
python-cg Documentation

Release 0.1

Jakub Stasiak

April 10, 2013

CONTENTS

1	python-cg	1
1.1	What is python-cg?	1
1.2	What's the state?	1
1.3	Requirements	1
1.4	Documentation	2
1.5	Building	2
1.6	Important information	2
1.7	Quickstart	2
1.8	Testing	3
1.9	License	4
2	Indices and tables	5

PYTHON-CG

1.1 What is python-cg?

python-cg is a Python wrapper for NVidia Cg Toolkit runtime. I've started it because I like Python, I like NVidia CG and I want to do some computer game/3d graphics prototyping and research. Also I still find C++ counterproductive as far as my needs are concerned and I don't want to waste my time doing boring stuff. Programming in Python is fun.

I know about some projects that were meant to bring CG to Python but as far as I know they're history now.

Project is hosted at GitHub: <https://github.com/jstasiak/python-cg>.

1.2 What's the state?

The project is in very early development stage. Overview of what's supported right now:

- Cg contexts
 - creating
 - destroying
- CgFX effects
 - creating from file
 - creating directly from source code
- accessing effects' techniques and their passes
- accessing effect parameters with their names, semantics and parameter-specific metadata (rows, columns etc.)
- setting sampler parameters and most of numerical parameters

What doesn't work at the moment and there's no plan to implement it:

- everything that's left (well, until I decide I need some of it or someone else does that)

1.3 Requirements

This project requires:

- NVidia Cg Toolkit 3.0

- Python interpreter (+ development files):
 - 2.x 2.6, or
 - 3.x 3.2
- C and C++ compiler

Python packages required to build and install *python-cg*:

- Cython 0.18
- numpy

To build documentation/run tests you also need:

- Mock 1.0
- Nose 1.2
- Sphinx ~ 1.2 (development version)

1.4 Documentation

Pregenerated documentation can be found at <https://python-cg.readthedocs.org/en/latest/>.

You can also build documentation all by yourself by calling:

```
sphinx-build -b html docs docs/build/html
```

Generated HTML files are placed in `docs/build/html/` directory.

1.5 Building

To build the project in place, run:

```
python setup.py build_ext --inplace
```

1.6 Important information

- This project works with OpenGL and OpenGL only
- It uses row-major matrices by default, just like numpy does

1.7 Quickstart

First you need to create an instance of `CG` class and use it to create new `Context`:

```
from cg import CG
```

```
cg = CG()
context = cg.create_context()
```

We want to use an effect to render some stuff so we're gonna create `Effect` from file:

```
effect = context.create_effect_from_file('effect.cgfx')
```

Note: This assumes that you have a file named `effect.cgfx` and that it contains a valid CG effect.

We now have access to Effect's techniques and parameters:

```
for technique in effect.techniques:
    # ...

for parameter in effect.parameters:
    # ...
```

For the sake of simplicity let's say we have a parameterless effect with only one `Technique`:

```
technique = effect.techniques[0]
```

Now we can access technique's passes. Each `Pass` has methods `begin()` and `end()` and the actual drawing has to take place between a call to `begin` and `end`:

```
gl.glClear(gl.GL_COLOR_BUFFER_BIT)

for pass_ in technique.passes:
    pass_.begin()

    gl.glBegin(gl.GL_TRIANGLES)
    gl.glVertex3f(-0.5, -0.5, 0)
    gl.glVertex3f(0.5, -0.5, 0)
    gl.glVertex3f(0, 0.5, 0)
    gl.glEnd()

    pass_.end()

# swap buffers
```

You can find complete, runnable example application in `example` directory. Please note that it requires (in addition to `python-cg` requirements):

- Development version of SFML 2
- Python packages listed in `example/requirements.txt`:

```
pip install -r example/requirements.txt
```

Then to run the example:

```
python setup.py build_ext --inplace
PYTHONPATH=. python example/main.py
```

1.8 Testing

To run tests, execute:

```
python runtests.py
```

1.9 License

© 2013, Jakub Stasiak

This project is licensed under MIT License, see LICENSE file for details.

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*