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 Designers find qds unappealing because the UI is outdated, not contemporary and does not follow standard design patterns of other design tools.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UI looks outdated and unappealing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

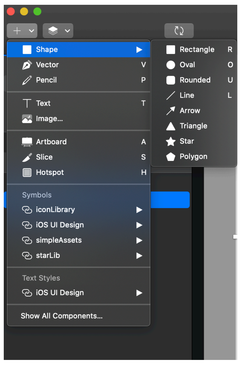
Photoshop

All Photoshop tools are available as a togglable icon, selected tool is clear through the cursor shape and pressed state of the button in the tool bar. Toolbar itself is free floating and can be rescaled, by default it is docked to the left of the screen in a way to maximise the design view screen size.

Illustrator

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

Sketch



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

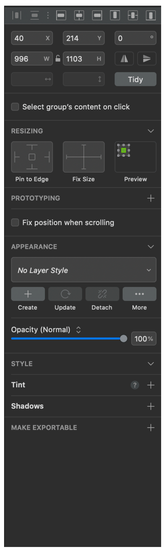
XD

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

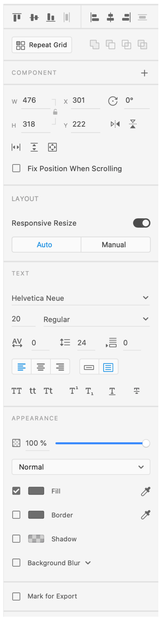
Property Panels

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

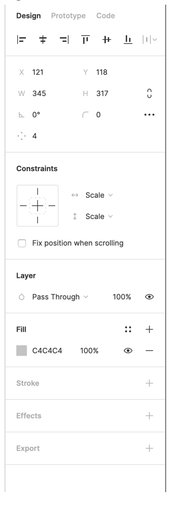
Sketch



XD



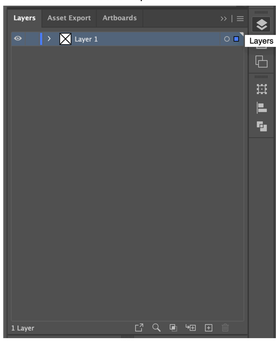
Figma



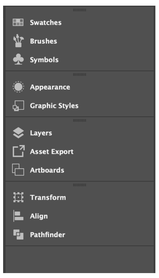
Adobe has arguably the most advanced property panels, when dragged to the narrowest point it collapses into a slim bar with icons only [3]. Dragging it a bit wider reveals labels for each icon [2] and clicking on one will toggle the property panels, which are all docable panels of their own [1].

Illustrator & Photoshop

1



2



3

Advanced Toolbars

Deconstucting 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 functinality 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

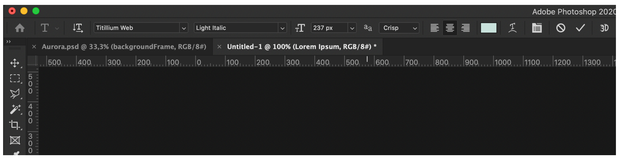
Photoshop

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

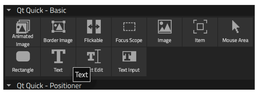
Most Import Text Properties - automatically propgated to top tool bar



More Properties can be “popped out” in their own dockable panel

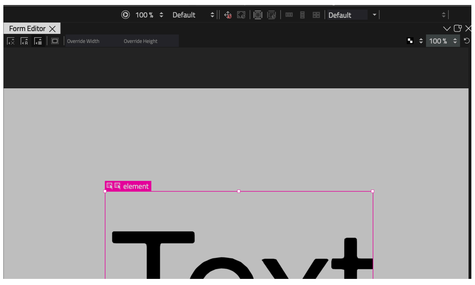
Qtds

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).

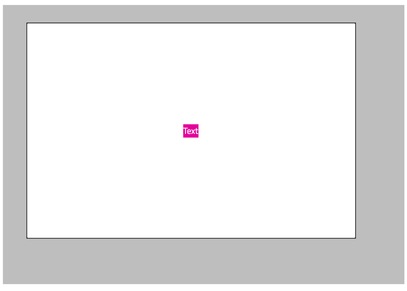


Item bounding box is cluttered and distracting

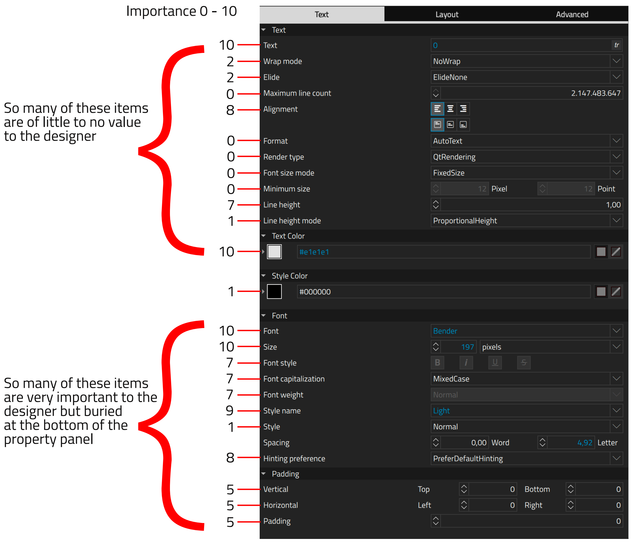
Top Menu is not contextual to the most import properties of the current item



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



Text Properties - Disorganised, difficult to find the relevant properties



Advanced Toolbars

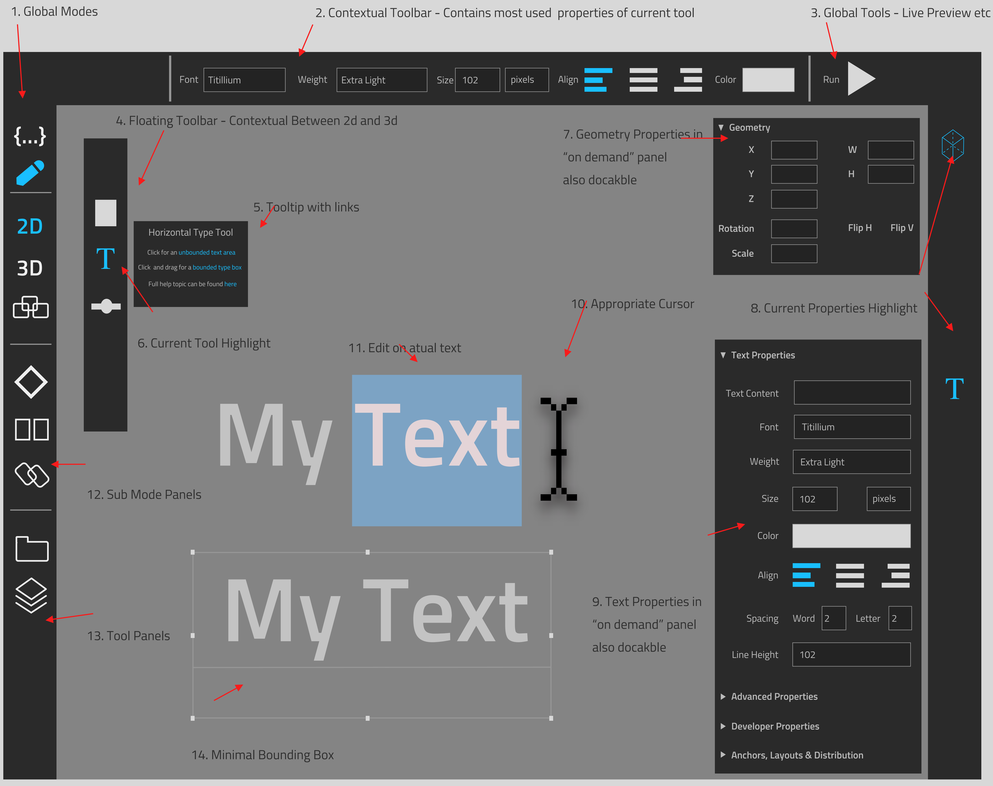
Text Tool - Concept Sketch

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, funtionality, 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 propoerties 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 follwing 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 seperate the geometry properties from the individual property sheets, as the geometry properties are a common set dshared by virtually all quick objects moving these into a seperate panel makes a lot more sense. Would require moving scale and rotation from advanced proprties 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 dragable or togglable menu, would require new icons, conforming to the new icon standard. Would require re-organising and prioriting 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. Approriate 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 signficant 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 intergrated for text objects.

Advanced Toolbars

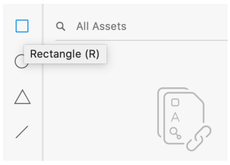
Deconstucting a single tool

Adding shapes should be simple and intuative, all other tools have shapes that can be drawn on directly into the workspace. Our method of using Libraries, having duplicate shapes with different properties, not having certain shapes at all (ovals, lines, n-sided polys) and having to drag annd drop shapes from a panel to the form editor is behind the times in terms of what designers expect.

Example One - Adding and Editing Rectangles - Qds VS XD

XD

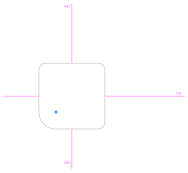
Tool that can be toggled on and off - simple tooltip (works with icon) + hotkey



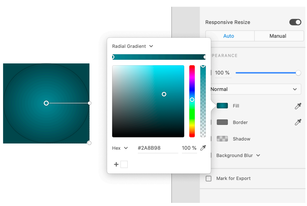
Cursors are conntextual to the tool (not in screenshot) rectangles can be drawn on intuitively.

Property Panel is clean and simple

Bounding box has interactive tool handles for scale, rotation and rounding.



Intuative gradient settings with “on shape: tool handles

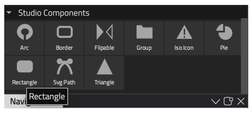


Qtds

Qt Design Studio Provides 2 different rectangles, one with very simple properties and one with a lot more complex and useful ones. By default you only see the simple rect, to get the one with individual corner rounding (essential) you have to know about imports and which library you need.

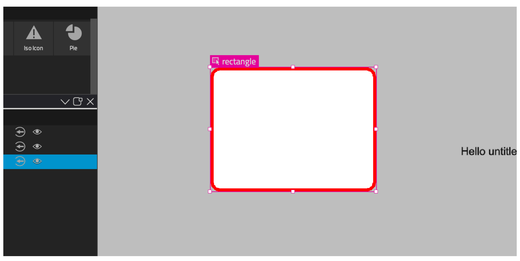
Basic Rectangle is avaialble by default but lacks important properties.

Studio Rectangle has the extra properties but you have to know which import you need to get it



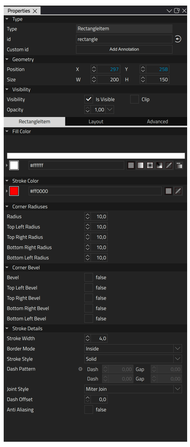
Rectangle is not a togglable tool but an object you have to drag into your scene, a different interaction patter than every other design tool. Cursor is not contextual and the whole process is uintuative

Bounding box lacks common features of other tools, no rotate cursor availalble from the corners and no control points for corner radius.

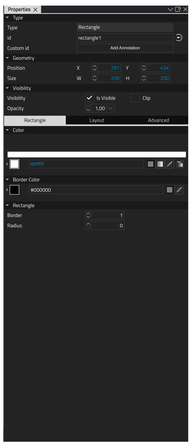


Basic Rectagle lacks properties designers find useful, advanced Rectangle has more useful properties but lacks discoverability.

Studio Rectangle



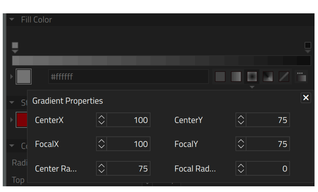
Basic Rectangle



Adding Gradients are essential for designers, it is one of the cornerstones of modern visual design. Our gradient UX is poor because: The gradient options are different between rectangle types, difficult to discover and have unituative controls.

Gradient Controls are not discoverable

Unintuative Gradient Controls



Advanced Toolbars

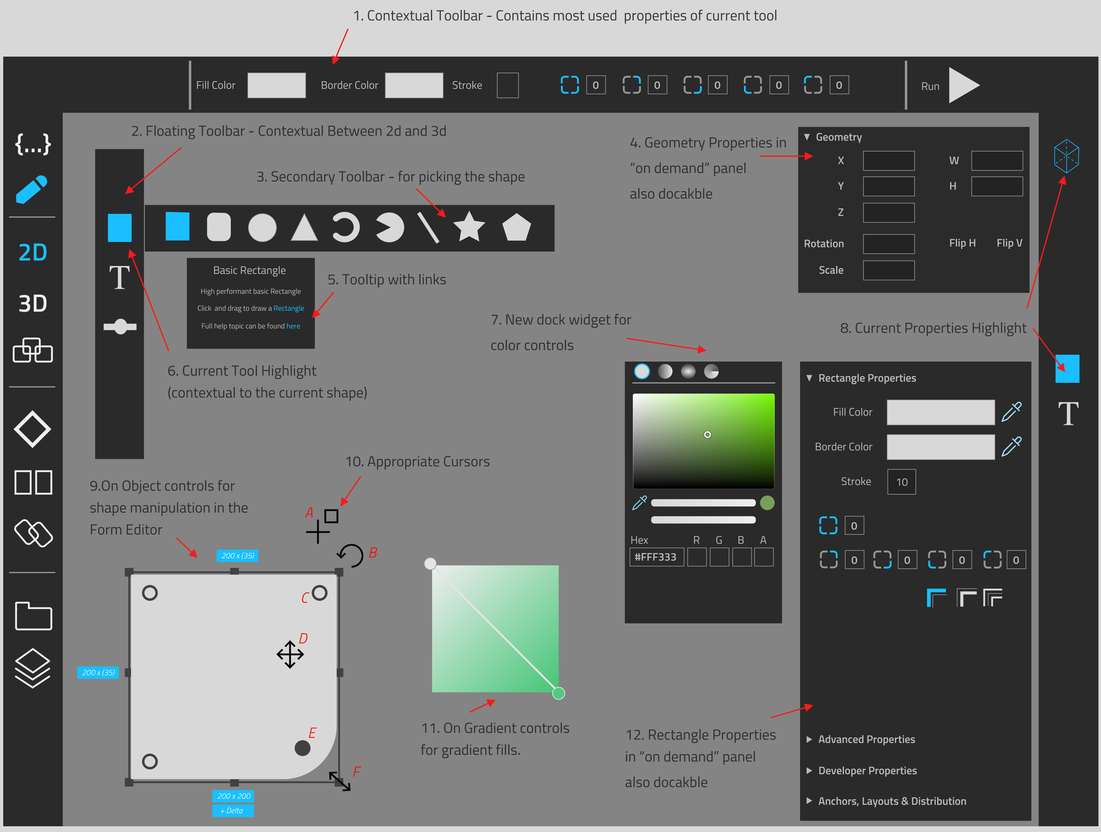
Rectangle Tool - Concept Sketch

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, funtionality, 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.

Rectangle Drawing Tool - Concept Sketch



Rectangle - Component List

1. 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 propoerties will be common between tools and can therefore be grouped.

2. 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 follwing the users drawn path.

3. Secondary Floating Toolbars

Some tools will require sub-toolbars for the different options within the tool, shapes, controls, 3d tools etc.

4. Geometry Properties in On Demand Dockable Panel

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

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. New Dock Widget for Color Controls

Move the color properties into a new dockable panel, improve the settings and controls for gradient pickers, add a eyedrop tool for picking colors from the screen.

8. Current Property Highlight.

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

9 &10. On Object controls and cursors

Introduce cursors, handles and lablels for improved "on object" editing, including scale, rotate and move cursors, handles for corner radius, labels for width, height, x, and y + deltas when moved.

11. On Object Gradient Controls

Intuative handles for controlling gradients with the form editor object itself.

12. Dockable Property Panel

Introduce modular approach to property panels with dockable panels for object properties and new responsive and customisable layout for the properties in the panel.

Advanced Toolbars

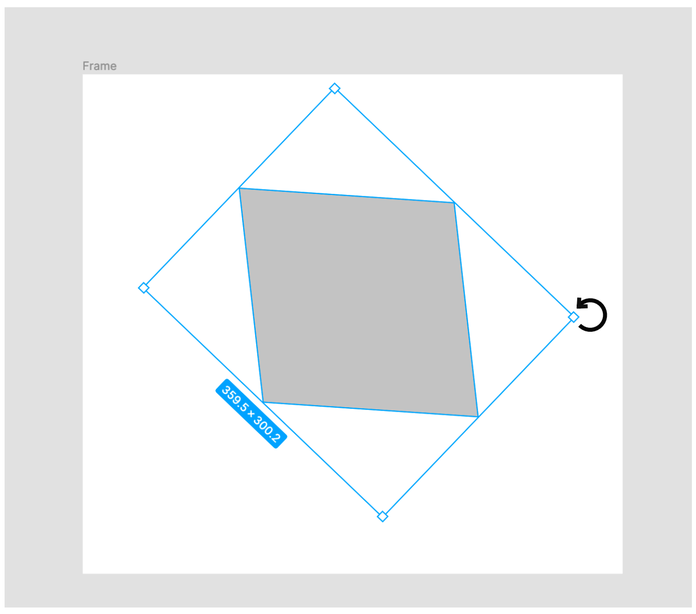
Deconstucting a single tool

The Rotation Tool is a good example of something that contains both low hanging fruit and more complicated issues. One simple improvement would just be to move the property into the Geometry section, making it more consistent with other design tools and more discoverable for our users. Having a discrete rotation tool or a cursor handle on the corners of the object would also improve the usability of Qds. The issue with being able to manipulate (re-scale, resize) the object after it has been rotated is more complex but also something we should do as all other design tools offer this functionality.

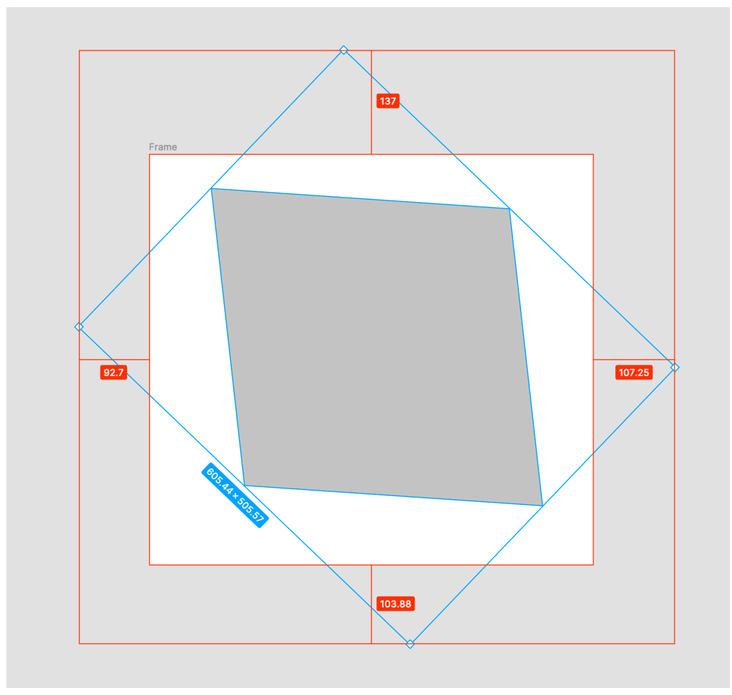
Example One - Rotating Objects - Qds VS Figma

Figma

Object can be rotated via a cursor that appears on the corner when the user hovers over it.

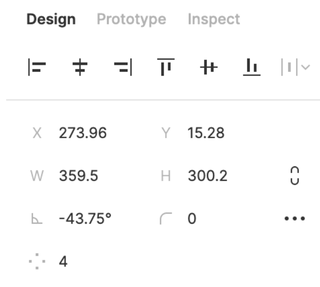


Rotated items can be resized via the corner handles after being rotated, resize guides appear when you do so.



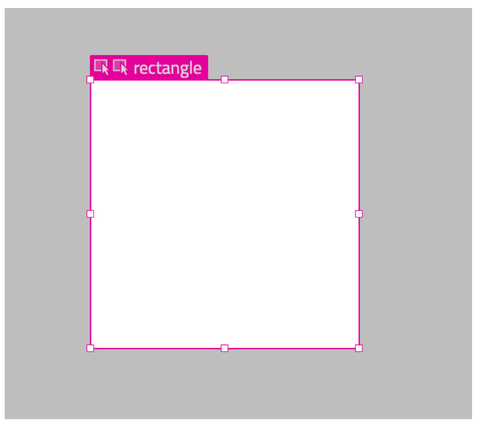
Property Panel is clean and simple,

Rotation is logically a property of the geometry of the object.

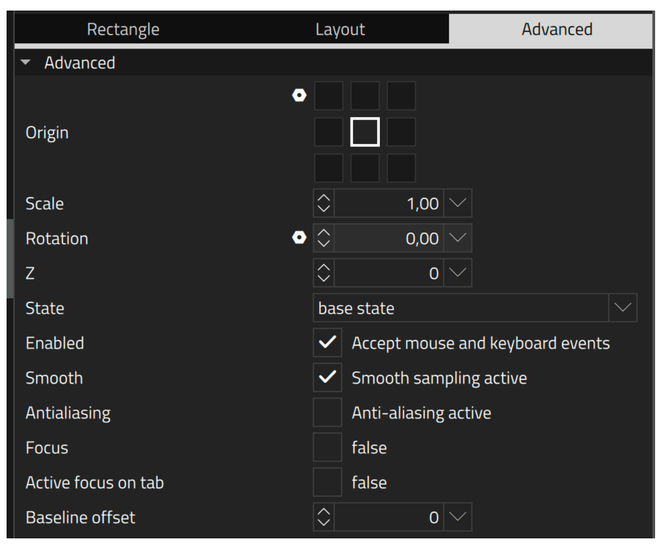


Qtds

No control handles on the object in the form editor

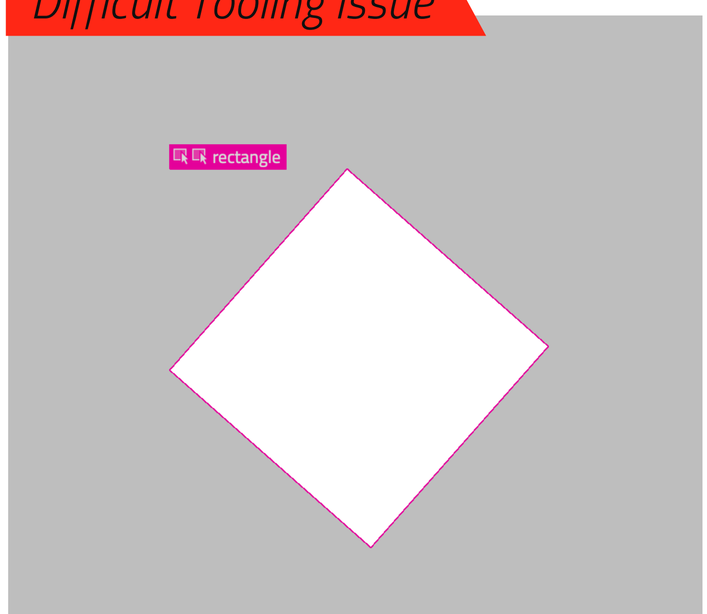


Rotation Property is not part of object geometry but buried in the advanced property panel, which is very un-intuative.



Once an object is rotated it can no longer be re-sized via the control handles at all.

Difficult Tooling Issue



Advanced Toolbars

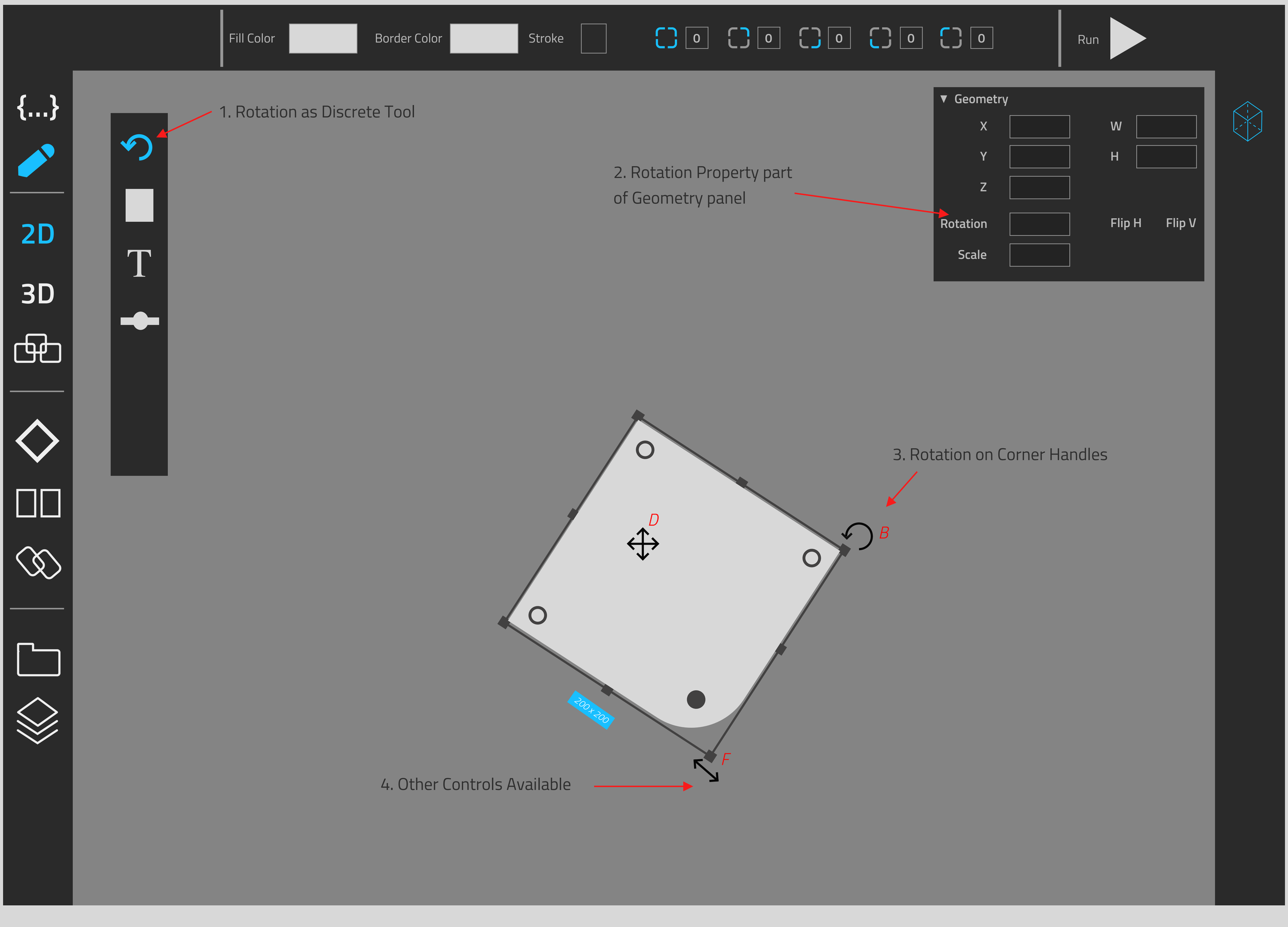
Rotation Tool - Concept Sketch

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, funtionality, 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



Rotation Tool - Component List

1. Discrete Rotation Tool

Many tools have a discrete rotation tool that can be used on any object in the form editor.

2. Rotation Property in the Geometry Panel

Rotation is a property of the objects geometry, moving our property here is low hanging fruit.

3. Rotation is available from hovering over the corner

Objects can be rotated via a control handle when you hover on the corner of the object in the form editor.

4. All other manipulation tools are availalable on a rotated object.

Even after a tool has been rotated all other handle tools are available.