otter

Jul 28, 2022

Quick Start

CHAPTER 1

Input file

The basic structure of the input file is:

```
import otter

viewports = [
  ...
]

colorbars = [
  ...
]

annotations = [
  ...
]

obj = {
  ...
  'viewports': viewports,
  'colorbars': colorbars,
  'annotations': annotations
}

if __name__ == '__main__':
    otter.render(obj)
```

- The input file is an actual python file and can be used like that, i.e. it can be executed and it will produce the output file.
- `viewports` is a list of dictionaries and it specifies areas with results. Each entry specifies a single area.
- `colorbars` is a list of dictionaries and it specifies result's color bars. Typically for some result with a mesh. Each entry specifies one color bar.
- `annotations` is a list dictionaries and it specifies things like text, images, or time annotation. Each entry specifies one annotation.

- `obj` can be either an image or a movie. The type is determined by listed parameters.
- As previously stated, each entry in `viewports`, `colorbars` and `annotations` is a dictionary where the key is the name of a parameter and the value is its value. Values can be of different types like `int`, `float`, `list`, etc.

CHAPTER 2

Object Parameters

2.1 Image

```
image = {
  'size': [ float, float],
  't', float,
  'time-unit': {'sec', 'min', 'hour', 'day'},
  'file': str,
  'background': [float, float, float],
  'gradient_background': bool,
  'background2': [float, float, float],
  'viewports': list,
  'colorbars': list,
  'annotations': list
}
```

size *width* and *height* of the image - either on screen of the physical size of the rendered image.

t simulation time the image is rendered for.

time-unit Time unit – used globally. For example, time annotation will pick this value.

file If specified, render into a file, otherwise render on screen in an interactive window.

background An array of three numbers between `0` and `1` where each entry represents red, green and blue component.

gradient_background If `True`, then the `background2` parameter is used and a linear background is rendered.

background2 Used if `gradient_background` is True.

viewports List of *Viewports*.

colorbars List of *Colorbar* s.

annotations List of *Annotations*.

2.2 Movie

```
movie = {
  'duration': float,
  'file': str,
  'size': [1280, 720],
  'times': [],
  'time-unit': 'min',
  'background': [float, float, float],
  'gradient_background': bool,
  'background2': [float, float, float],
  'viewports': list,
  'colorbars': list,
  'annotations': list
}
```

duration Duration in seconds of the final rendered movie.

file File name of the final movie.

size *width* and *height* of the movie.

times List of times for which we render the images. If not specified, time steps from the result file will be used.

time-unit Time unit – used globally. For example, time annotation will pick this value.

background An array of three numbers between `0` and `1` where each entry represents red, green and blue component.

gradient_background If `True`, then the `background2` parameter is used and a linear background is rendered.

background2 Used if `gradient_background` is True.

viewports List of *Viewports*.

colorbars List of *Colorbar* s.

annotations List of *Annotations*.

frame *Optional*. File name mask of rendered images.

location *Optional*. Location where the images are rendered. By default rendering happens in some temp location determined by the operation system.

2.3 Viewports

2.3.1 Exodus Result

```
vp = {
  'type': 'ExodusResult',
}
```

2.3.2 Plot Over Line

```
vp = {
  'type': 'PlotOverLine',
}
```

2.3.3 Vector Postprocessor Plot

```
text = {
  'type': 'VPPPlot',
}
```

2.4 Colorbar

```
colorbar = {
  'location': { 'left' | 'top' | 'right' | 'bottom' },
  'origin': [float, float],
  'viewport': [float, float, float, float],
  'layer': int,
  'width': float,
  'length': float,
  'primary': axis,
  'secondary': axis,
}
```

location Location of the numbers relative to the color bar.

origin Position of the color bar in the viewport.

viewport *left*, *bottom*, *right* and *top* of the viewport where this color bar is displayed.

layer Layer number.

width Width of the color bar relative to viewport.

length Length of the color bar relative to viewport.

primary Primary axis of the color bar.

secondary *Optional*. Secondary axis of the color bar.

2.5 Annotations

2.5.1 Text

```
text = {
  'position': [float, float],
  'opacity': float,
  'color': [float, float, float],
  'shadow': bool,
  'halign': {'left' | 'center' | 'right'},
  'valign': {'bottom' | 'middle' | 'top'},
```

(continues on next page)

(continued from previous page)

```
  'text': str,
  'font-size': float,
  'font-family': str,
  'bold': bool,
  'italic': bool
}
```

position The text position within the viewport, in relative coordinates.

opacity Set the text opacity.

color The text color.

shadow Toggle text shadow.

halign Set the font justification.

valign The vertical text justification.

text The text to display.

font-size The text font size.

font-family The font family.

bold Font bolding.

italic Italic type.

2.5.2 Image

```
text = {
  'position': [float, float],
  'width': float,
  'halign': {'left' | 'center' | 'right'},
  'valign': {'bottom' | 'middle' | 'top'},
  'opacity': float,
  'scale': float,
  'file': str,
}
```

position The position of the image center within the viewport, in relative coordinates.

width The logo width as a fraction of the window width, this is ignored if 'scale' option is set.

halign The position horizontal position alignment.

valign The position vertical position alignment.

opacity Set the image opacity.

scale The scale of the image. By default the image is scaled by the width.

file The PNG file to read, this can be absolute or relative path to a PNG or just the name of a PNG located in the `chigger/logos` directory.

2.5.3 Time

```
time = {
  'position': [float, float],
  'opacity': float,
  'color': [float, float, float],
  'shadow': bool,
  'halign': {'left' | 'center' | 'right'},
  'valign': {'bottom' | 'middle' | 'top'},
  'text': str,
  'font-size': float,
  'font-family': str,
  'bold': bool,
  'italic': bool,
  'format': str
}
```

position The text position within the viewport, in relative coordinates.

opacity Set the text opacity.

color The text color.

shadow Toggle text shadow.

halign Set the font justification.

valign The vertical text justification.

text The text to display.

font-size The text font size.

font-family The font family.

bold Font bolding.

italic Italic type.

format Formatting string for the time

2.6 Filters

2.6.1 Transform

```
transform = {
  'scale': [float, float, float]
}
```

scale Scaling factor for x, y and z direction.

2.6.2 Plane Clip

```
plane_clip = {
  'origin': [float, float, float],
  'normal': [float, float, float],
```

(continues on next page)

(continued from previous page)

```
  'inside_out': bool
}
```

origin The origin of the clipping plane.

normal The outward normal of the clipping plane.

inside_out When True the clipping criteria is reversed.

2.6.3 Box Clip

```
plane_clip = {
  'lower': [float, float, float],
  'upper': [float, float, float],
  'inside_out': bool
}
```

lower The lower corner of the clipping box.

upper The upper corner of the clipping box.

inside_out When True the clipping criteria is reversed.

2.7 Axis

```
axis = {
  'num-ticks': int,
  'range': [float, float],
  'font-size': int,
  'font-color': [float, float, float],
  'title': str,
  'grid': bool,
  'grid-color': [float, float, float],
  'precision': int,
  'notation': { 'standard' | 'scientific' | 'fixed' | 'printf'},
  'ticks-visible': bool,
  'axis-visible': bool,
  'labels-visible': bool,
  'scale': float
}
```

num-ticks The number of tick marks to place on the axis.

range The axis extents.

font-size The axis title and label font sizes, in points.

font-color The color of the axis, ticks, and labels.

title The axis label.

grid Show/hide the grid lines for this axis.

grid-color The color for the grid lines.

precision The axis numeric precision.

notation The type of notation, leave empty to let VTK decide. Can be 'standard', 'scientific', 'fixed', 'printf'.

ticks-visible Control visibility of tickmarks on colorbar axis.

axis-visible Control visibility of axis line on colorbar axis.

labels-visible Control visibility of the numeric labels.

scale Scale factor for the axis. Useful for changing units. For example, to go from *meters* to *centimeters* set this to *1e2*.

2.8 Camera

```
camera = {
  'view-up': [float, float, float],
  'position': [float, float, float],
  'focal-point': [float, float, float]
}
```

view-up ???

position The position of the camera.

focal-point The the focal point of the camera.

CHAPTER 3

Testing

Tests are located in `tests` dir.

3.1 Commands

To run tests:

```
$ python -m unittest discover -s tests -v
```

To generate code coverage:

```
$ coverage run -m unittest discover -s tests -v
$ coverage html
```

Open `htmlcov/index.html` in your browser.

CHAPTER 4

Deploy AppBundle for Mac OS X

This page describes how to build and deploy and App bundle for MacOS X.

4.1 Setup environment

```
$ virtualenv venv
$ . venv/bin/activate
$ pip install -r venv-requirements.txt
```

4.2 Mac OS X: Build icns

```
$ mkdir icon.iconset
$ sips -z 512 512 otter.png --out icon.iconset/icon_512x512.png
$ iconutil -c icns icon.iconset
$ rm -rf icon.iconset
```

4.3 Build the bundle

```
$ python setup.py py2app
```

4.4 Fix the install manually

```
$ cp /path/to/libffi.6.dylib /path/to/Otter.app/Contents/Frameworks/
```

Note: `libffi.6.dylib` can sit either on the system or in miniconda dir

4.5 NOTES

- if package dependencies change, generate a new `requirements.txt` by running:

```
$ pip freeze > venv-requirements.txt
```

4.6 Resources

Bibliography

[py2app] https://py2app.readthedocs.io/en/latest/

[metachris] https://www.metachris.com/2015/11/create-standalone-mac-os-x-applications-with-python-and-py2app/