ScrollViews

 ScrollViews were designed to scroll documents.   Documents by their very nature are unbounded and will always have the potential of being larger than the largest window on the largest screen ever made, and most graphic user interfaces support some sort of scrolling behavior so that a large document can be viewed in a small screen space.

 However, because of the general and powerful heirarchy of the AppKit View architecture, Controls such as Buttons and Sliders can certainly work inside of a ScrollView.   This was a design goal, not for user interface reasons, but to prove the generality and flexibility of the design.

 However, there is more to this issue than showing off the power of our View architecture.   As far as our visual metaphor goes, three-dimensional objects are used to imply concreteness.   To allow them to scroll around inside a ScrollView diminishes the power of the effect we are trying to achieve.   Controls are usually put on panels; a well-designed panel is compact and uncluttered.   Putting controls in a ScrollView is one way to allow panels to become large and monolithic.   Packing controls in ScrollViews shouldn't substitute for simpler panels of well-organized, easily-available controls.   In general, using ScrollViews (resent a large panel in a small area is a bad idea.   All controls for a panel should be visible when they can be effective. Inspector Panels get around this by having modes.   Seeing controls in ScrollViews should be a flag to review the UI choices made.   Consider substituting an Inspector-like interface, with context sensitive behavior, perhaps at the cost of a mode.