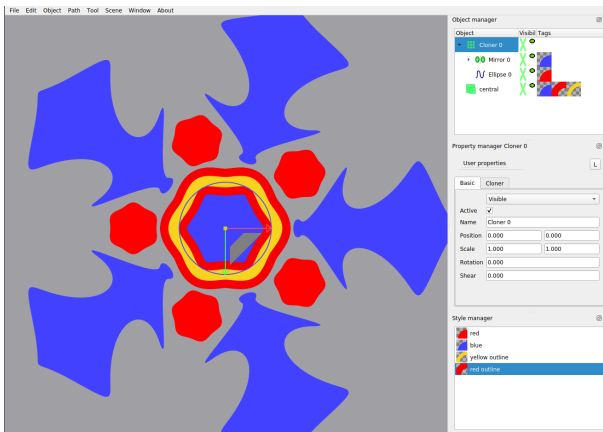# Interactive Procedural Non-Destructive 2D-Vector Graphics

Pascal Bies
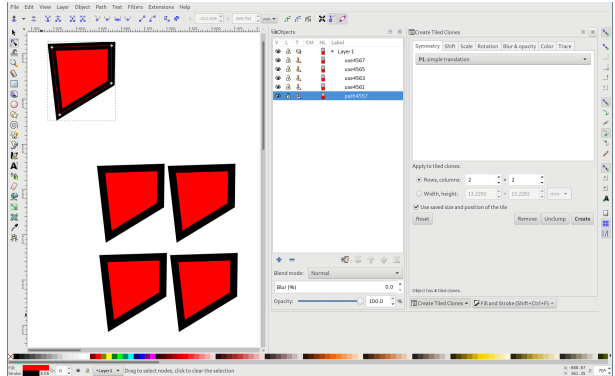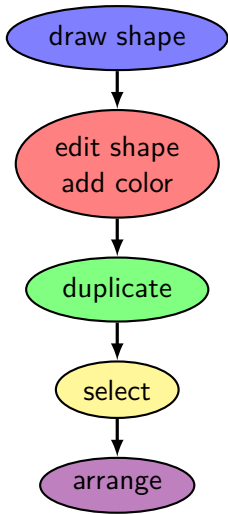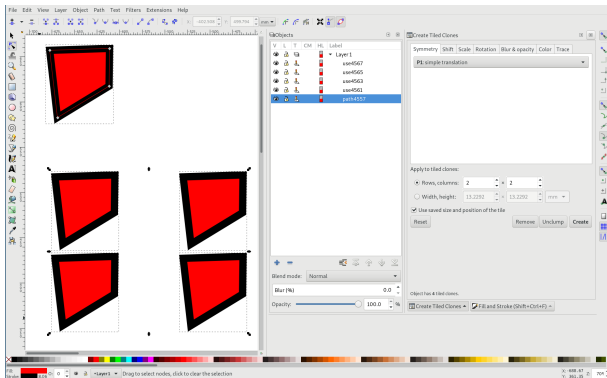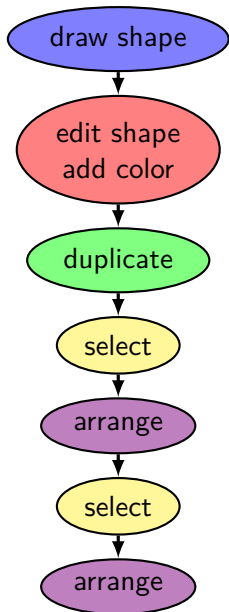
31.5.2019

What is this talk about?

- ▶ Advertising my latest development ~~Advertising my latest development~~
- ▶ What's (imho) wrong with existing vector graphic editors
- ▶ How to improve existing vector graphic editors
- ▶ … better start from scratch?

Example: Inkscape (I)

draw shape

↓

edit shape
add color

↓

duplicate

↓

select

↓

arrange

Example: Inkscape (II)

draw shape → edit shape add color → duplicate → select → arrange → select → arrange
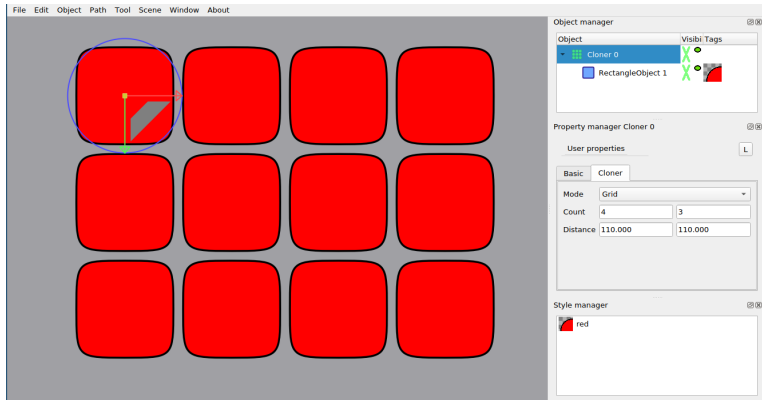
+ clone: editing original affects clones!

– replacing original hard (not possible?) <small>better had cloned a group</small>

– changing the arrangement later difficult <small>Object → Arrange…</small>

– many clones clutter object tree, difficult handling
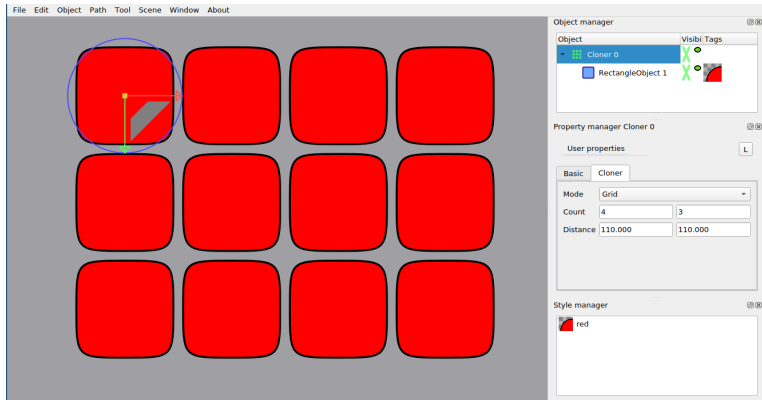
Idea: Special Generator-Objects (I)
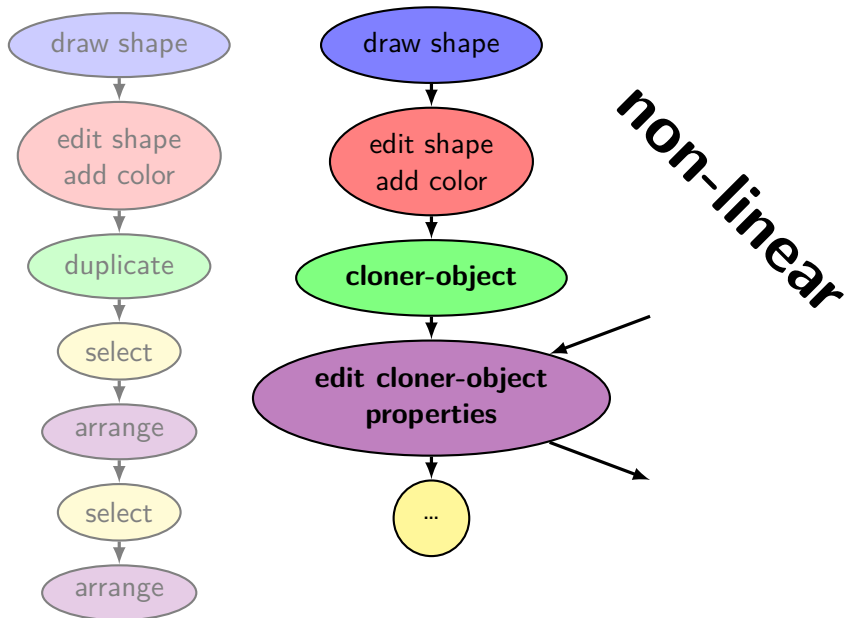
Idea: Special Generator-Objects (II)

- ▶ `Cloner` is parent of `Rectangle`
- ▶ editable special properties (mode, count, distance, …)
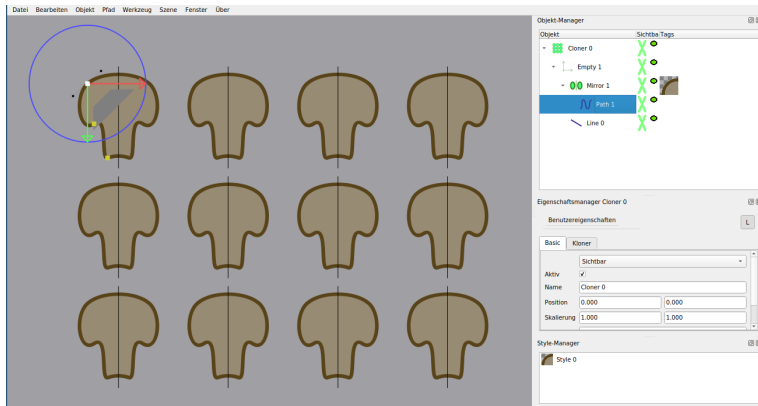- ▶ **object**-**oriented**, not *tool-oriented*

Idea: Special Generator-Objects (III)

+ replace/modify the template-object: easily anytime

+ arrangement-properties: edit anytime

+ handle only two objects, not many

Palette of Special Generator-Objects

- ▶ Cloner, Mirror, Instance, ...
- ▶ nested generators possible and **encouraged**
- ▶ e.g. clone mirrored shape

This Idea is not new

- ▶ programming
  - ▶ functions, loop, recusion, variables
  - ▶ plethora of paradigms: oo, procedural, ...
  - ▶ principles: DRY, KISS, ...
- ▶ LaTeX, TikZ
  - ▶ macros, variables
- ▶ 3D-modelling, -animation, ...
  - ▶ modifiers, instances, cloner, procedural modelling, ...

?? Why no 2D-WYSIWYG ??
object-oriented, structured, dry

Why no such 2D-WYSIWYG-editor?

- ▶ too abstract and complicated for artists?
  - ▶ for many use-cases that's true
- ▶ ... but it works for 3D
  - ▶ 3D is inherently more complicated
- ▶ use WYSIWYM instead?
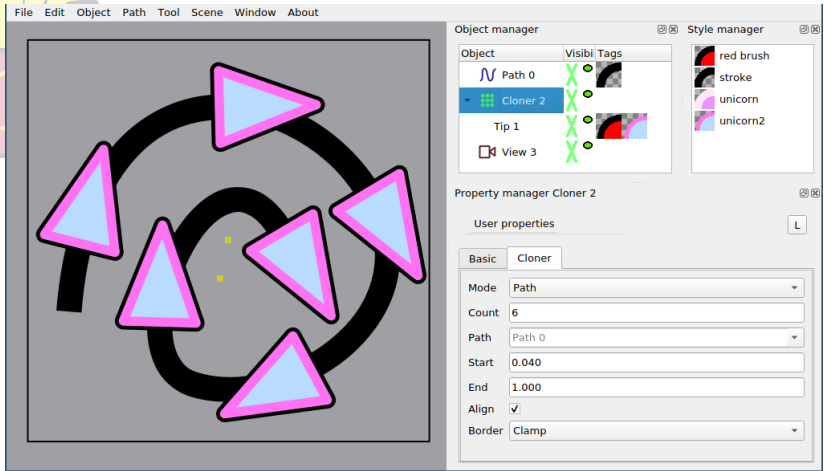
[by Andrew Michael Fitzsimon]

TikZ LATEX
**matplotlib**
[from https://matplotlib.org]

Who is left?

- ▶ Me and those who ...
- ▶ ... like abstraction, accustomed to coding or 3D
- ▶ ... like to have immediate visual feedback
- ▶ ... don't want to sustain duplication and unstructured documents

About Omm (sorry a.l.e....)

- ▶ implements **user-friendly** vector drawing
- ▶ object-oriented, structured, dry, non-destructive, non-linear
- ▶ brings 3D-workflow into 2D-world

About Omm (sorry a.l.e....)
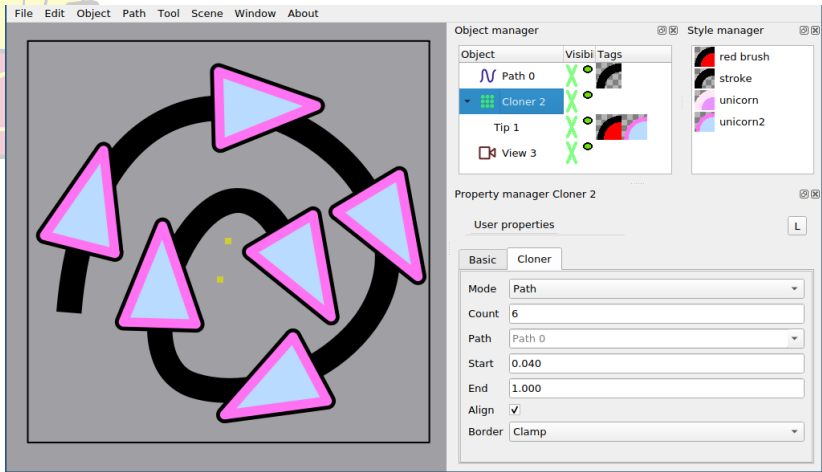
- **Don't Sacrifice Usability**: encourage, don't constrain
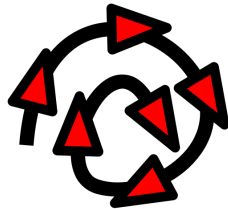- transparent, intuitive: user **understands** how it works

Omm—features (I)

- shapes: Rectangle, Ellipse, Line, Procedural-Path
- generators: Cloner, Instance, Mirror, Outline, Text
- image-object (pixel/svg/pdf)
- Bézier curves
- tools (select, knife, transform, …)
- scriptable 🐍 python™
  [No affiliation with python.org]
- simultaneous editing
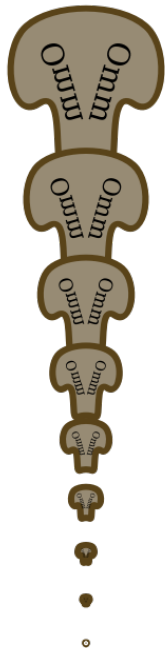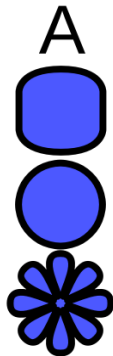
Omm—features (II)

- flexible gui: drag'n'drop, dockable widgets
- extend objects with tags
- separable styles (via tags)
- exhausitve undo/redo
- custom key-bindings   `M, Alt+Shit+X`
- internationalization
- load/save to json   [no affiliation with json.org]
- export pixel/svg/pdf [soon]
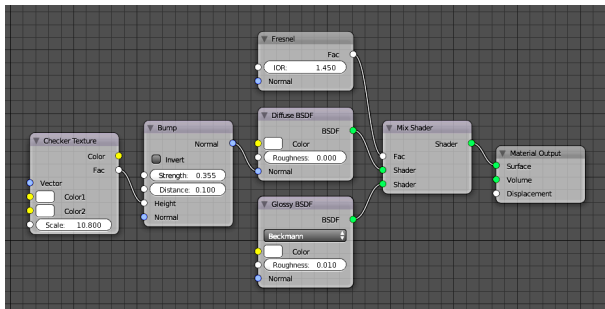
Future

- more generators, shapes and (few) tools
- **introduce animations**
- introduce modifiers (clone jitter, bend, bevel, …)
- improve styles (like blender nodes)
- procedural stuff (L-system, fractals, …)
- performance and workflow optimizations

get Code and Win64-exe
https://github.com/pasbi/ommpfritt

[Simon Oxley]

snapcraft

[The Blender 2.79 Manual • by the Blender Documentation Team • is licensed under a CC-BY-SA v4.0.]

Thanks! Questions?

TOPOLOGICAL CORRECTNESS

3D-workflow

python™

non-linear

non-destructive

transparency

Tools → **Object**

scriptable

semantic

intuitive

maintainability

animation

modularity

don't repeat yourself
instead: reuse