

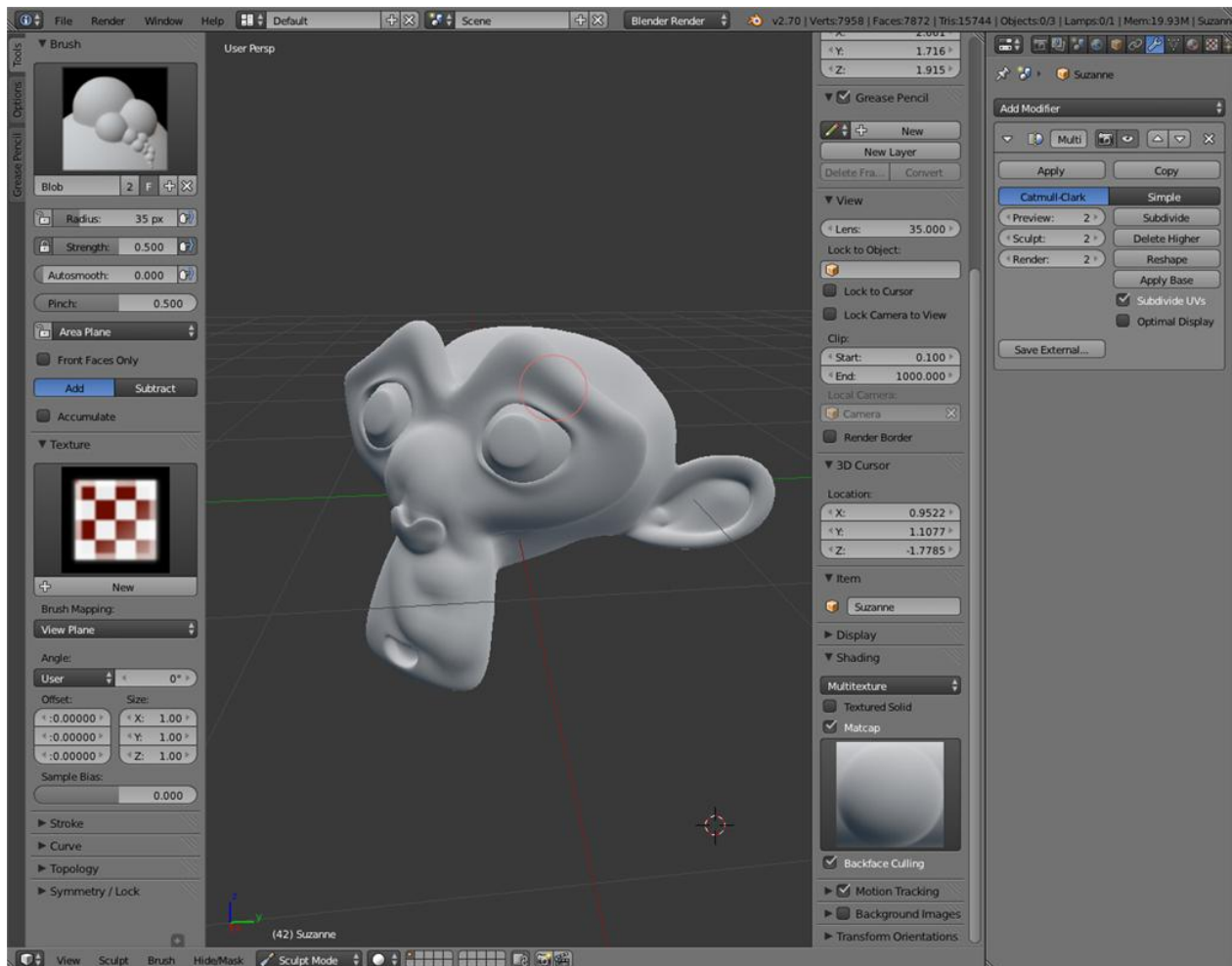
A Proposal for the Floating panels and the redesign of the T and N shelves.

courtesy of Paulius Mscichauskas (FreeMind)

The goal of this proposal is to make sure that the users are free to setup their layout to save as much screen space as possible. These additions would further improve the blender GUI without altering the current functionality and will add more customizability for blender users to make blender fit their unique workflow.

The Issues

Fig 1: A screenshot of blender as it is now (Blender 2.7) on a 1280x1024 screen, during sculpting

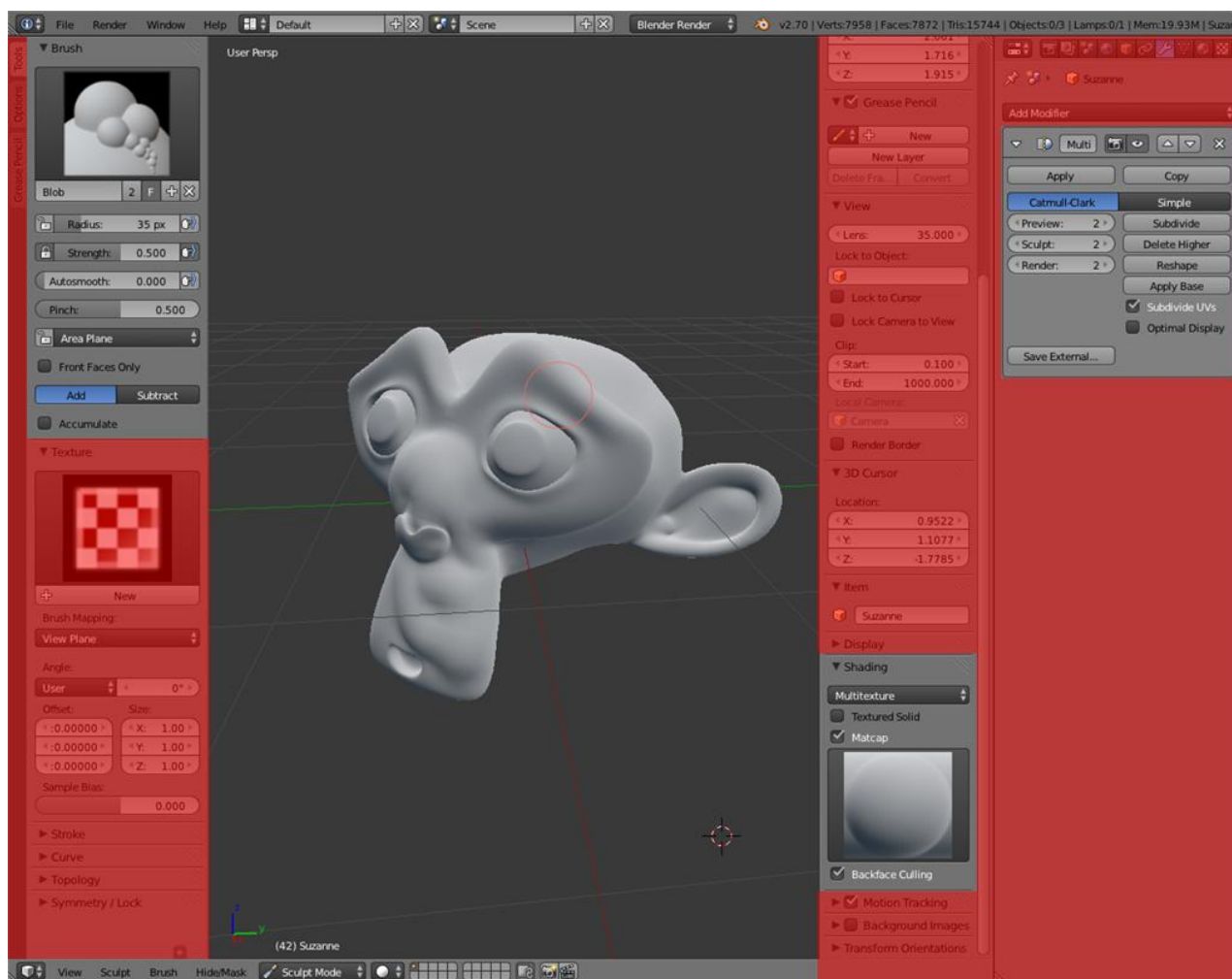


Let's say, in this scenario, the user is only focusing on these things: The 3D view to sculpt, The Brush panel to select his brushes, the Shading panel to switch his matcap and the multiresolution modifier to set his levels.

Issue 1 - The user is forced to see things he doesn't want or need to.

In order to see the multiresolution modifier, the brush panel and the shading panel, the user has to keep the "T" and "N" shelves on in the 3D view. Also, he is not allowed to go full screen on the 3D view, otherwise he wouldn't see the multiresolution modifier. There is no better, more compact way to display all of the mentioned wanted items at once. Such restrictions lead to seeing a lot of extra properties and buttons that this particular user does not need for this specific workflow. This makes the screen look and feel chaotic and overwhelming. The extras take a lot of space, especially on such small screens as 1280x1024 (They are still used widely).

Fig 2: Wasted screen space for the specific sculpting workflow by unwanted properties, marked as red



Issue 2 - 3D view size.

Because all of the extra menus bite the space out of the 3D view, it becomes quite tiny, longer vertically than horizontally, which is a bad aspect ratio for human eyes. This is not pleasant to work on.

Fig 3: Claustrophobic 3D view



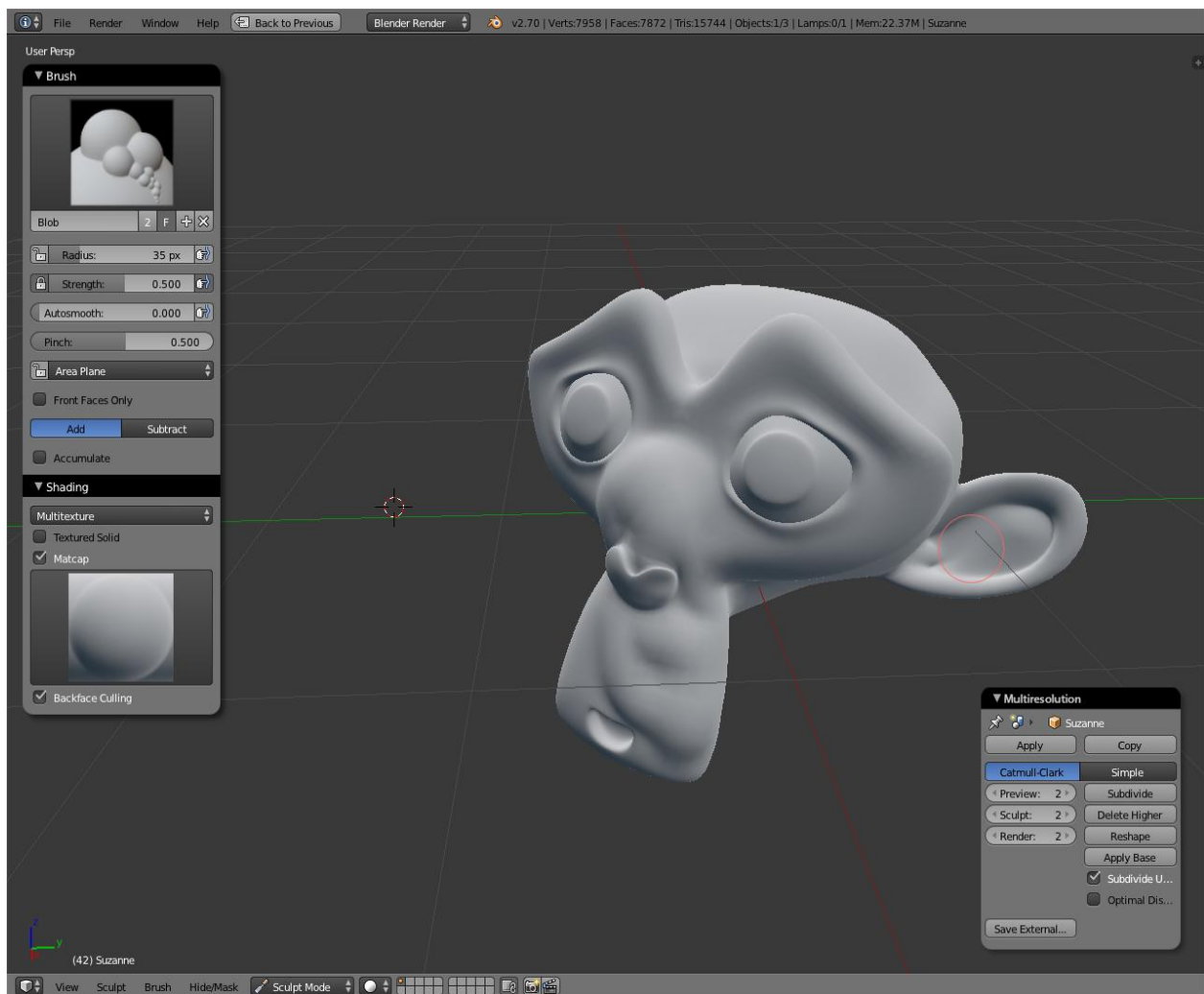
Blender has the "Region overlap" setting in the preferences, which helps, but it's still not the best solution. Even if the "T" and "N" shelves obscure the 3D view, instead of biting space out of it, the 3D view still feels narrow.

The issue of the small and narrow 3D view is far less dramatic on bigger screens, but 1280x1024 screens are still widely used. The proposed solution would benefit everyone and fix both issues.

Solution

The following will describe how blender could be designed to allow the users to lay out a workplace with not a bit of wasted space, no unwanted things to see, and a full screen 3D view. These changes will keep the essentials of the blender design intact. No current functionality would be crippled. Current workflows would be altered minimally. With these changes, there wouldn't be a "This is not blender anymore" feeling.

Fig 4: A screen layout customized by the user for his specific workflow. No space wasted. Full screen 3D view. Only wanted items for his specific sculpting workflow are displayed



Step 1 - The reorganization and redesign of the "T" and "N" shelves.

The following changes of Step 1 are not necessary for proposed new customizability options to work, but they are made with further proposed changes, content consistency, space saving and respect for old workflows in mind.

All of the panels that are related only to the scene or the selected object should be moved to the properties editor, where they belong. It's not consistent to have some of these with 3D view settings in one menu, and some in the properties. Such panels as "3D cursor" and "Grease pencil", would move to the properties editor, because they are global scene settings, not 3D view specific settings. Such panels as "Transform" and "Item" would be removed because they are duplicates of what already exists in the properties editor. The "N" shelf at this point should only have properties that affect that one specific 3D view editor, such as "Background images", "View", "Display" etc. For the current version of blender, it's important for these panels to be reached from the 3D view, but the reason for such changes is that the proposed new customizability options would make having these panels in the "N" shelf redundant.

Everything from the "T" shelf could be moved to the "N" shelf as a separate tab. We should have two tabs in the "N" shelf at this point: "3D view Settings" and "Tool shelf". "T" shelf is no longer required and can be removed. The reason why the "T" and "N" could be merged is that having more than one shelf in any editor and having the further proposed customizability would be a bit of an overkill.

There could be a third tab in the "N" shelf, which could be called "Custom". More information about this tab is written further in the paper.

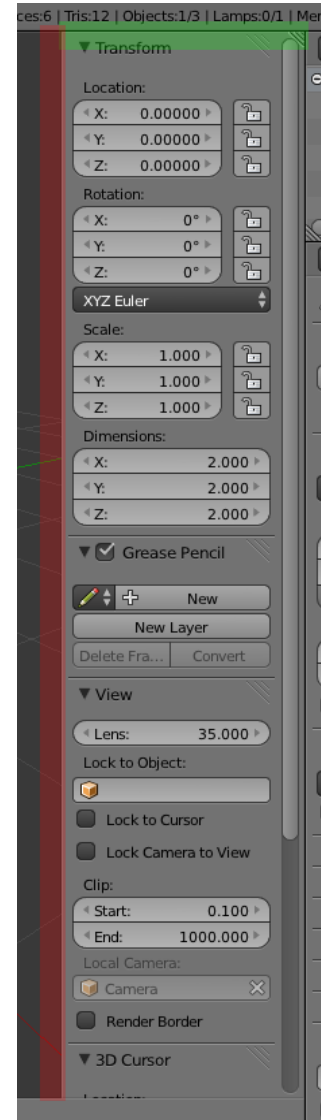
The tabs should not be vertical as it is done in Blender 2.7, because vertical tabs overall take far more space than horizontal tabs. Vertical tabs take horizontal space, which blender has a lack of.

Fig 5: Vertical versus horizontal tabs. Space occupation **comparison:**

Tabs should use icons, not text, because icons take far less space and are easier to recognize. Horizontal tabs that are displayed as icons are consistent with the rest of blender, for example - the properties editor. Blender should be consistent within itself.

Separation of "tool shelf" panels to avoid scrolling, as it is done in Blender 2.7, could be done by adding more horizontal tabs that only appear when the "Tool shelf" tab is active.

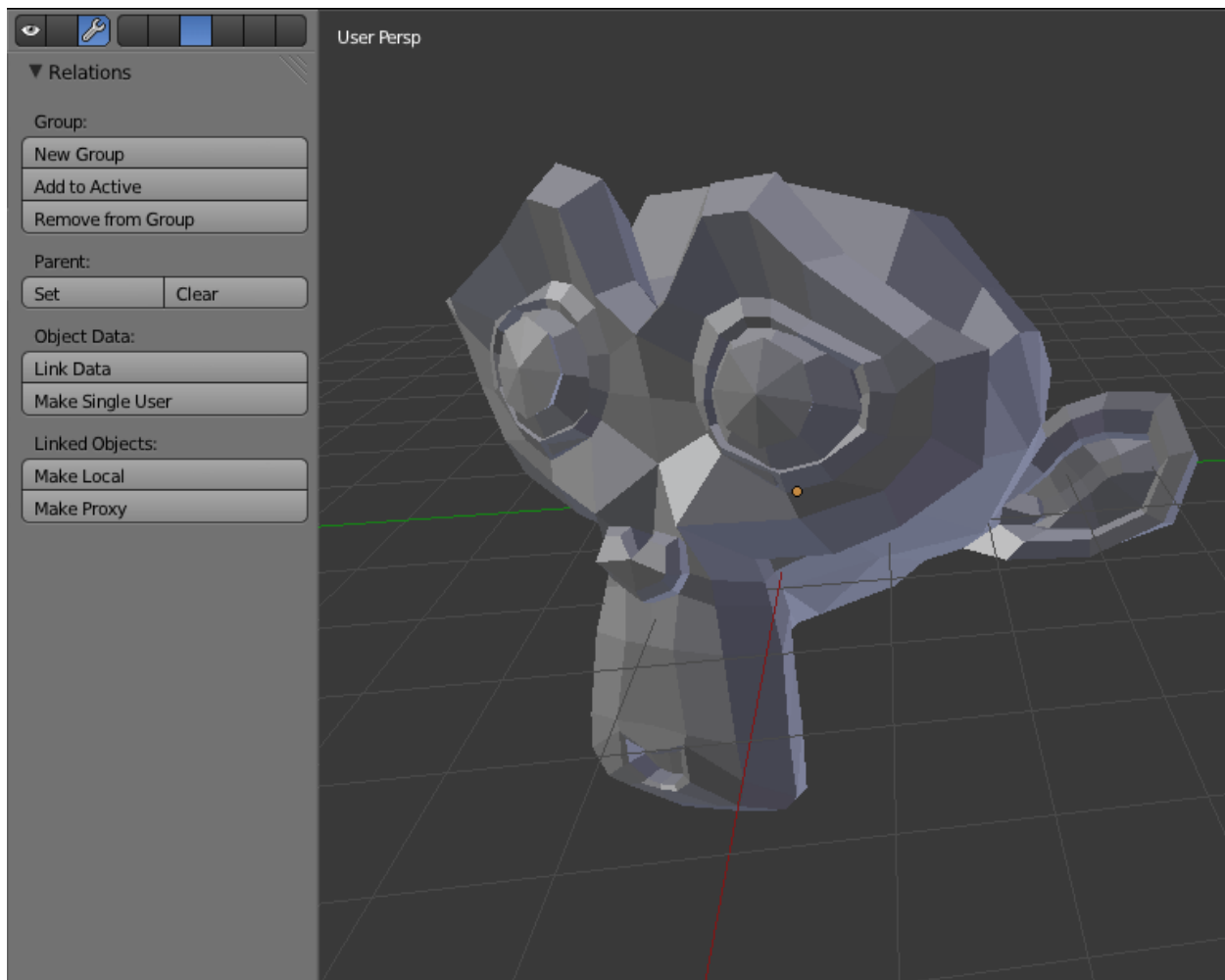
Scrolling wouldn't move the tab buttons. They would stay stationary. Like the header of the properties editor.



Context sensitivity of the "Tool shelf" and "3D view settings" would remain as it is currently.

Every editor that has "T" and "N" shelves should have these changes made to keep blenders interface consistent.

Fig 6: The merged, redesigned "N" shelf on the left. Tabs are displayed horizontally (Top left) as icons. The mockup lacks icons. From left to right, they are: 3D view settings, Custom, Tool shelf, tools, Create, relations, Animation, Physics, Grease pencil. The last six tabs are only available when the "Tool shelf" tab is active. The space between tab buttons indicate that the further tabs are dependent on the main tabs. Order of tabs may differ. The mockup displays the "Tool shelf" tab active, with the "Relations" subcategory for that tab.



Step 2 - Panel design.

In the current default panel design, each panel is only separated by thin lines. The headers only stand out because the text slightly differs in size, in comparison to other text. Header text is not aligned because of checkboxes on some of them. Panels don't have any boundaries. Mediocre design for fast eye scanning. Barely usable design for further proposed floating panels.

Repositioning behavior makes sense for the current panel design. There is no need to reposition the panels that often in blender, so the whole header length is dedicated to panel expansion/contraction. Repositioning is done by and dragging the triangle area on the right of the header.

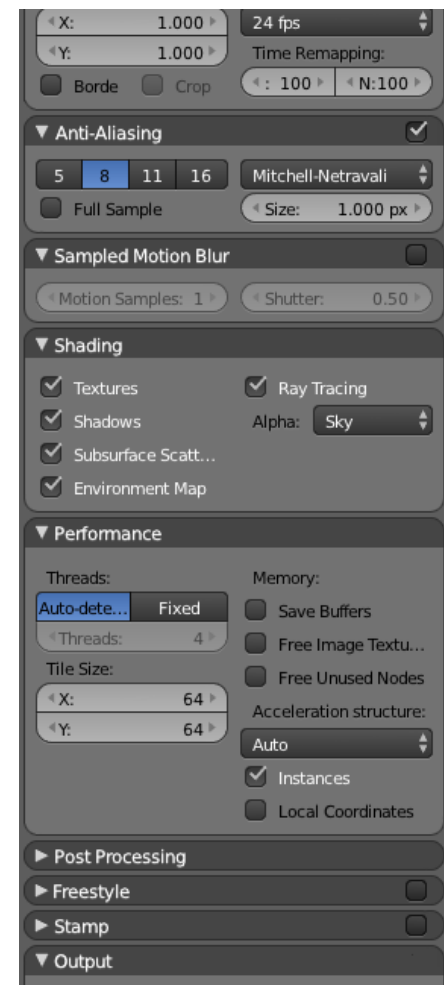
Fig 7: Current default panel design.

The default design of panels should aim for the panels to be clearly separated one from another and to have obvious headers. The GUI should clearly communicate that panels are separate entities one from another. The checkboxes on headers could be on the right side, so the header text could be aligned. Rounded corners could help communicate that there is more to scroll to, when all panels don't fit vertically. it's best if the same panel design would work for floating panels, so that the GUI communicate that panels in menus and floating panels are the same.

Just a basic design direction is proposed. Colors, shading, sizes and content are not proposed. The design should still differ from nodes to some degree, because floating panels should work in the node editor as well.

When floating panels are introduced, the grabbing of panels becomes far more relevant, so the behavior should adapt. Suggestion is, that the whole header area was for both, expansion/contraction and grabbing. Clicking on the header would expand or contract, clicking and dragging would move the panel.

Fig 8: Speculated panel design. This design will be used for further mockups.



Step 3 - Introduction of floating panels.

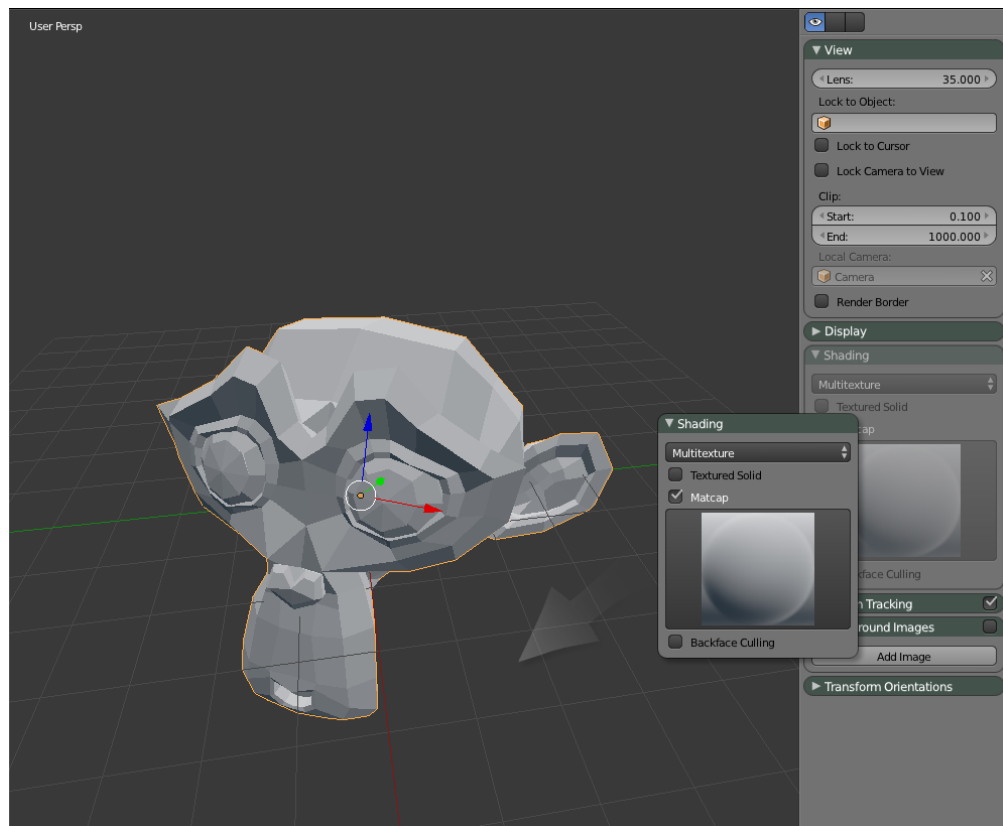
Floating panels is an UI change that will allow the user to lay out his workplace without any wasted space or unwanted things to see. This will give complete freedom of what and how many panels are displayed on the screen. This will bring comfort. For example, in a scenario where the user is sculpting with a multiresolution modifier, the user will no longer be forced to leave the 3D view to change his multiresolution level. Floating panels will also significantly reduce scrolling, time searching through menus and mouse movement distance. The user will have the ability to stay full screen in the 3D view and reach everything he needs for his specific workflow (See Fig 4)

Any panel from the properties editor or the "N" shelf would be able to become a floating panel for the 3D view. Also, any other editor that has the "N" shelf would have the ability to make any panel in the shelf become floating in those editors.

The user will not be allowed to add a panel from one editor to the other. For example, the user will not be allowed to place any panel from the UV editor into the 3D view as a floating panel. The only other editor the 3D view could get a panel from is the properties editor.

A user can make a panel floating by simple drag and drop. For example, dragging the "Shading" panel from the "N" shelf and placing it on the 3D view. Panels that are made floating do not disappear from their original location. Having floating panels does not cripple the functionality of panels in menus by any way. The user has the ability to make as many duplicates of a floating panel as he wants by dragging the same panel out multiple times. When the user has "Shading" as a floating panel, he is free to get the "N" shelf out of the way by hiding it with the "N" keyboard shortcut.

Fig 9: Dragging out the "Shading" panel from the "N" shelf, 3D view settings tab, into the 3D view.



Floating panel context sensitivity

Since the "3D view settings" and the "Tool shelf" are context sensitive, the floating panels are as well. The layouts for floating panels are completely separate entities for different modes ("Object" mode, "Edit" mode, "Sculpt" mode, etc.). So, for example, in the "Object" mode the user may have two floating panels - "Transform" and "Last operator", but when he changes it to "Sculpt" mode, he sees a different layout of floating panels - For example, "Brush" and "Multiresolution". Any newly created floating panels only remain in the mode they are created.

It's impossible to create any floating panels that are unrelated to the current mode, because, since the "3D view settings" and the "Tool shelf" are context sensitive, there is no place to drag and drop unrelated panels from onto the 3D view area.

Object selection sensitive panels, such as panels in the Properties > Object, would remain object sensitive as floating panels as well unless they are pinned. (Look for the pin icon on the multiresolution modifier floating panel in Fig 4). If the panel is pinned, it will display contents that are related to a specific object, independently on which object is selected.

Floating modifier or a constraint panels are a special case. The object may have more or less than one instance of the specific modifier/constraint. The solution to this is to give modifier floating panels extra controls. Such as a "+" button for objects that do not have that specific modifier. This button will add the modifier to the end of the modifier stack of the selected object. If an object that cannot have modifiers is selected (Such as the camera, for example) the floating modifier panel would display nothing or gets grayed out. If there are more than one instance of that modifier in the modifier stack, two extra icon buttons can be introduced - "Up" and "Down". These buttons will jump from one instance of the modifier to another in the modifier stack.

If two 3D view editors are displayed and they both have similar floating panels, for example, "View" or "Display", these floating panels would only control settings for the 3D view they are in. The user cannot control settings of one 3D view by changing values in another.

Same context sensitivity rules apply to all editors that has floating panels.

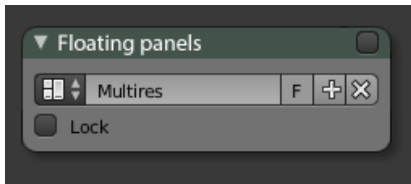
New panel in the "N" shelf.

The panel is visible in any mode, but is context sensitive to the mode blender is in. It's called "Floating panels". This panel gives the user an option to select one of multiple premade floating panel layout presets for the current mode or create new ones.

Simple Example of use: The user makes two layout presets of floating panels for sculpt mode, 3D view, and calls them "Multires" and "DynTop". The first layout contains the "Brush" panel and the "Multiresolution" modifier panel. The other contains the "Brush" panel and the "Topology" panel. The user may want to use one preset at some times, and the other at other times. Having presets will keep the user from having to re-add wanted floating panels and remove unwanted floating panels one by one every time he changes his workflow. "Multires" and "DynTop" only appear as options in the sculpt mode, the mode they were created in and relevant to.

Separate similar editors (for example, more than one 3D view) may have different floating panel layout presets selected in their "Floating panels" panel. If separate similar editors have the same floating panel layout preset selected, they display the same floating panels and changes to the panels on one editor would also affect the others.

Fig 10: The "Floating panels" panel from the "N" shelf.



The checkbox on the header would toggle the display of floating panels on the current editor. The default keyboard shortcut for this checkbox is "T". Such a keyboard shortcut would be comfortable to blender users, because so far it was used to show/hide the "T" shelf. An identical set of panels as the "T" shelf can be constructed out of floating panels, which is why having the original "T" shelf is redundant. Or alternatively, the classic "N" shelf can be reconstructed using the floating panels, and the current "N" shelf can display the "Tool shelf" tab.

The "Lock" checkbox locks the floating toolbars in place, but other functions, such as contraction may still work.

The panel is not an exclusion of the rule, that any panel from the "N" shelf can be dragged out to become a floating panel.

The alternative to drag and drop

If the user remembers the name, but not where to find the panel he's thinking of, or has any other reason, he could press "Space", type the name of the panel into the search and call the panel in as a floating panel. The search results should be context sensitive. Search should display an icon that an item is a panel, not a tool, because some panels have names that may be same as tools, "Transform" for example. This feature would allow the user to reach any possible wanted panel without ever leaving the 3D view or looking through the "N" shelf.

The same feature would also work in other editors.

For this to be most effective, Blender should avoid having different panels with same labels. Such as "Transform" that contains tools and "Transform" that contains coordinate number boxes.

More on the "Custom" tab in the "N" shelf

The whole purpose of this tab is for users who do not wish to use floating panels to still get the benefits they provide - The ability for the user to put the few panels that are required for the workflow in an easy to reach place. The tab would contain a way to select and create presets, a search bar and imported panels. The "Custom" tab presets are separate from floating panel presets.

Contents to the "Custom" tab could be added by either drag and drop from the properties editor or floating panels, or by right clicking on panels within the "3D view settings" or "Tool shelf" tabs or floating panels and selecting "Add to custom".

The "Custom" tab may contain duplicates of a panel if the user desires.

The same context sensitivity rules apply to the "Custom" tab as floating panels.

The "Last operator" panel

It was long debated on what the placement for the "Last Operator" panel should be. The proposition is for the answer to be - anywhere the user wants it to be.

By default, the last operator could be placed in the "Custom" tab and/or as a floating panel on the 3D view. "F6" would be the easiest way to call the panel in as a fully functional floating panel in any editor that supports floating panels.

Floating panel management

The floating panels could be moved around by clicking and dragging their headers, unless they are locked.

Floating panels could be scaled horizontally. Vertical scaling is not allowed. That is the same behavior as if the panels in menus.

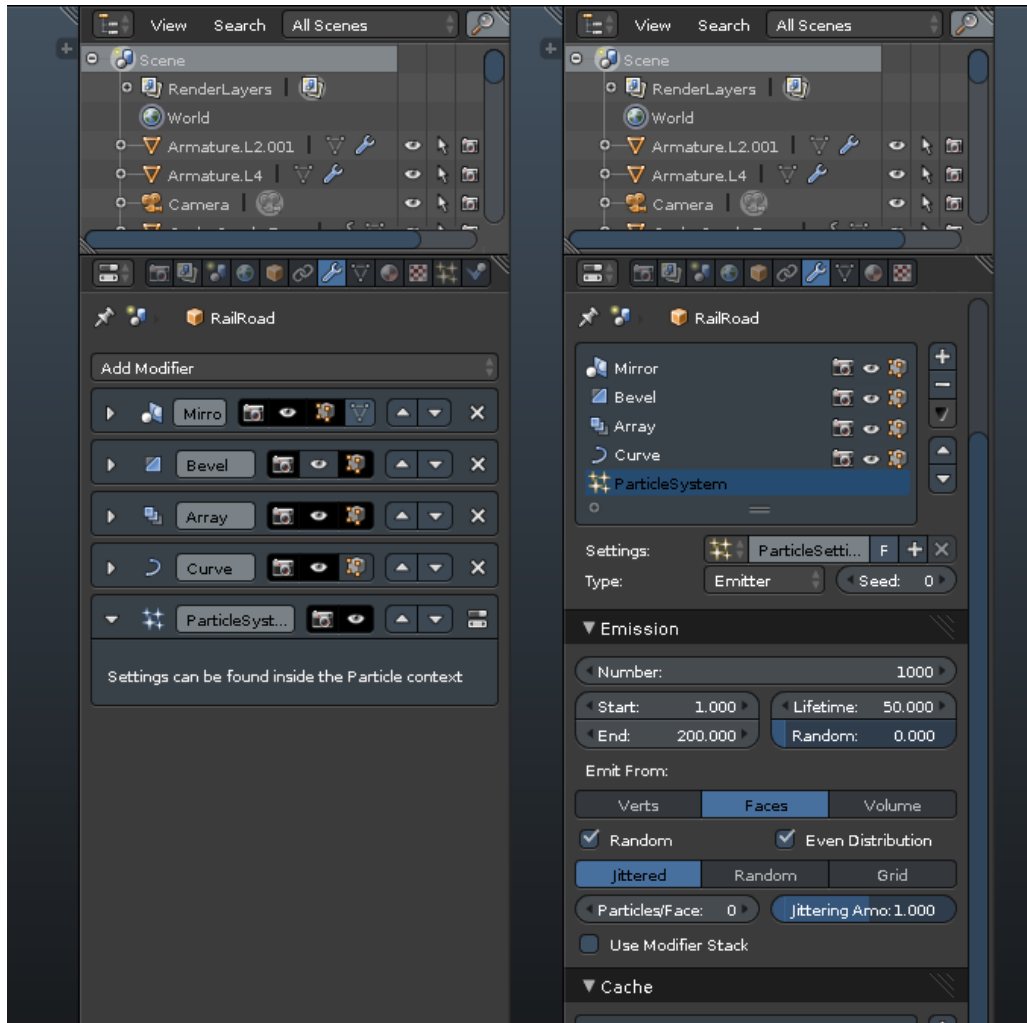
To remove a panel, the user would have to right click it and select "Close". There could also be an "x" on the panel header, unless floating panels are locked.

Floating panels snap to the edges of each other and the editor. If panels are snapped to the edges of the editor or one to another, they remain snapped, whether the editor is scaled up or down. Scaling the editor would keep relative floating panel positions.

Floating panels can be grouped into columns. Moving a panel right under another, would snap them together. This will make the two panels grouped into a column. In fig 4, the "Brush" panel and the "Shading" panel are grouped into a column. When multiple panels are grouped in a column, moving one would also move the others. To detach panels, the user has to make sure floating panels are unlocked, then right click on the header of the panel he wants to detach and select "detach" or drag it off with the mouse when holding "Ctrl". Collapsing single panels in a column group is allowed. Ctrl+L_Click expands one and collapses other panels in a column group. Same functionality as in menus.

If modifiers and constraints ever get designed as a list boxes (Fig 11), there could be a special case for Modifier and Constraint stack list box floating panels. Selecting a modifier or a constraint would automatically generate a column group under it for the contents of the selected modifier or constraint. Even if it is composed out of multiple panels, such as there are on particle systems or physics simulations.

Fig 11: Redesign of the modifier or constraint stack



Floating Panels are not allowed to be moved outside the boundaries of the editor they are in. That is, the user is not allowed put the panels in a way that any content is outside the view, unless the panel is taller or wider than the editor.

If a panel or a panel column group is taller than the editor, a scrollbar is added. Also, the user can scroll with the mouse wheel.

If most of the panel is on the lower half of the editor, clicking the header would collapse the panel down. If most of the panel is on the higher half of the editor, clicking the header would collapse the panel up.

"Tab" panel and other unique gizmos

The "Tab" panel is a unique panel that only works as a floating panel. It can be called in by searching or right clicking on a floating panel or a column group and selecting "Add Tab header" . The "Tab" panel only contains a header with horizontal tab buttons. The purpose of the "Tab" panel is to add a "mother" header to floating panels or a column group. Selecting other tabs would show different versions of the column group, that is, by hiding panels that are column grouped to the "Tab" panel and showing, or hide all of them at once.

The user can add more tabs by right clicking on the "Tab" panel and selecting "Add tab". The user can remove tabs by right clicking on the "Tab" panel and selecting "Remove tab". This would remove the currently active tab with all of the panels that are grouped to it.

The "Tab" panel can only be at the top or the bottom of a column group. A column group can have only one "Tab" panel. It is the only panel that doesn't move, when the user is scrolling on column group or a floating panel that doesn't fit the editor vertically.

Fig 12: The unique "Tab" panel. The header is larger than what the other panels have, because this panel contains only the header and buttons have to fit on it.

Example of use: Object mode. The user adds the "Tab" floating panel, then column groups the panels "Transform tools" and "Mesh tools" under it. The user adds another tab. When the other tab is active, the panels "Transform tools" and "Mesh tools" get hidden. The user then groups the "Add Meshes" panel to the second tab. At this point, the user has recreated two of the tabs of the Blender 2.7 version of the "T" shelf. This way, the user can recreate the whole "T" shelf place it on the left and lock floating panels. (See Fig 15)

Fig 13: Multiresolution modifier is grouped to the first tab, Dynamic topology is grouped to the second tab. One layout preset, two possible workflows, no wasted space.



Fig 14: Current (Blender 2.7) design of the "T" shelf.

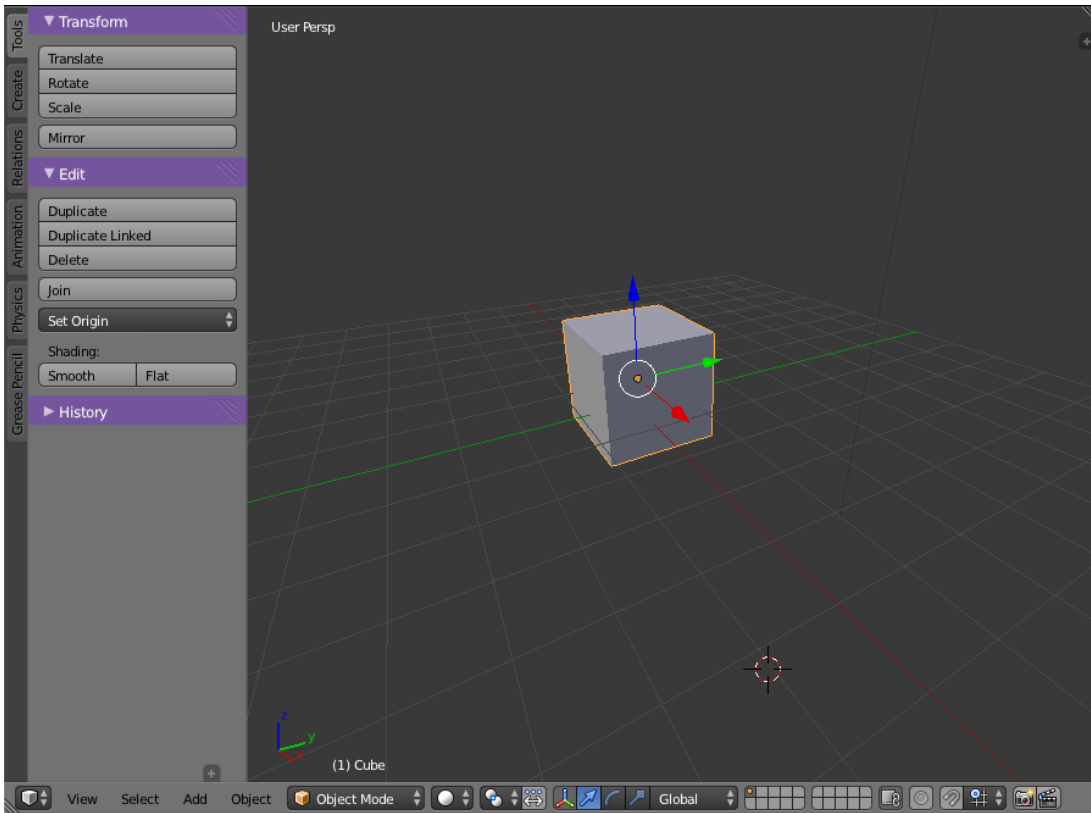
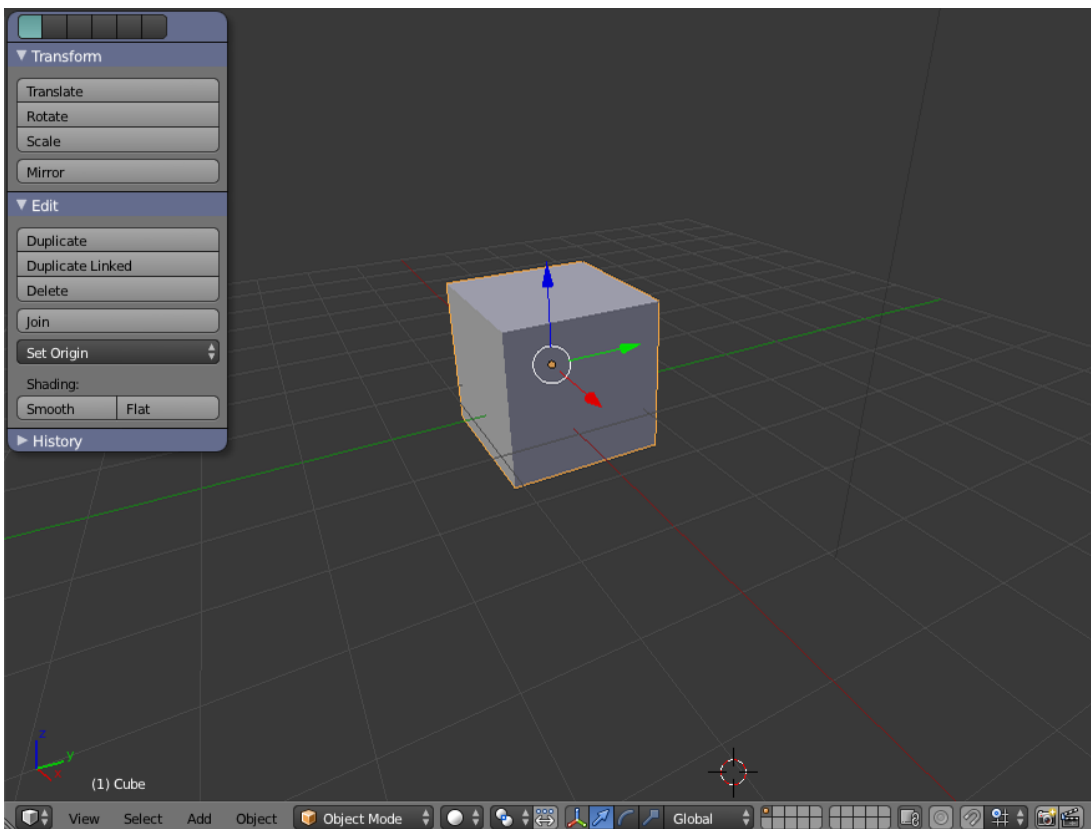
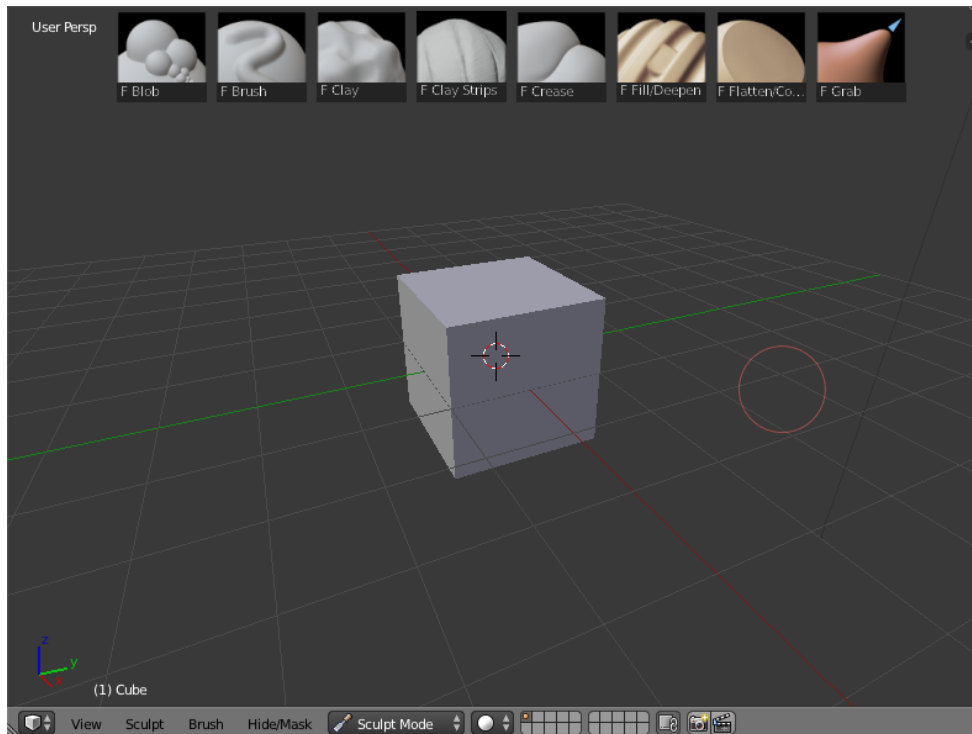


Fig 15: The "T" shelf reconstructed using floating panels with a "Tab" panel. Note - floating panels can be toggled on/off using the "T" keyboard shortcut, so it feels natural.



Apart from floating panels, there could be other optional floating gizmos, such as buttons to switch modes, a material row to select materials, a brush row, or anything similar. Floating gizmos should be defined using python, so python coders could have the ability to write floating gizmo add-ons.

Fig 16: A floating sculpt tool row implemented by a python coder:



Why let the user put panels in the middle of the editor? Why not just edges?

Just because most users would want their panels at the edges of the editor, doesn't mean there aren't any users that don't want to place their panels anywhere else. People have different workflows and different opinions on how the GUI should be laid out. That's why Users should have as much freedom as possible. Users are not forced to use all freedoms the UI provides.

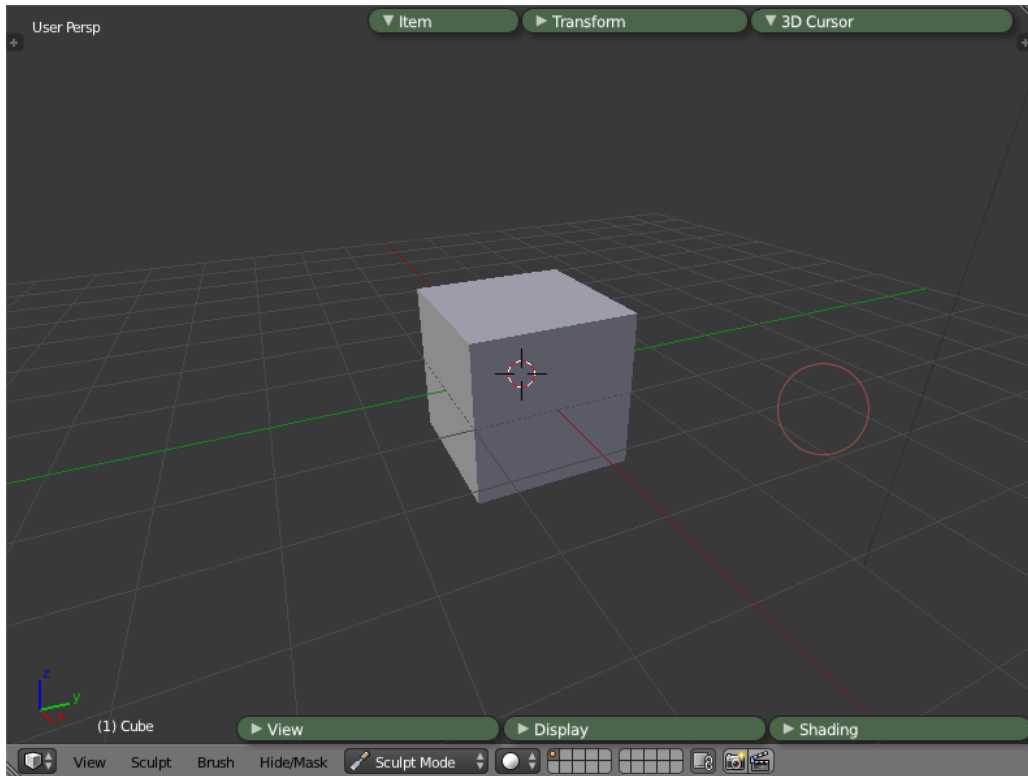
What if the user wants to make two or more panel columns next to each other?

What if the user wants collapsed rows of panels at the top and/or the bottom of the 3D view?

Such controls are not possible with the "T" and "N" shelf. Floating panels would have obvious controls for these actions and the best visual representation when doing them.

If floating panels were designed in a way, that they would be snap locked to the edges of the editor, even if the user doesn't want to leave the panels anywhere else but the edges, panels would feel awkward to reposition.

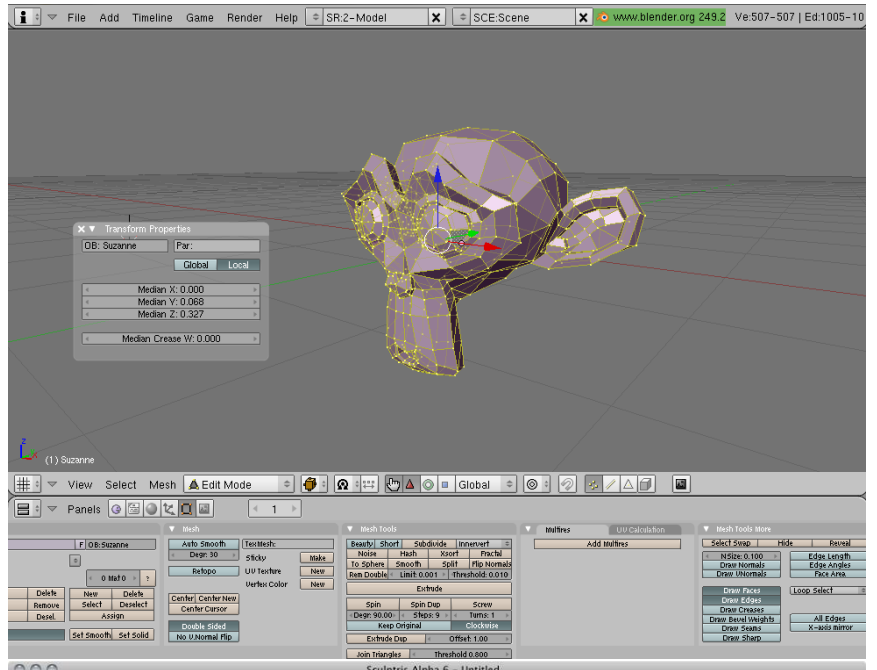
Fig 17: Rows of panels that can be achieved with floating panels, but not shelves. The top panels would expand down, the bottom panels would expand up, like it is described in the "Floating panel management" section.



Inspirations:

The old floating panel design in Blender 2.49. The UI design in 2.49 was terrible overall, but it had floating panels. Even though there were not so many panels that could float, the blender user was not forced to see all of these panels at once.

Fig 18: Blender 2.49 floating panels.



Sculptris. Sculptris is a program dedicated only to sculpting, therefore only sculpting settings and tools are displayed.

Everything is easy to reach and not hidden under menus. The 3D view is full screen. Blender users should have an ability to clean up their view to such a degree when they are just sculpting.

Fig 19: Sculptris



Some misconceptions and questions

These misconceptions and questions are gathered from the discussion in the Blenderartists.org forums.

"Floating panels tend to stick at the wrong positions. So you are busy with moving them around. Moving them around is against muscle memory. A tool should always stay available at the same position. Dock-able panels are the better choice in most cases therefore."

Proposed floating panels would need as much moving out of the way, as the current "T" and "N" shelves, when the "region overlap" setting is on.

"Why couldn't you use background images with sculpting?"

You could. The "N" shelf would still be there, fully functional, and it would still contain the panel for background images. The user could also drag and drop the background images panel on the 3D view area to make it floating if he wants.

"Floating panels that don't have a way to dock them are horrible (look at gimp prior to single window mode, or lightworks, I hate those designs"

Proposed floating panels wouldn't feel much different than the beloved design of Sculpttris. The differences are: The user is allowed to move the buttons around if he wants to, endless amount of customizability.

"un-doubling the double info from the properties and the N panel is a logic step, in a way, but if anything, it should be removed from the properties editor, as the info is still more useful in the N panel, then any other way as it's a 3D view setting."

By this logic, all panels in the properties editor should move to the "N" shelf.

"Merging T and N shelf is silly as they don't provide the same tools."

That is why it is proposed them to be in different tabs. With a subcategory of tabs for the tool shelf.

"Don't get the "tab" panel, maybe should be re-thought."

It is similar to the header of the properties editor, just that it's floating. Making things similar, means making it consistent.

"Do be careful about hiding stuff only you feel is superfluous."

The user is free to lay out his workplace any way he wishes.

"let's keep that non-overlapping philosophy in mind, because it's precious."

People have different tastes. In the proposal, the user is not forced to use floating panels. The user should not be forced to use shelves.

Conclusion

The author of this document proposes that the addition of floating panels provide far more control over the layout in Blender for the user. The improvements would allow the user to setup his layout for his specific workflow without wasting any space and letting the user reach wanted settings easier.

Also, the changes proposed would make the "N" shelf better organized. That is, the 3D view specific settings wouldn't be in the same menu as scene or object sensitive settings. Such grouping is inconsistent, but important for the current design, because those settings are important to reach from the 3D view. The proposal solves this inconsistency and still lets all of these settings be reached in the 3D view.

Current design Pros:

-

Current design Cons:

1. Inconsistencies in the "N" shelf.
2. The user is forced to see settings he doesn't need for current workflow.
3. The main work area, for example the 3D view, gets very small in some scenarios when the user setups his layout to see all the controls he needs for his workflow.

Proposed design Pros:

1. Much more customizability.
2. Blender allows the user a layout with no wasted space for his specific workflow
3. Consistency in the "N" shelf.
4. Blender allows the user to go full screen on the 3D view without losing accessibility to settings in the properties editor.
5. The user can setup his layout for all wanted panels to be at hand. No more need to jump through menus each time a specific panel is required for work.

Proposed design cons:

1. There might be some trouble until blender can edit multiple objects at once, as it was proposed in the original Blender 2.5 GUI design paper. More specifically - One of the rare panels that allow control over a multiple selections is the "Transform" panel in Edit mode, which would have no main placement that would make it consistent. This panel would either have to be always called in as a floating or kept in the "Custom" tab.

2. The fact that everyone's GUI would become customized for their own workflows might confuse beginners.

The author may be contacted for further discussion at pauliusmscichauskas@gmail.com or via private message in blenderartists.org forums.