pyRT - Computer Graphics in Jupyter Notebooks for Fun and Teaching
Image Generation using Pure Python

Prof. Martin Christen
FHNW – University of Applied Sciences and Arts Northwestern Switzerland
Institute Geomatics
Motivation

Source: retro64, Manual C16
Why do we want more Computer Graphics in Python?

Game Development

Severside Graphics Generation

- Simple to complex
- Speichern von Filmen z.B. als animiertes gif
- Teaching
  - Loops
  - Sorting Algorithms
  - ...

Streaming Content on Twitch / YouTube / …

- Creating Real-Time Content (Demo will follow)
Other Modules (selection)

2D Graphics

- Gizeh (https://github.com/Zulko/Gizeh)
- Pygame (https://github.com/pygame/pygame)
- Arcade (https://github.com/pvcraven/arcade)
- Pyxel (https://github.com/kitao/pyxel)
- Kivy (https://github.com/kivy/kivy)
- PyQt5 (and other GUI-Toolkits)
- MoviePy (https://github.com/Zulko/moviepy)
  und Prolog (https://github.com/Edinburgh-Genome-Foundry/Proglog)
Other Modules (Selection)

3D Graphics

- Vapory (https://github.com/Zulko/vapory)
- pythreejs (https://github.com/jupyter-widgets/pythreejs)
- three.py (https://github.com/stemkoski/three.py)
- PyOpenGL (https://github.com/mcfletch/pyopengl)
- ModernGL (https://github.com/moderngl/moderngl)
- PyQt5 (+ ModernGL oder QtOpenGL)
- Blender / Cinema 4D and Scripting
PyRT (pronounced pirate) is a raytracer/image generator for Python 3.5 and higher. This project is mainly done with the following in mind:

- Ray tracing in the Jupyter notebook
- Teaching computer graphics and ray tracing
- Exploring ray tracing concepts for geo data using Python.
- Rendering geo data, including large point clouds.
- Implementing new algorithms for rendering large 3D city models.
- Creating 3D-Maps from OpenStreetMap data
- Server-side rendering / cloud based rendering
- Having fun programming graphics stuff
**Installation**

`pip install pyrt`

(numpy & pillow are highly recommended, in theory nothing is required, but that makes displaying images hard)
First Steps: pyRT & Virtual Framebuffer

```python
from pyrt.renderer import RGBImage
from pyrt.math import Vec2, Vec3
import random

w = 320
h = 240
image = RGBImage(w, h)
image.clear(Vec3(0.0, 0.0, 0.4))

for i in range(5000):
    position = Vec2(random.randint(0, w - 1), random.randint(0, h - 1))
    color = Vec3(random.uniform(0, 1), random.uniform(0, 1), random.uniform(0, 1))

    image.drawPoint(position, color, 1)

image.framebuffer()
```
Let’s switch to Jupyter

Seeing this all in action makes more sense!

So let’s go!

Yay! Live Demo!!!

Source will be located at: https://github.com/martinchristen/EuroPython2020