

Interactivity

a Saturday class

Morning

Interactivity, the essence of all software

Interactivity is the source of value of all software. Web browsers, games, email, and spreadsheets all depend upon their interactivity for their delivered value. Conversation gives us the most illuminating example of interaction. Interactivity is an cyclic process in which two active agents alternately listen, think, and speak. Speaking is the easiest step, for we already know a great deal about how to express ideas in various media. The problem, then, is to restrain our tendency to concentrate on what we already do well. Thinking is much more difficult; it requires algorithms. The value we deliver to our customers arises from the algorithms we provide them with. But most designers have difficulty creating good algorithms. In order to listen well, you must permit the user to speak well. This is usually accomplished by means of good user interface design. However, good listening is more than just providing menus and pushbuttons. The software must be designed from the ground up to give the user maximum opportunity to express their desires -- and to clearly block what the software cannot do.

Break

The interactive loop

The interactive process presumes repetition. We expect the user to go round and round with the software. This process is usually convergent. In a game, the player converges to excellent play; in a web browser, the user converges to the information they seek; in a word processor, the user converges to the ideal form of the document they envision.

Common architectures for interactive software

One way of thinking about the interactive process is by preparing directed graphs of the interaction. Most applications graphed this way show lots of independent loops. Some educational and entertainment software uses scraggly trees or bushy trees. A variety of other architectures are explained, and their faults revealed.

Afternoon

The great polarity: thing versus process

Fundamental to the universe is the polarity between object and process. This polarity appears everywhere around us, and is of profound importance to the design of interactive software. The designer must concentrate on process, not object. This kind of thinking is alien to normal human thinking -- which is why good designers are so rare!

Break

Designing algorithms

Algorithms can be designed in a six step process: 1) express the algorithm as a causal relationship; 2) declare the output variable for that relationship; 3) determine the rele-

vant input variables; 4) decide the desired shape of the graph for the relationship; 5) apply the appropriate functions along with coefficients; 6) adjust coefficients to taste.

Metaphorical thinking applied to algorithm creation

Metaphors underlie most human creativity. This lecture shows how to use metaphorical thinking in the creation of new algorithms.