Geometric and Graphics Programming Laboratory: workshop 1

Alberto Paoluzzi

October 23, 2017

Outline: workshop 1

1. From PLaSM classic to Pyplasm
2. Jupyter notebook required
3. Minimal git/github instructions

From PLaSM classic to Pyplasm

Syntactical diffs (1/2)

PLaSM classic (FL)

- List

  ```
  <a,b,c,d>
  ```

- Application operator

  ```
  fun : args
  ```

- Composition operator

  ```
  f ~ g
  ```

- Construction operator

  ```
  [f,g,h]:x == <f:x,g:x,h:x>
  ```

pyplasm (Python)

- List (array)

  ```
  [a,b,c,d]
  ```

- Application operator

  ```
  fun(args)
  ```

- Composition operator

  ```
  COMP([f,g])
  ```

- Construction operator

  ```
  CONS([f,g,h])(x) ==
      [f(x),g(x),h(x)]
  ```

Syntactical diffs (2/2)

PLaSM classic (FL)

```
DEF name (arg::pred)(a1,a2::pred) = expr
WHERE
    local1 = expr1,
    local2 = expr2
END
```

pyplasm (Python)

```
def name(arg):
    local1 = expr1
    local2 = expr2
    def name1(a1,a2):
        return expr
    return name1
```

Workshop assignment

Convert some Classic PIASM scripts from Chapter 1 and/or Chapter 2 of book Geometric Programming for Computer-Aided Design (GP4CAD)

Free choice of number and type of scripts to Convert

Style specs (1/2)

- produce a notebook file, of type `.ipynb` (The ipynb file extension is associated with the IPython notebook and/or Jupiter, a rich architecture for interactive computing written in Python and available for various platforms.)

- alternate notebook cells with
  - Title and description (markdown)
  - PLaSM classic code (markdown teletype)
  - Python code
  - Image from execution (markdown)

Style specs (1/2)

- standard output: a single HPC value
- use meaningfull identificators (variables and parameters)
- use camelCase ids
- add Python `docstrings` (google for it)
- produce a single notebook file, named workshop_01.ipynb
- file path: `your_repos/ggpl/2017-10-23/workshop_01.ipynb`

Jupyter notebook required

Notebook tutorial

Notebook Basics

Minimal git/github instructions

Minimal git/github instructions (1/2)

create your local repository

```
$ mkdir development
$ cd development
$ git clone https://github.com/your-account/ggpl
$ cd ggpl
$ mkdir 2017-10-23
$ cd 2017-10-23
$ touch workshop_01.ipynb
```

Minimal git/github instructions (2/2)

commit your work

```
$ cd ..  # move to ggpl/
$ git add -A .
$ git commit -m "add a short note to commit"
$ git push origin master
```

References