

Advanced Toolbars

Outline of the current issue

Users find qds difficult to use because it hard to find the correct tool or property they need. Designers find qds unappealing because the UI is outdated, not contemporary and does not follow standard design patterns of other design tools.

Reasons:

- Toolbars are cluttered
- Property Panels are cluttered
- Icons are unclear in meaning
- Properties are unclear in order and meaning
- Tools are not displayed in context of tasks
- Qds lacks basic manipulation tools
- UI looks outdated and unappealing
- UI does not follow standard pattern of other Design Tools

General Examples of Design Tool Standards

Top Menus

Photoshop - Contextual Bar changes with selected layer item



Illustrator - Contextual Bar changes with selected layer item



Figma - Has the toolbar on the top, very simple design, easy to use

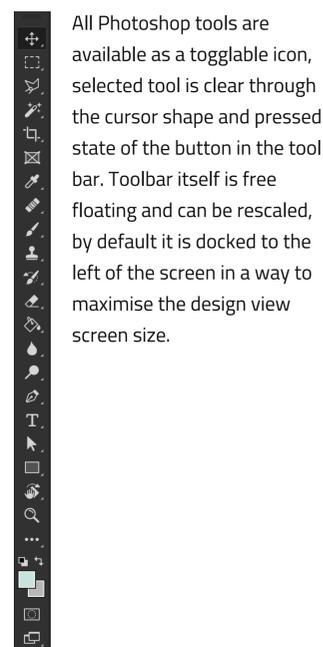


Sketch - Not Contextual but very clear design, lots of negative space, labels for added clarity



Toolbar Examples

Photoshop

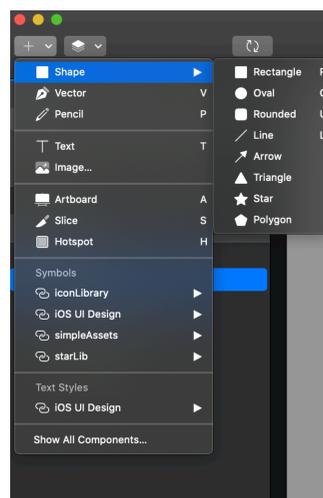


Illustrator



Much the same as Photoshop, like which it also keeps the color controls for foreground and background in the tool palette.

Sketch



Uses a nested menu to reach the tool palettes, clear icons plus both lables and hotkeys for extra discoverability.

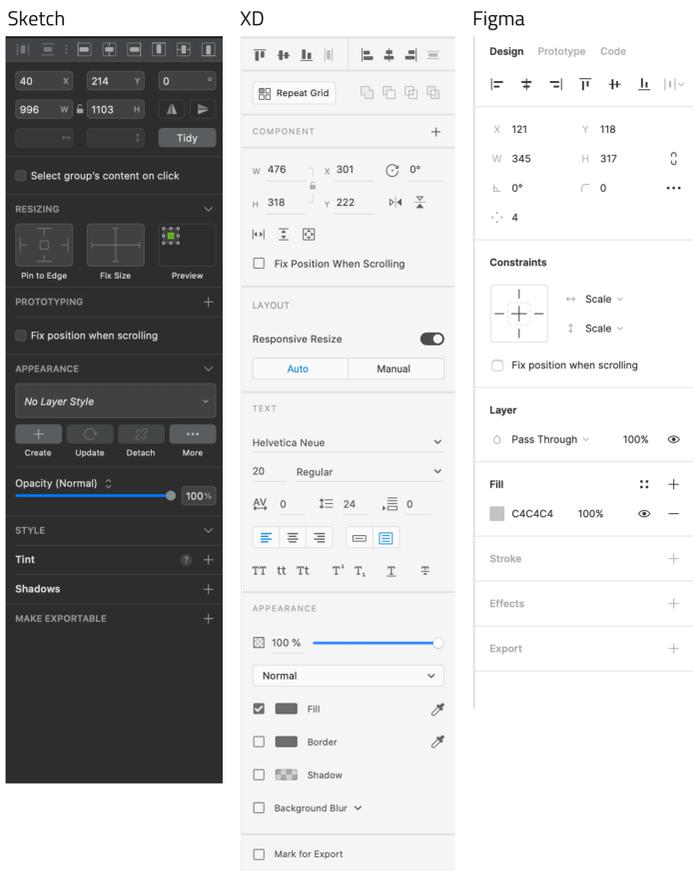
XD



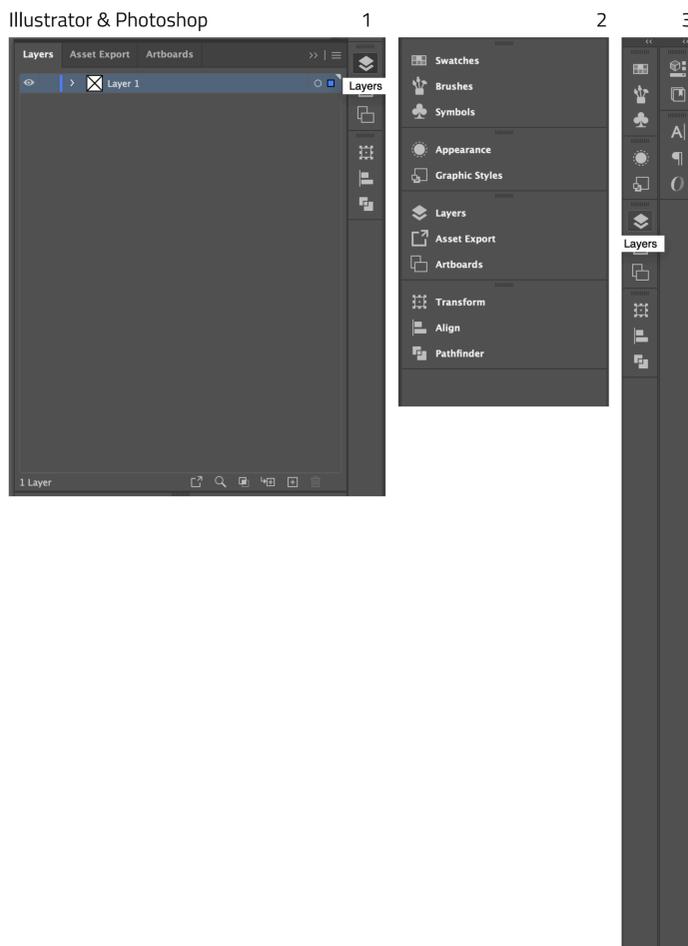
Super simple minimal design, clear icons and clear use of an interaction color to display the active tool.

Property Panels

Most modern design tools opt for very clean property panels, grouping the important properties together and using good icons with lots of negative space



Adobe has arguably the most advanced property panels, when dragged to the narrowest point it collapses into a slim bar with icons only [3]. Dragging it a bit wider reveals labels for each icon [2] and clicking on one will toggle the property panels, which are all docable panels of their own [1].



Advanced Toolbars

Deconstructing a single tool

In order to move to a new toolbar design in qtds we will have to analyse each of the individual tools required in each toolbar, this is split logically into the 2D and 3D tool, as each should have their own "Mode Sensitive" Toolbar with the most important tools available.

The reason we need to do this is there may be functionality missing that is required, this extra functionality may be solvable purely as Tooling or it may require changes to the Framework, indentifying the Framework issues early will be critical to the sucess, even better is to find appropriate solutions that do not require Framework changes.

For the 2D tools we need to consider having in the toolbar I want to make a non exclusive list of a few and then look in detail at a small subset of these. (Highlighted in Green)

- Geometry : Selection, Position, **Rotation**, Scale
- Shapes : **Rectangles**, Ovals, Arcs, Lines, Polygons
- Objects : **Text**, Quick Controls, Pen Paths

Example One - Adding and Editing Text - Qds VS Photoshop

Photoshop

Tool that can be toggled on and off - good tooltip + example



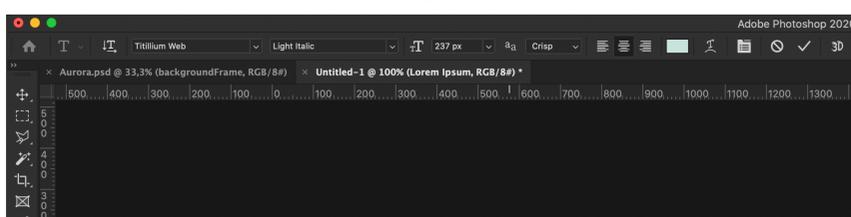
Bounding Box is Clean and doesn't interfere with the type



Editing is done on the actual font / size / color of the text



Most Import Text Properties - automatically propogated to top tool bar

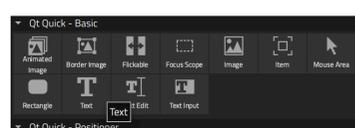


More Properties can be "popped out" in their own dockable panel

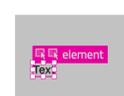


QtDs

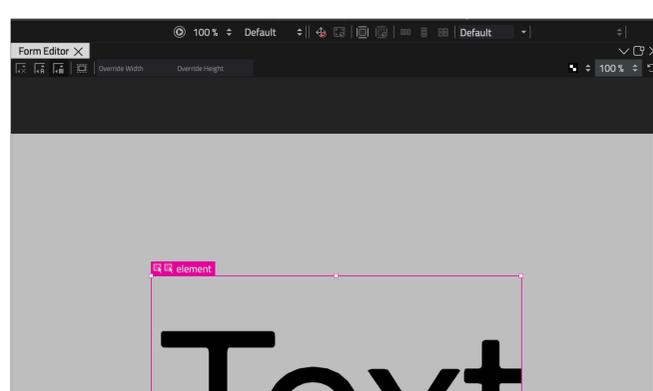
Text is not a tool that can be toggled on or off, but an object that must be dragged into the scene (non-standard and unintuitive) - Tooltip is only a repition of the name. Grouping is based on the Quick Group it comes from and not the group of tools you use the most (as is the common pattern in other design tools).



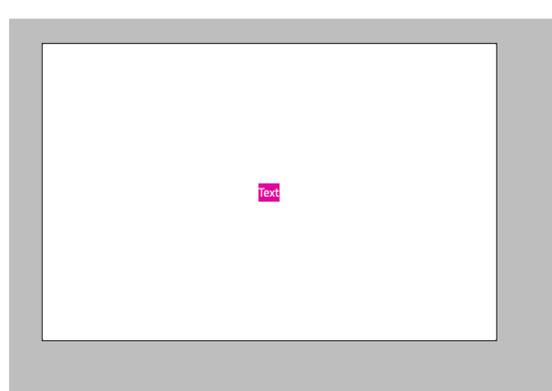
Item bounding box is cluttered and distracting



Top Menu is not contextual to the most import properties of the current item



Form Editor Text Edit - Not editing the actual text, small, unclear, poor contrast.



Text Properties - Disorganised, difficult to find the relevant properties

Importance 0 - 10

So many of these items are of little to no value to the designer

So many of these items are very important to the designer but buried at the bottom of the property panel

Property	Importance
Text	10
Wrap mode	2
Elide	2
Maximum line count	0
Alignment	8
Format	0
Render type	0
Font size mode	0
Minimum size	0
Line height	7
Line height mode	1
Text Color	10
Style Color	1
Font	10
Size	10
Font style	7
Font capitalization	7
Font weight	7
Style name	9
Style	1
Spacing	0,00 Word / 4,92 Letter
Hinting preference	8
Padding	5
Vertical	5
Horizontal	5
Padding	5

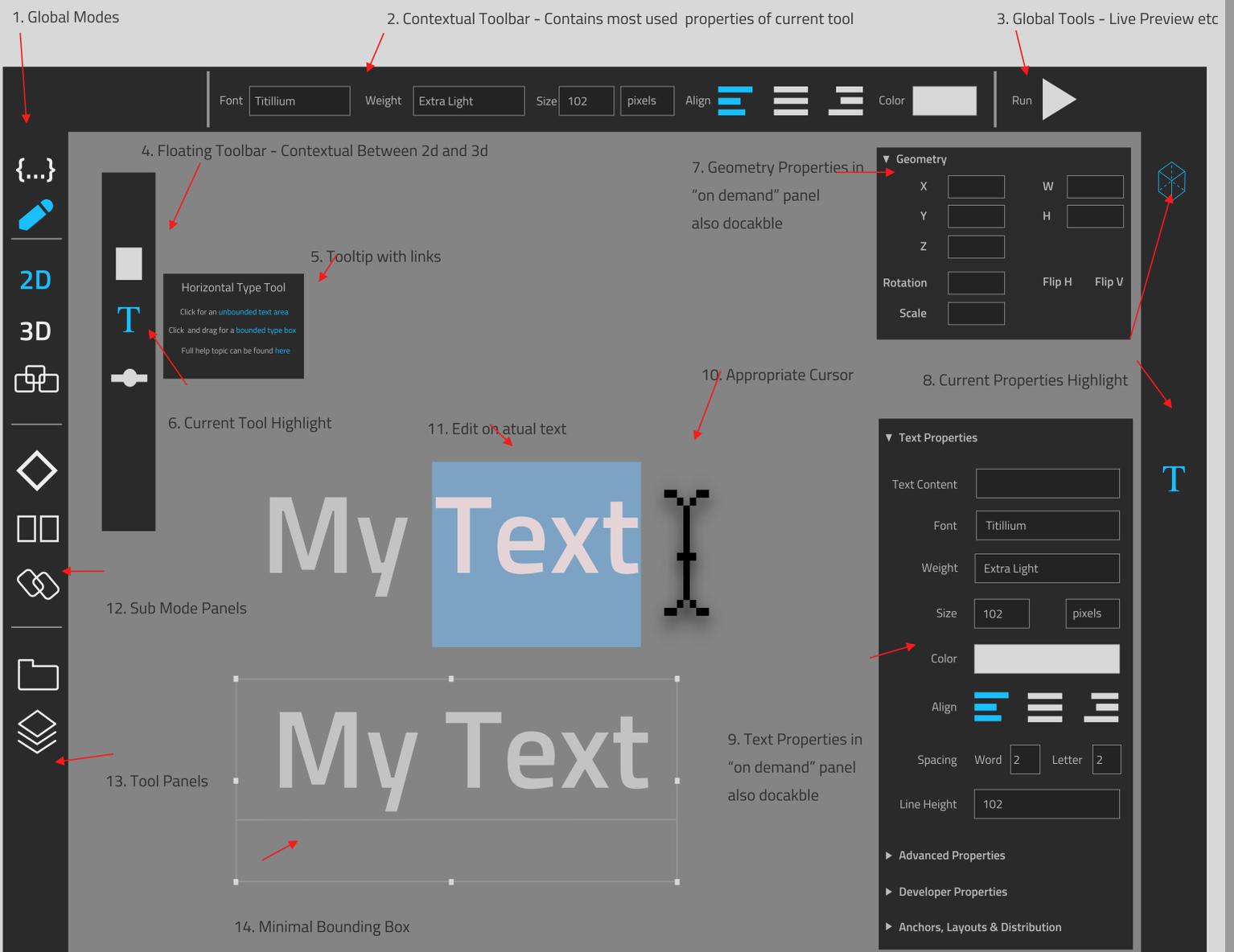
Advanced Toolbars

Text Tool - Concept Sketch

Sketching out this simple concept allows us to identify the UI areas where change are required, what the key features of these toolbars and panels are, and reveals some of the relationships between the UI components.

What it doesn't do is tell us anything about the final design specifications, measurements, functionality, color themes, icon systems or importantly, development effort required to fix all of these things.

From these Sketches of an idealised Design System, in the context of one specific tool. I want to try and map out the dependencies that it would create.

Text Editing Tool - Concept Sketch*Text Editing Tool - Component List*

1. Global Modes

Global Mode Toolbar is essentially reserved for switching between the Edit and Design Modes. Functionality already exists in current Mode Bar but called Edit and Design Mode and is grouped with the Home Page, Debug, Build, Setting and Help. We should rename Edit to Code, redesign the icon to a clearer code icon and remove all the other options into a file menu.

2. Contextual Toolbars

Entirely new Functionality that would change all the center menu items to the most used properties from the currently selected tool. Requires Development Effort and selection of all most important properties for the tool sets. Some properties will be common between tools and can therefore be grouped.

3. Global Tools

Some items are required in all design views, live preview launch, zoom factor, help access, canvas properties, etc. These should be arranged in the spaces on both the left and right of the context bar.

4. Floating Toolbars

Entirely new Functionality that would introduce a new toolbar with an entirely new way of interaction with Qt Quick Types. Contextual Toolbar that, for this use case, would select an text object, change the cursor, and then draw the new text object on the canvas following the users drawn path.

5. Tooltips

New Tooltips should be built in qml, allowing for easy adding of links to help topics and in the future adding something like gif animations or videos to demo the tool usage.

6. Current Tool Highlight

New Icons would be required, we would need to fix with a new icon standard, either pngs, svgs, icon font or painted.

7. Geometry Properties in On Demand Dockable Panel

A new property panel that would separate the geometry properties from the individual property sheets, as the geometry properties are a common set shared by virtually all quick objects moving these into a separate panel makes a lot more sense. Would require moving scale and rotation from advanced properties tab (where they make no sense to be anyway).

8. Current Property Highlight.

Entirely new Functionality that would introduce a new toolbar with the property panels accessible via a draggable or toggable menu, would require new icons, conforming to the new icon standard. Would require re-organising and prioritising items properties.

9. On demand Text Properties Panel

Re-Organised the panel so only the important properties for the designer appear to start with, tentatively introduce a concept of having a developer properties section, reserved for the many properties a developer might need but the designers not.

10. Appropriate Cursor

Most Design Tools use the cursor shape to indicate the currently selected tool, in the case of the text tool this should be the text cursor.

11. Edit on Actual Text

Entirely new Functionality that would allow users to edit the actual text as it appears in the form editor. May require significant development work on the form editor itself.

12. Sub Mode Panels

Context Sensitive Mode Switcher for the Sub Modes of the Design Mode. Would include switching between 2D, 3D and Flow Mode.

13. Library and Navigator

New method of launching the navigator / project browser and library panels.

14. Minimal Bounding Box

New Minimal Bounding Box with baseline integrated for text objects.