

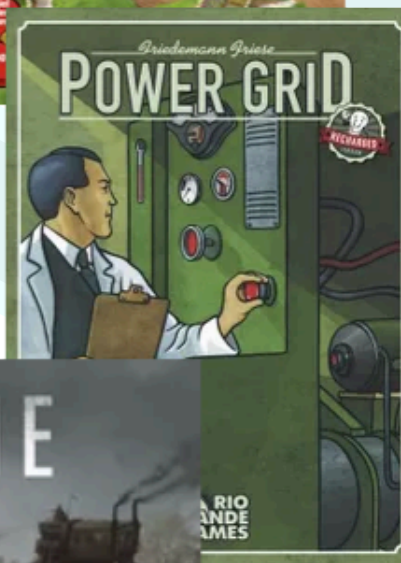
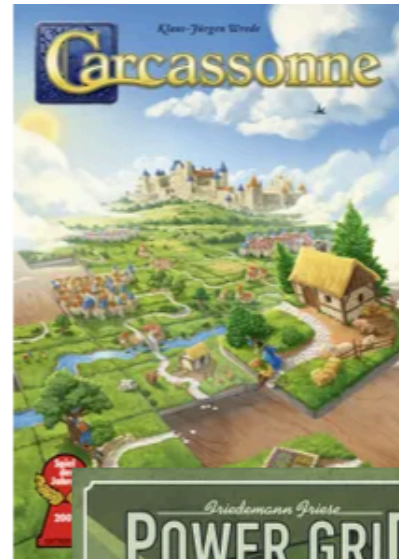
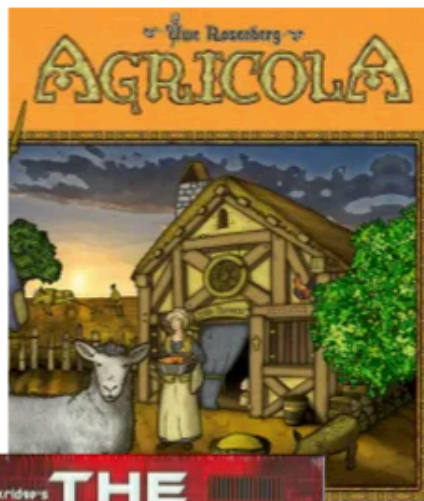
# Tabletop games < 3 FOSS

# Yes Tabletop games

- Boardgames
- Cardgames







# Me

Åke Forslund

Embedded software developer at Leine Linde

Mastodon: @forslund@mastodon.nu

E-mail: [ake.forslund@gmail.com](mailto:ake.forslund@gmail.com)

A large, stylized teal snowflake graphic with a black outline, positioned behind the text. The snowflake has six main branches, each with smaller sub-branches.

# Frozen Maze Games

# Our games







*Photo by @shelf\_no\_shame*



# Overview

- Making a Tabletop Game
- The tools we ended up with
  - The good parts
  - The bad parts

# Making a boardgame

1. Thinking it's a good idea...
2. Prototype -> Cut & Paste
3. Preview version -> Nice artwork
4. Mass-produced version

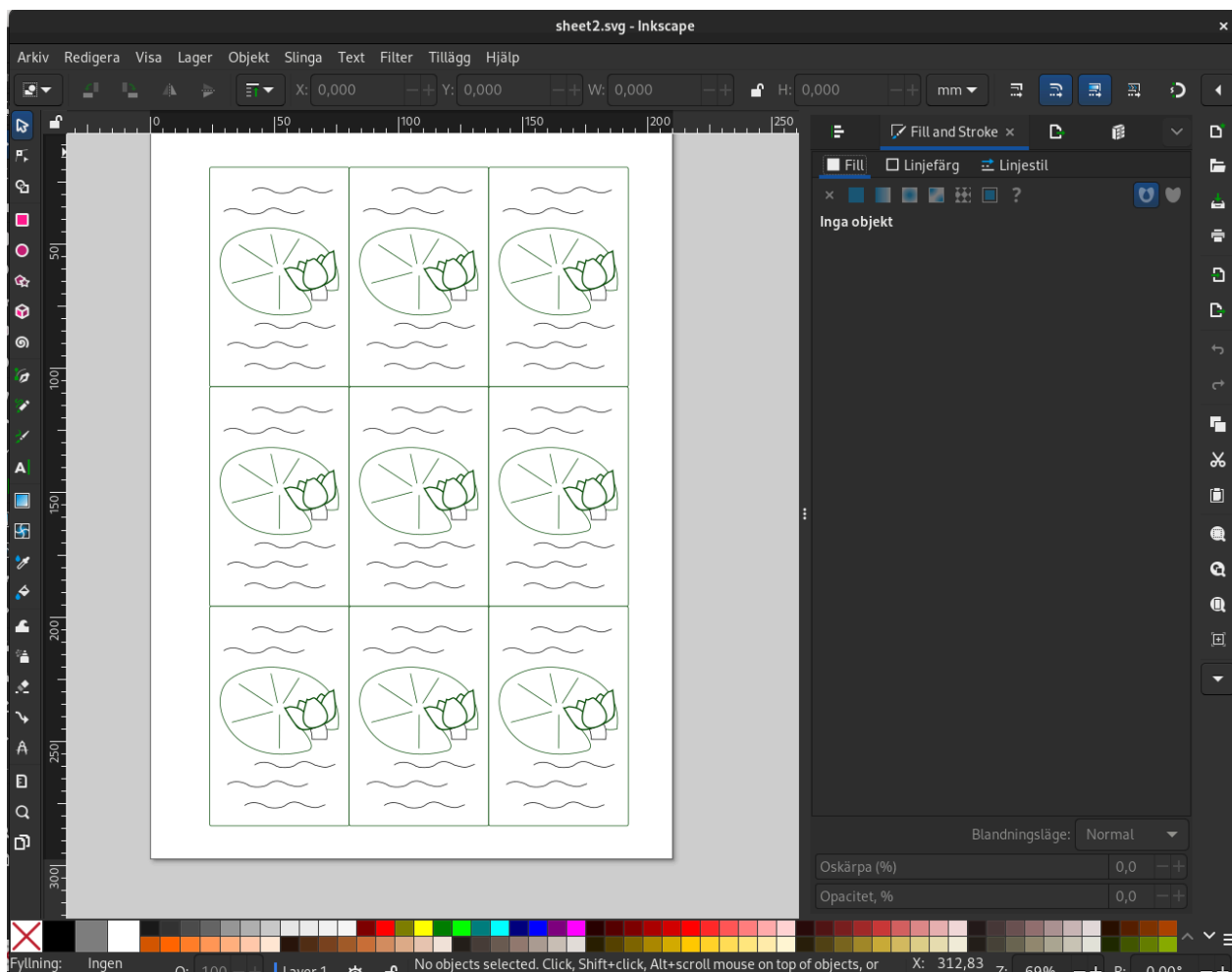


# Prototyping





# Inkscape



# What's good about it?

- Easy to move things around
- "Page" based
- High Skill ceiling
- "Modern" interface



# Creating a Preview version

All of a sudden it needs to look good...



# Working with an artist



# What do we need now?

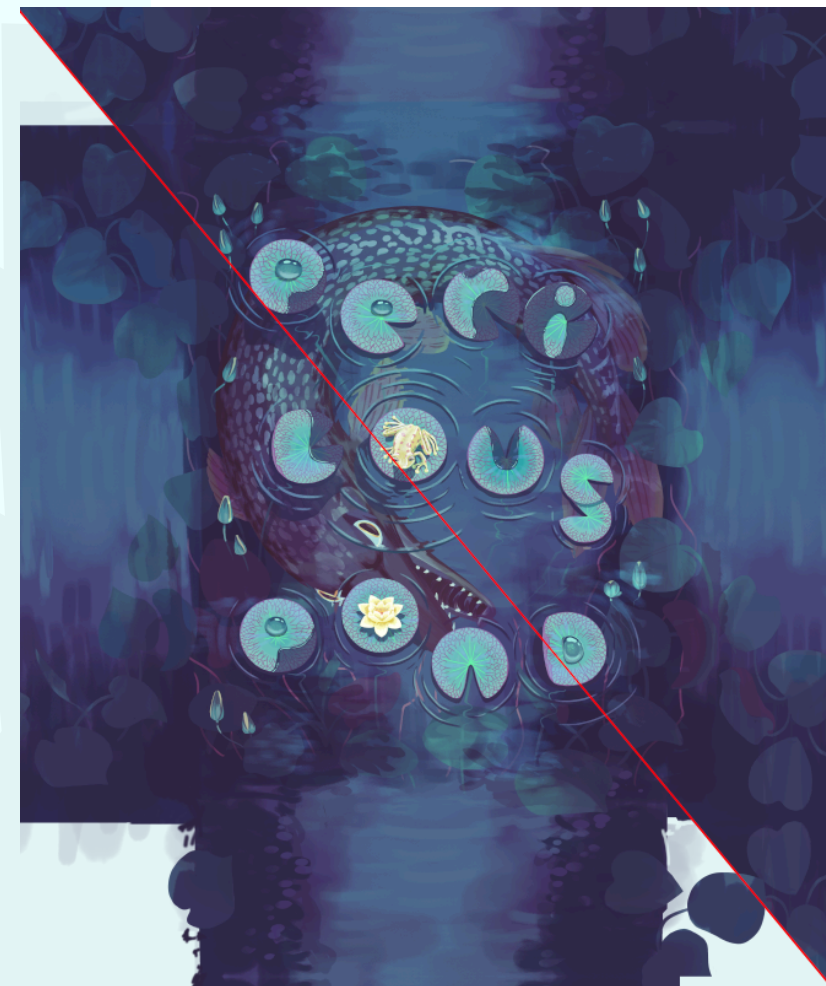
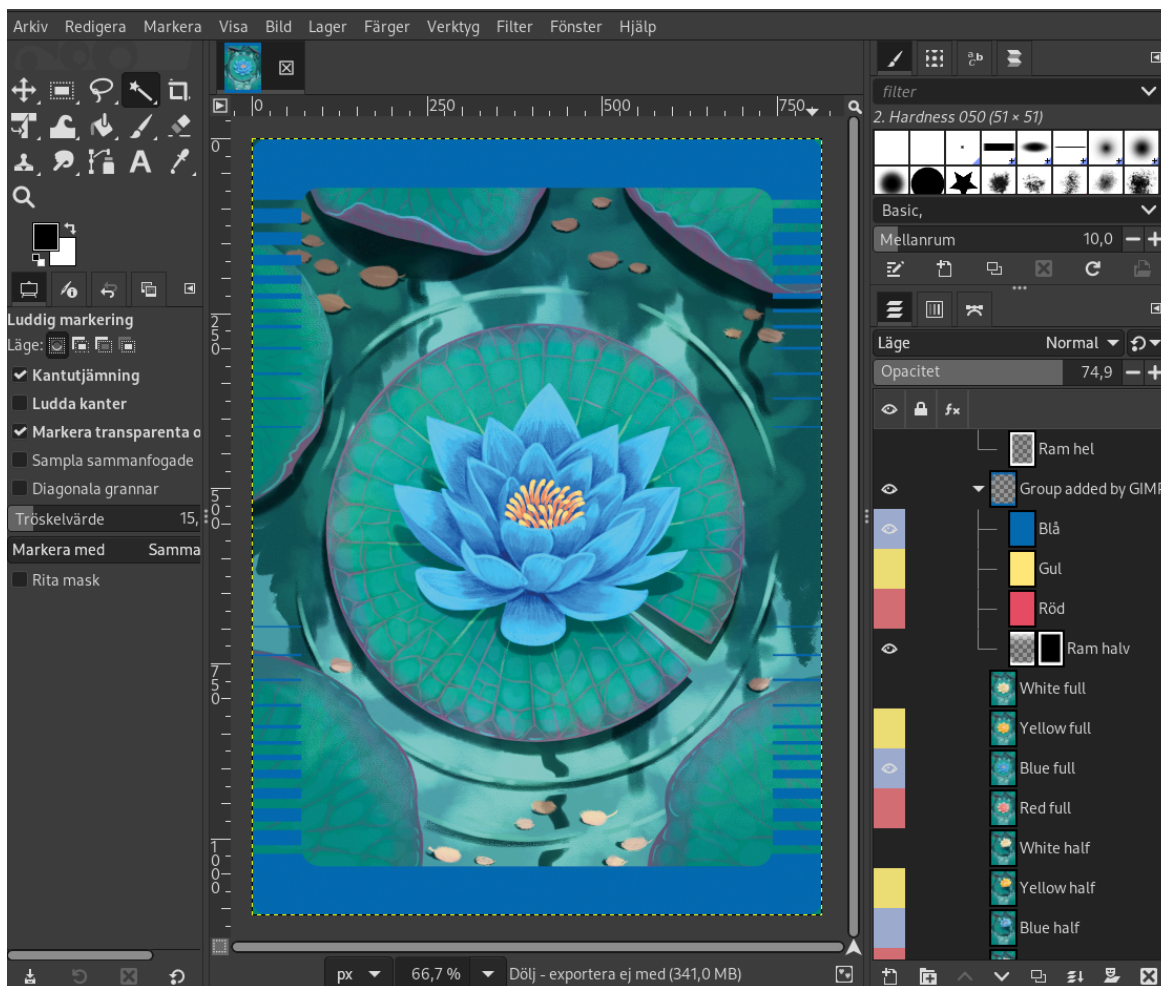
- Interoperability
- Ability to make changes



# GIMP

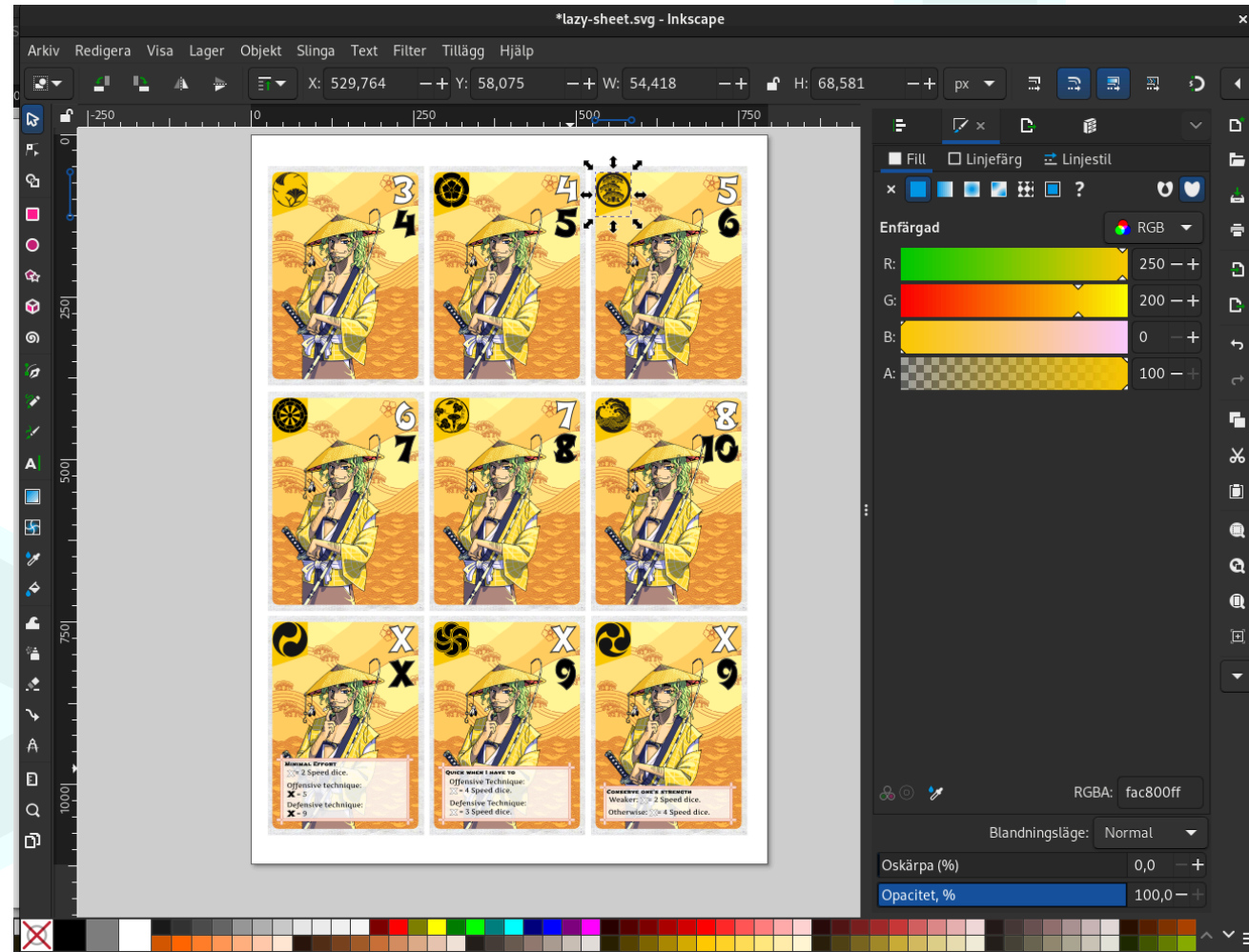
- Bitmap image editor
- Great for common actions (cropping, correcting levels etc.)
- Great support for reading Photoshop files
- Constantly improving







# Inkscape is still useful



# Wooden components





# Getting a game produced



# Working with a production company

Things get tricky...



# GENERAL REQUIREMENTS

Document preparation at a glance

**Your print-ready PDF/X files must be compliant with the following guidelines:**

- The content must be checked for errors before data delivery; an approval PDF serves only to check the data was transferred correctly to the printer print shop, not to check for errors of content.
- We only accept PDF files created with our **LF\_PDF job options** or in standard **PDF/X4:2008**.
- Image resolutions of 300 pixels per inch (dpi)
- Printing preferences **ISO coated\_v2\_300\_eci.icc (EC I)** (ECI <http://www.eci.org/de/downloads>)
- Black text must be set up in pure black (C: 0 %, M: 0 %, Y: 0 %, K: 100 %).
- At least 3 mm safety margin between important elements and the final format or the outline
- Bleed at least 3 mm
- A dieline must be placed for tableaus, stickers and box inserts

**Picture resolution:** All illustrations/images must have a resolution of at least **300 dpi**; line art and bitmaps with at least **1200 dpi**. Images can appear blurred or pixelated at lower resolutions.

**Colour:** All files must be delivered in CMYK format and the **ISO coated\_v2\_300\_eci.icc (ECI)** profile. Do **not use RGB format** for colour files as these do not print correctly. Please also note the maximum area coverage of **300 %**, as otherwise the products cannot be processed.

**Pure black:** If your texts are coloured black, please ensure that it is pure black (C: 0 %, M: 0 %, Y: 0 %, K: 100 %).

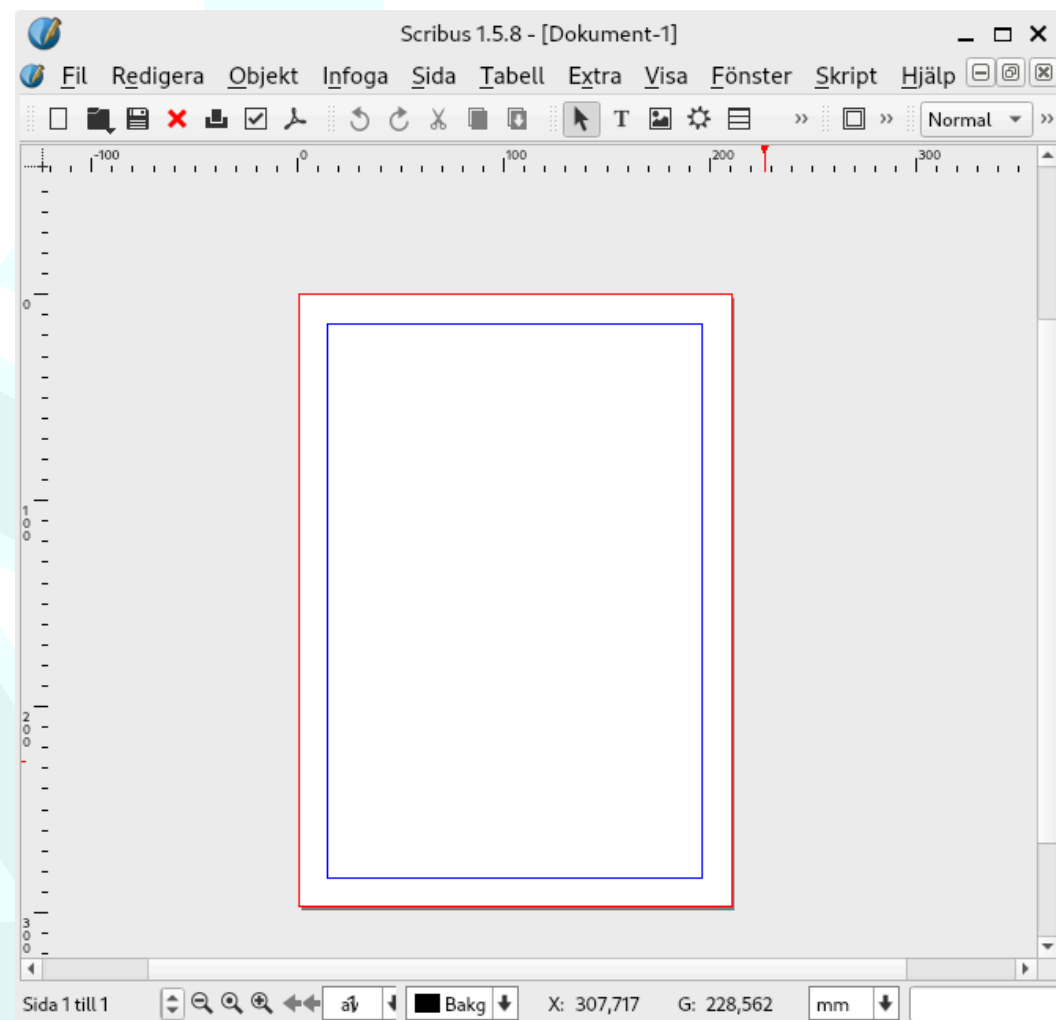
- Printer specifics
  - CMYK
  - Die lines
  - Bleed settings
  - Specific Pdf versions
- New file formats for wooden components

# What do we need now?



# Scribus

- Desktop publishing software
- Very competent
- Works great with SVGs
- Very good pdf export

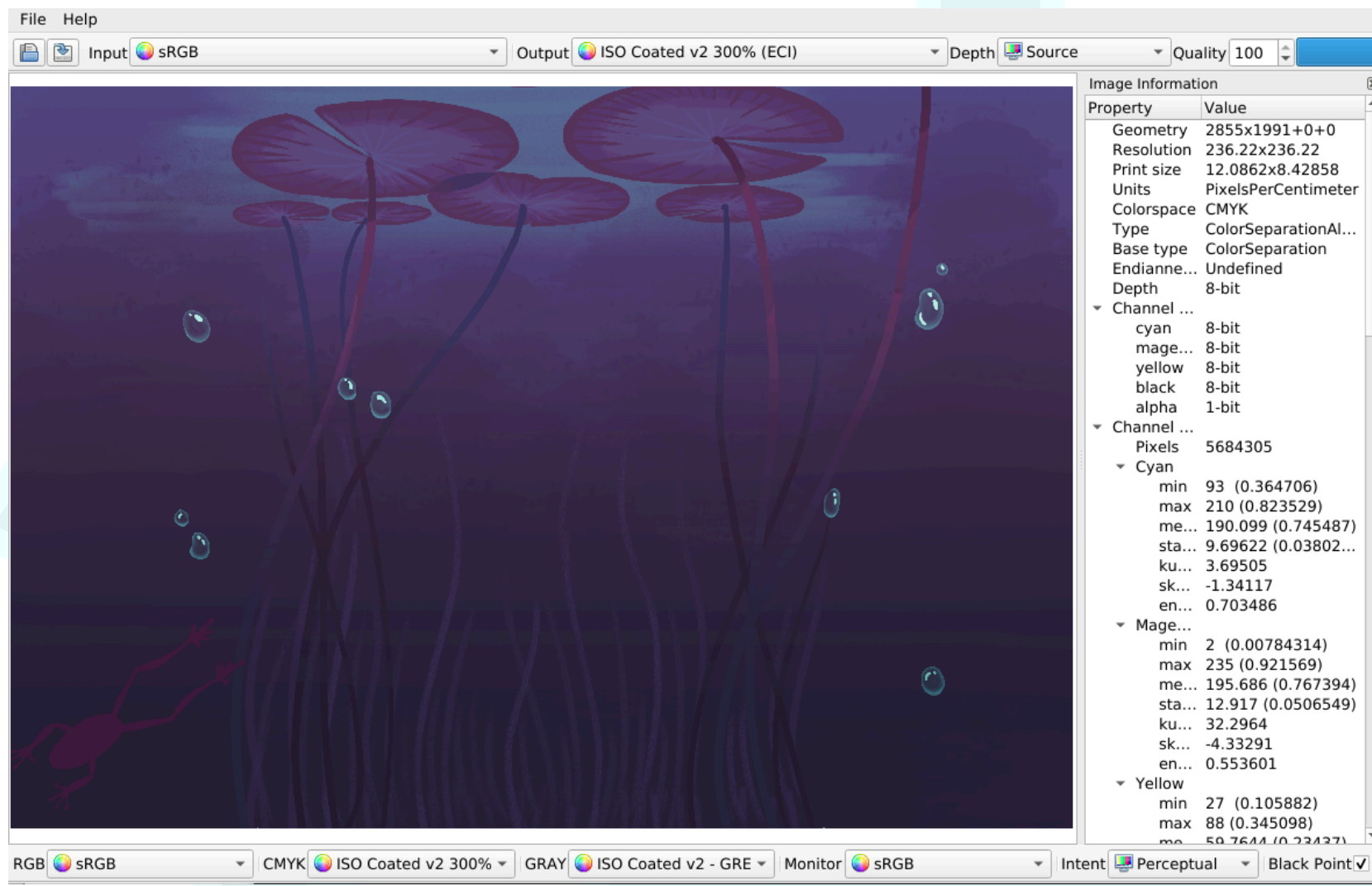




# CMYK

- ICC color profile
- Convert bitmap images & colors used in vector files

# Cyan



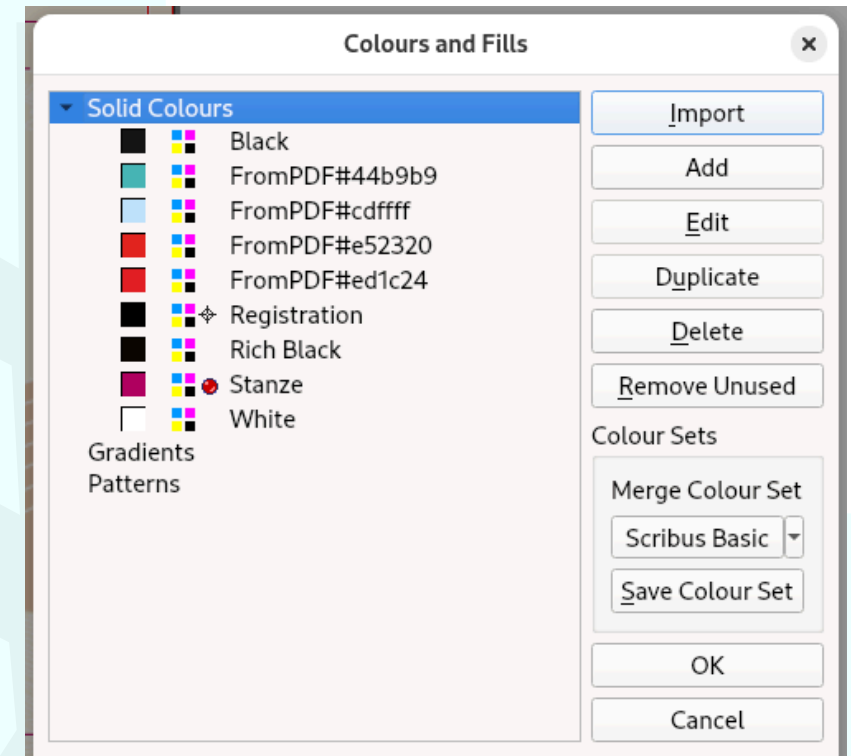
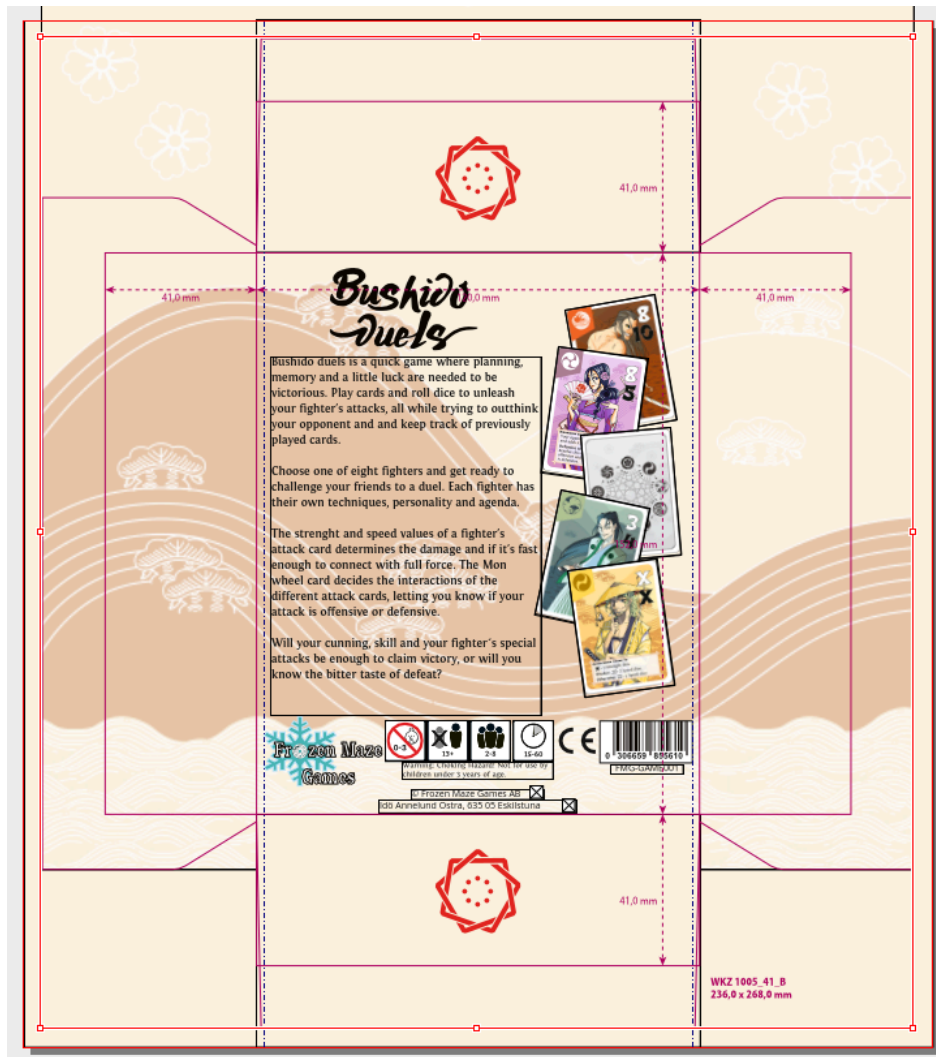
# Pdf version / variant

- Pdf 1.3: CMYK, Spot colors, Notes
- Pdf 1.4: 1.3 + Transparency

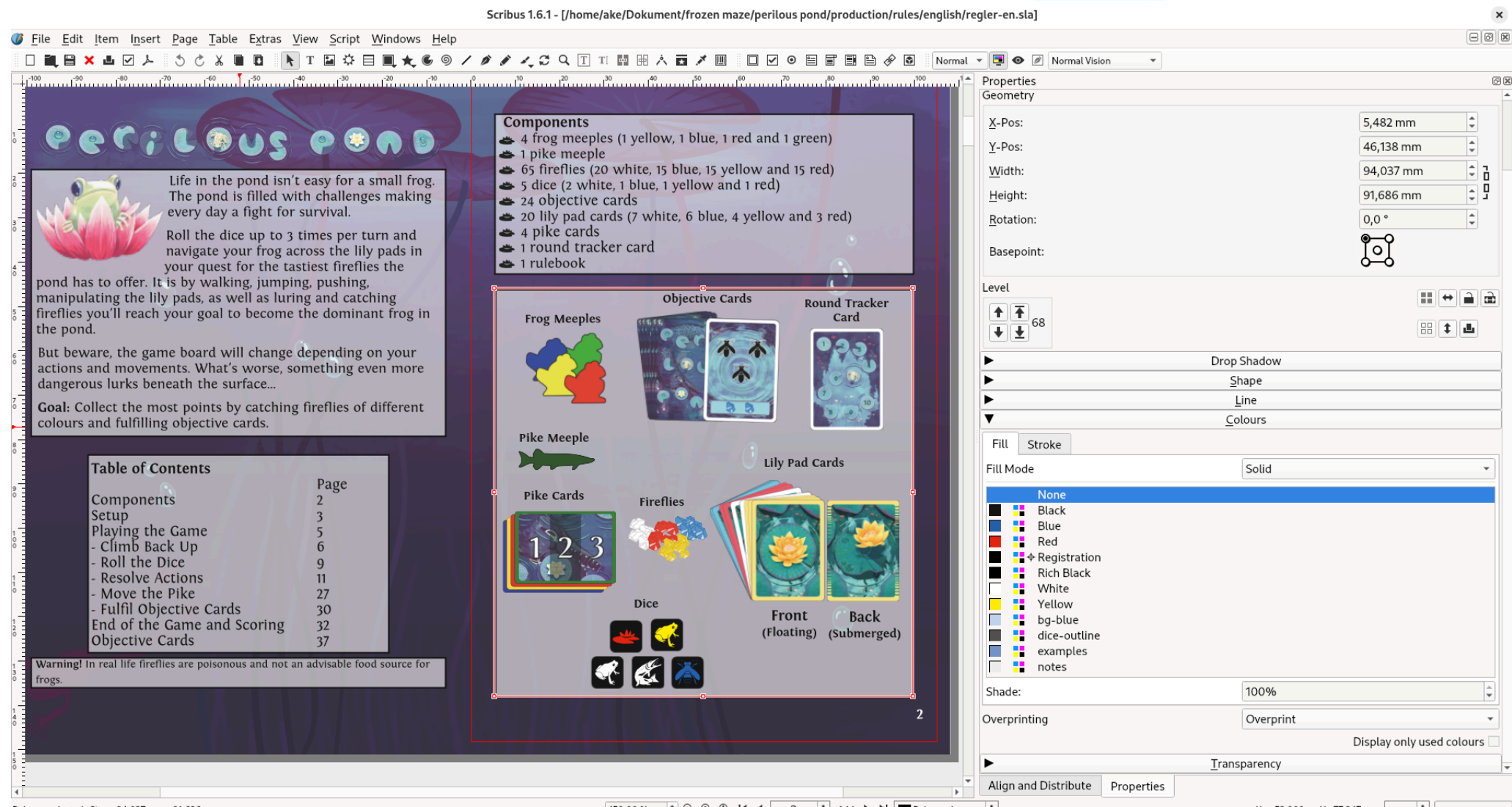
[...]

- Pdf/X-4: Embedded color profiles

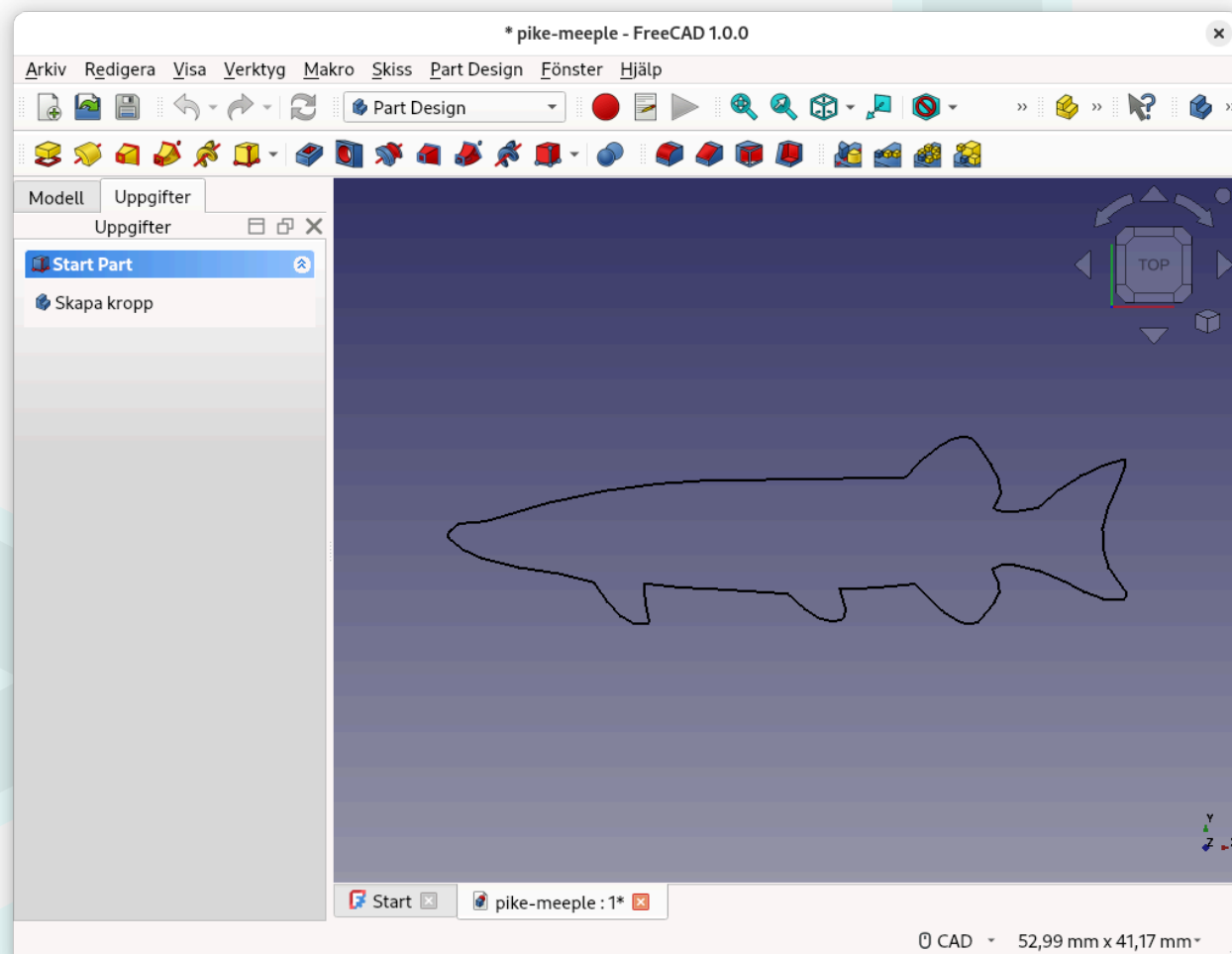
# Die-lines



# Over-print



# Wooden components



# Honorable mentions

## pdftools

```
$ pdftinfo presentation.pdf
```

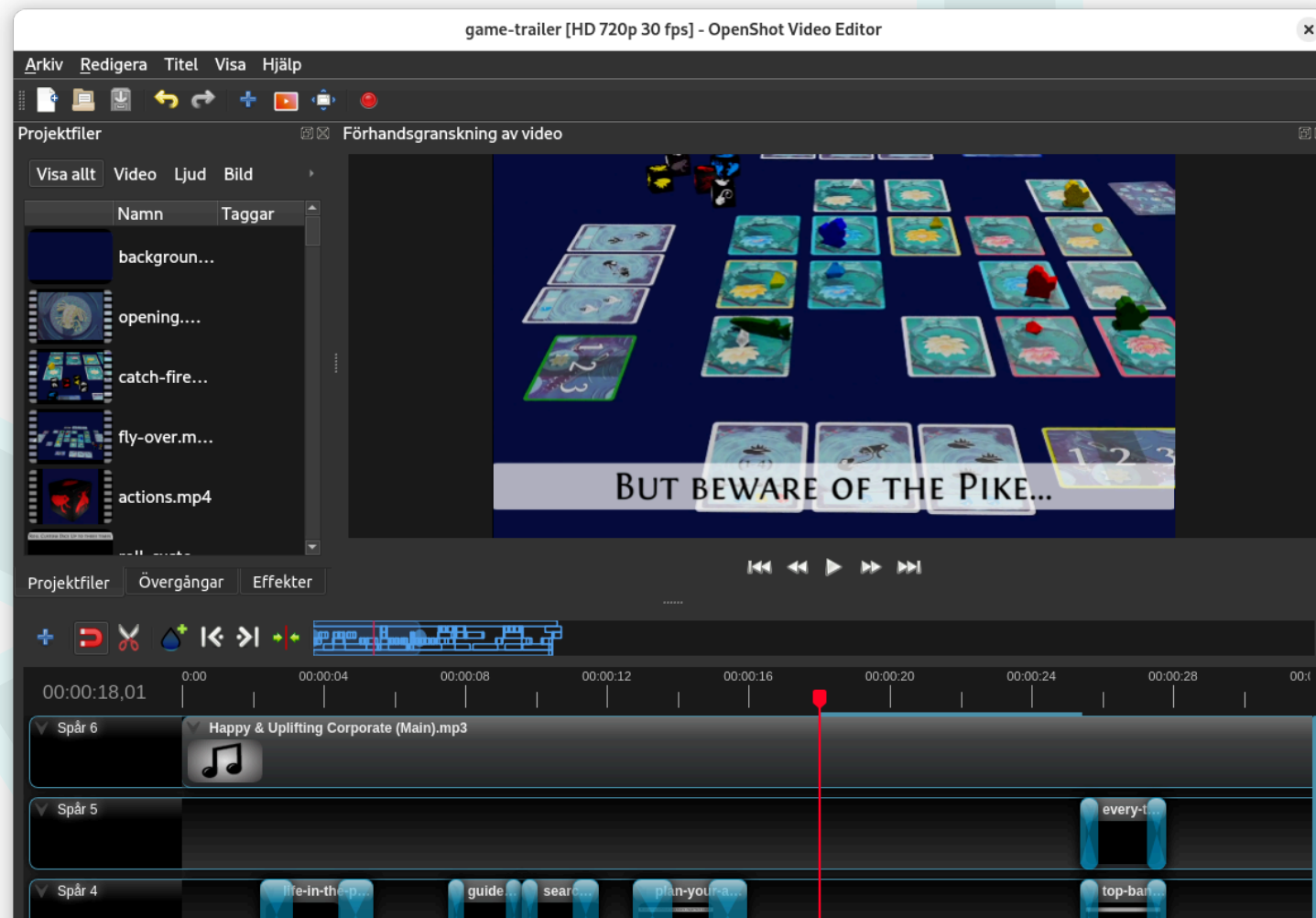
```
$ pdftimages -list presentation.pdf
```

# Blender





# Open Shot



# Isn't it awesome?



```

tree = ET.parse(argv[1])
root = tree.getroot()

black_things = [o for o in list(root.iter())
                 if o.attrib.get('PCOLOR') == 'Black'
                 or o.attrib.get('PCOLOR2') == 'Black']

without_overprint = [o for o in black_things
                     if o.attrib.get('doOverprint', '0') != '1']

coloured_things = [o for o in list(root.iter())
                   if o.attrib.get('PCOLOR') != 'Black'
                   and o.attrib.get('PCOLOR2') != 'Black']
coloured_overprint = [o for o in coloured_things
                      if o.attrib.get('doOverprint', '0') == '1']

# Remove text with no color from list
coloured_overprint = [o for o in coloured_overprint
                      if not (o.attrib.get('PTYPE', '0') == '4'
                              and 'PCOLOR' not in o.attrib
                              and 'PCOLOR2' not in o.attrib)]

lookup_ptype = {
    '2': 'Image',
    '4': 'Text',
    '5': 'Line',
    '6': 'Polygon',
    '8': 'Text on a Path',
    '7': 'Polyline',
    '9': 'LaTeX',
    '12': 'Unknown',
    '16': 'Unknown',

```

# Contributing back

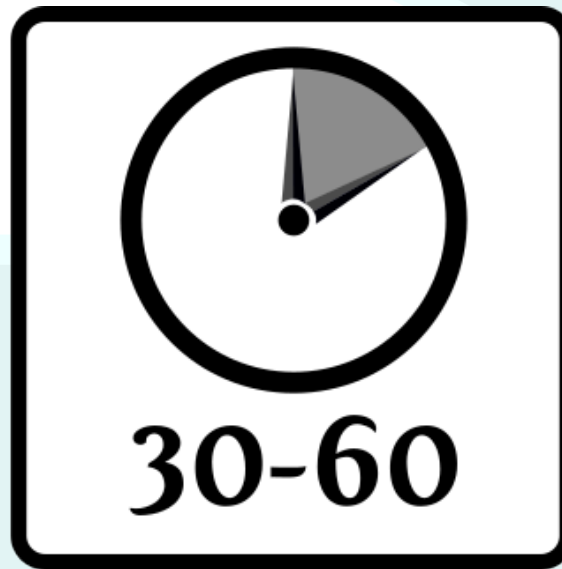
- Contributing €
- Translations



# Icon pack

<https://github.com/forslund/boardgame-icons>

Based on the original work of Sinister design



# Finally! It's over!

You've been a lovely audience

Questions?

