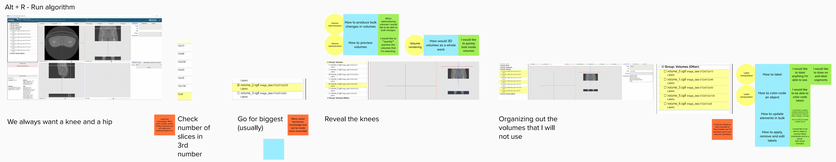
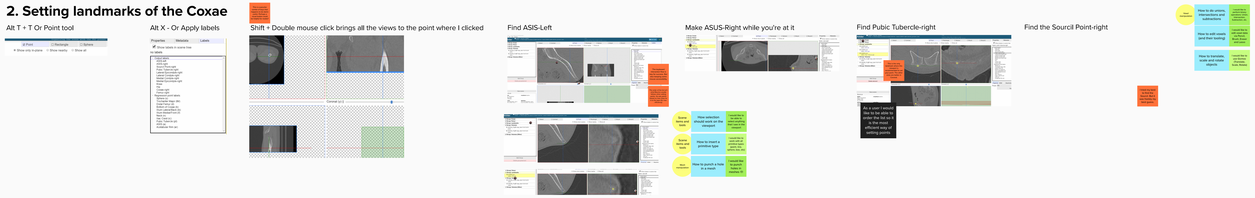
Arbiter experience analysis

1.- Classification / Automated Triage

Alt + R - Run algorithm

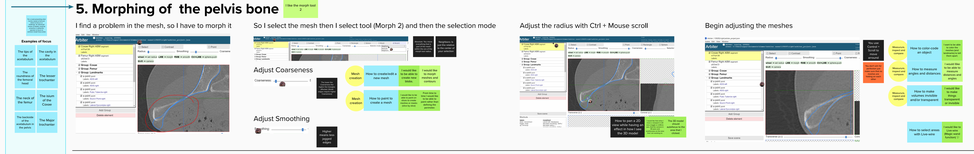


2. Setting landmarks of the Coxae



3. Setting landmarks of the Knees

5. Morphing of the pelvis bone



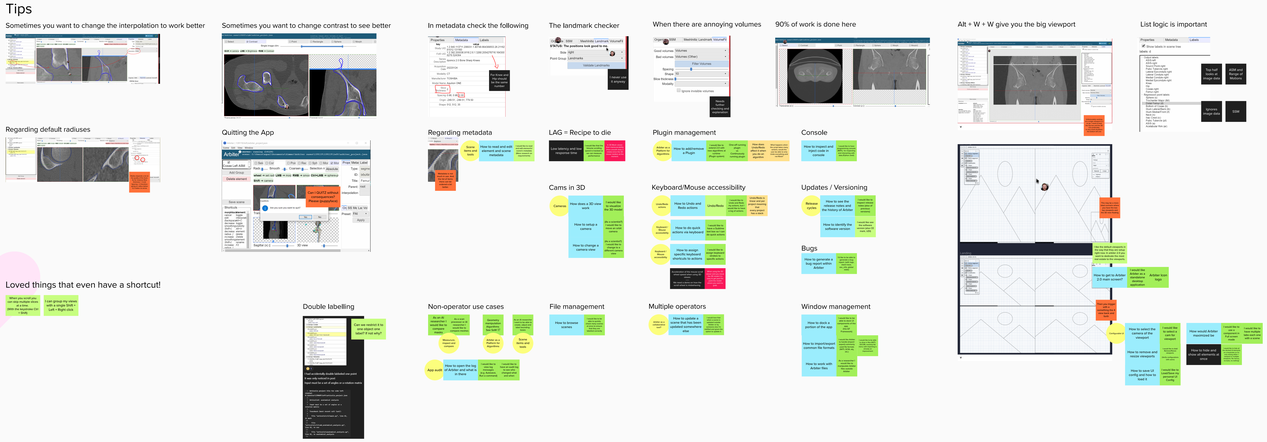
6. Morphing of the rim of the cup

Same process

7. Morphing of the rim of the femur

Same process

Tips



Summary

Per step of the experience

1.- Classification / Automated Triage

Assuming that landmarks will not be automatically placed. The best way to go in this step is to deep dive in the tree.

- Make it easier to apply/change labels to the landmarks
- Make it easier to group/ungroup/delete/hide items
- Make it possible to color code it (perhaps automatically)
- Consolidate the controls in both the tree, metadata, labels and properties panels into one.
- Make it possible to place landmarks + add a label to it entirely with the keyboard
- Have a basket of unused items. No need to delete them, they can all stay there nicely tucked away.
- Make it possible to simply drag/drop items within the tree in the same fashion as layers.

2. Setting landmarks of the Coxae

- Redesign the toolbar in a way that occupies less real-estate so more can be left for the viewports where the operator will work anyway.
- Make it easier to adjust the viewports to your needs.
- Reduce the interface's panels footprint. The most space left for the viewports the better.
- The 3D viewport does not necessarily need a full window of its own.
- Allow for viewport default configurations that make sense to the operator.
- Allow to save your workspace.
- Bring the shortcuts when they are contextually appropriate rather than having them always in the bottom left and top bar.
- For the noobs, Arbiter can do wonders at creating an expectation on where does certain landmark is.
- For most landmark placement it is the Coronal view that should be used. Create flow that supports that bias.

3. Setting landmarks of the Knees

- Autosave (and if possible Autoversion)
- When you close a project be forgiving. Allow the user to open it back and begin working where it left.
- Don't ask whether the user is sure to close Arbiter. Just give him/her the button to save the project.

4. Begin segmentation

- Locate the progress of the Command initialization via a bar close to the button where you clicked
- Allow for multiple instances of Arbiter so simultaneous work can be done while the computer is processing something.

5. Morphing of the pelvis bone

- Common areas of focus can be highlighted to bias the user to look in there at all times. (see Examples of focus)
- Coarsens, Smoothness, Morph 1, Morph 2 etc. Can benefit a lot from an explanation in a tooltip (sometimes an animated gif is all it takes to explain what is what)
- Same for Absolute and Neighbors. There is easy knowledge that can be conveyed on hover for instance. This in turn can ease the learning curve.
- Morphing something can greatly benefit from a large 3D projection at times so you can see the chain effect that is caused by the morph

6, 7. Morphing X

- Arbiter makes it easy to keep morphing until you have the perfect shape. However you only need to focus on a few areas and to get them right in order to have a good enough report. It would help to create a flow that encourages that behavior on the user. At the moment it is really easy to spend a ton of time just morphing and morphing. How good it good enough?

Tips

- The explanation of what interpolation does, and how the elements affect each other is too far apart.
- Metadata can be better arranged to represent the values that matter the most. Perhaps it can be accessible in a way that does not require so much real-estate.
- Landmark checker can either be integrated into the process or scrapped entirely. It is of no use in its current form. Furthermore you need to activate it. That seems odd.
- The volume fitter is one of those tools that are only used in very specific instances. It can easily be tucked away to give space to the more relevant content in the interface.
- Shortcuts are pecular it is worth bulding a map of them and see if there are any adjustments that need to be made (more used shortcuts should be shorter. Less used shortcuts don't need that. There may be missing shortcuts. There may be useless shortcuts too.)
- The list serves as both a applicator of labels and a checklist. This is really useful (2 functions in one) the elements in the list can be rearranged for a better default flow.
- The list can also benefit from custom arrangements. When you are doing 20 projects in a day, you will develop an idea of how to work best.
- The default size of radious has a relationship with the Slice thickness and Spacing. There is a curve of ideal settings for the default size based on those two factors.
- In metadata the user compares the number of the knee and the hip. It may be good to make thise numbers easily comaprable.
- Any lag in its use will kill the app usange. People just don't have the patience to wait even a second to their next click and drag.
- A console to run scripts and a log would be very helpful for power users
- There is a need for a space for the release history of Arbiter.
- Likewise aspace should be dedicated for Arbiter's version so operators can assist each other better.
- THere needs to be a fast and efficient bug reporting system that captures the log of the user, what they are seeing, their mood and an (optional) description box.
- Arbiter should never lose its keyboard accesibility. It's crucial. It only needs to comunicate it better.
- Arbiter should never lose the ability to open multiple instances of itself.