/lmtx
```

After installation, rebuild the ConTeXt database:

```
context --generate
```

Option 2: Manual Installation

1. Clone this repository: `git clone https://github.com/conradolandia/pauta.git`, or download as zip.
2. Copy the `doc` and `tex` folders to your ConTeXt tree and rebuild your database with `context --generate`. You can find more details about the process on the ConTeXt wiki.
3. Alternatively, call context with the `--path` flag, and provide it with the path of this folder, i.e:

```
context --path=/home/user/pauta
```

4. Alternatively still, simply place `t-pauta.mkxl` on the same directory as the file importing it.

Use

1. Invoke the `\Pauta` macro as many times as you want pages. Each invocation can have a different configuration. Each invocation will create one single page.
2. The data about the hand (the calligraphic term for "font") is autogenerated by the module and set into the header or footer, following the user configuration.
3. **Warning:** This module takes over the header and footer typesetting areas (see the layout article for more information about ConTeXt typesetting areas), and does not reset them properly yet. So if your document includes other content on those areas you will need to reset again to your liking by manually invoking `\setupheadertexts` and/or `\setupfootertexts`. This will be hopefully improved in the future.

Configuration Parameters

All parameters are optional. Defaults are as follows:

```
\Pauta[
  hand=, % Hand name. If not defined, will not show info on the left side of the header / footer
  handInfo=, % Some extra info for the hand. If not defined, will not show info on the right side of the header / footer
  infoPosition=footer, % Where to show the extra info (header | footer)
  infoLeft={\setup{pauta:content:leftmark}}, % If defined, will override autogenerated hand info on the left side of the header / footer
  infoRight={\setup{pauta:content:rightmark}}, % If defined, will override autogenerated hand info on the right side of the header / footer
  displayNibs=false, % Show nib-width marks (true | false)
  displayAngleMarks=false, % Display dotted guides for the nib angle (true | false)
  nibWidth=3mm, % Pen nib width (must include units, or it will default to big points)
  nibAngle=35, % Nib working angle in degrees
  ascenders=3, % Number of ascender lines (in nib widths)
  xHeight=4, % Number of x-height lines (in nib widths)
  descenders=3, % Number of descending lines (in nib widths)
  adjustment=0, % Sometimes it's necessary to adjust the height, because it can be longer than TextHeight. Still
not sure why it happens but it happpens... a value of 1 or 2 should solve it.
  mainColor={s=.4}, % Main color (lines that separate sections)
  secondaryColor={s=.6}, % Secondary color (lines separated by a nib width)
  tertiaryColor={s=.8}, % Tertiary color (nib width marks on the left margin and dotted angle lines)
]
```

Code Examples

Example 1: Basic Usage

```
\usemodule[pauta]

\startdocument
\Pauta[
  hand={Carolingian},
  handInfo={Tours school, VIII\high{th} century},
  infoPosition=header,
  displayNibs=true,
  displayAngleMarks=true,
  nibWidth=3mm,
  ascenders=2,
  xHeight=3,
  descenders=2,
  adjustment=0,
  mainColor={s=.6},
  secondaryColor={s=.8},
  tertiaryColor={s=.8},
]
\stopdocument
```

Example 2: Multiple Pauta Instances

```
\usemodule[pauta]

\startdocument
\Pauta[
 hand={Carolingian},
 handInfo={Tours school, VIII\high{th} century},
 infoPosition=header,
 displayNibs=true,
 displayAngleMarks=true,
 nibWidth=3mm,
 ascenders=2,
 xHeight=3,
 descenders=2,
 adjustment=0,
 mainColor={s=.5},
 secondaryColor={s=.6},
 tertiaryColor={s=.7},
]
```

```
% Overriding the header / footer info:

\Pauta[
  infoLeft={An exercise in Visigothic script},
 infoRight={from an Spanish manuscript, VII\high{th} century},
 infoPosition=footer,
 displayNibs=true,
 displayAngleMarks=false,
 nibWidth=2mm,
 ascenders=4,
 xHeight=3,
 descenders=4,
 adjustment=1,
 mainColor={s=.3},
 secondaryColor={s=.4},
 tertiaryColor={s=.5},
]
\stopdocument
```

Development

Creating a Release

To create a release package for distribution:

```
./make-release.sh
```

This script will: - Read the version from the `VERSION` file - Create a properly structured ZIP archive following TDS conventions - Exclude build artifacts (`.log`, `.tuc`, `.tuo` files) - Include all documentation PDFs - Create `pauta-YYYY.MM.DD.zip` ready for upload to modules.contextgarden.net

Module Structure

The module follows the ConTeXt module writing guidelines: - TDS-compliant directory structure - Self-documenting source code with `%D` comments - XML interface file for command documentation - Proper version management