



2.8 DESIGN PROPOSAL.

DESIGNING AN EXCELLENT INTER-
ACTION MODEL FOR BLENDER 2.8

WRITTEN BY
WILLIAM REYNISH



ABOUT THE AUTHOR



I'm William Reynish, a long time user of Blender since 2001. I have worked as an animator using Blender, as well as teaching it in schools. I've directed short films and commercials, each time using Blender, where I end up teaching it to fellow animators. I recognise its many qualities and am constantly reminded of it's shortcomings too.

INTRODUCTION & PURPOSE

Blender has come a long way since it was open sourced in 2002. It has gained an awesome array of capabilities, and has gone through a few UI overhauls.

Ten years ago (gosh!), before the release of Blender 2.5, I published a document with a list of user interface and workflow improvement proposals for Blender. I'm very happy that some of these concepts made it into 2.5, which was a big step forward for Blender and, with the hard work of many great developers, formed the basis on which the next decade of features were built.

We are once again at an interesting time for Blender. The forthcoming Blender 2.8 provides a new opportunity to review what was done for 2.5 and what should be done in order to make Blender relevant into the future.

This document is an attempt to tackle workflow and user interface issues from an interaction design perspective, in order to formulate a clear, coherent and well defined vision - something that can be difficult to do in the context of open source software.

While I have compiled a list of proposals, in many ways it's more important to identify and focus on the issues, rather than necessarily agreeing on solutions. There are many times more than one right answer to a problem, and identifying the issues is what drives good solutions.

This is a proposal, and does not necessarily reflect what 2.8 will finally end up being, but I hope to help refocus attention on the Blender user experience. I believe, with a series of improvements, that Blender's UI cannot simply become adequate, but *excellent*.



In order to start discussing problems and solutions with regards to the Blender UI, we first have to define who Blender is for. Blender can be many things to different people. It can be part of a pipeline, or it can be the pipeline. Is it a developer kit? A programming environment? This needs to be simple and clear:

Blender is a tool for artists to create 3D graphics & animations.

This means that solutions need to be tuned for *end-user* artists to use *as-is*. As UI designers and developers, this is where our focus should be.

BEGINNERS & PROS

There's a popular but false dichotomy often cited, that an interface is either aimed at pro users or beginners. The implication is that, when designing a pro tool such as Blender, values such as ease of use, clarity and simplicity don't matter. On the contrary, it turns out the opposite is true: Pro's, who are usually paid to create content on a deadline, have *less* time to suffer confusing interfaces, to search through illogical layouts and learn unnecessarily complex systems. In other words, to truly make Blender a pro tool, it has to be easy, pleasant and delightful to use. Creating 3D content is complicated enough, the last thing we should do is burden users with unnecessary complications that steals focus away from their creations.

Every experienced user starts as a beginner. The UI should assist users to *become* efficient. An efficient, elegant, clear and consistent interface is not something that should be scoffed at as something for 'noobs'. When done right, it will improve the workflow for everyone.



In order to analyse issues and solutions, here's a list of relevant values for us to base our solutions on:

Speed

Performing an action should be as quick as possible, so that users don't have to wait.

Ease of use

Doesn't matter how powerful something is, if you can't figure out how to use it. The more burdensome an interface is, the less mental energy, or 'locus of attention' is left over to focus on the actual task.

Consistency

Once you learn one area, you should be able to re-purpose this knowledge in other areas. The more consistent, the less users have to learn, and keep in mind while working.

Affordance

Blender isn't used in isolation. Where it makes sense, use standard conventions so that users don't have to juggle different sets of mental paradigms when switching between apps.

Clarity

Design is making clear choices and communicating them. The UI should be clear & readable, but also defer to the content.

A NOTE ON CUSTOMISATION



A common misconception is that UI design doesn't matter as long as you give the user many ways to customise the behaviour of an app. Why get a nice, comfortable hammer when you can just get a lump of metal and make your own? There are a few issues with this line of thinking:

1. It makes maintaining the app more difficult the more customisation options are included
2. Customisation should never be used as a crutch to excuse unclear, inefficient or poor interfaces.
3. Most users won't take the time to do much customisation. Again, Blender is a tool for artists to use *as-is*.
4. Tutorials become difficult to follow if all users have different setups.
5. If the UI is good enough, users won't need to do much customisation.

That doesn't mean we shouldn't strive to make Blender flexible. What it means is that we can't hide behind customisability. Design is about making decisions, and good design is making clear ones.

KEEPING WHAT IS GREAT VS CHANGE

Blender has a number of central ideas we should keep. We should stay true to the core principles that have made it great. Here's a few highlights:

Non-overlapping windows

The subdivided areas make it possible to view lots of information without overlaps, and makes it easy to resize areas of interest to fit the task at hand.

Speed

The speed of which artists can work is paramount. In some ways Blender is already a very efficient tool, and we should aim to keep it that way.

Flexibility

Blender is used for a wide variety of tasks, and the user interface should remain flexible enough to accommodate all of the areas that Blender is used in.

TOOLS 1: INTRODUCTION



While there are many areas of the user interface, the graphic design, organisation, naming and specific features that could be enhanced or cleaned up, I find the biggest issue in Blender is related to tools. This was one of the main focuses of the 2.5 workflow refactor, and with it we gained a tool shelf, as well as an 'operator tweak' (I'll refer to it as 'tool settings' from now on) system - a way to adjust, and visually see, the result of a tool after it is activated. However, these features were never really particularly well implemented or polished, resulting in a rather messy interface paradigm that feels incomplete, confusing and inconsistent. Here's an attempt to clear it up, and to define a clear, understandable paradigm for interacting with tools in Blender.

TOOLS 2: ONE APPROACH TO RULE THEM ALL

Let me start by going on a small tangent. Even though it's not directly related to the tools system, in many ways the *Cycles* render engine is a good model for how to improve an area in Blender. It took an old system comprised of a rather arbitrary hodgepodge of features, each of which used a different system and technique, resulting in an overall confusing system that was hard to extend and difficult to use. Cycles is not only more technically advanced than the old render engine, but it has a design that is *clearly defined* and *consistent*. Once you understand the basic principles of the engine, it's quite easy to work out the rest. And for developers, the clearly defined (dare I say *opinionated*) design makes it quite obvious how to extend it. This same clearly defined approach is what we should have for the tools system too.

As for tools, Blender has a vast array of them, and while they are greatly useful individually, they don't follow any well defined conventions, and there's not a clear overarching design in place that makes activating, adjusting and controlling them consistent.



Let me give you a few examples:

1. Transform tools (G, R, S etc) completely block the UI while enabled, and don't let you input transform settings into the tool settings area until *after* you've finished the initial transformation. The tool settings for transform are mostly located in the *header*, not the tool settings area.
2. Very few tools have manipulators (only the basic transform tools) and they are entirely separate from the tools they are associated with.
3. Other tools, such as Subdivide, *don't* block the UI and don't immediately grab the cursor to use as input to control the # of subdivisions. But really, why can't you click and drag to increase or decrease them?
4. Grease Pencil tools activate when clicking the tool icon, but then, bizarrely, drops the tool after every line you've drawn(!). You need to then activate the tool again and again to make many lines. (That is, unless you enabled 'Continuous Drawing', but why is that necessary?) It can also be enabled by holding down D, which is not consistent with any other tool. Grease Pencil is dropped by pressing Esc (why?).
5. Sculpt and paint tools are picked in a menu rather than a list, and stay active until you pick a different tool, but the tool settings aren't in the tool settings area (why?).

When you look at all these tools and systems as a whole, it's frankly a mess - so much so that your head will hurt. We have a *lot* of work to do in this area. Below is a description of each major tools system area, as well as a coherent solution to unify tools.

TOOLS 3: TOOL SETTINGS

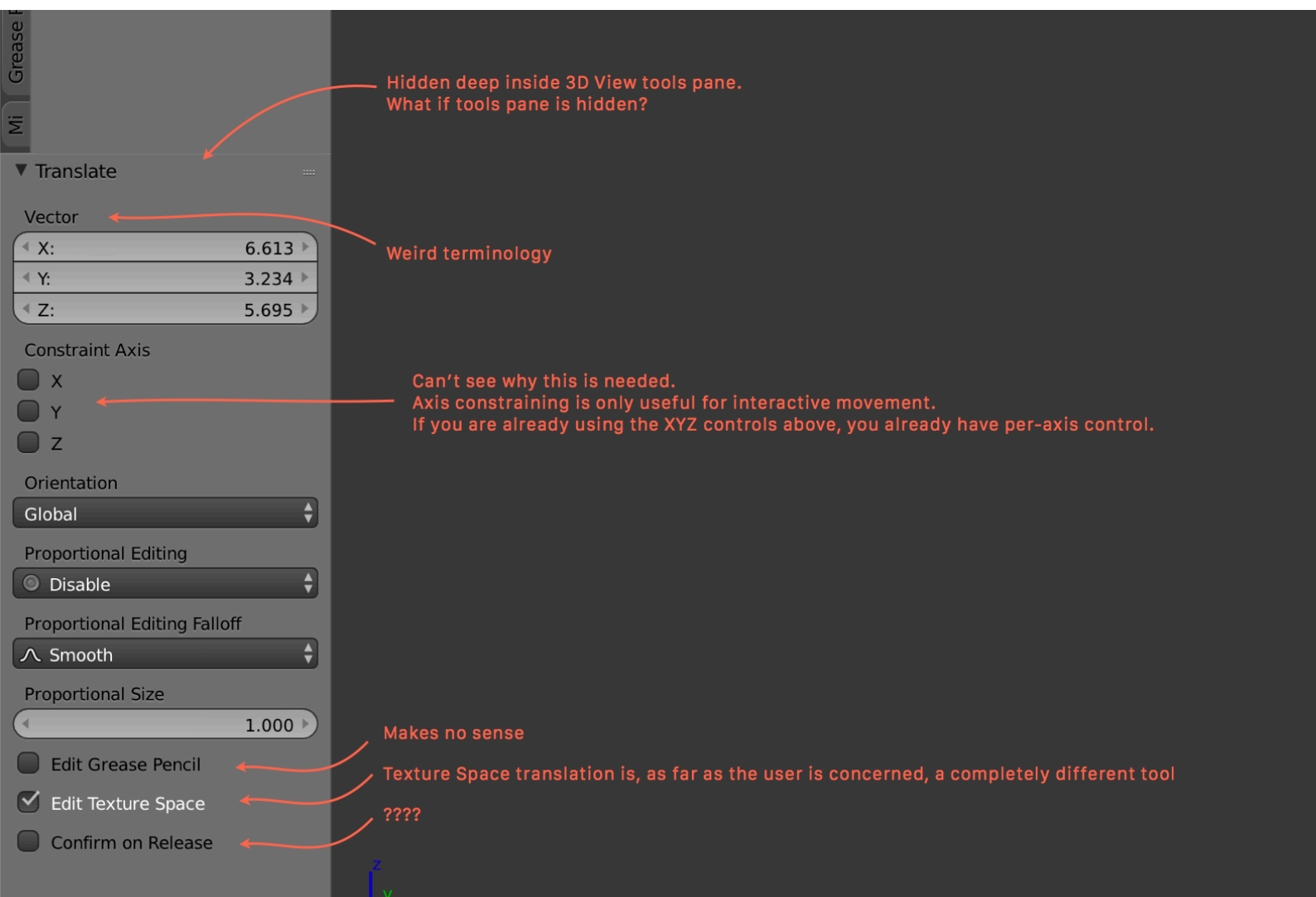
Analysis Since 2.5, we've had the ability to tweak tool settings after a tool is enabled. The workflow goes like this: You select an item -> pick a tool -> tweak settings, and see the results in real time. The tool settings are accessed in the lower portion of the tool pane (T key) or by pressing F6. However, in practice there are a number of issues that remain from 2.5 that were never fixed, often leaving tool settings to be incomprehensible, confusing or downright broken.

Here's an overview of the main issues:

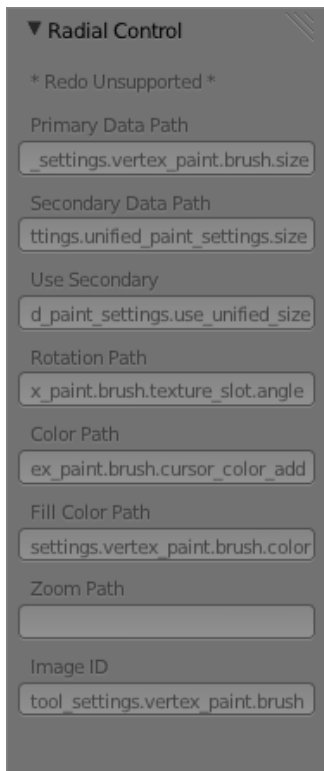
- Tool Settings are too hidden inside the toolbar. Users often close the toolbar to gain more screen space, but then it's not possible to tweak the tool settings, which is crucial for most tools. The F6 shortcut works, but is completely obscure considering how important this is.
- Even though Tool Settings are actually global and can relate to any Blender editor, it is shoe-horned into the 3D Editor. This creates a bizarre workflow where the user may use a tool in the UV Editor, while having to tweak settings in the 3D View.
- The contents of the tool settings pane is often poorly thought out, don't work or make no sense. Many times the tool settings pane is riddled with unrelated junk.
- Tool Settings are not respected for certain modes, such as any of the paint or sculpt modes, which just places the relevant tool settings in the tools pane itself, leaving the Tool Settings blank, greyed out, or filled with gibberish.
- Sometimes you can't edit the tool settings until *after* you've used a tool, such as with any of the transform tools.
- *Some* tool settings are placed in the area header, such as Snapping, Proportional Editing and Pivot Point.

I must say that these issues are pretty alarming. The ability to use tool settings are core to using Blender, and the fact that this area is so half baked is no good. Since it's such a core element, I think this is one of the areas that needs the most love and attention.

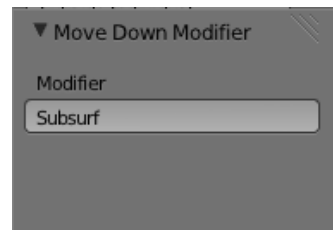
Below is a gallery of various tool settings, demonstrating just how incomplete and broken this interface currently is:



As you can see, many tool settings aren't really displaying useful user-facing options, but instead present internal operator flags that that either don't make any sense, or flat out don't work whatsoever.



Some tools, such as Move Up/Down Modifier (arrow in the Modifier list) was never meant to be tweaked.



Some tools don't support the Tool Settings panel, such as Radial Control in Sculpt Mode (F key). They currently show a list of meaningless settings that are greyed out.

Some tools have no settings. That's ok, but it's meaningless to have a popup appear with no settings.

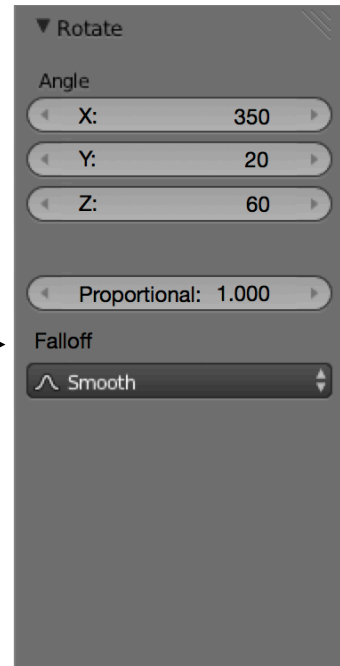
Only one field for a rotation in 3 dimensions? →

Not necessary if you can already type in XYZ values →

This doesn't work anyway →

Not necessary if you can set Proportional to zero →

Makes no sense to show this: →

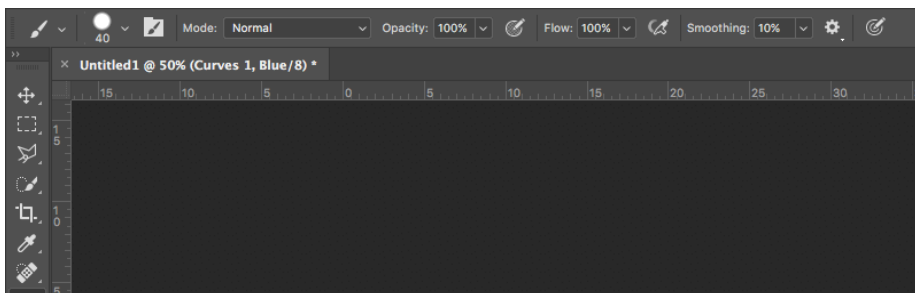




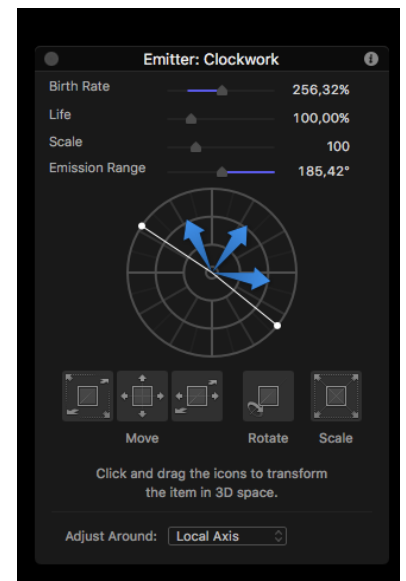
Proposal

To solve these issues, we have to take a step back. First, where and how does it even make sense to display the tool settings? Since they are global, the fact that the tool settings are crammed into the 3d View doesn't work. There needs to be a globally accessible way to tweak these settings, even if the 3D View is not available. I can think of four possible solutions here:

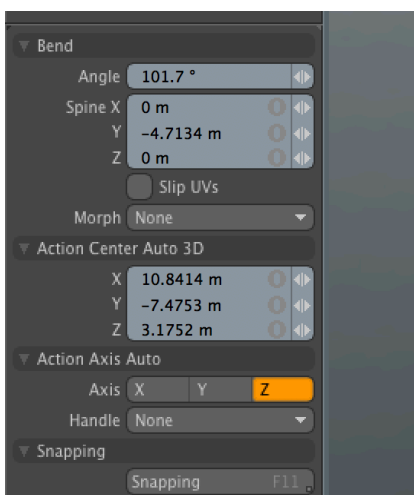
1. A Photoshop-style settings bar along the top
2. A Motion.app-style floating, global panel that appears when a tool is activated
3. A dedicated, global Tool Settings area, like Properties, but context sensitive, a la Modo
4. A new tab inside Properties itself



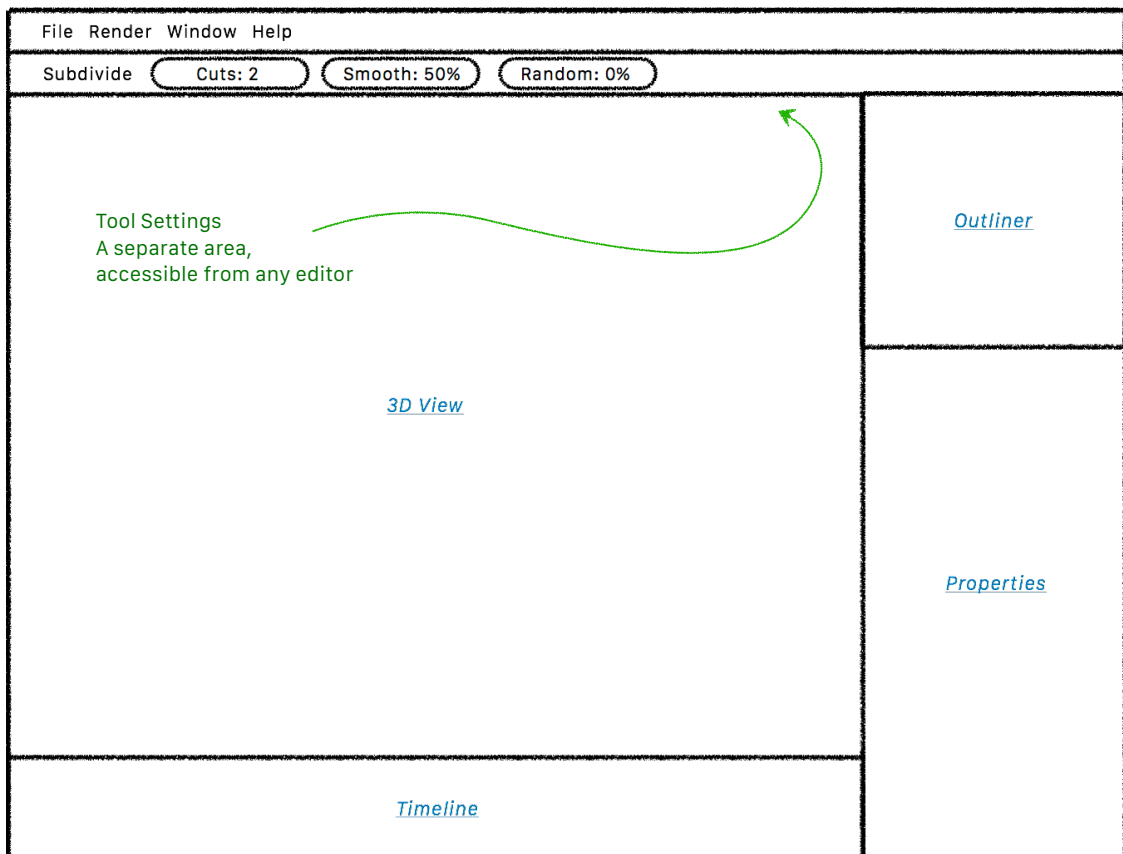
1: Photoshop



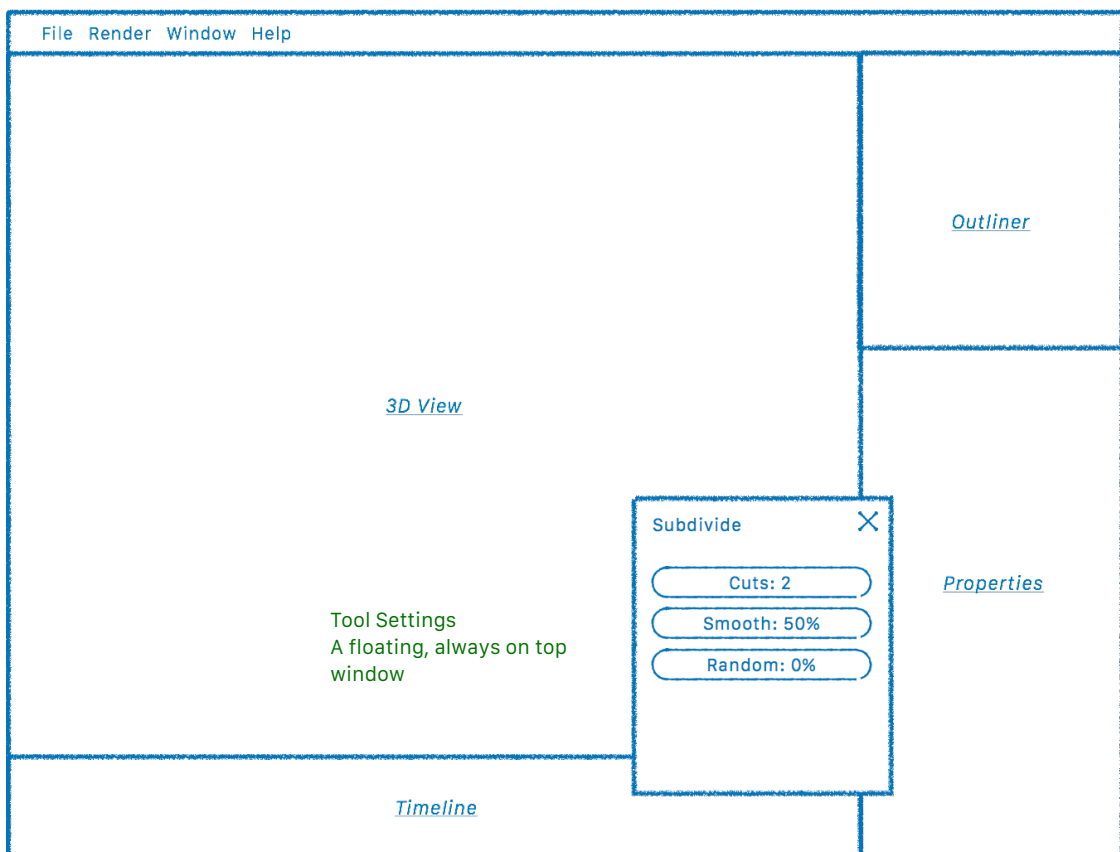
2: Apple Motion



3: Modo



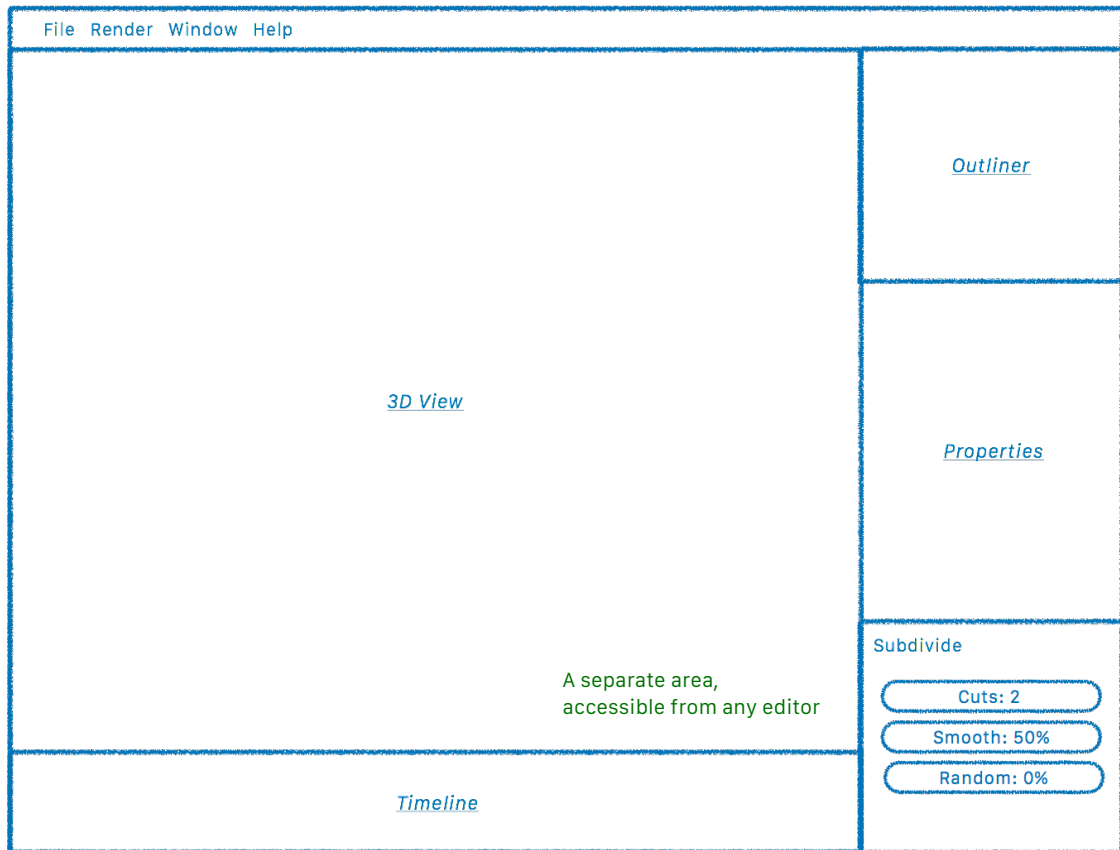
Tool Settings in always-present top bar



Tool Settings in floating window



3

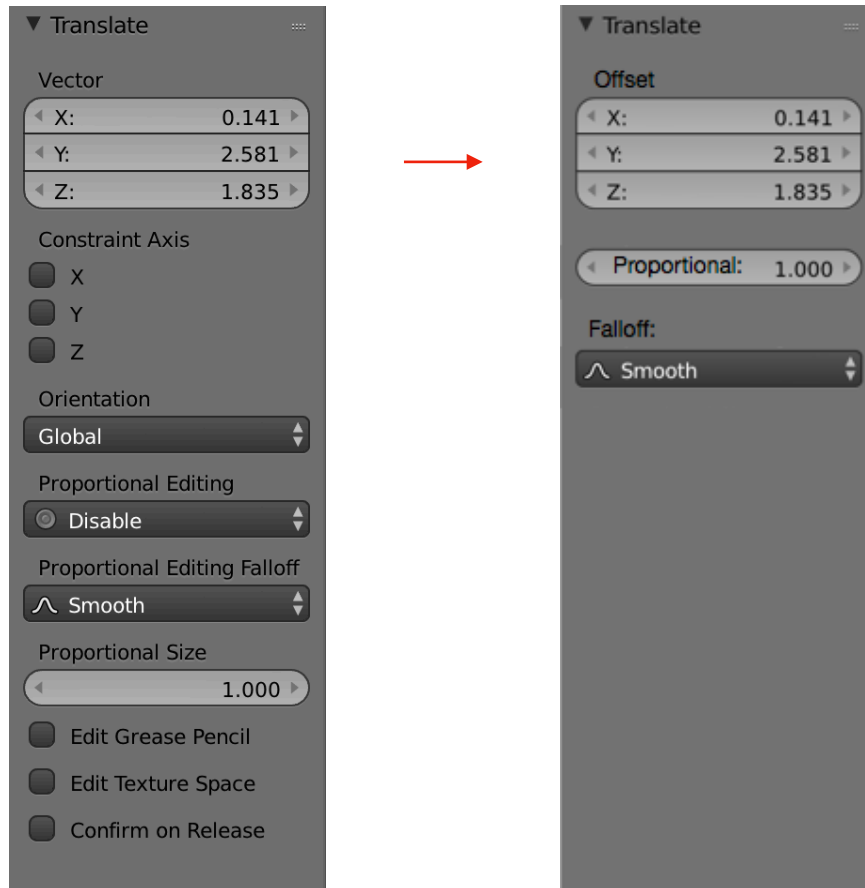


Tool Settings in dedicated area

I think solution #1 makes the most sense. It fits with Blender's paradigm, takes up little space and is accessible from any editor. A floating window violates Blender's non-overlapping principle, and the separate editor concept makes working in full screen mode impossible.



Additionally, we need to go through the actual settings themselves. The most basic tool in Blender, Grab/Move, has a tool settings pane associated with it, full of options that simply make no sense and don't work. This needs to be cleaned up, like so:

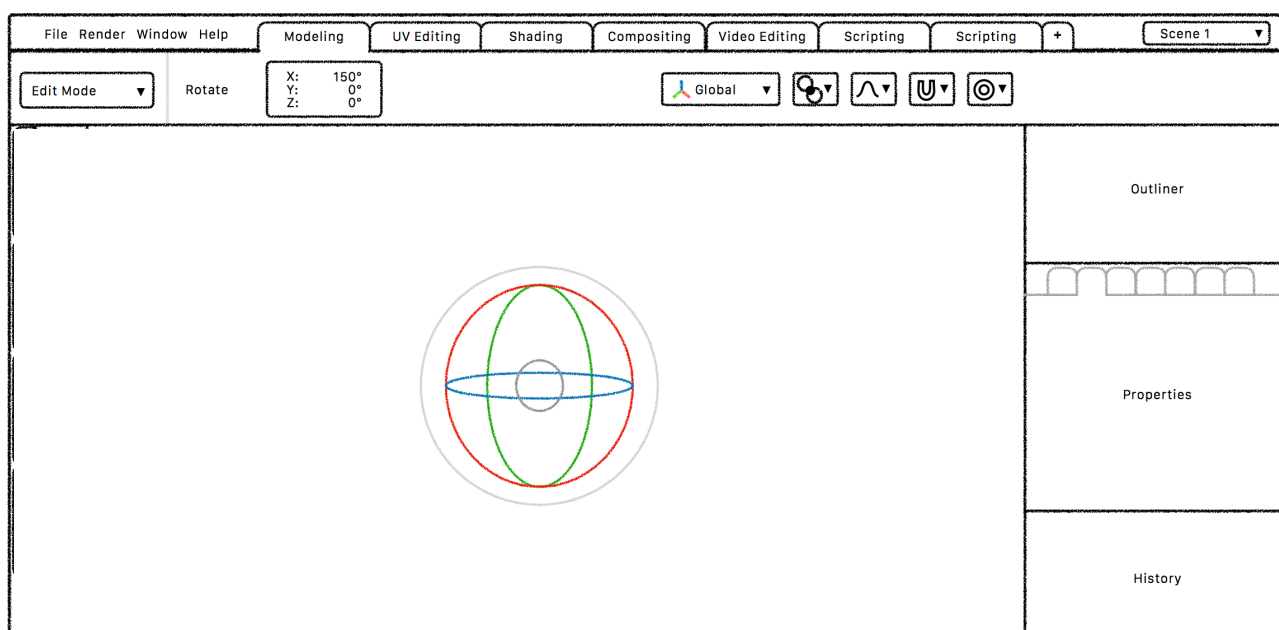


The same goes for other tool settings too. We have to go through all the tools and make sure the options are actually appropriate as user-facing settings, and not internal operator logic.



The tool settings area should be respected across all our tools and modes. So, whatever mode you are in, whatever tool you are using, the relevant settings are placed in the tool settings area. To summarise, here's what I think we should do:

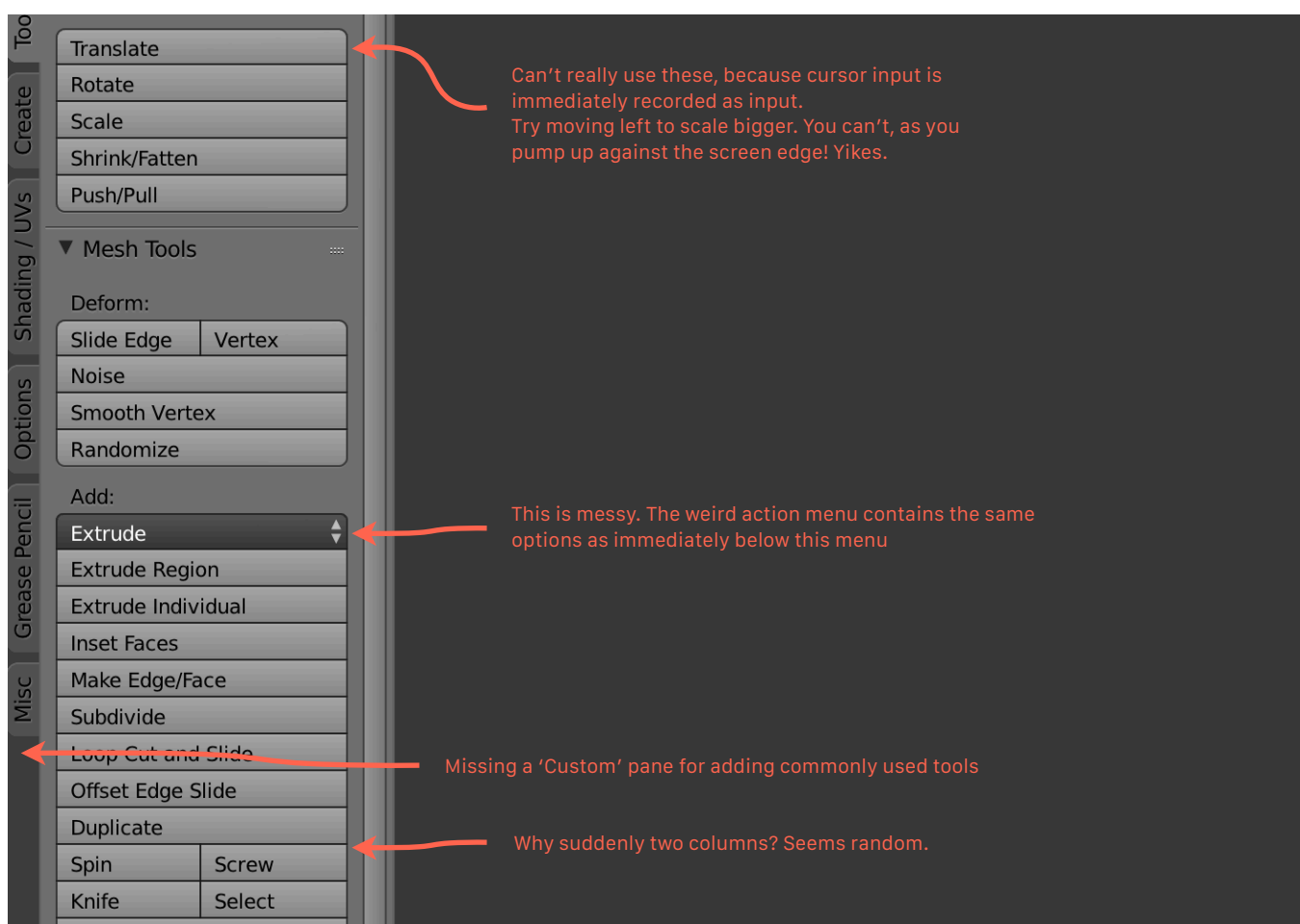
- Move the tool settings to a globally accessible area (most likely the top bar), so that you can access it when the toolbar is hidden.
- Remove the tool settings that temporarily hijack headers with obscure options
- Make it so that you can always edit and access tool settings while the current tool is active
- Unify the way tool settings are laid out with the most important options further to the left, and more obscure options further to the right.
- Clean up the tool settings themselves. Only present options that are actually useful to the user. Remove all the internal operator options that should not be user-facing.

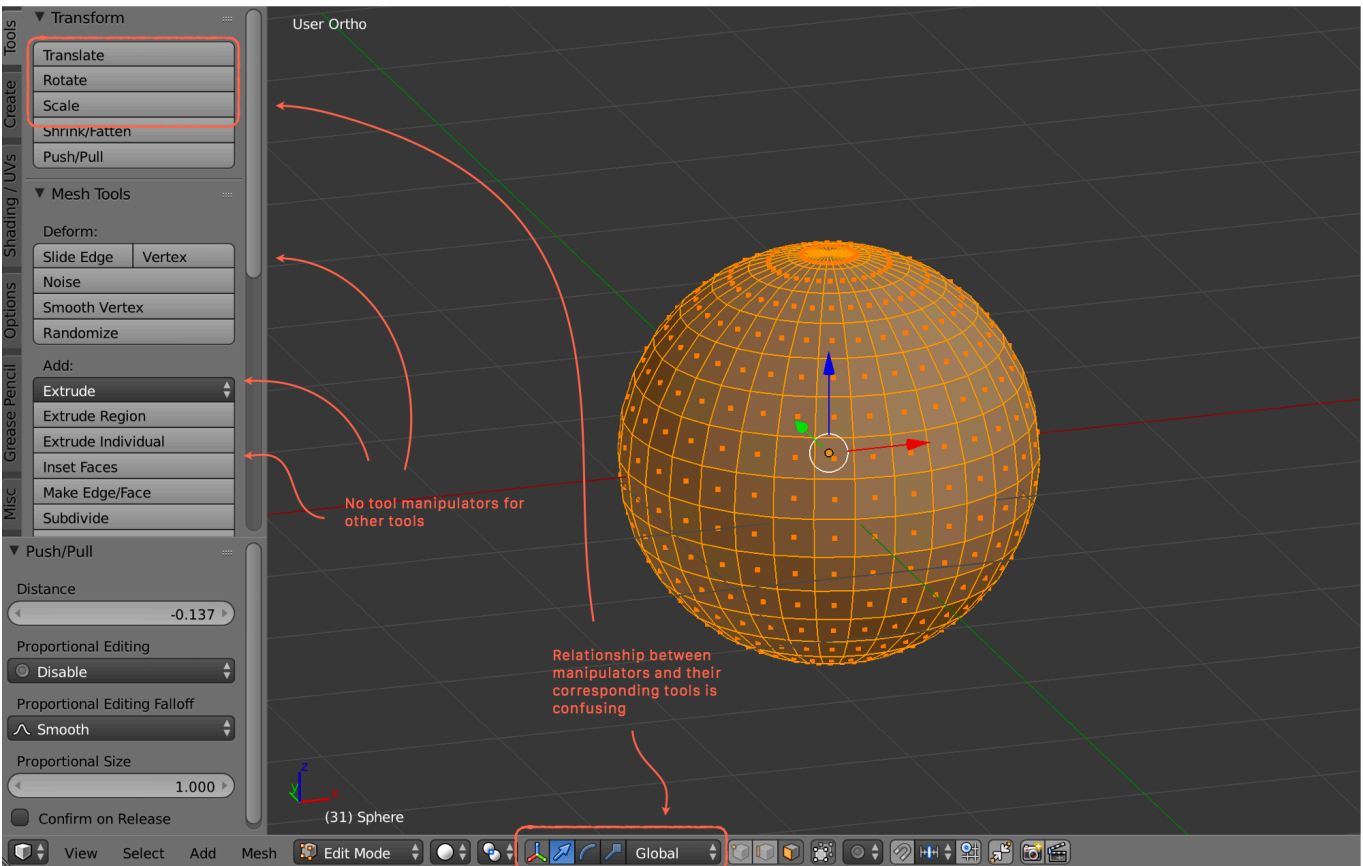
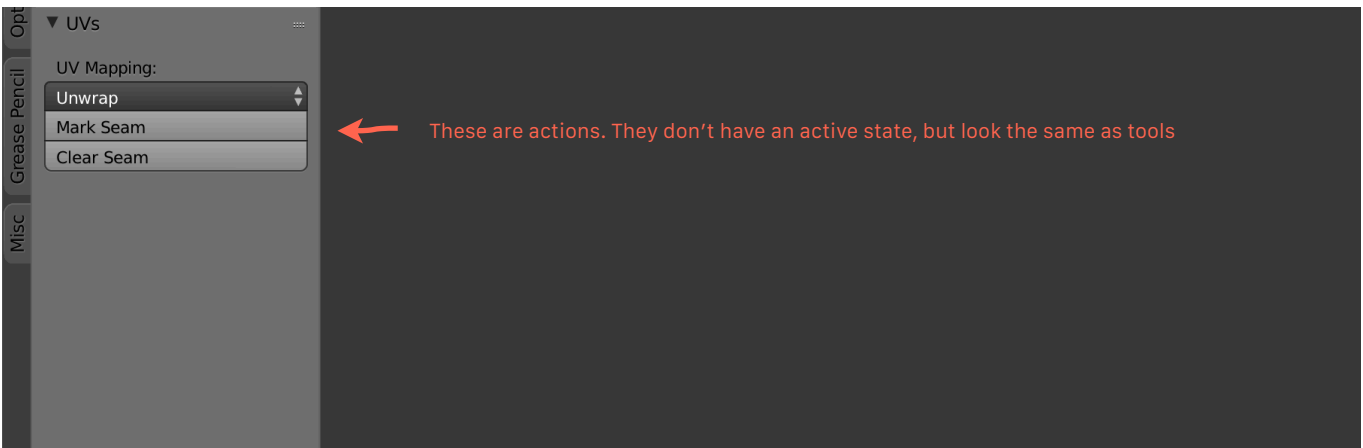
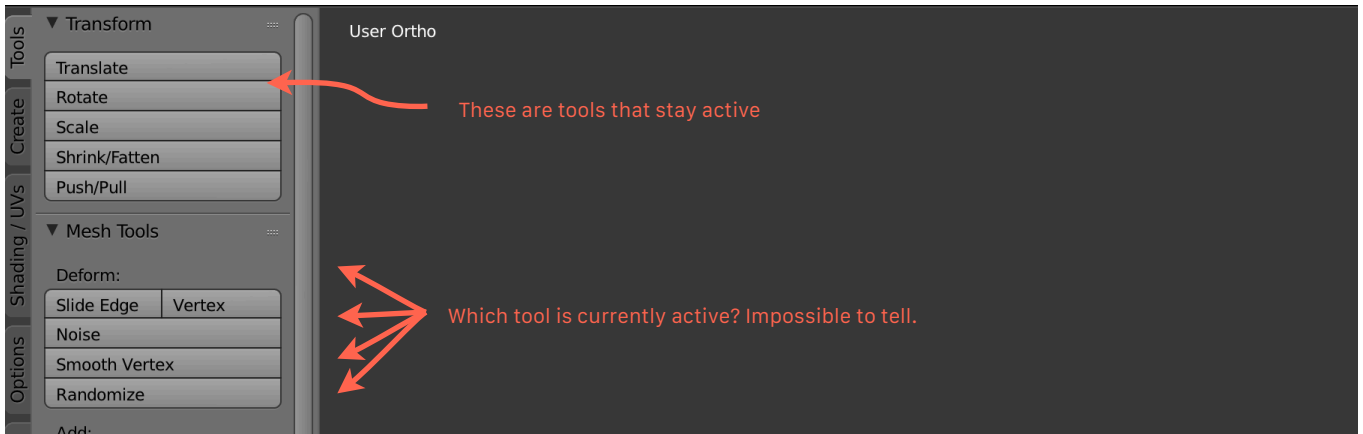


TOOLS 4: TOOLBAR

Analysis The toolbar was added in 2.5 to address the growing list of tools in Blender that had outgrown the old hotkey-only design where the user had to memorise hotkeys for every tool (oh no!). The old paradigm was feasible when Blender only had a handful of modelling tools available, but this quickly became unwieldy as the number of tools grew larger than the available keyboard keys. With 2.5, we added a contextually relevant list of tools for the user to browse through. However, the tool shelf still feels rather tacked on. Here's a list of issues:

- Certain tools immediately start interpreting cursor movement as tool input, even when activated via the tool shelf, making many tools extremely awkward to activate this way. These blocking 'immediate mode' tools are designed around a keyboard-only hotkey user interface, and is not compatible with a tool shelf or menus.
- Toolbar takes up a lot of screen space
- There are no icons or hotkeys visible
- The toolbar is missing it's single most useful aspect - an easy way to save your favourites.





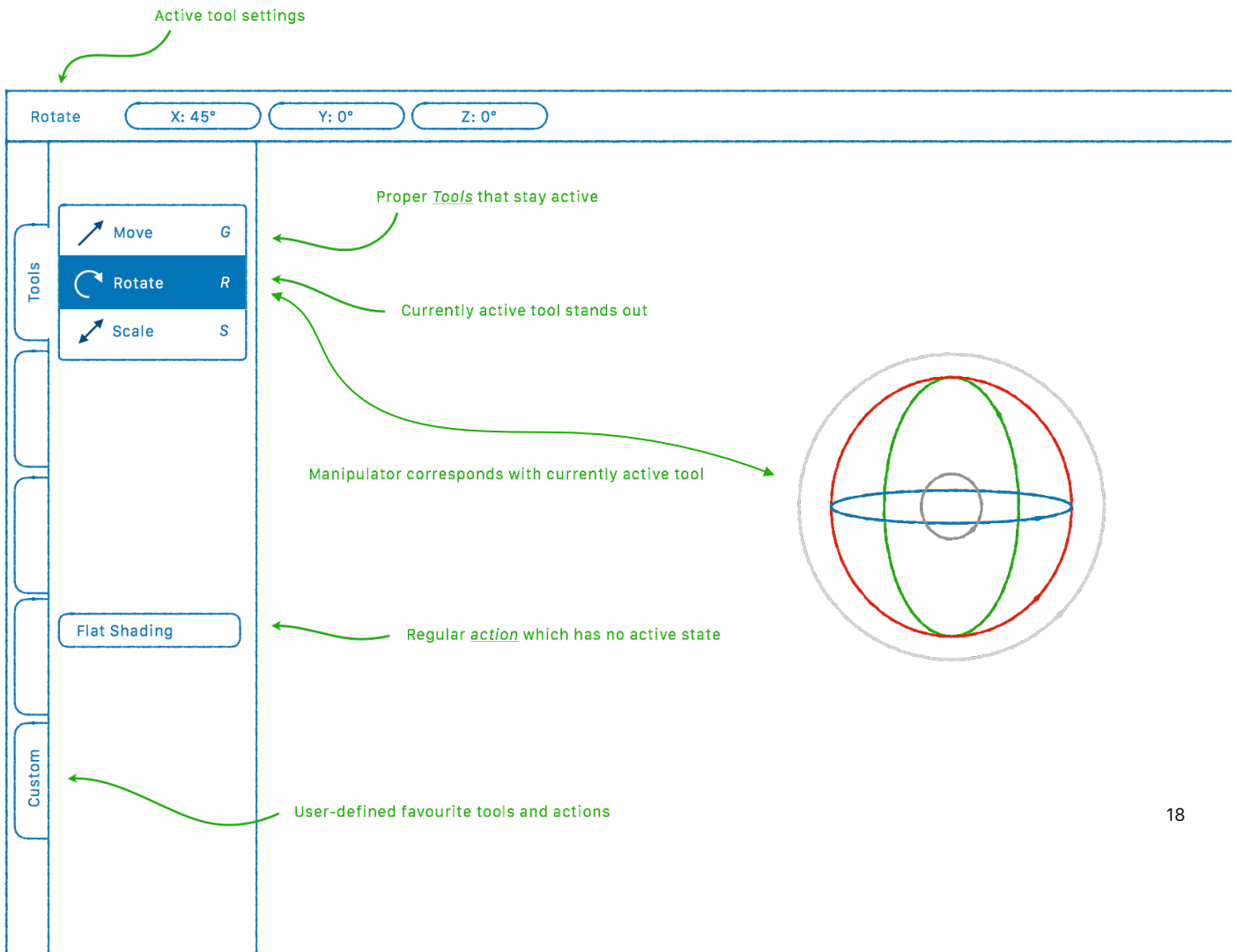


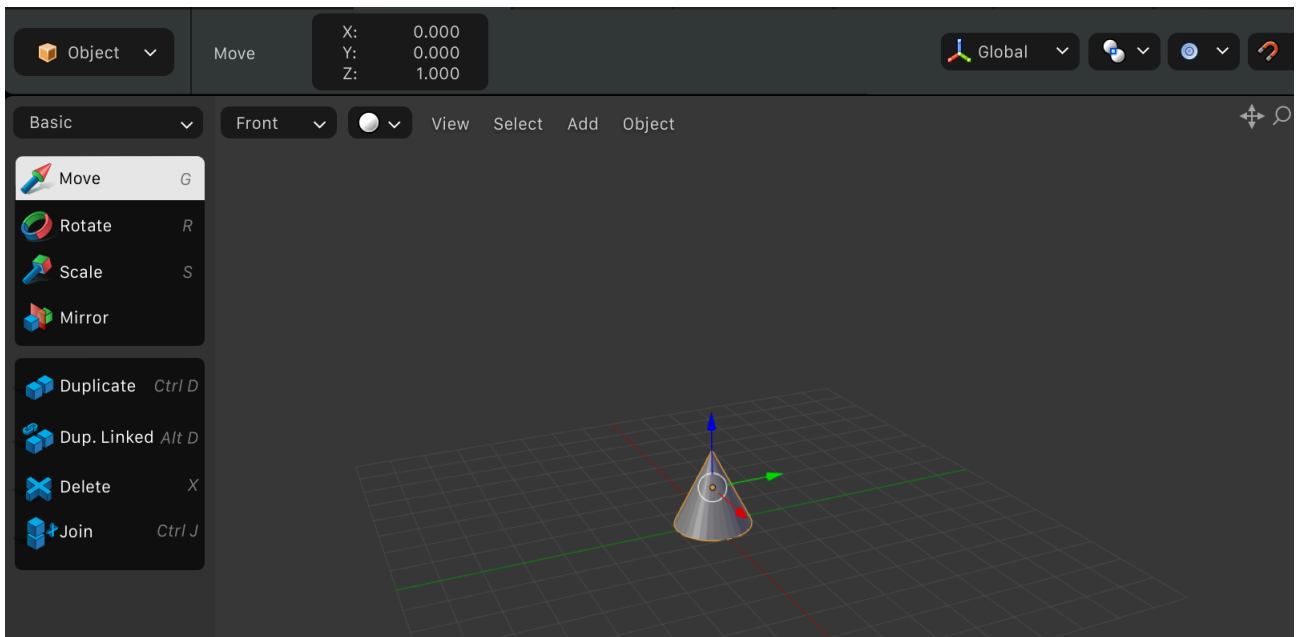
Proposal

Here's what we could do to improve the toolbar:

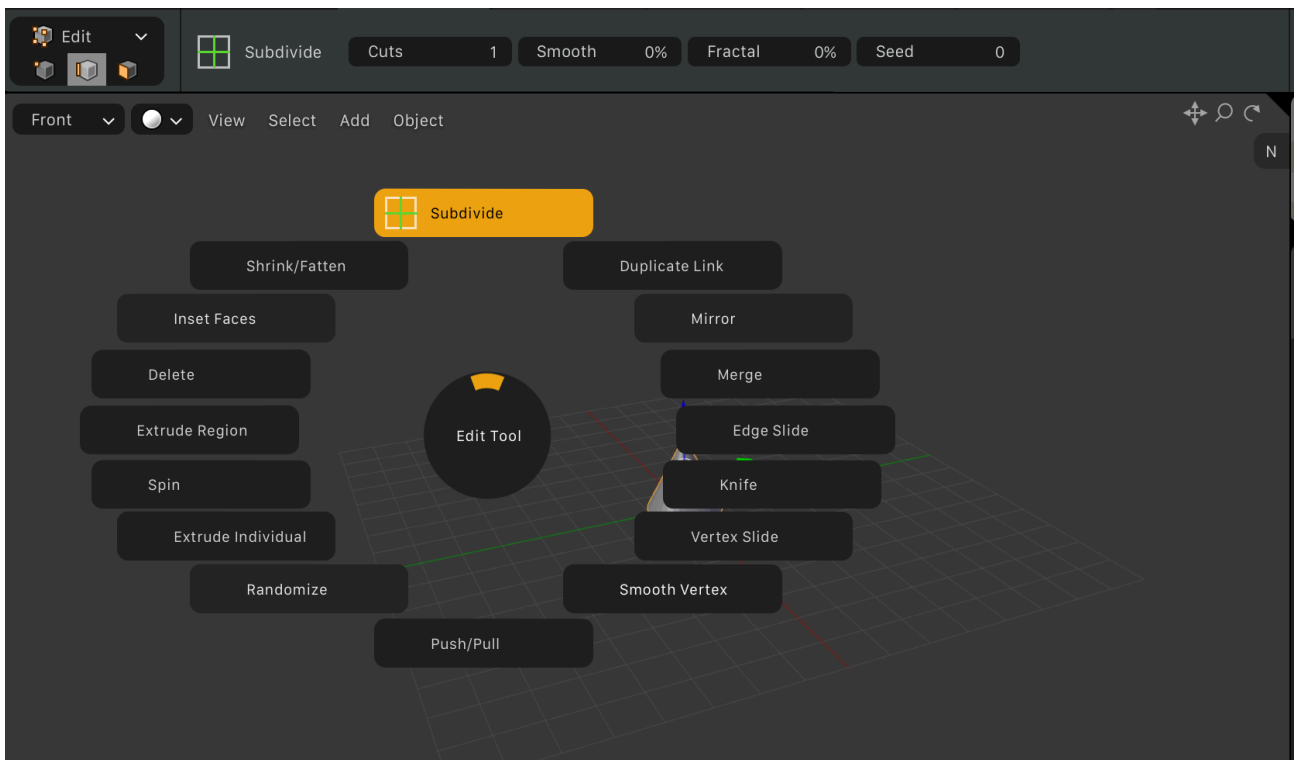
1. Drop the 'immediate mode' cursor input for tools such as Move, Rotate, Scale, Extrude etc. Just activate the tool, show the relevant manipulator and tool settings, and allow for dragging anywhere
2. Add a 'Custom' tools pane for the user to add favourite and commonly used tools.
3. Separate tool settings area from toolbar. This has two advantages: It lets the toolbar be slimmer and simpler, and it also makes it possible to hide it completely a lot of the time, if the user prefers to use hotkeys or contextual menus instead.
4. Place a bigger emphasis on contextual (radial?) menus, rather than a big always-visible, screen-gobbling toolbar

These changes can transform the toolbar from an odd wart to a useful, integrated part of using tools in Blender.





Example of visualised toolbar with active state, shortcut keys and separated tool settings area.



Here you can see the toolbar hidden, yet tool settings are still useable.



TOOLS 5: ACTIVE TOOLS

Analysis Blender does *kind of* have the concept of an active tool. In Edit Mode, click Subdivide, and the tool settings area will let you customise the subdivision settings live. The Subdivide tool is therefore active with the ability to tweak it, until you activate a different tool. However, several things are not all that clear:

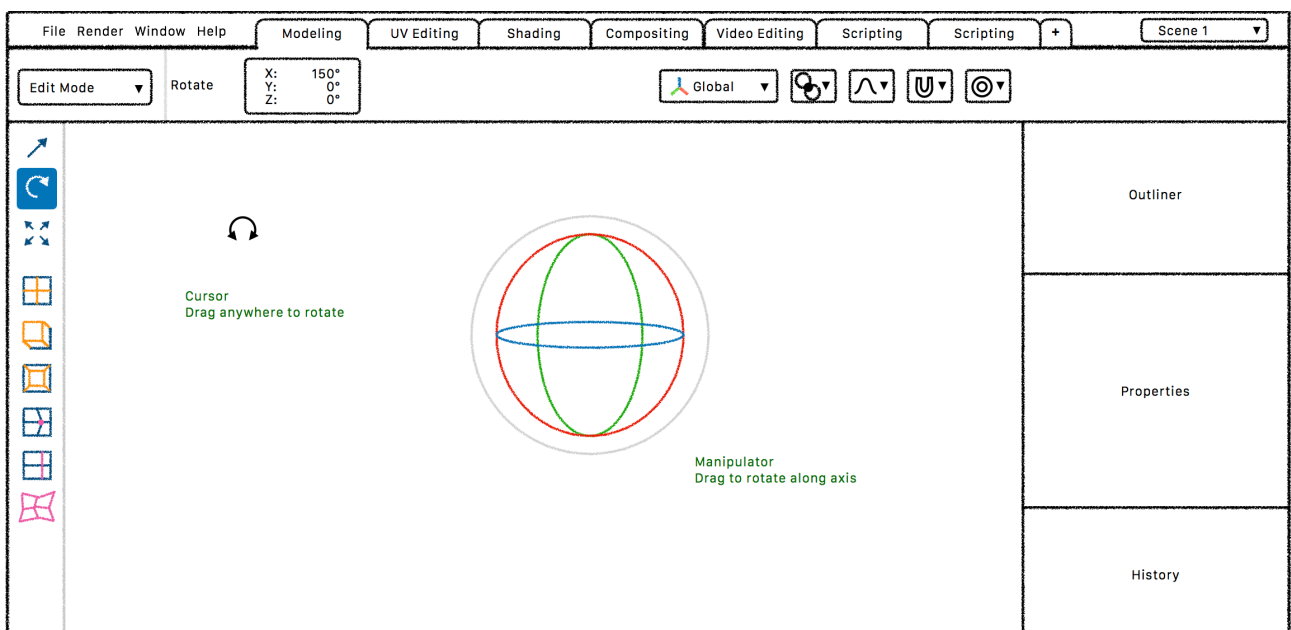
- There's no visual indication of which tool is active.
- There's no visual distinction between tools (something that remains active - eg Subdivide, Spin, Inset, Rotate etc) and simply actions (something that performs an action, with nothing to set, tweak or otherwise do - eg Shade Smooth, Mark Seam, Join Objects etc), so the user cannot know if he/she is activating something that requires user input, or something that just activates.
- When activating an action, such as Shade Smooth, it puts the action name inside the tool settings area, even though there are no relevant settings. This is misleading and confusing.
- Some tools in Blender are blocking and completely hijack the user input while in use. This makes it very awkward to input settings, which in this case happens by typing inscrutable data into the header, which is a holdover from before Blender had a proper tool settings system.

Proposal

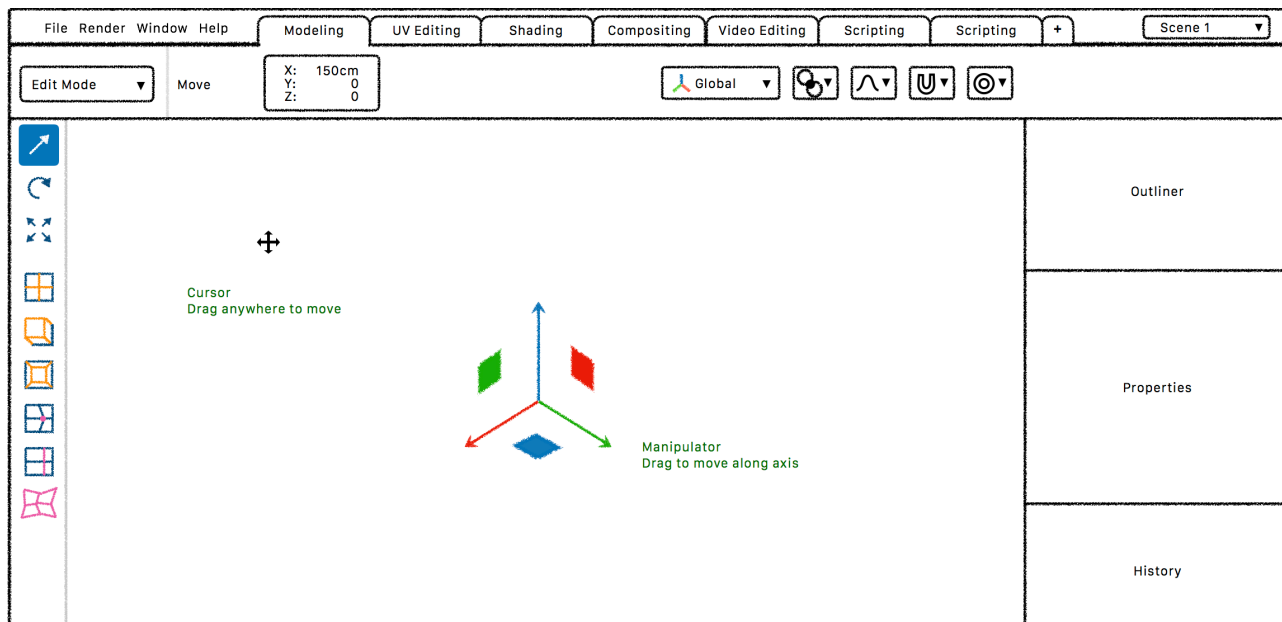
Active tools could be implemented in various ways. Here are the main points that would make it work:

- Make the currently active tool visually stand out. Just like the Paint Bucket tool in something like Photoshop is clearly distinctive when active, the currently active tool in Blender should be too. When Subdivide is active, the Subdivide tool button should clearly stand out.
- Unify the behaviour of the various tools so they all have a clear, useful active state. Remove the inscrutable and inconsistent 'header keyboard settings' concept.
- For users who like the directness of the keyboard-driven modal tools, add 'sticky keys' behaviour, so that holding, say, G, instantly starts a move operation that ends as soon as the user releases said key. This is even faster than before, because G-Move-LMB-Click becomes just G-Move-Release, without requiring a confirmation click.

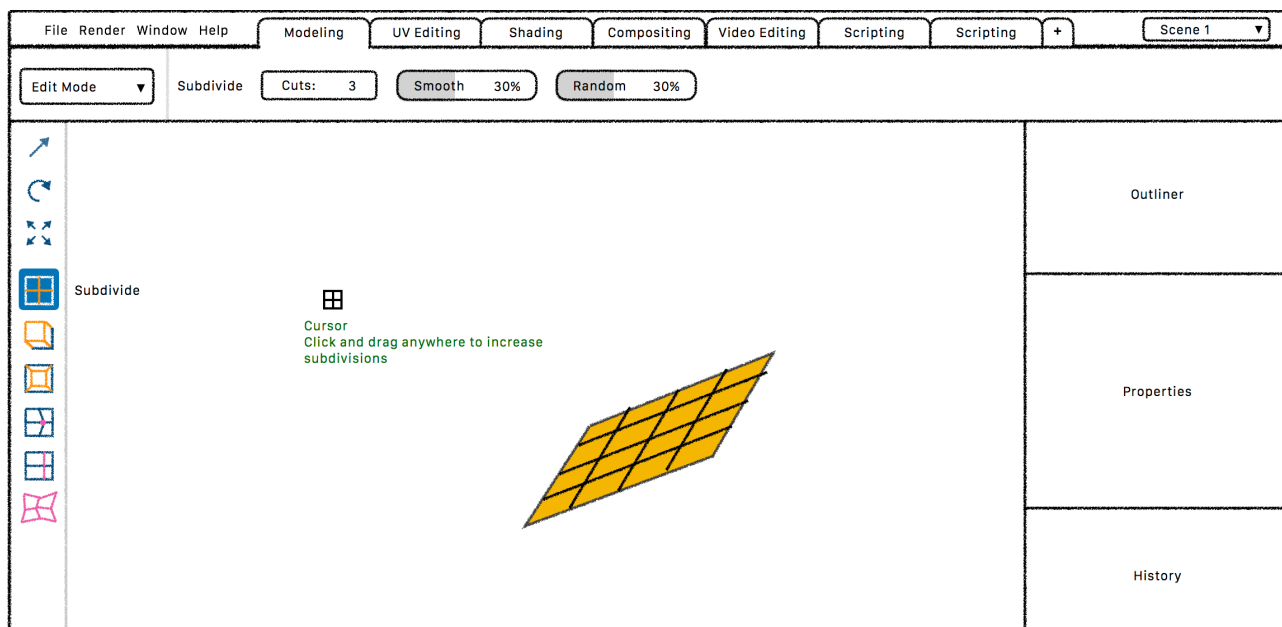
Below are several examples of how various tools would work with this system:



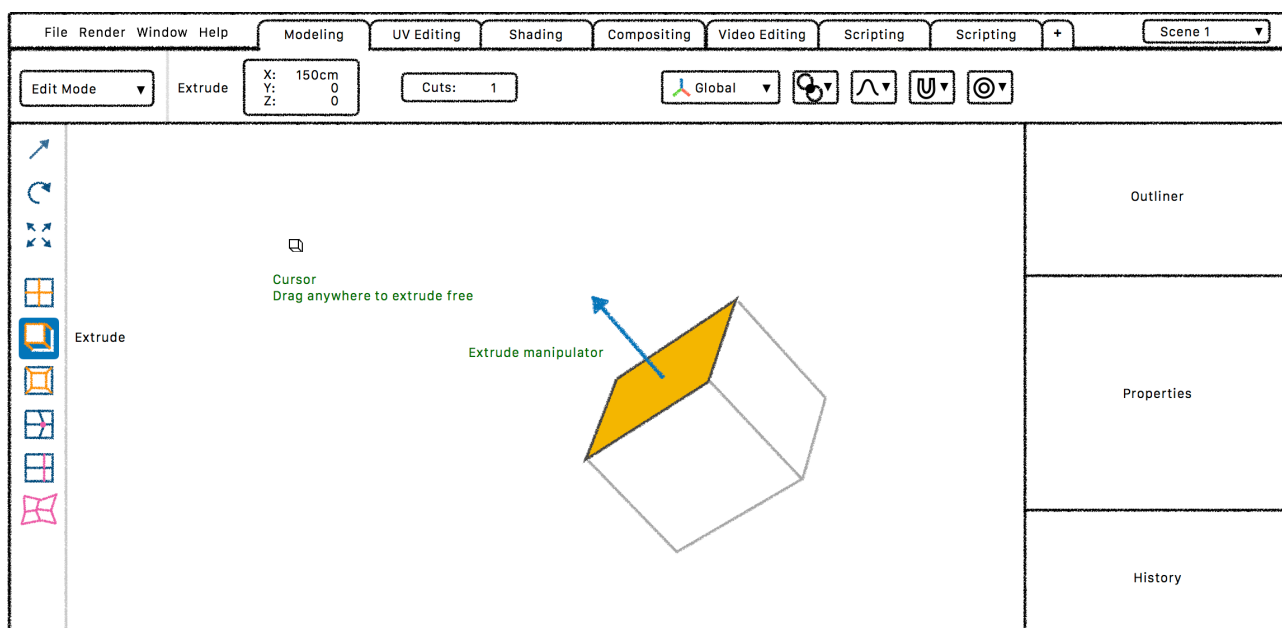
Rotate tool



Move tool



Subdivide tool



Extrude tool



Grease Pencil tools



Texture Paint tools

This way, tools can become more consistent, and also become useable when accessed from a toolbar. It also has the benefit of unifying various ways tools work and making tool settings more useful.



TOOLS 6: MANIPULATORS

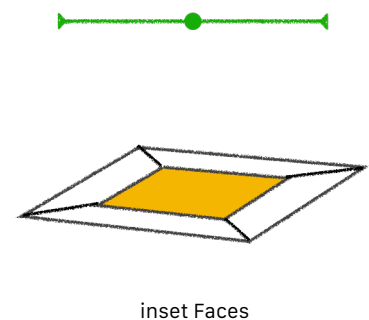
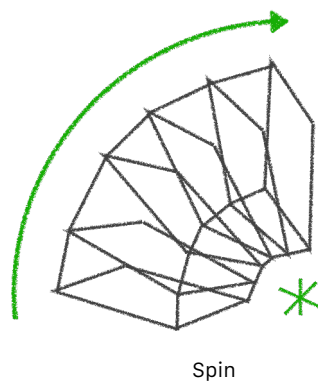
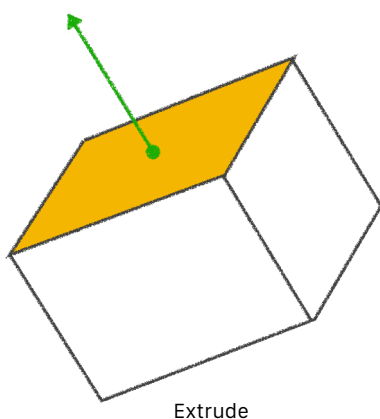
Analysis Many years ago, Blender added visual manipulators for three tools: *Grab*, *Rotate* and *Scale*. However, these have a few issues:

- Manipulators are disconnected from their respective tools. You can activate Rotate, yet the visual manipulator is still set to Scale
- They are only available for three tools, although many other tools would benefit from visual manipulators
- They are illogically and randomly placed in the window header, far away from the tool pane

Proposal Manipulators need some reworking. Here's what we could do to improve them:

- Remove the current transform-only manipulators from the header.
- Integrate manipulators into the concept of *active tools*. When a tool is activated, a visual manipulator gives the user direct manipulation over the tool inside the 3D view. You could see this being especially useful for translation tools such as Extrude, Vertex Slide, Knife, Inset etc.

Here are some examples of tools that would benefit from having manipulators added:



TOOLS 7: STICKY KEYS

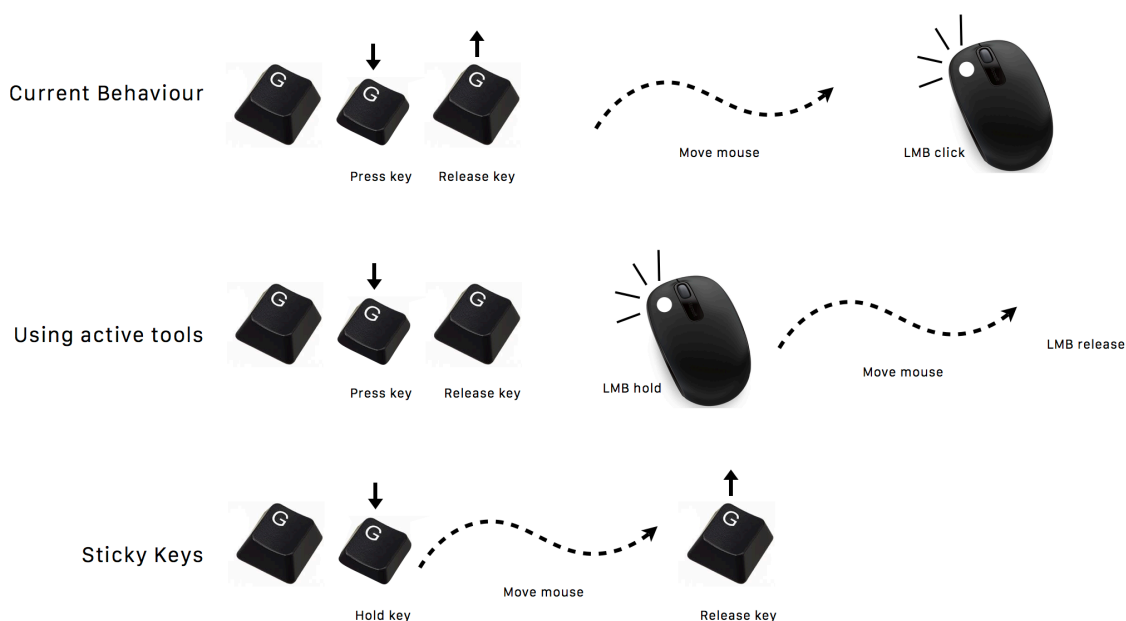
Proposal

If we switch to a tools workflow centred around active tools, an issue might be that some users would like a more immediate interaction mode also. One where you don't need to first pick a tool and *then* begin moving. Even though there isn't an extra click of the mouse, the immediacy of the old behaviour has something direct about it that would be good to keep around. Enter sticky keys!

What is sticky keys you might ask? Well, it's a way to quickly jump in and out of using a tool simply by holding a key. In fact, Blender already uses sticky keys, albeit in an inconsistent way: Grease Pencil uses it to allow you to temporarily activate annotation by letting you draw while holding down the D key.

Here's how I imagine it could work:

1. The user can tap G to activate the move tool. Then he/she can click anywhere on the screen to move perpendicular to the view plane. He/she can also, optionally, use the manipulators to move in a specific direction, or input numeric values into the tool settings area.
2. Alternatively, using sticky keys, the user can simply hold the same key (G), move the mouse and then release the key to perform a quick move. No activating required, and no clicking of the mouse necessary at all. This is much faster than the old Blender behaviour where the user must first tap G, then move, then click.





The beauty of this is that we could apply this principle to all the tools. Take Grease Pencil: You tap D (or click it in the toolbar) to enable the tool, then start drawing. The tool stays active until you pick a different tool. Or, if you just want to quickly draw a line without activating Grease Pencil, you can hold D, draw your line, and then release. Viola!

Let's take another example: Subdivide. Normally, the user activates it by locating it in the toolbar or hitting W. From here, the user can drag anywhere in the 3D view to increase/decrease cuts, or type in a number in the tool settings area. Alternatively, using sticky keys, the user can also hold W, move the mouse to increase the # of cuts, and then release W. Bang!

This is a super fast approach to using tools that, with a new consistent tools system can be applied globally to all tools.

TOOLS 8: ICONS & CURSORS

Analysis Icons provide a nice way to quickly identify different items in a list. They also allow for communicating more clearly what a tool does (A picture is worth a thousand words, after all). A good example is something like the Pivot Point menu with it's clear illustrations of what the various options mean.

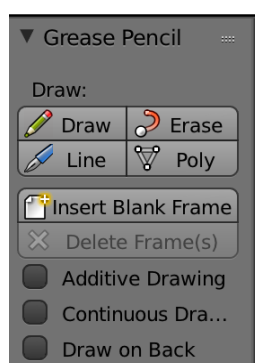
The nice thing about them is also that, because 3D apps, and Blender in particular, is very jargon heavy, you can supplement a very technical term with a clear illustration of what it does. Inset Faces, for example, is a tool that can be visualised to neatly describe what it means.

However, the current tools system has a rather inconsistent use of icons:

- Icons are present for paint modes, but not Particle Edit, Edit Mode or Object Mode
- Icons are used for Grease Pencil tools, but they are not bespoke - it's just re-using unrelated icons from other areas
- Blender already has some special cursors (think knife tool), but like the rest of the tools system, it's arbitrary, and it's not used throughout.



Edit Mode: No icons.



Grease Pencil: Reuse of wrong icons



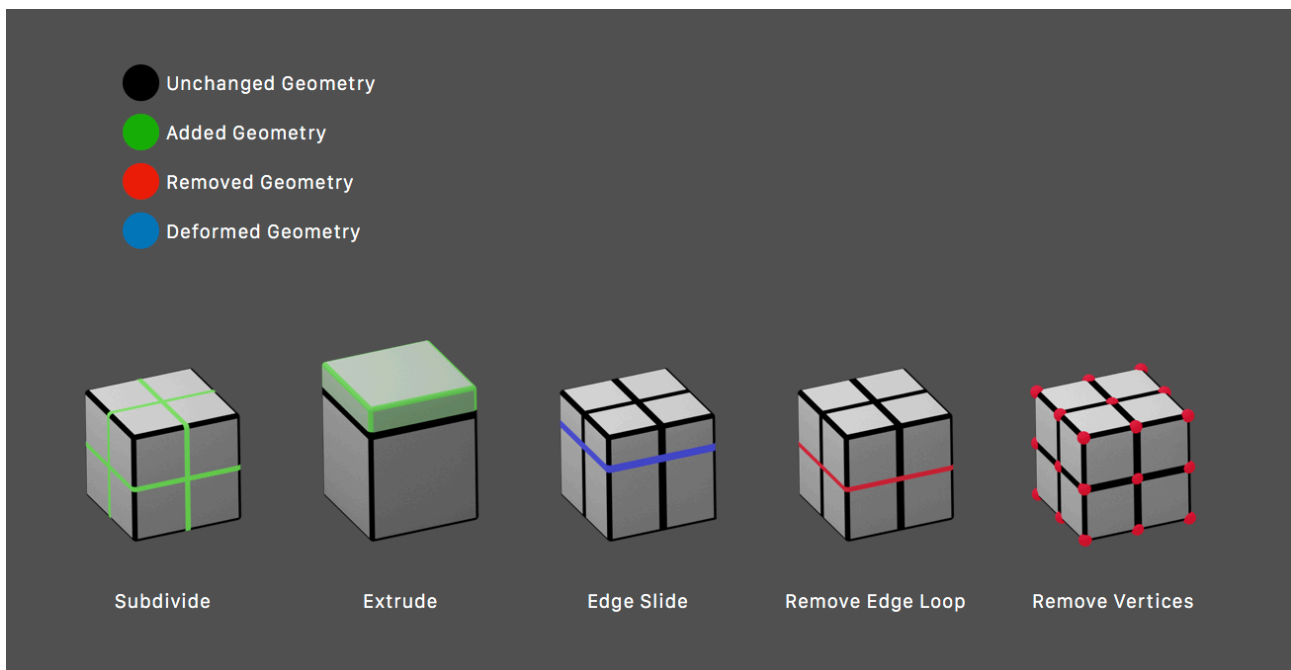
Sculpt Mode: Icon-menu. Only one tool visible at a time.

Proposal

Here's what I think we should do:

- Add icons for all the tool shelves in every mode
- For any tool that remains active, also have a matching cursor design that helps communicate which tool you are using.
- We should create very clear guidelines for these icons, so that we can easily draw new ones later when we add new tool
- These guidelines should have clearly defined rules for what added geometry looks like, what removed geometry looks like, and what deformed geometry looks like, so that we create a consistent visual language.

Here's an example of how icons for Edit Mode tools could look. I created them in Blender itself, which also makes adding more fairly easy. These do require some more work, but gives an idea of such a visual language:



TOOLS 9: RADIAL PIE MENUS

Proposal

Although it's very useful to have a toolbar, to visually present a list of useful tools relevant to each mode, it does come with some draw-backs. Namely, it takes up quite a lot of screen space, and that goes for *each* 3D view. If you have multiple 3D Views, users have to setup a complex layout where perhaps only one area has a toolbar.

For this reason, a lot of users hide the toolbar so they can regain this space. That creates a second issue: How then do you access the tools? Do you then have to remember the hotkeys for each and every tool, in every mode? That's quite daunting, as Blender has such a multitude of tools and modes.

Here's a way we could solve both of these issues:

- As an addition to the toolbar, we could add a consistent tools menu to switch between active tools, similar to how we currently have a consistent menu to add items across Blender (Shift+A)
- The tools menu could be invoked with a consistent shortcut key (perhaps replacing the inconsistent W key menu)
- The tools menu would include the same active tools found in the regular toolbar, and would make it possible to hide the toolbar completely

This approach has the advantage that users don't have to remember dozens of shortcuts for every tool in every mode. Instead, users can just worry about knowing *one* shortcut (W?) to select any relevant tool for each mode. On the following pages, there are visualisations of how this could look for several modes

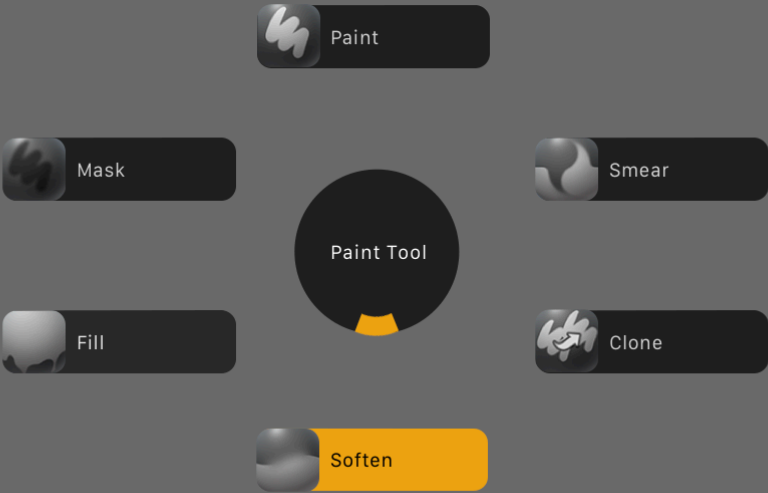
Sculpt Mode



Hair Editing

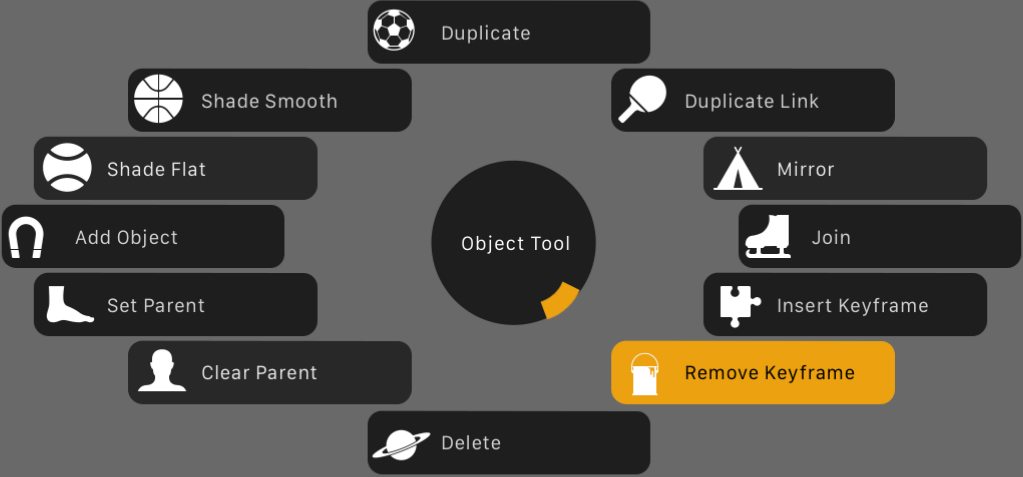


Paint Mode



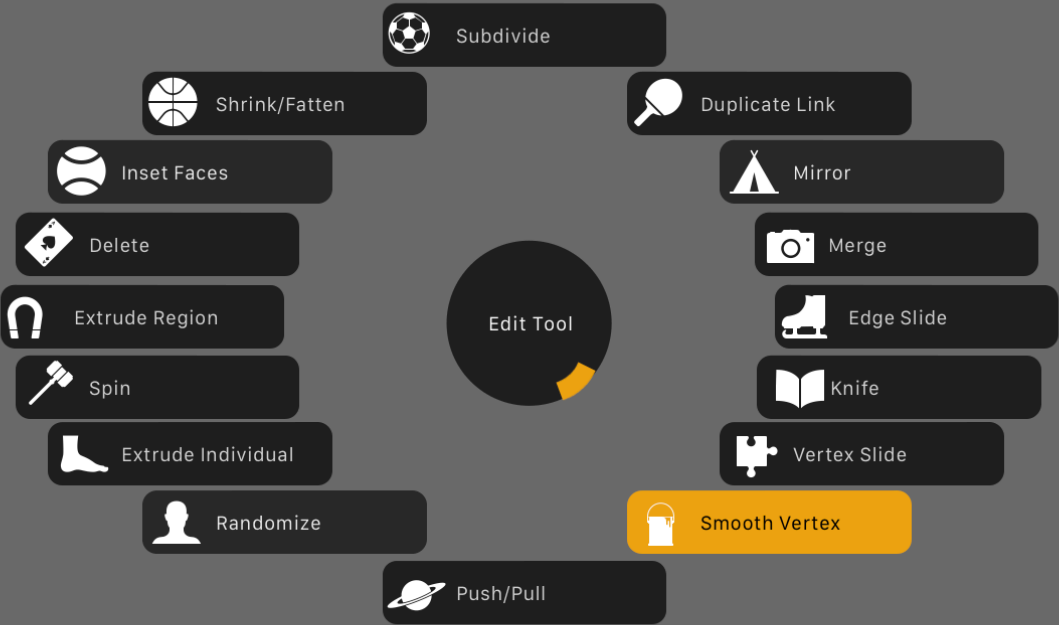
Object Mode

These icons are temporary

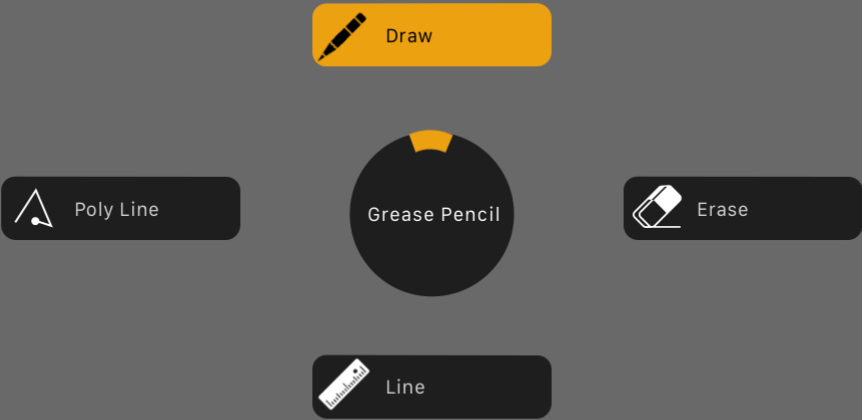


Edit Mode

These icons are temporary



Grease Pencil





TOOLS 10: WRAP-UP

Here I'd like to summarise all the points above. Here's how the various parts of the tool system all fit together:

- Tools should be separated into two buckets: *Tools* (active operators with settings. Requires user-input.) and *Actions* (Can't be active. Just a command).
- All tools should have a clear *active state*. When activated, they take over the tool settings area, and (optionally) adds relevant manipulators.
- Tools should not block the UI and steal cursor input.
- Tool settings need to be scrutinised and cleaned up. Internal operator flags should not be visible to the user
- Manipulators should be consolidated with their respective tools, rather than being an option on the side
- All tools should use the same tools settings area for their respective settings. No tool settings in the toolbar, header or other random places
- We should add tool icons to better identify tools and to help explain what they do, as well as matching cursors to make it clear which tool is active
- Sticky keys can be added consistently across the board, as a super quick way to engage tools

This would for a coherent and consistent tools paradigm, that is both quick to use, flexible and easy to use. The key is that it represents a clearly defined system that can be applied to Edit Mode tools, transform tools, paint tools and hair combing tools. It will no doubt require a lot of work and testing to implement, but I think it's worth it. As Blender expands over the next many years, such a fundamental area as the tools system needs a solid, well defined paradigm.



DEFAULT SETTINGS

Proposal

While often overlooked, the default settings are very important. New users use them, many people updating to new Blender versions use them. Whenever Blender is re-installed on a new computer, users will use these.

While I won't go into all the settings of Blender, there are some rather obvious defaults that I think would make sense to set as defaults, such as:

- Making Cycles the default renderer (The Blender Internal renderer is outdated and not being actively developed).
- Setting the Cycles renderer up with some more useful defaults, such as disabling caustics, GPU rendering enabled in Preferences, Dithering=1.000, etc.
- Default Cycles materials set to Principled BSDF in order to give users a quick way to create a vast array of materials without having to dive into the Node Editor.
- Units set to Metric, to give users a useable context for scale (Sorry USA :))
- Added meshes should have generated UV's on by default, so textures can be applied easily.
- Camera sensor size set to a sensible standard, such as either 35mm or super 35mm.

There's probably many more settings you could mention, but this is merely to say that the defaults needs an overhaul.



LMB SELECT

Proposal

One of the most baffling things about Blender for most new users is that we don't use the left mouse button (LMB) to select items in the 3D View. Instead, LMB is reserved for placing the 3D Cursor, while selection is done via RMB. I think it's time to consider if this is really the best idea:

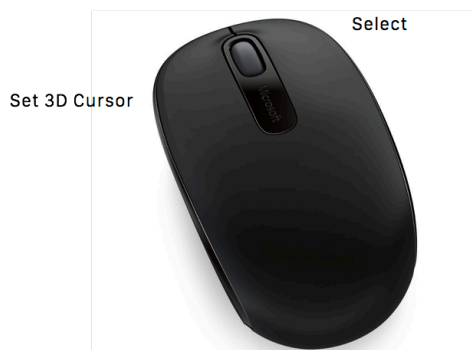
- As everyone knows, LMB to select is a convention so entrenched in any piece of software and any OS, that not using it is missing a *gigantic* affordance.
- If we are to not use LMB for select, we'd better have a damn good reason why not.
- The thing we've assigned to LMB, the 3D Cursor, is, I'd argue, not particularly important or useful. Or, said more eloquently, It's relative importance does not correlate with being assigned to the main mouse button. The 3D Cursor could easily be manipulated in a different way. (Ctrl+LMB? Or using a special 3D Cursor tool?)
- Using RMB for select is in many ways inconsistent with Blender itself. We use LMB to select buttons, items in lists, nodes etc.
- Using RMB for select means we can't use RMB for context sensitive menus, like we already do in the Properties area.
- Even when users have gotten used to RMB select, most pro's use multiple apps and switch between them constantly. RMB select makes this awkward.

One of the nicest things about using LMB to select (providing that we set the 3D cursor aside) is that we open up RMB for all sorts of useful things. We could use it to open a context sensitive menus that gives users relevant tools depending on the context he/she is in. It could, for many uses, reduce the dependance on the toolbar, thereby reclaiming lots of screen space. This would not only be extremely useful, it would also be consistent within Blender itself, as well as literally any other piece of user-facing software out there.



There are, however, a few issues to fix, in order to make LMB select work well as a default. For example, what about selecting faces while in paint mode? (Could be solved by adding a selection tool to paint mode). Also, what about generally selecting items when a tool is active? Should users have to first 'drop' the current tool, or should they switch to a specific select tool? I'm confident we can sort out these niggles though, through a process of experimentation and testing.

Current mouse button assignments for the 3D View



2.79 DEFAULT

Selecting with LMB is consistent with Blender itself, as well as leveraging enormous affordance



2.8 DEFAULT



DRIVERS

Analysis This is one area where the current workflow he have falls down completely. Adding drivers and relationships between items in Blender requires and obscene amount of steps to set up even a single driver. Let's take a simple use-case: Making the X location of Object A be driven by the Y location of Object B with a 2:1 relationship. Pretty simple.

To do that, these are the steps currently required:

1. With Object A selected, right-click on the X Location field.
2. Go To Drivers sub-menu
3. Click Manually Create Later (Single)
4. Split the viewport
5. Change the new area to the F-Curve Editor
6. Change the F-Curve Editor mode to Drivers
7. Select the Driver in the source list on the left
8. Expand the area Properties by clicking the + icon
9. Click the Drivers tab
10. Change Type to Sum Values
11. In the Variables section, click the droplet icon and select Object B
12. Change the variable type to 'Y location'
13. Select the 2nd vertex of the driver F-Curve in the Drivers area corresponding to the current driver
14. Click the F-Curve tab in the F-Curve Editor
15. Set the value to 2.000

That's 15 steps. I think it's pretty easy to see that it's too much work for a relatively simple action.



Proposal

There are a number of different ways we could improve the workflow for adding drivers. Here's one way:

1. With Object A selected, right-click on the X Location field.
2. Click Add Driver. A popup appears, like the one illustrated below
3. In the Driven By field, click the droplet icon and select Object B. A searchable list appears with Object B properties.
4. Select Y Location.
5. Set Multiplier to 2.000

We've just reduced a laborious 15-step process to a simple 5 steps. If the user wants to further tweak the driver F-Curve, he/she can click the Edit Driver Curve button, which opens up an F-Curve Editor.

Adding drivers could be so much faster and simpler

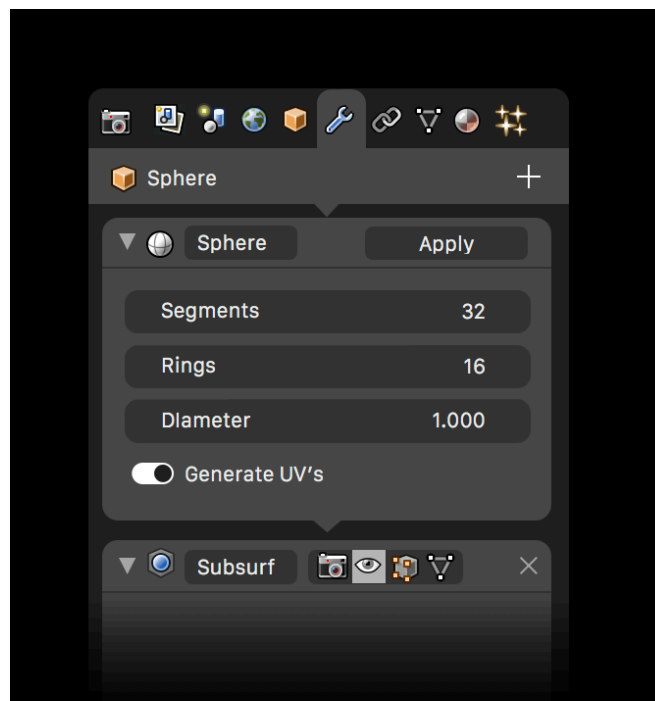


PARAMETRIC OBJECTS

Proposal While version 2.5 of Blender gave users the ability to tweak basic object parameters after adding items to the scene, these parameters currently get frozen as soon as the user performs another action. This makes the usefulness of this feature rather limited. We could use the 2.8 release to take a step forwards here.

There are several ways we could implement parametric objects, and a few issues to solve. Here's one way to go about it:

- When adding a new object, users get the same options as before
- This adds a new modifier to the modifier list that's stuck to the top
- Users can either apply this modifier, or enter Edit Mode to freeze the object. When entering Edit Mode, we could present a warning to let users know that they are now freezing the object parameters.



Parametric objects use the modifier system to give users persistent control over object parameters

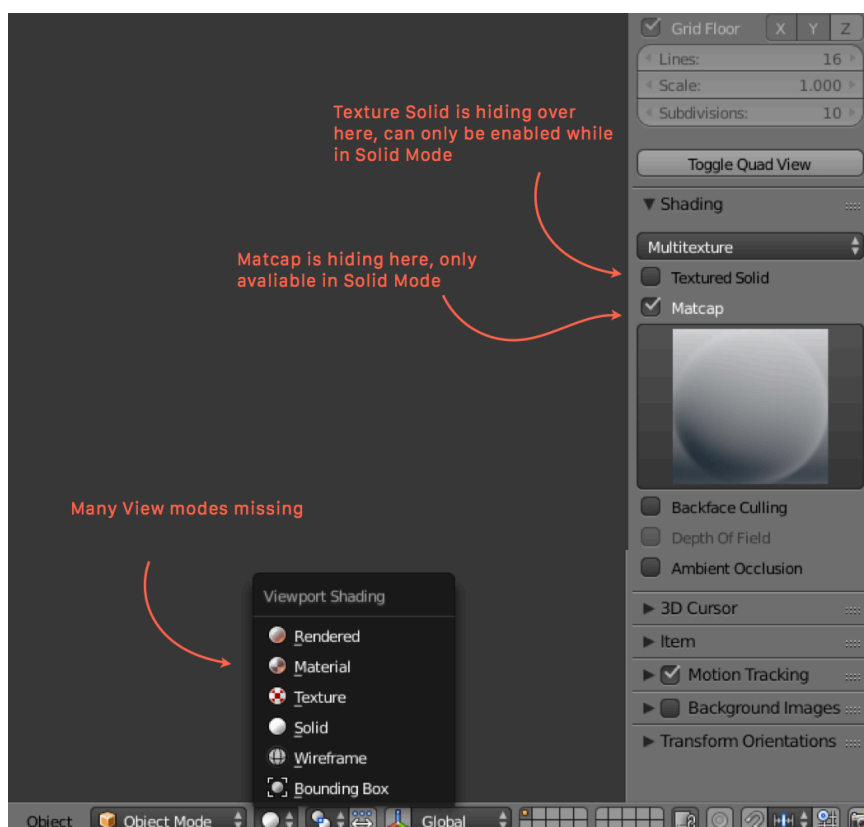
VIEW MODES

Analysis

Blender has many ways in which to change the way the user is able to visualise objects in 3D. However, this area is all a bit of a mess currently. There three main areas where the user can set this:

1. Viewport Shading menu in 3D View header (Bound Box, Wire, Solid, Texture, Material & Rendered)
2. Shading panel in N-key editor (contents of which confusingly changes depending on which Viewport Shading option is selected)
3. Display panel in Object Properties, which lets you set a 'Maximum Draw Type'

Not only are the view modes scattered about in the UI, the relationship between the various options is entirely non-obvious: Texture Solid and Matcap is only available in Solid mode, Shadeless is only available in Textured Mode, and the Maximum Draw Type only allows you to set a maximum quality level using the items in the Viewport Shading menu. Many things are also not currently possible: It's not currently possible to use Matcap for only one object - same goes for Textured mode which always applies to the entire scene.



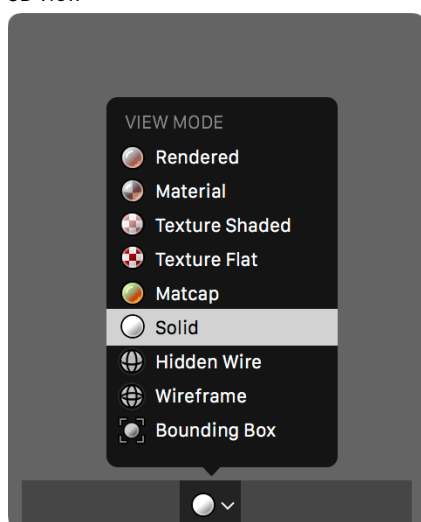
Proposal

All this can be simplified, cleaned up, and at the same time made more powerful. Here's how:

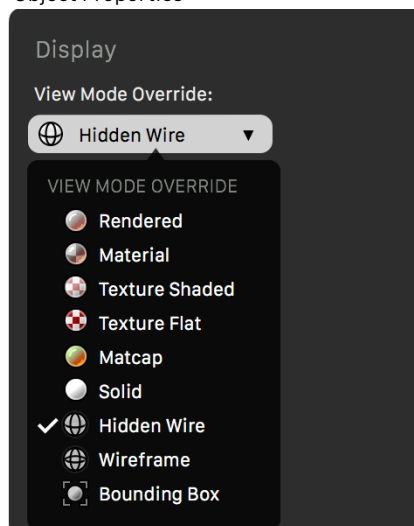
- Consolidate all the view modes, including Matcap and Texture Solid, into one place - inside the View Mode header menu. This is an area-specific option, so you can set one 3D view to be in Wireframe mode and another in Rendered mode.
- Remove the 'Maximum Draw Type' menu in Object Properties and replace it with 'View Mode Override', which mirrors the exact same options available in the 3D View View Mode menu.

This way it's both a lot simpler, because everything is consolidated into one place, and also a lot more powerful, because with View Mode Override, users can specify certain objects to be viewed in a specific View Mode. This way, you can view the entire scene in Wireframe, but set one object to Matcap.

3D view



Object Properties



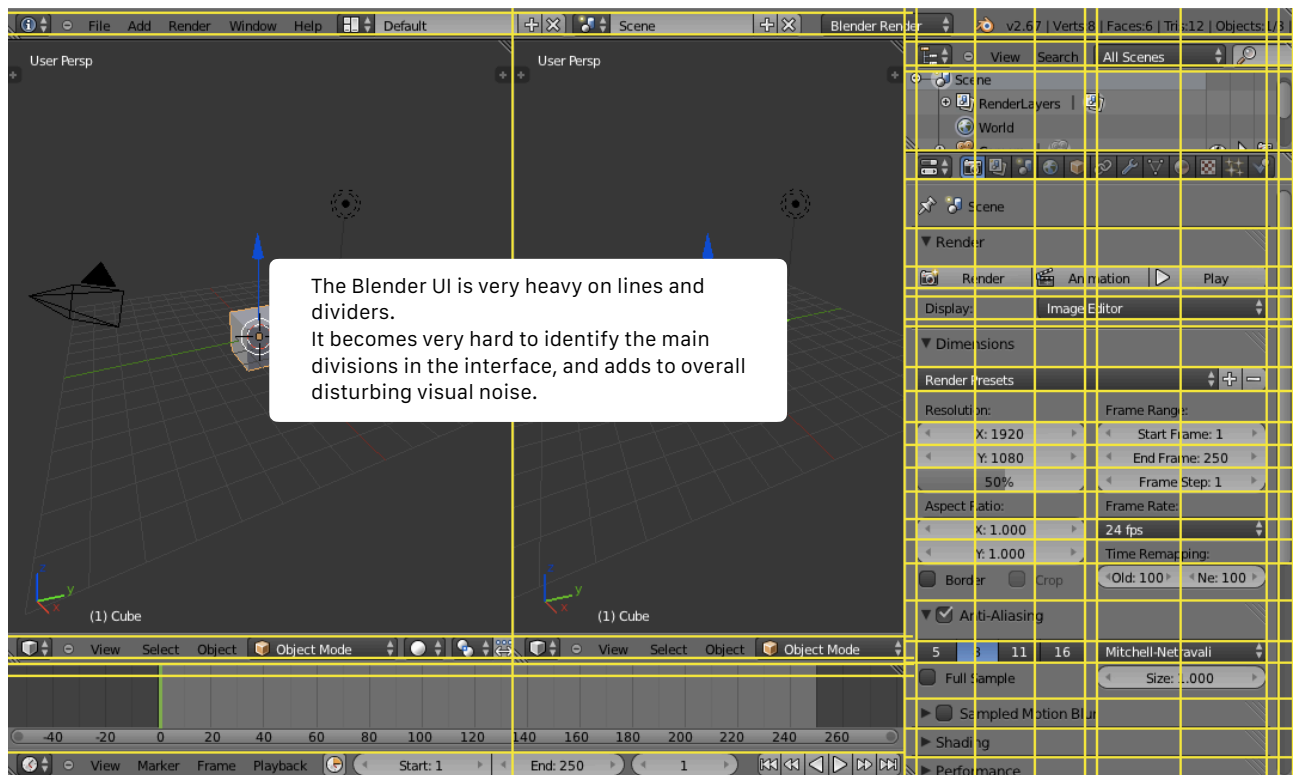
GRAPHIC DESIGN

Analysis

The up-coming move to 2.8 also represents an opportunity to revise the graphical, visual side of Blender. While user interface paradigms and workflows are more important, visual design also helps communicate functionality, and can also help artists focus on the contents they are producing by not stealing too much attention.

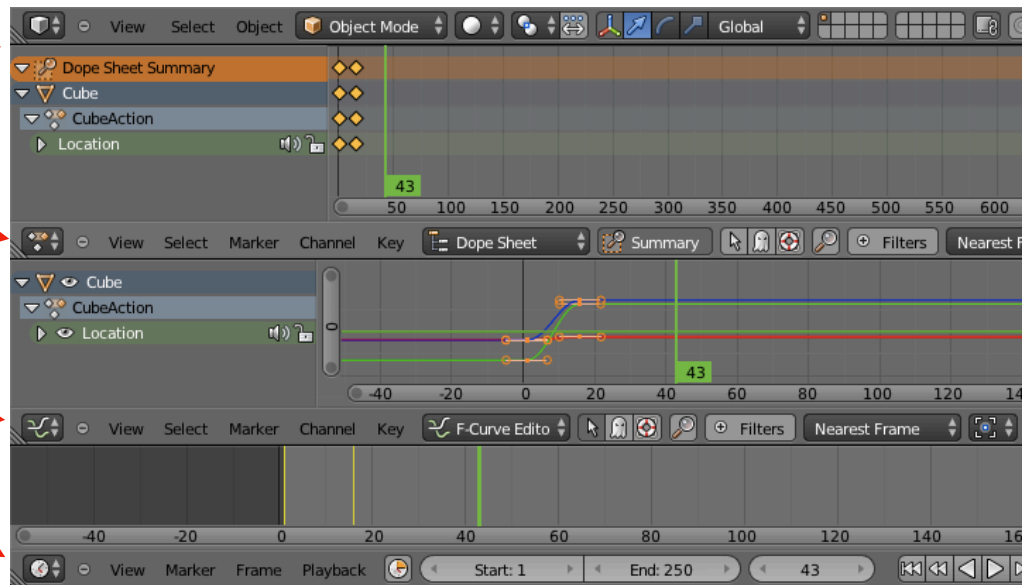
The default graphic design of Blender has stayed pretty much the same over the last decade, and this now gives us some perspective to critique it:

- The 2.5 UI is overly full of gradients, lines, arrows, boxes, making for an overly heavy, distracting and messy look
- The hierarchy of Blender is not well communicated visually: The Editor divisions are not clear enough, and it's too hard to see which editor headers belong to. This also applies to panels inside Properties, where headlines are not distinct enough from content
- There's a lack of distinction between radio buttons and tabs.





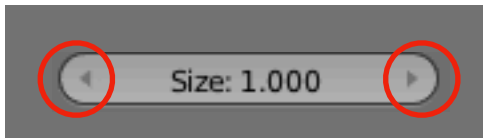
Hard to see which editor
each header belongs to



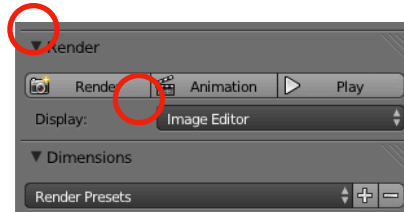
These controls are not needed at top level.
Can be placed inside the menu



This should be tabs



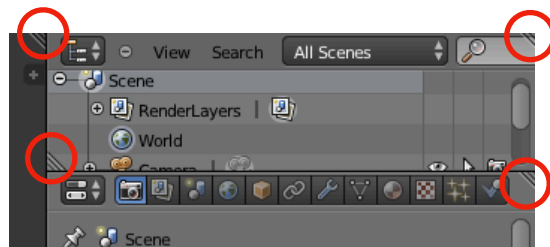
Arrows not needed unless you are interacting with the widget



Emboss effect doesn't work well with high DPI
retina displays



Groove not needed



Too messy, and the 'grabby' style is actually not
right, since you are not grabbing the corner

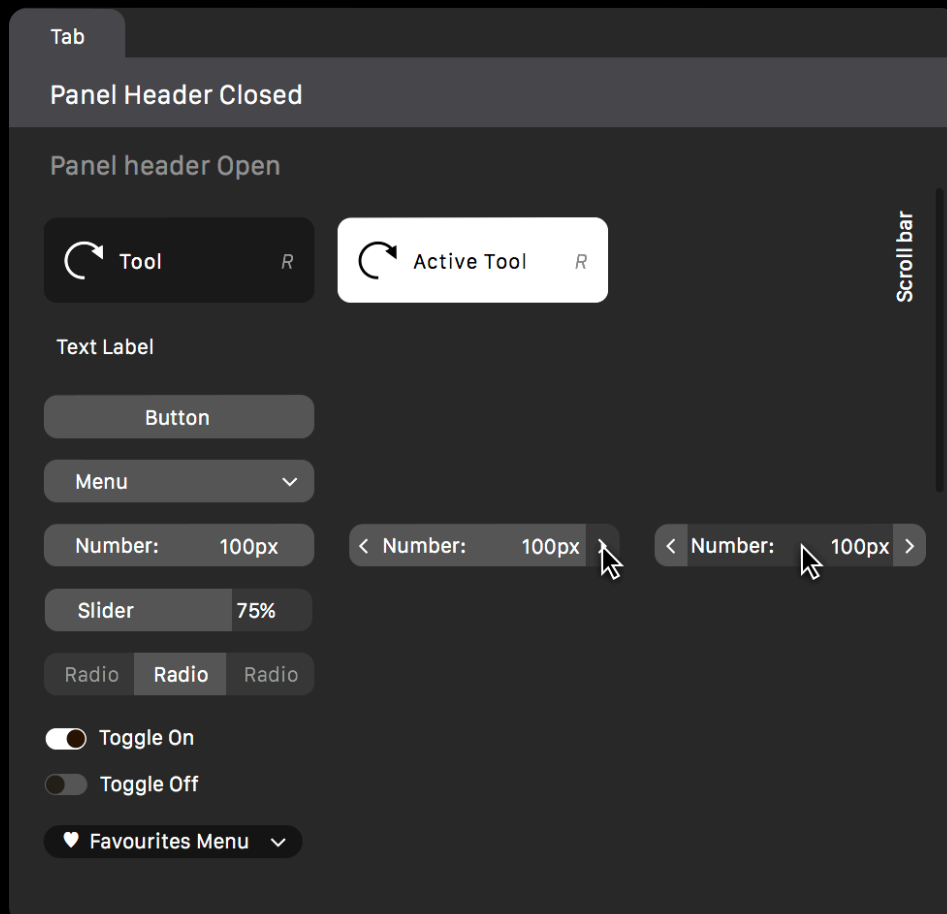


Proposal

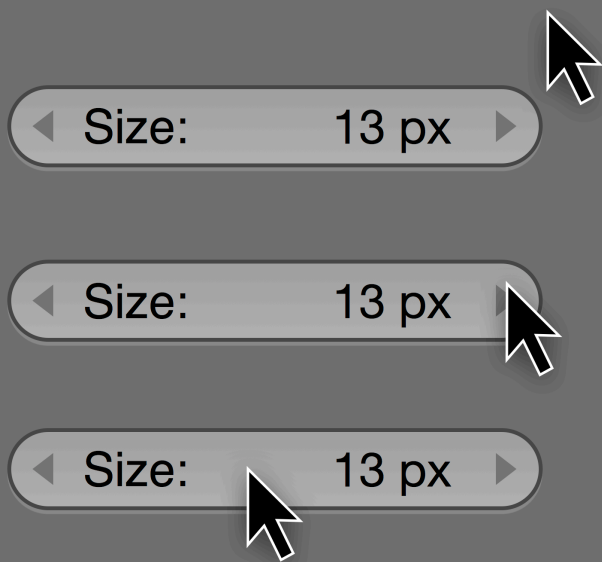
With our experience of using and seeing Blender being used since 2.5, here's how we can improve the visual design:

- We must calm down the graphical user interface elements and let them get out of the way and defer to user created content.
- Hide unnecessary graphical indicators when not in use, such as number field arrows
- Communicate information hierarchy using sizes and brightness.
- Use tabs where appropriate, instead of ratio buttons
- Simplified controls are more appropriate for high-DPI displays, where emboss effects don't really work.
- The simpler controls, dials and fields are quicker for OpenGL to draw on-screen.

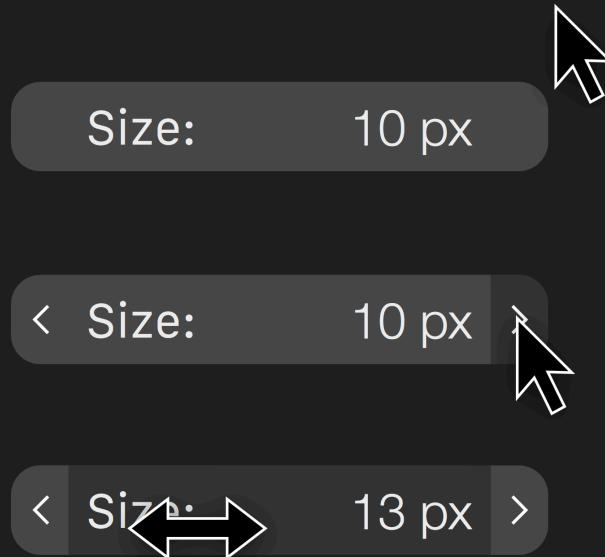
Here's how I imagine the basic controls could look:



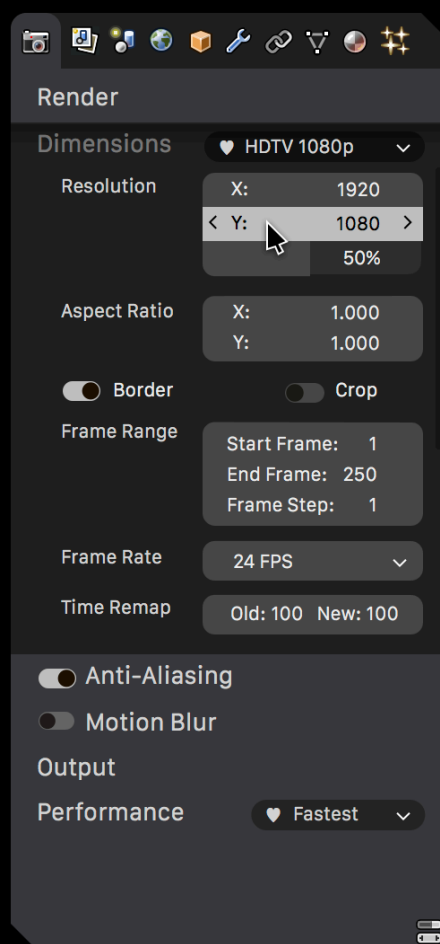
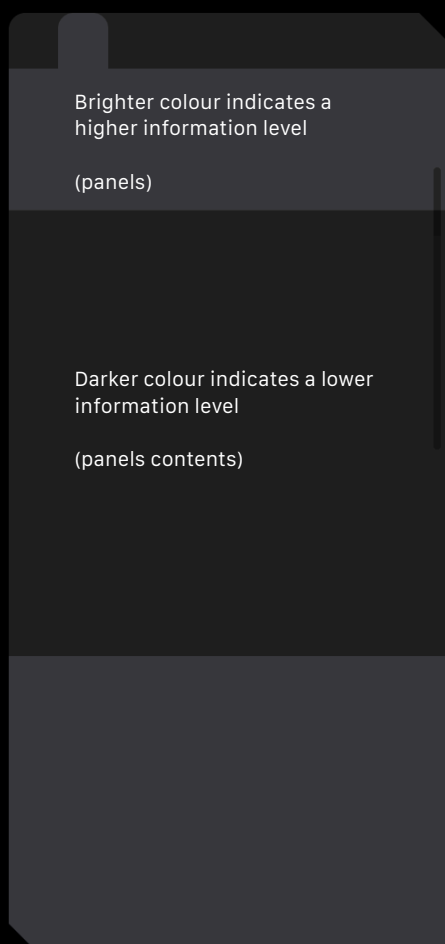
The simpler controls make a big difference when you have many controls visible at the same time. Together, the overall impression is much simpler and less busy looking.



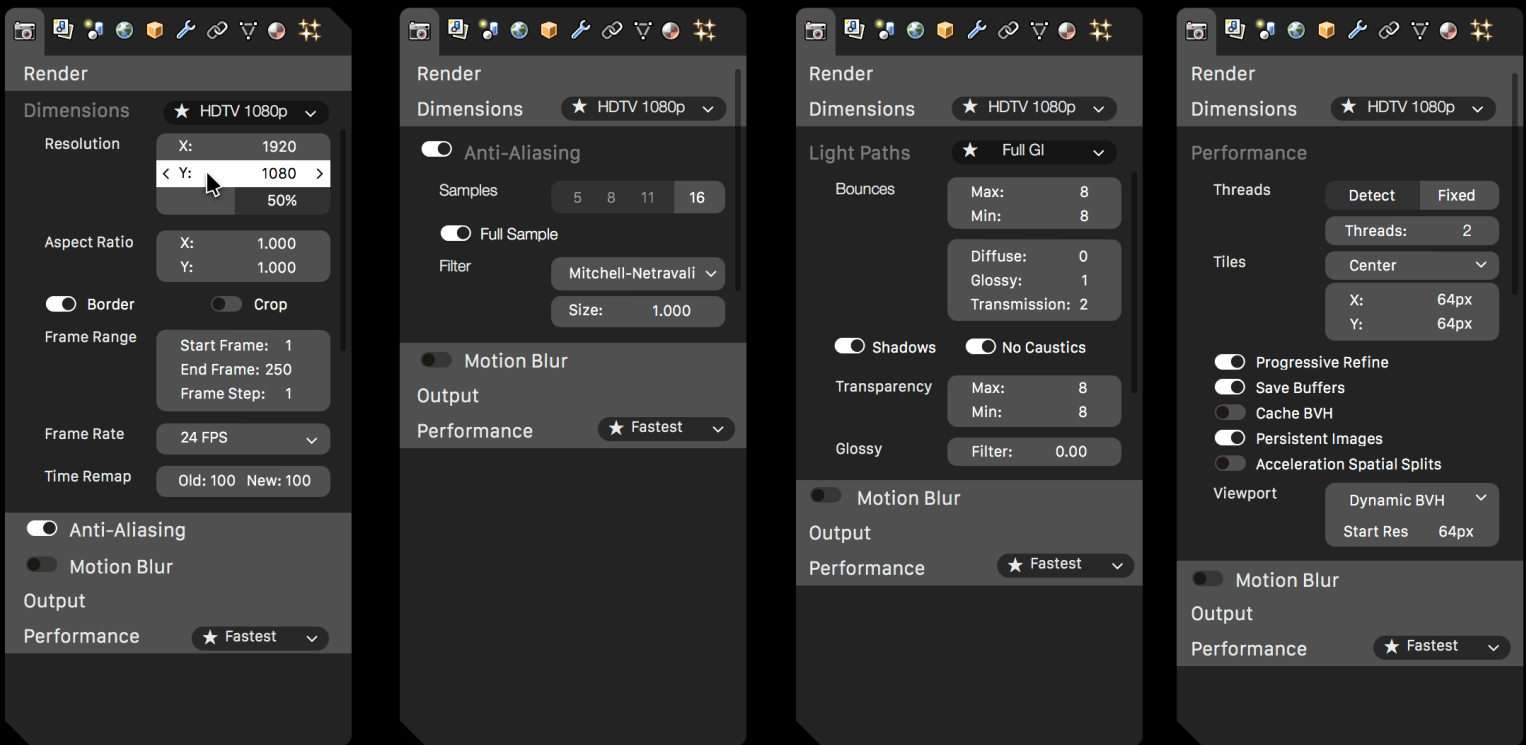
Before: controls look the same no matter where you hover



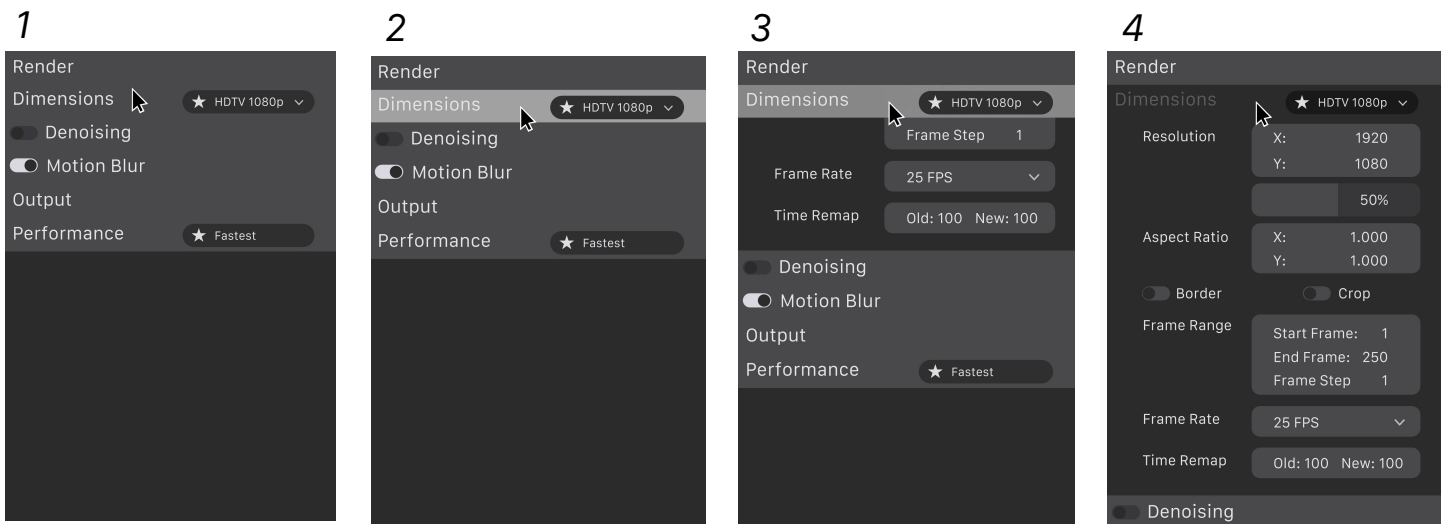
After: controls show progressively more information, and communicate the various hotspots in the number fields.



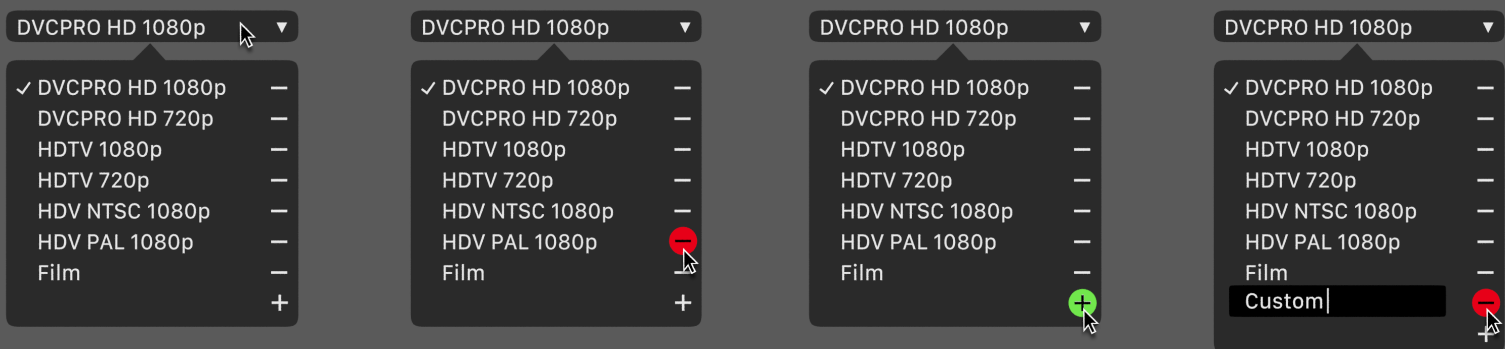
Clarified information hierarchy through visual cues



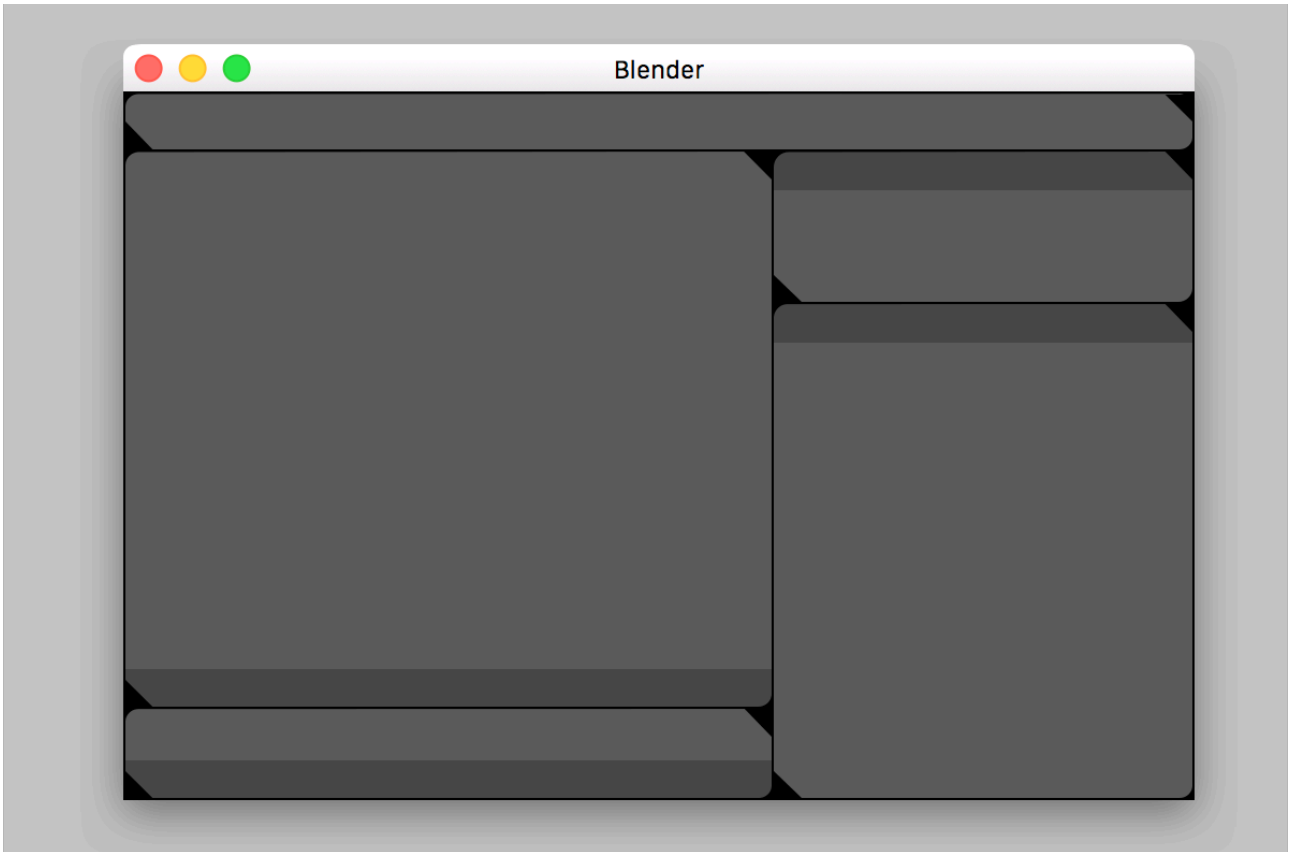
Properties



Opening panels transitions to a lower information level

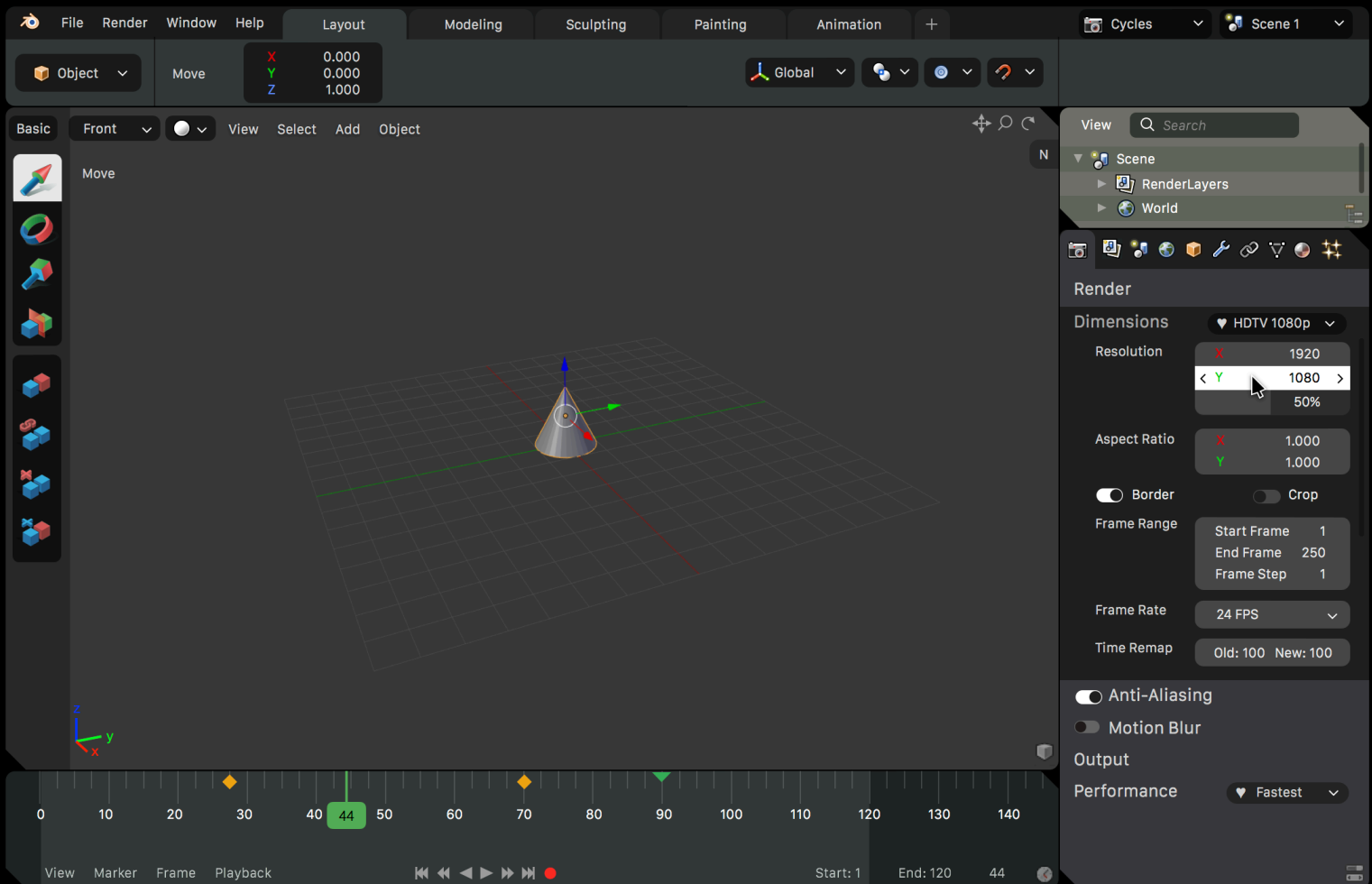


Dynamic menus communicate functionality with rollover highlight colours.



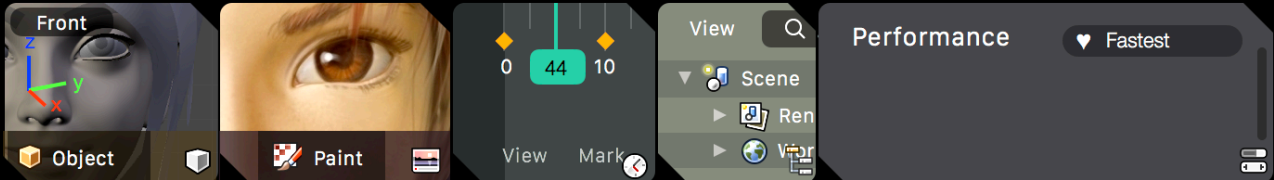
Area divisions are more distinctly separated than sub-areas inside them

Area distinctions are more clear. Global settings in global area. Simpler, cleaner controls. Tool icons by Aslam Cader





The headers and editors float together, and it's hard to quickly identify the various editors



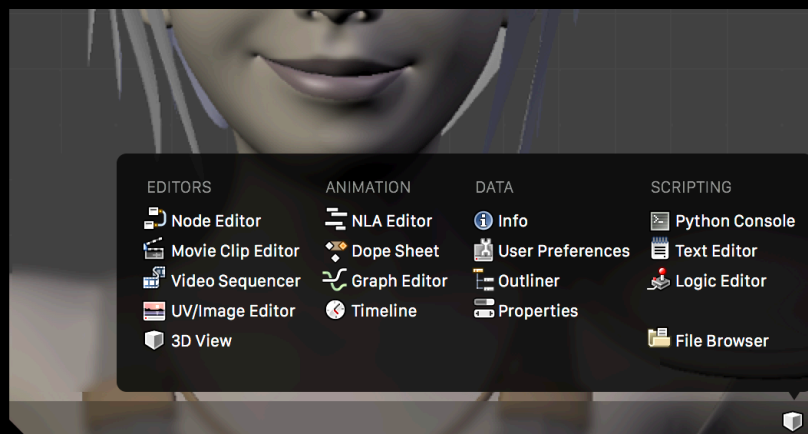
The rounded corners and editor colour tints, as well as the transparent headers, makes it easy to distinguish editors and to see the boundary between them

Rounded corners makes it easier to spot the boundaries of the editor

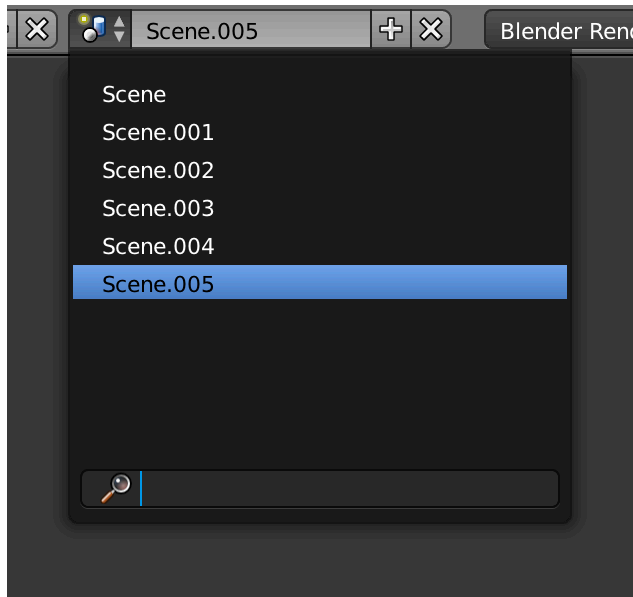
'Action' corner for splitting and combining editors

'Action' corner for splitting and combining editors

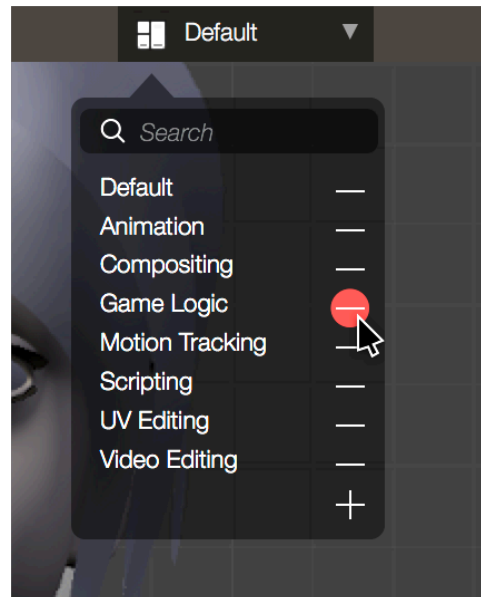
Slight transparency makes it clear which area the header pertains to



Editor menu broken into sections, a la modifiers



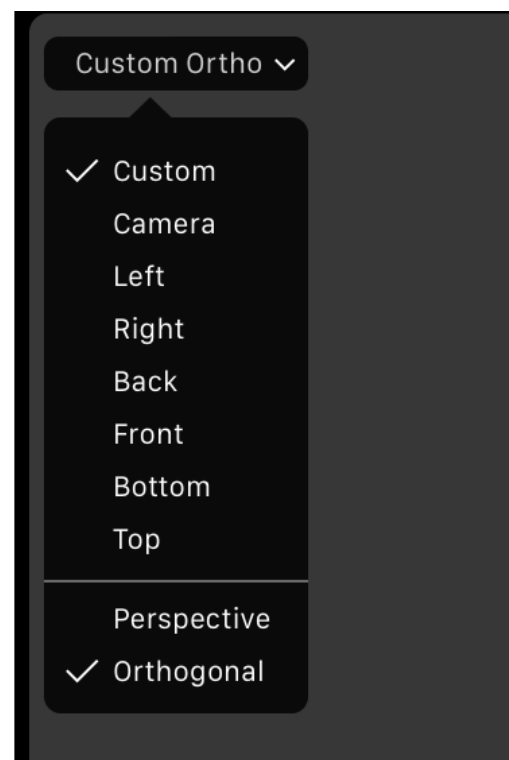
Before: Dynamic menus have always-visible + and - buttons at the top level, which looks busy. These controls are seldom used and can better be placed inside the menu itself.



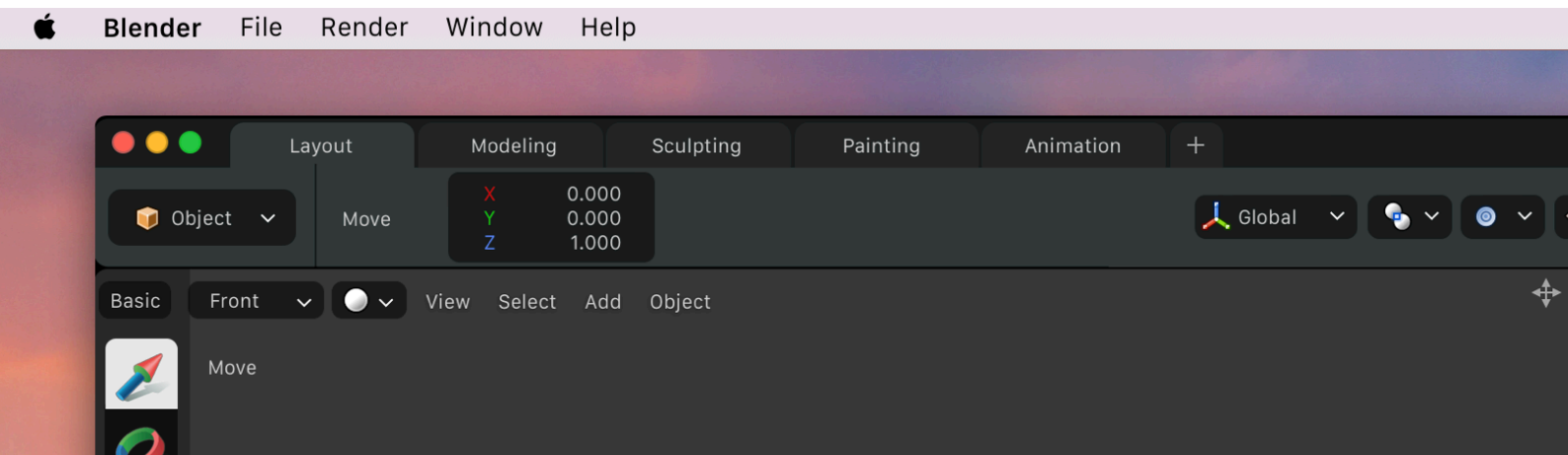
Dynamic menu with + and - inside.



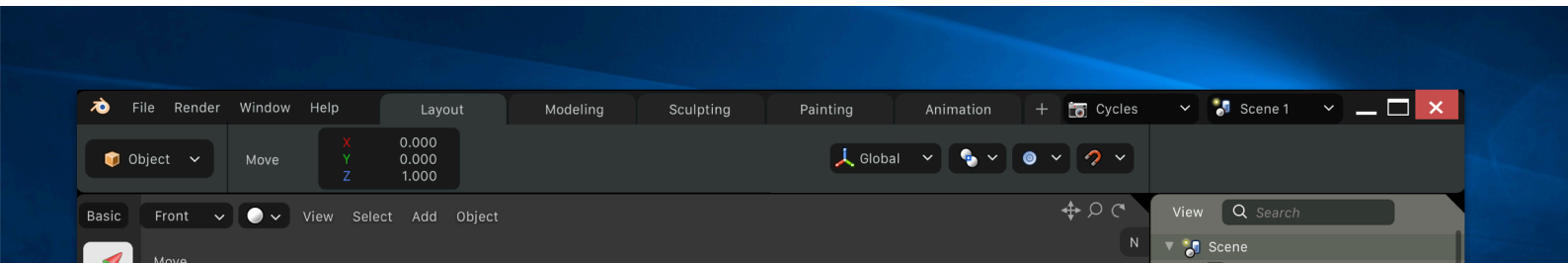
This is the top left of the 3D View. It tells you which view you are seeing, but you cannot use this to directly change it.



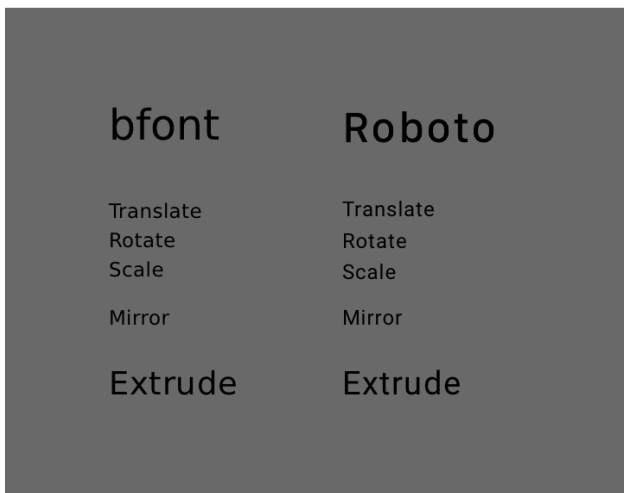
As illustrated here, this can be turned into a discreet menu, for a more direct connection between the label and setting.



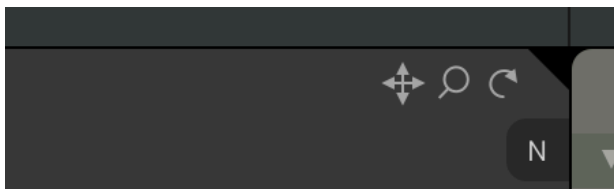
Blender using Mac menus and minimal app header



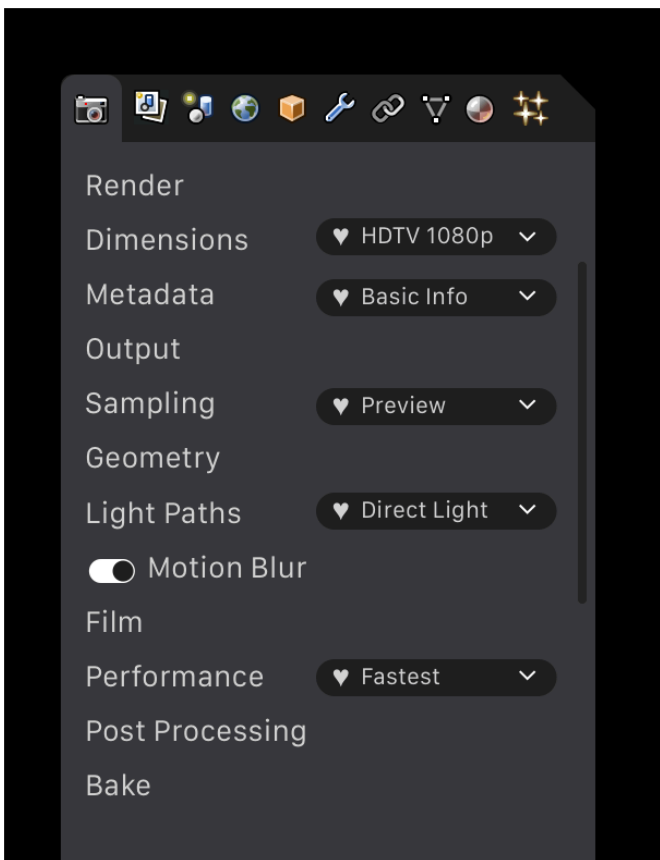
Windows version using platform conventions



The Roboto font is a excellent UI font that's more readable, under the Apache 2.0 licence. We could use it to replace the current bfont.



Discreet icons for panning, zooming and orbiting in 3D View.



Preset menus built into Properties headers. This makes it possible to set presets without having to open up each panel. Users could then keep more panels closed and avoid excessive scrolling.



MODIFIERS

Analysis

The user interface for modifiers needs some love. Main issues are:

- The order flow goes from the top to the bottom. This is not clear or communicated in any way.
- There's no way to quickly re-order modifiers by dragging them.
- The large arrows for re-ordering are clunky and take up too much space

Proposal

Here's how it could be improved:

- Add an indicator arrow in each modifier that shows how the result is handed down to the next item.
- Add ability to re-order modifiers, just like you can currently re-order panels
- Remove the re-ordering arrows

When you think about it, it's actually rather strange that re-ordering regular Properties panels, which is inconsequential to the app's functionality, is so much easier and more straight forward compared to modifiers, where re-ordering is more important. This is a smaller change, but the kind of quality of life attention to detail that would make Blender more delightful and communicative.

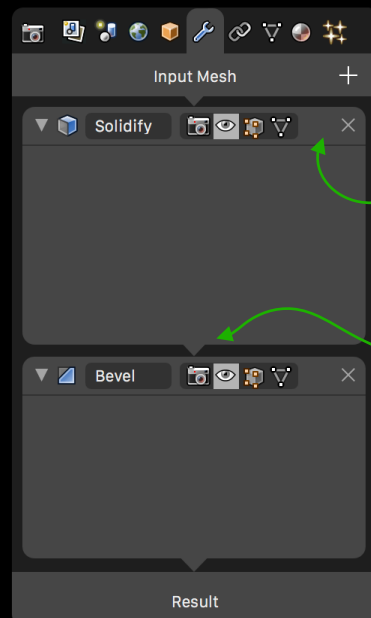


Arrows take up space, and make it impossible to quickly re-order modifiers

Instead, we should encourage re-ordering much more than with standard panels

There's no indication of evaluation order

Not enough visual separation



Add New, placed above remove X's

Drag to re-order

Clear evaluation order



EDITING MULTIPLE ITEMS

Proposal Blender has already gained the ability to edit values pertaining to multiple items. In 2.79, you can set a value, then right-click and choose *Apply to Selected*. Even better, in the 2.8 beta, you can simply type in a value with multiple items selected, and Blender will apply that setting to both automatically. This is a vast improvement, and is miles faster than before.

However, there's no clear visual indication to show differing values. This can be solved by using a visual indication, such as a dash, like so:



Controls in a state showing differing values.

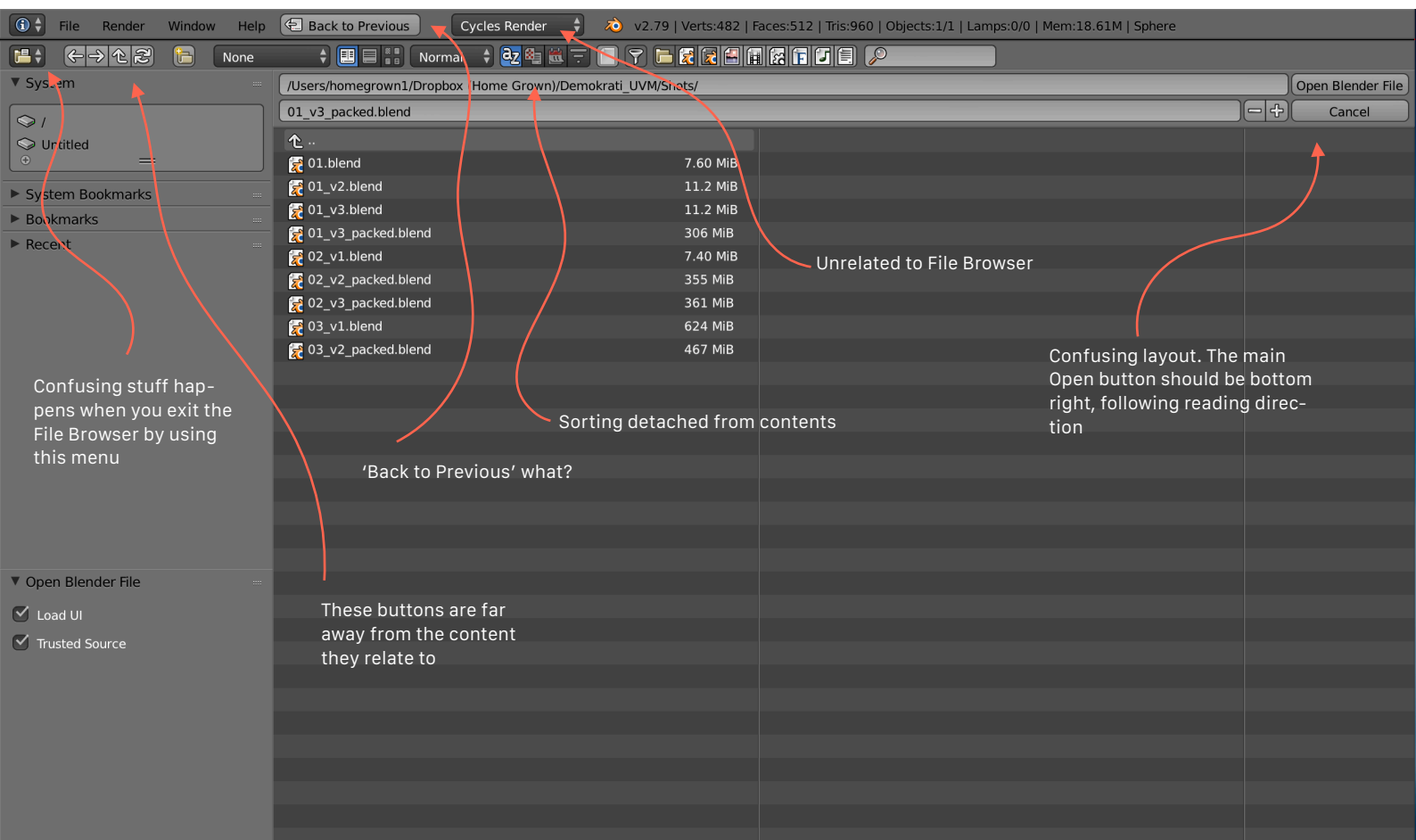


FILE BROWSER

Analysis The File Browser appears in Blender as a quasi-modal view, which takes over the entire window until you've selected your file or path, leaving only a strip at the top with a button saying Back to Previous, which returns the user to the state he/she was in before browsing.

This has some confusing ramifications however, as listed below:

- If, while the File Browser is open, you change the editor view from File Browser to, say, 3D View, you are stuck in a weird state. The 'Back to Previous' button is still there, and if you click it, you are yanked back to a previous layout. Additionally, the Info area then becomes a File Browser :) Bizarre!
- If, after going to open a file, the user closes the window, Blender itself quits
- You can still change the renderer inside the Info header while file browsing, even though it's completely unrelated
- The File Browser doesn't reflect design conventions of either Windows, MacOS, or any popular Linux distro, making it a rather jarring experience.

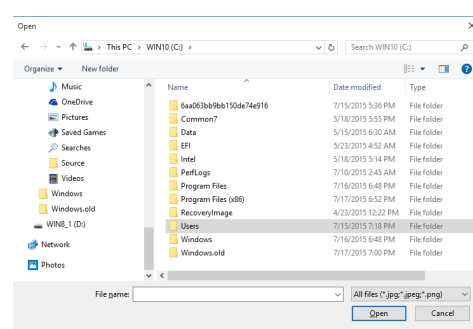
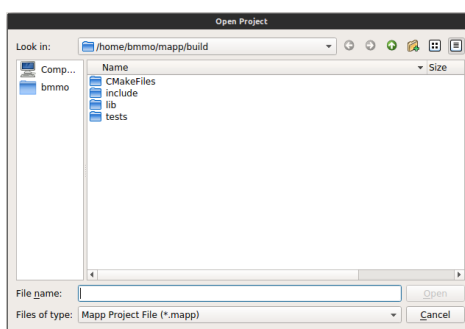
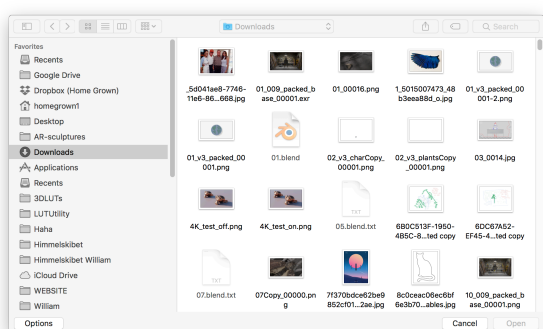




Proposal

- For something like file browsing, which blocks the UI, we should spawn a new window with the file browser. This way it's very clear how to get back (close the window), and you can't get stuck in a bizarre state. It's also clear what parts of the UI pertain to file browsing, and what is part of the normal UI chrome.
- Additionally, the layout of the File Browser is somewhat in need of scrutiny. The placement of buttons and controls are rather arbitrary.

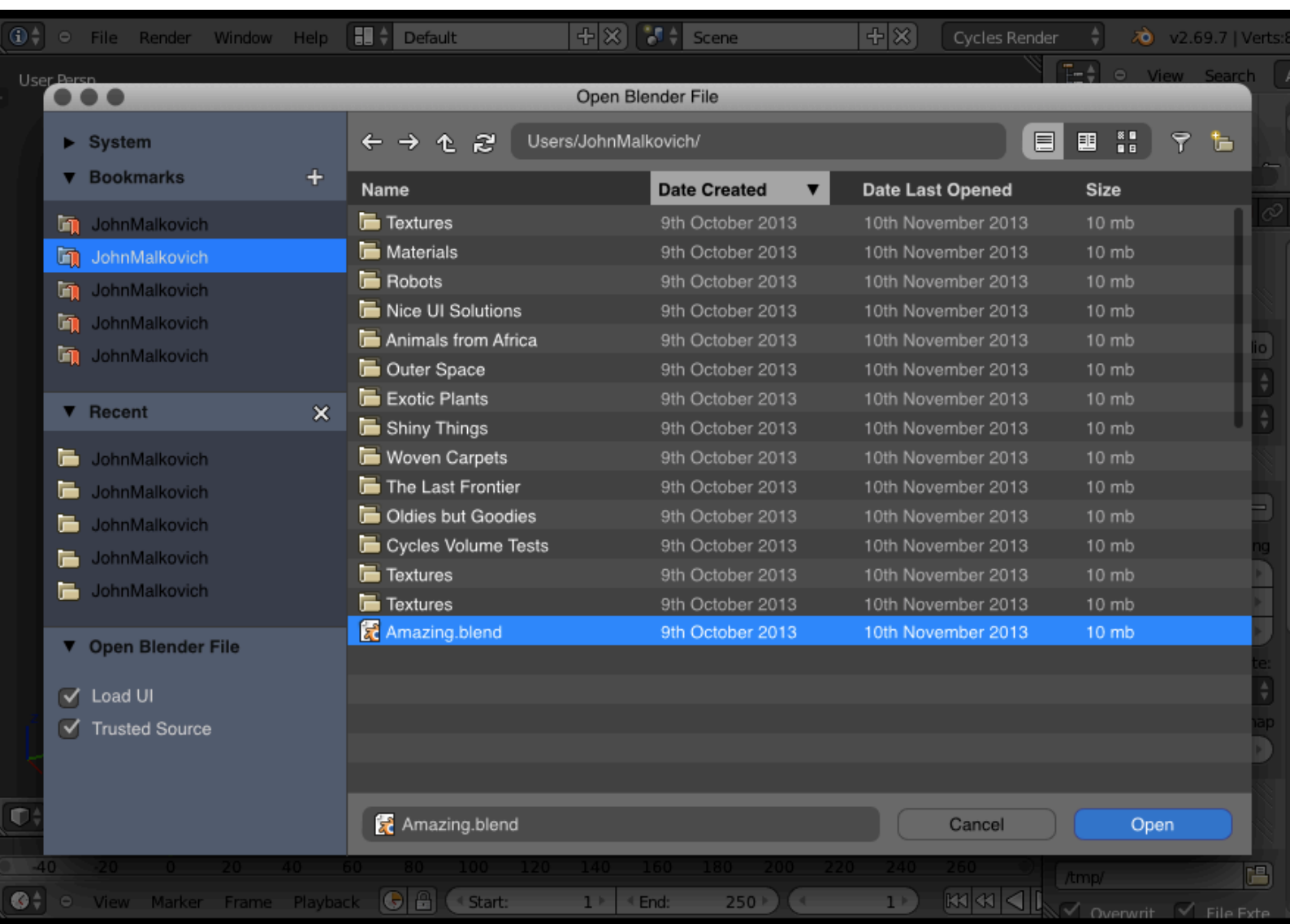
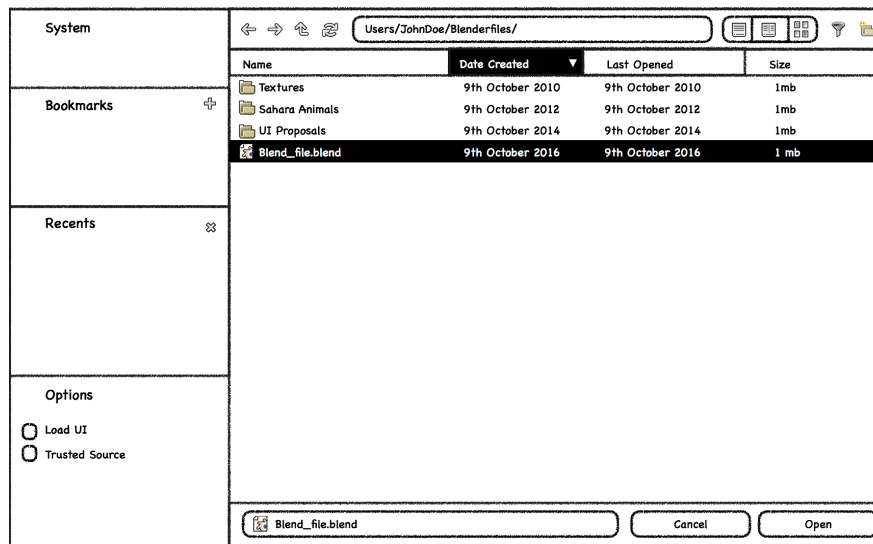
Here's a sample of what Open dialog boxes look like on the three major OS's:



As you can see, they are all fairly similar. Open/Save button is in bottom right. File path at the top. Name along the bottom. Sorting is done by tapping the relevant column. There's really no good reason why we shouldn't adhere to these general principles in Blender.



Here's one way to do a layout that more closely follows general conventions of the examples above:

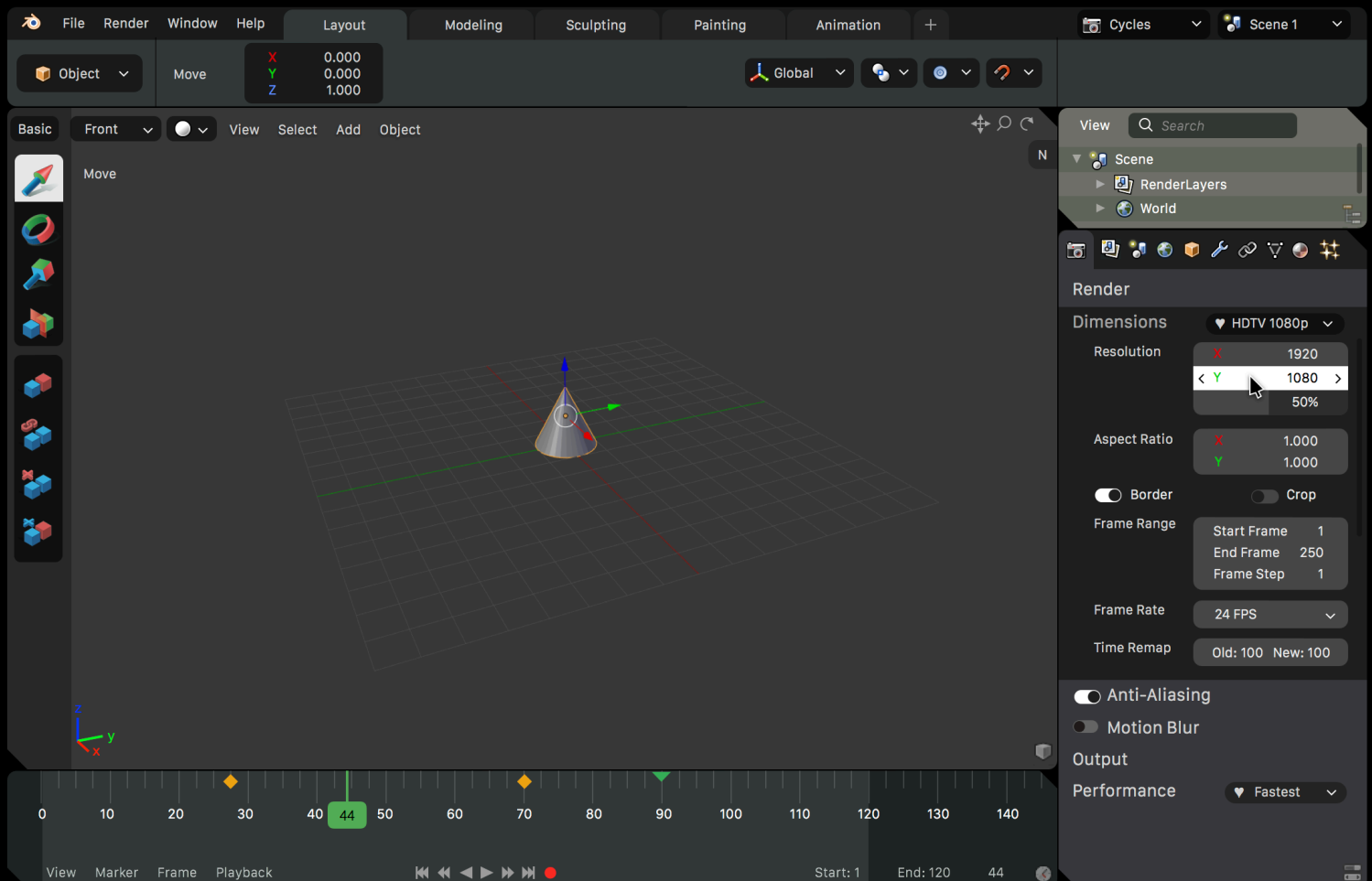




CONCLUSION

Hopefully, this document will help focus attention on some of the core, basic aspects of the Blender UI that still feel unfinished, broken or awkward. While there are many other things one could improve, clean up and fix, the basic interaction paradigm regarding tools, tool settings and tool manipulators is, to me, the area where Blender most severely needs attention and care. It's part of the scaffolding that many other tools will be built on during the next many years of Blender development, and, more than likely, there might not be a better time to actually get it right than now. The good thing is that, with persistent attention to detail and more well defined conventions, Blender's UI has the potential to become truly *excellent*.

William Reynish



Tool icons by Aslam Cader



REFERENCES

This wasn't written in a vacuum, but in a dialog and coinciding with other UI proposals and docs for Blender. Here's a list:

Blender UI Analysis

https://wiki.blender.org/index.php/Dev:Ref/Proposals/UI/Analysis_Bcon13

UI Workshop Write-up

Julian Eisel

https://wiki.blender.org/index.php/Dev:2.8/UI/Workshop_Writeup

UI analysis and challenges

Brecht van Lommel

<https://www.youtube.com/watch?v=ziPLNUfm7KA>

UI Proposal: Top Bar Reshuffle

Brecht van Lommel

https://wiki.blender.org/index.php/Dev:Ref/Proposals/UI/Top_Bar_Reshuffle

New Blender UI - The Proposal

Andrew Price

<https://www.youtube.com/watch?v=UWacQrEcMHk>