CMSI 370-01

Interaction Design

Fall 2012

Assignment 1120 Feedback

Chase Blokker

1e — No visible demonstration of affordance knowledge seen here; I was able to figure out the use of the Shift and Alt keys because I know that this is what I suggested. Ideally, some sort of visible sign that these were indeed your chosen controls should be there; even better if things were dynamic (i.e., something different appears when the Shift or Alt keys get held down). Not something in the 3D code or rendering, of course, because you have not learned about that yet. However, it remains eminently possible to do something the web page itself.

I will mark this as an **O** to indicate that this knowledge is not seen. It does not mean that you have to redo things in this assignment, but you should make sure to demonstrate it in the remaining ones. (**O**)

3a — You definitely advanced your user interface knowledge quite well with this assignment, this time in the area of how direct manipulation interfaces are constructed. (**+**)

3b — Your work shows further advancement in event-driven programming, specifically with regard to handling a rapid succession of events, typical of the direct manipulation interaction style. (**+**)

3c — Your understanding of MVC is clearly demonstrated in this assignment. (**+**)

3d — You are fairly successful in breaking down high-level user actions into lower-level events. There is one hiccup in your approach, in that the initial states that you record when the mouse goes down are no longer valid later in the sequence, if the user changes the type of manipulation midway. For example, start a pan (i.e., Shift key down). Move the tetrahedron to one side of the area. Then, let go of the Shift key. See the "jerk" effect? You might need to do this a few times. Same happens when toggling the Alt key while keeping the mouse down.

The bottom line here is that your state management needs a little more refinement while the mouse is in motion. Essentially, you need to "reset" your initial state when a change in modifier keys takes place. Other than that, things came out pretty nicely, I think. (**|**)

4a — Your code largely works, except for the aforementioned mid-drag behavior. There are also a few notes included in your code inline. (**|**)

4b — Your code demonstrates proper separation of concerns. (**+**)

4c — Your code is nicely readable and easy to understand. Comments provide appropriate annotations, and spacing is generally OK (though not perfect). (**+**)

4d — Your work generally shows good resource use and knowledge-seeking. (**+**)

4e — Your commit pattern shows fair phasing and messages. I can see all of this happening in three commits, but I think it would have been more scrupulous to break the work up into even finer chunks (e.g., rotate first, then scale; or something like that). (**|**)

4f — Submitted on time. (**+**)