Post Krita 5 Projects

After Krita 5.0 we should think of where we want to take Krita next. We have a couple of well known problems:

- Text
- Workspaces, libraries and projects vs single files
- No Krita cloud for exchanging assets and making money from subscriptions
- No mobile UI
- Missing productivity options in our tools
- Ancient docker implementation (KDDockWidgets)
- Ancient MDI implementation (no alternative…)
- Qt6 (but that's not going to happen in 2022, I'm pretty sure of that…)

And so on. In this document, we're looking at one of our competitors to see what they have to offer to their users that our users don't have, especially in the areas mentioned above.

So…

Forget Photoshop, forget Corel Painter, forget Medibang, forget AzPainter, forget Procreate. The only other digital painting application that really tries to provide productivity features for painters, illustrators and comic book artists these days is Clip Studio, former Manga Studio.

Despite having a cluttered and often confusing interface, with, just like Krita, sometimes many ways to achieve the same thing, there are places where Clip absolutely beats us. After Krita 5.0, it's a good time to define some larger projects again, rank them and start beating Clip so some of my favourite artists can switch…

Clip comes in two versions, EX and PRO, with EX being the full version and PRO being the amateur version.

Text

Text features CSP has that we don't have:

- Clip supports vertical text with Ruby
- BUT NOT word wrap (though it can fit a frame, without keeping track of spaces or adding hyphenation)
- But not justified text
- AND NOT text fit to shape
- On-Canvas editing (though it's really confusing… For instance, after selecting italic only the next type char is italic)
- ...

Every text object is a layer. The SQLite database at the heart of the CPS file format stores the unformatted text with hard line endings in the layer table. Then there are two binary blobs containing formatting information (including plain text font names).

For us, Text is still going to be a really major project, demanding the final evaluation of whether SVG(2) + some extensions are going to hack it or not.

Images vs Books/Projects

- We have the comic book manager Python plugin, but this is much more integrated in CSP than in Krita. A book project can consist of several pages, and CSP is fairly good in managing those pages, handling per-project assets, like patterns, fills, references. But also graphical page layout with bleeds and everything, including generating pdf, epub and kindle artefacts.
- There are more useful presets than Krita has, for instance for creating webtoons. (We will have to make it possible to create those extremely large images in Krita as well.)

Front Page

- Clip's front page is a separate executable that stays around or can be closed, ours is wedged in between the regular UI elements.
  - Our approach takes up space we could use to make the impression of Krita more attractive.
- It is simultaneously busy and attractive
- It contains: a menu to the left with options for starting clip, managing projects and managing assets, accessing documentation and community, a cloud login on top, a recent files area, then a small news area with three lines of news, and a html based screen that shows news items.
  - This is a good example: we need to move from a pure file based approach (though that still needs to be possible) to showing a projects overview, since people are expecting that nowadays.

Cloud

- Subscriptions
- Buy and sell assets/materials/tutorials/python plugins
- Manage materials: this is utterly confusing in CSP. If you want to download a material from the cloud, you can select it in CSP, then click OK, which takes you to the startup screen, where you have to do nothing but tab back to CSP where you have to wait a bit, and then your material is available.
  - Note: materials can be a lot of things, like screentones, patterns, 3D models, reference images, brushes, tool presets.

Mobile UI

Note: The iOS version of Clip Studio is *completely* identical with the desktop version, down to traditional menus and dialog windows. If we could publish Krita as is on iOS, we would be in exactly the same position as Clip Studio.

That said, moving new components and the proposed new startup screen to QML to make the UI more mobile friendly would be a good idea.
Note that file handling uses a kind of project-view thingy, but CSP doesn't start with that on iOS

- Projects view vs file view
- Responsive mobile ui
- …

Tracing over 3D models

This is something we've tried before, and which gets requested a lot. Basically, CSP contains a mini-poser or mini-daz3d thing, with only grayscale rendering of 3D models supported, but it does support posing, camera changes and other manipulations. Informal enquiry suggests that nobody uses the technology to autotrace the model, people just trace it manually.

Flexible Panelling

- We make Krita for comic book artists, so a flexible panelling tool/shape would be a good idea. Clip's solution is workable, but quite cumbersome.
    - We should make it possible to have connected lines and curves and give those a similar style. Wolthera has already been collecting examples of comic frames that CSP cannot make.
        - **CAVEAT**: we should NOT try to reuse the old KOffice/Calligra shape connection code! Frames should probably be a new shape with its own tool that uses line and curve shapes internally.

Vectors

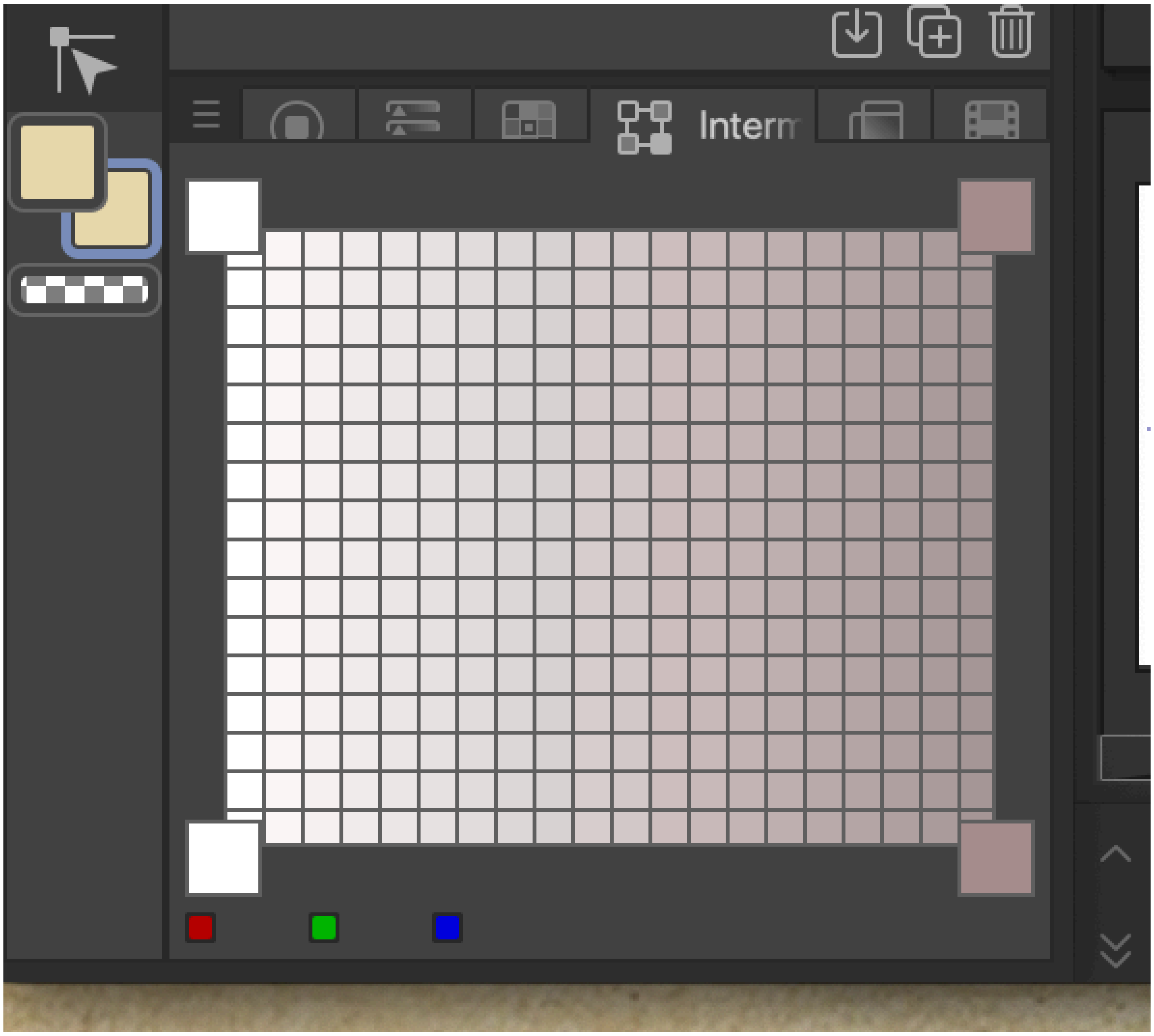
- Clip allows pasting SVG (but so do we)
- We really must have the ability to draw in freehand on a vector layer with a real brush, and have the result editable
- And we should probably also store this data in the undo for raster freehand strokes, so those become editable, too
- We need a much better calligraphy tool

Layer Management

- Layes can be tagged as Reference or Draft. Neither is taken into account when exporting an image, meaning you don't have to hide your sketch when exporting.
    - Our comic project management tool uses color labels for this, but that's a pain
    - Should we just add generic tagging to layers -- a few default tags, then a number of options that people can associate with a layer tag?

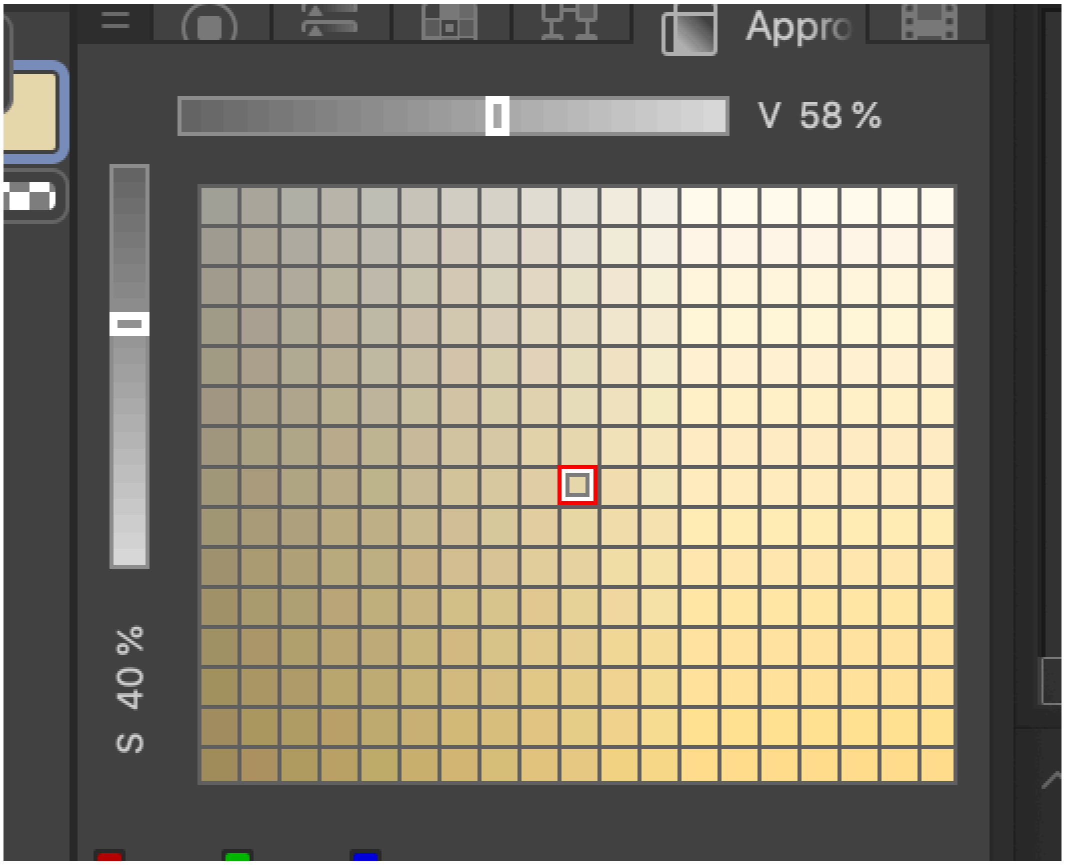
Color Selectors

- CSP has two interpolating color selectors that could be interesting to implement for us:



Interpolation between four colors -- note that clicking on the corners sets it to the current FG color, doesn't open a popup color selector. This is called Intermediate Color. (https://www.clip-studio.com/site/gd_en/csp/userguide/csp_userguide/540_color_plt/540_color_category_plt_chukanshoku.htm#XREF_91468)

The other one is called “Approximate Color” (https://www.clip-studio.com/site/gd_en/csp/userguide/csp_userguide/540_color_plt/540_color_category_plt_kinjishoku.htm)



Project Assets

There is a docker with per-project assets. Anything that can be loaded as a file can be in this docker, and be reused in different pages of a project.

New tools for easy filling

- Coloring between the lines is easy in CSP
- And this can also be done with gradients -- this is really important!

Various Notes

When working on this, Wolthera and I were frequently exasperated by Clip Studio, so here’s our list of complaints, but also various other observations:

- Does its own frickin’ window management, even on macOS, where it totally breaks down.
- Performance on a 2015 macbook pro is pretty abysmal
- Clip studio only provides CMYK soft-proofing. *But* this might be an advantage, since it forces people to use a proper workflow, instead of working in CMYK.
- Their settings dialog is not that different from ours, with a lot of options clearly being there for implementation reasons.
  - We probably should increase our default undo steps value from 30. CSP has 200 as default.
- Clips tools and subtools system is pretty complicated, though it makes saving a combination of tool options and brush options as one preset easy.
- CSP has a reference docker
- There is a separate docker that shows info about the current selection
- There is a separate docker that allows you to use a search term to show a subset of layers
- Files: project files are .cmc. These are SQLite files. Image files have the extension .clip and are binary files (in Manga Studio they also were SQLite files). https://github.com/rasensuihei/cliputils has a python file that can do some stuff, and which shows that the .clip files still are sqlite based.

Other Post 5.0 stuff?

For…

- Animation -- Krita is already light-years ahead of CSP, so our own plans should work for that.
- Resource Manager dialog and other resource stuff
- HDR -- post Qt6 port, if it’s still possible then.
- Shortcuts and conflicts with canvas input settings (inputrc being unreadable)
- Make touch configurable
- Extend Python api with undo/redo
- Put dockers in a settings page, instead of a long list
- Recorder/Actions
- Create vector libraries
- Text on path
  - Show blending mode, opacity next to the layer name in the layer docker, decrease indent?
- Improving clipping mask situation:
  - Loading/saving these from PSD
  - Figuring out a number of usability-issues regarding them.
- Eraser virtual stylli stuff.
- Talk to drawpile folks about their server.

Intel:

- Add benchmarks in the UI
- Check thread congestion
- Check only using P-Cores
- Implement jpeg-xl
- Machine learning?