FloatOmeter: User-Friendly Input of Floating-Point Numbers in Virtual Environments

Matthias Kreiser Jürgen P. Schulze Andrew S. Forsberg

Department of Computer Science, Brown University, Providence, RI 02912 USA
{kreiser,schulze,asf}@cs.brown.edu

Specifying floating-point numbers in immersive virtual environments (VEs) like CAVEs is not as straightforward as at the desktop because VEs usually lack keyboards and other obvious approaches like speech recognition are not generally robust enough yet. The most commonly used way to input floating-point numbers is by sliders. Sliders are widgets with a button the user can move along an axis to control a number; their main disadvantage is that they have lower and upper bounds. They reach their limits when the range of the number to be entered is not previously known, or when parts of the range need much finer control than others (e.g., to enter the distances of the planets from the sun). Newman [1968] proposed a light-pen-based method to change numbers continuously (similar to the iPod's touch wheel) and additionally to control the granularity of the changes. His approach works for numbers in a longer range than sliders, but this range is still limited.

Another established way to enter floating-point numbers is with a virtual numeric keypad: Chen et al. [2004] found that their virtual numeric keypad (V-Key) worked better than a slider-based interface. However, changing an existing number is cumbersome and requires placing a cursor next to the digit to be changed and typing in the new digit, or entering the entire number again.

General number input can be done just like text input. Bowman et al. [2002] compare four methods for text input in a VE: pen and tablet, pinch glove, chord keyboard, and speech. However, None of these techniques work well when a previously entered number just needs a small change.

The FloatOmeter

We have developed a method both to enter and change the individual digits of floating-point numbers. It works in virtual environments without using a physical or virtual keyboard. To interact with the number we use a virtual laser pointer: a ray is cast from a hand-held six degrees-of-freedom (6DOF) input device. We use a wand, but any 6DOF device with a button would work. Each digit of the floating-point number is displayed in a little widget box that highlights when the ray intersects it (see Figure 1). When a digit is highlighted, the user can change it by depressing the left wand button while rotating the wrist. Rotations to the right increase the digit's value, rotations to the left decrease it. When the digit is zero and the user rotates to the left, it is set to nine and the next higher valued digit is decreased by one. This process is recursive up to the highest valued digit. For instance, if the number is 2005.731 and the user decreases the "tens-place" digit by one, the number becomes 1995.731. This method reminds us of mechanical odometers, hence the name "FloatOmeter" (floating-point odometer).

In order to keep the digits panel as short as possible, our algorithm never displays leading or trailing zeros. If bigger numbers or more precision after the decimal point are required, the user can click on the arrows left or right of the number to display an additional digit with a default value of zero. A digit is automatically added if the most significant digit is increased above nine. A practical problem we ran into when removing trailing zeros was that, due to rounding errors in the storage of floating-point numbers, they sometimes have more non-zero digits after the decimal point than the user had entered. To solve this problem we store the number of displayed digits after the decimal point, and next time we display the number we round it off to that number of digits. To allow the input of negative numbers we display a little box with a plus or minus sign between the left arrow and the most significant digit; clicking on this box toggles the sign.

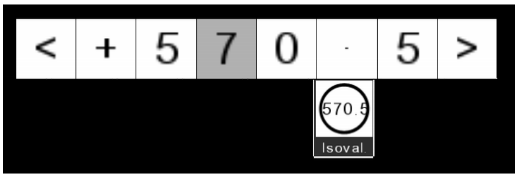

Figure 1: Dial knob (bottom) with FloatOmeter (top).

Because the FloatOmeter takes up a relatively large screen area (but less than a virtual keyboard), we use it as an extension of virtual dial knobs. Many parameters can be set with dials alone and do not need FloatOmeters. Our dials display a value that can be changed by intersecting the ray with the dial and holding the wand button while turning the wrist (just as changing individual digits of the FloatOmeter). If more control over the dialed value is needed, one can click the left wand button while the pointer is on the dial and move the laser pointer up. Once it reaches the the top sixth of the dial widget, a FloatOmeter shows up above it. It is always positioned so that the decimal point is directly above the dial, so that just by looking at the number of boxes on either side of the decimal point the user can estimate how big and how precise the number is. Once the value has been set, the FloatOmeter is closed by clicking anywhere outside of it.

A value set with the FloatOmeter can also be changed with the dial. The amount of change per ten-degree rotation is specified by clicking on a digit in the FloatOmeter with the right wand button to mark it as the step size, indicated by a gray background. In Figure 1 digit "7" is marked, so that subsequent dial rotations to the right change the value to 580.5, 590.5, etc.

In the future we want to run a user study to compare the FloatOmeter to other methods of number input in VEs.

References

BOWMAN, D., RHOTON, J., AND PINHO, M. 2002. *Text Input Techniques for Immersive Virtual Environments: An Empirical Comparison*. Proceedings of Human Factors and Ergonomics Society Annual Meeting, Baltimore, pp. 2154–2158.

CHEN, J., BOWMAN, D., WINGRAVE, C., AND LUCAS, J. 2004. *Designing Explicit Numeric Input Interfaces for Immersive Virtual Environments*. Technical Report, Virginia Tech Department of Computer Science, TR-04-13.

NEWMAN, W. 1968. *A Graphical Technique for Numerical Input*. Computer Journal, 11(1), pp. 63–64.